Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Unsupervised Learning of Human-Object Interactions with Neural Network Self-Organization

Dissertation

Dissertation submitted to the University of Hamburg
with the aim of achieving a doctoral degree at the
Faculty of Mathematics, Informatics and Natural Sciences,
Department of Informatics.

Luiza Mici

Hamburg 2018

Submitted:
September 12, 2018

Day of oral defence:
November 27, 2018

The following evaluators recommend the admission of the dissertation:

Prof. Dr. Stefan Wermter (advisor)
Department of Informatics,
Universität Hamburg, Germany

Prof. Dr. Loo Chu Kiong (reviewer)
Department of Artificial Intelligence,
University of Malaya, Malaysia

Prof. Dr. Frank Steinicke (chair)
Department of Informatics,
Universität Hamburg, Germany

Prof. Dr.-Ing. Wolfgang Menzel (deputy chair)
Department of Informatics,
Universität Hamburg, Germany

To Robert, Çezarina, Redi and Luigi

Abstract

Understanding human actions is crucial for establishing an effective interaction between an assistive system and humans in the real world. Humans are able to understand others' behavior by interpreting body movements and finding relevant contextual cues in their surroundings. Such ability is supported by a highly developed visual system that creates a coherent perceptual experience by effortlessly integrating different sources of information. Furthermore, the human brain is continuously projected into the future, hence anticipating the development and the intentions of the observed actions. Artificial systems, however, are far from a human-like performance in these tasks. The reliable recognition and anticipation of human actions from multiple visual cues still remain an open challenge.

In this thesis, we focus on human daily activities that involve interactions with objects. We aim at designing artificial learning systems for the recognition and prediction of human-object interactions while considering interdisciplinary aspects of neuroscience and human psychology for the two perception tasks. We apply hierarchical arrangements of self-organizing neural networks that resemble the cortical processing of action features with an increasing complexity of representation. We introduce a novel architecture that can segment and recognize the manipulated objects from a scene and map them to their action possibilities in an unsupervised manner. The spatiotemporal representations obtained through the self-organizing learning are then associated with symbolic labels for the classifications of the human-object interactions. We evaluate our model with two different corpora containing fine-grained human daily activities in home-like scenarios and demonstrate that our model is competitive with respect to supervised state-of-the-art approaches.

We address human action anticipation by focusing on both *what* will happen next and *how* the action will be performed. First, we present and discuss a novel hierarchical self-organizing architecture for the incremental learning and prediction of human motion patterns. Our experimental results demonstrate that self-organization can account for robust body motion prediction, yielding high performance during the online adaptation also in the presence of missing data samples. Then, we introduce a temporal association mechanism for storing goal-oriented action sequences of arbitrary lengths into our model. We demonstrate that both short-term and long-term temporal dependencies of the human actions can be learned with the same underlying neural mechanism, thereby allowing for the anticipation of actions in a longer activity sequence. Finally, we present and analyze an approach with top-down feedback connectivity that uses the classi-

fication error to modulate the neural growth of a self-organizing hierarchy. We show how the interplay between feedforward and feedback connectivity generates an adequate number of prototype neurons and promotes the learning of compact representations of actions from the sensory input.

This thesis contributes to the field of visual recognition and prediction of human-object interactions with a set of novel models that take inspiration from biological mechanisms of action perception serving as a stepping-stone for different future research directions.

Zusammenfassung

Menschliche Aktionen zu verstehen ist äußerst wichtig, um ein effektives Zusammenspiel zwischen einem Hilfssystem und Menschen in Szenarien der realen Welt herzustellen. Menschen sind im Stande das Verhalten anderer zu verstehen, indem sie ihre Körpersprache interpretieren und relevante, kontextbezogene Hinweise in der Umgebung finden. Solche Fähigkeit wird von einem hochentwickelten visuellen System unterstützt, das durch eine mühelose Integration verschiedener Informationsquellen ein einheitliches Wahrnehmungserlebnis hervorbringt. Zudem projiziert das menschliche Gehirn kontinuierlich die Zukunft und sagt so die zukünftige Entwicklung und die Absichten der beobachteten Aktionen voraus. Künstliche Systeme sind jedoch noch weit davon entfernt Aufgaben so auszuführen, wie es einem Menschen möglich ist. Die zuverlässige Erfassung und Voraussage menschlichen Handelns mittels verschiedener visueller Hinweise bleibt noch immer eine Herausforderung.

In der vorliegende Arbeit fokussieren wir uns auf menschlich Aktivitäten des täglichen Lebens, die den Umgang mit Objekten beinhalten. Wir streben an, ein künstliches Lernsystem für die Erkennung und Prognose von Interaktionen zwischen Mensch und Objekt zu entwickeln, während wir interdisziplinäre Aspekte der zwei Wahrnehmungsaufgaben betrachten. Wir wenden hierarchische Anordnungen selbstorganisierender neuronaler Netze mit Schicht für Schicht höher werdender Darstellungskomplexität an, die der kortikalen Verarbeitung von Handlungseigenschaften gleichen. Wir stellen eine neuartige Architektur vor, die manipulierte Gegenstände in einer Szene identifizieren kann und mögliche Handlungen für diese in unüberwachter Weise bestimmt. Die räumlichen und temporalen Darstellungen, die durch das selbst organisierte Lernen erlangt werden, werden mit symbolischen Labeln für die Klassification von Mensch-Objekt-Interaktionen verbunden. Wir evaluieren unser Modell an zwei unterschiedlichen Korpora, welche detailgenaue Tagesaktivitäten in alltäglichen Szenarien häuslich darstellen, und demonstrieren damit die Wettbewerbsfähigkeit unseres Modells mit den aktuellen Stand der Forschung.

Zuerst präsentieren und diskutieren wir eine neuartige, selbst organisierte und hierarchische Architektur für das inkrementelle Lernen und für die Vorhersage menschlicher Bewegungsmuster. Unsere experimentellen Ergebnisse demonstrieren, dass Selbstorganisation eine robuste Erkennung von Bewegungsmustern gewährleistet, was auch im Falle von fehlenden Beispieldaten guten Ergebnissen während der Online-Anpassung führt. Außerdem erweitern wir unser Model mit einem Assoziationssmechanismus zum Speichern von zielorientierten Aktionssequenzen be-

liebiger Länge. Wir demonstrieren anhand von verschiedenen Experimenten, dass kurzfristige und langfristige Abhängigkeiten menschlicher Handlungen mit demselben zugrundeliegenden neuralem Mechanismus erlernt werden können, wodurch die Antizipation von Handlungen in Langzeitsequenzen ermöglicht wird. Schließlich präsentieren und analysieren wir eine top-down Feedbackverbindung die Klassifizierungsfehler nutzt, um das neuronale Wachstum einer selbstorganisierten Hierarchie zu modulieren. Wir zeigen, wie das Zusammenspiel zwischen Feedforward- und Feedback-Konnektivität eine ausreichende Anzahl an Prototyp-Neuronen generiert und das Erlernen der aktionsrelevanten Repräsentationen aus sensorischem Input begünstigt.

Diese Arbeit trägt zum Feld der visuellen Erkennung und Vorhersage von Mensch-Objekt Interaktionen bei. Eine Reihe von Modellen, die von biologischen Mechanismen der Handlungswahrnehmung inspiriert sind, dienen dabei als Sprungbrett für zukünftige Forschungsrichtungen.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The ongoing development of robotics and the increasing number of elderly individuals have led to an increasing interest in applying assistive technologies in order to improve the quality of care (Scassellati et al., 2012; Kachouie et al., 2014; Amirabdollahian et al., 2013; Cecchi et al., 2016). Despite the many physical forms in which assistive robots come, their main functionalities include providing assistance, serving, and interacting with humans. Human behavior understanding through visual perception is one of the most important steps towards a successful and safe execution of each task (Aggarwal and Ryoo, 2011; Sciutti et al., 2018). Current systems, e.g., Apple Siri or Amazon Alexa, need to be given explicit vocal commands to take action, but this is not sufficient when monitoring patients or elderly individuals at home. While providing care, an interactive robot system should know if the patient has performed essential daily activities fundamental to the patient's well-being, such as drinking water and taking medications. On the other hand, for an anticipatory planning of a reactive response, e.g., fetching a glass of water when the person wants to drink, an assistive robot should predict human motion and infer the intention of a human activity beforehand (Ryoo, 2011).

Real-world domestic scenes are quite cluttered and diverse. On the one hand, they pose several challenges to the visual analysis systems but on the other, they also provide relevant contextual information about the human activities. Thus, different relevant action components should be detected from the scene, processed, and later integrated, such as the present objects, the human body pose and motion, and the relationship between humans and objects during the interaction. The human brain is highly skilled in integrating multiple contextual information in just a brief glimpse of visual input due to the importance that goal inference

has for survival and social activities, e.g., detecting a threat (Henderson, 2003) or accessing the emotional status and the intentions of a person (Karg et al., 2013). Hence, the underlying biological mechanisms for action perception remain a source of inspiration for the development of artificial systems which address the recognition and anticipation of human activities (Giese and Rizzolatti, 2015). From the computational perspective, one question is of central concern: How to process and represent the visual stimuli arising from human-object interactions, given their spatiotemporal components with different levels of abstraction and heterogeneous temporal dynamics.

1.2 Problem Statement and Research Objectives

In this thesis, we aim to develop and analyze neural network learning architectures for the recognition and anticipation of human activities in domestic scenarios. In particular, we want to investigate self-organizing architectures, which have mainly focused on the recognition and generation of human gestures and full-body actions (Kawashima et al., 2009; Coleca et al., 2015; Parisi et al., 2015), for the modeling of high-level cognitive functions such as the recognition and prediction of human-object interactions and the learning of hierarchical representations of human activities. The application of self-organizing architectures is appealing due to their capability to adapt in an online manner while resembling neurophysiological processes such as input-driven self-organization (Merzenich et al., 1983; Blakemore and Cooper, 1970; Hirsch and Spinelli, 1970) and synaptic plasticity (Hebb, 1949). Moreover, these models can learn in an unsupervised manner, i.e., when the manual annotation of the input is not provided. Hence, their investigation is an important step towards building autonomous systems.

Our first research question is: how can relationships between objects and human motion patterns be learned in an unsupervised manner during the observation of human-object interactions? Detecting objects in cluttered scenes and recognizing human actions are two important, yet challenging research topics which have received a lot of attention from the computer vision, computational neuroscience and cognitive science community. From the computational perspective, however, it is not clear how to link architectures specialized in object recognition and biological motion recognition, e.g., how to match between classes of objects and hand/arm movements. To answer this question, we take inspiration from different findings on the neural mechanisms in the human brain including the hierarchical processing of spatiotemporal patterns with an increasing complexity and abstrac-

tion of representation (Hasson et al., 2008; Taylor et al., 2015) and the integration of the manipulated objects with the body motion for action understanding and goal inference (Beauchamp et al., 2002; Baldassano et al., 2017). Keeping in mind a real-world application of our methods, we aim at technologies that require the least computational effort and that can operate in real-time. Thus, we rely on the 3D body tracking frameworks provided by inexpensive depth sensors, e.g., Kinect cameras.

Plenty of studies address the modeling of the temporal structure of human activities for action classification (Aggarwal and Ryoo, 2011). Yet, it remains largely unknown how to apply existing recognition approaches for the anticipation of goals before actions have been fully executed (Ryoo, 2011). Human activity prediction is a relatively new topic which can have significant implications for the assistive systems operating in domestic scenarios, for instance, the planning of reactive responses for assisting individuals according to their needs (Koppula and Saxena, 2016). Building upon well-studied computational mechanisms, such as the Hebbian learning rule (Hebb, 1949), we develop and analyze a temporally sensitive neural architecture which is capable of storing and recalling sequences of arbitrary lengths in order to both classify and predict an ongoing and the upcoming human-object interaction respectively. For this reason, we revisit ideas about the encoding of temporal sequences through neural self-organization and lateral weighted Hebbian connections, which have been successfully applied for robot control (Barreto et al., 2003) and the prediction of human motion (Parisi et al., 2016b). Our objective is to investigate similar learning mechanisms for the learning and the recall of atomic actions that compose longer human activities. In particular, we look at how can the most frequently activated lateral connections be used to encode the temporal order of the perceived body motion patterns during human-object manipulation and to develop goal-oriented neural chain activations (Chersi et al., 2014).

The last research objective we pursue in this thesis is the investigation of a novel top-down modulation mechanism for the optimization of the neural growth of a hierarchical architecture comprising growing self-organizing networks. In the human visual system, top-down connections outnumber bottom-up connections and have a strong influence in the shaping of the visual features (Gilbert and Sigman, 2007). In a hierarchy of growing self-organizing networks with an increasing depth of the temporal context, we let the emergence of prototype atomic actions with a shorter temporal context be influenced by the upper layer, composed of prototype sequences of relatively long temporal contexts. This is useful, for instance, for the learning of hierarchical representations of human activities on two different

semantic levels: atomic actions and activities performed over a longer duration. In this case, the activity labels can provide constraints on the development of atomic action prototypes in order to have better recognition of the actions and vice versa.

1.3 Contribution to Knowledge

This thesis contributes to the knowledge on neural self-organization with a set of methods, experiments and detailed analyses of self-organizing models for the learning and prediction of human actions with a particular focus on human-object interactions. The neural architectures take inspiration from a set of biological findings, such as the hierarchical processing of body pose and motion cues and the encoding of goal-oriented actions through chain-like neural activations, but do not attempt to model the underlying neural mechanisms in detail. We propose and evaluate possible extensions of the growing self-organizing networks in order to account for a set of visual tasks such as the recognition and prediction of human-object interaction, the online human motion prediction and the learning of hierarchical representations of human activities. Taking advantage of the capability of Growing When Required (GWR) networks (Marsland et al., 2002) to learn input data streams incrementally, we provide a detailed analysis of a novel hierarchical architecture for the incremental learning and prediction of human movement sequences. Furthermore, we propose a top-down modulation mechanism that can be applied to a GWR-based hierarchical architecture in order to optimize the process of neurogenesis and the topological organization of each layer according to the classification task. Through our experimental results, we discuss the learning properties of the architectures based on sensory-driven topology preserving networks and their advantages for real-world applications, especially in the case of noisy or even missing sensory information.

1.4 Structure of the Thesis

The thesis is organized into eight chapters. In Chapter 2, we provide an overview on neural mechanisms for the action perception in the brain together with an introduction to current trends and state-of-art approaches in human activity recognition, human motion prediction, and human action anticipation.

In Chapter 3, we present some of the biological findings providing evidence for the topographic organization of the input in several areas of the brain. We describe models of neural network self-organization, which similar to the cortical

organization, develop topology-preserving neural arrangements and connectivity patterns being driven by the distribution of the sensory input.

In Chapter 4, we introduce the first neural framework for the modeling of human-object interactions from RGB-D videos. Our approach consists of two network streams processing action cues in terms of body posture and the manipulated object which converge into a final layer mapping motion patterns to objects in an unsupervised manner. Moreover, we introduce a dataset of transitive actions that we have collected for the purpose of the current study, but which will be used to evaluate the architectures proposed in the following chapters as well. We provide an in-depth analysis of the architecture by analyzing the importance of the objects as contextual information and the neural responses to congruent and incongruent action-object pairs. Furthermore, we carry out a quantitative evaluation by comparing action recognition rates achieved on a benchmarking dataset to the state-of-the-art methods for the recognition of human-object interactions.

In Chapter 5, we investigate the use of hierarchical self-organizing learning together with a temporal association mechanism for the simultaneous learning and prediction of human motion patterns. We evaluate this architecture in the context of a human-robot interaction scenario in which the robot has to learn and reproduce visually demonstrated arm movements. To assess the prediction accuracy, we set up experiments whereby the training of the architecture is carried out by introducing action categories incrementally. We analyze the online response of the architecture during the introduction of a new input sequence, the prediction performance during incremental learning and the sensitivity of the model with respect to learning parameters. Additionally, we show the robustness of our model when dealing with occasionally missing sensory data during the training process.

In Chapter 6, we propose a model for the prediction of human-object interactions in which the temporal order of action sub-sequences emerges through asymmetric lateral connections between neurons. We investigate the use of the model for the prediction of plausible future actions as well as its capability to synthesize body motion patterns.

In Chapter 7, we propose a novel top-down modulation mechanism which modulates the neural growth and the topological structure in a hierarchical arrangement of growing self-organizing networks. We apply both learning mechanisms in order to learn the compositionality of human activities and capture temporal relations between composing actions. Experimental results with a benchmarking dataset show that we outperform the state-of-the-art approaches with respect to the recognition of high-level human activities while the application of the top-down

learning mechanism optimizes the internal representations according to the task.

Concluding in Chapter 8, the neural network architectures and the experimental results presented in this thesis are discussed from the perspective of our research questions. Furthermore, we discuss the advantages of topology preserving networks for the tasks of human action recognition and prediction as well as analogies and limitations with respect to biological findings from which we take motivation. Finally, we provide a series of possible future research directions.

Chapter 2

Human Activity Recognition and Anticipation

Action understanding and anticipation lie at the heart of social interaction. Knowing the goal of other persons' actions allows for anticipating what they are going to do next and planning one's own actions accordingly. Neurophysiological studies have identified a widely distributed and complex network of brain areas which are specialized in the visual encoding of biological motion and body parts, such as fingers, hands, face, and limbs, and the identification of manipulated objects (Downing and Peelen, 2011; Beauchamp et al., 2002; Rizzolatti et al., 2001). The human brain efficiently processes multiple streams of information regarding the action cues in order to infer others' goals as well as interact with the environment, a capability which is crucial for survival. Hence, the investigation of the underlying neuro-computational mechanisms for action recognition in the human brain is fundamental for the development of artificial systems which face several challenges such as cluttered environments and reasoning on complex scenes.

In Section 2.1, we give an introduction to how action understanding is achieved in the human brain and the underlying neural mechanisms. In Section 2.2, we describe state-of-the-art approaches for complex visual tasks such as transitive action recognition, motion prediction, and human action anticipation.

2.1 The Understanding of Actions in the Brain

2.1.1 How Do We Understand Others' Actions?

The capacity of humans to effectively use objects sets them apart from all other species (Johnson-Frey, 2003). Most human actions are *transitive actions*, i.e.,

involve manipulation or interaction with objects (Johnson and Grafton, 2003). The use of objects unlocks a variety of effects humans can achieve in their living environment, from cutting with a knife to contacting others through a mobile phone and traveling the world with various types of vehicles. The understanding of others' transitive actions, on the other hand, represents a key function of the human visual system for goal inference and social communication. Over the last decade, action processing and understanding has received a lot of attention in the neuroscience community as well as other disciplines, such as computer vision and robotics (Giese and Rizzolatti, 2015; Aggarwal and Ryoo, 2011; Demiris and Hayes, 2002). However, the neural basis of this visual capability remains only partially understood.

Researchers from the neuroscience community have argued that the *mirror neurons* are the key element of action/intention understanding (Gallese et al., 1996; Rizzolatti et al., 2001). Mirror neurons were originally found in the ventral premotor cortex of the macaque brain. The key characteristic of the mirror neurons is that they fire both when the monkey manipulates an object in a specific way and when it observes another monkey (or experimenter) perform the same action or a similar one. This distinguishes mirror neurons from other *sensory* or *motor* neurons whose discharge is associated either with observation or execution, but not both. So, action execution and observation are closely related and the ability to interpret others' actions requires the involvement of the own motor system. It was proposed that the mechanism underpinning action understanding and goal inference is a 'direct matching' between the observed motor acts (e.g., the trajectory of the hand) to the observer's motor repertoire. Given that the observer knows the outcome of an action when executing it, he/she can recognize the goal of the observed action when the same set of neurons are active during the execution of that action.

The actions associated with the mirror neurons are mainly transitive. For instance, whole-body motions such as walking, turning, and gazing typically do not recruit mirror-related areas. Also in the adult human brain, there is evidence for the existence of a *mirror neuron system*, i.e., brain regions (or a set of regions) which are activated both during the observation as well as during the execution of similar actions (Rizzolatti and Fogassi, 2014). Apart from action understanding (Umilta et al., 2001), many other high-level functions have been associated with the human mirror neuron system such as imitation (Carr et al., 2003), intention attribution (Iacoboni et al., 2005), and the evolution of language (Rizzolatti and Arbib, 1998). However, studies conducted at the cellular level are mostly available

for monkeys, leaving the underlying neural and computational mechanisms of the human mirror neurons system largely unknown (Kilner and Lemon, 2013).

Recent findings have contrasted the traditional view that mirror neurons are activated only by the type of motor act being observed/executed and have demonstrated that the objects knowledge plays a key role in action understanding. For instance, a gaze can sometimes recruit mirror areas when the gaze points to graspable objects. Rizzolatti and Craighero (2004) showed that the mirror neurons exhibited a decrease in response in the case actions were mimicked, i.e., the target object was absent. Furthermore, the category of the object being manipulated and the value the object has for the monkey, e.g., food vs an inedible object, has been recently found to modulate the mirror neurons' response (Caggiano et al., 2012). Neurophysiological data in the human brain concerning mirror areas and beyond confirm that only when the information about the object identity is added to the semantic information about the action, then the actions of other individuals can be completely understood (Saxe et al., 2004). Objects are also one of the factors having an impact on the intention inference and action prediction. From the experiments with a monkey observing demonstrations of the task of grasping to *eat* and grasping to *place* an object, Fogassi et al. (2005) found that the observer could only make predictions based on the object's identity. For instance, the presence of food led the monkey to anticipate the action of *eating*, while the presence of an inedible object led the monkey to anticipate the action *placing*.

The knowledge about objects and their action-related attributes is acquired gradually during the development as a result of both exploratory and observational learning. The process starts early on with innate reflexes such as the palmar-grasp reflex to objects inside the palm and then goes through self-exploration and observation of the own action consequences towards building concepts like affordances and the object's function within 9 months of age (Rosenbaum, 2009). The idea behind the concept of object affordances is that there exists a direct relation between low-level visual features of objects to the type of grasping possibly performed on that object (Gibson, 2014; Jamone et al., 2016). This is supported by neurophysiological evidence found in the so-called *canonical neurons* which are activated during the visual presentation of a given object and also during the grasping of that object (Murata et al., 1997; Rizzolatti and Fadiga, 1998). Behavioral and imaging studies in humans have confirmed, for instance, that passively viewing an object, i.e., without interacting with it, can activate basic movements for reaching and grasping it (Buccino et al., 2009). Building on top of the self-explored object affordances, the emulation of others' goal and action imitation become possible at

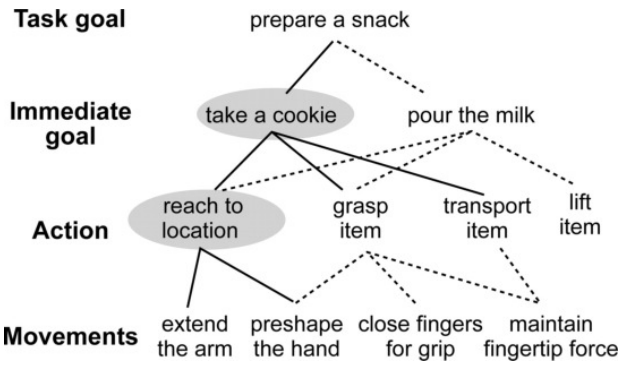


Figure 2.1: The hierarchical organization of goals (reprinted with permission from Hamilton and Grafton (2006)). A task goal may involve several immediate goals, achieved through a sequence of basic actions. Each action is composed of several movements.

the age of 12 months (Want and Harris, 2002). During the second year of life, a cognitive leap follows towards understanding scenarios that include the use of multiple tools and complex problem-solving. A number of researchers believe that the latter developmental stage requires not just the information that is directly perceived but also the ability to engage symbolic or relational thinking, although how this happens is still an unresolved issue (Bates, 2014).

Actions and their goals can be ordered hierarchically according to their level of abstractness or the time required for their completion. Neuroscientists distinguish mainly between motions (e.g., opening the hand), actions (often transitive actions e.g., reaching or grasping a cookie), immediate goals (e.g., take a cookie) and task goals (e.g., prepare a snack) (Hamilton and Grafton, 2006) (see Fig. 2.1). A great variety of these goals are encoded irrespective of the motor acts executed for achieving it (Rizzolatti and Fogassi, 2014). Moreover, goals can be inferred by observing only a few of the activity's motor acts. According to the mirror neuron literature, an internal simulation of the observed action, based on the observer's motor repertoire, allows the observer to infer others' goals. The underlying mechanism for this is believed to be the sequential activation of a subpopulation of neurons encoding subsequent motor acts. Different computational models have been proposed for modeling the chained neural activation, for instance, the *synfire chains* model first theorized by Abeles (1982), and the *neuronal chains* model proposed by Chersi et al. (2011). Unlike the first model, the neuronal chains model does not require synchronicity between the neurons and can generate the same sequence with varying durations of the composing motor acts determined by external regulatory signals.

The actions performed in an implausible or unusual way, e.g., turning on a light switch with the knee when both hands are free, seem to be an exception. A functional magnetic resonance imaging (fMRI) study conducted by Brass et al. (2007) showed that the observation of this type of actions elicits greater activity in the brain areas which lack mirror properties. This is presumably explained by the fact that the action does not allow a match with one's own motor repertoire. In addition to this, Van Overwalle and Baetens (2009) suggest that several brain areas beyond the mirror neuron system are activated when observed actions lead to multiple goals, due to requiring more mental processing for selecting one of them.

Even though the debate about how goals and intentions are encoded in the brain is on-going and often controversial (Cavallo et al., 2016), there exists a wide spectrum of physiologically inspired models for action processing and understanding (see Giese and Rizzolatti (2015) for a review). Many of these models have never been concretely implemented and have served only as conceptual frameworks. Furthermore, only a few of them address simple transitive actions such as grasping, placing and holding (Fleischer et al., 2013; Prevete et al., 2008; Tessitore et al., 2010). However, the insights they provide about biological motion processing and integration of the information regarding hand and the manipulated objects improve our understanding of the brain and can contribute to the development of artificial models of perception.

2.1.2 Neural Mechanisms for Transitive Action Perception

The processing of the perceived visual cues about body and objects produce distinct patterns of activity in the human cortex (Beauchamp et al., 2002). A schematic illustration of the brain containing a set of areas involved in visual processing of transitive actions is shown in Fig. 2.2. According to the two visual streams hypothesis, visual information is processed in two separate pathways in the primate cerebral cortex. The ventral pathway plays an important role in constructing semantic perceptual information about the objects, whereas the dorsal pathway is involved in spatial awareness and guidance of actions. Object recognition is carried out in the ventral visual stream which is composed of feedforward, hierarchically-organized cortical areas culminating in the Inferior Temporal Cortex (IT) (Felleman and Essen, 1991). With each layer in the hierarchy, the abstraction level of visual features is increased through the alternation of simple and complex cells decreasing sensibility to objects' location and scale (Hubel and Wiesel, 1962). Taken together, these findings have led to the successful development of biologically inspired architectures for object recognition (Fukushima,

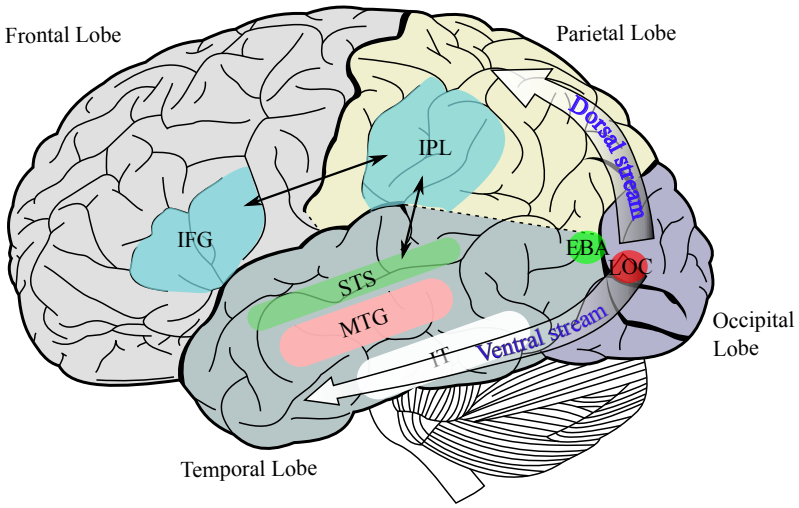


Figure 2.2: Schematic illustration of the location of the human brain areas involved in the perception of transitive actions. EBA, extrastriate body area; LOC, lateral occipital complex, IT, inferior temporal cortex; MTG, middle temporal gyrus; STS, superior temporal sulcus; IPL, inferior parietal lobule, IFG, inferior frontal gyrus (homologous to the macaque area F5). Image modified from Wikimedia (2007)

1980; Riesenhuber and Poggio, 1999; Serre et al., 2007). In contrast to the commonly accepted invariant representations with respect to the object's position and scale, the view-independent object representations have found no consensus among researchers. Three-dimensional rotation changes objects' shape due to the 2D retinal projections from the three-dimensional space. Some theories suggest the view-invariant representations of objects, i.e., the underlying neural representations respond similarly to an object across it views (Marr et al., 1980; Biederman, 1987; DiCarlo et al., 2012). Other theories suggest that objects representations are view-dependent, that is, they consist of several 2D views of an object (Poggio and Edelman, 1990; Perrett, 1996). Empirical evidence for the view sensitivity of the human object-selective cortex has been found recently by Grill-Spector (2013). The author hypothesizes that view invariance may be achieved utilizing a population code across neurons, which themselves are not view-invariant.

The neural mechanisms for the processing of the body pose is an extension of the shape-processing ventral pathway model. It continues to higher levels of cortical substrates consisting of 'snapshot neurons' selectively responding to body shapes, whose existence is supported by neurophysiological data and brain imaging experiments (Grossman and Blake, 2002). The highest hierarchy level of the body pose processing pathway consists of *motion pattern neurons*, which, according to physiological data, are possibly located in the superior temporal sulcus (STS) and

the ventral premotor cortex (F5) (Perrett et al., 1985). The motion pattern neurons summate the activity of all snapshot neurons that are active during the same movement pattern. Therefore the biological movements might be recognized as sequences of body poses corresponding to snapshots of complex movements (Giese and Poggio, 2003). Interestingly, the motion pattern neurons are highly selective to the temporal order of the snapshots, e.g., randomization of the temporal order of the frames of a movie typically disrupts the perception of a biological movement pattern. Not only posture but also motion plays a key role in biological motion perception (Oram and Perrett, 1996). The motion-processing dorsal pathway processes biological movements as optic-flow patterns and has, in principle, a similar hierarchical architecture as the form pathway. The two pathways converge at the level of the STS area.

Representations of human body parts reside in cortical areas distinct from representations of other object categories. Downing and Peelen (2011) identified a part of the human extrastriate cortex involved in the visual processing of the human body and body parts, namely the extrastriate body area (EBA). On the other hand, the identity of the objects is processed in the Lateral Occipital Complex (LOC) area (Grill-Spector, 2013). The functional and anatomical segregation for the processing of objects and body pose has been confirmed by experimental results on the brain's response during viewing of human object-manipulation images (Beauchamp et al., 2002). The STS was not activated during viewing of animated pictures of man-made objects. Instead, the activation occurred in the middle temporal gyrus (MTG) even when viewing static pictures of objects commonly associated with motion, e.g., a picture of a house does not activate the lateral temporal cortex. The mirror neuron literature suggests that the biological motion after having been processed and encoded in the STS is further transmitted to the inferior parietal lobule (IPL) area where the relationship with the object is specified and kinesthetic qualities are evaluated and then to the inferior frontal gyrus (IFG) which is the human homolog of area F5 in the macaque brain (Keysers and Perrett, 2004; Rizzolatti et al., 2001) (both areas are highlighted in blue in Fig. 2.2). The IFG is where the action goal coding occurs.

All these findings together suggest that the processing and integration of different visual information underlie the emergent representations of human-object interactions. From the computational perspective, an important question can be posed on the type of representations of body postures and manipulated objects involved in the learning of transitive actions and, in particular, on the way the two can be integrated. Representations of human-object interactions are not merely

the visual features of shape and motion of the action components but also higher order features which represent the interaction stimuli (Tunik et al., 2007; Baldassano et al., 2017). Translated to the computer vision task of the recognition of human transitive actions, this requires further reasoning on the visual and semantic features of full interactions, such as the spatial relationships between the manipulated objects and the body parts and their temporal dynamics (Yao and Fei-Fei, 2010b) or the identity of the objects involved.

2.2 Computational Models

2.2.1 Introduction to Challenges of Visual Activity Recognition

The goal of vision-based human activity recognition systems is to automatically detect and analyze human activities from the information acquired from visual sensors, e.g., a sequence of images captured by an RGB or an RGB-D camera. The task has been of strong interest for different fields of research since the early 1990s (Aggarwal and Ryoo, 2011). Major components of such recognition systems include feature extraction, action learning and classification, and action recognition and segmentation (Poppe, 2010). A simple recognition process for a learning-based algorithm consists of three steps, namely the detection of the human and/or his/her body parts, motion tracking, and then recognition using the tracking results. For example, to recognize the *waving* activity, the person's arms and hands are first detected and tracked, then spatiotemporal descriptions of the movement are extracted and compared to existing patterns in the data learned during training to finally determine the action class.

The literature suggests a conceptual categorization of human activities into four different levels depending on the complexity: gestures, actions, interactions, and group activities (Aggarwal and Ryoo, 2011; Ziaeefard and Bergevin, 2015; Aggarwal and Xia, 2014). Gestures are elementary movements of a person's body part and are the atomic components describing the meaningful motion of a person, e.g., *stretching an arm* or *raising a leg*. Actions are single-person activities that may be composed of multiple gestures such as *walking* and *waving*. Interactions are human activities that involve a person and one object (or multiple objects). For instance, *a person making a phone call* is a human-object interaction. Finally, group activities are the activities performed by groups composed of multiple persons or objects, e.g., *a group having a meeting*. While there has been extensive

work on gestures and actions, only in the last decade the field has moved towards the recognition of complex human activities involving objects or multiple persons.

Vision-based human activity recognition faces several challenges:

- *Intra-class variations and inter-class similarity:* Different individuals can perform the same action in a different manner. For example, the *walking* action can be performed with different speed and stride length. Furthermore, there are anthropometric differences between individuals which may increase the variations of the possible movement patterns within one action class. With the increasing number of action classes, there is more overlap between movement patterns as well. Two actions may become distinguishable by very subtle spatiotemporal details. This remains a major issue for a great number of existing approaches, especially those relying solely on the body pose and motion information (Wang et al., 2014; Shahroudy et al., 2016; Cippitelli et al., 2016).

- *Environment and recording settings:* Occlusions, cluttered environments, shadows, varying illumination conditions, and dynamic backgrounds may lead to erroneous body segmentations and may alter significantly the way actions are perceived. In addition to this, the same action observed from different viewpoints can lead to very different image sequences. This remains particularly a major issue for applications with traditional 2D sensors (e.g., RGB cameras) (Lea et al., 2016; Ma et al., 2017).

- *Temporal segmentation:* Many action recognition systems require actions to be manually segmented in time. Erroneous segmentations can have a negative impact on the performance, especially for systems that rely on a single representation for entire image sequences, for instance, bag-of-words representations built on spatiotemporal interest points (STIPs) descriptors (Wang et al., 2009; Rybok et al., 2014).

- *Unlabelled training data:* The performance and scalability of an action recognition system need to be analyzed through larger-scale experiments, which require a large amount of training and test sequences. Oftentimes training data labels are sparse or unavailable. Since manual annotation is expensive, the development of unsupervised or semi-supervised approaches is encouraged (Wu et al., 2015; Lan et al., 2015; Parisi et al., 2016b).

The introduction of depth sensing devices, such as the Microsoft Kinect and ASUS Xtion, represents a significant contribution to the field of action recognition since

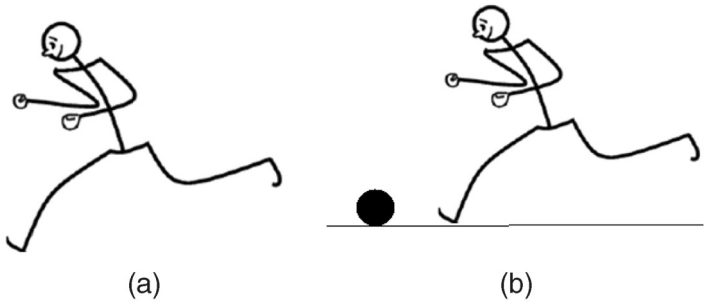


Figure 2.3: Action recognition requires contextual information. The same pose can have different meanings based on the context: (a) running and (b) kicking (reprinted with permission from Gupta et al. (2009)).

they largely alleviate some of the aforementioned low-level difficulties. This sensor technology provides depth measurements used to obtain reliable estimations of 3D human pose in cluttered environments, including a set of body joints in real-world coordinates and their orientations. The representation of the human body pose through a 3D skeletal configuration is more robust to varying illumination conditions and dynamic environments and can be used for building a view-invariant action recognition approach (Aggarwal and Xia, 2014). Although deep learning architectures have recently shown high accuracy in body segmentation from RGB images (Lea et al., 2016), capturing articulated human motion from sequences of RGB images may result in loss of information, due to for instance limb occlusion or bad point of view. Moreover, 3D body tracking through depth sensors remains the less computationally expensive method for motion segmentation and body pose estimation (Han et al., 2013).

Many human actions involve similar movement patterns but have different meanings according to their context, i.e., the visual cues from the surrounding environment which carry information about the action such as the manipulated objects (see Fig. 2.3). Behavioral studies show that context plays an important role in action recognition in the human visual system. Detecting visual abnormalities out of context can become crucial for survival (e.g., detecting an unattended bag in the airport) (Henderson, 2003). In computer vision, context has often been used in problems such as object detection and recognition (Divvala et al., 2009; Heitz and Koller, 2008), scene recognition (Murphy et al., 2004), and object segmentation (Shotton et al., 2006). The idea of using context information for action recognition has been tackled only in the last decade and has shown to significantly boost performances in action recognition tasks starting from the first experiments with static images (Yao and Fei-Fei, 2010b).

The recognition of human-object interaction activities relies heavily on the contextual information, e.g., the motion analysis and identification of the manipulated objects (Gupta et al., 2009; Yao and Fei-Fei, 2010a). However, object detection and recognition are subject to ambiguity due to clutter and occlusion or even bad light conditions as long as appearance features are used, i.e., color, texture, and shape cues. The difficulty is even higher when objects are being manipulated, due to being partially visible or completely occluded. All these can have a negative impact on the resulting action recognition performance. Another issue is the difference in the appearance of objects belonging to the same category, e.g., cups come in different sizes and shape, however, they all serve the purpose of *drinking*. For this, several approaches assign high-level, semantic attributes to each class of object instead of relying on the single object category (Farhadi et al., 2009; Lampert et al., 2009). Object attributes can describe parts (a car's wheels), shape (rectangular), and materials (metallic). Although this has been shown to generalize well to unseen objects and transfer information between classes, it has the disadvantage of being sensitive to the manual selection and assignment of the semantic attributes.

The understanding of human activities is essential for various applications, from surveillance systems in public places such as airports and subway stations to human-computer interaction systems. The reliable recognition of human activities is fundamental for real-time monitoring of people with disabilities, seniors, and babies in a residential context or for assessing the progress of patients during the at-home rehabilitation (see Fig. 2.4). Within this context, mobile robots may be designed to process the sensed information and assist people according to their necessities. There has been an increasing number of ongoing research projects aimed to develop assistive robots in smart environments for self-care and independence of the elderly at home (Amirabdollahian et al., 2013; Cecchi et al., 2016). As a result, advanced robotic technologies which comprise socially-aware human-robot interaction have been developed. This increasing interest and effort are closely related to the rising user acceptance (Torta et al., 2012). During the last decade, people seem to advocate the use of assistive robots in their homes, be it for physical or mental training, or for support for basic daily tasks such as reminding them when to take medication or drink/eat food. The positive effects on the senior's well-being through the use of socially assistive robots in domestic environments is supported by recent studies as well (see Kachouie et al. (2014) for a review). However, together with the promising results, robotic technologies introduce a vast set of challenges and technical concerns which need to be addressed.

Figure 2.4: A robot monitoring and assisting an elderly in a home environment. In this example, the Care-O-Bot 3 robot (Reiser et al., 2013) offers water to the elderly after detecting that she has not drunk enough water for a long time.

2.2.2 Recognition of Transitive Actions

Different approaches: Understanding human-object interactions requires the integration of complex relationships between features of human body action and object identity. From a computational perspective, it is not clear how to link architectures specialized in object recognition and motion recognition, e.g., how to bind different types of objects and hand/arm movements. Recently, Fleischer et al. (2013) proposed a physiologically inspired model for the recognition of transitive hand-actions such as grasping, placing, and holding. Nevertheless, this model works with visual data acquired in a constrained environment, i.e., videos showing a hand grasping balls of different sizes with a uniform background, with the role of the identity of the object in the transitive action recognition being unclear. Similar models have been tested in robotics, accomplishing the recognition of grip apertures, affordances, or hand action classification (Prevete et al., 2008; Tessitore et al., 2010).

Various approaches for the recognition of human-object interactions do not explicitly model the interplay between object recognition and body pose estimation. Typically, first, objects are recognized and activities involving them are subsequently recognized, by analyzing the objects' motion trajectories (Wu et al., 2007). The approach from Yang et al. (2015) infers actions by considering all possible trigrams $<$*Object1*, *Action*, *Object2*$>$ extracted from the sentences in the English Gigaword corpus. Pieropan et al. (2014b) proposed including action-related audio

cues in addition to the spatial relationship among objects in order to learn object manipulations for the purpose of robot learning by imitation. However, important descriptive visual features like body motion or fine-grained cues like the hand pose during manipulation were not considered.

Probabilistic approaches have been extensively used for reasoning upon relationships and dependencies among objects, motion, and human activities. Gupta et al. (2009) modeled hand trajectories with Hidden Markov Models (HMM) and applied a Bayesian network for integrating the appearance of manipulated objects, human motion, and reactions of objects. Following a similar approach, Ryoo and Aggarwal (2007) introduced an additional semantic layer providing feedback to the modules for object identification and motion estimation leading to an improvement of object recognition rates and better motion estimation. Nevertheless, the subjects' articulated body pose was not considered as input data, leading to applications in a restricted task-specific domain such as airport video surveillance. Other research studies have modeled the mutual context between objects and human pose through graphical models such as Conditional Random Fields (CRF) (Yao and Fei-Fei, 2012; Koppula et al., 2013; Kjellström et al., 2011). These types of models suffer from high computational complexity and require a fine-grained segmentation of the action sequences.

A different approach for the recognition of human-object interactions has been the extraction of novel low-level visual features encoding the spatial relationships between the human and the manipulated objects. Yao and Fei-Fei (2010a) proposed the *Grouplet* feature which captures the spatial organization of image patches encoded through Scale-Invariant Feature Transform (SIFT) descriptors (Lowe, 2004). Their method is able to distinguish between interactions or just co-occurrences of humans and objects in an image, but no applications on video data have been reported. Aksoy et al. (2011) proposed the Semantic Event Chains (SEC), i.e., a matrix whose entries represent the spatial relationship between extracted image segments for every video frame. Action classification is obtained in an unsupervised way through maximal similarity. While this method is suitable for teaching object manipulation commands to robots, the representation of the visual stimuli does not allow for reasoning upon semantic aspects such as the congruence of the action being performed on a certain object.

Early attempts to apply neural networks for the problem of understanding human-object interactions from visual perception yielded promising results. Shimozaki and Kuniyoshi (2003) proposed a Self-Organizing Map (SOM) based hierarchical architecture capable of integrating object categories, spatial relationships,

and movement and it was shown to perform well on simple 2D scenes of ball handling actions. However, compared to the static image domain, there is limited work on understanding human-object relationships from video data sequences with neural network architectures (Lea et al., 2016; Ma et al., 2017).

On object affordances: When reasoning about the environment in terms of actions and objects the concept of affordances also comes into play. For this reason, affordances have been a major focus of numerous robotic studies especially in scenarios of robots learning by demonstration and action planning. However, studies based on a practical interpretation of the concept do not provide a unified view of how to represent affordances for effectively using them in complex scenarios (see Jamone et al. (2016) for an overview). Early work from Fitzpatrick and Metta (2003) put forward the idea that a robot can learn about affordances by acting on objects and observing the effects. This idea was followed by a number of researchers, who implemented methods for the learning of the action effects in an unsupervised way and then clustering the stored experiences in order to discover object categories (Ugur et al., 2009; Ridge et al., 2010). In scenarios with multiple objects, Stoytchev (2005) investigated the learning of affordances as a tool-behavior pair that provides a desired effect but did not make associations between the distinctive features of the objects and their affordances.

Although useful for robot operations, it is not clear how to bootstrap the affordance knowledge acquired through self-exploration in order to generalize to previously unseen objects or to understand others' actions. To address this problem, different approaches describe objects in terms of the function of their geometrical parts (Schoeler and Wörgötter, 2016) or through the modeling of the interaction scene, for instance, by observing humans performing activities using objects and clustering them into functional classes (Pieropan et al., 2014a). While such systems are very useful in practice (e.g., for a service or collaborative robot), they do not provide insights into how humans use affordances to understand others' actions. Therefore, even though the idea of endowing an agent with the capability of reasoning about objects in terms of affordances is quite attractive, it also means that the affordances need to be encoded in terms of sensory data such that generalizations can be made in different scenarios, e.g., an affordance model should represent the roll-ability of an object but also the sit-ability of a chair, and this seems to be an open challenge.

Body features from RGB-D data: Since the introduction of the low-cost depth sensing devices, such as Microsoft Kinect and Asus Xtion, there has been an extensive work in human action recognition from depth data (Cippitelli et al.,

2016; Yang and Tian, 2014; Sung et al., 2012). Among a large number of human body representation approaches we can distinguish between two broad categories: 1) representations based on the RGB-D information , and 2) representations based on the 3D skeleton data. Some methods that belong to the first category use, for instance, 3D silhouettes and extract spatiotemporal features from the temporal evolution of the silhouettes during action performance (Li et al., 2010; Yang et al., 2012). 3D silhouettes-based algorithms are usually view-dependent, thus more suitable for describing actions parallel to the camera. A number of methods have explored the use of the spatio-temporal interest points (STIP) descriptor for RGB-D data. The advantage of this descriptor stands in its invariance to spatiotemporal shifts and scales and its capability to deal with occlusions, thereby being suitable for the recognition of human-object interactions. However, STIP-based methods require the whole video as input and are very slow to compute, thus limiting their real-time application. High computational cost and poor real-time performance is also the major limitation of approaches based on 3D optical flow or scene flow using RGB and depth (see Aggarwal and Xia (2014) for a review).

The representation of the human body as *skeletons*, i.e., an articulated system of rigid segments connected by joints, has been of great interest long before the proliferation of low-cost depth sensors. Back in 1973, Johannson's experiments evidenced the remarkable efficiency of the humans in recognizing actions by only seeing animated figures of light spots attached to a person's major joints (Johansson, 1973). In computer vision, researchers tried to extract skeletons from silhouettes (Fujiyoshi et al., 2004) or label main body parts such as arms, legs, torso, and head for activity recognition (Ben-Arie et al., 2002). By now, skeletal joints tracking algorithms are built into the Kinect device or are offered by freely available OpenNI libraries, thereby offering an easy access to the skeletal joint locations in real-time applications. The convenience of this technology has led to a great number of applications for the recognition of full-body actions and hand gestures (Aggarwal and Xia, 2014; Parisi et al., 2015).

Unlike the features from 3D silhouettes, the skeletal joint features are invariant to the camera location and subject appearance or to body size. Moreover, human action recognition schemes based on the skeletal joints features are better at modeling finer activities compared to the 3D silhouettes based approaches. The main limitation of the skeletal features is the lack of information about surrounding objects. For this, Wang et al. (2014) proposed a new 3D appearance feature called Local Occupancy Pattern (LOP) describing the depth appearance in the neighborhood of a 3D joint, and thus capturing the relations between the human

body parts, e.g., hands, and the environmental objects that the person is interacting with. Although their method produces state-of-the-art results, the identity of the objects is completely ignored, and the discriminative power of such features is unclear when the objects being manipulated are small or partially occluded. An alternative method would be to model human-object interactions considering the skeletal features combined with object recognition and tracking.

2.2.3 Motion Prediction

Motion analysis and prediction are an integral part of robotic platforms that counterbalance the imminent sensorimotor latency. Well-known methods for tracking and prediction are the Kalman Filter models, as well as their extended versions which assume non-linearity of the system, and the Hidden Markov Model (HMM)s. Kalman filter-based prediction techniques require a precise kinematic or dynamic model that describes how the state of an object evolves while being subject to a set of given control commands. HMMs describe the temporal evolution of a process through a finite set of states and transition probabilities. Predictive approaches based on dynamic properties of the objects are not able to provide correct long-term predictions of human motion (Vasquez et al., 2008) due to the fact that human motion also depends on other higher-level factors than kinematic constraints, such as plans or intentions.

Neural networks provide an alternative approach for motion prediction. They are known to be able to learn universal function approximations and thereby predict non-linear data even though dynamic properties of a system or state transition probabilities are not known (Schaefer et al., 2008; Saegusa et al., 2007). For instance, the Multilayer Perceptron (MLP) and the Radial Basis Function (RBF) networks, as well as Recurrent Neural Networks (RNNs) have found successful applications as predictive approaches (Mainprice and Berenson, 2013; Barreto, 2007; Ito and Tani, 2004; Zhong et al., 2012). A subclass of neural network models, namely the Self-Organizing Map (SOM) (Kohonen, 1990), is able to perform local function approximation by partitioning the input space and learning the dynamics of the underlying process in a localized region. The advantage of the SOM-based methods is their ability to achieve long-term predictions at much less expensive computational time (Simon et al., 2007).

Johnson and Hogg (1996) first proposed the use of multilayer self-organizing networks for the motion prediction of a tracked object. Their model consisted of a bottom SOM layer learning to represent the object states and the higher SOM layer learning motion trajectories through the leaky integration of neuron activations

over time. Similar approaches were proposed later by Sumpter and Bulpitt (2000) and Hu et al. (2004), who modeled time explicitly by adding lateral connections between neurons in the state layer, obtaining performances comparable to that of the probabilistic models.

Several other approaches use SOMs extended with temporal associative memory techniques (Barreto, 2007), e.g., associating to each neuron a linear Autoregressive (AR) model (Walter et al., 1990; Vesanto, 1997). A drawback which is common to these approaches is their assumption of knowing a priori the number of movement patterns to be learned. A better alternative would be to adopt growing extensions of the SOM such as the Growing When Required (GWR) algorithm (Marsland et al., 2002; Parisi et al., 2016a). The GWR algorithm has the advantage of a nonfixed, but varying topology and requires no specification of the number of neurons in advance. However, the prediction capability of the self-organizing approaches in the case of multidimensional data sequences has not been thoroughly analyzed in the literature.

2.2.4 Incremental Learning of Motion Patterns

In the context of learning motion sequences, an architecture capable of incremental learning should identify unknown patterns and adapt its internal structure in consequence. This topic has been the focus of a number of studies on the Programming by Demonstration (PbD) (Billard et al., 2016). Kulić et al. (2008) used Hidden Markov Models (HMMs) for segmenting and representing motion patterns together with a clustering algorithm that learns in an incremental fashion based on intra-model distances. In a more recent approach, the authors organized motion patterns as leaves of a directed graph where edges represented temporal transitions (Kulić et al., 2012). However, the approach was built upon automatic segmentation which required observing the complete demonstrated task, thereby becoming task-dependent. A number of other works have also adapted Hidden Markov Model (HMM)s to the problem of incremental learning of human motion (Takano and Nakamura, 2006; Billard et al., 2006; Ekvall et al., 2006; Dixon et al., 2004). The main drawback of these methods is their requirement for knowing a priori the number of motions to be learned or the number of Markov models comprising the learning architecture.

Ogata et al. (2004) proposed a model that considers the case of long-term incremental learning. In their work, an RNN was used to learn a navigation task in cooperation with a human partner. The authors introduced a new training method for the recursive neural network in order to avoid the problem of mem-

ory corruption during new training data acquisition. Calinon and Billard (2007) showed that the Gaussian Mixture Regression (GMR) technique can be successfully applied for encoding demonstrated motion patterns incrementally through a Gaussian mixture model (GMM) tuned with an expectation-maximization algorithm (EM). The main limitation of this method is the need to specify in advance the number and complexity of tasks in order to find an optimal number of Gaussian components. Therefore, Khansari-Zadeh and Billard (2010) suggested a learning procedure capable of modeling demonstrated motion sequences through an adaptive GMM. Cederborg et al. (2010) suggested to perform a local partitioning of the input space through kd-trees and training several local GMR models.

However, for high-dimensional data, the partitioning of the input space in a real-time system requires additional computational time. Regarding this issue, it is convenient to adopt self-organizing networks that perform in parallel the partitioning of the input space, through the creation of prototypical representations, as well as the fitting of necessary local models. The application of a growing self-organizing network, such as the GWR, allows for the learning of prototypical motion patterns in an incremental fashion (Parisi et al., 2016a).

2.2.5 Action Prediction

The goal of human activity prediction has been formally defined as the capability to infer an ongoing activity given an incomplete temporal observation (Ryoo, 2011). Subsequently, most of the existing recognition approaches which make a decision at the end of an action sequence cannot be directly applied to the problem of activity prediction. Several approaches have been proposed, often referred to as *early activity recognition*, with the primary goal to infer the activity label from just the initial part of the video sequence (see Fig. 2.5). Ryoo (2011) proposed two variants of the *bag-of-words* for capturing how the distribution of the spatiotemporal features changes over time and represents video sequences as histograms. Training activity sequences were then modeled as histograms and comparison between learned activity models and incomplete test observations were computed using a dynamic programming algorithm. However, this approach did not account for the sequential nature of the temporal events. Cao et al. (2013) extended Ryoo's work by dividing each activity into multiple temporal segments and estimating the activity likelihood at each segment and finally combining the likelihoods to achieve a global posterior probability. Lan et al. (2014) proposed the hierarchical *movemes*, a new human motion representation that allows for describing motion at multiple levels of granularity and developed a learning framework on top of it

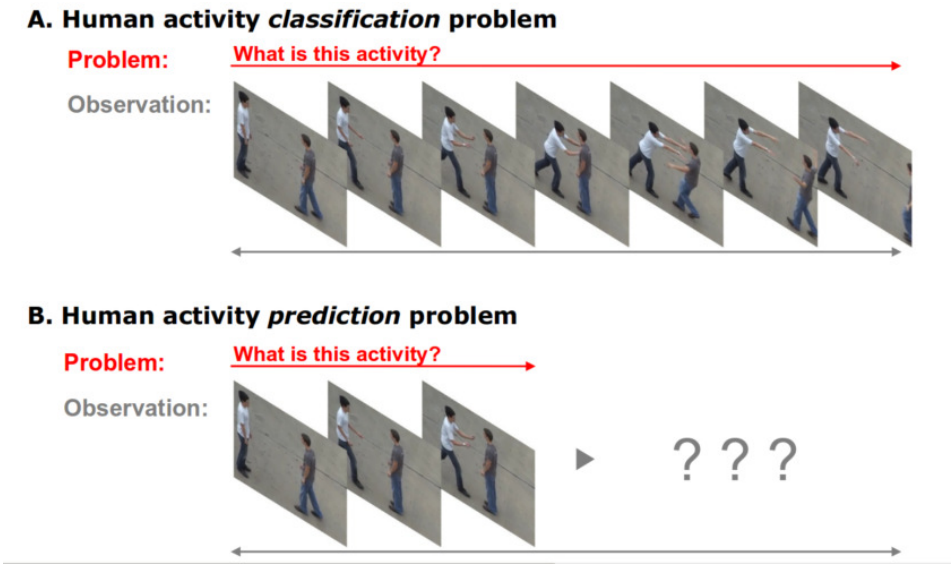


Figure 2.5: A comparison between the activity classification problem and the activity prediction problem (reprinted with permission from Ryoo (2011)).

for performing action prediction. SOM-based architectures have also been proposed for the purpose of action prediction (Ding et al., 2016) and motion sequence completion (Okada et al., 2004; Araujo and Barreto, 2002; Parisi et al., 2016b). However, the main goal of these approaches is to recognize an action while it is unfolding rather than what is likely to happen next. The latter would require to learn motion patterns and the temporal order of the atomic actions composing an activity. This would allow, for example, a better planning of a robot's response.

Typical hierarchical representations of the human activity rely on hybrid approaches in which perceptual sequences are learned, e.g., through a neural network model, at the lower level and are combined into more complex sequences, or activities, by assigning them arbitrary symbols or rules (Wermter, 2000; Taniguchi et al., 2016). However, these symbols are usually fixed and defined a priori by the designers based on their knowledge. For this purpose, many researchers believe that symbolic processing should be performed exclusively with analog dynamical systems, e.g., with neural network architectures (Arie et al., 2012; Nishimoto et al., 2008).

Chersi et al. (2014) have proposed a SOM-based computational model that combines principles of Hebbian learning and topological self-organization and reproduces the encoding of actions as well as language in the brain. The learning mechanism of this model establishes neural pools, i.e., neurons responding to similar visual input, and links among these pools which then form goal-directed

neuronal chains. Although this model is quite interesting by considering human cognition mechanisms, no results have been reported on real-world action recognition applications. The work of Koppula and Saxena (2016) addresses the problem of anticipation of human actions at a fine-grained level of atomic actions. The authors also focus on predicting not only *what* comes next but also *how* it is performed. However, they demonstrate the capability of their architecture to generate object trajectories, whereas no generation of body postures has been reported.

2.3 Summary

Humans possess an outstanding capability to easily infer and reason about abstract concepts such as the goal of actions which can vary from immediate goals related to a transitive action (e.g., take a cookie) to long-term intentions (e.g., prepare a romantic dinner) (Van Overwalle and Baetens, 2009). This capability requires processing of complex visual stimuli regarding body movement patterns and contextual cues, such as the manipulated objects as well as learning representations of the interaction stimuli. Since the discovery of the mirror neurons, the underlying neural mechanisms for action processing and goal inference have been of great interest to the neuroscience community (Rizzolatti and Fogassi, 2014). The study of the cortical areas activated during action perception has provided evidence for a highly distributed network of regions responsible for the coding and integration of the different action components. Body motion cues are processed through a hierarchy of spatiotemporal receptive fields with an increasing complexity of the representation, i.e., higher-level areas process information accumulated over larger temporal windows with increasing invariance to the position and the scale of stimuli. Segregated pathways are engaged in the processing and representation of the information about biological and non-biological stimuli, i.e., man-made objects (Beauchamp et al., 2002). The brain then integrates all streams of information as well as engages higher-level cortical areas in order to infer action goals in spite of apparent ambiguities. The underlying biological mechanisms for this process are largely unknown and complex, yet have been of fundamental importance to the development of basic artificial systems for the recognition or imitation of hand actions and provide a stepping stone to the robust recognition of whole-body transitive actions in real-world scenarios.

The goal of learning-based systems for the visual recognition of human activities is to automatically extract spatiotemporal descriptions of movements from sequences of images and learn action templates to compare to during the deter-

mination of the action label of a new sequence. The development of systems for the recognition of transitive actions is computationally more effortful than for simpler human activities such as gestures and single-person actions like *walking* and *jumping* due to requiring more fine-grained visual analysis. In the last decade, progress on transitive action recognition was accelerated by the latest technological advancements in visual sensing devices and the increase in the computational power of modern graphics processing units (GPU). In particular, the use of the low-cost depth sensing devices such as Microsoft Kinect and Asus Xtion cameras is quite promising due to the computational efficiency in sensory data processing and the real-time performance that such technology offers. Deep neural networks, on the other hand, are recently exhibiting high accuracy in terms of action recognition from large-scale datasets including actions that involve human-object interactions. However, despite recent progress in the field of action recognition, important questions remain open on how to extract and better process body features and how to encode spatial relationships between the human and the manipulated objects for effectively learning the complex dynamics of transitive actions in real-world scenarios. Further challenges have to be addressed such as the unreliable body tracks or systematic sensor errors affecting the integrity of the input stream. Most importantly, intelligent systems should not be limited to the recognition of human behavior but should also anticipate it while continuously adapting to the changing environment.

In the next chapters, we propose a set of neural network architectures for transitive action recognition and anticipation from RGB-D videos. We design our architectures taking into account some aspects of the biological transitive action perception and seek to achieve robust and online adaptable intelligent systems for enhancing human-robot interaction.

Chapter 3

Self-Organizing Networks

Neural network models of self-organization are inspired by biological findings such as the Hebbian learning and the brain maps plasticity (Hebb, 1949). These models have been applied to a variety of applications such as data compression and visualization, clustering, pre-processing of large datasets, classification, and regression. Moreover, they have shown their applicability in several high-level cognitive functions such as human action recognition as well as multi-modal perception (Shimozaki and Kuniyoshi, 2003; Ding et al., 2016; Parisi et al., 2016b). In this chapter, we provide an overview of existing self-organizing networks. We start with the competitive Vector Quantization (VQ) (Biehl et al., 2016) method and the Self-Organizing Map (SOM) network proposed by Kohonen (1990). Both comprise most of the computational ingredients that facilitate the understanding of the growing self-organizing networks described later on. In particular, we review the main properties, functionality, and drawbacks of each model while progressively moving towards hierarchical network arrangements for processing temporal sequences.

3.1 Self-Organization in the Brain

The main structures of the brain are determined genetically (Shatz, 1992). Humans are born with almost all neurons they will ever have. However, during lifetime, the brain becomes bigger because neurons grow in size and the number of connections increases. Continuous stimulation from the environment requires the brain to adapt by modifying its internal structure in order to achieve higher functional complexity and increase the probability of survival. There exists evidence showing that extrinsic factors such as sensory experience define the way patterns of connectivity and functions of the cortex are shaped. The earliest are studies from

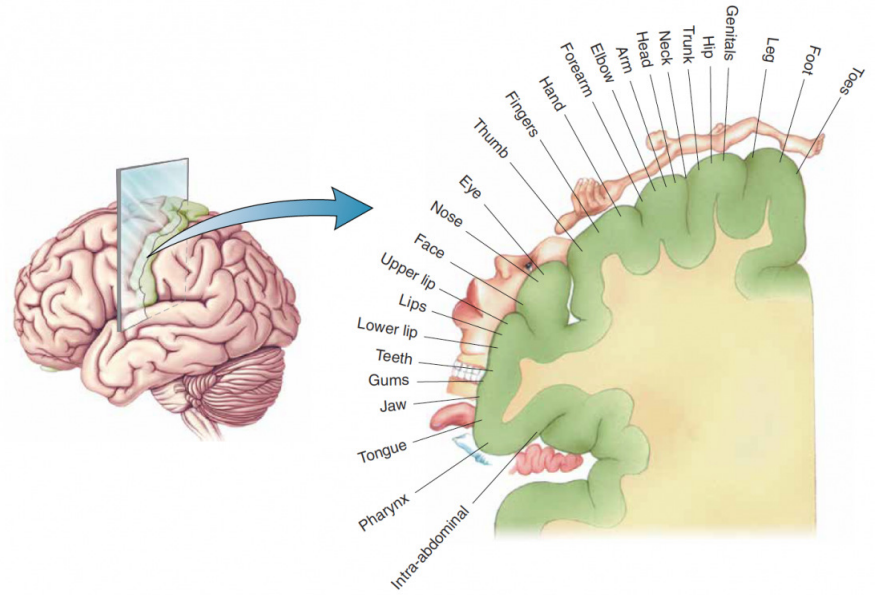


Figure 3.1: A somatotopic map of the body surface onto the primary somatosensory cortex. Neurons in each area are most responsive to the parts of the body illustrated above them[1].

the mammalian visual system, especially during the neonatal stages. Mountcastle (1957) and later Hubel and Wiesel (1962) found that certain neural cells in the cat's brain respond selectively to some specific sensory stimuli. These cells are arranged in the so-called *brain maps*, in which their topological location corresponds to some stimuli property, e.g., orientations, in an orderly fashion. Different studies suggest that the brain maps are strongly modified by visual experiences (Merzenich et al., 1983; Golarai et al., 2017) and that visual inputs are crucial for the normal cortical organization in general (Blakemore and Cooper, 1970; Hirsch and Spinelli, 1970). The brain's adaptation is also evident in case of injuries or sensory deprivation at a young age. A brain tumor, for instance, elicits the so-called cortical reorganization, meaning that different parts of the brain attempt to compensate for the functional deficit of the affected area (Fisicaro et al., 2016). These effects are explained by the brain's plasticity and they demonstrate the neural cells' self-organization that is mainly driven by sensory information.

Similar to the visual cortex, also other cortical areas of the brain exhibit topographic arrangements which are driven by the sensory information (Arbib, 2003). In the somatosensory cortical area, for instance, the inputs received from receptors of different body regions are organized topographically into the so-called *somato-*

[1]Figure adapted from: https://blogs.aalto.fi/neuroscience/

topic maps (see Fig. 3.1). How the topography is developed and maintained has been the subject of a number of studies (Buonomano and Merzenich, 1998), which have identified synaptic plasticity and local excitation and inhibition, i.e., neural cells exciting the closest neighbors and inhibiting the more distant ones, as two necessary conditions for the development of the somatotopy. Other examples are the tonotopic map (Reale and Imig, 1980) in which the spatial order of the cell responses correspond to the pitch of acoustic frequencies of tones perceived, or the semantic space recently analyzed and visualized in details by a fMRI study conducted by Huth et al. (2012). This study showed that the brains of different individuals represent object and action categories in a common semantic space that is mapped smoothly onto the cortical sheet so that nearby points in cortex represent semantically similar categories.

Synaptic plasticity is a process that affects the strengths of synaptic connections between neurons during the learning process and plays an important role in the brain's adaptation. The most well-known theory describing the basic mechanisms of synaptic plasticity was first proposed by Donald Hebb in 1949 (Hebb, 1949), postulating that simultaneous activation of two neurons leads to an increase in synaptic strength between them. This learning mechanism is at the core of most computational models (Floreano and Mattiussi, 2008). Hebb's rule implies strong locality of the plasticity since the modification of the synapses depends only on the presynaptic and the postsynaptic neurons. It also introduces the concept of activity-induced reinforcement or weakening of the synapses. There are several other neurophysiological processes that contribute to the brain's adaptation such as the neurogenesis (Boldrini et al., 2018; Sorrells et al., 2018), the growth and death of the connections and the molecular modifications of the neuron membrane (Tierney and Nelson III, 2009). However, these processes are less understood than the activity-dependent synaptic changes and are thus less frequently incorporated in computational modules of neural systems.

Attempts for modeling the brain maps date back in the 1970s with the work from Von der Malsburg (1973) and Grossberg (1976) who formulated biologically plausible models capable of self-organizing in an unsupervised environment. In these models, the emergence of feature-sensitive cells was implemented by the so-called *competitive learning*, i.e., the adaptation of the strongest activated cells to the given input made them become tuned to specific input features. These ideas were later embodied into the best-known and most popular model of self-organizing networks, namely the self-organizing maps proposed by (Kohonen, 1982). During training, SOMs build a neural map through the so-called *vector quantization*, a

process that finds prototype vectors for encoding a submanifold of the input space. Through such process, the network learns topological relations of the input space in an unsupervised manner.

3.2 Competitive Vector Quantization

Vector Quantization (VQ) is a quantization technique that models a probability density function through a finite set of prototype vectors which are often referred to as code-vectors, and the set of code-vectors is called a codebook. The standard vector quantization was introduced by Dirichlet (1850) with the so-called Dirichlet tessellation in two- and three-dimensional spaces and by Voronoi in spaces of arbitrary length (Voronoï, 1908). The idea behind vector quantization is to partition the input space into a finite number of regions, called Voronoi regions, and to find an optimal prototype vector for each region. In the end, the prototype vectors should represent the data as faithfully as possible. If we consider data samples from the Euclidean space, the goal would be to compute prototypes with the smallest Euclidean distance from the input.

Competitive VQ (Biehl et al., 2016) is a very basic scheme for unsupervised VQ which employs the concept of competitive learning (Rumelhart and Zipser, 1985), i.e., in each learning iteration the prototypes compete for updates. So, given a set of P data vectors $\{\mathbf{x}_i \in \mathbb{R}^N\}, i = 1, 2, ...P$, and a set of prototypes with the same dimensionality $W = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K\}$, the learning of the competitive VQ method is guided by a cost function called quantization error, defined as:

$$H_{VQ} = \sum_{i=1}^{P} \frac{1}{2} ||\mathbf{w}_b - \mathbf{x}_i||^2, \tag{3.1}$$

where $\mathbf{w}_b \in \mathbb{R}^N$ denotes the prototype with the smallest Euclidean distance from the input vector $\mathbf{x}_i \in \mathbb{R}^N$:

$$||\mathbf{w}_b - \mathbf{x}_i|| \leq ||\mathbf{w}_j - \mathbf{x}_i||, j = 1, 2, ..., K. \tag{3.2}$$

In words, the quantization error is the sum of the distances of all individual input vectors from their respective closest prototype. The competitive VQ follows a stochastic gradient descent approach for the minimization of the cost function H_{VQ}. At each time step, a single input vector $\mathbf{x}_i$ is randomly selected and the closest prototype vector, or *winner*, $\mathbf{w}_b$ is computed following Eq. 3.2. Then, the

winner prototype is updated according to the equation:

$$\Delta \mathbf{w}_b = \epsilon(t) \cdot (\mathbf{x}_i - \mathbf{w}_b), \tag{3.3}$$

where the learning rate $\epsilon(t) < 1$ controls how much the prototypes are updated. As in any stochastic gradient descent approach, the convergence of the prototype vectors is guaranteed by employing a time-dependent learning rate, initially set to an arbitrary value for then slowly approaching zero in the course of training (Robbins and Monro, 1985).

Typically, the prototypes are placed at randomly selected data points and are then moved during learning into regions with the highest density of the input space. However, the training outcome is very sensitive to the prototype initialization. If, for instance, the prototypes are initially placed in empty regions of the input space, they may never be identified as the winner of any data point, thus remaining unchanged during the whole training process. For this reason, the concept of neighborhood cooperativeness between prototypes was introduced by Kohonen as part of his biologically motivated model, the self-organizing maps.

3.3 The Self-Organizing Map

The Kohonen's Self-Organizing Map (SOM; Kohonen (1990)) was originally proposed as a biologically inspired model of the brain maps. In the SOM, the prototype vectors are neurons assigned with weights of the same dimensionality as the input space. Topological relations are imposed between neurons. Typically, their position is arranged in a two-dimensional grid, or so-called *neural lattice* A. Unlike the VQ method, the update of the weights affects not only the winner neuron, i.e., the neuron with the smallest Euclidean distance from the current input data sample but also the neurons in its immediate neighborhood defined by a Gaussian function:

$$h^{SOM}(t) = \exp\left(-\frac{||b - r||_A^2}{2\sigma(t)^2}\right), \tag{3.4}$$

where $||b - r||_A^2$ is the Euclidean norm, b and r are the positions of the winner neuron and its neighbor in the lattice A, and $\sigma(t)$ limits the neighborhood range. Then, all prototype neurons are updated according to:

$$\Delta \mathbf{w}_r = \epsilon(t) \cdot h^{SOM}(t) \cdot (\mathbf{x}_i(t) - \mathbf{w}_r). \tag{3.5}$$

Thus, all neurons within the defined neighborhood range are updated, though to a lesser extent. The learning rate $\epsilon(t)$ is a monotonically decreasing function of time between $[0, 1]$, for instance, the exponentially decreasing function defined as:

$$\epsilon(t) = \epsilon_0 \left(\frac{\epsilon_T}{\epsilon_0} \right)^{\frac{t}{T}}. \tag{3.6}$$

The neighborhood radius $\sigma(t)$ is a monotonically decreasing function of t as well. The values are fairly large at the beginning of the learning process in order to develop the rough topological ordering of the prototype vectors and are then gradually reduced to allow for the convergence towards optimal values.

The key feature of the SOM is that it provides a low-dimensional, topology-preserving representation of the input space in the following sense: input samples with small Euclidean distance are mapped either to identical or adjacent neurons on the map. Similarly, neurons which are neighbors in the map correspond to prototype vectors with a small Euclidean distance in the input space. This constitutes a powerful tool for the visualization of high-dimensional data and can be extended with posterior labeling for being employed in classification and regression tasks. However, a main drawback of the SOM is the fixed topology that may limit the resulting mapping accuracy. For more details on this matter and examples of applications and different implementations, we refer the reader to a recent review of the SOM by Kohonen (2013).

Neural Gas

Inspired by Kohonen's approach, Martinetz and Schulten (1991) developed a VQ scheme with neuron neighborhood cooperativeness, called the Neural Gas (NG) algorithm. Similarly to the SOM, the NG consists of a competitive layer with a fixed number of neurons that must be defined a priori. However, the network's structure is not fixed and adapts to the input data distribution through the learning process.

The main idea of the NG is to update several prototypes at a time, not based on their vicinity to the winner prototype neuron defined by the network's structure, but according to their rank with respect to the distance from the given sample. At each time step t, an input vector $\mathbf{x}$ is randomly selected from the dataset and the prototype neurons are ordered based on the distance to the given sample and are assigned a rank k. Then, each neuron $W = \{\mathbf{w}_j, j = 1, ..., K\}$ is adapted according

to:

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \epsilon \cdot \exp\left(-\frac{k_j}{\lambda}\right) \cdot (\mathbf{x} - \mathbf{w}_j), \tag{3.7}$$

where ϵ is the learning rate, the exponential function represents the neighborhood function which decreases with increasing ranks, and λ determines the range of the neighborhood with respect to the prototype ranks, thus defines the prototypes with significant updates. The learning rate ϵ and the range λ decrease with increasing t.

After a sufficient number of adaptation steps, the neurons will cover the data space with a minimum representation error. In fact, since the structure of the network is not constrained by a fixed topology, the NG has been shown to minimize the quantization error. However, the algorithm requires the number of neurons to be chosen a priori and cannot be changed over time. Depending on the relationship between inherent data dimensionality, some information on the topological arrangement on the input data may be lost when being mapped.

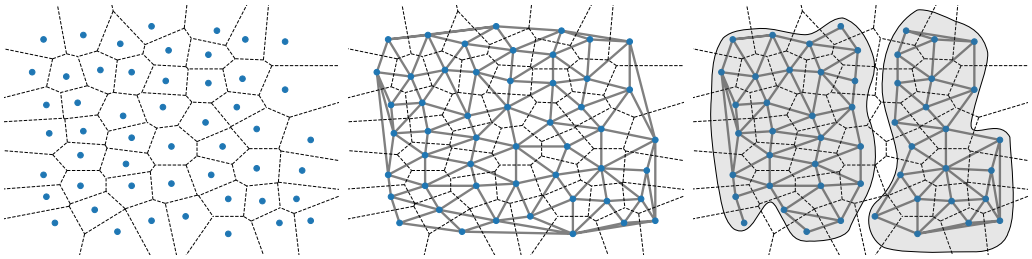
3.4 Growing Self-Organizing Networks

Growing networks address the limitations of the so far described models by creating (or removing) neurons to support the correct formation of topological maps. Models like the Growing Neural Gas (GNG) and the Growing When Required (GWR) network have the ability to incrementally add neurons based on representation errors and preserve the input's topology through the learning process by applying the Competitive Hebbian Learning (CHL) method (Martinetz, 1993). These models have been successfully applied for clustering human motion patterns in terms of multi-dimensional flow vectors (Parisi et al., 2014, 2015) as well as for learning object representations without supervision (Donatti et al., 2010). In this section, we provide a comparison between the GNG and the GWR models.

Competitive Hebbian Learning

As mentioned in Section 3.2, the standard vector quantization procedure partitions the input manifold $V \subset \mathbb{R}^N$ into a number M of Voronoi regions. These regions are defined as:

$$V_i = \{\mathbf{v} \in V : ||\mathbf{v} - \mathbf{w}_i||^2 \leq ||\mathbf{v} - \mathbf{w}_j||^2 \quad \forall i \neq j\}, \tag{3.8}$$



(a) (b) (c)

Figure 3.2: (a) The Voronoi tessellation. The dashed lines delineate the Voronoi regions for the given data points. (b) The solid lines are the edges of the Delaunay triangulation, connecting points with a neighboring Voronoi region. (c) The induced Delaunay triangulation. Edges exist only in areas where $P(\mathbf{x}) > 0$.

where all vectors in V have a distance to $\mathbf{w}_i$ not greater than their distance to $\mathbf{w}_j$. The set $\{V_i\}_{i=1}^{M}$ forms a partition of V and is known as a Voronoi tessellation or the Voronoi diagram of V, while the points $\{\mathbf{w}_i\}_{i=1}^{M}$ are called generating points or centers. The geometric dual, the Delaunay triangulation is a graph in which the nodes are the generating points and points of adjacent Voronoi regions are connected by an edge. These two closely related data structures are one of the most fundamental data structures in *computational geometry* (Aurenhammer, 1991).

The CHL as proposed by Martinetz (1993) provides a way to generate Delaunay triangulation graphs from a given set of centers. The principle of this method is:

> *For each input signal $\boldsymbol{x}$, an edge is inserted between the two closest nodes, measured by the Euclidean distance.*

The resulting graph is a subgraph of the Delaunay triangulation, called the "induced Delaunay triangulation". In this subgraph, two centers are only connected if the common border of their Voronoi regions lies at least partially in the input space, i.e., where $P(\mathbf{x}) > 0$. The induced Delaunay triangulation has been shown to preserve the input topology in a very general sense (Martinetz, 1993).

The idea underlying the GNG algorithm when originally proposed by Fritzke (1995) was the combination of the Neural Gas model, used for the purpose of vector quantization, i.e., for defining the centers of the Voronoi regions, with the CHL principles. In fact, the CHL is the essential component of both the GNG and the GWR networks since it is used for creating network edges which guide the local adaptation of the nodes. Additionally, the incremental networks adopt an edge-aging mechanism for the removal of the edges between nodes not belonging anymore to adjacent Voronoi regions after several updates. For this, each edge is associated with an *age* which can be incremented during learning. Those edges

with an age exceeding a predefined threshold are removed and nodes with no connections, i.e., isolated nodes, are removed consequently.

Growing Neural Gas

The GNG network starts with a set of $N = 2$ neurons in the input space. At each learning iteration, the algorithm is given a random input $\mathbf{x}(t)$ drawn from the input distribution $p(\mathbf{x})$. The closest neuron b and the second closest neuron s in N are computed as follows:

$$\begin{aligned} b &= arg \min_{n \in N} ||\mathbf{x}(t) - \mathbf{w}_n||, \\ s &= arg \min_{n \in N/\{b\}} ||\mathbf{x}(t) - \mathbf{w}_n||, \end{aligned} \tag{3.9}$$

and if the connection (b, s) does not exist, it is created. If the edge exists already, its age is reset to zero. The local quantization error of b is updated by $\Delta E_b = ||\mathbf{x} - \mathbf{w}_b||^2$. The weight vector of the first best-matching unit, $\mathbf{w}_b$, and the weights of all the topological neighbors of b, $\mathbf{w}_i$, are moved towards the input $\mathbf{x}$:

$$\begin{aligned} \Delta \mathbf{w}_b &= \epsilon_b \cdot (\mathbf{x}(t) - \mathbf{w}_b), \\ \Delta \mathbf{w}_i &= \epsilon_i \cdot (\mathbf{x}(t) - \mathbf{w}_i), \end{aligned} \tag{3.10}$$

where the learning rates are such that $\epsilon_i < \epsilon_b$. This means that the neighbors are updated to a lesser extent compared to the winner neuron.

If the number of learning iterations is a multiple of a predefined parameter λ, a new neuron is created halfway between the neuron with the largest accumulated error and its topological neighbor with the largest accumulated error. The connection-age-based mechanism takes care of removing rarely used connections and neurons without connections as a consequence. The algorithm stops when a criterion is met, i.e., some performance measure, network size, or a given number of training epochs.

The λ parameter has a significant impact on the performance of the algorithm. Setting the parameter low will results in a poor initial distribution of the nodes and the accumulated local error will be badly approximated in the first learning iterations. A higher λ, on the other hand, results in a slower network's growth and requires the algorithm to run for many iterations in order to achieve a good node distribution. The fixed value of the λ parameter leads to a constant neural growth which inhibits the network to adapt to rapidly changing distributions. For this reason, the author proposed the GNG with Utility Factor (GNG-U), which

relocates less useful nodes during training (Fritzke, 1997). However, this adds another parameter to the algorithm which, when not selected carefully, affects significantly the network's behaviour (Holmström and Gas, 2002).

Growing When Required

Unlike the GNG which creates new neurons at a fixed growth rate, the GWR algorithm proposed by Marsland et al. (2002) creates a new node whenever the network's activation is not sufficiently high. The amount of network activation at time t is computed as a function of the distance between the current input $\mathbf{x}(t)$ and its best-matching unit $\mathbf{w}_b$:

$$a(t) = \exp(-||\mathbf{x}(t) - \mathbf{w}_b||). \tag{3.11}$$

New neurons are added when the activity of the best-matching unit is not higher than a predefined insertion threshold a_T. In order to handle both the creation of new neurons as well as their adequate training, the GWR algorithm adopts a mechanism for measuring how often each neuron has fired. This firing rate is initially set to one and then decreases every time a neuron is trained in the following way:

$$\Delta h = \tau \cdot \kappa \cdot (1 - h) - \tau, \tag{3.12}$$

where τ and κ are constants controlling the behavior of the decreasing curve. These constants are set in a way to reduce faster the firing counter of the winner neuron than of its neighbors. The firing rate is considered also during the update step of the network. The position of the winner neuron and its neighbors are moved towards the input $\mathbf{x}(t)$:

$$\begin{aligned} \Delta \mathbf{w}_b &= \epsilon_b \cdot h_b \cdot (\mathbf{x}(t) - \mathbf{w}_b), \\ \Delta \mathbf{w}_i &= \epsilon_i \cdot h_i \cdot (\mathbf{x}(t) - \mathbf{w}_i). \end{aligned} \tag{3.13}$$

The use of the activation threshold and firing counters to modulate the growth of the network leads to create a larger number of neurons at early stages of the training process. Afterward, the created neurons are fine-tuned to the input data through subsequent training epochs. This makes the GWR algorithm more suitable than the GNG algorithm for learning representations of non-stationary datasets and for the incremental learning of sensory data since neurons are created immediately to represent the new region in the input space. The GWR algorithm iterates over the given data until a given stop criterion is met, e.g., a maximum network size or a

maximum number of iterations. The learning algorithm for GWR is illustrated in Appendix B.

The standard GNG and GWR learning algorithms do not account for temporal sequence processing. Therefore, there is a motivation to extend these networks while preserving desirable learning properties such as computational efficiency and network convergence.

3.5 Self-Organizing Networks for Temporal Sequences

In all the methods discussed so far the network's response and the activation values computed at time step t are based only on the input at time step t. The temporal aspect of the input data does not play a role during learning, thus such methods are not suitable for temporal sequence processing. In such cases, the goal would be to inspect an input signal at a given time step, taking into account its temporal context. For a topology-preserving algorithm, this means that input signals mapped to neighboring neurons should have both the value $x(t)$ and their temporal context similar. There have been quite a few extensions proposed for the SOMs, the GNG, and the GWR networks in order to learn spatiotemporal dynamics of the input. We will review here only the methods which are relevant for understanding our work.

3.5.1 Delay Embedding

One method for handling temporal sequences with a self-organizing network is by embedding time into the input data samples at each learning iteration. The delay-embedding technique, also often referred to as the *sliding time window* technique, comprises the specification of a temporal window over the input sequence and the concatenation of the corresponding successive items into a single vector of higher dimensionality. The advantage of this technique, being simply a pre-processing of the input data, is that it preserves all the network properties of the self-organizing algorithms. On the other hand, it may demand a higher computational effort because of the increase of the input dimensionality, which slows the network's training. However, such a strategy has been shown to produce very good results in recognition tasks because the relevant temporal information is always explicitly available (Parisi et al., 2014, 2015).

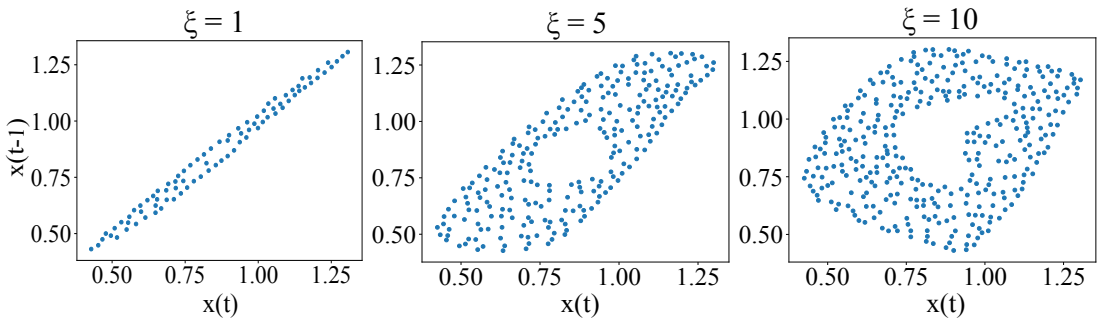


Figure 3.3: Distribution of the representations learned with a GWR network for the Mackey-Glass time series. From left to right, the values of the time window size and the lag parameter have been set to $q = \{2, 10, 20\}$ and $\xi = \{1, 5, 10\}$ respectively.

Referring to Takens's embedding theorem (Takens, 1981), it is possible to reproduce entirely the properties of a deterministic dynamical system starting from a series of observations. The embedding technique consists of grouping equidistant observations $\mathbf{x} = (\mathbf{x}_i)_{i=1,...,N}$ into vectors of dimension q such that:

$$\psi_i(\mathbf{x}) = \{\mathbf{x}_i, \mathbf{x}_{i-\xi}, ...\mathbf{x}_{i-(q-1)\xi}\}, i \in [q, N], \tag{3.14}$$

where q is the width of the time window and ξ is the so-called *time delay* or the *lag* parameter. Both q and ξ are data-dependent and are chosen in order to achieve a good input reconstruction. In general, the choice of the lag is selected in order to maximize the independence of the components in $\psi_i(\mathbf{x})$, but still keeping the value of ξ small. The lag parameter has been found to play an import role in obtaining meaningful spatiotemporal clusters as well (Simon et al., 2006).

An example for the processing and learning of continuous temporal sequences is the well-known Mackey-Glass time series described by the differential equation $\frac{dx}{d\tau} = bx(t) + \frac{ax(\tau-d)}{x(t-\tau)^10}$, using $a = 0.2$, $b = -0.1$, $d = 17$. To demonstrate the effects of the two delay embedding parameters, we train a GWR network on the series after being pre-processed following Eq. 3.14. We use three different parameter settings: $q = \{2, 10, 20\}$ and $\xi = \{1, 5, 10\}$, while keeping the learning parameters for the GWR algorithm unchanged. The distribution of the learned spatiotemporal representations is illustrated in Fig. 3.3. As can be seen from the figure, the greater the lag parameter ξ, the more spreads the distribution of the representations in the time series domain. Moreover, with the increasing value of each parameter, the GWR algorithm increases the number of generated neurons $N = \{80, 250, 340\}$ in order to cover the input space sufficiently.

Neuron Activation Trajectories

Another method for learning spatiotemporal dependencies without affecting the self-organizing network's learning dynamics is to train the network without considering the temporal dimension and then post-process the outputs with the so far described time window technique. One popular method of this type is the so-called *trajectory-based* SOM (Kohonen, 1988) which takes into account temporal relations among succeeding best-matching units. This means that for fixed time intervals of width q, the best-matching units are computed and their position is recorded on the map. Then, the spatiotemporal representations are built by concatenating the subsequent best matches of each temporal interval into single vectors:

$$\psi_i^{SOM}(\mathbf{x}) = \{b(\mathbf{x}_i), b(\mathbf{x}_{i-1}), ...b(\mathbf{x}_{i-(q-1)})\}, i \in [q, N], \tag{3.15}$$

where $b(\cdot)$ indicates the position of the winner neuron matching the input in each time step t. These spatiotemporal vectors can be visually illustrated as neuron activation paths or trajectories in the map, hence the name of this method.

The activation trajectory strategy cannot be applied in the current form to the growing self-organizing networks, for instance, to the GWR, due to the fact that the topological arrangement of the neurons is dynamic and changes during the learning process. So instead of the neurons' position, it is possible to concatenate the weight vectors associated with consecutively activated neurons:

$$\psi_i^{GWR}(\mathbf{x}) = \{\mathbf{w}_{b(\mathbf{x}_i)}, \mathbf{w}_{b(\mathbf{x}_{i-1})}, ...\mathbf{w}_{b(\mathbf{x}_{i-(q-1)})}\}, i \in [q, N]. \tag{3.16}$$

Trajectories of the neural activations from one network can be used as input for the training of a subsequent network in a self-organizing multi-layer architecture. In this way, it is possible to obtain neurons coding progressively increasing spatiotemporal dependencies of the input. This hierarchical learning strategy has been shown to produce very good results in the human body pose and motion processing and classification (Parisi et al., 2015).

3.5.2 Recurrent Connections

The temporal dynamics of an observed signal can be taken into account by a self-organizing network during local node adaptations as well. This can be obtained, for instance, by introducing recurrent connections, whereby past network outputs are fed back into the network and contribute to the current network's activation at each learning iteration. Early models often relied on leaky integrators, such as

the so-called temporal Kohonen maps (TKM; Chappell and Taylor (1993)) or the recurrent SOM (Varsta et al., 1997). More recent models incorporate an explicit representation of the temporal context. Such a context is attached to every neuron and is learned in a similar way as the neuron weights itself. Examples for this principle include Merge SOM (Strickert and Hammer, 2005), Merge NG (Strickert and Hammer, 2003), Merge GNG (Andreakis et al., 2009), γ-SOM (Estévez and Hernández, 2009), γ-NG (Estevez et al., 2011), γ-GNG (Estévez and Vergara, 2013), and γ-GWR (Parisi et al., 2017a). These methods differ in the way in which the temporal context is represented but rely on a similar treatment of the temporal dynamics of the input signal.

In general, the performance of the gamma models is better than those of the merge models with respect to the temporal quantization error metric (Voegtlin, 2002). This is due to the fact that for the gamma models the temporal context is extended in order to equip each neuron with an arbitrary number of context descriptors leading to an increase in memory depth and temporal resolution. In the gamma models, each neuron is equipped with a weight vector $\mathbf{w}_i$ and a set of context descriptors $C = \{c_1^i, c_2^i, ..., c_K^i\}, k = 1, ..., K$, where K is the Gamma filter order. The computation of the winner neurons in a network is as follows:

$$d_i(t) = \alpha_\omega \cdot ||\mathbf{x}(t) - \mathbf{w}_i||^2 + \sum_{k=1}^{K} \alpha_k \cdot ||\mathbf{C}_k(t) - \mathbf{c}_k^i||^2, \tag{3.17}$$

$$\mathbf{C}_k(t) = \beta \cdot \mathbf{c}_k^{I_{t-1}} + (1 - \beta) \cdot \mathbf{c}_{k-1}^{I_{t-1}} \quad \forall K = 1, ..., K, \tag{3.18}$$

where $\alpha, \beta \in (0; 1)$ are constant values that modulate the influence of the current input and the past, and $\mathbf{c}_0^{I_{t-1}} = \mathbf{w}^{I_{t-1}}$ with random $\mathbf{c}_k^{I_0}$ at $t = 0$. Both depth and temporal resolution are modulated by the value of β. The depth measures how far into the past the internal memory stores information and the resolution indicates the degree to which the information carried from each individual element of the input sequence is preserved. When using a $K = 1$, this approach reduces to the learning mechanism of the merge models.

Estévez and Vergara (2013) provide an extensive nonlinear time series analysis with the γ-GNG, showing that their model builds some kind of delay embedding using Gamma filters instead of delay coordinates. However, the model needed a careful selection of the β parameter and of the number of context descriptors K. The parameter selection was carried out through a grid search for the minimization of the temporal quantization error on a given dataset, followed by picking the model with the maximum mutual information from the first 10 results.

3.6 Summary

Computational models for self-organization constitute a highly attractive and versatile framework for the unsupervised learning of complex and potentially high-dimensional data. The conceptual simplicity of the models has allowed the development of a number of extensions and efficient training schemes for dealing with static input as well as temporal sequences. In addition to improving the understanding of cortical map organization via the development of simplified computational models, self-organizing networks have been successfully applied to a large number of tasks, from simple data analysis to more complex ones such as the recognition of human action sequences from multiple visual and auditory cues. In particular, growing self-organizing networks have been an effective model for clustering human motion patterns in terms of multi-dimensional flow vectors (Parisi et al., 2014, 2015) as well as for learning object representations without supervision (Donatti et al., 2010). The generative nature of this type of networks makes them particularly suitable for the task of learning human-object interactions when considering a possible generalization towards unseen action-object pairs.

One important strength of the self-organizing networks and, in general, of prototype-based systems is their flexibility with respect to the choice of similarity metrics. The Euclidean distance function takes into account all features with the same weight, however, depending on the nature of the problem at hand, other choices may be more suitable. A weighted Euclidean distance, for instance, is a good alternative in case the feature vector is an integration of different sensory sources or of different properties, e.g., the body motion expressed as three-dimensional joints in space and the identity of the manipulated objects during the perception of human-object interactions. If we want to give both data the same importance during distance computation, a weighted Euclidean distance must be applied as we will see in Chapter 6. The neural insertion criteria in the growing self-organizing models is another versatile component that can be adapted to the task at hand. For instance, if data labels are available, they can be used to optimize internal neural representations as we will see in Chapter 7.

So far, computational models of self-organization have shown their applicability in several high-level cognitive functions such as human action recognition and multi-modal perception (Parisi et al., 2016b). In the next chapters, we propose a set of learning architecture for the recognition of human-object interaction scenarios where the integration of visual context plays a decisive role in achieving a good performance. Furthermore, we show how simple modifications of the computational steps in a hierarchical GWR algorithm can be implemented for online

motion learning and prediction in robotic scenarios. We show how Hebbian learning of inter-modular neural connections can create an architecture sensitive to the temporal order of action segments and thus applied for action prediction. Finally, we show how hierarchical arrangements of GWR networks with different temporal resolutions, equipped with top-down modulation mechanisms, can contribute to the emergence of hierarchical action representations from visual input.

Chapter 4

Learning Human-Object Interactions with a Self-Organizing Architecture

4.1 Introduction

The recognition of transitive actions, i.e., actions that involve the interaction with an object, represents a key function of the human visual system that fosters learning and social interactions. Given the outstanding capability of humans to infer the goal of actions from the interaction with objects, the biological visual system represents a source of inspiration for developing computational models. The ability of computational approaches to reliably recognize human-object interactions can establish an effective cooperation between assistive systems and people in real-world scenarios, promoting learning from demonstration in robotic systems (Prevete et al., 2008; Tessitore et al., 2010).

From the computational perspective, an important question arises regarding the potential links between the representations of body postures and manipulated objects and, in particular, how these two representations interact and can be integrated. As discussed in Section 2.1, the information about body pose and objects are processed separately and reside in distinct areas of the human visual cortex (Beauchamp et al., 2002; Downing and Peelen, 2011; Grill-Spector, 2013). Neuroscientists have widely studied object and action perception, with a focus on where and how the visual cortex constructs invariant object representations (Hubel and Wiesel, 1962) and how neurons in the superior temporal sulcus (STS) area encode actions in terms of patterns of body posture and motion (Grossman and Blake,

2002; Giese and Poggio, 2003). As discussed in Section 2.1.2, the identity of the objects plays a crucial role for the complete understanding of human-object interactions (Saxe et al., 2004) and modulates the response of specific action-selective neurons (Gallese et al., 1996; Nelissen et al., 2005; Yoon et al., 2012). Yet, little is known about the exact neural mechanisms underlying the integration of actions and objects.

In this chapter, we propose a self-organizing neural architecture that learns to recognize human-object interactions from videos in real time. The design of the proposed architecture relies on the following assumptions:

- The visual features of body pose and man-made objects are represented in two distinct areas of the brain (Downing and Peelen, 2011; Grill-Spector, 2013; Beauchamp et al., 2002) (see Section 2.1).
- Input-driven self-organization defines the topological structure of specific visual areas in the brain (Miikkulainen et al., 2006) (see Section 3.1).
- The representations of object and action categories are based on prototypical examples. Prototype-based learning has its counterpart in cognitive psychology, which hypothesizes that a category is represented by a number of representatives and class membership is based on resemblance (Rosch and Mervis, 1975).
- The identity of the objects is crucial for the understanding of actions performed by other individuals (Saxe et al., 2004; Gallese et al., 1996).

We develop a hierarchical architecture with the use of growing self-organizing networks, namely the Growing When Required (GWR) network (Marsland et al., 2002), to learn prototypical representations of actions and objects and the resulting action-object mappings in an unsupervised fashion. The architecture consists of two network streams processing separately feature representations of body postures and manipulated objects. A second layer, where the two streams are integrated, combines the information in a self-organized manner for the development of action–object mappings. The visual identification and segmentation of the body pose from RGB videos are challenging due to the spatial transformations compromising the appearances, such as translations, the difference in the point of view, changes in ambient illumination, and occlusions. For this reason, we consider three-dimensional body skeletal representations, which are the most straightforward way of achieving invariance to the subjects' appearance and body size, for instance, through normalization. Moreover, three-dimensional articulated body

pose and motion in real-world can be easily obtained through widely available low-cost depth sensor technologies, such as the Asus Xtion cameras.

We evaluate our architecture with a dataset of RGB-D videos containing daily actions, which will be introduced in Section 4.2.1, acquired for the purpose of this study. In particular, we look into the role of the objects' identity as contextual information for distinguishing between different activities, the classification performance of our architecture in terms of recognition of human-object interaction activities, and the response of the network when fed with congruent and incongruent action-object pairs. Furthermore, we provide an evaluation of the performance of our architecture with respect to the state of the art in the transitive action recognition from RGB-D data with experiments on a publicly available benchmark dataset CAD-120 (Koppula et al., 2013). The actual body pose feature extraction and the segmentation and representation of the objects are described in Section 4.3. We present and discuss our results on both datasets in Section 4.5.

4.2 Datasets

In this chapter, we will provide experimental results on two datasets for the recognition of human-object interactions from RGB-D videos. Before analyzing the long human activities composing the CAD-120 benchmarking dataset, we will focus on a smaller scale dataset which we have acquired for the purpose of this study, namely the Transitive Actions dataset. The reason for this is twofold. First, running experiments with a cleaner and more controlled dataset allows us to understand the learning properties as well as the limitations of the proposed architecture. Recent studies from Torralba and Efros (2011) show that the evaluation of new algorithms on uncontrolled data collections (i.e., "in the wild") creates biased results and limits its progress. Second, by providing a dataset with low inter-class variability, e.g., *eating* and *drinking*, we can study the role of the object recognition module which is part of our hierarchical architecture.

4.2.1 The Transitive Actions Dataset

We collected a dataset of the following daily activities: *picking up* (an object), *drinking* (from a container like a mug or a can), *pouring* (from can to mug), *eating* (an edible object like a cookie), and *talking on the phone* (Fig. 4.1). The data collection was planned having in mind the role of the objects' identity in distinguishing the actions, in particular when the sole body motion information may not

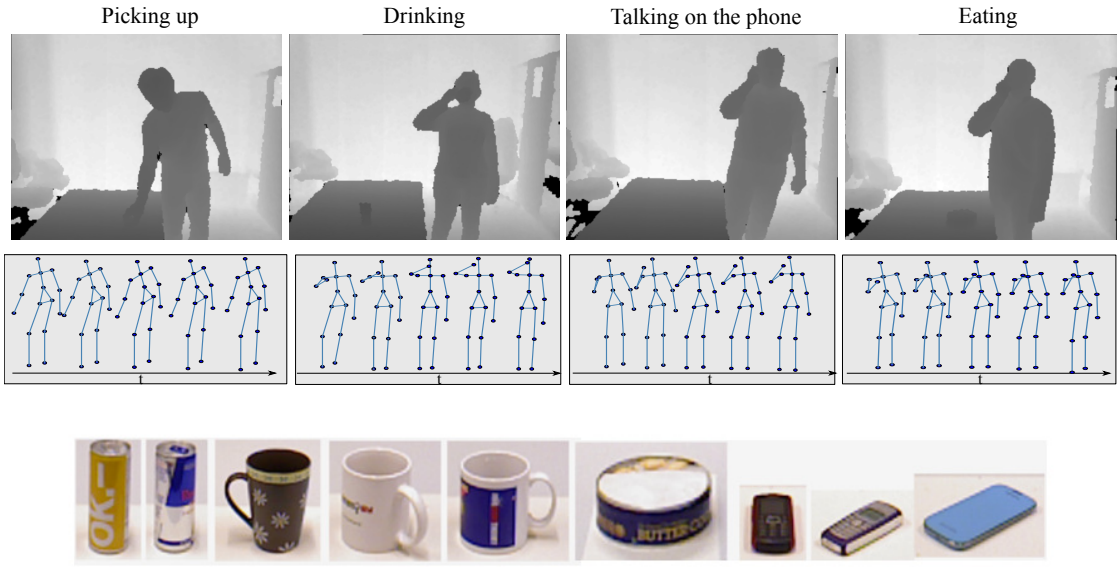


Figure 4.1: Examples of sequences of skeleton joints and objects taken from the Transitive Actions dataset. The object category labels are: *can*, *mug*, *biscuit box* and *phone*.

be sufficient to unequivocally classify an action. The actions were performed by 6 participants that were given no explicit indication of the purpose of the study nor instructions on how to perform the actions. The average duration of each action is of ≈ 75 frames corresponding to 2.5 seconds. The dataset was collected with an Asus Xtion depth sensor that provides synchronized RGB and depth frames at a frame rate of 30 Hz. The distance of each participant from the sensor was not fixed but maintained within the maximum range for the proper functioning of the depth sensor, i.e., 0.8 - 3.5 meters. The tracking of the skeleton joints was provided by the OpenNI framework[1]. To attenuate noise, we computed the median value for each body joint every 3 frames resulting in 10 joint position vectors per second. We added a mirrored version of all action samples to obtain invariance to actions performed with either the right or the left hand. Action labels were then manually annotated.

While the body segmentation and skeletal representations are automatically provided by tracking frameworks like OpenNI, the extraction of the objects' information requires additional computational steps. We chose the point-cloud-based table-top segmentation[2], which is a simple but effective method operating on a 3D representation of the scene (Aldoma et al., 2012; Rusu et al., 2009) and is

[1]OpenNI/NITE: http://www.openni.org/software
[2]Point Cloud Library: http://www.pointclouds.org/

commonly used in the robotics community. The position of the 3D points is given from the depth maps, whereas the color of each point can be extracted by the intensity of the corresponding pixel in the RGB image. The segmentation method is based on the extraction of a dominant scene plane, e.g., a table or the floor, and a Euclidean clustering step applied on the remaining points after the plane removal in order to obtain the objects' hypotheses. The clustering step is guided by a threshold, which indicates how close two points are required to be to belong to the same object. Therefore, for a successful segmentation, the method requires different objects to be standing from each other at a distance higher than the predefined threshold. After the clusters were individualized, we extracted the RGB region of the corresponding objects. The main assumption of the table-top segmentation method is the presence of an identifiable surface like a table with the objects standing on it. However, this type of scene configuration is quite common for human daily activities, e.g., having meal, and these activities are the focus of this dataset and of this chapter in general. In case false positives were obtained through the automatic segmentation, they were manually deleted. The obtained object images compose the training data for the object recognition module of the proposed architecture.

4.2.2 The CAD-120 Dataset

The Cornell Activity Dataset CAD-120 is an RGB-D benchmarking dataset containing object interactions, which has been collected and made publicly available by the Cornell University. This dataset consists of a total of 120 videos containing 10 long daily activities: *arranging objects*, *cleaning objects*, *having meal*, *making cereal*, *microwaving food*, *picking objects*, *stacking objects*, *taking food*, *taking medicine* and *unstacking objects*. These activities are performed by four different subjects (two males, two females and, of these four, one left-handed) repeating each action three to four times. The dataset has a total of 61.585 RGB-D video frames and the average duration of the activities is of ≈ 600 frames corresponding to 20 seconds. Similar activities are performed with different types of objects, e.g., the *stacking* and *unstacking* are performed with either pizza boxes, plates or bowls. Each video is annotated with the human skeleton tracks and the position of the manipulated objects across frames.

Sample images from the CAD-120 are shown in Fig. 4.2. This dataset provides significant variations in the way the subjects perform the activities and the scenes contain significant background clutter. In addition, subjects are partially occluded in different scenarios and not facing the camera. All these conditions can be

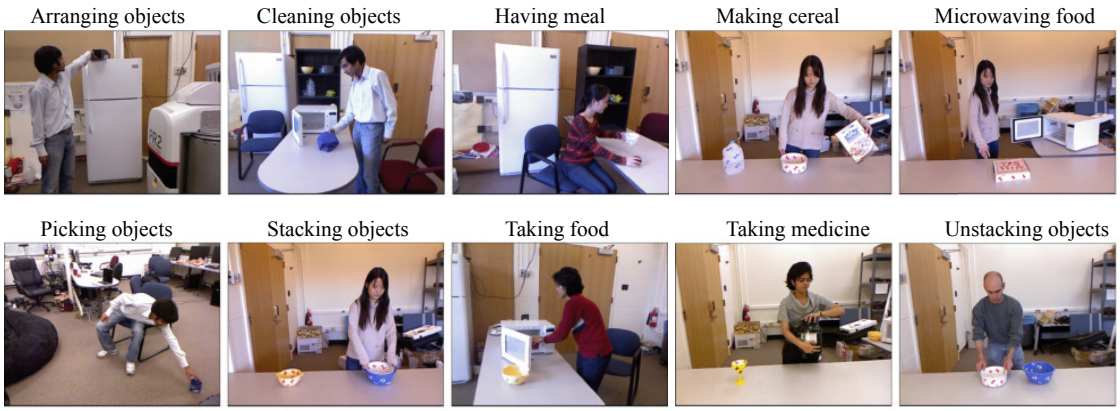


Figure 4.2: Examples of high-level activities from the CAD-120 dataset (Koppula et al., 2013)[3].

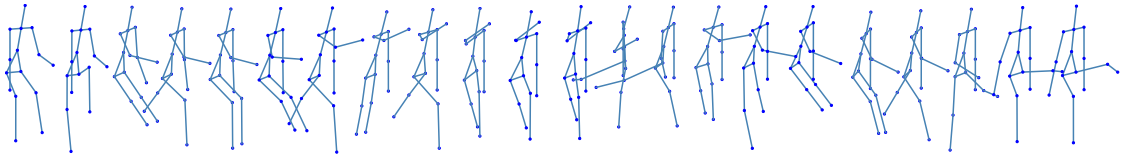


Figure 4.3: A skeleton sequence representing a seated person having meal (eating and drinking): the legs and feet have a very high tracking noise in this position.

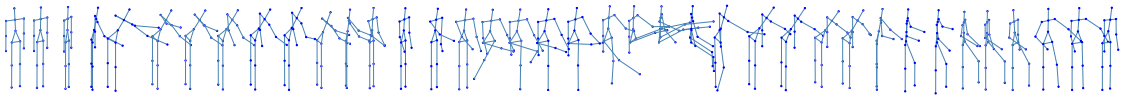


Figure 4.4: A skeleton sequence representing a person standing behind a table and microwaving food: the legs and feet have a very high tracking noise in this position due to not being visible.

challenging for the skeleton tracking algorithm yielding a noisy and often disrupted motion of the tracked joints (see Fig. 4.3 and 4.4). In particular, the tracking of feet and knee joints is mostly unreliable due to scenarios with subjects performing actions while sitting or standing behind objects. Thus, for our experiments, only the upper body joints are considered.

4.3 Feature Extraction

Both datasets introduced so far provide us with the three-dimensional body joint positions and the RGB images of the objects segmented from the scene. However, the processing of such information with our GWR-based architecture primarily requires a feature extraction step for both the body pose in order to achieve in-

[3]Images are taken from `http://pr.cs.cornell.edu/humanactivities/data.php`

variance to the translation scale and viewpoint, and the images of the manipulated objects in order to have a compact vectorial representation of objects discriminative enough for a successful classification.

4.3.1 Body Pose Features

We consider only the position of the upper body joints (*shoulders*, *elbows*, *hands*, center of *torso*, *neck* and *head*), given that they carry more significant information (than for instance the *feet* and *knee* joints) about the human-object interactions we focus on in this chapter (see Appendix C for the full list of joints provided by the OpenNI framework). However, the number of considered joints does not limit the application of our architecture for the recognition of full-body human-object interactions.

We extract the *skeletal quad* features (Evangelidis et al., 2014), which are invariant with respect to location, viewpoint as well as body-orientation. These features are built upon the concept of geometric hashing and have shown promising results for the recognition of actions and hand gestures. Given a quadruple of body joints $\{J_1, J_2, J_3, J_4\}$ where $J_i \in \mathbb{R}^3$, a local coordinate system is built by making J_1 the origin and mapping J_2 onto the vector $[1, 1, 1]^T$. The positions of the other two joints J_3 and J_4 are calculated with respect to the local coordinate system and are concatenated in a 6-dimensional vector $[\hat{j}_{3,1}, \hat{j}_{3,2}, \hat{j}_{3,3}, \hat{j}_{4,1}, \hat{j}_{4,2}, \hat{j}_{4,3}]$. The latter becomes the compact representation of the four body joints' positions. We empirically select two quadruples of joints: [*center torso, neck, left hand, left elbow*] and [*center torso, neck, right hand, right elbow*]. This means that the positions of the hands and elbows are encoded with respect to the torso center and neck. We choose the neck instead of the head position due to the noisy tracking of the head caused by occlusions during actions such as *eating* and *drinking*.

Composing such holistic body pose vectors, i.e., concatenations of joint positions, is quite convenient when employing a GWR network for the learning. In the case of missing joints in a data frame, due to, for example, noise or body occlusion, the best-matching unit for that input vector can be computed omitting the missing parts of the body pose vector. Self-organizing networks, such as SOMs and the GWR networks as their growing extension, are able to operate robustly in the case of missing values (Vatanen et al., 2015).

4.3.2 Object Features

The natural variations in RGB images such as variations in size, rotation, and lighting conditions, are usually so wide that objects cannot be compared to each

other simply based on the images' pixel intensities. For this reason, visual features are first extracted from the object images and subsequently encoded into compact vectorial representations in order to allow for an image comparison through vectorial metrics, such as the Euclidean distance function. What is important for our study is to have a classifier that generalizes to the objects' categories, despite of inter-class variations in shape and color. For this reason, we look into local appearance-based image descriptors to capture local shape convexities, for instance, the handle of a cup, or salient textures such as the keys in a keyboard etc. This can be obtained with the Scale-Invariant Feature Transform (SIFT) features which have been successfully applied to the problem of unsupervised object classification (Tuytelaars et al., 2010) and to learning approaches based on self-organization (Kinnunen et al., 2012). Moreover, SIFT descriptors are known to be, to some extent, robust to changes in illumination and image distortion.

In Lowe's original form (Lowe, 2004), the interest points are extracted from the grey-level image and then the image patches around each interest point are summarized through statistics of the local gradient directions of image intensities. When applying the SIFT descriptor to tasks such as object category recognition, experimental results have shown that better classification results are achieved by computing the SIFT descriptor over dense grids in the image domain as opposed to the sparse interest points obtained by the keypoint extractor. This improvement is explained by the fact that the descriptors computed over such a dense grid provide more information than the descriptors of a much sparser set of image points. Thus, we extract the dense SIFT features following the implementation provided by the VLFeat library[3]. In the dense SIFT features, the descriptors are of a fixed scale, thereby not accounting for the objects' scale variations between images. Therefore, multiple descriptors with four different window sizes are computed for each grid point on every image. The orientation of each dense SIFT descriptor is fixed and this relaxes the descriptors' invariance with respect to the object's rotation. With this kind of representation, we can train a *GWR* network and obtain neurons tuned to different object views, yet invariant to translation and scale.

We perform quantization followed by an image encoding step in order to have a fixed-dimensional vectorial representation of each object image. We apply the Vector of Locally Aggregated Descriptors (VLAD) (Jegou et al., 2012) encoding method (Fig. 4.5) which has shown higher discriminative power than the extensively used Bag of Visual Features (BoF) (Everingham et al., 2010; Szeliski, 2010) and has established itself as state of the art for the image retrieval problem (Arand-

[3]Dense SIFT from VLFeat library: http://www.vlfeat.org/

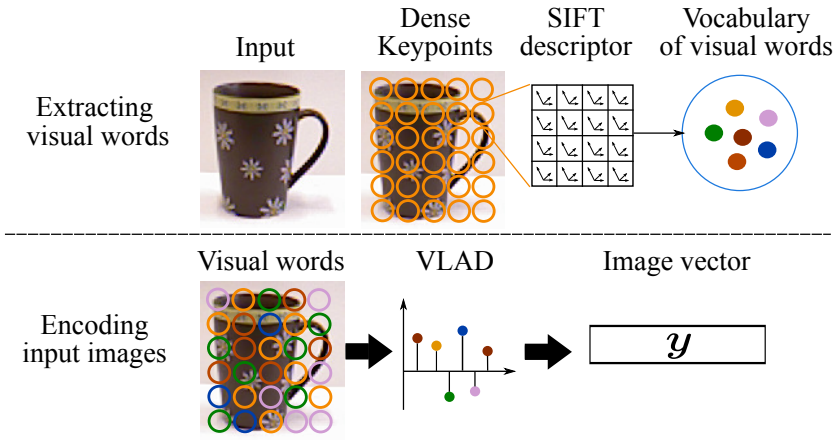


Figure 4.5: Illustration of the steps for encoding object images with the VLAD encoding method (Mici et al., 2018a).

jelovic et al., 2016). The BoF method simply computes a histogram of the local descriptors by hard assignment to a dictionary of visual words, whereas the VLAD method computes and traces the differences of all local descriptors assigned to each visual word. Some examples of words from the vocabulary of the VLAD encoder trained on two different object datasets are illustrated in Fig. 4.6. In the first row of the figure, examples from the Transitive Actions dataset are illustrated, and in the second row examples from the object-recognition benchmarking Washington Dataset are illustrated (classification results on the latter dataset are provided in Appendix D). As can be seen from the examples provided in the figure, some visual words do capture meaningful object parts, e.g., the handle of the cup or the keyboard keys. However, there are also cases (not illustrated in the figure) when the visual words represent simple oriented bars and corners without semantic or functional significance. These types of visual words can be quite convenient when the objects are distinguishable by their rich textures, such as books from their cover or cereal boxes from their logo and so on. It should be noted, however, that the image encoding process is completely unsupervised and no knowledge about the objects' categories is required during the feature extraction. Indeed, this can be one reason for the descriptors holding little category-related information.

The computational cost of the VLAD image encoding is moderate, given that the codebook size, K, is usually quite small, e.g., in our case, it is composed of only 64 visual words. So, the visual word assignment process, during which each feature of the new image should compute its distance with the visual words of the vocabulary, has a complexity of $\mathcal{O}(pKD)$. In our case, $K = 64$ and the PCA dimensionally-reduced SIFT descriptor, $p = 5$, and D is the number of descriptors

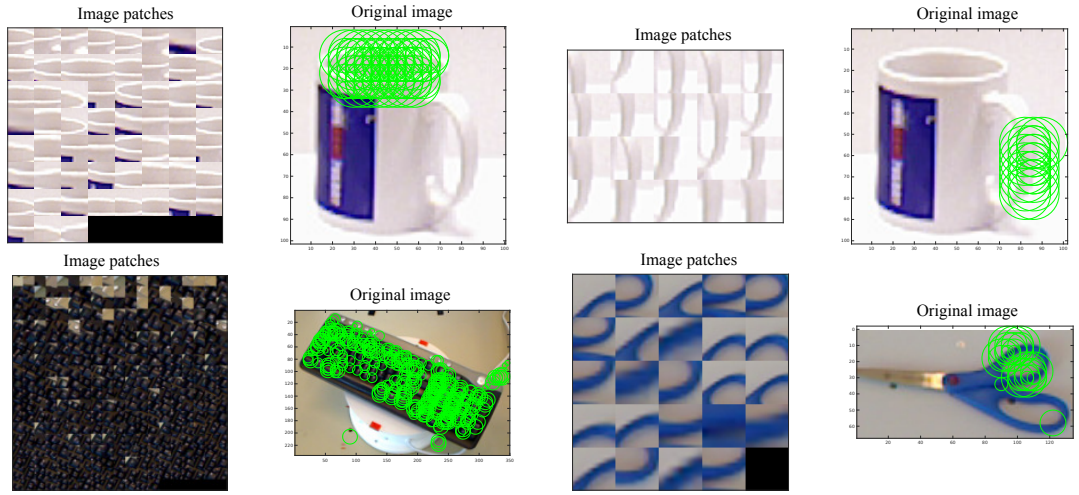


Figure 4.6: Examples of vocabulary words learned on two object datasets. The keypoints on the original object image matching one word are depicted with green circles and the corresponding image patches are on their left.

extracted from the image being encoded. Due to this low computational complexity SIFT features and compact image encodings based on them have been widely applied to real-time object retrieval and classification problems. However, their popularity in the second half of the last decade has been eclipsed by the multi-layer convolutional neural network (CNN) architectures, which have become state of the art in a variety of visual tasks (Zheng et al., 2017). Nowadays, there is a good choice of pre-trained CNN models which can be used out of the box for feature extraction and which are, as we speak, being outperformed by newer ones with a deeper and more complex neural structure (Lin et al., 2017). Of course, the similarity of the source used for training such models and the target images we want to encode and classify plays a critical role in the quality of features and defines how discriminative they are. Thus, quite often, fine-tuning is necessary before using such models for feature extraction on a new object dataset. For a deeper understanding of the differences between SIFT and CNNs, the reader can refer to the review by Zheng et al. (2017). Since our goal is the implementation of a system that can work in real time and does not require considerable computational power nor powerful GPUs, we opt for simple and efficient features like the body skeletal representations and SIFT for the objects. Also, as we will see in the rest of this chapter, SIFT features perform well on small datasets of objects particularly undergoing occlusions due to relying on local image representations.

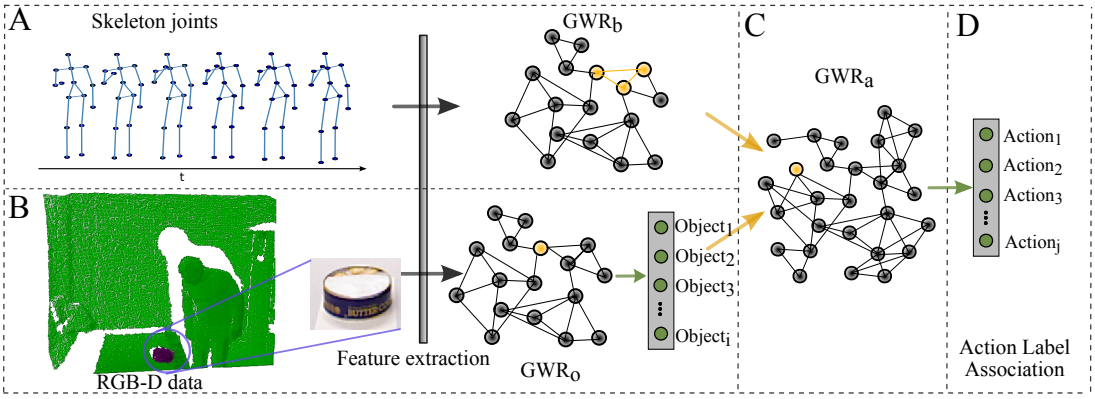


Figure 4.7: Overview of the neural architecture for the recognition of human-object interactions. (A) Processing of the body postures. A set of local features that encode the posture of upper body limbs is extracted and fed to the GWR_b network. (B) The input for the object recognition module is the RGB image of the manipulated object. If not provided by the dataset, the region of interest is automatically extracted through a point-cloud-based table-top segmentation. The object is represented as a compact feature vector and is fed to the GWR_o network which classifies the object. (C) The last network, GWR_a, learns the combinations of body postures and the object(s) involved in an action. (D) Action labels are associated with each neuron in the GWR_a network in order to evaluate the architecture's action classification performance (Mici et al., 2018a).

4.4 The Self-Organizing Hierarchical Architecture

The proposed architecture consists of two main network streams processing separately visual representations of the body postures and of the manipulated objects. The information from the two streams is then combined for developing action-object mappings. The building block of our architecture is the GWR network (Marsland et al., 2002), which is a growing extension of self-organizing networks with competitive learning. An overview of the architecture is depicted in Fig. 4.7.

The body pose cue is processed under the assumption that action-selective neurons are sensitive to the temporal order of prototypical patterns. Therefore, the output of the body pose processing stream is computed by concatenating consecutively activated neurons of GWR_b with a sliding time window technique. The object appearance cue is processed in order to have topological arrangements in GWR_o where different 2D views of 3D objects as well as different instances of

the same object category are mapped to proximal neurons in the prototypes domain. The advantage of having such a topological arrangement consists in mapping any unseen view of a known object into the corresponding views learned during the training. This capability resembles, to some extent, biological mechanisms for learning three-dimensional objects in the human brain (Poggio and Edelman, 1990; Perrett, 1996; Grill-Spector, 2013). Moreover, prototype-based learning approaches are supported by psychological studies claiming that semantic categories in the brain are represented by a set of most typical examples of these categories (Rosch and Mervis, 1975). For evaluating the architecture in terms of classification of human-object interaction activities, semantic labels are assigned to prototype neurons in GWR_a by extending the GWR algorithm with a labeling strategy.

4.4.1 Hierarchical Learning

We adopt hierarchical GWR learning (Parisi et al., 2015) for the data processing and subsequent action-object integration. Hierarchical training is carried out layer-wise and in an offline manner with batch learning. We first extract body pose, A, and object features, O, from the training image sequences, T. The obtained data is processed by training the first layer of the proposed architecture, i.e., GWR_b is trained with body pose data and GWR_o with objects (Fig. 4.7). After training is completed, the GWR_b network will have created a set of neurons tuned to prototype body pose configurations, and the GWR_o network will have learned to classify objects appearing in each action sequence.

The next step is to generate a new dataset T^* for the GWR_a network that integrates information coming from both streams (Fig. 4.8). In order to encode spatiotemporal dependencies within the body pose prototype space, we compute trajectories of the GWR_b best-matching units when having as input training action sequences. For all body pose frames $\mathbf{x}_i \in A$, the best-matching units are calculated (see Appendix B, Eq. B.1) and the corresponding neuron weights are concatenated following a temporal sliding window technique, as follows:

$$\psi(\mathbf{x}_i) = \mathbf{w}_{b(\mathbf{x}_i)} \oplus \mathbf{w}_{b(\mathbf{x}_{i-1})} \oplus ... \oplus \mathbf{w}_{b(\mathbf{x}_{i-q+1})}, i \in [q, m], \tag{4.1}$$

where $\oplus$ denotes the concatenation operation, m is the total number of training frames, and q is the width of the time window. We will refer to the computed $\psi(\mathbf{x}_i)$ by the name *action segment*.

The object data $\mathbf{y} \in O$ extracted from each action sequence is provided as input to the GWR_o network and the best-matching units $b(\mathbf{y})$ are calculated. Objects

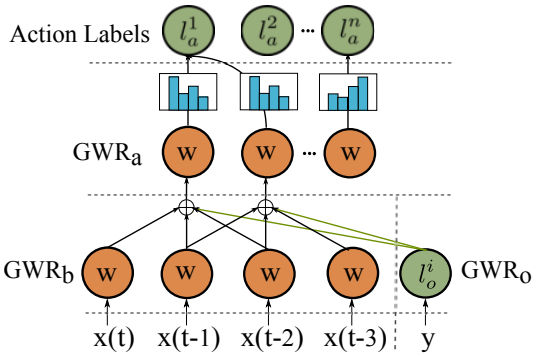


Figure 4.8: Schematic description of the hierarchical learning and of the association of action labels (not all neurons and connections are shown). At each time step t, the input data sample $\mathbf{x}(t)$ is represented by the weight $\mathbf{w}$ of the winner neuron which is then concatenated with the previous winner neuron weights (two previous neurons in this example) and the category label of the object l_o^i in order to compute the winner neuron in GWR_a. Each GWR_a neuron is associated with a histogram of action categories, and the most frequently matched class will be the recognized action (Mici et al., 2018a).

are extracted only at the beginning of an action sequence. Therefore, the object representations to be learned contain no temporal information and the computation of neural activation trajectories, reported in Eq. 4.1, is not performed. The label of the GWR_o best-matching unit is represented in the form of one-hot encoding, i.e., a vectorial representation in which all elements are zero except the ones with the index corresponding to the recognized object's category. When more than one object is segmented from the scene, the object data processing and classification with GWR_o is repeated as many times as the number of additional objects. The resulting labels are merged into one multiple-hot-encoded vector for the following integration step.

Finally, the new dataset T^* is computed by concatenating each action segment $\psi(\mathbf{x}_i)$ with the label of the corresponding object $l_o(\mathbf{y})$ as follows:

$$T^* = \{\phi_u(\mathbf{x}_i) \equiv \psi(\mathbf{x}_i) \oplus l_o(\mathbf{y}); \mathbf{x}_i \in A, \mathbf{y} \in O, u \in [q, m-q]\}. \tag{4.2}$$

Each pair ϕ_u, which we will refer to as an *action-object* segment, encodes both temporally-ordered body pose sequences and the identity of the object being manipulated during the action sequence. The GWR_a network is then trained with the newly computed dataset T^*, thereby learning the provided action-object pairs.

The resulting representative vectors of the body pose can have a very high dimension, which further increases when concatenating them through the temporal

window technique. Methods based on the Euclidean distance metric, as in our case, are shown to have a performance degradation when data lies in high-dimensional space (Aggarwal et al., 2001). Therefore, we apply the principal component analysis (PCA) technique to the neural weights of GWR_b. The number of principal components is chosen empirically in order to have a smaller-dimensional discrepancy with the object's label and maximize the classification performance. The new basis is then used to project weights of activated neurons in GWR_b before the concatenation of the activation trajectories and the subsequent integration step.

4.4.2 Classification

We extend the GWR algorithm with a labeling strategy for classification tasks while keeping the learning process unsupervised. We apply the majority vote strategy as in Strickert and Hammer (2005). For each neuron n_i, we store information about the category of the data points it has matched during the training phase. Thus, each neuron is associated with a histogram $hist(c, n_i)$ counting all cases of seeing a sequence with an assigned specific label c. Additionally, the histograms are normalized by scaling the bins with the corresponding inverse class frequency f_c and with the inverse neuron activation frequency f_{a,n_i}. In this way, class labels that appear less during training are less penalized, and the vote of the neurons is weighed equally regardless of how often they have fired. When the training phase is complete, each neuron that has fired during training, i.e., BMUs, will be associated with a histogram:

$$H(c, n_i) = \frac{1}{f_c \cdot f_{a,n_i}} \cdot hist(c, n_i). \tag{4.3}$$

At recognition time, given a test action sequence with length k, the best-matching units b_i are computed for each frame and the action label l is given by:

$$l = arg\max_c \left(\sum_{i=1}^{k} H(c, b_i) \right). \tag{4.4}$$

The classification of non-temporal data, e.g., object classification with the GWR_o network, is performed by applying majority vote only on the histogram associated to one best-matching unit H_{bmu}. This is a special case of Eq. 4.4, considering that $k = 1$ for non-temporal data.

In our case, action sequences are composed of smaller action-object segments as described in Section 4.4.1. Thus, the majority vote labeling technique described

so far is applied as follows. Let us assume we have a set of activity labels L_a along with our training data, for instance, *drinking* and *eating*. Therefore, each action-object segment $\phi \in T^*$ will be assigned with one of these labels and one action sequence will have the following form:

$$\Phi = \{(\phi_1, l_a^j), ..., (\phi_k, l_a^j), l_a^j \in L_a\}, \tag{4.5}$$

where l_a^j is the activity label and k is the number of action-object segments included in the sequence. During training of the GWR_a network on the action sequence Φ, the label l_a^j will be added to the histogram of the neurons activated for each of its composing segments ϕ. After the training is complete, the action sequence Φ will be classified according to the majority vote strategy (see Fig. 4.8). It should be noted that the association of neurons with symbolic labels does not affect the formation of topological arrangements in the network. Therefore, our approach for the classification of objects and actions remains unsupervised.

4.4.3 Training

In Table 4.1, we list the parameters used for training the proposed neural architecture throughout the experiments presented in Section 4.5. The selection of the range of parameters is made empirically while also considering the GWR algorithm learning factors. The parameters that we fix across all layers are the constants controlling the decreasing function of the firing rate variable (τ_b, τ_i and κ), the learning rates for the weights' update function (ϵ_b and ϵ_i) and the threshold for the maximum age of the edges (a_{max}). We set a higher insertion threshold parameter for the data processing layers, i.e., GWR_b and GWR_o, than for the integration layer GWR_a. The higher value chosen for the GWR_b and GWR_o networks leads to a greater number of neurons created and a better representation of the input data as a result, whereas the slightly lower value for GWR_a seeks to generate a set of neurons that tolerate more discrepancy in the input and generalize relatively more. The insertion threshold parameters are very close to each other and very close to 1, but their impact is perceptible given that the input data are normalized, i.e., take values within the interval $[0, 1]$. We train each network for 300 epochs over the whole dataset in order to ensure convergence, during which the response of the networks to the input shows little to no significant modifications. We choose 4 principal components for dimensionality reduction of the body pose spatiotemporal vectors prior to the concatenation with the corresponding object label.

In addition to the aforementioned parameters, the sliding window mechanism

Table 4.1: Training parameters of the GWR_b, GWR_o and GWR_a networks of our architecture for the classification of human-object interactions.

Parameter	Value
Insertion threshold	$a_T = \{0.98, 0.98, 0.9\}$
Firing threshold	$f_T = 0.1$
Learning rates	$\epsilon_b = 0.1, \epsilon_i = 0.01$
Firing rate behavior	$\tau_b = 0.3, \tau_i = 0.1, \kappa = 1.05$
Maximum edge age	$a_{max} = 100$
Training epochs	300

applied to the processed body pose data also has an impact on the growth of the GWR_a network. Wider windows lead to the creation of more neurons, albeit the slightly smaller number of data samples. This is an understandable consequence of the fact that the more temporal frames included in each time window, the higher the variance of the resulting data and the more prototype neurons created as a consequence. However, this parameter has to be set empirically according to the experimental training data distribution. We report the time window width parameter we set in each of our experiments in the following sections.

4.5 Experiments and Evaluation

We evaluated the proposed neural architecture both on the Transitive Actions dataset that we have acquired for the purpose of this study and on the publicly available action benchmark dataset, CAD-120 (Koppula et al., 2013), described in Section 4.2. In this section, we provide details on the classification performances obtained on these datasets, a quantitative evaluation of the integration module in the case of incongruent action-object pairs and a comparative evaluation on CAD-120.

4.5.1 Experiments with the Transitive Actions Dataset

Classification results

We now assess the performance of the proposed neural architecture for the classification of the actions described in Section 4.2.1. In particular, we want to evaluate the importance of the identity of the manipulated object(s) in disambiguating the activity that a subject performs. For this purpose, we conducted two sepa-

rate experiments, whereby we process body pose cues alone and in combination with recognized objects. Moreover, to further exclude any possible bias towards a particular subject, we followed a leave-one-subject-out strategy. Therefore, six different trials were designed by using video sequences of the first five subjects for training and using the remaining subject for the testing phase. This type of cross-validation is quite challenging since different subjects perform the same action in a different manner and with a different velocity.

We trained each GWR network with the learning parameters reported in Section 4.4.3. Since this dataset is composed of short temporal sequences, a time window of five frames was chosen for the concatenation of the processed body cues. This led to action-object segments of 0.5 seconds, considering 10 frames per second. When the training of the whole architecture was complete, the number of neurons reached for an input containing ≈ 6500 video frames was: 170 neurons for the GWR_b network, 182 for GWR_o and for the GWR_a network the number varied from 90 to 120 across different trials.

A plot showing the neural weights of the GWR_o network is depicted in Fig. 4.9. Given that the neural weights have a high dimensionality, i.e., the dimensionality of the VLAD descriptors, for illustration purposes we performed principal component analysis (PCA) and show the first two principal components. As can be seen from the plot, the neurons are topologically organized into clusters composed of different 2D views of the objects as well as different instances of the same object category. This is quite advantageous for our architecture since it allows for generalization towards unseen object views and, to some extent, towards unseen object instances. The overlap between the *can* and *mug* clusters suggests that the visual appearance of these object categories is more similar than compared to the others and, as a consequence, can be confused. However, this does not affect the action classification performance, since both of the objects are involved in the same activity, namely *drinking*.

We evaluated our architecture for the classification of human-object interactions using standard measurements (Van Rijsbergen, 1979):

$$Recall = \frac{TP}{TP + FN}, \tag{4.6}$$

$$Precision = \frac{TP}{TP + FP}, \tag{4.7}$$

$$F1\text{-}score = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision}, \tag{4.8}$$

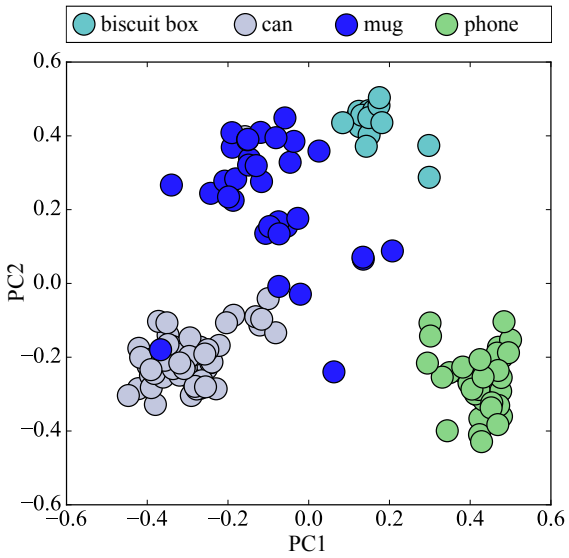


Figure 4.9: Neural weights of the GWR_o network after having been trained with the objects from the Transitive Actions dataset. The first two principal components have been chosen for the visualization in two dimensions (Mici et al., 2018a).

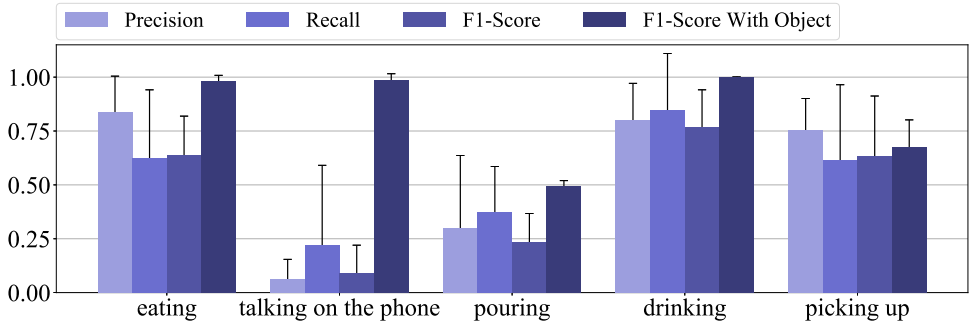


Figure 4.10: Classification results on the Transitive Actions dataset. Illustrated are precision, recall, and F1-score for the experiments without the object's information and the F1-score for the experiment setup considering the object. The mean values over 6 trials of cross-validation and the standard deviation are reported for each performance metric.

where, in a classification task with multiple classes $C = \{C_0, C_1, ..., C_i\}$, TP indicates *true positives*, i.e., the number of items correctly assigned to class C_i, FN indicates *false negatives*, i.e., the number of data items which were not recognized as examples of class C_i, and FP are the data examples that were incorrectly assigned to the class C_i. The precision score equals 1 (or 100%) for each class C when every item labeled as belonging to class C does indeed belong to class C, whereas a recall value that equals 1 (or 100%) means that every item from class

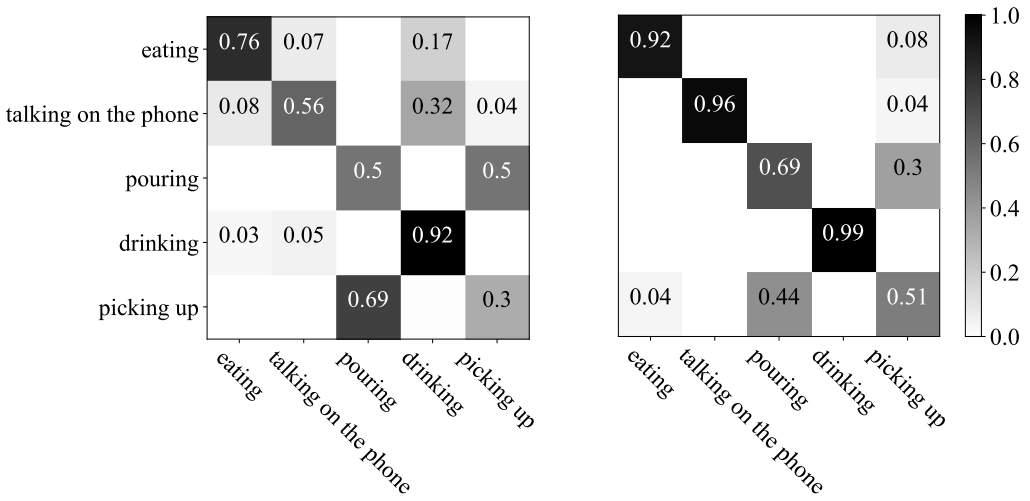


Figure 4.11: Normalized confusion matrices for the classification of the Transitive Actions dataset when **leaving out** the object's information and when **including** the object.

C was labeled as belonging to class C. The F1-score is a combination of precision and recall.

We report precision, recall, and F1-score for each class of activity without the object's information and the F1-score for the experiments considering the manipulated object. The mean values over the six trials and the standard deviation are illustrated in Fig. 4.10 and the normalized confusion matrices are illustrated in Fig. 4.11. For the *eating*, *drinking*, and *talking on the phone* actions, we obtained F1-score values greater than 0.95 when using the objects' identity information and lower values when using only body pose. For the *picking up* activity, on the other hand, the difference in the classification performance is marginal due to the fact that this action can be performed on all of the objects and the identity of a specific object does not play a decisive role. For the *pouring* activity the recognition rate does increase but is not as high as the other action classes. We assume that the reason for this is the similarity between the body pose during the *picking up* action and the *pouring* action. This points out the need for more fine-grained visual cues, for instance, the pose of the hand, which could lead to a higher recognition accuracy.

Experiments with incongruent action-object pairs

In addition to the classification experiments, we carried out a qualitative evaluation of the integration module when test data sequences of incongruent action-object pairs are given as input. We consider incongruent pairs to be unusual or function-

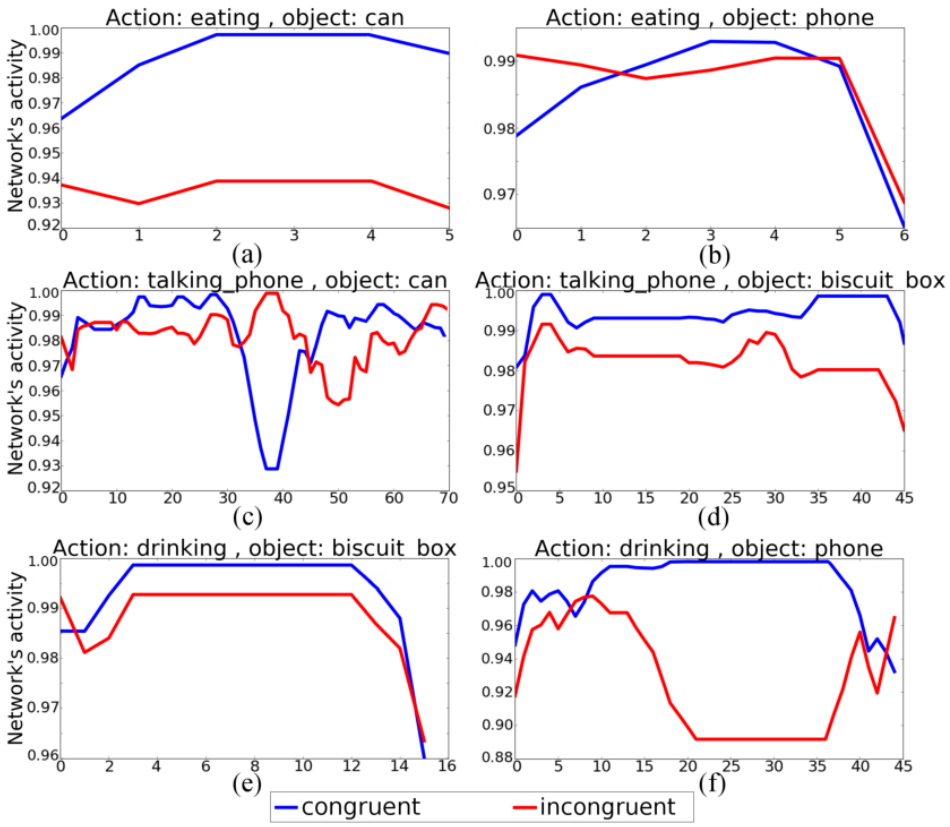


Figure 4.12: Comparison of the GWR_a network activations when having as input an action sequence combined with an incongruent object (in red) and one combined with the congruent one (in blue). The y axis represents the activation values, with 1 being the highest, and the x axis represents the number of frames of the illustrated data sequences. The number of frames can vary among different actions, e.g., the action *eating* is typically shorter than *talking on the phone* and *drinking* (Mici et al., 2018a).

ally irrelevant combinations of actions with objects, e.g., *drinking* with a *telephone* or *eating* with a *can*. As introduced in Section 2.1, several regions of the human brain have been found to be affected by object-action congruence (Yoon et al., 2012). The neural response in these areas is greater for actions performed on appropriate objects as opposed to unusual actions performed on the same objects. For this experiment, we artificially created a test dataset, for which we replaced the image of the object being manipulated in each video sequence with the image of an incongruent object extracted from a different action video.

We analyzed the activation values of the GWR_a BMUs (Eq. 3.11) on both the original action sequence and the manipulated one. A few examples of the obtained neural activations are illustrated in Fig. 4.12. We observed that the activations

were typically relatively low for the incongruent samples. This can be explained by the fact that the GWR_a prototypes represent the joint distribution of action segments and congruent objects taken from the congruent set. The activation of the network is expected to be lower when the input has been taken from a different data distribution than the one the model has learned to fit. The incongruent samples yield a higher discrepancy with respect to the prototype neurons, thereby leading to a lower network activation. However, we also noticed some exceptions, e.g., the incongruent pair <talking on the phone, can> depicted in Fig. 4.12.c. In this case, we can observe that the network activation becomes higher for the incongruent input at a certain point of the sequence, i.e., at a certain action-object segment. Nevertheless, a decreased network activation on the congruent input indicates that the network has a high quantization error for that particular action-object segment.

It should be noted that a small quantization error of the GWR network is not a requirement for a good performance in the action classification task. As described in Section 4.4.2, the classification of an action sequence is performed by considering the label histograms associated with the activated neurons. We can also notice some cases where the network activation on the incongruent input is not significantly low at the beginning of the sequence, but even slightly higher in the case of <eating, phone> (Fig. 4.12.b). A reason for this is the similar motion of the hand holding the object towards the head which may precede both *eating* and *talking on the phone* activities. Therefore, exchanging the object *biscuit box* with *phone* for the initial action segments has from little to no impact on the network's response.

4.5.2 Experiments with CAD-120

We evaluated the classification performance of our architecture on the publicly available benchmark dataset CAD-120. We computed skeletal quad features (described in Section. 4.4.1) for the encoding of the pose of the upper body, based on the three-dimensional position of skeletal joints provided in the dataset. Additionally, we extracted RGB images of manipulated objects from each frame and encoded them through the VLAD encoding technique as described in Section. 4.4.1. For the concatenation of the processed body pose cues, a time window of 9 frames was chosen. Since we down-sample the activity video frames to a rate of 10 fps, this leads to an action-object segment having a temporal duration of 0.9 seconds. After training the whole architecture with input data of $\approx$ 18.000 frames, the number of neurons reached in each GWR network was 460 for GWR_b, 410 for GWR_o, while for GWR_a the number varied from $\approx$ 3200 to $\approx$ 3700 across different trials of the

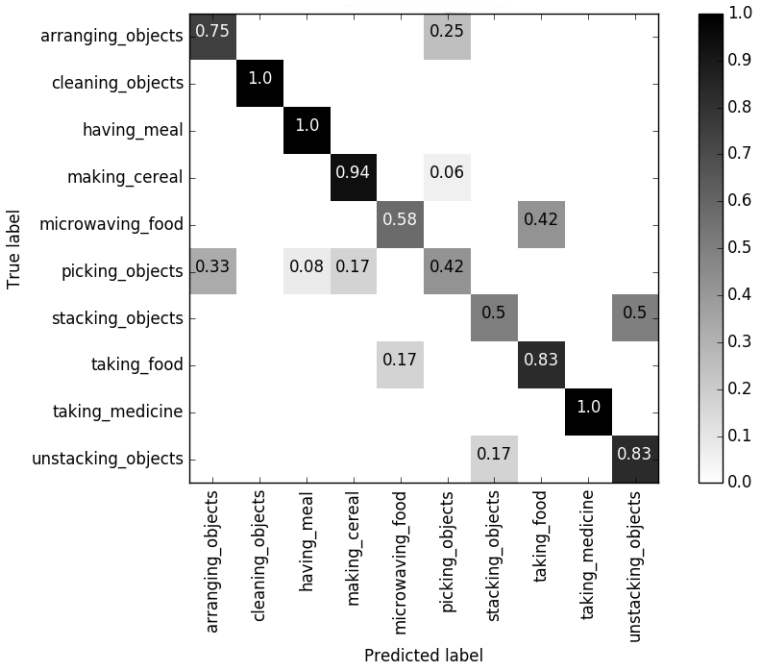


Figure 4.13: Confusion matrix for the 10 high-level activities of the CAD-120 dataset (Mici et al., 2018a).

cross-validation.

In Fig. 4.13, we show the confusion matrix for the 10 high-level activities of this dataset. We expected that the activities interchanged by our model were the ones including the same category of objects and similar body motions, e.g., *stacking objects* and *unstacking objects*, *microwaving food* and *taking food*. In fact, in the first two activities, the subjects repeat the same sequence of atomic actions: reaching, moving and placing objects, whereas the second two activities share the same atomic actions but in different orders: reaching, opening (microwave), moving, placing, and closing (microwave). The continuous interchange in the two mentioned examples is evident by looking at the architecture's output in Fig. 4.14. Also, the activity of *picking objects* was often confused with *arranging objects*, due to the fact that body pose segments of the *picking objects* activity are similar to the segments preceding the activity of *arranging objects*. In Table 4.2, we show a comparison of our results with the state of the art on the CAD-120 dataset with accuracy, precision, and recall as evaluation metrics. We obtained 79% accuracy, 80.5% precision, and 78.5% recall.

We reported only the approaches that do not use ground-truth temporal segmentation of the activities into smaller atomic actions or sub-activities (Hu et al.,

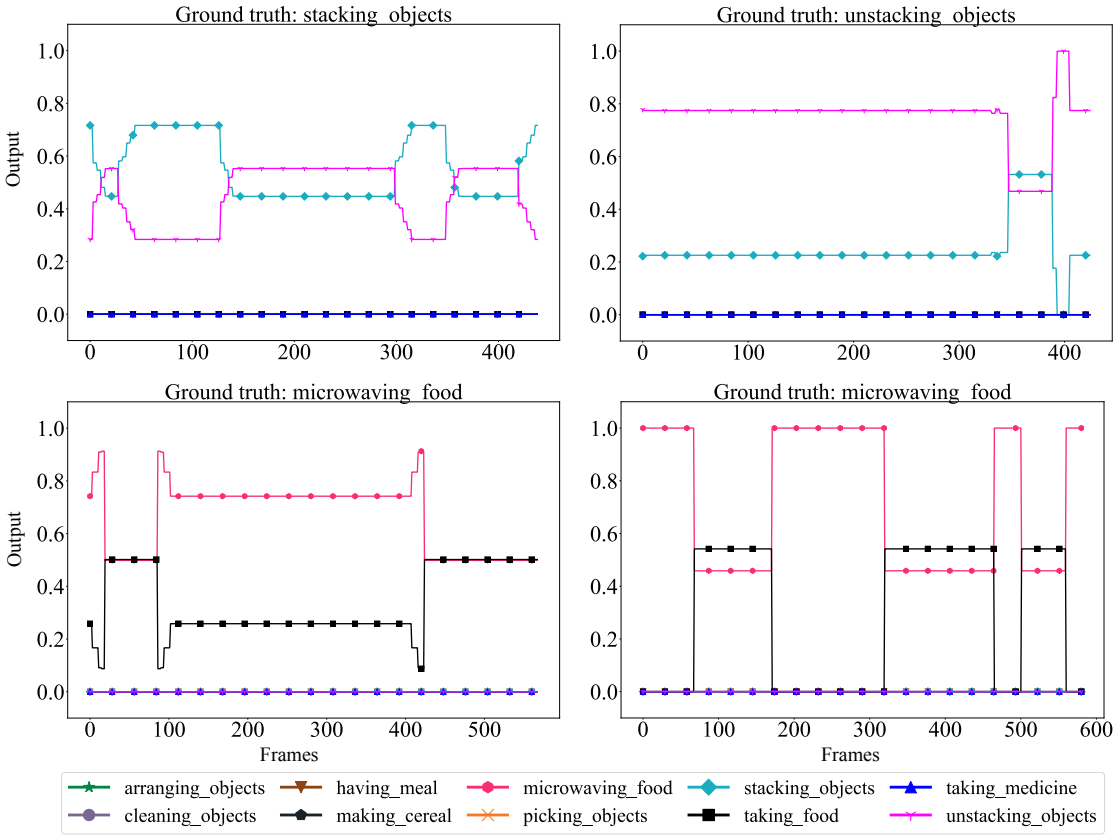


Figure 4.14: Output labels of the architecture during testing on unseen subjects of CAD-120. First row: the architecture interchanges *stacking objects* with *unstacking objects*, two activities involving the same objects and similar body motions. Similarly, in the second row, the architecture confuses *taking food* (from the microwave) with *microwaving food*.

Algorithm	**U**	**O. Rec.**	**O. Tr.**	**Acc.(%)**	**Prec.(%)**	**Rec.(%)**
Koppula and Saxena (2013), (*CRF*, *SVM*)	-	-	✓	83.1 ± 3.0	87.0 ± 3.6	82.7 ± 3.1
Koppula et al. (2013), (*CRF*, *SVM*)	-	-	✓	80.6 ± 1.1	81.8 ± 2.2	80.0 ± 1.2
Our approach, (GWR)	✓	✓	-	$\mathbf{79.0 \pm 3.4}$	$\mathbf{80.5 \pm 2.9}$	$\mathbf{78.5 \pm 3.6}$
Rybok et al. (2014), (*SVM*)	-	✓	-	78.2	-	-
Tayyub et al. (2015), (*SVM*)	-	-	✓	75.8 ± 6.8	77.9 ± 11.0	75.4 ± 9.1

Table 4.2: Results on the CAD-120 dataset for the recognition of 10 high-level activities. Reported are accuracy, precision and recall (in percentage) averaged over 4-fold cross-validation experiments. For comparison, we have included which of the reported methods is unsupervised (U), performs object recognition for the classification of the activities (O.Rec.) or relies on object tracking (O.Tr.).

2014; Taha et al., 2015). Our results are comparable with Rybok et al. (2014). Similar to our work, their method considers objects' appearance as contextual information which is then concatenated with body motion features represented as a bag of words. The best results were obtained by Koppula and Saxena (2013) reporting 83.1% accuracy, 87% precision, and 82.7% recall. In their work, spatiotemporal dependencies between actions and objects are modelled by a Conditional Random Field (CRF) which combines and learns the relationship between a number of different features such as the coordinates of the object's centroid, the total displacement and the total distance moved by the object's centroid in each temporal segment, the difference in (x, y, z) coordinates of the object and skeleton joint locations and their distances. After the generation of the graph, which models spatiotemporal relations, they use a Support Vector Machine (SVM) for classifying action sequences.

We assume that the tracking of the objects' position in the scene as well as the objects' distance from the subject's hand provides additional information that might improve our classification results. Therefore, the objects' position information will be considered in Chapter 7. Nonetheless, current results are quite promising considering that we extract less visual information compared to the other approaches and that the CAD-120 dataset contains complex scenes with varying points of view and considerable body occlusions leading to high tracking errors. The attenuation of noise may be to some extent achieved by the learning algorithm of the GWR networks. The algorithm is equipped with a mechanism to remove rarely activated neurons that may represent noisy input. Moreover, due to the firing counter mechanism of the GWR algorithm, well-trained neurons are trained less, thereby leading to less learning perturbations by slight input fluctuations.

4.6 Summary

In this chapter, we presented an approach based on neural self-organization for learning to recognize actions comprising human-object interaction from RGB-D videos. The proposed neural architecture relies on four assumptions that are consistent with evidence on neural mechanisms for transitive action recognition and on human psychological studies: 1) visual features of body pose and manipulated objects are processed in distinct pathways and are represented in distinct areas of the brain (Downing and Peelen, 2011; Grill-Spector, 2013; Beauchamp et al., 2002), 2) the visual input drives the arrangement of specific visual areas in the

brain through self-organization (Miikkulainen et al., 2006), 3) categories and concepts are learned as a set of typical examples or prototypes from observation and new observations are assigned to an existing category according to their similarity to the learned prototypes (Rosch and Mervis, 1975), and 4) the identity of the objects plays a crucial role in understanding the transitive actions (Saxe et al., 2004; Gallese et al., 1996).

Our architecture consists of two pathways of GWR networks processing respectively body pose and object appearance and identity, with a subsequent integration layer learning action-object mappings in an unsupervised way. The prototype-based learning mechanism of the GWR allows to attenuate input noise and to generalize towards unseen data samples. For the purpose of classification, we extended the GWR with a labeling technique based on majority vote.

The evaluation of our approach has shown good results on a dataset of human-object interactions collected specifically for the study on the importance of the identity of objects. The analysis of the neural response of the integration layer showed an overall lower network activation when given incongruent action-object pairs compared to the congruent pairs. Furthermore, the classification accuracy of our unsupervised architecture on a publicly available action benchmark dataset is competitive with respect to supervised state-of-the-art approaches. Unlike our approach, most state-of-the-art approaches rely on activity graphs which require fine-grained segmentation of body movements, usually done offline, making the framework computationally expensive and unsuitable for adaptive systems. Thus, the reported results motivate the application of our learning algorithm to assistive robot platforms, which will be able to extract the semantics of human activities perceived by the robot's vision system. At the current state, the proposed architecture can recognize human activities while being performed. However, in many real-world scenarios, the assistive system is required to identify an intended human activity before it is fully executed. For this reason, in the following two chapters we extend and evaluate our self-organizing approach towards more complex scenarios, such as simultaneous learning and prediction of human motion in HRI scenarios (see Chapter 5) and prediction of human-object interactions (see Chapter 6).

Chapter 5

Incremental Learning and Prediction of Human Motion with Self-Organization

5.1 Introduction

Real-time interaction with the environment requires robots to adapt their motor behavior according to perceived events. However, each sensorimotor cycle of the robot is affected by an inherent latency introduced by the processing time of sensors, transmission time of signals, and mechanical constraints (Mainprice et al., 2012; Zhong et al., 2012; Saegusa et al., 2007). Due to this latency, robots exhibit a discontinuous motor behavior which may compromise the accuracy and execution time of the assigned task. For social robots, delayed motor behavior makes Human-Robot Interaction (HRI) asynchronous and less natural. Synchronization of movements during HRI may increase rapport and endow humanoid robots with the ability to collaborate with humans during daily tasks (Lorenz et al., 2011). A possible solution to the sensorimotor latency is the application of predictive mechanisms which accumulate information from the robot's perceptual and motor experience and learn an internal model which estimates possible future motor states (Bahill, 1983; Behnke et al., 2003). The learning of these models in an unsupervised manner and their adaptation throughout the acquisition of new sensorimotor information remains an open challenge.

The efficient compensation of sensorimotor latencies caused by neural transmission delays plays a crucial role in human beings (Nijhawan and Wu, 2009). Predictive mechanisms in our sensorimotor system account for both motor pre-

diction and anticipation of the target movement during each action we take. The human cerebellum, for instance, is capable of estimating the effects of a motor command through an internal action simulation and a prediction model (Miall et al., 1993). Furthermore, there are additional mechanisms for visual motion extrapolation which account for the anticipation of the future position and movement of the target (Kerzel and Gegenfurtner, 2003). Internal models for sensorimotor prediction in humans constantly adjust to the sensory feedback (Rohde et al., 2014) as well as to the specific task (de la Malla et al., 2014). Similarly, artificial systems for the prediction of sensorimotor data and for delay compensation must be able to learn an internal model from sensorimotor observations and account for the continuous adaptation to the environment.

As discussed in Section 2.2.3, most of the existing prediction techniques mainly operate in a "learn then predict" approach, i.e., typical motion patterns are extracted and learned from training data sequences and then learned patterns are used for prediction (Zhong et al., 2012; Mainprice and Berenson, 2013; Ito and Tani, 2004; Levine et al., 2016). The main issue with this approach is that the adaptation of the learned models is interrupted by the prediction stage. However, it is desirable for a robot operating in natural environments to be able to learn incrementally, i.e., over a lifetime of observations, and to refine the accumulated knowledge over time. Therefore, the development of learning-based predictive methods accounting for both incremental learning and predictive behavior still need to be fully investigated.

In Chapter 4, we applied hierarchical self-organizing neural learning in order to map actions and manipulated objects for the classification of human-object interactions in an unsupervised manner. The experimental results showed that the hierarchical arrangement of two GWR networks with neurons encoding neural activation trajectories was able to successfully process three-dimensional body pose sequences and learn spatiotemporal action features. In this chapter, we propose a novel predictive mechanism based on the GWR learning algorithm (Marsland et al., 2002) which utilizes the neural trajectories to both learn the spatiotemporal input and predict future data samples in an online manner. Furthermore, we implement an architecture capable of compensating the sensorimotor delay of a small humanoid robot in the context of an imitation task in an HRI scenario. In this scenario, body motion patterns performed by a human demonstrator are mapped to trajectories of robot joint angles and then learned by the proposed neural architecture for subsequent imitation by the robot. We evaluate our system on a dataset of three subjects performing 10 different arm movement patterns.

We study the eligibility of the proposed neural framework for online sensorimotor delay compensation by measuring its prediction accuracy while being continuously trained. Experimental results reported in Section 5.4 show that the proposed architecture can adapt quickly to an unseen pattern and can provide accurate predictions albeit continuously incorporating new knowledge. Moreover, the system seems to maintain its performance even when training takes place with missing sensory information.

5.2 The Neural Framework

5.2.1 Overview

The standard GWR learning algorithm does not account for learning temporal sequences. This limitation has been addressed by different extensions described in detail in Section 3.5. The extension we made use of in the previous chapter was the hierarchy of GWRs augmented with a *window in time* memory (Parisi et al., 2015, 2016b; Mici et al., 2016). Now, our goal is to both encode data sequences and generate them. For this reason, we adopt the same approach, given that the relevant information regarding data samples in a window of time is always explicitly available. Furthermore, in contrast to the self-organizing networks equipped with Gamma filters, the sliding window technique does not affect the learning properties of the GWR algorithm.

The neural framework consists of a hierarchy of GWR networks (Marsland et al., 2002) which process input data sequences and learn inherent spatiotemporal dependencies (see Fig. 5.1) in an unsupervised manner. The outputs of the GWR_1 and GWR_2 networks are computed as the concatenation of the weights of consecutively activated neurons within a pre-defined temporal window q (see Fig. 5.2):

$$\mathbf{o}(t) = \mathbf{w}_{b(t)} \oplus \mathbf{w}_{b(t-1)} \oplus ... \oplus \mathbf{w}_{b(t-q+1)}, \tag{5.1}$$

where $\oplus$ denotes the concatenation operation. Moving up the hierarchy, the output $\mathbf{o}(t)$ will represent the input for the GWR network of the higher layer. In this way, the GWR_1 network learns a dictionary of prototypes of the spatial body configurations domain, while the GWR_2 and $P\text{-}GWR$ networks learn body motion patterns accumulated over a short and a longer time period respectively.

Such use of a multilayered GWR comes with three main advantages: First, it shapes a functional hierarchy that encodes spatiotemporal dependencies of the input in various timescales, e.g., for a sliding time window of width 3, the GWR_2

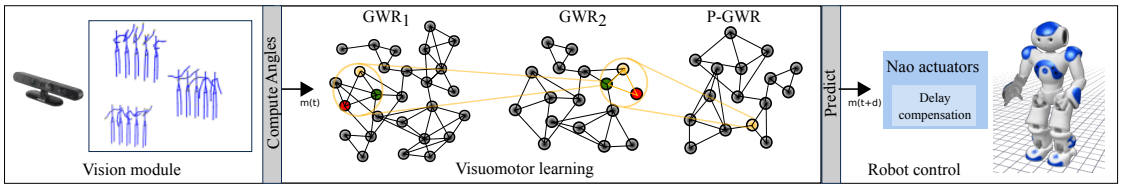


Figure 5.1: Overview of the proposed system for the sensorimotor delay compensation during an imitation scenario. The vision module acquires motion from a depth sensor and estimates the three-dimensional position of the joints. Shoulder and elbow angle values are extracted and fed to the visuomotor learning algorithm. The robot then receives predicted motor commands processed by the delay compensation module (Mici et al., 2018c).

network encodes 3 frames that correspond to 0.3 seconds and the *P-GWR* encodes 5 frames that correspond to 0.5 seconds of video when considering a discrete temporal sequence with a frame rate of 10 frames per second. This is consistent with evidence supporting increasingly large temporal receptive windows in the mammalian cortex (Giese and Poggio, 2003). Second, this scheme allows for a data compositionality, i.e., the sub-sequences learned and encoded by each neuron on a lower level can be re-used for representing different sequences on a higher level. Third, from the perspective of a system learning through a lifetime of observations, this hierarchical arrangement allows us to apply different neuron removal strategies in each layer in order to address the problem of forgetting rarely encountered, yet relevant information. More details about this point and on the online training strategy of the proposed neural framework are given in Section 5.4.1.

The hierarchical architecture is convenient for the application of a predictive mechanism due to the fact that the concatenations of consecutively matched prototypes, computed as the output of each layer, are explicitly mapping past values to the future ones. In fact, each vector can be split into two parts: the first carrying information about the input data at previous time steps, i.e., the *regressor* and the second representing the desired output of this mapping. Thus, if $\mathbf{x}(t)$ is the input vector fed to the *P-GWR* network, we can divide it into two parts:

$$\begin{aligned} \mathbf{x}^{in}(t) &= \mathbf{x}(t) \oplus \mathbf{x}(t-1) \oplus ... \oplus \mathbf{x}(t-p+1), \\ \mathbf{x}^{out}(t) &= \mathbf{x}(t+1), \end{aligned} \tag{5.2}$$

where $\mathbf{x}^{in}(t)$ is the regressor, $\mathbf{x}^{out}(t)$ is the desired output, and p denotes the maximum index of the past values. In the following section, we will see how to use the obtained regressor and output vectors in order to train the *P-GWR* network.

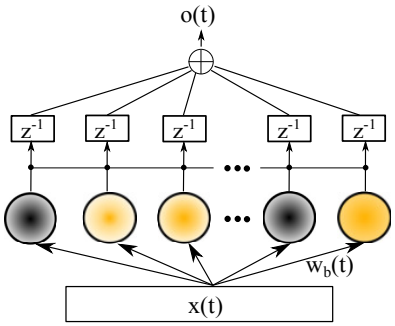


Figure 5.2: Schematic description of the output computation for the GWR_1 and GWR_2 networks (not all neurons and connections are shown). Given an input data sample $\mathbf{x}(t)$, the weight of the best-matching unit is concatenated with the weights of the previously activated neurons (depicted in fading yellow) in order to compute the output $\mathbf{o}(t)$. The length of the concatenation vector is a pre-defined constant q ($q = 3$ in this example). The z^{-1} blocks denote the time delay (Mici et al., 2018c).

5.2.2 The Predictive GWR Algorithm

The problem of one-step-ahead prediction can be formulated as a function approximation problem. Given a multi-dimensional time series denoted by $\{\mathbf{y}(\mathrm{t})\}$, the function approximation is of the form:

$$\hat{\mathbf{y}}(t+1) = \hat{f}\left(\mathbf{y}(t), \mathbf{y}(t-1), ..., \mathbf{y}(t-(p-1))|\Theta\right), \tag{5.3}$$

where the input of the function, or *regressor*, has an order of regression $p \in \mathbb{Z}^+$, with Θ denoting the vector of adjustable parameters of the model and $\hat{\mathbf{y}}(t+1)$ is the predicted value. In other words, the prediction function maps the past p input values to the observed value $\mathbf{y}(t+1)$ directly following them. We extend the GWR learning algorithm in order to implement this input-output mapping and apply this learning algorithm to the last layer of our architecture, i.e., to the *P-GWR* network.

The input samples fed to the *P-GWR* network are concatenations of the temporally ordered best-matching units (BMUs) from the preceding layer (Eq. 5.1) and are divided into two parts following Eq. 5.2. Each neuron of the *P-GWR* network will then have two weight vectors which we will call the input $\mathbf{w}^{in}$ and the output $\mathbf{w}^{out}$ weight vectors. During training, the input weight vector will learn to represent the input data regressor and the output weight vector will represent the corresponding predicted value. This learning scheme has been successfully applied to the *Vector-Quantized Temporal Associative Memory* (VQTAM) model (Barreto, 2007), shown to perform well on tasks such as time series prediction and predictive

control (Barreto et al., 2003).

The learning procedure for the Predictive GWR algorithm resembles the original GWR algorithm (see Appendix B) with a set of adaptations for temporal processing. During training, the first and the second best-matching units, b and s, at time step t are computed considering only the regressor part of the input:

$$\begin{aligned} b &= arg\min_{n \in A} ||\mathbf{x}^{in}(t) - \mathbf{w}_n^{in}||, \\ s &= arg\min_{n \in A/\{b\}} ||\mathbf{x}^{in}(t) - \mathbf{w}_n^{in}||, \end{aligned} \tag{5.4}$$

where $\mathbf{w}_n^{in}$ is the input weight vector of the neuron n and A is the set of all neurons. However, for the weight updates both $\mathbf{x}^{in}(t)$ and $\mathbf{x}^{out}(t)$ are considered:

$$\begin{aligned} \Delta\mathbf{w}_i^{in} &= \epsilon_i \cdot h_i \cdot (\mathbf{x}^{in}(t) - \mathbf{w}_i^{in}), \\ \Delta\mathbf{w}_i^{out} &= \epsilon_i \cdot h_i \cdot (\mathbf{x}^{out}(t) - \mathbf{w}_i^{out}), \end{aligned} \tag{5.5}$$

with the learning rates $0 < \epsilon_i < 1$ being higher for the BMUs (ϵ_b) than for the topological neighbors, as in the GWR algorithm, and h_i is the firing counter of the neuron. This learning mechanism guarantees that both the regressor space and the output space are vector-quantized. At each learning iteration, the quantization error of the output space is minimized following Eq. 5.5, hence the prediction error for the learned sequences is decreased.

The Predictive GWR algorithm operates differently from supervised prediction approaches. In the latter, the prediction error signal is the factor that guides the learning, whereas in the Predictive GWR the prediction error is implicitly computed and minimized without affecting the learning dynamics. Moreover, unlike the SOM-based VQTAM model, the number of input-output mapping neurons, or *local models*, is not pre-defined nor fixed but instead adapts to the input data.

5.2.3 Predicting Sequences

Given an input regressor $\mathbf{x}^{in}(t)$ at time step t, the one-step-ahead estimate is defined as the output weight vector of the *P-GWR* best-matching unit:

$$\hat{\mathbf{y}}(t+1) = \mathbf{w}_b^{out}, \tag{5.6}$$

where b is the index of the best-matching unit (Eq. 5.4). In the case that the desired prediction horizon is greater than 1, the multi-step-ahead prediction can be obtained by feeding back the predicted values into the regressor and computing

Eq. 5.4 recursively until the whole desired prediction vector is obtained. An alternative to the recursive prediction is the vector prediction which is obtained by increasing the dimension of the $\mathbf{x}^{out}$ vector with as many time steps as the desired prediction horizon h. Thus, the input regressor and the desired output would have the following form:

$$\begin{aligned} \mathbf{x}^{in}(t) &= \mathbf{x}(t) \oplus \mathbf{x}(t-1) \oplus ... \oplus \mathbf{x}(t-p+1), \\ \mathbf{x}^{out}(t) &= \mathbf{x}(t+1) \oplus \mathbf{x}(t+2) \oplus ... \oplus \mathbf{x}(t+h), \end{aligned} \tag{5.7}$$

where p denotes the index of the past values. The same dimensionality should be defined for the weight vectors $\mathbf{w}^{in}$ and $\mathbf{w}^{out}$ of the *P-GWR* neurons as well. This solution requires the training of the architecture with this setting of the weights.

5.3 Experimental Setup

The experimental setup consists of a simulated Nao robot incrementally learning a set of visually demonstrated body motion patterns and directly imitating them while compensating for the sensorimotor delay. We showcase the predictive capabilities of the proposed architecture in the context of an imitation scenario motivated by the fact that it can potentially imply behavior synchronization in the human-robot interaction. For humans, the synchronization of behavior is a fundamental principle for motor coordination and is known to increase rapport in daily social interaction (Lorenz et al., 2011). Psychological studies have shown that during conversation humans tend to coordinate body posture and gaze direction (Shockley et al., 2009). This phenomenon is believed to be connected to the mirror neuron system, suggesting a common neural mechanism for both motor control and action understanding (more details in Section 2.1.1). Interpersonal coordination is an integral part of human interaction, thus we assume that, applied to HRI scenarios, it may promote the social acceptance of robots.

5.3.1 System Description

A schematic description of our sensorimotor delay compensation system is given in Fig. 5.1. The system consists of three main modules:

1. The *vision module* which includes the depth sensor and the tracking of the 3D skeleton through OpenNI/NITE framework;[1]

[1]OpenNI/NITE: http://www.openni.org/software

2. The *visuomotor learning* module which receives angle values and provides future motor commands;

3. The *robot control* module which processes motor commands and relays them to the microcontrollers of the robot, which in our case is a locally simulated Nao.

We approach the demonstration of the movements through motion capture with a depth sensor, which provides us with reliable estimations and tracking of a 3D human body pose. Thus, the three-dimensional joint positions of the skeleton model constitute the input to the architecture. Then, the motor commands for the robot are obtained by mapping the user's arm skeletal structure to the robot's arm joint angles. This direct motion transfer allows for a simple, yet compact representation of the visuomotor states that does not require the application of computationally expensive inverse kinematics algorithms. Demonstrated motion trajectories are learned incrementally by training our hierarchical neural framework. This allows for extracting prototypical motion patterns which can be used for the generation of robot movements as well as the prediction of future target trajectories in parallel.

Although the current setup uses a simulated environment, we consider the same amount of motor response latency as it has been quantified in the real Nao robot, being between 30 to 40 ms (Zhong et al., 2012). This latency could be even higher due to reduced motor performance, friction or weary hardware. Visual sensor latency on the other hand, for an RGB and depth resolution of 640x480, together with the computation time required from the skeleton estimation middleware can peak up to 500 ms (Livingston et al., 2012). Taking into consideration also possible transmission delays due to connectivity issues, we assume a maximum of 600 ms of overall sensorimotor latency in order to carry out experiments described in Section 5.4.

5.3.2 Data Acquisition and Representation

The motion sequences were acquired by an Asus Xtion Pro camera at 30 frames per second. This type of sensor is capable of providing synchronized color information and depth maps at a reduced power consumption and weight, making it a more suitable choice than a Microsoft Kinect for being placed on a small humanoid robot. Moreover, it offers a reliable and markerless body tracking method (Han et al., 2013) which makes the interface less invasive. The distance of each participant from the visual sensor was maintained within the sensor's operational range, i.e., $0.8 - 3.5$ meters. To attenuate noise, we computed the median value for each body

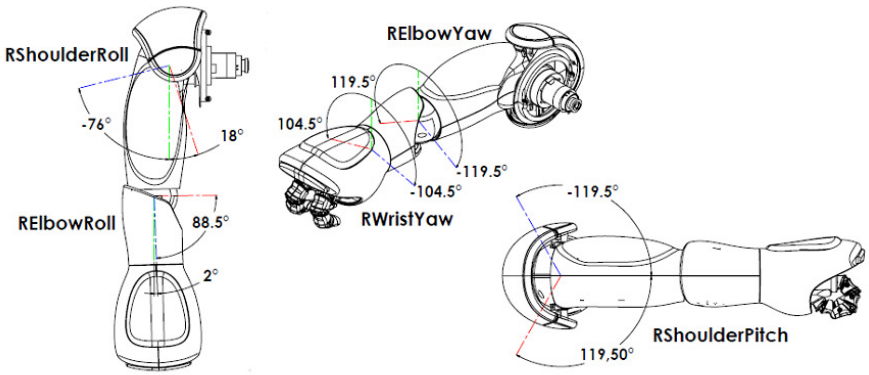


Figure 5.3: Nao's arm angles[2]. We consider only shoulder *pitch* and *yaw* and elbow *yaw* and *roll*. Wrist orientations cannot be extracted from the body skeletal representations.

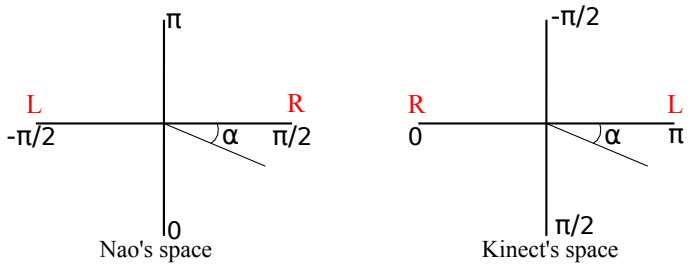


Figure 5.4: Mapping skeletons to Nao's joint angles. Left shoulder *roll* motion in Kinect and NAO spaces. Image drawn based on Rodriguez et al. (2014).

joint every 3 frames resulting in 10 joint position vectors per second (Parisi et al., 2016b).

We selected joint angles to represent the demonstrator's postures. Joint angles allow a straightforward reconstruction of the regressed motion without applying inverse kinematics, which may be difficult due to redundancy and leads to less natural movements. Nao's arm kinematic configuration differs from the human arm in terms of degrees of freedom (DoF)[2]. For instance, the shoulder and the elbow joints have only two DoFs (see Fig. 5.3) while human arms have three. For this reason, we compute only shoulder *pitch* and *roll* and elbow *yaw* and *roll* from the skeletal representation by applying trigonometric functions and map them to

[2]Software Documentation: http://doc.aldebaran.com

the Nao's joints by appropriate rotation of the coordinate frames (see Fig. 5.4):

$$\alpha_{LShoulderRoll} = \arccos((\overrightarrow{LShoulder} - \overrightarrow{RShoulder}) \cdot (\overrightarrow{LElbow} - \overrightarrow{LShoulder})) - \frac{\pi}{2},$$

$$\alpha_{LShoulderPitch} = 2\pi - \arcsin(z_{||\overrightarrow{LShoulder} - \overrightarrow{LElbow}||}),$$

$$\alpha_{LElbowRoll} = \arccos((\overrightarrow{LShoulder} - \overrightarrow{LElbow}) \cdot (\overrightarrow{LHand} - \overrightarrow{LElbow})) - \pi,$$

$$\alpha_{LElbowYaw} = \frac{\pi}{2} \cdot \frac{y_{(\overrightarrow{LHand} - \overrightarrow{LElbow})}}{\sin(\alpha_{LElbowRoll})}$$

For more details on the angle mapping see Rodriguez et al. (2014). Angle constraints are taken into account during this mapping. So, if a certain joint angle is impossible for the robot arm, the movement will stop at the maximal feasible value. Wrist orientations are not considered since they are not provided by the OpenNI/NITE framework. Considering the two arms, a frame contains a total of 8 angle values of body motion, which are given as input to the visuomotor learning module.

5.4 Experimental Results

We conducted experiments with a set of movement patterns that were demonstrated either with one or with both arms simultaneously: raise arm(s) laterally, raise arm(s) in front, wave arm(s), rotate arms in front of the body both clockwise and counter-clockwise. Some examples from these movement patterns are illustrated in Fig. 5.5. In total, 10 different motion patterns were obtained, each repeated 10 times by three participants (one female and two male) who were given no explicit indication of the purpose of the study nor instructions on how to perform the arm movements. In total, we obtained 30 demonstrations for each of the patterns. We first describe the incremental training procedure, then we assess and analyze in detail the prediction accuracy of the proposed learning method. We focus on the learning capabilities of the method while simulating a possible recurring malfunctioning of the visual system leading to the loss of entire data chunks. We conclude with a model for choosing the optimal predicted value for a system with a variable delay.

5.4.1 Hierarchical Training

The training of our architecture is carried out in an online manner. This requires that the GWR networks are trained sequentially with one data sample at a time.

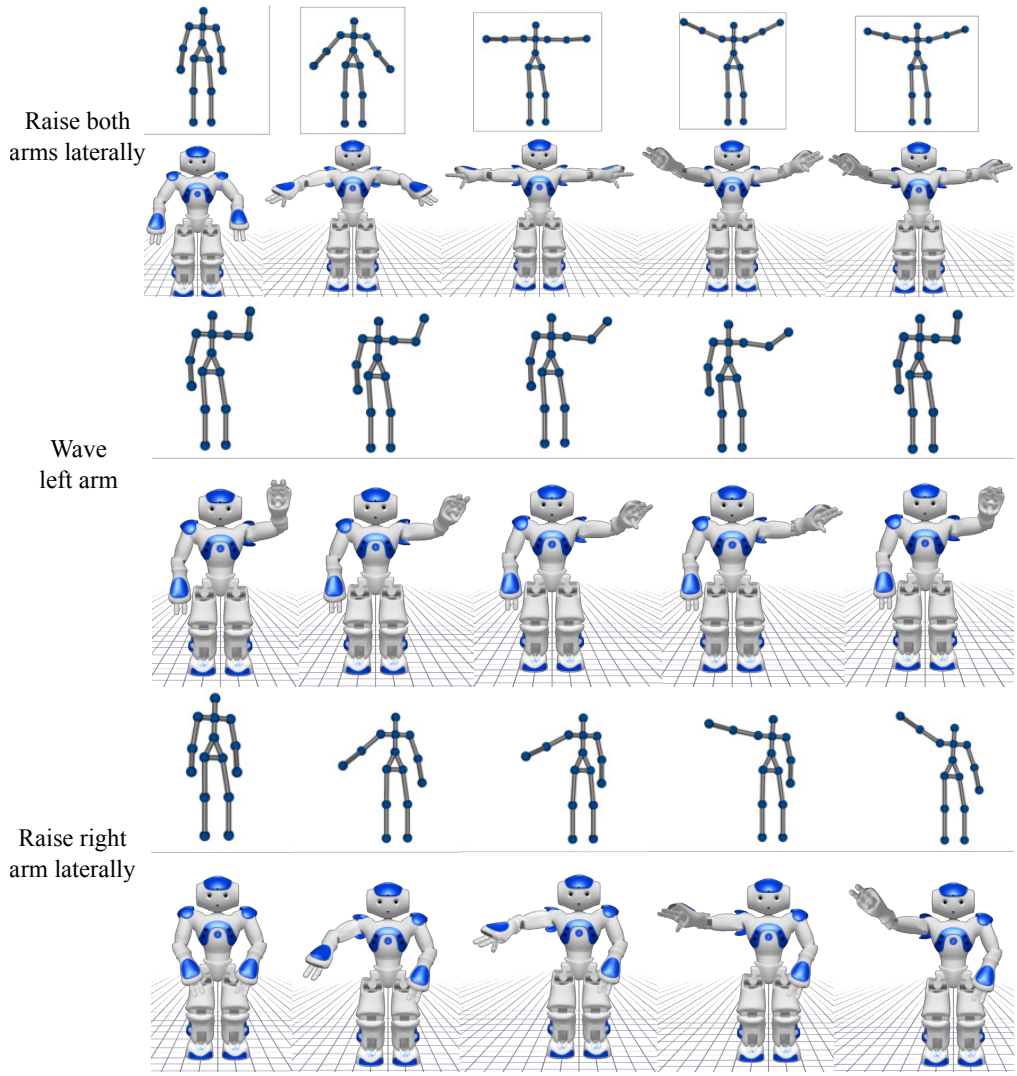


Figure 5.5: Examples of arm movement patterns. The visual input data are three-dimensional skeleton sequences which are mapped to the robot's joint angles (Mici et al., 2018c).

The networks are initialized with two neurons with random weight vectors. The GWR_1 network is trained to perform spatial vector quantization. Then the current sequence is gradually encoded as a trajectory of activated neurons as described in Eq. 5.1 and given as input to the GWR_2 network of the second layer. The same procedure is then repeated for the second layer until the training of the full architecture is performed. The learning of 30 demonstrations of one motion pattern from all three subjects constitutes one *training epoch*.

The learning parameters used throughout our experiments are listed in Table 5.1. The parameters have been empirically fine-tuned by considering the learning factors of the GWR algorithm. The firing threshold f_T and the parameters τ_b,

Table 5.1: Training parameters for the GWR_1, GWR_2, and *P-GWR* networks in our architecture for the incremental learning of sensorimotor patterns.

Parameter	Value
Activation threshold	$a_T = 0.98$
Firing threshold	$f_T = 0.1$
Learning rates	$\epsilon_b = 0.1, \epsilon_i = 0.01$
Firing counter behavior	$\tau_b = 0.3, \tau_i = 0.1, \kappa = 1.05$
Maximum edge age	{100, 200, 300}
Training epochs	50

τ_i, and κ define the firing counter decreasing function (see Appendix B, Eq. B.3) and were set in order to train a best-matching unit at least seven times before inserting a new neuron. It has been shown that increasing the number of trainings per neuron does not affect the performance of a GWR network significantly (Marsland et al., 2002). The learning rates are generally chosen to yield faster training for the BMUs than for their topological neighbors. However, given that the neurons' decreasing firing counter modulates the weights' update (see Eq. 5.5), an optimal choice of the learning rates has little impact on the architecture's behavior in the long run. The training epochs were chosen by analyzing the converging behavior of the composing GWR networks in terms of neural growth.

The activation threshold parameter a_T, which modulates the number of neurons, has the largest impact on the architecture's behavior. The closer to 1 this value is, the larger is the number of neurons created and the better is the data reconstruction during the prediction phase. Therefore, we kept a_T relatively high for all GWR networks. We provide an analysis of the impact of this parameter on the prediction performance of our architecture in Section 5.4.2. Finally, the maximum edge age parameter, which modulates the removal of rarely used neurons, was set increasingly higher with each layer. The neurons activated less frequently in the lower layer may be representing noisy input data samples, whereas in the higher layers the neurons capture spatiotemporal dependencies which may vary significantly from sequence to sequence. For instance, at the level of the GWR_1 network, which represents spatial body configurations, it is more probable that rarely seen input data samples are due to sensory noise. For the GWR_2 and P-GWR networks, on the other hand, rarely seen data samples are most probably due to sub-sequences encountered in the far past. For them, we set a higher edge age threshold so that neurons are removed more rarely.

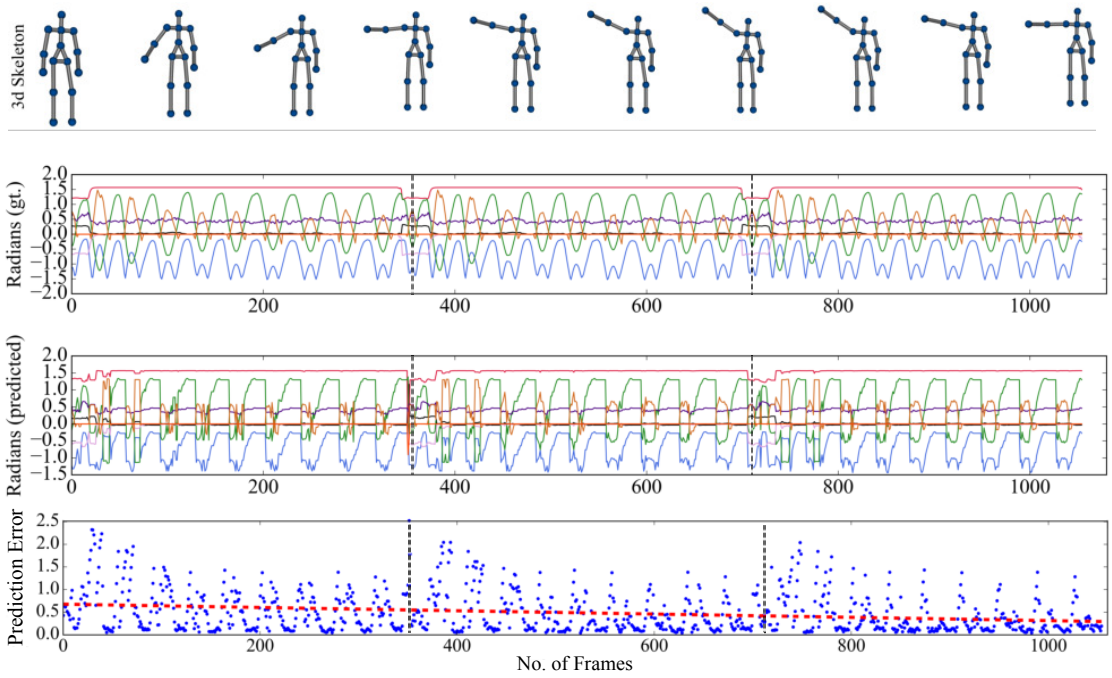


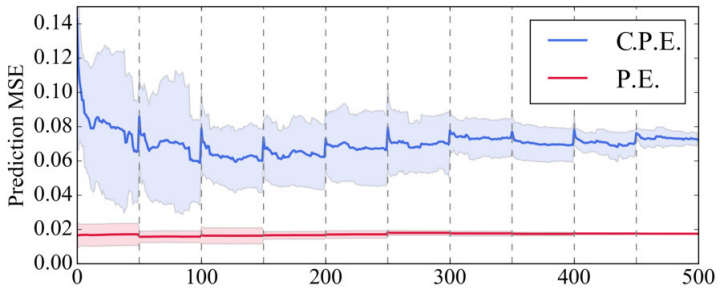
Figure 5.6: The behavior of the proposed architecture during training on an *unseen* sequence demonstrated by one subject (the sequence is presented three times to the network). From top to bottom illustrated are: the skeleton model of the visual sequence, the ground-truth data of robot joint angles, the values predicted from the network, and the Euclidean distance between predicted values and the ground truth over time (red dashed line indicating the statistical trend) (Mici et al., 2018c).

5.4.2 Predictive Behavior

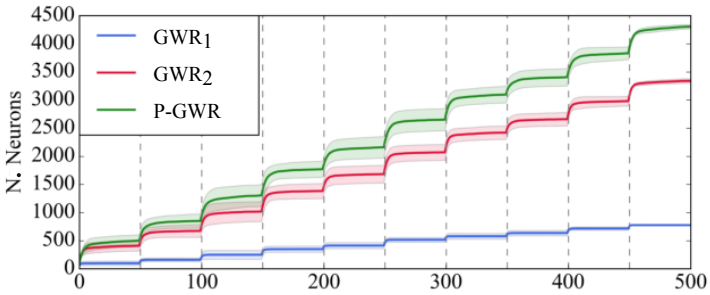
We now assess the predictive capabilities of the proposed method while the training is occurring continuously. Considering that the data sample rate is 10 fps, we set a prediction horizon of 6 frames in order to compensate for the estimated delay of 600 ms.

How fast does the architecture adapt to a new sequence?

An example of the online response of the architecture is shown in Fig. 5.6. We observed that, except in cases of highly noisy trajectories, the network adapted to an unseen input already after a few video frames, e.g., ≈ 100 frames which correspond to 10 seconds of the video sequence, and refined its internal representation after three presentations of the motion sequence demonstrated by one subject, i.e., after 30 demonstrations. This can be seen by the statistical trend of the prediction error.



(a)



(b)

Figure 5.7: (a) The cumulative prediction error (C.P.E) averaged over all learned sequences up to each learning epoch (in blue) and the prediction error (P.E.) computed between the predicted sequence and the sequence represented by the architecture (in red), (b) Average and standard deviation of the neural growth of the three GWR networks during learning (Mici et al., 2018c).

Behaviour analysis and prediction performance during incremental learning

We presented the movement sequences one at a time and let the architecture train for 50 epochs on each new sequence. The training phase was a total of 500 epochs for the whole dataset. Then, we re-ran the same experiment by varying the presentation order of the sequences and report the results averaged across all trials. In this way, the behavior analysis does not depend on the order of the data given during training. We analyzed the cumulative prediction error (C.P.E) of the model by computing the *mean squared error* (MSE) over all movement sequences learned up to each training epoch:

$$C.P.E = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{y} - \hat{\mathbf{y}})^2, \tag{5.8}$$

where m is the total number of frames seen so far. For comparison, we also computed the MSE between the values predicted by the model and the sensory input after being processed by the GWR_1 and the GWR_2 networks. For this, we substitute $\mathbf{y}$ in Eq. 5.8 with the weight vectors $\mathbf{w}_b^{GWR_2}$ of the GWR_2 neurons matching the input sequence. We refer to this performance measure as the prediction error (P.E.) since it evaluates directly the prediction accuracy of the *P-GWR* network while removing the quantization error propagated from the first two layers.

The flow of the overall MSE during training and the neural growth of the GWR networks composing the architecture are reported in Fig. 5.7. The moment in which we introduce a new motion sequence is marked by a vertical dashed line. As expected, the cumulative prediction error increases as soon as a new sequence is introduced (leading to the high peaks in Fig. 5.7.a.), for then decreasing immediately. However, the error does not grow but stays constant even though new knowledge is being added every 50 learning epochs. This is a desirable feature for an incremental learning approach. In Fig. 5.7.b., we observe that with the introduction of a new motion sequence there is an immediate neural growth of the three GWR networks followed by the stabilization of the number of neurons indicating a fast convergence. This neural growth is an understandable consequence of the fact that the movement sequences are very different from each other. In fact, the GWR_1 network, performing quantization of the spatial domain, converges to a much lower number of neurons, whereas the higher layers, namely the GWR_2 and the *P-GWR* network, have to capture a high variance of spatiotemporal patterns.

Impact of the activation threshold

In the described experiments, we set a relatively high activation threshold parameter a_T which led to a continuous growth of the GWR networks. Thus, we further investigated how a decreased number of neurons in the *P-GWR* network would affect the overall prediction error. For this purpose, we fixed the weight vectors of the first two layers after having been trained on the entire dataset, and ran multiple times the incremental learning procedure on the *P-GWR* network, each time with a different activation threshold parameter $a_T \in \{0.5, 0.55, 0.6, ..., 0.9, 0.95, 0.99\}$. We observed that a lower number of neurons, obtained through lower threshold values, led to quite high values of the mean squared error (Fig. 5.8). However, due to the hierarchical structure of our architecture, the quantization error can be propagated from layer to layer. It is expected that similar performances can be reproduced with a smaller number of neurons in the *P-GWR* network when a lower quantization error is obtained in the preceding layers.

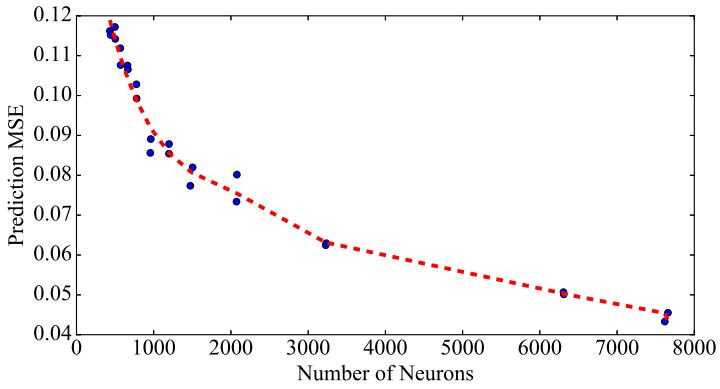


Figure 5.8: Prediction mean squared error (MSE) versus the number of neurons in the *P-GWR* network (Mici et al., 2018c).

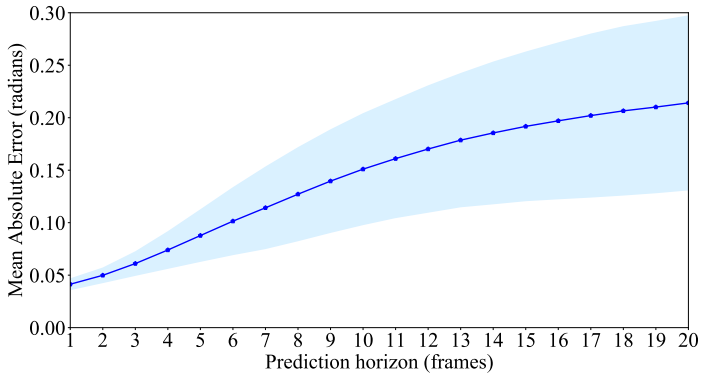


Figure 5.9: Mean absolute error (in radians) for increasing values of prediction horizons (in frames). In our case, 20 frames correspond to 2 seconds of a video sequence.

Sensitivity to the prediction horizon

We now take the architecture trained on the whole dataset and evaluate its prediction accuracy while increasing the prediction horizon up to 20 frames, which correspond to 2s of a video sequence. For achieving a multi-step-ahead prediction, we compute the predicted values recursively as described in Section 5.2.3. In Fig. 5.9, we report the mean absolute error and the standard deviation in radians in order to give a better idea of the error range. The results show that the magnitude of error and the standard deviation increase with larger prediction horizons. This should come as no surprise since producing accurate long-term predictions is a challenging task when dealing with human-like motion sequences. However, it seems that on average the error does not grow linearly but remains under 0.25 radians.

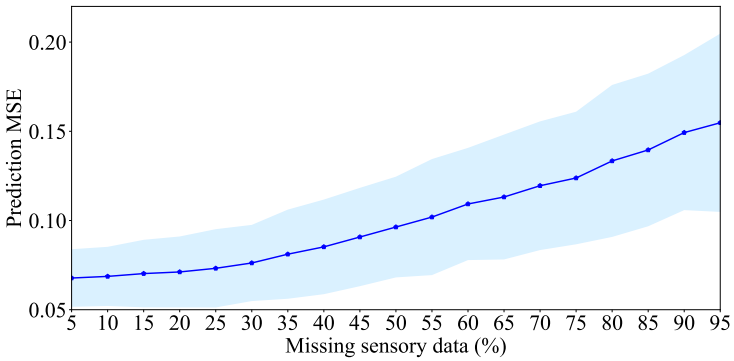


Figure 5.10: Prediction MSE averaged over 50 epochs of training on each motion pattern. For up to 30% of data loss the MSE does not grow linearly but rather stays almost constant. From this point on, the increasing percentage of data loss leads to the inevitable growth of the prediction error (Mici et al., 2018c).

5.4.3 Learning with Missing Sensory Data

In the following set of experiments, we analyze how the predictive performance of the network changes when trained on input sequences with missing data frames due to a faulty visual sensor or due to body occlusions. We simulate an occurring loss of entire input data chunks in the following way: during the presentation of a motion pattern, we randomly choose video frames where a second of data samples (i.e., 10 frames) is removed. The network is trained for 50 epochs on a motion sequence, each time with a different missing portion of information.

We repeat the experiment increasing the occurrence of this event in order to compromise up to 95% of the data and see how much the overall prediction error increases. Results are averaged over epochs and are presented in Fig. 5.10. As can be seen, the prediction MSE stays almost constant up to 30% of data loss. This means that the network can still learn and predict motion sequences even under such circumstances.

5.4.4 Compensating a Variable Delay

Experimental results reported so far have considered a fixed time delay which has been measured empirically by generating motor behavior with the real robot. However, the proposed architecture can also be used when the delay varies due to changes in the status of the hardware. In this case, given the configuration of the robot at time step t in terms of joint angle values $J_\xi(t)$, where ξ is the time delay estimation, the optimal predicted angle values to execute in the next step can be

chosen in the following way:

$$P^* = arg \min_{i \in [0,h]} ||J_\xi(t) - P(t+i)||, \tag{11}$$

where $P(t+i)$ are the predictions computed up to a maximum h of the prediction horizon.

The application of this prediction step requires a method for the estimation of the time delay ξ, which is out of the scope of this work. Current time delay estimation techniques mainly cover constant time delays, random delay with a specific noise characteristic, or restricted dynamic time delays, which nonetheless do not address uncertainty affecting real-world robot applications. Computational models inspired by biology have also been proposed for the time delay estimation (Sargolzaei et al., 2016). However, these models assume knowledge of the sensorimotor dynamics. The variable delay compensation technique needs further experiments which are not provided in this chapter.

5.5 Summary

Incremental learning and prediction of human motion patterns have been tackled by a great number of studies, which have adopted different methodologies, from Hidden Markov Models (HMM) to Gaussian Mixture Regression (GMR) and neural network architectures (see Section 2.2.3 and 2.2.4). In this chapter, we presented a self-organizing hierarchical neural architecture that achieved both tasks simultaneously and was evaluated in the context of a sensorimotor delay compensation system for a small humanoid robot. In particular, we evaluated the proposed architecture in an imitation scenario, in which the robot had to learn and reproduce visually demonstrated arm movements. Visuomotor sequences were extracted in the form of joint angles which can be computed from a body skeletal representation in a straightforward way. Sequences generated by multiple users were learned using hierarchically-arranged GWR networks equipped with an increasingly large temporal window. For the prediction of the visuomotor sequences we extended the original GWR algorithm with a temporal association mechanism, taking inspiration from the *Vector-Quantized Temporal Associative Memory* (VQ-TAM) model (Barreto, 2007). Experimental results demonstrated that the model can incrementally learn mappings of the regressors to the output vectors in the spatiotemporal domain with good precision. We conducted experiments with a dataset of 10 arm movement sequences showing that our system achieves low pre-

diction error values on the training data and can adapt to unseen sequences in an online manner. Experiments also showed that a possible system malfunction causing loss of data samples has a relatively low impact on the overall performance of the system. All these findings together suggest that the system is suitable for further applications to a robotic platform that operates in real environments and adapts continuously to sensorimotor feedbacks.

Similar to Chapter 4, we encoded temporal sequences through the sliding time window technique which comes with the disadvantage of increasing the computational cost due to the data's higher dimensionality. However, in our case, using angles as body pose features leads to a low-dimensional input compared to, e.g., raw images. Therefore, the training with long time windows does not pose a computational challenge. Furthermore, it has been shown that long-term predictions based on a sliding window are more accurate than recurrent approaches (Bütepage et al., 2017). The use of joint angles as visuomotor representations may seem to be a limitation of the proposed delay compensation system due to the fact that it requires sensory input and robot actions to share the same representational space. For instance, in an object manipulation task, this requirement is not satisfied, since the visual feedback would be the position given by the object tracking algorithm. This issue can be addressed by including both the position information and the corresponding robot joint angles as input to our architecture. Due to the generative nature of the self-organizing networks and their capability to function properly when receiving an incomplete input pattern, only the prediction of the object movement patterns would trigger the generation of corresponding patterns of the robot behavior.

Chapter 6

Prediction of Human-Object Interactions

6.1 Introduction

Human action analysis has been a major research topic since the early 1990s (Aggarwal and Ryoo, 2011) due to its relevance to a variety of applications such as health-care and assistive technologies as well as human-robot interaction and cooperation. As discussed in Chapter 4, learning to recognize complex human activities is more than just extracting body poses. For understanding human behavior, a more fine-grained visual analysis must be performed in order to extract more discriminative cues, for instance, the appearance and the identity of the objects during object manipulation. Due to this extra computational effort, the field of human action recognition has moved towards the recognition of realistic human activities involving objects or multiple persons only in the last decade. However, the focus has been mainly on the recognition of activities after a full observation, leaving the prediction of human actions an open challenge (Ryoo, 2011; Trong et al., 2017). The prediction of human activities before their full execution allows assistive robots to act anticipatorily and not just when given a command. For instance, when a robot sees a person holding a water carafe, it could infer that the person wants to drink and, consequently, it would react by fetching a cup.

As discussed in Section 2.2.5, existing recognition methodologies cannot be directly applied to the problem of activity prediction. State-of-the-art approaches for prediction (Lan et al., 2014; Koppula and Saxena, 2016), typically symbolic approaches, represent human activities as compositions of simpler entities, called atomic actions or action primitives. Similarly, in this chapter, we will approach

action prediction from the view of the hierarchical compositionality of activities, however, by means of an unsupervised neural framework. According to the hierarchical organization of goals proposed by Hamilton and Grafton (2006), human activities may involve several immediate goals, e.g., take a cookie or pour milk, each of which is achieved through a sequence of basic movements, e.g., extend an arm, preshape hand and close fingers. We will focus mainly on atomic actions that reach immediate goals in conjunction with a particular object without subsequent partitioning. For instance, we will consider the *drinking* activity as being composed of the atomic actions of picking up the cup and bringing the cup towards the mouth, but no elemental actions such as grasping or extend arm will be distinguished.

We will extend the hierarchical architecture described in Section 4 with a temporal association mechanism for the learning of consecutive body motion patterns during human-object manipulations. The architecture proposed in this chapter is novel in two main aspects: First, our learning mechanism develops distinctive mappings between objects and possible actions. This allows for the bidirectional retrieval of the information, i.e., it is possible to retrieve the appropriate object given a body action as well as to retrieve body motion patterns for manipulating a given object. Second, we use the same learning mechanism, i.e, the temporal Hebbian connectivity, for both learning the spatiotemporal dynamics of the body motion and the temporal order of the action sequences in longer activities. The application of both mechanisms allows for the emergence of *action chains*, i.e., temporally connected prototype neurons encoding consecutive action segments which are modulated by the specific target object. In this way, the architecture is able, when it receives only the initial action segment(s) starting a learned sequence, to carry out its most likely completion through an internal simulation of the full action sequence. Such a neural structure is reminiscent of the so-called *neural chains* (Chersi et al., 2011, 2014), which are believed to be the underlying neural mechanism for action recognition and execution in the human brain. Neural activity propagation through neural chains allows us to recognize activities by observing only a few of their composing motor acts. Neural chain activations are strictly modulated by the visual cues in the environment, for instance, a target object (Fogassi et al., 2005). Furthermore, neurons in one chain are not interchangeable with those of other chains even if they code the same motor act.

In Chapter 5, we implemented and analyzed a hierarchical neural framework for extracting and storing the temporal relationship between body motion patterns through self-organization. The multivariate prediction function was approximated

through a simple temporal association mechanism that mapped the regressor vectors to the corresponding output vector at each learning iteration (by following Eq. 5.2). Such input-output mappings are sufficient for motion prediction applications but cannot cope with a probabilistic interpretation of human activities. For instance, the action of *pick up can* can lead to both the action of *drinking* as well as the action of *pouring* from the can to a container like a mug. Such ambiguities will now be represented by multiple outgoing temporal connections whose weights are defined by the transitions frequency, as reported later in our experimental results.

We evaluate our architecture with the Transitive Actions dataset described in Section 4.2.1. Our experimental results show a high accuracy of the architecture in learning and anticipating plausible future actions. We also demonstrate the architecture's capability to synthesize body motion, thereby allowing for anticipating the way the next action will be performed. For robotic applications, the latter becomes relevant especially when robot planning takes place in environments shared with humans or other robots (Mainprice and Berenson, 2013).

6.2 A Self-Organizing Approach for the Prediction of Human-Object Interactions

The neural architecture consists of three network streams processing separately visual features of the body pose, motion, and the objects being manipulated. The information coming from the three streams is then integrated in order to develop spatiotemporal representations of action segments. The visual input is processed by three GWR networks (Marsland et al., 2002), whereas a GWR extended with the aforementioned temporal connections is applied for the integration of the processed information. In order to determine if the learned action segments are semantically meaningful, we associate each neuron of the integration module with an additional semantic layer comprising action labels. It should be noted, however, that the associative connections do not modulate the learning process of the integration module which remains unsupervised. An overview of the architecture is given in Fig. 6.1.

The proposed architecture has three main properties for the modeling of human-object interaction activities: First, activities are modeled as hierarchical structures in time, i.e., they are decomposed in sequences of atomic actions. Second, object identities are associated with actions in an unsupervised manner and serve as context information for disambiguating similar motion patterns. Third, human

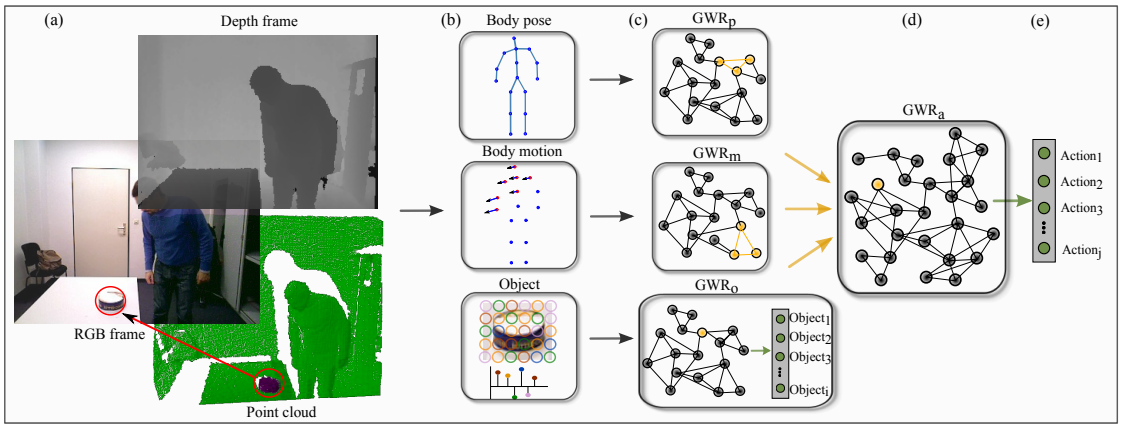


Figure 6.1: Extending the self-organizing architecture proposed in Section 4.4 towards action prediction from RGB-D videos. The current architecture is novel in three main aspects: 1) an additional network stream processes body motion information, 2) the GWR algorithm extended with asymmetric lateral connections is used for training the GWR_a network, and 3) associative connections are developed between the GWR_a neurons and the symbolic layer (Mici et al., 2018b). The PCA dimensionality reduction is not applied at the integration step of the current architecture in order to be able to generate action sequences.

motion trajectories are internally stored and can be retrieved at any point in order to predict and simulate how an action can be performed on a given object.

6.2.1 Learning Action-Object Segments

Except for the newly introduced body motion network stream, the hierarchical GWR learning adopted for the visual data processing remains similar to Section 4.4.1. First, we extract visual features of body pose, A, body motion, B, and manipulated objects, O, from the training image sequences, as described in Section 6.3.1. Then, we separately train the GWR_p network with the body pose features, the GWR_m with body motion, and the GWR_o with the objects. After the training is completed, the GWR_p will have created a set of prototype neurons representing typical pose configurations, the GWR_m will have neurons for prototype body motion vectors and the GWR_o network will have learned to classify objects appearing in each action sequence.

In order to encode spatiotemporal dependencies within the body features prototype space, we compute the neural activations of the GWR_p and GWR_m, i.e., the best-matching units $b(\cdot)$, and apply the delay embedding technique (Takens, 1981) which has been introduced in Section 3.5.1. For this, we take trajectories of

neural activations over time and group them into vectors of the form:

$$\psi_i(\mathbf{x}) = \{b(\mathbf{x}_i), b(\mathbf{x}_{i-\xi}), ...b(\mathbf{x}_{i-(q-1)\xi})\}, i \in [q, k], \tag{6.1}$$

where k is the total number of training frames and q and ξ are the embedding parameters denoting the width of the time window and the lag or delay between two consecutive frames, respectively. The choice of q is not critical as long as it is large enough. The lag parameter ξ, on the other hand, is chosen in order to maximize the independence of the delay vector components. The embedding parameters are data-dependent and can be set following a heuristic method or can be chosen empirically. Note that in the previous chapters the lag parameter has been fixed to $\xi = 1$, whereas now it will be further investigated for optimizing the performance of the neural architecture. By applying the delay embedding technique, we obtain two sets of spatiotemporal vectors with equal cardinality: one for the body pose $\psi_i(\mathbf{p})$ and one for the body motion $\psi_i(\mathbf{m})$, with $\mathbf{p} \in A$ and $\mathbf{m} \in B$.

The object data sample $\mathbf{y} \in O$ extracted at the beginning of each action sequence is provided as input to the GWR_o network and the corresponding best-matching units $b(\mathbf{y})$ are computed. The label of the GWR_o best-matching unit is represented in the form of a one-hot encoding, i.e., a vectorial representation in which all elements are zero except the ones with the index corresponding to the recognized objects' category. When more than one object appears in one action sequence, the object data processing and classification with GWR_o is repeated as many times as the number of additional objects. The resulting one-hot-encoded labels are merged into one fixed dimension vector for the following integration step.

Finally, all information processed by the GWR networks in the first layer of the architecture is integrated into a higher dimensional vector:

$$\phi_i = \psi_i(\mathbf{p}) \oplus \psi_i(\mathbf{m}) \oplus l_o(\mathbf{y}), i \in [q, k - q], \tag{6.2}$$

where $\oplus$ denotes the concatenation operator (see Fig.6.2). We will refer to the computed ϕ_i by the name *action-object segment*. Each segment is thus comprised of two parts:

1. the pre-processed visual sensor information about the body pose and motion,

2. the context information about the manipulated object, which is necessary to deal with ambiguities during the recognition and recall of segments which are shared among different action sequences.

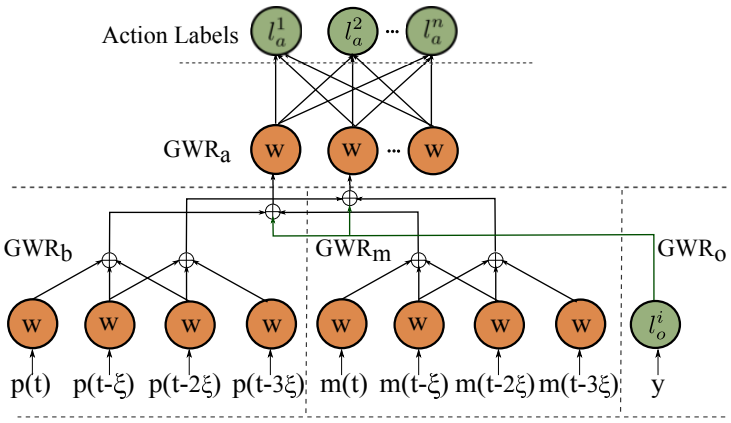


Figure 6.2: Schematic description of the hierarchical learning and of the association of action labels (not all neurons and connections are shown). At each time step t, the body pose $\mathbf{p}(t)$ and body motion $\mathbf{m}(t)$ are represented by the weight $\mathbf{w}$ of the winner neurons in GWR_b and GWR_m respectively. Then, each of these weight vectors is concatenated with the previous winner neuron weights (two previous neurons in this example) and the category label of the object l_o^i, in order to compute the winner neuron in GWR_a. Each GWR_a neuron is equipped with Hebbian connections to the semantic layer and the most frequently matched class will be the recognized action.

The set of newly computed spatiotemporal vectors is then used for training the GWR_a network.

6.2.2 Learning Goal-Oriented Action Chains

We now describe how we augment the GWR algorithm with two simple mechanisms in order to store and recall goal-directed action chains. For capturing the temporal aspects of human-object interaction sequences, we employ a time-delayed Hebbian learning rule in the GWR_a network which develops asymmetric temporal connections among the neurons. This learning mechanism has been successfully applied to the problem of trajectory learning with a self-organizing network (Araujo and Barreto, 2002). Interestingly, sequence completion driven by asymmetric connections between neurons is believed to be a feature of the human cortex (Mineiro and Zipser, 1998).

We define a fully connected matrix of weighted connections Ω among the neurons of the GWR_a network. The weights are adjusted in each learning iteration when the best-matching units are determined. The order of activation of the BMUs indicates the correct temporal order of the action-object segments that they rep-

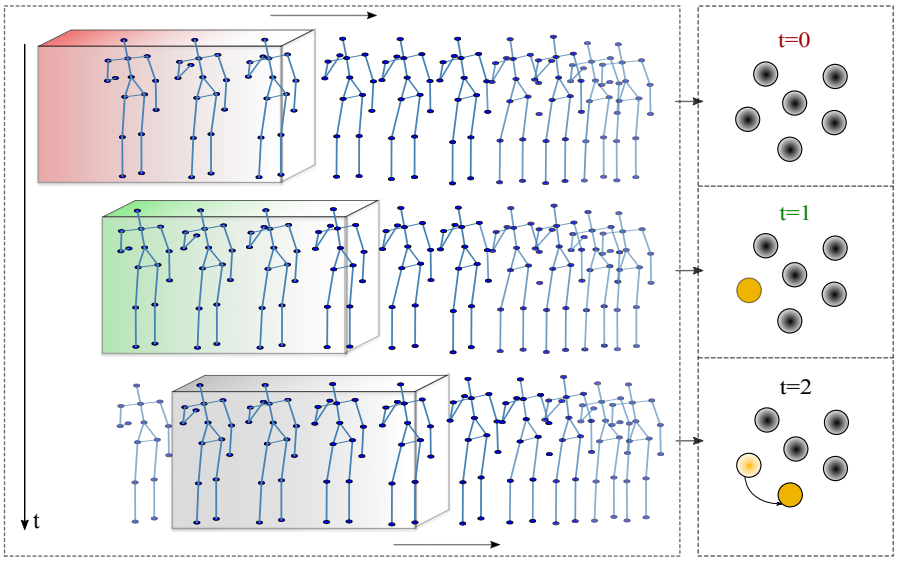


Figure 6.3: An illustration of how the temporal connections are established between consecutive BMUs in the GWR_a module. For simplicity, the action frames are depicted only as body skeletal configurations. At time $t = 0$, the delay embedded vector has a width lower than the defined time window (depicted as a sliding box). Thus, no response is obtained from the network yet. At time step $t = 1$, the first available action-object segment activates a neuron. An asymmetric lateral connection with a non-zero weight is established between the two consecutively activated neurons at time step $t = 2$.

resent (see Fig. 6.3). The learning rule is as follows:

$$\Delta\omega_{ij} = \mu \cdot a_j(t) \cdot a_i(t-1), \tag{6.3}$$

where $0 < \mu < 1$ is the temporal learning rate, and $a_i(t-1)$ and $a_j(t)$ are the activity values of the best-matching units at time step $t-1$ and t. The activity of a neuron is computed as a nonlinear function of the Euclidean distance between its weight $\mathbf{w}$ and the input data sample $\mathbf{x}(t)$ (see Eq. 3.11). Thus, following Eq. 6.3, when the two neurons i and j are consecutive BMUs, the temporal connection between them is strengthened in proportion to their activation, i.e., their similarity to the input data.

Taking the neurons' activation into account for the update of the temporal connections can help alleviate the problem of high quantization errors during the first learning iterations, while the network's growth is still taking place. Therefore, the farther the consecutive winner neurons are from the input they match, the less is the temporal connection between them strengthened. The temporal weights are

initialized to zero. Thus, non-zero connections are established only among consecutive winners and represent frequent transitions between action-object segments seen during training.

Since different trajectories, i.e., action sequences, should be handled by one single network, attention should be paid to have neurons responding to unique action-object pairs, e.g., *pick up mug* and not to *pick up phone* in order to have unambiguous goal-directed chains. For this purpose, we provide the objects' identity as a binary vector to each action-object segment as described in Section 6.2.1. However, the GWR neuron competition during training, which is based on the Euclidean distance function, does not guarantee that neurons specialize in unique action-object pairs. Thus, we apply a weighted Euclidean distance function to consider equally the two composing components of the action-object segments. The weights of the Euclidean distance are computed in the following way: For the pose and motion components of each action-object segment ($\psi_i(\mathbf{p})$, $\psi_i(\mathbf{m})$, see Eq. 6.2) the weights are given by an exponential function $\exp(-j)$, where $j \in [0, q]$, while for the object's identity the weights are set to 1. Then, the obtained weights are normalized such that their sum equals 1. With this type of configuration, a higher weight will be given not only to the manipulated object but also to the latest pose and motion frame in the action-object segment. Additionally, we modify the neuron insertion strategy in the following way: if the weight vector of the best-matching unit computed at time step t contains the identity of an object, o_b, different from the matched input, $o_{x(t)}$, then a new neuron is created.

6.2.3 Action Classification

While leaving the learning of the GWR_a network unsupervised, we simultaneously link each neuron to a symbolic action label $l \in L$, where L is the set of action classes. The GWR_a will then have a many-to-many relation with the symbolic layer. The set of weights Π, which are initialized to zero, are updated according to the rule:

$$\Delta\pi_{il_j} = \gamma \cdot a_i(t), \tag{6.4}$$

where $0 < \gamma < 1$ is the learning rate, $a_i(t)$ is the activity of the winner neuron at time step t and l_j is the target action label. After the training phase is complete, the weights are normalized by scaling them with the corresponding inverse class frequency and with the inverse neuron activation frequency. In this way, class labels that appear less during training are not penalized, and the vote of the neurons is weighed equally in spite of how often they have fired. The extended GWR

algorithm with the additional so-far described learning mechanisms is illustrated in Algorithm 1 (the modifications are highlighted in bold).

At recognition time, given one temporal segment of a human-object interaction at the time step t, the best-matching unit $b(t)$ is computed in the GWR_a module and the action label is given by:

$$l_j = arg\max_{l \in L}(\pi_{b,l}). \tag{6.5}$$

In order to classify an entire action sequence, a majority vote labelling technique is applied on the labels of its composing temporal segments.

Algorithm 1 The modified GWR algorithm (used for training the GWR_a module)

1. Create two random neurons with weights $\{\mathbf{w}_1, \mathbf{w}_2\}$
2. At each iteration t, generate an input sample $\mathbf{x}(t)$
3. Select the best and second-best matching neuron:
   $b = arg\min_{n \in A} ||\mathbf{x}(t) - \mathbf{w}_n||$, $s = arg\min_{n \in A/\{b\}} ||\mathbf{x}(t) - \mathbf{w}_n||$
4. Create a connection $E = E \cup \{(b, s)\}$ if it does not exist and set its age to 0.
5. (**New insertion condition**) If $(a(t) < a_T)$ and $(h_b < f_T))$ or $(o_b \neq o_{x(t)})$ then:
   - Add a new neuron r ($A = A \cup \{r\}$) with
     $\mathbf{w}_r = 0.5 \cdot (\mathbf{x}(t) + \mathbf{w}_b), h_r = 1$,
   - Update edges: $E = E \cup \{(r, b), (r, s)\}$ and
     $E = E/\{(b, s)\}$.
6. If no new neuron is added:
   - Update best-matching neuron and its neighbors i:
     $\Delta\mathbf{w}_b = \epsilon_b \cdot h_b \cdot (\mathbf{x}(t) - \mathbf{w}_b)$, $\Delta\mathbf{w}_i = \epsilon_i \cdot h_i \cdot (\mathbf{x}(t) - \mathbf{w}_i)$,
     with the learning rates $0 < \epsilon_i < \epsilon_b < 1$.
   - Increment the age of all edges connected to b by 1.
7. (**Newly introduced temporal connections**) If $b(t) \neq b(t-1)$ update the temporal connection weight between $b(t)$ and $b(t-1)$ following Eq. 6.3.
8. (**Newly introduced symbolic connections**) Update the symbolic connection weight between $b(t)$ and the target action label l_j following Eq. 6.4.
9. Reduce the firing counters of the best-matching neuron and its neighbors i:
   $\Delta h_b = \tau_b \cdot \kappa \cdot (1 - h_b) - \tau_b$, $\Delta h_i = \tau_i \cdot \kappa \cdot (1 - h_i) - \tau_i$
   with constant τ and κ controlling the curve behavior.
10. Remove all edges with ages larger than a pre-defined threshold and remove neurons without edges.
11. If the stop criterion is not met, repeat from step 2.

6.2.4 Action Prediction

During the prediction phase, each action sequence is presented to the trained architecture and the action-object segments are computed as described in Section 6.2.1. As can be seen in Fig. 6.3, the first winner neuron of the GWR_a is obtained at time $t = 1$, i.e., after the first q frames have been processed and the first composing temporal segment is available. The one-step-ahead prediction of the sequence can then be computed following the outgoing temporal connection with the maximal weight. In the case that the desired prediction horizon is greater than 1, the multi-step-ahead prediction can be obtained by recursively applying the one-step-ahead prediction computation. In both cases, the predicted action label for the last activated neuron is given by Eq. 6.5.

In contrast to Chapter 5, where we focused mainly on the motion prediction task, here we are interested in the higher-level action prediction problem, which is often indeterministic. For instance, the action of *pick up can* can lead to both the action of *drinking* as well as the action of *pouring* from the can to a container like a mug. In the current architecture, such ambiguities are represented by multiple outgoing temporal connections with non-zero weights. In other words, the maximal temporal weight gives the most probable, but not the only possible transition, after the observed action-object segment.

The proposed architecture has the advantage of self-organizing and learning sequences of arbitrary lengths in an unsupervised manner. The recall of a sequence can start at any point given one component action-object segment. Finally, the architecture can learn ordered action sequences, in our case called atomic actions, as a single long sequence, thereby providing a mechanism to recall the atomic action following the observed one.

6.3 Experimental Results

Now we assess the action label prediction capability of our architecture on an RGB-D dataset of human-object interactions, namely, the Transitive Actions dataset. The dataset consists of 4 simulated daily activities: *drinking* (from a container like a mug or a can), *eating* (an edible object like a biscuit), *pouring* (from a can into a mug) and *talking on phone* performed by 6 subjects. For the experiments reported here, each sequence was segmented into fine-grained atomic actions which define the activity: *pick up mug* and *drinking*, *pick up phone* and *talking on phone*, *pick up biscuit* from the biscuits box and *eating*, *pick up can* and *pouring* the liquid inside it into a mug. It should be noted that we distinguish between different *pick*

up sequences only for evaluation purposes. It allows us to better view the learned action-object chains and the consistency of the object in each sequence of atomic actions learned by the neural architecture. However, as mentioned also earlier, the learning process and the emergence of the internal representations are independent of the symbolic labels being used. In addition to the available activity sequences, we synthetically build longer ones in which the actions of *pick up mug* and *drinking* from mug follow the action of *pouring* from can to mug. With this type of sequence, we want to assess the prediction of more complex action sequences that lead to the inference of higher-level activities like, for example, *having meal*.

6.3.1 Feature Extraction

We consider only the position of the upper body joints *shoulders*, *elbows*, *hands*, center of *torso*, *neck*, *head* and *hips*, given that they hold all necessary information about the human-object interactions we focus on. However, the number of considered joints does not limit the application of our architecture for the task of recognition and prediction of full body actions. From each video frame, we extract the (x, y, z) position of each joint and we translate them into a coordinate system having the torso as the origin and concatenate them into one vector $\mathbf{p}$, which will then represent the body pose. Notice that we do not extract the skeletal quad features (Evangelidis et al., 2014) here since we are interested in not only the recognition but also the generation of body movement patterns.

We also consider the body motion vector $\mathbf{m}$, which we define as the differences in position of the upper body joints between two consecutive frames. We assume that these motion vectors, which encode the velocity of the movement between frames, hold significant information about apparently similar motion patterns, e.g. *pick up can* for *drinking* or *pick up can* for *pouring* its liquid into a mug. Behavioral studies with human infants have shown that the hands' motion velocity plays an important role in action anticipation (Stapel et al., 2015). Finally, the objects are extracted from the scene through a table-top segmentation algorithm and encoded by applying the Vector of Locally Aggregated Descriptors (VLAD) method (the same approach has been adopted in Section 4.3.2).

6.3.2 Predicting the Action Label

We follow the same cross-validation scheme described in Chapter 4, i.e., we train our architecture on activities performed by 5 subjects and test on activities of an *unseen subject*. The parameters used for training the architecture throughout our

Table 6.1: Training parameters for each GWR network in our architecture for the learning of human-object interaction sequences.

Parameter	Value
Activation Threshold	$a_T = 0.98$
Firing Threshold	$f_T = 0.1$
Learning rates	$\epsilon_b = 0.1, \epsilon_i = 0.01$
Firing counter behavior	$\tau_b = 0.3, \tau_i = 0.1, \kappa = 1.05$
Maximum edge age	{100,100,200}
Training epochs	50
Hebbian connections	$\mu = 0.3, \gamma = 0.5$

experiments were determined experimentally and are listed in Table 6.1. We define a time window width $q = 15$ and a lag $\xi = 3$ for the computation of the delay embedded vectors. In this way, we obtain action-object segments with a temporal length of 15 video frames.

In this set of experiments, we fix a prediction horizon of 500 ms (i.e, 15 frames at 30 fps) and compare the predicted action labels with the ground truth. In our dataset, the length of each atomic action is variable ranging from very short sequences, like *pick up* and *pouring* which can last less than a second, to very long sequences like *talking on phone* which can last up to 15 seconds. Thus, a prediction horizon of 500 ms is necessary not to penalize the short sequences in our dataset. The precision, recall, and F1-score for each action class, computed across all 6 folds, are reported in Fig. 6.4. Additionally, we report the confusion matrix both for the predicted and for the classified (ongoing) actions in Fig. 6.5 in order to further clarify the obtained results.

Analyzing the confusion matrices we can observe that the actions of *eating*, *drinking* and *talking on phone* are predicted with high accuracy, even though the test sequences have never been seen during training. *Pouring*, on the contrary, is not predicted with the same accuracy. We assume that the reason for this is twofold: (1) the misclassification of the pouring frames (which is evident from the classification confusion matrix) due to the fact that the body pose for *pick up can* and *pouring* are very similar (see Fig. 6.10), (2) the architecture often predicts what comes after *pouring* already, like *pick up mug* in order to drink. In both cases, the considerable number of false negatives causes the drop of the *pouring* recall metric, as can be seen in Fig. 6.4.

As for the prediction of the *pick up* sub-sequences, the confusion matrix is farther from the diagonal. However, the results demonstrate the temporal ambiguity

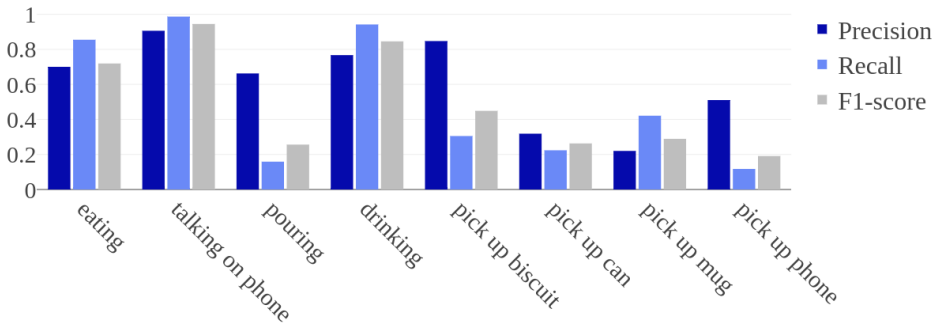


Figure 6.4: Action label prediction results on the Transitive Actions dataset. Illustrated are precision, recall, F1-score, averaged over 6 trials of cross-validation (Mici et al., 2018b).

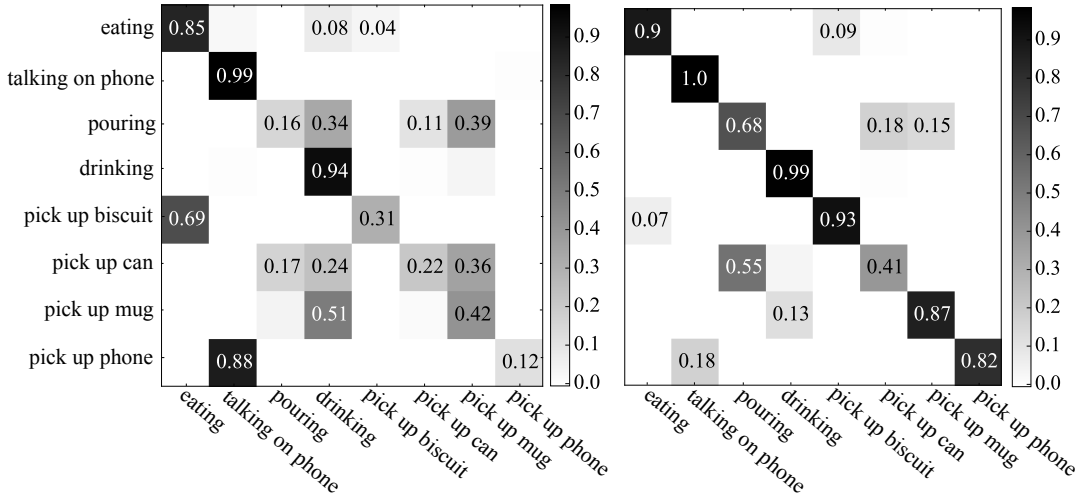


Figure 6.5: Normalized confusion matrices of the **predicted** and **classified** actions averaged over six trials (Mici et al., 2018b).

between consecutive atomic actions which can often lead to an imperfect segmentation. For instance, in the case of *pick up biscuit/mug/phone* the architecture predicts *eating*, *drinking* and *talking on phone*, respectively, quite early. However, this cannot be considered an error, but rather a desirable feature for real-time robotic applications, where the robot's response needs to be planned as much in advance as possible. Finally, there are obviously little to no implausible predictions such as *talking on phone* instead of *drinking* or *eating* and this shows that the architecture successfully performs in the task that has been assigned to it.

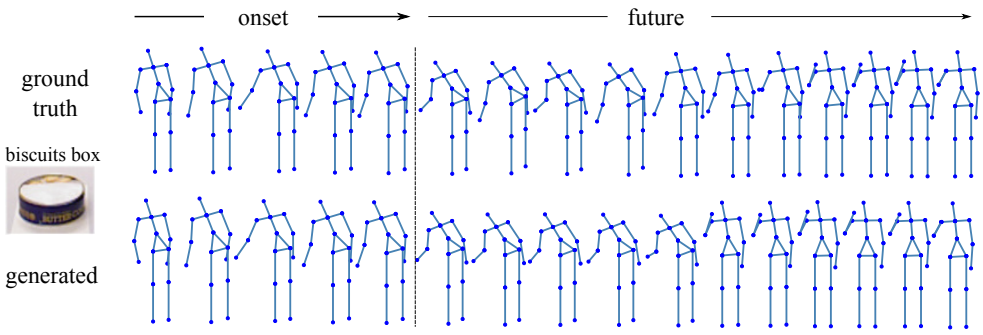


Figure 6.6: Body pose trajectories generated by the architecture when given the onset action-object segment of *pick up* (biscuit) which is followed by *eating*.

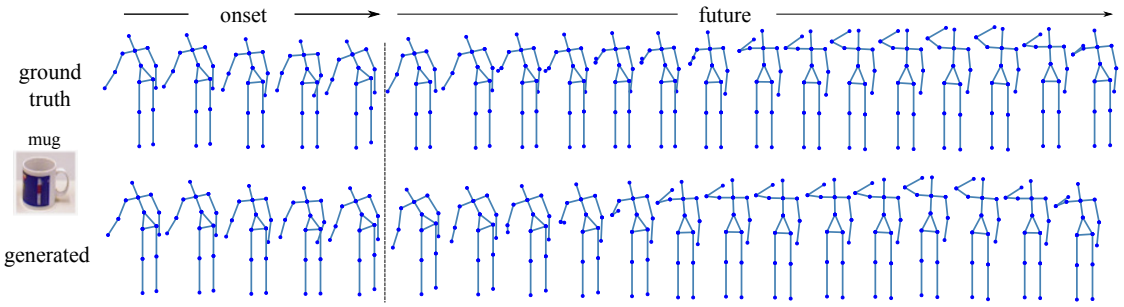


Figure 6.7: Body pose trajectories generated by the architecture when given the onset action-object segment of the action *pick up* (can) which is followed by *drinking*.

6.3.3 Visual Generation of Actions

We analyze the output of our architecture when simulating an entire action sequence given a specific object. While predicting action labels is important for a robotic platform when planning responses to those actions, predicting motion is crucial for planning robot motor commands in a shared workspace. In this round of experiments, we feed the trained architecture only the onset action-object segment starting a learned sequence and rely on the GWR_a's temporal connections for completing the sequence automatically. In order to do so, we recursively compute the one-step-ahead action-object segment, as described in Section 6.2.4. The iterations will stop when the current best-matching unit has no temporal connections to any other neurons - this indicates the end of the learned sequence. The performance of the architecture in this task is evaluated qualitatively, given that human motion synthesis is highly non-deterministic and its plausibility is hard to evaluate in a quantitative manner. This is evident by looking at the examples illustrated in figures 6.6, 6.7, and 6.8 (upper body joints are outputs of the GWR_a

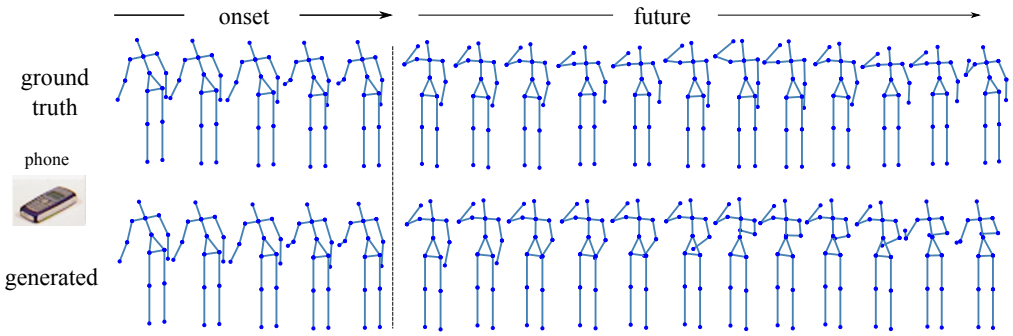


Figure 6.8: Body pose trajectories generated by the architecture when given the onset action-object segment of the action *pick up* (phone) which is followed by *talking on phone*.

module, whereas the feet are added for illustration purposes). For comparison, each generated sequence is depicted nearby its ground truth, i.e., the actual action sequence following the given onset action-object segment. As can be seen from the figures, the generated body pose trajectories are not strictly similar to the ground truth but do represent plausible actions. It can happen that the action following the *pick up* sub-sequence starts earlier, it is composed of fewer frames or it is performed with a different style than in the ground truth, e.g., the subject talking on the phone in Fig. 6.8 is also gesticulating with his left arm but not in the ground-truth sequence. This is an understandable consequence of the fact that the recall of the sequences is based on the most frequently seen body pose transitions during training and does not take into consideration different styles or speed of execution for each action. The neural activation trace of the GWR_a network when generating the sequence *pick up phone*→*talking on phone* is depicted in Fig. 6.9. For illustration purposes, the neuron weights have been projected into the 2D space through the Linear Discriminant Analysis (LDA) technique (Fukunaga, 2013). The figure reports an example of the action chains developed during training for the aforementioned sequence. It starts with a neuron belonging to the cluster *pick up phone* and transitions to the cluster *talking on phone*.

If the given onset action-object segment comprises the initial body postures of the atomic action *pick up can* and no further input is given to the architecture, the complete sequence composed by *pouring* and *drinking* will be generated. An example is illustrated in Fig. 6.10. However, if *pouring* is the only desired sequence, its action label, l^p, can be used to control the sequence generation in the following way: $j = arg \max_{w_{ij} \in \Omega}(w_{ij})$ if $l_j = l^p$. In words, among the outgoing temporal connections of the winner neuron, i, representing the current action-object segment, we choose

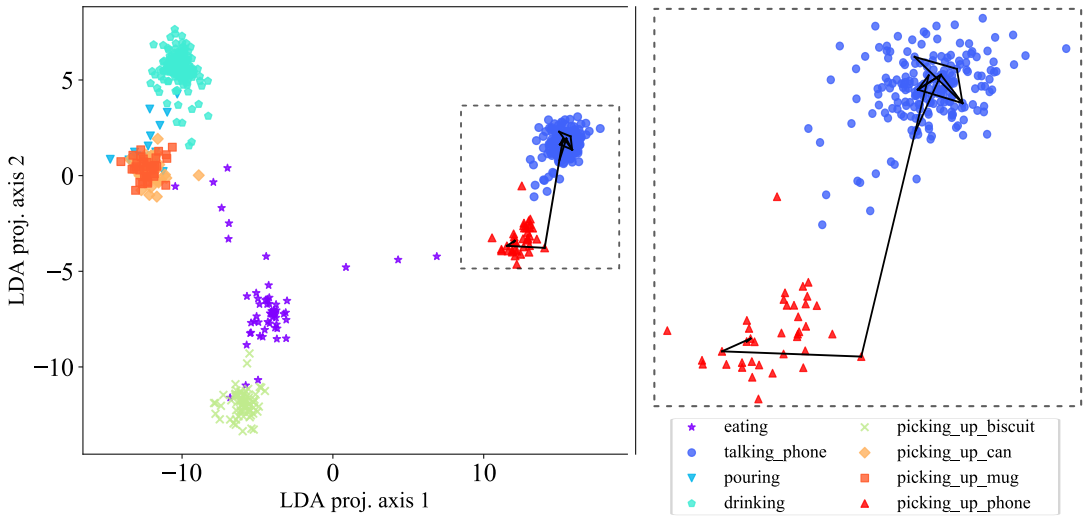


Figure 6.9: Example of an action chain developed for the sequence: *pick up phone* → *talking on phone*

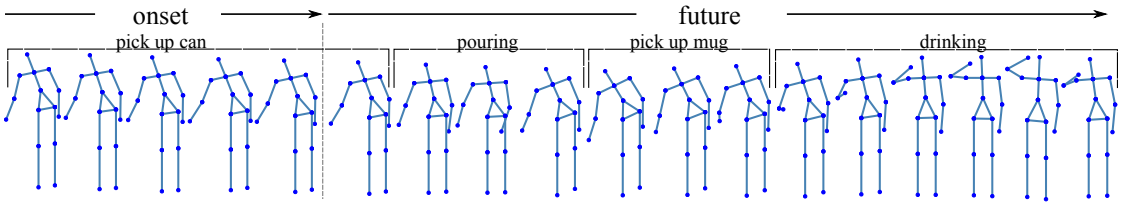


Figure 6.10: Body pose trajectories generated by the architecture when given the onset action-object segment of the action *pick up* (can). The action labels on top are output by the architecture. The sub-sequence *pouring* is automatically followed by *pick up* (mug) and *drinking*.

the one leading to the neuron with the specified action label. The same mechanism can also serve the purpose of choosing among multiple ambiguous states, which are common to different actions, in order to generate sequences that remain in the correct action class.

6.4 Summary

We presented a hierarchical neural network architecture for jointly learning to recognize and predict human-object interactions from RGB-D videos. In particular, we focused on how to extend a GWR learning algorithm in order to encode temporally ordered body motion patterns from sequences of arbitrary lengths. For this purpose, we employed the Hebbian learning in its simplest form in order to associate consecutive network activations. Additionally, actions were represented

as sequences of spatiotemporal segments consisting of body pose, motion, and the identity of the manipulated object(s). The temporal association mechanism together with the GWR network's self-organizing capability led to the formation of neural chains encoding goal-oriented actions. The formation of prototype neural chains resembles mechanisms for the execution and recognition of actions in the brain (Chersi et al., 2011). We showed that with the same underlying learning mechanism the architecture is able to predict body motion and scale to the prediction of action labels by learning ordered sequences of atomic actions. We evaluated our architecture on the Transitive Actions dataset showing that it can predict plausible future actions with high accuracy albeit being tested on sequences never encountered before. This dataset has low inter-class variability, i.e., the motion patterns are very similar across different action classes. This made the action prediction more challenging and allowed us to focus on analyzing in detail the internal neural chain representations developed within our architecture. However, further evaluation of our architecture should be performed as part of the future work on larger-scale datasets composed of more complex sequences of actions. Finally, the action generation results showed that our architecture can deal with receiving only some initial sensory input and internally simulate the rest of the action without being fed any further input.

The experiments with the generation of actions provided a qualitative analysis of the goal-oriented action chains developed within the architecture. The results showed that the learned sequences can be replicated accurately in the visual domain. This was our main focus, considering that not only predicting human actions but also anticipating how the actions are performed can be useful features for assistive robotic platforms that share the motion workspace with humans. Furthermore, the generative property of the proposed architecture makes it an attractive approach for robot action learning. For instance, instead of learning body posture trajectories it can be used to learn object state transitions, i.e., the object's position and pose over time, during the observation of object manipulation activities. Then, the same object trajectories can be replicated with a robotic platform by means of inverse kinematics. In Chapter 5, we approached the correspondence problem through direct angle mappings but this was more suitable for learning arm gestures. Moreover, the temporal connections developed between consecutively activated neurons can be seen as possible paths between states in an abstract representation of the robot workspace. On top of this representation, it is possible to benefit from the graph theory, which provides efficient path planning algorithms. Robot path planning can be performed, for instance, by searching

the shortest path when the weights of the neural connections refer to the distance from one state to another (Barreto et al., 2003). Thus, different extensions towards robotic scenarios are possible each of which introduces its own challenges.

Chapter 7

Learning Hierarchical Representations of Human-Object Interactions

7.1 Introduction

The network models presented in Chapter 4, 5, and 6 learn the spatiotemporal dependencies of the body movement patterns in an unsupervised manner being driven by bottom-up sensory information. However, it is known biologically that feedback, also called *top-down*, connections are widely present in the dorsal and ventral streams which process the visual sensory inputs in the human cortex. The top-down connections are thought to have a powerful influence in the shaping of the lower-level processes (Gilbert and Sigman, 2007). For instance, top-down connections from the premotor cortex play a role in the formation of the topographic class-grouped representations in the higher substrates of the visual processing hierarchy (Luciw and Weng, 2010). In this chapter, we will focus on a top-down learning mechanism which uses symbolic labels to modulate the neurogenesis as well as the topological arrangements in a hierarchical architecture of GWR networks. From the computational perspective, a top-down modulation mechanism is necessary in order to account for the classification error and not just the quantization error, which is not directly related to the classification performance.

In Chapter 6, we explored how learning the most frequent transitions among body postures by applying temporal Hebbian connections can lead to action anticipation. This anticipation mechanism was shown to enable also the mental simulation or the generation of human actions. The learning was conducted in

an unsupervised manner while the identity of the manipulated objects was used as the action context. In order to evaluate if the representations learned for the action-object segment sequences were semantically meaningful, we mapped neurons to action symbolic labels based on the co-occurrence of the visual stimuli and the corresponding label presented during training. In this chapter, we will see how to use the action labels in order to modulate learning, optimize the internal representations, and improve the action recognition performance.

The architecture proposed here is novel with respect to the fact that we introduce a new neuron insertion strategy that takes into account the classification error instead of just the quantization error. As will be shown later in this chapter, a neuron insertion strategy based on the minimization of the quantization error does not guarantee a better classification performance and may lead to the creation of an unnecessarily large number of neurons. We will analyze how this mechanism can be implemented in a hierarchical self-organizing architecture for the recognition of human activities on two semantic and temporal levels: atomic actions and long-term activities. Our goal is to consider both short-range and long-range relations between action-segments and use the feedback connections among layers for modulating the learning process.

Most of the previous work on the hierarchical recognition of human activities addresses activity and atomic actions recognition as separate tasks (Koppula and Saxena, 2013; Koppula et al., 2013), i.e., the action labels need to be inferred before the activity labels. In contrast to these approaches, we seek to jointly model actions and activities with one neural framework. By using the top-down modulation mechanism we aim to use the activity labels as a constraint for the atomic actions in order to have a better estimation of the actions and *vice versa*.

We will conduct a set of experiments with the CAD-120 dataset which has been previously used to evaluate the architecture for the recognition of human-object interactions (Chapter 4). Experimental results show that we outperform the state-of-the-art approaches with respect to the recognition of high-level activities. A qualitative analysis of the labels generated by the architecture during testing shows that semantically meaningful representations of the composing atomic actions emerge.

7.2 A New Neuron Insertion Strategy

The insertion criterion of new neurons in the original GWR algorithm is decided based on the local representation errors of the network. As discussed in Section 3.4,

the goal of a GWR algorithm is to estimate the unknown probability density of the input with the local density of the prototype vectors. If the activity of the best-matching unit (BMU) $b(t)$ at time t is lower than the insertion threshold a_T, then a new neuron will be inserted and will be placed halfway between $b(t)$ and the input $x(t)$. However, when target labels are available, the fact that the habituated BMU has been assigned a different label than the input it matches at time step t, for instance, can indicate that a new neuron should be inserted near the existing one. With this argument in mind, it seems reasonable to take the local classification error information into consideration for the neuron insertion criteria during training.

7.2.1 An In-depth Analysis

The GWR algorithm decides the moment and place for the insertion of a new neuron at each learning iteration. For this reason, each neuron should be equipped with a way of measuring how often it has misclassified. We associate each neuron i with a counter c_i, which is incremented whenever that neuron is the BMU of an input with a different label. In principle, this is very similar to the firing counter, or the neuron habituation, of the GWR algorithm which measures how often a neuron has fired. The neuron habituation prevents the creation of new neurons when the BMU has not been trained enough times. The misclassification counter prevents the creation of new neurons when the BMU has mismatched the input only a few times, which can happen when the BMU is a neuron only recently inserted or when the matched input sample is noisy. The idea of storing the local classification error is similar to the Supervised Growing Neural Gas (SGNG) algorithm (Holmström and Gas, 2002). The SGNG algorithm, however, has the extra computational cost of learning a set of output weights to a Radial Basis Function (RBF) output layer in parallel with the development of the prototype vectors.

Whenever the misclassification counter c_b of the habituated BMU at time step t exceeds a threshold m_T, a new neuron will be inserted between the badly matched winning neuron and the input and will take the label of the input. If there is no mismatch between the input and the BMU, then the algorithm will proceed normally with the weight updates. This insertion mechanism is independent of the labeling strategy being applied: it can be the frequency-based strategy, described in Section 4.4.2, or the weighted asymmetric Hebbian connections to a semantic layer as shown in Section 6.2.3.

If we include the quantization error in the neuron insertion condition described so far, then the condition would be the following: $((a(t) < a_T)$ *and* $(h_b < f_T)$ *and*

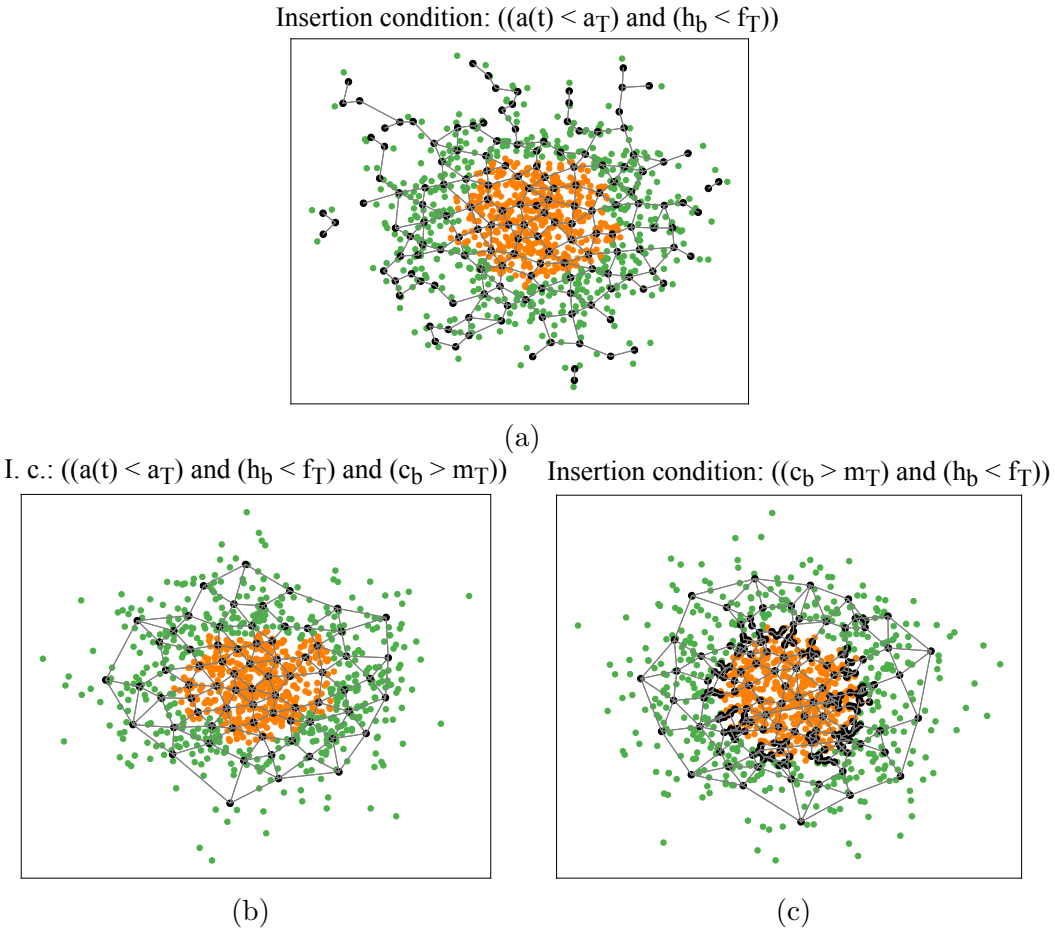


Figure 7.1: An experiment with a 2D dataset of two nested clusters demonstrating the neuron placement when applying different neuron insertion conditions. The neurons are depicted in black. (a) Results using only the quantization error (as applied in the GWR algorithm). (b) Results using both the quantization error and the classification error. (c) Results when only the classification error is considered as a neuron insertion condition. In each case, the habituation threshold is taken into account in order to assure sufficient training of the neurons.

$(c_b > m_T)$). Figure 7.1 illustrates an example of the neuron placement when using different neuron insertion strategies during classification. The dataset used for training the models is composed of one thousand data samples, drawn from a two-dimensional normal distribution, arranged in two nested clusters. The exact same parameters were used in each experiment: $f_T = 0.3$, $a_T = 0.9$, $\epsilon_b = 0.1$, $\epsilon_i = 0.01$, 50 training epochs and maximum edge age 50. We set a misclassification threshold $m_T = 10$. As can be easily noted in Fig. 7.1.a, the neurons in a GWR algorithm try to cover the whole data distribution in the best way possible, no matter what class each data point belongs to. In Fig. 7.1.b the use of the two insertion criteria in conjunction leads to the creation of a significantly smaller number of neurons,

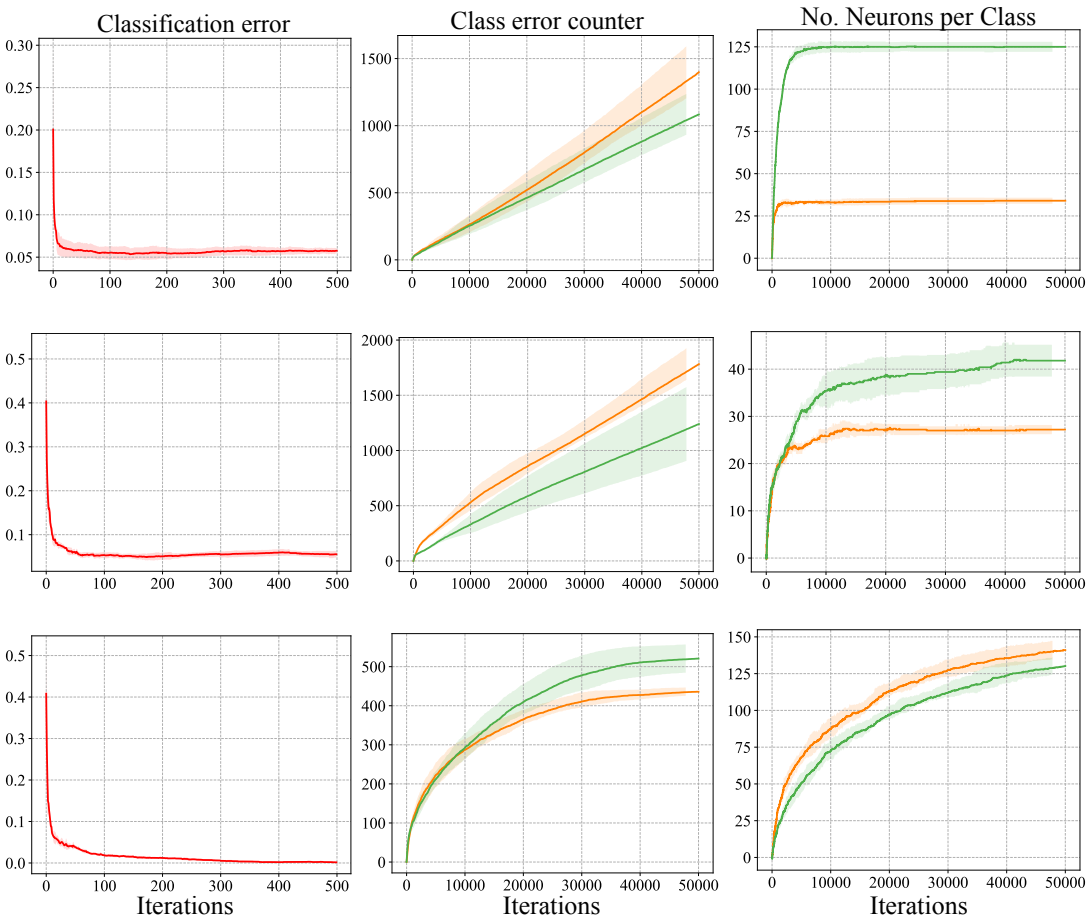


Figure 7.2: More details on the effect of different neuron insertion conditions during training on the nested clusters dataset. Each plot shows the average and the standard deviation of the results of 5 trials, each time randomly shuffling the input order. The first row shows results for the GWR with the original insertion condition, the second row shows results using both neuron insertion conditions, and the last row, using only the classification error. The first column shows the classification accuracy measured every 100 iterations, the second and the third columns show the error counter for each class and the class-specific neural growth respectively, during each learning iteration.

which are distributed evenly over the data samples. In Fig. 7.1.c, on the other hand, where the classification error is the sole insertion strategy, we can observe a greater number of neurons placed near the two clusters border, i.e., where most of the misclassifications occur.

Figure 7.2 illustrates the average classification error, the neural growth of each network during training and the class error counter, which is increased every time the label of the BMU is not matching the input's label. In the case of the GWR with the original neuron insertion condition, the neural growth stops way before

the first 10.000 learning iterations, i.e., the first epoch, albeit the growing error counter of both classes. In the same way, the classification error converges quite fast towards 0.05 but is not decreased anymore throughout learning. The effect of applying both insertion conditions is, in this example, slightly different. We can observe a smaller number of neurons created for each class and the growth does not halt but tries to counteract the growing error counter. However, this is not enough for the classification error to reach the 0 value. In the third case, we see that the error counter is much smaller for both classes and becomes constant after 50 epochs, whereas the classification error converges to almost 0 after 30 epochs. The neural growth is slower than the other two cases and does not stop until reaching the best performance.

The GWR network that bases its neural growth only on the classification error will tend to be small once it starts classifying all training samples correctly. When the classification error does not equal zero, on the other hand, the network will grow continuously. Thus, from all insertion conditions described so far, it seems more convenient to combine both the quantization error with the classification error while keeping the insertion threshold high, e.g., $a_T = 0.99$ for the current experiments. This would allow for the higher density of neurons in the regions where most misclassifications occur while guaranteeing that, at least, all the training data have a good prototype representation. Moreover, when the network has learned to represent the input data in the best way possible, the growth will stop even though misclassifications may still take place. It should be noted that the sensibility of the network's growth with respect to the value of the insertion threshold parameter is more relaxed. Finding an optimal value for this parameter is no longer necessary for maximizing the classification performance of the model, as long as both insertion conditions are used.

7.2.2 Modulating Neural Growth in a Hierarchical Architecture

The application of the proposed neuron insertion strategy in a hierarchy of GWR networks equipped with hierarchical spatiotemporal learning (see Section 4.4) may lead to: (1) a greater number of neurons in areas with higher class uncertainty, as previously demonstrated, and (2) the modulation of the neural growth of the GWR network in the lower layer of the hierarchy, thus optimizing resources according to the task being solved. The second outcome is achieved by simply propagating the delayed classification error information from the top layer to the previous one. We

assume that when one action segment is not well classified, the representation of the spatial data learned by the first layer needs to be adapted accordingly. If we take the example of the hierarchical learning of human actions, for instance, the neural growth of the first layer which processes body postures can be modulated by how well the second layer is classifying actions. The more subtle the differences between the sequence segments given in input to the second layer, the denser will be the first layer areas representing the constituent body postures, and, as a result, the more fine-grained and distinctive will become the subsequently generated spatiotemporal representations. It is expected that the neural structure and the internal representations learning through this bilateral process will vary depending on how well the current training data samples are being classified rather than based on a quantization error dependent neuron insertion threshold which doesn't change during learning.

Now, we demonstrate the validity of our hypotheses with an example. Fig. 7.3 illustrates the neural placement and the dynamics of the hierarchical architecture in terms of the class error counter and the number of class neurons during training on the Transitive Actions dataset introduced in Section 4.2.1. For comparison, we illustrate in Fig. 7.4 the results of the same experiment when the neural growth is driven by the quantization error and no top-down modulation is applied. Both architectures have been trained for 10 epochs and the results have been averaged over 6 trials (the Transitive Actions dataset is composed of actions executed by 6 different subjects). In the first column of the two figures, the neuron weights have been projected to the 2D space by means of the LDA technique (Fukunaga, 2013). The parameters of the projection matrix have been adapted by projection pursuit to yield a maximum Gaussian separation of the prototypes in the two-dimensional target space (Friedman and Stuetzle, 1981).

The most evident difference in the two experimental results is the decreased number of neurons in both layers of the first trained architecture. Secondly, the class with the highest number of neurons in both layers is the one being mostly misclassified, i.e., the *picking up* action. We see that this is not the case with the second architecture, whose second layer dedicates more neurons to the *talking on the phone* action. Moreover, it reaches a stable number of neurons between the second and the fourth epoch independent of the increasing classification error for *picking up*. By analyzing the 2D-projected neuron weights we can observe a slightly better separability between the clusters in the second layer of the first architecture. Hence, in the unprojected original space, a good separability can be expected. This is confirmed by the experiment: the first architecture reaches a

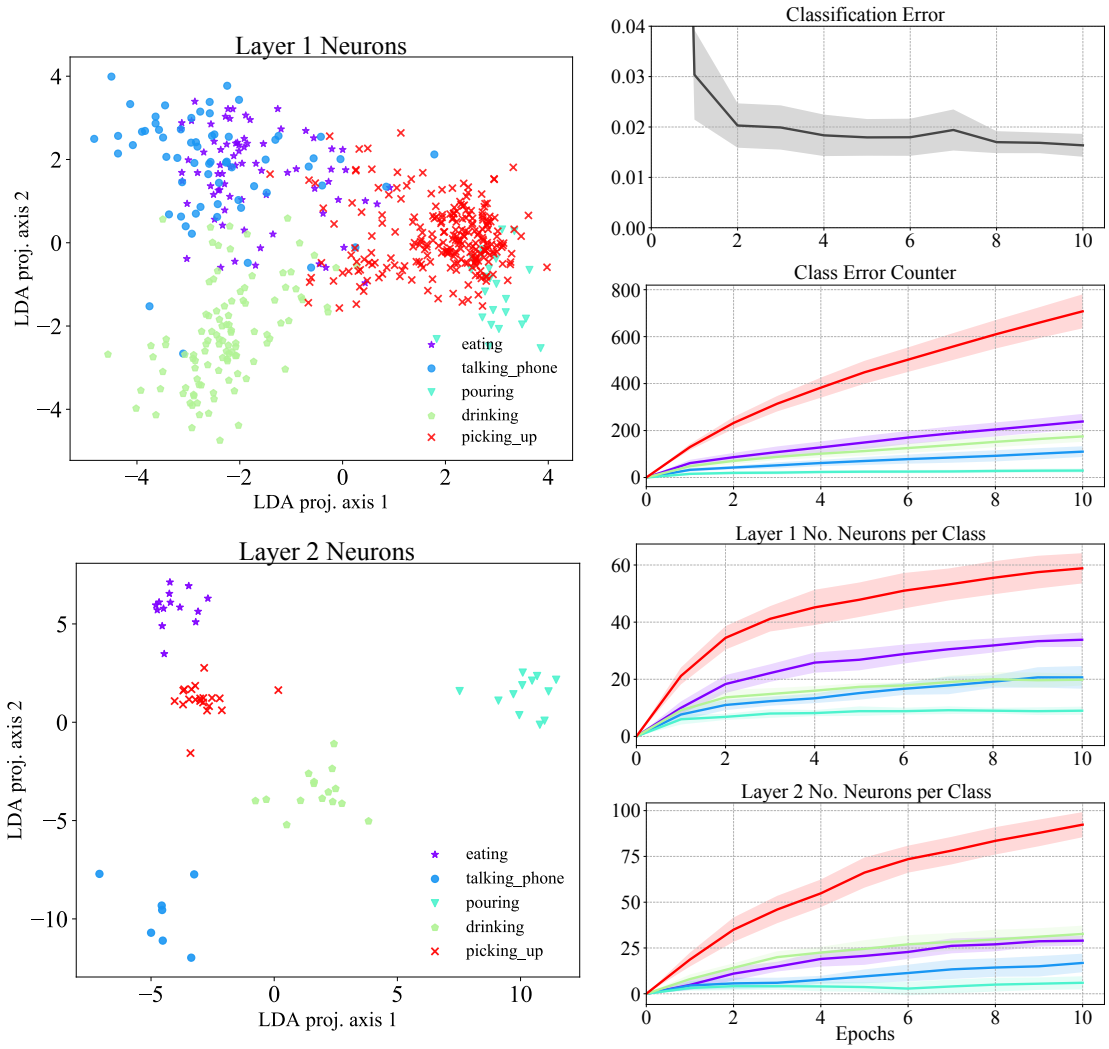


Figure 7.3: Experimental results on the Transitive Actions dataset when applying the top-down modulation mechanism. The LDA-projection of the weights is shown in the first column and the learning dynamics of the two-layer GWR hierarchy during training are shown in the second column. Each action category is illustrated with the same color in all the plots.

relatively low classification error by the end of the training session.

In Fig. 7.5 we illustrate the sensitivity of the architecture with respect to the misclassification threshold. We observe that the recognition rate for a threshold $m_T = 6$ is on average slightly higher than for $m_T = 0$ albeit the almost halved number of neurons. A misclassification threshold equal to 0 is equivalent to a hierarchy of GWR networks where neural growth occurs as soon as misclassifications occur. Looking at the standard deviation, the recognition rate oscillates from trial to trial and it might be higher than the average for lower thresholds. This means that increasing the threshold does not always result in better performance. This is

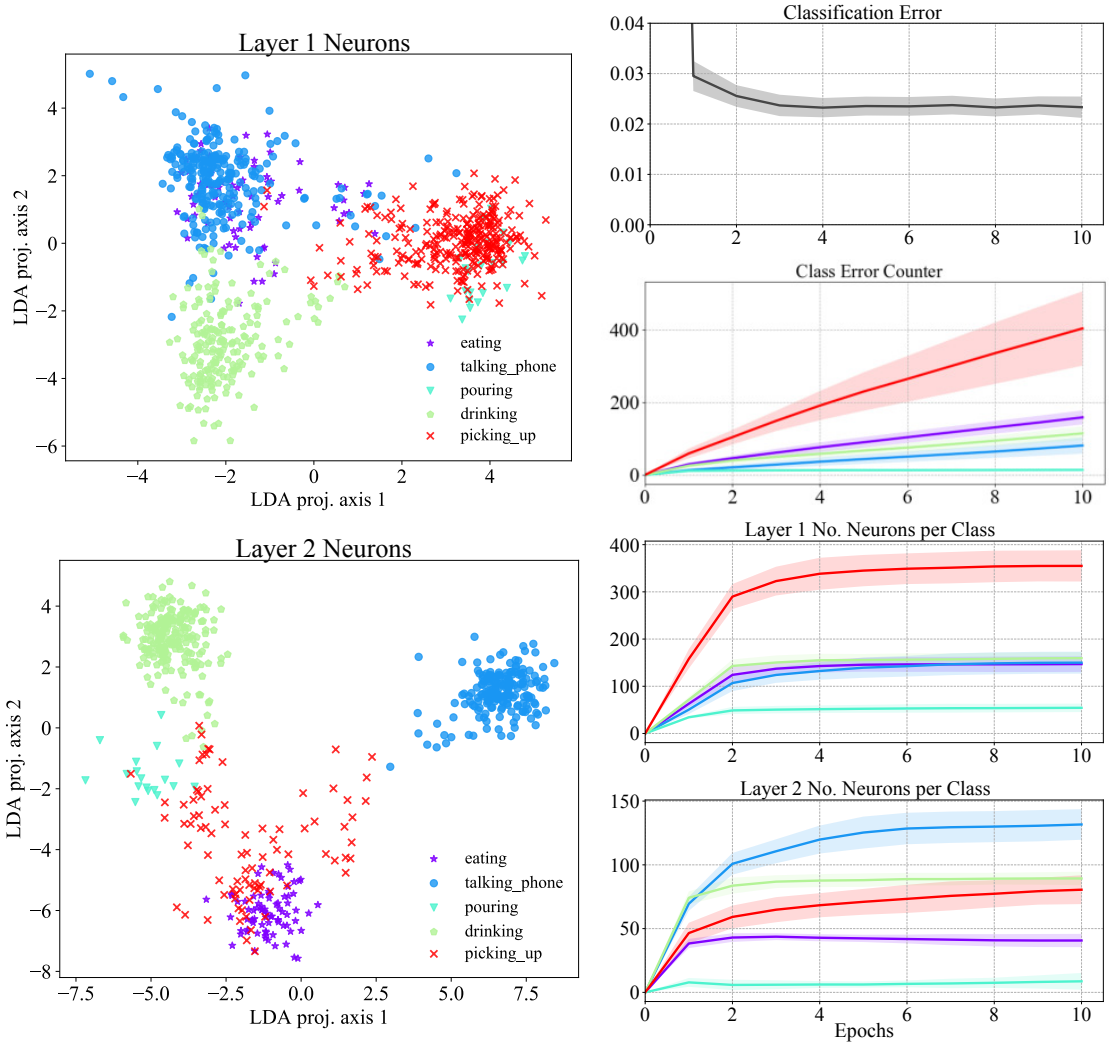


Figure 7.4: Experimental results on the Transitive Actions dataset when disabling the top-down modulation mechanism. Looking at the LDA-projection of the weights, we see that the trained neural architecture has a larger number of neurons than when the top-down modulation mechanism is applied (see Fig. 7.3). During learning, the neural growth for each action category is independent of the class error counter.

an understandable consequence of the fact that this top-down modulation mechanism does not aim at separating prototype neurons belonging to different classes, as in the case of the Learning Vector Quantization (LVQ) algorithm for instance, but rather concentrates resources in areas where classification is difficult. Moreover, the growth of the network becomes much slower with an increasing threshold and the available resources might not be enough to solve the classification task.

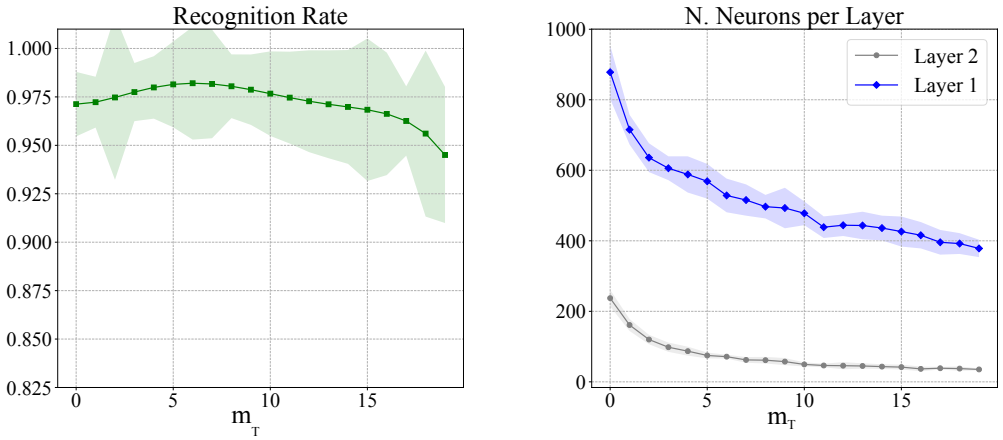


Figure 7.5: Sensitivity analysis of the misclassification threshold. The mean value and the standard deviation are computed over the 6 folds of the Transitive Actions dataset.

7.3 An Architecture for Learning the Compositionality of Human Activities

We now present a hierarchical architecture extended with the so far described top-down modulation mechanism for learning human actions on two levels of semantic and temporal complexity: 1) *atomic actions* such as *reaching*, *opening* which are completed in a relatively short period of time, and 2) the high-level *activities* that can be composed of different atomic actions. An overall diagram of the architecture is shown in Fig. 7.6.

The GWR_b, GWR_o, and GWR_1 networks process and subsequently integrate the body pose and the information about the manipulated object(s), while the GWR_2 network integrates spatiotemporal dependencies over longer time windows and learns to classify human activities. Both the GWR_1 and the GWR_2 networks capture different temporal ranges of actions by the accumulation of body movement patterns over a short and a longer time period respectively. The feedforward hierarchical computation of the spatiotemporal inputs is identical to the one introduced in Section 4.4.1. Besides the identity of the manipulated objects, we consider additional visual features capturing the object-object and object-body spatial relations, as will be described in Section 7.4.1.

We introduce delayed feedback connections and extend the traditional GWR learning algorithm with the proposed top-down modulation mechanism. Thus, during training, the error regarding the misclassification of the atomic actions and of the activities is propagated not only to the GWR_2 and GWR_1 respectively,

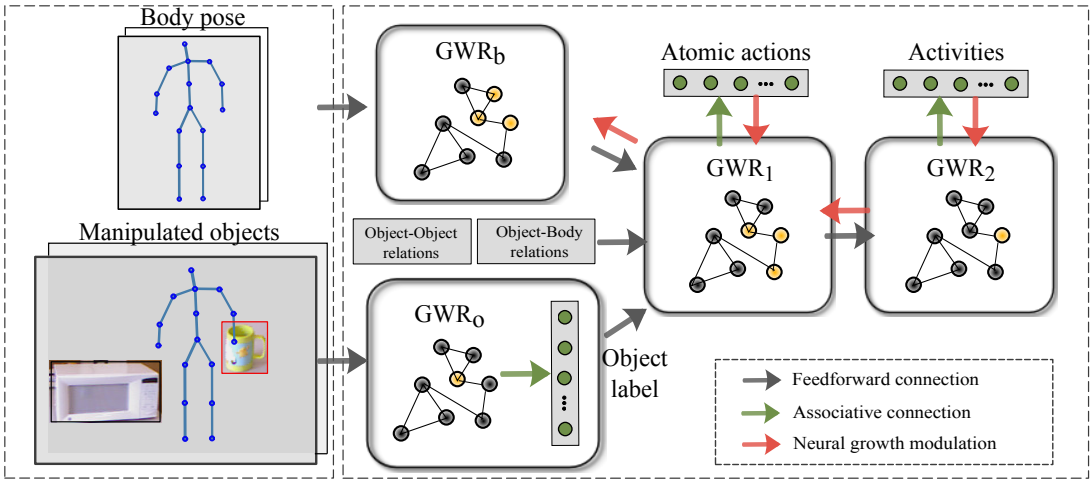


Figure 7.6: Illustration of the neural architecture for learning the compositionality of the human activities. This model extends the self-organizing architecture proposed in Section 4.4 with: 1) an additional network layer that captures spatiotemporal dependencies over longer time windows, 2) two associative connection matrices between GWR_1 and the atomic actions symbolic layer and between GWR_2 and the activities symbolic layer, and 3) the top-down modulation mechanism that modulates the learning of the GWR_b, GWR_1, and GWR_2 networks. Additional visual features capturing the object-object and object-body spatial relations are also provided as input to the current architecture.

but also to the network layers preceding them (the feedback connections are depicted with red arrows in Fig. 7.6). This is done in order to: 1) allow changes of the topological structures for all the body processing GWR networks, and 2) to better match the input space for jointly learning the atomic actions and the high-level activities. We apply the proposed neuron insertion strategy to each network layer. Notice that the action classification error is not propagated to the object recognition module which provides the identity of the manipulated objects at the beginning of each action sequence.

7.4 Experiments with the CAD-120 Dataset

We run experiments with the CAD-120 benchmarking dataset of human activities previously used to report activity recognition performances in Chapter 4. This dataset provides 120 videos of 10 long daily activities composed of a varying number of atomic actions (see Table 7.1). The dataset is challenging in the following aspects: 1) The activities in the dataset are performed by four different actors, which behave quite differently, e.g., use left or right hand or follow a different order of atomic actions. 2) There is a large variation even for the same activity, e.g.,

Table 7.1: The high-level activities of CAD-120 in terms of atomic actions. The order of the atomic actions can be different for some activities. The high-level activities are learnt at the top-most layer of our architecture and the atomic actions are learnt by network layer 2.

Activity	Reaching	Moving	Placing	Opening	Closing	Eating	Drinking	Pouring	Cleaning	Null
Making cereal	✓	✓	✓					✓		✓
Taking medicine	✓	✓	✓	✓		✓	✓			✓
Stacking objects	✓	✓	✓							✓
Unstacking objects	✓	✓	✓							✓
Microwaving food	✓	✓	✓	✓	✓					✓
Taking food	✓	✓	✓	✓	✓					✓
Picking objects	✓	✓								✓
Cleaning objects	✓	✓	✓	✓	✓				✓	✓
Arranging objects	✓	✓	✓							✓
Having meal	✓	✓	✓	✓					✓	

the atomic action *opening* can refer to opening a bottle or opening the microwave. Although both of them have the same label, they appear significantly different from each other in the video. 3) As also shown in Section 4.2, occlusion is a critical issue for this dataset, e.g., in some of the videos, legs are occluded by the table, leading to completely unreliable leg tracks (see Fig. 4.3 and 4.4).

Since in this set of experiments we will use the object motion information provided by the dataset, an additional issue is presented by the objects being occluded by other objects (e.g., the pizza box is not tracked while inside the microwave) or not being tracked due to their small size, e.g., the apple object appearing in the *having meal* activity. This means that object location annotations provided by the dataset are often unreliable.

7.4.1 Adding Objects' Motion and Spatial Relationships

The recognition of human activities can be guided by the information regarding the objects involved and the way their spatial relationships change over time. For instance, putting a pizza box inside the microwave indicates that the person is microwaving food or bringing the cup towards the mouth indicates that the person is drinking. The use of objects' spatial relationships as visual features, though, raises an important question: how can such features be invariant to the scene despite the varying number and type of objects appearing in it?

One way to represent object relationships is through the scene graphs proposed by Aksoy et al. (2017). However, their approach requires the manual definition of discrete labels for spatial relations. In contrast, our goal is to keep continuous position values. Thus, we take the tracked position of the objects and form a

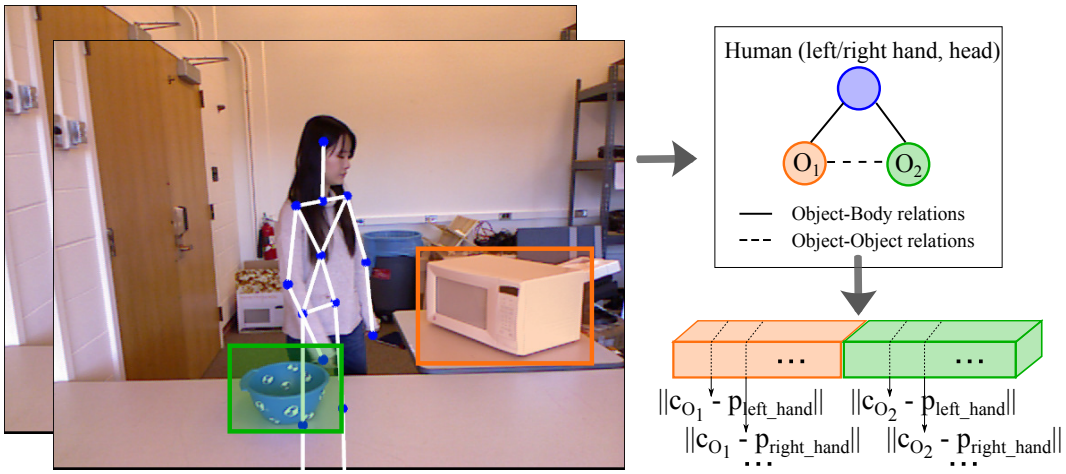


Figure 7.7: An illustration of how we represent the spatial relationships between objects and humans in a scene from the CAD-120 dataset. We extract the three-dimensional centroids of the objects, c_{O_1} and c_{O_2}, and compute the Euclidean distance between them and the *left hand*, *right hand*, and *head* joints. This is then concatenated to the Euclidean distances between the objects' centroids. In this example, the person first interacts with the microwave and then with the bowl. Hence, the tracks of the microwave will take the first place in the concatenated vector of the spatial relationships.

vector whereby the order is given by the manipulation order, e.g., if the activity sequence is composed of *opening* (microwave) $\rightarrow$ *moving* (bowl) into the microwave, then the object motion vector will contain the microwave tracks concatenated to the bowl's tracks (see Fig. 7.7). From the x, y coordinates given in pixels for the left upper corner and right bottom corner of the bounding boxes surrounding each tracked object, we extract the three-dimensional centroids from the corresponding depth image patches. We capture the body-objects relationship by computing the Euclidean distance between the centroid of the objects to the left hand, right hand, and the head joints of the body skeleton. The object-object relationships are computed as the Euclidean difference between the three-dimensional object centroids. To capture the objects' motion information, we compute the mean velocity and the displacement of the object's centroid along the x, y and z axis across consecutive video frames.

It should be noted that our representation of the objects' motion and spatial relationship comprises only a fraction of the input features used by the related work on the CAD-120 dataset. This is due to the fact that the input features provided by the dataset authors are suitable for learning with graphical models, such as the conditional random field (CRF) model. For instance, some features

about the objects' relative positions are provided for the first, middle and the last frame of the temporal segments, which are extracted before training the model. Unlike these methods, both the learning and the recognition phase of our architecture are performed on a continuous stream of input data and no prior temporal segmentation of the atomic actions is necessary.

7.4.2 Impact of the Top-down Modulation During Training

Now we evaluate the architecture described in Section 7.3 by running experiments with the CAD-120 dataset under two conditions: 1) considering only the architecture's feedforward connections and using the standard GWR neuron insertion strategy, and 2) considering both feedforward and top-down connections, thus applying the proposed neural growth modulation mechanism. For the first experimental setup the architecture is trained through the hierarchical learning strategy described in Section 4.4.1, thus the training remains unsupervised. For the second setup, at each learning iteration, the delayed classification errors of the activities and atomic actions are propagated from the semantic layer to the GWR_2 and GWR_1, respectively, and to the network layers preceding them.

The visual features computed from the skeletal body tracks and the RGB images of the manipulated objects are the same as the ones described in Section 4.3. For each experimental setup, we run 4 trials, each time leaving one subject out of training, and average the obtained results. We empirically set a time window width of $q = 30$ and a lag $\xi = 5$ for the GWR_1 network and $q = 5$, $\xi = 1$ for the GWR_1. Thus, the first network has a temporal depth of 3 seconds, given that the data has a frame rate of 10 fps due to the median filter applied every 3 frames for attenuating noise. The average duration of an atomic action in the CAD-120 dataset is around 3 seconds. The GWR_2 network will have a temporal depth of 3.5 seconds thus developing spatiotemporal segments representing frames from at least two atomic actions.

The recognition rates of the GWR_1 and GWR_2 networks during training for both experiments are illustrated in Fig. 7.8. The neural growth of the body pose processing networks is illustrated in Fig. 7.9. We can observe that the impact of the feedback connections on the neural growth are similar to what we saw with the Transitive Actions dataset. This means that the proposed top-down modulation mechanism scales up to deeper architectures with a more complex input. As can be seen from Fig. 7.9, the number of neurons developed during learning for the second experimental setup (illustrated in red) is significantly lower than for the first setup. Most importantly, the reduced number of neurons does not compromise

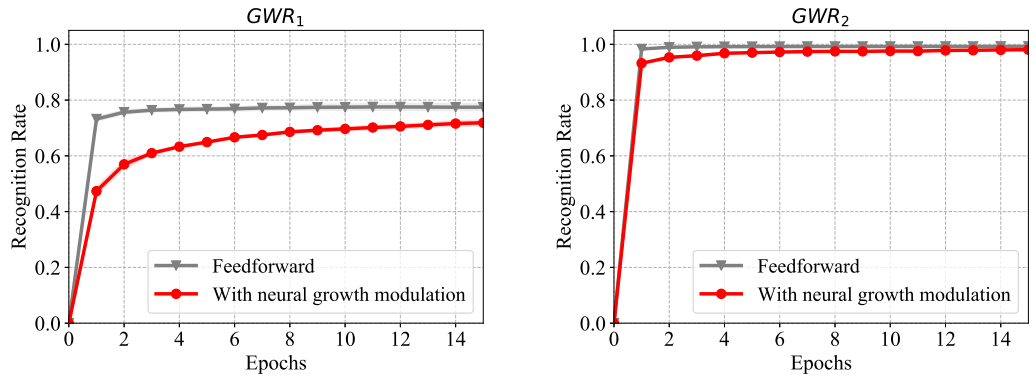


Figure 7.8: Comparison of the classification results on the training data of CAD-120 when training is conducted only with a feedforward input stream and when using the proposed neural growth modulation. a) The accuracy of the GWR_1 network which learns to classify atomic actions, and b) the accuracy of the GWR_2 network which learns to classify the activities. The results are averaged over 4-fold cross-validation experiments.

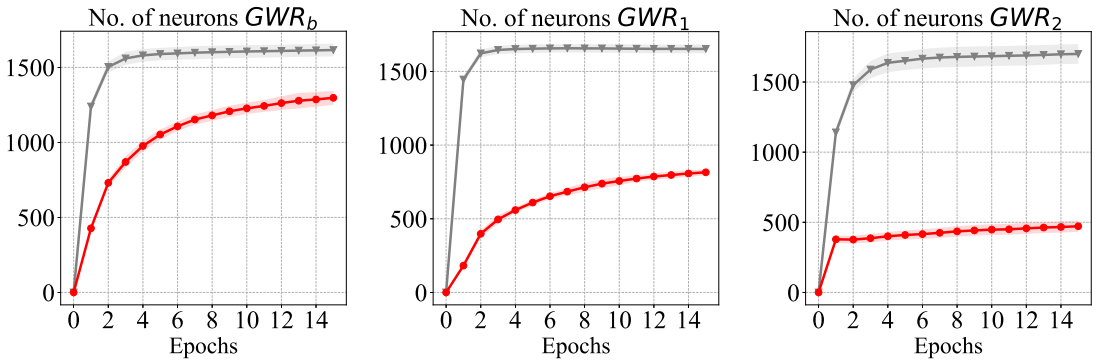


Figure 7.9: The number of neurons over the training epochs for the GWR networks with and without the top-down neural growth modulation. The results are averaged over 4-fold cross-validation experiments.

the classification accuracy of the activities. For the recognition of the atomic actions, on the other hand, the experimental setup with the feedback connections results in a slightly lower accuracy. One reason for this might be the fact that the two classification errors regarding the atomic actions and the activities are simultaneously intervening on the topographic organization of the GWR_2 network causing this slight performance decay. Another reason might simply be that the segmentation of the atomic actions of this dataset contains errors, thus causing higher confusion among classes. A few examples illustrating the second hypothesis will be shown in the following section.

Without ground-truth segmentation						
	Sub-activity			Activity		
Algorithm	Acc.(%)	Prec.(%)	Rec.(%)	Acc.(%)	Prec.(%)	Rec.(%)
Koppula and Saxena (2013), (*CRF*, *SVM*)	**70.3 ± 0.6**	**74.8 ± 1.6**	**66.2 ± 3.4**	83.1 ± 3.0	87.0 ± 3.6	82.7 ± 3.1
Koppula et al. (2013), (*CRF*, *SVM*)	68.2 ± 0.3	71.1 ± 1.9	62.2 ± 4.1	80.6 ± 1.1	81.8 ± 2.2	80.0 ± 1.2
Hierarchical feedforward, (GWR)	45.9 ± 3.8	45.0 ± 4.2	55.9 ± 7.1	**92.0 ± 3.6**	**92.5 ± 4.1**	**91.7 ± 3.7**
Rybok et al. (2014), (*SVM*)	-	-	-	78.2*	-	-
Tayyub et al. (2015), (*SVM*)	-	-	-	75.8 ± 6.8	77.9 ± 11.0	75.4 ± 9.1
With ground-truth segmentation						
Koppula and Saxena (2013), (*CRF*, *SVM*)	**89.3** ± 0.9	87.9 ± 1.8	**84.9** ± 1.5	93.5 ± 3.0	95.0 ± 2.3	93.3 ± 3.1
Koppula et al. (2013), (*CRF*, *SVM*)	86.0 ± 0.9	84.2 ± 1.3	76.9 ± 2.6	84.7 ± 2.4	85.3 ± 2.0	84.2 ± 2.5
Hierarchical with top-down, (GWR)	43.8 ± 3.4	41.3 ± 3.1	58.6 ± 6.1	93.5 ± 3.2	94.4 ± 3.4	93.3 ± 3.3
Hu et al. (2014), (*CRF*, *SVM*)	87.0 ± 1.9	**89.2** ± 4.6	83.1 ± 2.4	-	-	-
Tayyub et al. (2015), (*SVM*)	-	-	-	**95.2 ± 2.0**	**95.2 ± 1.6**	**95.0 ± 1.8**

Table 7.2: Classification results on the action hierarchy of the CAD-120 dataset. Reported are accuracy, precision and recall (in percentage) averaged over the 4-fold cross-validation experiments. ∗Note that Rybok et al. (2014) have not provided the standard deviation of their results.

7.4.3 Comparison to the Other Approaches

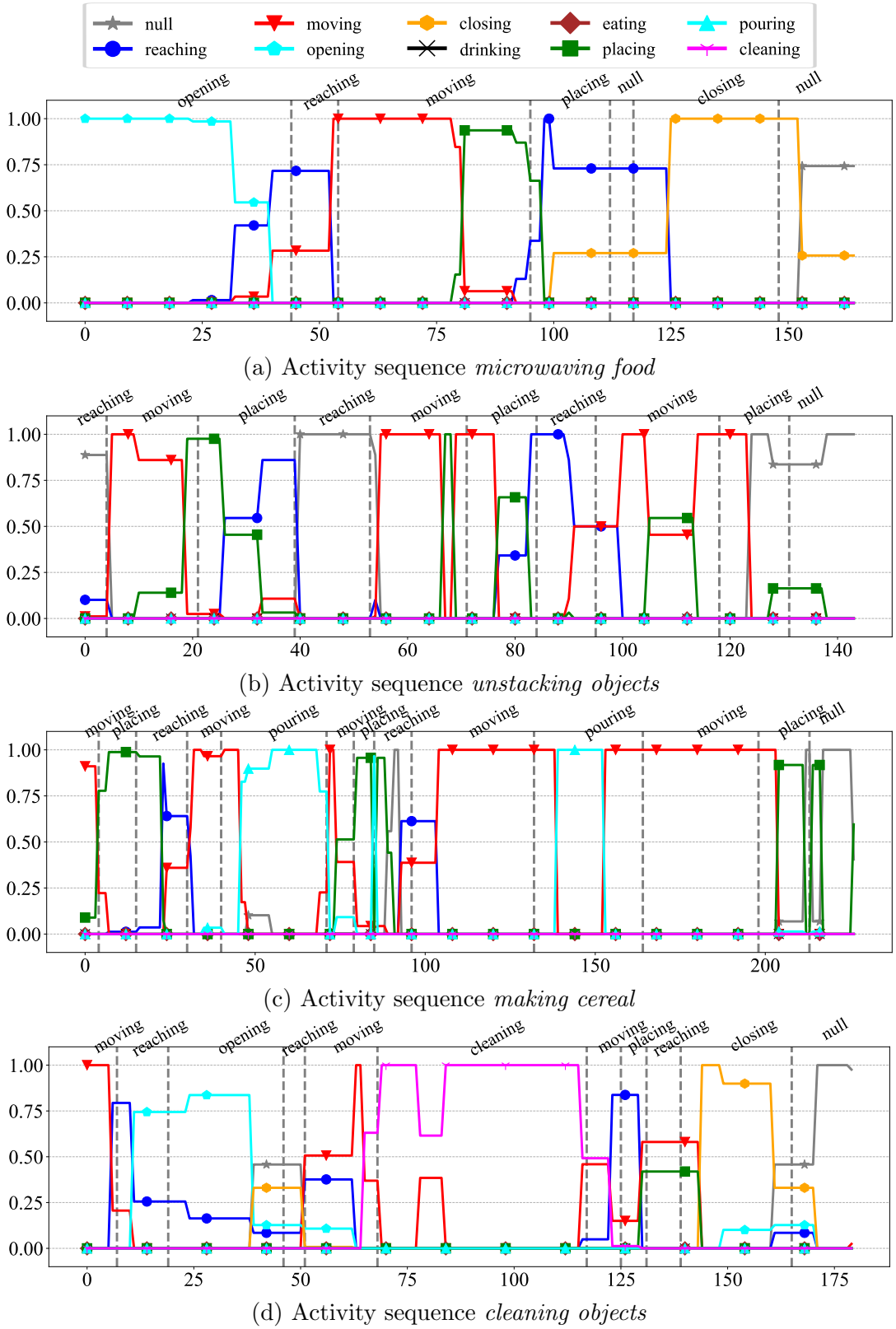
In Table 7.2, we report the accuracy, precision, and recall of our two models on both the atomic actions and the high-level activities of the CAD-120 dataset. We also compare our results to the other approaches on this dataset. Note that the authors of the dataset refer to the atomic actions with the name *sub-activities*. We report both the average values of the performance measurements as well as the standard deviation across the 4 validation folds. Our model equipped with the top-down modulation mechanism has been listed among approaches using ground-truth segmentation, due to the fact that we use the sub-activity labels during training to modulate the learning of the GWR_1 network. The model with only feedforward connections does not use the sub-activity labels for modulating learning but associates them with each neuron for evaluation purposes. The direct comparison of the results on this table needs some caution though. The other approaches use the input features provided by the authors of the dataset, which are computed at each ground-truth temporal segment, whereas in our approach the features are computed continuously at each video frame.

We observed that the model with top-down connections shows a better perfor-

mance regarding the classification of high-level activities and a slight decrease of the accuracy and precision on the sub-activities. Yet, the feedforward model performs better than state of the art on the high-level activities albeit the relatively low recognition accuracy on the sub-activities. This indicates that our approach does not require a fine-grained manual segmentation and a successful recognition of the atomic actions in order to correctly classify high-level activities. The reasons for the low accuracy on the sub-activities for both models need to be further investigated.

We visually analyzed the output labels of the GWR_1 network on the activity sequences from the test sets at each cross-validation trial. In Fig. 7.10, we illustrate some examples from the *unseen* subject 1. Each subfigure illustrates one activity sequence and the frame rate is of 10 fps. The ground-truth temporal segmentation provided by the dataset is depicted with vertical gray dashed lines and each plotted line interpolates the output of the best-matching neuron representing each video frame. An output of 1 indicates that the BMU has *one* Hebbian connection with non-zero weight towards that particular category label, whereas multiple lines indicate that the BMU is connected to multiple category labels in the semantic layer. The second case happens when the neuron has matched spatiotemporal segments belonging to different categories during training and this may be due to either the similarity in the feature space of these segments, due to the incorrect manual segmentation of the atomic-actions or due to the pre-defined temporal window including several atomic actions in it. The second reason is not to be excluded given that the segmentation of the atomic actions in this dataset is particularly fine-grained. In Fig. 7.10.c, for instance, we can see that the activity *making cereal* is composed of 10 atomic actions in only 100 video frames (corresponding to 10 seconds). There is a considerable overlap between the atomic actions of *reaching*, *moving*, and *placing*. These actions compose more than half of the instances of the CAD-120 dataset.

From the examples reported in Fig. 7.10 we can also observe the different temporal borders between the recognized atomic actions and the ground-truth segmentation. Again, the correctness of the ground-truth segmentation plays a role here. In Fig. 7.10.a, for instance, the sequence of atomic actions is *opening* (microwave), *reaching* (for an object), *moving* (the object), *placing* (the object inside the microwave), *null* (no action), *closing* (the microwave), and then *null*. In this example, the ground-truth segmentations did not take the atomic action *reaching* (the microwave) into account, which is, for instance, not the case in Fig. 7.10.d where there is *reaching* and then *opening* (the microwave). In the example re-



(a) Activity sequence *microwaving food*

(b) Activity sequence *unstacking objects*

(c) Activity sequence *making cereal*

(d) Activity sequence *cleaning objects*

Figure 7.10: Output labels of the GWR_1 network (atomic actions) for the test subject 1 of the CAD-120 dataset. The ground-truth temporal segmentation of the atomic actions is illustrated with vertical gray dashed lines. The ground-truth atomic action labels are reported on top of each plot.

ported in Fig. 7.10.a, however, although with incorrect temporal boundaries, the sequence of labels output from our model is plausible.

Finally, in comparison to the other approaches in Table 7.2, the proposed feed-forward model seems more advantageous than the model with the top-down modulation. However, for applications where human activities need to be learned incrementally during the lifetime of an intelligent agent, the second model provides a trade-off between high recognition rates and the optimization of the neural resources.

7.5 Summary

In this chapter, we presented a hierarchical self-organizing architecture for learning hierarchical representations of the human-object interactions. The architecture builds on top of the hierarchical learning scheme, successfully applied in the other chapters of this thesis, and is further equipped with a top-down mechanism for the modulation of the neural growth of each body feature processing GWR network of the hierarchy. In particular, we focused on analyzing in detail the learning effects of the proposed top-down mechanism by conducting experiments with both synthetic data as well as two real-world datasets composed of human-object interactions. Overall, we saw that the application of this mechanism can lead to the creation of a considerably low number of neurons and to a higher concentration of neurons in the areas where classification is harder.

The experimental results with the CAD-120 dataset demonstrated that the proposed architecture outperforms the state-of-the-art approaches with respect to the classification of the high-level activities. Experiments also showed that the average recognition accuracy for the atomic actions was lower than the other approaches and we analyzed a few possible reasons for this. Unlike the other methods, the proposed architecture operates on a continuous stream of information and the temporal boundaries between atomic actions are certainly hard to determine. However, this seemed to not affect the overall activity classification performance indicating that our approach is not sensitive to the correct segmentation and classification of the atomic actions. Moreover, a qualitative analysis of the atomic action labels generated by the architecture on the test data sequences showed that semantically meaningful representations had emerged. Thus, the reported results motivate further applications of the proposed architecture on other datasets for the learning of the compositionality of the human activities.

Chapter 8

Discussion and Conclusions

8.1 Summary of the Thesis

In this thesis, we focused on the recognition and the prediction of human daily activities, which are two crucial perception tasks required for establishing a natural and efficient interaction between an assistive system and a human (Vignolo et al., 2017). We presented a number of neural network architectures that learn spatiotemporal representations of human-object interactions from sequences of RGB and depth maps. First, we proposed an architecture motivated by neurobiological findings for the processing and integration of the body pose sequences and the manipulated objects. The architecture considered the object's identity as an important contextual information for disambiguating similar movement patterns of different action categories. The experimental results showed that the recognition accuracy of our approach is competitive with respect to supervised state-of-the-art approaches on a challenging benchmark dataset of high-level human daily activities. The architecture was then extended with a temporal association mechanism based on Hebbian learning in its simplest form in order to also address the anticipation of human-object interactions. We showed how the architecture could capture long-term temporal dependencies by both evaluating the action prediction accuracy on a transitive actions dataset and analyzing the learned spatiotemporal representations during the closed-loop generation of the learned action sequences.

Taking advantage of the online and incremental learning capability of the GWR algorithm, we proposed a neural framework for the learning and prediction of human motion which was then applied for the online sensorimotor delay compensation of a mid-size humanoid robot during an imitation task. Visuomotor sequences of arm movements were extracted in the form of joint angles, which could be directly mapped to the robot. Experimental results showed that our system can achieve

low prediction error values while being trained in an online manner.

Finally, we investigated a top-down modulation mechanism for the optimization of the architecture's neural growth during the learning of human-object interactions. In particular, we looked at the application of the modulatory signal during the emergence of hierarchical representations of the human activities. Experimental results demonstrated that the top-down mechanism provided a way to optimize neural resources according to the classification task rather than based on the quantization error, thus preventing the creation of an unnecessarily high number of neurons. Such results encourage the application of our method in autonomous systems that learn from continuous sensorimotor experiences based on limited processing and memory resources.

8.2 Discussion

8.2.1 Mapping Actions to Objects

Our first research question was how can relationships between objects and human motion patterns learned in an unsupervised manner from image sequences of human-object interactions. For this reason, in Chapter 4 we proposed a self-organizing hierarchical architecture composed of two network streams, each one processing the body pose features and the manipulated objects respectively. The GWR network that processed the objects' visual appearance was further connected to a semantic layer containing object category labels. Thus, the identity of the manipulated object was then concatenated to the prototype spatiotemporal segments representing body motion during the manipulation of that particular object. This is in line with the biological findings suggesting that there are separate functional and anatomical pathways processing the information about biological motion and man-made objects and that the identity of the object is crucial for a full understanding of a human activity. The use of prototype-based representations for objects is motivated by psychological studies on the nature of human categorization (Rosch and Mervis, 1975), suggesting that categories are typically learned as a set of prototypical examples and the similarity, or the so-called family resemblance, is used for class association.

Apart from the biological motivation, this solution benefits from the advantages of the modular architectures, for instance, when a new object instance is available (a box of a new brand of cereal) only the object recognition module should be trained and not the entire architecture. Consequently, the new learned

object will be associated with a previously learned action, e.g., *pouring* cereal. When considering hybrid architectures, it is quite straightforward to introduce an object recognition module with a higher classification accuracy, for instance, based on a state-of-the-art CNN architecture. As long as the output of the applied object recognition module is of the form of one-hot encoding, i.e., a vectorial representation in which all elements are zero except the ones with the index corresponding to the recognized object category, the integration of the information can be performed exactly in the same manner as when using the proposed dense SIFT- and GWR-based object recognition module. Another advantage of this approach is the possibility to retrieve the information from the learned pairs, i.e., it is possible to tell on what object a certain action can be performed as well as retrieve the body motion patterns that can be executed for manipulating a given object. The latter was demonstrated in Chapter 6 and it was achieved through an extension of the recognition architecture with a temporal association mechanism that allows for recalling the learned action sequences.

The proposed multi-stream neural architecture was shown to be competitive with the state-of-the-art approaches on a real-world dataset composed of long human daily activities performed in cluttered environments. Furthermore, it allowed us to extend our experiments beyond the classification of human activities and analyze, for instance, neural responses when the input is composed of incongruent action-object pairs. Interestingly, these experiments demonstrated a behavior resembling the action-selective neural circuits which show sensitivity to the congruence of the action being performed on an object (Yoon et al., 2012). Moreover, it allowed us to study the relevance of different contextual information such as the identity of the object and the spatial relationships between manipulated objects and the body in the scene. In Chapter 7, we saw that spatial relationships can help discriminate short atomic actions such as *pouring* and *opening*.

The association of an action with an object can be seen also as learning the functional properties of the objects or their action possibilities, which, according to Gibson, define the so-called object *affordances*. However, in this thesis, we do not claim to have modeled such a concept. It is argued that object affordances in the human visual system are not defined by the object's category, appearance, and shape but rather on the actions a human can perform with/on them. A bottle and a kettle, for instance, can be both used for pouring water into a cup due to being two types of containers. A person can sit on a chair as well as on a rigid box that can hold the person's weight even though the box is not categorized as a chair and does not even look like one. In other words, low-level features

are associated with action possibilities without the need to have a fully detailed model of the objects nor to recognize them semantically. However, it is unclear what visual attributes of the human-object interaction define object affordances. It may be that the visual attributes are different for different types of actions, e.g., the tilting movement of a bottle that contains water may define the *pourable* affordance but the *sitable* affordance of a chair is defined by the anthropomorphic shape (Grabner et al., 2011). If the affordances are defined as the spatiotemporal pairwise relationships between objects and the relationships between the human and the manipulated object(s) (Pieropan et al., 2014a), then we might have used this definition of object affordances in the hierarchical architecture proposed in Section 7.3. Although humans do not need to recognize an object in order to perceive its affordance and act on it, the semantic object recognition information has been shown to modulate the execution of the manipulation actions in a top-down fashion (Goodale, 2008).

8.2.2 Self-Organizing Neural Learning

Generative approaches based on self-organization learn an input probability distribution through a finite set of reference vectors associated with neurons. Moreover, they resemble the topological relationships of the input space through the neurons' organization. Growing self-organizing approaches such as the GNG (Fritzke, 1995) and the GWR networks (Marsland et al., 2002) are characterized by a dynamic topological structure able to adapt to the input data space through the mechanism of the competitive Hebbian learning (Martinetz, 1993). Unlike the GNG, where the network grows at a constant rate, the GWR algorithm is equipped with a learning mechanism that creates new neurons whenever the current input is not well represented by the prototype neurons. Thus, from the perspective of incremental learning, the GWR algorithm is more suitable than the GNG since new knowledge can be added to the network as soon as new data become available.

The parameters modulating the growth rate of a GWR network are the activation threshold and the firing counter threshold. The activation threshold establishes the maximum discrepancy between the input and the prototype neurons in the network. The larger we set the value of this parameter, the smaller is the discrepancy, i.e., the quantization error of the network. The firing counter threshold is used to ensure the training of recently added neurons before creating new ones. Thus, smaller thresholds lead to more training of existing neurons and the slower creation of new ones, favoring better network representations of the input. Intuitively, the less discrepancy between the input and the network representations,

the smaller is the input reconstruction error and this is a necessary condition during human motion prediction as shown in Chapter 5. However, less discrepancy means also more neurons. This proved to be not the main issue in our motion prediction experiments since the number of neurons did not affect significantly the computational complexity of the prediction function.

When it comes to the classification of the actions, there is no straightforward relationship between the number of neurons and the classification performance (see Section 7.2.1). In this case, the activation threshold should be set empirically in order to increase the classification performance. From this point of view, there is no clear advantage of the growing self-organizing networks with respect to networks with a fixed number of neurons, e.g., the SOM or the NG network. For the latter architectures, defining a smaller number of neurons also means that the resulting quantization error will be higher but not necessarily that the classification accuracy will decrease. The advantage of the growing self-organizing networks becomes clearer when the error measure used as the basis for neuron insertion is optimized for the task at hand (Fritzke, 1996). The arbitrary choice of an error measure for the neural growth is a central property of the growing models. As was shown in Chapter 7, using the classification error for the GWR neural growth concentrates network resources on the areas where classification is more difficult, thus avoiding the unnecessary insertion of neurons where the classification is correct.

8.2.3 Feature Extraction

One drawback of the self-organizing neural frameworks proposed in this thesis is the need to extract features from the raw images using other methods in order to perform classification and prediction tasks. For the processing of human body motion we rely on the extraction of a 3D skeleton model, from which, in some cases, we compute features describing body pose while maintaining a low-dimensional feature space and achieving scale and view-invariance. For the processing of the manipulated objects, we rely on a number of computer vision techniques, such as the object segmentation and tracking and the dense SIFT features for discriminating among objects. An end-to-end approach whereby features are extracted from the raw RGB and/or depth images through a hierarchy of self-organizing networks can also be implemented (Miikkulainen et al., 2006; Parisi et al., 2017a; Hankins et al., 2018). However, the performance of such models has been demonstrated on static images containing only objects or sequences of gestures and full-body actions. How well these approaches perform on image sequences of fine-grained human-object interaction activities needs further investigation.

More sophisticated feature extraction approaches are also offered by state-of-the-art pre-trained convolutional neural networks which are available as libraries in python deep learning APIs (e.g., Keras[1]). Rapid advances of this technology have led to models with outstanding performance in several classification tasks based on large-scale datasets in the wild. Thus, the application of this technology for feature extraction would presumably lead (although this needs to be proved) to a better performance for the recognition and prediction of human-object interactions in our architectures, especially considering the high amount of noise our models currently face due to the unreliable skeleton tracks.

It should be mentioned, however, that the skeleton body representations remain advantageous for tasks like body motion prediction and generation (Bütepage et al., 2017; Ghosh et al., 2017). Moreover, we saw in Section 5.3.2 that skeletal body representations are a low-dimensional description of the articulated human body that can be directly mapped to a humanoid robot during imitation learning. The extraction of the object motion and of the spatial relationships, of which we make use in Chapter 7, and the identification of humans/objects in one scene still remain perception tasks that require the application of separate dedicated deep learning architectures (Santoro et al., 2017; Johnson et al., 2017). This makes the application of the deep learning technology for the recognition of human-object interactions computationally more expensive (Guo et al., 2016; Ma et al., 2017), thus not ideal for real-time applications.

8.2.4 Hierarchies of Self-Organizing Networks

The architectures proposed in Chapters 4, 6, and 7 are hierarchical in the sense that visual information of body pose (and motion, when considered) was vector quantized prior to the integration with the objects and then prototype spatiotemporal segments were developed. One reason for this is to attenuate, to some extent, the noise of the body skeletal tracking which becomes considerable in the cases of body occlusion or when the subject touches objects in the background. The noise attenuation can be achieved by the GWR algorithm which is equipped with a mechanism to remove rarely activated neurons that may represent a noisy input. In addition to this, the firing counter of the neurons modulates the weights update, thereby leading to less learning perturbations of the well-trained neurons when slight input fluctuations occur. Another reason for the hierarchical arrangement of the GWR networks is to generalize with respect to the body motion. After training

[1]Keras: The Python Deep Learning library: https://keras.io/

the GWR on body pose frames, for instance, each neuron may still respond during slight joint translations, which may be due to noise or due to differences in motion execution between subjects.

The hierarchical arrangement of the GWR networks proposed in Chapter 5 has the advantage of increased computational efficiency by sharing neurons across multiple levels, e.g., prototype spatiotemporal patterns can be shared during the encoding of different motion sequences. This seemed to be an intuitive choice for the learning of visuomotor sequences for which resource-efficiency is desired. However, the extent to which neurons are reused is tightly coupled with the input distribution. In fact, in our experiments with input data samples represented as multi-dimensional vectors of both arms' shoulder and elbow angles, there was little to no overlap among training sequences. This led to the growth of the networks with each iteration over unseen sequences. One solution would be to implement a deeper hierarchy, similar to the work from Du et al. (2015), such that the whole skeleton is divided into parts, e.g., limbs, which then would feed separate network streams and develop spatiotemporal dependencies in parallel. The output of these network streams would then be combined to be the input of the *P-GWR* layer which then learns to predict future joint position or angle values. However, this solution relies on a correct tracking of each body limb. This is not the case in realistic scenarios whereby body tracking is affected by missing data due to body partial occlusions and self-occlusions (e.g., caused by a lateral view of the body). Thus, a holistic body representation is preferable due to its robustness to noise and missing information in the input data.

We followed a hierarchical learning architecture, meaning that the GWR networks in higher layers received as input the concatenation of the neural activation trajectories from lower-level layers. The temporally ordered neural activations obtained in this way resemble the sensitivity to the temporal order of the action-selective neurons in the STS area of the brain (Giese and Poggio, 2003). Interestingly, there is also neurophysiological evidence that actions are represented by sequences of poses over fixed temporal windows (Singer and Sheinberg, 2010). From the computational perspective, the sliding window technique allows for the extrapolation of spatiotemporal dependencies in the data sequences.

A limitation of the sliding time window technique for the encoding of temporal sequences is the high computational cost it introduces due to the data dimensionality increasing along the hierarchy. During our experiments with a skeletal representation of the human body, long time windows did not pose a computational challenge due to the low-dimensional input. However, for a high-dimensional in-

put, e.g., raw images, an alternative should be considered since, apart from the computational effort, the performance of the similarity measure degrades with the increasing data dimensionality. One alternative would be to apply the recurrent extensions of the growing self-organizing networks presented in Section 3.5.2. For a low dimensional input such as the skeleton body representation the sliding time window approach has been successfully applied also in other studies referenced in this thesis. Furthermore, it has been shown that long-term predictions based on a sliding window are more accurate than recurrent approaches (Bütepage et al., 2017).

The sliding time window allows for defining an arbitrary memory depth of the neurons at each level of the hierarchy, i.e., how far into the past the internal memory of each neuron stores information. This resulted in being quite useful for studying the compositionality of the action sequences, e.g., learning atomic actions and long human activities which are composed of atomic actions. A similar behavior can be obtained by applying a Gamma memory instead of the sliding time window computed at the output of each GWR layer. The γ-GNG (Estévez and Vergara, 2013) and the γ-GWR (Parisi et al., 2017a) models have an arbitrary number of temporal context descriptors and can be used thus in a hierarchical arrangement similar to the one presented in this thesis. However, for a low-dimensional input, the advantage of one approach over the other is not quite clear. Moreover, the Gamma models introduce additional hyperparameters that need to be optimized empirically (Estévez and Vergara, 2013) and the impact of these parameters on the final performance is not yet entirely understood.

8.3 Future Work

In this thesis, we saw how architectures built upon growing self-organizing networks can be quite flexible and can be successfully applied for the recognition and prediction of human activities in real-world scenarios. Thus, the obtained results motivate the extension of our approach into several future directions.

In our current work, we used depth information for the extraction of a three-dimensional skeleton model. We were motivated by the convenience of the depth sensing devices in real-time applications and the fact that they are the least computationally expensive method for motion segmentation and body pose estimation. However, the learning models based on tracked skeletons are susceptible to sensor noise and body occlusions. This issue becomes highly relevant during the segmentation of a body interacting with objects and can have a great impact on the

recognition and prediction accuracy considering the subtle hand/arm movements during object manipulation. For this reason, either the spatiotemporal features should be learned from depth images through a deep self-organizing architecture, similar to Parisi et al. (2017a), or hybrid architectures whereby action features are extracted through pre-trained deep CNN architectures should be applied. This would require, however, the application of additional (separation) mechanisms so that appropriate features are fed into each of the processing streams of the proposed architecture. Moreover, feature extraction with deep learning architectures has been shown to be either computationally expensive or it requires large amounts of training data.

The imitation scenarios presented in Chapter 5 were carried out in a simulated environment. Future studies with the real robot should address the introduction of overall body configuration constraints for learning the perceived motion. When the visual body tracking framework becomes unreliable, the provided body configurations may become unrealistic and cannot be mapped to the robot, or, in the worst case, when mapped to the robot may lead to hardware damage. For this reason, outlier detection mechanisms should be investigated in order to discard these unrealistic body configurations during training.

An additional future work direction would be to extend the neural framework presented in Chapter 5 towards the learning and imitation of manipulation tasks with a humanoid robot (Billard et al., 2016). The use of joint angles as visuomotor representations might be not sufficient in this case, since the visual feedback of the human demonstrator would include also the correct position and, perhaps, orientation of the manipulated object and not just the correct arm configuration. This issue can be addressed by including both the position information and the corresponding robot joint angles as input to the architecture. Due to the generative nature of self-organizing networks and their capability to function properly when receiving an incomplete input pattern, only the prediction of the object movement patterns would trigger the generation of corresponding patterns of the robot behavior. Moreover, the current results encourage further experiments towards learning by demonstration scenarios, whereby demonstrated motion patterns are stored and then recalled for the execution of different tasks with a robotic platform. For this reason, temporal association mechanisms and architectures similar to the ones presented in Chapter 4 can be extended to further consider the robot's hardware constraints.

So far, the proposed models consider only the visual stimuli of the human-object interactions. However, there are certain human-object interactions that cannot

be detected relying only on visual perception. It is, for instance, very difficult to detect whether a person is turning on an oven or boiling water with the kettle. One approach to tackle this limitation is to add other sources of information, such as for instance, the sound generated by the interaction. Multimodal learning of human actions and gestures has gained a lot of interest in the recent years (Stork et al., 2012; Teo et al., 2012; Parisi et al., 2016b). However, the audio-visual recognition of object manipulation actions has so far remained an open challenge (Pieropan et al., 2014b).

8.4 Conclusions

This thesis contributes to the field of visual recognition and prediction of human-object interactions with a set of self-organizing architectures that take inspiration from biological mechanisms of action perception. The proposed architectures stand among quite a few existing neural network approaches for the fine-grained understanding of human activities from video sequences.

Reported experiments showed that unsupervised learning with growing self-organizing architectures yields robust action-object representations, exhibiting comparable performance to the state-of-the-art, supervised, graph-based approaches. Such architectures can be extended to deal with both the recognition and the anticipation of human actions as well as human motion generation by applying similar underlying neural mechanisms. Symbolic labels of human actions, when available, can be used to modulate learning and can lead to compact spatiotemporal representations of the actions. Moreover, experimental results showed the robustness of the self-organizing architectures in learning streams of body motion information incrementally with no incurring overall performance decay.

Our findings demonstrate the suitability of the proposed approaches for being applied in real-world assistive systems, which should adapt continuously to the sensorimotor feedback, a changing environment, and most importantly to the humans' needs.

Appendix A

List of Abbreviations

BMU best-matching unit

BoF Bag of Visual Features

CHL Competitive Hebbian Learning

CRF Conditional Random Fields

EBA extrastriate body area

EM expectation-maximization algorithm

fMRI functional magnetic resonance imaging

F5 ventral premotor cortex

GMM Gaussian mixture model

GMR Gaussian Mixture Regression

GNG Growing Neural Gas

GNG-U GNG with Utility Factor

GPU graphics processing unit

GWR Growing When Required

HMM Hidden Markov Model

HRI Human-Robot Interaction

IFG inferior frontal gyrus

IPL inferior parietal lobule

IT Inferior Temporal Cortex

LDA Linear Discriminant Analysis

LOC Lateral Occipital Complex

LOP Local Occupancy Pattern

LVQ Learning Vector Quantization

MLP Multilayer Perceptron

MTG middle temporal gyrus

NG Neural Gas

PCA Principal Component Analysis

PbD Programming by Demonstration

RBF Radial Basis Function

RNN Recurrent Neural Networks

SEC Semantic Event Chains

SGNG Supervised Growing Neural Gas

SIFT Scale-Invariant Feature Transform

SOM Self-Organizing Map

STIP spatio-temporal interest points

STS superior temporal sulcus

SVM Support Vector Machine

VLAD Vector of Locally Aggregated Descriptors

VQ Vector Quantization

Appendix B

Supplementary Algorithms

Algorithm 2 Growing When Required (GWR) (Marsland et al., 2002)

1. Start with a set A of two random neurons with weights $\{\mathbf{w}_1, \mathbf{w}_2\}$ in the input space.

2. At each iteration t, generate an input sample $\mathbf{x}(t)$

3. Select the best and second-best matching neuron:
$$b = arg\min_{n \in A} ||\mathbf{x}(t) - \mathbf{w}_n||, \quad s = arg\min_{n \in A/\{b\}} ||\mathbf{x}(t) - \mathbf{w}_n|| \tag{B.1}$$

4. Create a connection $E = E \cup \{(b, s)\}$ if it does not exist and set its age to 0.

5. If $(a(t) < a_T)$ and $(h_b < f_T))$ or $(o_b \neq o_{x(t)})$ then:

   - Add a new neuron r $(A = A \cup \{r\})$ with $\mathbf{w}_r = 0.5 \cdot (\mathbf{x}(t) + \mathbf{w}_b), h_r = 1$,
   - Update edges: $E = E \cup \{(r, b), (r, s)\}$ and $E = E/\{(b, s)\}$.

6. If no new neuron is added:

   - Update best-matching neuron and its neighbors i:
$$\Delta\mathbf{w}_b = \epsilon_b \cdot h_b \cdot (\mathbf{x}(t) - \mathbf{w}_b), \quad \Delta\mathbf{w}_i = \epsilon_i \cdot h_i \cdot (\mathbf{x}(t) - \mathbf{w}_i), \tag{B.2}$$
     with the learning rates $0 < \epsilon_i < \epsilon_b < 1$.
   - Increment the age of all edges connected to b by 1.

7. Reduce the firing counters of the best-matching neuron and its neighbors i:
$$\Delta h_b = \tau_b \cdot \kappa \cdot (1 - h_b) - \tau_b, \quad \Delta h_i = \tau_i \cdot \kappa \cdot (1 - h_i) - \tau_i \tag{B.3}$$
   with constant τ and κ controlling the curve behavior.

8. Remove all edges with ages larger than a pre-defined threshold and remove neurons without edges.

9. If the stop criterion is not met, repeat from step 2.

Appendix C

The Skeleton Human Body Model

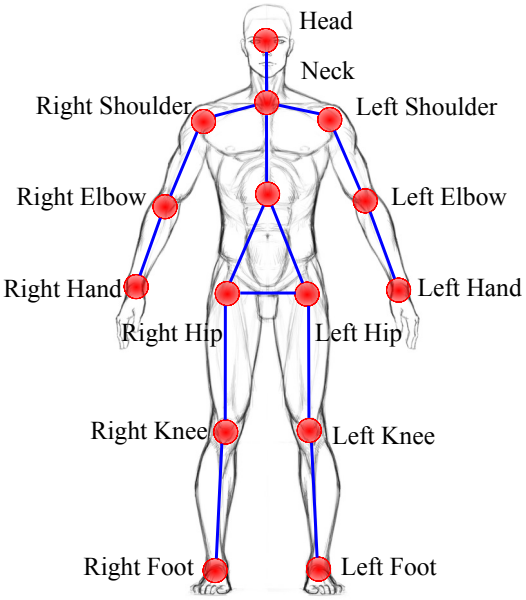


Figure C.1: The skeleton human body model obtained from the Asus Xtion Pro camera. The OpenNI library tracks the three dimensional position of the 15 illustrated joints. Image drawn based on the OpenNI library documentation[1].

[1]NITE: http://pr.cs.cornell.edu/humanactivities/data/NITE.pdf

Appendix D

Additional Results

This appendix shows the results obtained with the object recognition module presented in Section 4.4 on the Washington RGB-D Objects Dataset (Lai et al., 2011). This dataset comprises $\approx$ 42000 images of 300 object instances taken from multiple views and angles with a Kinect 3D camera and organized into 51 categories of common household objects. Only the RGB images of this dataset were used. The dataset is quite challenging because it contains objects without texture and several object categories have high intra-class similarity. The experiments reported here are conducted leaving one random object instance out for testing within each object category and training the classifier on all views of the remaining object instances. The images were encoded with the VLAD encoding technique described in Section 4.3.2.

We obtained an average classification accuracy of $74,5\%$. The highest recognition rate using the RGB images of this dataset is 82.4% (Bo et al., 2013). However, the results are not directly comparable due to the fact that we base our experiments on the first three (out of ten) trials provided for evaluation by the dataset authors.

Figure D.1: Examples of confused classes: a plate classified as a bowl, one mushroom labeled as garlic, an orange classified as peach due to shape similarities, a calculator classified as keyboard due to the common visual word representing the image patch of the keys. Images from Washington RGB-D object dataset (Lai et al., 2011).

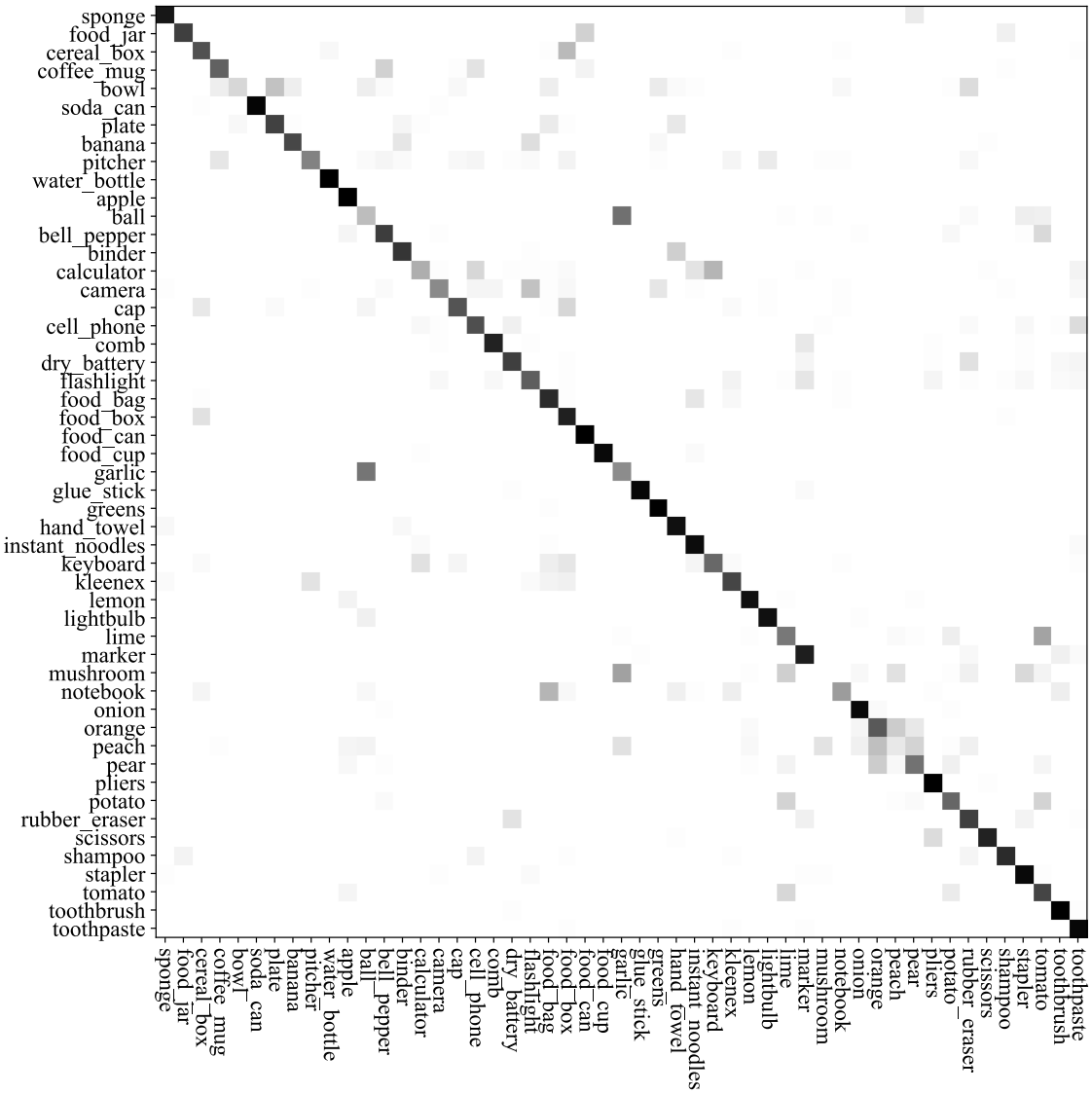


Figure D.2: The normalized confusion matrix for the first three trials of the Washington RGB-D Object Dataset.

Appendix E

Pubblications Originating from this Thesis

Journal Articles

- Mici, L., Parisi, G.I., Wermter, S. (2018) A self-organizing neural network architecture for learning human-object interactions. *Neurocomputing*(307), pages 14–24, doi:10.1016/j.neucom.2018.04.015.

- Mici, L., Parisi, G.I., Wermter, S. (2018) An Incremental Self-Organizing Architecture for Sensorimotor Learning and Prediction. *IEEE Transactions on Cognitive and Developmental Systems* (TCDS), vol. 10, no. 4, pages 918–928, doi:10.1109/TCDS.2018.2832844.

Conference Papers

- Mici, L. Parisi, G.I., Wermter, S. (2018) Recognition and Prediction of Human-Object Interactions with a Self-Organizing Architecture. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), pages 1197–1204.

- Mici, L., Parisi, G.I., Wermter, S. (2016) Recognition of Transitive Actions with Hierarchical Neural Network Learning. In Proceedings of the 25th International Conference on Artificial Neural Networks (ICANN), pages 472–479.

- Mici, L., Hinaut, X., Wermter, S. (2016) Activity Recognition with Echo State Networks using 3D Body Joints and Objects Category. In Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pages 465–470.

Appendix F

Acknowledgements

First, I would like to express my sincere gratitude to my advisor Prof. Stefan Wermter for his continuous support during my doctoral studies. His guidance and invaluable advice helped me in all the time of research and writing of this thesis. I also would like to thank Prof. Loo Chu Kiong for the interesting discussions and feedback on my work and the rest of my examination committee: Prof. Frank Steinicke and Prof. Wolfgang Menzel for their insightful comments and interesting questions. Special thanks to Dr. Cornelius Weber, Dr. Sven Magg, Katja Kösters and Erik Strahl for helpful analytical, technical, and administrative support respectively.

In addition, I would like to thank all my colleagues of the Knowledge Technology research group, especially the ones I had the pleasure to collaborate with: German I. Parisi, Johannes Twiefel, and Xavier Hinaut.

I am profoundly grateful to my family: my beloved father Robert whom I miss every day, my mother Çezarina and my brother Redi for the unconditional love and support. Thank you for always believing in me throughout writing this thesis and my life in general. A wholeheartedly thank you goes to my husband Luigi who has always been by my side since the very beginning.

Finally, I gratefully acknowledge the support of the University of Hamburg, the EU- and City of Hamburg-funded program Pro-Exzellenzia 4.0, the Transregio TRR169 on Crossmodal Learning, and the Hamburg Landesforschungsförderung.

Bibliography

Abeles, M. *Local cortical circuits: An electrophysiological study*, volume 6. Springer Science & Business Media, 1982.

Aggarwal, C. C., Hinneburg, A., and Keim, D. A. On the surprising behavior of distance metrics in high dimensional space. In *International Conference on Database Theory (ICDT)*, pages 420–434. Springer, 2001.

Aggarwal, J. K. and Ryoo, M. S. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3), 2011.

Aggarwal, J. K. and Xia, L. Human activity recognition from 3D data: A review. *Pattern Recognition Letters*, 48:70–80, 2014.

Aksoy, E. E., Abramov, A., Dörr, J., Ning, K., Dellen, B., and Wörgötter, F. Learning the semantics of object–action relations by observation. *The International Journal of Robotics Research*, pages 1229–1249, 2011.

Aksoy, E. E., Orhan, A., and Wörgötter, F. Semantic decomposition and recognition of long and complex manipulation action sequences. *International Journal of Computer Vision*, 122(1):84–115, 2017.

Aldoma, A., Marton, Z. C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R. B., Gedikli, S., and Vincze, M. Tutorial: Point cloud library: Three-dimensional object recognition and 6 DOF pose estimation. *IEEE Robotics Automation Magazine*, 19(3):80–91, Sept 2012.

Amirabdollahian, F., op den Akker, R., Bedaf, S., Bormann, R., Draper, H., Evers, V., Pérez, J. G., Gelderblom, G. J., Ruiz, C. G., Hewson, D., et al. Assistive technology design and development for acceptable robotics companions for ageing years. *Paladyn, Journal of Behavioral Robotics*, 4(2):94–112, 2013.

Andreakis, A., Hoyningen-Huene, N. v., and Beetz, M. Incremental unsupervised time series analysis using merge growing neural gas. In *International Workshop*

on Self-Organizing Maps, pages 10–18. Springer, 2009.

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J. Netvlad: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5307, 2016.

Araujo, A. F. and Barreto, G. A. Context in temporal sequence processing: A self-organizing approach and its application to robotics. *IEEE Transactions on Neural Networks*, 13(1):45–57, 2002.

Arbib, M. A. *The handbook of brain theory and neural networks*. MIT press, 2003.

Arie, H., Arakaki, T., Sugano, S., and Tani, J. Imitating others by composition of primitive actions: A neuro-dynamic model. *Robotics and Autonomous Systems*, 60(5):729–741, 2012.

Aurenhammer, F. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.

Bahill, A. A simple adaptive smith-predictor for controlling time-delay systems: A tutorial. *IEEE Control systems magazine*, 3(2):16–22, 1983.

Baldassano, C., Beck, D. M., and Fei-Fei, L. Human–object interactions are more than the sum of their parts. *Cerebral Cortex*, 27(3):2276–2288, 2017.

Barreto, G. A. Time series prediction with the self-organizing map: A review. In *Perspectives of neural-symbolic integration*, pages 135–158. Springer, 2007.

Barreto, G. D. A., Araújo, A. F., and Ritter, H. J. Self-organizing feature maps for modeling and control of robotic manipulators. *Journal of Intelligent and Robotic Systems*, 36(4):407–450, 2003.

Bates, E. *The emergence of symbols: Cognition and communication in infancy*. Academic Press, 2014.

Beauchamp, M. S., Lee, K. E., Haxby, J. V., and Martin, A. Parallel visual motion processing streams for manipulable objects and human movements. *Neuron*, 34 (1):149–159, 2002.

Behnke, S., Egorova, A., Gloye, A., Rojas, R., and Simon, M. Predicting away robot control latency. In *Robot Soccer World Cup*, pages 712–719. Springer,

2003.

Ben-Arie, J., Wang, Z., Pandit, P., and Rajaram, S. Human activity recognition using multidimensional indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1091–1104, 2002.

Biederman, I. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.

Biehl, M., Hammer, B., and Villmann, T. Prototype-based models in machine learning. *Wiley Interdisciplinary Reviews: Cognitive Science*, 7(2):92–111, 2016.

Billard, A. G., Calinon, S., and Guenter, F. Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics and Autonomous Systems*, 54(5): 370–384, 2006.

Billard, A. G., Calinon, S., and Dillmann, R. *Learning from Humans*, pages 1995–2014. Springer International Publishing, 2016.

Blakemore, C. and Cooper, G. F. Development of the brain depends on the visual environment. *Nature*, 228(5270):477, 1970.

Bo, L., Ren, X., and Fox, D. Unsupervised feature learning for RGB-D based object recognition. In *Experimental Robotics*, pages 387–402. Springer, 2013.

Boldrini, M., Fulmore, C. A., Tartt, A. N., Simeon, L. R., Pavlova, I., Poposka, V., Rosoklija, G. B., Stankov, A., Arango, V., Dwork, A. J., et al. Human hippocampal neurogenesis persists throughout aging. *Cell Stem Cell*, 22(4): 589–599, 2018.

Brass, M., Schmitt, R. M., Spengler, S., and Gergely, G. Investigating action understanding: inferential processes versus action simulation. *Current Biology*, 17(24):2117–2121, 2007.

Buccino, G., Sato, M., Cattaneo, L., Rodà, F., and Riggio, L. Broken affordances, broken objects: a TMS study. *Neuropsychologia*, 47(14):3074–3078, 2009.

Buonomano, D. V. and Merzenich, M. M. Cortical plasticity: from synapses to maps. *Annual review of neuroscience*, 21(1):149–186, 1998.

Bütepage, J., Black, M., Kragic, D., and Kjellström, H. Deep representation learning for human motion prediction and classification. *ArXiv preprint*

arXiv:1702.07486, February 2017.

Bütepage, J., Kjellström, H., and Kragic, D. Anticipating many futures: Online human motion prediction and synthesis for human-robot collaboration. *arXiv preprint arXiv:1702.08212*, 2017.

Caggiano, V., Fogassi, L., Rizzolatti, G., Casile, A., Giese, M. A., and Thier, P. Mirror neurons encode the subjective value of an observed action. *Proceedings of the National Academy of Sciences*, 109(29):11848–11853, 2012.

Calinon, S. and Billard, A. Incremental learning of gestures by imitation in a humanoid robot. In *Proceedings of the ACM/IEEE International Conference on Human-robot interaction (HRI)*, pages 255–262, 2007.

Cao, Y., Barrett, D., Barbu, A., Narayanaswamy, S., Yu, H., Michaux, A., Lin, Y., Dickinson, S., Mark Siskind, J., and Wang, S. Recognize human activities from partially observed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2658–2665, 2013.

Carr, L., Iacoboni, M., Dubeau, M.-C., Mazziotta, J. C., and Lenzi, G. L. Neural mechanisms of empathy in humans: a relay from neural systems for imitation to limbic areas. *Proceedings of the national Academy of Sciences*, 100(9):5497–5502, 2003.

Cavallo, A., Koul, A., Ansuini, C., Capozzi, F., and Becchio, C. Decoding intentions from movement kinematics. *Scientific Reports*, 6:37036, 2016.

Cecchi, F., Sgandurra, G., Mihelj, M., Mici, L., Zhang, J., Munih, M., Cioni, G., Laschi, C., and Dario, P. CareToy: An intelligent baby gym: Home-based intervention for infants at risk for neurodevelopmental disorders. *IEEE Robotics & Automation Magazine*, 23(4):63–72, 2016.

Cederborg, T., Li, M., Baranes, A., and Oudeyer, P.-Y. Incremental local online gaussian mixture regression for imitation learning of multiple tasks. In *Proceedings of IEEE/RSJ International Conference On Intelligent Robots and Systems (IROS)*, pages 267–274, 2010.

Chappell, G. J. and Taylor, J. G. The temporal kohønen map. *Neural networks*, 6(3):441–445, 1993.

Chersi, F., Ferrari, P. F., and Fogassi, L. Neuronal chains for actions in the parietal

lobe: A computational model. *PloS one*, 6(11):e27652, 2011.

Chersi, F., Ferro, M., Pezzulo, G., and Pirrelli, V. Topological self-organization and prediction learning support both action and lexical chains in the brain. *Topics in Cognitive Science*, 6(3):476–491, 2014.

Cippitelli, E., Gasparrini, S., Gambi, E., and Spinsante, S. A human activity recognition system using skeleton data from RGBD sensors. *Computational Intelligence and Neuroscience*, 2016.

Coleca, F., State, A., Klement, S., Barth, E., and Martinetz, T. Self-organizing maps for hand and full body tracking. *Neurocomputing*, 147:174–184, 2015.

de la Malla, C., López-Moliner, J., and Brenner, E. Dealing with delays does not transfer across sensorimotor tasks. *Journal of Vision*, 14(12):8, 2014.

Demiris, J. and Hayes, G. Imitation as a dual-route process featuring predictive and learning components; 4 biologically plausible computational model. *Imitation in animals and artifacts*, 327, 2002.

DiCarlo, J. J., Zoccolan, D., and Rust, N. C. How does the brain solve visual object recognition? *Neuron*, 73(3):415–434, 2012.

Ding, W., Liu, K., Cheng, F., and Zhang, J. Learning hierarchical spatio-temporal pattern for human activity prediction. *Journal of Visual Communication and Image Representation*, 35:103–111, 2016.

Dirichlet, G. L. Über die reduction der positiven quadratischen formen mit drei unbestimmten ganzen zahlen. *Journal für die reine und angewandte Mathematik*, 40:209–227, 1850.

Divvala, S. K., Hoiem, D., Hays, J. H., Efros, A. A., and Hebert, M. An empirical study of context in object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1271–1278, 2009.

Dixon, K. R., Dolan, J. M., and Khosla, P. K. Predictive robot programming: Theoretical and experimental analysis. *The International Journal of Robotics Research*, 23(9):955–973, 2004.

Donatti, G. S., Lomp, O., and Würtz, R. P. Evolutionary optimization of growing neural gas parameters for object categorization and recognition. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 1–8,

2010.

Downing, P. E. and Peelen, M. V. The role of occipitotemporal body-selective regions in person perception. *Cognitive Neuroscience*, 2(3-4):186–203, 2011.

Du, Y., Wang, W., and Wang, L. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1118, 2015.

Ekvall, S., Aarno, D., and Kragic, D. Online task recognition and real-time adaptive assistance for computer-aided machine control. *IEEE Transactions on Robotics*, 22(5):1029–1033, 2006.

Estevez, P., Hernández, R., Perez, C. A., and Held, C. Gamma-filter self-organising neural networks for unsupervised sequence processing. *Electronics Letters*, 47 (8):494–496, 2011.

Estévez, P. A. and Hernández, R. Gamma SOM for temporal sequence processing. In *International Workshop on Self-Organizing Maps*, pages 63–71. Springer, 2009.

Estévez, P. A. and Vergara, J. R. Nonlinear time series analysis by using gamma growing neural gas. In *Advances in Self-Organizing Maps*, pages 205–214. Springer, 2013.

Evangelidis, G., Singh, G., and Horaud, R. Skeletal quads: Human action recognition using joint quadruples. In *International Conference on Pattern Recognition (ICPR)*, pages 4513–4518, 2014.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. Describing objects by their attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1778–1785, 2009.

Felleman, D. J. and Essen, D. C. V. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1):1–47, 1991.

Fisicaro, R. A., Jost, E., Shaw, K., Brennan, N. P., Peck, K. K., and Holodny, A. I. Cortical plasticity in the setting of brain tumors. *Topics in magnetic resonance*

imaging: TMRI, 25(1):25, 2016.

Fitzpatrick, P. and Metta, G. Grounding vision through experimental manipulation. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 361(1811):2165–2185, 2003.

Fleischer, F., Caggiano, V., Thier, P., and Giese, M. A. Physiologically inspired model for the visual recognition of transitive hand actions. *The Journal of Neuroscience*, 33(15):6563–6580, 2013.

Floreano, D. and Mattiussi, C. *Bio-inspired artificial intelligence: theories, methods, and technologies*. MIT press, 2008.

Fogassi, L., Ferrari, P. F., Gesierich, B., Rozzi, S., Chersi, F., and Rizzolatti, G. Parietal lobe: from action organization to intention understanding. *Science*, 308 (5722):662–667, 2005.

Friedman, J. H. and Stuetzle, W. Projection pursuit regression. *Journal of the American statistical Association*, 76(376):817–823, 1981.

Fritzke, B. A growing neural gas network learns topologies. *Advances in Neural Information Processing Systems*, 7:625–632, 1995.

Fritzke, B. Growing self-organizing networks - why? In *European Symposium on Artificial Neural Networks (ESANN)*, volume 96, pages 61–72, 1996.

Fritzke, B. A self-organizing network that can follow non-stationary distributions. In *Proceedings of International Conference on Artificial Neural Networks (ICANN)*, pages 613–618. Springer, 1997.

Fujiyoshi, H., Lipton, A. J., and Kanade, T. Real-time human motion analysis by image skeletonization. *IEICE Transactions on Information and Systems*, 87(1): 113–120, 2004.

Fukunaga, K. *Introduction to statistical pattern recognition*. Academic press, 2013.

Fukushima, K. Neocognitron: A self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.

Gallese, V., Fadiga, L., Fogassi, L., and Rizzolatti, G. Action recognition in the premotor cortex. *Brain*, 119(2):593–609, 1996.

Ghosh, P., Song, J., Aksan, E., and Hilliges, O. Learning human motion models for long-term predictions. In *Proceedings of the IEEE International Conference on 3D Vision (3DV)*, pages 458–466, 2017.

Gibson, J. J. *The ecological approach to visual perception: classic edition.* Psychology Press, 2014.

Giese, M. A. and Poggio, T. Neural mechanisms for the recognition of biological movements. *Nature Reviews Neuroscience*, 4(3):179–192, 2003.

Giese, M. A. and Rizzolatti, G. Neural and computational mechanisms of action processing: interaction between visual and motor representations. *Neuron*, 88 (1):167–180, 2015.

Gilbert, C. D. and Sigman, M. Brain states: Top-down influences in sensory processing. *Neuron*, 54(5):677–696, 2007.

Golarai, G., Liberman, A., and Grill-Spector, K. Experience shapes the development of neural substrates of face processing in human ventral temporal cortex. *Cerebral Cortex*, 27(2):1229–1244, 2017.

Goodale, M. A. Action without perception in human vision. *Cognitive Neuropsychology*, 25(7-8):891–919, 2008.

Grabner, H., Gall, J., and Van Gool, L. What makes a chair a chair? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1529–1536. IEEE, 2011.

Grill-Spector, K. Representation of objects. *The Oxford Handbook of Cognitive Neuroscience, Volume 2: The Cutting Edges*, 2, 2013.

Grossberg, S. On the development of feature detectors in the visual cortex with applications to learning and reaction-diffusion systems. *Biological Cybernetics*, 21(3):145–159, 1976.

Grossman, E. D. and Blake, R. Brain areas active during visual perception of biological motion. *Neuron*, 35(6):1167–1175, 2002.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., and Lew, M. S. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27–48, 2016.

Gupta, A., Kembhavi, A., and Davis, L. S. Observing human-object interactions:

Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, Oct 2009.

Hamilton, A. F. d. C. and Grafton, S. T. Goal representation in human anterior intraparietal sulcus. *Journal of Neuroscience*, 26(4):1133–1137, 2006.

Han, J., Shao, L., Xu, D., and Shotton, J. Enhanced computer vision with Microsoft Kinect sensor: A review. *IEEE Transactions on Cybernetics*, 43(5): 1318–1334, Oct 2013.

Hankins, R., Peng, Y., and Yin, H. SOMNet: Unsupervised feature learning networks for image classification. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2018)*, pages 1221–1228, Jul 2018.

Hasson, U., Yang, E., Vallines, I., Heeger, D. J., and Rubin, N. A hierarchy of temporal receptive windows in human cortex. *Journal of Neuroscience*, 28(10): 2539–2550, 2008.

Hebb, D. The organisation of behavior - a neuropsychological theory. *Wiley, New York*, 1949.

Heitz, G. and Koller, D. Learning spatial context: Using stuff to find things. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 30–43. Springer, 2008.

Henderson, J. M. Human gaze control during real-world scene perception. *Trends in cognitive sciences*, 7(11):498–504, 2003.

Hirsch, H. V. and Spinelli, D. Visual experience modifies distribution of horizontally and vertically oriented receptive fields in cats. *Science*, 168(3933):869–871, 1970.

Holmström, J. and Gas, G. N. Growing neural gas, experiments with GNG, GNG with utility and supervised GNG. *Master's Thesis in the Department of Information Technology, Uppsala University*, 2002.

Hu, N., Englebienne, G., Lou, Z., and Kröse, B. Learning latent structure for activity recognition. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1048–1053, 2014.

Hu, W., Xie, D., Tan, T., and Maybank, S. Learning activity patterns using fuzzy self-organizing neural network. *IEEE Transactions on Systems, Man, and*

Cybernetics, Part B (Cybernetics), 34(3):1618–1626, 2004.

Hubel, D. H. and Wiesel, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1): 106–154, 1962.

Huth, A. G., Nishimoto, S., Vu, A. T., and Gallant, J. L. A continuous semantic space describes the representation of thousands of object and action categories across the human brain. *Neuron*, 76(6):1210–1224, 2012.

Iacoboni, M., Molnar-Szakacs, I., Gallese, V., Buccino, G., Mazziotta, J. C., and Rizzolatti, G. Grasping the intentions of others with one's own mirror neuron system. *PLoS biology*, 3(3):e79, 2005.

Ito, M. and Tani, J. On-line imitative interaction with a humanoid robot using a mirror neuron model. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1071–1076, 2004.

Jamone, L., Ugur, E., Cangelosi, A., Fadiga, L., Bernardino, A., Piater, J., and Santos-Victor, J. Affordances in psychology, neuroscience and robotics: a survey. *IEEE Transactions on Cognitive and Developmental Systems*, 2016.

Jegou, H., Perronnin, F., Douze, M., Sánchez, J., Perez, P., and Schmid, C. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012.

Johansson, G. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211, 1973.

Johnson, J., Hariharan, B., van der Maaten, L., Hoffman, J., Fei-Fei, L., Zitnick, C. L., and Girshick, R. B. Inferring and executing programs for visual reasoning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3008–3017, 2017.

Johnson, N. and Hogg, D. Learning the distribution of object trajectories for event recognition. *Image and Vision computing*, 14(8):609–615, 1996.

Johnson, S. H. and Grafton, S. T. From "acting on" to "acting with": the functional anatomy of object-oriented action schemata. In *Progress in brain research*, volume 142, pages 127–139. Elsevier, 2003.

Johnson-Frey, S. H. What's so special about human tool use? *Neuron*, 39(2):

201–204, 2003.

Kachouie, R., Sedighadeli, S., Khosla, R., and Chu, M.-T. Socially assistive robots in elderly care: A mixed-method systematic literature review. *International Journal of Human-Computer Interaction*, 30(5):369–393, 2014.

Karg, M., Samadani, A.-A., Gorbet, R., Kühnlenz, K., Hoey, J., and Kulić, D. Body movements for affective expression: A survey of automatic recognition and generation. *IEEE Transactions on Affective Computing*, 4(4):341–359, 2013.

Kawashima, M., Shimada, A., and Taniguchi, R. Early recognition of gesture patterns using sparse code of self-organizing map. In *International Workshop on Self-Organizing Maps*, pages 116–123. Springer, 2009.

Kerzel, D. and Gegenfurtner, K. R. Neuronal processing delays are compensated in the sensorimotor branch of the visual system. *Current Biology*, 13(22):1975–1978, 2003.

Keysers, C. and Perrett, D. I. Demystifying social cognition: a hebbian perspective. *Trends in cognitive sciences*, 8(11):501–507, 2004.

Khansari-Zadeh, S. M. and Billard, A. Bm: An iterative algorithm to learn stable non-linear dynamical systems with gaussian mixture models. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2381–2388, 2010.

Kilner, J. M. and Lemon, R. What we know currently about mirror neurons. *Current Biology*, 23(23):R1057–R1062, 2013.

Kinnunen, T., Kamarainen, J.-K., Lensu, L., and Kälviäinen, H. Unsupervised object discovery via self-organisation. *Pattern Recognition Letters*, 33(16):2102–2112, 2012.

Kjellström, H., Romero, J., and Kragić, D. Visual object-action recognition: Inferring object affordances from human demonstration. *Computer Vision and Image Understanding*, 115(1):81–90, 2011.

Kohonen, T. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982.

Kohonen, T. The neural phonetic typewriter. *Computer*, 21(3):11–22, 1988.

Kohonen, T. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

Kohonen, T. Essentials of the self-organizing map. *Neural Networks*, 37:52–65, 2013.

Koppula, H. S. and Saxena, A. Anticipating human activities using object affordances for reactive robotic response. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):14–29, 2016.

Koppula, H. S. and Saxena, A. Learning spatio-temporal structure from RGB-D videos for human activity detection and anticipation. In *International Conference on Machine Learning (ICML)*, pages 792–800, 2013.

Koppula, H. S., Gupta, R., and Saxena, A. Learning human activities and object affordances from RGB-D videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.

Kulić, D., Takano, W., and Nakamura, Y. Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden Markov chains. *The International Journal of Robotics Research*, 27(7):761–784, 2008.

Kulić, D., Ott, C., Lee, D., Ishikawa, J., and Nakamura, Y. Incremental learning of full body motion primitives and their sequencing through human motion observation. *The International Journal of Robotics Research*, 31(3):330–345, 2012.

Lai, K., Bo, L., Ren, X., and Fox, D. A large-scale hierarchical multi-view RGB-D object dataset. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1817–1824, 2011.

Lampert, C. H., Nickisch, H., and Harmeling, S. Learning to detect unseen object classes by between-class attribute transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 951–958, 2009.

Lan, T., Chen, T.-C., and Savarese, S. A hierarchical representation for future action prediction. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 689–704. Springer, 2014.

Lan, T., Zhu, Y., Roshan Zamir, A., and Savarese, S. Action recognition by hierarchical mid-level action elements. In *Proceedings of the IEEE International*

Conference on Computer Vision (ICCV), pages 4552–4560, 2015.

Lea, C., Reiter, A., Vidal, R., and Hager, G. D. Segmental spatiotemporal cnns for fine-grained action segmentation. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 36–52. Springer, 2016.

Levine, S., Pastor, P., Krizhevsky, A., and Quillen, D. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *arXiv preprint arXiv:1603.02199*, 2016.

Li, W., Zhang, Z., and Liu, Z. Action recognition based on a bag of 3D points. In *IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 9–14, 2010.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollar, P. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

Livingston, M. A., Sebastian, J., Ai, Z., and Decker, J. W. Performance measurements for the Microsoft Kinect skeleton. In *Proceedings of IEEE Virtual Reality Workshops (VRW)*, pages 119–120, 2012.

Lorenz, T., Mörtl, A., Vlaskamp, B., Schubö, A., and Hirche, S. Synchronization in a goal-directed task: Human movement coordination with each other and robotic partners. In *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 198–203, 2011.

Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

Luciw, M. and Weng, J. Top–down connections in self-organizing hebbian networks: Topographic class grouping. *IEEE Transactions on Autonomous Mental Development*, 2(3):248–261, 2010.

Ma, C.-Y., Kadav, A., Melvin, I., Kira, Z., AlRegib, G., and Graf, H. P. Attend and interact: Higher-order object interactions for video understanding. *ArXiv preprint arXiv:1711.06330*, November 2017.

Mainprice, J. and Berenson, D. Human-robot collaborative manipulation planning using early prediction of human motion. In *Proceedings of IEEE/RSJ International Conference On Intelligent Robots and Systems (IROS)*, pages 299–306,

2013.

Mainprice, J., Gharbi, M., Siméon, T., and Alami, R. Sharing effort in planning human-robot handover tasks. In *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 764–770, 2012.

Marr, D., Lal, S., and Barlow, H. Visual information processing: The structure and creation of visual representations. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 290(1038):199–218, 1980.

Marsland, S., Shapiro, J., and Nehmzow, U. A self-organising network that grows when required. *Neural Networks*, 15(8):1041–1058, 2002.

Martinetz, T. Competitive hebbian learning rule forms perfectly topology preserving maps. In *International Conference on Artificial Nerual networks (ICANN)*, pages 427–434. Springer, 1993.

Martinetz, T. and Schulten, K. A "neural-gas" network learns topologies. In *Artificial Neural Networks*, pages 397–402. Elsevier Science Publisher B.V., 1991.

Merzenich, M. M., Kaas, J., Wall, J., Nelson, R., Sur, M., and Felleman, D. Topographic reorganization of somatosensory cortical areas 3b and 1 in adult monkeys following restricted deafferentation. *Neuroscience*, 8(1):33–55, 1983.

Miall, R., Weir, D. J., Wolpert, D. M., and Stein, J. Is the cerebellum a smith predictor? *Journal of motor behavior*, 25(3):203–216, 1993.

Mici, L., Parisi, G. I., and Wermter, S. Recognition of transitive actions with hierarchical neural network learning. In *Artificial Neural Networks and Machine Learning (ICANN)*, pages 472–479. Springer International Publishing, 2016.

Mici, L., Parisi, G. I., and Wermter, S. A self-organizing neural network architecture for learning human-object interactions. *Neurocomputing*, 307:14–24, 2018a.

Mici, L., Parisi, G. I., and Wermter, S. Recognition and prediction of human-object interactions with a self-organizing architecture. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2018)*, pages 1197–1204, Jul 2018b.

Mici, L., Parisi, G. I., and Wermter, S. An incremental self-organizing architecture for sensorimotor learning and prediction. *IEEE Transactions on Cognitive and*

Developmental Systems, Apr 2018c.

Miikkulainen, R., Bednar, J. A., Choe, Y., and Sirosh, J. *Computational maps in the visual cortex*. Springer Science & Business Media, 2006.

Mineiro, P. and Zipser, D. Analysis of direction selectivity arising from recurrent cortical interactions. *Neural Computation*, 10(2):353–371, 1998.

Mountcastle, V. B. Modality and topographic properties of single neurons of cat's somatic sensory cortex. *Journal of neurophysiology*, 20(4):408–434, 1957.

Murata, A., Fadiga, L., Fogassi, L., Gallese, V., Raos, V., and Rizzolatti, G. Object representation in the ventral premotor cortex (area f5) of the monkey. *Journal of neurophysiology*, 78(4):2226–2230, 1997.

Murphy, K. P., Torralba, A., and Freeman, W. T. Using the forest to see the trees: A graphical model relating features, objects, and scenes. In *Advances in neural information processing systems*, pages 1499–1506, 2004.

Nelissen, K., Luppino, G., Vanduffel, W., Rizzolatti, G., and Orban, G. A. Observing others: Multiple action representation in the frontal lobe. *Science*, 310 (5746):332–336, 2005.

Nijhawan, R. and Wu, S. Compensating time delays with neural predictions: are predictions sensory or motor? *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1891):1063–1078, 2009.

Nishimoto, R., Namikawa, J., and Tani, J. Learning multiple goal-directed actions through self-organization of a dynamic neural network model: A humanoid robot experiment. *Adaptive Behavior*, 16(2-3):166–181, 2008.

Ogata, T., Sugano, S., and Tani, J. Open-end human robot interaction from the dynamical systems perspective: Mutual adaptation and incremental learning. In *Proceedings of IEA-AIE*, pages 435–444. Springer, 2004.

Okada, M., Nakamura, D., and Nakamura, Y. Self-organizing symbol acquisition and motion generation based on dynamics-based information processing system. In *Proceedings of the 2nd International Workshop on Man-machine Symbiotic Systems*, pages 219–229, 2004.

Oram, M. and Perrett, D. Integration of form and motion in the anterior su-

perior temporal polysensory area (STPa) of the macaque monkey. *Journal of neurophysiology*, 76(1):109–129, 1996.

Parisi, G. I., Weber, C., and Wermter, S. Self-organizing neural integration of pose-motion features for human action recognition. *Frontiers in Neurorobotics*, 9, 2015.

Parisi, G. I., Magg, S., and Wermter, S. Human motion assessment in real time using recurrent self-organization. In *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 71–76, Aug 2016a.

Parisi, G. I., Tani, J., Weber, C., and Wermter, S. Emergence of multimodal action representations from neural network self-organization. *Cognitive Systems Research*, 2016b.

Parisi, G. I., Tani, J., Weber, C., and Wermter, S. Lifelong learning of human actions with deep neural network self-organization. *Neural Networks*, 96:137–149, Dec 2017a.

Parisi, G. I., Tani, J., Weber, C., and Wermter, S. Lifelong learning of human actions with deep neural network self-organization. *Neural Networks*, 96:137–149, 2017b.

Parisi, G. I., Weber, C., and Wermter, S. Human action recognition with hierarchical growing neural gas learning. In *International Conference on Artificial Neural Networks (ICANN)*, pages 89–96. Springer, 2014.

Perrett, D. View-dependent coding in the ventral stream and its consequences for recognition. *Vision and Movement Mechanisms in the Cerebral Cortex*, pages 142–51, 1996.

Perrett, D., Smith, P., Mistlin, A., Chitty, A., Head, A., Potter, D., Broennimann, R., Milner, A., and Jeeves, M. A. Visual analysis of body movements by neurones in the temporal cortex of the macaque monkey: a preliminary report. *Behavioural brain research*, 16(2-3):153–170, 1985.

Pieropan, A., Ek, C. H., and Kjellström, H. Recognizing object affordances in terms of spatio-temporal object-object relationships. In *Proceedings of IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 52–58, 2014a.

Pieropan, A., Salvi, G., Pauwels, K., and Kjellström, H. Audio-visual classification and detection of human manipulation actions. In *Proceedings of IEEE/RSJ International Conference On Intelligent Robots and Systems (IROS)*, pages 3045–3052, 2014b.

Poggio, T. and Edelman, S. A network that learns to recognize three-dimensional objects. *Nature*, 343(6255):263–266, 1990.

Poppe, R. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

Prevete, R., Tessitore, G., Santoro, M., and Catanzariti, E. A connectionist architecture for view-independent grip-aperture computation. *Brain Research*, 1225: 133–145, 2008.

Reale, R. A. and Imig, T. J. Tonotopic organization in auditory cortex of the cat. *Journal of Comparative Neurology*, 192(2):265–291, 1980.

Reiser, U., Jacobs, T., Arbeiter, G., Parlitz, C., and Dautenhahn, K. *Care-O-bot 3 – Vision of a Robot Butler*, pages 97–116. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

Ridge, B., Skočaj, D., and Leonardis, A. Self-supervised cross-modal online learning of basic object affordances for developmental robotic systems. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 5047–5054, 2010.

Riesenhuber, M. and Poggio, T. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999.

Rizzolatti, G. and Arbib, M. A. Language within our grasp. *Trends in neurosciences*, 21(5):188–194, 1998.

Rizzolatti, G. and Craighero, L. The mirror-neuron system. *Annual Review of Neuroscience*, 27:169–192, 2004.

Rizzolatti, G. and Fadiga, L. Grasping objects and grasping action meanings: The dual role of monkey rostroventral premotor cortex (area F5). *Sensory guidance of movement*, 218:81–103, 1998.

Rizzolatti, G. and Fogassi, L. The mirror mechanism: Recent findings and perspectives. *Philosophical Transactions of the Royal Society B*, 369(1644), 2014.

Rizzolatti, G., Fogassi, L., and Gallese, V. Neurophysiological mechanisms underlying the understanding and imitation of action. *Nature reviews neuroscience*, 2 (9):661, 2001.

Robbins, H. and Monro, S. A stochastic approximation method. In *Herbert Robbins Selected Papers*, pages 102–109. Springer, 1985.

Rodriguez, I., Astigarraga, A., Jauregi, E., Ruiz, T., and Lazkano, E. Humanizing NAO robot teleoperation using ROS. In *Proceedings of IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 179–186. IEEE, 2014.

Rohde, M., van Dam, L. C., and Ernst, M. O. Predictability is necessary for closed-loop visual feedback delay adaptation. *Journal of vision*, 14(3):4–4, 2014.

Rosch, E. and Mervis, C. B. Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 7(4):573–605, 1975.

Rosenbaum, D. A. *Human motor control*. Academic press, 2009.

Rumelhart, D. E. and Zipser, D. Feature discovery by competitive learning. *Cognitive science*, 9(1):75–112, 1985.

Rusu, R. B., Blodow, N., Marton, Z. C., and Beetz, M. Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments. In *Proceedings of IEEE/RSJ International Conference On Intelligent Robots and Systems (IROS)*, pages 1–6, 2009.

Rybok, L., Schauerte, B., Al-Halah, Z., and Stiefelhagen, R. "Important stuff, everywhere!" Activity recognition with salient proto-objects as context. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 646–651. IEEE, 2014.

Ryoo, M. S. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1036–1043, 2011.

Ryoo, M. S. and Aggarwal, J. Hierarchical recognition of human activities interacting with objects. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007.

Saegusa, R., Nori, F., Sandini, G., Metta, G., and Sakka, S. Sensory prediction for autonomous robots. In *Proceedings of IEEE-RAS International Conference*

on Humanoid Robots (Humanoids), pages 102–108, 2007.

Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017.

Sargolzaei, A., Abdelghani, M., Yen, K. K., and Sargolzaei, S. Sensorimotor control: computing the immediate future from the delayed present. *BMC Bioinformatics*, 17(7), 2016.

Saxe, R., Carey, S., and Kanwisher, N. Understanding other minds: linking developmental psychology and functional neuroimaging. *Annual Review of Psychology*, 55:87–124, 2004.

Scassellati, B., Admoni, H., and Matarić, M. Robots for use in autism research. *Annual review of biomedical engineering*, 14:275–294, 2012.

Schaefer, A. M., Udluft, S., and Zimmermann, H.-G. Learning long-term dependencies with recurrent neural networks. *Neurocomputing*, 71(13):2481–2488, 2008.

Schoeler, M. and Wörgötter, F. Bootstrapping the semantics of tools: Affordance analysis of real world objects on a per-part basis. *IEEE Transactions on Cognitive and Developmental Systems*, 8(2):84–98, 2016.

Sciutti, A., Mara, M., Tagliasco, V., and Sandini, G. Humanizing human-robot interaction: On the importance of mutual understanding. *IEEE Technology and Society Magazine*, 37(1):22–29, 2018.

Serre, T., Oliva, A., and Poggio, T. A feedforward architecture accounts for rapid categorization. *Proceedings of the National Academy of Sciences*, 104(15):6424–6429, 2007.

Shahroudy, A., Liu, J., Ng, T.-T., and Wang, G. Ntu RGB+D: A large scale dataset for 3D human activity analysis. *arXiv preprint arXiv:1604.02808*, 2016.

Shatz, C. J. The developing brain. *Scientific American*, 267(3):60–67, 1992.

Shimozaki, M. and Kuniyoshi, Y. Integration of spatial and temporal contexts for action recognition by self organizing neural networks. In *Proceedings of IEEE/RSJ International Conference On Intelligent Robots and Systems (IROS)*, volume 3, pages 2385–2391, 2003.

Shockley, K., Richardson, D. C., and Dale, R. Conversation and coordinative structures. *Topics in Cognitive Science*, 1(2):305–319, 2009.

Shotton, J., Winn, J., Rother, C., and Criminisi, A. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 1–15. Springer, 2006.

Simon, G., Lee, J. A., and Verleysen, M. Unfolding preprocessing for meaningful time series clustering. *Neural Networks*, 19(6-7):877–888, 2006.

Simon, G., Lee, J. A., Cottrell, M., and Verleysen, M. Forecasting the CATS benchmark with the double vector quantization method. *Neurocomputing*, 70 (13):2400–2409, 2007.

Singer, J. M. and Sheinberg, D. L. Temporal cortex neurons encode articulated actions as slow sequences of integrated poses. *The Journal of Neuroscience*, 30 (8):3133–3145, 2010.

Sorrells, S. F., Paredes, M. F., Cebrian-Silla, A., Sandoval, K., Qi, D., Kelley, K. W., James, D., Mayer, S., Chang, J., Auguste, K. I., et al. Human hippocampal neurogenesis drops sharply in children to undetectable levels in adults. *Nature*, 555(7696):377, 2018.

Stapel, J. C., Hunnius, S., and Bekkering, H. Fifteen-month-old infants use velocity information to predict others action targets. *Frontiers in Psychology*, 6:1092, 2015.

Stork, J. A., Spinello, L., Silva, J., and Arras, K. O. Audio-based human activity recognition using non-markovian ensemble voting. In *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 509–514, 2012.

Stoytchev, A. Behavior-grounded representation of tool affordances. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3060–3065. IEEE, 2005.

Strickert, M. and Hammer, B. Neural gas for sequences. In *Proceedings of the Workshop on Self-Organizing Maps (WSOM03)*, pages 53–57, 2003.

Strickert, M. and Hammer, B. Merge SOM for temporal data. *Neurocomputing*,

64:39–71, 2005.

Sumpter, N. and Bulpitt, A. Learning spatio-temporal patterns for predicting object behaviour. *Image and Vision Computing*, 18(9):697–704, 2000.

Sung, J., Ponce, C., Selman, B., and Saxena, A. Unstructured human activity detection from RGBD images. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 842–849. IEEE, 2012.

Szeliski, R. *Computer vision: Algorithms and applications*. Springer Science & Business Media, 2010.

Taha, A., Zayed, H. H., Khalifa, M., and El-Horbaty, E.-S. M. Skeleton-based human activity recognition for video surveillance. *International Journal of Scientific & Engineering Research*, 6(1):993–1004, 2015.

Takano, W. and Nakamura, Y. Humanoid robot's autonomous acquisition of proto-symbols through motion segmentation. In *Proceedings of IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 425–431, 2006.

Takens, F. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981.

Taniguchi, T., Nagai, T., Nakamura, T., Iwahashi, N., Ogata, T., and Asoh, H. Symbol emergence in robotics: a survey. *Advanced Robotics*, 30(11-12):706–728, 2016.

Taylor, P., Hobbs, J., Burroni, J., and Siegelmann, H. The global landscape of cognition: Hierarchical aggregation as an organizational principle of human cortical networks and functions. *Scientific reports*, 5:18112, 2015.

Tayyub, A., JawadTavanai, Gatsoulis, Y., Cohn, A. G., and Hogg, D. C. Qualitative and quantitative spatio-temporal relations in daily living activity recognition. In *Computer Vision – ACCV 2014*, pages 115–130, Cham, 2015. Springer International Publishing.

Teo, C. L., Yang, Y., Daumé, H., Fermüller, C., and Aloimonos, Y. Towards a watson that sees: Language-guided action recognition for robots. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 374–381. IEEE, 2012.

Tessitore, G., Prevete, R., Catanzariti, E., and Tamburrini, G. From motor to

sensory processing in mirror neuron computational modelling. *Biological Cybernetics*, 103(6):471–485, 2010.

Tierney, A. L. and Nelson III, C. A. Brain development and the role of experience in the early years. *Zero to three*, 30(2):9, 2009.

Torralba, A. and Efros, A. A. Unbiased look at dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1521–1528, 2011.

Torta, E., Oberzaucher, J., Werner, F., Cuijpers, R. H., and Juola, J. F. Attitudes towards socially assistive robots in intelligent homes: Results from laboratory studies and field trials. *Journal of Human-Robot Interaction*, 1(2):76–99, 2012.

Trong, N. P., Nguyen, H., Kazunori, K., and Le Hoai, B. A comprehensive survey on human activity prediction. In *Proceedings of ICCSA*, pages 411–425. Springer, 2017.

Tunik, E., Rice, N. J., Hamilton, A., and Grafton, S. T. Beyond grasping: representation of action in human anterior intraparietal sulcus. *Neuroimage*, 36: T77–T86, 2007.

Tuytelaars, T., Lampert, C. H., Blaschko, M. B., and Buntine, W. Unsupervised object discovery: A comparison. *International Journal of Computer Vision*, 88 (2):284–302, 2010.

Ugur, E., Sahin, E., and Oztop, E. Affordance learning from range data for multi-step planning. In *International Conference on Epigenetic Robotics*, 2009.

Umilta, M. A., Kohler, E., Gallese, V., Fogassi, L., Fadiga, L., Keysers, C., and Rizzolatti, G. I know what you are doing: A neurophysiological study. *Neuron*, 31(1):155–165, 2001.

Van Overwalle, F. and Baetens, K. Understanding others' actions and goals by mirror and mentalizing systems: a meta-analysis. *Neuroimage*, 48(3):564–584, 2009.

Van Rijsbergen, C. J. *Information Retrieval*. Butterworth-Heinemann, 2nd edition, London, 1979.

Varsta, M., Heikkonen, J., Millan, J. d. R., et al. *Context learning with the self-organizing map*. Citeseer, 1997.

Vasquez, D., Fraichard, T., Aycard, O., and Laugier, C. Intentional motion on-line learning and prediction. *Machine Vision and Applications*, 19(5):411–425, 2008.

Vatanen, T., Osmala, M., Raiko, T., Lagus, K., Sysi-Aho, M., Orešič, M., Honkela, T., and Lähdesmäki, H. Self-organization and missing values in SOM and GTM. *Neurocomputing*, 147:60–70, 2015.

Vesanto, J. Using the SOM and local models in time-series prediction. In *Proceedings of Workshop on Self-Organizing Maps 1997*, pages 209–214, 1997.

Vignolo, A., Noceti, N., Rea, F., Sciutti, A., Odone, F., and Sandini, G. Detecting biological motion for human–robot interaction: A link between perception and action. *Frontiers in Robotics and AI*, 4:14, 2017.

Voegtlin, T. Recursive self-organizing maps. *Neural Networks*, 15(8-9):979–991, 2002.

Von der Malsburg, C. Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14(2):85–100, 1973.

Voronoï, G. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les paralléllòèdres primitifs. *Journal für die reine und angewandte Mathematik*, 134:198–287, 1908.

Walter, J., Riter, H., and Schulten, K. Nonlinear prediction with self-organizing maps. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 589–594, 1990.

Wang, H., Ullah, M. M., Klaser, A., Laptev, I., and Schmid, C. Evaluation of local spatio-temporal features for action recognition. In *BMVC 2009-British Machine Vision Conference*, pages 124–1. BMVA Press, 2009.

Wang, J., Liu, Z., and Wu, Y. Learning actionlet ensemble for 3D human action recognition. In *Human Action Recognition with Depth Cameras*, pages 11–40. Springer International Publishing, 2014.

Want, S. C. and Harris, P. L. How do children ape? Applying concepts from the study of non-human primates to the developmental study of 'imitation' in children. *Developmental Science*, 5(1):1–14, 2002.

Wermter, S. *Hybrid neural systems*. Number 1778. Springer Science & Business Media, 2000.

Wikimedia. Wikimedia commons. Image showing dorsal stream (green) and ventral stream (purple) in the human visual system., 2007.

Wu, C., Zhang, J., Savarese, S., and Saxena, A. Watch-n-patch: Unsupervised understanding of actions and relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4362–4370, 2015.

Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., and Rehg, J. M. A scalable approach to activity recognition based on object use. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

Yang, X. and Tian, Y. Effective 3d action recognition using eigenjoints. *Journal of Visual Communication and Image Representation*, 25(1):2–11, 2014.

Yang, X., Zhang, C., and Tian, Y. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 1057–1060. ACM, 2012.

Yang, Y., Li, Y., Fermüller, C., and Aloimonos, Y. Robot learning manipulation action plans by "watching" unconstrained videos from the world wide web. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 3686–3693, 2015.

Yao, B. and Fei-Fei, L. Grouplet: A structured image representation for recognizing human and object interactions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 9–16. IEEE, 2010a.

Yao, B. and Fei-Fei, L. Modeling mutual context of object and human pose in human-object interaction activities. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 17–24. IEEE, 2010b.

Yao, B. and Fei-Fei, L. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1691–1703, 2012.

Yoon, E. Y., Humphreys, G. W., Kumar, S., and Rotshtein, P. The neural selection and integration of actions and objects: an fMRI study. *Journal of Cognitive Neuroscience*, 24(11):2268–2279, 2012.

Zheng, L., Yang, Y., and Tian, Q. SIFT meets CNN: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 2017.

Zhong, J., Weber, C., and Wermter, S. A predictive network architecture for a robust and smooth robot docking behavior. *Paladyn*, 3(4):172–180, 2012.

Ziaeefard, M. and Bergevin, R. Semantic human activity recognition: A literature review. *Pattern Recognition*, 48(8):2329–2345, 2015.

Declaration on Oath

I hereby declare on oath, that I have written the presented dissertation by my own and have not used other than the acknowledged resources and aids.

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ort, Datum Unterschrift

Erklärung zur Veröffentlichung

Ich erkläre mein Einverständnis mit der Einstellung dieser Dissertation in den Bestand der Bibliothek.

Ort, Datum Unterschrift