



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Inspection of Echo State Networks for Dynamic Gestures

Dissertation

zur Erlangung des Doktorgrades

an der Fakultät für Mathematik, Informatik und Naturwissenschaften

Fachbereich Informatik

der Universität Hamburg

eingereicht beim Fach-Promotionsausschuss Informatik von

Dipl.-Inf. Doreen Jirak

Hamburg, 2017

Submission of the thesis:
24 of February of 2017

Date of oral defense:
24 of April of 2017

Dissertation Committee:

Prof. Dr. Stefan Wermter (reviewer)
Dept. of Computer Science
Universität Hamburg, Germany

Prof. Dr.-Ing. H. Siegfried Stiehl (reviewer)
Dept. of Computer Science
Universität Hamburg, Germany

Prof. Dr.-Ing. Timo Gerkmann (chair)
Dept. of Computer Science
Universität Hamburg, Germany

©2017 by Doreen Jirak

All the illustrations, except where explicitly stated, are work by Doreen Jirak and are licensed under the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0).

To view a copy of this license, visit <https://creativecommons.org/licenses/by-sa/4.0/>

Abstract

The modeling of neural processes in the human brain and the explanation of learning human behavior has attracted the research areas of computational neuroscience, artificial neural networks, and cognitive robotics. Of particular interest is the processing of sequences naturally occurring in the acquisition of cognitive skills like learning how to play an instrument or to use language for the expression of our thoughts. Gestures are a special and interesting research subject as they display motor acts with linguistic properties. They are an essential, non-verbal component of human communication and as such a significant body of research investigated their evolutionary and developmental role in cognition.

Well-established neural models like recurrent neural networks were long a valuable method to learn sequential data. However, their gradient-based training exhibited multiple convergence problems, which limited their application success. The *Reservoir Computing* paradigm encouraged the development of novel recurrent network implementations, suggesting a functional separation between the representation of input data by a loosely coupled set of recurrent neurons (the “reservoir”) and a simplified training using linear models. Their design allows a fast training circumventing the drawbacks from classic training algorithms, and thus inspired research into network models implementing this paradigm in the scientific community.

The present thesis investigates the Echo State Networks (ESN) for the task of gesture recognition. This area is hardly explored in the current field of research, however, it seems reasonable to exploit the benefits of simple training and the performance success of ESNs in sequence tasks for gestures.

Our first experiment addresses the gesture representations and the performance in a specific ESN architecture. We introduce two approaches for vision-based processing of the gestures to support a natural gesture performance. For the feature extraction we define a *simple* and a *complex* feature set, which represent different characteristics of our data. Our performance evaluation on both sets reveal that the processing differs, and that for *complex*, respectively, high-dimensional features the architectural variant of an ESN is beneficial.

In the second chapter, we introduce the recurrence analysis for an adequate visualization and quantification of gestures and reservoir dynamics. The rationale is to complement research on the underlying theoretical stability conditions for proper functioning of the network. We demonstrate the validity of using recurrence quantification analysis (RQA) to measure different intrinsic properties of the data and their representation in the reservoir. Finally, we present examples of the

reservoir stability with respect to the spectral radius of the reservoir matrix and suggest a criterion for the detection of the transition phase between stable and unstable state, expressed by the fluctuation of crucial RQA measures.

Finally, we investigate statements and explanations for the successful performance of ESN. In particular, we address the stochastic initialization process which is often debated in the research community due to high performance variations. We show that using orthogonal matrix is beneficial to obtain good performance network with reduced experimental variations. The results are contrary to the explanation, that the diversity of eigenvalue magnitudes in random reservoirs are responsible for a sufficient input representation. They also complement other research into deterministic rules of reservoir initialization and ESN architectures. Further, we address the sparsity factor and the reservoir size, which are also usually considered to be effective factors for ESN performance.

All three chapters investigate a different aspect of ESNs connected to the gesture input and the performance in prediction or classification tasks. We show, that ESNs are a complementary method for gesture recognition, and substantially add to possible applications in this area.

Zusammenfassung

Die Modellierung neuronaler Prozesse im menschlichen Gehirn und das Verstehen von Lernprozessen menschlichen Handelns stehen seit jeher im wissenschaftlichen Fokus der Neurowissenschaften, dem Gebiet der künstlichen Intelligenz und der kognitiven Robotik. Von besonderem Interesse ist die Verarbeitung von Sequenzen, die in vielerlei Ausprägungen kognitiver Kompetenzen auftreten, so z.B. wenn wir das Spielen eines Instrumentes erlernen oder uns mithilfe von Sprache verständigen wollen. Als Gesten werden vielfältige motorische Hand-, Arm-, oder Körperbewegungen beschrieben, die in weiten Teilen linguistische Eigenschaften beinhalten. Sie sind eine fundamentale Komponente menschlicher nonverbaler Kommunikation, und aufgrund ihrer Besonderheit der Kopplung von Motorik und Sprachelementen vielbeachteter Untersuchungsgegenstand zur Erklärung evolutionärer Sprachgenese, und der entwicklungspsychologischen Rolle von Gesten auf den Erwerb kognitiver Fähigkeiten.

Für das automatische Lernen von Sequenzen sind besonders rekurrente neuronale Netze geeignet. Traditionelle Trainingsalgorithmen solcher Netze basieren auf gradienten-basierten Verfahren, die jedoch zu verschiedenen Konvergenzproblemen führen können. Diese Problematik begrenzt ihren praktischen Einsatz. Das *Reservoir Computing* Paradigma hingegen beruht auf einer funktionellen Teilung der Netzwerkarchitektur: das sogenannte Reservoir besteht aus zufällig initialisierten, rekurrent verbundenen Neuronen, welche den Netz-Input repräsentieren. Lediglich die Verbindungen zwischen dem Reservoir und der Netzausgabe werden mit Hilfe einfacher linearer Modelle trainiert. Diese Trainingsstrategie zum Lernen rekurrenter neuronaler Netze ist, im Vergleich zu den bekannten gradienten-basierten Algorithmen, schnell und hat neue Wege zur Erforschung der entsprechenden Netzwerke eröffnet.

In der vorliegenden Arbeit untersuchen wir die Eigenschaften von Echo State Networks (ESN) unter dem Gesichtspunkt der Gestenerkennung. Über die Kopplung der Netze und der Applikation ist bisher wenig bekannt, dennoch ist die gute Performanz der Echo State Networks in der Verarbeitung von Sequenzen ein schlüssiges Argument zu Erforschung wesentlicher Eigenschaften dieser.

Im ersten Kapitel dieser Arbeit präsentieren wir Gestenrepräsentationen zur Gestenerkennung und untersuchen die entsprechende Performanz in einer speziellen ESN-Architektur. Dazu werden zwei verschiedene Verfahren zur Merkmalsextraktion vorgestellt, die zur Erstellung einfacher und komplexer Repräsentationen dienen. Unsere experimentellen Resultate zeigen, dass das Netzwerk Performanzunterschiede in Hinblick auf die Architekturgegebenheiten aufweist, wobei vor allem

die komplexe Repräsentation davon profitiert.

Im zweiten Kapitel fokussieren wir uns auf das Konzept der Stabilität in ESNs und demonstrieren, dass Analysemethoden von Rekurrenzen valide Instrumente zur Visualisierung und Quantifizierung dynamischer Prozesse im Reservoir sind. Sie ergänzen bisherige Prozeduren zur Bestimmung der Netzwerkstabilität. Die Analyse der Rekurrenzen ermöglicht die Bestimmung der intrinsischen Eigenschaften des Netzwerk-Inputs und deren Repräsentation im “reservoir”. Wir präsentieren schliesslich Beispiele der Stabilität bzgl. des spektralen Radius und schlagen ein Kriterium zur Bestimmung der Transitionsphase zwischen stabilem und instabilem Zustand vor.

Das dritte Kapitel untersucht bekannte Annahmen und Erklärungen für die Performanz der ESNs. Ein bedeutendes Thema ist dabei die zufällige Initialisierung vorallem im Reservoir, da dies zu einer breiten Streuung der einzelnen Resultate führt. Wir zeigen, dass durch die Initialisierung mit orthogonalen Matrizen bereits eine wesentliche Reduktion dieser Varianz erfolgt, und geben so einen weiteren Hinweis darauf wie strukturelle Veränderungen wesentliche Vorteile hinsichtlich der Kontrollierbarkeit der Netzwerke und ihrer Performanz bringen können, ähnlich wie es bereits andere Arbeiten zu deterministischen Strukturen gezeigt haben. Desweiteren gehen wir auf die sogenannte “sparsity” und die Reservoirgröße ein, die ebenfalls wesentliche Faktoren in der ESN Performanz sind.

Alle drei Kapitel untersuchen verschiedene Aspekte der ESN in Hinblick auf die Gesten als Input-Daten und in Prediktions- und Klasifikationsszenarien. Wir zeigen, dass die ESNs wertvolle Werkzeuge zur Gestenerkennung sind und wesentlicher Bestandteil zukünftiger Applikationen sein werden.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Objectives of the Thesis | 3 |
| 1.3 | Contributions of the Thesis | 3 |
| 1.4 | Thesis Outline | 4 |
| 2 | Gesture Types and Their Learning Models | 7 |
| 2.1 | Gesture Taxonomy | 7 |
| 2.2 | Gestures Involved in Cognitive Tasks | 11 |
| 2.3 | System Development Stages | 12 |
| 2.3.1 | Gesture Recordings | 12 |
| 2.3.2 | Techniques for Gesture Representations | 14 |
| 2.4 | Modelling and Learning Gestures | 20 |
| 2.4.1 | Probabilistic Graphical Models | 21 |
| 2.4.2 | Gesture Recognition with Artificial Neural Networks | 25 |
| 2.5 | Chapter Summary | 29 |
| 3 | Neural Processes and the Emergence of Computational Models | 31 |
| 3.1 | From Neurons to Cortical Structures | 31 |
| 3.1.1 | Computations on the Neuronal Level | 32 |
| 3.1.2 | Neural Plasticity Shapes the Brain | 33 |
| 3.1.3 | Functional Areas of the Cortex | 34 |
| 3.2 | Sequence Learning and Transient Dynamics | 38 |
| 3.2.1 | Random Recurrent Networks | 38 |
| 3.2.2 | Transient Dynamics Involved in Learning | 40 |
| 3.3 | Chapter Summary | 41 |
| 4 | Recurrent Neural Networks and Reservoir Computing | 43 |
| 4.1 | Echo State Networks | 46 |

| | | |
|----------|---|------------|
| 4.1.1 | Important Equations and Evaluation Measures for ESN | 47 |
| 4.1.2 | Network Parameters and Reservoir Configurations | 52 |
| 4.2 | Stability in Echo State Networks | 57 |
| 4.3 | Memory in an Echo State Network | 61 |
| 4.4 | Important Definitions and Concepts from Dynamical Systems | 63 |
| 4.5 | Chapter Summary | 66 |
| 5 | Processing Dynamic Gestures with Different Representations | 67 |
| 5.1 | Gesture Recordings and Preprocessing | 67 |
| 5.1.1 | Gesture Performance Description | 68 |
| 5.1.2 | Preprocessing and Feature Extraction | 70 |
| 5.2 | Gesture Recognition Experiment | 75 |
| 5.3 | Results and Evaluation | 77 |
| 5.4 | Insights from Single Reservoirs | 80 |
| 5.5 | Chapter Summary and Discussion | 84 |
| 6 | Recurrence Analysis for Gesture Sequences and the Reservoir | 87 |
| 6.1 | User-Independent Sequences | 88 |
| 6.2 | Visualization and Dynamics | 90 |
| 6.2.1 | Phase Space Reconstruction and Recurrence Plots | 91 |
| 6.2.2 | Recurrence Quantification Analysis | 100 |
| 6.2.3 | Reservoir Dynamics | 104 |
| 6.3 | Chapter Summary | 112 |
| 7 | Reservoir Initialization and Pruning | 113 |
| 7.1 | Getting Good Reservoirs is a Chance | 114 |
| 7.1.1 | Weight Matrix Initialization | 115 |
| 7.1.2 | Sensitivity of Reservoir Connections | 118 |
| 7.2 | Pruning Procedures in Reservoirs | 120 |
| 7.2.1 | Experimental Section | 123 |
| 7.2.2 | Graph Theoretic Analysis | 128 |
| 7.3 | Chapter Summary | 132 |
| 8 | Thesis Discussion | 135 |
| 8.1 | Gesture Representations | 136 |
| 8.2 | Visualization and Quantification of Reservoir Dynamics | 138 |
| 8.3 | Reservoir Initialization and Pruning | 139 |
| 8.4 | Limitations and Future Work | 140 |

| | |
|---|------------|
| 8.5 Conclusion | 141 |
| A Additional Technical Information and Equations | 143 |
| B Publications Originating from this Thesis | 145 |
| C Acknowledgements | 147 |
| Bibliography | 149 |

List of Figures

| | | |
|-----|--|----|
| 2.1 | Gesture types with increasing linguistic meaning, which makes gestures independent of accompanying speech. The figure follows Kendon’s continuum but differs from it by the addition of the dashed boxes. | 8 |
| 2.2 | Pointing gesture to different objects in a scenario with the humanoid NAO platform. | 10 |
| 2.3 | An emblem with different meanings including ok, zero, and money or coins. In some countries, this gesture is regarded as an insult. (Image source: Pixabay) | 10 |
| 2.4 | Processing pipeline for design decisions involved in the development of a gesture recognition system. | 12 |
| 2.5 | Devices for vision-based gesture recognition. Left: The NAO humanoid robot (Aldebaran company) and the NimbRo-OP (University of Bonn) equipped with cameras. The vertical arrangement of the two cameras on the NAO robot does not allow stereoscopic image caption because their visual fields do not overlap. (Photos from KT lab) | 14 |
| 2.6 | Gesture tracking devices. a) Hardware for specific finger movements (Jing et al., 2012). b) Nintendo Wii [®] controller, which uses inertial measurements to track dynamic arm movements (Schlömer et al., 2008) c) A glove for hand tracking from Fujitsu company in a slim version inspired by traditional data gloves. d) DG5-V Data Glove. Both devices can capture both finger- and hand movements but can be expensive and need to be newly calibrated when users change. | 15 |
| 2.7 | Examples of 10 postures on 3 backgrounds from Triesch and von der Malsburg (1996). | 16 |

| | | |
|------|--|----|
| 2.8 | Examples of hand gesture representations. Left: A pointing gesture with optical flow vectors using the Lucas-Kanade algorithm (blue arrows, from own recordings). Middle: Hand extraction from a usual web camera for a representation of the finger configurations, displayed by the red and green points. Right: Image from a depth sensor with significant body points (Parisi et al., 2014). | 19 |
| 2.9 | Examples of a hand model. Hand movements of a subject are tracked and rendered to display different hand poses (Hamester et al., 2013). . . | 20 |
| 2.10 | An HMM defined as $\{A, B, \pi\}$. X depicts the states and Y the emissions. Training an HMM aims to find a model which fits best the set of the hidden state sequences (state transition matrix A) with the most probable emission sequence (emission probability matrix B). The dashed lines show that only the emissions are available while the states and their transitions producing them is hidden. The initial start probability for a state is given by π . HMMs belong to the class of generative models and are directed acyclic graphs. | 22 |
| 2.11 | A undirected probabilistic graph model. The hidden Conditional Random Field contains an extra layer h serving as latent variables, which model specific parts of a sequence conditioned on the input X. Leaving the extra layer h provides the standard formulation for a CRF. In contrast to the HMM, Y assigns here a class label for the complete sequence. CRFs belong to the class of discriminative models. | 23 |
| 3.1 | Sketch of a neuron and the main parts responsible for the propagation of action potentials. The embedded figure displays a chemical synapse. . | 33 |
| 3.2 | The Hebbian learning rule states, that the synapse connection ω_{ij} between two neurons x_i and x_j strengthens when both neurons are activated simultaneously. The connection weakens if the two neurons get activated separately. | 34 |
| 3.3 | Sketch of the human brain. (Image modified from Pixabay) | 35 |
| 4.1 | The perceptron model with the input layer and an additional bias (constant 1 node). The sum of the input and their corresponding weights is passed to an activation function, here the Heaviside function. This model computes linear decision boundaries. | 44 |

| | | |
|------|--|----|
| 4.2 | Left: Elman network with an additional context layer. The weights are fixed to 1, as the hidden layer activations from the previous time step are copied into it. Right: Neural network with recurrently connected neurons in two hidden layers (dashed box). | 45 |
| 4.3 | Algorithms and network architectures primarily connected to the Reservoir Computing paradigm as proposed by Verstraeten et al. (2007). The Backpropagation Decorrelation learning rule is built on the Atiya-Parlos algorithm which approximates the weight changes in recurrent learning. Echo State Networks (ESN) have been widely adopted in machine learning while the Liquid State Machine with its specific neuron type and topology has gained more attention in the neuroscientific community. . . | 46 |
| 4.4 | The <i>tanh</i> activation function usually used for the reservoir. | 47 |
| 4.5 | Separation of data samples in spaces of different dimensionality. Left: a nonlinear function is needed to correctly classify data samples in \mathbb{R}^2 . Right: Transformation of the data with an additional dimension, i.e. here, \mathbb{R}^3 , provides a linear separable classification. The data is then separated by a hyperplane (here only indicated with the triangle for visualization purpose). | 47 |
| 4.6 | Echo State Network as introduced by Jaeger (2002) and referred to as standard ESN throughout the thesis. The M -dimensional input u projects via the weight matrix $W_{in} \in \mathbb{R}^N \times (1 + L)$ to the reservoir of size $N \times N$, whose connectivity is denoted by κ . The additional 1 denotes a bias term. The norm of the reservoir matrix W_{res} is denoted here as ρ (spectral radius). The parameter α denotes the leakage rate. The output weight matrix W_{out} (dashed red arrow) is the only network component which is trained via e.g. a regression. All matrices except W_{out} are randomly initialized and the connections stay fixed. Other architectural options use a direct input-output connection and interconnections between output neurons. The model also provides the feedback matrix W_{back} , which can be used for pattern generation tasks. | 50 |
| 4.7 | Reservoir activations from arbitrarily chosen neurons over time. | 50 |
| 4.8 | Training and prediction of the NARMA sequence using RLS learning and for 100 time steps. | 52 |
| 4.9 | Evolution of the weights using RLS on the NARMA task. | 52 |
| 4.10 | Weights drawn from a uniform probability $\mathcal{U} \in [-0.5;0.5]$ in a sparsely connected reservoir of size 100 with $\kappa = 0.1$. Only 10% of the neurons are connected. | 57 |

| | | |
|------|--|----|
| 4.11 | Sequence plot of Lorenz system with the parameters $\pi = 10$, $\rho = 28$ and $\gamma = 8/3$. The Lyapunov exponent is $\lambda_1 > 0$ | 64 |
| 4.12 | Phase portrait of the Lorenz attractor projected to the 2D plane. It shows the two distinctive areas where the system is attracted to but does not converge, i.e. there is no fix point. | 65 |
| 5.1 | Sketch of the gestures performed. From left to right: <i>circle</i> , <i>point left</i> , <i>point right</i> , <i>stop</i> , <i>turn</i> . The red crosses depict the directions. | 68 |
| 5.2 | The start and the end position during the recordings. | 68 |
| 5.3 | Examples from the recordings. Left image: the <i>circle</i> gesture. Right image: the <i>turn</i> gesture. | 69 |
| 5.4 | Variance of gesture performance of one subject for the five defined gestures carried out ten times. The y-axis denotes the time needed to perform the gestures in seconds. The + marks an outlier. | 70 |
| 5.5 | Variance of gesture performance of one subject for the five defined gestures carried out ten times. The y-axis denotes the time needed to perform the gestures in seconds. The + mark outliers. | 71 |
| 5.6 | For the connected component analysis we used the 8-pixel connectivity, i.e. considering the vertical, horizontal, and diagonal neighborhood of pixel p_i | 72 |
| 5.7 | Image sequence of a <i>stop</i> gesture and the resultant binary images yielding the hand and the face region. Note the noise corruption at this stage, which is observable from the small white areas and interferences in the hand and face area. | 72 |
| 5.8 | The preprocessing steps and the resultant hand extraction from frames of a <i>stop</i> gesture. An ellipse (light blue) is fitted to the global hand shape and the corresponding hand orientation is computed. The hand center (x, y -coordinates) is shown with the magenta cross. | 73 |
| 5.9 | Example of a receptive field with different pixel values. The <i>max</i> pooling operation compresses this image patch to its maximum value. This procedure is applied to all receptive fields of an image, whose concrete size depends on the predefined filter size. | 74 |
| 5.10 | A raw gesture sequence is fed to the MCCNN, where each image is processed by three channels. Each channel contains two convolutional layers, followed by a <i>max</i> pooling operation. | 75 |

| | | |
|------|--|----|
| 5.11 | Evaluation of the results from the test set using the <i>simple</i> feature set over the ensemble sizes. The boxplots show the distribution and the median of misclassifications over the 30 trials with varying reservoir sizes (4 – 9), the leakage rate $\alpha = 0.2$ and $\ell = 3$ remain fixed (from left to right, top to bottom). The dashed line displays the mean classification. Outliers are marked by \circ | 77 |
| 5.12 | Results for a varying leakage rate $\alpha = 0.1, 0.2, 0.3$ (top to bottom) and fixed reservoir size = 4 and $\ell = 4$. There are no outliers. | 79 |
| 5.13 | Evaluation of the results from the test set using the <i>complex</i> feature set over the ensemble sizes. The boxplots show the distribution and the median of misclassifications over the 30 trials with varying reservoir sizes (4 – 9), the leakage rate $\alpha = 0.2$ and $\ell = 3$ remain fixed (from left to right, top to bottom). The dashed line displays the mean classification. Outliers are marked by \circ | 80 |
| 5.14 | Top: Results from the tests of the <i>complex</i> feature set for a fixed reservoir size = 5 and leakage $\alpha = 0.1$. For $\ell = 3$, the experimental results display numerous outliers (\circ). Bottom: Incrementing the parameter to $\ell = 4$ shows a smoother picture and improved classification result. | 81 |
| 5.15 | Comparisons of both feature sets. a): Evaluation of the experiments using different ensemble sizes for the <i>simple</i> feature set. Experiment 1: $\alpha = 0.2$, $\ell = 3$ and #reservoir neurons=9. The dashed lines show the trend for misclassification. Experiment 2: $\alpha = 0.2$, $\ell = 3$ and #reservoir neurons=4. b) Evaluation of the experiments using different ensemble sizes for the <i>complex</i> feature set with the same parameter configurations as in a). | 82 |
| 5.16 | Confusion matrix depicting the classification performance on a reservoir with equal parameters ($\alpha = 0.1$, $\rho = 0.8$), but different reservoir sizes. Left: 100 neurons. Right: 200 neurons. | 83 |
| 5.17 | Confusion matrix showing the effect of the leakage rate. Keeping the 100 reservoir neurons but increasing the leakage rate from $\alpha = 0.1$ (left image) to $\alpha = 0.3$ (right image) decreases the number of misclassifications. | 83 |
| 5.18 | Example output and the corresponding activations (y-axis) of test sequences showing the influence of the gestures on each other. The gestures were fed in sequentially (x-axis). The <i>gesticulation</i> gesture is abbreviated as ‘none’ in the figure legend. | 84 |

| | | |
|------|--|----|
| 6.1 | A <i>circle</i> gesture sequence showing the horizontal movement from the <i>5DG</i> set (magenta) compared to two filter methods used in our analysis. The effect of the filter is mainly smoothing irregularities between sample points introduced by noise or postprocessing. | 89 |
| 6.2 | Example of a <i>circle</i> gesture from the <i>5DG+E</i> set: the original version (red), the filtered (green), and noisy version (gray) from this sequence. . | 90 |
| 6.3 | Illustration of recurrence in an idealized phase space for two trajectories $x(i\Delta t)$ and $x(j\Delta t)$ (Δt is the time interval) determined by the radius ϵ (red ring). If this value would be too small, only one line would be detected and thus no recurrence. In contrast, setting this value too high would include more lines and may distort the interpretation of dynamics. | 92 |
| 6.4 | Recurrence plot of an i.i.d. time-series with $\epsilon = 0.3$. The RP exhibits no regular line patterns but only single dots due to the uncorrelated nature of the input signal. | 93 |
| 6.5 | Recurrence plot of the Lorenz system with an embedding dimension $\mu = 3$ and delay $\tau = 4$. In contrast to the irregular pattern in Figure 6.4, a chaotic system exhibits small diagonal lines along the LOI. This indicates recurrences for only a few times in the phase space, alternating with white compartments. | 94 |
| 6.6 | Estimation of the delay parameter using the autocorrelation function yields $\tau = 6$ | 95 |
| 6.7 | Estimation of the percentage of presumably nearest neighbors for determination of the embedding parameter μ . We chose the <i>circle</i> sequences shown in Figure 6.2: the original sequence derived from the <i>5DG</i> recordings, and the filtered, respectively, noisy version from that sequence considering the x -direction. | 95 |
| 6.8 | Estimation of the percentage of presumably nearest neighbors for determination of the embedding parameter μ for <i>point left</i> | 96 |
| 6.9 | Recurrence plot of a sequence of <i>stop</i> gestures (y-direction, $\mu = 3$, $\tau = 18$, $\epsilon = 0.2$). Due to the low movements in the gesture performances, the plot displays many laminar states (black blocks). | 97 |
| 6.10 | Recurrence plot of a sequence of <i>turn</i> gestures (x-direction, $\mu = 6$, $\tau = 12$, $\epsilon = 0.7$), which exhibit more movements displayed by the smaller distances between the lines in contrast to the <i>stop</i> gestures. | 97 |

6.11 Recurrence plot of a sequence of filtered *turn* gestures shown in the upper plot (x-direction, $\mu = 8$, $\tau = 5$, $\epsilon = 0.7$), which exhibit more movements displayed by the smaller distances between the lines in contrast to the *stop* gestures. The high value of $\epsilon = 0.7$ is chosen for a better visualization. 98

6.12 Recurrence plot of a sequence of *turn* gestures shown in the upper plot (x-direction, $\mu = 6$, $\tau = 12$, $\epsilon = 0.7$) reflecting the influence of noise. Almost all periodic structure is lost. 99

6.13 Top: Time-series of the motion from a *circle* gesture recorded with the Kinect device. Bottom: According RP of the time-series. The embedding parameters were $\tau = 7$ and $\mu = 5$. Although periodic in nature, the time-series reveals an irregular pattern of the gesture, which is performed several times within the sequence. The plot shows spurious lines, which can be interpreted as a chaotic structure (but a careful analysis should be given). The RP was thresholded with an $\epsilon = 0.7$ 100

6.14 Validation of the alignment of reservoir states as shown in Bianchi et al. (2016a) for the input $\sin(\phi k)$, $\phi = 3/50$ and $k = 1 \dots 5000$ 106

6.15 Left: Activations of 50 reservoir neurons processing the *turn* gesture performed several times (i.e. the discretized time, respectively frame-wise processing). Between gesture pauses, i.e. end of a gesture and start of a new one, the activations remain steady, depicted by the column-like structure between time intervals (the RP would exhibit laminar states). . 107

6.16 Corresponding recurrence plot of reservoir activations. The laminar states occur for low activity in the reservoir. The diagonal lines display the periodic characteristics of the gesture. 108

6.17 Reservoir activations when $\rho = 1.5$. The neurons display abrupt changes in their activities. 109

6.18 The corresponding recurrence plot of reservoir activations when $\rho = 1.5$. 109

6.19 Lorenz system evolution of the x-component. 110

6.20 RP of the corresponding reservoir activations for the Lorenz system, which equals the unembedded phase portrait for the x-component. . . . 110

6.21 L_{max} values over the spectral radius ρ averaged over 10 trials. The curve shows a small decrease for values up to $\rho = 1.0$, but drops significantly for $\rho = 1.2$, indicating that the reservoir enters the chaotic regime. . . . 111

6.22 Recurrence rate RR for different spectral radii ρ averaged over 10 trials. The curve shows a similar trend as for L_{max} with only small variances up to $\rho = 1.1$, followed by a significant decrease. 111

| | | |
|------|---|-----|
| 7.1 | Left: Random matrix M generated with $n=200$ and scaled $M/\sqrt{(n)}$. The eigenvalues of M scatter around the unit circle with different magnitudes. Right: Eigenvalue distribution shows equal magnitudes for an orthogonal matrix, i.e. $MM^T = \mathbf{I}$ | 116 |
| 7.2 | Result of 30 trials using a 1-step ahead prediction on a <i>turn</i> gesture using random initialization of the reservoir (left) and orthogonal matrix (right). | 116 |
| 7.3 | Result of 30 trials using a 1-step ahead prediction on a <i>point left</i> gesture using random initialization of the reservoir (left) and orthogonal matrix (right). The + display outliers. | 117 |
| 7.4 | Display of the results shown in Figure 7.3 from a subset of the trials (1-10) using random initialization of the reservoir (left) and orthogonal matrix (right). The orthogonal initialization has the better performance and the graph displays a similar error distribution. In contrast, the random reservoir has a great maximum value, which becomes an outlier when averaged over more trials (cf. Figure 7.3). | 118 |
| 7.5 | Random pruning of reservoir connections of a network with $r_N = 50$. The figure shows the matrix W_{res} after 20 iterations. Red circles depict examples of synapse loss. | 122 |
| 7.6 | Track of the spectral radius when iteratively cropping elements from the reservoir of size $r_N = 50$. The red curve shows the average decay from 50 reservoirs. | 123 |
| 7.7 | Misclassification results from the 1st trial for set 1 (157 sequences). . . | 128 |
| 7.8 | Misclassification results from the 1st trial for set 2 (79 sequences). . . | 128 |
| 7.9 | Misclassification results from the 6th trial for set 1 (157 sequences). . . | 129 |
| 7.10 | Misclassification results from the 6th trial for set 2 (79 sequences). . . | 129 |
| 7.11 | Example of an adjacency matrix size 6×6 for a directed graph. An 1 is assigned whenever there is a connection between two nodes, else there is a 0 entry. A 1 on the diagonal (green) depicts a loop or self-reference ($i = j$). The out-degree δ^+ can be derived from the row, while the in-degree δ^- can be determined from the column. | 130 |
| 7.12 | Distribution of the neuron out-degree for a reservoir of size $r_N = 100$ and sparsity $\kappa = 0.1$ | 131 |
| 7.13 | Distribution of the neuron out-degree for a reservoir of size $r_N = 100$ and sparsity $\kappa = 0.25$ | 131 |

7.14 Distribution of the neuron out-degree from the reservoir of the 10th trial.
With 348 pruned connections its sparsity is low. A clear separation of the
different out-degrees compared to e.g Figure 7.13 is not visible. However,
the number of neurons with a high out-degree is decreasing. 132

A.1 Mackey-Glass time series with $\tau = 17$ 144

List of Tables

| | | |
|-----|---|-----|
| 2.1 | Overview of different gesture types | 11 |
| 4.1 | Lyapunov exponent | 65 |
| 5.1 | Description of the gesture performance | 69 |
| 5.2 | Range of parameter values | 76 |
| 6.1 | Gesture datasets | 90 |
| 6.2 | Some RQA measures for a <i>circle</i> gesture | 102 |
| 6.3 | Some RQA measures for a <i>point left</i> gesture | 102 |
| 7.1 | MSE results from synapse or neuron removal | 120 |
| 7.2 | Average sequence lengths | 124 |
| 7.3 | Variance explained | 125 |
| 7.4 | Results from the pruning of a 50-neuron reservoir | 127 |

Chapter 1

Introduction

Elle lui fait signe de la suivre. Il ne comprend pas ses mots mais il comprend ses gestes.

*La Petite Fille de Monsieur Linh,
Philippe Claudel*

1.1 Motivation

The *Reservoir Computing* paradigm for modeling and learning sequential data opened a new perspective on training recurrent neural networks (RNN). Traditional RNN architectures are trained with gradient-based techniques, where the backpropagation-through-time (BPTT) algorithm is the most popular implementation. However, the circulation of the error signal along the layers of an RNN leads to convergence failures, mostly due to exploding or vanishing gradients (Bengio et al., 1994; Hochreiter et al., 2001). Despite the introduction of training optimizations to overcome the problems of learning (Williams and Zipser, 1989; Martens and Sutskever, 2011; Bengio et al., 2013), the research interests in the last years shifted to other neural architectures, specifically deep neural networks, and lightweight algorithms exploring machine learning techniques.

Inspired by sequence processing in the human prefrontal cortex, neural networks based on *Reservoir Computing* build on a functional separation of mapping an input into feature representations in a high-dimensional “reservoir” which consists of random recurrently connected neurons and remains untrained. The information from the reservoir is then read out using a simple linear model like a regression. The established networks implementing the principles of *Reservoir Computing* are

the Liquid State Machines (LSM) (Maass et al., 2002) and the Echo State Networks (ESN, the focus of the present thesis) (Jaeger, 2002), both concurrently but independently developed. Both architectures differ in their neuron types, where the LSM implements spiking neurons and the ESN both analog and leaky-integrator neurons. The functioning of an ESN was long to be a “black box”, and research into this network type primarily considers conditions from dynamical system theory. As a consequence, Prokhorov (2005) even questioned the application of ESNs in the practical domain.

In the last decade, however, numerous applications in diverse research areas occurred for the language processing (Dominey, 2005; Tong et al., 2007; Hinaut and Dominey, 2013), robotics (Hartland and Bredeche, 2007; Waegeman et al., 2009; Oubbati et al., 2010) and prediction of time-series (Jaeger and Haas (2004); Hellbach et al. (2008); Deihimi and Showkati (2012)). Until today, surprisingly less is known about ESNs for vision-related tasks like action or gesture recognition. Complementary to these studies, the theoretical aspects of computations with ESN based on the introduction by Jaeger (2001a) were outlined in Buehner and Young (2006), Ozturk et al. (2007), Jaeger et al. (2007), and Yildiz et al. (2012), each addressing the particular conditions and parameters necessary for proper functioning of an ESN. It became apparent that application-driven research was misguided by some of the requirements introduced by theory (Caluwaerts et al., 2013), and that only tuning the ESN parameters using standard search procedures neglect investigations into reservoir dynamics and the parameter interplay.

A vital field of ESN research advances the coupling of both, the theoretical and practical facets. The principal investigations address the deviation from the random reservoir initialization (Rodan and Tiño, 2010; Rodan and Tino, 2011; Strauss et al., 2012), stability conditions and memory (Verstraeten and Schrauwen, 2009; Pascanu and Jaeger, 2011; Boedecker et al., 2012; Barancok and Farkas, 2014; Bianchi et al., 2016b), and the reservoir state-space organization (Gallicchio and Micheli, 2011). These approaches provide substantial advice for ESN design and help to unveil the computational and architectural properties of the networks. In addition, a thorough experimental guideline outlined by Lukoševičius and Jaeger (2009) further supports linking well-established criteria for an ESN with the varying tasks such a network can solve.

1.2 Objectives of the Thesis

In the present thesis, we aim at contributing to the controversial opinions in the ESN research connected to the task of gesture recognition. Gestures essentially carry information and are one of the building blocks in human communication. With the advent of new sensor technologies, gestures attract also more and more attention in the area of human-machine interaction. However, the most intuitive gesture performance for humans is still realized through vision-based techniques. Therefore, we are interested in the different gesture representations extracted from videos sequences and their influence on the classification performance in ESNs.

Another driving force of our thesis considers the recurrence and stability in Echo State Networks for varying parameter. We suspect that tracking the neuron activity in the reservoir phase space allows a visualization of specific states the underlying system revisits. The corresponding phase portrait allows interpretation of the input and their representation in the network. In particular, our main objective is to examine the validity of complex system science tools for the reservoir analysis and detection of stability borders.

Finally, we focus specifically on the reservoir in terms of their initialization and their connectivity for gesture recognition in a *one-shot* learning scenario. We are particularly interested in the role of the reservoir size, as well as the connectivity or sparsity in a single reservoir for the discrimination of gesture sequences varying in their motion profile.

1.3 Contributions of the Thesis

- We introduce a set of user-dependent command gestures for vision-based gesture recognition, followed by a scheme to expand this set with user-independent sequences to obtain variations.
- We introduce the notation of *simple* and *complex* features for the gesture representation and its influence on the performance of ESNs following the notation of ensembles. The *complex* feature set was derived from a deep neural network, where the training on the gesture sequences yields a set of image coefficients.
- We present an analysis of gesture data using recurrence plots, linking useful methodologies from complex systems to unveil specific characteristics of gesture sequence variants. Notably, the analysis is not restricted to our prob-

lem domain but may stimulate further explorations to other sequence-related tasks.

- We show that the recurrence quantification analysis (RQA) is a viable tool to determine input-driven reservoir stability, which complements current predictors based on the Lyapunov exponent. The design of a criterion allows detection of the *edge of stability*.
- We demonstrate that little alteration on the initialization improves the chance for good reservoirs, supporting current research on this topic.
- We introduce a pruning strategy and demonstrate which factors are specifically interesting for the gesture recognition of different gestures considering only a single reservoir.

1.4 Thesis Outline

Chapter 2 gives an overview of the area of gesture recognition. We start with the introduction of the most significant terms for the definition of different gesture types. The task of gesture recognition is split into several stages summarized in a system pipeline for general guidance. Particular focus is then given on hand and arm gestures, where we outline endorsed preprocessing and feature extraction methods for both static gestures (postures) and dynamic gestures. The last two sections of chapter 2 highlight traditional sequence models in the area of gesture recognition and core literature, followed by a review of literature employing artificial neural networks.

In chapter 3 we will present the neural information processing in the human brain, which serves as a basis to understand the abstractions in computer models. We will specifically focus on the neurophysiological basis of sequencing and emergence of the *Reservoir Computing* paradigm. We will also highlight the gestures being complementary to language.

Chapter 4 introduces the working principles of recurrent neural networks (RNN), where we highlight main differences between conventional RNN and architectures following the *Reservoir Computing* paradigm. As this thesis employs Echo State Networks, formulations for the functioning of this model will be given. Of special interest are the underlying dynamical system properties and rules to govern them, explicitly stability boundaries for proper functioning and corresponding theorems. The chapter closes with observations of memory properties in Echo State Networks.

In chapter 5, we will first present our gesture definition, recordings, and subsequent preprocessing schemes. We will then explain our experimental set up comprising the adoption of ESN-ensembles, followed by a performance evaluation on the used feature sets. The chapter is an entry to gesture recognition with this model per se and thus concludes with several questions concerning the underlying network properties and stability issues in light of the task.

In chapter 6, we introduce an expansion of the gesture dataset used in chapter 5 and set up a gesture analysis introducing, thus linking, the methodology of recurrence plots (RP) and recurrence quantification analysis (RQA), which originate from analysis of nonlinear dynamics and complex systems. We, therefore, gather the embedding-delay procedure and give examples on different sequences together with visualization examples demonstrating the usefulness of RPs for gesture analysis, and the effect on reservoir representations.

The individual RQA measures obtained are introduced and explained. We give examples of reservoir dynamics for different sequences and show the behavior of RQA measures when driving the reservoir into an unstable regime. We derive a criterion to capture the critical phase for a reservoir adding to existent approaches.

In chapter 7, we address specifically the reservoir. In an experiment on predictions, we show that only a little modification in the reservoir initialization decreases the experimental variability, which is often emphasized to hinder practical applications. In a second experiment, we consider pruning reservoir connections. This approach is inspired by pruning processes in the human brain and in the context of ESNs we have a tool to identify “useless” connections. We introduce a pruning strategy in a *one shot* learning scenario. We demonstrate, that in a single reservoir for the discrimination of different gestures the connectivity, respectively the sparsity, may play a different role than usually stated for ESN applications.

Finally, in chapter 8 we will summarize the thesis and discuss advantages, contributions, but also limitations of our work. We close this chapter with suggestions on how to extend the techniques presented in this thesis for future work.

Chapter 2

Gesture Types and Their Learning Models

Gestures complement our everyday communication besides speech, body language and facial expressions displaying emotions. Gestures are highly diverse and subject to interpretation in different contexts, for instance, diverse cultural environments. The different gesture types range from pure gesticulation to speech-replacing gestures, but no unified taxonomy exists until today. Gestures can have a standalone character or support verbal communication: for instance, doing a stop gesture, i.e. showing the palm in front of another person is understood as “do not approach any further”. We also use gestures in emotional events, for example moving the arm up with the hand as a fist when winning in sports, or putting both hands on the face hiding eyes when we feel despair or sadness.

We introduce the area of gesture recognition by explaining the different gesture types, each demanding a distinct way of processing gesture data. We proceed with the outline of most common feature extraction methods in this area, distinguishing between *static* and *dynamic* gestures. This links to some open access benchmark data descriptions and classification techniques. Furthermore, we outline probabilistic graphical models, which were traditionally used for gesture recognition. Finally, we give an overview of alternative, neural network-based approaches.

2.1 Gesture Taxonomy

Gestures, from the Latin word “gestura” or “gerereare”, are defined by the Oxford dictionary as “*A movement of part of the body, especially a hand or the head, to express an idea or meaning*” (Stevenson, 2010). This definition highlights gestures

as actions either performed manually or using the head, which is encountered in e.g. storytelling, where gestures are an important part of communication, or navigating an aircraft because the communication distance and noise hinder verbal communication. The expressions can also be symbolic and be used to transmit emotions: “*She was touched by his friendly gesture*” or intentions: “*He invited him as a conciliatory gesture.*”

Gestures as a subject of research became interesting when computers were equipped with graphical user interfaces, bringing computer technologies in every household. The research area of Human-Computer Interaction (HCI) emerged, which investigates gestures as input modality and usage in virtual reality or games. Within this field, an important subarea, Human-Robot Interaction (HRI), shows increased employment of gestures and their recognition. However, no consistent taxonomy for the various gesture types exists to date, although a standard reference in the area of gesture research is the work of McNeill (1992) and Kendon (1983). Kendon’s continuum is an attempt to classify distinct gestures types and is illustrated in Figure 2.1 in a revised form with the addition of *deictic* and *iconic* gestures. The continuum is arranged from the left to the right displaying gesture types with increasing level of linguistic features which the gestures convey, culminating in sign language which replaces verbal communication.

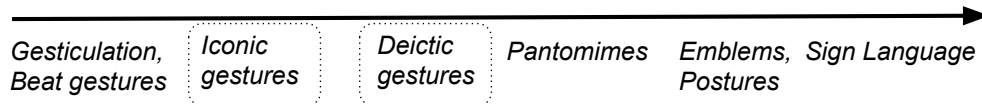


Figure 2.1: Gesture types with increasing linguistic meaning, which makes gestures independent of accompanying speech. The figure follows Kendon’s continuum but differs from it by the addition of the dashed boxes.

Gesticulation may be some form of seemingly random hand- or arm movements. In fact, we tend to gesticulate even in the absence of a communication partner, e.g. when vividly describing a happy event on a phone (de Ruiter, 1995; Wei, 2006), which led to the metaphor of “speaking hands”. A thorough investigation of gestures influencing higher cognitive tasks and gesture production of blind people is provided in Goldin-Meadow (2003).

In McNeill (1992) the connection between speech and gestures was termed *co-speech* gestures and describes four speech-related gestures, which we explain now. The *beat* gestures were introduced as a term for hand movements in front of the body, aligned with the rhythm while we are talking or when we emphasize something in our talk. This type of gesture is not captured by the Kendon continuum,

and as this type is not relevant to the present thesis, we summarize *gesticulation* and *beat* in Table 2.1.

Iconic gestures are elicited in a narrative and thus aligned with speech. Again, as there is no unified convention on gestures, *iconic* gestures are sometimes also categorized into the set of *gesticulation* or called synonymously as *metaphoric* (McNeill, 1992). Examples are “We rolled down the hill ” or “The book is so thick”, which implies also a high congruent relationship between what is said and what is shown.

Deictic gestures, often synonymously called *pointing* gestures, are used for a spatial reference to real persons, objects, locations or directions, or as part in a narrative. The latter usage explains why often there is a fuzzy transition between *iconic* and *pointing* gestures. We subdivided these gestures to account for their different requirements in an application, i.e. we assume that *iconic* gestures are connected to speech input while *pointing* gestures can also be understood without specific speech input (Figure 2.1). The *pointing* gesture has also attracted research in the area of developmental psychology, which investigates the role of this gesture type in infants. These studies may provide an understanding of communication in the prelinguistic phase (Melinder et al., 2015). How *pointing* can facilitate (joint) attention was investigated in Tomasello et al. (2007). Whether infants comprehend the intention of actions when a game is explained using gestures was for example addressed in Liszkowski (2014), including behavioral tasks in a non-structured environment. Finally, *pointing* gestures are used to spatially reference to an object (Cappuccio et al., 2013) (spatial cognition) as depicted in Figure 2.2 and were shown to support counting both in children (Alibali and DiRusso, 1999) and adults (Cappuccio et al., 2013)¹.

Emblems are typical pictograms like a *stop* sign when trespassing is prohibited. They can be understood without additional verbal input but also they heavily depend on the cultural background. The *OK* sign, i.e. index finger and thumb forming an “O” (see Figure 2.3) is well understood in the North American society that something is fine or finished but can cause negative confusion in the South American countries, where this gesture has a very rude connotation.

Within the set of emblems, also *pantomimes* are used to explain actions, most prominent the subset of tool-use. Examples are peeling a banana or to show how to use a hammer. *Pantomimes* are a significant part of object-related actions and have been investigated in neuroscience (Bartolo et al., 2007; Kroliczak et al., 2007; Osiurak et al., 2012) where they are also sometimes termed as transitive actions,

¹In Cappuccio et al. (2013) the authors declared pointing gestures to be ‘instrumental’

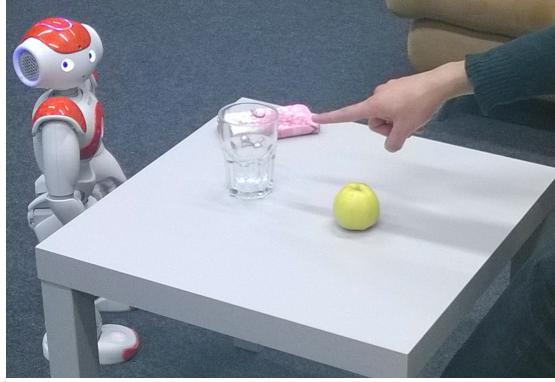


Figure 2.2: Pointing gesture to different objects in a scenario with the humanoid NAO platform.



Figure 2.3: An emblem with different meanings including ok, zero, and money or coins. In some countries, this gesture is regarded as an insult. (Image source: Pixabay)

and in computer vision, respectively, cognitive robotics, to extract significant object properties (affordances, Gibson (1979)) for reaching and grasping simulations (Goodale, 2011; Ugur et al., 2012).

Sign language provides a visual communication channel for deaf people and replaces speech by specific sequences of hand signs or finger spelling. In addition, so-called *visemes* display important features from lipreading and support the semantics of *sign language*. As an example, consider a person (pointing to oneself = “I”), who shows an action (open flat left hand and bend the right hand towards to = “to buy”). The meaning cannot be inferred from the hand gestures only, but also from mouth movement (the mouth forming an “o” as an expression for “what?”) and from other facial expressions. The latter is of special importance for a wider range of expressions, i.e. realizing also irony, sarcasm or metaphors which, while speaking, we would emphasize with special intonations, prosody or word concepts.

For an application in user interfaces, a model has to capture also the transitions between either different finger configurations (finger spelling) or gestures depicting words. This leads to additional challenges for an appropriate gesture

Table 2.1: Overview of different gesture types

| Gesture Type | Example | Style | Extra Modalities |
|--------------------------------|---------------------------------|--------------------|--|
| Gesticulation | (Rhythmic) hand movements | Dynamic | None |
| Iconic Gestures (Co-speech) | “The dog was so tall” | Dynamic | Speech Body posture Head information Facial expressions |
| Deictic Gestures | “Go to the right” | Static, Dynamic | Spatial reference |
| Pantomime | “Grasp the cup” | Static, Dynamic | None |
| Emblems | “OK” (Thumbs up) | Static, Dynamic | Facial expressions |
| Sign Language | Fingers and hands express words | Static, Dynamic | Body posture Head information Facial expressions |

representations and sequence models, which we describe in the sections below.

2.2 Gestures Involved in Cognitive Tasks

Developmental psychology investigates learning processes and the acquisition of cognitive skills in infants. That gestures are more than movements but help balancing cognitive load was pointed out by Goldin-Meadow and Wagner (2005). Children start counting with their fingers when asked how old they are or point to objects while counting them (Alibali and DiRusso, 1999; Berteletti and Booth, 2015). Also, they use their fingers to perform simple arithmetics like addition and it was actually found that performing multiplication and subtraction activated brain areas responsible for finger representations in the parietal cortex (Andres et al., 2012).

What distinguish gestures from actions? Recent studies showed that gestures do play a special role in communication settings by contrasting gestures from object-related actions. Although actions, like lifting a glass or cutting bread with

a knife, contain more visual information, researchers reported less confusion by participants when presenting incongruency between speech and action than showing an incongruent gesture together with a phrase (Kelly et al., 2010). The work concluded that gestures have a tighter link to language than simply being a companion. It can be argued that during evolution gestures were always involved in communication, whereas actions have a manipulative rather than communicative character and are associated with objects more than to speech or language.

2.3 System Development Stages

Careful experiment design and specification analysis for the implementation of a gesture recognition system are not only important from a software engineering point of view but also to facilitate subsequent steps involved in the chain from recording gestures to their recognition, and allow experimental reproducibility. A summary of crucial steps in system design is displayed in Figure 2.4.

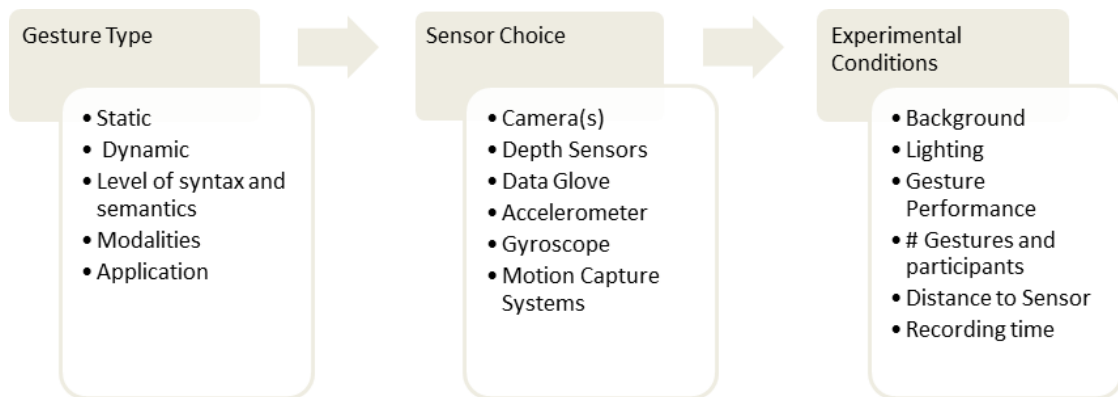


Figure 2.4: Processing pipeline for design decisions involved in the development of a gesture recognition system.

First, the gesture vocabulary needs to be defined. The set of gestures and their use in an application highly influences the subsequent choices on hardware and algorithms, as different gesture types consequently involve different design decisions.

2.3.1 Gesture Recordings

The different gesture types assign either specific finger, hand, or arm postures, which are usually subsumed under the term static gestures. They are primarily recorded with one camera because the focus is on the representation of hand shapes

and their corresponding finger configuration. In the case of fingerspelling as it occurs in sign language, also the order between different shapes matters. Posture databases thus typically contain a large number of images recorded with varying background and distinct camera angles. From these images, the challenge in the recognition or discrimination of postures comprises the development of hand shape descriptors based on the hand contour and computation of geometrical relations between finger and the center of the hand. We outline the techniques employed for static gesture recognition in section 2.3.2.

For dynamic gestures, the computation of a special finger configuration is not necessarily essential to the final recognition part. Depending on the selected set of gestures, constraints of the sensor choice can simplify recordings and the experimental settings. For instance, it may be sufficient to record hand- and arm gestures with a single camera, and, if the background should not be considered, depth images may facilitate the preprocessing even further. Inertial measurement units enable to use gestures for the control of smartphones, which does not need any lighting or background considerations at all. On the other hand, modeling finger configurations for sign language or grasping actions may need more sophisticated devices, e.g. a data glove. A multi-camera setup can be used to test for multiple perspectives or different subjects in the scene. An example is the exhaustive MOCAP database², comprising recorded actions from subjects wearing light markers on their body.

A plethora of hardware for gesture recordings exists, so we constrained our selection to the devices available which are most distinct in their recording ability and application to gesture types. Among the different devices, cameras provide the most natural interface for gesture performance. Figure 2.5 shows robots equipped with their cameras for vision-based gesture recognition, which can either be used on its own (e.g. a webcam as a robot camera or a Kinect (left) or a fish-eye lens (right)) or in an HRI setting.

For the NAO robot, the cameras are configured in a vertical fashion, which does not allow approximating a depth image from two horizontally arranged cameras. Applications in vision-based gesture recognition have benefitted from the release of cheap depth sensors in corresponding devices like the Microsoft Kinect[®] over the last 5 years. The availability of depth images in conjunction with publicly available software (notably the OpenNI framework) facilitated typical computer vision tasks like background subtraction and object detection in scenes.

Besides employing cameras, data gloves are preferably used when the hand

²e.g. MOCAP database <http://mocap.cs.cmu.edu/>

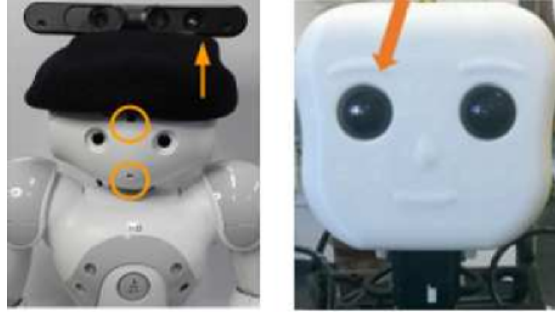


Figure 2.5: Devices for vision-based gesture recognition. Left: The NAO humanoid robot (Aldebaran company) and the Nimbro-OP (University of Bonn) equipped with cameras. The vertical arrangement of the two cameras on the NAO robot does not allow stereoscopic image capture because their visual fields do not overlap. (Photos from KT lab)

trajectory and a certain finger configuration play an important role (Figure 2.6d³). Although such devices can be expensive and hinder a natural gesture performance due to cables and calibration issues, a valid application scenario is to investigate the reaching trajectories and the hand preshaping. For pantomime gestures, this device can help in tracking the different hand configurations for object-related grasping actions. A lighter version was presented by Fujitsu⁴ for the hand tracking (Figure 2.6c).

In the past, a trend emerged with the publication of the Wii[®] remote controller allowing access to gyroscope and acceleration data when gestures were performed (Schlömer et al., 2008). In general, this kind of data is available from all devices equipped with an inertial measurement unit (IMU). Recent applications focus also on modern smartphones and use their IMUs to control the mobile device with gestures.

2.3.2 Techniques for Gesture Representations

In the previous section, we presented a pipeline with the first important steps involved in the process of gesture recognition. After recording the gesture data, the next question is which features best represent the gestures. In this section, we outline the preprocessing and feature extraction methods for the representation of gestures. In the following, we restrict the descriptions of techniques to hand- and arm gestures to keep in line with the focus of the thesis.

³<http://www.dg-tech.it/vhand3/>

⁴www.fujitsu.com/global/about/resources/news/press-releases/2014/0218-01.html

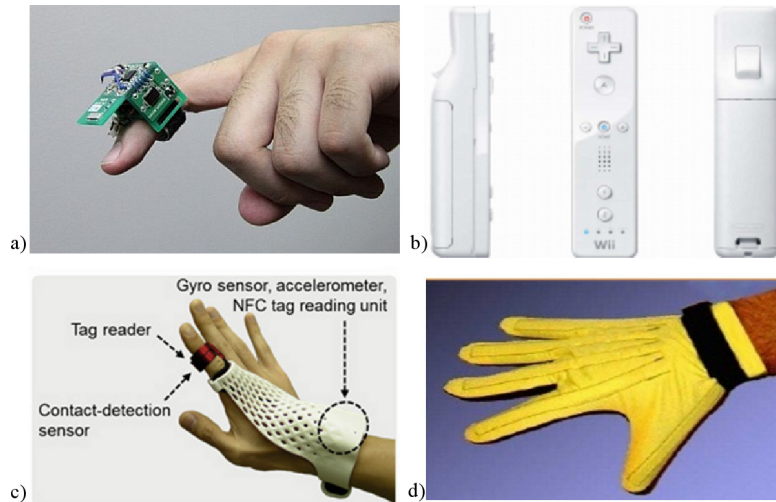


Figure 2.6: Gesture tracking devices. a) Hardware for specific finger movements (Jing et al., 2012). b) Nintendo Wii[®] controller, which uses inertial measurements to track dynamic arm movements (Schlömer et al., 2008) c) A glove for hand tracking from Fujitsu company in a slim version inspired by traditional data gloves. d) DG5-V Data Glove. Both devices can capture both finger- and hand movements but can be expensive and need to be newly calibrated when users change.

The first division of gestures concerns whether to deal with static gestures or dynamic gestures, as this has an impact on the representation. Static gestures or postures need descriptive features for the hand- and especially finger configuration. Figure 2.7 shows examples from the Triesch database (Triesch and von der Malsburg, 1996) for ten gestures and with varying backgrounds. The homogeneous background (pixel values either 0 or 255) in the first two rows enable a rather easy background subtraction. In contrast, the structured background scene displayed in the last row is a challenging task because the postures must not be confused with the background of the image. Some databases⁵ also provide sequences of postures and can be assumed as dynamic to a certain degree, but still, the processing and recognition of these sequences differ from gestures exhibiting dynamic motion profiles.

Techniques applied for the representations preferably provide also image invariance (usually affine transformations like translation or rotation) and are robust against occlusions or patterns in the background. Figure 2.7 shows examples of postures in front of different backgrounds.

In contrast, dynamic gestures describe varying spatiotemporal patterns and are

⁵e.g. Sebastien Marcel Dynamic Hand Postures <http://www.idiap.ch/resource/gestures/>

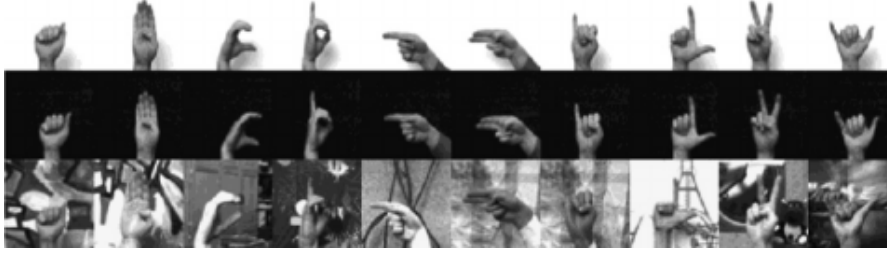


Figure 2.7: Examples of 10 postures on 3 backgrounds from Triesch and von der Malsburg (1996).

therefore usually represented by their motion profile. After identifying the hand or arm in the scene, the movement will be tracked over a number of frames, which denote the time component. Depending on the desired level of representation, the computation of the hand center points, movement velocity, motion estimation or the extraction of spatial-temporal points of interest are the most common choices for feature extraction. When gestures are supposed to be recognized in a continuous video stream, an additional step is to also model the *gesture spotting*, i.e. onset and end of a gesture performance. Having established a gesture scenario, the next aim is to extract meaningful features from the postures and movements, which are descriptive enough to generalize across different subjects. In the area of gesture recognition, two main directions are known: first, feature extraction techniques derived directly from the images or videos are summarized as *appearance-based* approaches. Second, models using geometrical figures for the representation of the hand and fingers are considered as *model-based* approaches.

Appearance-Based Approaches

Feature extraction on the image pixel level is usually performed on downsampled images to reduce the computational complexity. Further, images or videos captured by a camera in RGB-format are transformed into another color space (e.g. HSV, YCbCr) or reduced to grayscale values to facilitate segmenting visual cues. Another subsequent step is the noise reduction to erase high frequencies from images. Depending on a chosen pixel neighborhood (4- or 8- pixel neighborhood) the segmentation of an object (for gestures: arm, hands, and optionally the head or face) results in a binary image coding with **1** representing the area of a segmented object and **0** the area of the excluded parts.

After segmenting the regions of interest, the next step is then to extract significant features describing the object(s). A prominent feature computation technique usually applied to image reconstruction is to compute statistical properties of a

pixel distribution, called *image moments*. The advantage of those image descriptors is that the applied image statistics are invariant to affine object transformations like scale or translation invariance. While the calculations of the known Hu moments (Hu, 1962) are nowadays part of common image processing libraries⁶ and thus ready to use, the less prominent orthogonal moment calculations based on Zernike polynomials was shown to be superior in performance for e.g. sign language (Otiniano-Rodriguez et al., 2012). Zernike moments can also be used as reliable shape descriptors and for the retrieval of 3D objects (Novotni and Klein, 2004) and were extended to the temporal domain first introduced for gait classification by Shutler and Nixon (2006). Hu moments are calculated using Cartesian coordinates while orthogonal moments use the polar coordinates, which may explain their benefit over Hu moments, which in turn suffer from linear dependencies in the calculations. However, the technique of moment calculation, especially when considering the factorials for the Zernike moment computation, puts constraints on real-time performance. In recent years, using deep neural networks became a more popular way of feature extraction with the same invariance properties on objects in images, but with the advantage of processing the images directly and implementing a learning of the filter sizes dependent on the image input, which is further used for classification.

Another method to extract object shapes is to use descriptors (a comparison of image moments and Fourier descriptors for posture classification was done by Conseil et al. (2007)). As an example, consider the hand in an image. The Fourier analysis relies on the determination of frequencies in the image by a superposition of weighted sine and cosine functions, where the basic signal is the fundamental harmonic followed by signals with increasing frequencies. To describe a shape is then the composition of different frequency image content achieved by the mentioned weights or coefficients. Intuitively, the more details an image contains (like edges or contours) the higher the frequency portions to represent these. A disadvantage of the method is to find the right number of coefficients which sufficiently represent an object shape, e.g. the fingers of a hand. An approach extending the idea is the usage of wavelet descriptors. While the underlying idea of frequency decomposition remains the same, a wavelet allows flexible frequency filtering adding a time localization component.

The computation of the contour or silhouette is another prominent approach in *appearance-based* methods. For instance, the Elastic Graph Matching algorithm (EGM) relies on template models built on a graph structure and was used for

⁶e.g. OpenCV libraries for C++ and Python

object, face and posture recognition (Triesch and von der Malsburg, 2001). The edges in the graph model assign a distance between the nodes, which in turn are salient points in an image. They are described by a collection of so-called *jets*, which are composed of Gabor wavelets with varying frequencies and orientations. In general, a Gabor function was found to model the neural responses to changing orientations of visual stimuli in the V1 area of the cat's striate cortex, which is why the EGM has the connotation of being bio-inspired. Gabor filters provide invariant image descriptors (Kamarainen et al., 2006) but can be substituted with wavelets due to their improved time-frequency resolution. The EGM algorithm then tests a new image with a model graph in a corresponding database. Although the EGM approach does not need additional background subtraction, the creation and labeling of the initial graph models are time-consuming, and thus the database is difficult to extend with new posture templates.

Another example of contour extraction is depicted in Figure 2.8 (middle). Extracting the hand and setting the background to a unique pixel value (here black) simplifies the computation of the fingers, as they serve as the image boundaries. The reference point for the detection of the tip of the fingers is the center of mass, depicted as a big red circle. As the method provides a flexible description on the hand- and finger shape it can be used for both postures and dynamic gestures (cf. first example showing the *OK* posture vs. the movement of the index finger displayed in the third image column).

When considering dynamic gestures, i.e. hand and arm movements which are interpreted without the necessity of special finger configuration, the computation of the optical flow is a common approach. Instead of using the $\{x,y\}$ -coordinates of the image, the optical flow algorithms compute the motion velocity $\{u,v\}$ for both image directions. The underlying idea is that motion can be computed as the rate of luminance changes. This yields a flow vector field where the vector magnitude corresponds to the strength of motion. In Figure 2.8 (left) an example of optical flow computation using the Lucas-Kanade algorithm (Lucas and Kanade, 1981) is shown. The blue arrows depict the motion vector which is mainly distributed on the moving arm. The advantage of the optical flow computation is that no special segmentation procedure is needed to be applied beforehand, as the information about the object of interest is coded in the length and magnitude of the flow vectors. Most commonly known implementations of optical flow are the Lucas-Kanade algorithm which is fast but only a local motion estimator, and the Horn-Schunck method (Horn and Schunck, 1981) which allows estimation of the global motion for more than two frames, but which is also slow. For a comparison of both

methods see Bruhn et al. (2005).

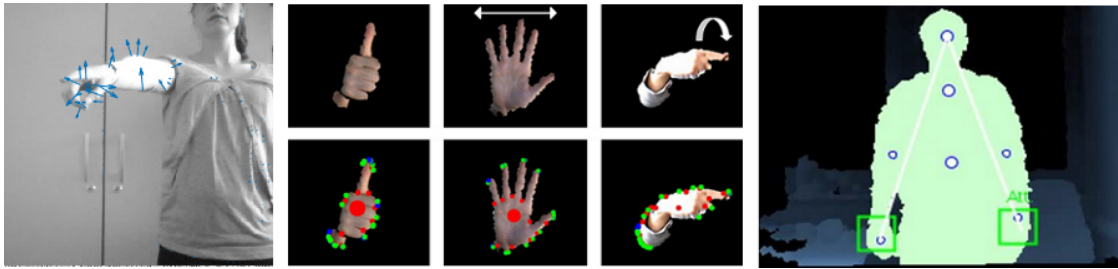


Figure 2.8: Examples of hand gesture representations. Left: A pointing gesture with optical flow vectors using the Lucas-Kanade algorithm (blue arrows, from own recordings). Middle: Hand extraction from a usual web camera for a representation of the finger configurations, displayed by the red and green points. Right: Image from a depth sensor with significant body points (Parisi et al., 2014).

Related to the depth sensors, Figure 2.8 shows also an example of a segmented human and the computation of body joints from which important features for gesture representations can be derived. In a gesture scenario for both command and iconic gestures presented by Parisi et al. (2014), a feature vector was derived from the motion in the three dimensions and additionally the angle and distance between head and hand. One benefit in the scenario with that recording was that no special preprocessing was needed to track both hands, as the motion was the most salient feature in the scene. This approach enabled to also refine the gesture definition for e.g. a symmetric gesture when one hand mirrors the motion of the other hand.

Model-Based Approaches

Model-based approaches aim at creating 3D representations of the hand for free articulation or for reliable hand tracking in a 3D scene. The benefit of such a model is to provide a flexible, markerless description to overcome the hand calibration issues and cumbersome usage of data gloves (as described in the previous section). One challenge for the derivation of these models is the complexity inherent in the hand- and finger configurations because a hand is a nonrigid object. Multiple joints and orientations involved in tracking a gesture span a large space of degrees of freedom (DoF), but several anatomical constraints can facilitate the gesture representation in that space. For instance, finger joints are constrained in their rotations by their bone structure, and also finger interactions are limited by the knuckles.

In Figure 2.9 some examples of a hand model and corresponding poses are shown based on the work presented by Hamester et al. (2013). The model is built on an approach developed by Iason Oikonomidis and Argyros (2011) but integrated findings on the finger movement interrelations to improve the model. Subsequently, a Principal Component Analysis (PCA) was applied to reduce dimensionality in the DoF space. Both models employed Particle Swarm Optimization (PSO) for hand tracking. In a direct comparison to Iason Oikonomidis and Argyros (2011), the model showed a faster convergence of the average error.

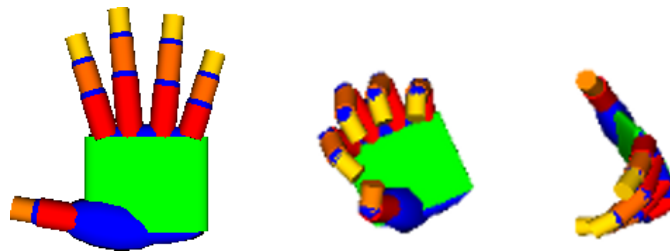


Figure 2.9: Examples of a hand model. Hand movements of a subject are tracked and rendered to display different hand poses (Hamester et al., 2013).

Another approach is to compute a mesh from the hand shape. In general, the basic idea is to triangulate over objects resulting in a mesh structure which can be parametrized to provide a flexible description of the object, specifically the hand (Vidal et al., 2012). This makes it an attractive tool for hand tracking serving as deformable hand template but the computations to derive a reasonable model involve time-critical steps. A hybrid approach taking into account a deep learning architecture processing depth images in a frame-by-frame manner with synthetically rendered hand poses is presented by Neverova et al. (2014).

2.4 Modelling and Learning Gestures

In this section, we outline graphical models, which have become standard over decades in modeling and learning gestures. As they are based on (joint) probability distributions, another direction is learning gestures using artificial neural networks. In this section, we review recent approaches in gesture recognition for both approaches including hybrid models.

2.4.1 Probabilistic Graphical Models

Probabilistic graph models (PGM) are reliable sequence models with application areas in computational linguistics and computer vision. Learning in PGMs is performed in a supervised manner, i.e. the output is known and according to a loss function, training optimizes the model parameters as to decrease the error. A notable algorithm is the Expectation-Maximization (EM) in which the underlying graph structure allows inference in appropriate computation time. This is the case for rather simple graph structures like a chain or tree. For more complex graphs inference can be approximated by e.g. Markov Chain Monte Carlo (MCMC). A standard graphical model is the of Hidden Markov Model (HMM). An HMM is a directed probabilistic graph and belongs to the class of generative models (see Figure 2.10). The term *hidden* comes from the fact that only the sequence output is observable, but not the state sequence producing the result. Basically, an HMM is defined as $\lambda = \{A, B, \pi\}$, where A denotes the state transition probability matrix for states S , B is the emission (or output) probability matrix for observations O , and π is the initial probability distribution that the HMM starts in a particular state. Context information is provided by the Markov property, i.e. the past information influences the current state depending on the order. Due to the combinatorial complexity of possible states and emission sequences, optimization of an HMM is realized by means of dynamic programming. Main objectives for proper use of an HMM are identified:

- Evaluation: $P(O|\lambda)$, the probability P for an observation sequence O given an HMM λ (forward algorithm)
- Decoding: Similar to the evaluation but the aim is to maximize the probability that a state sequence produced an observation sequence (Viterbi algorithm)
- Learning: Estimation of the model parameters of an HMM λ from training examples, i.e. supervised learning (Baum-Welch algorithm)

A thorough tutorial on learning HMMs with an application link to speech recognition is provided in Rabiner (1989). One drawback of HMM is the discretization of states for sequence modeling, as producing a vectorized codebook results in information loss.

Another class of graph models comprises undirected probabilistic graph approaches. The general concept is built on Markov Random Fields (MRF), which

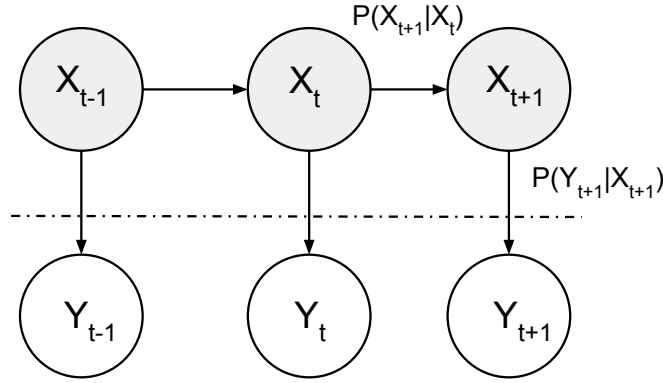


Figure 2.10: An HMM defined as $\{A, B, \pi\}$. X depicts the states and Y the emissions. Training an HMM aims to find a model which fits best the set of the hidden state sequences (state transition matrix A) with the most probable emission sequence (emission probability matrix B). The dashed lines show that only the emissions are available while the states and their transitions producing them is hidden. The initial start probability for a state is given by π . HMMs belong to the class of generative models and are directed acyclic graphs.

have been shown successful in image processing (Wang et al., 2013). The computational principles are borrowed from statistical physics which is reflected in the use of potentials in the models (Li, 2009) (see equation 2.2). Based on the MRF, Conditional Random Fields (CRF) and derivatives were developed and applied to object recognition (Quattoni et al., 2004; Wang and Ji, 2005; Zhong and Wang, 2006), activity and action recognition (Vail et al., 2007; Shimosaka et al., 2007), and further applied to parsing and labeling tasks in the language domain (Sha and Pereira, 2003; Cohn and Blunsom, 2005). Comprehensive introductions to the model are provided in Lafferty et al. (2001) and Wallach (2004).

One benefit of a CRF compared to HMM is the modeling of the whole sequence at once instead of computing the joint distribution (see Figure 2.11). Also, the structure of a CRF graph can be arbitrary, but a typical CRF has a tree structure or forms a linear chain.

Modeling the conditional probability of Y given the observations X , the distribution is defined as (Hammersley and Clifford (1971)):

$$P(Y|X) = \frac{1}{Z} \prod_n^N \phi(n, y_{n-1}, y_n, X) \quad (2.1)$$

As ϕ is a non-negative function, the potential is assumed to be exponential:

$$P(Y|X) = \frac{1}{Z} \prod_n^N \exp(\pi f(n, y_{n-1}, y_n, X)) \quad (2.2)$$

where Z is simply the sum of the entire product for normalization. Model parameters are optimized using maximum likelihood estimation (MLE), which is facilitated when the graph structure is simple⁷, e.g. a linear chain.

Because of their sequence modeling capability, probabilistic graphical models went through diverse extensions and are, until today, well-accepted models for action, activity, and gesture recognition. Variations on the HMM architecture include parametric (Wilson and Bobick, 1999) and parallel HMM (Vogler and Metaxas, 1999).

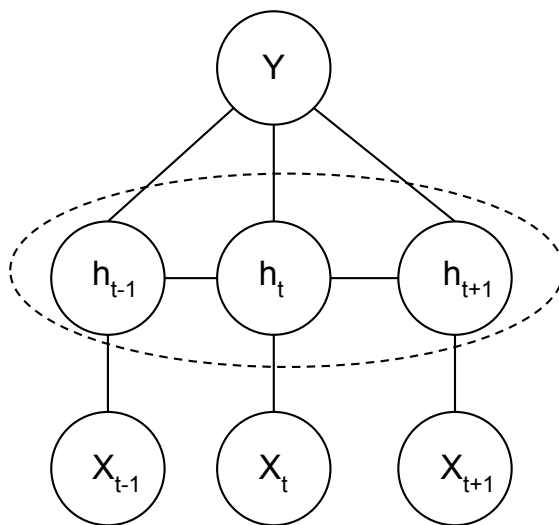


Figure 2.11: A undirected probabilistic graph model. The hidden Conditional Random Field contains an extra layer h serving as latent variables, which model specific parts of a sequence conditioned on the input X . Leaving the extra layer h provides the standard formulation for a CRF. In contrast to the HMM, Y assigns here a class label for the complete sequence. CRFs belong to the class of discriminative models.

One of the extensions on the CRF architecture called Hidden CRF (HCRF) was introduced in Quattoni et al. (2007). An extra layer L was constructed between the set of observations X and label set Y (see Figure 2.11). The latent variables in that layer allowed drawing inferences about the relation between image patches or relations between movements, which is an advantage over HMMs. As an application, the authors set up experiments on objects and gestures. After fitting the body shape of subjects from a 3D model the angles and joints resulting from performing different manipulative gestures with one arm were used as the feature set. On the set of arm gestures, the HCRF with multiclass classification showed

⁷functions of the form $\log \sum_i \exp x_i$ are concave and thus training converges towards a global minimum

superior performance expressed in accuracy compared to an HMM and a CRF. An additional window was also used to investigate the influence of contextual information, which led to an increase of performance for the HCRF to for a window size of 1. Interestingly, a drop was shown for the CRF and increasing window sizes, presumably due to overfitting.

A method based on the HCRF called latent discriminative CRF (LDCRF) was presented in Morency et al. (2007). In contrast to the HCRF, the model can learn labels between the variables in the hidden layer, which is a desired property for unsegmented data streams. Learning was achieved using belief propagation. The authors reported superior performance in the recognition of head gestures compared to HMM, CRF, and HCRF. Using this model, Song et al. (2012) presented a study on command gesture recognition for aircraft handling. Hence, a strict requirement of their approach was that users perform gestures in an unconstrained way and did not have to care special gesture start- or end phases. The authors used image streams from a stereo camera. After obtaining the depth images, two computational modules carried out the estimation of the body postures and four hand shapes. The obtained features were then merged for input into the LDCRF. The authors showed, that their additionally developed multilayer filtering method showed good performance on the NATOPS database.

A hybrid model exploiting the complementary features of an HMM and a CRF model was introduced for the task of recognizing ciphers between 0-9 (Elmezain et al., 2010). The focus was on modeling the gesture spotting, i.e. the start of a meaningful gesture sequence, thus random motion had to be considered in the model as well. An HMM was trained after obtaining 3D and color information. The subsequent inference task for the cipher classification was then realized with a CRF.

A hybrid approach combining neural computations with statistical models for sequence processing is outlined in Do and Artieres (2010). The authors selected a labeling task of an optical character recognition dataset and the TIMIT corpus (continuous speech). A deep network was used to enhance the input representation, as they are known to provide more high-level structures, while the inference task on the sequence was still obtained using a CRF structure. The architecture called NeuroCRF is built on a Restricted Boltzmann machine and trained in a semi-supervised fashion, i.e. supervised training with labeled data together with unlabeled data.

In summary, the application field of graphical models is highly diverse and the models outlined in this section are a reasonable choice for sequence tasks, including

gesture recognition. The main advantage of their usage is that the models allow tractable inference mechanisms on a sound computational foundation. However, from our perspective drawbacks of graphical models include scalability, for CRFS the derivation of the feature functions and possible intractability of complex models (i.e. beyond linear chains and trees), and no or only a little adaptivity to the underlying graph structure, which puts constraints on the modeling itself. As our work is motivated by the properties of brain-related information processing principles, we will now review some work being in our line of research using artificial neural networks.

2.4.2 Gesture Recognition with Artificial Neural Networks

In this section, we review gesture recognition approaches based on artificial neural networks (ANN). The difference to the learning of gestures with graphical models is that according to a specific gesture representation a network learns an input-output mapping with the desired target. This deviates from computing (joint) probability distributions and MLE but includes supervised learning with a teacher signal at the output or learning gestures in an unsupervised fashion, i.e. autonomously learning a specific input distribution. The concrete learning algorithms behind ANNs are presented in chapter 4.

Zhu and Sheng (2009) presented a hybrid hand gesture recognition system for five command gestures, *come*, *go fetch*, *go away*, *sit down*, and *stand up*, applicable in a Human-Robot interaction (HRI) scenario. The architecture comprised two main components: first to segment and discriminate between a concrete gesture (gesture spotting) and a non-gesture, and then to subsequently categorize the five defined gestures in an online fashion. For the task of gesture spotting and subsequent recognition, a feedforward neural network and a hierarchical HMM were implemented. The hand gestures were recorded using a wearable sensor which delivered the 3D axis acceleration and 3D angular information from finger movements, in total 6 features. The final feature vector serving as input to the network consisted of the mean and variance values from measured gesture sequences. The network was then trained using the backpropagation algorithm (see chapter 4) with labels provided by the experimenter. When a gesture segment was successfully identified (i.e. discriminated against a non-gesture), a subsequent hierarchical HMM was used for categorization. The hierarchy was introduced by defining five HMMs, i.e. one HMM per gesture at the lower level and an HMM with five states and corresponding observation symbols at the upper level. Additional preprocessing using a sliding window of 1 second length (corresponded to 20 data points) and subsequent

clustering was necessary to quantize the motion information as input to the low-level HMMs. The upper-level HMM was trained with Bayesian filtering for online classification. In the evaluation, the authors reported both accuracy of the neural network performance in the segmentation task and classification accuracy as a percentage between ground truth labels and labels provided from the HMMs. The accuracy from the individual HMM at the lower level showed a good performance for four gestures (between 0.7742% and 0.8929%), while the *sit-down* gesture performed worst (0.2581%). The performance significantly increased for the HMM and Bayesian filtering at the upper level (0.7419%), also for the *come here* gesture (from 0.8929% to 0.9286%). In contrast, the remaining gestures showed equal performance (*stand up*) gesture or only slight improvement. Unfortunately, the authors did not provide any suggestion or explanation for the noticeable difference in the gesture performance for the *sit-down* gesture, nor a reason why the deviation of accuracy between the low-level and upper-level HMM for the remaining gestures was rather low. This leaves the question to what extent the hierarchical HMM was an essential extension to standard HMM recognition procedures, or, put differently, what this specific system contributed more to other gesture recognition systems with comparable performance.

A neurally-inspired approach using Localist Attractor Networks (LAN) for the recognition of dynamic gestures was addressed in Yan et al. (2010). The authors focused on the variations in gesture performance across multiple subjects from which the gesture data is recorded. The reason is to enable a more intuitive interface to HRI. In fact, their system enabled users to define their own commands in their own manner. In total, 235 samples were collected. The recorded motion data was transformed into feature vectors for network input using Fourier-based computation for frequency decomposition of the different signals (see section 2.3.2). The LAN itself exploits two important aspects from psychology: the i) the prime and ii) the gang effect. The prime effect refers to the fact that network convergence into attractors is dependent on a number of visits, i.e. convergence into a specific attractor is faster when it has been visited more recently. The gang effect describes the strengthening of the basin of surrounding attractors. The proposed architecture was tested in a simulation with a wheeled robot guided by the gestures. First, the system worked in real-time, which is an important factor to provide a natural communication. Second, the authors reported a classification error of 99.15%. However, several aspects limit the freedom of unscripted gesture performance. The gesture data was collected via numerous body orientation sensors including arm tape and a wrist device for the motion capture. The additional,

very specific hardware is not available everywhere and for every user. The sensor choice is furthermore insensitive, as the performance of gestures only contained the movement of the arm and hand. This reveals the body sensors as an unnecessary system overhead.

The use of a Jordan network (Jordan, 1986) for the recognition of 10 predefined gestures was presented in Hikawa and Araga (2011). Various hand postures were detected in the first stage of their system and then assembled into sequences. The preprocessing was based on color detection in the images; binary quantization followed by two Discrete Fourier Transforms (DFT) to determine the horizontal and vertical spectrum of the image. The features derived from the computations were then further clustered using two approaches, namely a range check and a Self-Organizing Map (SOM) combined with Hebbian learning. In sum, 38 postures were defined. The Jordan network was then trained on 760 images to classify the correct gesture based on the composite indices of the beforehand clustered hand postures. The authors reported superior recognition performance for the hybrid SOM+Hebbian method with an average recognition rate of 96.2% for 20 frames sampled per gesture. A slight drop in the recognition was reported for 5- and 10 frames sampled per gesture (94.6% and 95.6%, respectively). The posture classification and the subsequent network training was carried out separately. However, the authors made no statements about the computational complexity of training and testing their system. Also, no experiments to test real-time recognition were carried out. Additionally, the system requires that users wear a red-colored glove to simplify background segmentation. Another constraint in the system is the usage of predefined postures, although the different spatial positions of hands can be captured by the preprocessing scheme.

Nagi et al. (2011) studied gesture recognition employing a Max-Pooling Convolutional Neural Network (MPCNN). Convolutional Neural Networks (CNN) resemble the processing of retinal stimuli in the visual cortex, where a cascade of local feature filtering and composition of high-level structures yields invariant image descriptors (see 2.3.2). The gestures were defined as numbers 1 to 6 to command a wheeled robot, yielding in total 6000 images. The image segmentation was simplified using an orange glove, where the hand was extracted after color transformation into the YCbCr space and applying a single Gaussian Model (SGM) to model the color distribution. The images were then fed into a with the ratio 60% for training and 40% for testing. Their approach yielded an error rate of 3.23% averaged over 100 training and test samples, which ranked among best compared to a system using feature extraction with Fast Fourier Transform (FFT) or using Hu moments

as input to a support vector machine classifier (25.32%, respectively 20.34%).

An unsupervised gesture recognition system based on a two-level SOM and usage of an ASUS Xtion recording device was developed by Parisi et al. (2014). The dataset used in the experiments was inspired by the ChaLearn 2013 data concerning the recognition of Italian co-speech (iconic) gestures. Also, command gestures were defined, as the main aim of the work was to present a flexible, yet robust neural architecture with minimal preprocessing for hand gestures using both only one hand or both. Additional audio signals were not used, but the authors made use of the fact that Italian co-speech gestures are usually understood taking into account head movements. The input to the SOM included the $\{x, y, z\}$ coordinates but also the head angle and hand-head distance calculated by the Euclidean metric. Gesture sequences were fed into the SOM architecture, where outliers were detected and removed in the first SOM layer. Subsequently, the trajectories, defined as the trace of best matching units (BMU), were matched against the trajectory representations of a pre-trained SOM in the second level. Although the system does not explicitly model time, the average recognition accuracy obtained from the model was 89% for the command dataset and 90% for the Italian dataset. An advantage of this approach is that the gesture sequences were recognized in real-time, which qualifies such a system for HCI or HRI scenarios.

Reservoir Computing Approaches

Only a few models exist for gesture recognition employing Reservoir Computing, which is the focus of the present thesis. One study using this learning paradigm is presented in Weber et al. (2008). Five hand gestures were defined, namely (anti-)clockwise movement, up and down, and a parametrized Lissajous curve describing the shape of the cipher 8 (therefore referred to as “figure-8”). Gestures were recorded with a regular web camera and the hand centroids extracted using the Camshift algorithm of the OpenCV library. The input was normalized and sequences fed with equal lengths into an Echo State Network. For training, the authors applied the recursive least squares (RLS) algorithm. Parameters were fixed in advance, where an orthonormal reservoir was used yielding a matrix with maximum spectrum 0.8. Test recognition rates showed good performance for the horizontal- and anticlockwise movements (91% and 96%, respectively). The vertical movement was recognized with 85% and the clockwise with 87%. The worst results both for training and testing were achieved for the “figure-8” pattern, yielding only 65% recognition rate. Although the results were promising, we could find no evidence following up on this approach. It would have been interesting to scale

up the approach to more useful gestures, specifically command gestures. Also, more investigations on the parameters would have probably revealed some interesting properties of the network behavior and accordingly the learning process. It is for example still an open question in the research community whether the conditions of a maximal spectral radius below unity needs to be obeyed. We will explain the mathematical underpinnings of Echo State Networks in the chapters 4 and 5. However, before going into the computational details, we want to describe the underlying information processing mechanisms in the brain in the next chapter.

2.5 Chapter Summary

In this chapter, we outlined the different gesture types, their representations and reviewed probabilistic graph models, which were state-of-the-art sequence models in the last decades. From the gesture recognition literature, we selected those which used command gestures and used neurally-inspired techniques. In the present thesis we focus on the learning paradigm of *Reservoir Computing*, however, research on this area in connection with gesture recognition is rather sparse. We will outline the neural processes in the human brain, the cortical organization, and the neurophysiological model for sequence processing, which substantially underpins the *Reservoir Computing* approach, in the next chapter.

Chapter 3

Neural Processes and the Emergence of Computational Models

The information processing capabilities of the human brain offered an incentive to translate neural phenomena into mathematical models of artificial neurons and neural networks. This chapter explains the neurobiological principles behind neuronal information processes in the brain and gives examples of their modeling.

We start with the broad description of the electrochemical processing on the neuronal level. With this basis, we explain further building blocks of brain computations including plasticity mechanisms shaping the cortical organization. We outline the sequencing model from Dominey (1995) for action selection, which introduces the special role of the prefrontal cortex (PFC) as a reservoir of random, recurrently-connected neurons. The *Reservoir Computing* paradigm substantially builds on this idea, however, the neuroscientific and the computer science developments emerged concurrently.

3.1 From Neurons to Cortical Structures

In this section, we describe the basic neural information processing, the plasticity mechanisms structuring the brain, and give an overview of the main functions of the four cortices.

3.1.1 Computations on the Neuronal Level

Computations in the neural tissue are fast and efficient in energy consumption. This is due to a specialized structure of local functional areas but also due to a specific global connectivity pattern. The neural information processing can be described from the microscopic to the macroscopic view. The microscopic description dates back to the findings of Golgi and Cajal, who presented histological evidence of anatomical features of neurons and postulated the electrochemical propagation of impulses involving synapses.

The signal transmission from neuron to neuron, and thus their communication is realized as follows: a neuron consists of arborescent fibers called dendrites, which receive action potentials (i.e. electrical impulses, AP) from their neighboring neurons. For the excitation of a neuron, it is necessary that the incoming current is high enough and exceeds a certain threshold. If this is the case, the considered neuron fires, otherwise it remains silent. This phenomenon is also known as the “all-or-nothing”. For a neuron to fire, the membrane becomes semipermeable. This enables extracellular positive ions in the cell tissue to enter the inner, negatively charged, cell. The neurons’ state then changes from hyperpolarization to a depolarization, i.e. the voltage increases to approximately +20mV, opposed to the approximately -70 to -90 mV resting state of the neuron. An AP is generated at the neurons’ soma and propagated along the axon, which has a special structure. While the myelin sheaths act as an electrical isolation, the nodes of Ranvier in between allow the regeneration of the AP. The electrical impulse thus “jumps” from node to node, yielding a fast transmission along the nerve fiber. This process is also called saltatory conduction. The transmission ends at the axon terminals, called synapses. They can either be a chemical or an electrical synapse. In the first case, the synapse comprises vesicles with neurotransmitters, which are released into the post-synaptic gap. An example of a neurotransmitter is dopamine (excitatory). Electrical current is passed via gap junctions between the membrane of synapses. Figure 3.1 depicts this process.

After the emission of an AP, a time interval follows, where the ion concentration is reversed back again. During this *absolute refractory period*, the neuron is incapable transmitting newly arriving signals, and this *relative refractory period* describes the time interval to go back to resting state.

A model to describe the neural activity and the underlying mechanisms of action potential propagation were introduced by Hodgkin and Huxley (1952), who described the neural dynamics as a set of differential equations. Due to the models’ complexity, which prohibited the analytical solution of the nonlinear dynamics,

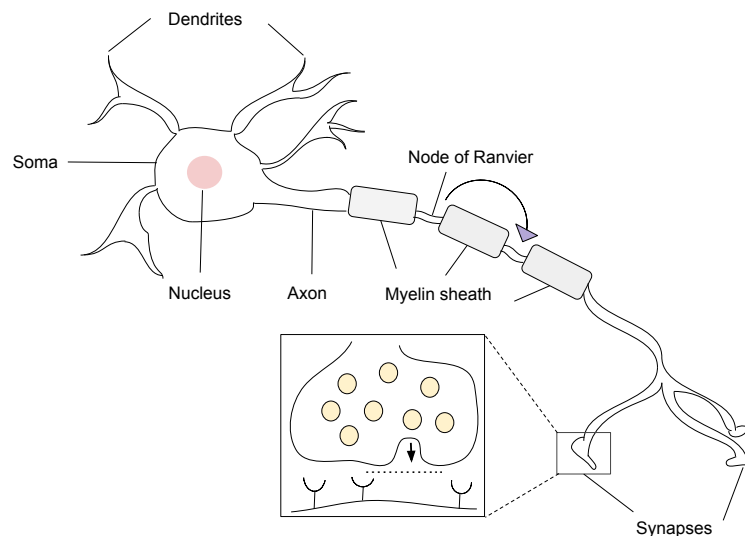


Figure 3.1: Sketch of a neuron and the main parts responsible for the propagation of action potentials. The embedded figure displays a chemical synapse.

more simplified spiking neuron models were introduced (FitzHugh, 1955; Izhikevich, 2007).

3.1.2 Neural Plasticity Shapes the Brain

A property of the human brain is its noticeable organization. In the following, we describe the underlying adaptive mechanisms responsible for the brain plasticity. As described in the previous section, the arborization of each neuron yields a local synapse neighborhood. This led to the research question whether the strength of the connectivity influences neural responses. One contribution approaching this topic was Donald Hebb’s postulation (Hebb, 1949), which is often summarized as: “what fires together wires together”. In essence, the more the synapses generate action potentials concurrently, the more this process strengthens their connections, respectively, synapses. Vice versa, the synapse weight is weakened if the two neurons are activated distinctively. This claim was the foundation into research of *synaptic plasticity* (SP) and cell assemblies, constituting an unsupervised organization principle of neurons shaping the cortical areas. In computational neuroscience and cognate disciplines, this postulate formed the basis for associative learning known as *Hebbian learning*. Figure 3.2 sketches two neurons x_i and x_j

with a synapse weight ω_{ij} , which is intensified in the case of concurrent neural activation. Computationally the basic model is:

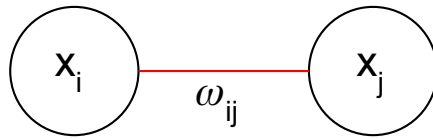


Figure 3.2: The Hebbian learning rule states, that the synapse connection ω_{ij} between two neurons x_i and x_j strengthens when both neurons are activated simultaneously. The connection weakens if the two neurons get activated separately.

$$\Delta\omega_{ij} = \eta x_i x_j \quad (3.1)$$

where the Δ quantifies the change of strengthening and η is a learning rate, which determines the speed of weight update. Grossberg (1968) introduces a weight decay term $\gamma \in [0; 1]$ to prevent unbounded growth of the weights:

$$\Delta\omega_{ij} = \eta x_i x_j - \gamma \omega_{ij} \quad (3.2)$$

The SP has also an effect on the excitability of a neuron itself, specifically on its electrical properties. This *intrinsic plasticity* (IP) describes the neurons' ability to generate action potentials, where the corresponding output distribution varies (Turrigiano et al., 1994). An information-theoretic approach to model IP is, for example, presented in Triesch (2007). Until today, the relevance of this mechanism is not yet fully understood, but experimentally this type of plasticity was shown to be part of homeostatic regulation and involved in learning and creation of memory capabilities (Zhang and Linden, 2003).

3.1.3 Functional Areas of the Cortex

The anatomy of the human brain comprises three components: the brain stem with the medulla oblongata, the cerebellum and the cerebrum (cf. Figure 3.3). The brain stem and cerebellum are mainly (but not exclusively) responsible for the afferent and efferent processing of motor signals, while the cerebrum processes numerous perceptual stimuli from the environment to enable human beings performing cognitive tasks (decision-making, reasoning, language acquisition, etc). It is separated into a left and right cortex with a functional division into the occipital, temporal, parietal and frontal brain areas. The information flow is from posterior to anterior, involving forward, lateral and recurrent connections. The

display of complex behavior results usually from an orchestrated biological neural network, for example, the default network (theory of mind) (Fair et al., 2008), the limbic network (generation of affective states) (Thomas Yeo et al., 2011) or the salience network (body-related attention and behavior) (Seeley et al., 2007). In the following, we highlight some of the significant functions of the four cortices.

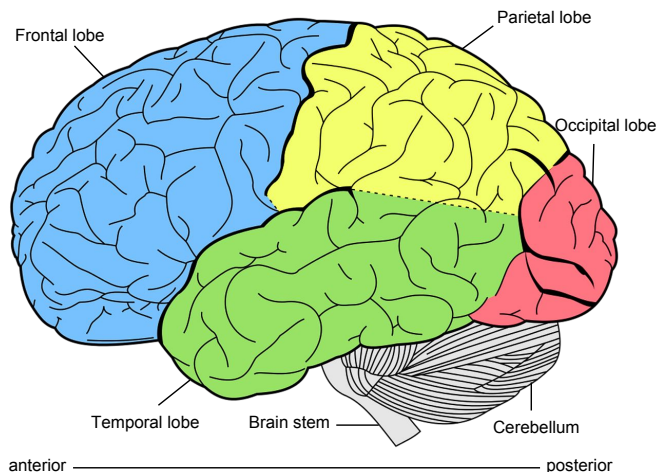


Figure 3.3: Sketch of the human brain. (Image modified from Pixabay)

The occipital brain area forms the visual cortex. Retinal stimuli from the surrounding are processed via a layered neuronal architecture. Neurons in the first layer, area V1, compute rather simple image structures like edges and pass this information to successive layers, where neurons get more and more tuned to a particular image content, i.e. code for specific shapes and colors. This hierarchical process provides a robust image description because, once identified, similar or equal images are recognized even under affine transformations and with occlusions. A prominent characteristic of the visual cortex is the *ventral-dorsal* dichotomy (Ungerleider and Mishkin, 1982), commonly referred to as the 'what' and 'where' path. As the names indicate, the ventral ('what') stream encodes perceptual properties, while the dorsal stream ('where') computes the spatial location of objects in the scene and is hypothesized to play a vital role in coding for actions, e.g. in affordance processing. Affordances trigger a specific motor behavior, e.g. grasping a cup, when the handle elicits properties which make it graspable. Information processed in the ventral stream is passed to the temporal cortex, while the dorsal stream merges into the parietal cortex. Areas relevant for gesture recognition are V2 (BA 18) and V3 (BA 19) for discrimination of finger gestures and detecting motion patterns.

Located in the lower part of the brain is the temporal lobe, mainly involved in memory-related tasks and audio-, respectively, language processing. Prominent structures are the hippocampus, which is activated in navigation tasks and processes emotions modulated by the amygdala, the primary auditory cortex (BA 41/BA 42) for sound perception like a pitch but also the auditory working memory and Wernicke's area (BA 22) located in the left hemisphere. The latter plays a crucial role in language processing, specifically in sentence generation and semantic processing. It was also found that it is involved in deductive reasoning. Another interesting area is the inferior temporal gyrus (ITG, BA 20), which was found to be active in metaphor comprehension (left hemisphere) and object integration into a more complex scene (right hemisphere). Finally, the fusiform gyrus (BA 37) has become popular when the neuronal activity was recorded specifically tuned to the recognition of faces. It has become evident that there are different areas separating the processing of faces from object recognition or object detection. Also, episodic memory and sign language are encoded here. Finally, other cognitive processes are distributed along the temporal cortex, for example, inferential reasoning, irony understanding (right hemisphere), lexicosemantic processing (left hemisphere) and the experience of emotional states in the temporopolar area.

The parietal cortex is mainly involved in sensorimotor integration. Information from the visual cortex about the spatial location of objects and other visual cues are transformed into motor responses. It is further subdivided in a superior (SPL, BA 5/BA 7) and an inferior part (IPL). The SPL is mainly involved in tool-use gestures (left hemisphere), visuomotor attention and visuospatial processing, and provides working memory for verbal, motor, emotional and auditory stimuli. More high-level computations comprise emotion and self-reflection for decision making, temporal context recognition, and chaotic pattern processing. The IPL (BA 39/BA 40) shares some area with the Wernicke area and is active in processing action sequences and executive control of behavior (angular gyrus, BA 39) as well as gesture imitation, motor planning, visually-guided grasping and integration of proprioceptive information (supramarginal gyrus, BA 40). More cognitive functions encoded here are deductive reasoning, semantic processing, and verbal creativity. The somatosensory cortex (SI) covers the postcentral gyrus, which is made up of three main regions (BA 1, BA 2, BA 3). These areas process elementary sensory stimuli like pain, touch, and vibrations, and are active for finger proprioception and voluntary hand movements.

Particularly interesting is the posterior parietal cortex (PPC) and the somatosensory cortex. Neurons in primate homologue (usually macaque mulatta,

homolog brain areas F2-F7) showed distinct firing patterns for visual and audio stimuli simulating an action (Rizzolatti et al., 1996; Gallese et al., 1996; Kohler et al., 2002; Hamzei et al., 2003; Rizzolatti and Craighero, 2004). As the firing characteristics were especially shown for imitation tasks, i.e. a primate observing a demonstrator performing an object-related action like grasping a cup, these neurons were coined *mirror neurons*, located both in the temporal and frontal lobe of the primate brain.

The control and execution of motor actions are processed in the frontal lobe. The primary motor cortex (BA 4) and the premotor cortex (BA 6) are active when planning movements and learning motor sequences involving visuomotor tasks and memory (motor memory and working memory). Interestingly, also language-related tasks are processed in different areas of the frontal lobe. In particular, the supplementary motor area (SMA), which is a medial extension from the premotor cortex, is involved in language initiation and voluntary speech production. The same applies to the frontal eye field (FEF, BA 8), as it shares an area with the SMA. Most prominent is the inferior frontal gyrus (IFG) with its two main areas BA 44 and BA 45, which presumably host the mentioned *mirror neurons*. In some literature both, areas are summarized under the term *Broca's area*, while other approaches report that it is only the pars triangularis (BA 45). Both areas belong to the putative human mirror neuron system and are actively involved in grammatical processing on a syntactical and phonological level, grapheme to phoneme conversion, sentence comprehension, and syntactic working memory. For the majority of humans, language is processed in the left hemisphere. In contrast, the right hemisphere is active in the more high-level processing of language and contextual features, which includes (affective) prosodic information processing, metaphor processing, melody generation and expression of emotions.

The prefrontal cortex (PFC, BA 10) is engaged in decision-making, emotion processing (unpleasant vs. pleasant) and joint attention. Its special role was also revealed in memory-related tasks (short-term memory, working memory) and language-specific processes in the dorsolateral PFC (dlPFC, BA 46), particularly for syntactic and semantic processing. Eventually, its functional repertoire yields substantial sequence processing capabilities. This is exploited in several models, of which we present the most noticeable sequence model which is related to the *Reservoir Computing* paradigm.

3.2 Sequence Learning and Transient Dynamics

Human behavior or processing language are inherently structured and follow a temporal order, e.g. learning a specific action or the syntax in a sentence. A key brain area involved in sequence learning is the prefrontal cortex (PFC). It receives signals from sensory and motor areas but has no direct connections to the primary motor cortex (MI). The dorsolateral area of the PFC (dlPFC, BA 46) has connections to the frontal eye field (FEF, BA 8) and basal ganglia, which are involved in goal-oriented behavior and evoking automatic behavior. The backprojections to the inferior temporal area (IT) are presumably involved in the recall of visual memory (Miller et al., 1991).

In the context of structured sequence processing, a neurophysiological model on sensorimotor transformations involving the PFC presented in Dominey (1995) is usually referred to as an inspiration for a new perspective on learning in computational models, namely the *Reservoir Computing* paradigm. Interestingly, Lukoševičius and Jaeger (2009) stated that researchers involved in *Reservoir Computing* but with a different focus, i.e. neuroscientific evidence and investigations on the PFC on the one hand, and concrete neural network implementations, on the other hand, became aware of each others' work not until 2008. We will summarize the neurophysiological model introduced by Dominey (1995), as it demonstrated that random recurrent networks with a separation of the input representation and the training of neuronal connections is a valid method to model sequence processing.

3.2.1 Random Recurrent Networks

Dominey (1995) suggested a sequence generation model for action selection based on an experiment presented in Barone and Joseph (1989), which revealed spatial preference properties of PFC neurons of macaque monkeys, also referred to as “context neurons”. Their experimental setting considered sequences of spatially arranged light flashes (above, left, and right wrt. a fixation point). The monkeys first learned the order of illuminated buttons and were then trained to touch them in the correct order, e.g. 231 if light 2 was turned on first, then light 3 and finally light 1.

Based on this experiment, Dominey (1995) proposed a cortico-striatal model, which aimed at reproducing the transformations of internal states triggered by the visual cues to corresponding sequential actions. A core component in the model comprised the mechanisms between the caudate nucleus (CD) and the prefrontal

cortex (PFC). The latter was modeled as a recurrent neural network to achieve the desired dynamics emulating neural activity patterns over time. The novelty in the network implementation was that the connection weights in the PFC model were not modified during training, but were randomly initialized and stayed fixed (the “reservoir”). The rationale behind this was to provide a pool of neurons representing the feature diversity coming from IT cells. This information storage over a short timescale, the short time memory, is realized by the *transient dynamics* exhibited by the network dynamics. Only the synapses projecting from the PFC to the CD were learned to employ a reinforcement scheme. Dopaminergic neurons strengthened the connections whenever a correct saccade was generated, the actual reward. This essentially implemented an associative mechanism of sequence encodings in the PFC and the corresponding action selection by the CD.

The model was further exploited for sequential processing of abstract representation and their temporal binding, extending it to the language domain. It was shown to preserve syntax in sentence generation and incorporation of semantics (Dominey, 2005; Dominey et al., 2006). The connection between neurophysiological findings to grammatical constructions using *Reservoir Computing* is presented in Dominey (2013).

Complementary to Dominey’s model, Barak et al. (2013) studied the statistical properties of response profiles in PFC neurons involved in a discrimination task considering three network models, ranging from attractor-based networks to random recurrent networks (RRN). In the latter case, only the readout weights were trained corresponding to the scheme in Dominey’s model (Dominey, 1995). In addition, the initial network states showed spontaneous fluctuations. The authors claimed that, despite the random network performing well, none of the models were fully able to reproduce the data. Therefore, the authors suggested that a hybrid model of a random network and trained parts in the recurrent and readout layer may serve as an explanatory model for the underlying data.

In a recent study, Enel et al. (2016) argued that the RRN provides inherently *mixed selectivity*, which explains the representational diversity in the PFC (Rigotti et al., 2013). This approach highlighted further the connection between the PFC reservoir and current neural network implementations. Following the principle of RRN include further studies of context processing in the PFC (Mante et al., 2013) and working memory models (Pascanu and Jaeger, 2011). Other models adopting the reservoir approach even considered the assumption of a “critical” or “chaotic” brain, which is responsible for the acquisition of cognitive skills. We outline some representative work on this topic in the next section.

3.2.2 Transient Dynamics Involved in Learning

Learning and the acquisition of higher cognitive abilities in humans were long described as a convergence of neural trajectories into stable attractors¹, specifically fixed point attractors. However, another explanatory model of cognition is supported by the observation of the mentioned transient dynamics (Babloyantz and Loureno, 1994; Durstewitz and Deco, 2008; Rabinovich et al., 2008; Rabinovich and Varona, 2011) and the concept of chaotic itinerancy (Tsuda, 2009; Faure and Korn, 2001).

Almost three decades ago, Skarda and Freeman (1990) investigated the functional role of chaos beyond the general assumption that chaos appears as noise or as an effect of brain malfunctioning. The findings were based on data recorded from the olfactory bulb and highlighted the role of nonlinear dynamics. The authors put further emphasis on the fast transitory behavior of the brain when confronted with a novel stimulus. Another aspect is the information generation. Due to an internal construction of chaotic activity patterns, a selection mechanism is triggered where the brain filters the significant information necessary for learning and behavior. The work, therefore, suggested revising models for data-driven analyses in neuroscience. Korn and Faure (2003) highlighted chaotic brain activities in the olfactory bulb, exhibiting bursts of gamma and theta brain waves. A recent approach supporting the view on transient dynamics accounting for olfactory processes and odor categorization was presented in Buckley and Nowotny (2012). The idea of chaotic activity in connection with the reservoir concept influenced the work on central pattern generators. Sussillo and Abbott (2009) presented a study of movement generation, arguing that motor capabilities are subject to re-organizational principles of spontaneous cortical activity. They were able to show that the network indeed reproduced different movement patterns, resembling data from the motor and premotor cortex. Boström et al. (2013) reproduced arm movements from electromuscular measurements following the line of argumentation of a “self-active” reservoir.

In neuroscience, the studies of the complex dynamical investigate “selforganized criticality” (SOC, Levina et al. (2007); Shew and Plenz (2013); Shew et al. (2015)), basically a term from physics Bak et al. (1988). The interesting aspect in view of this hypothesis is that the brain activity tends towards a critical state in sensory processing, but keeps distant to the transition border to stay stable (Priesemann et al., 2014). This leads to the yet unsolved question, why the brain favors a regime with less computational power, and how the brain can “measure” itself

¹An exemple is the Hopfield model (Hopfield, 1982).

to not cross the critical border to unstable dynamics. Studies, which specifically investigate phase-transitions and criticality for attention-driven visual processing are, for instance, outlined in Tomen and Ernst (2013); Tomen et al. (2014).

As a metaphor of the *reservoir* in the context of transient dynamics, throwing a stone into the water the surface produces ripples which fade over a certain time interval until the water surface is flat again. This is the final *attractor* state indicating that the system is stable and thus converges (however, as discussed, instabilities can be a desired effect for learning patterns). The transient dynamics can then be thought of the short-term memory of the features, which created a specific wave profile on the water depending on the angle and power the stone was thrown into the water.

A review of transient dynamics, their models as well as their influence on memory and in decision-making processes is provided in Durstewitz and Deco (2008). The perspective on learning in terms of transient dynamics becomes more and more an active research area and leads to shifts in the understanding of computational neural network models, specifically Recurrent Neural Networks.

3.3 Chapter Summary

In this chapter, we explained the computational principles of information processing in the human brain. Starting at the neuronal level, we showed that the cortex has four functional areas and we gave an overview of their role in perception and cognition. We highlighted the prefrontal cortex (PFC) as a “sequence” area based on neuroscientific experiments and models. The introduction of random recurrent networks influenced the perspective on computing with recurrent neural networks as an alternative to attractor-based learning. In line with this research, we presented some work considering transient dynamics and brain criticality as a functional element for learning. We will come back to these aspects in the next chapter.

Chapter 4

Recurrent Neural Networks and Reservoir Computing

In this chapter, we bridge the neurobiological principles in the human brain to their computational counterparts. The advent of digital machines created new research directions like cybernetics and artificial intelligence, with the aim to explain brain processes formally and to equip machines with autonomous behavior. McCulloch and Pitts (1943) suggested an artificial neuron model capable of computing Boolean functions $F : \{0, 1\}^n \mapsto \{0, 1\}$, which became one of the most influential basis of neural network implementations. Their approach was extended by Rosenblatt (1958) with the introduction of the *perceptron* model with randomly initialized weights as synapses, whose connection strengths were determined via training. A sketch of the perceptron is shown in Figure 4.1.

The artificial neuron consists of *input* vectors \mathbf{x} , the size of the input dimension and the corresponding *weights* \mathbf{w} , respectively synapses, which are passed to a summation unit and thresholded according to an *activation function*. The neuron fires if the following holds:

$$\mathbf{x} \cdot \mathbf{w} \geq \theta \tag{4.1}$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is the n -dimensional input and $\mathbf{w} = (w_1, w_2, \dots, w_n)$ are the corresponding weights. Instead using a specific threshold θ , the vectors are usually augmented with an additional *bias* β defined as a constant $\mathbf{x} = (x_1, x_2, \dots, x_n, \mathbf{1})$ and $\mathbf{w} = (w_1, w_2, \dots, w_n, -\theta)$ yields:

$$\mathbf{x} \cdot \mathbf{w} \geq 0 \tag{4.2}$$

The geometrical interpretation of the equation is that a decision boundary is

computed between different classes spanned by the input space.

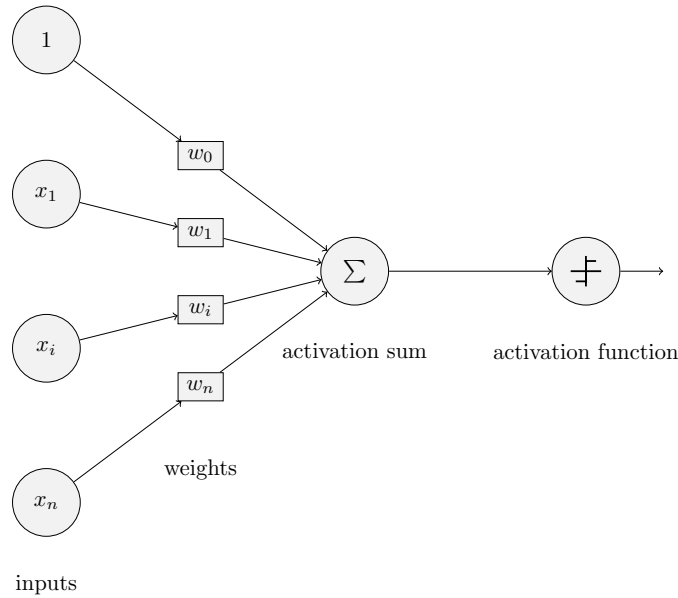


Figure 4.1: The perceptron model with the input layer and an additional bias (constant 1 node). The sum of the input and their corresponding weights is passed to an activation function, here the Heaviside function. This model computes linear decision boundaries.

For more complex problems, the neural processes are modeled as dynamical systems where the weight dynamics are functions of the time. The amount of sensory information available through sensors demand models to learn the spatio-temporal, varying audio and visual information and their sequential correlation is sometimes referred to as long-term dependencies. To achieve that, the model should also implement memory capabilities. Recurrent Neural Network (RNN) provide such an architecture to handle sequences and to provide contextual information over several time spans. The RNN is set up with additional layers of recurrently connected neurons yielding the desired memory property in the network. One of the first neural architectures fulfilling the requirements was the Elman network (Elman, 1990). It extends the notation of a feedforward network by introducing a context layer, which holds at time t copies of the hidden layer activations from time $t - 1$. An example of this network type and of a more general RNN is depicted in Figure 4.3.

Training RNNs is commonly performed by employing the Backpropagation through time (BPTT) algorithm, which computes a gradient along the error landscape w.r.t. the network weights unfolded over time. “Unfolding” here means, that the synaptic network weights are copied spatially according to a time step t . Although the BPTT algorithm was useful for neural networks to solve non-linear and sequential tasks, it also came along with several drawbacks introduced

by the partial derivative calculations. The computed gradient assigns the rate of weight changes to minimize a loss function by propagating back the error through the network. Adding hidden layers may be useful for complex tasks but also the learning speed may differ between the layers. As an example, it was shown that the gradient was biggest in the last layer of an RNN, but exponentially decayed to zero going backward in the network. This means that the first layer(s) contribute only little from learning. The according term to describe this phenomenon was coined *vanishing* gradient, which can cause a problem when the network should learn long-term dependencies. Vice versa, if the gradient grows exponentially it is called the *exploding* gradient (Bengio et al., 1994).

However, also other problems cast application of BPTT-based neural architectures difficult, namely: classic BPTT RNNs are slow in convergence, may get stuck in local minima, and are sensitive to bifurcation occurrences (Doya, 1992). Methods to improve gradient-based learning includes Hessian-free optimization (Martens and Sutskever, 2011) or stochastic gradient descent (SGD). Nesterov (1983) also suggested a first-order optimization method altering the momentum update which was further simplified in Bengio et al. (2013).

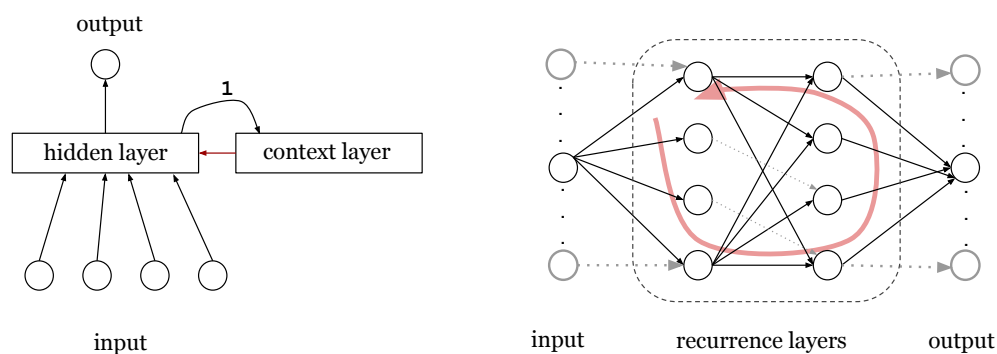


Figure 4.2: Left: Elman network with an additional context layer. The weights are fixed to 1, as the hidden layer activations from the previous time step are copied into it. Right: Neural network with recurrently connected neurons in two hidden layers (dashed box).

The mentioned problems and difficulties in applying RNNs inspired a new way of thinking about RNN training.

An alternative to well-known BPTT training, Atiya and Parlos (2000) proposed the Atiya-Parlos algorithm (APRL). The computation of the loss function w.r.t. weights was reformulated as an optimization problem, resulting in a gradient approximation rather than the exact calculation. The authors reported performance

improvement and faster convergence and were able to derive an online learning rule. Further analysis of the network dynamics for online learning in RNNs was provided in Schiller and Steil (2005). Comparing the weight changes for real-time recurrent learning (RTRL) with a developed version based on the APRL the authors revealed a faster update in the output layer than in the hidden layers. As a result, the authors claimed that their algorithm and the results derived from their study may serve as a transitional model between training networks with RTRL and networks based on *Reservoir Computing* (RC) principles, subsuming the Backpropagation-Decorrelation algorithm (BPDC) (Steil, 2007), together with the Echo State Networks (ESN) (Jaeger, 2002; Jaeger and Haas, 2004) and the spiking equivalent Liquid State Machines (LSM) (Maass et al., 2002).

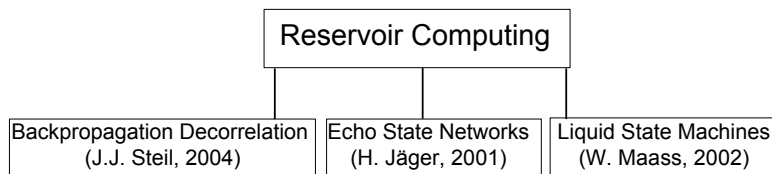


Figure 4.3: Algorithms and network architectures primarily connected to the Reservoir Computing paradigm as proposed by Verstraeten et al. (2007). The Backpropagation Decorrelation learning rule is built on the Atiya-Parlos algorithm which approximates the weight changes in recurrent learning. Echo State Networks (ESN) have been widely adopted in machine learning while the Liquid State Machine with its specific neuron type and topology has gained more attention in the neuroscientific community.

4.1 Echo State Networks

The Echo State Networks (ESN) Jaeger (2002) differ from usual RNNs in the way that the single recurrent layer remains untrained and acts as the dynamical part providing transient responses to the memoryless output which connections are trained using a linear model. The learning is fast and circumvents training issues for conventional RNNs.

The role of the reservoir is the following: the input is mapped into a high-dimensional space spanned by the number of reservoir neurons. This “rich representation” based on nonlinear transforms by the reservoir neurons (usually employing the *tanh* function, see Figure 4.4) facilitates the subsequent regression step. The motivation behind follows machine learning procedures, where a kernel function is employed on the input, resulting in a transformation into a higher

dimensional feature space. Eventually, the separation of the data is linear, thus decision boundaries are easier to compute (an example is depicted in Figure 4.5).

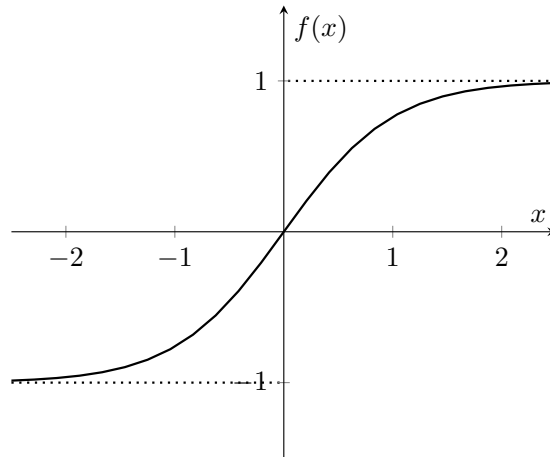


Figure 4.4: The \tanh activation function usually used for the reservoir.

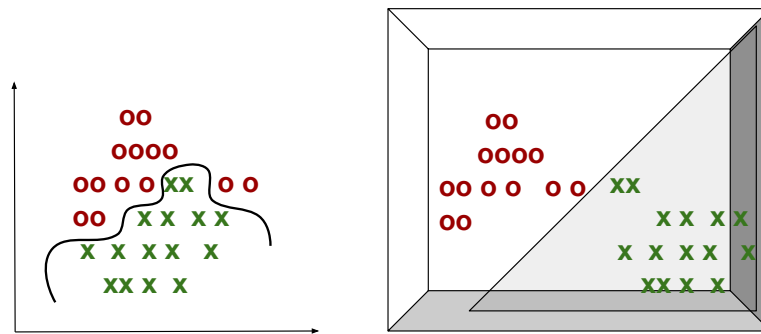


Figure 4.5: Separation of data samples in spaces of different dimensionality. Left: a nonlinear function is needed to correctly classify data samples in \mathbb{R}^2 . Right: Transformation of the data with an additional dimension, i.e. here, \mathbb{R}^3 , provides a linear separable classification. The data is then separated by a hyperplane (here only indicated with the triangle for visualization purpose).

4.1.1 Important Equations and Evaluation Measures for ESN

We now introduce the basic computations considering an ESN with leaky integrator neurons (LI-ESN) (Jaeger et al., 2007). The standard definition of an ESN with analog neurons (Jaeger, 2002) can be derived from the update equation by setting $\alpha = 1$, the leakage rate which we will explain below.

Let u be an external input and x be the reservoir states evolving over time with a predefined time constant c :

$$\dot{x} = c^{-1} \left(-\alpha x + f(W_{in}u + W_{res}x + W_{back}y) \right) \quad (4.3)$$

Discretization according to step δ yields:

$$x(n+1) = \left(1 - \frac{\alpha\delta}{c}\right)x(n) + \frac{\delta}{c}f(W_{in}u((n+1)\delta) + W_{res}x(n) + W_{back}y(n)) \quad (4.4)$$

The reservoir states are then updated according to:

$$x(n+1) = rx(n) + f(W_{in}u(n+1) + W_{res}x(n) + W_{back}y(n)) + \nu(n+1) \quad (4.5)$$

where f is the network activation function, the matrices W_* connect the layers and ν is an additive noise term. It was shown that ν is important when dealing with the feedback modus as to stabilize the system (Jaeger, 2002). A significant regulation term is the retainment factor $r = 1 - \alpha$ and $\alpha \in (0, 1]$ is the leakage rate (see section 4.1.2). Figure 4.7 shows some neuron activations x in the reservoir.

The output is computed as:

$$y(n+1) = f_{out}(W_{out}(u(n+1), x(n+1), y(n))) \quad (4.6)$$

where f_{out} denotes the output activation function, usually the *id* or a sigmoid function. W_{out} are the weights between the reservoir states and the output, which are the only set of weights which are trained.

Training the weights W_{out} simplifies to a (regularized) linear regression¹, here Tikhonov regularization:

$$W_{out} = YX^T(XX^T + \tau\mathbf{I})^{-1} \quad (4.7)$$

where τ is the regularization coefficient and \mathbf{I} is the identity matrix. Regularization prevents overfitting, i.e. unbounded growth of W_{out} weights.

Setting $\tau = 0$ yields the Wiener-Hopf equation as a special case:

$$W_{out} = YX^T(XX^T)^{-1} \quad (4.8)$$

which can be rewritten using the pseudoinverse notation as:

$$W_{out} = (YX^\dagger)^T \quad (4.9)$$

¹Regression matrix notation.

For supervised learning tasks, let the known teacher output be denoted as \hat{y} and the network performance is based on an error measure between the desired output $\hat{y}(n)$ and the actual network output $y(n)$. A standard error measure is the mean squared error (MSE):

$$MSE = \frac{1}{N} \sum_{n=0}^{N-1} (\hat{y}(n) - y(n))^2 \quad (4.10)$$

for two signals of length N . Another measure used is the root mean square error (RMSE):

$$RMSE = \sqrt{MSE} \quad (4.11)$$

In case the two signals differ in their amplitudes, the error can be normalized by the variance σ^2 of the desired signal (NMSE):

$$NMSE = \frac{1}{N\sigma_{\hat{y}}^2} \sum_{n=0}^{N-1} (\hat{y}(n) - y(n))^2 \quad (4.12)$$

which can be further expressed as normalized root MSE (NRMSE):

$$NRMSE = \sqrt{NMSE} \quad (4.13)$$

Equation 4.7 is used for offline learning and is usually applied to classification tasks, while online learning is used for experiments in non-stationary environments. The reason is that for offline-classification the datasets are usually cleaned beforehand in the sense of noise- or outlier removal. The data were usually recorded in settings where environmental variables like lighting or background are set optimally.

In other applications, the objective is to train a network given uncertain environmental settings (e.g. when a robot navigates through unknown terrain). Adaptive filter theory (Haykin, 1996) allows handling these uncertainties. Among others, the Kalman filter and, specifically, the Recursive Least Squares Filter (RLS) are linear estimators of current inputs. For nonlinear state estimations, the Extended Kalman Filter (EKF) offers a statistical alternative to gradient-based RNN.

The objective function for RLS is to minimize the sum of the squared errors weighted by ϕ , which is an exponential forgetting factor $0 < \phi \leq 1$ and thus weighs current values in the update procedure stronger than past information. The error is iteratively computed as:

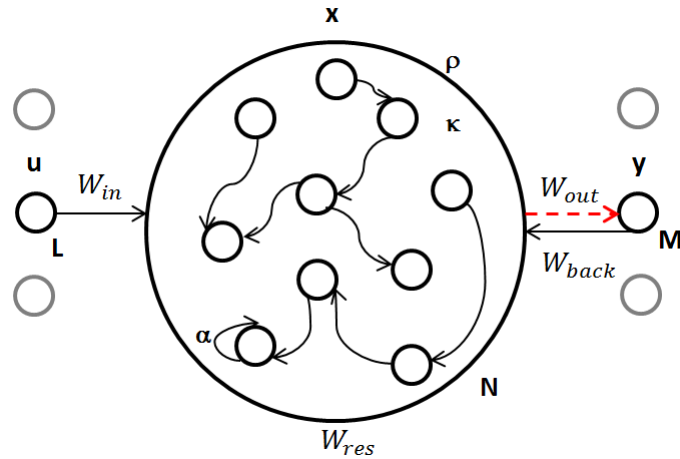


Figure 4.6: Echo State Network as introduced by Jaeger (2002) and referred to as standard ESN throughout the thesis. The M -dimensional input u projects via the weight matrix $W_{in} \in \mathbb{R}^N \times (1+L)$ to the reservoir of size $N \times N$, whose connectivity is denoted by κ . The additional 1 denotes a bias term. The norm of the reservoir matrix W_{res} is denoted here as ρ (spectral radius). The parameter α denotes the leakage rate. The output weight matrix W_{out} (dashed red arrow) is the only network component which is trained via e.g. a regression. All matrices except W_{out} are randomly initialized and the connections stay fixed. Other architectural options use a direct input-output connection and interconnections between output neurons. The model also provides the feedback matrix W_{back} , which can be used for pattern generation tasks.

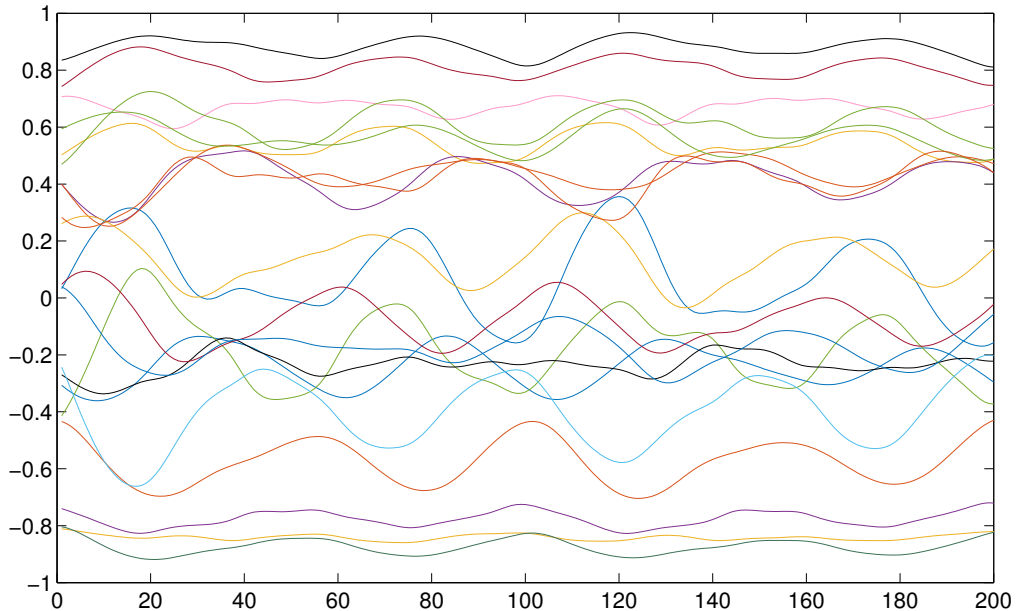


Figure 4.7: Reservoir activations from arbitrarily chosen neurons over time.

$$\varepsilon(n) = \phi^{n-i} \sum_{i=0}^n (y(i) - \hat{y}(i))^2 \quad (4.14)$$

where y is the target response and \hat{y} is the current signal, modeled as:

$$\hat{y} = \sum_{k=0}^{L-1} w_k f(n-k) \quad (4.15)$$

where w_k are the filter coefficients of length L , which need to be updated stepwise and $f(n)$ is an input signal to be filtered. Solving

$$\mathbf{w}_n = \mathbf{R}^{-1}(n)\mathbf{P}(n) \quad (4.16)$$

with:

$$\mathbf{R}(\mathbf{n}) = \sum_{i=0}^n \phi^{n-i} \mathbf{f}(i)\mathbf{f}^T(i) \quad (4.17)$$

$$\mathbf{P}(\mathbf{n}) = \sum_{i=0}^n \phi^{n-i} \mathbf{f}(i)y(i) \quad (4.18)$$

Due to the factor ϕ and the necessity of matrix inversion as shown in (4.16) we cannot apply numerically feasible algorithms. This is why the matrix inversion lemma is used. In general, given a square, nonsingular matrix A and its inverse A^{-1} , the following holds:

$$(\mathbf{A} + \mathbf{BCB})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}^{-1} \quad (4.19)$$

with \mathbf{B} is a $n \times k$ and \mathbf{C} is a $k \times k$ non-singular matrix. Figure 4.8 depicts the prediction using the RLS algorithm on a Nonlinear Autoregressive Moving Average (NARMA, see appendix) task and the computed weights. The NARMA task is a benchmark time-series for prediction tasks, often used when novel ESN architectures are introduced.

Another training scheme called First-Order-Reduced-Error (FORCE) was introduced by Sussillo and Abbott (2009) to enable pattern generation with ESN inspired by the involvement of the cerebellum in learning motor sequences. The reservoir is initialized to exhibit spontaneous activity, resembling a chaotic precondition similar to neuroscientific findings (cf. chapter 3). Given a target function used to learn a specific function and using RLS, FORCE training enables rapid error decrease at the very beginning of the learning stages, yielding finally a network stably reproducing a desired pattern. The approach was further extended to a reward-modulated Hebbian learning (Hoerzer et al., 2012) and arm movement

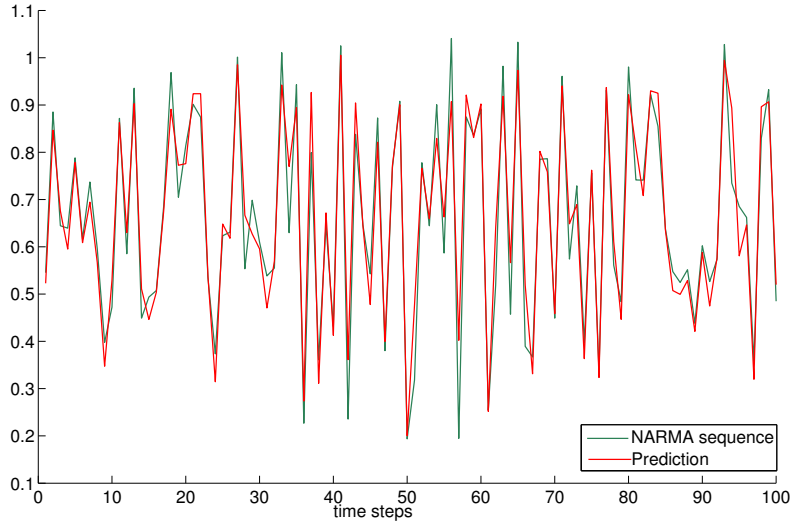


Figure 4.8: Training and prediction of the NARMA sequence using RLS learning and for 100 time steps.

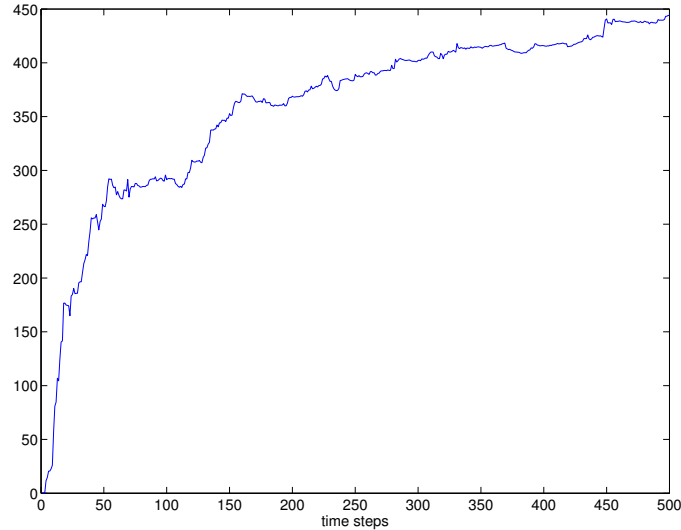


Figure 4.9: Evolution of the weights using RLS on the NARMA task.

modulation based on electromyography data (Boström et al., 2013), promoting the RC paradigm to model motor behavior and motor sequencing tasks.

4.1.2 Network Parameters and Reservoir Configurations

So far, we described the set-up in an ESN and the training procedures involved in learning. Essential to achieving a network with a good performance, may it be in terms of prediction, accuracy or signal reconstruction, is to understand the parameters involved and their effect on the network behavior. The central parameters in

an ESN are:

1. reservoir size r_N
2. leakage rate α in LI-ESN
3. connectivity κ
4. spectral radius ρ
5. input scaling ι

1. As explained, the reservoir acts kernel-like by providing a new feature space for better separation of the input. This is also referred to as the *separation property* of the reservoir. As a loose rule-of-thumb it was long stated that the more reservoir neurons, the more diverse and thus high-performant a network can be. However, results of a study considering the task of multiple imposed oscillators (MSO, see Appendix B) Koryakin et al. (2012) showed that rather small reservoir sizes performed superior to bigger reservoirs.

2. The parameter α was introduced to extend the ESN model with leaky integrator neuron. It constitutes a local memory mechanism for a neuron as α regulates the influence of past neural activity to the current neural state. Moreover, this parameter introduces a time constant to the network by controlling the network ability to respond to new information from e.g. sensors. For small values of α , past information stays longer in the network and is, therefore, suitable for slowly varying patterns while higher values of α can deal better with high-frequency signals, i.e. when abrupt changes occur in sensor data. Whether or not the leak rate may be an effective parameter depends on the task at hand. The work presented in Butcher et al. (2010) introducing a random projection layer to the standard ESN, for example, showed no effect of α for a 4-th order polynomial modeling task but for a noisy version of the isolated spoken digit classification task.

An adaptive mechanism to tune the leak rate α was presented in Dasgupta et al. (2013). The adaption criterion was based on intrinsic plasticity (IP, cf. chapter 3) and an information-theoretic measure, the local adaptive information. The idea behind this approach was to equip a reservoir with flexible timescales depending on event occurrences and their delays, which showed good performance in a robot navigation task for short and long mazes, supporting the hypothesis that individual neuron leak rates scale the temporal resolution of the network supporting also long range dependencies. Also, the spectral radius was shown to exceed unity.

For the following parameters we want to formulate some statements found across ESN literature, but we want to analyze them more critically in order to anchor the investigations in the present thesis.

3. *Random connections are sufficient to achieve good network performance.*

As shown in Figure 4.6 only the connections from the reservoir to the output W_{out} are trained. The synaptic weights for the randomly initialized but fixed matrices are usually drawn from a discrete probability distribution (probability mass function, pmf), e.g. the Bernoulli distribution:

$$Pr(w = \pm 1) = \frac{1}{2} \quad (4.20)$$

yielding a densely coupled network.

Another common choice for continuous weight values is the Gaussian probability density function (pdf):

$$w \sim N(0, \sigma^2) \quad (4.21)$$

where σ^2 is the variance and w denotes the elements $w_{ij} \in W_{res}$. Also, the uniform distribution within a particular range or even constant values are allowed (Jaeger, 2002). An additional connectivity factor $\kappa \in [0; 1]$ can be applied to set up reservoirs with a desired level of sparseness, i.e. $w_{ij} \in W_{res} = 0$. This parameter determines whether the weights drawn from probability distributions should be selected rather small or high - for a dense network it is advised to scale down the weights and vice versa, sparse networks are equipped with higher valued weights.

Networks with a randomly chosen but unaltered weight matrix showed good performance on important benchmark data, e.g. the n-th order NARMA task, the k-delay memory task, or the prediction for the Mackey-Glass time series (see appendix). However, quoting from Lukoševičius and Jaeger (2009): “*just simply creating a reservoir at random is unsatisfactory*” shows, that even researchers in the field of ESN demand a sensible handling of that issue. The main drawback we see in the application of random reservoir weight matrices is that the error between multiple trials deviates significantly (cf. chapter 5). Also, the parameters involved need to be quite fine-tuned, which narrows the acquired freedom that comes with the simplified training and also the area of application.

Although in principle the statement is underpinned by current literature using ESN on an application level (for an overview see Lukoševičius and Jaeger (2009)), some of the approaches motivate their network choice due to the findings in the prefrontal cortex (PFC) and the sequencing mechanisms (see chapter 3). Also, current literature from the neuroscience research is provided to further support of this view by considering the *mixed selectivity* of randomly connected neurons

(Rigotti et al., 2010; Enel et al., 2016). The synaptic wiring based on Hebb’s rule (Hebb, 1949), as well as the functional cortical organization with certain properties (e.g. small-world Watts and Strogatz (1998) or scale-free property Eguiluz et al. (2005)), however, are driven by certain criteria, diverging from the randomness.

Despite the success of random network application in ESN, authors in this field recognize serious variations in ESN performance. Therefore, the ambitions in the research of ESNs were directed altering the reservoir using several techniques. The attempts opened the discussion on the repeatability of experiments when ESNs were employed, without the overhead of additional parameter optimization routines in favor of a more principled way setting up reservoir networks.

A variant of the reservoir initialization was presented in Rad et al. (2010), which made use of the properties of Kronecker matrices. An initial, small reservoir was set up and the parameters optimized using e.g. evolutionary search. When an optimal parameter configuration was found, the Kronecker product can be used to create a large reservoir.

Ferreira et al. (2013) also employed evolutionary search but for optimizing all reservoir neurons which is limiting when considering the search space of a big reservoir. A similar approach using differential evolution for reservoir optimization was presented in Otte et al. (2016) and showed superior performance for the MSO task (see Appendix B) compared to a random set up.

A study changing the algebraic properties of the reservoir weight matrix was conducted by Strauss et al. (2012). Using normal matrices (permutation matrix, sparse orthogonal matrix) and simple reservoir connectivity like neuron chains showed good performance on the benchmark sets like Mackey-Glass-17.

Motivated by the question whether a deterministically constructed ESN with minimal architectural complexity can compete with a randomly initialized standard ESN, Rodan and Tino (2011) presented three alternative ESN topologies on a thorough selection of time-series and tasks evaluated in the RC community: 3 NARMA tasks (order 10 and 20 and a randomized NARMA, Jaeger (2002)), the Hénon map (also, Hénon attractor), the Santa Fe laser dataset², the sunspot data, isolated digit speech recognition (Schrauwen et al., 2007), IPIX Radar data set Xue et al. (2007), nonlinear system with observational noise (Gordon et al., 1993), and nonlinear communication channel model (Jaeger and Haas, 2004). The newly introduced architectures used in the comparison experiments using the data comprised reservoirs constructed as a delay line (DL), a DL with feedback, and a cyclic chain reservoir called SCR. The input- and reservoir weights were set determin-

²<http://www-psych.stanford.edu/%7Eandreas/Time-Series/SantaFe.html>

istically with either a varying sign pattern. For all architectures, either linear or *tanh* reservoir activations were used. For evaluation, the NMSE was used. The authors revealed a good performance for all the presented topologies in comparison with results obtained from the standard ESN. Especially the SCR showed superior performance for the NARMA tasks and was a second best model³ for the laser dataset, the nonlinear channel model, and Hénon map, where the standard ESN showed an overall best performance. The recognition task on spoken digits was the only high-dimensional input data (86 frequency channels obtained by post-processing speech data (Schrauwen et al., 2007)) and all introduced ESNs achieved good performance. The authors highlighted the role of the reservoir size and the nonlinearity in the reservoir to be eminent for solving the tasks, although the performance increase as a function of the reservoir size was shown to be not monotonic (in line with this finding, just naively increasing the reservoir size led even to performance deterioration (Qiao et al., 2016)). As a global result, the study showed that simpler ESN architectures solve equally well typical benchmark tasks but provide more control over the network settings.

We mentioned above the sparsity or connectivity factor κ , usually set to a small value to obtain a loosely coupled reservoir weight matrix (a sparse matrix has only a small fraction of non-zero elements; an example is depicted in Figure 4.10). Gallicchio and Micheli (2011) showed that the sparsity factor was misinterpreted, as both sparse and dense connectivity yield the same performance. Thus, sparsity can no longer serve as an explanation for the good performance of ESN. Nevertheless, sparsity can be a crucial factor in terms of computational performance for time-critical applications.

4.&5. *A stable ESN needs to have a spectral radius $\rho < 1$.*

In the past years, discussions emanated in the Reservoir Computing community about the necessity of obeying this specification, as the concept of the ESP has led to misunderstandings and hence to a lack of investigation of the real network potential. Especially when involved in applications, authors often lack to explain the connection of their parameter settings to the actual functioning of the network for their specified task. This is why several authors put the emphasis in their work to unveil the question *how* reservoir networks achieve their computational capabilities, and which dynamical regimes are best suited to which tasks. Because the number of tasks for employment reservoir computing techniques differs in their input statistics, network memory requirements and application modes,

³We assume that the best models for each architecture was obtained after averaging over trials and parameter configurations.

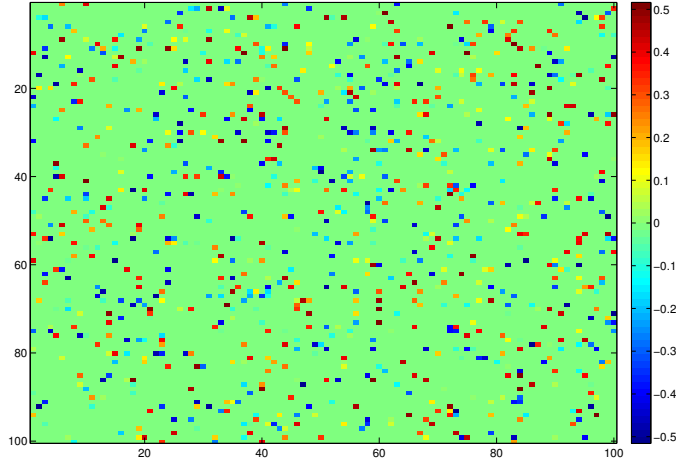


Figure 4.10: Weights drawn from a uniform probability $\mathcal{U} \in [-0.5;0.5]$ in a sparsely connected reservoir of size 100 with $\kappa = 0.1$. Only 10% of the neurons are connected.

no general rule about the selection of spectral radius ρ was established. An intuitive attempt to set some parameter ranges beforehand lie in the dynamics of the task itself, whether the network response and hence its dynamics should be slow or fast. For instance, learning to accurately predict long sentences requires more memory to learn the correlations between the words (or another representation of the sentence). In contrast, navigation tasks can benefit from a network with fast switching capabilities.

4.2 Stability in Echo State Networks

An important theoretical statement about the proper functioning of an ESN is expressed as the Echo State Property (ESP). The recurrently connected neurons, the states, *echo* their activation for a finite time. In a stable system, respectively in a stable ESN, the influence of the input decreases gradually and does not persist nor gets amplified. The system evolution is then independent of its initial conditions and always converges to a fixed point.

The ESP was first outlined in Jaeger (2002). Here, we state the so-called *forward* specification of the ESP and refer to Yildiz et al. (2012) for more details.

For the compactness condition regarding the input, let $\{u^{+\infty} = (u_1, u_2, \dots) | u_k \in U, \forall k \geq 1\}$ denote the right-infinite input and $\{x^{+\infty} = (x_0, x_1, \dots) | x_k \in X \forall k \geq 0\}$ the right-infinite state sequences for compact sets U and X , respectively. The ESP can then be stated as follows:

Theorem 4.2.1. *A network $F : X \times U \rightarrow X$ satisfies the echo state property (ESP)*

w.r.t. U iff it is uniformly state contracting, thus if there exists a null sequence $(\delta_k)_{k \geq 0}$ such that $\forall u^{+\infty} \in U^{+\infty}$ and $\forall x^{+\infty}, x'^{+\infty} \in X^{+\infty}$ compatible with $u^{+\infty}$, it holds that $\forall k \geq 0, \|x_k - x'_k\| \leq \delta_k$.

Intuitively, the theorem serves as a guarantee that two state space trajectories x, x' asymptotically converge independent by their starting conditions (here the L_2 -norm (Jaeger, 2002) and holds generally for any L -norms). Regardless of any perturbations of the states the trajectories stably converge. A dynamical system with this characteristics is then said to be *uniformly state-contracting*. Computation of the Lyapunov exponent (see section 4.4) is used to express the state behavior of a dynamical system in a quantitative way.

Some remarks on the definitions:

- The compactness of the activation space holds when a squashing function like the *tanh* function is applied component-wise, which is a common choice in the ESN literature. In general, every function which satisfies the Lipschitz condition can be applied: $|f(a) - f(b)| \leq |a - b|, a, b \in \mathbb{R}$.
- No feedback into the reservoir is considered.
- The ESP is defined for any input thus including the null sequence.

Let further be $\Lambda(W_{res})$ the spectrum of the reservoir matrix $W_{res} \in \mathbb{R}^{n \times n}$ with eigenvalues $\lambda_i \in \Lambda$ and with the spectral radius defined as $\rho(W_{res}) = \max\{|\lambda| : \lambda \in \Lambda(W_{res})\}$. To obtain a network with fading memory property and stable state behaviour, Jaeger (2002) characterized a network to have the ESP:

Theorem 4.2.2. *An echo state network with a fixed random reservoir matrix W_{res} and *tanh* state activation function has the echo state property if the maximal spectral radius $\lambda(W_{res}) < 1$.*

The system fulfilling the criterion is referred to as local asymptotically stable⁴.

In practice, Jaeger suggested to choose a value v and scale the reservoir matrix accordingly to achieve the desired spectral radius. The ESP is violated for $\rho > 1$ given the definitions above rather than necessarily scaling W_{res} below unity. This led to some misunderstandings in the RC community. We also provide the notation of the *sufficient* condition for the sake of completeness (global asymptotic stability)(Jaeger, 2002):

⁴The system is marginally stable if $\rho = 1$

Theorem 4.2.3. *An echo state network with a fixed random reservoir matrix W_{res} and \tanh state activation function has the echo state property if the maximal singular value $\sigma(W_{res}) < 1$.*

For the introduced criteria it holds that for a matrix A , $\rho(A) \leq \sigma(A)$.

One proposition was introduced in Buehner and Young (2006) shown to be less restrictive than the singular value and applicable to specifically structured reservoir matrices, e.g. triangular matrices (lower or upper off-diagonal):

$$\sigma(AW_{res}A^{-1}) < 1 \quad (4.22)$$

for any matrix A which has full rank.

As a consequence about the criticism that the *necessary* or *sufficient* conditions do not hold in practice, Yildiz et al. (2012) defined another criterion on the ESP for any input based on the Schur stability for the reservoir matrix:

Theorem 4.2.4. *An echo state network satisfies the echo state property for any input if the internal weight matrix W_{res} is diagonally Schur stable, i.e. there exists a diagonal $P > 0$ such that $W_{res}^T P W_{res}$, where P is negative definite.*

Gallicchio and Micheli (2011) investigated the relationships between the contractivity properties implied by the network boundary conditions and the task performance of an ESN. The authors identified the *Markovian bias*, respectively, the *Markovianity* of an ESN independent of any specific topology by derivation of a contractivity coefficient based on the singular value σ of the reservoir weight norm. The question of an appropriate ESN architecture was investigated further under four aspects, namely the input variability, inherent multiple time scales, neuronal nonlinear interactions, and reservoir dimensionality. The two latter factors were also identified to be crucial factors of ESN performance in the work of Rodan and Tino (2011), but Gallicchio and Micheli (2011) focused rather on an answer whether the enhanced feature space or the increase of the interactions using nonlinear neurons adds to the reservoir diversity necessary for successful employment of the subsequent (linear) regression.

An alternative to avoid any tuning of the spectral radius ρ based on information theoretic measures is proposed in Ozturk et al. (2007) called the ASE-ESN. Being critical about the procedure creating random reservoirs and setting different parameters for the spectral radius, the authors introduced the average state entropy (ASE) to measure information content for solving a specific task, and then optimize an additional bias to maximize the ASE. This way, the stability margin of the network and thus the spectral radius was now controlled by the bias. The work

reduced the strategy of parameter searching for an optimal network for a given task to an optimization problem for one parameter, rendering the computation of the cost function easy. Their network design compared to the standard ESN showed superior performance in a memory task, parity check, and system identification. The authors highlighted additionally that the best performance was achieved at the stability border. The system is then said to be in a *critical state*.

The Edge of Stability

Criticality is achieved at the system transition border (Langton, 1990) for which the (approximated) Lyapunov exponent $\lambda \approx 0$. This computational regime is often referred to as edge of chaos, but as limit cycles and oscillations may occur without being classified as chaos, we prefer the notation of the *edge of stability* (EOS). This also better describes the stability margins when tuning the parameters⁵.

As the stability is tuned by the spectral radius ρ several approaches in the ESN literature reported setting this parameter to obey the ESP. But the argumentation for setting the parameter values for real world scenarios should rather emerge from the task itself and not from the definition. Both statements, either to tune the spectral radius below or above unity, have no general validity (see section 4.1.2). Also, the input scaling and the choice of activation function plays a crucial role. The limits of the *tanh* function provide state boundaries already, which is why the spectral radius can also be much higher than unity. Until today, the question of the correct scaling of the global parameters of an ESN has to be solved for every specific task. An analytical approach in connection with the stated conditions (section 4.2) is given in Yildiz et al. (2012).

But why is the *edge of stability* interesting? In chapter 3 we introduced the hypothesis that learning in the human brain is performed in a (sub)critical state and that spontaneous, chaotic activity may have a functional role in cognitive tasks. This is further supported by the assumption that information processing is maximized at the critical border.

The work in Boedecker et al. (2009) presented a study for ESN to examine the information maximization hypothesis in a computational context. The authors defined measures based on the transfer entropy (first presented in Obst et al. (2010)) and average information storage (AIS). As a result of a memory task and on a 30th-order NARMA task, the authors reported best performance (normalized root mean squared error, NRMSE) for networks scaled to the EOS. A proposition

⁵A thorough paper on that topic for LSM is presented in Bertschinger and Natschlaeger (2004).

following from the results was that the measures may guide the self-organization in the network.

Barancok and Farkas (2014) connected network stability at the EOS and memory when driven with input. For their experiments, the authors used an ESN with one input and one output, and *tanh* reservoir activation. The signals under investigations were the Mackey-Glass prediction, a 30-th order NARMA task and a memory task (Jaeger, 2001a). Setting up reservoirs with different sparsity levels, the work highlighted that sparsity is a crucial factor influencing stability and memory, which in general is known to be reciprocal, i.e. the more stable a network is the less memory can be obtained from the network. For all signals and sparsity levels, the memory capacity (MC) was computed. As a result, the MC was maximized at the EOS in relation to the sparsity, and the theoretical limit as indicated by Jaeger ($MC \leq N$), the number of reservoir neurons being an indicator of input history stored therein) was also verified in the experiments. The authors emphasized that results on the MC may vary when using other activation functions than the *tanh* function (the validity of this approach is mentioned above and considers all activation functions satisfying the Lipschitz condition).

4.3 Memory in an Echo State Network

Jaeger (2001b) introduced a measure for the calculation of the short-term memory (STM) obtained from the reservoir. Let s_n be a signal of length n fed into the network, let k denote the delay which the network should be able to recall, and y_k are the output units. A measure for the input signal s_n in relation to the current output by the trained network was defined by Jaeger (2001b) as:

$$d[W_k^{out}](s(n-k), y_k(n)) = \frac{cov^2(s(n-k), y_k(n))}{\sigma^2(s(n))\sigma^2(y_k(n))} \quad (4.23)$$

where *cov* measures the covariance between the delayed version of an input signal s and the calculated output value for the delay k , and σ^2 denotes the variance.

The STM capacity for a delay k is computed as:

$$MC_k = \max d[W_k^{out}](s(n-k), y_k(n)) \quad (4.24)$$

The total STM capacity is then computed as:

$$MC = \sum_{k=1}^{\infty} MC_k \quad (4.25)$$

The defined memory capacity MC ranges between $[0; 1]$ and quantifies the signal reconstruction. If $MC = 1$ the signal was perfectly reconstructed and $k \leq N$. In case $k \geq N$ MC would drop to zero.

A bound on the memory capacity was studied by Jaeger (2001b) for independent and identically distributed input (i.i.d.) and a linear output function f being the necessary conditions for the following proposition:

The memory capacity for recalling an i.i.d. input by an N -unit RNN with linear output units is bounded by N .

Assuming for the reservoir and output the identity function \mathbf{id} and again i.i.d. input it was also shown in the paper that in that case $MC = N$ with a monotonically decreasing memory trend.

For the creation of the reservoir, several initialization techniques and usage of reservoir activation function exist. The approach in White et al. (2004) the STM was investigated for an orthogonal reservoir with linear activation and noise injection. One way to model that is to use a distributed shift register (DSR), a network which implements a variant of a delay line network, where the input signal is expanded to the different, orthogonal directions. This has a crucial impact on the subsequent signal reconstruction realized by the memory capacity in that network. For zero noise, all neurons in the reservoir contributed to the network memory, where with increasing noise the capacity dropped. In general, for a defined memory function $m(k)$, where k denotes the length of signal history for retrieval, a value of 0 assigns no recall from the network and 1 is achieved for perfect signal reconstruction. As the suggested network is error-prone to neuron loss (structural noise), the authors used a fully connected DSR. An optimal spectral radius ρ was found to be ≈ 1 and in the case of nonzero noise 1.

Also, a usual random network was used in their study. For fully connected matrices initialized as a random Gaussian, the authors claimed that presence of noise, in fact, regularizes the system but did not contribute to give extensive memory capabilities. Indeed, the generic fashion of such a network is desired for diverse applications, however, constraints to the spectral radius and noise type may hinder a general statement about memory in these networks and need more investigations.

A study presented in Ganguli et al. (2008) in line with the aforementioned work focused on the limits of memory in linear networks. They further introduced the Fisher memory curve (FMC) as an analytical tool to describe the signal-to-noise ratio (SNR) both for the network and the input. The authors showed poor memory performance for generic networks with a normal reservoir matrix which is in contrast to the results from Jaeger (2001b) and Maass et al. (2002). Therefore,

they suggested to model memory properties in the PFC or hippocampus rather with nonnormal reservoirs.

An extension of the investigations of the memory in a reservoir using high-dimensional input was presented in Hermans and Schrauwen (2010). To shed light on the memory distribution, a Principal Component Analysis (PCA) was applied beforehand to decorrelate the input. A memory function was derived based on the equations firstly presented in Jaeger (2001b) where the total amount of memory capacity was summed from the MC results of the individual input channels. The architecture varied in the number of neurons, input scaling, and spectral radius ρ ; for activation, the *tanh* function was used. Based on their experiments, the authors showed that higher input scaling values led to a memory decrease, which results from the saturation range of the employed function. As a consequence, a drop of the spectral radius was observable.

Furthermore, the connection between the PCA representation of input data and memory revealed that the first principal components (PC), i.e. eigenvectors with its associated eigenvalues representing most of the underlying data variance, benefit most from the memory, which has an impact on the signal reconstruction. In relation to ρ being close to unity, the authors inferred empirically that the memory is distributed over the first PCs approximately proportional to the square root of the PC energy, i.e. standard deviation of the PC. In turn, a decrease of parameter ρ led to a uniform spread of memory covering also parts of the signal less contributing to signal approximation represented by weaker PCs.

4.4 Important Definitions and Concepts from Dynamical Systems

Dynamical systems and their behavior are described as a set of differential equations yielding the systems states. Their changes over time can be traced in their corresponding phase space. The state space trajectories are commonly referred to as *orbits*. The points a system settles into are called *attractors* or basin of attraction.

In the perspective of a dynamical system in (computational) neuroscience *attractors* serve as models for memory (Durstewitz et al., 2000) or motor behaviour (Schoener et al., 1995). To process or retrieve the information it is assumed that higher cognitive tasks like sequencing or classification follow along attractors, where perturbations like noise enable the network to switch between different attractor

states.

Stable state behavior is described in its simplest form for fixed point and limit cycle attractors (periodic phenomena like heartbeat or pacemaker). In this case, two phase space trajectories which initial conditions differ about an ϵ close to zero, converge. Vice versa, if the two trajectories diverge the system is called chaotic. The Lorenz system describing heat convection in the atmosphere is a classic example (see Figure4.11). The phase portrait for some given parameter values for which the system exhibits chaotic behavior is depicted in Figure4.12.

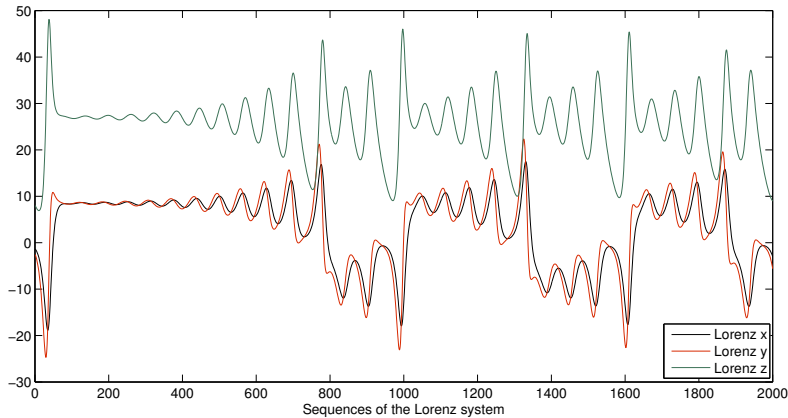


Figure 4.11: Sequence plot of Lorenz system with the parameters $\pi = 10$, $\rho = 28$ and $\gamma = 8/3$. The Lyapunov exponent is $\lambda_1 > 0$.

Attractors can have different shapes such as lines, spheres, tori or manifolds, and combinations. Chaotic attractors have no shape descriptions and thus no geometrical description. Also, there can be strange attractors which have non-integer system dimensions.

The different categories of system state behavior have implications on the predictability, i.e. while for stable systems the predictability is rather high, it drops significantly for chaotic systems. Besides the computation of the Lyapunov exponent, other tools are available for the detection of stability in a dynamical system (Milnor, 1985). This will be the topic of chapter 6.

Whether a system is stable or not can be determined by the Lyapunov exponent. The system behavior in an n -dimensional space is characterized by the so-called Lyapunov spectrum of n Lyapunov exponents $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ sorted in decreasing order. From a geometrical point of view, these exponents measure the sensitivity to the initial conditions in a dynamic system. Say we have two orbits, in a chaotic system the distance of their trajectories will increase exponentially:

$$\frac{d}{d_0} = e^{\lambda(t-t_0)} \quad (4.26)$$

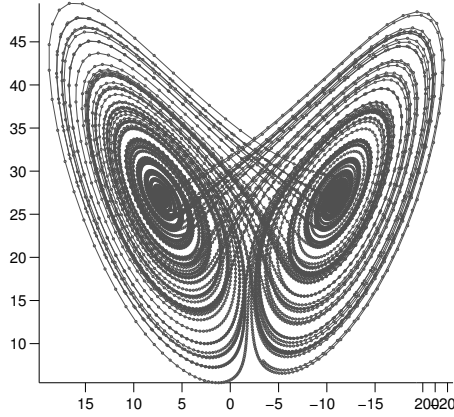


Figure 4.12: Phase portrait of the Lorenz attractor projected to the 2D plane. It shows the two distinctive areas where the system is attracted to but does not converge, i.e. there is no fix point.

Table 4.1: Lyapunov exponent

| stable state | transition | unstable state |
|---------------|---------------------|----------------|
| $\lambda < 0$ | $\lambda \approx 0$ | $\lambda > 0$ |

where d_0 determines the initial displacement, d the current displacement picked at a time-point $t > t_0$ and t_0 the start. The definition used to determine the system behavior is:

$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \frac{1}{(t_i - t_{0i})} \ln \left| \frac{d_i}{d_{0i}} \right| \quad (4.27)$$

The term of a local Lyapunov exponent (LLE) is used to better distinguish between the theoretical base of infinite iterations to calculate λ and the approximation. As mentioned above, the order of the exponents is decreasing, hence the maximal value in the spectrum is denoted by λ_1 . This information is used to classify a system into chaotic or non-chaotic, i.e. $\lambda_1 > 0$ iff the system is chaotic. Vice versa, $\lambda_1 \leq 0$ iff system is not chaotic. For $\lambda \approx 0$ the system reaches the stability border.

An approach based on the estimation of the Local Lyapunov Exponent, and the effects of parameters ρ , σ , and input scaling was presented in Verstraeten and Schrauwen (2009). In addition, the authors also considered the structural value μ known from control theory. Instead of tuning the parameters, the authors derived the LLE using the Jacobian $\mathbf{J}(W_{res})$. The \tanh function was used for the reservoir activations, resulting in a simplified notation of the Jacobian in (only considering the diagonal of the Jacobian). With that approach, the work contributed to measure stability and highlighted the role of the LLE as a predictor of performance for three experiments commonly used in the ESN literature, namely 30th-order

NARMA task, Mackey Glass prediction ($\tau = 17$), and speaker identification. A mathematical analysis of the LLE state using plasticity learning rules in random networks was presented in Siri et al. (2008).

4.5 Chapter Summary

In this chapter, we introduced the Recurrent Neural Networks and explained the differences between traditional training procedures and the principles of *Reservoir Computing* (RC). We introduced the theoretical framework of Echo State Networks (ESN), which are a particular architecture following the RC paradigm. In this context, we outlined the underlying equations, parameters, and conditions for proper implementation of such a network. We specifically highlighted the controversy regarding the statements of parameter settings and obedience of the echo state property. We concluded this chapter presenting the studies on the memory capacity of ESNs.

Chapter 5

Processing Dynamic Gestures with Different Representations

In this chapter, we present our first experiment on gesture recognition with ensemble Echo State Networks (ESN)(Jaeger et al., 2007). The aim is to investigate the influence of two, very distinct gesture representations on the classification capabilities for the mentioned networks.

First, we introduce our gesture vocabulary and how the gesture was performed for the video recordings. We then explain our vision-based preprocessing scheme for the detection of the hands and the feature extraction techniques for a representation, which we call the *simple* set. Second, we call the counterpart of this set the *complex* feature set, derived from a specific deep neural network architecture. We delineate potential and limitations of each gesture representation coupled with the performance of the ensemble ESN.

5.1 Gesture Recordings and Preprocessing

The application of Echo State Networks to the task of gesture recognition is relatively sparse (cf. chapter 2). Hence, the comparison of other gesture recognition studies using an ESN in terms of their representations, their processing in the network, and finally the performance is difficult. We, therefore, decided to conduct experiments adopting an ensemble ESN approach introduced by Jaeger et al. (2007), which complements their study.

As we are interested in gestures, which potentially can replace speech, we recorded dynamic command gestures. They are sketched in Figure 5.1 and the specification of the actual gesture performance is summarized in Table 5.1.



Figure 5.1: Sketch of the gestures performed. From left to right: *circle*, *point left*, *point right*, *stop*, *turn*. The red crosses depict the directions.

We recorded gestures both from the built-in robot NAO camera and a commercial webcam, but due to the low frame-rate from the robot, we decided to proceed with webcam recordings. The device had a frame-rate of 30f/s and a 640×480 image resolution. We chose the distance to the camera as to mimic a conversation, i.e. approximately 1 meter.

5.1.1 Gesture Performance Description

The performance of a dynamic gesture comprises the pre- and post-stroke phase, assigning the start and end of the gesture (chapter 2). In our setting, the pre- and post-stroke phase were always the same: the subject in the scene held the arms and hands down, and faces the sensor as shown in Figure 5.2. The actual gesture performance is referred to as the *stroke* phase. We describe the correct execution in Table 5.1.

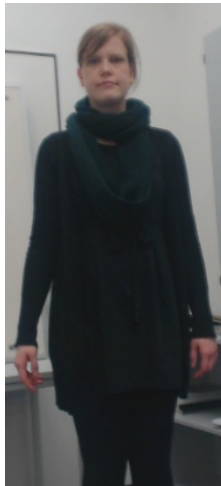


Figure 5.2: The start and the end position during the recordings.

All gestures were constrained to be performed with the right hand but with

Table 5.1: Description of the gesture performance

| Gesture | Realization | Samples | Class |
|-------------|---|---------|-------|
| Circle | Right arm trajectory in xy-plane forming a round shape (2x) | 20 | 1 |
| Point Left | Right arm moves to the left, index finger shows the direction | 20 | 2 |
| Point Right | Right arm moves to the left, index finger shows the direction | 20 | 3 |
| Stop | Right arm moves up, hand palm frontal to the camera. | 20 | 4 |
| Turn | Right arm trajectory in xz-plane forming a round shape (3x) | 20 | 5 |

the subjects' individual speed. Further, we did not use any markers or colored gloves to promote an intuitive gesture execution. For a better visualization of the differences between the *circle* and *turn* gesture, we show two samples from the recordings in Figure 5.3



Figure 5.3: Examples from the recordings. Left image: the *circle* gesture. Right image: the *turn* gesture.

Dynamic gestures performed across individual subjects introduces at least two sources of variations in the execution. As a demonstration, we recorded the performance time of the five defined gestures from two persons uninformed about the actual experiment. With this disjoint set, we got an unbiased time track of gesture performance, as participants in the concrete experiments aim at performing

the same gesture as similar as possible. The results are depicted in Figure 5.4 and Figure 5.5. Two sources of variations can be identified from the plots: first, the *intra-subject variability*, which denotes the performance variations within each gesture class shown by the boxplots. As an example, the *circle* gesture carried out ten times in Figure 5.4 ranges from 2.7 seconds to 3.3 seconds. Apparently, the gesture realizations fluctuate for each gesture. In contrast, the evaluation of the time durations shown in Figure 5.5 vary less in each gesture class. However, when compared to the first subject we observe an *inter-subject variability*, i.e. deviations in the gesture production between the two subjects. A two-sample t-test with a significance level $\alpha = 0.01$ supports the relevance of performance variance, i.e. the hypothesis that the samples share the same distribution with equal means was rejected.

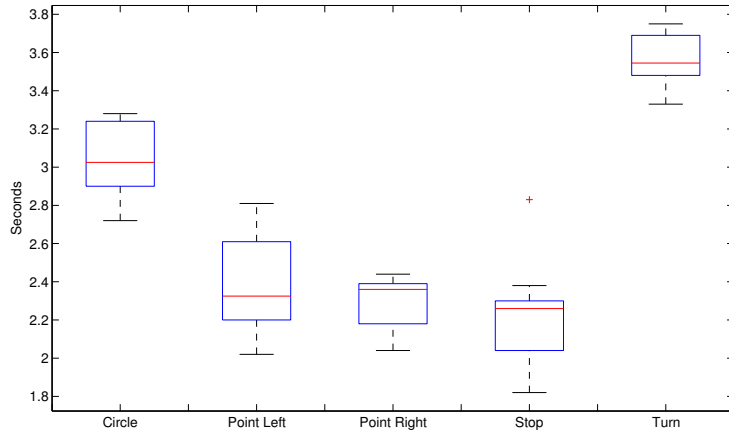


Figure 5.4: Variance of gesture performance of one subject for the five defined gestures carried out ten times. The y-axis denotes the time needed to perform the gestures in seconds. The + marks an outlier.

5.1.2 Preprocessing and Feature Extraction

We recorded the gestures as described and converted the videos into frames using the Linux `mplayer` tool (see Appendix B for concrete command). We inspected the data and deleted redundant frames from the pre- and post-stroke phase, as we were creating a so-called isolated gesture dataset. In contrast, in a continuous setting, these phases would belong to the incremental learning of meaningful gestures and gesture pauses.

The resultant sequence lengths obtained from this postprocessing ranged between approximately 60 frames for the *stop* gesture and approximately 130 frames for the *turn* and *circle* gesture.

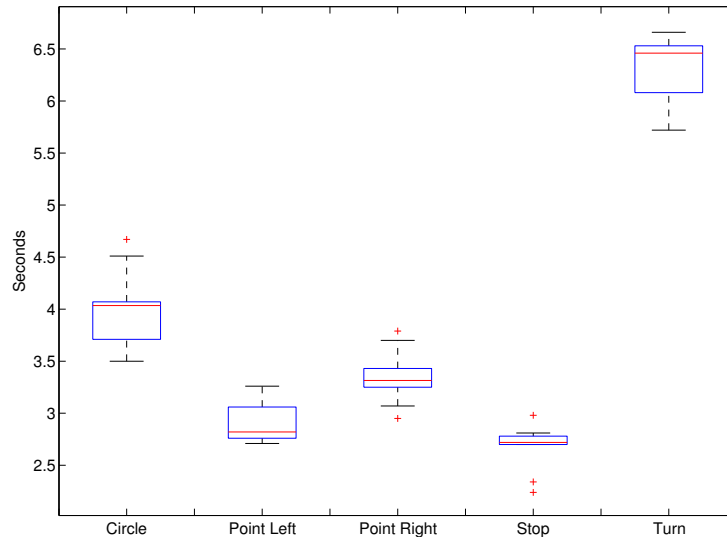


Figure 5.5: Variance of gesture performance of one subject for the five defined gestures carried out ten times. The y-axis denotes the time needed to perform the gestures in seconds. The + mark outliers.

The extracted images in the RGB format were converted into the YCbCr color space to account for luminance changes and to facilitate extracting the skin color. We applied a pixel interval on the Cb and thresholded the images in this new color space on the Cb and Cr components according to a predefined pixel interval on skin color (Kakumanu et al., 2007). The resultant binary images contained then only the hand and the face region. In more detail, areas from the skin detection were labeled 1, while all other areas not containing skin information were labeled 0 (logical representation). Also, the sensor introduced noise, which corrupted our regions of interest. Some pixels actually belonging to the same object region like the face or hand were incorrectly labeled as 0. In general, this directly influences the subsequent object labeling, as one object could be detected as two or more. Hence, the hand or the face could be falsely classified and, in a later stage, wrongly tracked. We, therefore, applied additionally a hole filling procedure, a morphological technique in image processing to create image blobs (binary large objects). We performed a connected component analysis based on the 8-pixel connectivity (see Figure 5.6). This step basically labels pixels in the desired object regions.

A threshold on the resultant blob sizes yielded the face and hand region, where we only considered the latter for the feature extraction. Figure 5.7 shows a sequence of raw images and the binary images resulting from the preprocessing. Especially the face region depicts the influence of noise.

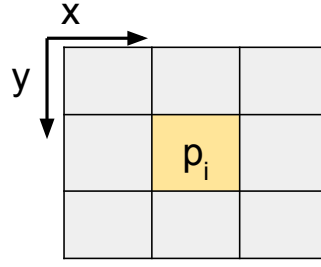


Figure 5.6: For the connected component analysis we used the 8-pixel connectivity, i.e. considering the vertical, horizontal, and diagonal neighborhood of pixel p_i .

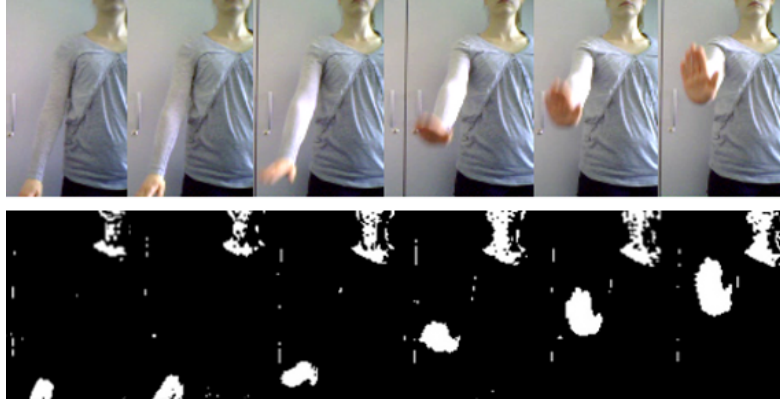


Figure 5.7: Image sequence of a *stop* gesture and the resultant binary images yielding the hand and the face region. Note the noise corruption at this stage, which is observable from the small white areas and interferences in the hand and face area.

Simple Features Set

From the procedure described above, we obtained the hand area and extracted a small feature set from it. We computed the center of mass $\{x, y\}$ from the hand, which described the corresponding position over a gesture sequence. Additionally, we implemented an ellipse descriptor to flexibly fit the global hand shape. An ellipse in its parametric form is described as:

$$\begin{aligned} x(\theta) &= \alpha_0 + \alpha_x * \sin(\theta) + \beta_x * \cos(\theta) \\ y(\theta) &= \beta_0 + \alpha_y * \sin(\theta) + \beta_y * \cos(\theta) \end{aligned} \quad (5.1)$$

where (α_0, β_0) are the ellipse center points (here equal to the center of mass), (α_x, α_y) the major axis vector and (β_x, β_y) the minor axis vector. An example is depicted in Figure 5.8. The resultant feature set $\{x, y, \theta\}$ is referred throughout the work as *simple* feature set.

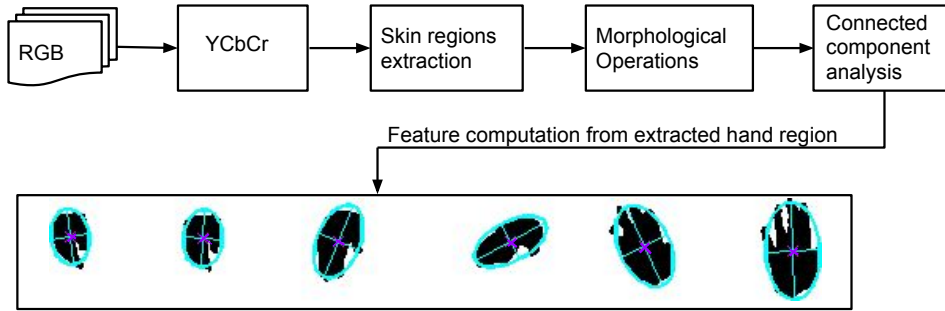


Figure 5.8: The preprocessing steps and the resultant hand extraction from frames of a *stop* gesture. An ellipse (light blue) is fitted to the global hand shape and the corresponding hand orientation is computed. The hand center (x, y -coordinates) is shown with the magenta cross.

A Deep Network For Complex Features

For a different gesture representation, we also set up a feature set extracted from a Multichannel Convolutional Neural Network (MCCNN) (Barros et al., 2014), an extension of a convolutional neural network (CNN). The latter is a feedforward network trained in a supervised way. It is a model reflecting the hierarchical information processing in the visual cortex. While neurons in the V1 area are activated by low-level visual features like edges, neurons in the higher cortical areas assemble this information into shapes and finally compositions of shapes into objects.

In a CNN, this is achieved by an alternating computational scheme consisting of convolution and pooling. First, an image is separated into receptive fields which are convolved with a predefined number of filter maps or *kernels* with prespecified sizes. Intuitively, the receptive fields are image patches which are convolved with a filter matrix sliding through the image. Then, after the convolution, a pooling operation is applied. Figure 5.1.2 shows an example of this operation, which in effect subsamples the image. The whole procedure is then carried out again in the consecutive layer, this time with a different set of filter maps. This hierarchical processing yields image features invariant to scaling and translation. Finally, a feature vector resulting from this computational scheme is fully connected to a set of neurons in an additional hidden unit. These neurons finally project to the output, where the number of e.g. object categories determines the size of the layer. What is trained in this architecture are the filter coefficients or weights

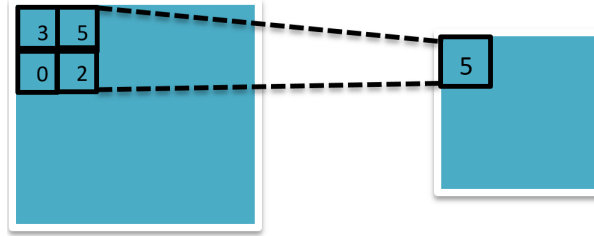


Figure 5.9: Example of a receptive field with different pixel values. The *max* pooling operation compresses this image patch to its maximum value. This procedure is applied to all receptive fields of an image, whose concrete size depends on the predefined filter size.

of the mentioned filter matrix, usually learned by applying the backpropagation algorithm (see chapter 4).

As motivated in section 5.1 gesture performance can vary significantly between subjects. Thus, an invariant representation according to different scales or positions (translation) might be advantageous to account for the mentioned *inter-subject variability*. Another benefit using a CNN architecture is that we do not need any additional image preprocessing. Instead, the raw images obtained from a camera are directly fed into the network. The only exception is that images are downsampled beforehand for computational reasons.

The MCCNN used for our experiments extends the convolution operation to a cubic kernel in the first layer. In our experiment, the images were reduced to an image size of 28×28 . They were fed into three channels, where a Sobel filter was applied for the edge detection in both image directions. The third channel converted the 3D RGB images into grayscale images. A cubic kernel in the first layer is then applied to the output of the three channels. We used 50 filters in the first layer with a size 5×5 . The pooling size was set to 4×4 . For the second layer, 70 filters were used each with a size 2×2 , the pooling size was again 4×4 . For the classification into the 5 gesture classes, an additional hidden layer comprising 500 hidden units connected the two layers with the corresponding output.

We used a batch size of 20 in the learning process, i.e. the weights were updated after presenting 20 images to the MCCNN architecture. The activation function we chose for the hidden units was the *tanh* function and we set the learning rate¹ to 0.1. As a result, we obtained a 70-dimensional feature vector. We call this representation the *complex* feature set. Figure 5.1.2 shows an overview of the MCCNN architecture selected for our experiments.

¹determines the speed of weight update

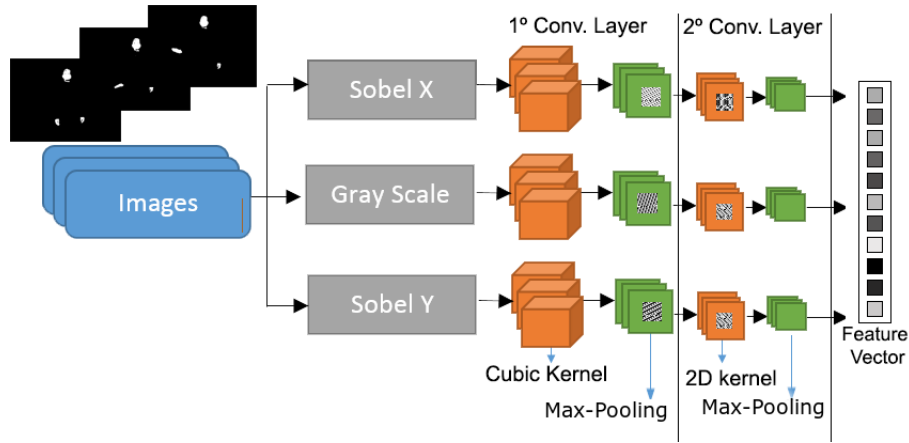


Figure 5.10: A raw gesture sequence is fed to the MCCNN, where each image is processed by three channels. Each channel contains two convolutional layers, followed by a *max* pooling operation.

5.2 Gesture Recognition Experiment

For the experiments we decided to make use of ensemble ESNs, an idea presented first for a speaker identification task in Jaeger et al. (2007). We made use of the ESN toolbox along with the paper. The motivation behind using this scheme was that the experiments should exploit the limits of computations in an ESN when feeding it with different input, here rather simple versus quite complex. Using only one ESN would need a high number of reservoir neurons for the *complex* feature set while using the same setting for the *simple* set would render the task useless either due to overfitting or due to perfect classification. On the other hand, using a single reservoir with only a few neurons for the *simple* feature set would be sufficient, but would clearly not result in the computational capacity needed for the *complex* set. But as we want to compare both sets in terms of the gesture representation and the classification abilities in ESNs, this approach appeared reasonable to us.

We initialized the ESN parameters as follows: we initialized the input W_{in} and reservoir matrices W_{res} randomly in the range $[-1; 1]$. The feedback matrix W_{back} and the noise term ν were set to 0, as we use the ESNs here in a supervised learning task. The input was scaled with a factor of 1.5 and the spectral radius was set below unity. As we were interested in an application for gesture recognition concentrating on the features for the gesture representation, we settled the parameter comparable to the current literature on ESNs.

From the recordings and after postprocessing we obtained 103 gesture sequences. They were then split into disjoint sets $train \cap test = \emptyset$ and randomly

subsampled 75% sequences for the training set and 25% for the test set. The ESNs were created from small leaky integrator ESNs (LI-ESN). Specifically, 500 small ESNs were initialized with the parameters described and used as individual classifiers. The ensemble character is reflected in the way the ESNs were combined. We merged ESNs according to a predefined number of sets. In more detail, the 500 networks can be used separately, yielding 500 classifiers, or joined in equally-sized sets which make up the ensembles. This means that only the division of 500 with integers modulo 0 are allowed. Hence, we used the following set sizes: [1 4 5 10 25 50 100 125 250 500]. A consequence of this scheme is that the classifiers produce varying responses used for the final classification. Thus, the responses from all classifiers in a set were averaged constituting a “vote” (Jaeger et al., 2007).

As outlined in chapter 4, training an ESN basically involves a regression. For the experiments, we used the following scheme (Jaeger et al., 2007): Let \mathbf{s}_n^i denote the i -th state sequence from a training sample set with length l . The vector \mathbf{s}_n^i is composed of the input sample $\mathbf{u}_i(n)$ and the state activations $\mathbf{x}_i(n)$ (cf. also chapter 4 for the notation). Further, let h_c^i be the hypothesis that the i -th sample belongs to the c -th gesture class, where $1 \leq c \leq 5$. Following Jaeger et al. (2007), we chose a small value ℓ , which $\mathbf{s}_i(n)$ will then be equidistantly subdivided with. Simply put, ℓ serves as a factor at which position in the matrix X the states can be collected and put into a vector for subsequent regression.

We created output matrices with the size $l \times c$, where the column corresponding to the desired class contained the value 1 (frame-wise classification). All other classes consequently have zero columns. As the results of a regression are real-valued, a *max* operation was applied on the output for the c classes and the entry of the resultant column (each column codes for a gesture class) was set to 1, all other entries to 0. From this procedure, we computed the number of misclassifications, i.e. the number of wrongly recognized gesture classes.

In our experiment, we focused on the following parameters and their combinations: the reservoir size (#neurons), ℓ , and the leakage rate α as depicted in Table 5.2.

Table 5.2: Range of parameter values

| #neurons | ℓ | α |
|----------|--------|-----------------|
| 4-9 | 3-6 | {0.1, 0.2, 0.3} |

5.3 Results and Evaluation

We ran 30 trials and averaged the classification results across the samples and the different ensemble sizes. For the evaluation, we calculated the mean number of misclassification. We also show the median in the corresponding boxplots to account for the influence of outliers on the mean.

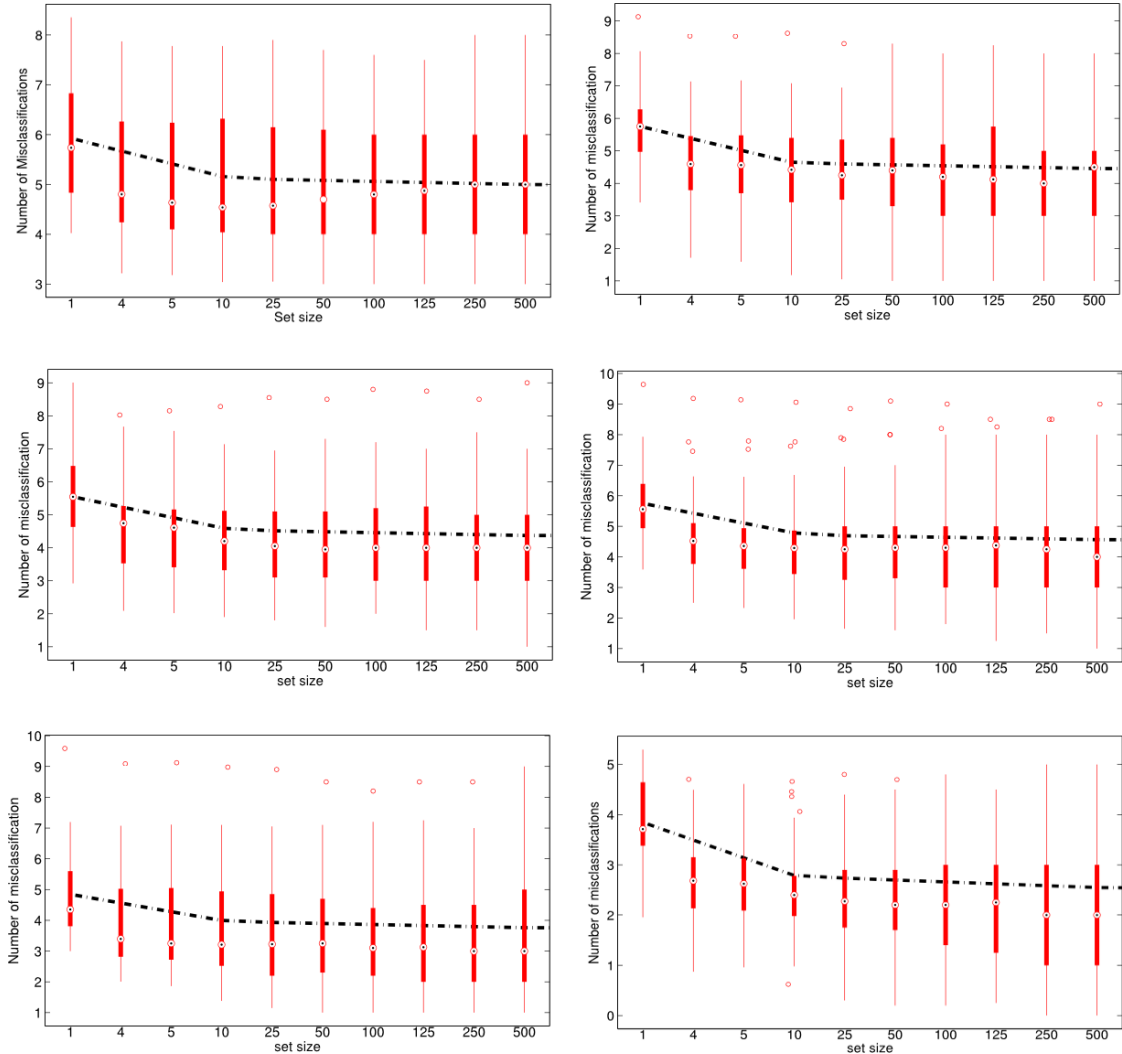


Figure 5.11: Evaluation of the results from the test set using the *simple* feature set over the ensemble sizes. The boxplots show the distribution and the median of misclassifications over the 30 trials with varying reservoir sizes (4 – 9), the leakage rate $\alpha = 0.2$ and $\ell = 3$ remain fixed (from left to right, top to bottom). The dashed line displays the mean classification. Outliers are marked by \circ .

In Figure 5.11 we show the evaluation using boxplots over the different ensemble sizes when varying the reservoir size incrementally from 4 to 9. The leakage rate

was fixed to $\alpha = 0.2$ and $\ell = 3$. Increasing the number of neurons in the reservoirs of individual ESNs decreased the number of misclassifications. Also, the boxplot span, i.e. the minimum and maximum assigned by the boxplot whiskers, reduced significantly. In the first plot of Figure 5.11, the graphs display a minimum from around 3 misclassifications and a maximum number of misclassifications ranging up to 8. With an increasing number of reservoir neurons this interval decreased, however, we observed the occurrence of outliers (the \circ in all plots except the first in Figure 5.11). These outliers negatively influence the mean classification results depicted by the black, dashed line. In contrast, the median of the results is stable around 3 – 4 misclassifications across the set sizes with the best performance.

The single reservoir achieved the best results for the *simple* feature set across all experiments. However, we also observed a higher variance on the results from the 30 trials, depicted by the mentioned whiskers in the boxplot. This effect is less noticeable for the different ensemble sizes. Based on the results, the ensemble produced more consistent responses and thus less varying numbers of misclassifications. The 'hit-rate' on the performance is thus more stable. The single reservoirs delivered the best performance, but the random initialization creates a certain "chance" to get a good performance network.

The variation of the leakage rate α with fixed reservoir sizes and ℓ showed no significant effect on both of these datasets. In Figure 5.12, we show this for a reservoir size=4 and $\ell=4$. The setting for $\alpha = 0.1$ and $\alpha = 0.2$ shows similar results, while for $\alpha = 0.3$ the performance is slightly worse.

The averaged number of misclassifications is rather low for the *complex* feature set, i.e. across the trials, the training error is nearly 0, also across the different ensemble sets (Figure 5.13). The test results showed in the worst case for overfitting (1.5 misclassifications on average). In comparison, the *simple* feature set showed highly varying misclassifications across the experiments. We wanted to investigate the cause of the misclassification and inspected the test results. We identified the *circle* and *turn* gestures, and the *turn* and the pointing gestures were mostly confused with each other.

Our results also display outliers. We identified a positive effect on outlier behavior when increasing the value of ℓ . In Figure 5.14 the effect can be seen for the *complex* feature set when increasing this parameter, while the reservoir size and the leakage rate stay fixed.

We also directly compared the performance of the two representations as shown in Figure 5.15. The leakage rate α and ℓ were fixed to 0.2 and 3 respectively. The first experiment shows the evaluation for a reservoir size = 4 and the second

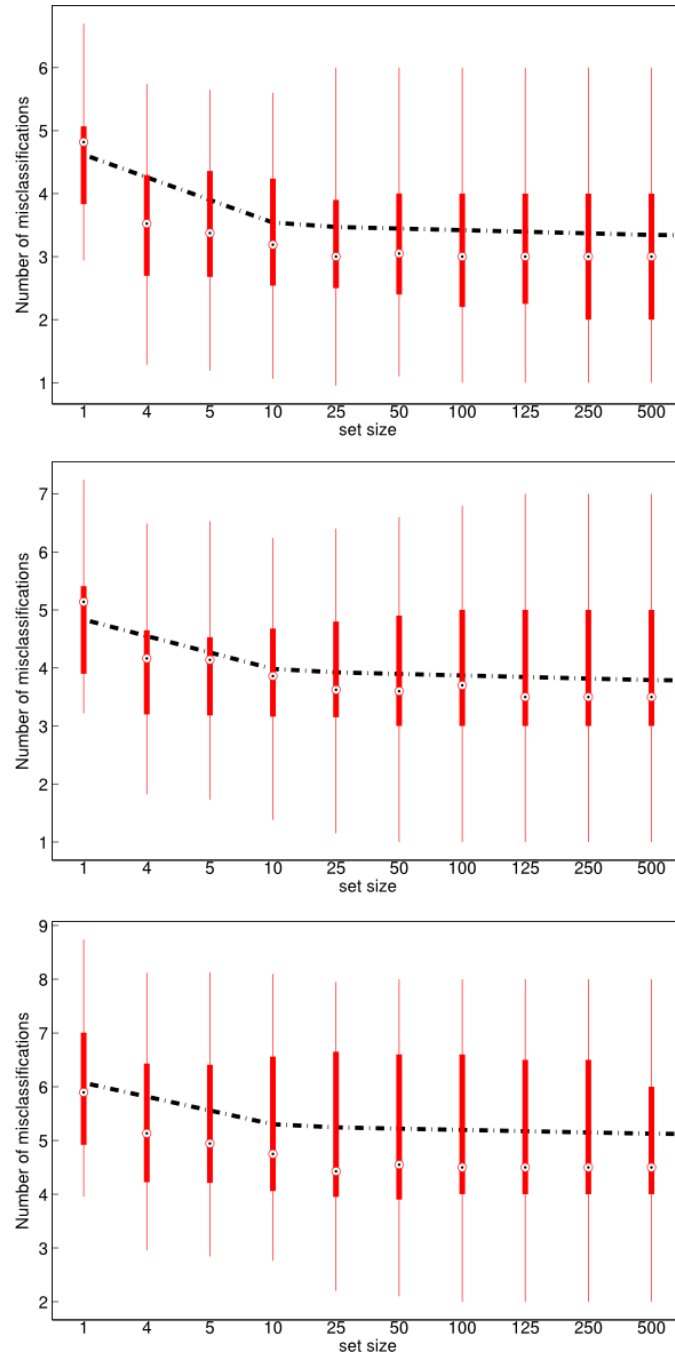


Figure 5.12: Results for a varying leakage rate $\alpha = 0.1, 0.2, 0.3$ (top to bottom) and fixed reservoir size = 4 and $\ell = 4$. There are no outliers.

experiment considered a reservoir size = 9. In the upper image, we observe that for the *simple* feature set the best average number of misclassification is achieved for the single reservoir, displayed by the decreasing trend of the black lines. In contrast, the trend lines in the lower image for the *complex* feature set show good

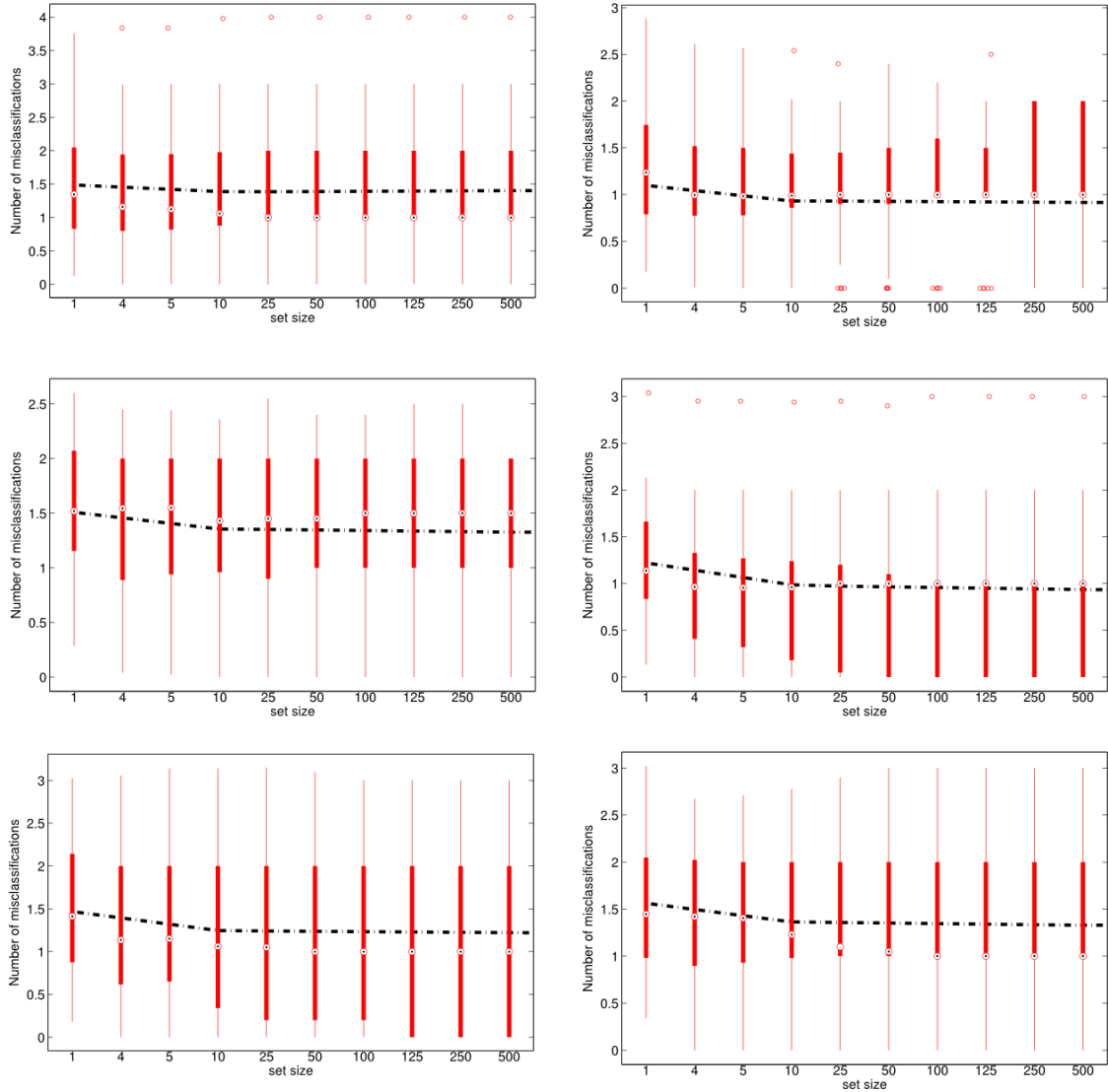


Figure 5.13: Evaluation of the results from the test set using the *complex* feature set over the ensemble sizes. The boxplots show the distribution and the median of misclassifications over the 30 trials with varying reservoir sizes (4 – 9), the leakage rate $\alpha = 0.2$ and $l = 3$ remain fixed (from left to right, top to bottom). The dashed line displays the mean classification. Outliers are marked by \circ .

performance for different combinations of ensemble sizes and an increase in the number of misclassifications for the single reservoir.

5.4 Insights from Single Reservoirs

In the previous section, we showed the results of varying reservoir sizes and leakage rates for ensemble ESNs. As these two parameters are crucial parameters in

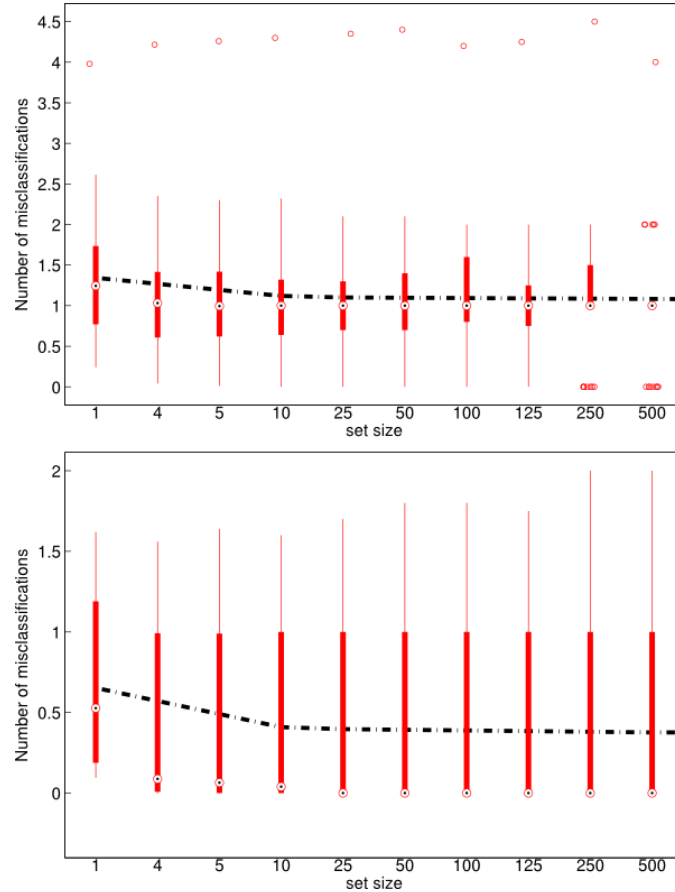


Figure 5.14: Top: Results from the tests of the *complex* feature set for a fixed reservoir size = 5 and leakage $\alpha = 0.1$. For $\ell = 3$, the experimental results display numerous outliers (\circ). Bottom: Incrementing the parameter to $\ell = 4$ shows a smoother picture and improved classification result.

standard ESNs, we want to specifically focus on them in this section. We also refer to the negative influence of similar gesture performances on the classification, also when only parts of the movements are similar (sub gestures). We will investigate this and the discrimination ability of a standard ESN against *gesticulation*, i.e. meaningless arm or hand movements, with the following experimental setup.

We first increased our data by creating gesture sequences from the original *5DG* with variance $\vartheta = 10\%$. This way we doubled the dataset and additionally extended it with *gesticulation* sequences. They consist of randomly subsampled pieces from other gesture sequences of our data, thus containing no structure or consistent gesture trajectory. Introducing a *gesticulation* class has the benefit that unintentional arm or hand movements will not necessarily be classified as a gesture in real scenarios. We also add more complexity to the existing dataset, as the

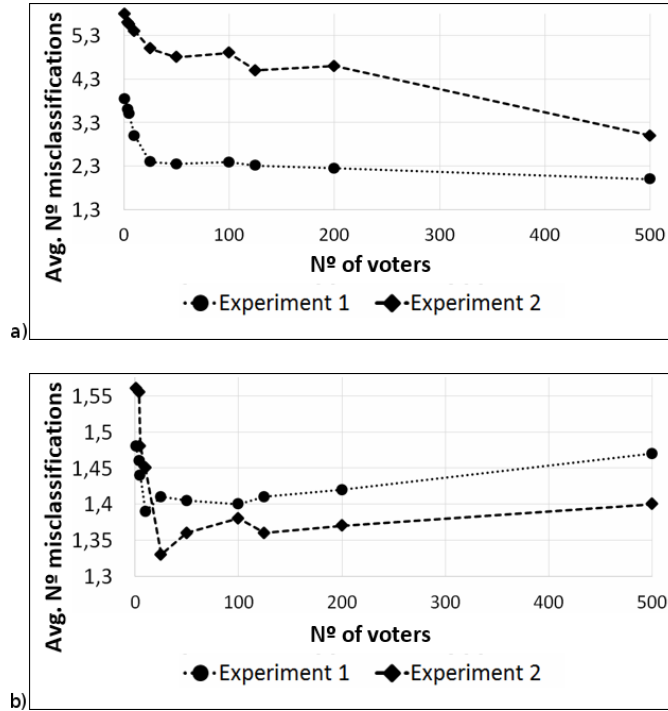


Figure 5.15: Comparisons of both feature sets. a): Evaluation of the experiments using different ensemble sizes for the *simple* feature set. Experiment 1: $\alpha = 0.2$, $\ell = 3$ and $\#$ reservoir neurons=9. The dashed lines show the trend for misclassification. Experiment 2: $\alpha = 0.2$, $\ell = 3$ and $\#$ reservoir neurons=4. b) Evaluation of the experiments using different ensemble sizes for the *complex* feature set with the same parameter configurations as in a).

sequences were not completely random or simply noise, but contain small sequence parts which a classifier can easily be confused with and thus is another challenge. For the experiments, the extended dataset was randomly subsampled with a ratio 2/3 of the sequences for training and 1/3 for testing, i.e. we obtained 157 training sequences and 79 test sequences (236 sequences (103*2+30 gesticulation streams)). We ran 30 trials and averaged the individual results as displayed in the confusion matrices. The letters correspond to the gesture type: C: *circle*, PL/PR: *point Left/Right*, S: *stop*, T: *turn around*, and G: *gesticulation*.

In the ESN literature, it is often stated that increasing the reservoir size leads to a better performance (chapter 4). Our experiments revealed indeed a decrease in misclassifications when doubling the number of neurons (while all parameters are fixed), see Figure 5.16. Despite the fact that parameter tuning is always heavily dependent on the task at hand, Figure 5.17 shows the effect of the leakage rate α , which decreased the number of misclassification. For both experiments, we fixed

the spectral radius $\rho = 0.8$ (cf. Weber et al. (2008)).

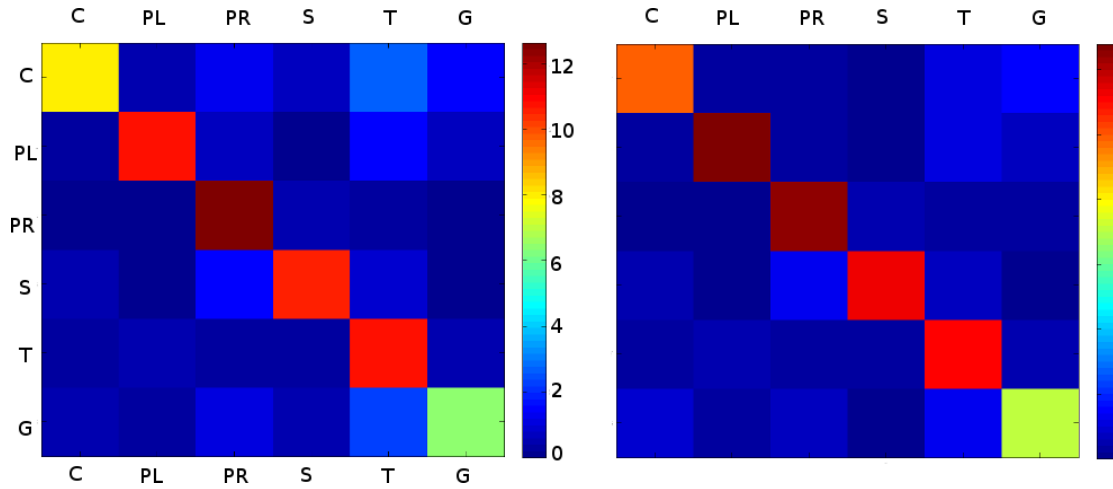


Figure 5.16: Confusion matrix depicting the classification performance on a reservoir with equal parameters ($\alpha = 0.1$, $\rho = 0.8$), but different reservoir sizes. Left: 100 neurons. Right: 200 neurons.

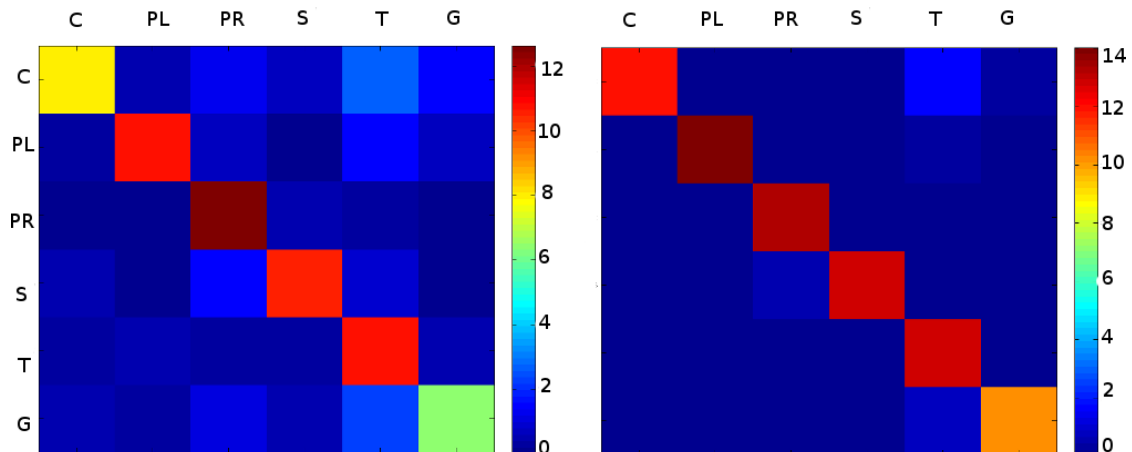


Figure 5.17: Confusion matrix showing the effect of the leakage rate. Keeping the 100 reservoir neurons but increasing the leakage rate from $\alpha = 0.1$ (left image) to $\alpha = 0.3$ (right image) decreases the number of misclassifications.

Figure 5.18 shows the test output of the gesture recognition task on the extended gesture set. The concrete parameter setting is here of minor concern, as the figures' purpose is to highlight the variabilities within gestures. We fed the network subsequently with the gestures ordered as depicted in the figure legend. The y -axis denotes the output activity, where high values denote a clear response to the presented input and less confusion with other gestures. The most significant

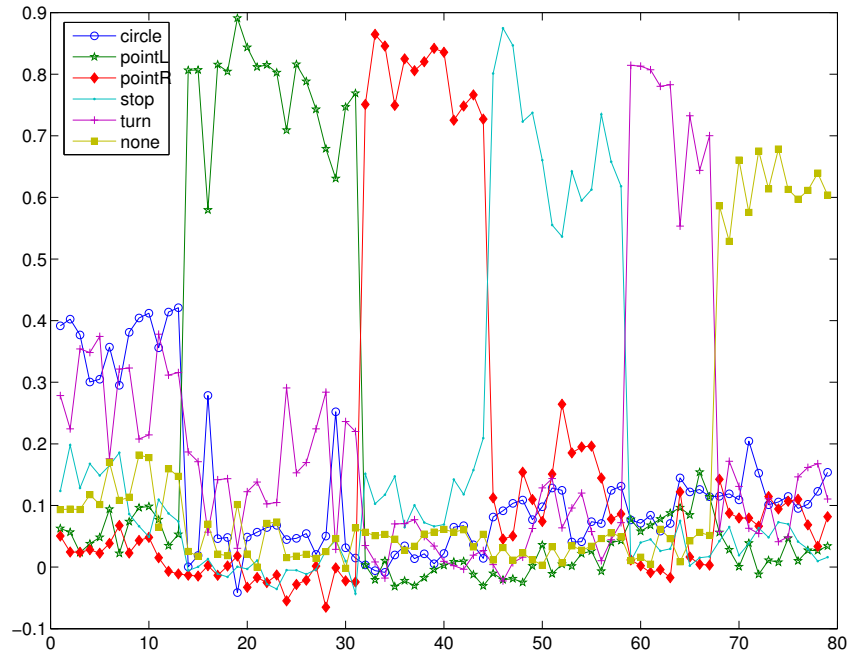


Figure 5.18: Example output and the corresponding activations (y-axis) of test sequences showing the influence of the gestures on each other. The gestures were fed in sequentially (x-axis). The *gesticulation* gesture is abbreviated as ‘none’ in the figure legend.

confusion is revealed between the *circle* (blue curve) and *turn* (magenta curve) gestures. This is explained by the information loss we gain when recording these gestures with a usual camera, i.e. the 2D projection of both gestures is a challenge for a gesture recognition system discriminating the two motions (see Table 5.1). Another effect can be seen between *turn* and *point left* (green curve), and *point right* (red curve) and *stop* (light blue curve). In these cases, one gesture is a sub gesture of the other, i.e. the upwards movement of the *stop* gesture is contained also in the other two gestures. Good discrimination was achieved between the gestures *point right* and *point left*. Also, the *gesticulation* shows a separation from the other gesture classes, despite some influence from the *circle* and the *turn* gestures. This can be explained by the fact that these two gestures are performed more vividly opposed to the other gestures.

5.5 Chapter Summary and Discussion

In this chapter, we presented experiments on a set of command gestures using two different feature representations. We introduced two distinct feature extraction

techniques: one is based on the detection of hands in gesture sequences involving numerous image processing methods. We computed the hand position and the actual hand orientation per frame, which constituted the *simple* feature set. In contrast, we extracted features from gesture sequences using an extension of a convolutional neural network, the MCCNN. We called this representation the *complex* feature set. Our motivation was to investigate the two significantly different representations in a classification with ensemble ESNs, which were introduced by (Jaeger et al., 2007).

Our results showed the best performance of the ESN when merged into one reservoir for the *simple* feature set and hardly any benefit using ensembles. However, the plateau-like black trend lines may be indicators that for this feature set rather small ESNs would suffice for a proper classification. From the insight into individual samples, we believe that the *simple* features and thus probably insufficient representation caused rather confusions among the classifiers. However, we also observed that these single reservoirs displayed a huge variance of the results of the experiments. In contrast, the ensembles displayed higher confusions along the different parameter configurations, but their variances were smaller. We assume that the effects of the random initialization are smoothed across the ensembles, yielding an implicit controlling mechanism,

In our experiments, we chose 30 trials per experiment, but no general rule exists until today how to set this number. Thus, the experimenter faces rather a “chance” of good performance and chooses the number of trials based on a gut feeling or comparable studies in the ESN literature. However, in real world scenarios, we need to rely on stable networks which allow experimental repeatability and plausibility. This leads to the question to what extent we can consider (architectural) design decisions based on the data, input statistics and the knowledge of the parameter interplay, and different reservoir topologies.

The evaluation of the *complex* features showed fewer misclassifications across the experiments, and especially when using ESN ensembles. We explain this effect by the fact that the derived features for this set can be assumed as image coefficients representing significant image parts. Using ensembles provide a mechanism to encode these different representations. This may be an indicator that a different ESN architecture than the standard ESN (cf. chapter 4) may also contribute to the performance, or can at least be a sensible choice for a specific problem. Evidence of ESN variations successfully applied to ESN benchmark data support this view. In some cases, it even offers a better understanding and analysis of the reservoir principles, than exhaustively tuning an ESN promoting its “black box” behavior

to a non-expert.

We also compared directly the performance between the *simple* and *complex* feature set. We showed a notable trend that the *simple* feature set achieved the best performance for the single reservoir, while the *complex* feature set profited from the ensemble structure. We conclude that a standard ESN (i.e. only one reservoir) is a viable tool for gesture recognition based on our results from the *simple* feature set. However, other tasks from the vision domain, e.g. action recognition, often extract high-dimensional features to sufficiently represent sequences of body actions. In this case, the usage of the ensemble techniques may be beneficial to obtain good performance.

Our experiments did not highlight a special role of the leakage rate on the actual performance. As we used a rather small dataset, we suggest performing more investigation on this parameter using a bigger dataset with distinct gesture types varying in their motion.

Also, we detected confusion between gestures which are similar in their performance or contain subgestures. The effect changed slightly when also increasing ℓ , but having both, high values of reservoir neurons and ℓ , resulted in divergent training- and test error.

A more elaborate study on the main parameters in an ESN would also include the influence of the input scaling ι and the spectral radius ρ on the performance. As our main concern was to provide a study on the gesture representations in connection with a classification scheme, we omitted this in favor of the interpretation of our results and comparability to similar studies by Jaeger et al. (2007). However, based on the literature on the spectral radius and computations at the *edge of stability* (EOS) as explained in chapter 3 and chapter 4, we also aim at investigating under which conditions the reservoir produces valid results for different sequences, and which role the EOS hypothesis plays. We will deepen this topic in the next chapter.

Chapter 6

Recurrence Analysis for Gesture Sequences and the Reservoir

In chapter 4, we outlined the significant parameters and conditions for the proper functioning of Echo State Networks. We also discussed that due to the variety of tasks both in complexity and applicability no universally valid ESN configuration exists that applies to all of them. A significant parameter in every ESN architecture is the spectral radius ρ which from a theoretical perspective is an indicator for network stability (sufficient condition, cf. chapter 4). The correct setting of this parameter is often debated in the research community, and studies we outlined in chapter 4 showed evidence that different tasks need a different setting of the spectral radius. It is, therefore, common practice to use any search algorithm to tune the network for a specific task. An opposing trend, however, aims at introducing methods to characterize the ESN processing capabilities which may substitute tweaking the algebraic properties of the reservoir matrix.

In this chapter, we particularly address the stability issues in ESN for gesture data. For the understanding of the dynamics in a system, and whether it is in a stable or unstable state, it is crucial to investigate the time evolution of network states, usually done by an approximation of the Lyapunov exponent¹. Wolf et al. (1985) suggested a scheme using perturbations with a specific noise level. To determine the stability in an ESN, the convergence behavior of two activation state trajectories x and $x' + \varepsilon$ (ε is some noise) would be measured. The system is said to be stable, when the initial and the perturbed state sequences converge, showing that they are independent of their initial condition. If the displacement between these two trajectories is exponential, the system is referred to as chaotic (but may

¹as a reminder, the Lyapunov exponent is defined in the limit ∞

also exhibit oscillations or limit cycles). In the latter case, the (approximated) Lyapunov exponent is positive. A novel scheme introduced by Verstraeten and Schrauwen (2009) focused on the Jacobian matrix of the reservoir. Their results not only supported that the constraint on the spectral radius (i.e. $\rho < 1$) might not hold in all application cases, but also that their so-called Local Lyapunov exponent (LLE) serves as a better performance predictor than the reservoir properties like the spectral radius or the singular value.

A methodology to investigate different state behavior is a significant research topic in complex system sciences. The conception of *recurrence plots* (RP) (Eckmann et al., 1987) allowed the construction of phase portraits from distinct time-series using the embedding-delay technique. These specific plots enabled further the computation of numerous characteristics of the underlying system, culminating in the *recurrence quantification analysis* (RQA)(Webber and Zbilut, 1994). Using these two approaches provide a visualization technique and an analysis tool for the investigation of the reservoir.

We explain the construction of RPs with different gesture sequences, both from our originally recorded set introduced in the previous chapter (5DG) and variants computed from them. We proceed with the introduction of the RQA measures determined from the RPs. Especially interesting are those quantities which indicate a transition from a stable to a chaotic system behavior. We investigate their effect on variously initialized reservoirs and propose a criterion which hints to this transition.

For the experiments, we used the ESN toolbox (Jaeger, 2002) and the CRP library (Marwan et al., 2002, 2007). This way, we also validated results demonstrated in Bianchi et al. (2016a).

6.1 User-Independent Sequences

Before going into detail, we describe first the sequences used in this chapter and motivate our procedure. To determine the dynamical properties in the reservoir we created gesture sequences of different level of difficulty. We point out, that for gesture recognition it is common to also use user-independent sequences as is the case here. The reason is that we are interested in the time-series themselves in connection with the principles of the reservoir and their corresponding analysis rather than performing a study keeping into account the possible different experiences or background from different users in a gesture performance scenario.

As described in the previous chapter, we postprocessed the resultant images

from the gesture recordings and deleted frames after their thresholding contained no information. To keep these frames resulted in deterioration of the gesture trajectories, thus this procedure did not introduce any information loss, but we got reliable sequences for further input into a network. An example sequence and the result of applying low-pass filters is depicted in Figure 6.1. For the construction of the phase portrait however, we want to assure a clean representation of our data and a reasonable interpretation. Therefore, we additionally filtered our gesture sequences from the 5DG recordings (see Appendix B), resulting in the 5DG+E basic set. The *gesticulation* gestures were not filtered because their randomness is a desired effect in our dataset.

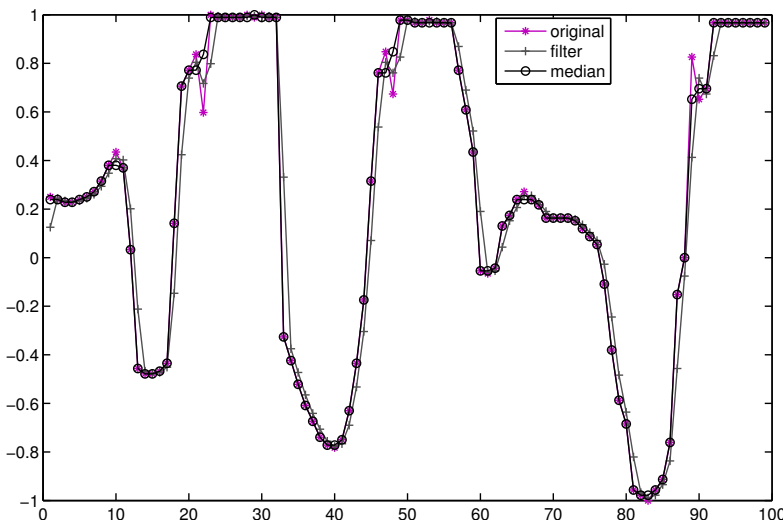


Figure 6.1: A *circle* gesture sequence showing the horizontal movement from the *5DG* set (magenta) compared to two filter methods used in our analysis. The effect of the filter is mainly smoothing irregularities between sample points introduced by noise or postprocessing.

To achieve some variability on our gestures to account for the gesture performance variations, we introduced the $\vartheta\%$ which denotes the amount of variance from which we created new sequences (5DG+E). Again, from experimentation with different configurations, we chose $\vartheta = 10$, which produced reasonable noisy sequences, i.e. a trade-off between still structured but also challenging sequences.

Finally, we also perturbed our data from the extended dataset with Gaussian noise², the 5DG+E noise. We provide a summary of the datasets and operations on them in Table 6.1. Figure 6.2 shows the filtered, original and noisy version of the x-direction from the *circle* gesture.

²A form of white noise, but of course other noise variants are also possible

Table 6.1: Gesture datasets

| Dataset | Operation |
|-------------|--|
| 5DG | Preprocessing |
| 5DG+E | Extension of 5DG set Doubled gestures by $\vartheta\%$ variance + Gesticulation |
| 5DG+E basic | Lowpass-filtered data to serve as basic signal set |
| 5DG+E noise | Addition of Gaussian noise |

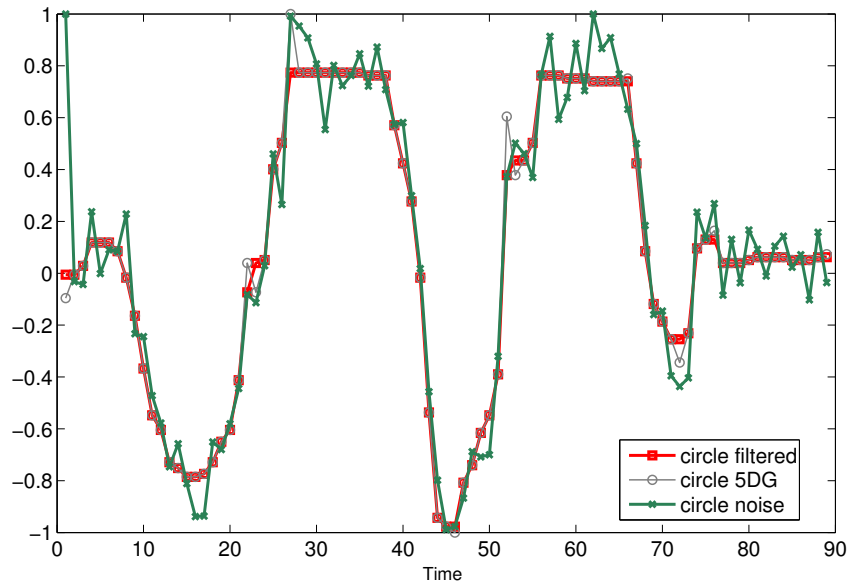


Figure 6.2: Example of a *circle* gesture from the *5DG+E* set: the original version (red), the filtered (green), and noisy version (gray) from this sequence.

6.2 Visualization and Dynamics

In the following, we want to explore gesture data for further analysis of their underlying properties employing recurrence plots (RP). Their usage as a visualization tool is further motivated by the fact that dynamical invariants can be computed from the line distribution in an RP, known as Recurrence Quantification Analysis (RQA). The measure links the predictability of a system, indicating stable or unstable behavior (chaotic or stochastic). The latter characteristics can also be

exploited in the reservoir, connecting thus the stability analysis with the RQA.

In the *Reservoir Computing* community it is often stated that the performance is highly dependent on a specific task, thus report of optimized parameter configurations should be critically evaluated. Input scaling, for example, is necessary to provide the network with nonlinearity in case the input has values around zero; in contrast for high-amplitude data, the scaling leads to the saturation area of the commonly used *tanh* activation function, yielding a network with binary switching power only.

Therefore, the input fed into the network and the network design should not be uncoupled but analyzed from both sides. Visualization techniques can help to identify the network behavior given the input relevant to the task and to clarify the parameter interplay. For some common network architectures, such visualizations are well established. For deep neural networks such as those resembling the hierarchical processing of visual stimuli, it helps to uncover which features actually emerged from the cascades of convolution filtering and pooling operations. Also, for classic RNN the usage of heatmaps can be a useful tool detecting the weight organization during or after training. While working on the thesis, we asked whether it is possible to show the insights of the reservoir for different settings, especially when considering criticality.

6.2.1 Phase Space Reconstruction and Recurrence Plots

For the visualization of data exhibiting different characteristics, Eckmann et al. (1987) introduced the Recurrence Plots (RP) which show the time evolution of states in the data phase space. For instance, if the underlying data reveals periodicity the phase space will highlight similar trajectories shifted in time. In contrast, chaotic time-series the trajectories diverge and thus show rather irregular patterns in an RP with only (see chapter 4). In the case the RP shows only spurious lines and points, the underlying data comes from a stochastic source. In general, the RP serves as a visualization of a specific time-series and thus further analysis is mandatory before interpreting and classifying data into one of these categories.

As an ESN is a dynamical system, we can investigate the gesture representation in the reservoirs phase space in terms of recurrence and the significant parameters driving the dynamics.

Before going into detail we provide the basic computational steps to derive the visualization in a recurrence plot and the representation in the corresponding phase space. Figure 6.3 sketches a trajectory in the phase space. A predefined neighborhood ϵ assigns the recurrence of two trajectories visiting a similar point

in this space.

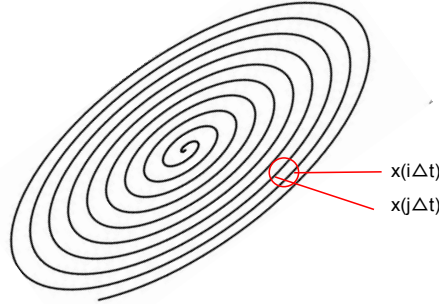


Figure 6.3: Illustration of recurrence in an idealized phase space for two trajectories $x(i\Delta t)$ and $x(j\Delta t)$ (Δt is the time interval) determined by the radius ϵ (red ring). If this value would be too small, only one line would be detected and thus no recurrence. In contrast, setting this value too high would include more lines and may distort the interpretation of dynamics.

From this intuitive idea, the recurrence plot can be computed as follows:

$$R_{i,j} = \Theta(\epsilon - \|x_i - x_j\|) \quad x_i, x_j \in \mathbb{R}^m \quad x_i, x_j = 1 \dots N \quad (6.1)$$

The norm usually taken is $\|\cdot\|_2$ but any other norm can be used as well. The Heaviside function Θ evaluates every point $R(i, j)$ to either 1, assigning there is a recurrence and 0 in case there is no recurrence given a threshold ϵ :

$$R_{i,j} = \begin{cases} 1: & \vec{x}_i \approx \vec{x}_j \\ 0: & \vec{x}_i \not\approx \vec{x}_j, \end{cases} \quad (6.2)$$

In an RP this is depicted by a black dot for 1, and a white dot for 0. These dots are formed into lines in case there is a trajectory in the phase space, as is the case for e.g. periodic signals. The RP is symmetric, i.e. $R_{i,j} \equiv R_{j,i}$ for a fixed threshold ϵ . The choice is crucial for the interpretation of an RP: a very small value, i.e. a narrow ϵ -neighborhood, may be too restrictive and produces an almost empty RP (because then even close recurrent points may not be included and set to 0). High values, and therefore a rather generous ϵ -radius, overemphasizes recurrence point relations and is sensitive to data noise, misleading subsequent interpretations. The determination of criteria how to set ϵ more flexibly is an important subject in current research but goes beyond the objective of this thesis.

The line of identity (LOI, R_{ii}) is the main diagonal and longest line in the RP. This is intuitively clear as every point is maximal close to itself and therefore always included in the computations of equation 6.1. Interpretations on the pattern displayed by the RP consider primarily the diagonal structures parallel to it (periodic signal), vertical blocks (laminar states) as well as small diagonal fragments (chaotic behavior). In case the underlying time-series is time-independent or, respectively, uncorrelated the RP shows no regular or line pattern. An example for an independently and identically distributed (i.i.d.) signal is depicted in Figure 6.4.

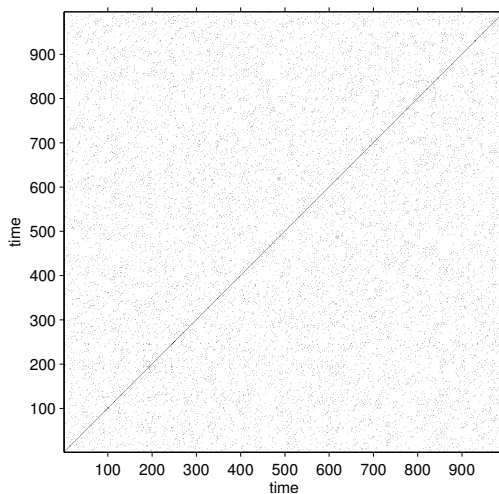


Figure 6.4: Recurrence plot of an i.i.d. time-series with $\epsilon = 0.3$. The RP exhibits no regular line patterns but only single dots due to the uncorrelated nature of the input signal.

In contrast to a stochastic signal, which has consequently no inherent structure, a deterministic chaotic system is characterized by short recurrence times in the phase space, caused by the high divergence of trajectories. The RP displays this behavior with a division of small fragments of diagonal lines and white compartments, which represent areas of no recurrence. An example of the chaotic Lorenz system, we introduced in chapter 4, is shown in Figure 6.5.

Takens theorem (Takens, 1981) ensures finding an appropriate embedding for the phase space reconstruction and is usually used for the construction of an RP. In general, for any univariate time-series x time-delayed copies from the original system with an appropriate embedding can be computed as follows:

$$x = (x(i), x(i + \tau), \dots, x(i + (\mu - 1)\tau)) \quad (6.3)$$

where μ is the embedding parameter and τ determines the lag.

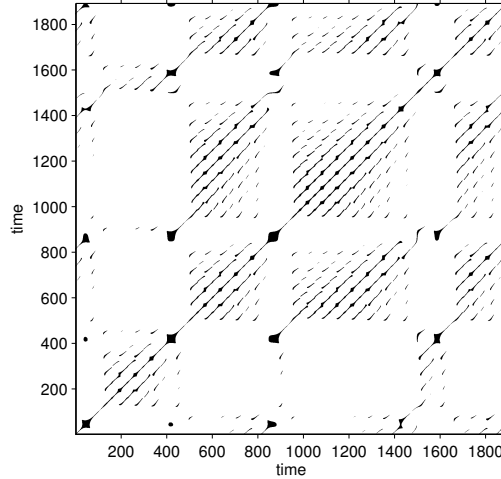


Figure 6.5: Recurrence plot of the Lorenz system with an embedding dimension $\mu = 3$ and delay $\tau = 4$. In contrast to the irregular pattern in Figure 6.4, a chaotic system exhibits small diagonal lines along the LOI. This indicates recurrences for only a few times in the phase space, alternating with white compartments.

We used the first zero-crossing of the autocorrelation function for the estimation of the lag parameter τ (an example is shown in Figure 6.6).

The result of the embedding-delay technique is a μ -dimensional trajectory of the time-series in the phase space. The RP accordingly then represents a 2D representation of the μ -dimensional orbit produced by the underlying dynamical system. This visualization allows inspection of large and small-scale characteristic of the dynamical system, which had basically produced this time-series (in a later section, we will also show how properties from the RP can be quantified).

The embedding dimension μ can then be computed using the False Nearest Neighbor algorithm (Kennel et al., 1992):

$$v_{fnn} = \frac{\|x_{i+1} - x_{j+1}\|}{\|x_i - x_j\|} > r_{tol} \quad (6.4)$$

where the point-wise distances of two vectors are calculated and compared to a predefined threshold r_{tol} . The value v_{fnn} is the percentage of false neighbors along the dimension μ , which is taken as embedding value when v_{fnn} reached zero. An example is depicted in Figure 6.7.

The procedure iteratively tests different values μ_k for embedding the time-series and computes the distance change over given k . If points get detached significantly for dimension μ_{k+1} , they are revealed as false nearest neighbors revealing that the underlying attractor is insufficiently unfolded for dimension μ_k . In an RP this can be seen e.g. when lines are orthogonal to diagonals. The embedding dimension

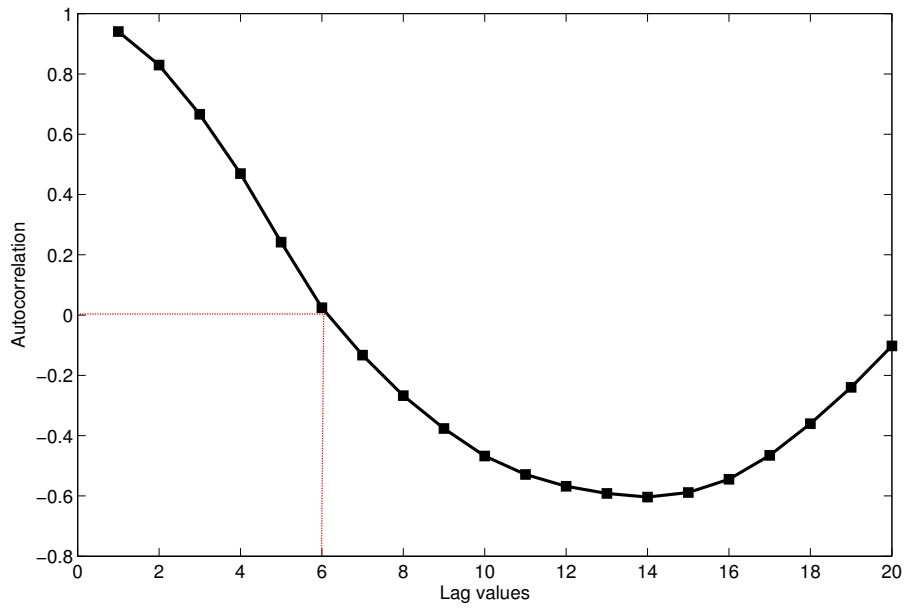


Figure 6.6: Estimation of the delay parameter using the autocorrelation function yields $\tau = 6$.

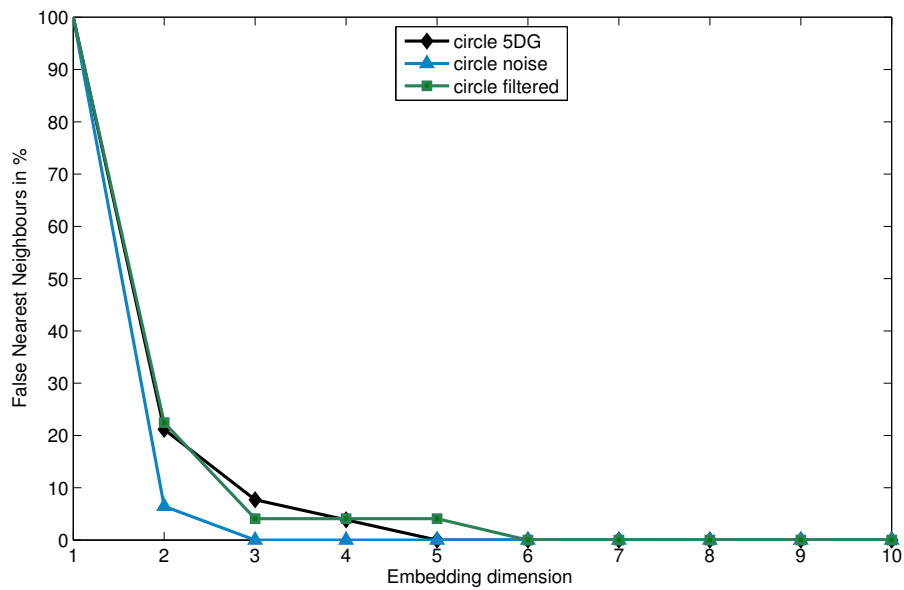


Figure 6.7: Estimation of the percentage of presumably nearest neighbors for determination of the embedding parameter μ . We chose the *circle* sequences shown in Figure 6.2: the original sequence derived from the *5DG* recordings, and the filtered, respectively, noisy version from that sequence considering the *x*-direction.

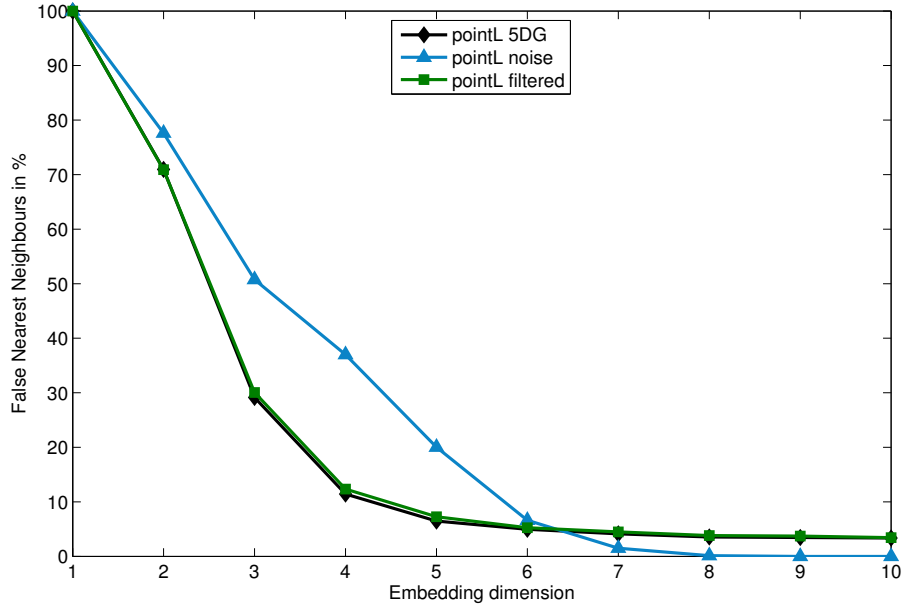


Figure 6.8: Estimation of the percentage of presumably nearest neighbors for determination of the embedding parameter μ for *point left*.

has to be large enough to unfold the geometric structure of the attractor, resulting in $\mu \geq 2n + 1$ (as this is a sufficient condition, the embedding value can also be less than stated). So, the embedding technique preserves the original topology of the underlying n -manifold.

With these parameters, we are able to construct the recurrence plots (visualizations CRP library (Marwan et al., 2002)). For all plots, we used the Euclidean distance. Figure 6.9 and Figure 6.10 show two gesture examples from our dataset (5DG), which differ in their motion characteristics and are thus suitable to demonstrate how recurrence plots can identify the different states over time or the occurrences of noise. We chose sequences from the *stop* and the *turn* gesture and plot 1000 sample points. The RP of the *stop* gestures displays a periodic structure with 4 prominent laminar states between at the time-instances 450, 600, 750, and 900. Laminar states occur whenever there are no changes. For the gesture, this is plausible, as the only motion for *stop* considers the arm (lift up and put down). The concrete *stroke* phase mostly consists of holding the hand in front of the camera. In contrast, the recurrence plot *turn* gestures shows periodic patterns with higher frequencies, i.e. the lines are less distant to each other than in the plot for the *stop* gesture. The laminar states assign start and end phases of the gesture (hands are not moving, cf. gesture settings described in chapter 5). The recurrence plot of the filtered *turn* gesture sequences reveal better the periodic compartments (Fig-

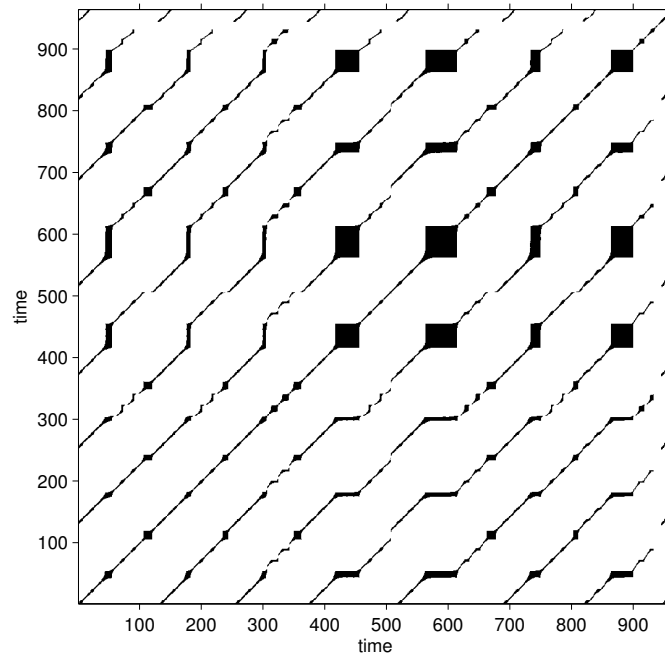


Figure 6.9: Recurrence plot of a sequence of *stop* gestures (y-direction, $\mu = 3$, $\tau = 18$, $\epsilon = 0.2$). Due to the low movements in the gesture performances, the plot displays many laminar states (black blocks).

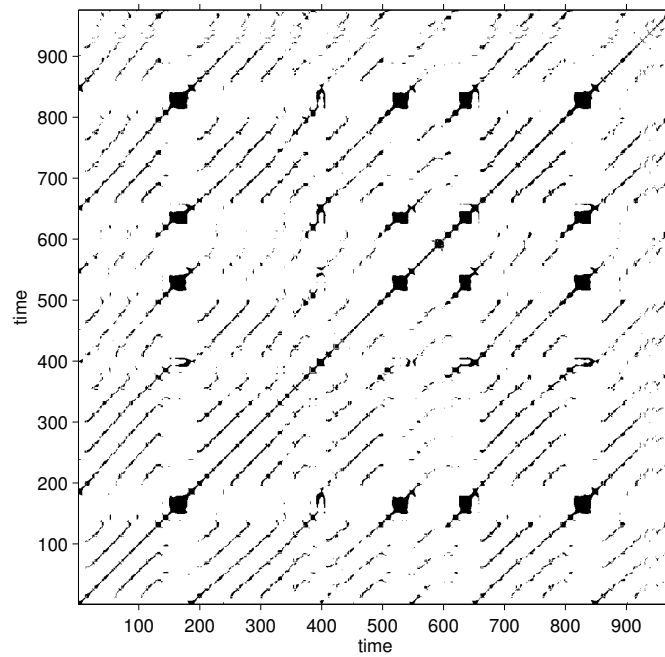


Figure 6.10: Recurrence plot of a sequence of *turn* gestures (x-direction, $\mu = 6$, $\tau = 12$, $\epsilon = 0.7$), which exhibit more movements displayed by the smaller distances between the lines in contrast to the *stop* gestures.

ure 6.11). In contrast, the noisy version of these sequences shown in Figure 6.12 display almost no periodic structure (we used the same values for the embedding μ and delay τ as for the original *turn* gesture in Figure 6.10).

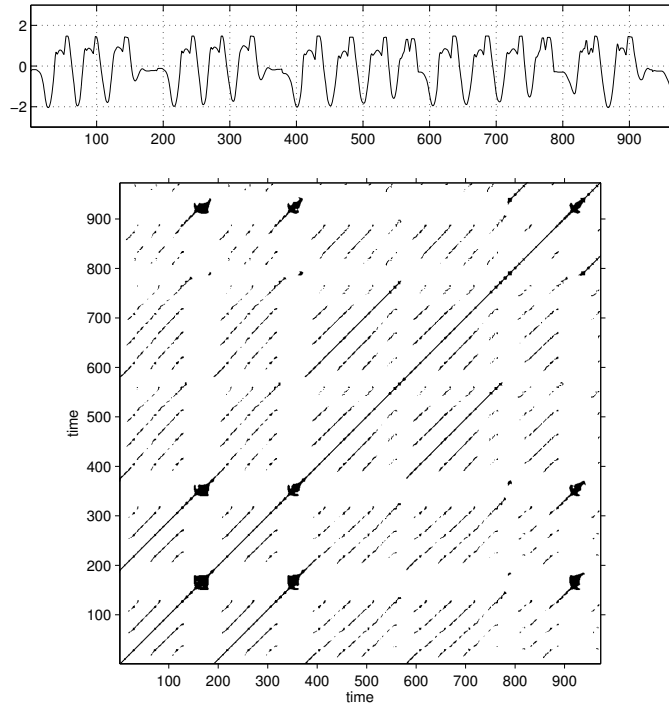


Figure 6.11: Recurrence plot of a sequence of filtered *turn* gestures shown in the upper plot (x-direction, $\mu = 8$, $\tau = 5$, $\epsilon = 0.7$), which exhibit more movements displayed by the smaller distances between the lines in contrast to the *stop* gestures. The high value of $\epsilon = 0.7$ is chosen for a better visualization.

The recurrence plots (RP) highlighted periodic segments, however, the plots display also spurious lines which come from noise or other fluctuations in the sequence. Thus, the motion profile of especially the *circle* and *turn* gestures may not be periodic, which can have an influence in prediction or classification tasks, when the data is used for input. For a qualitative comparison, we also selected the *circle* gesture recorded from a Kinect device, repeatedly performed within one sequence (Parisi et al., 2014). The gesture was represented using a skeleton model and the feature set comprised five features per frame (motion in x, y, z directions, distance and angle between head and hand). We upsampled the trajectory by interpolating the sequences with a median filter (factor 3) to compensate for the windowing mechanism introduced in Parisi et al. (2014), where only every 3rd frame was selected as input into a classification stage. In Figure 6.13 the movement in the x -direction is shown. The time-series depicts an irregular trajectory, which is also visible in the corresponding RP.

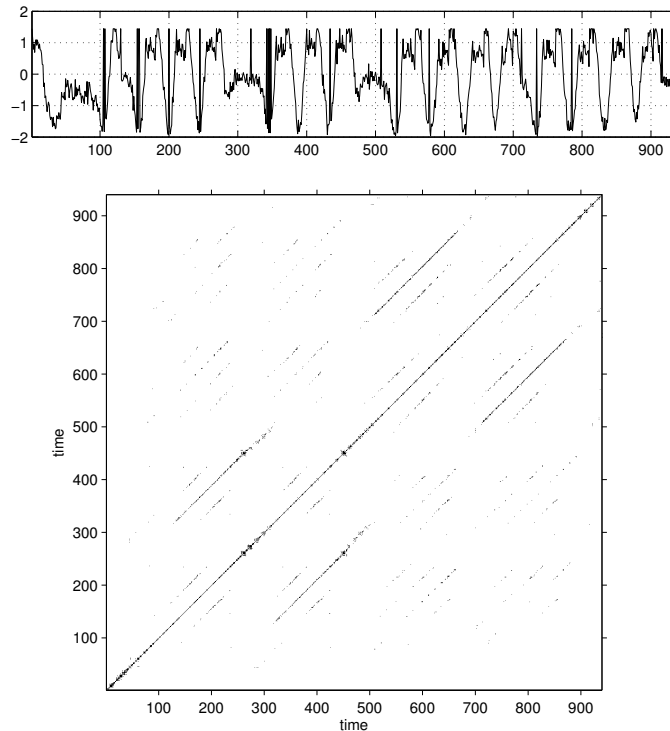


Figure 6.12: Recurrence plot of a sequence of *turn* gestures shown in the upper plot (x-direction, $\mu = 6$, $\tau = 12$, $\epsilon = 0.7$) reflecting the influence of noise. Almost all periodic structure is lost.

Summarized, we introduced the recurrence plots (RP) as a viable tool to investigate gesture sequences varying in performance (e.g. *turn* vs. *stop*) and level of signal properties, from smooth, filtered to noisy sequences. While RPs are used in many fields especially life sciences (Marwan, 2008), to the best of our knowledge that was not considered in any gesture analysis before. We assume that the movements of gestures, the same as for actions, are considered to be mainly periodic. However, we showed how from recordings that this is not necessarily the case, especially when taking into consideration sensor noise and presumably poor preprocessing of the data. Investigating the RPs for data is a valuable tool in the data (pre-) processing as it enables visualization of suspicious subsequences which may have an influence on a subsequent prediction or classification task.

In addition, the RPs provide several measures by considering the specific line distribution of a time-series. The *Recurrence Quantification Analysis* (RQA) (Marwan and Kurths, 2005) provides a method to quantitatively specific measures derived from the RPs (e.g. the recurrence rate) and relates also to dynamical invariants like the correlation entropy (Grassberger and Procaccia, 1983). In the discussion whether the embedding procedure has an influence on the estimation

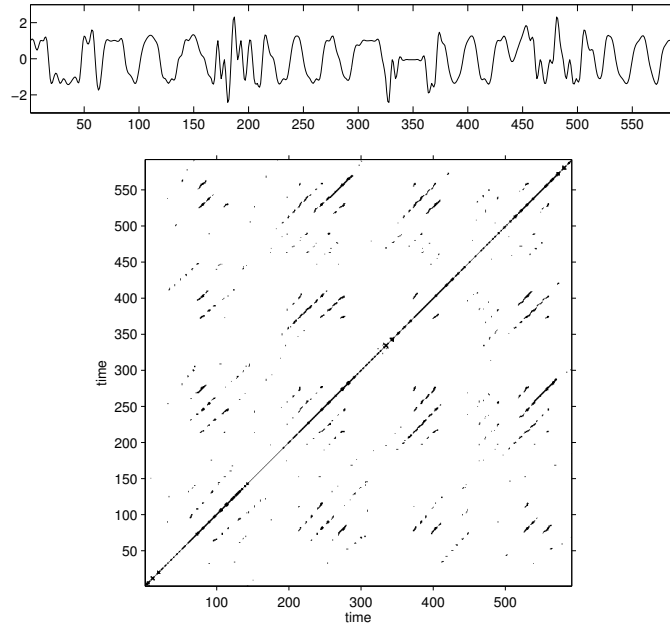


Figure 6.13: Top: Time-series of the motion from a *circle* gesture recorded with the Kinect device. Bottom: According RP of the time-series. The embedding parameters were $\tau = 7$ and $\mu = 5$. Although periodic in nature, the time-series reveals an irregular pattern of the gesture, which is performed several times within the sequence. The plot shows spurious lines, which can be interpreted as a chaotic structure (but a careful analysis should be given). The RP was thresholded with an $\epsilon = 0.7$.

of dynamical invariants was investigated by the work of March et al. (2005). The authors applied their scheme to the mutual information between two variables derived from geomagnetic data and showed the effect in a cross recurrence plot (CRP) visualizing the areas in such a plot contributing to it. From their analytical derivation of a new formulation of the both quantities using the correlation sum, the authors showed evidence that unembedded RPs made available all information and thus support the work of Iwanski and Bradley (1998) that the embedding dimension μ has no effect on dynamical invariants.

6.2.2 Recurrence Quantification Analysis

The RP depicts a specific distribution of a time-series comprising black lines, blocks or dots to assign recurrence, laminar states or, respectively, individual spots, and white compartments which show that the system evolution does not return to this part of the phase space. To determine some quantitative measures from the pattern Webber and Zbilut (1994); Marwan and Meinke (2004) introduced the following:

- *RR*: Recurrence Rate

$$RR = \frac{1}{N} \sum_{i,j=1}^N R(i, j) \quad (6.5)$$

- *DET*: Determinism

$$DET = \frac{\sum_{l=l_{min}}^N l(P(l))}{\sum_{i,j=1}^N R(i, j)} \quad (6.6)$$

where $P(l)$ is the distribution of diagonal lines of length l :

$$P(l) = \sum_{i,j=1}^K (1 - R_{i-1,j-1})(1 - R_{i+l,k+l}) \prod_{k=0}^{l-1} R_{i+k,j+k} \quad (6.7)$$

Given that $DET \in [0; 1]$, a value close to 1 indicates good predictability for a system, while a low value close to 0 hints to a chaotic or stochastic system.

- L_{max} : longest diagonal line distribution except the LOI

$$L_{max} = \max\{l_i\}_{i=1}^{\mathcal{L}_l}, \mathcal{L}_l = \sum_{l \geq l_{min}} P(l) \quad (6.8)$$

where L_{max} refers to the longest diagonal line except the line of identity (LOI)

- *DIV*: divergence

$$DIV = \frac{1}{L_{max}} \quad (6.9)$$

$DIV \in [0; 1]$ and is maximal for chaotic systems, as recurrence in the state space is unlikely opposed to stable systems, where recurrent points within a ϵ -radius have a probability of 1.

- *LAM*: laminarity

$$LAM = \frac{\sum_{v=v_{min}}^K v(P(v))}{\sum_{v=1}^K vP(v)} \quad (6.10)$$

where $P(v)$ is the amount of vertical lines:

$$P(v) = \sum_{i,j=1}^K (1 - R_{i,j})(1 - R_{i,j+v}) \prod_{k=0}^{v-1} r_{i,j+k} \quad (6.11)$$

and v_{min} is the minimal vertical line length chosen for the *LAM* computation. Laminarity occurs when the states do not or only slowly change. That is why

Table 6.2: Some RQA measures for a *circle* gesture

| | RR | DET | LAM | L_{max} | TT |
|----------|--------|--------|--------|-----------|--------|
| filtered | 0.1318 | 0.7791 | 0.8634 | 13 | 5.9799 |
| original | 0.1236 | 0.7293 | 0.8161 | 12 | 6.3200 |
| noisy | 0.0552 | 0.2083 | 0.3079 | 3 | 2.0781 |

Table 6.3: Some RQA measures for a *point left* gesture

| | RR | DET | LAM | L_{max} | TT |
|----------|--------|--------|--------|-----------|---------|
| filtered | 0.2672 | 0.9866 | 0.9959 | 124 | 14.1522 |
| original | 0.2672 | 0.9852 | 0.9949 | 124 | 14.0416 |
| noisy | 0.2259 | 0.6083 | 0.7315 | 19 | 3.4496 |

this measure is related to the trapping time TT , which computes the average vertical line length:

$$TT = \frac{\sum_{v=v_{min}^N} v P(v)}{\sum_{v=v_{min}^N} P(v)} \quad (6.12)$$

Inspecting the line distribution yields dynamical invariants like the correlation entropy, being a link to the Lyapunov exponent by providing a lower bound of summed positive exponents (as a reminder, a positive Lyapunov exponent indicates instabilities in the system up to chaos). The measure DIV has a special role when thinking about how trajectories behave in the phase space. As we described in chapter 4, trajectories diverging exponentially fast reveal the underlying dynamical system to be chaotic (as an example we introduced the Lorenz system). Therefore, that measure can be related to the Lyapunov exponent (LE), but both quantities should be distinguished. In an RP this is depicted by small diagonal lines, showing that trajectories recur only for a small time interval.

For consistency, we performed an RQA on the *circle* gesture sequences shown in Figure 6.2. The advantage of RP and RQA is, that even short time-series can be used for an analysis (Marwan, 2011). We extracted some of the measures to show the effects of the altered version of the original gesture from the $5DG+E(\vartheta = 10\%)$ set, which is depicted in table 6.2. For the recurrence rate (RR) we see a drop from the filtered to the noisy version of the time-series. More important, the determinism DET decreases significantly and also $L_{max} \setminus LOI$. That result follows from the characteristics of the data, as the filtered and preprocessed streams yield a smooth trajectory and exhibit periodicity. This, in turn, leads to a predictable

pattern, opposed to the noisy version, where the phase portrait yields smaller diagonal lines according to the L_{max} and states are less likely to recur. This prevents a consistent pattern detection and hence decreases the predictability. As was outlined above, the L_{max} has a reciprocal relationship with DIV ; clearly, the longer states run in parallel, resulting in a high value for L_{max} , the less the trajectory diverges, and vice versa. Therefore, while in the first two data series DIV is quite low, the value increases for the noisy version. Besides the diagonal line distribution, also the vertical spacing and blocks in an RP play a role, expressed in the laminarity LAM (%) and trapping time TT (average number of vertical lines), where the noisy time-series shows the smallest value. The evolution over the time is, in this case, not (literally) trapped.

When considering the *point left*, *point right* and *stop* gestures we have to point out that their intrinsic dynamics are rather low. This is not surprising, as several datasets usually contain both lively and rather moderate movements, e.g. the KTH action dataset comprising *running* but also *hand clapping* and *boxing*. One challenge for a classifier is then to cope with the varying motion elements while at the same time representing and learning them sufficiently. For our scenario, we found that especially these three gestures overlap in motion and thus cause misclassifications (see chapter 5). In this particular section, however, we are interested in the characteristics they exhibit for the RQA. As mentioned, for the extracted time series the movement elements are quite low, so we decided to capture half of the sequences from the datasets explained above. As the start and end position is similar in all sequences, we concatenated the sequences.

For the *point left* gesture we used half the sequences from the $5DG + E$ ($\vartheta = 10\%$), which yielded in sum a time-series of 1535 sample points. The results are reported in table 6.3. As for the *circle* gesture, we compared the selected measures on the original sequences and their modification, the filtered and the noisy version. The determinism DET is higher than for the *circle* gesture, which is explained by the mentioned small movements in the gesture. Observations of the sequences of pointing gesture reveal a periodic-like pattern of different times, which is highly predictable. The significant effect of noise can then be seen from the drop to $DET = 0.6083$. In line with the observations from the *circle* gesture a decrease in the laminarity LAM , trapping time TT and the L_{max} is recorded. Please note, that due to the processing of a longer sequence the value for L_{max} may appear large, opposed to considering a rather short time-series.

6.2.3 Reservoir Dynamics

An important topic in current research is to shed light on reservoir dynamics and stability connected to network performance. The main hyperparameter for investigating the stability is the norm of reservoir weights bounded by some criteria we outlined in chapter 4. One significant property in ESN is the echo state property (ESP). However, some insistent misunderstandings about the ESP still exist (Yildiz et al., 2012; Caluwaerts et al., 2013). For clarification on this topic, some authors published studies demonstrating that networks can exhibit or not the ESP depending on the driving input and scaling coefficients. As a reminder, both results were reported in the RC community: reservoirs scaled with a spectral radius ρ below unity showing oscillating behavior (Yildiz et al., 2012) and (input driven) networks exceeding the spectral radius $\rho \geq 1$ while still possessing the ESP (Verstraeten and Schrauwen, 2009; Caluwaerts et al., 2013).

Investigation of the network dynamics links also to the question at which regime a network achieves best results. Studies using the LSM reported good performance for networks at the transition between a stable and unstable regime (Legenstein and Maass, 2007), while for ESN it was shown that optimality is achieved in a stable network state but close to the stability border (yielding eventually the ESP) (Jaeger, 2002). For both networks, the detection of the “edge of stability” (EOS) is a vital research field. It is, therefore, a standard procedure to search for an optimal value of the spectral radius ρ of the reservoir matrix and to possibly identify the EOS.

A method of investigating the dynamics in a reservoir substituting to tune the spectral radius directly was introduced by Verstraeten and Schrauwen (2009). For ESNs with *tanh* activation, they established the Local Lyapunov Exponent (LLE) from the corresponding Jacobian matrix \mathbf{J} quantifying the reservoir dynamics. The advantage gained is that the temporal profile of activations traced in the Jacobian is flexible, and the LLE was shown to be even a better predictor when the network reaches the stability transition border. In general, the Jacobian \mathbf{J} is a matrix of partial derivatives and helps identify solutions for a set of nonlinear equations by using linearization around a specific value \mathbf{a} for a local linear approximation, e.g. for $f : \mathbb{R}^m \rightarrow \mathbb{R}$:

$$\nabla_{\mathbf{a}} f = \left(\frac{\partial f}{\partial x_1}(\mathbf{a}), \frac{\partial f}{\partial x_2}(\mathbf{a}), \dots, \frac{\partial f}{\partial x_m}(\mathbf{a}) \right) \quad (6.13)$$

For $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ the notation generalizes to:

$$\mathbf{J}_a \mathbf{f} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_m} \end{pmatrix}$$

which gives the Jacobian matrix evaluated at \mathbf{a} . Due to the simple derivative, for a *tanh*-activated reservoir with states $\mathbf{x} = [x_1, x_2, \dots, x_n]$, $x_i = \{1 \dots n\}$ the Jacobian \mathbf{J} is given by Verstraeten et al. (2007):

$$\mathbf{J}(x(t)) = \begin{pmatrix} 1 - (x_1(t))^2 & 0 & \cdots & 0 \\ 0 & 1 - (x_2(t))^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 - (x_n(t))^2 \end{pmatrix} \times W_{res}$$

where $x_i(t)$ is the i -th activated neuron at time³ t . As shorthand notation:

$$\mathbf{J}(x(t)) = \text{diag}[1 - x_1^2(t), 1 - x_2^2(t), \dots, 1 - x_n^2(t)] W_{res} \quad (6.14)$$

As the Jacobian mirrors the temporal evolution of the reservoir, the Lyapunov exponent approximation $\hat{\lambda}$ can be computed as Verstraeten et al. (2007):

$$\hat{\lambda} = \log\left(\prod_{t=1}^T (r_k)^{1/T}\right) \quad (6.15)$$

where T is the length of the trajectory and r_k denotes the k -th eigenvalue.

In an extended work, Verstraeten and Schrauwen (2009) introduced the minimal singular value⁴ μ of the Jacobian \mathbf{J} of reservoir activations as another reliable descriptor for reservoir dynamics. The computations provide a way to track and determine the dynamics of an input-driven reservoir.

The link to RP and RQA is obvious: the introduced measures *DIV* for divergence and L_{max} which connects to a similar understanding of stability, as e.g. stable systems have long diagonals in their RP phase portrait, and instabilities can be detected by fluctuations in the RP and discontinuous diagonal lines. RQA measures allow expressing different system states quantitatively for comparison under different conditions.

In the context of ESN, it is an important research branch to unveil the characteristics of the global parameters⁵ and the causes on the stability when driving the

³as a reminder, the discretized model is used and thus time refers to an increment of 1 in step size

⁴A measure in control theory.

⁵see chapter 4

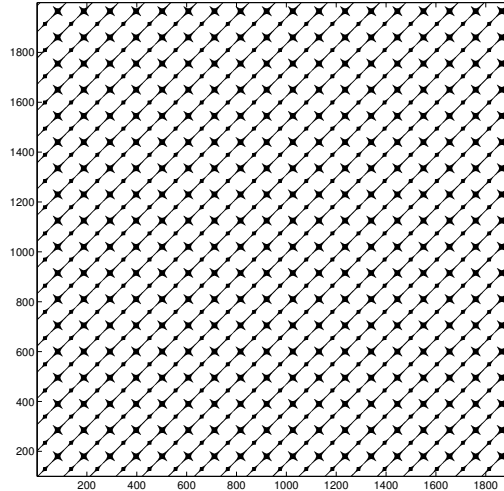


Figure 6.14: Validation of the alignment of reservoir states as shown in Bianchi et al. (2016a) for the input $\sin(\phi k)$, $\phi = 3/50$ and $k = 1 \dots 5000$.

network in different modes or parameter configuration. Recent work presented in Bianchi et al. (2016a) used recurrence quantification analysis (RQA) to determine dynamical properties from the reservoir layer in a standard ESN. To highlight the nature of input signals triggering the activation states, the authors chose the periodic sinusoidal signal and the Mackey-Glass time-series ($\tau = 17$, MG-17). They showed that reservoir activation patterns align with the input when the spectral radius $\rho < 1$. In contrast, the RPs revealed an irregular phase portrait when setting $\rho > 1$. We verified their results to assure comparability with our approach, and so we repeated the experiment for the sinusoidal (Figure 6.14) and, to stay consistent in the thesis, the Lorenz time-series (Figure 6.20).

For a visualization when we scale the reservoir matrix W_{res} above unity, we let an ESN run on the *circle* gesture but with $\rho = 1.5$. From the RP depicted in Figure 6.18, it can be shown that despite the LOI the patterns are spurious and contain only small diagonal lines. The recurrence rate obtained was $RR = 0.0145$. The visualization of the reservoir shows an irregular pattern in contrast to e.g. Figure 6.16.

In the thesis, we were also interested in identifying the computational boundaries and underlying dynamics for the gestures. We are interested in the question how the recurrence behavior can best be traced and whether from this it is possible to obtain a similar quantification as was done using e.g. the (L)LE. Interestingly, using the RP and RQA approach provides tools, where the neural activations in the reservoir form a vector in phase at every time instant $t, 1 \leq t \leq T$, T is the length of the time-series. The RQA computations apply as were introduced in the

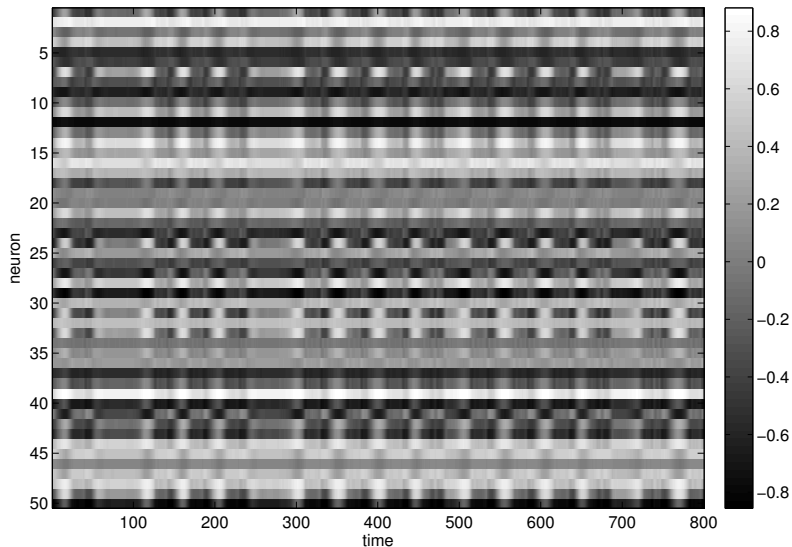


Figure 6.15: Left: Activations of 50 reservoir neurons processing the *turn* gesture performed several times (i.e. the discretized time, respectively frame-wise processing). Between gesture pauses, i.e. end of a gesture and start of a new one, the activations remain steady, depicted by the column-like structure between time intervals (the RP would exhibit laminar states).

section above.

In the previous section, we showed several plots on the reservoir RPs from stable to unstable (chaotic) regime. For quantification, we can make use of the measure introduced and choose *point left* and *circle* which displayed different characteristics in the RPs. It is obvious that different line distributions impact the RQA measures.

For the experiment, we performed a 20-step ahead prediction with $r_N = 50$, and split the data into 900/900/100 for training, test, and number of dropout samples, respectively. W_{in} and W_{res} were randomly initialized following a uniform distribution $W_* \sim U \in [-0.5; 0.5]$. Training was performed with regularization⁶ with $\phi = 1e - 8$. We varied the spectral radius ρ within the ranges $[0.8 - 1.5]$ with stepsize 0.1 except for the critical border where we also put $\rho = 0.99$. We ran 10 ESN instances and averaged the results. We focused especially on the measure RR , DET , and L_{max} because we are interested in the ESN stability. The value for DIV is reciprocal to L_{max} and can thus be easily calculated from.

As expected, values below unity showed high values for all three measures. For $\rho > 1$ the difference between 1.0 and 1.1 is smooth, but drops between 1.1 and 1.2. A significant change in the values can be reported for $\rho \geq 1.2$. Figure 6.21 supports this for the *point left* gesture and L_{max} .

⁶cf. formular in section 4.1

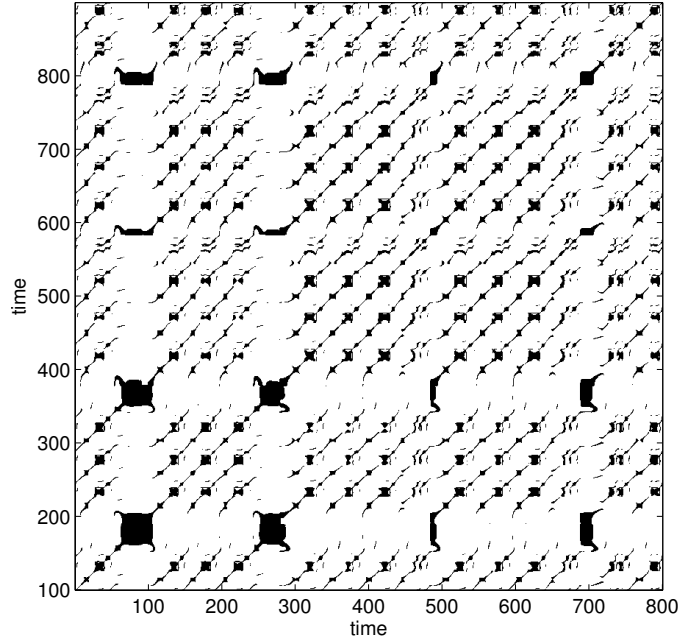


Figure 6.16: Corresponding recurrence plot of reservoir activations. The laminar states occur for low activity in the reservoir. The diagonal lines display the periodic characteristics of the gesture.

Inspecting the individual trial values reveals a high discrepancy for all three RQA measure. In the case of the *point left* gesture, the RR was minimum $min = 0.00085$ and maximum $max = 0.18727$. The DET values showed an even more drastic picture ranging from 0 to 0.97064. As DET is coupled with L_{max} these fluctuations apply as well and is also 0 in the worst case.

When computing the RQA measures it becomes evidence that values for a certain parameter configuration hardly differ, while for configurations at the presumable EOS the RQA measures substantially change (see also Figures 6.21 6.22). Therefore, we propose a criterion to automatically detect these changes. While Bianchi et al. (2016a) defined a criterion based on the variances of measures introducing an additional threshold, we employ a simple differencing scheme.

First, the RQA measures are averaged over the trials:

$$RQA_{total}^c = \sum_{i=1}^n RQA_i^c \quad (6.16)$$

is the total amount of an RQA measure for n trials (for the experiments presented here $n = 10$) and for a certain parameter configuration c , where the average is simply:

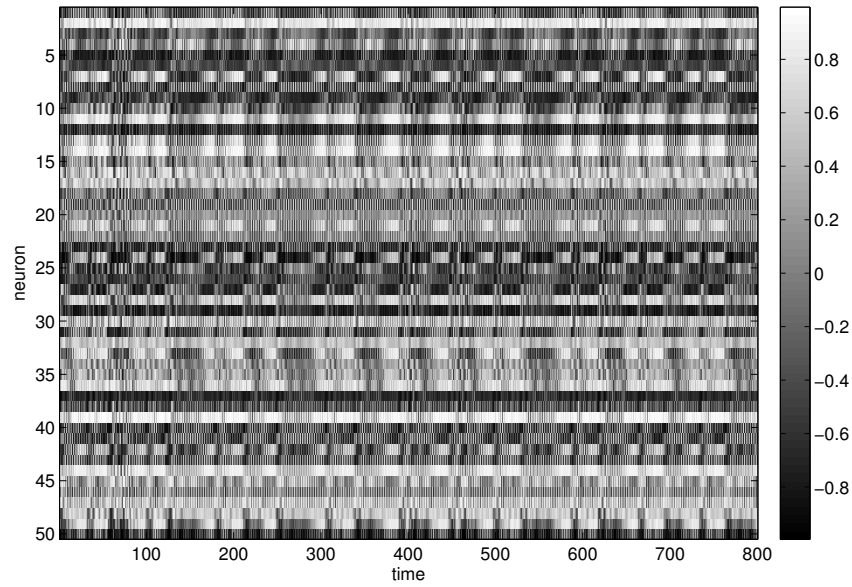


Figure 6.17: Reservoir activations when $\rho = 1.5$. The neurons display abrupt changes in their activities.

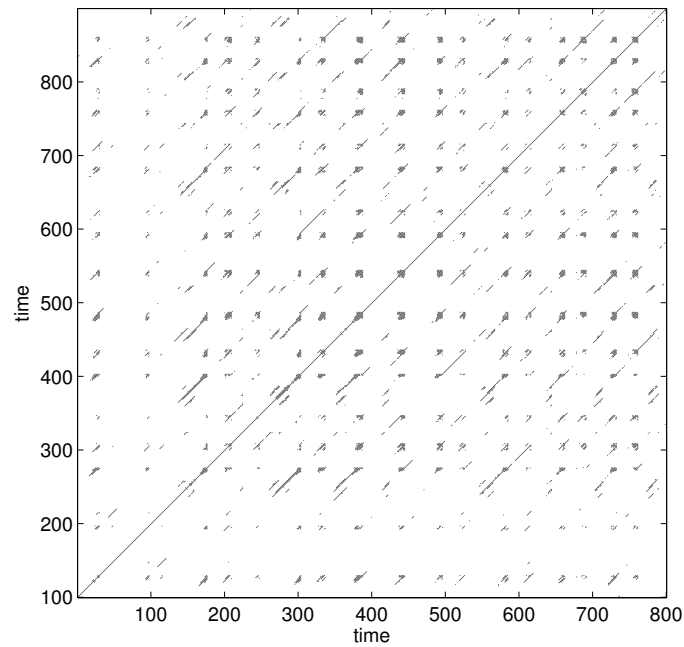


Figure 6.18: The corresponding recurrence plot of reservoir activations when $\rho = 1.5$.

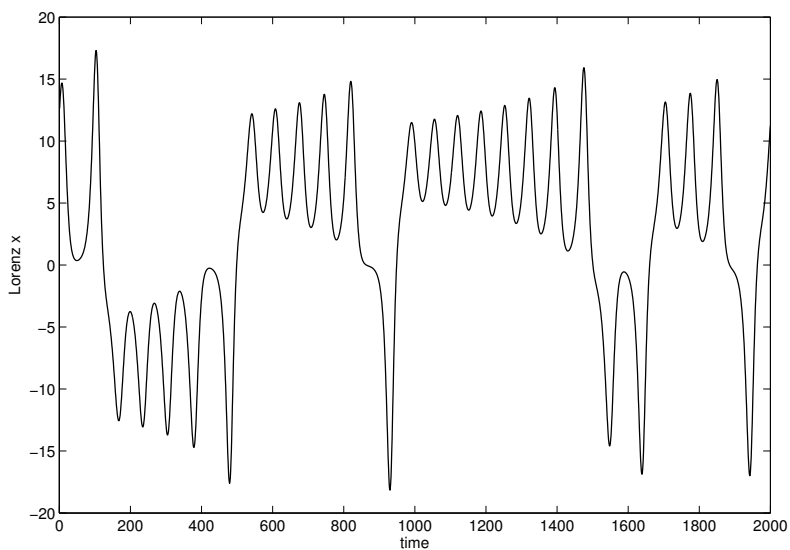


Figure 6.19: Lorenz system evolution of the x-component.

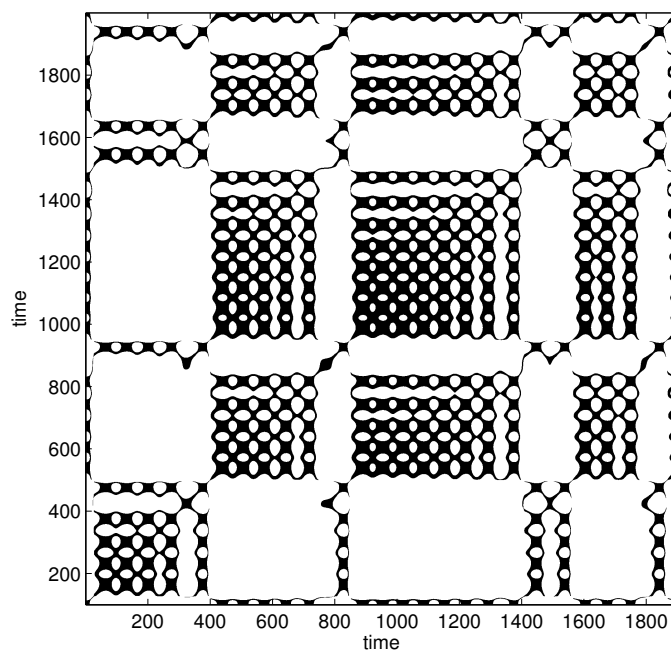


Figure 6.20: RP of the corresponding reservoir activations for the Lorenz system, which equals the unembedded phase portrait for the x-component.

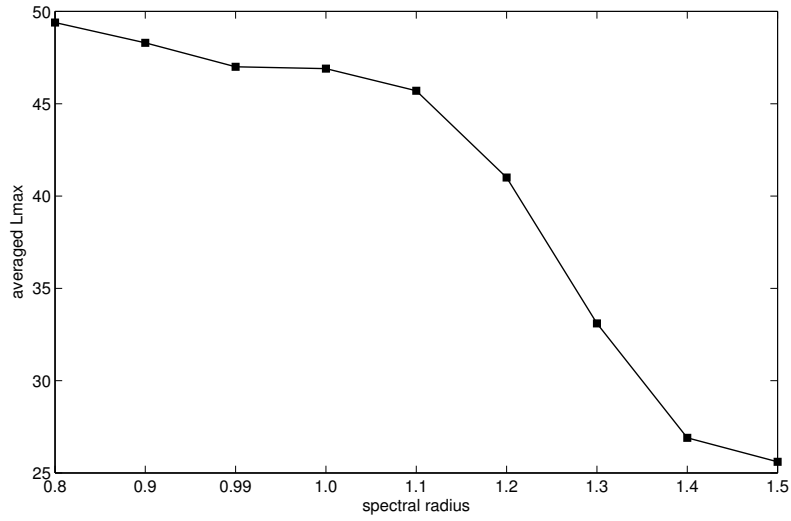


Figure 6.21: L_{max} values over the spectral radius ρ averaged over 10 trials. The curve shows a small decrease for values up to $\rho = 1.0$, but drops significantly for $\rho = 1.2$, indicating that the reservoir enters the chaotic regime.

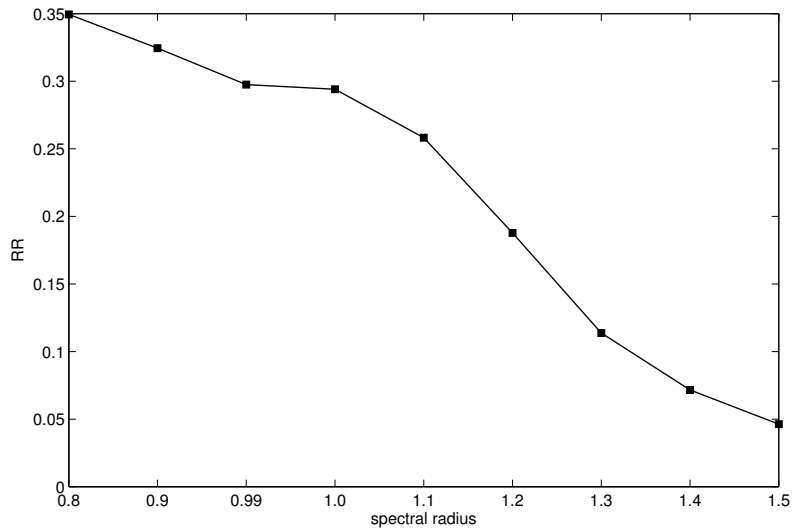


Figure 6.22: Recurrence rate RR for different spectral radii ρ averaged over 10 trials. The curve shows a similar trend as for L_{max} with only small variances up to $\rho = 1.1$, followed by a significant decrease.

$$RQA_{avg}^c = RQA_{total}^c/n \quad (6.17)$$

We then iteratively compute the differences of RQA_{avg}^c measured over all considered parameter ranges c_j and perform the *max* operation:

$$\Delta_{RQA}^{c_j} = RQA_{avg}^{c_j} - RQA_{avg}^{c_{j+1}} \quad (6.18)$$

$$\Delta_{RQA}^{c_{all}} = \max_j \{ \Delta_{RQA}^{c_j} \} \quad (6.19)$$

where c_{all} includes all parameter values considered, here the 9 values of ρ used for the experiments. The input to equation 6.19 is a vector the size of c_{all} (i.e. 9) and thus from the index of the computed value the parameter value c can be extracted. As an example, we take all averaged values for $L_{max} \{1.1, 1.3, 0.1, 1.2, 4.7, 7.9, 6.2, 1.3\}$. $\Delta_{RQA}^{c_{all}} = 7.9$ which is the 6-th position and thus $c = \rho = 1.2$. The result confirms the trend visible in Figure 6.21.

6.3 Chapter Summary

We introduced the methodology of recurrence plots (RP) and recurrence analysis (RQA) as a valuable visualization and quantification tool to identify data characteristics serving as input to an ESN and the specific reservoir dynamics under different settings. We showed that the reservoir in a stable condition aligns with the input, while this property is lost when the reservoirs got unstable, supporting recent finding from Bianchi et al. (2016b). Introducing the RQA measures allowed assessment of values characterizing both the underlying time-series and the reservoir dynamics. Further, we defined an effective criterion complementing the metrics for the temporal evolution introduced.

We want to emphasize that the introduced methods are not restricted to gestures, but can be applied the same way for any activity pattern or for the analysis of any other time-series with repetitive characteristics, e.g speech frequencies (actually, first applications of RP used geophysical, climate or medical data). Thus, the work presented here can be expanded to the analysis of reservoir dynamics and the influence of parameter settings for different tasks, which may stimulate further debate and investigations on network settings opening a more global view on RC and its applicability.

Chapter 7

Reservoir Initialization and Pruning

In this chapter, we investigate the structural properties of the reservoir. One critique about the applicability of ESN is that their stochastic initialization yields networks with varying performance. It reduces to a “chance” to get good networks which satisfiable solve a task, which makes the ESN a rather generic network. As a consequence, ESN application for a specific task leads to exhaustive parameter tuning which is computationally demanding and hinders insights into the actual working principles behind. In Sussillo and Abbott (2009) this procedure was highlighted as yielding “another unintelligible network”. The usual scheme is to perform a grid search on the parameter space, but Bergstra and Bengio (2012) showed that random search yields similar or even better results being computationally more efficient at the same time.

We show that for the prediction of gesture trajectories even small changes in the initialization procedure decreases this variability, supporting findings from other studies. As the experiments still consider the norm of the reservoir and thus are rather static, we propose a novel algorithm based on pruning the reservoir. This approach is motivated by two facts: first, by pruning the reservoir we resign to set the sparsity beforehand. Instead, a specific topology emerges from the input driving the reservoir, preferably with a high sparsity. Gallicchio and Micheli (2011) sd evidence that it is not essential for the performance but plays a role for time-critical applications. However, the question whether or not a certain topology or reservoir setting is beneficial for a set of tasks is not fully satisfiable answered yet. Approaches with a specific topology did not show any superior performance over completely random networks, however, Lukoševičius and Jaeger (2009) stated that “This, however, does not serve as a proof that similar approaches are futile”.

Second, the pruning procedure has a significant impact on the algebraic properties of the reservoir matrix. The scaling procedure is still a matter of debate and in connection with the results from chapter 6, we concluded that it might be beneficial to establish a mechanism which tunes this according to the presented input.

For our implementations, we used the ESN toolbox as a starting point Jaeger (2002).

7.1 Getting Good Reservoirs is a Chance

The reasons to have a closer look on the reservoir topology and possible optimizations are manifold. We introduced the recurrence quantification analysis and how the data and accordingly the reservoir structure can be qualitatively and quantitatively described by means of recurrence plots and derived measurements from the line distributions. The insights gained by the procedures help the network design and can be applied to any other data, which makes this approach less restricted to a particular dataset.

In general, the reservoir is initialized with weights drawn from a probability distribution (see section 4.1.2), which stay fixed. The connectivity κ , i.e. the synapses between the reservoir neurons, is mainly sparse and may follow a certain topology. Experiments are then performed over a predefined number of trials giving an average for any evaluation measure (e.g. number of misclassifications, MSE). To minimize an error or to maximize the recognition needs thorough tuning of the system parameters. Using search algorithms raise the following questions: Why a certain parameter configuration emerged and is the result comprehensible? Usually, authors report the values without elaborating further on that. This leads to the next question: How sensitive is the network to parameter changes? This further connects to point that authors sometimes lack to explain the parameter interplay, which decreases the applicability of a network. In the previous chapter, we approached the topic about the spectral radius and the input scaling. We will now access the reservoir properties: we first contrast a random initialization scheme with an orthogonal one and highlight the differences in their performance and their performance variability.

7.1.1 Weight Matrix Initialization

We described approaches which aim at optimizing a particular network structure. One of the reasons is that neural information processing operates at the EOS or slightly below it (Levina et al., 2007; Priesemann et al., 2014). Furthermore, neural connectivity is shaped by plasticity mechanisms as described in chapter 3 and chapter 4. Also, initialization of a specific reservoir structure, as was for instance presented in Maass et al. (2002) resembling the microcolumns in the visual cortex, is a vital research field. All approaches deviate from the randomness in the reservoir, which however was shown to achieve good performance. The aim, therefore, for ESNs is to find “good” reservoirs.

The aim in applications with reservoirs is to decrease the variance of experimental outcomes inherent in the random initialization process even when all parameters stay fixed. In the following, we compare the standard procedure with a small alteration in the initialization, that is setting up an orthogonal matrix serving as the reservoir. We observed a decrease in the variability of performance for a prediction task, but not necessarily also a better performance. This supports findings from other studies comparing different reservoir structures.

In the following, we present some experiments highlighting different weight initialization strategies and their influence on performance. We conducted some initial experiments with a rather simple ESN structure to investigate the performance variations given a prediction task for gestures. We employed a standard ESN and initialized the reservoirs both as a random matrix and as an orthogonal matrix, i.e. a matrix M is orthogonal if $MM^T = \mathbf{I}$, i.e. vectors v_i in a matrix M have unit length ($v_i v_i = 1$) and are pairwise orthogonal as the dot product of two column vectors $v_i v_j = 0, i \neq j$. These two criteria can be used in general to check for orthogonal matrices.

The characteristic of the eigenvalues is an important measure to determine the reservoir matrix algebraic properties, and thus its stability (as outlined in chapter 6). The distribution of the eigenvalues for a random and an orthogonal network on the unit disk is depicted in Figure 7.1. The randomly initialized matrix exhibits diverse magnitudes of eigenvalues, which was generally shown by Girkos law to increase when $n \rightarrow \infty$ for a $n \times n$ random matrix. In contrast, eigenvalues of an orthogonal matrix have uniform magnitudes. This has direct implication also on the short term memory capability of a network as shown in Boedecker et al. (2009).

We performed a 1-step ahead prediction and evaluated the performance using the MSE on 30 trials for both random and orthogonal reservoirs.

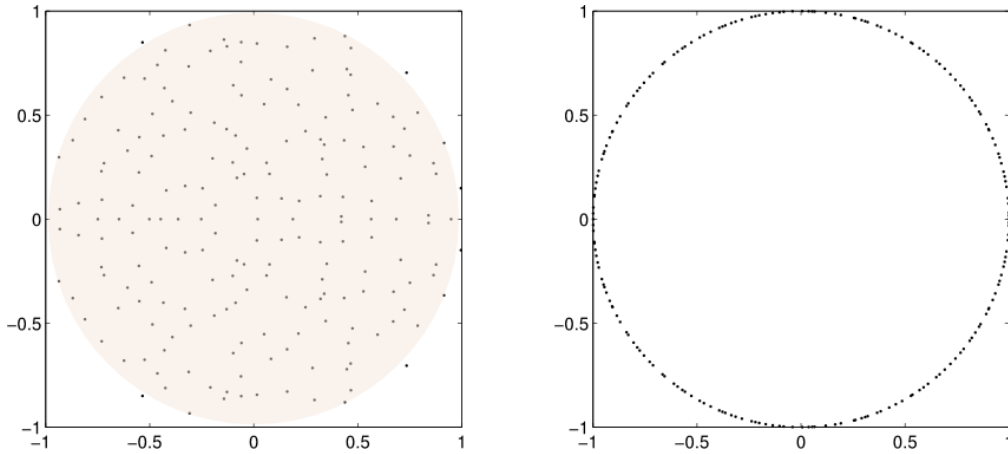


Figure 7.1: Left: Random matrix M generated with $n=200$ and scaled $M/\sqrt{(n)}$. The eigenvalues of M scatter around the unit circle with different magnitudes. Right: Eigenvalue distribution shows equal magnitudes for an orthogonal matrix, i.e. $MM^T = \mathbf{I}$.

For the *turn* gesture, we observe that the MSE found for the random initialization strategy is lower, but the boxplot reveals (in line with results from chapter 5) high variations within results. We observe that in this case also the performance of some ESN instances is worse compared to the orthogonal initialization. The latter strategy yields on average a higher MSE but also less variance over the trials.

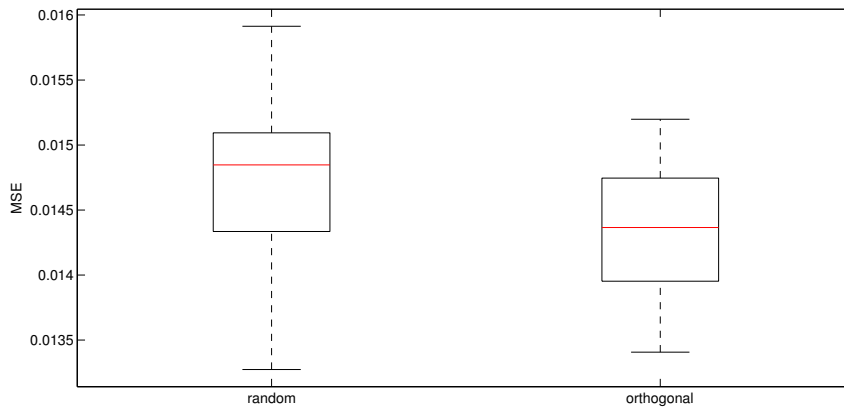


Figure 7.2: Result of 30 trials using a 1-step ahead prediction on a *turn* gesture using random initialization of the reservoir (left) and orthogonal matrix (right).

The prediction of the *point left* gesture shows a clearer picture, where for the orthogonal matrices the minimum and the median value is less. Again, the realization of 30 trials yields only little variation in the MSE results compared to the random initialization.

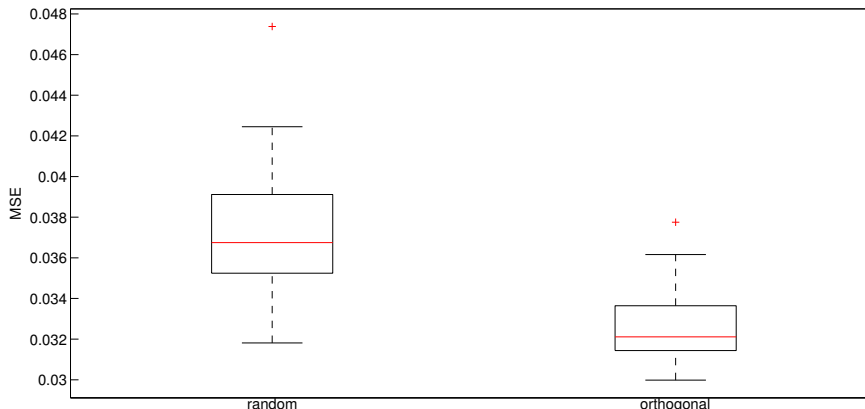


Figure 7.3: Result of 30 trials using a 1-step ahead prediction on a *point left* gesture using random initialization of the reservoir (left) and orthogonal matrix (right). The + display outliers.

Although the network initialization is still randomly driven, we showed that even a little modification on the reservoir initialization decreased experimental variances over the trials and thus performance is more robust regarding the error evaluation; casually speaking, we can be more confident about the actual performance when results vary less. To find further evidence, we inspected the first 10 results from the trials, as 10 is an often chosen trial number in the ESN literature. The two subsets revealed a better performance for the orthogonal reservoir in terms of the least MSE value (random: 0.0327, orthogonal: 0.0304). In addition, the worst performance with an MSE value of 0.474 found over all 30 trials was already present in the subset of 10 first trials, while the minimum (the best performance network) was not. Increasing the number of trials thus might be beneficial to gain proper ESNs, while for the orthogonal network this would lead to overhead. Although we set up only a small experiment, we believe that due to the inherently different structure of the reservoir matrices the findings in a greater setting would apply as well.

The results imply that orthogonal reservoirs yield some kind of determination of error behavior over trials, while the error diffusion for random reservoirs shows the dependence of a reasonably good performance reservoir on the number of trials and thus chance. In addition, the small error ranges resulting from experiments with orthogonal matrix reduces the necessity of guessing also the number of trials. The randomness in ESN requires quite some expertise in the application field of ESN and thus more principled ways of experimental settings are desirable. Finally, our results complement findings from other studies on reservoir weight matrix modifications (cf. chapter 4).

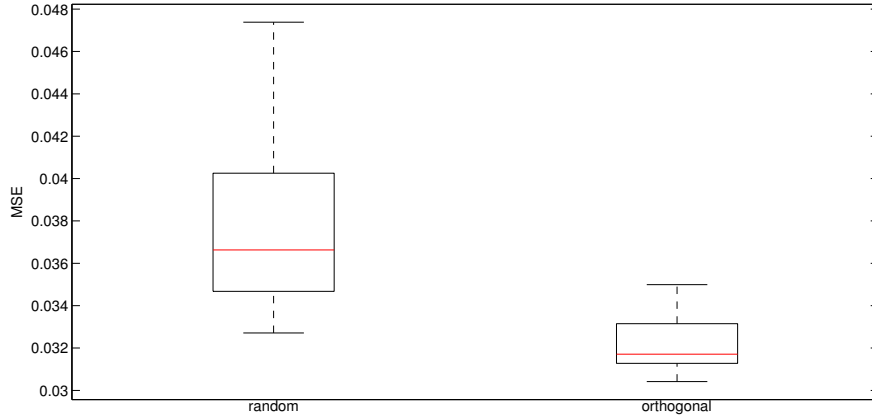


Figure 7.4: Display of the results shown in Figure 7.3 from a subset of the trials (1-10) using random initialization of the reservoir (left) and orthogonal matrix (right). The orthogonal initialization has the better performance and the graph displays a similar error distribution. In contrast, the random reservoir has a great maximum value, which becomes an outlier when averaged over more trials (cf. Figure 7.3).

7.1.2 Sensitivity of Reservoir Connections

Reservoir Computing is usually emphasized to be a neurologically plausible model for information processing in the cortex. Besides the question whether it suffices to use a random network or not, cortical processing is prone to synaptic- or neuron loss. In the healthy brain, this causes no severe failure to accomplish a task, while we will show the crucial impact on neuron loss on the computational performance of the reservoir. That has special implications for using ESNs as an autonomous pattern generator, especially when used in a “chaotic” network precondition (Vincent-Lamarre et al., 2016), where connectivity disruptions yield different phase space trajectories and can thus not be used as a stable attractor representing, for instance, a particular movement. This has direct implications of using an ESN as e.g. a central pattern generator (CPG).

While the dynamical system view on the networks involved are quite well underpinned, the question how robust these networks are against changes in their fixed topology is less investigated. However, reasoning about the neurobiological inspiration of RC models contains also principles like neural rewiring or plasticity (e.g. Hebbian learning, intrinsic plasticity). These aspects form the basis of robust neuronal information processes and development of functional cortical microcircuits. Losing synapse connectivity or even neurons do not affect brain computations. Translated to the ESN architecture, a lost connection is the deletion of a node or set of nodes in the reservoir matrix W_{res} . Very recent work in the

line of this research (Vincent-Lamarre et al., 2015, 2016) highlighted this aspect for the autonomous neural architectures introduced by Sussillo and Abbott (2009) and Laje and Buonomano (2013) work on the timing networks with spontaneous fluctuations. They showed the huge impact of a synapse or neuron removal on trained networks (e.g. using FORCE algorithm, see chapter 4) resulting in significantly error-prone networks in test scenarios. This has direct negative impact on the learned attractors, i.e. trajectories cannot be longer correctly generated.

In the case of prediction or supervised classification, the perspective on neural loss is different. As outlined in chapter 4 numerous variants of reservoir optimization were introduced to account for the variability due to random initializations, and to benefit from neurobiological principles yielding robust networks and algorithms which facilitate understanding the working basis of ESNs.

For the experiment, we used a 1-step ahead prediction on both the Mackey-Glass benchmark ($\tau = 17$, MG-17) and for the gestures with different levels of connectivity. As we were particularly interested in the impact of neuron or synapse loss, we set $\rho = 1$. The train- and test lengths were equal and set to 900 timesteps, where for the training phase the first 100 steps were discarded.

The MG-17 data when trained with a fully connected reservoir shows increase in the MSE for the 1-step ahead prediction when the used architecture suffered from a synapse removal. The table 7.1 depicts the result of the 1-step ahead prediction task on the ESN configuration described above. The data was selected on the basis of their complexity, i.e. *stop* reveals the least motion, while the Mackey-Glass (MG-17) task is the most complex one. Interestingly, the great difference between the gestures and MG-17 is that the latter performs best for a fully connected network, but when deleting one node the performance reduces drastically and is even worse in the case a complete connection row is deleted from the network. In contrast, the performances for the gestures are worse for the full network and show increase in the error for the node deletion, with the difference that the network seem to even out the absence of connections. As a consequence, the drop in performance is less influential. We explain the performance difference with the fact, that the gestures show less complexity (especially when interpolated and thus smoothed) while the chaotic MG-17 time-series is a trajectory with abrupt changes. Disruptions of certain parts of the network are thus crucial as they may have captured an important signal part needed for correct prediction. The same applies to the signal generation, when considering an ESN as an autonomous pattern generator, as was pointed out by Vincent-Lamarre et al. (2015). The rather simple experiment is in line with recent work on the validity of RC resembling neural processing

| | fully connected | loss of a synapse | loss of a neuron |
|--------|-----------------|-------------------|------------------|
| stop | 0.017637 | 0.022742 | 1.0723 |
| pointR | 0.035826 | 0.057853 | 0.30626 |
| turn | 0.018839 | 0.025622 | 1.5651 |
| MG17 | 4.0254e-08 | 0.61222 | 14.4069 |

Table 7.1: MSE results from synapse or neuron removal

mechanisms (Vincent-Lamarre et al., 2016) and encourages to investigate further the effect of connectivity for classification. Although the error differences may not be crucial here, we think that when scaling up learning for a broader range of gestures and their representations, the error may add up fast and lead to bad performance for generalization. In addition, data from a continuous video stream transfer specific noise to the input. The trained architecture, however, should be able to cope with this data.

7.2 Pruning Procedures in Reservoirs

In chapter 4, we introduced the significant network parameters along with the general guideline setting up an ESN. In the literature, it is often stated that sparsity has a minor role in ESN settings and that a certain topology has no significant improvement over completely random networks. However, the converse argument does not imply to neglect investigations into the reservoir organization. Especially for the task of gesture recognition, it is interesting to reveal the emerging features of an “optimal” reservoir.

The common practice in the reservoir design is to have a) a large number of reservoir neurons for capturing intrinsic signal properties and to provide a high-dimensional feature space to allow application of simple, linear models b) a sparse connectivity between reservoir neurons and c) a random initialization following a certain probability distribution for all considered network matrices. This procedure comes along with some drawbacks: a big reservoir may contain redundancy, thus it is advised to use a regularization like ridge regression to avoid overfitting. This introduces an additional parameter in the subsequent optimization procedure. For grid search, every new parameter adds exponentially to the search routine. The underlying properties of the reservoir or a network topology, however, remain unexplained (although the intuition for sparsely connected random networks is that they exhibit small-world properties, which we refer to in the sections below). Re-

dundancy in the reservoir was also revealed for symbolic input using PCA, which was explained by the suffix representation in the reservoir state space (Gallicchio and Micheli, 2011). This means that symbolic streams with similar suffixes are closely mapped in the corresponding state space, exploiting also the clustering behavior of RNNs with small weights prior to training (Tino et al., 2004). This led to the notation of the *Markovian architectural bias* in state space models.

Another misconception is the sparsity factor in ESN providing possible sub-reservoirs or small-world structure which sufficiently represent the various signal parts. Studies revealing that reservoirs with dense connectivity perform equally good however contradict this assumption, as we outlined in chapter 4. Therefore, other properties need to explain the reservoirs “richness”.

Finally, the random reservoir initialization process yields no superior performance over deterministic approaches (Rodan and Tino, 2011) but impede experimental repeatability and reproduction of results for validation and comparison with other architectures diverging from the randomness in the ESN.

Dutoit et al. (2008) showed that pruning connections in W_{out} confirm regularization effects due to sparsification of the output layer. The criterion to prune neurons was based on the validation error after training and when using the network in free-run mode (i.e. includes the feedback matrix W_{back} and teacher-forcing). The procedure followed a less principled way by assuming equal probability of pruning, however as remarked by the author’s other heuristics can be used to refine the strategy. An example is directly provided by the authors themselves, showing the effect of neuron pruning based on Fishers Discriminant. A downside of this study was the lack of investigations on the reservoir dynamics. Scardapane et al. (2014) introduced a correlation-based measure to determine the pruning of synapse connections in an online fashion, and further expanded their work to neuron pruning. Their studies showed that although the pruning did not necessarily lead to an improved performance, the applied criterion supported an automatic tuning of the reservoir itself to an optimal sparsity and reservoir size. Pruning the reservoir was also studied in Butcher et al. (2010). Also, hardware implementations based on RC principles has gained more attention in the last years, where an optimal reservoir with a reduced set of connections is beneficial.

In the previous section, we demonstrated how an orthogonal network may be favorable for a prediction task. However, a fixed spectral radius may limit the flexibility necessary for picking the information when a network is fed with sequential variations.

Instead of searching for an optimal reservoir size and sparsity in a reservoir

used for gesture recognition, we introduce an alternative scheme keeping into consideration the input-driven behavior of the reservoir (as e.g. presented in chapter 6). While parameter search algorithms test for permutations and can thus be introducing exponential search for every parameter included, we first naively set up a reservoir and optimize the reservoir directly by pruning. The hypothesis is to obtain an optimal reservoir size and reservoir sparsity κ while keeping the good performance for the given task.

As reducing the reservoir matrix element or neuron-wise has an impact on the spectral radius ρ , we first investigate its changes by random pruning. It causes a consequent shrinkage of the parameter ρ and is 0 in the limit if no intermediate rescaling procedure is applied. Reservoir adaptation based on plasticity mechanism, as e.g. implemented with Hebbian learning, yields a similar effect, reducing the norm of W_{res} (Siri et al., 2008) and hence its spectral radius. Note, that the element-wise pruning of the reservoir yields a synaptic disruption, while a complete neuron loss can be realized by setting a row of W_{res} to 0.

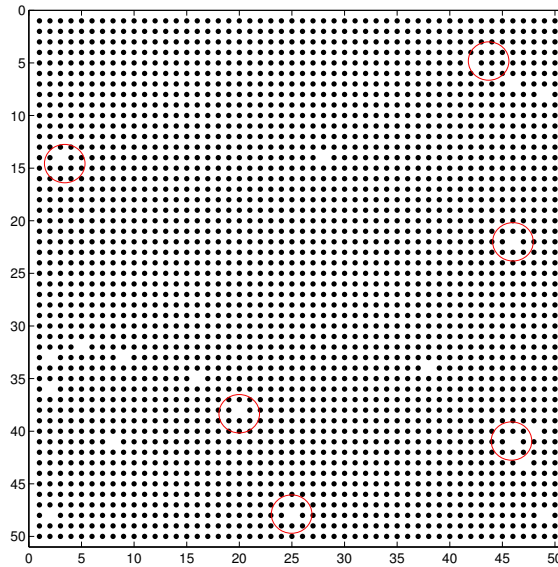


Figure 7.5: Random pruning of reservoir connections of a network with $r_N = 50$. The figure shows the matrix W_{res} after 20 iterations. Red circles depict examples of synapse loss.

We want to show the influence of pruning towards the spectral radius ρ and that this procedure can also cause an increase in ρ , possibly leading to instabilities in the network. We created 50 reservoir matrices initialized as $W_{res} \sim G$ (Gaussian) and scaled them to get a spectral radius $\rho = 1$. We iteratively pick a random element from the reservoir and set it to 0. The matrix gets sparse as depicted in Figure 7.5, resulting in a decrement of the spectral radius and convergence to 0. Interestingly,

this behavior is not monotonic but contains fluctuations where values of ρ can also increase again, supporting findings from Scardapane et al. (2014). Figure 7.6 shows the results from the 50 reservoirs, where the red curve denotes the average spectral radius. The question remains, whether pruning enables an automatic tuning of the spectral radius given the task-specific input or imposes a constraint to use pruning due to possible instabilities.

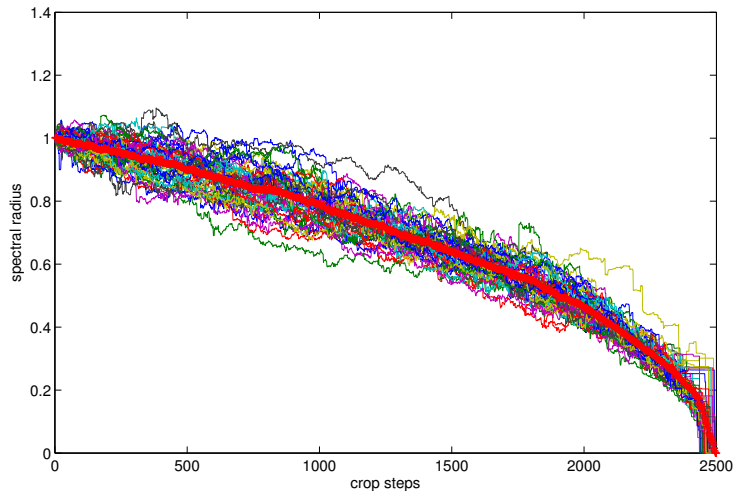


Figure 7.6: Track of the spectral radius when iteratively cropping elements from the reservoir of size $r_N = 50$. The red curve shows the average decay from 50 reservoirs.

7.2.1 Experimental Section

We set up experiments starting with a dense reservoir connectivity. Instead of setting the reservoir size and the sparsity factor beforehand, we argue that pruning the reservoir connections will yield an optimal reservoir. This assumption is based on the observations that big reservoir are redundant and needs additional regularization. The redundancy can be avoided by orthogonalization (hence decorrelation), which we demonstrated decreases experimental variability. The difference to other studies is that we use a classification task.

Reservoir Variance for Gesture Prototypes

The motivation behind the pruning process of the reservoir is that large reservoirs produce redundancy and tend to overfit the data. A way to investigate redundancy, respectively, correlations between variables is to employ a principal component analysis (PCA). In brief, a PCA decorrelates an input by transforming the original data space into a new space spanned by so-called eigenvectors e_i with

its corresponding eigenvalues λ_i . The eigenvalues are derived from the data covariance matrix, which captures the data distribution. The largest eigenvalue with the associated eigenvector points into the direction of the greatest variance. This eigenvector is referred to as the first principal component (PC). The second PC is an orthogonal projection of the first PC, showing the direction of the second-largest variance. The construction of the PCA space repeats until a predefined threshold is reached, say 90% of the data variance. Considering all eigenvalues and their corresponding eigenvectors is thus just a transformation of the original data space.

To get an insight into the reservoir activations triggered by the different gesture sequences, we selected five sequences from the *5DG* dataset, each representing a gesture class (prototypes).

Table 7.2: Average sequence lengths

| | [mean] | median |
|-------------|--------|--------|
| circle | 74 | 68 |
| point left | 84 | 84 |
| point right | 128 | 134 |
| stop | 120 | 124 |
| turn | 190 | 183 |

The sequence choice is based on the simple heuristic of sorting the sequence lengths in an ascending order and taking the sequence indexed by the median. The order is arbitrary since the median is taken from the cardinality $|c|$ of sorted sequences per gesture class c . This procedure allows direct access to the data. Taking the mean would shift the average sequence lengths towards outlier, i.e. too short or too long sequences (the brackets in 7.2 stands for the floor operation).

We collected the state activations from the sequences from a reservoir with $r_N = 100$ and dense connectivity (only the states are relevant here). We then performed a PCA on the state activations.

We report the amount of variance explained (equation 7.1) by the first four PCs ($e_i \lambda_i$, for $i = 1, 2, 3, 4$) in Table 7.3, which capture almost all variance from the reservoir. The values correspond to the computation:

$$\lambda_i / \sum_{j=1}^n \lambda_j \quad (7.1)$$

Table 7.3: Variance explained

| | circle | pointL | pointR | stop | turn |
|------|--------|--------|--------|-------|-------|
| 1.PC | 42.77 | 75.53 | 73.30 | 58.16 | 49.29 |
| 2.PC | 37.82 | 19.62 | 10.80 | 32.00 | 35.88 |
| 3.PC | 10.65 | 2.08 | 8.16 | 6.59 | 5.99 |
| 4.PC | 1.66 | 0.97 | 2.28 | 1.10 | 3.44 |

where λ_i are the first selected eigenvalues from the set of all eigenvalues λ_j , the total amount of variance. The result indicates that a smaller reservoir might be sufficient for the processing of the sequences. However, we only used the prototypical sequences and more investigations are needed for the complete sequence set. Specifically, we want to explore the processing capabilities in a single reservoir for the recognition of the six gesture classes (introduced in chapter 5) and the role on sparsity in terms of pruning reservoir connections.

Experiments on all Gestures Sequences

A key ingredient to the successful application of ESNs is to initialize the reservoir with a large number of neurons. However, big reservoirs tend to overfit the data which substantially decreases the performance gain. As a consequence, ESNs are trained with additional regularization, a common procedure in machine learning for those ill-posed problems. While this approach might be useful for performance comparisons on e.g. benchmark data, the ESN remains a rather generic network, neglecting specific properties emerging from the reservoir or the network learning. The network characteristics for a particular application task, however, would give helpful indications for proper configurations, similar to what we have demonstrated for the reservoir initialization in the previous section.

The PCA on the reservoir activations of prototypical gesture sequences hinted to reservoir redundancy. However, the interesting question is how a reservoir is able to discriminate the gesture different classes. We investigate the role of connectivity, respectively, sparsity in the reservoir and the influence of a pruning strategy on the performance and the resultant reservoir topology. In particular, we are interested in learning gesture sequences in a *one-shot* learning setting. This means that we only use the prototypical sequences introduced in the section below to train an Echo State Network.

The network is initialized with input and reservoir weights randomly drawn

from a uniform probability distribution $\sim \mathcal{U} \in [-0.5; 0.5]$. Instead of setting a specific sparsity factor, all neurons are densely connected. An adjacency matrix A_{ij} can be derived from the different iteration stages to keep track of the emergent reservoir structure. Clearly, $A_{ij} = 1 \forall i, j$ for a dense matrix. From either matrices W_{res} or A_{ij} the resultant sparsity can be computed to determine the ratio between start and end configurations. We run the reservoir with the prototypical sequences (see section below) and use the training set (Set 1: 157 sequences) and test set (Set 2: 79 sequences) introduced in chapter 5 both as test data. We do this to show later the differences in the evaluation for the random network in chapter 5, and the results we obtained from a very small dataset but with interesting characteristics emerging from the pruning strategy (Algorithm 1).

Algorithm 1 Synapse Sparsification

Input Data

Weight Matrix Initialization

$$W_{in}, W_{res} \sim \mathcal{P}$$

$$W_{back} = 0$$

Parameter Initialization

$$\kappa = 1, \rho_{desired}, r_N$$

Reservoir Rescaling

$$W_{res} \leftarrow \rho_{desired} \frac{W_{res}}{\rho(W_{res})}$$

Set adjacency matrix

Let the reservoir run with prototype sequences of c classes

Collect reservoir state activations X

Compute the pairwise correlation between X

Define a correlation threshold θ

if $\text{corr}(X) \in [-\theta; \theta]$ **then**

Get the corresponding row and column indices from the correlation matrix

Prune: set $W_{res} = 0$ accordingly

end if

Test the ESN with the pruned reservoir matrix

We fix the spectral radius $\rho = 0.9$ as is a common procedure in ESN application and chose an interval $[-0.1; 0.1]$ (θ in Algorithm 1). As described above, pruning the synapse connections in the reservoir results in a decrease of the reservoir matrix norm, however, we are interested whether the input driving the network will have an influence on this parameter. The results for a reservoir with $r_N = 50$ are reported in Table 7.4 for 10 trials. It is evident, that the spectral radius fluctuates

Table 7.4: Results from the pruning of a 50-neuron reservoir

| Pruned connection | Set 1 eval | Set 2 eval | spectral radius |
|-------------------|------------|------------|-----------------|
| 328 | 36 | 16 | 0.9267 |
| 342 | 37 | 18 | 0.8726 |
| 306 | 37 | 18 | 0.9603 |
| 334 | 38 | 17 | 0.8625 |
| 338 | 40 | 17 | 0.8867 |
| 326 | 33 | 18 | 0.9815 |
| 306 | 36 | 21 | 0.8874 |
| 290 | 31 | 25 | 0.9226 |
| 332 | 32 | 19 | 0.9684 |
| 348 | 34 | 19 | 0.9503 |

around the fixed value. The best performance was achieved for $\rho = 0.9815$, which is close to the critical theoretical border. Note, that the number of pruned synapses is rather low. This observation may be explained by the low number of reservoir neurons, where the connections might have a positive influence on the information processing of the gestures.

An interesting note is that increasing the reservoir size r_N does not lead to a performance improvement. We tried $r_N = 100$ und $r_N = 200$ and expected an increase. However, especially for a spectral radius below 0.9, the performance was worse. For the reservoir of size $r_N = 50$, increasing the threshold θ led to no improvement. Summarized, we suspect that the input-driven regime of the reservoir the recognition of different gestures is mainly driven by the spectral radius and low sparsity. Especially the tuning of the spectral radius towards unity is an interesting phenomenon emerging only from a simple pruning strategy. Further, the usage of larger reservoirs for a rather small feature set results in overfitting.

Finally, we want to demonstrate the evaluation results from Set 1 and Set 2 in terms of misclassification. Although the *stop* gesture performed worst in both sets and caused much of the performance errors reported in Table 7.4, more dynamic gestures were learned reasonably well (see Figures 7.7, 7.8, see Figures 7.9, 7.10 for the 6th trial). This is an interesting result in comparison with findings from chapter 5, as here only one gesture sequence per class served as input to the ESN.

Also, the rather simple representation seems to be sufficiently represented in the network of 50 neurons.

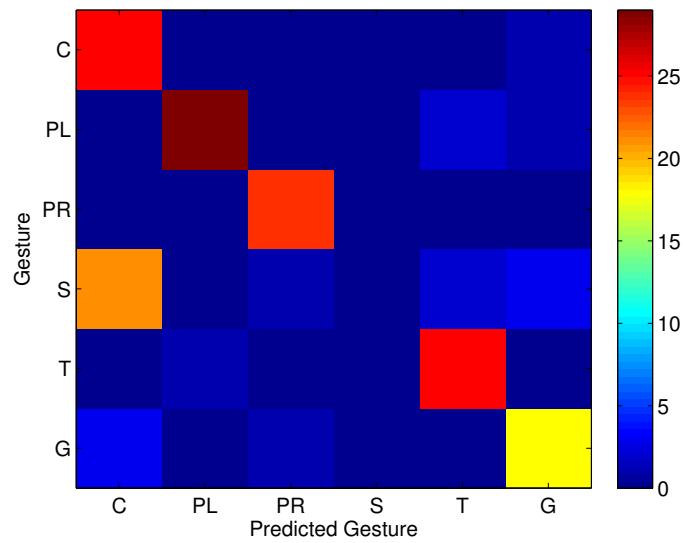


Figure 7.7: Misclassification results from the 1st trial for set 1 (157 sequences).

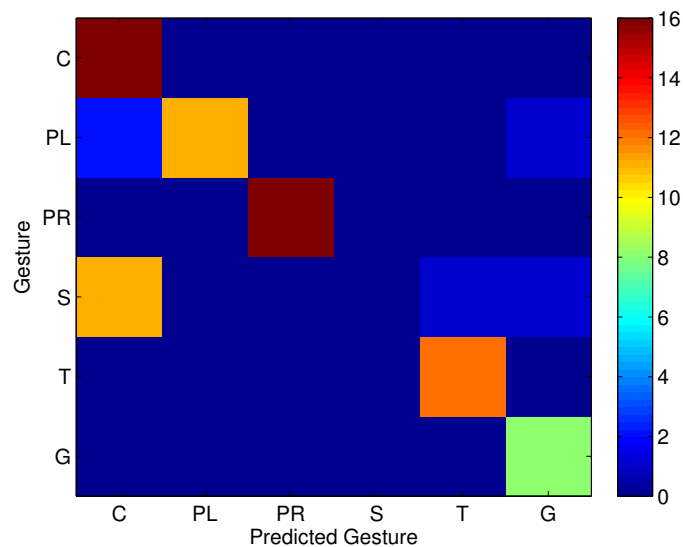


Figure 7.8: Misclassification results from the 1st trial for set 2 (79 sequences).

7.2.2 Graph Theoretic Analysis

Despite the performance, the topology provides also measures to characterize the resultant reservoir. From a graph-theoretic perspective, the reservoir is a directed graph including loops or, respectively, self-reference. This view on the neural

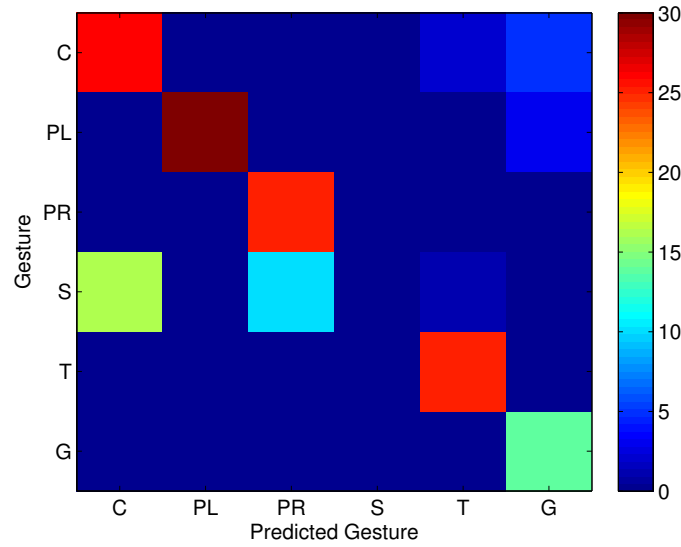


Figure 7.9: Misclassification results from the 6th trial for set 1 (157 sequences).

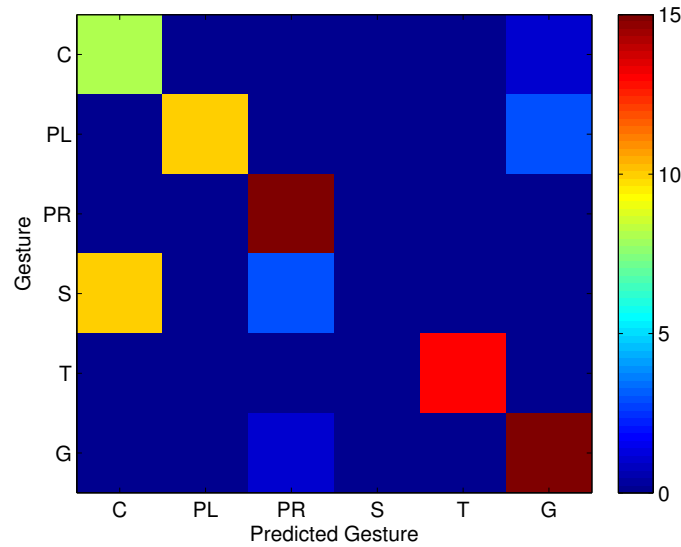


Figure 7.10: Misclassification results from the 6th trial for set 2 (79 sequences).

connectivity allows characterization of the emerging topology from our pruning algorithm. In this context, synaptic connections are used synonymously for *edges*, while neurons are the *vertices*. The connectivity of a vertex ν_i can be described by its degree, i.e. the sum of δ^- of incoming edges and δ^+ the number of outgoing connections. Based on the intuition that an outgoing edge of a vertex ν_i is an incoming edge of vertex ν_j , the corresponding degrees can be derived from the adjacency matrix A by either summing over the i th row of A (out-degree of vertex ν_i) or summing over the j th column (in-degree of vertex ν_j). A compact notation is provided in equation 7.2 and equation 7.3. The diagonal of A is set if neurons

have a self-reference, i.e. $a_{ii} = 1$, $a_{ii} \in A$. This loop adds 2 in the computation of the total vertex degree. Figure 7.11 gives an example of what has been said before.

$$\delta^-(\nu_j) = \sum_{k=1}^n a_{kj} \quad (7.2)$$

$$\delta^+(\nu_i) = \sum_{k=1}^n a_{ik} \quad (7.3)$$

| | | | | | | | |
|---|------------|---|---|---|---|---|------------|
| | j | | | | | | |
| i | 0 | 1 | 0 | 0 | 0 | 1 | δ^+ |
| | 0 | 1 | 1 | 1 | 0 | 0 | |
| | 1 | 0 | 0 | 1 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 1 | 1 | |
| | 0 | 1 | 1 | 0 | 1 | 0 | |
| | 1 | 0 | 0 | 0 | 1 | 0 | |
| | δ^- | | | | | | |

Figure 7.11: Example of an adjacency matrix size 6×6 for a directed graph. A 1 is assigned whenever there is a connection between two nodes, else there is a 0 entry. A 1 on the diagonal (green) depicts a loop or self-reference ($i = j$). The out-degree δ^+ can be derived from the row, while the in-degree δ^- can be determined from the column.

The distribution of vertex degrees characterizes different topologies for graph structures. One major aspect of the brain is its functional separation, yielding neural clusters with a high local connectivity degree and so-called brain hubs, connecting information from different areas. The precuneus for example was identified as such a hub. Due to their functional role hubs can be identified by high values of their out-degree. Figures 7.12 7.13 show the out-degree δ^+ for two reservoirs with different sparsity factor κ . It shows that only a small fraction of neurons have a high out-degree, which indicates a clustered organization with hubs.

The property of a system comprising locally dense clusters and hubs is called *small world*, and research of the brain connectome showed evidence that our cortex has small world properties. The structure is then called accordingly small-world topology. An example of such a structure is the Watts-Strogatz model, situated between completely random graphs (Erdős-Rényi model) and scale-free

graphs (Barabási-Albert model, known for the preferential attachment in social networks¹).

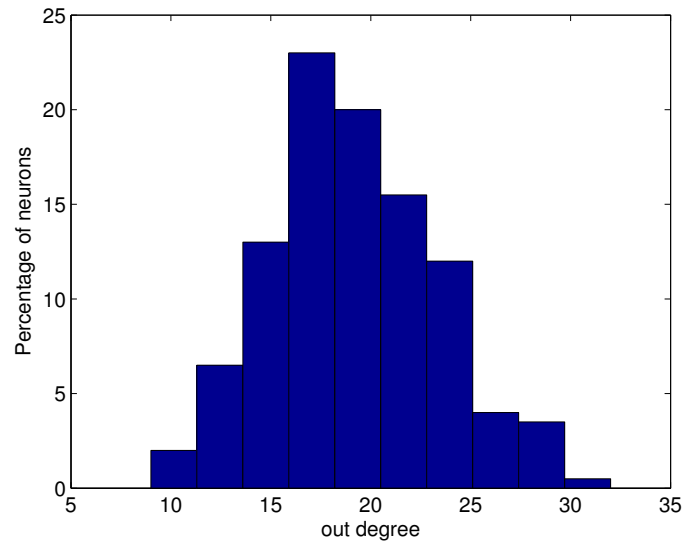


Figure 7.12: Distribution of the neuron out-degree for a reservoir of size $r_N = 100$ and sparsity $\kappa = 0.1$

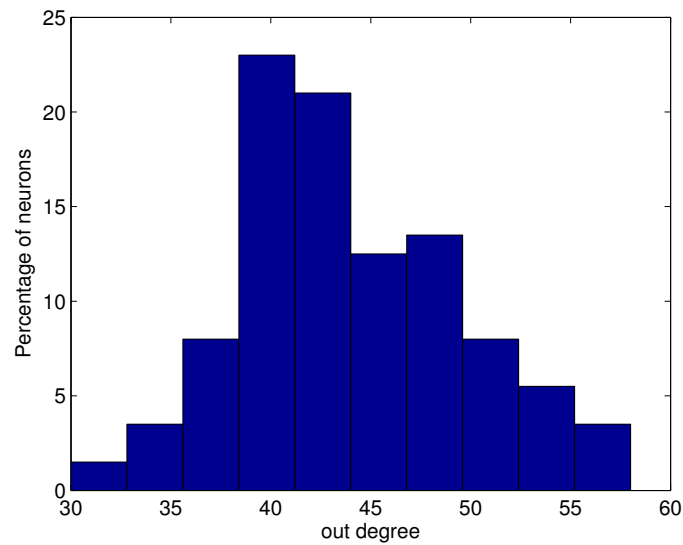


Figure 7.13: Distribution of the neuron out-degree for a reservoir of size $r_N = 100$ and sparsity $\kappa = 0.25$

Our pruning experiments did not give significant indications that the reservoir arrives at a certain topology. From the 10th experimental trial, which gave most of the pruned connections, we can only carefully state that a more exhaustive pruning

¹A popular example is the “6 degrees of Kevin Bacon”

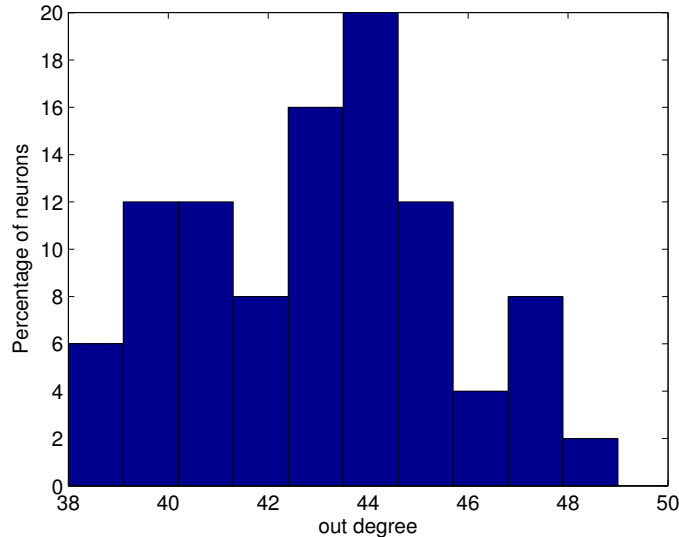


Figure 7.14: Distribution of the neuron out-degree from the reservoir of the 10th trial. With 348 pruned connections its sparsity is low. A clear separation of the different out-degrees compared to e.g Figure 7.13 is not visible. However, the number of neurons with a high out-degree is decreasing.

might lead to a reservoir topology similar to e.g. the *small world*. However, our experiments, which are the first addressing an input-driven optimization of the reservoir for the processing of different gestures, are a potential basis for further investigation on the complex dynamics for the task of gesture recognition.

7.3 Chapter Summary

In this chapter, we focused on the reservoir in different experiments. First, we showed that the orthogonal reservoir initialization decreases the undesired performance variability which is displayed by a random initialization. We also keep into consideration the biological inspiration of the reservoir. In particular, we were interested in the network behavior in case of synapse or neuron loss, which is typical in the human brain. The second experiment thus concentrated on the pruning and the effect on the gesture recognition in a *one-shot* learning scenario. We introduced a pruning strategy based on the pairwise correlation between reservoir activations. In contrast to Hebbian learning, which considers strengthening synapses between co-activated neurons, we use the correlation measure to prune synapse when the correlation is low. We showed, that the reservoir is able to discriminate the gestures when only considering prototypical sequences from our dataset. We observed an interesting behavior for the spectral radius ρ : the experiments revealed fluctu-

ations around the fixed value of 0.9, where especially the increase of ρ yielded a better performance for the two test sets. We suspect that pruning the input-driven reservoir might lead to interesting self-tuning mechanisms as is implemented with plasticity rules. Further, for the task of gesture recognition in a single reservoir, the sparsity factor might play a role, as the amount of pruned connections was rather low. We explain this by the role of connectivity on the complex dynamics in the reservoir.

Chapter 8

Thesis Discussion

The introduction of the *Reservoir Computing* paradigm shifted the learning of recurrent neural networks from gradient-based techniques to models with a functional separation between input representations through a random and untrained recurrent layer, and a memoryless readout using machine learning methods. Especially Echo State Networks (ESN), a concrete neural network implementation, received a lot of attention in the last years. In domains like language processing, robotics, and time-series prediction, ESNs were already successfully applied and evaluated on various data. Until today, however, only little is known about the processing and performance in vision-based tasks like gesture recognition, despite the fact that gestures are essential for human communication, and an integral part in human-machine interfaces and interactions with cognitive robots. Learning gestures employing the *Reservoir Computing* techniques benefit from the fast training of the recurrent layer in the network architecture and would consequently enrich various applications in robotics and the modeling of the complementary modalities language and gestures.

From our perspective, the lack of research into the topic of gesture recognition, especially for dynamic gestures, exploiting the benefits of Echo State Networks is surprising. A great number of studies in the research literature, preferably in the practical domain, highlighted the good ESN performance in sequencing tasks and emphasized their fast training and easy implementation in contrast to traditional learning in recurrent neural networks. However, from our literature research, we also experienced a gap between the theoretical requirements of the proper functioning of ESNs and experimental design of applications. The misconception of some of the ESN conditions or relevance of parameters created partially false arguments and explanations of the working principles of ESNs. Our research was, therefore, guided by the three principal questions:

- Are Echo State Networks an appropriate architecture to solve the task of gesture recognition?
- How can we visualize and quantify the recurrence in ESNs with gesture input, which gives insight into the network stability?
- Which reservoir modifications are beneficial and which properties emerge from the network for the task?

We summarize and discuss the particular motivations and research questions we addressed in our individual chapters. Furthermore, we suggest ideas derived from the thesis we find worth for future explorations. The last section concludes this thesis.

8.1 Gesture Representations

In our first experiments on gesture recognition described in chapter 5, we defined a set of command gesture sequences. They belong to the type of gestures, which can be understood without additional speech input and thus qualify to be an integral part in intelligent vision systems. Due to the lack of appropriate benchmark data on command gestures, we recorded our own dataset for further vision-based processing of the acquired image sequences. We believe that a gesture recognition system should allow the most natural gesture performance, thus we diverging from other works in this field using gloves (Nagi et al., 2011; Hikawa and Araga, 2011; Lamberti and Camastra, 2011), additional tools (Weber et al., 2008; Yan et al., 2010) or simplifications on the hand detection and hand tracking using a skeleton model on body joints (Parisi et al., 2014), which neglects the hand shape at all. As outlined in chapter 2, a large set of different methods are available to compute representative features from gesture sequences, resulting in feature sets of very different dimensionalities. In our research, we were interested in the legitimacy of the application of Echo State Networks for gesture recognition and we, therefore, asked:

What would be a “sufficient” gesture representation for our set of dynamic gestures? Are there differences in the network classification of Echo State Networks given distinct complexities of features?

To address the questions we decided to investigate two boundaries of gesture representations, which we called the *simple* feature set and the *complex* feature set.

The *simple* feature set was derived from a description of the hand gesture trajectory acquired from gesture performance recordings and the corresponding hand shape, emanating in a frame-wise feature vector (x, y, θ) . To extract these features we implemented the preprocessing stages necessary to obtain the hand performing the gesture. The components (x, y) represent the center of the hand, θ specifies the hand orientation.

In contrast, the *complex* feature set was obtained from a variant of a deep neural network architecture, the Multi Channel Convolution Neural Network (MCCNN) Barros et al. (2014). The advantage of such a network is that we can omit a preprocessing procedure. The features are directly computed by alternating filters in the network layers inspired by the processing of sensory information in the visual cortex. The drawbacks, however, which come along with the architecture are 1) a computationally intensive training for the correct tuning of the filters and 2) the resultant features are abstract representations of the input images and are thus not accessible to human interpretation. For our purpose, we assumed that these features represent particular image coefficients for our gesture sequences. We captured a 70-dimensional feature vector from the final layer of the MCCNN architecture, hence the name *complex*.

We adopted the ensemble ESNs approach introduced by Jaeger et al. (2007) for learning the gestures. The idea to use differently sized reservoirs provided a unifying architecture for a comparison of the two distinct feature sets. As the aim of this chapter was to investigate the gesture representations and their influence on the classification abilities, we concentrated on three main parameters as suggested in the literature. This way, our work allows comparison with other data from a similar problem domain.

We chose the number of misclassification as the evaluation measure over all experiments. Our results showed that both gesture representations are suitable for the gesture recognition task. However, we observed differences in the particular processing of the feature sets. Our results revealed that the *complex* feature set benefits from the ensemble ESN structure. Contrary to this observation, the *simple* feature set achieved best results when the ESN comprised only one reservoir. Our experiments suggest that in time-critical applications, e.g. in interaction with a robot, a rather simplistic gesture representation using a single reservoir is a valid approach to achieve a good gesture recognition performance in feasible time. However, we constrain this statement to a setting with controlled experiments conditions and a clear gesture protocol on the correct gesture performance as introduced in the chapter. We expect that for a larger set of subjects and

environmental changes like the illumination and varying camera perspective, the *simple* feature set would insufficiently represent the distinct gestures, which will cause an error-prone classification. In addition, fewer specifications on the gesture execution would introduce more *intra* and *inter-subject variability* as identified for our gesture set. Therefore, our experiments suggest that the proposed scheme of obtaining invariant, representative features from a deep neural network as input to ensemble ESNs is advantageous for a robust recognition under challenging experimental settings and for gestures performed “in the wild”.

8.2 Visualization and Quantification of Reservoir Dynamics

A key functional characteristic of Echo State Networks is their stability. Only if a network is stable it can be guaranteed to produce reliable results. The conditions for the ESN stability were introduced by Jaeger (2001a) as the “echo state property” (ESP), which basically recontextualize concepts from dynamical system theory. The ESP indicates how to properly configure an ESN to obtain contractive dynamics, where usually the spectral radius and the singular value of the reservoir matrix are considered. A controversial debate emerged in the research community about the ESP conditions (Buehner and Young, 2006; Ozturk et al., 2007; Yildiz et al., 2012; Caluwaerts et al., 2013), especially considering their uncritical application in practical tasks.

The different opinions about the network requirements and specific experimental settings for a particular task motivated us to diverge from the process of tweaking network parameters, but instead to unveil the reservoir dynamics when a network processes the gesture sequences. The argumentation about the ESP and our intuition about the temporal reservoir activation profiles for ESN motivated us to examine the introduction of recurrence plots (RP) for the analysis of both time-series and reservoir activations. The advantage of these plots is that they give a 2D representation of the underlying data, which provides information about the recurrence in phase space. From the line distribution of the plot, we also investigated the validity of recurrence quantification analysis (RQA) as a computational method complementary to the approximation of the Lyapunov exponent (Verstraeten and Schrauwen, 2009; Barancok and Farkas, 2014).

First, we suggested an approach for the extension of our original data to obtain sequence diversity. Then, we introduced the embedding-delay technique for the

analysis of the gesture sequences and showed that the RPs are a valuable tool for a qualitative judgment of the data. This technique highlights different characteristics like periodicity, signal noise or chaotic behavior, which can give additional information on the data sources and the design of prediction or classification procedures. The subsequent computations of crucial RQA measures showed first evidence that they are useful indicators for the differences in the data, for instance, filtered and noisy sequences.

In a second step, we investigated the methodology on the reservoir for different settings of the spectral radius. We observe, that the reservoir captured the intrinsic factors of the input signals (periodic, chaotic) as long as the system is in a stable state. The representational power of the reservoir vanished with increasing instabilities, which we demonstrated both with the corresponding RP and RQA measures. We defined a criterion to compute when actually the network enters an unstable mode, which further supported our findings. From our results, we conclude that for the network design and determination of the dynamics the RQA approach is a relevant tool in ESN research. Finally, our work adds to the investigations introduced by (Bianchi et al., 2016b).

8.3 Reservoir Initialization and Pruning

The stochastic reservoir initialization and the role of the connectivity are another subjects of debate among ESN researchers. We showed evidence that only a little modification in the initialization process is beneficial for the prediction of gestures. The number of trials is usually set according to the experimenter’s knowledge or driven by any error criteria. However, keeping into consideration the special structure of other matrix types, here an orthogonal matrix, may facilitate the trial-and-error process.

We further elaborated on reservoir properties for gesture recognition using *one-shot* learning. We demonstrated that only by a selection of some prototype gestures the ESN was able to discriminate between gestures exhibiting more dynamics. A pruning strategy was introduced to shed light on the resultant sparsity of a reservoir because in the ESN literature a high sparsity value is recommended. However, our experiments showed that a rather dense network is beneficial for the task of gesture recognition using only a single reservoir. Increasing the number of reservoir neurons showed no significant improvements on the results.

Finally, we observed that the pruning procedure led to an increase in the spectral radius ρ , where the performance was best for the highest value. This observa-

tion might give an indication that the reservoir pruning might be a tool tuning a reservoir only by its input.

8.4 Limitations and Future Work

Our research was motivated by two factors: on the one hand, the task of gesture recognition receives more and more attention in the research areas of human-machine interfaces and cognitive robots. On the other hand, the introduction of the *Reservoir Computing* paradigm introducing a simplified training of recurrent neural networks attracted the use of corresponding network models like Echo State Networks for sequential processing. As the experimental experience and knowledge about important properties for vision-related tasks are sparse, we decided to limit our gesture data set to a small number of gestures in favor of analysis and interpretation of our results. An obvious extension of our work would be to establish a bigger command gesture corpus, and we encourage to use the tools presented in this thesis as well as to re-evaluate our findings and statements. Especially the observations for the pruning strategy should be more investigated for the tuning of an input-driven reservoir, substituting the manual tweaking of the reservoir matrix or using time-consuming search algorithms.

We emphasized the importance of gesture recognition in interaction scenarios. Therefore, another consequent step would be the application of gesture recognition in robot scenarios. Recent research demonstrated the effectiveness of Echo State Networks for language understanding on a humanoid robot, thus gestures would be a valuable complementary component for the further development of multi-modal, cognitive architectures. Moreover, a model for the coupling of language and gestures would stimulate research into developmental and embodied robotics to evaluate the role of gestures on language acquisition and other cognitive skills.

Finally, Echo State Networks can also be exploited as autonomous pattern generators for gesture production. All gestures presented in this thesis can be represented as functions, which can be learned in an ESN architecture with feedback. We encourage the investigations of resultant neural controllers and investigations on the stability on a humanoid robot to push forward a *proactive* human-robot interaction.

8.5 Conclusion

The thesis contributes to the research area of gesture recognition demonstrating the appropriateness of Echo State Networks for this task. The introduction of the recurrence analysis is a valuable tool complementing current studies on stability issues in the Echo State Network. The investigations on the gesture data and the representation, as well as keeping into consideration modifications of the reservoir might stimulate further research into the direction of effective network architectures following the *Reservoir Computing* paradigm and the successful application of gesture recognition.

Appendix A

Additional Technical Information and Equations

We provide technical details as well as equations for the benchmark data used for testing Echo State Networks as outlined throughout the thesis.

- The gestures were recorded with a standard web camera. The resultant video files in `avi` format were converted into images in `png` format using the `mplayer` (Linux OS):

```
for i in *avi; do mplayer ${i} -vo png; mv *png ${i%.*}; done
```

- Activation intervals (a, b) for the initialization of the random matrices:
 $M := a + (b - a) \cdot \text{rand}(\cdot, \cdot)$. Note, that with Matlab 7.7 and newer, usage the random generators with `seed` is substituted by `rng`.
- For the data filtering, we used the Matlab functions `filter` and `medfilt1` to smoothen our gesture sequences. The `filter` operates along 1D creating a moving-average mechanism for the individual x and y streams. The filter operation relies on the definition of a rational transfer function on the input, where coefficients for the numerator n and denominator m have to be chosen beforehand. After some experimentations we set $n = 3$ and $m = 1$. The median filter also smoothes the data by computing the median of three values. The choice of the filter depends on the underlying data and we chose to use the median filter for all gestures except for the *turn* gestures, which were filtered using the `filter` function.
- The n -th order Nonlinear Autoregressive Moving Average function (NARMA). For benchmark tests a typical choice is $n = 10$ and $n = 30$:

$$y(t+1) = 0.3y(t) + 0.05y(t) \sum_{i=0}^n y(t-i) + 1.5x(t-n)s(t) + 0.1 \quad (\text{A.1})$$

where $s(t)$ is the input at time t randomly sampled from a uniform distribution and $y(t+1)$ the output, which is the target for prediction tasks in neural networks. The inherent structure of producible timeseries is challenging due to the demand of nonlinearity and long memory.

- The Mackey Glass differential equation is computed as:

$$\frac{dx}{dt} = \beta \frac{x(t-\tau)}{1 + (x(t-\tau))^n - \gamma x(t)} \quad (\text{A.2})$$

where β and γ are two nonnegative parameters (e.g. set to $\beta = 0.25$ and $\gamma = 0.1$), n is a shape parameter (e.g. $n = 10$) and τ is the nonnegative time delay. It is common to set $\tau = 17$ to create a mildly chaotic dynamical system, and $\tau = 30$ for strong chaos, exhibiting a hard task for prediction.

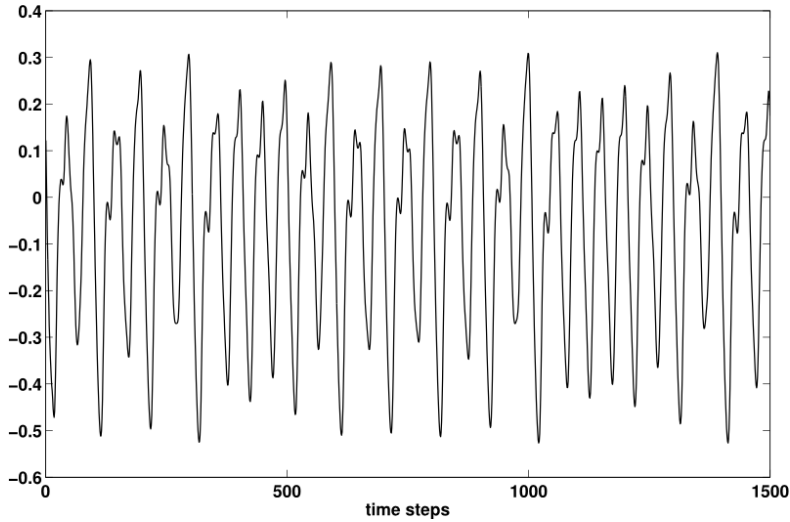


Figure A.1: Mackey-Glass time series with $\tau = 17$.

- The multiple superimposed oscillator (MSO) task describes a coupled system of sinusoidal signals with noninteger frequencies:

$$y(t) = \sin(0.2t) + \sin(0.311t) + \sin(0.42t) + \sin(0.51t) + \sin(0.63t) + \sin(0.74t) \quad (\text{A.3})$$

Appendix B

Publications Originating from this Thesis

- Jirak, D., Barros, P., Wermter, S. Dynamic gesture recognition using Echo State Networks. Proceedings of 23th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN'15, pp. 475-480, Bruges, Belgium, 2015.

Other publications include the research area of computer vision, gesture recognition and sequence modeling

- Barros, P., Jirak, D., Weber, C., Wermter, S. Multimodal emotional state recognition using sequence-dependent deep hierarchical features. Neural Networks, Volume 72, Pages 140-151, December, 2015.
- Barros, P., Parisi, G. I., Jirak D. and Wermter, S. Real-time gesture recognition using a humanoid robot with a deep neural architecture. Proceedings of the IEEE-RAS International Conference on Humanoid Robots (Humanoids 2014), pp. 83-88, Spain, 2014.
- Parisi, G. I., Jirak, D., Wermter, S. HandSOM: Neural clustering of hand motion for gesture recognition in real time. Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2014), pp. 981-986, Edinburgh, Scotland, UK, 2014.
- Hamester, D., Jirak, D., Wermter, S. Improved estimation of hand postures using depth images. Proceedings of the 16th International Conference on Advanced Robotics (ICAR 2013), Montevideo, UY, November 2013.

- Meins, N., Jirak, D., Weber, C., Wermter, S. Adaboost and Hopfield Neural Networks on different image representations for robust face detection. Proceedings of the 12th International Conference on Hybrid Intelligent Systems (HIS 2012), pp. 531-536, IEEE. Pune, IN, December 2012.

Other publications not included in this thesis

- Sakreida K., Efnert I., Thill S., Menz M.M., Jirak D., Eickhoff C.R., Ziemke T., Eickhoff S.B., Borghi A.M., Binkofski F., Affordance processing in segregated parieto-frontal dorsal stream sub-pathways *Neuroscience & Biobehavioral Reviews*, Vol. 69, pp. 89-112, Elsevier October 2016
- Jirak, D., Menz, M.M., Buccino, G., Borghi, A., Binkofski, F. Grasping language - A short story on embodiment. *Consciousness & Cognition*. Vol. 19(3), pp. 711-720, Elsevier, 2010.

Appendix C

Acknowledgements

Firstly, I would like to express my deepest gratitude to my advisor Prof. Stefan Wermter for his continuous support and encouragement of my Ph.D. study. I am particularly thankful for his valuable advises in all the phases of my research, which will also now guide my future research. Being part of his group gave me the precious opportunity to work in an international team, to take over responsibility in university teaching, and to benefit from his experience and knowledge to work on my academic career.

My sincere thanks also go to my reviewer Prof. H. Siegfried Stiehl. I am more than happy that he kept his interest into my research and encouraged my way of thinking. His feedback on my thesis and his interesting scientific questions are highly valuable for my next steps in research.

I would like to also thank my thesis committee chair Prof. Timo Gerkmann for his insightful comments on my work broadening my research perspective, and for his suggestions considering my life in academia.

I have greatly benefited from the scientific spirit in the group and interesting discussions with my lab fellows. I am particularly grateful for the diverse projects I could realize during my studies and the inspiring talks at the coffee machine. It is a pleasure and honor to work with you guys.

Ich möchte insbesondere meinen Freunden danken, die mir während der ganzen Zeit bedingungslos zur Seite standen (und stehen). Ich danke vorallem Johanna für ihr Einfühlungsvermögen und ihr offenes Ohr. Unsere Gespräche haben mir stets geholfen, die Dinge wieder im richtigen Licht zu betrachten und meinen Weg weiterzugehen. Romy danke ich für eine fast 20 Jahre währende Freundschaft; ihr Humor und ihre offene, liebe Art haben mich immer wieder neu aufgebaut. Bei Christian bedanke ich mich für seine immense Geduld mit mir und für all die

entspannenden Dinnerabende. Meinen Freunden Benni und Philip danke ich nicht nur ganz herzlich für eine aufregende gemeinsame Studienzeit, sondern auch für ihren stetigen Rückhalt und wunderbare Gespräche.

A minha profunda gratidão e apreciação vai para Pablo, que sempre me ajudou e encorajou. Quero agradecê-lo, especialmente, por sua paciência comigo em todas as fases estressantes. Não poderia ser mais abençoada por poder compartilhar minha vida com você. Seu amor é meu lar, ao qual sempre posso retornar. Eu amo você.

Bibliography

- Alibali, M. W. and DiRusso, A. A. (1999). The function of gesture in learning to count: more than keeping track. *Cognitive Development*, 14(1):37 – 56.
- Andres, M., Michaux, N., and Pesenti, M. (2012). Common substrate for mental arithmetic and finger representation in the parietal cortex. *NeuroImage*, 62(3):1520 – 1528.
- Atiya, A. F. and Parlos, A. G. (2000). New results on recurrent network training: Unifying the algorithms and accelerating convergence. *IEEE Transactions on Neural Networks*, 11(3):697–709.
- Babloyantz, A. and Loureno, C. (1994). Computation with chaos: a paradigm for cortical activity. *Proceedings of the National Academy of Sciences of the United States of America*, 91(19):9027–9031.
- Bak, P., Tang, C., and Wiesenfeld, K. (1988). Self-organized criticality. *Physical Review A*, 38:364–374.
- Barak, O., Sussillo, D., Romo, R., Tsodyks, M., and Abbott, L. F. (2013). From fixed points to chaos: Three models of delayed discrimination. *Progress in Neurobiology*, 103.
- Barancok, P. and Farkas, I. (2014). Memory capacity of input-driven echo state networks at the edge of chaos. In Wermter, S., Weber, C., Duch, W., Honkela, T., Koprinkova-Hristova, P., Magg, S., Palm, G., and Villa, A., editors, *Artificial Neural Networks and Machine Learning ICANN 2014*, volume 8681 of *Lecture Notes in Computer Science*, pages 41–48. Springer International Publishing.
- Barone, P. and Joseph, J. P. (1989). Prefrontal cortex and spatial sequencing in macaque monkey. *Experimental Brain Research*, 78(3):447–464.
- Barros, P., Magg, S., Weber, C., and Wermter, S. (2014). A multichannel convolutional neural network for hand posture recognition. In Wermter, S., Weber, C.,

- Duch, W., Honkela, T., Koprinkova-Hristova, P., Magg, S., Palm, G., and Villa, A., editors, *Artificial Neural Networks and Machine Learning ICANN 2014*, volume 8681 of *Lecture Notes in Computer Science*, pages 403–410. Springer International Publishing.
- Bartolo, A., Daumler, M., Sala, S. D., and Goldenberg, G. (2007). Relationship between object-related gestures and the fractionated object knowledge system. *Behavioral Neurology*, 18(3):143–147.
- Bengio, Y., Boulanger-Lewandowski, N., and Pascanu, R. (2013). Advances in optimizing recurrent networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8624–8628.
- Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.
- Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305.
- Berteletti, I. and Booth, J. R. (2015). Perceiving fingers in single-digit arithmetic problems. *Frontiers in Psychology*, 6:226.
- Bertschinger, N. and Natschlaeger, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, 16(7):1413–1436.
- Bianchi, F. M., Livi, L., and Alippi, C. (2016a). Investigating echo state networks dynamics by means of recurrence analysis. *CoRR*, abs/1601.07381.
- Bianchi, F. M., Livi, L., and Alippi, C. (2016b). Investigating echo-state networks dynamics by means of recurrence analysis. *IEEE Transactions on Neural Networks and Learning Systems*, PP(99):1–13.
- Boedecker, J., Obst, O., Lizier, J. T., Mayer, N. M., and Asada, M. (2012). Information processing in echo state networks at the edge of chaos. *Theory in Biosciences*, 131.
- Boedecker, J., Obst, O., Mayer, N. M., and Asada, M. (2009). Initialization and self-organized optimization of recurrent neural network connectivity. *HFSP Journal*, 3(5):340–349.

- Boström, K. J., Wagner, H., Prieske, M., and de Lussanet, M. (2013). Model for a flexible motor memory based on a self-active recurrent neural network. *Human Movement Science*, 32(5):880–898.
- Bruhn, A., Weickert, J., and Schnörr, C. (2005). Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231.
- Buckley, C. L. and Nowotny, T. (2012). Transient dynamics between displaced fixed points: An alternate nonlinear dynamical framework for olfaction. *Brain Research*, 1434:62 – 72. Selected papers presented at the International Workshop on Neural Coding, Limassol, Cyprus, 29 October - 3 November 2010.
- Buehner, M. and Young, P. (2006). A tighter bound for the echo state property. *IEEE Transactions on Neural Networks*, 17(3):820–824.
- Butcher, J., Verstraeten, D., Schrauwen, B., Day, C., and Haycock, P. (2010). Extending reservoir computing with random static projections: a hybrid between extreme learning and RC. In *European Symposium on Artificial Neural Networks, 18th, Proceedings*, pages 303–308. D-Side.
- Caluwaerts, K., Wyffels, F., Dieleman, S., and Schrauwen, B. (2013). The spectral radius remains a valid indicator of the echo state property for large reservoirs. In *The 2013 International Joint Conference on Neural Networks*, pages 1–6.
- Cappuccio, M. L., Chu, M., and Kita, S. (2013). Pointing as an instrumental gesture: Gaze representation through indication. *Humana.Mente: Journal of Philosophical Studies*, 24:125–149.
- Cohn, T. and Blunsom, P. (2005). Semantic role labelling with tree conditional random fields. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 169–172, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Conseil, S., Bourennane, S., and Martin, L. (2007). Comparison of fourier descriptors and hu moments for hand posture recognition. In *Signal Processing Conference, 2007 15th European*, pages 1960–1964.
- Dasgupta, S., Wörgötter, F., and Manoonpong, P. (2013). Information dynamics based self-adaptive reservoir for delay temporal memory tasks. *Evolving Systems*, 4(4):235–249.

- de Ruiter, J. (1995). Why do people gesture at the telephone? In Biemans, M. and Woutersen, M., editors, *Proceedings of the CLS opening Academic Year 1995-1996*.
- Deihimi, A. and Showkati, H. (2012). Application of echo state networks in short-term electric load forecasting. *Energy*, 39(1):327 – 340. Sustainable Energy and Environmental Protection 2010.
- Do, T. M. T. and Artieres, T. (2010). Neural conditional random fields. *Journal of Machine Learning Research - Proceedings Track*, 9:177–184.
- Dominey, P. (2013). Recurrent temporal networks and language acquisition from corticostriatal neurophysiology to reservoir computing. *Frontiers in Psychology*, 4:500.
- Dominey, P. F. (1995). Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. *Biological Cybernetics*, 73(3):265–274.
- Dominey, P. F. (2005). From sensorimotor sequence to grammatical construction: Evidence from simulation and neurophysiology. *Adaptive Behavior*, 13:347–361.
- Dominey, P. F., Hoen, M., and Inui, T. (2006). A neurolinguistic model of grammatical construction processing. *Journal of Cognitive Neuroscience*, 18(12):2088–2107.
- Doya, K. (1992). Bifurcations in the learning of recurrent neural networks. In *IEEE International Symposium on Circuits and Systems*, pages 2777–2780.
- Durstewitz, D. and Deco, G. (2008). Computational significance of transient dynamics in cortical networks. *European Journal of Neuroscience*, 27(1):217–227.
- Durstewitz, D., Seamans, J. K., and Sejnowski, T. J. (2000). Neurocomputational models of working memory. *Nature Neuroscience*, 3 supp:1184–1191.
- Dutoit, X., Schrauwen, B., Van Campenhout, J., Stroobandt, D., Van Brussel, H., and Nuttin, M. (2008). Pruning and regularization in reservoir computing: a first insight. In *Proceedings of 16th European Symposium on Artificial Neural Networks*, page 6. d-side Publications.
- Eckmann, J. P., Kamphorst, O. S., and Ruelle, D. (1987). Recurrence plots of dynamical systems. *Europhysics Letters*, 4:973+.

- Eguíluz, V. M., Chialvo, D. R., Cecchi, G. A., Baliki, M., and Apkarian, A. V. (2005). Scale-free brain functional networks. *Physical Review Letters*, 94:018102.
- Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.
- Elmezain, M., Al-Hamadi, A., Sadek, S., and Michaelis, B. (2010). Robust methods for hand gesture spotting and recognition using hidden markov models and conditional random fields. In *Proceedings IEEE International Signal Processing and Information Technology*, pages 131–136.
- Enel, P., Procyk, E., Quilodran, R., and Dominey, P. F. (2016). Reservoir computing properties of neural dynamics in prefrontal cortex. *PLoS Computational Biology*, 12(6):1–35.
- Fair, D. A., Cohen, A. L., Dosenbach, N. U. F., Church, J. A., Miezin, F. M., Barch, D. M., Raichle, M. E., Petersen, S. E., and Schlaggar, B. L. (2008). The maturing architecture of the brain’s default network. *Proceedings of the National Academy of Sciences*, 105(10):4028–4032.
- Faure, P. and Korn, H. (2001). Is there chaos in the brain? i. concepts of nonlinear dynamics and methods of investigation. *Comptes Rendus de l’Académie des Sciences - Series III - Sciences de la Vie*, 324(9):773–793.
- Ferreira, A. A., Ludermir, T. B., and de Aquino, R. R. B. (2013). An approach to reservoir computing design and training. *Expert Systems with Applications*, 40(10):4172–4182.
- FitzHugh, R. (1955). Mathematical models of threshold phenomena in the nerve membrane. *The bulletin of mathematical biophysics*, 17(4):257–278.
- Gallese, V., Fadiga, L., Fogassi, L., and Rizzolatti, G. (1996). Action recognition in the premotor cortex. *Brain*, 119 (Pt 2):593–609.
- Gallicchio, C. and Micheli, A. (2011). Architectural and markovian factors of echo state networks. *Neural Networks*, 24(5):440–456.
- Ganguli, S., Huh, D., and Sompolinsky, H. (2008). Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences*, 105(48):18970–18975.
- Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. New Jersey: Lawrence Erlbaum Associates.

- Goldin-Meadow, S. (2003). *Hearing gesture: how our hands help us think*. Belknap Press of Harvard University Press Cambridge, MA 2003.
- Goldin-Meadow, S. and Wagner, S. M. (2005). How our hands help us learn. *Trends in Cognitive Sciences*, 9(5):234 – 241.
- Goodale, M. A. (2011). Transforming vision into action. *Vision Research*, 51(13):1567 – 1587. Vision Research 50th Anniversary Issue: Part 2.
- Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEEE Proceedings F (Radar and Signal Processing)*, 140:107–113(6).
- Grassberger, P. and Procaccia, I. (1983). Measuring the strangeness of strange attractors. *Physica D Nonlinear Phenomena*, 9:189–208.
- Grossberg, S. (1968). Some nonlinear networks capable of learning a spatial pattern of arbitrary complexity. *Proceedings of the National Academy of Sciences*, 59(2):368–372.
- Hamester, D., Jirak, D., and Wermter, S. (2013). Improved estimation of hand postures using depth images. In *International Conference on Advanced Robotics*, pages 1–6.
- Hammersley, J. M. and Clifford, P. E. (1971). Markov random fields on finite graphs and lattices. Unpublished manuscript.
- Hamzei, F., Rijntjes, M., Dettmers, C., Glauche, V., Weiller, C., and Büchel, C. (2003). The human action recognition system and its relationship to Broca’s area: an fMRI study. *Neuroimage*, 19(3):637–644.
- Hartland, C. and Bredeche, N. (2007). Using echo state networks for robot navigation behavior acquisition. In *IEEE International Conference on Robotics and Biomimetics*, pages 201–206.
- Haykin, S. (1996). *Adaptive Filter Theory (3rd Ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Hebb, D. O. (1949). *The Organization of Behavior*. Wiley, New York.
- Hellbach, S., Strauss, S., Eggert, J. P., Körner, E., and Gross, H.-M. (2008). Echo state networks for online prediction of movement data — comparing investigations. In *Proceedings of International Conference on Artificial Neural Networks Part I*, pages 710–719, Berlin, Heidelberg. Springer-Verlag.

- Hermans, M. and Schrauwen, B. (2010). Memory in reservoirs for high dimensional input. In *International Joint Conference on Neural Networks*, pages 1–7.
- Hikawa, H. and Araga, Y. (2011). Study on gesture recognition system using posture classifier and jordan recurrent neural network. In *Proceedings of International Joint Conference on Neural Networks*, pages 405–412.
- Hinaut, X. and Dominey, P. F. (2013). Real-time parallel processing of grammatical structure in the fronto-striatal system: A recurrent network simulation study using reservoir computing. *PloS one*, 8(2):e52946.
- Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In Kremer and Kolen, editors, *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press.
- Hodgkin, A. L., A. L. and Huxley, A. F., A. F. (1952). Propagation of electrical signals along giant nerve fibers. *Proceedings of the Royal Society B Biological Sciences*, 140(899):177–183.
- Hoerzer, G. M., Legenstein, R., and Maass, W. (2012). Emergence of complex computational structures from chaotic neural networks through reward-modulated hebbian learning. *Cerebral Cortex*.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558.
- Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.
- Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory*, 8(2):179–187.
- Iason Oikonomidis, N. K. and Argyros, A. (2011). Efficient model-based 3d tracking of hand articulations using Kinect. In *Proceedings of the British Machine Vision Conference*, pages 101.1–101.11. BMVA Press.
- Iwanski, J. S. and Bradley, E. (1998). Recurrence plots of experimental data: To embed or not to embed? *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 8(4):861–871.

- Izhikevich, E. M. (2007). *Dynamical systems in neuroscience: the geometry of excitability and bursting*. MIT press.
- Jaeger, H. (2001a). The echo state approach to analysing and training recurrent neural networks - with an erratum note. Technical report, German National Research Center for Information Technology.
- Jaeger, H. (2001b). Short term memory in echo state networks. GMD-Report 152, GMD - German National Research Institute for Computer Science.
- Jaeger, H. (2002). Tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the "echo state network" approach. Technical report, German National Research Center for Information Technology.
- Jaeger, H. and Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80.
- Jaeger, H., Lukoševičius, M., Popovici, D., and Siewert, U. (2007). Optimization and applications of echo state networks with leaky- integrator neurons. *Neural Networks*, 20(3):335 – 352. Echo State Networks and Liquid State Machines.
- Jing, L., Zhou, Y., Cheng, Z., and Huang, T. (2012). Magic ring: A finger-worn device for multiple appliances control using static finger gestures. *Sensors*, 12(5):5775.
- Jordan, M. I. (1986). Serial order: A parallel, distributed processing approach. Technical Report 8604, Institute for Cognitive Science, University of California, San Diego.
- Kakumanu, P., Makrogiannis, S., and Bourbakis, N. (2007). A survey of skin-color modeling and detection methods. *Pattern Recognition*, 40(3):1106–1122.
- Kamarainen, J. K., Kyrki, V., and Kalviainen, H. (2006). Invariance properties of gabor filter-based features-overview and applications. *IEEE Transactions on Image Processing*, 15(5):1088–1099.
- Kelly, S. D., Ozyürek, A., and Maris, E. (2010). Two sides of the same coin: speech and gesture mutually interact to enhance comprehension. *Psychological Science*, 21(2):260–267.
- Kendon, A. (1983). The study of gesture: Some remarks on its history. In Deely, J. N. and Lenhart, M. D., editors, *Semiotics 1981*, pages 153–164. Springer US, Boston, MA.

-
- Kennel, M. B., Brown, R., and Abarbanel, H. D. I. (1992). Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*, 45:3403–3411.
- Kohler, E., Keysers, C., Umiltà, M. A., Fogassi, L., Gallese, V., and Rizzolatti, G. (2002). Hearing sounds, understanding actions: action representation in mirror neurons. *Science*, 297(5582):846–848.
- Korn, H. and Faure, P. (2003). Is there chaos in the brain? ii. experimental evidence and related models. *Comptes Rendus Biologies*, 326(9):787–840.
- Koryakin, D., Lohmann, J., and Butz, M. V. (2012). Balanced echo state networks. *Neural Networks*, 36:35–45.
- Kroliczak, G., Cavina-Pratesi, C., Goodman, D. A., and Culham, J. C. (2007). What does the brain do when you fake it? an fMRI study of pantomimed and real grasping. *Journal of Neurophysiology*, 97(3):2410–2422.
- Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01*, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Laje, R. and Buonomano, D. V. (2013). Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nature Neuroscience*, 16.
- Lamberti, L. and Camastra, F. (2011). Real-time hand gesture recognition using a color glove. In Maino, G. and Foresti, G. L., editors, *International Conference on Image Analysis and Processing*, pages 365–373, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Langton, C. G. (1990). Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1):12 – 37.
- Legenstein, R. and Maass, W. (2007). What makes a dynamical system computationally powerful? In Haykin, S., Principe, J. C., Sejnowski, T. J., and McWhirter, J. G., editors, *New Directions in Statistical Signal Processing: From System to Brains*, pages 127–154. MIT Press.
- Levina, A., Herrmann, J. M., and Geisel, T. (2007). Dynamical synapses causing self-organized criticality in neural networks. *Nature Physics*, 3.

- Li, S. Z. (2009). *Markov Random Field Modeling in Image Analysis*. Springer Publishing Company, Incorporated, 3rd edition.
- Liszkowski, U. (2014). Two sources of meaning in infant communication: preceding action contexts and act-accompanying characteristics. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 369(1651).
- Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial intelligence*, pages 674–679.
- Lukoševičius, M. and Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149.
- Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560.
- Mante, V., Sussillo, D., Shenoy, K. V., and Newsome, W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84.
- March, T., Chapman, S., and Dendy, R. (2005). Recurrence plot statistics and the effect of embedding. *Physica D: Nonlinear Phenomena*, 200(1):171–184.
- Martens, J. and Sutskever, I. (2011). Learning recurrent neural networks with hessian-free optimization. In Getoor, L. and Scheffer, T., editors, *International Conference on Machine Learning*, pages 1033–1040. Omnipress.
- Marwan, N. (2008). A historical review of recurrence plots. *The European Physical Journal Special Topics*, 164(1):3–12.
- Marwan, N. (2011). How to avoid potential pitfalls in recurrence plot based data analysis. *International Journal of Bifurcation and Chaos*, 21(4):1003–1017.
- Marwan, N. and Kurths, J. (2005). Line structures in recurrence plots. *Physics Letters A*, 336(45):349 – 357.
- Marwan, N. and Meinke, A. (2004). Extended recurrence plot analysis and its application to ERP data. *International Journal of Bifurcation and Chaos*, 14(02):761–771.

- Marwan, N., Romano, M. C., Thiel, M., and Kurths, J. (2007). Recurrence plots for the analysis of complex systems. *Physics Reports*, 438(56):237 – 329.
- Marwan, N., Wessel, N., Meyerfeldt, U., Schirdewan, A., and Kurths, J. (2002). Recurrence-plot-based measures of complexity and their application to heart-rate-variability data. *Physical Review E*, 66:026702.
- McCulloch, W. S. and Pitts, W. H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133.
- McNeill, D. (1992). *Hand and Mind: What Gestures Reveal about Thought*. University of Chicago Press, Chicago.
- Melinder, A. M. D., Konijnenberg, C., Hermansen, T., Daum, M. M., and Gredebäck, G. (2015). The developmental trajectory of pointing perception in the first year of life. *Experimental Brain Research*, 233(2):641–647.
- Miller, E., Li, L., and Desimone, R. (1991). A neural mechanism for working and recognition memory in inferior temporal cortex. *Science*, 254(5036):1377–1379.
- Milnor, J. (1985). On the concept of attractor. In Hunt, B., Li, T.-Y., Kennedy, J., and Nusse, H., editors, *The Theory of Chaotic Attractors*, pages 243–264. Springer New York.
- Morency, L.-P., Quattoni, A., and Darrell, T. (2007). Latent-dynamic discriminative models for continuous gesture recognition. In *Conference Proceedings IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8.
- Nagi, J., Ducatelle, F., Di Caro, G. A., Ciresan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., and Gambardella, L. M. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *Conference Proceedings IEEE International Signal and Image Processing Applications*, pages 342–347.
- Nesterov, Y. (1983). A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. *Doklady Akademii Nauk AN SSSR (translated as Soviet. MatDoklady)*, 269:543–547.
- Neverova, N., Wolf, C., Taylor, G. W., and Nebout, F. (2014). Hand segmentation with structured convolutional learning. In *Asian Conference on Computer Vision*.

- Novotni, M. and Klein, R. (2004). Shape retrieval using 3D zernike descriptors. *Computer-Aided Design*, 36(11):1047 – 1062. Solid Modeling Theory and Applications.
- Obst, O., Boedecker, J., and Asada, M. (2010). Improving recurrent neural network performance using transfer entropy. In Wong, K. W., Mendis, B. S. U., and Bouzerdoum, A., editors, *ICONIP 2010. Lecture Notes in Computer Science*, volume 6444, pages 193–200. Springer.
- Osiurak, F., Jarry, C., Baltenneck, N., Boudin, B., and Le Gall, D. (2012). Make a gesture and I will tell you what you are miming. pantomime recognition in healthy subjects. *Cortex*, 48(5):584–592.
- Otiniano-Rodriguez, K., Camara-Chavez, G., and Menotti, D. (2012). Hu and zernike moments for sign language recognition. In *The 2012 International Conference on Image Processing, Computer Vision, and Pattern Recognition*.
- Otte, S., Butz, M. V., Koryakin, D., Becker, F., Liwicki, M., and Zell, A. (2016). Optimizing recurrent reservoirs with neuro-evolution. *Neurocomputing*, 192:128 – 138. Advances in artificial neural networks, machine learning and computational intelligence Selected papers from the 23rd European Symposium on Artificial Neural Networks.
- Oubbati, M., Kord, B., and Palm, G. (2010). Learning robot-environment interaction using echo state networks. In Doncieux, S., Girard, B., Guillot, A., Hallam, J., Meyer, J.-A., and Mouret, J.-B., editors, *Proceedings From Animals to Animats 11: 11th International Conference on Simulation of Adaptive Behavior*, pages 501–510. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Ozturk, M. C., Xu, D., and Príncipe, J. C. (2007). Analysis and design of echo state networks. *Neural Computation*, 19(1):111–138.
- Parisi, G., Jirak, D., and Wermter, S. (2014). Handsom - neural clustering of hand motion for gesture recognition in real time. In *IEEE International Symposium on Robot and Human Interactive Communication*, pages 981–986.
- Pascanu, R. and Jaeger, H. (2011). A neurodynamical model for working memory. *Neural Networks*, 24(2):199–207.
- Priesemann, V., Wibral, M., Valderrama, M., Pröpper, R., Le Van Quyen, M., and Geisel, T. (2014). Spike avalanches in vivo suggest a driven, slightly subcritical brain state. *Frontiers in System Neurosciences*, 8.

- Prokhorov, D. (2005). Echo state networks: appeal and challenges. In *Proceedings IEEE International Joint Conference on Neural Networks, 2005.*, volume 3, pages 1463–1466 vol. 3.
- Qiao, J., Li, F., Han, H., and Li, W. (2016). Growing echo-state network with multiple subreservoirs. *IEEE Transactions on Neural Networks and Learning Systems*, PP(99):1–14.
- Quattoni, A., Collins, M., and Darrell, T. (2004). Conditional random fields for object recognition. In *Conference on Neural Information Processing Systems*, pages 1097–1104. MIT Press.
- Quattoni, A., Wang, S., Morency, L.-P., Collins, M., and Darrell, T. (2007). Hidden conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1848–1853.
- Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.
- Rabinovich, M., Huerta, R., and Laurent, G. (2008). Transient dynamics for neural processing. *Science*, 321(5885):48–50.
- Rabinovich, M. and Varona, P. (2011). Robust transient dynamics and brain functions. *Frontiers in Computational Neuroscience*, 5:24.
- Rad, A. A., Hasler, M., and Jalili, M. (2010). Reservoir optimization in recurrent neural networks using properties of Kronecker product. *Logic Journal of IGPL*, 18(5):670–685.
- Rigotti, M., Barak, O., Warden, M. R., Wang, X.-J., Daw, N. D., Miller, E. K., and Fusi, S. (2013). The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451):585–590.
- Rigotti, M., Ben Dayan Rubin, D., Wang, X.-J., and Fusi, S. (2010). Internal representation of task rules by recurrent dynamics: the importance of the diversity of neural responses. *Frontiers in Computational Neuroscience*, 4:24.
- Rizzolatti, G. and Craighero, L. (2004). The mirror-neuron system. *Annual Review of Neuroscience*, 27:169–192.
- Rizzolatti, G., Fadiga, L., Gallese, V., and Fogassi, L. (1996). Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3(2):131–141.

- Rodan, A. and Tino, P. (2011). Minimum complexity echo state network. *IEEE Transactions on Neural Networks*, 22(1):131–144.
- Rodan, A. and Tiño, P. (2010). Simple deterministically constructed recurrent neural networks. In *Proceedings of the 11th International Conference on Intelligent Data Engineering and Automated Learning*, pages 267–274, Berlin, Heidelberg. Springer-Verlag.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408.
- Scardapane, S., Nocco, G., Comminiello, D., Scarpiniti, M., and Uncini, A. (2014). An effective criterion for pruning reservoir’s connections in echo state networks. In *2014 International Joint Conference on Neural Networks*, pages 1205–1212.
- Schiller, U. D. and Steil, J. J. (2005). Analyzing the weight dynamics of recurrent learning algorithms. *Neurocomputing*, 63:5 – 23. New Aspects in Neurocomputing: 11th European Symposium on Artificial Neural Networks.
- Schlömer, T., Poppinga, B., Henze, N., and Boll, S. (2008). Gesture recognition with a wii controller. In *Proceedings of the 2Nd International Conference on Tangible and Embedded Interaction*, pages 11–14, New York, NY, USA. ACM.
- Schoener, G., Dose, M., and Engels, C. (1995). Moving the frontiers between robotics and biology dynamics of behavior: Theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, 16(2):213 – 245.
- Schrauwen, B., Defour, J., Verstraeten, D., and Van Campenhout, J. (2007). The introduction of time-scales in reservoir computing, applied to isolated digits recognition. In *Proceedings of the 17th International Conference on Artificial Neural Networks*, pages 471–479, Berlin, Heidelberg. Springer-Verlag.
- Seeley, W. W., Menon, V., Schatzberg, A. F., Keller, J., Glover, G. H., Kenna, H., Reiss, A. L., and Greicius, M. D. (2007). Dissociable intrinsic connectivity networks for salience processing and executive control. *Journal of Neuroscience*, 27(9):2349–2356.
- Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology -*

-
- Volume 1*, pages 134–141, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Shew, W. L., Clawson, W. P., Pobst, J., Karimipannah, Y., Wright, N. C., and Wessel, R. (2015). Adaptation to sensory input tunes visual cortex to criticality. *Nature Physics*, 11(8):659–663.
- Shew, W. L. and Plenz, D. (2013). The functional benefits of criticality in the cortex. *The Neuroscientist*, 19(1):88–100.
- Shimosaka, M., Mori, T., and Sato, T. (2007). Robust action recognition and segmentation with multi-task conditional random fields. In *IEEE Proceedings of International Conference of Robotics and Automation*, pages 3780–3786.
- Shutler, J. and Nixon, M. (2006). Zernike velocity moments for sequence-based description of moving features. *Image and Vision Computing*, 24(4):343 – 356.
- Siri, B., Berry, H., Cessac, B., Delord, B., and Quoy, M. (2008). A mathematical analysis of the effects of hebbian learning rules on the dynamics and structure of discrete-time random recurrent neural networks. *Neural Computation*, 20(12):2937–2966.
- Skarda, C. A. and Freeman, W. J. (1990). Chaos and the new science of the brain. *Concepts in Neuroscience*, 1(2):275–285.
- Song, Y., Demirdjian, D., and Davis, R. (2012). Continuous body and hand gesture recognition for natural human-computer interaction. *ACM Transactions on Interactive Intelligent Systems*, 2(1):5:1–5:28.
- Steil, J. J. (2007). Online reservoir adaptation by intrinsic plasticity for back-propagationdecorrelation and echo state learning. *Neural Networks*, 20(3):353 – 364.
- Stevenson, A. (2010). Oxford dictionary of english.
- Strauss, T., Wustlich, W., and Labahn, R. (2012). Design strategies for weight matrices of echo state networks. *Neural Computation*, 24(12):3246–3276.
- Sussillo, D. and Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557.

- Takens, F. (1981). Detecting strange attractors in turbulence. In Rand, D. and Young, L.-S., editors, *Dynamical Systems and Turbulence, Warwick 1980*, volume 898 of *Lecture Notes in Mathematics*, pages 366–381. Springer Berlin Heidelberg.
- Thomas Yeo, B. T., Krienen, F. M., Sepulcre, J., Sabuncu, M. R., Lashkari, D., Hollinshead, M., Roffman, J. L., Smoller, J. W., Zöllei, L., Polimeni, J. R., Fischl, B., Liu, H., and Buckner, R. L. (2011). The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *Journal of Neurophysiology*, 106(3):1125–1165.
- Tino, P., Cernansk, M., and Benuskov, L. (2004). Markovian architectural bias of recurrent neural networks. *IEEE Transactions on Neural Networks*, 15(1):6–15.
- Tomasello, M., Carpenter, M., and Liszkowski, U. (2007). A new look at infant pointing. *Child Development*, 78(3):705–722.
- Tomen, N. and Ernst, U. (2013). Phase transitions in cortical dynamics explain improved information processing under attention. *BMC Neuroscience*, 14(1):P126.
- Tomen, N., Rotermund, D., and Ernst, U. (2014). Marginally subcritical dynamics explain enhanced stimulus discriminability under attention. *Frontiers in Systems Neuroscience*, 8:151.
- Tong, M. H., Bickett, A. D., Christiansen, E. M., and Cottrell, G. W. (2007). Learning grammatical structure with echo state networks. *Neural Networks*, 20(3):424 – 432. Echo State Networks and Liquid State Machines.
- Triesch, J. (2007). Synergies between intrinsic and synaptic plasticity mechanisms. *Neural computation*, 19(4):885–909.
- Triesch, J. and von der Malsburg, C. (1996). Robust classification of hand postures against complex backgrounds. In *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition*, pages 170–, Washington, DC, USA. IEEE Computer Society.
- Triesch, J. and von der Malsburg, C. (2001). A system for person-independent hand posture recognition against complex backgrounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1449–1453.
- Tsuda, I. (2009). Hypotheses on the functional roles of chaotic transitory dynamics. *Chaos*, 19(1):015113.

- Turrigiano, G., Abbott, L., and Marder, E. (1994). Activity-dependent changes in the intrinsic properties of cultured neurons. *Science*, 264(5161):974–977.
- Ugur, E., Sahin, E., and Oztop, E. (2012). Self-discovery of motor primitives and learning grasp affordances. In *International Conference on Intelligent Robots and Systems*, pages 3260–3267.
- Ungerleider, L. G. and Mishkin, M. (1982). *Two Cortical Visual Systems*, chapter 18, pages 549–586. MIT Press.
- Vail, D. L., Veloso, M. M., and Lafferty, J. D. (2007). Conditional random fields for activity recognition. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 235:1–235:8, New York, NY, USA. ACM.
- Verstraeten, D. and Schrauwen, B. (2009). On the quantification of dynamics in reservoir computing. In Alippi, C., Polycarpou, M., Panayiotou, C., and Ellinas, G., editors, *Artificial Neural Networks ICANN 2009*, volume 5768 of *Lecture Notes in Computer Science*, pages 985–994. Springer Berlin Heidelberg.
- Verstraeten, D., Schrauwen, B., DHaene, M., and Stroobandt, D. (2007). An experimental unification of reservoir computing methods. *Neural Networks*, 20(3):391 – 403. Echo State Networks and Liquid State Machines.
- Vidal, V., Wolf, C., and Dupont, F. (2012). Combinatorial mesh optimization. *The Visual Computer*, 28(5):511–525.
- Vincent-Lamarre, P., Lajoie, G., and Thivierge, J.-P. (2015). Extreme sensitivity of reservoir computing to small network disruptions. *BMC Neuroscience*, 16(1):1–2.
- Vincent-Lamarre, P., Lajoie, G., and Thivierge, J.-P. (2016). Driving reservoir models with oscillations: a solution to the extreme structural sensitivity of chaotic networks. *Journal of Computational Neuroscience*, pages 1–18.
- Vogler, C. and Metaxas, D. (1999). Parallel hidden markov models for american sign language recognition. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 116–122 vol.1.
- Waegeman, T., Antonelo, E., Wyffels, F., and Schrauwen, B. (2009). Modular reservoir computing networks for imitation learning of multiple robot behaviors. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 27–32.

- Wallach, H. M. (2004). Conditional random fields: An introduction. Technical report, Department of Computer and Information Science, University of Pennsylvania.
- Wang, C., Komodakis, N., and Paragios, N. (2013). Markov random field modeling, inference & learning in computer vision & image understanding: A survey. *Computer Vision and Image Understanding*, 117(11):1610 – 1627.
- Wang, Y. and Ji, Q. (2005). A dynamic conditional random field model for object segmentation in image sequences. In *IEEE Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 264–270.
- Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of small-world networks. *Nature*, 393(6684):440–442.
- Webber, C. L. and Zbilut, J. P. (1994). Dynamical assessment of physiological systems and states using recurrence plot strategies. *Journal of Applied Physiology*, 76(2):965–973.
- Weber, C., Masui, K., Mayer, N. M., Triesch, J., and Asada, M. (2008). Reservoir computing for sensory prediction and classification in adaptive agents.
- Wei, C. Y. (2006). Not crazy, just talking on the phone: Gestures and mobile phone conversations. In *IEEE International Professional Communication Conference*, pages 299–307.
- White, O. L., Lee, D. D., and Sompolinsky, H. (2004). Short-term memory in orthogonal neural networks. *Physical Review Letters*, 92(14):148102.
- Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280.
- Wilson, A. D. and Bobick, A. F. (1999). Parametric hidden markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):884–900.
- Wolf, A., Swift, J. B., Swinney, H. L., and Vastano, J. A. (1985). Determining Lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*, 16(3):285 – 317.
- Xue, Y., Yang, L., and Haykin, S. (2007). Decoupled echo state networks with lateral inhibition. *Neural Networks*, 20(3):365–376.

- Yan, R., Tee, K. P., Chua, Y., and Tang, H. (2010). A gesture recognition system using localist attractor networks for human-robot interaction. In *IEEE Proceedings International Conference on Robotics and Biomimetics*, pages 1217–1222.
- Yildiz, I. B., Jaeger, H., and Kiebel, S. J. (2012). Re-visiting the echo state property. *Neural Networks*, 35:1–9.
- Zhang, W. and Linden, D. J. (2003). The other side of the engram: experience-driven changes in neuronal intrinsic excitability. *Nature Reviews Neuroscience*, 4:885–900.
- Zhong, P. and Wang, R. (2006). Object detection based on combination of conditional random field and markov random field. In *Proceedings International Conference on Pattern Recognition*, volume 3, pages 160–163.
- Zhu, C. and Sheng, W. (2009). Online hand gesture recognition using neural network based segmentation. In *Proceedings International Conference on Intelligent Robots and Systems*, pages 2415–2420.

Declaration of Oath

Eidesstattliche Versicherung

I hereby declare, on oath, that I have written the present 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.

Hamburg, Date 24th February 2017
City and Date
Ort und Datum

Doreen Jirak
Signature
Unterschrift

