Data Science for Neuroscience

The Brain as Inspiration, Model and Data Source

Inaugural-Dissertation

to obtain the academic degree

Doctor rerum naturalium (Dr. rer. nat.)

submitted to the Department of Biology, Chemistry and Pharmacy

of the Freie Universität Berlin

By

Christopher John Häusler

from Brisbane, Australia

2014

© Christopher John Häusler, 2014.

Typeset in LaTeX 2$_\varepsilon$.

The research presented in this dissertation was carried out from December 2010 until February 2014 at the Theoretical Neuroscience & Neuroinformatics group, Institute of Biology, Freie Universität Berlin, under the supervision of Prof. Dr. Martin P. Nawrot

1st Reviewer: Prof. Dr. Martin P. Nawrot - Freie Universität Berlin

2nd Reviewer: Prof. Dr. Manfred Opper - Technische Universität Berlin

Date of defense: 04/04/2014

Acknowledgments

First and foremost, I would like to thank my supervisor Prof. Martin Nawrot for his guidance, support, sense of humour and willingness to share a beer. I would like to THANK THANK THANK the members of the great 'neuro-inf' lab for creating a friendly, fun and stimulating work environment and for generally being great people, you've all grown very close to my heart. Particular thanks to Jan Soelter and Thomas Rost for being kaggle buddies and explaining all the concepts I didn't get. Thanks to Michael Schmuker for bringing me into the lab many years ago as a lowly masters student and for keeping our beloved compute servers up and running. Thanks to Farzad Farkhooi for sharing his office with me and understanding all things mathematical. Thank you to Prof. Manfred Opper for providing invaluable feedback at our committee meetings and letting me occupy space in your lab from time to time.

A big thank you to my experimental collaborators over the last 4 years, Björn Kampa, Patricia Molina, Andrea Kühn and especially Julius Hübl. You not only provided me with the valuable data and domain insight I needed for my work, but also with a supportive and collaborative environment and a great opportunity to see other working groups in action.

More thanks than I can mention to the BCCN and GRK 1589-1 for training me and funding me over the last few years, for welcoming me to Berlin and making me a much smarter person than I was when I arrived. Special call out to Vanessa Casagrande, Margret Franke and Julia Schäffer for providing incredible support to myself and everyone else who has been lucky enough to study at the BCCN during your tenure.

Thanks to Alex Susemihl for authoring 3 papers with me, being my NIPS companion and generally a great friend. Without your help, this thesis would have been many pages shorter. Also, thanks for introducing me Fernanda and Hugo, your awesome family who made for fun times in Berlin. We hope to see you soon!

A big thanks to Berlin for being a rad place to live, even if it is damn cold almost all of the time.

Of course, A Super Big Thanks to my family. To my parents and brother for always feeding me and supporting me in whatever I did and for not getting upset when I went to the other side of the world to live. And to Katrin and Florence, the two loves of my life who make everyday just damn fantastic!

List of Publications

This dissertation is based on the following three manuscripts:

Temporal Autoencoding Improves Generative Models of Time Series

Authors: Chris Häusler[*,1,2], Alex Susemihl[*,1,3], Martin P Nawrot[1,2], Manfred Opper[1,3]

Author Contributions: Research idea, execution, analysis and manuscript by CH and AS. Revision of the manuscript by MPN and MO

Manuscript Status: Submitted to Neurocomputing on December 17, 2013. http://www.journals.elsevier.com/neurocomputing/

Natural image sequences constrain dynamic receptive fields and imply a sparse code

Authors: Chris Häusler[*,1,2], Alex Susemihl[*,1,3], Martin P Nawrot[1,2]

Author Contributions: Research idea, execution, analysis and manuscript by CH, AS and MPN

Manuscript Status: Published. http://dx.doi.org/10.1016/j.brainres.2013.07.056

Decoding of voluntary hand movements in local field potential and extracellular spiking activity from patients with Parkinson disease and Dystonia

Authors: Chris Häusler[*,1,2], Julius Hübl[*,4], Andrea Kühn[4], Martin P Nawrot[1,2]

Author Contributions: Research Idea and Manuscript by CH, JH, MPN, AK. Analysis by CH, JH and MPN. Experiments by JH and AK

Manuscript Status: A revised version of this manuscript will be submitted for publication in an international peer reviewed journal.

Author Affiliations

1. Bernstein Centre for Computational Neuroscience, Berlin
2. Neuroinformatics & Theoretical Neuroscience, Institute of Biology, Freie Universität Berlin
3. Methods of Artificial Intelligence Group, Berlin Institute of Technology, Germany
4. Department of Neurology, Charitá Universitätsmedizin Berlin, Campus Virchow, Berlin, Germany

* Authors contributed equally to the work.

Abstract

Data Science is a fast growing buzzword in both industry and academia. Despite the hype, the term well reflects how a growing number of technically orientated scientists moving into traditionally less technical disciplines such as biology approach their day to day research. This body of work is an example of this approach, taking core disciplines from data science such as modelling, machine learning, statistics and data analysis and applying them to the field of neuroscience. The thesis is broken into three self-contained manuscripts, each addressing a key neuroscientfic problem in a data driven way. In the first manuscript we take inspiration from information processing in the brain and using temporal artificial neural networks and unsupervised learning, devise an approach to improve neuron based generative models. In the second manuscript we use the brain as a model and apply the approach developed in the first manuscript to the problem of dynamic representation learning in the mammalian visual cortex. In the third manuscript we use the brain as a data source and apply statistical and machine learning techniques to help understand neural representation of movement in the human basal ganglia.

Zusammenfassung

Data Science ist ein schnell wachsendes Modewort in Industrie und Forschung. Trotz des anfänglichen Hypes ist eine Vielzahl von technisch orientierten Wissenschaftlern zu vermerken, die ihren Schwerpunkt auf weniger technische Forschungsdisziplinen, wie z.B. die Biologie, verlagern.

Die vorliegende Arbeit ist ein Beispiel dieser Entwicklung, Fachdisziplinen wie Modellierung, maschinelles Lernen, Statistik und Datenanalyse im Bereich der Neurowisseschaften anzuwenden. Diese Doktorarbeit setzt sich aus drei eigenständigen Manuskripten zusammen, die neurowissenschaftliche Problematiken datengesteuert lösen.

Das erste Manuskript geht von der Informationsverarbeitung im Gehirn aus und bedient sich temporaler künstlicher neuraler Netze und unüberwachtem Lernen, um Neuron-basierte generative Modelle zu verbessern. Im zweiten Manuskript fungiert das Gehirn als Modell. Der Ansatz der ersten Arbeit wird auf die Problematik des Lernens dynamischer Darstellungen im visuellen Kortex von Säugetieren angewendet. Das dritte Manuskript sieht das Gehirn als Datenquelle und versucht mittels statistischer und maschineller Lernverfahren die neurale Darstellung der Bewegung von menschlichen Basalganglien zu veranschaulichen.

Contents

1

Introduction

1.1 Data Science

Data Science is a buzzword that is growing fast in both industry and academia. Definitions for the term abound but the technical work of a Data Scientist can generally be distilled to a few core disciplines. *Data acquisition and Processing*: the ability to acquire data relevant to the question at hand and reshape it into a useable format for further analysis, *Statistical Analysis*: the application of statistical methods to gain insight into trends and relationships within the data, *Modelling and Machine Learning*: the ability to build models that mimic key principles of the system being studied. The ability to utelise state-of-the-art learning algorithms to pry out non-obvious dependencies within the data and use them to fill in missing data points, group data into logical clusters or predict future evolutions of the data at hand (to name but a few

applications), *Data Visualisation and Story Telling*: the ability to condense the results of ones work into a visual, verbal and/or written form that clearly communicates it's key findings.

These foci coupled with traditional attributes of science such as expert domain knowledge, creativity, curiosity and experimental design, allow the Data Scientist to make compelling discoveries from large and often imperfect data. In shorter form, Data Science is an empirical science, a set of methods and systems for extracting knowledge from data [1].

Post graduate training courses in Data Science are springing up around the world as a deluge of new data sources requires ever more people with advanced analytical training. Top educators such as New York University's Centre for Data Science, Columbia University's Institute for Data Sciences and Engineering and the University of California, Berkley now offer Masters Degrees in the field whilst the University of Edinburgh's Centre for Doctoral Training in Data Science offers PhD level training.

1.2 Data Science for Neuroscience

Despite the hype, the term well reflects how a growing number of technically orientated scientists moving into traditionally less technical disciplines such as biology approach their day to day research. This transition toward data driven research is being defined by a changing research landscape where the experimental data being produced is often so complex and voluminous that it is difficult to develop the specific competencies required to succeed as both experimentalist and analyst. The resulting demand for scientists apt at data manipulation and analysis is drawing researchers from traditionally more technical fields such as computer science into these areas as the technical problems to be solved become more and more challenging.

Neuroscience is a prime example of this migration, where interdisciplinary collaborations between biologists and technical scientists are the norm. It is in fact so common that this area of crossover, most often named Computational Neuroscience, has it's own research networks [2, 3], journals [4, 5] and annual conferences [6–8] across the globe.

As experimental technology advances and the cost of computation decreases, the amount of data being produced in the natural sciences is growing. In the field of Neuroscience, the area upon which this thesis is focused, it is not uncommon for individual experiments to produce many gigabytes (if not terabytes) of data. Using this often noisy data efficiently to extract relevant information and confirm or disprove hypotheses is an ongoing challenge.

1.3 Machine Learning

The challenge of dealing with huge datasets has been partly met by leveraging the methods being developed in a field called Machine Learning. Machine Learning (ML) is a branch of Artificial Intelligence that is interested in the construction of computational systems that can learn from data (a spam filter for emails is a classic application). The increase in available computing power over the last decades has led to a resurgence in the study of a particular area of ML known as artificial neural networks (ANNs), a type of model heavily inspired by the information processing structures of the brain. ANNs are often trained using the backpropogation method [9], a supervised learning approach which requires the model to be shown both the data and the expected answer for each sample within the training set. A drawback to supervised learning is that each training example must be labeled with the the correct answer before being provided to the model. Labeling the data is often a task that must be completed manually.

Artificial Neural Networks were a hot topic in the 80's and early 90's [10–13] but were eclipsed by other machine learning methods, namely the support vector machine [14], shortly thereafter. This was in part due to the practical difficulties in training large networks with millions of parameters, the requirement of vast amounts of labeled data to perform supervised learning and the extensive computational cost of getting such networks to converge. Twenty years later, two things have changed. Computation is cheap, and an efficient unsupervised approach to initialising the many parameters in ANNs has been found in the Restricted Boltzmann Machine [15]. Unsupervised training is an approach where the model learns about the structure of the data itself,

reducing the need for labeled training sets. The Restricted Boltzmann Machine (RBM) is a specialised 2 layer ANN adept at learning useful feature representations [16] from data in an unsupervised manor. The application of RBMs along with it's kindred spirit, the Autoencoder (AE) [17], for parameter intialisation in ANNs has allowed for efficient training of multi-layer neural networks with billions of parameters [18], an area of study referred to as *Deep Learning*. These techniques have become so successful that they are now a thriving field of research in academia [16, 19–25] and industry alike, with active working groups at Google, Facebook, Baidu and Microsoft being headed up by founders of the field such as Geoff Hinton and Yann Lecun.

1.4 Thesis Overview

This thesis pursues research in areas intersecting machine learning (particularly Deep Learning) and Neuroscience, whilst drawing on the core disciplines of a data scientist. The work is broken into three self-contained manuscripts, each addressing a key neuroscientfic problem in a data driven manner, using the brain as inspiration, model and data source.

In the first manuscript *Temporal Autoencoding Improves Generative Models of Time Series* (chapter 2), we work with the brain inspired Artificial Neural Network and specifically the generative and feature learning properties of Restricted Boltzmann Machines. Much research has been done assessing and improving the generative performance of RBMs, but little of this work has focused on temporal versions of the model. Here we advance the state-of-the-art by developing a novel training method called *Temporal Autoencoding* and show that it can be used to increase the generative performance of two RBM based temporal models, the Conditional RBM (CRBM) and the Temporal RBM (TRBM).

In the second manuscript *Natural image sequences constrain dynamic receptive fields and imply a sparse code* (chapter 3), we use the brain as a model of computation and investigate applications of *Temporal Autoencoding* to the problem of dynamic representation learning in the mammalian visual cortex. Many studies address coding strategies

employed in V1 and hypothesise how these neural representations may be learnt directly from the statistics of natural images. Only a small subset of this work however addresses the problem of developing such coding strategies in a dynamic environment, where visual input is constantly changing. In this manuscript we apply the *Temporal Autoencoding* training introduced in chapter 2 to learn dynamic representations of natural image sequences. We show that the representations learned not only capture important statistics of natural images but that they also result in a temporally sparse encoding of natural image sequences, a desirable property in systems such as the brain that have limited metabolic resources.

The third manuscript *Decoding of voluntary hand movements in local field potential and extracellular spiking activity from patients with Parkinson disease and Dystonia* (chapter 4), utelises the brain as a data source and we apply statistical and machine learning techniques to help better understand neural representation of movement in the human basal ganglia. We show that strong correlations exist between neural activity in both the subthalamic nucleus and the globus pallidus interna and movement parameters, further strengthening a body of literature that links the basal ganglia to processing of motor control. Additionally, we show that it is possible to use Extracellular and Local Field Potential recordings from this region to reconstruct the patients hand position using a linear regressor.

2

Temporal Autoencoding Improves Generative Models of Time Series

Abstract

Restricted Boltzmann Machines (RBMs) are generative models which can learn useful representations from samples of a dataset in an unsupervised fashion. They have been widely employed as an unsupervised pretraining method in machine learning. RBMs have been modified to model time series in two main ways: The Temporal RBM stacks a number of RBMs laterally and introduces temporal dependencies between the hidden layer units; The Conditional RBM, on the other hand, considers past samples of the dataset as a conditional bias and learns a representation which takes these into account. Here we propose a new training method for both the TRBM and the CRBM, which enforces the dynamic structure of temporal datasets. We do so by treating the temporal models as denoising autoencoders, considering past frames of the dataset as corrupted versions of the present frame and minimising the reconstruction error of the present data by the model. We call this approach Temporal Autoencoding. This leads to a significant improvement in the performance of both models in a filling-in-frames task across a number of datasets. The error reduction for motion capture data is 56% for the CRBM and 80% for the TRBM. Taking the posterior mean prediction instead of single samples further improves the model's estimates, decreasing the error by as much as 91% for the CRBM on motion capture data. We also trained the model to perform

forecasting on a large number of datasets and have found TA pretraining to consistently improve the performance of the forecasts. Furthermore, by looking at the prediction error across time, we can see that this improvement reflects a better representation of the dynamics of the data as opposed to a bias towards reconstructing the observed data on a short time scale. We believe this novel approach of mixing contrastive divergence and autoencoder training yields better models of temporal data, bridging the way towards more robust generative models of time series.

2.1 Introduction

Good statistical models of data are generally thought to yield good representations for discriminative or predictive tasks. One class of statistical model which has received a great deal of attention in recent literature is the Restricted Boltzmann Machine [15]. The Restricted Boltzmann Machine (RBM) is a simple graphical model which is easily trainable using contrastive divergence (CD) learning [26]. A marked advantage of this class of model in addition to its strong feature learning capabilities [16, 18] is that it allows for sample generation from the learned data distribution. The Restricted Boltzmann Machine has been extended in two canonical ways to model temporal data: The Temporal RBM [27]; and the Conditional RBM [28], both of which have had notable success. Statistical modelling of temporal data is a problem of great interest in machine learning. Not only because many data sources are intrinsically temporal, but also because of the growing number of applications that interact with users in real time, requiring efficient and scalable handling of large streams of temporal data.

The TRBM learns temporal correlations between latent representations of each temporal sample, whilst the CRBM learns a latent representation for the whole data sequence (see section 2.2 below). For TRBMs and CRBMs, contrastive divergence learning seeks to approximately maximize the likelihood of sequences of observed data (in the case of the TRBM) or the conditional likelihood of present data given the past (in the case of the CRBM), without any regard to it's underlying dynamics. They learn a dynamical model of the data, but without explicitly training on the data's temporal characteristics they fail to exploit much of the information available to them. We propose a simple method to enforce the dynamics of the data in the model's learnt representations. We achieve this by training the models as a neural network for prediction, similar to the approach of denoising autoencoders [17]. We refer to this method as Temporal Autoencoding (TA), which by itself it does not yield good generative models. However, by initializing TRBM and CRBM models with Temporal Autoencoding and then applying contrastive divergence training, one can bias their structure toward the temporal dynamics of the data, resulting in better

generative performance.Here we extend on our previous work [29] and assess Temporal Autoencoding as a general pre-training algorithm for both the TRBM and CRBM models on a number of datasets. Additionally, we investigate the effect of different weight initialisation strategies along with an adaptive learning rate.

We show that Temporal Autoencoding pre-training improves the performance of both generative models across the considered datasets by as much as 80% in approximately the same time as those models trained in the conventional manner. These findings hold across different modalities of data, such as human motion capture data [28] and for the datasets included in the M3 forecasting competition [30] which encompass yearly, quarterly, monthly along with non-specific periodicity temporal data. The fact that the proposed pre-training betters model performance across datasets for both the CRBM and TRBM confirms that the method provides a robust improvement in the generative performance of both RBM models. Furthermore, the performance increase is not limited to short time-scales, but can be seen to hold even for longer periods of time, extending further than the memory encoded directly by the method.

Autoencoders have recently been cast into a new light by considering them as generative models [31]. Though we do not take that approach here, we firmly believe that autoencoder training can improve the performance of generative models greatly. This has been shown for the temporal models considered here, and we expect this to lead to a significant improvement towards training temporal generative models.

2.2 Methods

We propose a new pre-training method for both the TRBM and the CRBM, based on a denoising autoencoder approach through time. To this end we shortly discuss the RBM, the denoising autoencoder and the temporal models used. Throughout the paper we will denote the activation of visible layers by $\mathbf{v} = (v_1, v_2, \ldots, v_N)$ and the activation of hidden layers by $\mathbf{h} = (h_1, h_2, \ldots, h_M)$, where N is the number of visible units and M the number of hidden units. In the case of temporal models we will denote the present state of the visible and hidden layers by $\mathbf{v}^T = (v_1^T, v_2^T, \ldots, v_N^T)$ and

$\mathbf{h}^T = (h_1^T, h_2^T, \ldots, h_M^T)$, where T is the number of delayed units considered, and the subsequential delayed units by $\mathbf{v}^k = (v_1^k, v_2^k, \ldots, v_N^k)$ and $\mathbf{h}^k = (h_1^k, h_2^k, \ldots, h_M^k)$, where $k \in \{0, \ldots, T-1\}$. The naming convention is shown in figure 2.1 for $T = 2$ delayed units.

2.2.1 Restricted Boltzmann Machines

Restricted Boltzmann Machines are generative models which assume all-to-all symmetric connectivity between the visible and hidden variables (see figure 2.1a) and seek to model the structure of a given dataset. They are energy-based models, parametrized by an N-by-M-dimensional weight matrix $\mathbf{W}$, a bias for the visible layer $\mathbf{b}^v = (b_1^v, b_2^v, \ldots, b_N^v)$ and a bias for the hidden layer $\mathbf{b}^h = (b_1^h, b_2^h, \ldots, b_M^h)$. The energy of a given configuration of activations $\mathbf{v}$ and $\mathbf{h}$ is given by

$$E_{RBM}(\mathbf{v}, \mathbf{h}|\mathbf{W}, \mathbf{b}^v, \mathbf{b}^h) = -\sum_{i,j} W_{ij}\, v_i\, h_j - \sum_i b_i^v\, v_i - \sum_j b_j^h\, h_j,$$

and the probability of a given configuration is given by

$$P(\mathbf{v}, \mathbf{h}) = \exp\left(-E_{RBM}(\mathbf{v}, \mathbf{h}|\mathbf{W}, \mathbf{b}^v, \mathbf{b}^h)\right)/Z(\mathbf{W}, \mathbf{b}_v, \mathbf{b}^h),$$

where $Z(\mathbf{W}, \mathbf{b}_v, \mathbf{b}^h)$ is the partition function. One noted advantage of the RBM is that the visible units are independent of each other when conditioned on the hidden units and vice-versa. This allows for efficient sampling, and for the exact calculation of a number of averages. Namely, we can evaluate exactly the conditional distributions

$$P(v_i = 1|\mathbf{h}) = \sigma\left(\sum_j W_{ij} h_j + b_i^v\right),$$

and

$$P(h_j = 1|\mathbf{v}) = \sigma\left(\sum_i W_{ij} v_i + b_j^h\right),$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function.

One can extend the RBM to continuous-valued visible variables by modifying the energy function, to obtain the Gaussian-binary RBM

$$E_{RBM}(\mathbf{v}, \mathbf{h}|\mathbf{W}, \mathbf{b}^v, \mathbf{b}^h, \{\sigma_i^2\}) = -\sum_{i,j} \frac{1}{\sigma_i^2} W_{ij}\, v_i\, h_j + \sum_i \frac{(b_i^v - v_i)^2}{2\sigma_i^2} - \sum_j b_j^h\, h_j.$$

This then leads to the conditional distributions

$$P(v_i|\mathbf{h}) = \mathcal{N}\left(\sum_j W_{ij}h_j + b_i^v\,,\,\sigma_i^2\right),$$

where $\mathcal{N}(\mu, \sigma^2)$ is the normal distribution with mean μ and variance σ^2 and

$$P(h_j = 1|\mathbf{v}) = \sigma\left(\sum_i \frac{W_{ij}v_i}{\sigma_i^2} + b_j^h\right).$$

Often the variances are constrained to have the same value across dimensions, or simply taken to be constant value of 1. It is possible to learn them directly from the data, however, one must take extra care to deal extremely small variance values. Like most statistical models, RBMs can be trained by maximizing the log likelihood of the data. This, however proves to be intractable even for the case of the RBM, and we are left with maximizing surrogate functions. The derivative of the log likelihood of an observed visible state D can be written as

$$\frac{\partial \log P(D)}{\partial \theta} = -\left\langle \frac{\partial E}{\partial \theta}\middle| D\right\rangle_{\mathbf{h}} + \left\langle \frac{\partial E}{\partial \theta}\right\rangle_{\mathbf{v},\mathbf{h}},$$

where θ is any of the parameters of the model. Note that the first term is easy to compute, but the second one involves averages over the full distribution $P(\mathbf{v}, \mathbf{h})$, which is intractable. RBMs are therefore usually trained through contrastive divergence, which approximately follows the gradient of the cost function

$$\begin{aligned} CD_n(\mathbf{W}, \mathbf{b}^v, \mathbf{b}^h) = \quad & KL(P_0(\mathbf{v}|\mathbf{W}, \mathbf{b}^v, \mathbf{b}^h)||P(\mathbf{v}|\mathbf{W}, \mathbf{b}^v, \mathbf{b}^h)) \\ & -KL(P_n(\mathbf{v}|\mathbf{W}, \mathbf{b}^v, \mathbf{b}^h)||P(\mathbf{v}|\mathbf{W}, \mathbf{b}^v, \mathbf{b}^h)), \end{aligned}$$

where P_0 is the data distribution, P_n is the distribution of the visible layer after n Markov chain Monte Carlo (MCMC) steps and $KL()$ is the Kullback-Leibler divergence [26]. The samples from the data distribution are simply taken from the data, whereas the samples from P_n are taken by running a MCMC for n steps. The function CD_n gives an approximation to maximum-likelihood (ML) estimation of the weight matrix $\mathbf{W}$. Further approximation is still needed, as the CD_n cost still involves intractable averages, but it is generally found that the approximate parameter update given by

$$\Delta\theta \propto -\left\langle \frac{\partial E_{RBM}}{\partial \theta}\middle| D\right\rangle_{\mathbf{h}} + \left\langle \frac{\partial E_{RBM}}{\partial \theta}\right\rangle_{n},$$

already gives very good results. The weight updates then become

$$\Delta W_{ij} \propto \frac{1}{\sigma_i^2} \langle v_i h_j \rangle_0 - \frac{1}{\sigma_i^2} \langle v_i h_j \rangle_n .$$

In general, $n = 1$ is already sufficient for practical purposes [15].

2.2.2 Autoencoders

Autoencoders are deterministic models with two weight matrices $\mathbf{W}^1$ and $\mathbf{W}^2$ representing the flow of data from the visible-to-hidden and hidden-to-visible layers respectively (see Figure 2.1b).[1] AEs are trained to perform optimal reconstruction of the visible layer, often by minimizing the mean-squared error (MSE) in a reconstruction task. This is usually evaluated as follows: Given an activation pattern in the visible layer $\mathbf{v}$, we evaluate the activation of the hidden layer by $h_j = \sigma(\sum_i W_{ij}^1 v_i + b_j^h)$. These activations are then propagated back to the visible layer through $\hat{v}_i(v_i) = \sigma(\sum_j W_{ij}^2 h_j + b_i^v)$ and the weights $\mathbf{W}^1$ and $\mathbf{W}^2$ are trained to minimize the distance measure between the original and reconstructed visible layers. Therefore, given a set of Q image samples $\{D_k\}$ we can define the cost function. Using the squared euclidean distance between the original data and the reconstructed data for example, $\hat{\mathbf{v}}_k = (\hat{v}_1(D_k), \hat{v}_2(D_k), \ldots, \hat{v}_N(D_k))$, we have the loss function

$$\mathcal{L}(\mathbf{W}^1, \mathbf{W}^2, \mathbf{b}^v, \mathbf{b}^h | \{D_k\}) = \frac{1}{Q} \sum_d \|D_k - \hat{\mathbf{v}}(D_k)\|^2.$$

The weights can then be learned through stochastic gradient descent on $\mathcal{L}$. Autoencoders often yield better representations when trained on corrupted versions of the original data, performing gradient descent on the distance to the uncorrupted data. This approach is called a denoising autoencoder [17]. Note that in the AE, the activations of all units are continuous and not binary, and usually take values between 0 and 1.

[1] Often one only uses one matrix and propagates up throught $\mathbf{W}^1$ and down through its transpose $(\mathbf{W}^1)^\top$

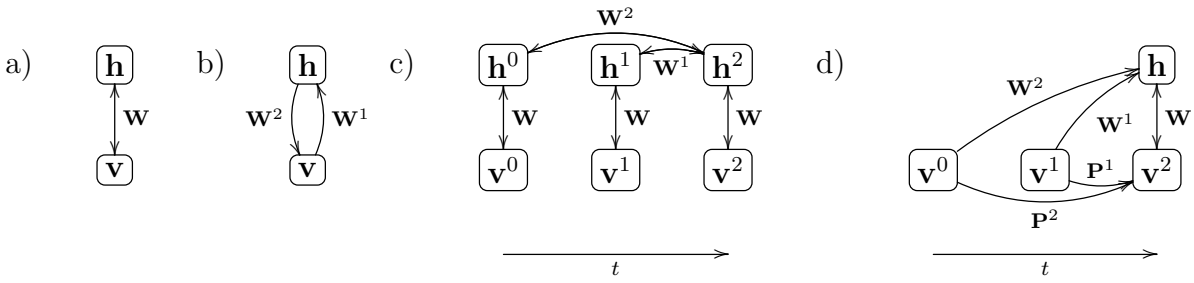

Figure 2.1: Described model architectures: a) RBM; b) Autoencoder; c) Temporal RBM and d) Conditional RBM.

2.2.3 Temporal Restricted Boltzmann Machine

Temporal Restricted Boltzmann Machines (TRBM) are a temporal extension of the standard RBM whereby connections are included from previous time steps between hidden layers, from visible-to-hidden layers and from visible-to-visible layers. Learning is conducted in the same manner as a normal RBM using contrastive divergence and it has been shown that such a model can be used to learn non-linear system evolutions such as the dynamics of a ball bouncing in a box [27]. A more constrained version of this model, discussed in [32] can be seen in figure 2.1c and only contains temporal connections between the hidden layers. We restrict ourselves to this model architecture throughout the paper.

The energy of the model for a given configuration of the visible layers $\mathcal{V} = \{\mathbf{v}^0, \ldots, \mathbf{v}^T\}$ and hidden layers $\mathcal{H} = \{\mathbf{h}^0, \ldots, \mathbf{h}^T\}$ is given by

$$E(\mathcal{H}, \mathcal{V} | \mathcal{W}, \mathcal{B}) = \sum_{t=0}^{T} E_{RBM}(\mathbf{h}^t, \mathbf{v}^t | \mathbf{W}, \mathbf{b}^v, \mathbf{b}^h) - \sum_{t=0}^{T-1} \left(\sum_{jk} W_{jk}^{(T-t)} h_j^T h_k^t \right), \qquad (2.1)$$

where we have used $\mathcal{B} = \{\mathbf{b}^v, \mathbf{b}^h\}$ and $\mathcal{W} = \{\mathbf{W}, \mathbf{W}^1, \ldots, \mathbf{W}^T\}$, where $\mathbf{W}$ are the static weights and $\mathbf{W}^1, \mathbf{W}^2, \ldots \mathbf{W}^T$ are the delayed weights for the temporally delayed hidden layers $\mathbf{h}^{T-1}, \mathbf{h}^{T-2}, \ldots, \mathbf{h}^0$ (see figure 2.1c). Because the hidden layers are coupled, the expectations in the CD cost can not be simply evaluated as in the RBM, and must be estimated by MCMC sampling, making training and sampling in this model more difficult. More specifically note that the conditional distribution $P(\mathcal{H}|\mathcal{V})$ is already intractable. A simple way to deal with this is the so-called filtering approximation

where past hidden layers are sampled from whilst ignoring the present hidden layer and then the the present hidden layer is sampled from conditioned on the past.

2.2.4 Conditional Restricted Boltzmann Machines

One way to overcome the problems of the TRBM has been proposed in the Conditional Restricted Boltzmann Machines [28]. The CRBM has only one hidden layer which receives input from all visible layers, past and present and the present visible layer additionally receives input from past visible layers. Unlike the TRBM, only the present hidden and visible layers are considered to be free, whereas the past visible states are conditioned on. The energy of the model can be written as

$$E_{CRBM}(\mathbf{h}^T, \mathbf{v}^T | \mathbf{v}^0, \ldots, \mathbf{v}^{T-1}, \mathcal{W}, \mathcal{B}, \mathcal{P}) = E_{RBM}(\mathbf{h}^T, \mathbf{v}^T | \mathbf{W}, \mathbf{b}_h, \mathbf{b}^v) \\ - \sum_{t=0}^{T-1} \left(\sum_{ij} W_{ij}^{(T-t)} \, v_i^t \, h_j^T + \sum_{il} P_{il}^{(T-t)} \, v_i^t \, v_l^T \right),$$

where $\mathcal{P} = \{P_0, \ldots P_{T-1}\}$ are the visible-to-visible weights. The model architecture can be seen in Figure figure 2.1d. Using this formulation, the hidden layer can still be easily marginalized over, allowing for more efficient training using contrastive divergence. The CRBM is possibly the most successful of the temporal RBM models to date and has been shown to both model and generate data from complex dynamical systems such as human motion capture data and video textures [33].

2.2.5 Temporal Autoencoding Training

Standard CD training for the TRBM and CRBM seeks to maximize the likelihood of the data observed. This usually works quite well and has been shown to allow the trained models to reproduce complex temporal data such as video of a bouncing ball or human motion capture. This training method however ignores the current time frames causal dependence on the past. In time series data it is a natural assumption that the future states are given by some function of the past states, latent variables and possibly noise. We seek to explore this property, by explicitly learning a representation which captures these dynamics.

We do so by treating the hidden layers of the model as an information bottleneck, similar to what is done in the training of the denoising autoencoder [17]. We treat the past states of the time series up to a number of delays as a noisy representation of the present state, and propagate their values through the model, considering it as a neural network with sigmoidal activation functions and perform gradient descent on the quadratic error of the reconstructed present state. In this way, we explicitly constrain the model to represent the dynamic structure of the data.

This amounts to supervised learning for reconstruction using the architectures shown in figure 2.2. Though the idea behind the training procedure is the same for both models, the specifics are slightly different and as such we consider them separately below.

Temporal Autoencoding for the TRBM

Let us first consider the TRBM. The energy of the model is given by equation (2.1) and is essentially an T-th order autoregressive RBM which is usually trained by standard contrastive divergence. Here we propose training it with a novel approach, highlighting the temporal structure of the stimulus. First, the individual RBM visible-to-hidden weights $\mathbf{W}$ are initialized through contrastive divergence learning with a sparsity constraint on static samples of the dataset. After that, to ensure that the weights representing the hidden-to-hidden connections ($\mathbf{W}^t$) encode the dynamic structure of the ensemble, we initialize them by pre-training in the fashion of a denoising Autoencoder. For this, we consider the model to be a deterministic Multi-Layer Perceptron with continuous activation in the hidden layers. We then consider the T delayed visible layers as features and try to predict the current visible layer by projecting through the hidden layers. In essence, we are considering the model to be a feed-forward network, where the delayed visible layers would form the input layer, the delayed hidden layers would constitute the first hidden layer, the current hidden layer would be the second hidden layer and the current visible layer would be the output as is pictured in figure 2.2. Given sample activations of the visible layers $\mathcal{V}_d = \{\mathbf{v}_d^0, \mathbf{v}_d^1, \ldots, \mathbf{v}_d^{T-1}, \mathbf{v}_d^T\}$, we can then write the prediction of the network as $\hat{\mathbf{v}}^T(\mathbf{v}_d^0, \mathbf{v}_d^1, \ldots, \mathbf{v}_d^{T-1}; \mathcal{W}, \mathcal{B})$, where the

d index runs over the Q data points. The exact format of this function is described in algorithm 1. We minimize the reconstruction error given by

$$\mathcal{L}(\mathcal{W}, \mathcal{B}) = \frac{1}{Q} \sum_{d} \left\| \mathbf{v}_d^T - \hat{\mathbf{v}}^T(\mathbf{v}_d^0, \mathbf{v}_d^1, \ldots, \mathbf{v}_d^{T-1}; \mathcal{W}, \mathcal{B}) \right\|^2,$$

where the sum over d goes over the entire dataset. After the Temporal Autoencoding is completed, the whole model (both visible-to-hidden and hidden-to-hidden weights) is trained together using contrastive divergence (CD) training. A summary of the training method is described in table 2.1.

Table 2.1: Autoencoded TRBM Training Steps

Step	Action
1. Static RBM Training	Constrain the static weights $\mathbf{W}$ using CD on single frame samples of the training data
2. Temporal Autoencoding	Constrain the temporal weights $\mathbf{W}^1$ to $\mathbf{W}^T$ using a denoising autoencoder on multi-frame samples of the data
3. Model Finalisation	Train all model weights together using CD on multi-frame samples of the data

Temporal Autoencoding for the CRBM

The procedure is very similar for the CRBM. First the static weights $\mathbf{W}$ are initialized with contrastive divergence training. After that, we reconstruct the present frame from its past observations by passing it through the hidden layer. The obtained reconstruction is then a function of the past observations and the matrices $\mathcal{W}$ and the biases $\mathcal{B}$, we can write $\hat{\mathbf{v}}^T(\mathbf{v}_d^0, \mathbf{v}_d^1, \ldots, \mathbf{v}_d^{T-1}; \mathcal{W}, \mathcal{B})$. We then perform stochastic gradient descent on the reconstruction error

$$\mathcal{L}(\mathcal{W}, \mathcal{B}) = \frac{1}{Q} \sum_{d} \left\| \mathbf{v}_d^T - \hat{\mathbf{v}}^T(\mathbf{v}_d^0, \mathbf{v}_d^1, \ldots, \mathbf{v}_d^{T-1}; \mathcal{W}, \mathcal{B}) \right\|^2.$$

After this step is finished we proceed to train the CRBM with normal contrastive divergence to fine tune the weights for better generation. A summary for the training procedure is given in table 2.1 and a complete description of the temporal autoencoding step is given in algorithm 2.

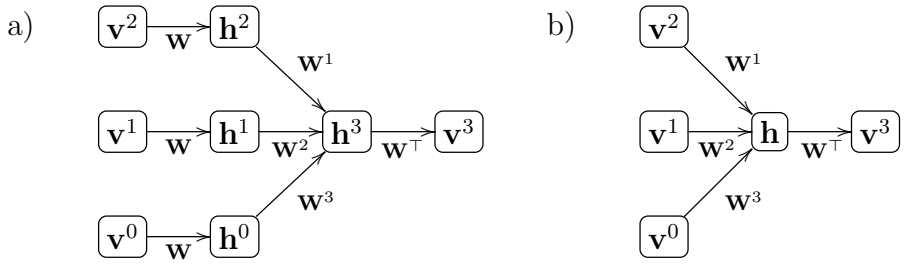

Figure 2.2: TRBM (a) and CRBM (b) temporal autoencoding architectures.

Implementation

Gradient descent on the cost functions explained above involves backpropagation through the hidden layers. This has been made relatively simple by automatic differentiation packages such as Theano [34]. We have implemented the temporal autoencoding training as a MLP and then proceeded to perform stochastic gradient descent on the loss using mini-batches.

Adaptive Learning Rate

Restricted Boltzmann Machines are notoriously sensitive to model parameters [35] including the choice of the learning rate η. A bad choice of η can cause the model to diverge during learning, though one can attempt to circumvent this issue by adapting η as training advances [36, 37]. Here we take a simplistic but successful approach that incrementally increases the learning rate by 10% when the error is less than the mean error of the last 5 iterations and shrinks the learning rate by 20% when the error is greater.

2.3 Results

The quality of a generative model can be measured by sampling from it and comparing the samples to the true dataset. This approach is generally called *filling-in* and is particularly well-suited to temporal applications as we can condition on the observations

Algorithm 1 Pre-Training Temporal weights through Autoencoding for the TRBM

Given a learning rate η

for each sequence of data frames $I(t-T), I(t-(T-1)) \ldots, I(t)$ **do**

take $\mathbf{v}^T = I(t), \ldots, \mathbf{v}^0 = I(t-T)$ and

for $j = 1$ **to** T **do**

for $i = 1$ **to** j **do**

$h_l^{T-i} = \sigma(\sum_k W_{kl}\, v_k^{T-i} + b_l^h)$

end for

$h_l^T = \sigma(\sum_{j=1}^T \sum_m W_{lm}^j h_m^{T-j} + b_l^h)$

$\hat{v}_n^T = \sigma(\sum_l W_{nl}\, h_l^T + b_n^v)$

$\epsilon(\mathbf{v}^T, \hat{\mathbf{v}}^T) = |\mathbf{v}^T - \hat{\mathbf{v}}^T|^2$

$\Delta \mathbf{W}^d = \eta\, \partial\epsilon / \partial \mathbf{W}^d$

end for

end for

up to a certain time and fill in the missing frames by sampling from the model. The performance can then be quantified using the Mean squared error (MSE) or the Mean Absolute Percentage Error (MAPE) between the true and sampled data.

We have applied our pre-training method to the CRBM and TRBM using two datasets. The motion-capture data described in [28] and the M3 competition dataset [30]. For both datasets we separated the data into a training and a test set, then trained our models on the training set and evaluated them in a filling-in-frames task on the test set. For all experiments we used a Gaussian-binary RBM model with variance fixed to 1.

2.3.1 Motion-Capture Data

We assessed the impact of our pre-training method by applying it to the 49 dimensional human motion capture data described in [28] and using this as a benchmark, comparing

Algorithm 2 Pre-Training Temporal weights through Autoencoding for the CRBM

Given a learning rate η

for each sequence of data frames $I(t-T), I(t-(T-1)) \ldots, I(t)$ **do**

take $\mathbf{v}^T = I(t), \ldots, \mathbf{v}^0 = I(t-T)$ and

for $j = 1$ **to** T **do**

$h_l = \sigma(\sum_{t=T-j}^{T-1} \sum_k W_{kl}^{(T-t)} v_k^t + b_l^h)$

$\hat{v}_i^T = \sigma(\sum_l W_{il} h_l + b_i^v)$

$\epsilon(\mathbf{v}^T, \hat{\mathbf{v}}^T) = |\mathbf{v}^T - \hat{\mathbf{v}}^T|^2$

$\Delta \mathbf{W}^d = \eta \, \partial \epsilon / \partial \mathbf{W}^d$

end for

end for

the performance to the models without pre-training [2]. We also investigate the impact that different initialisation strategies have on model performance along with the benefits of using an adaptive learning rate. All the models were implemented using Theano [34], have a temporal dependence of 6 frames and were trained using minibatches of 100 samples for 500 epochs[3]. The training time for the models was approximately equal and the weight matrices were initialised randomly from the distribution $\mathcal{N}(0, 0.1)$ unless otherwise stated. Training was performed on the first 2000 samples of the dataset after which the models were presented with 1000 snippets of the data not included in training set and required to generate the next frame in the sequence. Generation from the TRBM is done using the filtering approximation, that is, by taking a sample from the hidden layers at $t-6$ through $t-1$ and then Gibbs sampling from the RBM at time t while keeping the others fixed as biases. Generation from the CRBM is more straightforward, activations from the visible layers at $t-6$ through $t-1$ are fed to

[2] In this section we refer to the reduced TRBM model referenced in [32] with only hidden-to-hidden temporal connections

[3] For the TRBM and CRBM, training epochs were broken up into 100 static pre-training and 400 epochs for all the temporal weights together. For the TA pretrained models, aTRBM and aCRBM, training epochs were broken up into 100 static pre-training, 50 Autoencoding epochs per delay and 100 epochs for all the temporal weights together, totalling to the same number of training epochs (500)

the hidden layer then Gibbs sampling is performed for the visible units at time t. For both models, the visible layer at time t is initialized with noise and we sample for 100 Gibbs steps from the model. The results of a single trial prediction for 4 random dimensions of the dataset can be seen in Figure 2.3 and the mean squared error and standard deviations of the model predictions over 100 repetitions of the task can be seen in Table 2.2.

The models trained with Temporal Autoencoding significantly outperform their CD-only trained counterparts. The CRBM shows an improvement of approximately 56%, while the TRBM shows an improvement of almost 80% on this dataset. Surprisingly, initialising the network with weights of 0 instead of the random distribution described above had no impact on performance for either the standard CD trained models or the TA trained ones. The results can be further improved by taking the mean of the estimate by sampling from the hidden layer multiple times and taking the average prediction. This is akin to taking the Bayesian posterior mean estimator and leads to a further decrease in the MSE of 78% for the CRBM and 91% for the TRBM relative to straight CD training.

One could argue that the improved performance of the TA pre-trained model simply shows that a deterministic neural network is more well suited to the task at hand. To make sure that the gain is due to the interplay of both training approaches, we also trained a deterministic multi-layer perceptron (MLP) with the architecture shown in figure 2.2. This results are shown in the rightmost column in figure 2.3 and one can see that the simple deterministic approach outperforms the CD-trained model, but not the model trained with Temporal Autoencoding.

These improvements also hold for longer time scales if we keep feeding the models predictions back into it and let it generate autonomously. The TA pre-training significantly lowers the prediction error. Even after 6 frames, when all the visible layer frames were generated by the model, the MSE is still approximately as low or lower than when filling in one frame from the data without pre-training. The prediction errors for our models are shown in figure 2.4.

The use of an adaptive learning rate plays an important role in these results enabeling the models to perform much better than with any the many fixed learning rates we experimented with. A good example is in the results of CD-trained CRBM where the introduction of an adaptive learning rate lowered the error by approximately 50% over our previously published results in [29].

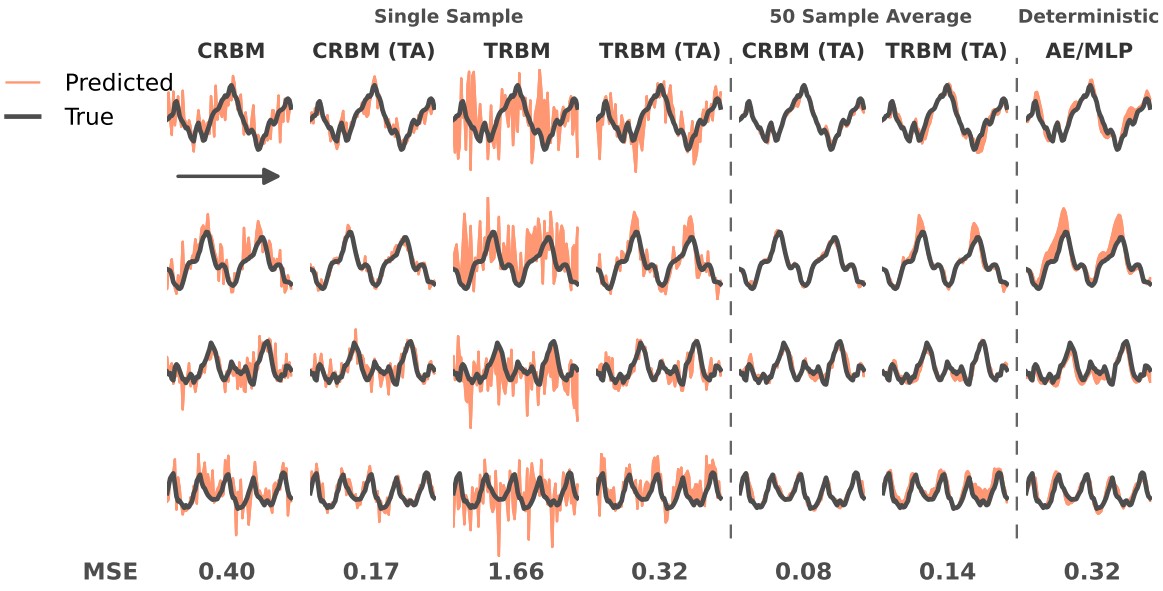

Figure 2.3: The CRBM and TRBM are used to fill in data points from motion capture data [28] with and without TA pre-training. 4 dimensions of the motion data are shown along with the their model reconstructions from a single trial (left group), mean prediction over 50 samples (middle group) and deterministically (right group).

2.3.2 M3 Forecasting Competition Data

The motion capture experiments have shown great results for our proposed training method, but it reflects a lot of structure specific to the origin of the data. To assess how the method works on a more generalised dataset, we applied it to the datasets of the M3 forecasting competition. The M3 forecasting competition [30] pitted forecasting algorithms against one another on 3003 different datasets, ranging from microeconomical

Table 2.2: Prediction results on the motion capture dataset

Model	**Architecture and Training**	**MSE (± SD)**
TRBM	100 hidden units, 6 frame delay	1.59 (± 0.12)
TRBM (TA)	100 hidden units, 6 frame delay	0.32 (± 0.03)
TRBM (TA), 50 sample mean	100 hidden units, 6 frame delay	0.14 (± 0.03)
CRBM	100 hidden units, 6 frame delay	0.40 (± 0.05)
CRBM (TA)	100 hidden units, 6 frame delay	0.17 (± 0.02)
CRBM (TA), 50 sample mean	100 hidden units, 6 frame delay	0.08 (± 0.02)

to financial and industrial data. The data are univariate, but through state augmentation we can use our method to generate predictions for future data points. We have done so by taking chunks of 4 consecutive observations and used successive chunks as our multivariate data. With these we have trained the model to generate forecasts.

Figure 2.5 shows the average performance of our algorithm on the four different kinds of data. They are separated into yearly, quarterly, monthly and other, the main categories of the competition. Here we measure the model performance using MAPE as was used in the competition. Although the datasets are generally small if compared to the usual unsupervised learning case, our training method still fares relatively well. Furthermore, TA pre-training continues to show a strong improvement over straight CD learning across the board. The robust performance of the TA pre-training on these datasets strongly suggests our method will generally yield improvements.

2.4 Discussion and Future Work

We have introduced a new training method for temporal RBMs that we call Temporal Autoencoding and have shown that it can achieve a significant performance increase in a filling-in-frames task across a number of datasets. The gain in performance from

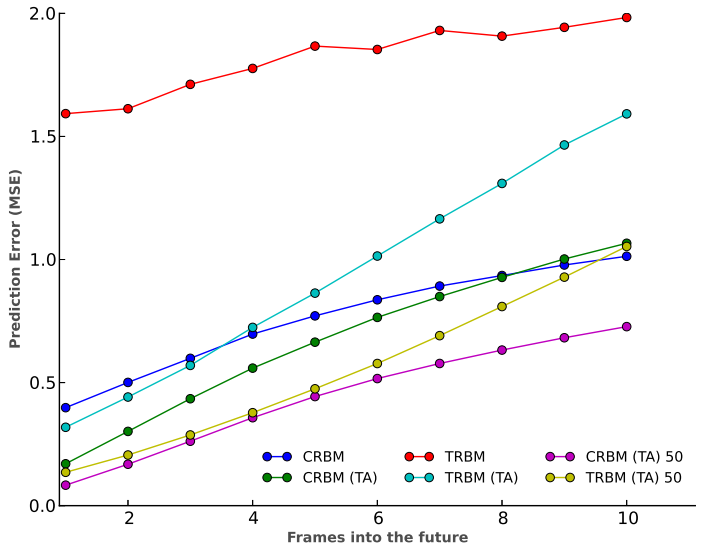

Figure 2.4: CRBM, TRBM are used to fill in data points from motion capture data [28] with and without TA pre-training. The plot shows the evolution of the MSE after the input is killed and the model is left to generate samples on its own.

our pre-training approach is robust and holds for both the CRBM and the TRBM, allowing for more effective training of temporal generative models.

Our approach combines the supervised approach of backpropagating prediction errors through the network with the unsupervised approach of Contrastive Divergence learning. We have also shown that neither method by itself can achieve the performance we achieve by combining both.

In the M3 contest dataset, specifically, the approach is shown to consistently improve the MAPE in a forecasting task, across a number of different types of data. On motion capture data, on the other hand, we were able to improve the MSE of the generative model by as much as 90% in some cases.

It is our opinion that the approach of autoencoding temporal dependencies gives the model a more meaningful temporal representation than is achievable through contrastive divergence training alone. The TA training seeks to constrain the model to reproduce the dynamics observed in the data and as such it is not surprising that the improvement in generation also leads to an improvement in the prediction performance

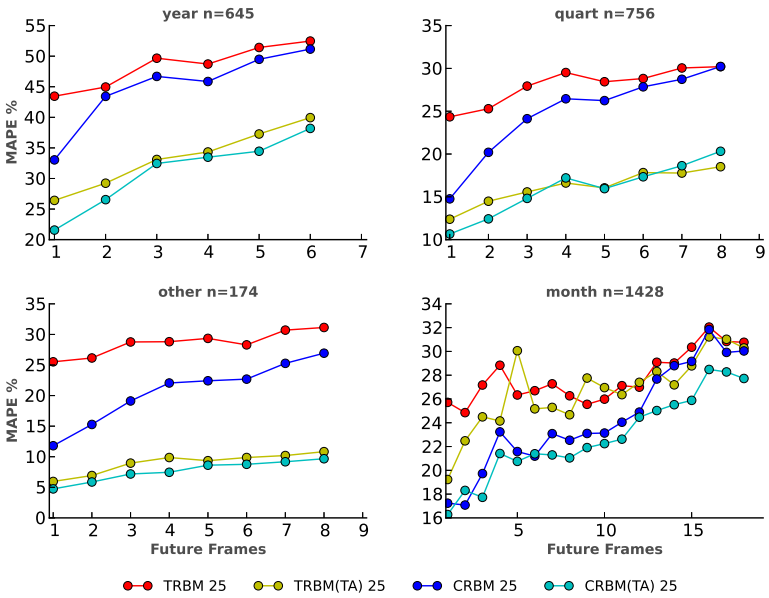

Figure 2.5: CRBM, TRBM are used to fill in data points from the M3 forecasting competition with and without TA pre-training. The plot shows the evolution of the MAPE after the input is killed and the model is left to generate samples on its own. In all four data categories, the Temporal Autoencoded models out perform those without TA training

of the models considered. We believe the inclusion of Autoencoder training in temporal learning tasks will be beneficial in a number of contexts, as it enforces the causal structure of the data on the learned model. In future work, it would be interesting to try to replace Gibbs sampling of the posterior mean by an approximate inference approach in order to speed up the prediction.

3

Natural image sequences constrain dynamic receptive fields and imply a sparse code

Abstract

In their natural environment, animals experience a complex and dynamic visual scenery. Under such natural stimulus conditions, neurons in the visual cortex employ a spatially and temporally sparse code. For the input scenario of natural still images, previous work demonstrated that unsupervised feature learning combined with the constraint of sparse coding can predict physiologically measured receptive fields of simple cells in the primary visual cortex. This convincingly indicated that the mammalian visual system is adapted to the natural spatial input statistics. Here, we extend this approach to the time domain in order to predict dynamic receptive fields that can account for both spatial and temporal sparse activation in biological neurons. We rely on temporal restricted Boltzmann machines and suggest a novel temporal autoencoding training procedure. When tested on a dynamic multivariate benchmark dataset this method outperformed existing models of this class. Learning features on a large dataset of natural movies allowed us to model spatio-temporal receptive fields for single neurons. They resemble temporally smooth transformations of previously obtained static receptive fields and are thus consistent with existing theories. A neuronal spike response model demonstrates how the dynamic receptive field facilitates temporal and population sparseness. We discuss the potential mechanisms and benefits of a spatially and temporally sparse

representation of natural visual input.

3.1 Introduction

Physiological and theoretical studies have argued that the sensory nervous systems of animals are evolutionarily adapted to their natural stimulus environment [for review see 38]. The question of how rich and dynamic natural stimulus conditions determine single neuron response properties and the functional network connectivity in mammalian sensory pathways has thus become an important focus of interest for theories of sensory coding [for review see 39, 40].

For a variety of animal species and for different modalities it has been demonstrated that single neurons respond in a temporally sparse manner [38, 40–42] when stimulated with natural time-varying input. In the mammal this is intensely studied in the visual [43–49] and the auditory [42, 50, 51] pathway as well as in the rodent whisker system [41, 52]. Sparseness increases across sensory processing levels and is particularly high in the neocortex. Individual neurons emit only a few spikes positioned at specific instances during the presentation of a time-varying input. Repeated identical stimulations yield a high reliability and temporal precision of responses [48, 53]. Thus, single neurons focus only on a highly specific spatio-temporal feature from a complex input scenario.

Theoretical studies addressing the efficient coding of natural images in the mammalian visual system have been very successful. In a ground breaking study, [54] learned a dictionary of features for reconstructing a large set of natural still images under the constraint of a sparse code to obtain receptive fields (RF), which closely resembled the physiologically measured RFs of simple cells in the mammalian visual cortex. This approach was later extended to the temporal domain by [55], learning rich spatio-temporal receptive fields directly from movie patches. In recent years, it has been shown that a number of unsupervised learning algorithms, including the denoising Autoencoder (dAE) [17] and the Restricted Boltzmann Machine (RBM) [15, 16, 56], are able to learn structure from natural stimuli and that the types of structure learnt can again be related to cortical RFs as measured in the mammalian brain [57–59].

Considering that sensory experience is per se dynamic and under the constraint of a temporally sparse stimulus representation at the level of single neurons, how could the

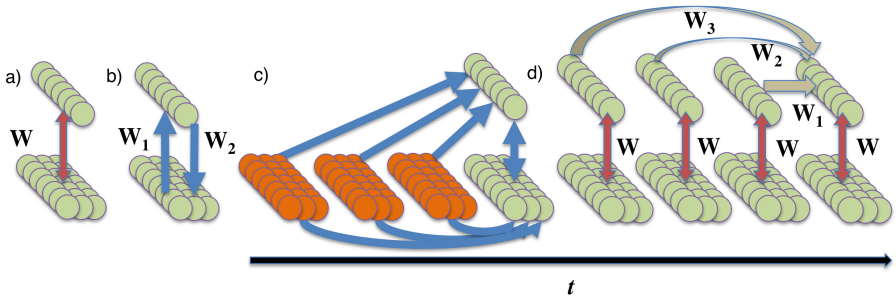

FIGURE 3.1: Described model architectures: a) Autoencoder; b) RBM; c) Conditional RBM and d) Temporal RBM. In the CRBM (figure 3.1C; see also section 3.4), there is a *hidden* layer only at the current sample time whose activation is defined by weights connecting the current as well as previous activations of the *visible* layer. The TRBM (figure 3.1D) has a *hidden* layer instantiation for each sample time within the models delay dependency and the temporal evolution of the model is defined by lateral connections between the *hidden* units of consecutive time steps.

static RF model, i.e. the learned spatial feature, extend into the time domain? Here we address this question with an unsupervised learning approach using RBMs as a model class. Building on an existing model, the Temporal Restricted Boltzmann Machine (TRBM) introduced by [27], we introduce a novel learning algorithm with a temporal autoencoding approach to train RBMs on natural multi-dimensional input sequences. For validation of the method, we test the performance of our training approach on a reference dataset of kinematic variables of human walking motion and compare it against the existing TRBM model and the Conditional RBM (CRBM) as a benchmark [28]. As an application of our model, we train the TRBM using temporal autoencoding on natural movie sequences and find that the neural elements develop dynamic RFs that express smooth transitions, i.e. translations and rotations, of the static receptive field model. Our model neurons account for spatially and temporally sparse activities during stimulation with natural image sequences and we demonstrate this by simulation of neuronal spike train responses driven by the dynamic model responses. Our results propose how neural dynamic RFs may emerge naturally from smooth image sequences.

3.2 Results

We outline a novel method to learn temporal and spatial structure from dynamic stimuli - in our case smooth image sequences - with artificial neural networks. The hidden units (neurons) of these generative models develop dynamic RFs that represent smooth temporal evolutions of static RF models that have been described previously for natural still images. When stimulated with natural movie sequences the model units are activated sparsely, both in space and time. A point process model translates the model's unit activation into sparse neuronal spiking activity with few neurons being active at any given point in time and sparse single neuron firing patterns.

3.2.1 The Model

We rely on the general model class of RBMs (see section 3.4.1). The classic RBM is a two layer artificial neural network with a *visible* and a *hidden* layer used to learn representations of a dataset in an unsupervised fashion (figure 3.1 A). The units (neurons) in the *visible* and those in the *hidden* layers are all-to-all connected via symmetric weights and there is no connectivity between neurons within the same layer. The input data, in our case natural images, activate the units of the *visible* layer. This activity is then propagated to the *hidden* layer where each neuron's activity is determined by the input data and by the weights $\mathbf{W}$ connecting the two layers. The weights define each hidden neuron's filter properties or its RF, determining its preferred input.

Whilst the RBM has been successfully used to model static data, it lacks in the ability to explicitly represent the temporal evolution of a continuous dataset. The CRBM (figure 3.1 C) and TRBM (figure 3.1 D) are both temporal extensions of the RBM model, allowing the hidden unit activations to be dependent on multiple samples of a sequential dataset. The models have a *delay* parameter which is used to determine how long the integration period on a continuous dataset is.

The CRBM has an instantiation of the *visible* layer for each sample time within the model's delay range, each of which is connected directly to the single *hidden* layer at the current sample point. In the TRBM (figure 3.1 D; see also section 3.4.1) the

temporal dependence is modelled by a set of weights connecting the *hidden* layer activations at previous steps in the sequence to the current hidden layer representation. The TRBM and CRBM have proven to be useful in the modelling of temporal data, but each again has its drawbacks. The CRBM does not separate the representations of *form* and *motion*. Here we refer to *form* as the RF of a hidden unit in one sample of the dataset and *motion* as the evolution of this feature over multiple sequential samples. This drawback makes it difficult to interpret the features learnt by the CRBM over time as the two modalities are mixed. The TRBM explicitly separates representations of *form* and *motion* by having dedicated weights for the *visible* to *hidden* layer connections (*form*) and for the temporal evolution of these features (*motion*). Despite these benefits, the TRBM has proven quite difficult to train due to the intractability of its probability distribution (see section 3.4).

In this work we develop a new approach to training Temporal Restricted Boltzmann Machines that we call Temporal Autoencoding (we refer to the resulting TRBM as an autoencoded TRBM or aTRBM) and investigate how it can be applied to modelling natural image sequences. The aTRBM adds an additional step to the standard TRBM training, leveraging a denoising Autoencoder to help constrain the temporal weights in the model. Table 3.1 provides an outline of the training procedure whilst more details can be found in the section 3.4.1.

In the following sections we compare the filters learnt by the aTRBM and CRBM models on natural image sequences and show that the aTRBM is able to learn spatially and temporally sparse filters having response properties in line with those found in neurophysiological experiments

3.2.2 Learning Temporal Filters from Natural Image Sequences

We have trained a CRBM and an aTRBM on natural image sequence data taken from the Hollywood2 dataset introduced in [60], consisting of a large number of snippets from various Hollywood films. From the dataset, 20x20 pixel patches are extracted in sequences 30 frames long. Each patch is contrast normalized (by subtracting the mean

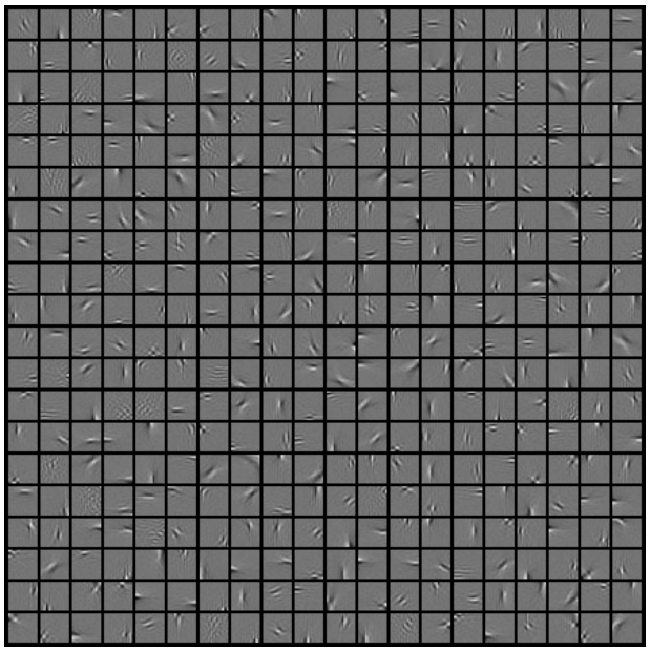

Figure 3.2: Static filters learned by the aTRBM on 20x20 image patches. Note the mostly gabor like filters of varying orientation and frequency selectivity.

and dividing by the standard deviation) and ZCA whitened [61] to provide a training set of approximately 350,000 samples. The aTRBM and CRBM models, each with 400 hidden units and a temporal dependency of 3 frames, are trained initially for 100 epochs on static frames of the data to initialize the static weights **W** and then until convergence on the full temporal sequences. Full details of the models' architecture and training approaches are given in the Methods section.

Static RFs

The static filters learned by the aTRBM through the initial contrastive divergence training can be seen in Figure 3.2 (note that the static filters are pre-trained in the same way for the CRBM and aTRBM, therefore the filters are equivalent). We obtain Gabor-like patch filters resembling simple cell RFs in V1, reproducing the typical result for a variety of methods (see Introduction), statistics of which can be seen in figure 3.3. The RFs of the hidden units are spatially located across the entire image patch with some distinct clustering along the borders (Figure 3.3A). In 2D Fourier

space (Figure 3.3B) one can see a good coverage of the space, representing frequency and direction selectivity, both of these results being in agreeance with those found in similar studies (see [62] and [61], for example). The filters also display a preference for cardinal (horizontal and vertical) orientations (Figure 3.3C), a phenomenon that has often been reported in electrophysiological experiments of primary visual cortex [eg. 63, 64].

Dynamic RFs

We then analysed how the static filters are connected through the temporal weights learned during autoencoder training by visualizing their evolution over time. The filters discussed were learned by the aTRBM (see equation (3.1)) with our training algorithm described in section 3.4.1. To visualize the dynamic RF of a hidden unit we clamped the activation of that unit to 1 and set all other units to be inactive in the most delayed layer of the aTRBM. We then proceeded to sample from the distribution of all other hidden layers and chose the most active units in every delay. This is shown in figure 3.4. We have shown the most active units when a hidden unit is active for the 80 units with highest temporal variation among the subsequent filters. This however, only gives us a superficial look into the dynamics of the RF's. One way to look further is to consider the n most active units at the second-furthest delay and then sequentially clamp each of these to an active state and look at the resulting activations in the remaining layers. If one does this sequentially, we are left with a tree of active units, 1 at time $t-T$, n at time $t-(T-1)$, and n^T at time t. We can then look at what these units code for. We have performed this procedure with two hidden units, and to visualize what they code for we have plotted the center of mass of the filters in frequency and position space. This is shown in figure 3.5.

Visualizing the temporal RFs learnt by the CRBM is simpler than for the aTRBM. We display the weight matrix $\mathbf{W}$ and the temporal weights $\mathbf{W}_1$ to $\mathbf{W}_d$ for each hidden unit directly as a projection into the visible layer (a 20x20 patch). This shows the temporal dependence of each hidden unit on the past visible layer activations and is plotted with time running from top to bottom in figure 3.4B. The aTRBM learns richer

filter dynamics with a longer temporal dependency, whereas the CRBM only seems to care about the visible layers at times t and $t-1$, possibly because most of the variation is captured by the visible-to-visible weights.

The temporal profile of excitation versus inhibition for the aTRBM can also be seen from the profile of the connectivity matrix between it's hidden units. This is shown in figure 3.3(E) and one can note a transition from self-excitation at $delay = 1$ to self-inhibition at $delay = 3$.

In figure 3.5 we analyse the filter histories of the aTRBM for $n = 3$ and visualize for two of the *hidden* layer units, their preference in image space, frequency and direction.

For the unit in figure 3.5A there is a clear selectivity for spatial location over its temporal evolution and activations remain spatially localised. In contrast there is no apparent preference for orientation. The unit depicted in figure 3.5B on the other hand, displays strong orientation selectivity, but the spatial selectivity is not accentuated. These results are representative of the population and provide evidence for preferential connectivity between cells with similar RFs, a finding that is supported by a number of experimental results in V1 [65, 66].

3.2.3 The dynamic RF model facilitates sparse coding

The temporal evolution of the spatial filter structure expressed by single units in the dynamic RF model (figure 3.4 and figure 3.5) renders individual units to be selective to a specific spatio-temporal structure of the input within their classical RF. This increased stimulus specificity in comparison to a static RF model implies an increased sparseness of the units' activation. To test this hypothesis we quantified temporal and spatial sparseness for both model approaches.

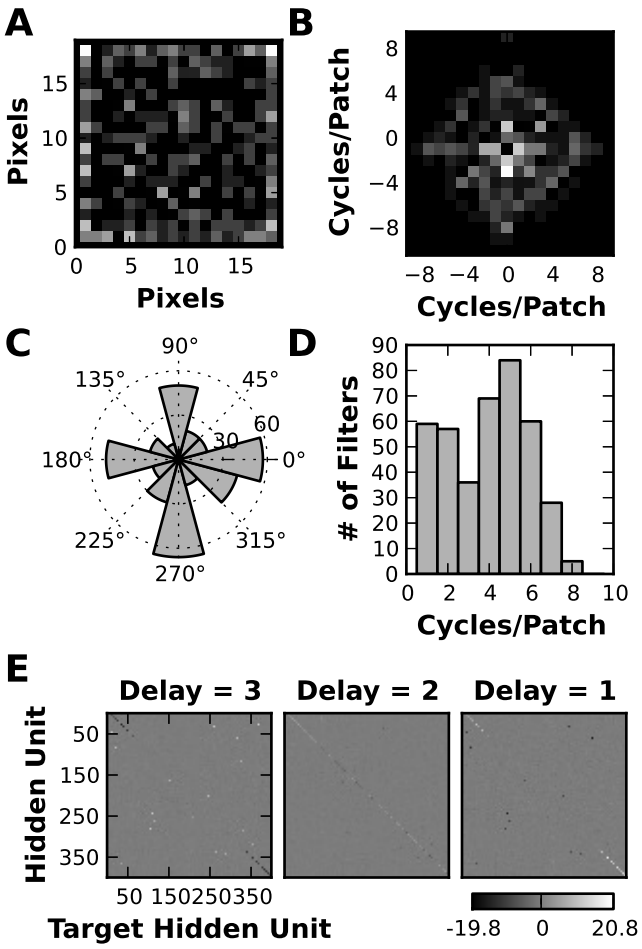

FIGURE 3.3: Static filter statistics - aTRBM: a) Histogram of the filters spatial location; b) Histogram of the filters spatial frequency; c) Histogram of the filters preferred direction (showing a clear preference for cardinal directions) and d) frequency. e) Visualization of the temporal transition weights for 3 time delays for the aTRBM. Note the strong self excitation at $delay = 1$ and self inhibition at $delay = 3$

Temporal sparseness

We measured temporal sparseness of the single unit activation h using the well established sparseness index S (equation (3.2)) introduced by [67] and described in section 3.4.2. The higher the value of S for one particular unit, the more peaked is the temporal activation profile $h(t)$ of this unit. The lower the value of S, the more evenly distributed are the activation values $h(t)$. The quantitative results across the population of 400 hidden units in our aTRBM model are summarized in figure 3.6 A. As expected, units are temporally sparser when the dynamic RF is applied with a mean

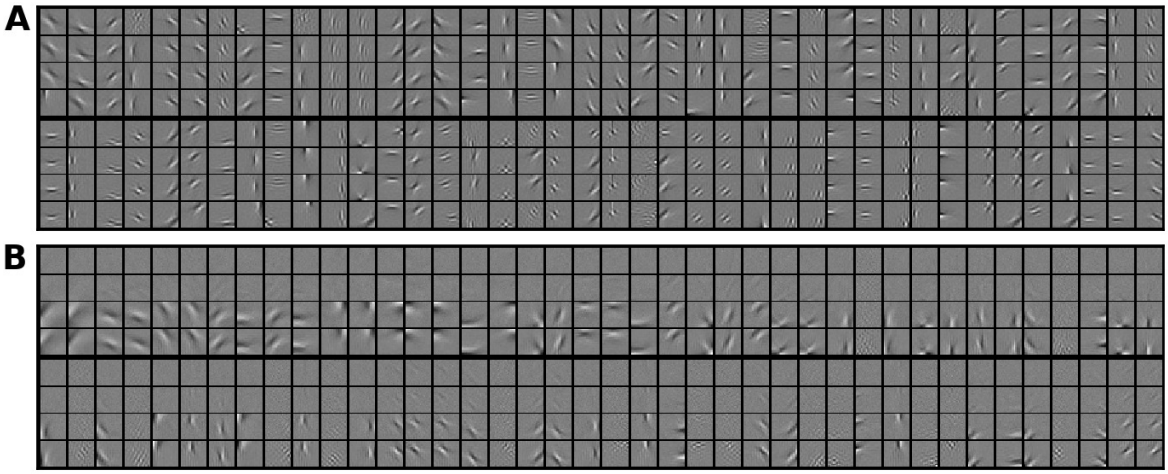

Figure 3.4: Dynamic RFs. 80 out of 400 hidden units with the highest temporal variation from an aTRBM (A) and a CRBM (B). For the pre-trained TRBM, we plot the most active units as described in the text. Each group of 4 images represents the temporal filter of one hidden unit with the lowest patch representing time t and the 3 patches above representing each of the delay steps in the model. The units are displayed in two rows of 40 columns with 4 filters, with the temporal axis going from top to bottom.

sparseness index of 0.92 (median: 0.93) compared to the mean of 0.69 (median: 0.82) for the static RF. This is also reflected in the activation curves for one example unit shown in figure 3.6 D1 for the static RF (blue) and the dynamic RF (green) recorded during the first 8 s of video input.

In the nervous system temporally sparse stimulus encoding finds expression in stimulus selective and temporally structured single neuron firing patterns where few spikes are emitted at specific instances in time during the presentation of a time varying stimulus (see section 3.1). In repeated stimulus presentations the temporal pattern of action potentials is typically repeated with high reliability (e.g. [53]). In order to translate the continuous activation variable of the hidden units in our aTRBM model into spiking activity we used the cascade model depicted in figure 3.6 C and described in section 3.4.2. The time-varying activation curve (figure 3.6 D1) is used as deterministic intensity function of a stochastic point process model. This allows us to generate repeated stochastic point process realizations, i.e. single trial spike trains, as shown for the example unit in figure 3.6 D2. Clearly, the repeated simulation trials based

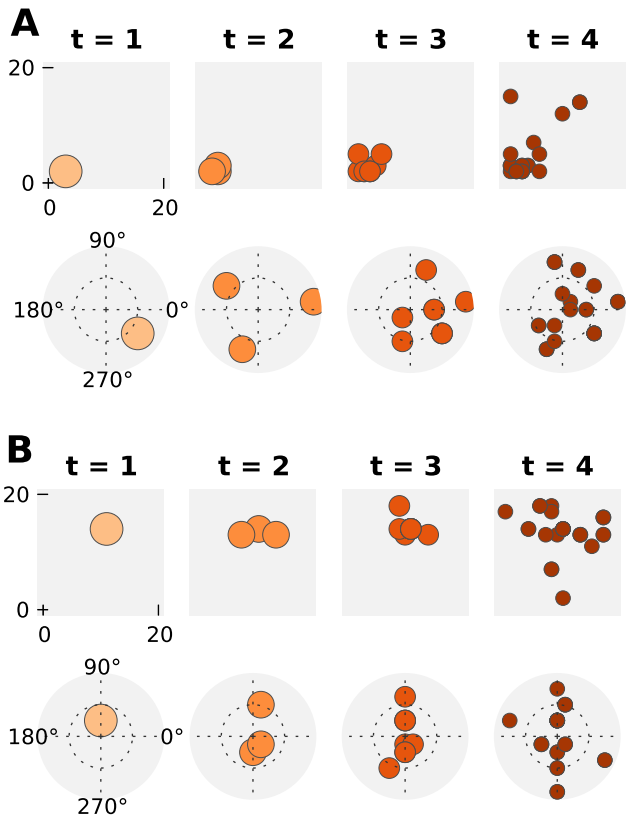

FIGURE 3.5: Spatial and angular evolution of two hidden units in the aTRBM (sub figures A & B). The upper row shows the center of each units receptive field in pixel space for the most active units in in the temporal evolution of one unit. The lower row shows the strongest frequency component of the filters for this same evolution. The unit in subfigure A shows a clear spatial preference but is orientation agnostic whilst the unit in subfigure B is less spatially selective but shows a clear preference for vertically oriented filters.

on the dynamic RF activation (green) exhibit a spiking pattern, which is temporally sparser than the spiking pattern that stems from the static RF activation (blue). This also finds expression in the time histogram of the trial-averaged firing rate shown in figure 3.6 D3. The firing rate is more peaked in the case of the dynamic RF, resembling the deterministic activation curve in figure 3.6 D1.

Spatial sparseness

Spatial sparseness (also termed population sparseness) refers to the situation where only a small number of units are significantly activated by a given stimulus. In the

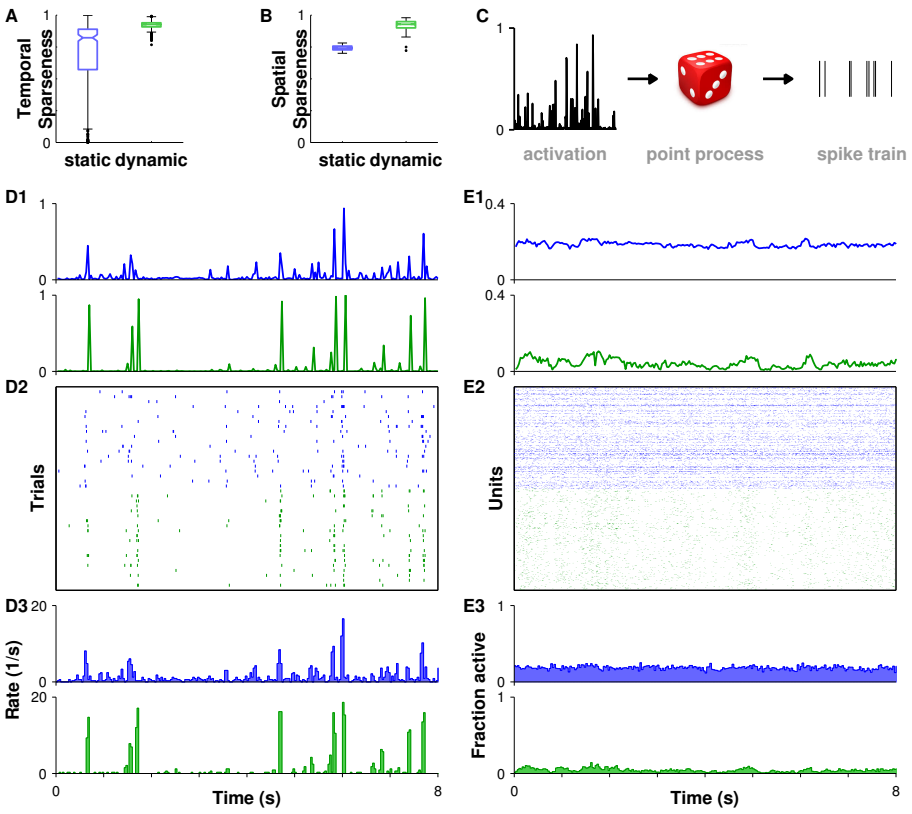

Figure 3.6: Temporal and spatial sparseness of neuronal activity for static and dynamic RF responses. (A) Temporal sparseness measured in 400 hidden layer units during 30 s of video stimulation is significantly larger for the dynamic (right) than the static (left) RF model ($P < 10^{-5}$; Wilcoxon signed rank test). (B) Spatial sparseness measured across all 400 neurons is significantly increased ($P < 10^{-5}$; Wilcoxon signed rank test) in the dynamic (right) RF model as compared to the static RF model (left). (C) Sketch of cascade model for spike train generation. During video stimulation the activation curve of a hidden layer neuron (left) expresses the deterministic probability of being active in each frame. A stochastic point process model (center) generates action potentials (rigth) according to a time-varying intensity proportional to the activation curve. (D1-D3) Temporal sparseness during 8 s of video stimulation. (D1) Activation curve of one hidden neuron for the static RF (blue) and the dynamic RF (green) model with a temporal sparseness of $S = 0.82$ and $S = 0.94$, respectively. (D2) Repeated point process realizations (n=20) using the activation curves in (D1). (D3) Firing rate estimated as time histogram from 100 repetitions for static (blue) and dynamic (green) RF model. (E1-E3) Spatial sparseness in the population of hidden layer neurons during video stimulation. (E1) Average activation curves of hidden layer units for the static (blue) and dynamic (green) RF model. (E2) Spike trains of $N = 50$ hidden layer neurons when using the static (red) or dynamic (blue) RF model. (E3) The fraction of active neurons per video frame in the total population of 400 hidden units is considerably smaller for the dynamic RF model.

natural case of time-varying stimuli this implies a small number of active neurons in any small time window while the rest of the neuron population expresses a low baseline activity. Again, we use S (equation (3.2)) to quantify spatial sparseness from the population activation $\mathbf{h}$ of hidden neurons and for each time step separately. The results depicted in Fig. 3.6 B show a significantly higher spatial sparseness when the dynamic RF was applied with a mean (median) of 0.92 (0.93) as compared to the static RF with a mean (median) of 0.74 (0.74).

We demonstrate how the spatial sparseness for the static and the dynamic RF model in the population of hidden units affects spiking activity using our cascade point process model. Fig. 3.6 E2 shows the simulated spiking activity of all 400 neurons based on the activation $\mathbf{h}(t)$ of the hidden neurons during 8 s of recording. Overall the static RF (blue) results in higher firing rates. The stimulus representation in the ensemble spike train appears more dense for the static RF (blue) than in the case of a dynamic RF (green). As shown in Fig. 3.6 E3, fewer neurons were active at any given point in time when they were driven by the dynamic RF model.

3.3 Discussion

We suggested a novel approach to unsupervised learning of spatio-temporal structure in multidimensional time-varying data. We first define the general topology of an artificial neural network (ANN) as our model class. Through a number of structural constraints and a machine learning approach to train the model parameters from the data, we arrive at a specific ANN which is biologically relevant and is able to produce activations for any given temporal input (section 3.2.1). We then extend this ANN with a Computational Neuroscience based cascade model and use this to generate trial variable spike trains (section 3.2.3).

The proposed aTRBM model integrates the recent input history over a small number of discrete time steps. This model showed superior performance to other models on a recognized benchmark dataset. When trained on natural videos that represent

smooth sequences of natural images the units in the hidden layer developed dynamic receptive fields that retain the properties of the static receptive field and represent smooth temporal transitions of their static receptive field structure. This time-extension of the previously obtained static receptive fields increase the input selectivity of each hidden unit. Consequently, each hidden unit is activated in a highly sparse manner by only specific spatio-temporal input scenarios.

3.3.1 Temporal Autoencoding model

We have introduced a new training method for TRBMs called Temporal Autoencoding and validated it by showing a significant performance increase in modelling and generation from a sequential human motion capture dataset (Figure 3.7). The gain in performance from the standard TRBM to the pre-trained aTRBM model, which are both structurally identical, suggests that our approach of autoencoding the temporal dependencies gives the model a more meaningful temporal representation than is achievable through contrastive divergence training alone. We believe the inclusion of autoencoder training in temporal learning tasks will be beneficial in a number of problems, as it enforces the causal structure of the data on the learned model.

We have shown that the aTRBM is able to learn high level structure from natural movies and account for the transformation of these features over time. The statistics of the static filters resemble those learned by other algorithms, namely gabor like patches showing preferential orientation of the filters along cardinal directions (Fig. 3.2). The distribution of preferred position, orientation and frequency (Fig. 3.3) is in accordance with results previously found by other methods [e.g. 61, 68], and the simple cell like receptive fields and cardinal selectivity is supported by neurophysiological findings in primary visual cortex [63, 64]. Importantly the temporal connectivity expressed in the weights $\mathbf{W}_M$ learned by the model is also qualitatively similar to the pattern of lateral connections in this brain area. Preferential connection between orientation-selective cells in V1 with similar orientation has been reported in higher mammals [65, 66, 69]. These lateral connections are usually thought to underlie contour integration in the visual system. Here they arise directly from training the aTRBM model to reproduce

the natural dynamics of smoothly changing image sequences. One could say that, in an unsupervised fashion, the model learns to integrate contours directly from the dataset.

The aTRBM presented here can be easily embedded into a deep architecture, using the same training procedure in a greedy layer-wise fashion. This might allow us to study the dynamics of higher-order features (i.e. higher order receptive fields) in the same fashion as was done here for simple visual features. In this way one could envisage applications of our approach to pattern recognition and temporal tasks, such as object tracking or image stabilization.

3.3.2 The dynamic RF is a potential mechanism of sparse stimulus encoding

There is strong evidence that encoding of natural stimuli in sensory cortices - specifically in the visual and auditory system - is sparse in space and time (see Section 3.1). Sparse coding seems to be a universal principle widely employed both in vertebrate and invertebrate nervous systems and it is thought to reflect the sparsity of natural stimulus input [40, 44, 70]. Deciphering the neuronal mechanisms that underlie sparse coding at the level of cortical neurons is a topic of ongoing research.

Population sparseness critically depends on the network topology. An initially dense code in a smaller population of neurons in the sensory periphery is transformed into a spatially sparse code by diverging connections onto a much larger number of neurons in combinations with highly selective and possibly plastic synaptic contacts. This is particularly well studied in the olfactory system of insects where feed-forward projections from the antennal lobe diverge onto a much larger number of Kenyon cells in the mushroom body with random and weak connectivity [71] and thereby translate a dense combinatorial code in the projection neuron population into a sparse code in the Kenyon cell population [72, 73]. Also in the mammalian visual system the number of retinal cells at the periphery, which employ a relatively dense code, is small compared to the cortical neuron population in the primary visual cortex [40]. Another important mechanism responsible for spatial sparseness is global and structured lateral inhibition

that has been shown to increase population sparseness in the piriform cortex [74] and to underlie non-classical receptive fields in the visual cortex [48].

A network architecture of diverging connections and mostly weak synapses is reflected in the RBM models introduced here (see section 3.4 and figure 3.1). Initially an all-to-all connection between the units in the input and in the hidden layer is given, but due to the sparsity constraint most synaptic weights become effectively zero during training. By this, hidden layer units sparsely mix input signals in many different combinations to form heterogeneous spatial receptive fields (figure 3.2) as observed in the visual cortex [46, 49, 75]. A novelty of the aTRBM is that the learning of sparse connections between hidden units also applies to the temporal domain resulting in heterogeneous spatio-temporal receptive fields (figure 3.4 A). Our spike train simulations (figure 3.6) match the experimental observations in the visual cortex: sparse firing in time and across the neuron population [e.g. 46, 49].

Experimental evidence in the visual cortex suggests that temporally sparse responses of single neurons to naturalistic dynamic stimuli show less variability across trials than responses to artificial noise stimuli [48, 53]. Equally, in the insect olfactory system the temporally sparse stimulus responses in the Kenyon cells have been shown to be highly reliable across stimulus repetitions [76]. In our model approach, response variability is not affected by the choice of a static or dynamic RF model. The trained aTRBM provides a deterministic activation $\mathbf{h}$ across the hidden units. In the cascade model (Fig. 3.6 C) we generated spike trains according to a stochastic point process model. Thus the trial-to-trial spike count variability in our model is solely determined by the point process stochasticity and is thereby independent of the RF type. Spike frequency adaptation [SFA, 77] is an important cellular mechanism that increases temporal sparseness [78, 79] and at the same time reduces the response variability of single neuron [80–83] and population activity [78, 84, 85]. Other mechanisms that can facilitate temporal sparseness are feed-forward [86] and feed-back inhibition [87].

3.3.3 Why sparse coding?

Encoding of a large stimulus space can be realized with a dense code or with a sparse code. In a dense coding scheme few neurons encode stimulus features in a combinatorial fashion where each neuron is active for a wide range of stimuli and with varying response rates (stimulus tuning). Dense codes have been described in different systems, prominent examples of which are the peripheral olfactory system of invertebrates and vertebrates [e.g. 88–91], and the cortical motor control system of primates [e.g. 92, 93].

In sensory cortices a sparse stimulus representation is evident (see section 3.1). Individual neurons have highly selective receptive fields and a large number of neurons is required to span the relevant stimulus space. What are the benefits of a sparse code that affords vast neuronal resources to operate at low spiking rates? We briefly discuss theoretical arguments that outline potential computational advantages of a sparse stimulus encoding.

The first and most comprehensive argument concerns the energy efficiency of information transmission. Balancing the cost of action potential generation relative to the cost for maintaining the resting state with the sub-linear increase of information rate with firing rate in a single neuron leads to an optimal coding scheme where only a small percentage of neurons is active with low firing rates [94–96].

The argument outlined above is limited to the transmission of information and conditioned on the assumption of independent channels. Nervous systems, however, have evolved as information processing systems and information transmission plays only a minor role. Then the more important question is how does sparse coding benefit brain computation? We consider three related arguments. In a spatially sparse code, single elements represent highly specific stimulus features. A complex object can be formed only through the combination of specific features at the next level, a concept that is often referred to as the binding hypothesis [97]. In this scheme, attentional mechanisms could mediate a perceptual focus of objects with highly specific features by enhancing co-active units and suppressing background activity. In a dense coding scheme, enhanced silencing of individual neurons would have an unspecific effect.

A spatially sparse stimulus representation can facilitate the formation of associative

memories [98]. A particular object in stimulus space activates a highly selective set of neurons. Using an activity-dependent mechanism of synaptic plasticity allows the formation of stimulus-specific associations in this set of neurons. This concept is theoretically and experimentally well studied in the insect mushroom body where the sparse representation of olfactory stimuli at the level of the Kenyon cells [99, 100] is thought to underlie associative memory formation during classical conditioning [73, 101–103]. This system has been interpreted in analogy to machine learning techniques that employ a strategy of transforming a lower dimensional input space into a higher dimensional feature space to improve stimulus classification [73, 104, 105].

Theories of temporal coding acknowledge the importance of the individual spike and they receive support from accumulating experimental evidence [e.g. 41, 47, 106]. Coding schemes that rely on dynamic formation of cell assemblies and exact spike timing work best under conditions of spatially and a temporally sparse stimulus representations and low background activity.

3.4 Methods

3.4.1 Machine Learning Methods

To develop the Temporal Autoencoding training method for Temporal Restricted Boltzmann Machines used in this work, we have extended upon existing work in the field of unsupervised feature learning.

Existing Static Models of Unsupervised Learning

Two unsupervised learning methods well known within the Machine Learning community, Restricted Bolzman Machines (RBMs) and Autoencoders (AEs) [107, 108] form the basis of our temporal autoencoding approach. Both are two-layer neural networks, all to all connected between the layers but with no intra-layer connectivity. The models consist of a visible and a hidden layer, where the visible layer represents the input to the model whilst the hidden layer's job is to learn a meaningful representation of

the data in some other dimensionality. We will represent the visible layer activation variables by v_i, the hidden activations by h_j and the vector variables by $\mathbf{v} = \{v_i\}$ and $\mathbf{h} = \{h_j\}$ where $i = [1..N]$ and $j = [1..S]$ index the individual neurons in the visible and hidden layers respectively.

Restricted Boltzmann Machines are stochastic models that assume symmetric connectivity between the visible and hidden layers (see Figure 3.1a) and seek to model the structure of a given dataset. They are energy-based models, where the energy of a given configuration of activations $\{v_i\}$ and $\{h_j\}$ is given by

$$E_{RBM}(\mathbf{v}, \mathbf{h}|\mathbf{W}, \mathbf{b}_v, \mathbf{b}_h) = -\mathbf{v}^\top \mathbf{W}\mathbf{h} - \mathbf{b}_v^\top \mathbf{v} - \mathbf{b}_h^\top \mathbf{h},$$

and the probability of a given configuration is given by

$$P(\mathbf{v}, \mathbf{h}) = \exp\left(-E_{RBM}(\mathbf{v}, \mathbf{h}|\mathbf{W}, \mathbf{b}_v, \mathbf{b}_h)\right)/Z(\mathbf{W}, \mathbf{b}_v, \mathbf{b}_h),$$

where $Z(\mathbf{W}, \mathbf{b}_v, \mathbf{b}_h)$ is the partition function. One can extend the RBM to continuous-valued visible variables by modifying the energy function, to obtain the Gaussian-binary RBM

$$E_{RBM}(\mathbf{v}, \mathbf{h}|\mathbf{W}, \mathbf{b}_v, \mathbf{b}_h) = -\frac{\mathbf{v}^\top}{\sigma^2}\mathbf{W}\mathbf{h} + \frac{\|\mathbf{b}_v - \mathbf{v}\|^2}{2\sigma^2} - \mathbf{b}_h^\top \mathbf{h}.$$

RBMs are usually trained through contrastive divergence, which approximately follows the gradient of the cost function

$$\begin{aligned} CD_n(\mathbf{W}, \mathbf{b}_v, \mathbf{b}_h)) = \quad & KL(P_0(\mathbf{v}|\mathbf{W}, \mathbf{b}_v, \mathbf{b}_h)||P(\mathbf{v}|\mathbf{W}, \mathbf{b}_v, \mathbf{b}_h)) \\ & -KL(P_n(\mathbf{v}|\mathbf{W}, \mathbf{b}_v, \mathbf{b}_h)||P(\mathbf{v}|\mathbf{W}, \mathbf{b}_v, \mathbf{b}_h)), \end{aligned}$$

where P_0 is the data distribution and P_n is the distribution of the visible layer after n MCMC steps [26]. The function CD_n gives an approximation to maximum-likelihood (ML) estimation of the weight matrix $\mathbf{w}$. Maximizing the marginal probability $P(\{\mathbf{v}\}_D|\mathbf{W}, \mathbf{b}_v, \mathbf{b}_h)$ of the data $\{\mathbf{v}\}_D$ in the model leads to a ML-estimate which is hard to compute, as it involves averages over the equilibrium distribution $P(\mathbf{v}|\mathbf{W}, \mathbf{b}_v, \mathbf{b}_h)$. The parameter update for an RBM using CD learning is then given by

$$\Delta\theta \propto \left\langle \frac{\partial E_{RBM}}{\partial \theta} \right\rangle_0 - \left\langle \frac{\partial E_{RBM}}{\partial \theta} \right\rangle_n,$$

where the $\langle\rangle_n$ denotes an average over the distribution P_n of the hidden and visible variables after n MCMC steps. The weight updates then become

$$\Delta \mathbf{W}_{i,j} \propto \frac{1}{\sigma^2} \langle v_i h_j \rangle_0 - \frac{1}{\sigma^2} \langle v_i h_j \rangle_n .$$

In general, $n = 1$ already gives good results [15].

Autoencoders are deterministic models with two weight matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ representing the flow of data from the visible-to-hidden and hidden-to-visible layers respectively (see Figure 3.1b). AEs are trained to perform optimal reconstruction of the visible layer, often by minimizing the mean-squared error (MSE) in a reconstruction task. This is usually evaluated as follows: Given an activation pattern in the visible layer $\mathbf{v}$, we evaluate the activation of the hidden layer by $\mathbf{h} = sigm(\mathbf{v}^\top \mathbf{W}_1 + \mathbf{b}_h)$, where we will denote the bias in the hidden layer by $\mathbf{b}_h$. These activations are then propagated back to the visible layer through $\hat{\mathbf{v}} = sigm(\mathbf{h}^\top \mathbf{W}_2 + \mathbf{b}_v)$ and the weights $\mathbf{W}_1$ and $\mathbf{W}_2$ are trained to minimize the distance measure between the original and reconstructed visible layers. Therefore, given a set of image samples $\{\mathbf{v}^d\}$ we can define the cost function. For example, using the squared euclidean distance we have a cost function of

$$\mathcal{L}(\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_v, \mathbf{b}_h | \{\mathbf{v}^d\}) = \sum_d \|\mathbf{v}^d - \hat{\mathbf{v}}^d\|^2.$$

The weights can then be learned through stochastic gradient descent on the cost function. Autoencoders often yield better representations when trained on corrupted versions of the original data, performing gradient descent on the distance to the uncorrupted data. This approach is called a denoising autoencoder (dAE) [17]. Note that in the AE, the activations of all units are continuous and not binary, and in general take values between 0 and 1.

Existing Dynamic Models of Unsupervised Learning

To date, a number of RBM-based models have been proposed to capture the sequential structure in time series data. Two of these models, the Temporal Restricted Boltzmann Machine and the Conditional Restricted Boltzmann machine, are introduced below.

Temporal Restricted Boltzmann Machines (TRBM) [27] are a temporal extension of the standard RBM whereby feed-forward connections are included from previous time steps between hidden layers, from visible to hidden layers and from visible to visible layers (see Figure 3.1d). Learning is conducted in the same manner as a normal RBM using contrastive divergence and it has been shown that such a model can be used to learn non-linear system evolutions such as the dynamics of a ball bouncing in a box [27]. A more restricted version of this model, discussed in [32] can be seen in figure 3.1d and only contains temporal connections between the hidden layers. We will restrict ourselves to this model architecture in this paper.

Similarly to our notation for the RBM, we will write the visible layer variables as $\mathbf{v}^0, \ldots, \mathbf{v}^T$ and the hidden layer variables as $\mathbf{h}^0, \ldots, \mathbf{h}^T$. More precisely, $\mathbf{v}^T$ is the visible activation at the current time t and $\mathbf{v}^i$ is the visible activation at time $t-(T-i)$. The energy of the model for a given configuration of $\mathcal{V} = \{\mathbf{v}^0, \ldots, \mathbf{v}^T\}$ and $\mathcal{H} = \{\mathbf{h}^0, \ldots, \mathbf{h}^T\}$ is given by

$$E(\mathcal{H}, \mathcal{V}|\mathcal{W}) = \sum_{t=0}^{T} E_{RBM}(\mathbf{h}^t, \mathbf{v}^t|\mathbf{W}, \mathbf{b}) - \sum_{t=1}^{M} (\mathbf{h}^T)^\top \mathbf{W}_{T-t} \mathbf{h}^t, \tag{3.1}$$

where we have used $\mathcal{W} = \{\mathbf{W}, \mathbf{W}_1, \ldots, \mathbf{W}_M\}$, where $\mathbf{W}$ are the static weights and $\mathbf{W}_1, \mathbf{W}_2, \ldots \mathbf{W}_M$ are the delayed weights for the temporally delayed hidden layers $\mathbf{h}_{T-1}, \mathbf{h}_{T-2}, \ldots, \mathbf{h}_0$ (see figure 3.1d). Note that, unlike the simple RBM, in the TRBM, the posterior distribution of any unit in the hidden layer conditioned on the visible layer is not independent of other hidden units, due to the connection between the delayed RBM's. This makes it harder to train the TRBM, as sampling from the hidden layer requires Gibbs sampling until the system has relaxed to its equilibrium distribution. This has led researcher to consider other types of probabilistic models for dynamic data.

Conditional Restricted Boltzmann Machines (CRBM) as described in [28] contain no temporal connections from the hidden layer but include connections from the visible layer at previous time steps to the current hidden and visible layers. The model architecture can be seen in Figure 3.1c. In the CRBM, the past nodes are conditioned on, serving as a trial-specific bias. These units are shown in orange in

Figure 3.1c. Again, learning with this architecture requires only a small change to the energy function of the RBM and can be achieved through contrastive divergence. The CRBM is possibly the most successful of the temporal RBM models to date and has been shown to both model and generate data from complex dynamical systems such as human motion capture data and video textures [33].

Temporal Autoencoding Training for TRBMs

Much of the motivation for this work is to gain insight into the typical evolution of learned hidden layer features or RFs present in natural movie stimuli. With the existing CRBM this is not possible as it is unable to explicitly model the evolution of hidden features without resorting to a deep network architecture. Sparse coding models, as proposed by [68] overcome this restriction by learning complex filters, allowing for phase dynamics by multiplying the filters by complex weights whose dynamics are governed by phase variables. However, the evolution of the filters is indirectly modelled by the phase variables, not allowing for a direct biological interpretation.

The TRBM, in comparison, provides an explicit representation of the evolution of hidden features but, as we show, can be difficult to train using the standard algorithm. While this model does not have a direct biological influence, it's artificial neural network structure allows for a biological interpretation of its function and indeed, producing a spiking neural network implementation of this approach would make for interesting future research. Here, we present a new pre-training method for the TRBM called Temporal Autoencoding (aTRBM) that dramatically improves its performance in modelling temporal data.

Training Procedure The energy of the model is given by equation (3.1) and is essentially an M-th order autoregressive RBM which is usually trained by standard contrastive divergence [27]. Here we propose to train it with a novel approach, highlighting the temporal structure of the stimulus. A summary of the training method is described in table 3.1. First, the individual RBM visible-to-hidden weights $\mathbf{W}$ are initialized through contrastive divergence learning with a sparsity constraint on static

samples of the dataset. After that, to ensure that the weights representing the hidden-to-hidden connections ($\mathbf{W}_t$) encode the dynamic structure of the ensemble, we initialize them by pre-training in the fashion of a denoising Autoencoder as will be described in the next section. After the Temporal Autoencoding is completed, the whole model (both visible-to-hidden and hidden-to-hidden weights) is trained together using contrastive divergence (CD) training.

One can regard the weights $\mathbf{W}$ as a representation of the static patterns contained in the data and the $\mathbf{W}_t$ as representing the transformation undergone by these patterns over time in the data sequences. This allows us to separate the representation of *form* and *motion* in the case of natural image sequences, a desirable property that is frequently studied in natural movies (see [62]). Furthermore, it allows us to learn how these features should evolve along time to encode the structure of the movies well. In the same way as static filters learned in this way often resemble RFs in visual cortex, the temporal projections learned here could be compared to lateral connections and correlations between neurons in visual cortex.

TABLE 3.1: Autoencoded TRBM Training Steps

Step	Action
1. Static RBM Training	Constrain the static weights $\mathbf{w}$ using CD on single frame samples of the training data
2. Temporal Autoencoding	Constrain the temporal weights $\mathbf{w}_1$ to $\mathbf{w}_d$ using a denoising autoencoder on multi-frame samples of the data
3. Model Finalisation	Train all model weights together using CD on multi-frame samples of the data

Temporal Autoencoding The idea behind many feature extraction methods such as the autoencoder [17] and reconstruction ICA [109] is to find an alternative encoding for a set of data that allows for a good reconstruction of the dataset. This is frequently combined with sparse priors on the encoder. We propose to use a similar framework for TRBM's based on filtering (see [110]) instead of reconstructing through the use of a denoising autoencoder (dAE). The key difference between an AE and a dAE is that random noise is added to each training sample before it is presented to the network, but the training procedure still requires the dAE to reproduce the *original* training

data, before the noise was added, thereby *denoising* the training data. The addition of noise forces the model to learn reliable and larger scale structure from the training data as local perturbations from the added noise will change each time a sample is presented and are therefore unreliable.

In the aTRBM, we leverage the concept of denoising by treating previous samples of a sequential dataset as *noisy* versions of the current time point that we are trying to reproduce. The use of the term *noise* here is somewhat of a misnomer, but is used to keep in line with terminology from dAE literature. In the aTRBM case, no noise is added to the training data, but the small changes that exist between consecutive frames of the dataset are conceptually considered to be *noise* in the terms that we want to remove these changes from previous samples to be able to correctly reproduce or predict the data at the current time point. We can therefore use a dAE approach to constrain the temporal weights. In this sense, we consider the activity of the time-lagged visible units as noisy observations of the systems state, and want to infer the current state of the system. To this end, we propose pre-training the hidden-to-hidden weights of the TRBM by minimizing the error in predicting the present data frame from the previous observations of the data. This is similar to the approximation suggested by [32], where the distribution over the hidden states conditioned on the visible history is approximated by the filtering distribution. The training is done as follows. After training the weights $\mathbf{W}$ we consider the model to be a deterministic Multi-Layer Perceptron with continuous activation in the hidden layers. We then consider the M delayed visible layers as features and try to predict the current visible layer by projecting through the hidden layers. In essence, we are considering the model to be a feed-forward network, where the delayed visible layers would form the input layer, the delayed hidden layers would constitute the first hidden layer, the current hidden layer would be the second hidden layer and the current visible layer would be the output. We can then write the prediction of the network as $\hat{\mathbf{v}}_d^T(\mathbf{v}_d^0, \mathbf{v}_d^1, \ldots, \mathbf{v}_d^{T-1})$, where the d index runs over the data points. The exact format of this function is described in algorithm 3. We therefore

minimize the reconstruction error given by

$$\mathcal{L}(\mathcal{W}) = \sum_d \left\| \mathbf{v}_d^T - \hat{\mathbf{v}}^T(\mathbf{v}_d^0, \mathbf{v}_d^1, \ldots, \mathbf{v}_d^{T-1}) \right\|^2,$$

where the sum over d goes over the entire dataset. The pretraining is described fully in algorithm 3.

We train the temporal weights $\mathbf{W}_i$ one delay at a time, minimizing the reconstruction error with respect to that temporal weight stochastically. Then the next delayed temporal weight is trained keeping all the previous ones constant. The learning rate η is set adaptively during training following the advice given in [35].

Algorithm 3 Pre-Training Temporal weights through Autoencoding

for each sequence of data frames $I(t-T), I(t-(T-1)) \ldots, I(t)$, we take $\mathbf{v}^T = I(t), \ldots, \mathbf{v}^0 = I(t-T)$ and **do**

  for $d = 1$ **to** M **do**

    for $i = 1$ **to** d **do**

      $\mathbf{h}^{T-i} = sigm(\mathbf{W}\,\mathbf{v}^{T-i} + \mathbf{b}_h)$

    end for

    $\mathbf{h}^T = sigm(\sum_{j=1}^{d} \mathbf{W}_j \mathbf{h}^{T-j} + \mathbf{b}_h)$, $\hat{\mathbf{v}}^T = sigm(\mathbf{W}^\top \mathbf{h}^T + \mathbf{b}_v)$

    $\epsilon(\mathbf{v}^T, \hat{\mathbf{v}}^T) = |\mathbf{v}^T - \hat{\mathbf{v}}^T|^2$

    $\Delta \mathbf{W}_d = \eta\, \partial\epsilon / \partial \mathbf{W}_d$

  end for

end for

3.4.2 Model Analysis

Sparseness index

To measure spatial and temporal sparsenes we employ the sparseness index introduced by [67] as

$$S = 1 - \frac{(\Sigma\,|a|/n)^2}{\Sigma\,(a^2/n)}. \tag{3.2}$$

where a is the neural activation and n is the total number of samples used in the calculation. To quantify sparseness of the hidden unit activation we stimulate the aTRBM model that was previously trained on the Holywood2 dataset (cf. Section 3.2.2) with a single video sequences of approx. 30 s length at a frame rate of 30 /s (total 897 frames) and measure the activation $\mathbf{h}$ of all hidden units during each video frame. *Spatial sparseness* refers to the distribution of activation values across the neuron population and is identical to the notion of population sparseness [111]. To quantify spatial sparseness we employ S to the activation values $\mathbf{h}$ across all 400 units for each of the time frames separately, resulting in 897 values. We use the notion of *temporal sparseness* to capture the distribution of activation values across time during a dynamic stimulus scenario [48]. High temporal sparseness of a particular unit indicates that this unit shows strong activation only during a small number of stimulus frames. Low temporal sparseness indicates a flat activation curve across time. Our definition of temporal sparseness can easily be related to the definition of lifetime sparseness [48] if we consider each video frame as an independent stimulus. However, natural videos do exhibit correlations over time and successive video frames are thus generally not independent. Moreover, the dynamic RF model learns additional time dependencies. We employ S to quantify the temporal sparseness across the 897 single frame activation values for each neuron separately, resulting in 400 single unit measures.

Temporal and spatial sparseness are compared for the cases of a static RF and a dynamic RF. The static RF is defined by looking at the response of the aTRBM when all temporal weights are set to 0. This is equivalent to training a standard RBM.

Cascade spike generation model

From the activation variable h of the hidden units in our aTRBM model we generated spike train realizations using a cascade point process model [112] as described in (figure 3.6 C). For each hidden unit we recorded its activation h during presentation of a video input. This time-varying activation expresses a probability between 0 and 1 of being active in each video frame. We linearly interpolated the activation curve

to achieve a time resolution of 20 times the video frame rate. We then used the activation curve as intensity function to simulate single neuron spike train realizations according to the non-homogeneous Poisson process [113]. This can be generalized to other rate-modulated renewal and non-renewal point process models [85, 114]. The expectation value for the trial-to-trial variability of the spike count is determined by the point process stochaisticity [114] and thus independent of the activating model. We estimated neural firing rate from a single hidden neuron across repeated simulation trials or from the population of all 400 hidden neurons in a single simulation trial using the Peri Stimulus Time Histogram [115–117] with a bin width corresponding to a single frame of the video input sequence.

3.4.3 Benchmark Evaluation - Human Motion Dynamics

We assessed the aTRBM's ability to learn a good representation of multi-dimensional temporal sequences by applying it to the 49 dimensional human motion capture data described by [28] and, using this as a benchmark, compared the performance to a TRBM without our pretraining method and Graham Taylor's example CRBM implementation[1]. All three models were implemented using Theano [34], have a temporal dependence of 6 frames [as in 28] and were trained using minibatches of 100 samples for 500 epochs[2]. The training time for all three models was approximately equal. Training was performed on the first 2000 samples of the dataset after which the models were presented with 1000 snippets of the data not included in the training set and required to generate the next frame in the sequence. For all three models, the visible-to-hidden connections were initialized with contrastive divergence on static snapshots of the data. For the TRBM we then proceeded to train all the weights of the model through contrastive divergence, whereas in the aTRBM case we initialized the weights through

[1]CRBM implementation available at *https://gist.github.com/2505670*

[2]For the standard TRBM, training epochs were broken up into 100 static pretraining and 400 epochs for all the temporal weights together. For the aTRBM, training epochs were broken up into 100 static pretraining, 50 Autoencoding epochs per delay and 100 epochs for all the temporal weights together

temporal autoencoding as described in algorithm 3, before training the whole model with CD. The CRBM was also trained using contrastive divergence. In addition, we created a deterministic model which has the same structure as the aTRBM but was trained using only the first two training steps listed in table 3.1 which we will refer to as an Autoencoded Multi Layer Perceptron (AE/MLP).

Data generation in the aTRBM is done by taking a sample from the hidden layers at $t-6$ through $t-1$ and then Gibbs sampling from the RBM at time t while keeping the others fixed as biases. This is the filtering approximation from [32]. The visible layer at time t is initialized with noise and we sample for 30 Gibbs steps from the model. Data generation from the AE/MLP is done deterministically whereby the visible layers at $t-6$ through $t-1$ are set by the data and the activation is the propagated through to the visible layer at t for the sample prediction. We are interested in the performance of the AE/MLP to determine whether or not their is an advantage to the stochatiscity of the RBM models in this prediction task. To this end, we also tested the deterministic performance of the three RBM models discussed here but the results were much poorer than those where the model generated data stochastically.

The results of a single trial prediction for four random dimensions of the dataset and the mean squared error (MSE) of the RBM model predictions over 100 repetitions for all 49 dimensions of the task can be seen in can be seen in Figure 3.7. While the aTRBM is able to significantly outperform both the standard TRBM and CRBM models in this task during single trial prediction (3 leftmost columns), the deterministic AE/MLP model (middle column) predicts with an even lower error rate. In the 3 rightmost columns, we produce 50 single trial predictions per model type and take their mean as the prediction for the next frame in order to see if averaging over trials reduces the inherent variance of a single trial prediction. The performance of the CRBM and the aTRBM improve markedly and the aTRBM outperforms all other models. It should be noted that this process is not the same as taking the mean activation of the model (ie. a deterministic pass through the model with no sampling) which severely under performs the results shown here. Instead, averaging over multiple stochastic samples of the model proves to be advantageous in creating a low error estimate of the next

frame. These results show not only the advantage of the aTRBM over the CRBM in this task, but also that of the stochastic models over the deterministic AE/MLP. Although single trial predictions from the aTRBM are not quite as accurate as those of the AE/MLP, the aTRBM is able to generate unique predictions stochastically at each trial, something the deterministic AE/MLP is not able to achieve. If one is interested purely in minimising the MSE of the prediction, one can still use the aTRBM to generate and average over multiple trials which reduces the MSE and out performs the AE/MLP.

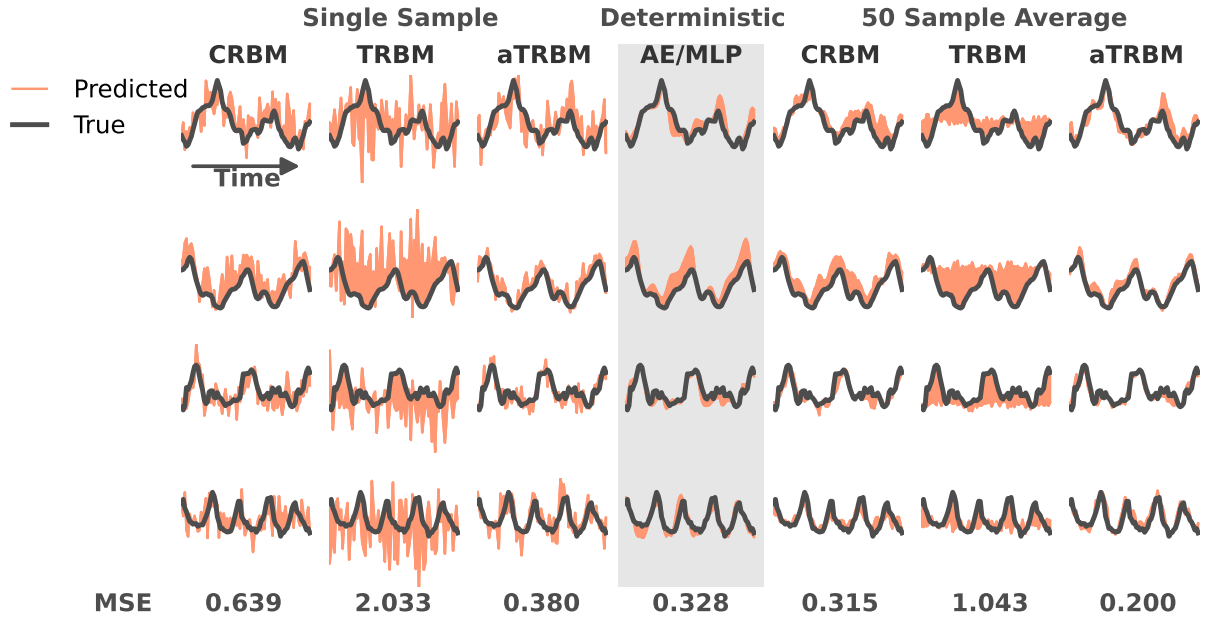

Figure 3.7: CRBM, TRBM, aTRBM and AE/MLP are used to fill in data points from motion capture data [28]. Four random dimensions of the motion data are shown along with the their model reconstructions from a single trial (three leftmost columns), deterministically (middle column, grey), and as an average of 50 generated trials (three rightmost columns). At the bottom of each column, one can see the Mean Squared Error (MSE) of the reconstruction over all 49 dimensions of the entire 1000 sample test data. The aTRBM is the best performer of the single trial predictors, producing a lower MSE than the CRBM and TRBM. The deterministic AE/MLP has marginally better MSE performance than the aTRBM, although at the cost of no longer being a generative model. We find however, that if one generates 50 single trial predictions from the aTRBM model and then takes the average of these, the MSE is reduced ever further, allowing the aTRBM to far outperform the AE/MLP. From this point of view, the aTRBM is the more advantageous model in the respect that it can generate non-deterministic single trial predictions, and if one is interested in reducing the MSE as far as possible, can be averaged over a number of trials, thereby reducing the single trial variation and increasing the predictor performance.

4

Decoding of voluntary hand movements in local field potential and extracellular spiking activity from patients with Parkinson disease and Dystonia

Abstract

The primate basal ganglia (BG) play an important role in the control and feedback of voluntary movements. Here we studied the neural representation of voluntary movements in the human BG. We utilized micro-electrode recordings during electrode implantation for deep brain stimulation (DBS) in patients with Parkinson disease and Dystonia and analyzed neural activity in the subthalamic nucleus (STN) or the globus palidus (GPi), respectively. In an experimental paradigm of self-paced wrist movements we found significant representations of different kinematic movement parameters such as position, velocity, or acceleration, both in multi-unit spiking activity (MUA) and in the band-limited power of the local field potential (LFP). The quantitative correlation with specific movement parameters varied considerably across patients. In a decoding approach we predicted the movement trajectory from the neural signals during single trials. High decoding performance with up to 86% correlation of predicted and actual movement trajectory was achieved when using multiple signal types simultaneously. Our results indicate that kinematic movement parameters are encoded at the level of neuronal populations in the human STN and GPi and that movements can be successfully decoded in the single trial.

4.1 Introduction

Deep brain stimulation (DBS) of the basal ganglia (BG) is a very efficacious and relatively safe treatment option for patients with severe movement disorders. It has been established and refined over the past 25 years and has now become widely available in most developed countries. Patients with Parkinsons disease (PD) whose symptoms such as tremor, bradykinesia and rigidity cannot be adequately controlled with medication or in whom side effects to the medication (mainly fluctuations) occur are typical candidates for DBS therapy [118]. Patients with dystonia make another group that can largely benefit from DBS treatment [119, 120]. Dystonia may result in involuntary muscle activity leading to abnormal postures, muscle twitching or repetitive movements [121]. In most cases no satisfactory treatment using only medication is available. Deep Brain Stimulation involves the surgical implantation of a brain pacemaker which emits high frequency electrical stimulation via deep brain electrodes to the local brain region, usually targeted to the subthalamic nucleus (STN for PD) or the internal part of the globus pallidus (GPi for dystonia), causing changes in the brains activity and a reduction in symptoms. Whilst this technique is a clinically approved method that can greatly improve quality of life for the patients treated, the precise mechanism of action of DBS is still not completely understood. DBS also provides a rare opportunity to record signals from deep within the human brain and to gain a better understanding of neural processing within the affected regions [122].

The STN and the GPi are located within the BG, which is a group of subcortical nuclei that are connected with the cerebral cortex through different functional loops [123]. The intrinsic BG circuitry can broadly be divided into a so-called direct and indirect pathway. The direct pathway is associated with facilitation of motor plans that are generated in the cortex while the indirect pathway simultaneously inhibits competing motor plans. Within this circuitry both the STN and GPi occupy strategic positions: the GPi is the main output nucleus of the BG, exerting a substantial inhibitory control on the thalamus which in turn has excitatory projections to the cortex. The STN is part of the indirect pathway and in turn has excitatory projections

on the GPi, thus controlling the output of the BG. Amongst others, the STN and GPi are an area associated with voluntary movement control, action selection and feedback [124, 125]. Neural disfunction within this region is strongly related to a number of movement disorders including Parkinson's disease and dystonia, making it a primary target for DBS. It is an ongoing area of research to determine how movements are represented in this brain region which in turn could allow us to better understand how this process breaks down in disease [126–128].

Here, we investigate movement related neuronal signals in the Basal Ganglia. We recorded both, the LFP and extracellular spiking signals intraoperatively from four patients receiving a DBS implant. The patients were asked to perform self paced movements of their wrists during the experiment and we use measurements of wrist position along with the neural recordings to investigate the neural representations of movement. We are specifically interested in the relationship between distinct kinematic parameters such as position, velocity and acceleration and the neural activity.

4.2 Methods

4.2.1 Experimental Paradigm

Four patients (2 PD, 2 dystonia patients) participated in this experiment and informed consent was given (see table 4.1 for details). PD patients were withdrawn from their dopaminergic medication overnight to allow for a better assessment of their predominant symptoms in surgery. The experiment was performed intraoperatively during the surgical implantation of a DBS electrodes and before chronic stimulation began. As a standard procedure during surgery the patients are woken up from propofol anesthesia to allow for tuning of the DBS electrode, providing a short time period in which to conduct the experiments. The paradigm was started only when the patients were awake and aware of the situation and showed full responsiveness. Patients were pre-coached on the experimental paradigm prior to surgery. The paradigm required them to perform self paced movements of one of their hands (contralateral to the electrode

placement) extending their wrist and then returning it to a resting position.

Table 4.1: **Patient Details**

Patient #	**1**	**2**	**3**	**4**
Sex	M	M	M	M
Diagnosis	MDS	PD	PD	CD
Disease duration	44	7	23	1
Age at Surgery	47	60	55	45
Target	GPi	STN	STN	GPi
Coordinates (mm)	x=20.9 y=-4.4 z=-3.6	x=11.7 y=2.8 z=-4.87	x=14.4 y=2.5 z=-5.0	x=20.4 y=-3.5 z=-4.87
Pre-OP UPDRS-III OFF/ON	TH: 15.6 AC-PC: 22.5	TH: 18.5 AC-PC: 22.5	TH: 17.8 AC-PC: 23.0	TH: 17.8 AC-PC: 24.7
Pre-OP TWSTRS	N.A.	26/19	44/12	N.A.
Pre-OP Medication	None	800 mg Levodopa	48 mg Apomorphine 100 mg Levodopa 200 mg Amantadine	10.5 mg Tetrabenazine 200 mg Katadolon 300 mg Allopurinol

m = male; MDS = myoclonus dystonia syndrome; PD = Parkinson's disease; CD = cervical dystonia; target coordinates are given in millimeters with respect to the midcommisural point (x-axis), to the midline of the third ventricle (y-axis) and to the AC-PC line (z-axis); UPDRS-III = Unified Parkinson's disease Rating Scale Part III - Motor Examination; OFF/ON = without/with dopamingergic medication; TWSTRS = Toronto Western Spasmodic Torticollis Rating Scale.

4.2.2 Data Acquisition

Three signal types were recorded during the experiment. The patients movements were recorded using an electronic goniometer (SG65, Biometrics Ltd, Newport, UK) which converted the angle of their wrists into a voltage trace. The goniometer was not pre-calibrated to provide an exact angular reading but is appropriate for capturing general movement parameters. Two types of neural signals were recorded in the form of Local Field Potential (LFP) and high-frequency spiking activity along the trajectory that

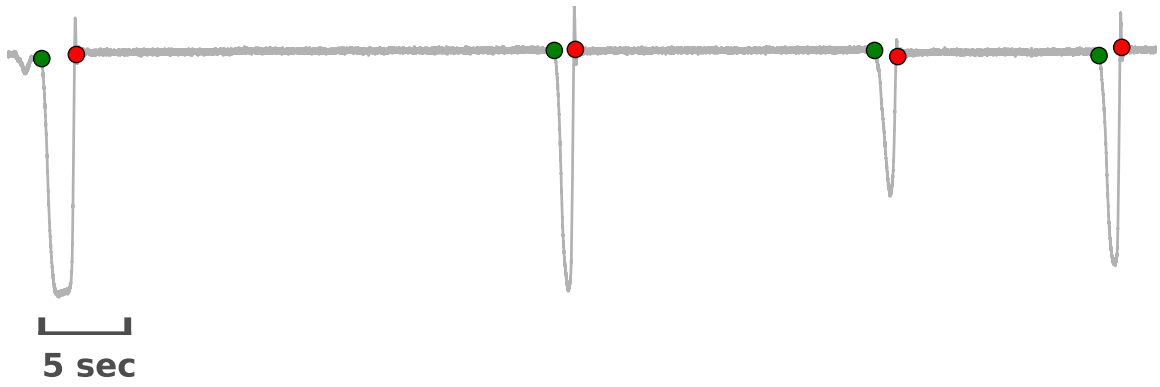

Figure 4.1: **Raw recording of the goniometer.** Individual wrist movements result in deflections of the goniometer signal that represent the movement extent at any given point in time. Green and red markers indicate onset and offset respectively of individual self-paced wrist movements

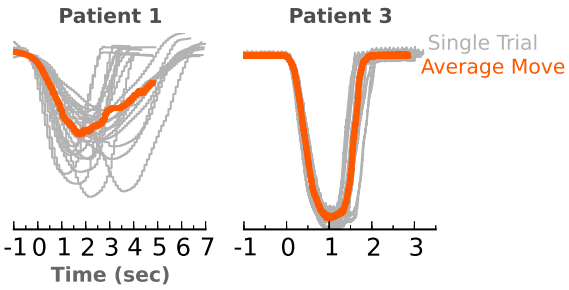

Figure 4.2: **Movement variability.** This figure shows both single trial and averaged movements (aligned to movement onset) as recorded by the goniometer.

aimed through the STN and GPi respectively. The neural activity was recorded via a tetrode (Thomas Recording, Giessen, Germany) which encompassed four platinum-tungsten fibers in a glass coat with one contact each at the end. Three of the tetrode contacts were circularly arranged around a central contact at the tip of the electrode. The distance between the outer contacts was 29 μm and the distance between the central contact and the outer contacts was 5 μm. Impedance of the contacts during recording ranged between 500 and 1500 kΩ and amplification was in the order of 4000 - 20000 times. Recordings were performed in an exploratory manner for 60 seconds every millimeter starting from approximately 15 millimeters above the calculated target point, i.e. the antero-ventral border of the STN or the medio-ventral border of the

GPi right above the optic tract, respectively. High-frequency spiking activity was visualized online on screen and was captured on hard disk. The MUA signal was bandpassed using a hardware filter from 500 Hz to 10 kHz. Local Field Potentials were recorded from one of the four tetrode contacts referenced through a common ground and lowpassed using a hardware filter at 141 Hz while extracellular signals were simultaneously recorded from all 4 tetrode contacts. The recordings for the goniometer paradigm were started in proximity of the calculated target point when one or more units were clearly distinguishable from background activity (i.e. excellent signal-to-noise ratio) based on online visual inspection and when the peak to peak amplitude of spiking activity was stable over approximately 30 seconds. For patients 1 and 4, the intended target point was the GPi and for patients 2 and 3 the STN. All signals were sampled at 25 kHz and stored on hard disk for offline analysis.

4.2.3 Data Processing and Feature Extraction

In this section we describe the steps taken to process the recorded signals for further analysis. Custom software developed in Python and Matlab was used to perform the analysis. For this experiment we are particularly interested in movement related neural signals and accordingly cutouts were taken around the movement times and the rest of the recordings were discarded. After the initial processing described below, all signals were down-sampled to 1000Hz for ease of analysis.

Movement (Goniometer)

The voltage trace from the goniometer is used as a proxy for the patients hand position and movement start and end times were defined by visually inspecting the trace of each patient (see figure 4.1). Due to the self paced nature of the experiments along with the partially ongoing effect of anesthetics used during surgery, movements varied widely in length, extension and speed both between single trials of a given patient and between the patients themselves (see figure 4.2). Velocity and acceleration of the movements were calculated from the raw signal using a first and second order Savitzky Golay filter

(total width of 501 data points) respectively from the FIND matlab toolkit [129, 130]. After identifying the start and end times of each movement, cutouts of the three signal types (movement, velocity and acceleration) were made with a buffer of 1 second on each side and baselined by subtracting the mean of the 1 second interval leading up to movement onset. For each of the signals we then calculated their average over trials when aligned by movement onset, minima (the turnaround point of the movement) and offset. As each of the trials were different lengths, the average signal was only computed for time points where at least 60% of the trials were ongoing. Additional features were also calculated including the length in seconds of each movement, the maximum velocity and acceleration reached during each movement and the maximum extent of the wrist in each movement.

Electrophysiology

Extracellular Signals For each patient, all 4 extracellular signals were processed. Cutouts were made around movement onset and offset as described in section 4.2.3. From the extracellular recording we derive two further measures of neural activity, Multi Unit Activity (MUA) and Background Unit Activity (BUA). The MUA is generated by the action potentials of nearby neurons within the range of up to 300 μm. Beyond this range the MUA is not distinguishable from the background activity. MUA was calculated by taking the absolute of the Hilbert transform [131] of the extracellular signal and then smoothing it using a causal exponential filter with a sigma of 50 ms [130]. This type of extracellular signal captures the bulk spiking activity of many neurons within the recording volume and has a long tradition [132, 133]. The MUA has previously been shown to carry movement-specific information in the monkey motor cortex [134].

The BUA is derived from the MUA signal and reflects sub-noise level spiking activity from the same local radius as the MUA. The BUA was calculated as described in [135]. The spike times used in the BUA calculation were found by thresholding the extracellular trace at its mean plus two times the standard deviation. Each upward threshold crossing was considered a spike. The MUA and BUA were baselined and

onset, minima and offset aligned averages were calculated as described in section 4.2.3. For an example of the resulting signals, see figure 4.3.

Local Field Potential The Local Field Potential (LFP) is mainly generated through the summation of dendritic currents of excitatory and inhibitory synapses of local neuronal populations within the range of up to 2-3 millimeters. A spectrogram of the raw LFP was calculated using the specgram function in matplotlib [136] with a window of 400 ms that was shifted by 1ms for each calculation. Again, cutouts were made around movement onset and offset as described in section 4.2.3. The spectrogram was then divided into seven bands (delta 1-4Hz, theta 4-8Hz, alpha 8-14Hz, beta 14-30Hz, gamma 30-70Hz, high gamma 70-130Hz and very high gamma 130-200Hz , other frequencies were discarded), normalised to zero mean and unit variance, and the average signal power of each band was computed over time. The LFP bands were then baselined and onset, minima and offset aligned averages were calculated as descried in section 4.2.3. For an example of the resulting LFP signals, see figure 4.3. It should be noted that the band range of the very high gamma signal overlaps with the hardware based lowpass filter of 141Hz. However we still find meaningful signal in this range during analysis (likely due to the slow attenuation of such filters) and as such do not discard the frequencies between 141Hz and 200Hz.

Additionally, for each of the MUA, BUA and LFP band signals, the average response in a 100 ms window around movement onset, minima and offset was calculated.

4.2.4 Feature Correlation Analysis

Correlation between features of the movement and neural based signals were calculated over trials using Pearson's Correlation. Correlations were also calculated between the averages of each of the neural signals and the movement signals over time. Correlations with a p-value > 0.05 were considered insignificant and set to 0. Results of the correlation analysis can be seen in figures 4.4 and 4.5 and interpretation of the results

is found in section 4.3.1.

4.2.5 Decoding Movement

Decoding of the neural signals to reconstruct each patient's movements on a trial by trial basis was achieved using the scikit-learn implementation of Linear Regression [137]. Both movement and neural signals were downsampled to 100Hz using the decimate function from the scipy package for python [138]. For each movement sample to be predicted, the Linear Regression model received the last 100 samples (1 second) of neural response directly preceding the prediction point. The window size of 100 samples was chosen as the best performer after testing of a number of possible values. Movement responses were predicted using *a leave one trial out* methodology where the classifier is trained on all trials except the one to be predicted. Prediction performance was then quantified by measuring the Pearson's correlation coefficient between the real and predict movement data on a trial by trial basis. The correlation over trials were averaged after using Fisher's z transform as described in [139] to reach a single correlation value representing decoding performance.

4.3 Results

In each patient we first perform a correlation analysis to quantify the relation of different movement parameters and the neural activity. In a second approach we use different frequency bands of the LFP and the MUA and BUA signals to predict the wrist position in single movement trials.

4.3.1 Neural representation of kinematic parameters

Patients were instructed to perform wrist movements (flexion) in a self-paced manner. This could result in highly variable movements as for patient 1 or in highly stereotypical trajectories as for patient 2 (see figure 4.2). Average movement durations were in the

range of 2-4s. Figure 4.3 shows a single trial movement trajectory as recorded with the goniometer (bottom) together with the corresponding neural activity traces for patients 1 and 3. The power evolution of the LFP was measured in seven separate frequency bands (green traces in figure 4.3; see Methods) and population spiking activity was estimated in the MUA and BUA signals (blue traces). Relations between neural signals and movement are difficult to assess on a single trial basis. We therefore performed correlation analysis between kinematic movement parameters and neural activity in two ways.

First, we performed correlations of the trial-averaged time-resolved movement parameters (position, velocity, acceleration) with the trial-averaged time-resolved neural activity traces. Figure 4.4, A and B show examples for the dynamic changes in amplitude and velocity compared to the time-resolved MUA amplitude. We computed the linear correlation coefficient between all seven LFP bands, MUA and BUA and the three movement parameters and repeated this for different temporal alignments of trials (figure 4.4). It shows that high correlations exist between one neural signal type and different movement parameters, indicating a mixed representation of different parameters in the neuronal population signals. At the same time, one movement parameter could show high correlation with different signal types.

Whilst each patient shows a strong correlation to at least one of the signal pairs, they do not share a common pattern as to which signals are correlated and which are not. Subplots D-F have movement parameters plotted against neural response across trials. Again we see that each patient correlates strongly with at least one signal pair but that there is no commonality across patients. Figure 4.5 shows a heat map of all assessed movement/neural combinations for each patient. A large number of signal pairs show a non-zero correlation, however it also becomes clear that each patient is unique as to which pairs have a relationship. We can however conclude that properties of the movement trajectory including position, velocity and acceleration along with

4.3.2 Movement prediction from neural signals

In order to predict the patients movement on the basis of the neural activity we used a linear regression algorithm. We use all but one trial for training the linear regressor and tested the prediction on the unused trial. Performance was computed as the Pearson's correlation coefficient of the true versus the decoded signal. The procedure was repeated for each trial once (see chapter 4.2.5). This allowed us to find the best collection of neural signals to use as features by assessing the performance of all possible signal combinations separately. The results are provided in Table 4.2. For one patient we reached a correlation between actual and predicted trajectory as high as 0.86, while in case of patient 2 the best prediction yielded an average correlation of only 0.27. Actual and predicted time course of wrist extension amplitude are shown in figure 4.6 for the two patients that yielded the best decoding performance. The predictions closely follow the actual movements suggesting that the signals recorded with the electrode are sucient for accurately reconstructing this type of hand movement.

Figure 4.7 shows the percentage of the maximum decoding performance achieved as the number of features available to the regressor grows. All patients except #4 reach peak performance at 3 features. Again we find little consistency across patients as to which features perform best (see table 4.2). We do however see that the bulk of the decoding performance for each patient is won from the best single feature with only marginal gain achieved by adding more. The predictions closely follow the actual movements suggesting that the signals recorded with the electrode are sucient for accurately reconstructing this type of hand movement.

4.4 Discussion and Future Work

In this work we have shown a strong representation of movement in the basal ganglia, in line with many previously published studies [128, 140–142]. We find significant correlations between all of the neural signals assessed and parameters of movement such as position, velocity and acceleration. However, we do not find any unifying

Table 4.2: **Best Features for Decoding.** This table shows the best single and multi-feature combinations for decoding movement in each patient. The performance metric used is trial-by-trial correlation. For all patients, the bulk of the decoding performance is won with the first feature and only marginal increases are achieved by adding more.

Patient	Single Feature	Performance	Multiple Features	Performance
1	Bua	0.48	Bua, Beta, Gamma	0.61
2	Theta	0.21	Theta, Beta, Alpha	0.27
3	Mua	0.84	Mua, Very High Gamma, Gamma	0.86
4	Mua	0.64	Mua	0.64

trends in this relationship across patients, that is, what correlates well for one patient does not necessarily do so for another. This could be due to a number of factors. First, the sample size of our cohort was only four and only two for each target point. Second, one has to assume that at a microscopic level recordings were made in different locations for all patients. Furthermore the variability of neuronal discharge in the pallidum or in the subthalamic nucleus in response to active or passive limb movements was also observed in previous studies [eg. 142, 143]. This is a by-product of the clinical nature of the experiments, where the primary goal is not controlled scientific research but to help get the patient well. Additionally, the free movement nature of the experiment has resulted in some patients having very rhythmic and homogeneous movements whilst others are sporadic and heterogeneous. This makes it again more difficult to compare across patients as across-trial averages can be more or less reliable, depending on the regularity of the movements, while other parameters such as movement length become redundant if each movement takes the same amount of time. A compounding factor is the strong relationship between the movement parameters themselves. Velocity, Acceleration, movement length and maximum extension are all results of the hand position itself tracked over time, making it near impossible to investigate these factors independently of each other.

Nonetheless, we have shown that it is possible to accurately decode a patients hand position using neural recordings from the basal ganglia and a linear regressor. This would suggest that not only is *some* movement represented in the basal ganglia, but

that it is represented at a fidelity high enough for full movement reconstruction. We also find that no single neural signal *best* represents this type of movement, but that the best signal is different across patients or brain region. Though in the three patients with reasonable decoding performance, the electrophysiological signals (MUA and BUA) explained more of the movement parameter than those from the LFP. Decoding the neural activity of voluntary movements can serve a lot of clinical applications, e.g. in the area of brain-computer interfaces [144] where brain signals have to be translated reliably into commands that operate a prosthetic arm. We hope that this work can serve as a guide for the design of future experiments.

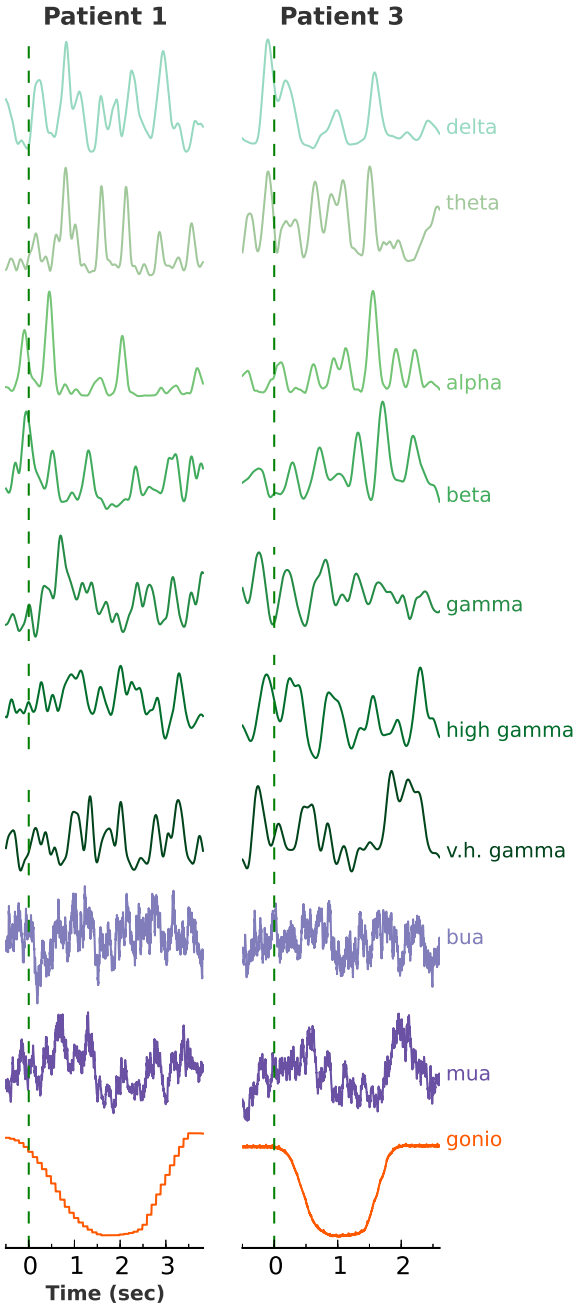

Figure 4.3: **Neural signals and movement trajectory.** This figure shows the 1 movement (red) and 9 neural (7 LFP band-limited power in green and 2 extracellular signals in blue) signals for a single trial in two patients. Movement onset is marked by the dashed line

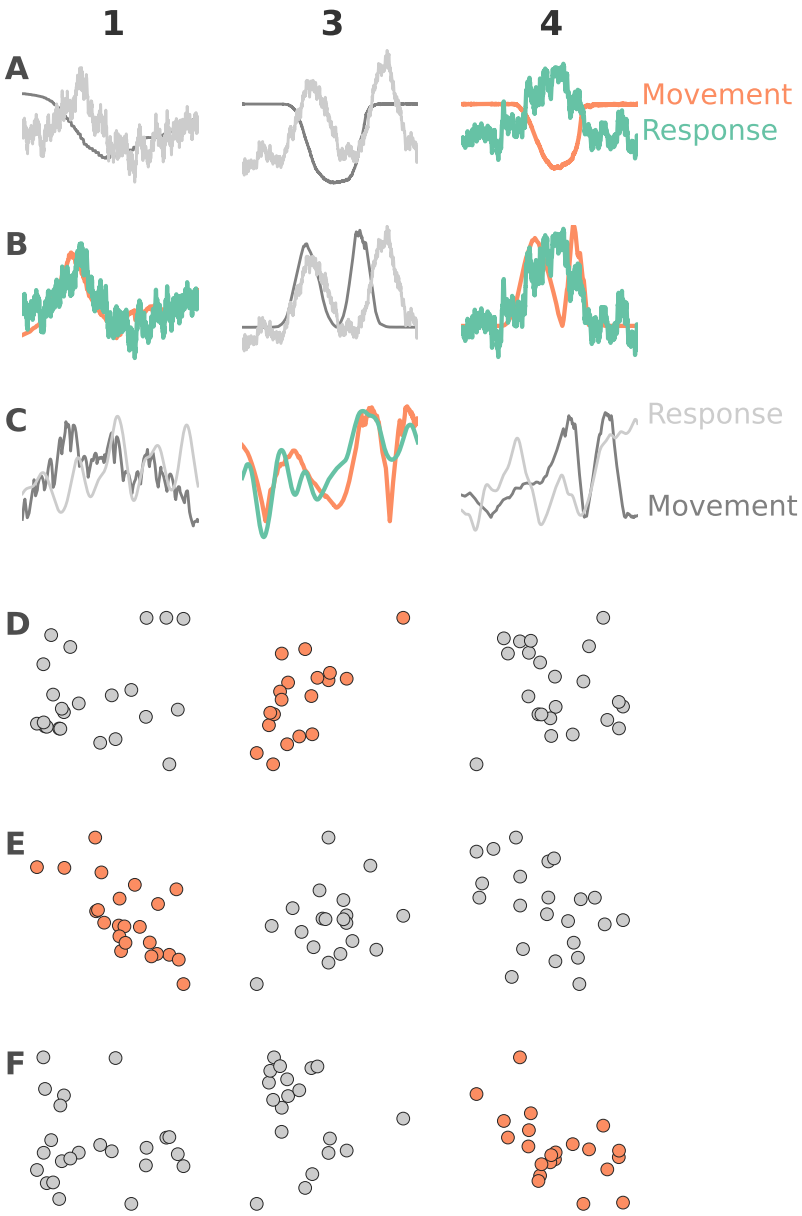

Figure 4.4: **Relationship between neural signals and kinematic movement parameters.** For 3 Patients, (A-C) Plot Movement Features against Neural Signal (Trial Averaged) with Time on the x-axis and signal on the y-axis, (D-F) Plot Movement parameters against Neural Signal over Trials. A) Movement plotted against MUA (onset aligned). B) Absolute Velocity plotted against MUA (Onset Aligned). C) Absolute Acceleration plotted against Alpha (aligned to movement minima) D) Velocity at Trial Offset (x-axis) plotted against MUA rate at offset (y-axis). E) Extent at Movement Minima (x-axis) plotted against Very High Gamma rate at Minima (y-axis) F) Offset Velocity (x-axis) plotted against Gamma (y-axis) at movement minima. Where a significant correlation above 0.5 exists, plots are shown in colour.

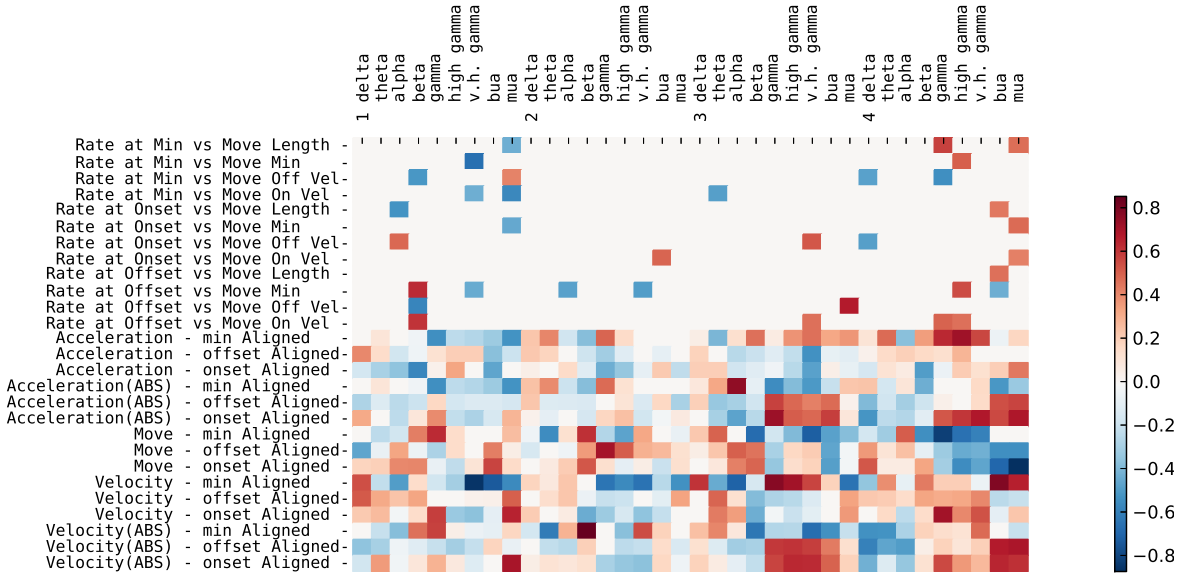

Figure 4.5: **Correlation comparison between neural signals and kinematic movement parameters.** This figure compares the significant correlation (p ¡ 0.05) of different signal and movement features across patients. The correlation value for each pair is represented by the heat map and non-significant correlations are set to zero value. Whilst many signal combinations result in a non-zero correlation, there is very little consistency across patients as to which combinations correlate most.

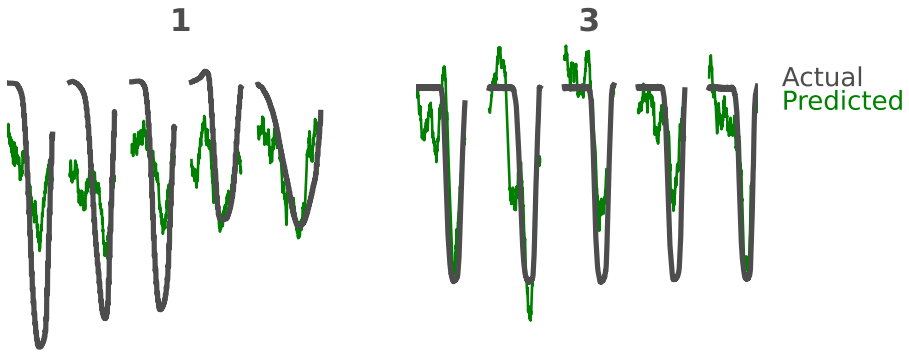

Figure 4.6: **Decoding Accuracy.** Actual and decoded trajectory for 2 patients. The correlation between the actual and decoded trajectories are 0.61 and 0.86 for patients 1 and 3 respectively.

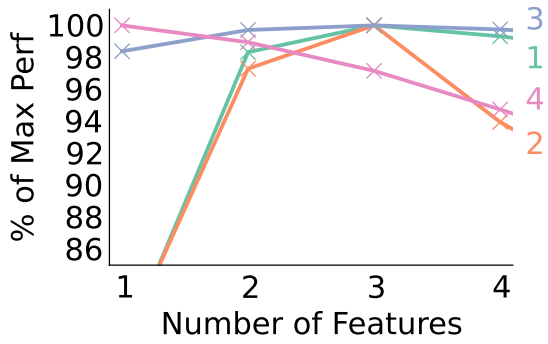

Figure 4.7: **Movement Decoding for multiple features.** Change in prediction power as a percentage of the best overall prediction. The number of neural signals used increases from left to right. The maximum for all but one patient is reached at $n = 3$

5

General Discussion

In the manuscripts above we have worked in areas intersecting Data Science and Neuroscience.

The work in chapter 2 is focused on machine learning research and specifically, the brain inspired framework of Artificial Neural Networks. We introduced a new training method for temporal RBMs called *Temporal Autoencoding* and showed that our pre-training method achieves a significant performance increase in both generative and predictive tasks across multiple datasets. The approach is robust and proves fruitful for both Temporal and Conditional RBMs, advancing the state-of.the-art in temporal RBM training. In our opinion, TA training allows the networks to learn a more meaningful temporal representation of the data than is possible through contrastive divergence alone. We believe the inclusion of Autoencoder training in temporal learning tasks will be beneficial in a number of contexts, as it enforces the causal structure

of the data on the learned model

In chapter 3 we extend our work with *Temporal Autoencoding* and apply it to modelling the process of dynamic representation learning in the mammalian visual cortex. Using *Temporal Autoencoding* we show that TRBMs are able to learn dynamic representations of natural image sequences that capture both important statistics of the dataset and echo the receptive field properties found experimentally in V1 neurons. In addition, we show that the learned features provide for a temporally sparse code, an attribute desirable in systems with finite metabolic resources such as the brain.

Chapter 4 moves away for from the modelling approach and uses the brain as a data source to investigate the relationship of wrist movement and neural activity in the basal ganglia. We are able to show not only that strong correlations exist between movement parameters and activity in the subthalamic nucleus and the globus pallidus interna, but that the neural signals from these regions can be used to reconstruct the patients wrist trajectory via the use of a linear regressor.

5.1 Neuroscience and Machine Learning

These manuscripts together allude to the importance of machine learning techniques in neuroscience and paint a picture of the two part relationship between these fields. On the one hand, machine learning is heavily utelised by neuroscientists as a tool in the analysis of experimental data [145–147], for decoding of neural signals in brain machine interfaces [148–152] and even to peer into our thoughts [153, 154], to name but a few applications.

It is perhaps the second meeting point between these fields that is most interesting. The use of machine learning models as a starting point for understanding brain functionality, and the application of our understanding of neural function as a source for new and improved machine learning algorithms. This back and forth of information and ideas has been seen in areas such as ANNs, their neuromorphic counterparts and slow feature analysis (SFA) amongst others. Of the many forms of ANNs studied, their uniting factor is that they all draw the concept for their basic computational unit, the

artificial neuron, from the brain [9, 155, 156]. This approach to learning, an idea of massively parallel networks of computational units, has spawned a huge focus of research with many applications. An extension of this field has been the investigation of hardware based ANN implementations known as neuromorphic hardware [105, 157] . Neuromorphic hardware has the advantage of a truly parallel implementation in hardware (where as software based ANNs are still bound to the pipeline bottlenecks of the Von Neumann architecture) allowing computations to run at speeds similar to, if not faster than biology [158]. Additionally, neuromorphic chips are able to implement more biologically realistic spiking neuron models such as integrate-and-fire [159] and Hodgkin-Huxley [160], the computational burden of which usually limits the size and/or speed of such networks implemented in software. This has opened up new opportunities to investigate computational networks inspired by biologically, taking inspiration from the honeybee olfactory system [161–163] or the human retina [164–166].

In return, ideas such as the Hopfield network [167], a form of recurrent ANN designed to learn collections of patterns, have provided a possible framework for associative memory in the brain. SFA [168], a form of unsupervised machine learning that extracts slowly varying features from a more quickly varying signal, has been shown to account for the self organisation of complex-cell receptive fields in visual cortex ([169]; closely related to our work in chapter 3) and the formation of place cells in the hippocampus [170]. RBMs have even been suggested as a generalised framework for cortical representation learning across multiple sensory systems [171]. The combined work presented in this thesis also addresses this two part relationship, with chapter 4 relying on machine learning techniques to decode patients hand movements whilst chapters 2 and 3 take inspiration from the brain and utelise state-of-the-art machine learning techniques to suggest how natural image sequences could be encoded using a neural structure.

5.2 A Data Scientific Approach to Neuroscience

The unifying theme across this work is Neuroscience, but it is also a data scientific approach to research, be it the through development of new machine learning techniques (chapter 2), modelling representation learning from our natural environment (chapter 3) or the analysis of neural and behavioural data in tandem (chapter 4). The individual challenges that arose in each of these projects required the application of cutting edge computational techniques, drawn from the toolkit of the Data Scientist. These techniques allowed us to arrange and interrogate gigabytes of noisy and multivariate temporal data and to create automated analysis work flows, allowing for fast assessment of multiple dataset and parameter combinations. It allowed us to investigate the data in an interactive manner, and learn directly from it with Machine Learning techniques.

5.2.1 Python for Data Science

In particular, this thesis is a testament to the versatility and maturity of the Python programming language and it's framework for scientific data analysis. The results presented above would not have been possible if not for the fantastic array of open source tools that cover every aspect of the Data Science pipeline. For *Data acquisition and Processing*: numpy [172], pandas [173], for *Statistical Analysis*: scipy [138], for *Modelling and Machine Learning*: theano [34], scikits-learn [137], ipython parallel [174], scikitCVcluster [175], and for *Data Visualisation and Story Telling*: matplotlib [136] and prettyplotlib [176]. Python's strength as a go-to language for scientific computing is becoming more widely acknowledged [177] which will hopefully in turn grow the community that contributes to it's great tools.

5.3 Outlook

As the complexity of experiments and the volume of data they produce increases, the field of neuroscience, along with other biological sciences, will become more and more

reliant on those trained in Data Scientific techniques to help make sense of their results. It has been said that Data Science will be the 'sexiest job of the 21st century' [178]. Such statements are of course pure hype, but the core disciplines of data science are already reflected in what many scientists in any number of fields do on a day to day basis, and as the data grows, so will the demand for their skill set.

References

[1] U. o. E. Centre for Doctoral Training in Data Science. *What is data science?* (2014). URL http://datascience.inf.ed.ac.uk/what-is-data-science/. 2

[2] *Gatsby computational neuroscience unit* (2014). URL http://www.gatsby.ucl.ac.uk/. 2

[3] *Bernstein network computational neuroscience* (2014). URL http://www.nncn.de/en. 2

[4] *Journal of computational neuroscience* (2014). URL http://www.springer.com/biomed/neuroscience/journal/10827. 2

[5] *frontiers in computational neuroscience* (2014). URL http://www.frontiersin.org/computational_neuroscience. 2

[6] *Computational and systems neuroscience (cosyne)* (2014). URL http://www.cosyne.org/. 2

[7] *Computational neuroscience meeting* (2014). URL http://www.cnsorg.org/.

[8] *Bernstein conference* (2014). URL http://www.bernstein-conference.de/. 2

[9] Y. Chauvin and D. E. Rumelhart. *Backpropagation: theory, architectures, and applications* (Psychology Press, 1995). 3, 85

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning representations by back-propagating errors.* Nature **323**(6088), 533 (1986). 3

[11] Y. LeCun, J. S. Denker, S. A. Solla, R. E. Howard, and L. D. Jackel. *Optimal brain damage.* In *NIPs*, vol. 2, pp. 598–605 (1989).

[12] Y. LeCun and Y. Bengio. *Convolutional networks for images, speech, and time series.* The handbook of brain theory and neural networks **3361** (1995).

[13] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. *Backpropagation applied to handwritten zip code recognition.* Neural computation **1**(4), 541 (1989). 3

[14] C. Cortes and V. Vapnik. *Support-vector networks.* Machine learning **20**(3), 273 (1995). 3

[15] G. Hinton and R. Salakhutdinov. *Reducing the dimensionality of data with neural networks.* Science **313**(5786), 504 (2006). 3, 11, 15, 33, 51

[16] A. Mohamed, T. Sainath, G. Dahl, B. Ramabhadran, G. Hinton, and M. Picheny. *Deep belief networks using discriminative features for phone recognition.* In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 5060–5063 (IEEE, 2011). 4, 11, 33

[17] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol. *Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.* The Journal of Machine Learning Research **11**, 3371 (2010). 4, 11, 15, 18, 33, 51, 54

[18] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng. *Building high-level features using large scale unsupervised learning.* In *International Conference in Machine Learning* (2012). 4, 11

[19] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. *Why does unsupervised pre-training help deep learning?* The Journal of Machine Learning Research **11**, 625 (2010). 4

[20] H. Lee, C. Ekanadham, and A. Y. Ng. *Sparse deep belief net model for visual area v2.* In *NIPS*, vol. 7, pp. 873–880 (2007).

[21] N. Srivastava and R. Salakhutdinov. *Multimodal learning with deep boltzmann machines.* In *NIPS*, pp. 2231–2239 (2012).

[22] S. M. Plis, D. R. Hjelm, R. Salakhutdinov, and V. D. Calhoun. *Deep learning for neuroimaging: a validation study.* arXiv preprint arXiv:1312.5847 (2013).

[23] Y. Bengio. *Deep learning of representations: Looking forward.* In *Statistical Language and Speech Processing*, pp. 1–37 (Springer, 2013).

[24] M. Längkvist, L. Karlsson, and A. Loutfi. *A review of unsupervised feature learning and deep learning for time-series modeling.* Pattern Recognition Letters (2014).

[25] M. Denil, B. Shakibi, L. Dinh, N. de Freitas, *et al. Predicting parameters in deep learning.* In *Advances in Neural Information Processing Systems*, pp. 2148–2156 (2013). 4

[26] M. Carreira-Perpinan and G. Hinton. *On contrastive divergence learning.* In *Artificial Intelligence and Statistics*, vol. 2005, p. 17 (2005). 11, 14, 50

[27] I. Sutskever and G. Hinton. *Learning multilevel distributed representations for high-dimensional sequences.* In *Proceeding of the Eleventh International Conference on Artificial Intelligence and Statistics*, pp. 544–551 (2007). 11, 16, 34, 52, 53

[28] G. Taylor, G. Hinton, and S. Roweis. *Modeling human motion using binary latent variables.* Advances in neural information processing systems **19**, 1345 (2007). 11, 12, 17, 21, 24, 26, 34, 52, 58, 61

[29] C. Häusler, A. Susemihl, and M. P. Nawrot. *Natural image sequences constrain dynamic receptive fields and imply a sparse code.* Brain research **1536**, 53 (2013). 12, 24

[30] S. Makridakis and M. Hibon. *The m3-competition: results, conclusions and implications*. International journal of forecasting **16**(4), 451 (2000). 12, 21, 24

[31] Y. Bengio, L. Yao, G. Alain, and P. Vincent. *Generalized Denoising Auto-Encoders as Generative Models*. ArXiv e-prints (2013). 1305.6663. 12

[32] I. Sutskever, G. Hinton, and G. Taylor. *The recurrent temporal restricted boltzmann machine*. Advances in Neural Information Processing Systems **21** (2008). 16, 22, 52, 55, 59

[33] G. Taylor. *Composable, distributed-state models for high-dimensional time series*. Ph.D. thesis (2009). 17, 53

[34] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. *Theano: a CPU and GPU math expression compiler*. In *Proceedings of the Python for Scientific Computing Conference (SciPy)* (2010). Oral Presentation, URL http://www.iro.umontreal.ca/~lisa/pointeurs/theano_scipy2010.pdf. 20, 22, 58, 86

[35] G. Hinton. *A practical guide to training restricted boltzmann machines*. Momentum **9**, 1 (2010). 20, 56

[36] K. Cho, A. Ilin, and T. Raiko. *Improved learning of gaussian-bernoulli restricted boltzmann machines*. In *Artificial Neural Networks and Machine Learning–ICANN 2011*, pp. 10–17 (Springer, 2011). 20

[37] A. Senior, G. Heigold, M. Ranzato, and K. Yang. *An empirical study of learning rates in deep neural networks for speech recognition*. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 6724–6728 (IEEE, 2013). 20

[38] P. Reinagel. *How do visual neurons respond in the real world?* Current opinion in Neurobiology **11**(4), 437 (2001). 33

[39] E. Simoncelli and B. Olshausen. *Natural image statistics and neural representation.* Annual review of neuroscience **24**(1), 1193 (2001). 33

[40] B. A. Olshausen, D. J. Field, *et al. Sparse coding of sensory inputs.* Current opinion in neurobiology **14**(4), 481 (2004). 33, 46

[41] S. Jadhav, J. Wolfe, and D. Feldman. *Sparse temporal coding of elementary tactile features during active whisker sensation.* Nature neuroscience **12**(6), 792 (2009). 33, 49

[42] T. Hromádka, M. DeWeese, and A. Zador. *Sparse representation of sounds in the unanesthetized auditory cortex.* PLoS biology **6**(1), e16 (2008). 33

[43] Y. Dan, J. Atick, and R. Reid. *Efficient coding of natural scenes in the lateral geniculate nucleus: experimental test of a computational theory.* The Journal of Neuroscience **16**(10), 3351 (1996). 33

[44] W. Vinje and J. Gallant. *Sparse coding and decorrelation in primary visual cortex during natural vision.* Science **287**(5456), 1273 (2000). 46

[45] P. Reinagel and R. Reid. *Precise firing events are conserved across neurons.* The Journal of neuroscience **22**(16), 6837 (2002).

[46] S. Yen, J. Baker, and C. Gray. *Heterogeneity in the responses of adjacent neurons to natural stimuli in cat striate cortex.* Journal of neurophysiology **97**(2), 1326 (2007). 47

[47] P. Maldonado, C. Babul, W. Singer, E. Rodriguez, D. Berger, and S. Grün. *Synchronization of neuronal responses in primary visual cortex of monkeys viewing natural images.* Journal of Neurophysiology **100**(3), 1523 (2008). 49

[48] B. Haider, M. R. Krause, A. Duque, Y. Yu, J. Touryan, J. A. Mazer, and D. A. McCormick. *Synaptic and network mechanisms of sparse and reliable visual cortical activity during nonclassical receptive field stimulation.* Neuron **65**(1), 107 (2010). 33, 47, 57

[49] K. A. Martin and S. Schröder. *Functional heterogeneity in neighboring neurons of cat primary visual cortex in response to both artificial and natural stimuli.* The Journal of Neuroscience **33**(17), 7325 (2013). 33, 47

[50] C. Chen, H. Read, and M. Escabí. *Precise feature based time scales and frequency decorrelation lead to a sparse auditory code.* The Journal of Neuroscience **32**(25), 8454 (2012). 33

[51] N. L. Carlson, V. L. Ming, and M. R. DeWeese. *Sparse codes for speech predict spectrotemporal receptive fields in the inferior colliculus.* PLoS Computational Biology **8**(7), e1002594 (2012). 33

[52] J. Wolfe, A. Houweling, M. Brecht, *et al. Sparse and powerful cortical spikes.* Current opinion in neurobiology **20**(3), 306 (2010). 33

[53] R. Herikstad, J. Baker, J.-P. Lachaux, C. M. Gray, and S.-C. Yen. *Natural movies evoke spike trains with low spike time variability in cat primary visual cortex.* The Journal of Neuroscience **31**(44), 15844 (2011). 33, 41, 47

[54] B. Olshausen *et al. Emergence of simple-cell receptive field properties by learning a sparse code for natural images.* Nature **381**(6583), 607 (1996). 33

[55] J. H. van Hateren and D. L. Ruderman. *Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex.* Proceedings of the Royal Society of London. Series B: Biological Sciences **265**(1412), 2315 (1998). 33

[56] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. *Improving neural networks by preventing co-adaptation of feature detectors.* Arxiv preprint arXiv:1207.0580 (2012). 33

[57] A. Saxe, M. Bhand, R. Mudur, B. Suresh, and A. Ng. *Unsupervised learning models of primary cortical receptive fields and receptive field plasticity.* Advances in neural information processing systems (2011). 33

[58] H. Lee, C. Ekanadham, and A. Ng. *Sparse deep belief net model for visual area v2.* Advances in neural information processing systems **20**, 873 (2008).

[59] H. Lee, Y. Largman, P. Pham, and A. Ng. *Unsupervised feature learning for audio classification using convolutional deep belief networks.* Advances in neural information processing systems **22**, 1096 (2009). 33

[60] M. Marszalek, I. Laptev, and C. Schmid. *Actions in Context.* In *IEEE Conference on Computer Vision & Pattern Recognition* (2009). 36

[61] A. J. Bell and T. J. Sejnowski. *The independent components of natural scenes are edge filters.* Vision Research **37**(23), 3327 (1997). URL http://www.sciencedirect.com/science/article/pii/S0042698997001211. 37, 38, 45

[62] C. Cadieu and B. Olshausen. *Learning intermediate-level representations of form and motion from natural movies.* Neural Computation pp. 1–40 (2012). 38, 54

[63] G. Wang, S. Ding, and K. Yunokuchi. *Difference in the representation of cardinal and oblique contours in cat visual cortex.* Neuroscience letters **338**(1), 77 (2003). 38, 45

[64] D. Coppola, L. White, D. Fitzpatrick, and D. Purves. *Unequal representation of cardinal and oblique contours in ferret visual cortex.* Proceedings of the National Academy of Sciences **95**(5), 2621 (1998). 38, 45

[65] W. Bosking, Y. Zhang, B. Schofield, and D. Fitzpatrick. *Orientation selectivity and the arrangement of horizontal connections in tree shrew striate cortex.* The Journal of Neuroscience **17**(6), 2112 (1997). 39, 45

[66] D. Field and A. Hayes. *Contour integration and the lateral connections of v1 neurons.* The visual neurosciences **2**, 1069 (2004). 39, 45

[67] B. Willmore and D. J. Tolhurst. *Characterizing the sparseness of neural codes.* Network: Computation in Neural Systems **12**(3), 255 (2001). 40, 56

[68] C. F. Cadieu and B. A. Olshausen. *Learning Transformational Invariants from Natural Movies.* In *Advances in Neural Information Processing Systems 21*, pp. 1–8 (2008). 45, 53

[69] S. Van Hooser. *Similarity and diversity in visual cortex: is there a unifying theory of cortical computation?* The Neuroscientist **13**(6), 639 (2007). 45

[70] C. Zetzsche and U. Nuding. *Nonlinear and higher-order approaches to the encoding of natural scenes.* Network: Computation in Neural Systems **16**(2-3), 191 (2005). 46

[71] S. J. Caron, V. Ruta, L. Abbott, and R. Axel. *Random convergence of olfactory inputs in the drosophila mushroom body.* Nature (2013). 46

[72] R. A. Jortner, S. S. Farivar, and G. Laurent. *A simple connectivity scheme for sparse coding in an olfactory system.* The Journal of neuroscience **27**(7), 1659 (2007). 46

[73] R. Huerta and T. Nowotny. *Fast and robust learning by reinforcement signals: explorations in the insect brain.* Neural computation **21**(8), 2123 (2009). 46, 49

[74] C. Poo and J. S. Isaacson. *Odor representations in olfactory cortex:sparse coding, global inhibition and oscillations.* Neuron **62**(6), 850 (2009). 47

[75] D. S. Reich, F. Mechler, and J. D. Victor. *Independent and redundant information in nearby cortical neurons.* Science **294**(5551), 2566 (2001). 47

[76] I. Ito, R. C.-y. Ong, B. Raman, and M. Stopfer. *Sparse odor representation and olfactory learning.* Nature neuroscience **11**(10), 1177 (2008). 47

[77] J. Benda and A. V. Herz. *A universal model for spike-frequency adaptation.* Neural computation **15**(11), 2523 (2003). 47

[78] F. Farkhooi, A. Froese, E. Muller, R. Menzel, and M. P. Nawrot. *Cellular adaptation accounts for the sparse and reliable sensory stimulus representation.* arXiv:1210.7165 (2012). 47

[79] M. P. Nawrot. *Dynamics of sensory processing in the dual olfactory pathway of the honeybee.* Apidologie **43**(3), 269 (2012). 47

[80] M. J. Chacron, A. Longtin, and L. Maler. *Negative interspike interval correlations increase the neuronal capacity for encoding time-dependent stimuli.* The Journal of Neuroscience **21**(14), 5328 (2001). 47

[81] M. P. Nawrot, C. Boucsein, V. Rodriguez-Molina, A. Aertsen, S. Grün, and S. Rotter. *Serial interval statistics of spontaneous activity in cortical neurons¡ i¿ in vivo¡/i¿ and¡ i¿ in vitro¡/i¿.* Neurocomputing **70**(10), 1717 (2007).

[82] F. Farkhooi, M. F. Strube-Bloss, and M. P. Nawrot. *Serial correlation in neural spike trains: Experimental evidence, stochastic modeling, and single neuron variability.* Physical Review E **79**(2), 021905 (2009).

[83] M. P. Nawrot. *Analysis and interpretation of interval and count variability in neural spike trains.* In *Analysis of parallel spike trains*, pp. 37–58 (Springer, 2010). 47

[84] M. J. Chacron, L. Maler, and J. Bastian. *Electroreceptor neuron dynamics shape information transmission.* Nature neuroscience **8**(5), 673 (2005). 47

[85] F. Farkhooi, E. Muller, and M. P. Nawrot. *Adaptation reduces variability of the neuronal population code.* Physical Review E **83**(5), 050905 (2011). 47, 58

[86] C. Assisi, M. Stopfer, G. Laurent, and M. Bazhenov. *Adaptive regulation of sparseness by feedforward inhibition.* Nature neuroscience **10**(9), 1176 (2007). 47

[87] M. Papadopoulou, S. Cassenaer, T. Nowotny, and G. Laurent. *Normalization for sparse encoding of odors by a wide-field interneuron.* Science **332**(6030), 721 (2011). 47

[88] R. W. Friedrich and G. Laurent. *Dynamics of olfactory bulb input and output activity during odor stimulation in zebrafish.* Journal of neurophysiology **91**(6), 2658 (2004). 48

[89] R. I. Wilson, G. C. Turner, and G. Laurent. *Transformation of olfactory representations in the drosophila antennal lobe.* Science Signaling **303**(5656), 366 (2004).

[90] S. Krofczik, R. Menzel, and M. P. Nawrot. *Rapid odor processing in the honeybee antennal lobe network.* Frontiers in computational neuroscience **2** (2008).

[91] M. F. Brill, T. Rosenbaum, I. Reus, C. J. Kleineidam, M. P. Nawrot, and W. Rössler. *Parallel processing via a dual olfactory pathway in the honeybee.* The Journal of Neuroscience **33**(6), 2443 (2013). 48

[92] A. P. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey. *On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex.* The Journal of Neuroscience **2**(11), 1527 (1982). 48

[93] J. Rickert, A. Riehle, A. Aertsen, S. Rotter, and M. P. Nawrot. *Dynamic encoding of movement direction in motor cortical neurons.* The Journal of Neuroscience **29**(44), 13870 (2009). 48

[94] W. B. Levy and R. A. Baxter. *Energy efficient neural codes.* Neural Computation **8**(3), 531 (1996). 48

[95] S. B. Laughlin *et al. Energy as a constraint on the coding and processing of sensory information.* Current opinion in neurobiology **11**(4), 475 (2001).

[96] P. Lennie. *The cost of cortical computation.* Current biology **13**(6), 493 (2003). 48

[97] A. Knoblauch, G. Palm, *et al. Pattern separation and synchronization in spiking associative memories and visual areas.* Neural networks: the official journal of the International Neural Network Society **14**(6-7), 763 (2001). 48

[98] G. Palm. *On associative memory.* Biological Cybernetics **36**(1), 19 (1980). 49

[99] J. Perez-Orive, O. Mazor, G. C. Turner, S. Cassenaer, R. I. Wilson, and G. Laurent. *Oscillations and sparsening of odor representations in the mushroom body.* Science **297**(5580), 359 (2002). 49

[100] K. S. Honegger, R. A. Campbell, and G. C. Turner. *Cellular-resolution population imaging reveals robust sparse coding in the drosophila mushroom body.* The Journal of Neuroscience **31**(33), 11772 (2011). 49

[101] R. Huerta, T. Nowotny, M. García-Sanchez, H. Abarbanel, and M. Rabinovich. *Learning classification in the olfactory system of insects.* Neural computation **16**(8), 1601 (2004). 49

[102] S. Cassenaer and G. Laurent. *Conditional modulation of spike-timing-dependent plasticity for olfactory learning.* Nature **482**(7383), 47 (2012).

[103] M. F. Strube-Bloss, M. P. Nawrot, and R. Menzel. *Mushroom body output neurons encode odor–reward associations.* The Journal of neuroscience **31**(8), 3129 (2011). 49

[104] R. Huerta. *Learning pattern recognition and decision making in the insect brain.* In *American Institute of Physics Conference Series*, vol. 1510, pp. 101–119 (2013). 49

[105] T. Pfeil, A. Grübl, S. Jeltsch, E. Müller, P. Müller, M. A. Petrovici, M. Schmuker, D. Brüderle, J. Schemmel, and K. Meier. *Six networks on a universal neuromorphic computing substrate.* Frontiers in neuroscience **7** (2013). 49, 85

[106] A. Riehle, S. Grün, M. Diesmann, and A. Aertsen. *Spike synchronization and rate modulation differentially involved in motor cortical function.* Science **278**(5345), 1950 (1997). 49

[107] H. Larochelle and Y. Bengio. *Classification using discriminative restricted boltzmann machines.* In *Proceedings of the 25th international conference on Machine learning*, pp. 536–543 (ACM, 2008). 49

[108] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. *Greedy layer-wise training of deep networks*. Advances in neural information processing systems **19**, 153 (2007). 49

[109] Q. Le, A. Karpenko, J. Ngiam, and A. Ng. *Ica with reconstruction cost for efficient overcomplete feature learning* (NIPS, 2011). 54

[110] D. Crisan and B. Rozovskii. *The Oxford Handbook of Nonlinear Filtering* (Oxford University Press, Oxford, 2011). URL http://econpapers.repec.org/RePEc:oxp:obooks:9780199532902. 54

[111] B. D. Willmore, J. A. Mazer, and J. L. Gallant. *Sparse coding in striate and extrastriate visual cortex*. Journal of neurophysiology **105**(6), 2907 (2011). 57

[112] A. V. Herz, T. Gollisch, C. K. Machens, and D. Jaeger. *Modeling single-neuron dynamics and computations: a balance of detail and abstraction*. science **314**(5796), 80 (2006). 57

[113] H. C. Tuckwell. *Introduction to theoretical neurobiology: Volume 2, nonlinear and stochastic theories*, vol. 8 (Cambridge University Press, 2005). 58

[114] M. P. Nawrot, C. Boucsein, V. Rodriguez Molina, A. Riehle, A. Aertsen, and S. Rotter. *Measurement of variability dynamics in cortical spike trains*. Journal of neuroscience methods **169**(2), 374 (2008). 58

[115] D. H. Perkel, G. L. Gerstein, and G. P. Moore. *Neuronal spike trains and stochastic point processes: I. the single spike train*. Biophysical journal **7**(4), 391 (1967). 58

[116] M. Nawrot, A. Aertsen, and S. Rotter. *Single-trial estimation of neuronal firing rates: from single-neuron spike trains to population activity*. Journal of neuroscience methods **94**(1), 81 (1999).

[117] H. Shimazaki and S. Shinomoto. *A method for selecting the bin size of a time histogram*. Neural Computation **19**(6), 1503 (2007). 58

[118] M. S. Okun. *Deep-brain stimulation for parkinson's disease*. New England Journal of Medicine **367**(16), 1529 (2012). 66

[119] J. Volkmann, A. Wolters, A. Kupsch, J. Müller, A. A. Kühn, G.-H. Schneider, W. Poewe, S. Hering, W. Eisner, J.-U. Müller, *et al. Pallidal deep brain stimulation in patients with primary generalised or segmental dystonia: 5-year follow-up of a randomised trial*. The Lancet Neurology **11**(12), 1029 (2012). 66

[120] R. A. Walsh, C. Sidiropoulos, A. M. Lozano, M. Hodaie, Y.-Y. Poon, M. Fallis, and E. Moro. *Bilateral pallidal stimulation in cervical dystonia: blinded evidence of benefit beyond 5 years*. Brain **136**(3), 761 (2013). 66

[121] S. Fahn. *The varied clinical expressions of dystonia*. Neurologic clinics **2**(3), 541 (1984). 66

[122] M. L. Kringelbach, N. Jenkinson, S. L. Owen, and T. Z. Aziz. *Translational principles of deep brain stimulation*. Nature Reviews Neuroscience **8**(8), 623 (2007). 66

[123] G. E. Alexander, M. R. DeLong, and P. L. Strick. *Parallel organization of functionally segregated circuits linking basal ganglia and cortex*. Annual review of neuroscience **9**(1), 357 (1986). 66

[124] A. M. Graybiel, T. Aosaki, A. W. Flaherty, and M. Kimura. *The basal ganglia and adaptive motor control*. Science **265**(5180), 1826 (1994). 67

[125] A. Currà, A. Berardelli, R. Agostino, N. Modugno, C. C. Puorger, N. Accornero, and M. Manfredi. *Performance of sequential arm movements with and without advance knowledge of motor pathways in parkinson's disease*. Movement disorders **12**(5), 646 (1997). 67

[126] A. A. Kühn, D. Williams, A. Kupsch, P. Limousin, M. Hariz, G.-H. Schneider, K. Yarrow, and P. Brown. *Event-related beta desynchronization in human subthalamic nucleus correlates with motor performance*. Brain **127**(4), 735 (2004). 67

[127] P. Brown and D. Williams. *Basal ganglia local field potential activity: character and functional significance in the human.* Clinical Neurophysiology **116**(11), 2510 (2005).

[128] C. Brücke, J. Huebl, T. Schönecker, W.-J. Neumann, K. Yarrow, A. Kupsch, C. Blahak, G. Lütjens, P. Brown, J. K. Krauss, *et al. Scaling of movement is related to pallidal γ oscillations in patients with dystonia.* The Journal of Neuroscience **32**(3), 1008 (2012). 67, 75

[129] A. Savitzky and M. J. Golay. *Smoothing and differentiation of data by simplified least squares procedures.* Analytical chemistry **36**(8), 1627 (1964). 71

[130] R. Meier, U. Egert, A. Aertsen, and M. P. Nawrot. *Finda unified framework for neural data analysis.* Neural Networks **21**(8), 1085 (2008). 71

[131] R. N. Bracewell and R. Bracewell. *The Fourier transform and its applications*, vol. 31999 (McGraw-Hill New York, 1986). 71

[132] W. Kruse and R. Eckhorn. *Inhibition of sustained gamma oscillations (35-80 hz) by fast transient responses in cat visual cortex.* Proceedings of the National Academy of Sciences **93**(12), 6112 (1996). 71

[133] R. Eckhorn, R. Bauer, W. Jordan, M. Brosch, W. Kruse, M. Munk, and H. Reitboeck. *Coherent oscillations: A mechanism of feature linking in the visual cortex?* Biological cybernetics **60**(2), 121 (1988). 71

[134] E. Stark and M. Abeles. *Predicting movement from multiunit activity.* The Journal of neuroscience **27**(31), 8387 (2007). 71

[135] A. Moran and I. Bar-Gad. *Revealing neuronal functional organization through the relation between multi-scale oscillatory extracellular signals.* Journal of neuroscience methods **186**(1), 116 (2010). 71

[136] J. D. Hunter. *Matplotlib: A 2d graphics environment.* Computing In Science & Engineering **9**(3), 90 (2007). 72, 86

[137] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. *Scikit-learn: Machine learning in Python.* Journal of Machine Learning Research **12**, 2825 (2011). 73, 86

[138] E. Jones, T. Oliphant, P. Peterson, *et al. SciPy: Open source scientific tools for Python* (2001–). URL http://www.scipy.org/. 73, 86

[139] N. C. Silver and W. P. Dunlap. *Averaging correlation coefficients: Should fisher's z transformation be used?* Journal of Applied Psychology **72**(1), 146 (1987). 73

[140] A. M. Graybiel. *The basal ganglia: learning new tricks and loving it.* Current opinion in neurobiology **15**(6), 638 (2005). 75

[141] T. Wichmann, H. Bergman, and M. DeLong. *The primate subthalamic nucleus. iii. changes in motor behavior and neuronal activity in the internal pallidum induced by subthalamic inactivation in the mptp model of parkinsonism.* Journal of Neurophysiology **72**(2), 521 (1994).

[142] J. T. Gale, D. C. Shields, F. A. Jain, R. Amirnovin, and E. N. Eskandar. *Subthalamic nucleus discharge patterns during movement in the normal monkey and parkinsonian patient.* Brain research **1260**, 15 (2009). 75, 76

[143] M. DeLong. *Activity of pallidal neurons during movement.* Journal of neurophysiology **34**(3), 414 (1971). 76

[144] J. P. Donoghue. *Bridging the brain to the world: a perspective on neural interface systems.* Neuron **60**(3), 511 (2008). 77

[145] M. Kim, G. Wu, and D. Shen. *Unsupervised deep learning for hippocampus segmentation in 7.0 tesla mr images.* In *Machine Learning in Medical Imaging*, pp. 1–8 (Springer, 2013). 84

[146] M. Sahani. *Latent variable models for neural data analysis.* Ph.D. thesis, California Institute of Technology (1999).

[147] K. Obermayer, H. Ritter, and K. Schulten. *A principle for the formation of the spatial structure of cortical feature maps.* Proceedings of the National Academy Of Sciences **87**(21), 8345 (1990). 84

[148] S. Dähne, F. C. Meinecke, S. Haufe, J. Höhne, M. Tangermann, K.-R. Müller, and V. V. Nikulin. *Spoc: A novel framework for relating the amplitude of neuronal oscillations to behaviorally relevant parameters.* NeuroImage **86**, 111 (2014). 84

[149] B. Blankertz, G. Dornhege, M. Krauledat, K.-R. Müller, and G. Curio. *The non-invasive berlin brain–computer interface: fast acquisition of effective performance in untrained subjects.* NeuroImage **37**(2), 539 (2007).

[150] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue. *Brain-machine interface: Instant neural control of a movement signal.* Nature **416**(6877), 141 (2002).

[151] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. Nicolelis. *Learning to control a brain–machine interface for reaching and grasping by primates.* PLoS biology **1**(2), e42 (2003).

[152] F. Pereira, T. Mitchell, and M. Botvinick. *Machine learning classifiers and fmri: a tutorial overview.* Neuroimage **45**(1), S199 (2009). 84

[153] S. Nishimoto, A. T. Vu, T. Naselaris, Y. Benjamini, B. Yu, and J. L. Gallant. *Reconstructing visual experiences from brain activity evoked by natural movies.* Current Biology **21**(19), 1641 (2011). 84

[154] M. A. van Gerven, F. P. de Lange, and T. Heskes. *Neural decoding with hierarchical generative models.* Neural computation **22**(12), 3127 (2010). 84

[155] F. Rosenblatt. *The perceptron: a probabilistic model for information storage and organization in the brain.* Psychological review **65**(6), 386 (1958). 85

[156] W. S. McCulloch and W. Pitts. *A logical calculus of the ideas immanent in nervous activity.* The bulletin of mathematical biophysics **5**(4), 115 (1943). 85

[157] D. Brüderle, M. A. Petrovici, B. Vogginger, M. Ehrlich, T. Pfeil, S. Millner, A. Grübl, K. Wendt, E. Müller, M.-O. Schwartz, *et al. A comprehensive workflow for general-purpose neural modeling with highly configurable neuromorphic hardware systems.* Biological cybernetics **104**(4-5), 263 (2011). 85

[158] K. Boahen. *Neuromorphic microchips.* Scientific American **292**(5), 56 (2005). 85

[159] . 85

[160] A. Hodgkin and A. Huxley. *Propagation of electrical signals along giant nerve fibres.* Proceedings of the Royal Society of London. Series B, Biological Sciences pp. 177–183 (1952). 85

[161] C. Häusler, M. P. Nawrot, and M. Schmuker. *A spiking neuron classifier network with a deep architecture inspired by the olfactory system of the honeybee.* In *Proceedings of the 5th International IEEE EMBS Conference on Neural Engineering, Cancun, Mexico*, pp. 198–202 (2011). 85

[162] B. Kasap and M. Schmuker. *Improving odor classification through self-organized lateral inhibition in a spiking olfaction-inspired network.* In *Neural Engineering (NER), 2013 6th International IEEE/EMBS Conference on*, pp. 219–222 (IEEE, 2013).

[163] M. Schmuker, T. Pfeil, and M. P. Nawrot. *A neuromorphic network for generic multivariate data classification.* Proceedings of the National Academy of Sciences p. 201303053 (2014). 85

[164] P. Lichtsteiner, C. Posch, and T. Delbruck. *A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor.* Solid-State Circuits, IEEE Journal of **43**(2), 566 (2008). 85

[165] T. Delbruck. *Silicon retina with correlation-based, velocity-tuned pixels.* Neural Networks, IEEE Transactions on **4**(3), 529 (1993).

[166] A. Andreou and K. Boahen. *A contrast sensitive silicon retina with reciprocal synapses.* Advances in Neural Information Processing Systems (NIPS) **4**, 764 (1991). 85

[167] J. J. Hopfield. *Neural networks and physical systems with emergent collective computational abilities.* Proceedings of the national academy of sciences **79**(8), 2554 (1982). 85

[168] L. Wiskott and T. J. Sejnowski. *Slow feature analysis: Unsupervised learning of invariances.* Neural computation **14**(4), 715 (2002). 85

[169] P. Berkes and L. Wiskott. *Slow feature analysis yields a rich repertoire of complex cell properties.* Journal of Vision **5**(6), 9 (2005). 85

[170] M. Franzius, H. Sprekeler, and L. Wiskott. *Slowness and sparseness lead to place, head-direction, and spatial-view cells.* PLoS Computational Biology **3**(8), e166 (2007). 85

[171] A. M. Saxe, M. Bhand, R. Mudur, B. Suresh, and A. Y. Ng. *Unsupervised learning models of primary cortical receptive fields and receptive field plasticity.* In *NIPS*, pp. 1971–1979 (2011). 85

[172] T. E. Oliphant. *Python for scientific computing.* Computing in Science & Engineering **9**(3), 10 (2007). 86

[173] W. McKinney. *Data structures for statistical computing in python.* In S. van der Walt and J. Millman, eds., *Proceedings of the 9th Python in Science Conference*, pp. 51 – 56 (2010). 86

[174] F. Perez and B. E. Granger. *Ipython: a system for interactive scientific computing.* Computing in Science & Engineering **9**(3), 21 (2007). 86

[175] C. Häusler. *scikitCVcluster: A wrapper of ipcluster around scikits learn classifiers to perform parallel cross validation* (2013). URL https://github.com/chausler/scikitCVcluster. 86

[176] O. Botvinnik. *prettyplotlib: Painlessly create beautiful matplotlib plots* (2013). URL https://github.com/olgabot/prettyplotlib. 86

[177] F. Pérez, B. E. Granger, and J. D. Hunter. *Python: an ecosystem for scientific computing*. Computing in Science & Engineering **13**(2), 13 (2011). 86

[178] T. H. Davenport, D. Patil, *et al. Data scientist: the sexiest job of the 21st century*. Harvard business review **90**(10), 70 (2012). 87