Pruning over-representations in stochastic latent encoder models

Carolin Scholl

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2018

Abstract

Inspired by early information-theoretic approaches to network reduction, we aimed to find the "optimal brain damage" in restricted and deep Boltzmann machines fit to image patterns. Using Fisher information as a measure of parameter sensitivity, we iteratively removed the least important weights and units from initially over-parameterized models while monitoring their generative and encoding performance. Their fit could be preserved even after removing the vast majority of weights. Pruning the most important parameters instead resulted in a loss of critical units that even a full retraining could not compensate for. As it is still difficult to predetermine the optimal architecture of artificial neural networks, our results suggest that a strategy pursued by the brain – an initial over-parameterization and subsequent network reduction – may be advantageous. Notably, our estimate of Fisher Information is theoretically available to an individual neuron as an indicator of self-relevance. We thus argue that our experiments may model aspects of naturally occurring experience-dependent pruning of synapses and neurons in the early development of sensory systems.

Acknowledgements

I wish to thank my supervisor, Matthias Hennig, for the guidance and support throughout this project. Furthermore, I like to thank Michael Rule who was essentially a second supervisor to me.

Their unconventional way of thinking and approaching problems leave me impressed. I was encouraged to keep asking any questions and I was never afraid to do so. With great patience, they were able to explain fundamentals and details in a way for me to understand. I am very grateful to have been given the opportunity to learn more about computational neuroscience, a field which I would have not been able to encounter on my own.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Carolin Scholl*)

To deltic and kingbird.

Table of Contents

Abbreviations

ANN	Artificial neural network
CD	Contrastive divergence
DBM	Deep Boltzmann machine
FI	Fisher Information
FIM	Fisher Information Matrix
KL divergence	Kullback-Leibler divergence
MI	Mutual Information
OBD	Optimal brain damage
OBS	Optimal brain surgeon
PCD	Persistent contrastive divergence
RBM	Restricted Boltzmann machine
RGC	Retinal ganglion cell
SVM	Support vector machine

Chapter 1

Introduction

> "[The human brain] has about 10^{14} parameters and we only live for about 10^9 seconds. Synapses are much cheaper than experiences, so it makes sense to throw a lot of synapses at each experience."
>
> Geoffrey Hinton, 2016

Geoffrey Hinton, one of the leading figures in artificial intelligence, recently made a disputable suggestion on what may underlie the extraordinary capacity of the human brain given the magnitude of information it is confronted with in a lifetime. While most would agree that the synaptic connections are the cornerstone for neural information transmission and the processing of sensory experience, the numerical juxtaposition of synapses vs. seconds may well be controversial. Does the number of synapses define an upper bound on the experiences a human can make in their lifetime? What is the minimum number of synapses needed to encode a single experience? Can we only have one experience per second? Is the number of synapses constant throughout life?

This thesis aims to challenge the currently prominent "the more, the merrier" philosophy in the field of artificial intelligence, a discipline that has been and is still being inspired by the computational capacity and efficiency of the biological brain. Neuronal plasticity, i.e. the change of the anatomy and function of areas of the brain as a response to its environment, is a key mechanism of human intelligence. The number and strength of synapses varies continuously throughout life. This dynamic adaptation is seen as the basic mechanism that allows learning.

Yet strikingly, a lot of synapses and neurons are lost during healthy early brain development. Why would this happen if the key to information encoding and storage was the sheer number of synapses? It rather suggests that the optimal wiring for an individual environment cannot be predetermined and has to be achieved through experience.

This optimal wiring does not make use of all initially available neurons and synapses. As a consequence, they are pruned away.

Some would even argue that synapses are not that cheap after all, considering their order of magnitude. Relative to the metabolism of the rest of the body, the brain is energetically expensive (Mink et al., 1981). From an economic point of view, the brain mass and energy needed for widespread connectivity can be seen as costs. It has been argued that the natural development and organization of neural circuits can be interpreted as a means to minimize these "wiring costs" (Bullmore and Sporns, 2012). Still, the brain needs to maintain reserves in order to be adaptive and resilient against damage. Indeed, there is evidence for widespread redundant wiring of multiple synapses between two cortical neurons which may improve learning (Hiratani and Fukai, 2018). In the light of the negative effects of over-pruning associated with neurodegenerative and psychiatric disorders, there seems to be a sweet spot between pruning and over-pruning in order to preserve a healthy level of plasticity in the brain.

We aim to find the minimal possible size of a biologically inspired model of visual sensory encoding that satisfies the trade-off between minimizing wiring costs and preserving functional learning. The models we are using are Boltzmann machines. In essence, they are Artificial neural networks (ANNs) whose units (neurons) belong to different layers and have weighted connections (synapses) between them.

In an information-theoretic approach, Rule et al. (2018) recently demonstrated that the Fisher Information Matrix (FIM) reveals the optimal size for such models. The entries of this matrix indicate the relative importance of individual parameters. Aiming to model aspects of natural occurring pruning in the development of sensory systems, we start with an initially over-parameterized model and iteratively reduce its size by removing synapses and neurons that have a relatively low Fisher Information (FI). Strikingly, the estimation of FI of weights in these models is based on computing the covariance between the firing rates of presynaptic and postsynaptic neurons. This measure of self-relevance is theoretically computable for an individual neuron, contributing to biological plausibility. In a comparison of removing the most vs. least important parameters we aim to find insights into the causes and dynamics of pruning and apoptosis in natural brain development and disease. Furthermore, a closer investigation into the role of individual parameters and units in ANNs may contribute to basic research in machine learning and artificial intelligence, leading the way to more efficient, biologically inspired learning in models of reduced size.

This dissertation is structured as follows: Chapter 2 will provide the biological

background in order to motivate our chosen model class which will be explained in detail. Alternative pruning approaches will be critically examined before we introduce FI as our pruning criterion. Chapter 3 comprises our experimental methods. After introducing the two datasets used, we will explain the details of model fitting including hyperparameter choices and their justification. We will also discuss the advantages and disadvantages of several evaluation measures. Different pruning criteria and schedules will be explained. In Chapter 4 we present our results that are roughly divided into two parts, depending on the dataset used. In Chapter 5 we thoroughly discuss the relevance as well as the limitations of our work. Lastly, Chapter 6 serves a brief overall summary of the project.

Chapter 2

Theoretical Background

2.1 Pruning and Apoptosis in the Development of the Visual System

Biological neurons communicate with each other by sending neurotransmitters over the synaptic cleft from the axonal branches of a presynaptic to the dendrites of a postsynaptic neuron where they bind to the appropriate receptors. The synapses can be broadly divided into excitatory and inhibitory: they either lead to a depolarization of the postsynaptic neuron or a hyperpolarization. If the voltage of the postsynaptic neuron exceeds a threshold, it produces an action-potential. It is said to fire or spike, eventually leading to the secretion of neurotransmitters at its synapses with other neurons. This is the fundamental principle of neural information processing. Now what enables animals to learn is plasticity, i.e. the dynamic strengthening and weakening of synapses as a response to sensory experiences. Yet there are certain phases during natural brain development characterized by very high plasticity which unintuitively entail a drastic reduction of neurons and synapses. This section summarizes relevant findings on plasticity of the early visual system in order to motivate the models and pruning mechanism we use for our simulations.

2.1.1 Developmental Exuberance and Critical Periods

[1]The human brain experiences a period of exuberant over-production of synapses, axons and axonal branches in its early development (Innocenti and Price, 2005). The

[1]This chapter was inspired from and consists of extensions to my Informatics Project Proposal (Scholl, 2018).

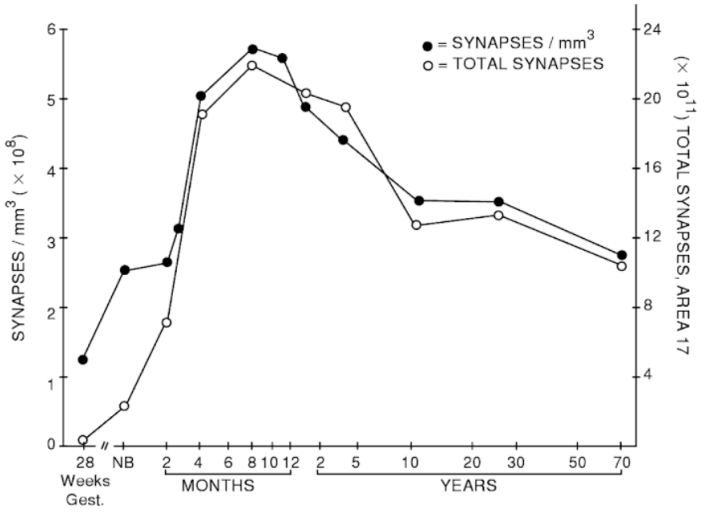


Figure 2.1: Total number of synapses and synaptic density in Brodmann area 17 (primary visual cortex) as a function of age. Figure reprinted from Huttenlocher (1990).

number of synapses increases dramatically in late gestation, a growth that extends over the first two years of life (Huttenlocher et al., 1997). Yet surprisingly, only about half of these early synaptic connections between neurons survive until adolescence (Chechik et al., 1999). This is because the initial surplus of synapses is followed by a period of excessive synaptic pruning. Similar rates of over-production and subsequent decline have been observed in other species as well (e.g. Innocenti, 1995).

Figure 2.1 visualizes said structural changes observed in morphometric studies of the primary visual cortex (Huttenlocher, 1990). Both the synaptic density and total number of synapses first increase rapidly, at which point the process is suddenly reverted until reaching a plateau. The shape of this curve is typical for the development of neuronal circuits across all areas of the brain. Yet the primary sensory and motor areas experience these progressive and regressive phases first. Considering associated functions, this makes sense as they define the basis for the development of higher cognitive functioning (Casey et al., 2005).

If the desired cortical mass and wiring was known before birth, this initial overproduction would be a waste of both time and resources. It has thus been suggested that the exact phenotype of the cortex is indeed not entirely predefined. Rather, it seems that it is adjusted after birth in a complex interplay between neurons and their cellular surroundings (Innocenti, 1995). This hypothesis can be extended to the macroperspective: perhaps the exact architecture of neural circuits is delayed in order to be dynamically adaptable to the sensory input an infant receives through interactions with

its environment. This is the core idea behind the theory of adaptive plasticity (Johnston, 2004).

Johnston (2004) metaphorically speak of "sculpting" the brain: infants show increased synaptic density and plasticity which is "under construction". The elevated synaptic density and plasticity allows for refinement to the individual environment reflected in the sensory input. Yet the metaphor can be misleading. Importantly, it is not the case that the brain is set to stone once the synaptic density has stabilized in late adolescence, as seen in Figure 2.1. To the contrary, lifelong neuroplasticity evident from synaptic re-organization, adult synaptogenesis, and even neurogenesis allows for continuous learning, adaptability and neurorehabilitation after injuries (e.g. Berlucchi, 2011; Lledo et al., 2006). Rather, the potency of sensory experiences to form and sculpt neural circuits is greatly elevated early in life and balances afterwards (Takesian and Hensch, 2013).

These early phases of high neuroplasticity are also known as critical periods. The term is motivated from the observation that sensory deprivation during these phases may lead to long-lasting or even irreversible malfunctioning of sensory systems. Hence exposure to the habitual environment of an organism is critical for healthy neurodevelopment. Classical studies by Hubel and Wiesel (1962) demonstrated the detrimental effects of monocular deprivation for the development of the visual system in kittens. Notably, after re-allowing binocular vision, the originally deprived eye had only a small number of neurons responding to visual stimuli while the other eye overcompensated with abnormally high firing rates. Connections and neurons were lost as a consequence of missing sensory input during a critical period of development. Their studies were among the first to motivate the idea of adaptive plasticity and give reason to assume that synaptic restructuring is elicited in an experience-dependent and activity-dependent manner. While some connections are strengthened, others become redundant and regress. Entirely unconnected neurons can safely be removed since they no longer contribute to neural information processing. It has been argued that such orphan cells that fail to wire need to be removed in order for the network to function (Meier et al., 2000).

The orderly self-destruction of a complete cell is called apoptosis and is yet another natural occurring regressive event that happens in large numbers during healthy brain development (Yuan and Yankner, 2000). For example, twice as many Retinal ganglion cells (RGCs) have been observed in new-born rats and hamsters compared to adult animals (Perry et al., 1983; Sengelaub and Finlay, 1982). The development of initially

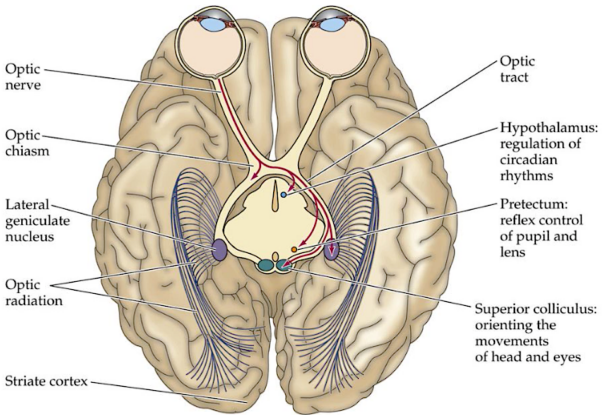


Figure 2.2: Axial view of the human brain. Red arrows indicate the projections of axons of RGCs along the optic nerve. Targets are various structures of the midbrain such as the pretectum or superior colliculus. Figure reprinted from Purves et al. (2008).

diffuse and wide connections followed by subsequent refinement is comparably well studied in these early stages of visual processing. RGCs transmit sensory information about light that was first transducted by the photoreceptors of the retina to different areas of the midbrain (see Figure 2.2). Their axons leave the retina at the so-called blind spot and form the optic nerve. Main targets are neurons belonging to the pretectum or superior colliculus (Purves et al., 2008).

Previous studies suggest that there are two main interacting signals that determine the survival or apoptosis of RGCs. First, the cells compete for establishing synapses with their target cell population (see Figure 2.3). Specifically, their survival is dependent on so-called neurotrophic factors that are secreted in limited amounts from the target neurons (Meier et al., 2000; Meyer-Franke et al., 1998). Lack thereof leads to RGCs dying from “trophic withdrawal” (Becker and Bonni, 2004).

However, the trophic factors alone are not sufficient for an RGC to survive. Additionally, their intracellular levels of cAMP need to be increased in order to be responsive to the neurotrophic factors (Meyer-Franke et al., 1998). cAMP levels are increased by depolarization through calcium inflow in the cell which is caused by electric activity of the neuron (Becker and Bonni, 2004; Franklin and Johnson, 1992; Meyer-Franke et al., 1998). It has further been shown that electrical activity not only supports the survival, but also accelerates the axon growth of RGCs (Goldberg et al., 2002). These findings are in line with our hypothesis that cell death is triggered in an experience- and activity-dependent manner: lack of neuronal activity may indicate the irrelevance of an RGC for the precise encoding of stimuli, and motivate its self-destruction.

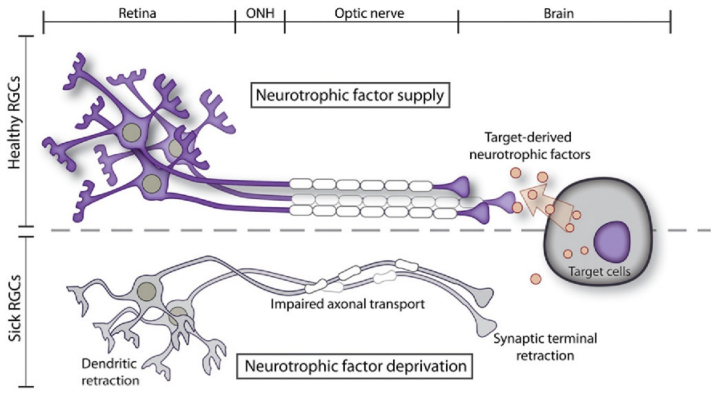


Figure 2.3: RGCs compete for limited neurotrophic factor supply from the target cells. Inactive RGCs that are unresponsive to the trophic factors do not establish connections and die of trophic withdrawal. Figure reprinted from Levkovitch-Verbin (2015).

2.1.2 Maladaptive Pruning

The cellular processes determining synaptic pruning and apoptosis are fragile. As mentioned above, sensory deprivation during critical periods may lead to irreversible changes in the neural circuits. Generally, the matching of neurons with target cells has been described as stochastic and imprecise (Meier et al., 2000). Apart from that, genetic mutations may disrupt plasticity and cause disadvantageous synaptic wiring (Flavell and Greenberg, 2008). Although pruning and apoptosis occur in large numbers in healthy brain development, they need to be executed within boundaries in order to be adaptive.

That is because over-pruning may result in neuronal dysfunction expressing itself in clinical disorders or cognitive impairments (Johnston, 2004). Both excessive apoptosis (Yuan and Yankner, 2000) and axonal degeneration (Low and Cheng, 2006) are suspected to be involved in the pathophysiology of neurodegenerative and psychiatric disorders such as schizophrenia (e.g. Feinberg, 1982; Sekar et al., 2016). Given that pruning occurs in natural brain development, but is linked to diseases if over-presented, we aim to find the “optimal brain damage” (LeCun et al., 1990) in a model of visual encoding. Specifically, this may simulate aspects of the activity-dependent pruning and over-pruning of RGCs and their sensory afferents. Since pathological and natural cell death may follow similar molecular mechanisms (Yuan and Yankner, 2000), understanding the factors contributing to apoptosis in healthy brain development and disease may help in getting a more coherent view of adaptive and maladaptive plasticity.

2.2 RBMs as a Model of Visual Encoding

The biological processes we aim to simulate specify certain requirements for the model choice. First of all, we need a layered model that can be organized into pre- and post-synaptic neurons. Second, if we want to model visual encoding, certain neurons should not be directly determined. Since the target cells affect pruning as well, the connections between neurons should be bidirectional and the model should be expandable in its number of layers. Lastly, the learning signal should be unsupervised and locally available for an individual neuron, thereby conforming to biological plausibility.

Restricted Boltzmann machines (RBMs) generally fulfill all these requirements. Rule et al. (2018) recently used RBMs to simulate the sensory encoding of image patterns through RGCs. RBMs have served as a model of RGC responses to visual stimuli before (e.g. Ganmor et al., 2011; Zanotto et al., 2017). Strikingly, several RBMs can be connected in order to build a multi-layer model of hierarchical visual processing (e.g. Reichert, 2012). This makes the model class particularly interesting for our purposes.

2.2.1 Model architecture

Standard RBMs as introduced by Smolensky (1986) are generative ANNs consisting of n binary units (neurons) which are organized in two layers: a visible layer $\mathbf{v}$ that directly corresponds to a given input vector and a hidden (or latent) layer $\mathbf{h}$ which may encode higher-order features of the input data. Hence, they are also referred to as latent encoder models. The hope is to learn an abstract representation of the causes of the data, in order for the model to be able to generate such data itself. Furthermore, the hidden unit representations may be useful features for a classifier to train on, instead of using the original data (e.g. Hinton, 2007). In that sense, RBMs are often used as feature detectors in order to preprocess data for supervised learning problems.

In Bernoulli RBMs the value of a neuron is binary, i.e. its state is either on (it is firing) or off (it is silent). Importantly, all neurons fire stochastically and can thus be described by probability distributions. Each visible layer neuron v_i has undirected weighted connections (synapses) to each hidden neuron h_j and vice versa, but neurons within a layer are not connected to each other (see Figure 2.4). The weights are stored in a matrix $\mathbf{W} \in \mathbb{R}^{n_v \times n_h}$, where n_v is the number of visible and n_h is the number of hidden units. Each unit further has a so-called bias, b_i^v or b_j^h respectively. A bias can be thought of as yet another weight to an additional, imaginary neuron that is always in

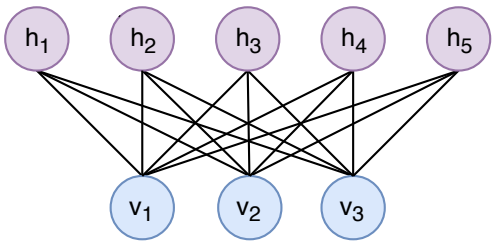


Figure 2.4: Sketch of a standard RBM. Lines denote weights between neurons, where the letter indicates their layer: v for visible, h for hidden. RBMs are only connected between layers, not within.

the on-state. It corresponds to the excitability of a neuron. As we mostly aim for sparse representations – and indeed cortical neurons are predominantly silent (Shoham et al., 2006) – biases are often initialized to negative values (Hinton, 2016). They can be summarized to a bias vector per layer, $\mathbf{b}^{\mathbf{v}}$ and $\mathbf{b}^{\mathbf{h}}$ respectively. All of these to-be-learnt parameters make up the set $\phi = (\mathbf{W}, \mathbf{b}^{\mathbf{v}}, \mathbf{b}^{\mathbf{h}})$.

RBMs are energy-based-models in that they assign energy to each configuration of the observed variables (i.e. visible units v_i) that they try to find statistical dependencies in (LeCun et al., 2006). The energy function for a configuration of all visible and hidden neurons in an RBM is given by (Hinton, 2012):

$$E^{\phi}_{\mathbf{v},\mathbf{h}} = -\sum_{i=1}^{n_v} b^v_i v_i - \sum_{j=1}^{n_h} b^h_j h_j - \sum_{i=1}^{n_v}\sum_{j=1}^{n_h} v_i h_j w_{ij} \tag{2.1}$$

This can be converted to a joint probability of the configuration, with Z being the partition function that sums over all possible configurations of visible and hidden neurons (Hinton, 2012):

$$p(\mathbf{v},\mathbf{h}) = \frac{1}{Z} e^{-E^{\phi}_{\mathbf{v},\mathbf{h}}} \tag{2.2}$$

From the bipartite graph structure, conditional independence between the two layers can be inferred. This independence assumption dramatically reduces the number of states to consider when computing the sum over configurations. Yet, the partition function is still intractable in many cases (LeCun et al., 2006), especially for larger models.

2.2.2 Boltzmann Learning

In RBMs, the probability of a visible pattern $\mathbf{v}$ can be computed by summing over all configurations of hidden neurons (Hinton, 2012):

$$p(\mathbf{v}) = \frac{1}{Z}\sum_{\mathbf{h}} e^{-E^{\phi}_{\mathbf{v},\mathbf{h}}} \tag{2.3}$$

The training objective then is to adjust the parameters ϕ such that the energy for a particular training pattern is lowered compared to the energies of competing patterns in the training set (Hinton, 2012). Lowering its energy corresponds to assigning a greater probability to that pattern.

Taking the logarithm of equation 2.3, it can be rewritten as:

$$\log p(\mathbf{v}) = \log(\sum_{\mathbf{h}} e^{-E^{\phi}_{\mathbf{v},\mathbf{h}}}) - \log Z \tag{2.4}$$

Remembering that the partition function comprises all possible configurations, we aim to maximize the first term, i.e. the unnormalized log probability assigned to the training vector $\mathbf{v}$, and to minimize the second term, i.e. the log probability assigned to all others (Reichert, 2012). Thereby, we maximize $p(\mathbf{v})$.

This can be formulated as a standard optimization problem with equation 2.4 as an objective function. For RBMs, the partial derivative of the log probability for one training instance $\mathbf{v}$, in our case the pixels of a binary image, with respect to a weight is given by (Hinton, 2012):

$$\frac{\partial \log p(\mathbf{v})}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \tag{2.5}$$

The angle brackets indicate expectations, the respective distributions are in subscripts: *data* stands for $p(\mathbf{h}|\mathbf{v}^n)$, and *model* denotes the joint distribution $p(\mathbf{v},\mathbf{h})$. For a detailed derivation, we refer to Reichert (2012). Maximizing the log likelihood of the data by finding appropriate parameters is equivalent to minimizing the Kullback-Leibler divergence (KL divergence) between these two distributions (Hinton et al., 2006). Generally, equation 2.5 cannot be solved analytically. Approaching the problem with stochastic gradient ascent instead, the weights would be iteratively updated for one randomly chosen training image at a time, scaled by the learning rate η (Hinton, 2012):

$$\Delta w_{ij} = \eta(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}) \tag{2.6}$$

Alternatively, this difference can be computed for more than one training image before applying a weight update. The differences then need to be summed and averaged over the number of training instances in a so-called mini-batch.

Notably, this learning procedure is unsupervised and local. The updates are essentially based on correlations between any two neurons of different layers, which resembles the fundamental idea of Hebbian learning: "neurons wire together if they fire together" (Lowel and Singer, 1992, p. 211). The biases b are updated based on the averages of the respective visible or hidden neurons:

$$\begin{aligned} \Delta b_i^v &= \eta(\langle v_i \rangle_{data} - \langle v_i \rangle_{model}) \\ \Delta b_j^h &= \eta(\langle h_j \rangle_{data} - \langle h_j \rangle_{model}) \end{aligned} \tag{2.7}$$

The expectations are usually approximated by sampling. However, obtaining an unbiased sample is unproblematic only for one of the two terms in equation 2.5. Approximating $\langle v_i h_j \rangle_{data}$ can be thought of as a forward-pass through the network. The visible neurons are clamped to the values of a particular training example $\mathbf{v}$ at a time. Due to the restricted connections, RBMs provide the factorial posterior distribution over the hidden states given the visibles (Le Roux et al., 2011). Exploiting this conditional independence of a layer given another, the probability for each hidden unit to be in the on-state given a configuration of the visible layer $\mathbf{v}$ is then computed as:

$$p(h_j|\mathbf{v}) = \sigma\left(b_j^h + \sum_i v_i w_{ij}\right), \tag{2.8}$$

where σ denotes the logistic sigmoid function (Hinton, 2012). This is also called the positive or wake phase, in contrast to the negative or sleep phase (e.g. Bengio and Delalleau, 2009; Hinton et al., 1995). During the latter, the model is said to "run free" or "dream" without input data in order to sample from the joint model distribution. More specifically, the visible layer neurons are initialized to a random state. Equation 2.8 is then used to update the binary states of all hidden neurons in parallel. Based on this sampled hidden layer vector $\mathbf{h}$, the states of all visible neurons are re-updated with the analogous equation (Hinton, 2012):

$$p(v_i|\mathbf{h}) = \sigma\left(b_i^v + \sum_j h_j w_{ij}\right) \tag{2.9}$$

This sampling procedure is called alternating Gibbs sampling. As a Markov chain Monte Carlo algorithm it produces a Markov chain as seen in Figure 2.5. It is run for many iterations for the samples to be unbiased and originate from the equilibrium

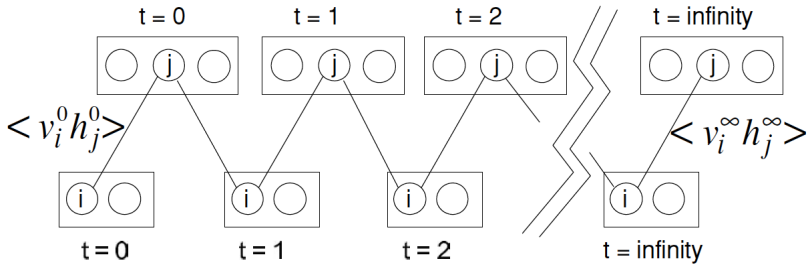


Figure 2.5: Illustration of the Markov chain traversed by alternating Gibbs sampling. A rectangle in the top layer represents a hidden layer at a particular time step t, analogously the visible layer is shown below. Layer by layer, their neurons are updated in parallel based on the current fixed state of the opposite layer using equations 2.8 and 2.9 in alternation for $t \to \infty$ time steps. Figure reprinted from Hinton et al. (2006).

distribution of the model (Hinton et al., 2006). Metaphorically, we want the model to move away from the random initialization and explore the full distribution. Due to the conditional independence of the layers, the sampling procedure can be parallelized and thus massively profits from GPU-accelerated computing.

2.2.3 Contrastive Divergence

Unfortunately the learning procedure described above tends to be slow, even with said computational resources. Contrastive divergence (CD) is an alternative objective function that speeds up and facilitates the learning process since its partial derivatives can be approximated more efficiently (Hinton, 2002). In CD the weight update Δw_{ij} is shortened to (Hinton, 2002):

$$\Delta w_{ij} = \eta(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{reconstruction}) \tag{2.10}$$

The key idea is to approximate the expectation over the model distribution in the negative phase with a single reconstruction in lieu of many samples (Reichert, 2012). Instead of initializing the visible layer to a random state, its neurons are set to the values of a training instance. The hidden unit activations are then computed using equation 2.8, based on which a reconstruction is generated using equation 2.9 (Hinton, 2002). Specifically, this is known as one-step CD or CD-1 since the Markov chain is effectively truncated after just one iteration or Gibbs step. The alternating sampling can be repeated k times, motivating the names k-step CD or CD-k. The process is depicted and contrasted to standard Boltzmann learning in Figure 2.6.

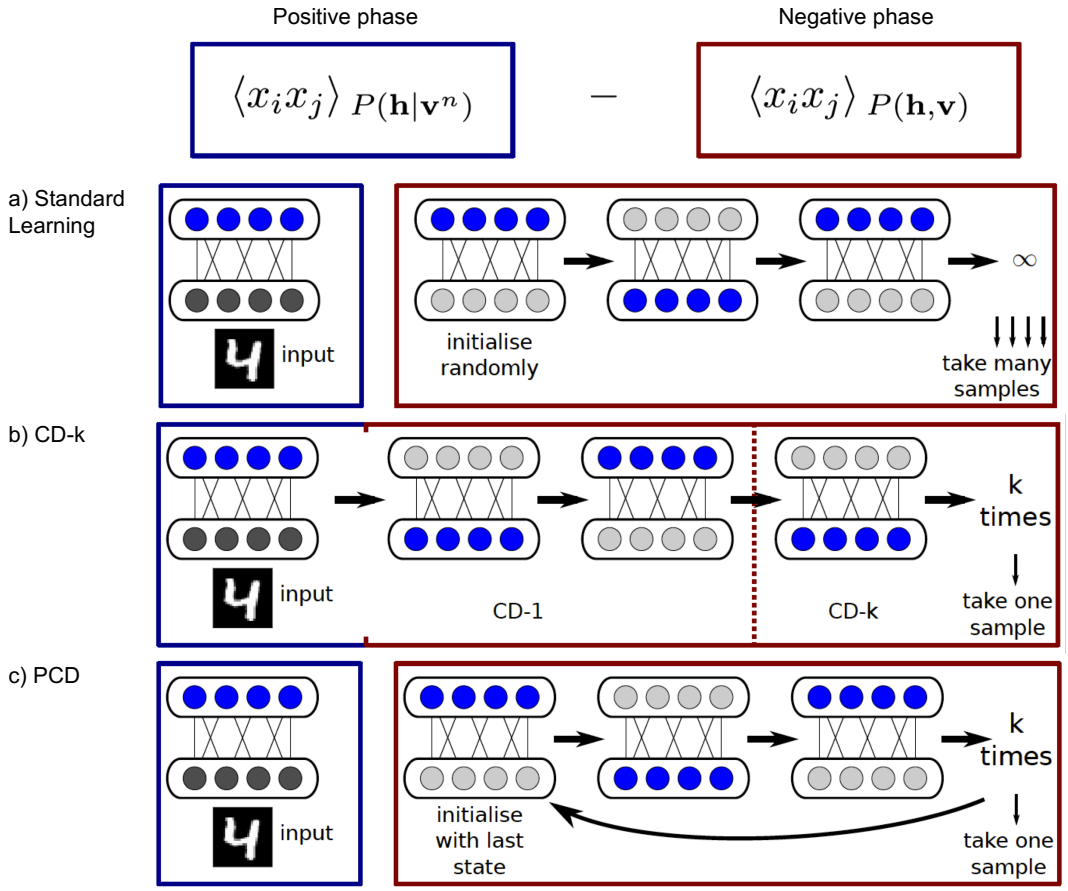


Figure 2.6: Illustration of the positive-negative algorithm for training RBMs. Parameters are updated after each passage of the two phases that are both based on sampling the states of the layers and computing correlations between the two, in parallel for all their neurons. The positive phase is initialized by fixing the visibles to a training example. The posterior distribution over the hidden neurons is inferred and a value for each hidden neuron is sampled from this distribution. The negative phase consists of letting the model "run free". (a) In standard learning the negative phase is initialized with a random state of the visibles, after which the sampling is alternated between the two layers many times until hypothetically reaching equilibrium. (b) In CD-k the two phases are merged together: the negative phase directly continues after the initialization to a training example and the alternating Gibbs sampling is run for k times. (c) In Persistent contrastive divergence (PCD), the Markov chain is persistent because it is not initialized with the next training example, but with the state after k steps. Figure adapted and reprinted from Reichert (2012).

The logic of the positive-negative algorithm still applies, only now the two phases can be thought of as being merged together: the positive phase is initialized with a training instance as before, but the negative phase does not start with a separate random initialization anymore. Instead, the sampling is continued starting from the same initialization. The correlations at the k^{th} Gibbs step now serve to compute the contribution of the negative phase (Reichert, 2012). After one cycle of a positive and a negative phase, the process restarts with a new training example as initialization. The weight update is applied after passing through all training vectors of a mini-batch.

It is indeed surprising that CD-k works well enough for most application purposes as it is not directly approximating the gradient of our initial objective function from Equation 2.4 (Hinton, 2002, 2012; Sutskever and Tieleman, 2010). Bengio and Delalleau (2009) justify CD by showing that the residual term after truncating the Markov chain converges to zero when $k \to \infty$. They interpret the method as reconstruction error learning. Larger values for k are recommended by some (e.g. Tieleman, 2008), but oftentimes even CD-1 is reported to work reasonably well.

Yet it should be noted that accelerating the learning process with CD may still come at a price. Reichert (2012) warned that a model may demonstrate poorer generative than reconstruction performance if trained using CD-k as it hinders a full exploration of the model distribution. As a consequence, the model may have to be run for a long time for its samples to be less correlated. A means to correct this is PCD (Tieleman, 2008), where the Markov chain is not re-initialized with a new training vector after each positive-negative phase (see Figure 2.6). That way, the sampling is persistent as it is always based on the previous state.

2.2.4 RBMs with Localized Receptive Fields

The number of connections in RBM may be further restricted by introducing localized receptive fields. A receptive field of a sensory neuron is defined by the region of receptors it is connected to and receives input from. If we consider vision, the receptive field corresponds to a number of photoreceptors of the retina that cover a small region of the visual field. A stimulus in this small field of view may then evoke firing in the respective neuron.

The introduction of receptive fields in RBM leads to a hidden neuron not being connected to all visible neurons anymore. Analogous to a receptive field of a sensory neuron, the hidden unit now only receives input from the part of the image that is

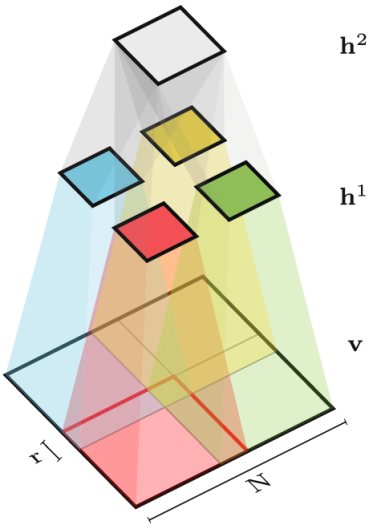


Figure 2.7: Illustration of a RBM with receptive fields. As indicated, the hidden units $\mathbf{h}^1$ all have their individual receptive fields, yet they overlap with each other (here, by r). Furthermore, there is an additional layer, $\mathbf{h}^2$, that is fully connected to all hidden neurons from layer $\mathbf{h}^1$. Figure reprinted from Eslami et al. (2014).

covered by the visible units it is connected to. A sketch of this architecture can be seen in Figure 2.7.

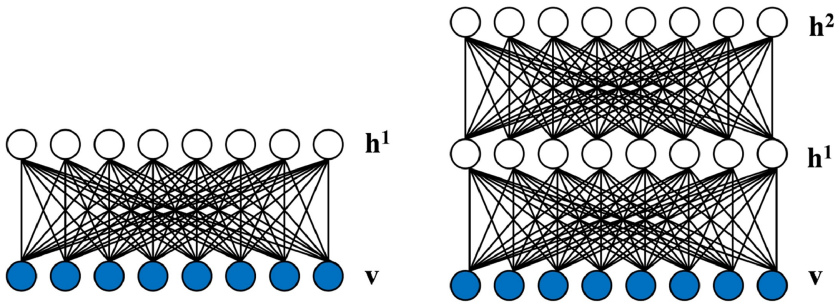
However, to combine the information from separate receptive fields again, the model needs a further extension. As illustrated in Figure 2.7, another layer $\mathbf{h}^2$ is added on top of hidden layer $\mathbf{h}^1$. Importantly, it is a fully connected layer, i.e. each of its units is connected to all units from the previous layer, just like in a standard RBM. This is essentially how Deep Boltzmann machines (DBMs) are constructed: multiple individual RBMs, with or without receptive fields, are stacked on top of each other (Hinton and Salakhutdinov, 2006).

2.2.5 Deep Boltzmann Machines

A DBM essentially consists of serially connected RBMs, where the hidden unit activations of one RBM function as a visible layer for the next in line. Eventually the DBM will have as many layers $\mathbf{h}^\ell$ as RBMs were pre-trained following this relay logic. A two-layer DBM is contrasted against a standard RBMs in Figure 2.8

The same restrictions for connections apply as in standard RBMs: adjacent layers are connected between each other, but there are no lateral connections within a layer. For instance, the visible layer $\mathbf{v}$ is only connected to $\mathbf{h}^1$, and $\mathbf{h}^1$ is wired to $\mathbf{v}$ and $\mathbf{h}^2$.

In essence, an RBM models the statistical dependencies and correlations in the



(a) RBM Model (b) DBM Model

Figure 2.8: Comparison of a standard RBM with one hidden layer and a DBM with two hidden layers. A DBM can be thought of consisting of multiple RBMs. Figure reprinted from Wu et al. (2018).

variables corresponding to the visible units or input data. Although standard one-layer RBMs are indeed capable of modeling higher-order correlations (Köster et al., 2014) – and with enough hidden units theoretically any discrete distribution (Le Roux and Bengio, 2008) – additional layers may help in finding a more efficient representation. The basic idea is that if there are still correlations in the latent units, they can themselves be modeled by further layers.

Strikingly, it is possible to train DBMs in a similar unsupervised learning algorithm as we have seen for RBMs (Salakhutdinov and Larochelle, 2010). The energy function of a two-layer DBM is given by adding a layer with its weights and biases (see Equation 2.11). Our parameter set ϕ would now consist of three sets of biases $\theta^v, \theta^{h^1}, \theta^{h^2}$ and two weight matrices, $\mathbf{W^1}$ and $\mathbf{W^2}$.

$$E^{\phi}_{\mathbf{v},\mathbf{h}} = -\sum_{i=1}^{n_v} \theta_i^v v_i - \sum_{j=1}^{n_{h^1}} \theta_j^{h^1} h_j^1 - \sum_{k=1}^{n_{h^2}} \theta_k^{h^2} h_k^2 - \sum_{i=1}^{n_v}\sum_{j=1}^{n_{h^1}} v_i h_j^1 w_{ij} - \sum_{j=1}^{n_k}\sum_{k=1}^{n_{h^2}} h_j^1 h_k^2 w_{jk} \quad (2.11)$$

Learning in DBMs generally follows the positive-negative algorithm, yet layer-wise: each layer of hidden units aims to find an appropriate representation of the distribution over the variables in its preceding layer in order to generate it. For example, the derivative of the log probability of a visible pattern $\mathbf{v}$ with respect to the weight w_{ij}^1 connecting visible neuron v_i with hidden neuron h_j^1 is now computed as (Salakhutdinov and Larochelle, 2010):

$$\frac{\partial \log p(\mathbf{v})}{\partial w_{ij}^1} = \langle v_i h_j^1 \rangle_{data} - \langle v_i h_j^1 \rangle_{model} \quad (2.12)$$

Yet again, exact maximum likelihood learning is intractable in these models (Salakhutdinov and Larochelle, 2010). But for DBMs, even classic CD learning is too slow and thus fails (Hinton et al., 2006). Instead, variational inference is used: a mean-field approach serves to estimate expectations with respect to the data in the positive phase (Salakhutdinov and Larochelle, 2010).

The variational distribution $Q(\mathbf{h})$ that shall approximate the true posterior $P(\mathbf{h}|\mathbf{v})$ is set to be a factorial $Q(\mathbf{h}) = \prod_i Q_i(h_i)$. We would then have such a variational distribution for each hidden layer ℓ (Salakhutdinov and Larochelle, 2010). Since all our neurons take on binary values, the parameterization of each variational distribution Q^ℓ requires only one mean-field parameter $\mu_i^\ell = Q_i^\ell(h_i^\ell = 1)$. Now the approximation is achieved through optimizing this parameter μ^ℓ with the goal of minimizing the variational free energy and thus the KL divergence between $Q(h^\ell)$ and $P(h^\ell|v)$ (Reichert, 2012). For each μ^ℓ there is a fixed mean-field equation involving input from the layers it is connected to. For example, for the first hidden layer h^1, we have (Salakhutdinov and Larochelle, 2010):

$$\mu_j^1 = \sigma\left(\sum_{i=1}^{n_v} W_{ij}^1 v_i + \sum_{j=1}^{n_{h^1}} W_{jk}^2 \mu_k^2\right), \tag{2.13}$$

where again σ denotes the logistic sigmoid function and μ_k^2 comprises the mean-field approximation of the units from the second layer. Each μ_j^ℓ is learned iteratively. The updates are computed layer by layer, but can be parallelized within each layer (Salakhutdinov and Larochelle, 2010). Mind that a μ_i^ℓ is set to the activation of a hidden unit h_i^ℓ, rather than sampling them given their activation probabilities. Thus, the joint learning is based on propagating activations through the network to compute the correlations in the positive phase. The negative phase is similar to before, except that PCD is used to compute the respective correlations (Reichert, 2012).

Approaching the problem of encoding high-dimensional sensory data into an abstract representation with more than one layer of non-linear processing is both theoretically and biologically intriguing (Salakhutdinov and Larochelle, 2010). Not only may it reduce computational complexity compared to representing the same data with a shallow, but wider network (Bengio et al., 2009), but it is also in line with theories of hierarchical sensory processing from cognitive and neuroscience (e.g. Clark, 2013; Sutton et al., 1988). Importantly, the brain circuits for early visual processing, parts of which we aim to model, also underlie a hierarchical organization with increasingly more complex representations (e.g. Hochstein and Ahissar, 2002). In short, the idea is

that sensory experiences can induce changes in these representation which themselves influence perception through top-down signaling (Ahissar and Hochstein, 2004). This is nicely reflected in the undirected graph architecture and unsupervised learning algorithm of RBMs and DBMs.

2.3 Computational Approaches to Pruning

Pruning and subsequent network reduction is intriguing not only from a computational neuroscience perspective. For instance, the hope for increased efficiency in the light of limited computational resources has traditionally motivated the search for smaller models and adequate pruning techniques in the machine learning community (e.g. Reed, 1993). But not only may a reduction in the number of parameters decrease space and time complexity, it is also desirable in order to prevent over-fitting. That is because over-parameterized models tend to remember their training data while not generalizing well to unseen data. On the other hand, too few parameters may prevent the network from learning the relevant structure of the data (Hassibi et al., 1993). Similar to what we observe in the brain, there seems to be a sweet spot between over- and under-parameterization of ANNs.

The investigation of the time course of learning and the emergence of important and unimportant parameters is closely related to this research area. For instance, Achille et al. (2017) disrupted the learning of ANNs by perturbing the input at different points during training. Similar to critical periods in natural brain development, early disruptions were more harmful than later ones. Yet we do not want to simulate sensory deprivation. Instead of modifying the input to the network, we want to manipulate its "hardware" by selectively removing synapses and neurons. This is the logic behind computational pruning methods. Generally, they start by building a large model that has satisfactory performance. Afterwards, the network size is reduced by removing weights and/or complete units (see Figure 2.9). The main challenge then is to identify the parameters to be removed, acknowledging their importance for a given task is indeed heterogeneous (Kirkpatrick et al., 2017).

The exploratory nature of pruning experiments reveals how little is currently known about the role of individual parameters in ANNs. For instance, Morcos et al. (2018) recently investigated the effect of removing units that mainly respond to one category of images such as cats and are thus easily interpretable (so-called "cat-neurons"). Yet they found that these are not generally more important than units with more diffuse

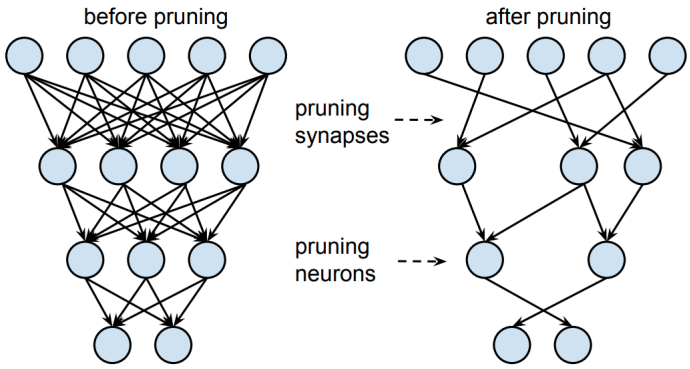


Figure 2.9: A network is pruned: both weights and unconnected neurons can be removed. Figure reprinted from Han et al. (2015).

activation patterns. Others aimed for diversity of the hidden units by merging the ones that were similar to each other (Mariet and Sra, 2016). However, it is unclear how a biological neuron should compare itself to, let alone merge itself with others since the strengthening and weakening of biological synapses happens locally, i.e. without considering the global network structure and performance each time.

In supervised classification settings, it is a logical approach to remove units with least discriminative power. Based on the notion that a unit with constant activation independent of the input does not contribute to distinguish between categories, Sánchez-Gutiérrez et al. (2017) aimed to identify the most discriminative units for a classification task. They used an RBM as a feature detector by feeding its hidden layer representation to a classifier. Using various discriminative measures such as Mutual Information (MI), they checked how different the activation of a given hidden unit was between classes and removed the ones that discriminated least. More than half of the hidden units could be removed without increasing the error. Yet again, a pruning rule depending on the performance in a supervised classification task is biologically implausible. After all, brain plasticity is unsupervised.

Berglund et al. (2015) on the other hand were interested in how much input information is encoded in each hidden unit of an RBM, irrespective of a classification objective. They measured the relevant activity of a particular hidden neuron by computing the MI between a training vector and each hidden unit. But apart from MI not being a locally available statistic, we understand that synaptic pruning precedes the death of a neuron in a biological network. We would thus also want to be able to prune individual weights in addition to removing units.

A simple metric to decide which connections to prune may be by the magnitude of

weights (e.g. Han et al., 2015; Tostado et al., 2017). However, this makes the strong and potentially naive assumption that magnitude equals importance (LeCun et al., 1990). In fact, the most sophisticated and theoretically justified pruning techniques may be among the earliest ones: both Optimal brain damage (OBD) (LeCun et al., 1990) and Optimal brain surgeon (OBS) (Hassibi et al., 1993) are information theoretic approaches to network reduction making use of Fisher Information (FI).

2.4 Fisher Information

OBD and OBS are based on the idea that the importance of parameters can be estimated by slightly changing their values and re-evaluating an error function. Importance then equates to the effect it has on said error. This can be locally approximated by the Hessian of the objective function, which tells us about the curvature with respect to small parameter changes. The Fisher Information Matrix (FIM) gives us exactly this. While OBS computes the full Hessian, that is, the change in the error with respect to each parameter pair, OBD makes the simplifying assumption of a diagonal matrix. That is because computing only the diagonal elements is computationally less expensive.

Only recently, Kirkpatrick et al. (2017) revived the usefulness of FI for indicating parameter importance in order to overcome catastrophic forgetting in ANNs. For each weight, they included its respective diagonal FIM entry into the regularized loss function. This protected important parameters from being overwritten when learning tasks sequentially. Yet employing a global loss function and adjusting parameters with respect to this through backpropagation opposes the basic principles of neural information processing in the brain. In a similar approach to Deistler et al. (2018), who found a locally computable alternative to attenuate forgetting in Hopfield networks, we want to transfer FI motivated pruning to the local regime in RBMs and DBMs.

Rule et al. (2018) provide us with the framework for our pruning experiments. They derived the FIM and local computation of its diagonal entries specifically for RBMs. As explained above, Boltzmann learning is an unsupervised learning algorithm which aims to minimize the difference between the data distribution and the model distribution (see Equation 2.5). Now the FIM locally approximates the KL divergence for the model distribution with the current parameter set ϕ and a model with slightly modified parameters ϕ_i and ϕ_j. An entry of the FIM for an RBM with a visible and a

hidden layer has the form (Rule et al., 2018):

$$F_{ij}(\phi) = \sum_{v,h} P_{v,h} \frac{\partial^2 E_{v,h}}{\partial \phi_i \partial \phi_j}, \tag{2.14}$$

Rule et al. (2018) expanded the derivatives, which involves averaging over the model distribution $P(\mathbf{v},\mathbf{h})$. Again, this can be computed through sampling. If an FIM entry tends towards zero, the involved parameters can be traded exactly for one another, i.e. they are redundant. Indeed, Rule et al. (2018) showed that the FIM becomes sparse with increasing hidden layer size, so more and more parameters were irrelevant. Furthermore, the important weights and biases tended to line up with a few overall highly sensitive, so-called "stiff" (e.g. Daniels et al., 2008; Gutenkunst et al., 2007), latent units.

Importantly, the diagonal entries that indicate the sensitivity of a single parameter are indeed locally available, i.e. for an individual neuron to potentially compute on its own (Rule et al., 2018). The importance of a bias is approximated by the variance of the activation of the neuron:

$$\begin{aligned} F_{b_i^v, b_i^v} &= \sigma_{v_i}^2 \\ F_{b_i^h, b_i^h} &= \sigma_{h_i}^2 \end{aligned} \tag{2.15}$$

The importance of a weight may be computed as the covariance of the activation of a visible v_i and a hidden neuron h_j:

$$F_{w_{ij}, w_{ij}} = \langle v_i^2 h_j^2 \rangle - \langle v_i h_j \rangle^2 \tag{2.16}$$

This corresponds to tracking the pre- and postsynaptic firing rates as well as their coincidences. It has recently been suggested that neurons are indeed capable of computing this through multiple synapses (Hiratani and Fukai, 2018). We thus argue that this parameter-wise computation of FI for a weight can indeed serve as a local measure of sensitivity of a synapse. Furthermore, the average FI of all parameters characterizing one latent unit, i.e. its bias and weights, may indicate the overall importance of a neuron.

Admittedly, the diagonal of the FIM may only be an approximation of parameter sensitivity. This also motivated the criticism of OBD which assumes the FIM to be a diagonal matrix. Yet, this is only true if the ellipse in parameter-space described by the FIM aligns with the parameter axes. As Hassibi et al. (1994) warned, it may lead

to pruning away the wrong parameters and introducing noise into the system through required retraining. Still, since Rule et al. (2018) observed a homogeneous separation into important and unimportant units even when computing the full FIM, we mostly rely on the heuristic of just computing the diagonal.

2.5 Outline

We now want to continue the size-accuracy trade-off experiments by Rule et al. (2018). Instead of building and comparing separate RBMs of different sizes, we begin with an initially over-parameterized model and use the local computation of FI to iteratively reduce its size. This could potentially simulate aspects of natural synaptic pruning and apoptosis elicited in an activity-dependent manner and support the hypothesis that certain connections and neurons are not needed for the precise encoding of stimuli.

Given that the majority of parameters have been demonstrated to be redundant in ANNs (Denil et al., 2013) and the fit of an RBM saturates with increasing hidden layer size (Rule et al., 2018), we expect that we can prune a significant amount of weights and units without detrimental effects on the fit. The point before the fit decreases would then mark the optimal model size.

Yet as the title of the thesis suggests, we further aim to over-prune the model to a suboptimal size. In a comparable approach to Hinton et al. (1993) who investigated reading impairments due to over-pruning, we want to simulate damage to visual brain circuits while monitoring the model's encoding and generative performance.

Chapter 3

Methodology

In our experiments, we iteratively reduced the size of initially over-parameterized models of visual encoding. RBMs as well as DBMs were used. We tried to identify and remove irrelevant parameters in these models through the local computation of FI. This section explains the details of implementation, training and evaluation methods, the results of which shall be addressed in the following chapter.

3.1 Datasets

Generally, separate experiments were conducted on two different datasets both of which have their caveats for evaluation (see Section 3.4).

First, continuing the study by Rule et al. (2018), $90,000$ circular patches of different sizes were randomly selected from images of the CIFAR-10 dataset (Krizhevsky and Hinton, 2009). This approach is based on the notion that natural images are generally not completely random patterns, but show typical statistical properties and correlations (Field, 1987). The smaller circles then ought to resemble receptive fields of sensory neurons. The images were converted to gray scale, quantized and binarized according to their median pixel intensity.

Second, the handwritten digits dataset MNIST (LeCun et al., 2010) was used. The square 28×28 images were downscaled to a 20×20 pixel version by cropping two pixels on either side. This led to a visible layer size of 400 units. The gray scale digits were binarized according to their mean pixel intensity. The training set consisted of $60,000$ and the held-out test set of $10,000$ images from ten different categories $(0-9)$.

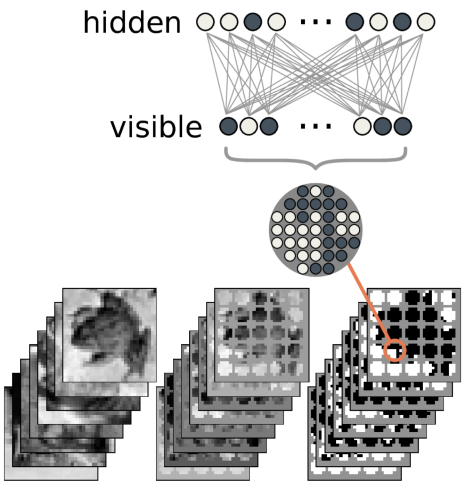


Figure 3.1: Circles of different radii were sliced out of the original CIFAR-10 images. Data was binarized and served as input to the visible units of an RBM. Figure reprinted from Rule et al. (2018).

3.2 Model Fitting and Sampling

All models were implemented in TensorFlow (Abadi et al., 2015) making use of open-source code (`github.com/monsta-hd/boltzmann-machines`). Alternating Gibbs sampling ran on NVIDIA GeForce GTX 980 GPUs. A state of each layer was stored as a sample after every 200^{th} Gibbs step for them to be uncorrelated. The number of samples depended on the training set: we always sampled as many states for a layer as there were training vectors. For DBMs the sampling propagated through all layers, entailing feedforward and feedback influence of the neighbouring layers. Again, the state of the whole machine, i.e. of all layers, was kept as a sample after every 200^{th} Gibbs step.

3.2.1 Fitting RBMs to CIFAR-10

Single-layer RBMs were fitted to $90,000$ CIFAR-10 circles of different sizes using CD-1. The radius of the circles determined the number of pixels and visible units (see Figure 3.1). Weights were initialized randomly from a zero-centered Gaussian with a standard deviation of 0.1. The number of hidden units varied depending on the visible layer size and the number of preceding pruning events. In any case, we started with a larger hidden than visible layer since we aimed for sparse representations and uncorrelated latent units. Initializing all hidden biases to -2 was another means to

encourage sparseness in the firing of hidden neurons. Other than that, we did not define a sparsity target.

As recommended by Hinton (2012), visible biases were initialized to $\log(p(v_i)/(1-p(v_i))$ where $p(v_i)$ stands for the fraction of training images where neuron i was in the on-state. We did not use mini-batches, thus updates were applied after each training instance. The models were first trained for two epochs. The learning rate was set to 0.1 for the first epoch and 0.01 for the second. Momentum was set to 0.9. We did not apply any L2-regularization.

3.2.2 Composing a DBM for MNIST

The comparably large number of 400 pixels of each MNIST image necessitated the use of receptive fields to prevent the mean activation of the hidden units from becoming prohibitively low. However, since we aimed to use the complete encodings of the digits for evaluation purposes, we could not select smaller areas from the images without re-combining the information again. Thus an additional fully-connected layer was required, resulting in a DBM (see Figure 3.2). This architecture also accounted for desirable near-zero correlations and sparse activations in the final hidden layer.

The first hidden layer had the same number of units as the visible layer (400). Each hidden unit can be thought of as having its small rectangle window out of which it observes the input. It is blind to the remaining visible units outside of this window. The receptive fields cover the input pixel by pixel, with each pixel being the center once. That way, the receptive fields of different hidden units overlapped with each other. Two neighbouring pixels to either side were selected leading to the maximal shape of 5×5. This also led to the receptive fields being smaller at the borders of the image. They can also be thought as being padded with zeros.

Two individual RBMs were pre-trained for 20 epochs using CD-1. The first RBM was trained on the $60,000$ binarized, downsized MNIST training images. Each of its 400 hidden units had a receptive field as explained above. This led to a reduction of originally $400 \times 400 = 160,000$ to $8,836$ weights. After training the first RBM, its hidden unit activation probabilities were computed for each training image using Equation 2.8. This can be thought of as a forward-pass through the layer, with the visible units clamped to a training vector at a time. The hidden unit activation probabilities were converted to binary values and served as input to the second RBM which had 676 fully connected hidden units (see Figure 3.2).

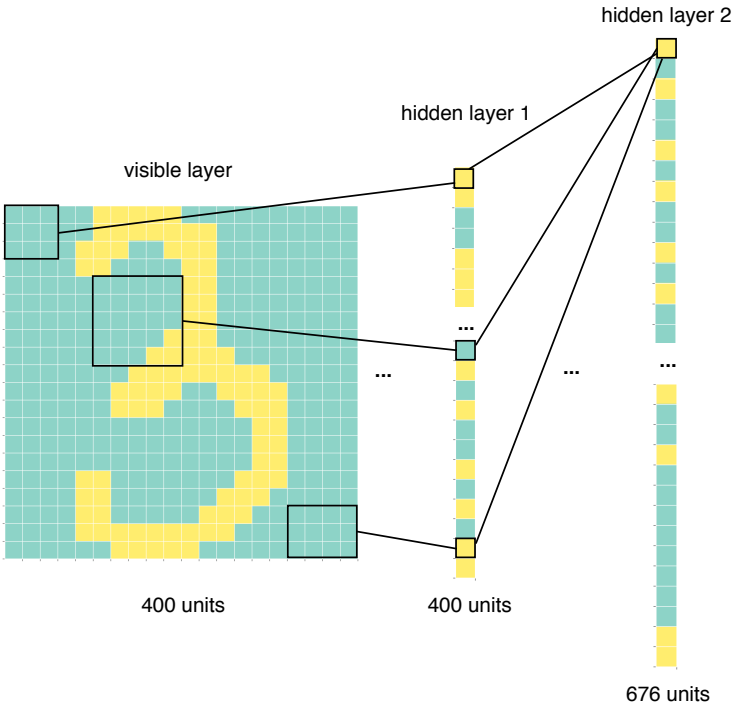


Figure 3.2: DBM architecture. The first hidden layer had the same number of units as the visible layer. All units were binary. Here, yellow indicates a unit being in the on-state while turquoise corresponds to an off-state. The black rectangles surrounding small areas of the input image represent the receptive fields. A unit from hidden layer 1 was connected by individual weights to each of the pixels/visible units located inside its receptive field (not shown in this scheme for clarity reasons). The maximum shape of the receptive fields was 5×5 in the center of the image. At the borders they were smaller. The second hidden layer had 676 units and was fully connected, i.e. each of its units had connections to each of the units in the previous layer. Only few connections are indicated due to clarity.

Most hyperparameters for the two RBMs were the same: again, weights were initialized randomly from a zero-centered Gaussian with a standard deviation of 0.1. All hidden biases were initialized with -2 and the visible biases with -1. No sparsity targets and costs were defined. Momentum was set to 0.5 for the first 5 epochs and then increased to 0.9 as recommended by Hinton (2012). The learning rates for the first RBMs were reduced evenly with each epoch on a logscale starting at 0.01 and ending at 0.0001. The same procedure was used to reduce the learning rates for the second RBM, except that the first rate was 0.1, evenly decreasing to 0.0001. Again, no mini-batches or L2 regularization was used.

The two RBMs served as building blocks for a DBM. Since our model consisted of only two hidden layers, their weights did not need to be halved which is required for intermediate layers in DBMs with more than two layers (Reichert, 2012). The biases of the hidden units of the DBM were computed by averaging the hidden and visible biases of the respective RBM layers. The DBM was then trained jointly for 20 epochs following the mean-field variational inference approach described in Section 2.2.5 as an approximation of the positive phase. In the negative phase, PCD was run for 100 persistent Markov chains. For this, the visible particles were initialized to the training examples and the hidden particles to binary hidden unit samples of the respective RBMs, given each training example. Mean-field updates to each μ^ℓ (see Equation 2.13) were applied until a maximum number of updates of 50 was reached or until the update fell below a threshold of 1×10^{-7}. Parameters were updated immediately after computing the positive and negative phase for a training image, i.e. no mini-batches were used. A maximum norm constraint of 6 was defined in order to keep the absolute values of the weights in a reasonable range and prevent extreme values. A decay rate of 0.8 was defined for the hidden unit firing probabilities. Neither a sparsity target nor L2 regularization were applied.

3.2.3 Justification of Architecture and Hyperparameters

Generally, fitting RBMs requires a certain expertise, which also motivated Hinton (2012) to write a practical guide for training them. Many decisions were thus inspired by his recommendations. The architecture of the DBM with receptive fields was deemed necessary after unsuccessfully fitting standard RBMs and DBMs to MNIST. Yet it was also inspired by Reichert (2012) who used a similar architecture.

The MNIST digits were cropped to a 20×20 version since we approximated the

FIM diagonal by sampling. Due to limited GPU memory, less visible and consequentially less hidden units were desirable. The removal of two pixels on each size almost halved the size of the visible layer compared to when using the original MNIST version.

Mind that hyperparameters for MNIST were not tuned in order to optimize a classification performance on a development or test set. Rather, we checked if an individual RBM was able to approximate the distribution of its input data using various heuristics as described in Section 3.4.1.

3.3 Pruning Procedure

The pruning experiments can broadly be divided into two different categories. Either we directly removed hidden units, or weights were pruned which may or may not lead to certain hidden units being entirely unconnected and thus removable. In both cases the FIM provided us with the criterion to identify the weights or units to delete.

3.3.1 Pruning Criteria

As explained in Section 2.4, the estimation of the FIM entries relies on sampling the states of the visible and hidden layer, corresponding to tracking the pre- and postsynaptic firing rates of neurons. Generally, the FIM is a square matrix of order $(n_v + n_h + n_v \times n_h)$, where n_v stands for the number of visible units and n_h counts the hidden units. For small models with a limited number of units, computing the full FIM was feasible. That way, we could also compute the first eigenvector of the matrix which served as a pruning criterion for our first experiments with RBMs that were fit to CIFAR-10 patches. An eigenvector had n_v entries for the visible biases, n_h entries for the hidden biases and $n_v \times n_h$ entries for the weights. The code for this was available from (`github.com/martinosorb/rbm_utils`).

As the size of the visible and hidden layers increased, the full FIM was not feasible anymore. Instead, the given code was adapted in order to compute only the diagonal elements of the matrix, following the OBD paradigm (LeCun et al., 1990). For all $n_v + n_h$ biases and $n_v \times n_h$ weight values, the respective diagonal entry was directly used as an estimate of their relative importance. For the DBM, we computed the FIM diagonal layer-wise assuming weakly correlated units in the hidden layers.

Since it is still not certain if neurons can track the coincidences of pre- and post-

synaptic firing rates which would be necessary to approximate the diagonal entries of the FIM, we further aimed at finding an alternative local indicator of weight sensitivity. Rule (personal communication, July 2018) derived an additional local estimate of weight importance. Using a mean-field approach and assuming independence in the hidden units, the derivation shows that the diagonal FIM entry for a weight can be approximated by the synapse strength W_i and the mean firing rates $\langle v \rangle$ and $\langle h \rangle$, without tracking their coincidences $\langle vh \rangle$ directly:

$$\langle vh \rangle \approx \langle h \rangle \, \sigma \left(\sigma^{-1} \left(\langle v \rangle \right) + W_i (1 - \langle h_i \rangle) \right) \tag{3.1}$$

The full derivation can be found in Appendix A. This heuristic estimate was used as an alternative weight pruning criterion for the joint pruning of the DBM fit to MNIST.

Depending on the experiment, we also compared the two FI inspired pruning criteria to random pruning and/or removing the *most* important units or weights, which we call anti-FI pruning.

3.3.2 Removal of Hidden Units

For RBMs trained on CIFAR-10 circular patches of limited size, the sensitive parameters tended to line up with a few latent units. Thus, the overall importance of a unit was computed as the maximum of the sum of the FIM diagonal entries for all its weights versus the entry for its bias. The hidden units were then ordered according to this metric and a set number of highly sensitive units was kept while the unimportant units were removed.

The RBM was re-initialized with the remaining number of hidden units and unchanged biases of both visible and latent units. The weights were scaled by the ratio of the number of previous hidden units and the number of current hidden units. Its fit was evaluated as described in Section 3.4.1 and the RBM was retrained for two epochs with the same hyperparameters as before.

3.3.3 Weight Pruning

Synaptic pruning was simulated by setting selected weights to zero and masking them during further training (see Section 3.3.5). Depending on the experiment, weights that did not meet a pre-defined FI threshold or a percentile of all weights in a hidden layer were removed. This will be explained in detail along with the results of a particular experiment (see Chapter 4). Whenever we pruned weights, we refrained from adjusting

the remaining parameters. That is because the adjustment of the weights by lost hidden units was not applicable if the pruning did not lead to a removal of units. Hence the pruned model was initialized with the remaining weights and biases from immediately before pruning. The repeated weight pruning alternated with re-training phases, which was deemed an automatic re-adjustment of parameters.

3.3.4 Pruning Schedules for DBM

The DBM confronted us with the choice of which layer to prune first. Since we further wanted to explore the direction and loci of pruning, we designed three different pruning schedules: Either the pruning affected one layer at a time or both layers were pruned at the same time.

The first schedule started by pruning the first layer, after which the second layer was pruned. The second schedule followed the same logic, but in the opposite direction, i.e. starting with the pruning of the second followed by the first layer. In both cases, the network was evaluated immediately after each pruning event. Furthermore, the network was retrained after each pruning of either layer. Like this, the network was retrained two times and evaluated four times in each of ten pruning iterations or sessions:

$$\text{[Prune layer]} \rightarrow \text{[Evaluate]} \rightarrow \text{[Retrain]} \rightarrow \text{[Evaluate]}$$
$$\rightarrow \text{[Prune other layer]} \rightarrow \text{[Evaluate]} \rightarrow \text{[Retrain]} \rightarrow \text{[Evaluate]} \rightarrow \text{[Repeat]}.$$

Unconnected hidden units in the last layer were removed, while unconnected units in the intermediate layer remained in the network. This was convenient regarding space complexity since the DBM is built from individual RBMs that had to be re-initialized after each pruning event. Without deleting units in the intermediate layer, the TensorFlow graph of the first RBM could be re-used.

In the third schedule both layers were pruned at the same time, i.e. based on the same samples. The model was evaluated, retrained and re-evaluated:

$$\text{[Prune both layers]} \rightarrow \text{[Evaluate]} \rightarrow \text{[Retrain]} \rightarrow \text{[Evaluate]} \rightarrow \text{[Repeat]}.$$

We also refer to this schedule as “joint pruning”. Since this schedule was more efficient regarding time and space complexity, unconnected hidden units from both hidden layers were removed, even if they were unconnected from just one neighbouring layer. The remaining biases and weights were initialized with their parameter values from immediately before pruning.

3.3.5 Implementation of Weight Pruning and Receptive Fields

The open-source code from `github.com/monsta-hd/boltzmann-machines` was extended in order to allow for receptive fields and weight pruning. Both architectural modifications target the restriction of connections and were thus implemented in a similar way. The trick is a simple element-wise multiplication of the weight matrix with a Boolean mask that indicates which connections to keep $(= 1)$ and which ones to delete $(= 0)$. When creating an RBM object, one can now specify a Boolean mask that freezes the indicated weights to zero. This corresponds to the pruning of a synapse between two neurons. The mask needs to have the same size as the weight matrix in order to be active. Otherwise, a mask of ones is created, leading to the weights being updated as in the training of a default, fully connected network.

One can also specify the desired shape of the receptive fields. The program then creates a Boolean mask by iterating over the weight matrix for each hidden unit and masking all weights to visible units that are not located inside its receptive field.

Both Boolean masks are multiplied with the matrices containing the weight updates during training. That way, indicated weights remain zero and no connection is established.

3.4 Experimental Design and Evaluation

Generally, we started with fitting a sufficiently large model to a fixed visible layer size of either CIFAR-10 circular patches or MNIST digits. For both datasets, the learning followed the positive-negative algorithm and thus was completely unsupervised. Assuming the initial model was over-parameterized, we reduced its size through pruning selected hidden neurons and/or weights according to their FI while monitoring the model fit. Unfortunately, evaluating the fit of RBMs is not obvious and contributes to the difficulties of training them.

3.4.1 Comparison of Distributions

For CIFAR-10, the only way to evaluate the fit of the RBM was to compare aspects of the data distribution with the model distribution $P(v,h)$, estimated from sampling states of all units.

First, we compared the mean estimated probability of a visible unit being active $\langle P(v_i)\rangle$ according to the data versus the model distribution. This corresponded to av-

eraging over the states of a particular pixel in the training images versus the samples of the respective visible unit. We further compared the covariance between each two visible units over all data images versus all samples.

Second, the samples of the visible layer were examined as a whole. We compared the frequency of generated patterns with the frequency of actual data patterns.

3.4.2 Encoding and Generative Performance

The above stated heuristics to compare the data and estimated model distribution served as sanity-checks when searching for an appropriate model architecture to train on the MNIST dataset. Importantly, the labels of the digits were not used to find a suitable fit of the initial RBMs and the DBM. They were only used for evaluation purposes between pruning events.

Metaphorically, we can look at the DBM from two sides in order to evaluate its fit (see Figure 3.3). Beginning from the side of the visible layer, the primary learning objective of RBMs and DBMs is to approximate the data distribution with the model distribution. Hence it is a valid approach to evaluate the fit of the machine by looking at the patterns it generates from sampling from its model distribution. For MNIST, these should ideally be digit-like patterns. In order to have a somewhat objective evaluation measure, an off-the-shelf classifier from scikit-learn (Pedregosa et al., 2011) was trained on the original binarized and downsized 20×20 digits. Specifically, we used a multinomial logistic regression classifier minimizing the cross-entropy loss using the 'sag' solver and an L2 regularization of 1. The DBM-generated images were fed to this classifier which returned a probability for an image to be an instance of either digit class. The maximum of these was chosen to be the winning class and the probabilities were interpreted as confidence in a decision. For each class, the mean confidence was computed by averaging over the winning probabilities for the samples ending up in that class. We refer to the average of these values as the quality of generated digits.

Furthermore, we counted the samples ending up in each class in order to monitor the diversity of generated images. A diversity score was computed as the minimum number of counts among the ten classes divided by the maximum number of counts. If the classes were rather balanced, this score would approach one. Very imbalanced classes would manifest themselves in a score approaching zero, which is the minimum if a class was not represented in the samples at all and thus had a zero-count.

Looking at the hidden layer, RBMs and DBMs encode the data into a set of latent

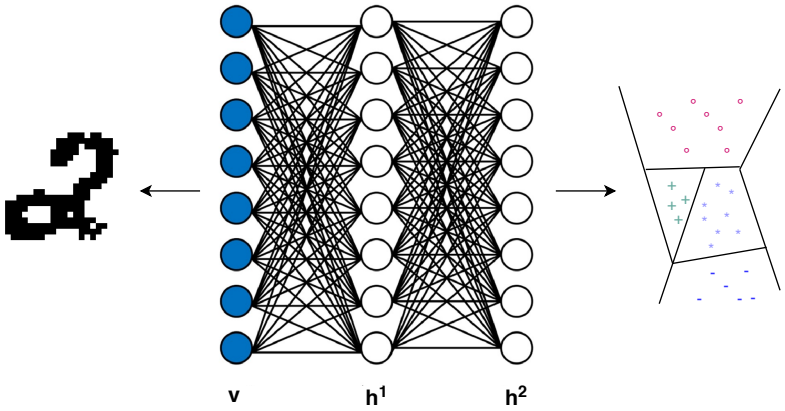


Figure 3.3: The DBM fit to the MNIST training set was evaluated from two sides. First, as indicated on the left side, the generated digits were assessed regarding their resemblance to handwritten digits. Second, the encoded representations in the latent units of the final hidden layer $\mathbf{h}^2$ were used to train a classifier instead of using the original digits. The classification accuracy then served as a measure of the model's encoding performance. Both performance measures were monitored during the iterative pruning experiments. DBM illustration in the centre adapted and reprinted from Wu et al. (2018)

variables. Similar to an auto-encoder approach, the post-learning latent activations of the DBM may be more useful features to train a classifier than the raw images. The activation probabilities of the units in the final hidden layer were computed by a forward-pass through the network, with the visible units clamped to one image at a time. At each evaluation, we trained a new classifier on the latent encodings of the $60,000$ training images. In fact, two classifiers were built each time: first, a logistic regression classifier with the same parameters as the one trained on the raw digits served as an exemplary linear classifier. Second, a Support vector machine (SVM) with a radial-basis function kernel and a penalty term of $1,000$ was used as an exemplary non-linear classifier. Afterwards, the classification accuracy on the encodings of the $60,000$ images was computed for each classifier. We refer to these accuracy scores as the encoding quality.

In order to monitor the fit of the model, we defined two baselines to compare to. The first one was the performance of the initial unpruned DBM regarding the two evaluation measures. A second baseline was the performance of classifiers trained on the raw digits. For the generative performance, this was the confidence about the $10,000$ handwritten test digits to belong to a certain class. For the encoding performance, it was the accuracy score on the test set of a comparable classifier trained on the original

raw images. As a side note, we were less interested to improve the current state of the art or set a new benchmark in this classification problem. Rather, the admittedly simple classifiers were a means to monitor and evaluate the fit of the DBM.

3.4.3 Minimal models

As a final experiment, we built so-called minimal models and evaluated them regarding their encoding and generative performance. We refer to a minimal model as a newly trained DBM with its weights masked and the number of hidden units set according to the final state of a model pruned for ten iterations. The two RBMs making up the minimal DBM were initialized with default hyperparameters (see Section 3.2.2), except with a pre-defined weight mask that was recovered from the final state of the pruned DBM. After pre-training both of the RBMs for 20 epochs, they were entered as building blocks for the DBM which was then jointly trained for 20 more epochs, again with the same hyperparameters as we used for the joint training of the initial DBM.

Chapter 4

Experiments and Results

The results are broadly divided into two sections. First, we report on our exploratory experiments with RBMs that were fit to circular patches cut out of CIFAR-10 natural images. The second section covers the pruning of a DBM that was fit to images of MNIST handwritten digits.

4.1 Explorations with RBMs on CIFAR-10

As mentioned before, fitting RBMs requires a certain expertise. Furthermore, the goodness of fit of a model is not directly available in a single metric. These first exploratory experiments on the CIFAR-10 dataset thus mainly served the familiarization with the model class in order to prepare building a DBM.

4.1.1 Full FIM vs. FIM Diagonal

Our first experiments started with the training of RBMs of different hidden layer sizes to CIFAR-10 circular patches, in order to have comparable stimulus material to Rule et al. (2018) who mainly inspired this work. We were able to replicate the clear separation into important and unimportant latent units for small visible layer sizes. Figure 4.1 shows this for an RBM that had 40 hidden units and was trained on 90,000 circular patches of 13 pixels for two epochs (see Section 3.2.1 for more information on hyperparemeters).

Two main observations can be drawn from this plot. First, the latent units were clearly separable into so-called “stiff” vs. “sloppy” units (e.g. Daniels et al., 2008; Gutenkunst et al., 2007): There were only a few stiff units with overall high impor-

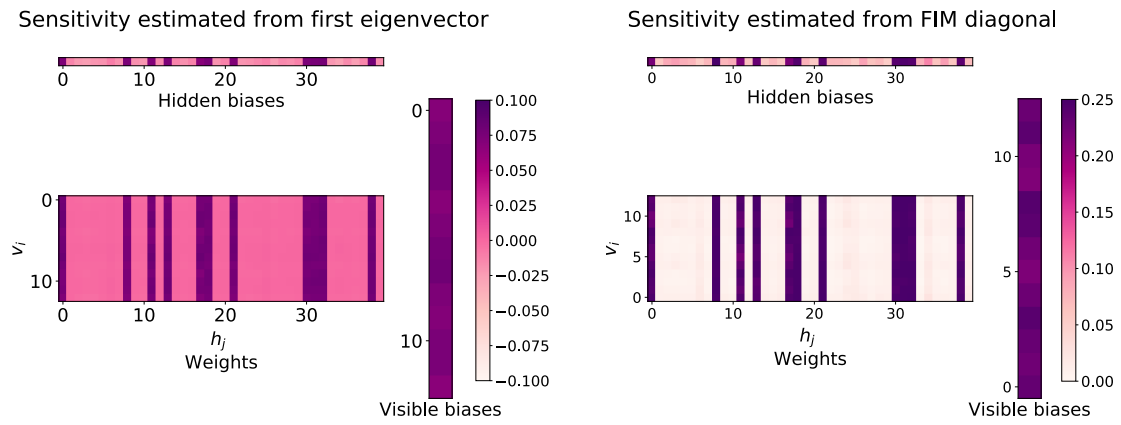


Figure 4.1: Comparison of parameter-wise sensitivity estimated from the first eigenvector of the FIM vs. its diagonal. Mind that the two measures have different ranges. Yet we see that they mostly agree with each other. There is a clear separation in sensitive and less sensitive latent units.

tance, i.e. their bias and all their weights had relatively high sensitivity. On the other hand, the remaining latent units had low sensitivity and may thus be seen as candidates for removal.

Second, we can identify these stiff and sloppy units from both looking at a metric derived from the full FIM, that is from its first eigenvector, and from its diagonal. This confirmed us in our approach of pruning based on the latter. Noting that this model was trained for only two epochs, we can also conclude that the directions of high sensitivity appear early during learning, potentially allowing early pruning as well.

We also observed a strong relationship between the average hidden unit activity $\langle p(h_j = 1) \rangle$ and the overall importance of a unit computed as the mean FI of all its weights ($r^2 = 0.992$ if the importance was computed based on the first eigenvector, $r^2 = 0.989$ if it was computed based on the FIM diagonal). The median hidden unit activity was 0.084 with a range between 0.056 and 0.684, indicating sparse representations which is desirable for a biologically plausible model of visual information processing (e.g. Olshausen and Field, 1997). Importantly, the correlation between the absolute value of a weight and its sensitivity was much lower ($r^2 = 0.473$ if estimated from the first eigenvector, $r^2 = 0.480$ if FIM diagonal was used). This observation questions a pruning procedure based on parameter magnitude as pursued by e.g. Han et al. (2015) or Tostado et al. (2017). Rather it confirmed the argument of LeCun et al. (1990), namely that magnitude does not necessarily equal sensitivity.

4.1.2 Removal of Hidden Units from RBMs

In our first pruning experiments, we removed half of the hidden units from an RBM that initially had 100 latent units and was fit to input patterns of 13 pixels for 2 epochs (see Section 3.2.1 for details on hyperparameters). The overall sensitivity of a unit was computed as the maximum of the sum of the FIM diagonal entries for all its weights and the respective entry for its bias. The hidden units were then ordered according to this metric and a set number of 50 "stiff" units was kept while the less important or "sloppy" units were removed. We refer to this as the FI motivated pruning.

The RBM was re-initialised with the 50 remaining hidden units and unaltered biases. The opposite pruning criterion was used as well, resulting in the removal of the *most* important units. That is, we used the inverse of the diagonal to order the units. We refer to this as anti-FI pruning. Lastly, we compared these two pruning criteria to random pruning.

In all cases, the weights were scaled by the ratio of the number of previous hidden units and the number of current hidden units. In this particular case, this led a multiplication of the weights by the factor 2. This was done to compensate for the loss of activation of the removed latent units. The number of synapses a neuron has with others is a locally available statistic, thus conforming to biological plausibility. After pruning, the RBMs were retrained for two epochs with the same hyperparameters as we used for training the initial model.

Figure 4.2 compares the parameter values of the model immediately after FI pruning and after refitting. The weight values generally did not change drastically. Yet

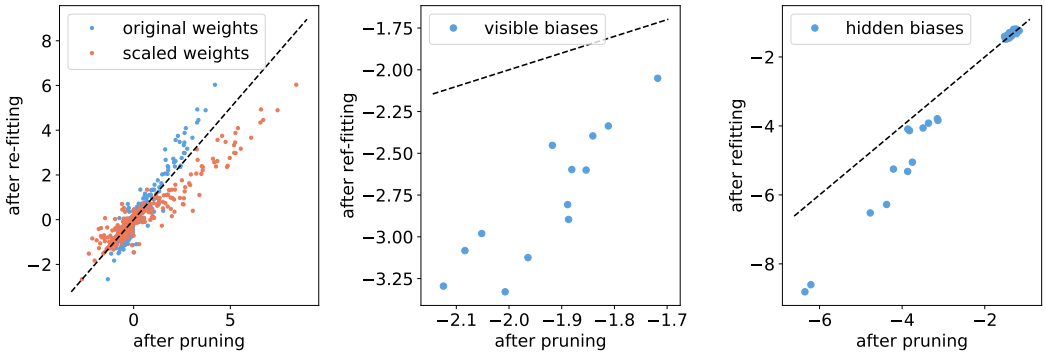


Figure 4.2: Comparison of parameter values of an FI pruned RBM before and after retraining. The weights were scaled by the ratio of number of hidden units (number lost)/(number before pruning). The scaling is compared to initializing the pruned RBM with unaltered weights (original).

we see that the naive re-scaling of the weights gives a valid correction for the slightly negative and near-zero weights, but deviates for larger positive ones. Apparently there is a more complicated non-linear effect that has to be taken into account when correcting for the loss of hidden units. However, we argue that a retraining phase may be seen as an automatic way of re-adjusting parameters as part of a self-reorganization of the network. It is also plausible from a biological perspective. After all, the sensory systems are continuously exposed to stimuli. The induced activity alters both perception and representation mechanisms (Ahissar and Hochstein, 2004), corresponding to a retraining.

The visible biases generally decreased during retraining. This makes sense considering sensitivity was highly correlated with the mean activity of latent units. Consequentially FI pruning protected the most active hidden units in the network. In order to compensate for the loss of inhibiting input from mostly silent hidden neurons, their biases became more negative. This protected the network from becoming overly active.

The fit of all pruned RBMs was evaluated immediately after removal and after retraining. Figure 4.3 shows a rank-frequency plot of the actual data patterns $\mathbf{v}_{data}$ vs. the sample patterns $\mathbf{v}_{model}$ generated by each model. Ideally, they should match as closely as possible. For the CIFAR-10 dataset we generally observed the data distribution to follow Zipf's law, i.e. the probability of patterns was approximately proportional to $1/rank$.

As can be seen, the frequencies of the generated patterns from the original model matched the frequencies of data patterns well. Immediately after pruning, the fit de-

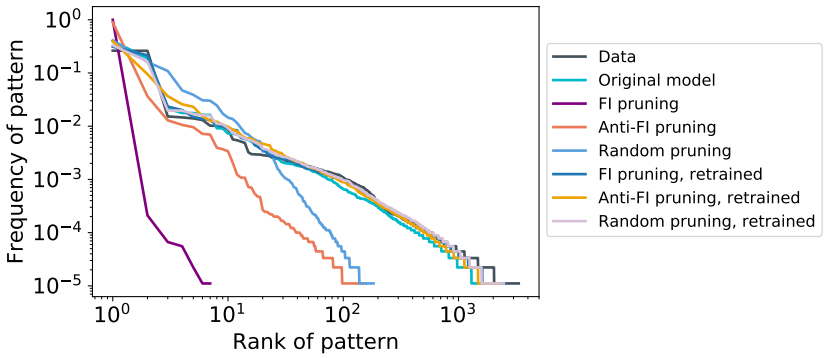


Figure 4.3: Rank-frequency plot of data patterns and generated patterns from the different models. The same original RBM with 100 hidden and 13 visible units was pruned based on different criteria. Either the 50 most important (FI pruning) or the 50 least important (anti-FI pruning) or 50 random units were removed.

teriorated in all three cases. For FI pruning the distribution seems to be even further off than with random or anti-FI pruning. It over-produced frequent patterns, but generated less infrequent ones. As an attempt to explain this, we add that RBMs are energy-based models that are defined to have a certain temperature, again a concept borrowed from physics (Hinton and Sejnowski, 1986): during annealing, a metal is first heated and then cooled down, thereby lowering the energy state of the metal. We have $P(x) \propto \exp(-E(x)/T)$ where T denotes the temperature and controls the phase of the system. A small temperature below the phase transition indicates an energy landscape with very steep valleys corresponding to few states with high probability. The remaining states on the other hand are very unlikely. Overall, the model distribution then has low entropy. Possibly, the FI pruning led to a cooling of the RBM which then got stuck in such a steep local energy minimum. Considering the sensitivity was highly correlated with the mean activity of the latent units, the network after FI pruning was most likely dominated by very active units leading to a skewed model distribution. This may indicate that the biases need to be adjusted in addition to the weights in order to correct the parameters after FI pruning.

Yet, all three pruned models were able to recover after retraining seen in a quick convergence to a suitable new configuration. Interestingly, the distribution for the model that had its *most* sensitive latent units removed (anti-FI) did not match the very frequent patterns as well anymore. Other than that, there were no heuristically noticeable differences after retraining.

Looking at the FIM diagonal after pruning and retraining, we see that the estimated parameter-sensitivity after anti-FI pruning was still very heterogeneous (see Figure 4.4). The short retraining did not yet lead to a re-attribution of important parameters to a few latent units. At this point, it would be difficult to order the hidden units

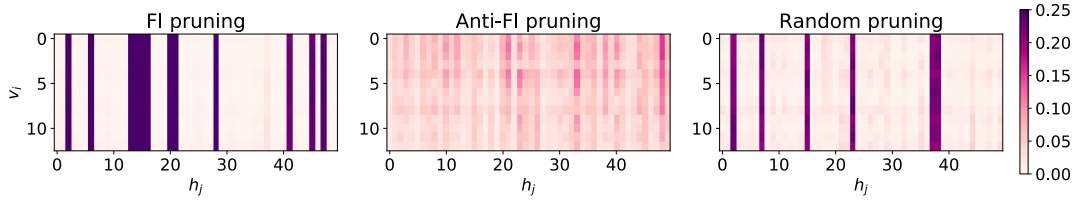


Figure 4.4: Diagonal entries of the FIM for weights after pruning half of all latent units of an RBM and retraining it for two epochs. For FI and random pruning the alignment of important parameters with few latent units remains. Pruning away the most important units destroyed said structure.

according to their average importance while this is still possible for the other two other models. That is because for both random and FI motivated pruning the organization into important and unimportant latent units remains, even more so for the latter. The random pruning may have led to the deletion of some of the stiff units, while the FI motivated pruning protected all of them. The preservation of this sensitivity structure technically allows fast repeated pruning to even smaller sizes. However, if allowed training for more epochs, any model with the same number of hidden units should theoretically show a comparable division into important and unimportant units, regardless of the initialization of parameters.

If the parameter sensitivity within a latent unit is more heterogeneous, the removal of complete hidden units is difficult to justify. Strikingly, when increasing the visible layer size, the latent units are not as well separated either. The FI then is more heterogeneous among all hidden units and requires a way of removing single weights corresponding to synaptic pruning.

4.1.3 Weight Pruning in RBMs

The following experiments were conducted during the implementation of the weight pruning and should thus rather be seen as a demonstration of the concept. We then applied this technique to a DBM fit to MNIST.

We present the results of an exemplary RBM that was fit to circular CIFAR-10 patches of 37 pixels for two epochs. As expected, the distribution of FI among the hidden units was slightly more heterogeneous for this larger visible layer size (see Figure 4.5). We then repeatedly removed weights which had a lower FIM diagonal entry than a pre-defined threshold of 0.05 for three iterations. After each pruning event the RBM was retrained for two epochs with lowered learning rates of 0.01 for the first and 0.001 for the second epoch (corresponding to a slow-down by factor 10 compared to the initial training). Other than that the hyperparameters remained the same (see Section 3.2.1). Unconnected hidden units were not removed from the model.

The initial model comprised $5,550$ weights and did not have a reasonable fit. The sampled states of its visible units did not match the means of the data pixels well. After three iterations of pruning and retraining the fit slightly improved regarding the mean and covariance of the visible units. Yet according to the rank-frequency plot, one cannot argue that the fit necessarily improved. Again, we note that it is difficult to evaluate the encoding of an RBM using these heuristics.

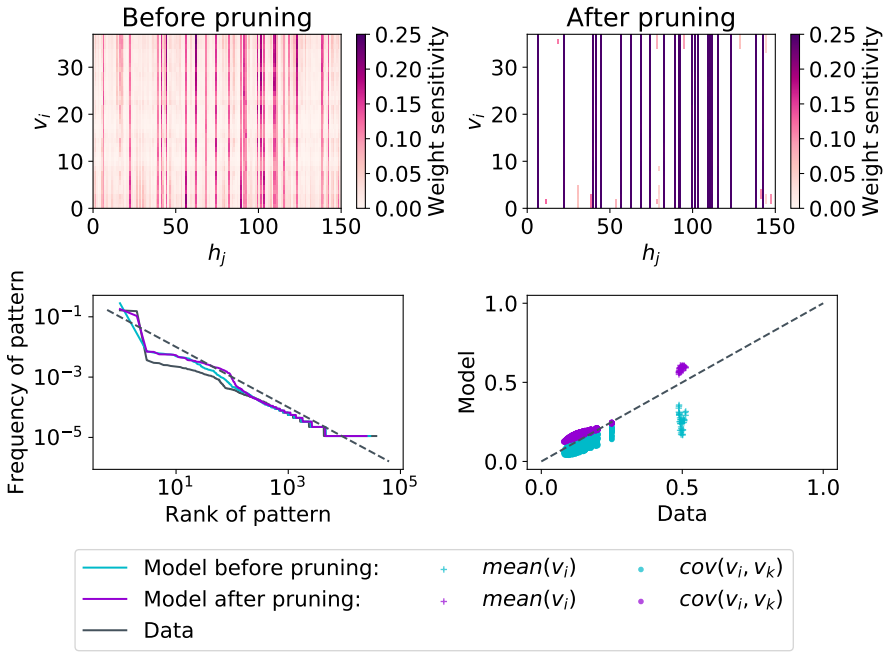


Figure 4.5: Comparison of model fit and FI before and after pruning the weights of an RBM for three iterations. White space in the top-right plot corresponds to pruned weights.

The FI pruning led to the majority of weights being removed in the first iteration: 79.982% of all weights had an estimated sensitivity of less than the threshold resulting in the removal of almost a third of the hidden units (see Table 4.1). After that, only a few more weights were pruned away, still leaving several more latent units unconnected. Interestingly, the overall importance of parameters seen in the sum of the diagonal, i.e. the trace of the FIM, remained fairly constant with a slight drop after the second pruning event (see Table 4.1). This is also reflected in the increase of importance of the weights of a few hidden units while others were deleted so that their FI was set to zero. The parameter-wise FI was "thinned out" between these stiff latent units (see top-plots of Figure 4.5). Presumably the sloppy, completely or partly unconnected units could now be removed from the network, a strategy that we followed in later experiments.

Given that the network apparently had not yet converged to an adequate energy minimum seen in the initial unsatisfactory fit, we argue that the pruning happened during early learning. This may correspond to natural pruning in a critical period. While Achille et al. (2017) found that the FIM trace decreased after a maximum in these early epochs, we observed that it remained rather constant although the RBM

Pruning Event	$n_{connected\ v}$	$n_{connected\ h}$	$n_{weights}$	$Tr(FIM)$
initial	37	150	5550	231.52
1	37	53	1116	240.88
2	37	38	912	218.14
3	37	35	894	230.79

Table 4.1: Number of active weights and remaining connected visible and hidden units over time. The exemplary RBM was pruned for three iterations with a threshold of 0.05 for the respective FIM diagonal entry. The last column shows the trace of the FIM, i.e. the sum of the diagonal entries.

had lost many of its parameters. Yet it is possible that the trace would decrease if the network was trained further in a given configuration without deleting any more parameters or units. However, the generally early separation into rather important and unimportant connections is in line with the observations made by Achille et al. (2017). This also becomes apparent in Figure 4.6 which shows the development of parameter sensitivity over the course of the three pruning events. The weights diverged into important and unimportant ones early and remained their relative sensitivity. Mind that pruning only targeted weights in this experiment, so the hidden biases were not taken into account during pruning. Yet when a unit became entirely unconnected, the sensitivity of its bias was set to zero as well.

Said temporary drop of the FIM trace became evident in a parameter-wise decrease of the sensitivity of all biases and weights. Perhaps the maximum and subsequent de-

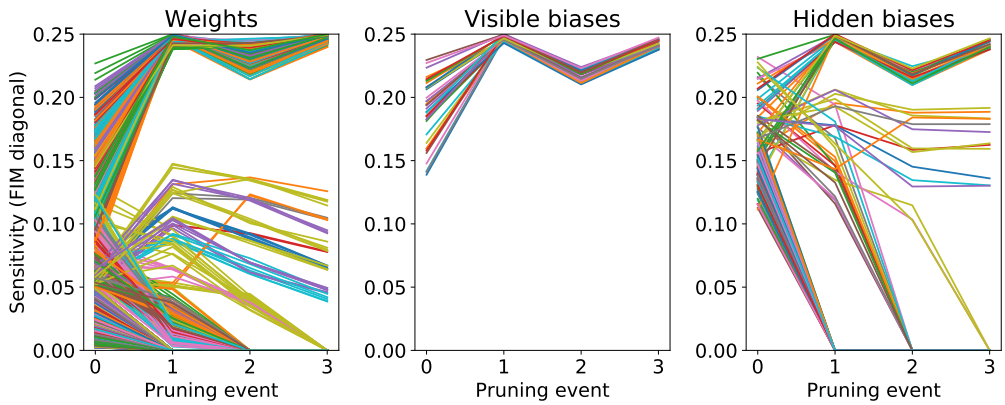


Figure 4.6: Development of parameter sensitivity over time shown separately for weights, visible and hidden biases.

crease of the FIM trace may carry useful information about when to stop pruning. Yet it should be noted that the FIM trace is not a locally available statistic for an individual neuron. As shall be discussed later, the time course of the emergence of sensitive directions during learning and pruning is generally worth further investigation, yet was out of the scope of this project.

The FI motivated weight pruning led to a selective deletion of the connections between the input and the hidden units. We further observed that, quite intuitively, the weights to a few neighbouring pixels remained protected. Figure 4.7 shows the pre- and post-pruning receptive fields of hidden units. Generally, we observe that the receptive fields of the latent units that survived the three iterations of pruning did not experience a drastic change. Even before pruning they covered different restricted and coherent areas of the input space and may encode spatially simple features comparable to biological on-center RGCs (Dayan and Abbott, 2001). On the other hand, the hidden neurons with initially diffuse and large receptive fields experienced a great loss of weights. Either they became entirely disconnected or relatively weak connections to very few input pixels remained. Supposedly they could also be removed in another iteration of pruning.

Although we conclude that these preliminary results may inspire many new research questions, we refrained from further explorations with RBMs fit to CIFAR-10 patches. That is because we still aimed for more objective evaluation criteria of the pre- and post-pruning model fit. After all, if retraining allows model recovery even from the loss of its most important parameters, what is the advantage of FI motivated pruning? Our main experiments with a DBM trained on MNIST aimed at answering this question.

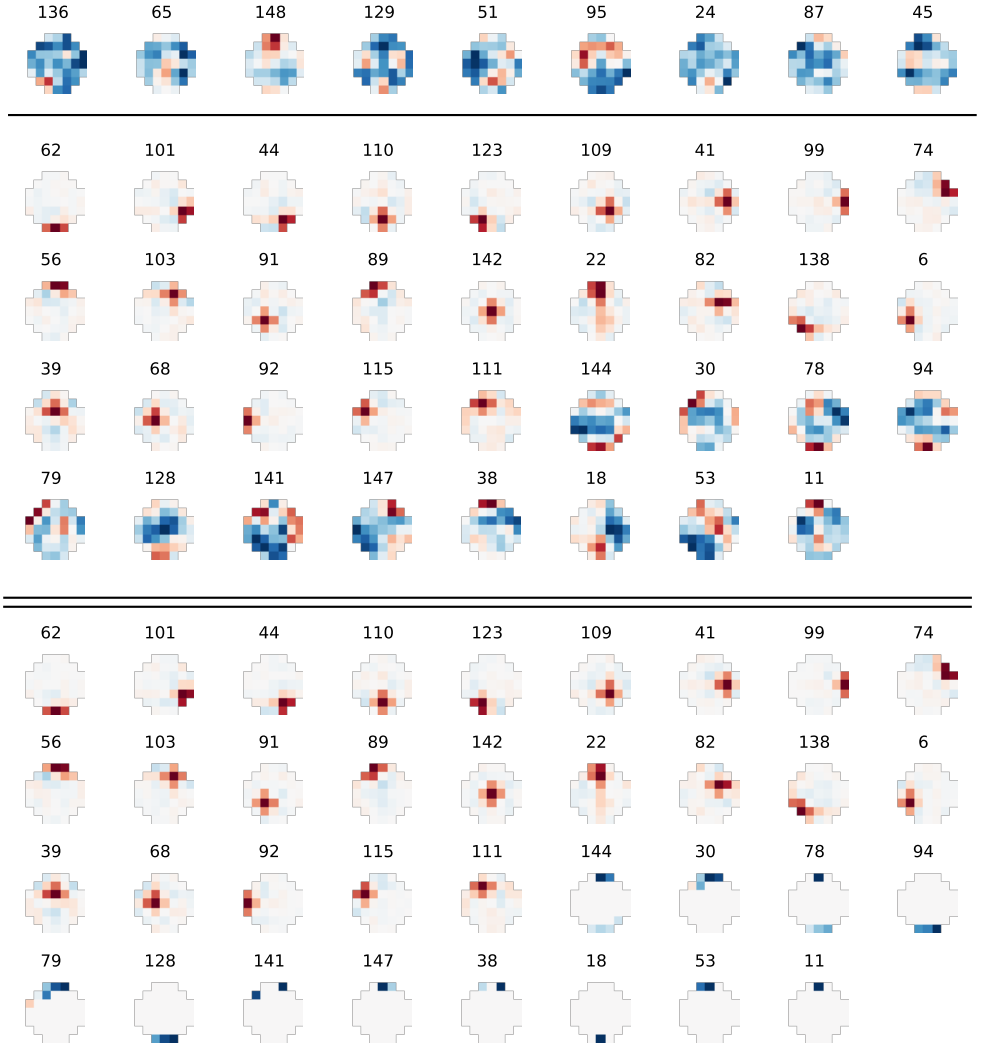


Figure 4.7: Receptive fields of latent units before and after FI motivated pruning. Top: receptive fields of nine random hidden units before pruning. None of them survived the three iterations of pruning, i.e. they belong to the 115 of initially 150 latent units that ended up disconnected to the visible units. Middle: receptive fields of the 35 remaining latent units before pruning, i.e. in the fully connected RBM. Bottom: receptive fields of these same 35 hidden units after pruning, ordered by their final relative sensitivity computed as the sum of the FIM diagonal entries of all their weights (left-to-right, then top-to-bottom starting at unit with index 62 beneath the double line).

4.2 Simulations with DBMs on MNIST

The iterative FI motivated removal of weights from a DBM was the main focus of this project. We thus first introduce the initial model that was the starting point for each of the following pruning experiments.

4.2.1 Initial Model and Baselines

The initial DBM was trained according to the procedure described in Section 3.2.2, while its fit was monitored regarding the two heuristic evaluation criteria mentioned in Section 3.4.1. It had 400 visible units, 400 units in the first hidden layer $\mathbf{h^1}$ and 676 units in the second hidden layer $\mathbf{h^2}$. Note that the connections in $\mathbf{h^1}$ were pre-restricted due to receptive fields with a maximal size of 5×5.

Figure 4.8 shows aspects of the initial fit. As can be seen in the top-left plot, the DBM had a satisfactory fit regarding the mean and covariance of the 400 input

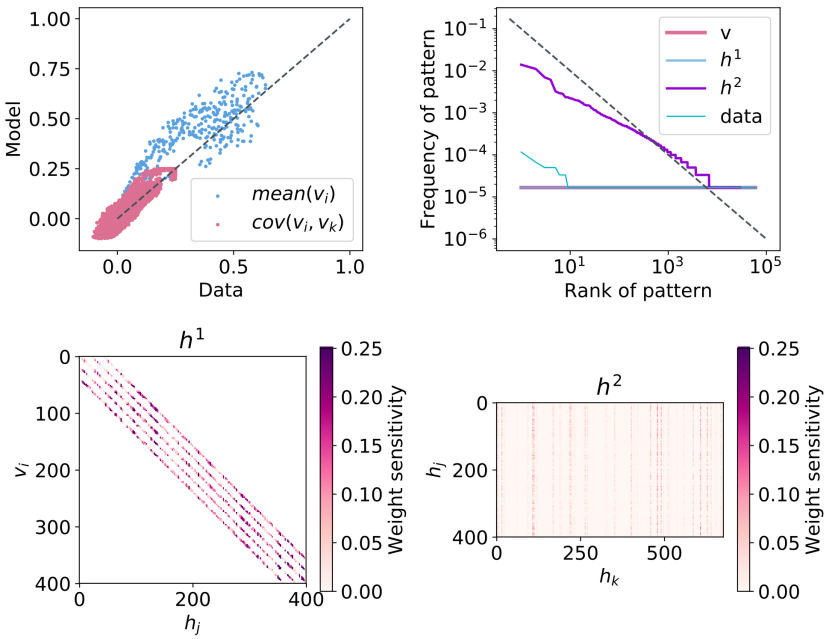


Figure 4.8: Initial fit of the to-be-pruned DBM. Top left: comparison of mean and covariance of the pixels of the training data and the respective visible units of the model estimated from $60,000$ generated samples. Top right: Frequency vs. rank of generated samples. Zipf's law only emerges for $\mathbf{h^2}$ samples. Bottom: Weight sensitivity computed as the diagonal entries of the FIM for $\mathbf{h^1}$ and $\mathbf{h^2}$. Non-existing weights in $\mathbf{h^1}$ correspond to white space.

pixels. We further see that the visible $\mathbf{v}$ and first hidden layer $\mathbf{h^1}$ produced solely unique patterns across the $60,000$ samples. Zipf's law only seemed to be present in the samples from the second hidden layer $\mathbf{h^2}$. The weight sensitivities computed as the diagonal entries of the FIM were generally high in the first hidden layer. It did not seem as if there were certain latent units that were overall more important than the rest in $\mathbf{h^1}$. Mind that the FI had this diagonal structure because of the receptive field that drastically restricted the connections to neighbouring areas of the input area. Instead of $160,000$ weights that a model without receptive fields would have had, there were only $8,836$ active weights in $\mathbf{h^1}$.

The second layer on the other hand was fully connected. The known organization into slightly more important and unimportant latent units was already noticeable. Yet, it also became evident that the majority of weights were irrelevant and thus candidates for pruning. In fact, the FI for $217,860$ of the $270,400$ weights (80.56%) in $\mathbf{h^2}$ was estimated to be zero due to arithmetic underflow. Remembering that the FI is computed as the covariance of the pre- and postsynaptic neuron, it is not surprising that most of these weights belonged to neurons in $\mathbf{h^2}$ that were silent across all samples and hence did not vary at all.

The difference in average weight sensitivity between layers may also be reflected in the mean activity of the neurons. While we observed a mean activation probability of 0.381 in $\mathbf{h^1}$ averaged over all its units, it was much lower in $\mathbf{h^2}$ (0.017). The activity was thus extremely sparse in the second hidden layer. There were still correlations present between the latent units in $\mathbf{h^1}$ (mean $r^2 = 0.084$, range $3.85 \times 10^{-07} - 1.0$) which justifies adding a second layer in order to model these. The correlation coefficients in $\mathbf{h^2}$ were not computable due to said non-varying units, yet the covariance ranged between 0.0 and 0.249.

Table 4.2 compares the classification accuracy of two classifiers trained on either the raw downsized and binarized 20×20 pixel digits vs. the final hidden layer rep-

Training Data	SVM	Logistic Regression
Raw Digits	0.9762	0.9134
$\mathbf{h}^2$	0.9544	0.9539

Table 4.2: Classification accuracy of SVMs with a radial-basis function kernel and logistic regression classifiers trained on the raw digits versus the final hidden layer representations of the DBM.

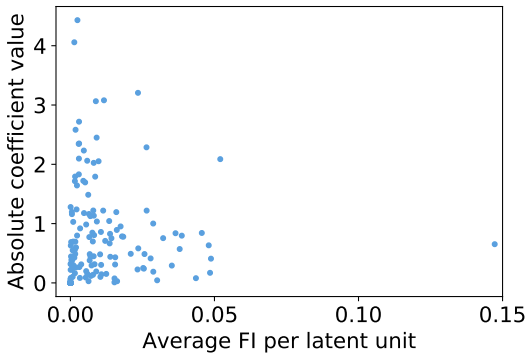


Figure 4.9: Absolute logistic regression coefficients against the average sensitivity of a latent unit computed as the mean of the FIM diagonal entries of all its weights.

resentations, i.e. the activation probabilities of the latent units in $\mathbf{h^2}$ for each training image. We see that the performance of the SVM trained on $\mathbf{h^2}$ is worse compared to one trained on the digits themselves. Yet we note that a classification accuracy of 0.9762 is already extremely high for a standard SVM and, as a side note, comparable to an SVM trained on the original 28×28 grey-scale digits which had an accuracy of 0.9751. The logistic regression classifier on the other hand profited from being trained on the hidden layer activation probabilities instead of on the raw digits. These are our two baselines. For each pruned model, we can now check when the classification performance falls below the accuracy rates of both the classifiers trained on the initial $\mathbf{h^2}$ and on the raw data.

The logistic regression classifier has another interesting property: since it is a linear classifier, its absolute coefficient or weight values indicate the relative feature importance. Mind that there are as many coefficients as there are feature variables, i.e. the 676 latent units in $\mathbf{h^2}$. Figure 4.9 shows the absolute weight values against the average FI of each hidden unit computed as the mean estimated FIM diagonal entry for all its weights. There is apparently no clear relationship between the feature importance for classification and average sensitivity of a latent unit. Remembering that RBMs and DBMs learn in a completely unsupervised fashion this makes sense. Their primary goal is finding a model of the data distribution at hand, irrespective of this encoding being potentially useful for a classifier.

Regarding the generative performance, we compared the confidence of the classifier fit to the raw MNIST training digits about the $10,000$ unseen images from the test set with its confidence about generated digits from the model (see Section 3.4.2 for further explanation). Exemplary digits can be seen in Figure 4.10. While the classifier

Figure 4.10: 100 randomly selected generated patterns corresponding to visible samples from the initial DBM. Samples were stored after every 200^{th} Gibbs step.

was very confident about the digits in the test set to belong to a certain category (mean probability for winning class $= 0.913$), its confidence about the generated samples was lower (mean probability for winning class $= 0.854$). Notably, even small deviations in this metric may well indicate a great difference in digit quality. This becomes evident in a mean confidence of 0.793 that the same classifier had about random patterns of the same input size. Thus, we also qualitatively inspected the images regarding their resemblance to digits after each pruning event.

As can be seen, the model seems to be biased towards certain patterns such as zeros or eights. For instance, according to the class probabilities of the classifier, only 157 out of $60,000$ samples from the DBM were believed to be ones. This bias is also reflected in a low diversity score of 0.011, yet as an anticipatory note, was rarely higher in any of the pruned models.

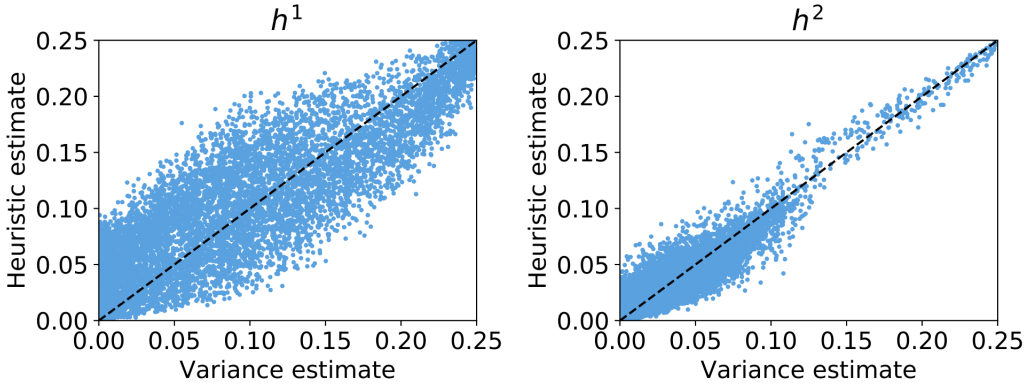


Figure 4.11: Initially, Rule et al. (2018) estimated the FIM diagonal entries as the covariance of the firing of the pre- and postsynaptic neuron, naming the variance estimate of parameter sensitivity. The heuristic estimate is an alternative approximation.

In order to justify our alternative local pruning criterion, we compared the original estimate (see Equation 2.16) with the new heuristic estimate of the FI per weight (see Appendix A). Figure 4.10 plots the two estimates against each other, separately for the two hidden layers. Acknowledging that the heuristic estimate assumes a large hidden layer and independence between its units, it makes sense that the estimate is more accurate for $\mathbf{h^2}$ which has both more neurons and lower correlations between its units. Both the original variance and the heuristic estimate were later used to indicate the relative importance of connections in the DBM in order to reduce its size.

4.2.2 Pruning One Layer at a Time

Two layer-wise pruning schedules were compared against each other. In the first schedule, we started pruning weights from $\mathbf{h^1}$. The model was evaluated, retrained and re-evaluated as described in Section 3.4.2. Then we pruned weights from $\mathbf{h^2}$. Again, the model was evaluated, retrained and re-evaluated before the whole procedure was repeated. Each joint retraining ran for 15 instead of initially 20 epochs. All other hyperparameters were the same as described in Section 3.2.2. The second pruning schedule followed the same logic, except that we started with pruning $\mathbf{h^2}$ followed by $\mathbf{h^1}$. In both cases unconnected hidden units in the intermediate layer were kept in the network. The pruning criterion was the 10^{th} percentile of weights with lowest FI according to the original variance estimate of the FIM diagonal elements. Weights were iteratively pruned 10 times.

Figure 4.12 shows the active latent units in each hidden layer separately for the two pruning schedules. Here, an active unit is defined as one that still has connections to the preceding layer in order to encode the input. Mind that although our intention was to prune 10% of weights in each iteration, the fact that the estimated FI was zero for as many as 80.56% of weights in $\mathbf{h^2}$ led to the 10^{th} percentile having a value of zero. As a consequence, all of these were pruned away in the first iteration. After that the weight pruning continued at a slower pace, leaving only a few more hidden units in $\mathbf{h^2}$ disconnected. For the first layer on the other hand it was never the case that the 10^{th} percentile was zero, resulting in relatively fewer units of the initially 400 becoming disconnected. Mind that due to the receptive fields $\mathbf{h^1}$ had far less active weights to begin with and a less sparse FIM diagonal as seen in Figure 4.8.

At each evaluation point, the $60,000$ training images were fed to the DBM in order to compute the hidden unit activation probabilities. These were then used to train

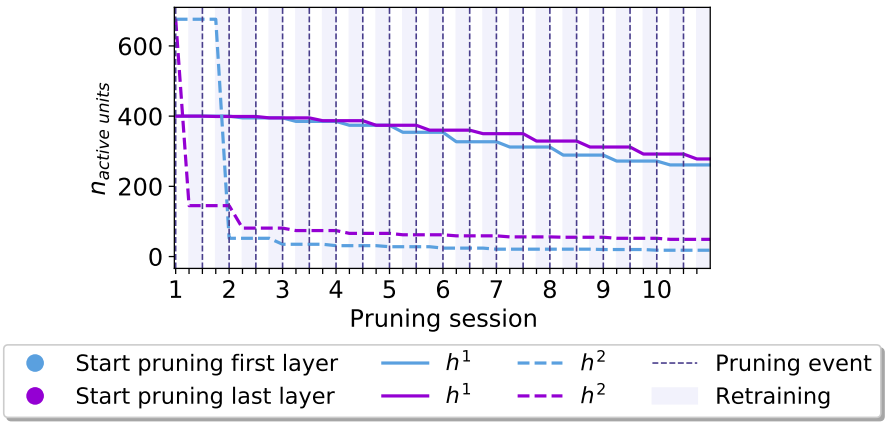


Figure 4.12: Active units in both layers $\mathbf{h^1}$ and $\mathbf{h^2}$ of the DBM over the time course of ten pruning sessions following the alternating layer-wise pruning schedules of starting with the first or the last layer. Each pruning event was followed by a retraining phase.

a logistic regression classifier and an SVM. The classification accuracy of these two classifiers were seen as a measure of the encoding quality. Figure 4.13 shows said performance over time separately for the two layer-wise pruning schedules.

In both cases, we see a drastic performance drop right after the first pruning event. The damage was even more pronounced when starting with pruning the first layer. However, this could be almost fully recovered from retraining following both schedules. After that the performance remained fairly constant for the schedule that starts pruning the second layer. Yet in later iterations we generally see a pattern for this schedule: the performance seemed to be almost unaffected after the first pruning event of each iteration, i.e. the pruning of the second layer. After pruning the first layer however, the classification accuracy dropped, leading to a zig-zag with short plateaus. For the opposite schedule, we observed a similar pattern in the later iterations, mirroring the same behavior in a shifted curve.

The classification accuracy of the SVMs never exceeded the baseline threshold of a respective classifier trained on the raw digits, regardless of the schedule. Furthermore, the performance compared to the second baseline of the initial DBM steadily decreased for both schedules. Yet again, this decrease is more pronounced for the schedule starting with the first layer.

On the other hand, the logistic regression classifier profited from the hidden layer representations even after the ten iterations of pruning and a final loss of 65.17% of all initial weights in $\mathbf{h^1}$ and 95.72% in $\mathbf{h^2}$, at least following the schedule of starting with the last layer. Similar to before, the opposite schedule led to worse performance and we

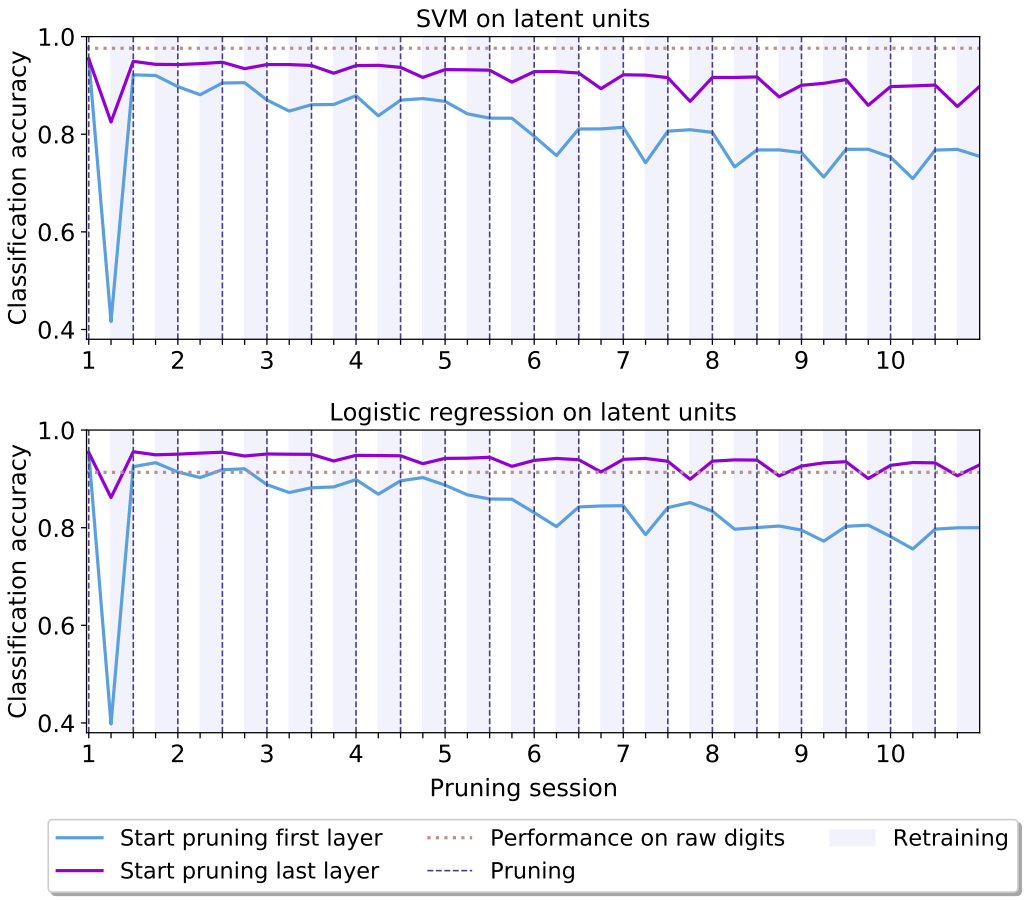


Figure 4.13: Classification accuracy of two different classifiers trained on the final hidden layer representations of the DBM over ten iterations. During each iteration, the model was evaluated four times: after each pruning event and after each retraining phase. The evaluation points are marked by ticks between the number indicating the iteration. The performance of comparable classifiers trained on the original digits is marked by the horizontal dotted line.

observe the same zig-zag interrupted by short plateaus when pruning the second layer. Generally, pruning the first layer on its own seemed to be more detrimental for the encoding performance according to both classifiers. Yet we note that compared to the performance of the initial DBM, the classification accuracy fell below that level after the fourth iteration. After ten iterations of pruning and retraining, the final accuracy of the classifier was 0.9287 and hence lower than the initial one of 0.9539.

Figure 4.14 shows the generative performance over the time course of ten pruning sessions for both pruning schedules. For the digit quality we see a very similar behavior to the encoding performance. After an initial performance drop following the first pruning event of the first iteration, the zig-zag plateau pattern returned, favoring

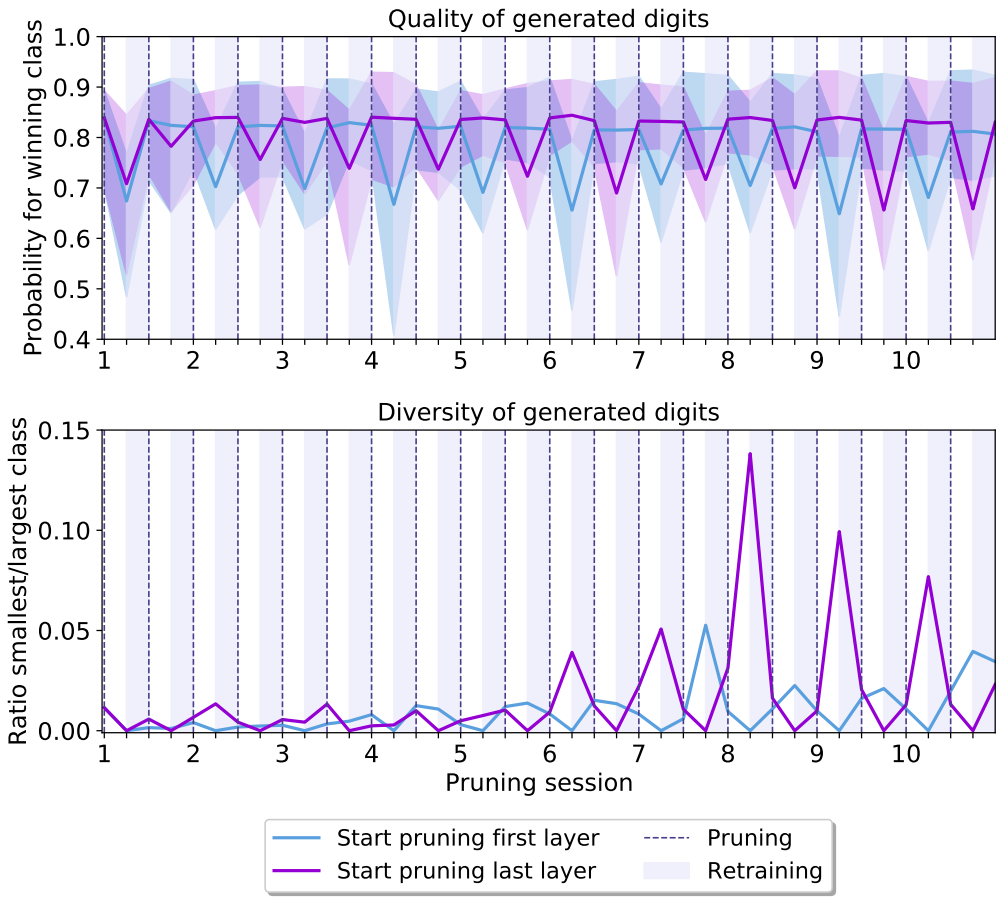


Figure 4.14: Quality and diversity of generated samples from the DBM that were fed into a classifier trained on the original MNIST dataset. For each given sample, the classifier assigned a probability for each digit class. The maximum probability of these was chosen to be the digit class that the sample belongs to (the winning class). The means across these are illustrated by the lines in the above plot. The shadows denote the range of quality across the ten classes. Bottom: Diversity of digits computed as the minimum divided by the maximum counts of samples assigned to the digit classes.

the pruning of the second layer. Other than that there was again a slight general performance advantage for this schedule. Each time after pruning the first layer, the digit quality was dramatically impaired. From qualitatively inspecting the samples during pruning, we can confirm that they did not look digit-like and were also very similar to another. This is also reflected in the diversity score that dropped to zero each time the first layer was pruned, indicating that there is at least one digit class that has zero counts. Although the diversity was generally low across the ten digit classes, the model pruned according to the schedule starting with the second layer suffered less from pruning. Quite contrary, the digit diversity even seemed to profit from pruning

in later iterations. Note that our initial DBM was biased towards producing samples with many visible units or pixels in the on-state (such as eights or zeros). The weight pruning may have decreased the mean activity of the visible units due to a reduced number of connections and thus potentially less excitatory input from $\mathbf{h^1}$, allowing for the generation of digits with less on-state-pixels such as ones or sevens.

4.2.2.1 Remarks on Initial Performance Drop

The initial drop of the encoding performance following the first pruning of either layer motivated a closer investigation. Specifically, we were interested if it was possible to preserve the encoding performance by somehow compensating for the loss of units and weights. Considering that the first iteration of pruning led to the numerically largest loss of weights and units, one may intuitively attribute it to this abrupt and strong lesion of the network. Like in a stroke many neurons and their respective synapses suddenly die and lead to severe functional deficits. Yet the performance drop also occurred and was even more pronounced when starting with pruning the first layer, which lost just 10% of its weights like in all other iterations. We thus repeated the beginning of the first iteration for the schedule starting with the pruning of the second layer and instead of deleting all weights with an FI of zero just selected 10% of these. Strikingly, we observed a comparable performance drop.

We further checked if the pruning led to sudden high correlations in the hidden layers, yet this was not the case. They were comparably low as to before pruning. Re-initializing the pruned DBM with the weights scaled by the number of previous and lost hidden units as seen in Figure 4.2 did not result in better performance either. What seemed to be promising was an adjustment of the hidden biases. We saw that the biases of the remaining latent units after the pruning of the second layer were extremely negative ranging between -39.78 and -2.64. Re-initializing the hidden biases with the previous mean activation probability of a neuron[1] (range $0-1$) alleviated the encoding performance drop. Remembering the temperature of the system that we attributed the distorted model distribution to in our first experiments with CIFAR-10, perhaps this performance drop is also associated with a cooling of the DBM. A low temperature is generally seen in relation to high absolute parameter values. Yet we stress that this hypothesis needs further investigation.

[1] An initialization with $-1/(1+exp(mean\ activity))$ for the hidden biases to be negative gave even better results.

4.2.3 Pruning Both Layers at the Same Time

The third schedule involved a joint pruning. Both layers were pruned based on the simultaneous sampling from all layers of the DBM. The model was evaluated, retrained and re-evaluated before the next iteration began. In the previous experiment we observed that both the SVM and logistic regression classifier behaved similarly. In order to accelerate the pruning procedure and since an SVM never profited from the encodings of our DBMs, we continued our next experiments with a logistic regression classifier only. The generally lower space and time complexity that is associated with the joint pruning schedule allowed us to compare different pruning criteria as well.

Specifically, we compared three criteria. First, we used the sensitivity of the weights computed as the variance estimate of the FIM diagonal as before (FI pruning). Second, we used the alternative heuristic FI estimate (Heuristic FI pruning, see AppendixA). Third, we compared these two criteria to pruning away the *most* important weights, i.e. the ones with the highest FIM diagonal entries computed by the original variance estimate (anti-FI pruning). Again, the 10% of weights with lowest sensitivity (or highest in case of anti-FI pruning) were deleted. Unconnected hidden units from both hidden layers were removed from the computational graph of the network, even if they just lost connections to one side, i.e. they were dead-ends.

Figure 4.15 shows the number of latent units left in the network over the time course of ten iterations of pruning. Similarly to the layer-wise pruning schedules, there

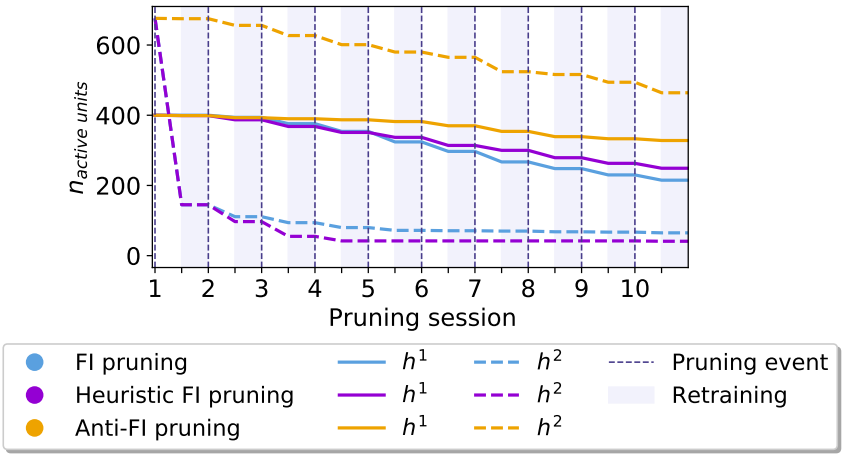


Figure 4.15: Active units in both layers $\mathbf{h^1}$ and $\mathbf{h^2}$ of the DBM over the time course of ten pruning sessions following the joint pruning schedule with three different pruning criteria. Either the least sensitive weights (FI and heuristic FI pruning) or the most important weights were pruned (anti-FI pruning).

was a harsh reduction in the number of hidden units in the first pruning session when pruning the least important weights. That is again due to the $> 10\%$ of weights with an estimated FI of zero. On the contrary, anti-FI pruning never led to the deletion of more than 10% of weights. The two FI pruning criteria left a slightly different number of units unconnected in the two hidden layers. For the heuristic FI pruning more neurons survived in $\mathbf{h^1}$ and less remained in $\mathbf{h^2}$ compared to the standard FI pruning.

Figure 4.16 shows the model performance during the ten iterations of pruning. Starting with the encoding performance, we see that there are profound differences between the pruning criteria and indeed standard FI pruning was superior to both the heuristic and the anti-FI pruning. Strikingly, there was no performance drop in the first iteration of pruning when using both estimates of FI. Quite contrary, the classification accuracy remained stable even immediately after pruning, i.e. before retraining. It then slowly decreased over the ten iterations, falling below the baseline of the classifier trained on the original digits in the 7^{th} pruning session. The second baseline given by the classifier trained on top of the initial DBM however was never exceeded. The heuristic FI pruning seemed to behave similarly in the first iteration, yet led to deteriorating performance after that. As an additional note, our monitoring of the relationship between the two estimates during pruning revealed that there were sometimes many outliers, possibly leading to the removal of the wrong weights. The classification accuracy decreased until the 5^{th} pruning session, after which it stabilized, but remained under the level of both baselines.

Anti-FI pruning led to a severe performance drop in the first iteration, similar to the one we observed in layer-wise pruning. This is remarkable given that numerically far less weights were pruned as seen in Figure 4.15. The model did not fully recover through retraining and the accuracy of classifiers trained on its encodings steadily decreased. After ten iterations of pruning, it remained lowest compared to the other criteria although it had most weights and units left in both layers. This nicely demonstrates that the key to performance does not lie in the sheer number of parameters.

Continuing with the generative performance, we see a similar pattern for all three pruning criteria. Immediately after each pruning event the samples were distorted, but this could largely be recovered through retraining. Yet we want to stress again that small differences in the digit quality score may indicate large differences in the actual similarity of generated patterns to digits. For example, the two dotted lines denoting the same measure for the minimal models introduced in Section 4.2.4 may not lie far apart. Yet the exemplary samples shown in Figure 4.19 differ a lot in their resemblance

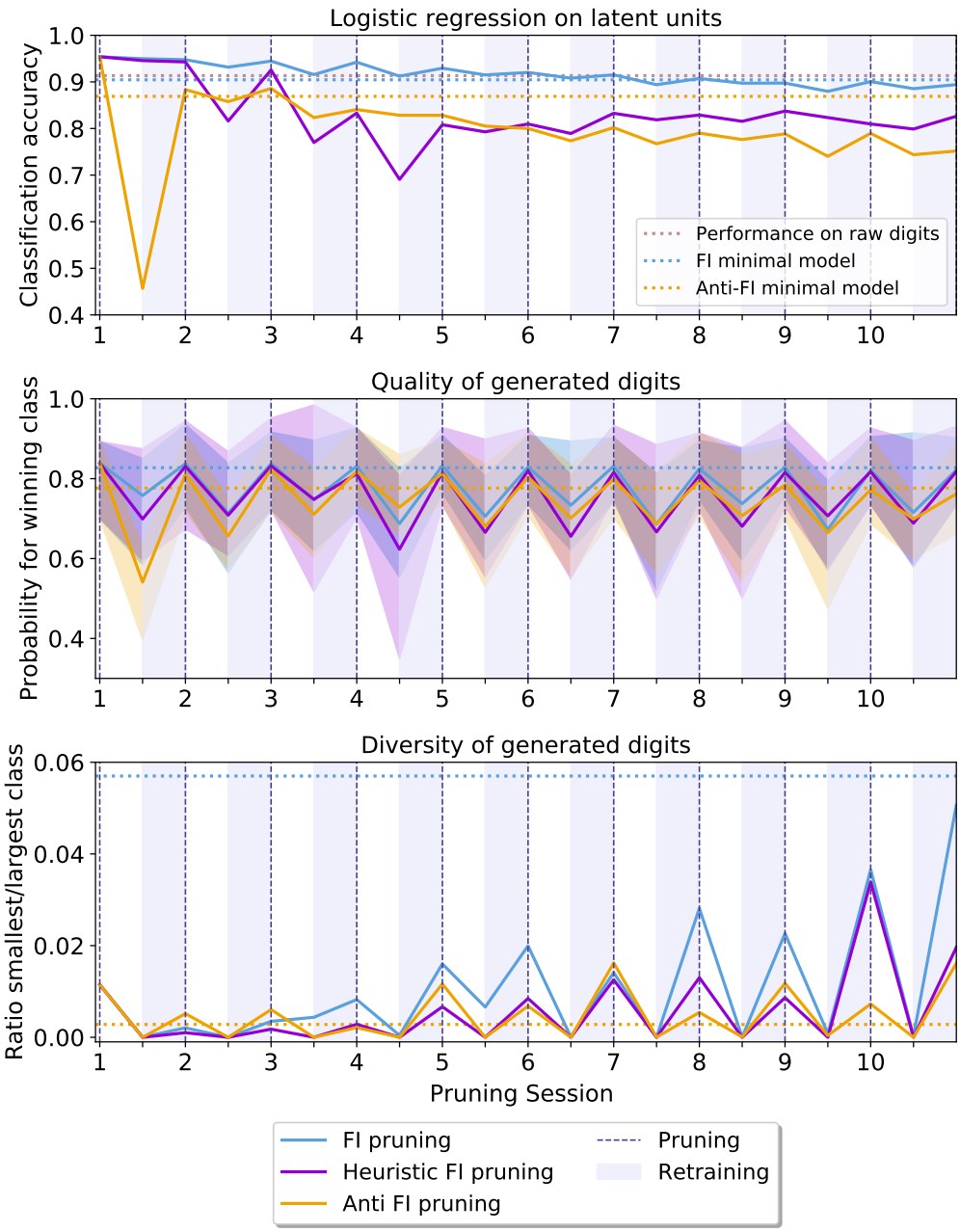


Figure 4.16: Evaluation measures of the DBM pruned over ten iterations following the joint pruning of both hidden layers according to three different pruning criteria. The dotted lines indicate the respective performance measures of either the minimal models (see Section 4.2.4) or, for the encoding performance only, of a classifier trained on the raw data. Digit quality refers to the confidence a classifier had that a sample belonged to a particular digit category. The shadows indicate the range of this measure among the ten classes. Digit diversity then was computed as the ratio of number of samples in the smallest and the largest class that the samples were assigned by the classifier.

to handwritten digits.

The diversity of the digits follows a similar pattern as we saw in the layer-wise pruning. After each pruning event, the samples were vastly indistinguishable so that most of them get assigned to the same category. This could also be confirmed from a qualitative analysis of the samples during pruning. In the later iterations, the diversity of digits even seemed to profit from FI pruning.

An inspection of the indices of the visible units that became disconnected as a result of pruning weights from $\mathbf{v}$ to $\mathbf{h}^1$ provides further interesting insights. Figure 4.17 compares the final visible layers after the ten pruning sessions based on the three different criteria. Anti-FI pruning left supposedly highly informative pixels in the center of the input disconnected. On the other hand, the two FI pruning criteria led to disregarding the pixels at the edges of the input. They are intuitively less informative since they vary little across the samples: their state is almost always off. The pixels in the center on the other hand differ between the samples of different categories. It was thus interesting to see if the anti-FI pruned model was still capable to encode the data distribution without input from these presumably highly informative pixels.

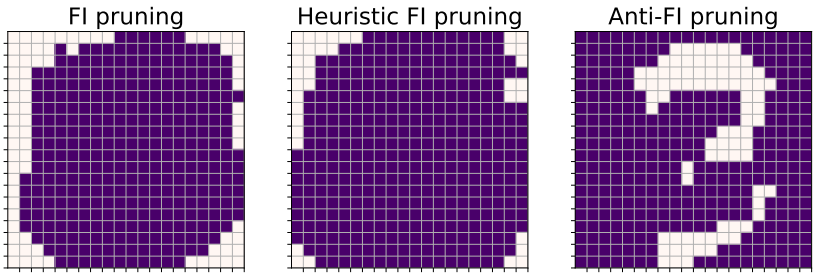


Figure 4.17: Comparison of the visible layers after ten iterations of joint pruning following three different pruning criteria. The dark pixels resemble the units that are still connected while the light ones denote unconnected visible units.

4.2.4 Minimal Models

The comparison of the final visible layers of the DBMs motivated us to build so-called minimal models with an architecture as indicated after ten iterations of joint FI vs. anti-FI pruning. This experiment aimed at answering two questions. First, we wanted to explore if it is beneficial to start with a large model and iteratively prune it compared to training the minimal model from scratch. That is, the minimal model has less redundant connections and thus needs to use its architecture more efficiently to encode the

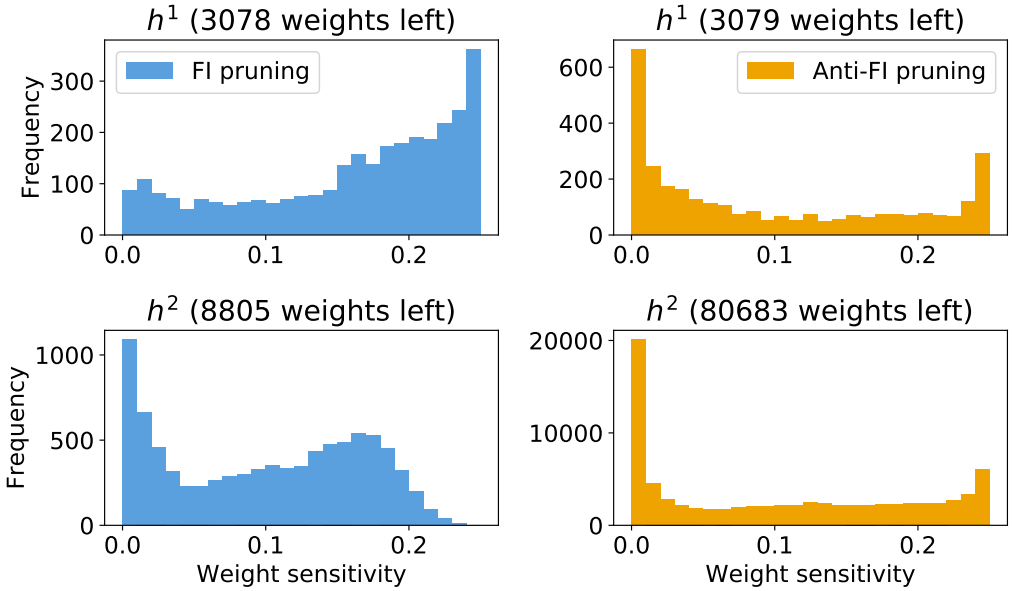


Figure 4.18: Comparison of the sensitivity of weights of the two minimal models per layer. Mind that the anti-FI minimal model had a lot more active weights leading to different ranges on the y-axes. Weight sensitivity was estimated through respective diagonal entries of the FIM.

data. Second, we were interested to see if pruning away the most important weights topologically damages the network in a way that it cannot recover from, even when allowing for a pre-training of the individual RBMs. Note that the retraining between the pruning events only involved a joint training of the DBM.

Specifically, two minimal models were built. Either it had the same weights masked and number of hidden units left such as the DBM after ten iterations of FI pruning or after anti-FI pruning. We refer to the first one as the FI minimal model and the second one as the anti-FI minimal model.

As can be seen in Figure 4.16, the FI minimal model outperformed the anti-FI one in both the encoding and generative performance regarding the digit quality and diversity. The FI minimal model reached a performance comparable to the DBM after ten iterations of FI pruning. The anti-FI minimal model performed better than its counterpart after ten iterations of anti-FI pruning, yet it did not reach the performance level of the FI minimal model. This demonstrates that the pruned weights and units were indeed important and topologically necessary for a satisfactory encoding of the data distribution. The result is also remarkable in light of the different numbers of parameters of the networks. In Figure 4.18 we see that the anti-FI minimal model had far more active weights than the FI one (total of $83,762$ vs. $11,833$). Yet most of these

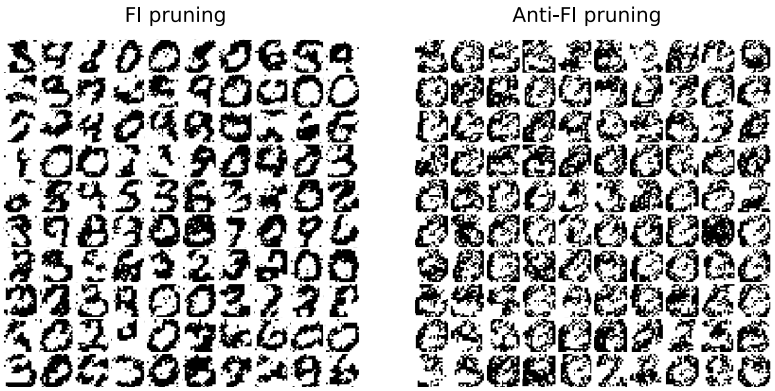


Figure 4.19: 100 randomly selected visible samples from the two minimal models. A sample was stored after every 200^{th} Gibbs step.

had low sensitivity, comparable to the overall low sensitivity in the second layer of the initial DBM (see Figure 4.8). Most weights of this model were irrelevant and could further be removed. Yet the network is missing connections relevant for encoding the input and would thus probably suffer even more from further pruning.

As a final comparison, Figure 4.19 contrasts samples from both minimal models. While the generated patterns from the FI minimal model still partly resembled digits, the samples from the anti-FI one were completely distorted. This is also reflected in the low quality score of 0.776 which is below the score for random patterns (0.793).

4.3 Conclusion of Results

While our initial experiments with RBMs fit to CIFAR-10 circular patches were limited due to the difficulties of evaluation, the main pruning experiments of a DBM fit to MNIST provided interesting insights.

With our local estimation of FI, we were able to reduce the number of weights of an initially over-parameterized ANN by more than 60% in the first and more than 90% in the second layer and still reach a satisfactory classification accuracy of the digits compared to a baseline of a classifier trained on the raw digits. Yet, according to the second baseline of a classifier trained on the initial hidden representations, the network arrived at a sub-optimal size over the course of ten pruning sessions. For the joint pruning and the schedule starting with the pruning of the first layer, the classifier did not even profit from the encodings anymore, compared to one being trained on the raw digits. Indeed, we over-pruned the model.

We also compared different loci of pruning and found a surprising pattern: the deletion of weights from the first layer generally seemed to be more detrimental than from the second. Interestingly, the quality of digits did not suffer after pruning of the second layer only. This was not the case when jointly pruning the DBM. Here, both the digit quality and diversity were impaired after pruning, yet could be recovered from retraining. The encoding performance on the other hand was not as negatively affected by the joint pruning. It slowly but steadily decreased over the course of ten iterations of pruning. The final visible layers of the models pruned according to three different criteria indicated that removing the most important weights topologically damaged the network in a way that it could not compensate through retraining. The building of two minimal models confirmed this: the minimal model with its weights masked according to ten iterations of anti-FI pruning performed worse regarding the encoding and generation of digits although its weights numerically outnumbered the ones of the FI minimal model. This nicely demonstrates that are indeed connections in an ANN that are more important or informative than others and that these can be identified by locally estimating their FI.

Chapter 5

Discussion and Limitations

5.1 Significance

We consider our experiments as a first empirical study of network reduction of RBMs and DBMs based on a local estimation of parameter-wise FI. We argue that compared to pruning criteria such as parameter-magnitude (e.g. Han et al., 2015; Tostado et al., 2017) or MI (e.g. Berglund et al., 2015; Sánchez-Gutiérrez et al., 2017), our rule can be justified from both an information-theoretic and biological perspective. Our results may thus provide insights into the naturally occurring removal of synapses (pruning) and neurons (apoptosis) in the developing brain. The fact that the majority of weights could be pruned away without a significant performance decrease supports our hypothesis that they may not be needed for the precise encoding of stimuli. A high correlation between the mean activity of hidden neurons and their average sensitivity supports the view that the death of synapses and neurons may indeed be triggered in an activity-dependent manner.

FI motivated pruning guided us in the reduction of initially over-parameterized RBMs and DBMs. In light of the difficulties of training and pre-determining the optimal size of such models, it may indeed be an advantage to start with a larger network and let FI indicate the optimal model size. This may be a strategy applied by biological systems as well.

By monitoring both the encoding and generative performance, we observed that pruning in the shallow, first hidden layer may be more harmful. Furthermore, repeatedly removing the most sensitive weights led to a damage of the network that it could not recover from through retraining. This demonstrated that FI may not just indicate the optimal network size (Rule et al., 2018), but also the optimal network topology.

That is, there are certain units and connections that could not be replaced by the same number of parameters elsewhere in the network, i.e. their spatial position seems to be critical. An inspection of the visible layers illustrated this: FI pruning led to supposedly uninformative pixels at the edges of the image becoming disconnected. Anti-FI pruning, on the other hand, cut connections to pixels in the center of the input, which are highly informative. While it may seem obvious that the central parts of the handwritten digits are more important, this may not be the case for each task and stimulus material. For instance, the circular patches with which we started our experiments were randomly selected from all locations of CIFAR-10 natural images. As we saw in Figure 4.7, the receptive fields of the important hidden neurons after pruning essentially covered all input pixels. But the ordering of the hidden units by their relative sensitivity indicated that in this example the most important pixels may have been at the bottom of the input space rather than the top or center. This may not have been as obvious before. The relative importance of weights and units gets even less obvious once we consider deeper layers of the network. Thus, the fact that FI pruning resulted in a disconnection of the pixels at the edges and anti-FI pruning to a disconnection of central pixels for the MNIST dataset should rather be seen as a proof of concept for the method.

Intuitively, the pixels that vary little are less relevant, e.g. the centered handwritten digits rarely cover the the corners and edges of the square images. This intuitive understanding of variance indicating importance also underlies our local pruning rule: the sensitivity of a weight was estimated as the covariance between the firing of the pre- and postsynaptic neuron, i.e. the variability of a unit from one layer with one from a neighboring layer (Rule et al., 2018). As mentioned before, a very recent study supported the view that biological neurons are able to track this variability through multiple synapses (Hiratani and Fukai, 2018), yet this is still not certain. What motivated us to derive an alternative local estimate of FI was the idea that the mean firing rates may be a more readily available local statistic, since they are also needed for homeostatic regulation. Homeostasis in this context describes the observation that neurons have a characteristic activity level that they tend to preserve and return to (e.g. Turrigiano and Nelson, 2000). Strikingly, our alternative heuristic estimate of the FI of a weight (see Appendix A) only uses the strength of a synapse and the mean pre- and postsynaptic firing rates instead of their coincidences. We saw that this heuristic estimate is a legitimate approximation in layers with many units and low correlations between them – the initial DBM is an example of this (see Figure 4.11). Yet from the second

iteration of pruning onwards it resulted in worse encoding and generative performance than the standard FI pruning (see Figure 4.16). Admittedly, if the presynaptic and post-synaptic mean firing rates can be tracked, a neuron may well also keep record of their coincidences.

The fact that our initial variance estimate of the FIM diagonal outperformed both other pruning criteria generally stands in favour of it. However, we still want to point out several limitations of the current work in order to motivate further investigation of the subject matter.

5.2 Limitations and Suggestions for Future Work

In our experiment that compared the two schedules of starting with the pruning of first or the second hidden layer we saw that the removal of weights in the first layer seemed to be generally more detrimental. This was a surprising result; one could have assumed that the locus of pruning has differential effects on the two evaluation measures, i.e. we would not have expected that pruning the first layer is more harmful to both the encoding and generation of digits. The fact that the schedule starting with the pruning of last layer continually outperformed its counterpart may support the view that sensory neurons depend on signals they receive from their target cells, which are then propagated backwards (e.g. Meyer-Franke et al., 1998).

Yet it is not clear if the pruning in the two layers is truly comparable in our experimental set-up. The aim of pruning the 10^{th} percentile of weights in each layer was to remove the same relative number of weights. However, this may not be the best strategy. First of all, the connections to the first hidden layer were already extremely restricted and pre-organized due to the receptive fields ($8,836$ weights instead of $160,000$). The average sensitivity of weights was thus much higher in the first than in the last layer (see Figure 4.8). One could argue that the percentage of weights to be removed in the first layer should be smaller considering they are already restricted by more than 90%. In the first iterations, the pruning by percentile in the second layer automatically led to more weights being pruned since the percentile was zero. In order to compare the schedules, criteria, and location of the pruning, one should carefully rethink how to set the threshold that determines how many weights will be removed. Between schedules and criteria the number of removed weights should be the same over time. Regarding the location, i.e. the first vs. the second and potentially deeper layers, further exploratory experiments should investigate if the advantage of starting

with pruning deeper layers remains if the connections are not as restricted by small receptive fields. One could start examining this by simply increasing the size of the receptive fields, thus remaining more weights to the first hidden layer and allowing for a larger overlap between the receptive fields.

We further note that the computation of a percentile is not local. It would be more biologically plausible to have a cut-off value that does not require the comparison with all other neurons in a layer. Ideally, there should be a stochastic local mechanism determining the removal of a weight or unit in order to truly resemble biological pruning and apoptosis. Thinking further, one may even want to model synaptogenesis, i.e. the (re-)adding of weights between units as pursued by e.g. Berglund et al. (2015). Potentially the same local computation of FI could not only guide network reduction but also network growth, resulting in an integrative model of neuronal plasticity. In relation to this, one could also allow neurons to have a "grace period": At the moment, the iterative joint pruning schedule radically removes any disconnected neurons. Perhaps they should have a chance to re-connect before they are removed.

However, the adding of connections most likely requires an adjustment of the other parameters which leads us to our next limitation: currently, we re-initialized the DBM with the remaining post-pruning biases and weights without any scaling. Except for when pruning only the second layer, we always observed that the generated samples were completely distorted immediately after pruning. The encoding performance repeatedly decreased as well. A re-training phase of 15 epochs allowed the network to recover to a comparable performance level as before. Although we argue that re-learning may indeed correspond to an automatic adaptation of the sensory plastic systems, it would be interesting to see if it is possible to derive a scaling of the remaining parameters that immediately compensates for the loss (or adding) of weights and units. A first investigation of parameter adjustment in order to avoid the initial performance drop in the first pruning iteration (see Figure 4.13) showed that the absolute parameter values were extremely large after pruning. A re-initialization of the biases with the mean firing rates instead alleviated the performance loss.

These observations fit with our suggestion that pruning may lead to a cooling of the system, yet this hypothesis needs a more thorough investigation. Admittedly, it is not trivial to determine the current temperature of an RBM or DBM, but one could start by introducing the parameter T and vary its value in a range close to 1 while monitoring the FI for samples. The FI is expected to fall off on both sides from the critical point, which would then indicate the approximate temperature (Machta et al., 2013). A

scaling of the weights may be an alternative or additional parameter adjustment. However, the naive re-scaling by the ratio of number of previous and post-pruning units only seemed to be adequate for weights close to zero and over-estimated the positive weights (see Figure 4.3). The current scaling involves a multiplication of the weights by a factor larger than 1. Given that the importance of a unit correlated highly with its mean activity, perhaps a rescaling should rather lead to a decrease of the weights in order to prevent the network from becoming dominated by highly active units. Altogether, the appropriate re-adjustment of parameters after pruning remains an open question and is subject to further research.

In close relation to this, one may also consider investigating the time-course of convergence after pruning and during re-training. The heterogeneous FI for the latent units after removal of the most sensitive units (see Figure 4.4) suggests that an advantage of FI pruning may indeed be a faster convergence to a new suitable configuration. Yet we did not explicitly direct an experiment at this hypothesis. A secondary observation was that the FIM trace stayed fairly constant during the first weight pruning experiments in RBMs (see Section 4.1.3). A recent study of the effects of perturbing the input during different time points of training in ANNs suggests that a critical learning period may be visible in a plateau of the FIM trace (Achille et al., 2017). It would be interesting to examine if such critical periods exist not just for changes in the "software" (training data) of a network, but its "hardware" (topology and number of parameters) as well. If this works, the development of FIM trace over time may also indicate when to stop further pruning. Yet again we note that the trace is computed by looking at the network globally and thus is not directly accessible for an individual neuron.

Generally, it would be desirable to accelerate the pruning procedure. Instead of removing a large number of weights between phases of full re-training, one could target fewer connections during (re-)learning for fewer epochs with small learning rates. Yet this approach brings forth the question of how to monitor the model fit, noting that our initial intention of fitting a DBM to MNIST was to have a more objective evaluation criteria. In the current work, each evaluation of the encoding performance consisted of training a new classifier on $60,000$ vectors comprising the hidden representations of each input image. Not only is this costly in terms of time, it also raises the question of compatibility of the classification accuracy across different evaluation points. Differences in the accuracy score may also be attributed to the quality of fit of the classifiers to the varying input data. Since we removed unconnected latent units from the last layer, the training data even had varying dimensionality. A way to circumvent this

would be to integrate the classifier into the model instead of building a separate one on top of the hidden unit representations. Strikingly, it is possible to do exactly this by adding a group of n_c visible units that receive as input the hidden unit representations, where n_c is the number of classes (Hinton et al., 2006). Either only one of this group of binary units is allowed to switch on, or if combined into a "softmax" unit, it can take on exactly one of n_c states (Reichert, 2012). The training of these classification units follows a standard supervised learning paradigm in that the label is provided for each training example. During testing, its states remain unclamped and need to be inferred from the hidden unit representations (Reichert, 2012). A follow-up study could repeat our iterative pruning procedure with this integrated classification mechanism as a means to evaluate the model fit between pruning events.

Regarding the generative performance, we realized that the logistic regression classifier trained on the raw digits was overly confident for the patterns it received as input. This became evident in the high digit quality score assigned to random patterns. During the set-up of the experiment, we temporarily used an SVM to assign class probabilities to the samples instead. This classifier seemed to be less confident in assigning a sample to a particular category (i.e. the maximum predicted probability among the 10 classes was generally lower). This possibly more fine-grained distinction between digit classes was in accordance with our qualitative inspection of the generated patterns. However, the SVM was much slower during testing time compared to the logistic regression which made us favour the latter in order to speed up the evaluation. Yet a simple fix of the over-confidence may also be to increase the L2 regularization penalty for the logistic regression classifier in order to bound the confidence of its predictions. Overall, one may consider alternative measures to evaluate the quality of the patterns generated by a DBM, also regarding their diversity.

Originally, we aimed to model visual hallucinations as well, thereby continuing the research by Reichert et al. (2013) on Charles Bonnet syndrome. Here, eye diseases lead to a complete loss of vision, yet many patients report having rich, vivid hallucinations of people, objects and scenes. In an attempt to simulate this loss of vision, Reichert et al. (2013) removed the visible layer of a DBM and successfully let the model hallucinate. Now we argue that our pruning approach of DBMs allows a further investigation of hallucinations in a partly lesioned visual system, given a carefully designed research question and experiment. This claim is based on the notion that pruning over-representations are also associated with visual hallucinations, as seen in various neurodegenerative and psychiatric diseases.

Schizophrenia is probably the most prominent disorder associated with auditory, olfactory, but also visual hallucinations (Waters et al., 2014). Notably, the Feinberg hypothesis (1982) that first linked schizophrenia with abnormal levels of synaptic pruning recently found strong support from a large genetic study (Sekar et al., 2016). In short, two complement component genes named C4A and C4B have the strongest known genetic association with schizophrenia and are likely to be involved in synaptic pruning during childhood and adolescence. Interestingly, complement genes are also involved in activity-dependent synaptic pruning of RGCs as taggers for unneeded synapses (Stevens et al., 2007). Sekar et al. (2016) confirmed expression levels of C4 in the RGCs of mice and further demonstrated that animals deficient of C4 had higher overlap between the RGCs of the two eyes, suggesting redundancy and less synaptic refinement. The authors concluded that elevated C4 expression may cause excessive synaptic pruning during adolescence, also explaining the typical time onset of the heritable brain disorder. Other examples of neurodegenerative diseases associated with visual hallucinations are several forms of dementia where the symptoms are correlated with brain volume loss in associative visual areas (Sanchez-Castaneda et al., 2010). Furthermore, a proportion of patients with Alzheimer's and Parkinson's disease commonly report experiencing visual hallucinations. Notably, these are also seen in connection with a decline of RGCs (Archibald et al., 2009; Armstrong, 2009).

Current theories of hallucinations stress to view their perceptual aspect in concordance with prior beliefs about the likelihood and precision of sensory input (Fletcher and Frith, 2009). Similarly, Reichert et al. (2013) interpreted the visual hallucinations seen in Charles Bonnet syndrome as a result of the cortex employing a hierarchical, generative model of the sensory input. Instead of completely depriving the network of its visible layer, our pruning technique locally damages the model. This may have effects on both the encoding and generative performance. As an attempt to investigate hallucinations, one could start by presenting a learned pattern to a "healthy," i.e. unpruned model. Propagating it forward and backward, it should roughly reconstruct the presented image. In an over-pruned model, we would expect the reconstruction error to be larger. That might be due to missing inhibitory or excitatory synapses (corresponding to negative and positive weights). One could also present random noise as input to the models and compare their encodings of such patterns. If the precision of sensory input is over-estimated compared to the precision of prior beliefs (Adams et al., 2015), an over-pruned model may be overly confident in encoding spurious correlations. Yet it is unclear how to measure such confidence. A first idea is to look at the responses of

the “stiff” hidden units to frequent and rare patterns. If they rather respond to frequent stimuli, their activity may correspond to a confidence or narrow posterior belief about stimuli. Although admittedly these hypotheses are premature, we deem the simulations of hallucinations in an over-pruned model to be an exciting research topic that is worth further investigation.

Chapter 6

Conclusions

In this project, we simulated naturally occurring synaptic pruning and cell death in the developing visual system. RBMs and DBMs were used to model the visual encoding of stimuli through RGCs and their sensory afferents. Two alternative local estimations of FI were used to compute the relative sensitivity of synapses and neurons.

First experiments of removing hidden units from RBMs trained on CIFAR-10 circular image patches provided interesting insights into the separation of "stiff" and "sloppy" units in the latent layer. As the visible layer increased, the importance of weights belonging to a unit became more heterogeneous, which motivated us to address the removal of individual connections. Here, the FI-motivated deletion of weights led to a restriction of the visual input to a few neighboring pixels per hidden unit while most of them became unconnected. However, the difficulties with evaluating the fit of RBMs inspired us to use labeled data in order to have two alternative evaluation measures. A DBM was trained on the handwritten digit dataset MNIST. RBMs and DBMs encoded the input data into hidden unit representations in order to reproduce it, making them generative models. This provided us with two evaluation measures. First, we evaluated the quality of the generated samples regarding their resemblance to digits. Second, we evaluated the hidden unit representations regarding their usefulness for a classifier.

Generally, pruning the first layer seemed to be more detrimental than starting to prune the second one. Yet, we discussed the limited comparability of the two layers in our experimental set-up due to receptive fields pre-restricting the connections in the first layer. We thus continued with a joint pruning of both layers at the same time. Here, we compared the removal of the *least* (FI pruning) with the *most* important parameters (anti-FI pruning). Strikingly, the anti-FI pruned model continuously performed worse

with respect to both evaluation measures. Initializing new DBMs with the weights deleted as indicated after the final pruning event, the performance difference between the models remained. This irreversible damage to the network induced by removing the most important parameters according to FI can be seen as a proof of concept for our method. It also suggests that there are highly sensitive parameters that cannot be replaced by others. The fact that the anti-FI pruned and worse performing model had 89% more parameters than the FI-pruned one opposes the "the more, the merrier" view that is present in today's machine learning community. The network size as in the sheer number of its parameters is clearly not the only aspect of sufficient task performance.

Similar to ANNs, the ideal size of sensory systems is difficult to determine a priori. Instead, the brain experiences an initial over-production of neurons and synapses, followed by subsequent pruning and apoptosis. We argue that the local computation of FI based on the covariance of presynaptic and postsynaptic firing rates may simulate the experience- and activity-dependent death of cells and synapses at a high level. Given that excessive pruning is associated with cognitive and perceptual impairments such as hallucinations, future research could apply our pruning technique to further investigate what may go wrong in diseases related to over-pruning.

Appendix A

Derivation of local FIM heuristic

The following heuristic estimate of the FIM diagonal entries for the weights of RBMs was derived by Rule (personal communication, July 2018).

The FIM value for weights is

$$\langle vh \rangle \left(1 - \langle vh \rangle\right)$$

This can be computed if there is some local mechanism for tracking and storing the correlation of pre- and post-synaptic firing rates $\langle vh \rangle$. This expectation is closely related to some more readily available local statistics, like the mean rates. Since v and h are binary in $\{0,1\}$, the expectation $\langle vh \rangle$ amounts to estimating the probability that v and h are simultaneously 1, i.e. $\Pr(v{=}1, h{=}1)$. One possible way to estimate this is noting the chain rule of conditional probability:

$$\Pr(v{=}1, h{=}1) = \Pr(v{=}1 | h{=}1) \cdot \Pr(h{=}1) = \langle h \rangle \Pr(v{=}1 | h{=}1)$$

Assuming that the hidden layer size is large and the hidden units are indepent, we may approximate the activity of all *other* hidden units apart from a particular h_i using mean-field. The activation of a visible unit is:

$$\Pr\left(v{=}1 | h\right) = \sigma\left(b^v + W\mathbf{h}\right),$$

where σ denotes the logistic sigmoid function. For mean-field, the hidden units are replaced with their mean firing rate:

$$\Pr\left(v{=}1 | \langle \mathbf{h} \rangle\right) = \sigma\left(b^v + W \langle \mathbf{h} \rangle\right)$$

If we want the contribution of all units *except* h_i, we would compute:

$$\Pr\left(v{=}1 | \langle \mathbf{h} \setminus h_i \rangle\right) = \sigma\left(b^v + W \langle \mathbf{h} \rangle - W_i \langle h_i \rangle\right)$$

If we want the mean-field activation assuming $h_i = 1$,

$$\begin{aligned}\Pr(v{=}1|h{=}1) &\approx \Pr\left(v{=}1|\langle \mathbf{h} \setminus h_i \rangle, h_i = 1\right) \\ &= \sigma\left(b^v + W\langle \mathbf{h} \rangle - W_i \langle h_i \rangle + W_i \cdot 1\right) \\ &= \sigma\left(b^v + W\langle \mathbf{h} \rangle + W_i(1 - \langle h_i \rangle)\right)\end{aligned}$$

Due to correlations, the mean-field might not be terribly accurate and we might want to get a better estimate by using the actual empirical mean rate for v instead. The empirical activation is then $\sigma^{-1}(\langle v \rangle)$ which can replace the $b^v + W\langle \mathbf{h} \rangle$ terms, leading to:

$$\Pr(v{=}1|h{=}1) \approx \sigma\left(\sigma^{-1}(\langle v \rangle) + W_i(1 - \langle h_i \rangle)\right)$$

$$\langle vh \rangle \approx \langle h \rangle \sigma\left(\sigma^{-1}(\langle v \rangle) + W_i(1 - \langle h_i \rangle)\right)$$

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Achille, A., Rovere, M., and Soatto, S. (2017). Critical learning periods in deep neural networks. *arXiv preprint arXiv:1711.08856*.

Adams, R. A., Huys, Q. J., and Roiser, J. P. (2015). Computational psychiatry: towards a mathematically informed understanding of mental illness. *J Neurol Neurosurg Psychiatry*, pages jnnp–2015.

Ahissar, M. and Hochstein, S. (2004). The reverse hierarchy theory of visual perceptual learning. *Trends in cognitive sciences*, 8(10):457–464.

Archibald, N. K., Clarke, M. P., Mosimann, U. P., and Burn, D. J. (2009). The retina in Parkinson's disease. *Brain*, 132(5):1128–1145.

Armstrong, R. A. (2009). Alzheimer's disease and the eye. *Journal of Optometry*, 2(3):103–111.

Becker, E. B. and Bonni, A. (2004). Cell cycle regulation of neuronal apoptosis in development and disease. *Progress in neurobiology*, 72(1):1–25.

Bengio, Y. and Delalleau, O. (2009). Justifying and generalizing contrastive divergence. *Neural computation*, 21(6):1601–1621.

Bengio, Y. et al. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127.

Berglund, M., Raiko, T., and Cho, K. (2015). Measuring the usefulness of hidden units in boltzmann machines with mutual information. *Neural Networks*, 64:12–18.

Berlucchi, G. (2011). Brain plasticity and cognitive neurorehabilitation. *Neuropsychological rehabilitation*, 21(5):560–578.

Bullmore, E. and Sporns, O. (2012). The economy of brain network organization. *Nature Reviews Neuroscience*, 13(5):336.

Casey, B., Tottenham, N., Liston, C., and Durston, S. (2005). Imaging the developing brain: what have we learned about cognitive development? *Trends in cognitive sciences*, 9(3):104–110.

Chechik, G., Meilijson, I., and Ruppin, E. (1999). Neuronal regulation: A mechanism for synaptic pruning during brain maturation. *Neural Computation*, 11(8):2061–2080.

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences*, 36(3):181–204.

Daniels, B. C., Chen, Y.-J., Sethna, J. P., Gutenkunst, R. N., and Myers, C. R. (2008). Sloppiness, robustness, and evolvability in systems biology. *Current opinion in biotechnology*, 19(4):389–395.

Dayan, P. and Abbott, L. F. (2001). *Theoretical neuroscience*. Cambridge, MA: MIT Press.

Deistler, M., Sorbaro, M., Rule, M. E., and Hennig, M. H. (2018). Local learning rules to attenuate forgetting in neural networks. *arXiv:1807.05097 [q-bio]*. arXiv: 1807.05097.

Denil, M., Shakibi, B., Dinh, L., De Freitas, N., et al. (2013). Predicting parameters in deep learning. In *Advances in neural information processing systems*, pages 2148–2156.

Eslami, S. A., Heess, N., Williams, C. K., and Winn, J. (2014). The shape Boltzmann machine: a strong model of object shape. *International Journal of Computer Vision*, 107(2):155–176.

Feinberg, I. (1982). Schizophrenia: caused by a fault in programmed synaptic elimination during adolescence? *Journal of psychiatric research*, 17(4):319–334.

Field, D. J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *Josa a*, 4(12):2379–2394.

Flavell, S. W. and Greenberg, M. E. (2008). Signaling mechanisms linking neuronal activity to gene expression and plasticity of the nervous system. *Annu. Rev. Neurosci.*, 31:563–590.

Fletcher, P. C. and Frith, C. D. (2009). Perceiving is believing: a Bayesian approach to explaining the positive symptoms of schizophrenia. *Nature Reviews Neuroscience*, 10(1):48.

Franklin, J. L. and Johnson, Eugene M, J. (1992). Suppression of programmed neuronal death by sustained elevation of cytoplasmic calcium. *Trends in neurosciences*, 15(12):501–508.

Ganmor, E., Segev, R., and Schneidman, E. (2011). Sparse low-order interaction network underlies a highly correlated and learnable neural population code. *Proceedings of the National Academy of Sciences*, 108(23):9679–9684.

Goldberg, J. L., Espinosa, J. S., Xu, Y., Davidson, N., Kovacs, G. T., and Barres, B. A. (2002). Retinal ganglion cells do not extend axons by default: promotion by neurotrophic signaling and electrical activity. *Neuron*, 33(5):689–702.

Gutenkunst, R. N., Waterfall, J. J., Casey, F. P., Brown, K. S., Myers, C. R., and Sethna, J. P. (2007). Universally sloppy parameter sensitivities in systems biology models. *PLoS computational biology*, 3(10):e189.

Han, S., Pool, J., Tran, J., and Dally, W. (2015). Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pages 1135–1143.

Hassibi, B., Stork, D. G., and Wolff, G. (1994). Optimal brain surgeon: Extensions and performance comparisons. In *Advances in neural information processing systems*, pages 263–270.

Hassibi, B., Stork, D. G., and Wolff, G. J. (1993). Optimal brain surgeon and general network pruning. In *Neural Networks, 1993., IEEE International Conference on*, pages 293–299. IEEE.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

Hinton, G. E. (2007). To recognize shapes, first learn to generate images. *Progress in brain research*, 165:535–547.

Hinton, G. E. (2012). A practical guide to training restricted Boltzmann machines. In *Neural networks: Tricks of the trade*, pages 599–619. Springer.

Hinton, G. E. (2016). Can the brain do back-propagation? Talk presented at the Stanford EE Computer Systems Colloquium, Stanford University. Available online from `https://web.stanford.edu/class/ee380/Abstracts/160427.html`.

Hinton, G. E., Dayan, P., Frey, B. J., and Neal, R. M. (1995). The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

Hinton, G. E., Plaut, D. C., and Shallice, T. (1993). Simulating brain damage. *Scientific American*, 269(4):76–82.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Hinton, G. E. and Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1:282–317.

Hiratani, N. and Fukai, T. (2018). Redundancy in synaptic connections enables neurons to learn optimally. *Proceedings of the National Academy of Sciences*, page 201803274.

Hochstein, S. and Ahissar, M. (2002). View from the top: Hierarchies and reverse hierarchies in the visual system. *Neuron*, 36(5):791–804.

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

Huttenlocher, P. R. (1990). Morphometric study of human cerebral cortex development. *Neuropsychologia*, 28(6):517–527.

Huttenlocher, P. R., Dabholkar, A. S., et al. (1997). Regional differences in synaptogenesis in human cerebral cortex. *Journal of comparative Neurology*, 387(2):167–178.

Innocenti, G. M. (1995). Exuberant development of connections, and its possible permissive role in cortical evolution. *Trends in neurosciences*, 18(9):397–402.

Innocenti, G. M. and Price, D. J. (2005). Exuberance in the development of cortical networks. *Nature Reviews Neuroscience*, 6(12):955.

Johnston, M. V. (2004). Clinical disorders of brain plasticity. *Brain and Development*, 26(2):73–80.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, page 201611835.

Köster, U., Sohl-Dickstein, J., Gray, C. M., and Olshausen, B. A. (2014). Modeling higher-order correlations within cortical microcolumns. *PLoS computational biology*, 10(7):e1003684.

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, Citeseer.

Le Roux, N. and Bengio, Y. (2008). Representational power of restricted boltzmann machines and deep belief networks. *Neural computation*, 20(6):1631–1649.

Le Roux, N., Heess, N., Shotton, J., and Winn, J. (2011). Learning a generative model of images by factoring appearance and shape. *Neural Computation*, 23(3):593–650.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. (2006). A tutorial on energy-based learning. *Predicting structured data*, 1(0).

LeCun, Y., Cortes, C., and Burges, C. (2010). The MNIST database of handwritten digits. Available from: `http://yann.lecun.com/exdb/mnist`.

LeCun, Y., Denker, J. S., and Solla, S. A. (1990). Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605.

Levkovitch-Verbin, H. (2015). Retinal ganglion cell apoptotic pathway in glaucoma: initiating and downstream mechanisms. In *Progress in brain research*, volume 220, pages 37–57. Elsevier.

Lledo, P.-M., Alonso, M., and Grubb, M. S. (2006). Adult neurogenesis and functional plasticity in neuronal circuits. *Nature Reviews Neuroscience*, 7(3):179.

Low, L. K. and Cheng, H.-J. (2006). Axon pruning: an essential step underlying the developmental plasticity of neuronal connections. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 361(1473):1531–1544.

Lowel, S. and Singer, W. (1992). Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity. *Science*, 255(5041):209–212.

Machta, B. B., Chachra, R., Transtrum, M. K., and Sethna, J. P. (2013). Parameter space compression underlies emergent theories and predictive models. *Science*, 342(6158):604–607.

Mariet, Z. and Sra, S. (2016). Diversity networks. *Proceedings of ICLR*.

Meier, P., Finch, A., and Evan, G. (2000). Apoptosis in development. *Nature*, 407(6805):796.

Meyer-Franke, A., Wilkinson, G. A., Kruttgen, A., Hu, M., Munro, E., Hanson Jr, M. G., Reichardt, L. F., and Barres, B. A. (1998). Depolarization and cAMP elevation rapidly recruit TrkB to the plasma membrane of CNS neurons. *Neuron*, 21(4):681–693.

Mink, J. W., Blumenschine, R. J., and Adams, D. B. (1981). Ratio of central nervous system to body metabolism in vertebrates: its constancy and functional basis. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 241(3):R203–R212.

Morcos, A. S., Barrett, D. G., Rabinowitz, N. C., and Botvinick, M. (2018). On the importance of single directions for generalization. *arXiv preprint arXiv:1803.06959*.

Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research*, 37(23):3311–3325.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Perry, V., Henderson, Z., and Linden, R. (1983). Postnatal changes in retinal ganglion cell and optic axon populations in the pigmented rat. *Journal of Comparative Neurology*, 219(3):356–368.

Purves, D., Augustine, G. J., Fitzpatrick, D., Hall, W., LaMantia, A., McNamara, J., and White, L. (2008). Neuroscience. *Sinauer Associates, Sunderland, Mass*.

Reed, R. (1993). Pruning algorithms-a survey. *IEEE transactions on Neural Networks*, 4(5):740–747.

Reichert, D. P. (2012). *Deep Boltzmann Machines as Hierarchical Generative Models of Perceptual Inference in the Cortex*. PhD thesis, University of Edinburgh, Edinburgh, UK.

Reichert, D. P., Series, P., and Storkey, A. J. (2013). Charles Bonnet syndrome: evidence for a generative model in the cortex? *PLoS computational biology*, 9(7):e1003134.

Rule, M. (2018). Derivation of local FIM heuristic. Personal communication.

Rule, M., Sorbaro, M., and Hennig, M. (2018). Optimal encoding in stochastic latent-variable models. *arXiv preprint arXiv:1802.10361*.

Salakhutdinov, R. and Larochelle, H. (2010). Efficient learning of deep Boltzmann machines. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 693–700.

Sanchez-Castaneda, C., Rene, R., Ramirez-Ruiz, B., Campdelacreu, J., Gascon, J., Falcon, C., Calopa, M., Jauma, S., Juncadella, M., and Junque, C. (2010). Frontal and associative visual areas related to visual hallucinations in dementia with Lewy bodies and Parkinson's disease with dementia. *Movement Disorders*, 25(5):615–622.

Sánchez-Gutiérrez, M., Albornoz, E. M., Rufiner, H. L., and Close, J. G. (2017). Post-training discriminative pruning for rbms. *Soft Computing*, pages 1–15.

Scholl, C. (2018). Pruning over-representations in latent encoder models. *Informatics Project Proposal*.

Sekar, A., Bialas, A. R., De Rivera, H., Davis, A., Hammond, T. R., Kamitaki, N., Tooley, K., Presumey, J., Baum, M., Van Doren, V., et al. (2016). Schizophrenia risk from complex variation of complement component 4. *Nature*, 530(7589):177.

Sengelaub, D. and Finlay, B. (1982). Cell death in the mammalian visual system during normal development: I. Retinal ganglion cells. *Journal of Comparative Neurology*, 204(4):311–317.

Shoham, S., OConnor, D. H., and Segev, R. (2006). How silent is the brain: is there a dark matter problem in neuroscience? *Journal of Comparative Physiology A*, 192(8):777–784.

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. *Parallel distributed processing: explorations in the microstructure of cognition*, 1.

Stevens, B., Allen, N. J., Vazquez, L. E., Howell, G. R., Christopherson, K. S., Nouri, N., Micheva, K. D., Mehalow, A. K., Huberman, A. D., Stafford, B., et al. (2007). The classical complement cascade mediates cns synapse elimination. *Cell*, 131(6):1164–1178.

Sutskever, I. and Tieleman, T. (2010). On the convergence properties of contrastive divergence. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 789–795.

Sutton, J., Beis, J., and Trainor, L. (1988). A hierarchical model of neocortical synaptic organization. *Mathematical and Computer Modelling*, 11:346–350.

Takesian, A. E. and Hensch, T. K. (2013). Balancing plasticity/stability across brain development. In *Progress in brain research*, volume 207, pages 3–34. Elsevier.

Tieleman, T. (2008). Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071. ACM.

Tostado, P., Wiest, M., and Yepremyan, A. (2017). Biologically-inspired sparse restricted Boltzmann machines.

Turrigiano, G. G. and Nelson, S. B. (2000). Hebb and homeostasis in neuronal plasticity. *Current opinion in neurobiology*, 10(3):358–364.

Waters, F., Collerton, D., Ffytche, D. H., Jardri, R., Pins, D., Dudley, R., Blom, J. D., Mosimann, U. P., Eperjesi, F., Ford, S., et al. (2014). Visual hallucinations in the psychosis spectrum and comparative information from neurodegenerative disorders and eye disease. *Schizophrenia bulletin*, 40(Suppl_4):S233–S245.

Wu, J., Mazur, T. R., Ruan, S., Lian, C., Daniel, N., Lashmett, H., Ochoa, L., Zoberi, I., Anastasio, M. A., Gach, H. M., et al. (2018). A deep Boltzmann machine-driven level set method for heart motion tracking using cine mri images. *Medical image analysis*, 47:68–80.

Yuan, J. and Yankner, B. A. (2000). Apoptosis in the nervous system. *Nature*, 407(6805):802.

Zanotto, M., Volpi, R., Maccione, A., Berdondini, L., Sona, D., and Murino, V. (2017). Modeling retinal ganglion cell population activity with restricted Boltzmann machines. *arXiv preprint arXiv:1701.02898*.