Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

Johannes Wiger Austefjord

An antiferromagnetic and insulating leaky integrate-and-fire neuron and its application

Master's thesis in Applied Physics and Mathematics
Supervisor: Alireza Qaiumzadeh

July 2022

NTNU
Norwegian University of
Science and Technology

Johannes Wiger Austefjord

An antiferromagnetic and insulating leaky integrate-and-fire neuron and its application

Master's thesis in Applied Physics and Mathematics
Supervisor: Alireza Qaiumzadeh
July 2022

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

NTNU
Norwegian University of
Science and Technology

Abstract

As the size of transistors begins to reach the extent of a couple of atoms, keeping up with Moore's law has become increasingly difficult. Furthermore, new methods of computing are pursued by researchers in the quest for more energy-efficient hardware. The intersection between analog and neuromorphic computing is one candidate to spark a transition into a new computing regime with specialized hardware for specific tasks. In this thesis, numerical micro-magnetic simulations of an antiferromagnetic and insulating spiking neuron yield a proof-of-concept for a new computing unit, that seeks to overcome equivalent metallic and ferromagnetic neurons in terms of speed and energy consumption. The spiking neuron could be a part of a hardware realization of a neural network. The neurons are set up to mimic the behavior of leaky integrate-and-fire neurons. Two different versions of the neuron will be given, and they will be compared in detail. The proposed spiking neuron is a new way of envisioning how applied spintronics can yield energy-efficient specialized computing hardware. The state of the proposed neuron is coded into the position of a magnetic domain wall, which can be controlled by exciting spin waves in the antiferromagnet and a voltage-controlled magnetic anisotropy. The simulations will show that in principle a spiking neuron that operates at pico-second time scale is possible. Furthermore, results indicate that the neurons could potentially operate at an energy consumption of $\sim 100\,\mathrm{aJ}$ per spike.

Sammendrag

Ettersom størrelsen på transistorer begynner å tilsvare utstrekningen til et par atomer, blir det stadig vanskeligere å holde tritt med Moore's lov. Samtidig forskes det på nye metoder for mer energieffektiv maskinvare. Krysningen mellom analog og nevromorf prosessering gir muligheter til å utløse en overgang til et nytt regime innen datateknologi, med datamaskiner som er spesialisert til å løse spesifikke oppgaver. I denne masteroppgaven vil numeriske mikromagnetiske simuleringer av et antiferromagnetisk og isolerende impulsnevron gi innblikk i det som kan bli en ny prosesseringsenhet. Dette er med den hensikt i å overgå metalliske og ferromagnetiske tilsvarende impulsnevron når det gjelder hastighet og energiforbruk. Nevronene er designet for å etterligne funksjonaliteten til lekkende integrer-og-fyr nevroner. To forskjellige versjoner av nevronet vil bli foreslått og sammenlignet. Det antiferromagnetiske impulsnevronet er et forslag på hvordan anvendt spintronikk kan gi energieffektive datamaskiner som er spesialisert til å løse konkrete og krevende oppgaver. Tilstanden til det foreslåtte nevronet er kodet inn i posisjonen til en magnetisk domenevegg som kontrolleres av spinnbølger og spenningskontrollert magnetisk anisotropi. Simuleringene vil vise at impulsnevronet i prinsippet kan operere på en picosekund tidsskala. Videre indikerer resultatene at nevronet kan ha et energiforbruk på $\sim 100\,\mathrm{aJ}$ per impuls.

Preface

This thesis finalizes a Master of Science in Applied Physics and Mathematics with a specialization in Technical Physics at the Department of Physics, Norwegian University of Science and Technology (NTNU). The work has been carried out at the Center for Quantum Spintronics (QuSpin), and supervised by Researcher Alireza Qaiumzadeh.

I want to use this opportunity to express my gratitude towards the people that have helped me with this thesis. First, I would like to thank Alireza for being my supervisor during the final year of my studies, and always providing me with assistance when required. I am glad I got the opportunity to challenge myself with this exciting project.

Furthermore, I wish to thank Verena Johanna Brehm for helping me with this project and providing me with help and ideas whenever I have asked for it. Additionally, I would like to mention Thorvald Molthe Ballestad for teaching me how to make better figures, and for being a general sparring partner when discussing ideas. Acknowledgments are also in order for Dr. Serban Lepadatu[1] for graciously providing help whenever needed with his computational software, Boris computational spintronics.

Thanks to my family for always supporting me, and especially to my mother and father for all the help you have provided.

-Johannes Wiger Austefjord
Trondheim, Norway
July, 2022

---

[1] Jeremiah Horrocks Institute for Mathematics, Physics and Astronomy - The University of Central Lancashire.

Contents

Symbols and conventions

This thesis includes ideas and concepts from a wide range of scientific fields. Most prominent are probability and information theory, machine learning, neuroscience and spintronics. An effort has been made to keep to standard notation, but in some cases, this has been impossible. The symbol H (Hamiltonian, fields, entropy) is perhaps particularly overloaded. The following list indicates where notation may differ from what is the standard practice in some fields, and hopefully removes confusion where similar symbols are used for quite different quantities. All variables are defined as they are introduced in the text.

- H is the Hamiltonian.
- $\boldsymbol{\mathcal{H}}$ is effective fields.
- $\boldsymbol{h}$ is applied external magnetic field.
- $\boldsymbol{l}$ is the Néel vector.

Furthermore the following notation is used

a	Scalar.
$\boldsymbol{a}$	Vector.
$\boldsymbol{A}$	Matrix.
$\overleftrightarrow{\boldsymbol{a}}$	Tensor.
$\mathbf{a}$	Stochastic variable.
∂_x	Derivative with respect to x.

List of abbreviations

AFM	antiferromagnet
ANN	artificial neural network
CMOS	complementary metal-oxide-semiconductor
DMI	Dzyaloshinskii–Moriya interaction
DW	domain wall
HM	heavy metal
IP	in-plane
iSHE	inverse spin Hall effect
FM	ferromagnet
FNN	feedforward neural network
LCP	left circularly polarized
LIF	leaky integrate-and-fire
LLG	Landau–Lifshitz–Gilbert
MP	McCulloch-Pitts
OOP	out-of-plane
RCP	right circularly polarized
SHA	spin Hall angle
SHE	spin Hall effect
SNN	spiking neural network
sLLG	stochastic Landau–Lifshitz–Gilbert
SRM_0	standard response model of zeroth order
STT	spin-transfer torque
VCMA	voltage-controlled magnetic anisotropy

1 Introduction

The computing paradigm of today hinges on complementary metal-oxide-semiconductor (CMOS) transistors [1]. Logic operations are achieved by moving charges in the form of electrons. For several decades the development of new transistors has followed the famous *Moore's law* that states that the number of transistors one can fit on a chip will grow exponentially, or equivalently that transistors will decrease in size correspondingly [2]. This has allowed the digital age we today live in. Less famously, there exists another rule describing the innovation of new transistors called *Dennard scaling* which states that the power consumption of transistors is constant as their size decreases. To this day Moore's law is still followed. However, the new technology is no longer capable of complying with Dennard scaling [3]. This comes as a consequence of transistors starting to reach the size of a couple of atoms, causing larger current leaks caused by direct tunneling through the transistor gate. This has sparked an interest in finding new ways of building logic units capable of arithmetic, integrating different types of computing hardware that each has its specialization in a single computer [4].

It is difficult to relate the computational power of the brain to a computer. However, when making a human compete against a computer at a given task, one can get some insight by comparing results and performance. In 2016, AlphaGo, a computer program by Google Deepmind played 18 times world champion Lee Sedol in a 5-game match in the board game *Go* [5]. The program calculated the next move by a search tree method combined with a deep neural network. It is speculated that the power consumption of the AlphaGo program during this match was $\sim 1.2\,\mathrm{MW}$ [6]. On the other side, the power consumption of a human brain is about $20\,\mathrm{W}$. Although AlphaGo won $4-1$, one can assume that the players were somewhat evenly matched. The fact that the human was able to compete with the machine with a 60 000 times smaller power consumption is quite remarkable.

It is not just in power efficiency the brain outperforms machines. Humans outperform machines in some tasks that come naturally to us but are difficult to describe formally in a computer program [7]. Such tasks could be image and speech recognition [8]. The introduction of deep learning has spawned a new age for artificial intelligence, and can now begin to compete with humans. However, there are still aspects where the brain exceeds the computer. All this is although computers operate at substantially slower frequencies than the brain.

Neuromorphic computing is the concept of engineering either software or hardware that operates somewhat equal to the human brain [9]. The goal of neuromorphic computing is to identify how the brain works and then implement systems that mimic some functionality of the brain, to gain some advantage over conventional computing, such as improved power efficiency. A famous example of this is artificial neural networks. A subset of artificial neural networks that tries to imitate the spiking behavior of neurons in the brain is spiking neural networks. These types of networks can be implemented with great energy efficiency on specialized neuromorphic chips, as will be discussed in chapter 3. An example of this is the *SpiNNaker* machine, which is capable of modeling spiking neural networks with the size of a rat brain in real-time [8].

In the human brain, memory is stored in the same location where the computation happens [10]. This is a crucial distinction from how a conventional computer operates. In 1945 Von Neumann envisioned some rules for how a computer could be built, inspired by how the human brain functions [11]. This is today called the *Von Neumann architecture*, and conventional computers are built on many of these principles. Von Neumann identified the main components of a computer, among others memory, M, and a control arithmetic unit, CA. Today they respectively correspond to RAM and the CPU. Furthermore, Von Neumann identified that data would have to be transferred from M to CA and back to M for each computation. This is the case in modern conventional computers as they still have separated memory and logic units, as opposed to the human brain. This transfer of data sets a cap on computational performance and is called the *Von Neumann bottleneck* [12]. Additionally, it requires a substantial amount of energy to move a lot of information back and forth between memory and logic unit [13]. Implementing hardware that is not limited by the Von Neumann bottleneck is a crucial problem in neuromorphic computing. The *Loihi* chip by Intel is an example of such a device that implements non-Von Neumann architecture [14, 15], built with regular CMOS technology, that operates with great power efficiency.

Analog computing is a contrary paradigm to digital computing. It is based on building a physical system that is governed by the same equations that one is trying to solve [16]. The idea is that by initiating the system with some given initial conditions, it will naturally evolve towards the solution. Famously analog computers were used to predict tides, long before the entrance of digital electronic computers. Traditionally one would build analog computers consisting of gears and wheels. In the 1980s digital computers dethroned analog computers. Nevertheless, researchers are now considering analog computers as a potential candidate for supplementing digital computers when problems become computationally expensive. Analog computing trades accuracy and reproducibility for shorter computing times and power consumption [17]. Today analog computers are used in conjunction with standard digital computers to form hybrid computers, and it is believed that analog computers can provide a more energy-efficient alternative than digital computers [18].

In recent years there has been development in using spintronic components to realize such systems [13] and the combination of neuromorphic computing and spintronics in *neuromorphic spintronics* has gained a lot of interest. Spintronics is a field of condensed matter physics that concerns itself with understanding the properties of the intrinsic spin of the electrons, and how it can be used to create new electrical components [19]. This would be a fundamentally different approach to building computing machines than what is offered by CMOS technology. The prospect of using the spin of electrons as information carriers rather than electrons themselves provides the opportunity for a more power-efficient computing regime. Spintronic devices share some characteristics of the brain, such as non-linearity and plasticity [20], and could achieve a substantial reduction in the energy cost of computing.

One of the objectives of neuromorphic and analog computing is to bridge the gap in power consumption between the human brain and computers. In all electrical components except for super-conductors, there is some power dissipation from electrical energy to heat due to *Joule heating* [21]. Joule heating is a consequence of electrons colliding as they move through the wire. An insulating material that is stimulated through electron spin and not moving electrons do not suffer from this drawback and can be more efficient. Both ferromagnetic and antiferromagnetic spintronic devices is proposed. However, we will see that antiferromagnetic may have a better potential than ferromagnetic devices as they work at higher frequencies, giving an edge when it comes to operating speed.

This master's thesis revolves around a proof-of-concept numerical study of a spintronic device, that could be implemented in a hardware realization of a spiking neural network. It consists of an insulating antiferromagnetic material, which can be stimulated without the use of electrical currents, in an attempt to tackle the issue of Joule heating [22]. Furthermore, it utilizes its physical behavior to do arithmetic operations such as integration. Moreover, it has a natural short-term memory embedded into its behavior, that could help transcend the Von Neumann bottleneck. It is both a neuromorphic and analog computing unit. Chapters 2 and 3 will give introduction to the necessary theory to understand the proposed device. Then chapter 4 will explain the implementation of the neuron, and how it works, while chapter 5 will give an example of how it may be used to implement a spiking neural network. At last, a summary and an outlook will be given in chapter 6.

2 Introduction to spintronics

This chapter will introduce concepts of magnetism and spintronics that will lay the foundation for the implementation of a hardware spiking neuron based on antiferromagnetic domain wall movement. The first section will concern general magnetization dynamics. More specialized topics will follow in the subsequent section and will give the necessary insight to understand the physics behind the proposed neuron in chapter 4.

2.1 Magnetization dynamics

This section explores the concept of magnetism in terms of electron spin and the free energy of magnetic materials.

2.1.1 Electron spin and magnetization

Elementary particles has angular momentum that comes about in two forms, orbital and intrinsic. In quantum mechanics these quantities are described as *operators*, rather than vectors. The total angular momentum operator $\tilde{\boldsymbol{J}}$ is the sum of the two contributions [23]

$$\tilde{\boldsymbol{J}} = \tilde{\boldsymbol{L}} + \tilde{\boldsymbol{S}} \tag{2.1}$$

where $\tilde{\boldsymbol{L}}$ and $\tilde{\boldsymbol{S}}$ is the orbital and intrinsic angular momentum operators. One typically refers to $\tilde{\boldsymbol{S}}$ as simply the *spin* of the particle. A charged particle with angular momentum will create a magnetic moment

$$\tilde{\boldsymbol{\mathcal{M}}} = \gamma_L \tilde{\boldsymbol{L}} + \gamma_S \tilde{\boldsymbol{S}}. \tag{2.2}$$

The terms γ_L and γ_S is the respective gyromagnetic ratios. In general the gyromagnetic ratio of a particle is given as $\gamma \equiv gq/2m$ where g is called the *g-factor*. q and m is the charge and the mass of the particle. For many magnetic materials the contribution from $\tilde{\boldsymbol{L}}$ is quenched [24], and for the electron one can approximate

$$\tilde{\boldsymbol{\mathcal{M}}} \approx \gamma_e \tilde{\boldsymbol{S}}, \quad \gamma_\mathrm{e} = g_\mathrm{eff} \frac{q_\mathrm{e}}{2m_\mathrm{e}} \tilde{\boldsymbol{S}} \tag{2.3}$$

where $g_e \approx 2$ [25], and q_e is the electron charge. γ_e is called the gyromagnetic ratio of the electron[2]. The magnetic moment operator is in other words dependent of the spin of the electron. To connect the classical field theory to the quantum theory of spins, one can write the continuous magnetization vector $\boldsymbol{M}(\boldsymbol{r})$ as

$$\boldsymbol{M}(\boldsymbol{r}) = \frac{\langle \sum_{i \in \mathcal{V}_\mathrm{av}} \tilde{\boldsymbol{\mathcal{M}}}_i \rangle}{\mathcal{V}_\mathrm{av}(\boldsymbol{r})} \tag{2.4}$$

where $\mathcal{V}_\mathrm{av}$ is an small averaging volume around the position $\boldsymbol{r}$ and $\tilde{\boldsymbol{\mathcal{M}}}_i$ magnetic moment operator from an electron i. $\boldsymbol{M}(\boldsymbol{r})$ describes the magnetic moment per unit volume at $\boldsymbol{r}$. Note that the averaging procedure above reduced the operator $\tilde{\boldsymbol{\mathcal{M}}}$ to a real-valued vector. This can be seen as a semi-classical micromagnetic approach for magnetism, and the rest of this thesis will utilize this regime [26]. In contrast to the micromagnetic approach, there is the atomistic approach which treats magnetic moment as a vector *per atom*.

Throughout this paper one assumes that the amplitude of the magnetization as constant and finite in both time and space, i.e. $|\boldsymbol{M}(\boldsymbol{r}, t)| = M_\mathrm{s}$, where M_s is the saturation magnetization. It is then common to introduce the normalized magnetization vector

$$\boldsymbol{m} \equiv \frac{\boldsymbol{M}}{M_\mathrm{s}}. \tag{2.5}$$

In the rest of this text, we will use the name ”magnetization” for $\boldsymbol{m}$, as this is the order parameter that will be used to describe magnetic systems. The fact that $|\boldsymbol{m}| = 1$ can be used to deduce a restriction on $\boldsymbol{m}$. Taking the time derivative on both sides of $\boldsymbol{m} \cdot \boldsymbol{m} = 1$ one deduces the following important relation

$$\boldsymbol{m} \cdot \frac{\partial \boldsymbol{m}}{\partial t} = 0. \tag{2.6}$$

This restricts the motion of $\boldsymbol{m}$ at a given point in space to the unit sphere. We will use this equation later when deriving the dynamics of $\boldsymbol{m}$.

---

[2] γ_e is negative, as $q_\mathrm{e} < 0$. In some text γ_e is defined as its absolute value. To avoid confusion $|\gamma_\mathrm{e}|$ will be used in the rest of this thesis, such that the subsequent equations follows the most common sign convention.

2.1.2 Free energy of ferromagnetic materials

The motion of $\boldsymbol{m}$ is a determined by the free energy $F[\boldsymbol{m}]$ which is a functional of $\boldsymbol{m}$. The free energy, F, is functional of $\boldsymbol{m}(\boldsymbol{r})$. F is expressed as an integral of the free energy density, f, taken over the sample volume V [27, 26]

$$F[\boldsymbol{m}(\boldsymbol{r})] = \int_V f(\boldsymbol{m}, \nabla\boldsymbol{m}, \boldsymbol{r}) d^3r. \tag{2.7}$$

The free energy density can be written as a sum of relevant contributions. The contributions stem from $\boldsymbol{m}$ interacting with itself, and fields from the outside world. Here we will only consider the relevant interactions for describing the AFM neuron in chapter 4.

Exchange interaction

The exchange interaction is one of the most dominant interactions in magnetic materials [28]. It comes as a consequence of interaction between electrons at neighboring atoms, and describes how spins will seek to align with other spins in their vicinity. The exchange is written as [26]

$$f_{\text{exchange}} = A\left[\left(\frac{\partial\boldsymbol{m}}{\partial x}\right)^2 + \left(\frac{\partial\boldsymbol{m}}{\partial y}\right)^2 + \left(\frac{\partial\boldsymbol{m}}{\partial z}\right)^2\right] = A\nabla\boldsymbol{m} : \nabla\boldsymbol{m}, \tag{2.8}$$

where A is a constant that parametrizes the interaction strength [26]. The notation $\boldsymbol{A} : \boldsymbol{B} \equiv \mathrm{Tr}(\boldsymbol{A}\boldsymbol{B}^\dagger)$ is called the *Frobenius inner product* [29].

Crystalline anisotropy

Crystalline anisotropy generates an energy contribution where the magnetization becomes dependent on the crystal axes [28]. It can arise in a variety of forms, here we shall only consider the simple case of

$$f_{\text{anisotropy}} = -K_{\text{easy}}(\boldsymbol{m} \cdot \mathbf{e}_{\text{easy}})^2 + K_{\text{hard}}(\boldsymbol{m} \cdot \mathbf{e}_{\text{easy}})^2, \tag{2.9}$$

where $K_{\text{easy}} \geq 0$ and $K_{\text{hard}} \geq 0$. $\mathbf{e}_{\text{easy}}$ and $\mathbf{e}_{\text{hard}}$ are unit vectors along the so called easy- and hard axis. Anisotropy can vary several orders of magnitude. In general, the coefficients K_{hard} and K_{easy} can be made position dependent [30], as we shall see in section 2.2.

DMI

The *Dzyaloshinskii–Moriya interaction* (DMI), also known as asymmetric exchange, appears in materials with low symmetry and is given as

$$f_{\mathrm{DMI}} = D\boldsymbol{m} \cdot (\nabla \times \boldsymbol{m}), \tag{2.10}$$

where D is a constant [28]. This interaction favours a tilting of neighboring spins, competing with the exchange interaction and can give rise to structures like domain walls and skyrmions.

One way to deal with the derivative of $\boldsymbol{m}$ along the boundaries is

$$\frac{\partial \boldsymbol{m}}{\partial \hat{n}} = \frac{2}{A}\hat{n} \times \boldsymbol{m}, \tag{2.11}$$

where A is the exchange constant from Eq. (2.8), and $\hat{n}$ is the unit surface normal vector.

Zeeman interaction

The final interaction we will consider is the *Zeeman interaction.* It arises when applying an external field to a magnetic material. The field will couple to magnetization

$$f_{\mathrm{Zeeman}} = -M_{\mathrm{s}}\boldsymbol{m} \cdot \mu_0 \boldsymbol{h}, \tag{2.12}$$

where μ_0 is the magnetic permeability of vacuum [27].

Total free energy for ferromagnets

Collecting all the terms from above, we can write the total free energy density as

$$\begin{aligned} f^{\mathrm{FM}} =& A\left[\left(\frac{\partial \boldsymbol{m}}{\partial x}\right)^2 + \left(\frac{\partial \boldsymbol{m}}{\partial y}\right)^2 + \left(\frac{\partial \boldsymbol{m}}{\partial z}\right)^2\right] - K_{\mathrm{easy}}(\boldsymbol{m} \cdot \mathbf{e}_{\mathrm{easy}})^2 \\ &+ K_{\mathrm{hard}}(\boldsymbol{m} \cdot \mathbf{e}_{\mathrm{hard}})^2 + D\boldsymbol{m} \cdot (\nabla \times \boldsymbol{m}) - M_{\mathrm{s}}\boldsymbol{m} \cdot \mu_0 \boldsymbol{h}. \end{aligned} \tag{2.13}$$

Here the superscript FM indicates that this expression is valid for ferromagnets. It is worth noting that all terms in f^{FM} are invariant under the flip of a spin, except for the Zeeman energy term from the applied external magnetic field $\boldsymbol{h}$. Ignoring this term one can in other words write $f^{\mathrm{FM}}(\boldsymbol{m}) = f^{\mathrm{FM}}(-\boldsymbol{m})$. This leads to a two-fold degenerate ground state of the system.

2.1.3 Ferromagnetism and antiferromagnetism

Materials that exhibit some sort of magnetic order is often separated into five different categories; *diamagnetism*, *paramagnetsim*, *ferromagnetism*, *antiferromagnetism* and *ferrimagnetism* [31]. Of these only the last three have magnetic ordering in the absence of any applied magnetic field. Fig. 2.1 displays a ferromagnetic and antiferromagnetic ordering and will be at the center of attention in this thesis. It is the exchange interaction that determines whether a material is a ferromagnet or an antiferromagnet. Considering only the exchange interaction in Eq.(2.8), an energy minimum exists when $\nabla \boldsymbol{m} = 0$ given that $A > 0$. This is the trademark of a ferromagnet (FM), i.e. that the state of the system is achieved when all spins align in the same direction, enforced by a positive exchange constant.

An antiferromagnet (AFM) works inherently in a different manner. In a discretized model it corresponds to an negative exchange interaction between neighboring spins, yielding a $\uparrow\downarrow$ ground state for a two spin system [28]. However, in the continuos case it is commonly modelled by two superimposed vector fields $\boldsymbol{m}_\mathrm{A}(\boldsymbol{r})$ and $\boldsymbol{m}_\mathrm{B}(\boldsymbol{r})$.

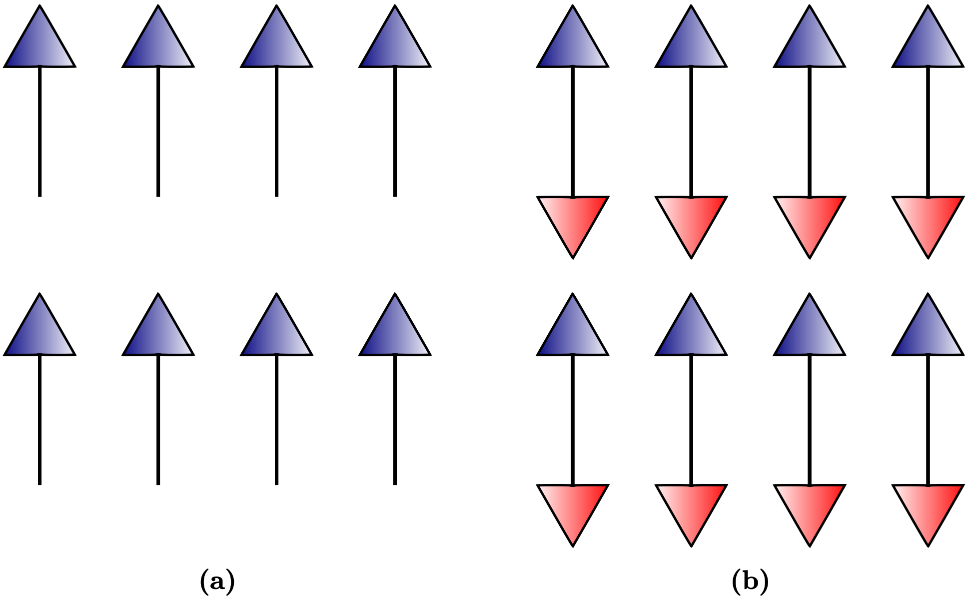


Figure 2.1: Ground states of magnetic system with only exchange interactions. **(a)** ferromagnet and **(b)** antiferromagnet.

Fig. 2.1b differs a bit from another common way of modeling antiferromagnets [28]. One will often see that the two sublattices are shifted with respect to each other, creating an alternating up-down pattern, ↑↓↑↓ as opposed to ↕↕↕↕. However, in the continuous regime, there is no distinction between the two models. The reason for this divergence from the widespread practice is simply because the numerical method that is used in this thesis [26], which will be described in detail later, discretize $\boldsymbol{m}_\mathrm{A}(\boldsymbol{r})$ and $\boldsymbol{m}_\mathrm{B}(\boldsymbol{r})$ at the same points in space. This yields a similar model of the system as in fig 2.1b. It is worth noting that although figures display a discrete model, the equations are given in a continuous model.

In this numerical model, the only interaction between the two sublattices A and B is the *antiferromagnetic homogenous exchange* and contributes to the free energy density as [27, 26]

$$f_i^{\text{ex, h}} = 4A_\mathrm{h}\boldsymbol{m_i} \cdot \boldsymbol{m_j}, \quad i, j \in \{\mathrm{A, B}\} \wedge i \neq j. \tag{2.14}$$

A_h parametrizes the interaction strength, and favours an antiferromagnetic configuration when positive. This can be understood as $f_i^{\text{ex, h}}$ has a minimum for when $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$ points in opposite directions. We shall consider one other term that can couple the two sublattices together and that is called *homogenous DMI* [26, 32]

$$f_i^{\text{DMI, h}} = -\eta_i(\boldsymbol{D}_\mathrm{h} \times \boldsymbol{m}_j) \cdot \boldsymbol{m}_i, \quad i, j \in \{\mathrm{A, B}\} \wedge i \neq j, \tag{2.15}$$

where $\eta_\mathrm{A} = -\eta_B = 1$, and $\boldsymbol{D}_\mathrm{h}$ is the homogenous DMI vector, and determines the strength of the interaction and the spin tilt direction. The total free energy density for each sublattice is then

$$f_i = f^{\mathrm{FM}} + f_i^{\text{ex, h}} + f_\mathrm{i}^{\text{DMI, h}}, \tag{2.16}$$

where f^{FM} is given in Eq. (2.13). Due to previously discussed symmetry of f^{FM} some of characteristic of two sublattices $\boldsymbol{m}_\mathrm{A}$ and $\boldsymbol{m}_\mathrm{B}$ is similar, just in opposite directions $\boldsymbol{m}_\mathrm{A} \simeq -\boldsymbol{m}_\mathrm{B}$ as the antiferromagnetic homogeneous exchange is relatively strong compared to the other terms.

It is possible and common to describe AFMs by two different order parameters than $\boldsymbol{m}_\mathrm{A}$ and $\boldsymbol{m}_\mathrm{B}$, namely the staggered order parameter, or *Néel vector*, $\boldsymbol{l}$, and the magnetization field $\tilde{\boldsymbol{m}}$ [33, 34]. They are defined as

$$\boldsymbol{l} = \frac{\boldsymbol{m}_\mathrm{A} - \boldsymbol{m}_\mathrm{B}}{2}, \quad \tilde{\boldsymbol{m}} = \frac{\boldsymbol{m}_\mathrm{A} + \boldsymbol{m}_\mathrm{B}}{2}. \tag{2.17}$$

In the limit where there is a strong interaction between lattices A and B, one can write

$$|\tilde{\boldsymbol{m}}| \to 0, \quad \text{when} \quad A_\mathrm{h} \gg 1. \tag{2.18}$$

It is then possible to use only $\boldsymbol{l}$ as an order parameter.

When it comes to building computing components, AFM has some advantages over FM. Since AFM has a small net magnetization $\tilde{\boldsymbol{m}}$ they do not induce large stray fields [35]. This is advantageous, as AFMs can be stacked close to each other without interfering. Another desirable feature of AFMs is that they display resonance frequencies in the THz range, as opposed to GHz in FMs. This yields the opportunity for faster computing units.

In conclusion, one can numerically simulate an AFM, by simulating two superimposed FMs, coupled together through the antiferromagnetic homogenous exchange, and the homogenous DMI. This makes the equation of motion for describing FMs and AFMs quite similar [26], as we will see in the coming section. The next subsection describes the equations of motion for FMs, which are easily adapted to AFMs by solving for two superimposed FMs coupled by Eq. 2.14.

2.1.4 The Landau-Lifshitz-Gilbert equation

As previously stated, magnetic materials is described by their magnetization $\boldsymbol{m}$. The magnetization will evolve over time and the *Landau-Lifshitz-Gilbert equation* (LLG) determines the dynamics of $\boldsymbol{m}$ [20]

$$\frac{\partial \boldsymbol{m}}{\partial t} = -|\gamma_\mathrm{e}|\boldsymbol{m} \times \mu_0 \boldsymbol{\mathcal{H}}_\mathrm{eff} + \alpha \boldsymbol{m} \times \frac{\partial \boldsymbol{m}}{\partial t} + \boldsymbol{\tau}. \tag{2.19}$$

Here γ_e is the electron gyromagnetic ratio and $\boldsymbol{\mathcal{H}}_\mathrm{eff}$ is an effective magnetic field, given by the functional derivative of the free energy [26]

$$\boldsymbol{\mathcal{H}} = -\frac{1}{\mu_0 M_\mathrm{s}} \frac{\delta F[\boldsymbol{m}(\boldsymbol{r}, t)]}{\delta \boldsymbol{m}}. \tag{2.20}$$

The first term $-|\gamma_\mathrm{e}|\boldsymbol{m} \times \mu_0 \boldsymbol{\mathcal{H}}$ will cause $\boldsymbol{m}$ to precess around $\boldsymbol{\mathcal{H}}$. The second term in the LLG Eq. (2.19) is called the *Gilbert damping* and is parametrized by the Gilbert damping constant α. This term will seek to minimize the free energy of the system given in Eq. (2.13) by relaxing $\boldsymbol{m}$ towards $\boldsymbol{\mathcal{H}}$. More generally one can state about the first two terms of Eq. (2.19) that $-|\gamma_\mathrm{e}|\boldsymbol{m} \times \mu_0 \boldsymbol{\mathcal{H}}$ determines the eigenmode frequency, while the Gilbert damping term determines the eigenmode lifetime [25]. Fig. 2.2 indicates both the precession term and the Gilbert damping term. The last term $\boldsymbol{\tau}$ is a spin-transfer torque (STT). STT can appear if one were to apply an electric current to a metallic and magnetic material. The spins of the electrons in the applied current could interact with the localized electron spins, causing a change in $\boldsymbol{m}$ [36].

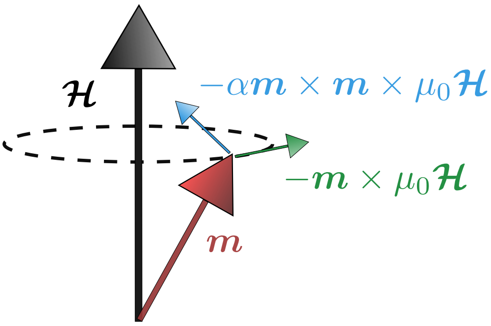


Figure 2.2: Motion of $\boldsymbol{m}$ due to an effective field $\boldsymbol{\mathcal{H}}$ and Gilbert damping. $\boldsymbol{m}$, $-\boldsymbol{m} \times \mu_0\boldsymbol{\mathcal{H}}$ and $-\boldsymbol{m} \times \boldsymbol{m} \times \mu_0\boldsymbol{\mathcal{H}}$ are all perpendicular to each other.

For computational reasons it may be preferable to rewrite the LLG Eq. (2.19) so that the derivative is isolated on one side of the equation. By inserting the expression for $\partial_t\boldsymbol{m}$ from Eq. (2.19) back into the derivative on the right hand side, and applying the condition in Eq. (2.6) one can write the transformed and explicit LLG equation

$$\frac{\partial \boldsymbol{m}}{\partial t} = \frac{-|\gamma_{\mathrm{e}}|}{1+\alpha^2}\left[\boldsymbol{m} \times \mu_0\boldsymbol{\mathcal{H}} + \alpha\boldsymbol{m} \times \boldsymbol{m} \times \mu_0\boldsymbol{\mathcal{H}} + \tilde{\alpha}\boldsymbol{\tau} \times \boldsymbol{m}\right] + \boldsymbol{\tau} \tag{2.21}$$

where $\tilde{\alpha} = \alpha/|\gamma_{\mathrm{e}}|$.

2.1.5 Calculation of the effective field, $\boldsymbol{\mathcal{H}}_{\text{eff}}$

The effective field $\boldsymbol{\mathcal{H}}$ remains to be calculated in Eq. (2.20). The functional derivative of the right hand side can be written [37]

$$\frac{\delta F[\boldsymbol{m}(\boldsymbol{r},t)]}{\delta \boldsymbol{m}} = \frac{\partial f(\boldsymbol{m}, \nabla\boldsymbol{m}, \boldsymbol{r}, t)}{\partial \boldsymbol{m}} - \nabla \cdot \frac{\partial f(\boldsymbol{m}, \nabla\boldsymbol{m}, \boldsymbol{r}, t)}{\partial(\nabla\boldsymbol{m})} \tag{2.22}$$

By substituting the expression for f in Eq. (2.13) and doing the derivatives one obtains

$$\begin{aligned}\boldsymbol{\mathcal{H}} =& \frac{2}{\mu_0 M_\mathrm{s}}\Big[A\nabla^2\boldsymbol{m} + K_\mathrm{easy}(\boldsymbol{m}\cdot\mathbf{e}_\mathrm{easy})\mathbf{e}_\mathrm{easy} - K_\mathrm{hard}(\boldsymbol{m}\cdot\mathbf{e}_\mathrm{hard})\mathbf{e}_\mathrm{hard} \\ & - D\nabla\times\boldsymbol{m}\Big] - \mu_0\boldsymbol{h}.\end{aligned} \tag{2.23}$$

Together with the LLG Eq. (2.19) one can now solve the motion of $\boldsymbol{m}$. For AFMs the additional fields of Eqs. (2.14) and (2.15) is

$$\boldsymbol{\mathcal{H}}_{\text{eff, i}} = -\frac{4A_{\text{h, ex}}}{\mu_0 M_\mathrm{s}}\boldsymbol{m}_j, \quad \boldsymbol{\mathcal{H}}_{\text{eff, i}} = \frac{\eta_i}{\mu_0 M_\mathrm{s}}\boldsymbol{D}_\mathrm{h}\times\boldsymbol{m}_j. \tag{2.24}$$

2.1.6 Temperature in the LLG equation

The LLG Eq. (2.19) can be expanded to include the effect of temperature in the magnetic system. This is achieved by adding a new field, $\boldsymbol{\mathcal{H}}_\mathrm{th}$ to the already existing field $\boldsymbol{\mathcal{H}}$. The thermal field is added phenomenological to account for random fluctuations caused by the finite temperature. Due to the stochastic fluctuations of $\boldsymbol{\mathcal{H}}$ the new and updated equation is called *stochastic-Landau-Lifshitz-Gilbert equation* (sLLG) and reads [38]

$$\frac{\partial\boldsymbol{m}}{\partial t} = -|\gamma_\mathrm{e}|\boldsymbol{m}\times\mu_0(\boldsymbol{\mathcal{H}}+\boldsymbol{\mathcal{H}}_\mathrm{th}) + \alpha\boldsymbol{m}\times\frac{\partial\boldsymbol{m}}{\partial t} + \boldsymbol{\tau}. \tag{2.25}$$

$\boldsymbol{\mathcal{H}}_\mathrm{th}$ will have the following restrictions

$$\langle\boldsymbol{\mathcal{H}}_\mathrm{th}(\boldsymbol{r},t)\boldsymbol{\mathcal{H}}_\mathrm{th}(\boldsymbol{r},t')\rangle = \frac{2\alpha k_B T}{|\gamma|\mu_0 M_\mathrm{s}}\delta(\boldsymbol{r}-\boldsymbol{r'})\delta(t-t') \tag{2.26}$$

and

$$\langle\boldsymbol{\mathcal{H}}_\mathrm{th}(\boldsymbol{r},t)\rangle = 0. \tag{2.27}$$

Here k_B is the Boltzmann constant and T is the temperature of the system. The Dirac delta function $\delta(x)$ is defined by [39]

$$\delta(x) = 0 \text{ for } x\neq 0 \quad \text{and} \quad \int_{-\infty}^{\infty}\delta(x)\,\mathrm{d}x = 1. \tag{2.28}$$

The Gaussian noise caused by temperature is in competition with the magnetic ordering spawned by $\boldsymbol{\mathcal{H}}$ [28]. In a numerical simulation where both time and space is discretized, the thermal field could be drawn from a Gaussian distribution

$$\boldsymbol{\mathcal{H}} \sim \mathcal{N}\left(0, \sqrt{\frac{2\alpha k_B T}{|\gamma|\mu_0 M_\mathrm{s} V\Delta t}}\right), \tag{2.29}$$

where V is and Δt is the discretization volume and time [26]. Replacing the Dirac delta functions with $\frac{1}{\Delta V \Delta t}$ Eq. 2.29 will fulfil the restrictions above.

Given a large enough temperature in a FM called the *Curie temperature*, T_C, the system will lose its magnetic ordering and become dominated by the unsystematic nature of $\boldsymbol{\mathcal{H}}_{\text{th}}$ [40]. This principle also applies to AFMs, where the critical temperature is called the *Néel temperature*, T_N. Increasing temperature above these critical temperatures results in a phase transition in the magnetic material.

2.2 Voltage-controlled magnetic anisotropy

In Eq. (2.9) the anisotropy constants are taken as constants in space. However, there exists a method called *voltage-controlled magnetic anisotropy* (VCMA) that allows for a spatially varying anisotropy parameter $K \to K(\boldsymbol{r})$ [30]. VCMA is a relatively newly discovered method of controlling anisotropy in a magnetic material. This is achieved by applying an electrical field to the magnetic sample. In FMs the VCMA is believed to occur when the applied electrical field causes relative changes in the electronic occupation state in $3d$ orbitals. In an insulating material, the electrical field will vanish in the bulk of the material [21], therefore VCMA is only possible in thin materials. In [41] it is proposed that an anisotropy gradient can control a topological magnetic structure called a skyrmion in an AFM. Furthermore, in [42] control over in-plane (IP) and out-of-plane (OOP) magnetization using VCMA is observed. In this thesis, we will assume that we can control the anisotropy coefficients by applying an electrical field to the AFM, and the strength of the anisotropy modulation is dependent on the amplitude of the electrical field. This thesis will employ VCMA to obtain control over a topological structure, similar to the work mentioned in [41].

2.3 Spin waves

A spin wave is a non-uniform excitation of the spin lattice that progresses from one location to another. Unlike uniform precession where the whole lattice precesses with the same phase, spin waves have a position-dependent phase. The study of spin waves is called *magnonics*, as their quanta is named *magnons* [30]. It is a research field of great interest, as spin waves are considered to be a possible replacement for moving electrons as information carriers.

Just as fluid waves, spin waves are transportation of phase and energy, rather than an actual particle. Figure 2.3 shows a spin wave on a one-dimensional chain of spins. Considering a FM and assuming a small deviation $\delta\boldsymbol{m}$, from some fixed ground state $\boldsymbol{m}_0$, and further assuming the excitations to be planer waves one can write

$$\boldsymbol{m}(\boldsymbol{r},t) = \boldsymbol{m}_0 + \delta\boldsymbol{m}, \quad |\delta\boldsymbol{m}| \ll |\boldsymbol{m}| \tag{2.30}$$

where $\boldsymbol{k}$ is the wave vector and ω is the angular frequency of the wave [1].

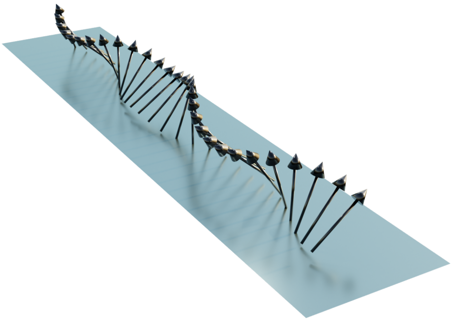


Figure 2.3: Spin wave on a one dimensional chain of spins.

Neglecting quadratic terms in $\delta\boldsymbol{m}$ and doing a temporal Fourier transform one arrives at the linearized LLG equation [43, 1]

$$\mathtt{i}\omega\delta\boldsymbol{m} = -|\gamma_e|\boldsymbol{m} \times \mu_0\boldsymbol{\mathcal{H}}_{\text{eff}} + \mathtt{i}\omega\alpha\boldsymbol{m}_0 \times \delta\boldsymbol{m}, \tag{2.31}$$

$\mathtt{i}$ being the imaginary unit. This equation has spin wave solutions on the form $\delta\boldsymbol{m}e^{\mathtt{i}(\omega(\boldsymbol{k})t+\boldsymbol{k}\cdot\boldsymbol{r})}$ where $\omega(\boldsymbol{k})$ is the dispersion relation. The group velocity $\boldsymbol{v}_{\text{g}}$ and the phase velocity $\boldsymbol{v}_{\text{p}}$ of the spin wave is then given as [1]

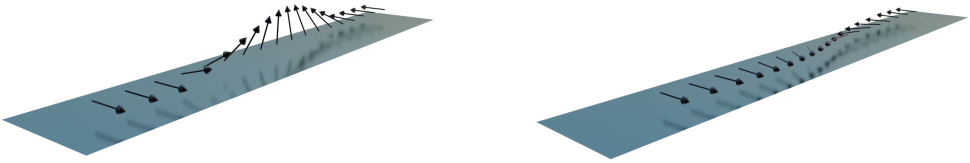
$$\boldsymbol{v}_{\text{g}} = \nabla_{\boldsymbol{k}}\boldsymbol{\omega} \qquad \boldsymbol{v}_{\text{p}} = \frac{\boldsymbol{k}\boldsymbol{\omega}}{|\boldsymbol{k}|^2}. \tag{2.32}$$

Spin waves exist in both FMs and AFMs alike. FM spin waves have typically ω in the GHz range, and $|\boldsymbol{v}_{\text{g}}| \approx 1\,\text{km/s}$. In nano-devices this corresponds to a pico second time scale. There are numerous examples of how spin waves in FM can be utilized in new and modern computing devices [44, 9]. However, AFM spin waves reach THz frequencies, which yields the possibilities for extremely fast and energy-efficient computing.

Spin waves can be excited in several ways. Among others is by VCMA modulation [44], spin-transfer torques, thermal excitation and laser pulses [30]. Thermal excitations suffer from the incoherent nature of temperature, while spin-transfer torques require some kind of applied current which will give rise to Joule heating. VCMA modulation and laser pulse excitation is promising as they can be localized to nanometre scale, is tunable, and does not require applied currents [30, 45]. However, these techniques are still in development and not commercially available on large scales.

2.4 Domain walls

Domain walls (DW) are topological structures that separate two magnetic domains. They are commonly separated into two types; Bloch and Néel [28]. Bloch DWs are characterized by that the rotation of $\boldsymbol{m}$ over the DW happens out of the DW plane. On the contrary, Néel DW rotates in the DW plane. This is illustrated in Fig. 2.4.



(a) Bloch domain wall.

(b) Néel domain wall.

Figure 2.4: Bloch and Néel type domain walls.

Considering a Bloch DW as in fig. 2.4a, the magnetization can we written in spherical coordinates

$$\boldsymbol{m}(\phi(y)) = \sin\phi\,\mathbf{e}_x + \cos\phi\,\mathbf{e}_z \tag{2.33}$$

ϕ is the azimuthal coordinate, or angle from the x-axis, and the wall is taken to be in the x-z plane. A configuration like this can be obtained by a system with exchange and easy axis along the x-axis. The free energy is

$$F = \int \left[A \left(\frac{\partial \boldsymbol{m}}{\partial y} \right)^2 - K_{\text{easy}} \left(\boldsymbol{m} \cdot \mathbf{e}_x \right)^2 \right] \mathrm{d}y \tag{2.34}$$

To find the DW thickness one can find the energy minimum with respect to $\phi(x)$. Inserting for $\boldsymbol{m}$ this reduces to

$$\frac{\delta F}{\delta \phi} = \frac{\delta}{\delta \phi}\left[\int A\left(\frac{\partial \phi}{\partial y}\right)^2 + K_{\text{easy}} \sin^2 \phi - K_{\text{easy}}\right] \mathrm{d}y = 0. \tag{2.35}$$

Using the Euler-Lagrange eq. (2.22), multiplying with $\partial\phi/\partial y$ and exploiting the fact that $\partial(\partial\phi/\partial y)^2/\partial y = 2\partial\phi/\partial y\, \partial^2\phi/\partial y^2$ one arrives at

$$\frac{\mathrm{d}\phi}{\mathrm{d}y} = \sqrt{\frac{K \sin^2 \phi}{A}}. \tag{2.36}$$

Finally making use of $m_x = \cos\phi$ one obtains

$$\frac{\mathrm{d}m_x}{1 - m_x^2} = -\sqrt{K/A}\,\mathrm{d}x \Longrightarrow m_x = -\tanh\left(\sqrt{\frac{K}{A}}y\right). \tag{2.37}$$

The width of the domain wall δ_{DW} thus scales as $\delta_{\text{DW}} \sim \sqrt{A/K}$. A strong anisotropy shortens the width, while a large gradient in $\boldsymbol{m}$ increases the exchange energy. Although this is a simple system, it still provides some intuition into some qualitative properties of DWs.

Domain walls behave similarly in AFMs. It can be noted that the two sides of a DW in an AFM will appear identical, due to the nature of the sublattice model and the strong antiferromagnetic homogenous exchange interaction. One important difference is that DW velocities in AFMs can become much larger than in FMs [27]. FM DWs can reach velocities of ∼ 100 m/s while ∼ 10 000 m/s is possible in some AFMs.

2.5 Spin pumping and the spin hall effects

The final features of spintronics necessary for the implementation of the proposed spiking neuron are spin pumping and the spin hall effects, and here a quick introduction to those phenomenons will follow.

2.5.1 Spin currents and the spin Hall effects

Spin currents are flow of angular momenta, and in Cartesian coordinates it is expressed as a rank two tensor [46], and will in this text be denoted $\overleftrightarrow{\boldsymbol{z}}$. This section will follow the work of Dyakonov in [47]. Phenomenologically the current density $\boldsymbol{j}$ and the components of the spin current $(\overleftrightarrow{\boldsymbol{z}})_{ij} = z_{ij}$ can be written as

$$j_i^{(0)} = \zeta q_e n E_i - D q_e \frac{\partial n}{\partial x_i}, \tag{2.38}$$

$$z_{ij}^{(0)} = -\zeta n E_i P_j - D \frac{\partial P_j}{\partial x_j}, \tag{2.39}$$

where ζ and D are the mobility and diffusion constants, $\boldsymbol{E}$ is the electric field, $\boldsymbol{P}$ is the spin polarization density[3] and n is the electron concentration. The z_{ij} component gives the spin flowing in direction $\mathbf{e}_i$ with polarized along $\mathbf{e}_j$.

In the presence of spin-orbit coupling the two currents couples together and for a isotropic material with inversion symmetry the total currents becomes

$$j_i = j_i^{(0)} + q_e \beta \epsilon_{ijk} z_{jk}^{(0)}, \quad z_{ij} = z_{jk}^{(0)} - \beta \epsilon_{ijk} j_k^{(0)}, \tag{2.40}$$

where $\epsilon_{ijk} \equiv (\overleftrightarrow{\boldsymbol{\epsilon}})_{ijk}$ is the *Levi-Cevita tensor*, also known as the unit antisymmetric tensor, and β parametrizes the spin-orbit coupling strength, and the Einstein summation convention is applied.

Eq. 2.40 can be written explicitly as [47]

$$\frac{\boldsymbol{j}}{q_e} = \zeta n \boldsymbol{E} + D \nabla n + \beta \zeta \boldsymbol{E} \times \boldsymbol{P} + \beta D \nabla \times \boldsymbol{P}, \tag{2.41}$$

and

$$z_{ij} = -\zeta E_i P_j - D \frac{\partial P_j}{\partial x_i} + \epsilon_{ijk} \beta (\zeta n E_k + D \frac{\partial n}{\partial x_k}). \tag{2.42}$$

Additionally there is a continuity restriction for $\boldsymbol{P}$

$$\frac{\partial P_j}{\partial t} + \frac{\partial z_{ij}}{\partial x_i} + \frac{P_j}{\tau_s} = 0, \tag{2.43}$$

where τ_s is the spin relaxation time. The two first terms in Eq. 2.41 is already accounted for in Eq. 2.38. The third term $\beta \zeta \boldsymbol{E} \times \boldsymbol{P}$ is called the *anomalous Hall effect*, and corresponds to a resistivity dependent on spin polarization. The last term $\beta D \nabla \times \boldsymbol{P}$ indicates that a non-uniform spin polarization density can induce an electric current. This is called the *inverse spin Hall effect* (iSHE). The third term in Eq. 2.42 $\epsilon_{ijk} \beta \zeta n E_k$ describes the *spin Hall effect* (SHE). This effect allows for the possibility that an electrical

---

[3] Here we follow the convention used in [47] and use $\boldsymbol{P}$ instead of the normal spin density $\boldsymbol{S} = \boldsymbol{P}/2$, sometimes also called the spin accumulation [48].

field can cause a spin current. The last term is a diffusive counterpart to the SHE and indicates that a diffusive electron gradient can also induce a spin current.

Lets consider applying an electrical field $\boldsymbol{E} = E_0\mathbf{e}_x$ to a rectangular magnetic material situated at the origin with dimensions L_x, L_y and L_z. We wish to investigate the spin polarization at the $y = 0$ boundary. Following the procedure in [47] and working to the first order in $\boldsymbol{E}$ and using Eqs. 2.42 and 2.43 while assuming stationary conditions, we get

$$D\frac{\partial^2 P_j}{\partial y^2} = -\frac{P_j}{\tau_s}. \tag{2.44}$$

At the y boundary of the material, we must have a zero spin current in the y-direction $q_{yj} = 0$. Using this we get from Eq. 2.42

$$\left.\frac{\partial P_x}{\partial y}\right|_{y=0} = 0, \qquad \left.\frac{\partial P_y}{\partial y}\right|_{y=0} = 0, \qquad \left.\frac{\partial P_z}{\partial y}\right|_{y=0} = \frac{\beta\zeta nE}{D}, \tag{2.45}$$

due to the antisymmetry of $\overleftrightarrow{\boldsymbol{\epsilon}}$ and the fact that the applied field only as an x-component. Solving Eq. 2.44 we get

$$P_z(y) = -\frac{\beta\zeta nEL_s}{D}e^{-y/L_s}, \quad P_x = P_y = 0, \tag{2.46}$$

where $L_s \equiv \sqrt{D\tau_s}$ is some characteristic spin diffusion length. A similar approach at the $y = L_y$ boundary would yield the same result just shifted $y \to y - L_y$. An important note to be made is that an applied electrical field in the x-direction yields a non-zero spin polarization density at the y-boundaries with polarization in the z-direction. In general $\boldsymbol{E}$ (or $\boldsymbol{j}$), the polarization of z_{ij} and the flow direction of z_{ij} are all perpendicular to each other.

There exists another common way write the SHE and iSHE terms. We introduce the explicit terms [48]

$$\boldsymbol{j}^{\mathrm{iSHE}} = \theta_{\mathrm{SHA}} D\frac{q_e}{2\mu_{\mathrm{B}}}\nabla \times \boldsymbol{P} \tag{2.47}$$

and

$$\overleftrightarrow{\boldsymbol{z}}^{\mathrm{SHE}} = \theta_{\mathrm{SHA}}\frac{\mu_{\mathrm{B}}}{q_e}\overleftrightarrow{\boldsymbol{\epsilon}}\overleftrightarrow{\boldsymbol{\sigma}}\boldsymbol{E}. \tag{2.48}$$

Here $\theta_{\mathrm{SHA}} \equiv 2\mu_B/\beta$ is the spin hall angle that parametrizes the strength of the SHE, and $\overleftrightarrow{\boldsymbol{\sigma}}$ is the electrical conductivity tensor of the material.

2.5.2 Spin pumping

Attaching a heavy metal (HM) to a magnet will create a strong spin-orbit coupling between the two materials. A dynamical magnetization generates a spin current that will be absorbed by the HM, a phenomenon known as spin pumping. A changing magnetization will induce a spin current into the HM [48]. The spin pumping effect can be utilized together with the iSHE to detect the position of a DW [49]. Spin pumping is the reciprocal effect of spin-transfer torques. For FMs and AFMs the spin current can be written

$$\begin{aligned} \boldsymbol{j}_{\mathrm{S}}^{\mathrm{FM}} =& \frac{\mu_{\mathrm{B}}\hbar}{q_e^2}\left[\mathrm{Re}\{G^{\uparrow\downarrow}\}\boldsymbol{m}\times\frac{\partial\boldsymbol{m}}{\partial t}+\mathrm{Im}\{G^{\uparrow\downarrow}\}\frac{\partial\boldsymbol{m}}{\partial t}\right] \\ \boldsymbol{j}_{\mathrm{S}}^{\mathrm{AFM}} =& \frac{\mu_{\mathrm{B}}\hbar}{q_e^2}\left[\mathrm{Re}\{G^{\uparrow\downarrow}\}\left(\boldsymbol{l}\times\frac{\partial\boldsymbol{l}}{\partial t}+\tilde{\boldsymbol{m}}\times\frac{\partial\tilde{\boldsymbol{m}}}{\partial t}\right)+\mathrm{Im}\{G^{\uparrow\downarrow}\}\frac{\partial\tilde{\boldsymbol{m}}}{\partial t}\right], \end{aligned} \tag{2.49}$$

where $\boldsymbol{l}$ and $\tilde{\boldsymbol{m}}$ is defined in Eq. (2.17) and μ_{B} is the Bohr magneton, $\mu_{\mathrm{B}} \equiv q_e\hbar/2m_e$, and $\hbar$ being the reduced Planck constant [48, 50]. Note that the vector components of the above equations gives the direction of polarization and not flow. The direction of flow is given by the geometry of the setup, i.e. where one places the HM on the magnet.

Inserting for $\boldsymbol{l}$ and $\tilde{\boldsymbol{m}}$, the cross-terms $\boldsymbol{m}_i \times \partial_t \boldsymbol{m}_j$ cancels and the spin pumping current for AFM reduces to

$$\boldsymbol{j}_{\mathrm{S}}^{\mathrm{AFM}} = \frac{\mu_{\mathrm{B}}\hbar}{q_e^2}\left[\mathrm{Re}\{G^{\uparrow\downarrow}\}\boldsymbol{\xi}+\mathrm{Im}\{G^{\uparrow\downarrow}\}\left(\frac{\partial\boldsymbol{m}_{\mathrm{A}}}{\partial t}+\frac{\partial\boldsymbol{m}_{\mathrm{B}}}{\partial t}\right)\right], \tag{2.50}$$

where the vector $\boldsymbol{\xi}$ is defined as

$$\boldsymbol{\xi} \equiv \boldsymbol{m}_{\mathrm{A}}\times\frac{\partial\boldsymbol{m}_{\mathrm{A}}}{\partial t}+\boldsymbol{m}_{\mathrm{B}}\times\frac{\partial\boldsymbol{m}_{\mathrm{B}}}{\partial t}. \tag{2.51}$$

The idea we are going to employ is that when a DW moves through a magnet, it naturally causes a change in $\boldsymbol{m}$ or $\boldsymbol{l}$. This will lead to spin pumping in the presence of an attached HM. If the polarization of the spin current is perpendicular to the direction of the spin current, a current will be induced in the HM, and this could be readily detected.

2.6 Summary - Introduction to spintronics

This chapter has provided an introduction to some features of spintronics that will be utilized in the AFM spiking neuron. We have studied FMs and AFMs, and seen that one can model the AFM as two superimposed FMs. Furthermore, the topological phenomenon of DWs and spin waves have been introduced. The concepts of VCMA and the spin Hall effects yield opportunities for interaction with magnetic materials from the outside world. The foundation for understanding the physics of the proposed spiking neuron has been given. The next chapter will provide the context in which we want the neuron to operate, and answer why we would want a spiking AFM neuron.

3 Introduction to spiking neural networks

Before introducing the proposed spiking neuron, an introduction to spiking neural networks will follow. This is to provide insight into what function the artificial neuron will serve in a computational setting. The chapter will introduce the biological neuron, and mathematical models describing it. This chapter features an introduction to ensembles of such neurons called artificial neural networks. A description of the general machine learning algorithm will be given. Finally spiking neural networks will be described, and the leaky integrate-and-fire model will be explained. Together with chapter 2 this provides the foundation to understand how the proposed spiking neuron functions, and how its behavior could be utilized in a physical implementation of a neural network.

3.1 The biological neuron and its first mathematical models

This section will give investigate the biological neuron found in the brain of animals. Furthermore, we will investigate the first mathematical models of the neuron that were used to build computing units.

3.1.1 The biological neuron

The neuron is an electrically excitable cell type, and essential to the nervous tissue of almost all animals [10]. It is a fundamental processor of information and interconnects with other neurons to create vast networks. In the human brain, there are several billion neurons [51]. The neuron consist of four main parts: the *soma*, *dendrites*, *the axon* and *the synapse*. Figure 3.1

shows a schematic picture of the components. The soma, or cell body, takes care of the main biological tasks necessary for a cell to live. The neurons communicate with each other through electrical signals, also called spikes, which are transmitted by the axon. The axon splits into several branches to reach neurons at different destinations. The dendrites connect to the axons, collecting the signals. In the interconnection between dendrite and axon lies the synapse. The synapse interprets the input from the axon and sends an output to the dendrites depending on some function that is adaptable for each synapse. A presynaptic neuron sends signals via its synapses to its postsynaptic neurons. The state of a neuron is determined by the potential of its soma. This is a function of signals from presynaptic neurons. If the potential reaches a threshold value, a spike is transmitted from the neuron to its corresponding postsynaptic neurons.

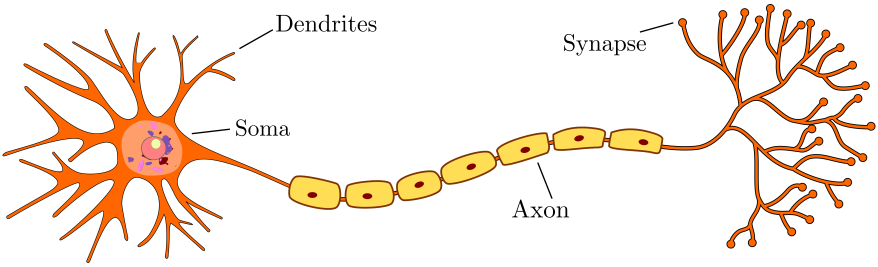


Figure 3.1: Schematic of a single neuron. Figure from [52], adapted to add labels.

3.1.2 McCulloch-Pitts neuron model

The first mathematical proposal of a neuron was given by McCulloch and Pitts in 1943 [53]. The model for the neuron is now known as *McCulloch-Pitts* (MP) neuron. Fig. 3.2 shows the setup of the neuron. Their model consist of a set of binary input variables $\boldsymbol{x} = \{x_j\}$ and a single binary output y_i. The function $g(\boldsymbol{x})$ is a simple sum of the inputs

$$g(\{x_j\}) = \sum_j x_j = G_i. \tag{3.1}$$

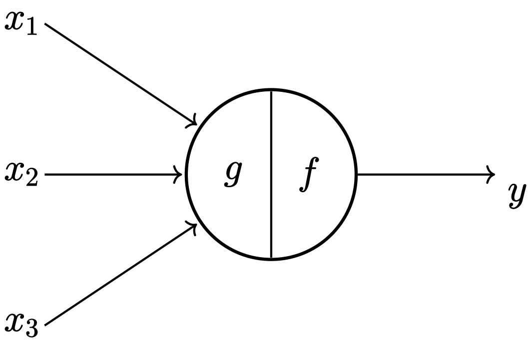


Figure 3.2: McCulloch-Pitts neuron. The inputs $\{x_j\}$ are binary, and so is the output y.

The function $f(G)$ then compares G to some threshold value T, and gives a binary output

$$y_i = f(G_i) = \begin{cases} 1, & \text{for } f(G_i) \geq T_i \\ 0, & \text{for } f(G_i) < T_i. \end{cases} \tag{3.2}$$

McCulloch and Pitts showed in their paper that one could build logical gates such as AND and OR with this simple neuron model.

3.1.3 Rosenblatts neuron and Hebbian learning

In 1958, Rosenblatt expanded upon the MP neuron, by introducing tunable weights w_{ij} into the model [54]. The gate value G is now given by

$$G_i = g(\{x_j\}; w_{ij}) = \sum_j w_{ij} x_j. \tag{3.3}$$

The function $f(G)$ is still the same as in Eq. 3.2. The Rosenblatt neuron is able to learn a mathematical function by changing the weights w_{ij}. The weights can be updated according to Hebb's learning rule [10]

$$w_{ij} \to w_{ij} + \lambda y_i x_j \tag{3.4}$$

where y_i and λ is some learning parameter. However, even with adaptable weights, the Rosenblatt neuron can not learn the XOR logical function [7]. Learning the functions that can not be linearly separated requires a non-linear activation function f.

3.2 Artificial neural networks

This section will provide an introduction to artificial neural networks (ANNs) in the context of computer science and machine learning. Parallels will be drawn between the machine learning algorithms and the previously discussed neuron model. Artificial neural networks employ a different kind of neuron from the Rosenblatt neuron. The new type of neuron allows for a non-binary continuous output and has a non-linear activation function.

3.2.1 The general machine learning algorithm

Any machine learning algorithm is composed of four main components: a dataset, a *cost function*, an optimization method and a model. The following provides a general introduction to a supervised machine learning algorithm based on [7].

Consider a conditional probability distribution $P_{\text{data}}(\mathbf{y} \mid \mathbf{x})$. Our dataset consists of observations of inputs $\boldsymbol{x}$ and outputs $\boldsymbol{y}$ of the random variables $\mathbf{x}$ and $\mathbf{y}$ governed by P_{data}. The model is set of functions $P_{\text{model}}(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta})$. Here $\boldsymbol{\theta}$ is a set of parameters that shapes the probability distribution. The goal of machine learning is to learn the true data generating distribution P_{data}. This is achieved by first guessing what kind of set of functions P_{model} should be, and then tweaking the parameters $\boldsymbol{\theta}$ so that it mimics P_{data}.

Let the set of observations be denoted $\{\boldsymbol{x}^{(i)}\}$ for inputs and $\{\boldsymbol{y}^{(i)}\}$ for outputs. Choosing $\boldsymbol{\theta}$ is commonly done through a maximum likelihood principle. This implies choosing the parameters such that the probability of observing the output given the input is maximized. It can be written

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} P_{\text{model}}(\{\boldsymbol{y}^{(i)}\} \mid \{\boldsymbol{x}^{(i)}\}; \boldsymbol{\theta}), \tag{3.5}$$

where the $\arg\max_x$ of the function $f : X \to Y$ is defined as

$$\arg\max_{x} f(x) \equiv \{x : f(c) \leq f(x) \quad \forall\, c \in X\}. \tag{3.6}$$

Assuming that we have m observations in the dataset that are independent and identically distributed allows us to write (3.5) as

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} P_{\text{model}}(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}). \tag{3.7}$$

This maximization problem does not change if one takes the logarithm and divides by the factor m. Doing this makes the problem a bit more convenient, as it can be expressed in terms of minimizing the cross entropy between the empirical probability distribution, $\hat{P}_{\text{data}}$, obtained from the dataset

$$\begin{aligned}\boldsymbol{\theta}_{ML} &= \arg\max_{\boldsymbol{\theta}} \frac{1}{m}\sum_{i=1}^{m} \ln P_{\text{model}}(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{y},\mathbf{x}\sim\hat{P}_{\text{data}}} \ln P_{\text{model}}(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta}) \\ &= \arg\min_{\boldsymbol{\theta}} \mathcal{S}(\hat{P}_{\text{data}}, P_{\text{model}}).\end{aligned} \tag{3.8}$$

The notation $\mathbb{E}_{\mathrm{x}\sim P}[f(x)]$ can be read as the expectation value of $f(x)$ with respect to $P(x)$. The cross entropy between two probability distributions $P(x)$ and $Q(x)$ is defined to be $\mathcal{S}(P, Q) \equiv -\mathbb{E}_{\mathrm{x}\sim P} \ln Q(x)$.

The cost function, typically denoted $J(\boldsymbol{\theta})$, could simply be the cross entropy, $J(\boldsymbol{\theta}) = \mathcal{S}(\hat{P}_{\text{data}}, P_{\text{model}})$. However, it is possible and common to add terms to this, for instance regularization terms that decrease the variance of P_{model}.

3.2.2 Feedforward neural networks

When learning non-linear functions it is difficult to guess what family of functions P_{model} should be. The *feedforward neural network* (FNN) provides a solution to this problem [7]. The feedforward neural network generally consists of 3 types of layers of neurons, the input layer, hidden layers and output layers. Connections are drawn between each neuron in each layer. Figure 3.3 provides an overview of the network layout. When describing the FNN it is preferable to adopt a vector notation by letting each layer l be described as $\boldsymbol{x}^l = \{x_i^l\}$, where $\{x_i^l\}$ is the set of all neurons in layer l. Doing this for hidden and output layers makes equations more compact. The machine learning as described in the previous subsection takes place by feeding the input data $\boldsymbol{x}$ into the input layer. The information flow from input to output is illustrated by Fig. 3.3.

The neurons in the first hidden layer is calculated from the input by a weighted sum of all neurons from the input layer in addition to some input independent bias, and then transformed by a non-linear transformation called the activation function f^1

$$\boldsymbol{h}^1 = f^1(\boldsymbol{W}^1\boldsymbol{x} + \boldsymbol{b}^1), \tag{3.9}$$

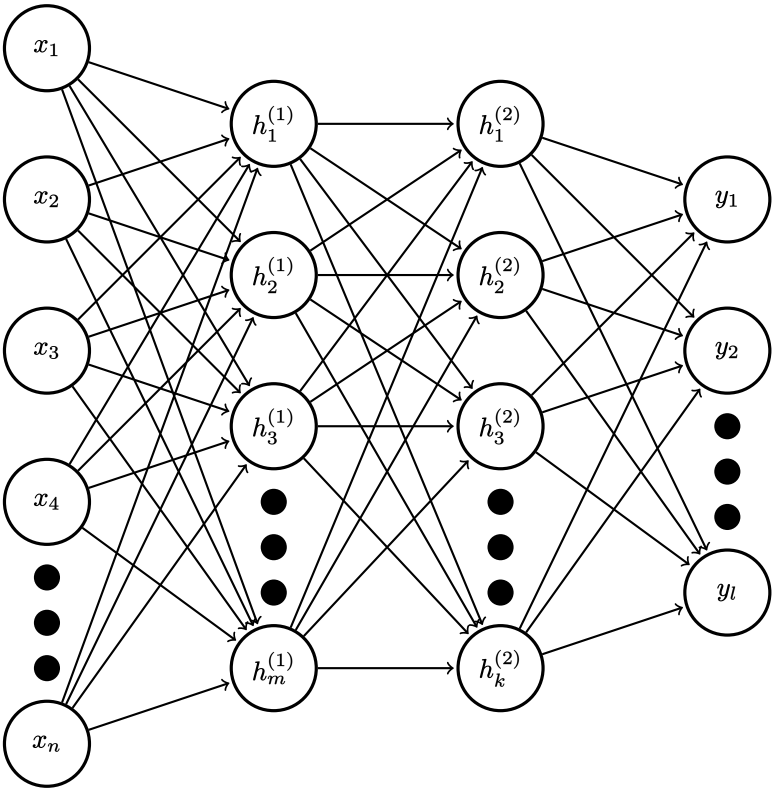


Figure 3.3: Graph structure of neural network with two hidden layers, and arbitrary depth in each layer. Input layer is $\{x_i\}$ and output layer is $\{y_i\}$.

where $\boldsymbol{b}^1$ is the biases, and $\boldsymbol{W}^1$ is the weight matrix connecting $\boldsymbol{x}$ to $\boldsymbol{h}^1$. The next layer of neurons, $\boldsymbol{h}^2$ is then computed based on the results of Eq. (3.9), in a similar way. This procedure is reiterated all the way through to the output layer

$$\boldsymbol{y} = f^3(\boldsymbol{W}^3\boldsymbol{h}^2 + \boldsymbol{b}^3). \tag{3.10}$$

A typical example of an activation function is the *sigmoid function*

$$\sigma(x) = \frac{1}{1 + \exp(-x)}. \tag{3.11}$$

Common for activation functions is that they map the output to a given interval, setting a limit on how large, either negative or positive, the state of a neuron can be. The sigmoid function maps all real numbers to an interval between 0 and 1. To have such non-linear activation functions is essential for

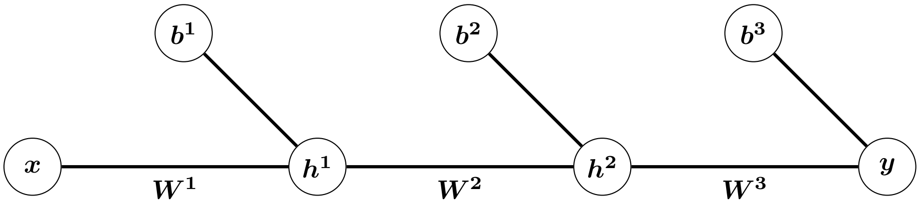


Figure 3.4: Neural network with two hidden layers, and arbitrary depth in each layer.

machine learning with artificial neural networks. Another activation function that was commonly used before is $\tanh(x)$, which was also used to describe domain walls in section 2.4. The two functions are depicted in Fig. 3.5. It can be shown that a single neuron with a non-linear activation function can represent the XOR function, which the Rosenblatt neuron was unable of [7].

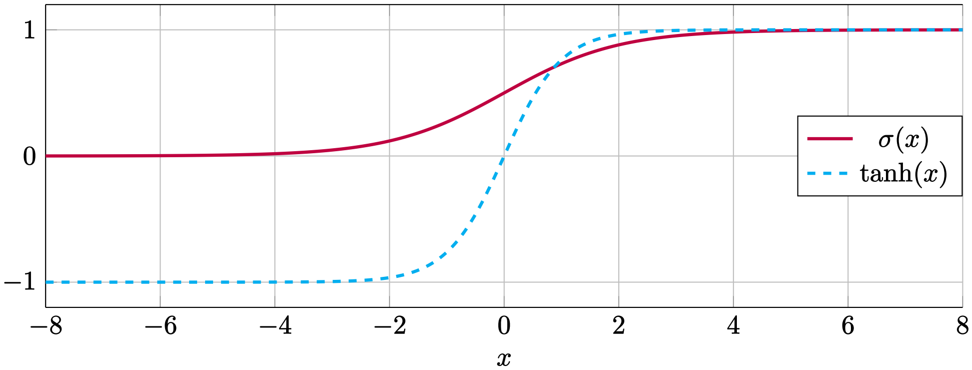


Figure 3.5: Two non-linear activation functions. The sigmoid function $\sigma(x)$ and hyperbolic tangent function $\tanh(x)$.

The machine learning as described in the previous section takes place by initiating the network with randomly selected weights and biases. The network architecture consists of the number of hidden layers and their depth and is together with activation functions chosen before the learning procedure starts. Then $\boldsymbol{y}_{\text{model}}$ is calculated based on the input $\boldsymbol{x}_{\text{data}}$. Then the cross entropy between $\boldsymbol{y}_{\text{model}}$ and the empirically observed $\boldsymbol{y}_{\text{data}}$ is then minimized by tweaking the parameters $\boldsymbol{\theta}$ which in this case consists of both $\boldsymbol{W}$ and $\boldsymbol{b}$. The selection of activation functions and network architecture might seem like choosing the family of functions P_{model} which was the problem we were trying to avoid. However, this is not the case as stated in the *universal approximation theorem* [55, 56]. The theorems interpretation is that a FNN with at least one hidden layer with enough units and an activation function like the sigmoid function, can represent any Borel measurable function [7] from one finite dimensional space to another. A definition of a Borel measurable function is given in [55], but it suffices to say that any continuous function on a closed and bounded subset of $\mathbb{R}^n$ is Borel measurable, where $\mathbb{R}^n$ is the n dimensional set of real numbers. The term represent means in this case approximate with any desired non-zero amount of error. Here within lays the true power of the FNN.

3.3 Spiking neural networks

A spiking neural network (SNN) takes the inspiration from human brain activity into computer science one step further than the previously discussed ANN. SNNs are coined ”the third generation of neural network models”, the first generation referring to networks based on McCulloch-Pitts neurons, and the second generation referring to ANNs with a non-linear activation function [57]. The graph structure and connectivity of SNNs are similar to ANNs. However, information is encoded as spikes, and the network has an explicit time dependency. The system is event-driven as opposed to clock-driven as in ANNs. As mentioned in the introduction, chapter 1, specialized neuromorphic chips implementing SNNs have already been developed.

3.3.1 Mathematical description of the model

The spiking neuron Ξ is represented in Fig. 3.6. Compare this to the MP neuron in Fig. 3.2. A formal definition of SNNs is given in [57]. Let V be a finite set of spiking neurons, connected by a set of $E \subseteq V \times V$ synapses. For each synapse $\langle i, j\rangle \in E$ between presynaptic neuron j and postsynaptic neuron i there are associated a response function ϵ_{ij} and a weight w_{ij}. The state variable $u_i(t)$ of neuron i is then given by

$$u_i(t) = \eta(t - t_i^{\text{prev}}) + \sum_j \sum_f w_{ij}\epsilon_{ij}(t - t_j^{(f)}) + u_0, \tag{3.12}$$

which is a combination of the zeroth order spike response model (SRM$_0$) in [58] and the formulation in [57]. Here u_0 is the equilibrium potential, i.e.

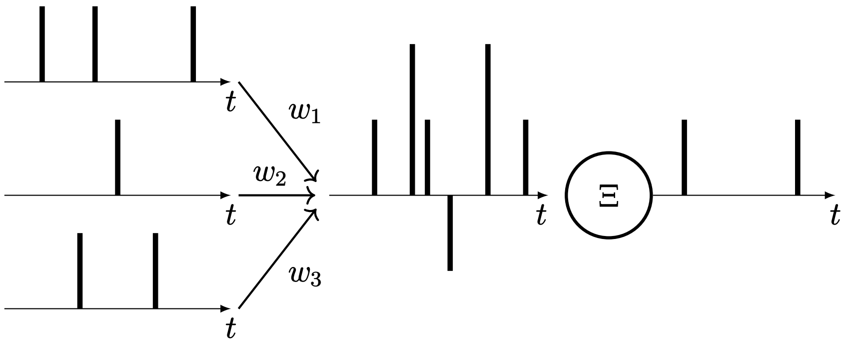


Figure 3.6: Simple spiking neuron. Three spike trains or three synapses lead to a spiking neuron Ξ. The spike trains are multiplied by weights w_i and merged before they get sent into Ξ. A non-linear function determines whether the neuron should fire as a consequence of stimuli from its synapses. Here $w_2 < 0$ to illustrate that synapses might be inhibitory as well as excitatory, while $w_1 = 1$ and $w_3 > 1$. Spikes are drawn as vertical lines, indicating Dirac delta functions $\delta(t - t_j)$.

the value of $u_i(t)$ when no stimuli has affected the neuron. $t_j^{(f)}$ indicates the firing times, where f is the label of each spike. Here t_i^{prev} is the time of the most recent spike from neuron i

$$t_i^{\text{prev}} = \max\{t_i^{(f)} | t_i^{(f)} < t\}. \tag{3.13}$$

The function $\eta(t - t_i^{\text{prev}})$ describes the spike form of $u_i(t)$ and is often written [58]

$$\eta(t - t_i^{\text{prev}}) = \begin{cases} 1/\Delta t & \text{for } \ 0 < t - t_i^{(f)} < \Delta t \\ -\eta_0 \exp(-\frac{t - t_i^{\text{prev}}}{\tau_\eta}) & \text{for } \ \Delta t < t - t_i^{(f)}. \end{cases} \tag{3.14}$$

The parameters η_0 and τ_η determine how a negative spike-afterpotential may appear, which is more related to modelling biological neurons. In the limit $\Delta t \to 0$ we get $\eta(t - t_i^{(f)}) \to \delta(t - t_i^{(f)})$. Thus, $u_i(t)$ is immediately reset to u_0 once a spike is fired. ϵ determines the response for postsynaptic neuron i from stimuli from presynaptic neuron j. Repeated spikes from neuron i gives rise to a spike train $S_i(t)$

$$S_i(t) = \sum_f \delta(t - t_i^{(f)}). \tag{3.15}$$

In general the firing time $t = t_i^{(f)}$ of a neuron i is set when $u_i(t)$ reaches a threshold value $u_{\text{threshold}}$ from below (or above given that $u_0 > u_{\text{threshold}}$)

$$u_i(t) = u_{\text{threshold}} \quad \text{and} \quad \operatorname{sign}(u_i(t) - u_0)\frac{du_i(t)}{dt} > 0 \quad \Longrightarrow t = t_i^{(f)}, \tag{3.16}$$

where $\operatorname{sign}(x) \equiv x/|x|$ is the sign-function. It is worth noting that Eq. 3.12 assumes no time delay as signals travel the synapses. This could easily be added as with a delay time d_{ij} for each synapse [59].

To give the SNN information about the outside world, we introduce a set of input neurons $V_{\text{input}} \in V$. These neurons do not have any presynaptic neurons. Instead, they send spikes as a function of some input data. We will see in section 3.4 how to convert regular data, such as an image, into spikes.

There is a strong resemblance when comparing Eq. 3.12 to Eqs. 3.9 and 3.10. The biggest difference is the time dependence of $\eta(t - t_i^{\text{prev}})$ and $\epsilon_{ij}(t - t_i^{(f)})$, which corresponds to the time dependence of SNNs themselves.

Similar to ANNs, training a SNN means to tune the weights or set of parameters $\boldsymbol{\theta} = \{w_{ij}\}$ in order to minimize some cost function, and thus make some P_{model} resemble a data generating distribution P_{data}.

3.3.2 Leaky integrate-and-fire neuron model

The fact that neuroscientists and computer scientists have two different perspectives when trying to model a neuron again becomes evident when discussing neuron models. Neuroscientists often seek to model a neuron to get insight into the functionality of the brain, while computer scientists rather want to have efficient computational units. The rather general Eq. 3.12 could be used to model a variety of neuron models. The following will introduce a common neuron model, used by computer scientists when creating SNNs.

Leaky integrate-and-fire model

The leaky integrate-and-fire (LIF) model are one of the most prominent neuron models [58]. It can be modelled by a RC circuit as shown in Fig. 3.7. The neuron voltage corresponds to the capacitor voltage $u(t)$. The LIF model is described by the differential equation

$$\tau\frac{du}{dt} = -u(t) + RI(t), \quad \tau = RC, \tag{3.17}$$

τ being the time constant of the capacitor, R is the resistance of the resistor and C is the capacitance of the capacitor. In the sense of neurons, $I(t)$ corresponds to presynaptic stimuli, while the term $-u(t)$ ensures a leaky behavior. Eq. 3.17 can be rewritten as

$$C\frac{du}{dt} = I(t) - \frac{u(t)}{R}, \tag{3.18}$$

where we see that if we short the resistor part of the circuit, i.e. letting $R \rightarrow \infty$ we get a perfect integrate-and-fire circuit with no leaky factor, $Cd_t u = I(t)$. This is a more simple neuron model that is also used in SNNs. The incoming current $I(t)$ is

$$I_i(t) = \sum_j w_{ij} \sum_f \delta(t - t_j^{(f)}). \tag{3.19}$$

The weights w_{ij} determine the connection strength from presynaptic neuron j to postsynaptic neuron i. The sum $\sum_f$ is over all presynaptic spike times (f).

The LIF model can be generalized to a non-linear leaky integrate-and-fire model

$$\tau\frac{du}{dt} = F(u) + G(u)I(t), \tag{3.20}$$

where the functions $F(u)$ and $G(u)$ are arbitrary functions. Here the neuron response to input $I(t)$ is a function of u while in Eq. 3.12 it is a function of the time since last spike.

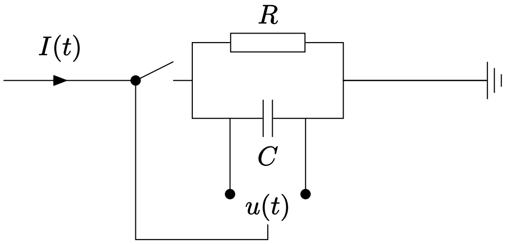


Figure 3.7: Leaky integrate-and-fire circuit. A capacitor, C, and resistor, R, are wired in parallel. The voltage over the capacitor $u(t)$ integrates the current input, while it leaks to ground. When $u(t)$ reaches a threshold value, a switch controlling the input wire is flipped, stopping new currents into the system for a refractory period. During the refractory period charge is completely depleted from the capacitor.

3.4 Neural coding

Information in SNNs is naturally encoded as spike trains, rather than the usual binary information coding schemes used in regular computers. Converting binary information to spikes requires a transformation into time-related domain. The spike train has a natural temporal component, which is inherently different from binary code. The following two subsections will consider two spike encoding schemes, called rate code and temporal code. In neuroscience, the problem of neural coding is not yet solved, but it is assumed that it is probably a mix of the two schemes [58].

3.4.1 Rate code

Rate code encodes information as some average value or frequency of spikes. Consider a black and white image as input.

A problem with rate code is that the temporal averaging suffers the loss of information in the temporal dimension. As previously discussed, the time dependence in SNNs is thought to be one of its strong points. On the other hand, rate code makes the SNNs more similar to ANNs, and can thus be simpler to work with. Another problem is that it is an inefficient way of encoding in a digital computer. Running SNNs on a clock-driven system with discrete time steps will require 2^N time steps to code N bits of information.

The simplest way of rate encoding is just averaging the number of spikes n_s over a predetermined time interval Δt [58].

$$\nu = \frac{n_\mathrm{s}}{\Delta t} \tag{3.21}$$

A stronger signal could correspond to a higher frequency. Rate code works well on slowly varying or constant signals. Essentially rate code corresponds to a Fourier transform of the spike train in Eq. 3.15.

3.4.2 Temporal code

Temporal code, or spike code as it is also known, is the alternative to rate code and is a set of different coding schemes that encode information in the timing between individual spikes. There are several ways to do this, and here we will consider three distinct methods. In temporal coding, neurons need only to fire once for each set of new presynaptic stimuli. The idea behind temporal coding is that the time it takes to reach the threshold value in neurons such as LIF neurons is correlated to the strength of the stimuli [60].

Time-to-first-spike

Time-to-first-spike encodes information as the time between the arrival of stimuli to a spike is sent. If a neuron transmits a signal shortly after stimuli, it can be considered a strong interaction. This requires a reference signal that spikes can be timed after. In theory, this could code for infinitely many bits. However, in a more realistic setup, with time resolution δt and N neurons, one could in a time interval Δt encode $\log_2(\Delta t/\delta t)^N$ bits.

Rank-order-coding

Rank-order-coding is another spike code, utilize the timing between spikes to encode information [60]. Consider a set of $\boldsymbol{x} = \{x_i\}$ neurons. Their output spikes are ordered in time, and the order determines the information. If t_i is the time neuron x_i sends a spike, then the order $t_2 > t_1 > t_0$ encodes something different than $t_2 > t_0 > t_1$. A layer of N neurons can with rank-order-coding transmit $\log_2 N!$ bits of information, as there are $N!$ ways of arranging $\{t_i\}$.

Latency coding

Latency coding is the most efficient temporal coding scheme mentioned here. It codes information as the timing between different spikes. Spikes are orders as in rank-order coding, but instead of just using the sequence of spikes, latency coding employs the time difference between spikes as $\Delta t_{ij} = t_i - t_j$ [61]. Latency coding and the other temporal coding methods trade the robustness of rate coding for superior efficiency.

3.5 Summary - Introduction to spiking neural networks

This chapter has introduced neural networks and neuron models of increasing complexity. The SNN has been studied in detail, together with the LIF neuron. Furthermore, an introduction to the general machine learning algorithm has been given. We can now understand why we would want a physical spiking unit. All necessary theory has now been concluded, and we are ready to study the proposed spiking AFM insulating neuron.

4 Neuron implementation

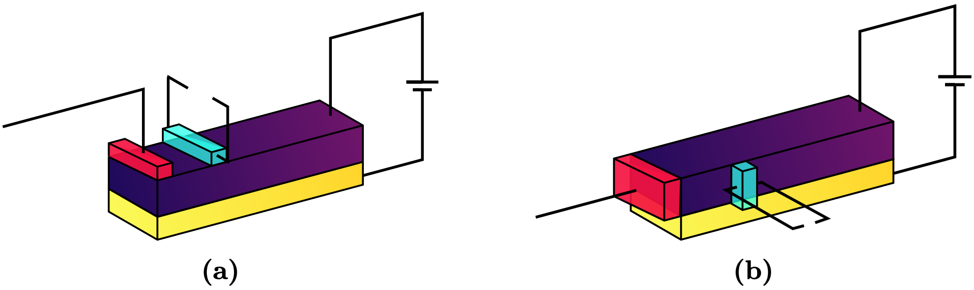
This chapter will introduce the proposed AFM spiking neuron. The AFM spiking neuron will correspond to Ξ in Fig. 3.6. Two different variations of the neuron will be discussed, namely one with OOP magnetization and one with IP magnetization because they exhibit different properties in some areas. The computational method will then be described before some important features will be investigated in detail. Finally, a finite temperature will be introduced to the system, and its effect on the neuron will be discussed.

4.1 Setup and functionality

The AFM neuron differs from the proposed neuron in [62] in two major aspects. The first is that the neuron is an AFM and not an FM. Furthermore, the AFM neuron is an insulating material as opposed to conducting. In [62] STTs from charge currents are used to move the DW. However, the insulating material in the AFM spiking neuron makes this a poor choice as it will cause power dissipation through Joule heating.

4.1.1 Geometry

The AFM material in the neuron is an elongated strip. Fig. 4.1 displays the geometry of the setup. The IP and OOP neuron variations require slightly different geometric setups as displayed in Fig. 4.1. The reason behind this will be explained in detail in sections 4.3 and 4.4. For now, it suffices to say that the excitatory antenna has to be placed parallel to the direction of the Néel vector in the area that is in direct contact with the antenna. Furthermore, it is necessary to place the detector such that it is normal to the Néel vector at the DW, $\boldsymbol{l}(x_{\mathrm{DW}})$. This gives the two different setups.


(a)

(b)

Figure 4.1: AFM neuron setup. Dimensions are not to scale. **(a)** OOP, and **(b)** IP. The injector is colored red, detector in blue, and the VCMA insulator and electrode in yellow. A voltage is applied between the VCMA electrode and the AFM.

4.1.2 Free energy

The free energy of the IP neuron is as given in Eq. (2.16), excluding the homogenous DMI. The IP magnetization is supported by an easy axis in the x-direction and a hard axis in the y-direction. $K_{\text{easy}} = K_x$ and $K_{\text{hard}} = K_y$. The OOP neuron has the same free energy as the IP neuron except for the DMI interaction, which is absent, and $K_{\text{easy}} = K_z$. The different parameters and constants are given in table 4.1. Furthermore, the easy axis has a spatial dependence

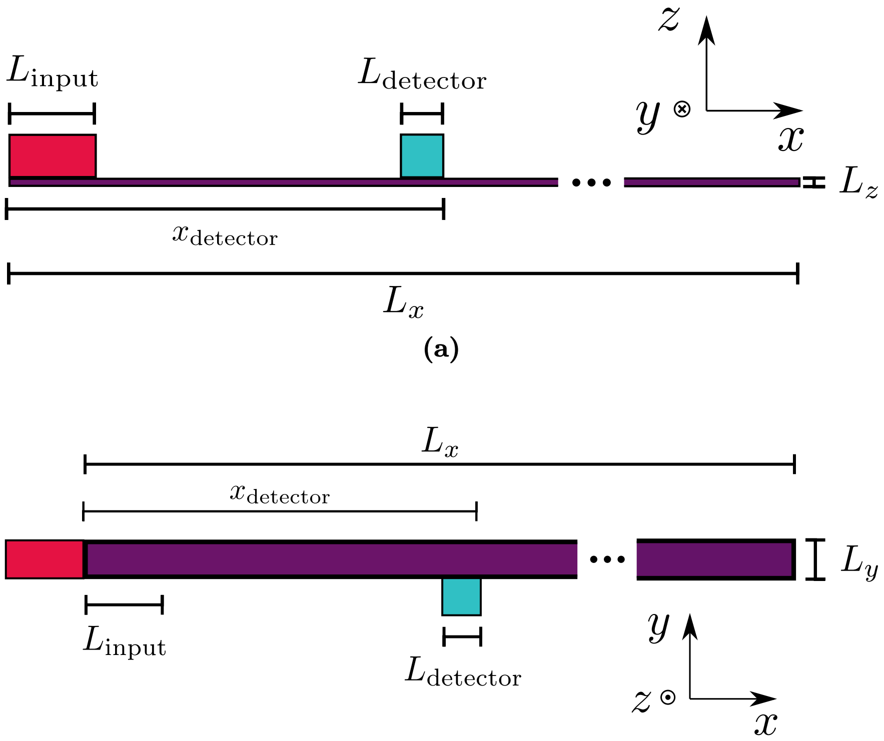
$$K_{\text{easy}}(x) = \left(\frac{x}{L_x} - \frac{2}{3}\right)^2 + 1, \tag{4.1}$$

that is created by the VCMA effect as described in section 2.2. Figure 4.4 depicts the starting configuration of the OOP and IP neuron. The minimum value of $K_{\text{easy}}(x)$ creates an equilibrium position x_0 for x_{DW}

$$\frac{dK_{\text{easy}}(x)}{dx} = 0 \Longrightarrow x_0 = \frac{2}{3} \cdot L_x \sim 333\,\text{nm}. \tag{4.2}$$

The IP neuron has magnetization mainly in the x-direction, with a DW in the z-direction, which is an opposite arrangement from the OOP neuron. The DMI causes the DW to be slightly skewed from the y-direction, as opposed to the OOP DW.

There is one additional parameter that has a spatial variation, and that is the Gilbert damping coefficient α. For $0\,\text{nm} < x < 480\,\text{nm}$ we have $\alpha = 0.002$ as in table 4.1. However, at the far right end of the AFM $x \geq 480\,\text{nm}$, we have $\alpha = 0.1$. This increase in damping is added for two practical reasons.



(a)

(b)

Figure 4.2: To scale AFM neuron setup. **(a)** OOP, and **(b)** IP. L_{input} for the IP indicates that the applied field is set to propagate 20 nm into the material. In other words the applied field for both neurons work in the entire volume of the AFM with $x \leq 20\,\text{nm}$.

The first is to minimize the effect of magnon reflection at the end of the AFM, and the second is to create a boundary such that the DW does not move so far right that it disappears out of the strip, rendering a uniform $\boldsymbol{l}$.

4.1.3 Functionality

The main functionality of the neuron is similar to that in [62]. The position of the DW, x_{DW} encodes the state of the system. By exciting magnons, which in our case is done by applying a circularly polarized external field $\boldsymbol{h}$, one can move the position of the DW. By sending signals similar to that in a SNN one can move the DW towards the end of the strip, where a detector

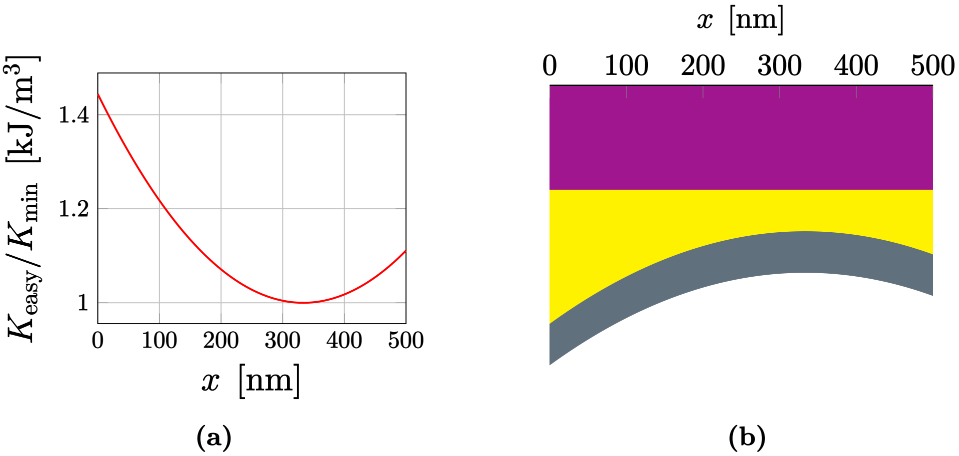
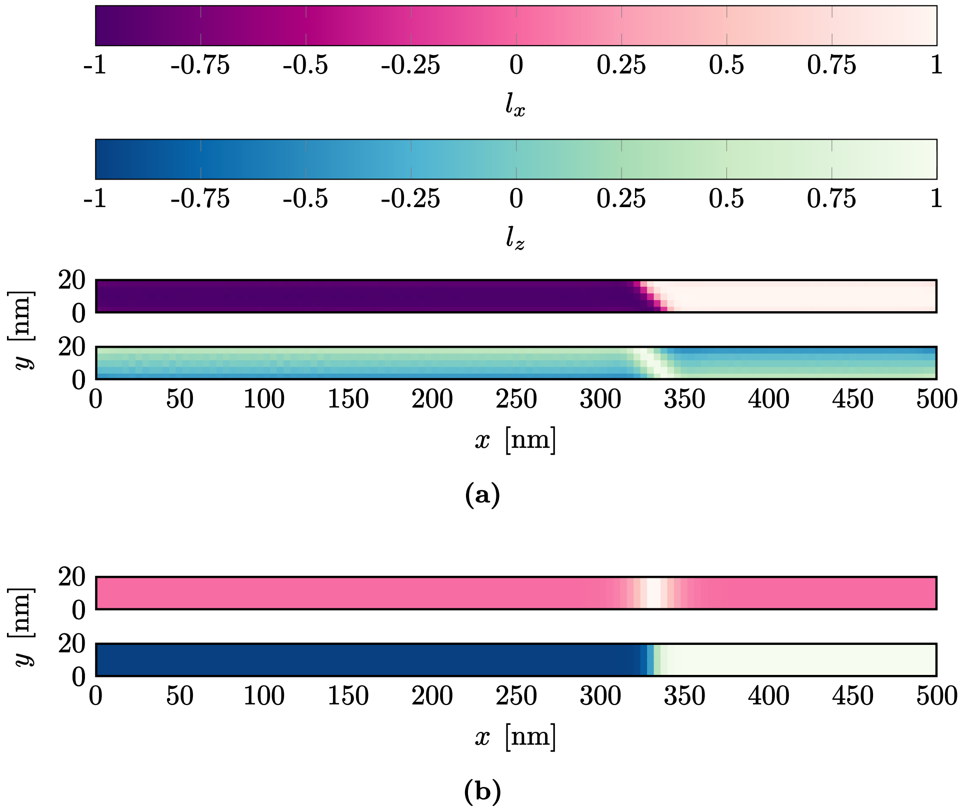


Figure 4.3: VCMA coefficient. **(a)** shows the spatial variation of $K_{\mathrm{easy}}(x)$. **(b)** indicates AFM, insulator and electrode layered from top to bottom, which ensures a spatially varying electrical field, and in turn results in the spatial variation of $K_{\mathrm{easy}}(x)$.

is placed. This corresponds to the integrating part of a LIF neuron which was discussed in subsection 3.3.2. Once enough signals have been sent into the neuron, the DW will eventually have reached the detector at the end. The detector registers the DW and puts the neuron into fire mode. A signal is then sent from the detector of the next postsynaptic neurons that are coupled to this firing neuron. Once the neuron has fired, a switch will break the input line for the time needed to relax the DW back into the equilibrium position. The spatial dependence of K_{easy} will then ensure that the DW moves back into the equilibrium position, x_0. This effect is always present, and not just when the neuron is refractory. This then corresponds to the leaky functionality of a LIF neuron. In neuroscience, the refractory period is the time needed for Na and K pumps to restore the concentration of ions outside and inside the cell [58], while here it is the time it takes for the DW to move from the detector position back to x_0. After the refractory time has passed, the switch controlling the input line is then turned back, allowing new incoming signals to the neuron, starting the process of moving the DW towards the detector once more.



(a)

(b)

Figure 4.4: Equilibrium configuration of DW for **(a)** IP and **(b)** OOP. Both l_x and l_z components are shown.

4.2 Computational method

The LLG equation (or sLLG when a finite temperature is present) is solved using Boris Computational Spintronics [26], which is a rather new and versatile computational spintronics research software. It uses finite differences to solve the equation with the RK4 method. In the simulations, a cell size of $4\,\text{nm} \times 4\,\text{nm} \times 2\,\text{nm}$ yielding $125 \times 5 \times 2$ grid with the given dimensions of the AFM. Spin waves are excited at the injector and modeled by a circularly polarized magnetic field. This results in monochromatic magnons at the left end, emulating an incoming laser beam. The frequency is close to the resonance frequency of the AFMs, and given in table 4.1 together with all other relevant parameters. To simulate the firing event when the DW passes under the detector, the spatial average of the relevant component of $\boldsymbol{\xi}$ is measured under the detector area. We assume this to be proportional to the signal that is measured in experiments that use the inverse spin hall effect and spin pumping.

Table 4.1: Coefficients used in simulations.

Name	Symbol	Value	Unit
Length of AFM	L_x	500	nm
Width of AFM	L_y	20	nm
Height of AFM	L_z	4	nm
Detector position	x_{detector}	100	nm
Detector length	L_{detector}	10	nm
Input length	L_{text}	20	nm
Exchange constant	A	1×10^{-12}	J/m
Homogeneous exchange constant	A_h	-200×10^3	$\mathrm{J/m^3}$
Easy-axis anisotropy constant	K_{easy}	20×10^3	$\mathrm{J/m^3}$
Hard-axis anisotropy constant	K_{hard}	-10×10^3	$\mathrm{J/m^3}$
Saturation magnetization	M_s	2.1×10^3	A/m
Gilbert damping	α	0.002	1
DMI coefficient (only IP)	D	300×10^{-6}	$\mathrm{J/m^2}$
Applied field frequency	ω	6.25×10^{13}	$\mathrm{rad\,s^{-1}}$

4.3 Domain wall movement

A crucial part of the neuron is the possibility to move the DW. As previously mentioned, this is done with exciting magnons with a circularly polarized external field. Fig. 4.5 shows the effect of applying one pulse of RH and one pulse of LH circularly polarized field. Here, the qualitative distinction between the OOP and IP neuron becomes evident. The magnons that are excited at the end of the sample travel down the AFM and work with an effective force on the DW. For the OOP neuron, the force is attractive, i.e. that the DW starts to move towards the place where the magnons are excited. In the case of the IP neuron, changing the handedness of the applied field changes the direction of the DW movement. In other words the sign of its velocity, v_{DW} changes. This is not present in the OOP neuron. This is summarized in table 4.2. Furthermore, the shift in DW position, Δx_{DW}, is larger in the IP case, despite having a weaker field applied. Simulations indicated that without the presence of DMI in the IP neuron, there was only an attractive force on the DW. This complies with the results of [34], where they found that DMI gave the possibility to have both attractive and repulsive movement in a DW with IP magnetization on a one-dimensional chain of spins.

Table 4.2: Whether or not changing handedness of excitatory field changes direction of DW movement.

	DMI	No DMI
IP	Yes	No
OOP	No	No

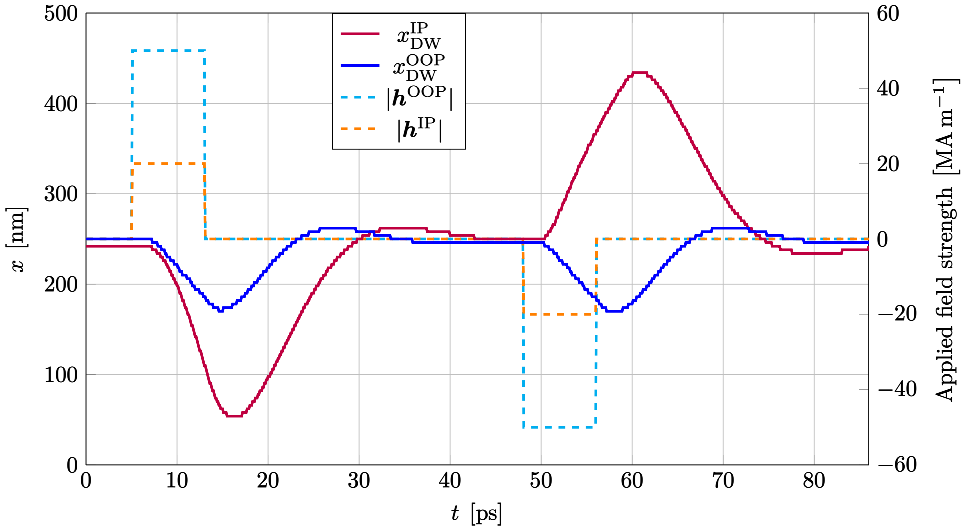


Figure 4.5: DW movement resulting from one right handed and one left handed excitatory signal. The duration of each of spike is 8 ps. The left y-axis refers to position along the x-axis of the AFM, while the right indicates applied field strength and handedness. Note that the sign here indicates whether the signal is LCP, $|\boldsymbol{h}^{\mathrm{OOP}}| > 0$, or RCP ,$|\boldsymbol{h}^{\mathrm{OOP}}| < 0$, and not the sign of $\boldsymbol{h}$ itself.

The figure also indicates a DW velocities $v_{\mathrm{DW}}^{\mathrm{IP}} \sim 20\,\mathrm{km\,s^{-1}}$ and $v_{\mathrm{DW}}^{\mathrm{OOP}} \sim 10\,\mathrm{km\,s^{-1}}$. This concurs with known DW speeds for other AFMs [63]. It is also possible to observe a delay from the applied signal, before the DW starts to move. This is a consequence mainly due to the fact that the magnons have to travel from the excitation area, which is between $[x = 0\,\mathrm{nm}, x = 20\,\mathrm{nm}]$, to the DW which is situated at the equilibrium position $x_0 = 333\,\mathrm{nm}$. This gives a spin wave group velocity of roughly $v_\mathrm{g}^\mathrm{SW} = 150\,\mathrm{km\,s^{-1}}$, around one order of magnitude faster than v_DW.

4.4 Spike readout

This section will provide and explain a method to determine if x_DW passes some threshold, and thus if the neuron should "fire". An explanation as to why the detector is placed differently from the IP to the OOP case will also be given. The focus lies on finding a significant qualitative measurement. We will make the assumption that $\mathrm{Re}\{G^{\uparrow\downarrow}\} \gg \mathrm{Im}\{G^{\uparrow\downarrow}\}$ and that the output signal to be measured is proportional to $\boldsymbol{\xi}$, where $\boldsymbol{\xi}$ is as defined in Eq. 2.51. There has been no attempt to estimate the parameters in Eqs. 2.47 and 2.50.

4.4.1 Spike readout in the OOP neuron

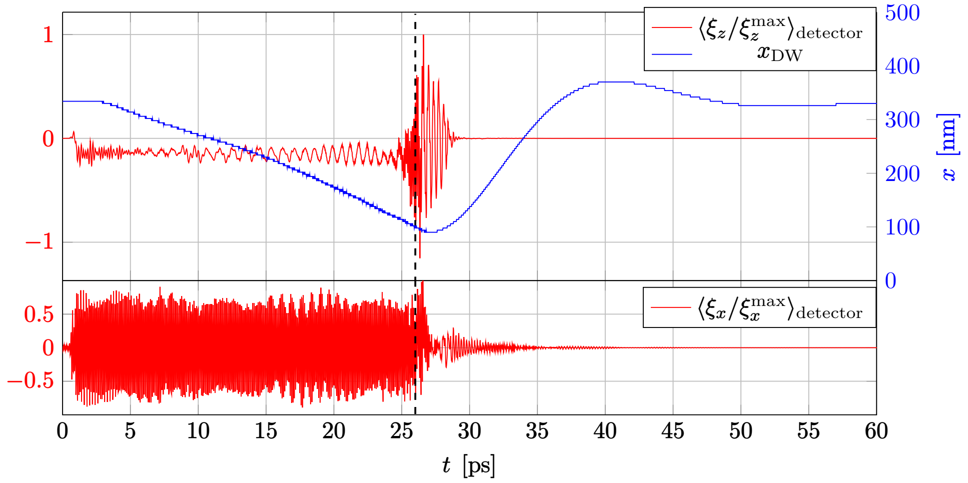


Figure 4.6: Spin pumping effect in OOP neuron. A field with amplitude $H_0 = 5 \times 10^7\,\mathrm{A\,m^{-1}}$ is applied from $t = 0\,\mathrm{ps}$ until $x_\mathrm{DW} = x_\mathrm{detector}$ at $t = 27\,\mathrm{ps}$, and is then turned off. The notation $\langle\cdot\rangle_\mathrm{detector}$ indicates that it is a spatial average under the detector area. The left y-axis indicates the normalized detected signal, while the right indicates the position along the AFM.

To know whether or not the neuron should spike, it is necessary to measure x_{DW}. Here, it is proposed to detect by utilizing the effects of spin pumping and the iSHE, mentioned in section 2.5. The detector consists of a HM that is in direct contact with the AFM. Once the DW passes under the HM, it will induce a large spin pumping effect. This will induce a spin current in the HM, which in turn will create a measurable potential difference in the HM through the iSHE. The spin pumping effect for the OOP neuron is shown in Fig. 4.6. More specifically it shows the spacial average of $\boldsymbol{\xi}$ under the detector area. When the DW reaches the detector position $x_{\mathrm{detector}} = 100\,\mathrm{nm}$ at $t = 27\,\mathrm{ns}$, there is a large increase in the fluctuation amplitude. Furthermore, the spin current into the HM, which will be in the z-direction, will be polarized along $\boldsymbol{l}_{\mathrm{DW}}$. In the OOP case, this is along the x-direction. This will then create a potential difference in the HM across the y-direction, which can be measured to determine that the DW has indeed reached the detector due to the iSHE. A signal can then be sent to the postsynaptic neurons, and the neuron itself can be put into refractory mode. The smaller fluctuations at $t < 27\,\mathrm{ps}$ is a consequence of the applied magnons. At $t > 27\,\mathrm{ps}$ the input signals are turned off, and magnons are no longer excited in the system. The fluctuations quickly dissipate after the DW has moved away from the detector.

4.4.2 Spike readout in the IP neuron

Spike readout in the IP neuron works generally the same way as in the OOP case. However, the fact that $\boldsymbol{l}_{\mathrm{DW}}$ now lies in the z-direction, and not in the x-direction has an impact on the readout, as the geometry of DW and detector placement matters for the outgoing signal. Fig. 4.7 indicates all components of $\boldsymbol{\xi}$. In the OOP case, it is ξ_z that is measured, along the same direction as $\hat{n}_{\mathrm{easy}}$. In the IP case, this would be equivalent to measuring ξ_x. However, this is geometrically unavailable, as the AFM lies in the x-direction. A detector can only be placed on the AFM at $x_{\mathrm{detector}} = 100\,\mathrm{nm}$ in the y or z-direction. Nevertheless, Fig. 4.7 indicates changes in both ξ_y and ξ_z as $x_{\mathrm{DW}} < x_{\mathrm{detector}}$. The most considerable changes happen in ξ_y and that is why the detector is placed on the side of the AFM instead of on top as in the OOP case, as in Fig. 4.1. In the IP neuron x_{DW} overshoots x_{detector}, due to the inertia of the AFM DW. At $t = 19\,\mathrm{ns}$ a new and weaker output signal can be seen in all three components of $\boldsymbol{\xi}$.

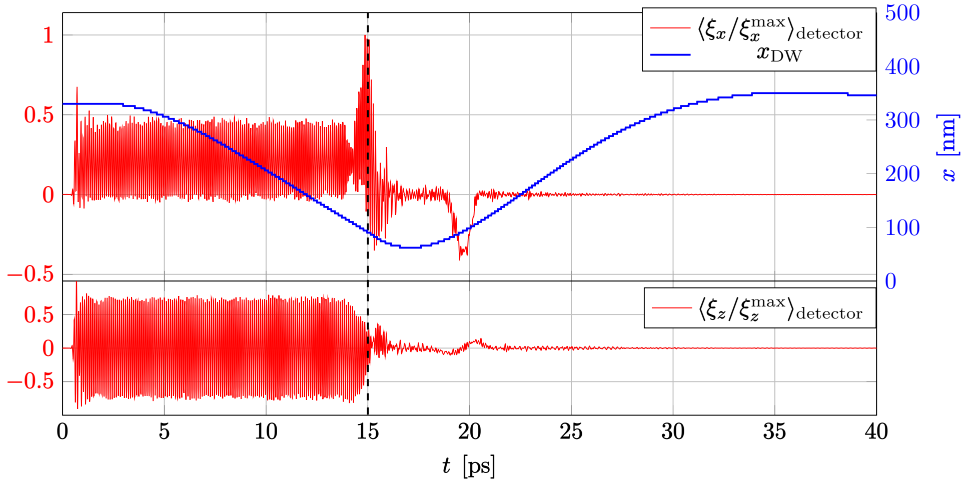
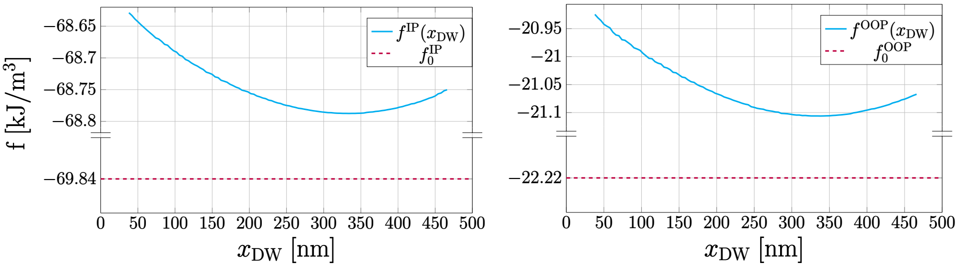


Figure 4.7: Spin pumping effect in IP neuron. A field with amplitude $H_0 = 2 \times 10^7\,\mathrm{A\,m^{-1}}$ is applied from $t = 0\,\mathrm{ps}$ until $x_{\text{DW}} = x_{\text{detector}}$ at $t = 15\,\mathrm{ps}$, and is then turned off. The left y-axis indicates the normalized detected signal, while the right indicates the position along the AFM.

4.5 Energy consumption

As mentioned in chapter 1 one of the advantages of analog and neuromorphic is the possibility of substantially more energy-efficient computing units than what regular CMOS-technology can provide. Chapter 2 makes the case that operating the neuron without electrical currents avoids the problem of Joule heating. This section seeks to make a simple estimate of how much energy it takes to excite a neuron from equilibrium to firing, i.e. how much energy it takes to move $x_{\text{DW}} = x_0$ to $x_{\text{DW}} = x_{\text{detector}}$. We introduce the total free energy of the AFM as $f(x_{\text{DW}}) = f_{\text{A}} + f_{\text{B}}$, see Eq. (2.16). Fig. 4.8 shows $f(x_{\text{DW}})$ with no external excitations, $\boldsymbol{h} = 0$. The function mirrors the spatial dependence of $K_{\text{easy}}(x)$, as seen in Fig. 4.3. Here we have only considered the energy difference from having the DW at different positions along the AFM. Moving the DW requires excitations of spin waves in the material. Fig. 4.9 displays the change in f as the DW is excited from x_0 to x_{detector}. It becomes evident from the figure that exciting spin waves across the whole lattice require more energy than it takes to move the DW. In the first picoseconds, spin waves travel down the magnet from the antenna. Referring to the spin wave velocity in section 4.3, it takes about 3 − 4ps for



(a) IP. **(b)** OOP.

Figure 4.8: The free energy density of the AFM neuron with no external fields, as a function of DW position.

the spin waves to travel the full length of the AFM. This can be seen as the initial rapid increase of $\Delta f(t) = f(t) - f(t = 0)$ in Fig. 4.9.

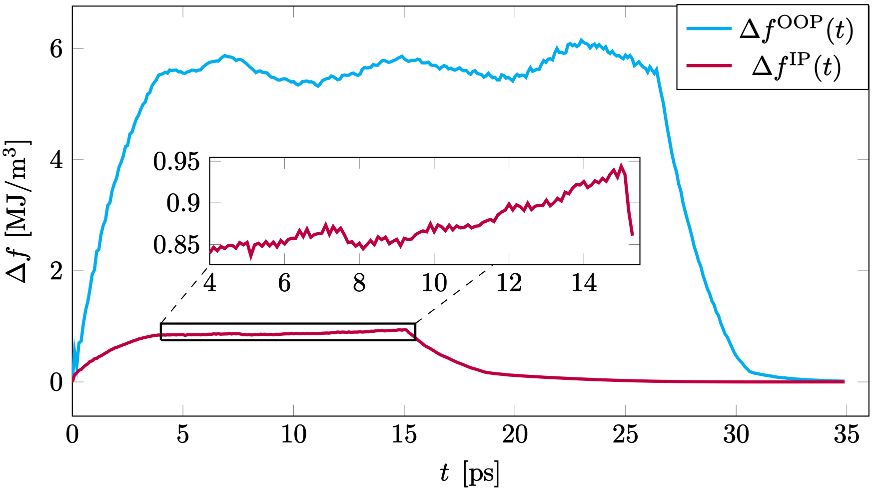


Figure 4.9: Change in free energy density as a function of time, during the excitation of x_{DW} from x_0 to x_{detector}.

As the free energy density in Figs. 4.8 and 4.9 are averages over the whole sample volume, the integral in equation Eq. (2.7) becomes trivial, and we get the simple expression for excitation energy ΔF

$$\Delta F = \left[f(x_{\mathrm{DW}} = x_{\mathrm{detector}}) - f(x_{\mathrm{DW}} = x_0)\right] L_x L_y L_z. \tag{4.3}$$

Using the equation for ΔF and the data displayed in Fig. 4.8 one can estimate the excitation energy needed to simply move the DW from x_0 to x_{DW}. This will be denoted as ΔF_{min} as it is the minimum energy required for putting a neuron into fire mode. However, this would yield a too low estimate for the energy needed to excite the DW, as one would have to excite spin waves in the whole AFM to move the DW. The results in table 4.3 indicate that much of the energy used to move the DW is used on exciting spin waves in the AFM.

Table 4.3: Energy required for moving the DW from x_0 to x_{detector}. ΔF_{min} is just the increase in energy from simply translating the DW, while ΔF_{max} is the energy required form exciting spin waves and moving the DW.

	ΔF_{min}	ΔF_{max}
IP	3.9 zJ	38 aJ
OOP	4.3 zJ	246 aJ

These energy estimates do not consider the generation of $\boldsymbol{h}$, the VCMA or readout circuitry. That being said, energy consumption in the zJ[4] range is still substantially lower than what is reported in [62], where they proposed a FM neuron where the DW is excited by an applied current, that operates at a fJ scale. Furthermore, [64] reports a LIF neuron built with CMOS technology that operates in the pJ range.

4.6 Temperature in the neuron

This section will focus on the effects of introducing a finite temperature to the neuron. It will feature both the Néel temperature and the effect of temperature on x_{DW}.

4.6.1 Néel temperature

The Néel temperature of the AFM neuron is found by examining the component the Néel vector which is in the ground state, $\langle l_z \rangle$ for OOP and $\langle l_x \rangle$ for IP. For these simulations, the parameter variation in K_{easy} and pinning at the far right of the neuron is turned of. The system is initiated with a uniform $\boldsymbol{l}$ along $\mathbf{e}_{\text{easy}}$ and then relaxed for $t = 25\,\text{ps}$ without any applied

---

[4] z = zepto = 1×10^{-21}. a = atto = 1×10^{-18}. f = femto = 1×10^{-15}. p = pico = 1×10^{-12}.

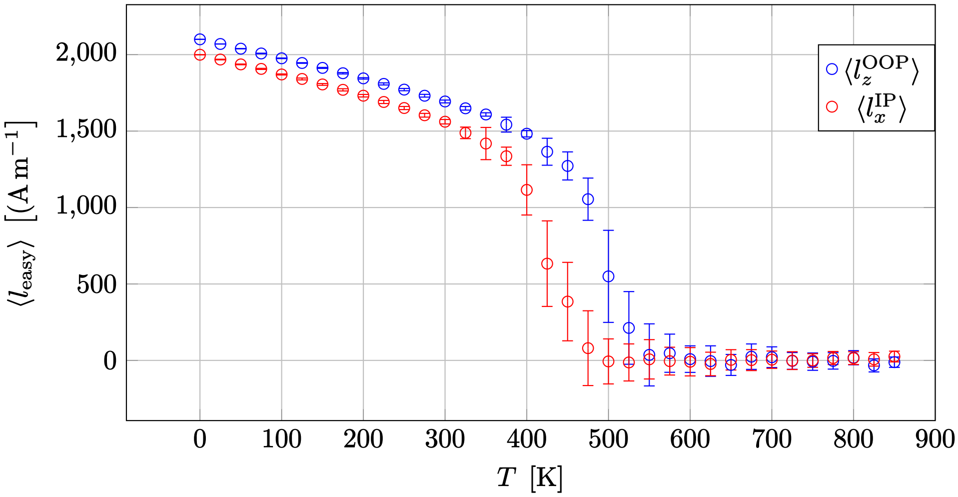


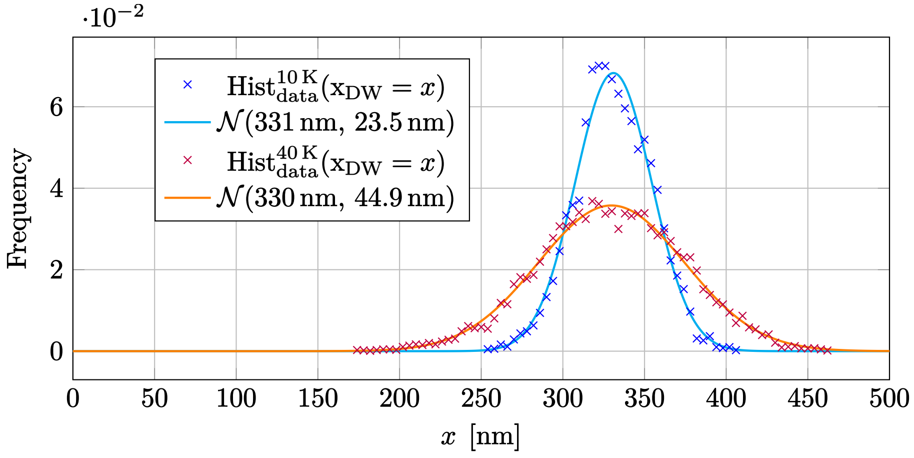
Figure 4.10: Néel temperature of IP and OOP neuron. Magnetic ordering is lost as temperature increases. The graph is the average over 50 simulations. Error bars indicate the standard deviation.

fields. The value of $\langle \boldsymbol{l} \rangle \cdot \mathbf{e}_{\mathrm{easy}} = \langle \boldsymbol{l}_{\mathrm{easy}} \rangle$ is then recorded. Here the average is done over the whole volume of the AFM. This is done for temperatures between $T = 0\,\mathrm{K}$ to $T = 850\,\mathrm{K}$, and 50 simulation is done for each value of T. The resulting graph is shown if Fig. 4.10. It indicates a Néel temperature of about 450 K for OOP and 350 K for IP. We observe that $\langle l_x^{\mathrm{IP}} \rangle < \langle l_x^{\mathrm{OOP}} \rangle$ for all temperatures above the Néel temperatures. Even at $T = 0\,\mathrm{K}$ we have that $\langle l_x^{\mathrm{IP}} \rangle < M_s$. This is a consequence of the DMI that is only present in the IP neuron. As mentioned in section 2.1, the DMI causes a tilting of spins, and the ground state of the system is not the uniform $\boldsymbol{l} = \pm M_s \mathbf{e}_{\mathrm{easy}}$, but has some texture along the edges of the material. This can also be observed in Fig. 4.1b, where there are some variations in l_z, as opposed to the more inform l_x for OOP in Fig. 4.1a.

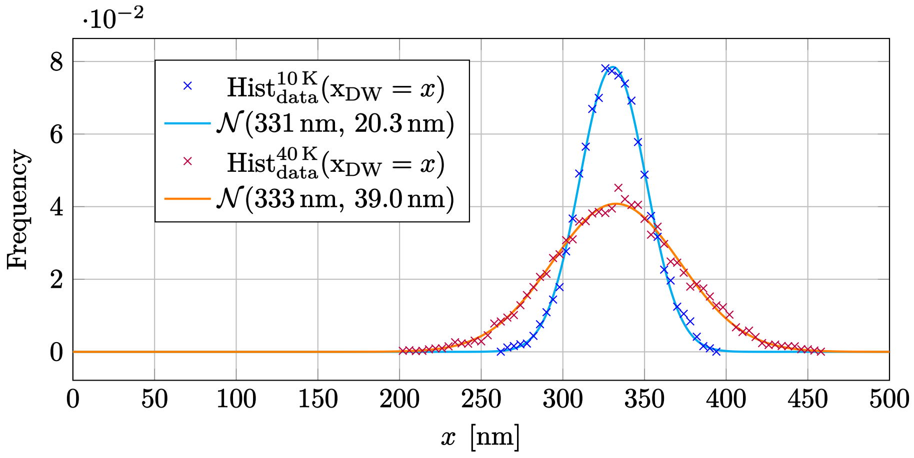
4.6.2 Temperature-induced Domain wall movement

The loss of magnetic order in the neuron is detrimental to any computing application. However, we see that T_{N} is above room temperature. This however does not necessarily mean that the proposed neuron would work at $T = 300\,\mathrm{K}$. Having some magnetic ordering is not necessarily sufficient for maintaining the desired capabilities. As temperature increases, the DW in the chip will start to fluctuate around its ground state. The quadratic form of $K_{\text{easy}}(x)$ is to ensure the DW does not move too far from x_0. Despite this, given a large enough temperature, the DW will be able to move so far left that it can initiate a spike, and even move so far left or right that it disappears out of the neuron. The increase in damping on the far right edge helps in this regard but is not enough to ensure that the DW does not move out of the system, resulting in a uniformly magnetized chip. The required temperature to cause this is far less than the Néel temperature. Fig. 4.11 shows histograms over the now stochastic variable $\mathrm{x_{DW}}$ for temperatures at $T = 10\,\mathrm{K}$ and $T = 40\,\mathrm{K}$. The fitted normal distributions indicate almost a doubling of the standard deviation σ in both the IP and OOP neurons. The slight shift in x_0 towards the left can be understood as a consequence of the increased damping at the right end of the AFM, which makes the system unsymmetrical around x_0. We also observe that the OOP neuron has smaller temperature-induced fluctuations than the IP neuron.

Considering that the energy potential well keeping the DW in place is $\sim$ zJ it is remarkable that the DW does not fluctuate more as a consequence of temperature. Furthermore, as noisy the movement of the magnetization is induced by temperature, the spin wave signal traveling down the chip gets weaker and distorted. In [65] they model a noisy neuron by adding a Wiener process [66] to a noiseless neuron. This is quite similar to the temperature-induced DW motion. In [67] they argue that stochastic neurons can be important for probabilistic calculations in the brain.



(a)



(b)

Figure 4.11: Temperature induced DW movement for **(a)** IP neuron and **(b)** OOP neuron. The introduction of finite temperature turns DW position into a stochastic variable. $\mathrm{x_{DW}}$ was observed for $t = 5\,\mathrm{ns}$. $\mathrm{Hist}_{\mathrm{data}}^{T}(\mathrm{x_{DW}} = x)$ are normalized histograms over the recorded $\mathrm{x_{DW}}$ over the observation time, and indicates draws from the data generating probability distribution. Normal distributions $\mathcal{N}(\mu, \sigma)$ are fitted for both $\mathrm{Hist}_{\mathrm{data}}^{10\,\mathrm{K}}(\mathrm{x_{DW}} = x)$ and $\mathrm{Hist}_{\mathrm{data}}^{40\,\mathrm{K}}(\mathrm{x_{DW}} = x)$.

4.7 Summary - Neuron implementation

This chapter has explained the idea and setup for the proposed AFM neurons. The position of the DW encodes the state of the neuron, which exhibits leaky integrate-and-fire behavior. Methods to move the DW and read out the position of the DW have been included. The OOP and IP neurons are quite similar, although they differ slightly in geometrical setup. The biggest difference is that the IP neuron with DMI interaction exhibits both attractive and repulsive movement of x_{DW} under the change of handedness of the applied field $\boldsymbol{h}$. Furthermore, the IP neuron can operate at lower applied field amplitudes than the OOP neuron. However, the OOP neuron is slightly more robust against temperature fluctuations than the IP neuron. Since the IP neuron is more sensitive to lower amplitudes of $\boldsymbol{h}$ one could increase the value of K_{easy} to obtain a more stable neuron in regards to temperature-induced DW fluctuations, and still operate at similar field amplitude values as the OOP neuron. Upon comparison with the ferromagnetic and metallic spiking neuron proposed in [62] our neuron indicates quicker dynamics and better energy efficiency.

5 Neuromorphic computing application

This chapter will focus on finding a surrogate model that describes the dynamics of the spiking neuron, and discuss a potential neural network application. This chapter will only focus on the IP neuron and exclude temperature.

5.1 Surrogate model for the AFM spiking neuron

The full-scale simulations of the spiking neuron are quite computationally expensive. To investigate neural networks consisting of this type of neuron would require simulating a large quantity at once, and is not pragmatic. Therefore it is desirable to find a surrogate model for the spiking neuron. Furthermore, such a model could yield additional insight into the system. The spiking neuron is so complex and non-linear, that a LIF neuron model as previously mentioned in section 3.3.2 will not be sufficient to describe the behavior. To see this, consider that the DW has a tendency to overshoot when relaxing towards x_0, as in Fig. 4.6. Considering a LIF neuron without external stimuli, the state of the system is completely described by the voltage u. In the AFM spiking neuron, we substitute $u \to x_{\mathrm{DW}}$. However, if the system is completely described by x_{DW} alone, then overshooting x_0 should not be possible at the same time as just staying still at x_0, which of course must be possible. This section will consider finding a neuron model that is computationally less expensive than the micro-magnetic simulations in Boris, yet captures the complex behavior of the neuron. The starting approximation is to describe the system by two variables, the domain wall position and velocity along the x-axis. The state vector of the system $\boldsymbol{X}$ is then

$$\boldsymbol{X} = \begin{bmatrix} x_{\mathrm{DW}} \\ v_{\mathrm{DW}} \end{bmatrix}. \tag{5.1}$$

x_{DW} is obviously essential for any surrogate model, as it is this variable that record the integration of the inputs, and determines when the spiking event occurs. v_{DW} is perhaps a more subtle inclusion, however it is important to include this as AFMs displays some damping behavior in the neuron dynamics.

The ansatz for the time evolution of $\boldsymbol{X}$ is on the following form

$$\frac{d\boldsymbol{X}}{dt} = \begin{bmatrix} v_{\text{DW}} \\ w(\boldsymbol{h}) \cdot I(t - t_{\text{delay}}(x_{\text{DW}})) \cdot f(x_{\text{DW}}) + g(x_{\text{DW}}) + d(v_{\text{DW}}) \end{bmatrix}, \tag{5.2}$$

where $w(\boldsymbol{h})I(t - t_{\text{delay}})$ is the incoming presynaptic spikes weighted by $w(\boldsymbol{h})$ which is a function of the applied field strength. $f(x_{\text{DW}})$ is a function that determines the strength of the spiking signal at the DW position, $g(x_{\text{DW}})$ is the function that describes the relaxation force caused by the VCMA, and $d(x_{\text{DW}})$ encodes the damping effect of the DW. In other words, $I(t - t_{\text{delay}}(x_{\text{DW}})) \cdot f(x_{\text{DW}})$ ensures integration of the input into the system, $g(x_{\text{DW}})$ is the leaky factor, and $d(v_{\text{DW}})$ is a consequence of the characteristic physics of the AFM neuron. The term "force" is applied loosely here, as it perhaps is more correct to talk about DW acceleration. Here we will only focus on one specific applied field strength, and set this value of $w(\boldsymbol{h}) = 1$.

The time t is the time of the input signal being applied. The delay time in $t_{\text{delay}}(x_{\text{DW}})$ is a consequence of the time it takes for the spin waves to travel from x_{detector} to x_{DW}. Although the spin waves move fast, modelling without considering the time delay deemed unfeasible. Using the estimate for spin wave velocity v_{g}^{SW} in section 4.3 and furthermore assuming a constant velocity we get

$$t_{\text{delay}}(x_{\text{DW}}) = \frac{x_{\text{DW}}}{v_{\text{g}}^{\text{SW}}} \sim \frac{x_{\text{DW}}}{L_x} 3.33\,\text{ps}. \tag{5.3}$$

Some simple and general observations could be made about the function $g(x_{\text{DW}})$, that will aid in making a reasonable ansatz. Firstly, it has to be a odd function around the equilibrium position x_0, to ensure that it constitutes a force that always points towards x_0. This is a consequence of the symmetric design of the VCMA around x_0, see Fig 4.3. Secondly, we have to require that $g(x_{\text{DW}} = x_0) = 0$ as any movement due to VCMA at x_0 is contradictory to the fact that it is a steady state position of x_{DW}. An simple ansatz for $g(x_{\text{DW}})$ that meet this requirements is

$$g(x_{\text{DW}}) = b_1(x_{\text{DW}} - x_0), \tag{5.4}$$

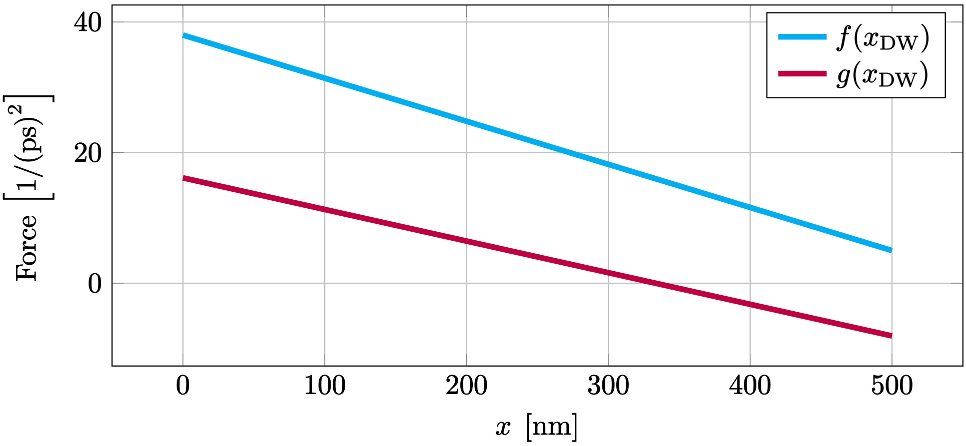


Figure 5.1: Spike force and relax force in the neuron.

where b_2 and b_1 are some coefficients to be determined.

The damping function $d(v_{\mathrm{DW}})$ as to act in opposite direction of v_{DW} as the opposite would be unphysical. The equation $\mathrm{sign}(d(v_{\mathrm{DW}})) = -\mathrm{sign}(v_{\mathrm{DW}})$ is satisfied by the simple

$$d(v_{\mathrm{DW}}) = -c_1 \cdot v_{\mathrm{DW}}. \tag{5.5}$$

Furthermore, we take $f(x_{\mathrm{DW}})$ has to be directed towards x_{detector} for and towards L_x for . Here we only consider, and take $f(x_{\mathrm{DW}})$ to be

$$f(x_{\mathrm{DW}}) = -a_1(x_{\mathrm{DW}} - x_0) + a_0. \tag{5.6}$$

The coefficients in Eqs. (5.4), (5.5) and (5.6) is given in table 5.1.

Table 5.1: Coefficients used in neuron surrogate model.

Constant	Value	Unit
a_0	38	$\mathrm{nm}/(\mathrm{ps})^2$
a_1	0.069	$1/(\mathrm{ps})^2$
b_1	0.0504	$\mathrm{nm}/(\mathrm{ps})^2$
c_1	0.275	$1/\mathrm{ps}$

Even though this is a simplification of a rather complex system, the model displays correspondence with the micro-magnetic simulation, as shown in Fig 5.2. The surrogate model indicates that it is possible to reduce the AFM spiking neuron down to two coupled differential equations of two variables, x_{DW} and v_{DW}. The surrogate model is comparable to the differential equations describing a harmonic oscillator which is perturbed by some force [31]. Furthermore, we observe that the quadratic function K_{easy} results in a linear relaxation force in Eq. 5.4. The spike force is also a linear function of x_{DW}. It indicates that the perturbing effect of exciting spin waves on the DW gets weaker the further the DW is from the injector. This can probably be understood as a consequence of spin waves being damped as they propagate through the material, even though our damping coefficient is rather small, $\alpha = 0.002$. Observing Fig. 5.2 we see that the domain wall moves a relatively large distance on the first two spikes. This is despite the fact of a weaker spike force $f(x_{\mathrm{DW}})$. It appears the weaker relaxation force g_{DW} allows for a large excitation of the DW position.

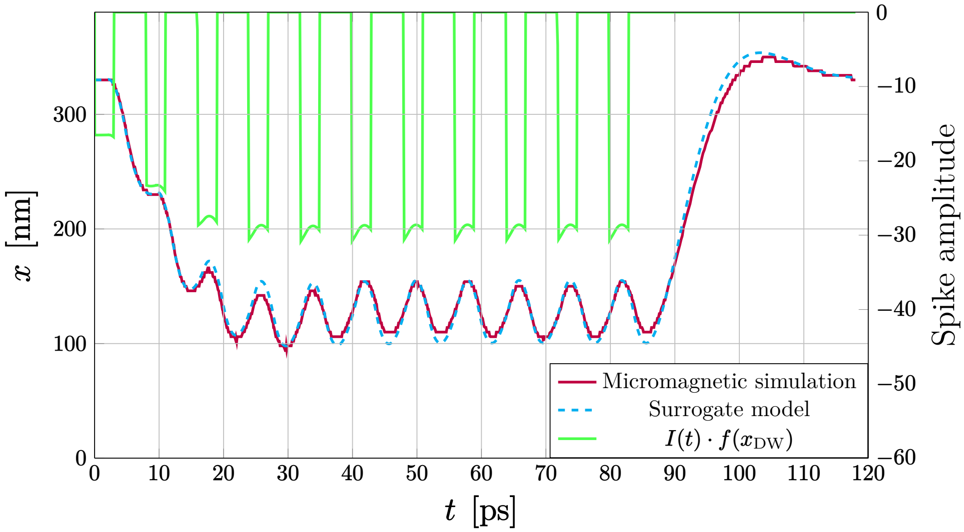


Figure 5.2: Micromagnetic simulation and surrogate model. Both models are subjected to the same input $I(t)$ where all spikes have the same amplitude. Note that the presynaptic spikes are given as $I(t)$ and not $I(t - t_{\mathrm{delay}})$, i.e. the spikes plotted at the time they were applied, and not when they acted on the DW. The left y-axis indicates DW position and the right axis indicates the amplitudes of spikes.

5.2 Neural networks

Up until now, we have only considered a single neuron. To be able to learn complicated functions, it is necessary with an ensemble of interconnected neurons, as described in section 3.2. A crossbar synapse as in [62] could be employed to connect neurons together. Simulations indicate that how strong DW movement is dependent on the applied field strength, as in Eq. 5.2. Tweaking parameters two learn a function could be achieved by somehow adjusting the $w(\boldsymbol{h})$ function. This is however outside the scope of this thesis.

Once the simpler equation of the surrogate model is obtained, it becomes straightforward to model neural networks. There exists software for simulating spiking neural networks on conventional computers [68]. The surrogate model could easily be integrated into such a system to yield information about how and if the neuron could be capable of actual machine learning.

6 Further work and conclusion

This chapter will serve as an epitome for the thesis in general. Suggested further work will feature before a final conclusion is given.

6.1 Further work

Experiments will have to be conducted on the effects of voltage-controlled anisotropy on antiferromagnets. It is necessary to investigate whether it is possible to achieve a 40% difference in anisotropy coefficients in antiferromagnets as we have assumed in this thesis. Exploration of materials that inhibits the imperative characteristics for building a spiking neuron will be of importance in the future. This thesis has not dealt with how to efficiently convert the electrical signal measured at the detector into coherent spin waves again, which will be crucial for building a hardware network of these computing units. The idea of a system that could utilize spintronic technologies for all aspects such as excitation, signal detection, and perhaps most importantly signal transfer and fan-out is compelling. As energy efficiency is an important concept of both analog and neuromorphic computing, it is important to seek a setup that can function in room temperature conditions. This is to avoid spending energy on cooling the system. Additionally, it could be of interest to investigate if temperature-induced domain wall fluctuations could provide an efficient gateway for stochastic spiking neural networks. We have seen that energy is lost as spin waves are damped when propagating through the material. It would be interesting to investigate excitations that are local to the domain wall, such that the spin waves are not dependent on moving significant distances before reaching their desired destination.

6.2 Conclusion

Numerical simulations of an antiferromagnetic and insulating spiking neuron have demonstrated a successful proof-of-concept. The state of the neuron is described by the position of a domain wall, which is modulated by voltage-controlled anisotropy and coherent spin wave excitation. This is not dependent on the application of electrical currents, mitigating the issue of Joule heating. The neuron displays leaky integrate-and-fire properties, similar to a biological neuron and the models used for simulating spiking neural networks. Compared to other hardware neuron proposals, it indicates the possibility of a significantly improved power efficiency. Furthermore, it can operate at pico second timescales. We have also seen that it is possible to create a surrogate model that simplifies simulations yet still captures the dynamics of the neuron. The simulations show promising results for energy-efficient and fast computing, inspiring further work in neuromorphic computing based on spintronics. The work in this thesis will be the foundation for a paper that will be submitted to the journal *Neuromorphic Computing and Engineering* for their special edition on *Topological Solitons for Neuromorphic Systems* [69].

Bibliography

[1] A. Mahmoud et al. “An Introduction to Spin Wave Computing”. In: *Journal of Applied Physics* 128.16 (Oct. 28, 2020), p. 161101. ISSN: 0021-8979, 1089-7550. arXiv: 2006.12905. URL: http://arxiv.org/abs/2006.12905 (visited on 03/28/2022).

[2] G. Moore. “Cramming More Components Onto Integrated Circuits”. In: *Proceedings of the IEEE* 86.1 (Jan. 1998). Reprinted from Gordon E. Moore, “Cramming More Components onto Integrated Circuits,” Electronics, pp. 114–117, April 19, 1965., pp. 82–85. ISSN: 1558-2256.

[3] M. Bohr. “A 30 Year Retrospective on Dennard’s MOSFET Scaling Paper”. In: *IEEE Solid-State Circuits Society Newsletter* 12.1 (2007). Conference Name: IEEE Solid-State Circuits Society Newsletter, pp. 11–13. ISSN: 1098-4232.

[4] R. S. Williams. “What’s Next? [The end of Moore’s law]”. In: *Computing in Science Engineering* 19.2 (Mar. 2017). Conference Name: Computing in Science Engineering, pp. 7–13. ISSN: 1558-366X.

[5] *AlphaGo: The story so far*. Deepmind. URL: https://deepmind.com/research/case-studies/alphago-the-story-so-far#what_is_go_ (visited on 12/17/2021).

[6] E. Goi et al. “Perspective on photonic memristive neuromorphic computing”. In: *PhotoniX* 1.1 (Mar. 3, 2020), p. 3. ISSN: 2662-1991. URL: https://doi.org/10.1186/s43074-020-0001-6 (visited on 12/17/2021).

[7] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. URL: https://www.deeplearningbook.org/ (visited on 10/04/2021).

[8] S. Furber and P. Bogdan. *SpiNNaker - A Spiking Neural Network Architecture*. OCLC: 1286310625. 2020. ISBN: 978-1-68083-652-3 978-1-68083-653-0. URL: https://directory.doabooks.org/handle/20.500.12854/67941 (visited on 06/06/2022).

[9] D. Christensen et al. “2022 roadmap on neuromorphic computing and engineering”. In: *Neuromorphic Computing and Engineering* (Jan. 12, 2022).

[10] T. Geszti. *Physical Models of Neural Networks*. Budapest: World Scientific Publishing Co. Pte. Ltd., 1990. ISBN: 981-02-0012-9.

[11] J. von Neumann. “First draft of a report on the EDVAC”. In: *IEEE Annals of the History of Computing* 15.4 (1993). Conference Name: IEEE Annals of the History of Computing, pp. 27–75. ISSN: 1934-1547.

[12] F. Peper. “The End of Moore’s Law: Opportunities for Natural Computing?” In: *New Generation Computing* 35.3 (July 1, 2017), pp. 253–269. ISSN: 1882-7055. URL: https://doi.org/10.1007/s00354-017-0020-4 (visited on 12/19/2021).

[13] J. Grollier et al. “Neuromorphic spintronics”. In: *Nature Electronics* 3.7 (July 2020). Number: 7 Publisher: Nature Publishing Group, pp. 360–370. ISSN: 2520-1131. URL: https://www.nature.com/articles/s41928-019-0360-9 (visited on 06/20/2022).

[14] M. Davies et al. “Loihi: A Neuromorphic Manycore Processor with On-Chip Learning”. In: *IEEE Micro* 38.1 (Jan. 2018). Conference Name: IEEE Micro, pp. 82–99. ISSN: 1937-4143.

[15] M. Davies et al. “Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook”. In: *Proceedings of the IEEE* 109.5 (May 2021), pp. 911–934. ISSN: 0018-9219, 1558-2256. URL: https://ieeexplore.ieee.org/document/9395703/ (visited on 03/18/2022).

[16] Y. Tsividis. *Not Your Father’s Analog Computer*. IEEE Spectrum. Section: Computing. Dec. 1, 2017. URL: https://spectrum.ieee.org/not-your-fathers-analog-computer (visited on 06/20/2022).

[17] S. Köppel et al. “Using analog computers in today’s largest computational challenges”. In: *Advances in Radio Science* 19 (Dec. 17, 2021), pp. 105–116. ISSN: 1684-9973. arXiv: 2102.07268[physics]. URL: http://arxiv.org/abs/2102.07268 (visited on 06/20/2022).

[18] S. Köppel, A. Krause, and B. Ulmann. *Analog Computing for Molecular Dynamics*. Number: arXiv:2107.06283. July 13, 2021. arXiv: 2107.06283[physics]. URL: http://arxiv.org/abs/2107.06283 (visited on 06/20/2022).

[19] R. Lebrun et al. “Tunable long-distance spin transport in a crystalline antiferromagnetic iron oxide”. In: *Nature* 561.7722 (Sept. 2018), pp. 222–225. ISSN: 0028-0836, 1476-4687. URL: http://www.nature.com/articles/s41586-018-0490-7 (visited on 06/30/2022).

[20] W. Yu, J. Xiao, and G. E. W. Bauer. *A Hopfield neural network in magnetic films with natural learning*. Number: arXiv:2101.03016. Jan. 8, 2021. arXiv: 2101.03016[cond-mat]. URL: http://arxiv.org/abs/2101.03016 (visited on 06/20/2022).

[21] D. J. Griffiths. *Introduction to electrodynamics*. 4th ed. Cambridge: University Press, 2017. XVIII, 599. ISBN: 978-1-108-42041-9.

[22] S. Baierl et al. “Terahertz-Driven Nonlinear Spin Response of Antiferromagnetic Nickel Oxide”. In: *Physical Review Letters* 117.19 (Nov. 1, 2016), p. 197201. ISSN: 0031-9007, 1079-7114. URL: https://link.aps.org/doi/10.1103/PhysRevLett.117.197201 (visited on 06/30/2022).

[23] P. C. Hemmer. *Kvantemekanikk*. 5. utg. Book Title: Kvantemekanikk. Trondheim: Tapir akademisk forl, 2005. 398 pp. ISBN: 978-82-519-2028-5. URL: https://www.nb.no/search?q=oaiid:"oai:nb.bibsys.no:990505369154702202"&mediatype=bÃÿker (visited on 12/20/2021).

[24] S. Chikazumi and C. D. Graham Jr. *Physics of Ferromagnetism*. 2nd ed. Vol. 94. International Series of Monographs on Physics. Oxford University Press Inc., New York, 1997. ISBN: 978-0-19-956481-1.

[25] B. Hillebrands, K. Ounadjela, and A. Thiaville. *Spin dynamics in confined magnetic structures: 1*. Vol. vol. 83. Topics in applied physics. Berlin: Springer, 2002. XVI, 336. ISBN: 978-3-540-41191-8.

[26] S. Lepadatu. “Boris computational spintronics—High performance multi-mesh magnetic and spin transport modeling software”. In: *Journal of Applied Physics* 128.24 (Dec. 28, 2020). Publisher: American Institute of Physics, p. 243902. ISSN: 0021-8979. URL: https://aip.scitation.org/doi/full/10.1063/5.0024382 (visited on 11/08/2021).

[27] P. Li et al. “Numerical methods for antiferromagnetics”. In: *IEEE Transactions on Magnetics* 56.4 (Apr. 2020), pp. 1–9. ISSN: 0018-9464, 1941-0069. arXiv: 1911.01717. URL: http://arxiv.org/abs/1911.01717 (visited on 05/09/2022).

[28] R. Skomski. *Simple models of magnetism*. Oxford graduate texts. Oxford: University Press, 2008. XVI, 349. ISBN: 978-0-19-857075-2.

[29] T. H. Pate. “Hölder type matrix inequalities of Pate, Blakley, and Roy extended to the inner product of Frobenius”. In: *Linear Algebra and its Applications.* Special Issue in Honor of José Perdigão Dias da Silva 436.6 (Mar. 15, 2012), pp. 1763–1769. ISSN: 0024-3795. URL: https://www.sciencedirect.com/science/article/pii/S0024379511006525 (visited on 12/15/2021).

[30] B. Rana and Y. Otani. “Towards magnonic devices based on voltage-controlled magnetic anisotropy”. In: *Communications Physics* 2.1 (Dec. 2019), p. 90. ISSN: 2399-3650. URL: http://www.nature.com/articles/s42005-019-0189-6 (visited on 03/21/2022).

[31] M. S. Rogalski. *Advanced university physics.* In collab. with S. B. Palmer. 2nd ed. Boca Raton, Fla: Chapman & Hall/CRC, 2006. XV, 1006. ISBN: 978-1-58488-511-5.

[32] O. V. Pylypovskyi et al. “Curvature-driven homogeneous Dzyaloshinskii–Moriya interaction and emergent weak ferromagnetism in anisotropic antiferromagnetic spin chains”. In: *Applied Physics Letters* 118.18 (May 3, 2021), p. 182405. ISSN: 0003-6951, 1077-3118. URL: https://aip.scitation.org/doi/10.1063/5.0048823 (visited on 06/28/2022).

[33] V. Baltz et al. “Antiferromagnetic spintronics”. In: *Reviews of Modern Physics* 90.1 (Feb. 15, 2018), p. 015005. ISSN: 0034-6861, 1539-0756. URL: https://link.aps.org/doi/10.1103/RevModPhys.90.015005 (visited on 05/07/2022).

[34] A. Qaiumzadeh, L. A. Kristiansen, and A. Brataas. “Controlling chiral domain walls in antiferromagnets using spin-wave helicity”. In: *Physical Review B* 97.2 (Jan. 9, 2018), p. 020402. ISSN: 2469-9950, 2469-9969. URL: https://link.aps.org/doi/10.1103/PhysRevB.97.020402 (visited on 05/07/2022).

[35] E. Haltz et al. “Domain wall dynamics in antiferromagnetically-coupled double-lattice systems”. In: *Physical Review B* 103.1 (Jan. 27, 2021), p. 014444. ISSN: 2469-9950, 2469-9969. arXiv: 2010.15443[cond-mat]. URL: http://arxiv.org/abs/2010.15443 (visited on 06/20/2022).

[36] A. Brataas, A. D. Kent, and H. Ohno. “Current-induced torques in magnetic materials”. In: *Nature Materials* 11.5 (2012). Num Pages: 10 Publisher: Nature Publishing Group, The Macmillan Building London N1 9XW United Kingdom, pp. 372–381. ISSN: 1476-1122. URL: https://www.proquest.com/docview/1031325250?pq-origsite=primo& (visited on 10/20/2021).

[37] H. Goldstein, J. L. Safko, and C. P. Poole. *Classical mechanics*. Third edition. Always learning. Harlow, England: Pearson, 2014. XVIII, 638. ISBN: 978-1-292-02655-8.

[38] R. Khoshlahni et al. "Ultrafast generation and dynamics of isolated skyrmions in antiferromagnetic insulators". In: *Physical Review B* 99.5 (Feb. 21, 2019). Publisher: American Physical Society, p. 054423. URL: https://link.aps.org/doi/10.1103/PhysRevB.99.054423 (visited on 10/20/2021).

[39] E. Kreyszig. *Advanced engineering mathematics*. 9th ed. Hoboken, N.J: Wiley, 2006. XVII, 1094, 129. ISBN: 978-0-471-72897-9.

[40] Y. Xu, D. D. Awschalom, and J. Nitta, eds. *Handbook of Spintronics*. Dordrecht: Springer Netherlands, 2016. ISBN: 978-94-007-6891-8 978-94-007-6892-5. URL: http://link.springer.com/10.1007/978-94-007-6892-5 (visited on 06/27/2022).

[41] L. Shen et al. "Dynamics of the antiferromagnetic skyrmion induced by a magnetic anisotropy gradient". In: *Physical Review B* 98.13 (Oct. 30, 2018), p. 134448. ISSN: 2469-9950, 2469-9969. URL: https://link.aps.org/doi/10.1103/PhysRevB.98.134448 (visited on 03/16/2022).

[42] M. Ghidini et al. "Non-volatile voltage control of in-plane and out-of-plane magnetization in polycrystalline Ni films on ferroelectric PMN–PT (001) $_{\mathrm{pc}}$ substrates". In: *Journal of Applied Physics* 129.15 (Apr. 21, 2021), p. 154101. ISSN: 0021-8979, 1089-7550. URL: https://aip.scitation.org/doi/10.1063/5.0040258 (visited on 06/27/2022).

[43] I. D. Mayergoyz, G. Bertotti, and C. Serpico. *Nonlinear Magnetization Dynamics in Nanosystems*. Elsevier series in electromagnetism. Oxford: Elsevier Science & Technology, 2008. ISBN: 978-0-08-044316-4.

[44] A. V. Chumak et al. "Roadmap on Spin-Wave Computing". In: (Oct. 30, 2021). arXiv: 2111.00365. URL: http://arxiv.org/abs/2111.00365 (visited on 05/09/2022).

[45] J. R. Hortensius et al. "Coherent spin-wave transport in an antiferromagnet". In: *Nature Physics* 17.9 (Sept. 2021). Number: 9 Publisher: Nature Publishing Group, pp. 1001–1006. ISSN: 1745-2481. URL: https://www.nature.com/articles/s41567-021-01290-4 (visited on 05/09/2022).

[46] X. R. Wang. "Anomalous spin Hall and inverse spin Hall effects in magnetic systems". In: *Communications Physics* 4.1 (Dec. 2021), p. 55. ISSN: 2399-3650. URL: http://www.nature.com/articles/s42005-021-00557-9 (visited on 05/12/2022).

[47] M. I. Dyakonov. *Spin Hall Effect.* Number: arXiv:1210.3200. Oct. 11, 2012. arXiv: 1210.3200[cond-mat,physics:physics,physics:quant-ph]. URL: http://arxiv.org/abs/1210.3200 (visited on 06/25/2022).

[48] S. Lepadatu. "Unified treatment of spin torques using a coupled magnetisation dynamics and three-dimensional spin current solver". In: *Scientific Reports* 7.1 (Oct. 11, 2017). Number: 1 Publisher: Nature Publishing Group, p. 12937. ISSN: 2045-2322. URL: https://www.nature.com/articles/s41598-017-13181-x (visited on 05/12/2022).

[49] V. T. Pham et al. "Electrical detection of magnetic domain walls by inverse and direct spin Hall effect". In: *Applied Physics Letters* 109.19 (Nov. 7, 2016), p. 192401. ISSN: 0003-6951, 1077-3118. URL: http://aip.scitation.org/doi/10.1063/1.4967171 (visited on 06/28/2022).

[50] R. Cheng et al. "Spin Pumping and Spin-Transfer Torques in Antiferromagnets". In: *Physical Review Letters* 113.5 (July 29, 2014), p. 057601. ISSN: 0031-9007, 1079-7114. URL: https://link.aps.org/doi/10.1103/PhysRevLett.113.057601 (visited on 05/12/2022).

[51] D. A. Woodruff. *What is a neuron?* Nov. 22, 2016. URL: https://qbi.uq.edu.au/brain/brain-anatomy/what-neuron (visited on 12/19/2021).

[52] S. N. *English: Structure of Neuron (unlabeled).* Feb. 23, 2018. URL: https://commons.wikimedia.org/wiki/File:Structure_of_Neuron.png (visited on 12/19/2021).

[53] W. S. McCulloch and W. Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *Bulletin of mathematical biology* 52.1 (1990). Place: NEW YORK Publisher: Elsevier Ltd, pp. 99–115. ISSN: 0092-8240.

[54] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain". In: *Psychological review* 65.6 (1958). Place: United States Publisher: American Psychological Association, pp. 386–408. ISSN: 0033-295X.

[55] K. Hornik, M. Stinchcombe, and H. White. "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5 (1989). Place: OXFORD Publisher: Elsevier Ltd, pp. 359–366. ISSN: 0893-6080. URL: https://search.proquest.com/docview/25289495?pq-origsite=primo (visited on 12/20/2021).

[56] G. Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of control, signals, and systems* 2.4 (1989), pp. 303–314. ISSN: 0932-4194.

[57] W. Maass. "Networks of spiking neurons: The third generation of neural network models". In: *Neural Networks* 10.9 (Dec. 1, 1997), pp. 1659–1671. ISSN: 0893-6080. URL: https://www.sciencedirect.com/science/article/pii/S0893608097000117 (visited on 06/22/2022).

[58] W. Gerstner. *Spiking neuron models: single neurons, populations, plasticity.* In collab. with W. M. Kistler. Cambridge: University Press, 2002. XIV, 480. ISBN: 978-0-521-81384-6.

[59] S. M. Bohte, J. N. Kok, and H. La Poutré. "Error-backpropagation in temporally encoded networks of spiking neurons". In: *Neurocomputing (Amsterdam)* 48.1 (2002). Publisher: Elsevier B.V, pp. 17–37. ISSN: 0925-2312.

[60] S. Thorpe and J. Gautrais. "Rank Order Coding". In: Computational Neuroscience: Trends in Research. Dec. 1, 1998, pp. 113–118. ISBN: 978-1-4613-7190-8.

[61] F. Ponulak and A. Kasiński. "Introduction to spiking neural networks: Information processing, learning and applications". In: *Acta neurobiologiae experimentalis* 71 (Jan. 1, 2011), pp. 409–33.

[62] A. Agrawal and K. Roy. "Mimicking Leaky-Integrate-Fire Spiking Neuron Using Automotion of Domain Walls for Energy-Efficient Brain-Inspired Computing". In: *IEEE Transactions on Magnetics* 55.1 (Jan. 2019). Conference Name: IEEE Transactions on Magnetics, pp. 1–7. ISSN: 1941-0069.

[63] O. Gomonay, M. Kläui, and J. Sinova. "Manipulating antiferromagnets with magnetic fields: Ratchet motion of multiple domain walls induced by asymmetric field pulses". In: *Applied Physics Letters* 109.14 (Oct. 3, 2016). Publisher: American Institute of Physics, p. 142404. ISSN: 0003-6951. URL: https://aip.scitation.org/doi/full/10.1063/1.4964272 (visited on 06/20/2022).

[64] S. Dutta et al. "Leaky Integrate and Fire Neuron by Charge-Discharge Dynamics in Floating-Body MOSFET". In: *Scientific reports* 7.1 (2017). Place: LONDON Publisher: Springer Nature, pp. 8257–7. ISSN: 2045-2322.

[65] W. Olin-Ammentorp et al. "Stochasticity and robustness in spiking neural networks". In: *Neurocomputing* 419 (Jan. 2021), pp. 23–36. ISSN: 09252312. (Visited on 06/29/2022).

[66] N. G. van Kampen. *Stochastic Processes in Physics and Chemistry.* North-Holland personal library. Amsterdam: Elsevier Science & Technology, 2007. ISBN: 978-0-444-52965-7.

[67] J. Fiser et al. “Statistically optimal perception and learning: from behavior to neural representations”. In: *Trends in Cognitive Sciences* 14.3 (Mar. 2010), pp. 119–130. ISSN: 13646613. URL: https://linkinghub.elsevier.com/retrieve/pii/S1364661310000045 (visited on 06/29/2022).

[68] H. Hazan et al. “BindsNET: A Machine Learning-Oriented Spiking Neural Networks Library in Python”. In: *Frontiers in Neuroinformatics* 12 (2018). ISSN: 1662-5196. URL: https://www.frontiersin.org/article/10.3389/fninf.2018.00089 (visited on 05/21/2022).

[69] *Focus issue on Topological Solitons for Neuromorphic Systems - Neuromorphic Computing and Engineering - IOPscience.* URL: https://iopscience.iop.org/journal/2634-4386/page/focus-issue-on-topological-solitons-for-neuromorphic-systems (visited on 07/01/2022).

Appendix - Specialization project

For completeness, the specialization project from the fall semester 2021 is attached to this master's thesis. It should be viewed as a completely independent text from the master's thesis. It provides insight into another neuromorphic application based on spintronics.

Simulating the physical realization of a Hopfield neural network for associative memory in magnetic films with neuromorphic spintronics

Author: Johannes Wiger Austefjord
Supervisor: Alireza Qaiumzadeh

July 1, 2022

Abstract

The plasticity and non-linearity of the stripe skyrmions that arise in magnetic thin films with a large Dzyaloshinskii-Moriya interaction is utilized for simulating a hardware based artificial neural network. By applying voltages to the thin films one is able to both compute and store the weights of the network within the films. The computational architecture in the magnetic films are fundamentally different then that of conventional computers, and is not limited by the Von Neumann bottleneck. The idea is conceptualized by numerical simulations of a 4 node Hopfield neural network able of practicing associative memory. A theoretical framework for both spintronics and neural networks is provided. The results try to emulate the results in the paper *Hopfield neural network in magnetic textures with intrinsic Hebbian learning* by Yu et al. [1]. This paper will use the software *Boris Computational Spintronics* [2], as opposed to Yu et al. who ran their simulations on *COMSOL Multiphysics®* [3] with a costumed built module for micromagnetics [4]. The ramification of the proceedings is a proof-of-concept for the realization of a physical Hopfield neural network.

Contents

Acknowledgments

I would like the opportunity to thank the following people for aiding me in my work with this report.

Serban Lepadatu for graciously providing help whenever needed with his computational software, *Boris computational spintronics* [2].

Dennis Meier for providing me with sources and some of his images, and allowing me to use them in this report.

Finally I would like to thank Alireza for supervising me this semester, and giving me the opportunity to challenge myself with this exciting project.

-Johannes Wiger Austefjord

Introduction

The famous *Moore's law* states that the number of transistors one can fit on a chip will grow exponentially [5]. This have been the case for several decades, but is predicted to come to a halt as transistors start to reach the size of a couple of atoms. This has sparked an interest in finding new ways of building logic units capable of arithmetic, integrating different types of computing hardware that each has its own specialization in a single computer [6].

It is difficult to relate the computational power of the brain to a computer. However, when making a human compete against a computer at a given task, one can get some insight by comparing results and performance. In 2016, AlphaGo, a computer program by Google Deepmind played 18 times world champion Lee Sedol in a 5-game match in the board game *Go* [7]. The program calculated the next move by a search tree method combined with a deep neural network. It is speculated that the power consumption of the AlphaGo program during this match was $\sim 1.2\,\mathrm{MW}$ [8]. On the other side, the power consumption of a human brain is about $20W$. Although AlphaGo won $4-1$, one can assume that the players were somewhat evenly matched. The fact that the human was able to compete with the machine with a 60000 times smaller power consumption is quite remarkable.

Neuromorphic computing is the concept of engineering either software or hardware with brain like features. The goal is to identify how the brain works, and then implement systems that mimic some functionality of the brain. A famous example of this is artificial neural networks. In recent years there have been development in using spintronic components to realize such systems [9] and the combination of neuromorphic computing and spintronics in *neuropmorphic spintronics* has gained a lot of interest. Spintronics is a field of condensed matter physics that concerns itself with understanding the properties of the intrinsic spin of the electrons, and how it can be used to create new electrical components. Spintronic devices share some characteristics of the brain, such as non-linearity and plasticity [1]. In 2017, Prychynenko et al. suggested using magnetic skyrmions in magnetic films for reservoir computing [10] and in 2021 Yu et al. simulated a physical Hopfield neural network capable of associative memory [1]. This project report will seek to give a theoretical framework for how a Hopfield neural network could be realized in magnetic thin films with a maze structure, and emulate and explain the results of Yu et al.

Symbols and conventions

This paper includes ideas and concepts from a wide range of scientific fields. Most prominent is probability and information theory, machine learning, neuroscience and spintronics. An effort has been made to keep to standard notation, but in some cases this has been impossible. The two symbols S (entropy, spin) and H (Hamiltonian, fields, entropy) is perhaps particularly overloaded. The following list indicates where notation may differ from what is the standard practice in some fields, and hopefully removes confusion where similar symbols are used for quite different quantities. All variables are defined as they are introduced in the text.

- H is the Hamiltonian
- $\boldsymbol{\mathcal{H}}$ is effective fields
- $\boldsymbol{h}$ is applied external magnetic field
- S is electron spin
- $\mathcal{S}$ is entropy
- $\boldsymbol{V}$ is the state of a system in a fully connected neural network, or set of voltages, which in this text is considered to be the same thing
- $\boldsymbol{\sigma}$ is used as the Pauli spin vector in calculating spin transfer torques, and in the rest of the paper it is conductivity

Furthermore the following notation is used

a	Scalar
$\boldsymbol{a}$	Vector
$\boldsymbol{A}$	Matrix
$\overset{\leftrightarrow}{\boldsymbol{a}}$	Tensor
$\mathbf{a}$	Random variable

1 Introduction to ferromagnetic spintronics

This chapter establishes the basics of ferromagnetic spintronics. It gives an introduction to magnetism and spin dynamics. Then the electrical transport theory of magnetic films is discussed with a special focus on magnetoresistance. Finally, a special phenomenon called stripe skyrmions is introduced.

1.1 Magnetism

This section explores the concept of magnetism in terms of electron spin and the free energy of magnetic materials.

1.1.1 Electron spin and magnetization

Electrons has angular momentum that comes about in two forms, orbital and intrinsic. In quantum mechanics these quantities are described as *operators*, rather than vectors. The total angular momentum operator $\tilde{\boldsymbol{J}}$ is the sum of the two contributions

$$\tilde{\boldsymbol{J}} = \tilde{\boldsymbol{L}} + \tilde{\boldsymbol{S}} \tag{1.1}$$

where $\tilde{\boldsymbol{L}}$ and $\tilde{\boldsymbol{S}}$ is the orbital and intrinsic angular momentum operators [11]. One refers to $\tilde{\boldsymbol{S}}$ as simply the *spin* of the electron. As the electron is charged, one can associate a magnetic moment, $\tilde{\boldsymbol{\mathcal{M}}}$, with the angular momentum

$$\tilde{\boldsymbol{\mathcal{M}}} = \gamma_L \tilde{\boldsymbol{L}} + \gamma_S \tilde{\boldsymbol{S}}. \tag{1.2}$$

The terms γ_L and γ_S is the respective gyromagnetic ratios. In general the gyromagnetic ratio of a particle is given as $\gamma \equiv gq/2m$ where g is called the *g-factor*. q and m is the charge and the mass of the particle. For many magnetic materials the contribution from $\tilde{\boldsymbol{L}}$ is quenched [12], and for the electron one can approximate

$$\tilde{\boldsymbol{\mathcal{M}}} \approx \gamma_{\text{eff}} \tilde{\boldsymbol{S}} = g_{\text{eff}} \frac{e}{2m_e} \tilde{\boldsymbol{S}} \tag{1.3}$$

where $g_{\text{eff}} \approx -2$ [13]. The magnetic moment operator is in other words dependent of the spin of the electron. To connect the classical field theory to the quantum theory of spins, one can write the continuous magnetization vector $\boldsymbol{M}(\boldsymbol{r})$ as

$$\boldsymbol{M}(\boldsymbol{r}) = \frac{\langle \sum_{i \in \mathcal{V}_{av}} \tilde{\boldsymbol{\mathcal{M}}}_i \rangle}{\mathcal{V}_{av}(\boldsymbol{r})} \tag{1.4}$$

where $\mathcal{V}_{av}$ is an small averaging volume around the position $\boldsymbol{r}$ and $\tilde{\boldsymbol{S}}_i$ denotes the quantum spin operator. $\boldsymbol{M}(\boldsymbol{r})$ describes the magnetic moment per unit volume at $\boldsymbol{r}$. By dividing with the saturation magnetization, M_s, one obtains the dimensionless vector $\boldsymbol{m}(\boldsymbol{r}) = \boldsymbol{M}(\boldsymbol{r})/M_s$. This is going to be used as the order parameter to describe the physics of the magnets.

1.1.2 Free energy of magnetic materials

The free energy of the system is a functional of the magnetization $F = F[\boldsymbol{m}(\boldsymbol{r})]$, and it is important for the dynamics of $\boldsymbol{m}$. A discrete model of a many spin system has free energy [14]

$$F = -\sum_{\langle ij \rangle} J_{ij} \boldsymbol{m}_i \cdot \boldsymbol{m}_j - \sum_i K_n (m_i \cdot \hat{n})^2 - \sum_{\langle ij \rangle} \boldsymbol{D}_{ij} \cdot (\boldsymbol{m}_i \times \boldsymbol{m}_j) - M_s \mu \sum_i \boldsymbol{m}_i \cdot \boldsymbol{h}_i^{\text{ext}} \tag{1.5}$$

The first terms is called the *direct exchange interaction* and is a consequence of electric interaction and the Pauli exclusion principle. The term J_{ij} parameterize this interaction. If $J_{ij} > 0$ then this term will strive to align spins i and j. This is the trademark of a ferromagnetic structure. Note that the summing index $\langle ij \rangle$ indicates that the sum is over nearest neighboring spins. The second term is called magnetic anisotropy. This determines an axis $\hat{n}$ which may be favorable or unfavorable direction for $\boldsymbol{m}_i$ to point, depending on the sign of K_n. If $K_n > 0$ then $\boldsymbol{m}_i$ will seek to align with $\hat{n}$. $\hat{n}$ is then called an "easy axis". However, if $K_n < 0$, $\boldsymbol{m}_i$ will rotate away from $\hat{n}$ trying to minimize its free energy by being normal to $\hat{n}$ making it a "hard axis". This effect can be the result of a wide range of causes, among others structural differences, or tension, in the material. The term $\sum_{\langle ij \rangle} \boldsymbol{D}_{ij} \cdot (\boldsymbol{m}_i \times \boldsymbol{m}_j)$

is called the *Dzyaloshinskii-Moriya interaction* (DMI) or antisymmetric exchange. The DMI will try to tilt neighboring spins along the vector $\boldsymbol{D}_{ij}$. This is in competition with the exchange interaction mentioned previously. The forth and final term is called the *Zeeman interaction*. This is a result of the spin coupling to an external magnetic field $\boldsymbol{h}_i^{\text{ext}}$. μ is the magnetic susceptibility of the material. It can also be written $\mu = \mu_0\mu_r$, where μ_0 is the magnetic permeability of vacuum, and μ_r is the relative permeability of the material. The consequence of an external magnetic field is that the spins will tend to align along it.

The same Eq. (1.5) can also be written on a continuous form. The free energy is now expressed as an integral of the free energy density, f, over the sample volume

$$F[\boldsymbol{m}(\boldsymbol{r})] = \int_V f(\boldsymbol{m}, \nabla\boldsymbol{m}, \boldsymbol{r}) d^3r. \tag{1.6}$$

The free energy density can be written similarly as

$$f = A(\nabla\boldsymbol{m})^2 - K_n(\boldsymbol{m}\cdot\hat{n})^2 - D\boldsymbol{m}\cdot(\nabla\times\boldsymbol{m}) - M_s\mu\boldsymbol{m}\cdot\boldsymbol{h}^{\text{ext}} \tag{1.7}$$

where A is a new material constant proportional to J in Eq. (1.5) [2, 13, 15].

1.2 Spin dynamics

This section will consider how localized spins behave as a function of time, based on interaction with other localized spin, external fields, and applied currents.

1.2.1 The Landau-Lifshitz-Gilbert equation

As previously stated, magnetic materials is described by their magnetization $\boldsymbol{m}$. The magnetization will evolve over time and the *Landau-Lifshitz-Gilbert* equation (LLG) determines the dynamics of $\boldsymbol{m}$ [1]

$$\frac{\partial\boldsymbol{m}}{\partial t} = -\gamma_e\boldsymbol{m}\times\mu_0\boldsymbol{\mathcal{H}}_{\text{eff}} + \alpha\boldsymbol{m}\times\frac{\partial\boldsymbol{m}}{\partial t} + \boldsymbol{\tau}. \tag{1.8}$$

Here γ_e is the electron gyromagnetic ratio and $\boldsymbol{\mathcal{H}}_{\text{eff}}$ is an effective magnetic field, given by the functional derivative of the free energy [2]

$$\boldsymbol{\mathcal{H}}_{\text{eff}} = -\frac{1}{\mu_0 M_s}\frac{\delta F[\boldsymbol{m}(\boldsymbol{r},t)]}{\delta\boldsymbol{m}}. \tag{1.9}$$

The first term $-\gamma_e \boldsymbol{m} \times \mu_0 \boldsymbol{\mathcal{H}}_{eff}$ will cause $\boldsymbol{m}$ to precess around $\boldsymbol{\mathcal{H}}_{eff}$. The second term in the LLG Eq. (1.8) is called the *Gilbert damping* and is parameterized by the Gilbert damping constant α. This term will seek to minimize the free energy of the system given in Eq. (1.7) by relaxing $\boldsymbol{m}$ towards $\boldsymbol{\mathcal{H}}_{eff}$. More generally one can state about the first two terms of Eq. (1.8) that $-\gamma_e \boldsymbol{m} \times \mu_0 \boldsymbol{\mathcal{H}}_{eff}$ determines the eigenmode frequency, while the Gilbert damping term determines the eigenmode lifetime [13].

The last term $\boldsymbol{\tau}$ is a spin-transfer torque. The electrons in the current have spins themselves and will interact with the localized electron spins, causing change in $\boldsymbol{m}$. Figure 1.1 illustrates this interaction. There are several different ways a current can interact with the $\boldsymbol{m}$. This paper will focus on the *s-d interaction* and *spin-orbit interaction*, and will be discussed in the following section.

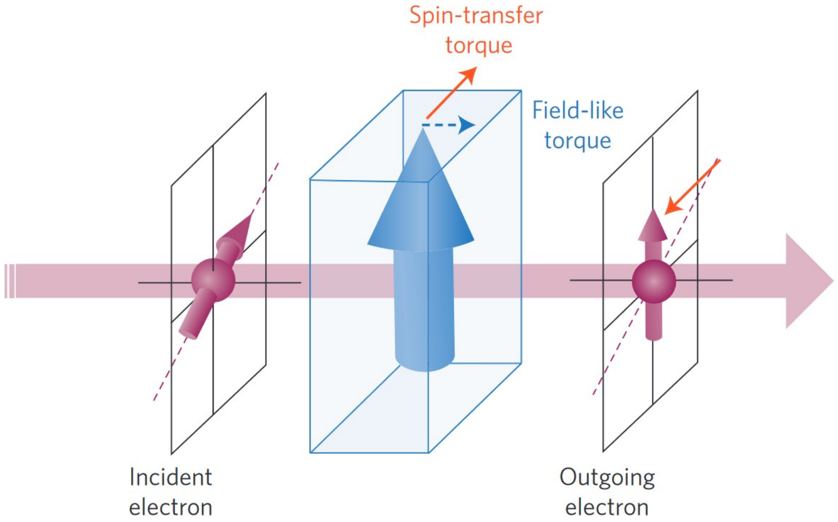


Figure 1.1: An incident free electron from a current has spin not aligned with the spin of the localized electron in the material, illustrated by the blue arrow. As electrons pass by the local $\boldsymbol{m}$ field, the free electrons spin aligns with $\boldsymbol{m}$. By conservation of angular momentum, the torque required to turn the electron spin must be equal and opposite to the torque on the localized electron. This is called the spin-transfer torque. Figure courtesy of [16].

Throughout this paper one assumes that the amplitude of the magnetic moment is constant and finite, i.e. $|\boldsymbol{m}(\boldsymbol{r}, t)| = 1$. The variation in $\boldsymbol{m}$ is solely in the direction of the vector. Taking the time derivative on both sides of

$\boldsymbol{m} \cdot \boldsymbol{m} = 1$ one deduces the following important relation

$$\boldsymbol{m} \cdot \frac{\partial \boldsymbol{m}}{\partial t} = 0. \tag{1.10}$$

For computational reasons it may be preferable to rewrite the LLG Eq. (1.8) so that the derivative is isolated on one side of the equation. By inserting the expression for $\frac{\partial \boldsymbol{m}}{\partial t}$ from Eq. (1.8) back into the derivative on the right hand side, and applying the condition in Eq. (1.10) one can write the transformed LLG equation

$$\frac{\partial \boldsymbol{m}}{\partial t} = \frac{-\gamma_e}{1+\alpha^2}[\boldsymbol{m} \times \boldsymbol{\mathcal{H}}_{eff} + \alpha \boldsymbol{m} \times \boldsymbol{m} \times \boldsymbol{\mathcal{H}}_{eff} + \tilde{\alpha} \boldsymbol{\tau} \times \boldsymbol{m}] + \boldsymbol{\tau} \tag{1.11}$$

where $\tilde{\alpha} = \alpha/\gamma_e$.

1.2.2 s-d and spin orbit interaction and the resulting spin transfer torques

s-d interaction

The s-d interaction gets its name from the interaction between local electrons that constitute the $\boldsymbol{m}$ field, usually in the d-orbital, and incoming free electrons from a current, usually in an s-orbital. The Hamiltonian can be written as [13]

$$H_{\mathrm{sd}} = J_{\mathrm{sd}} \boldsymbol{\sigma} \cdot \boldsymbol{m} \tag{1.12}$$

where $\boldsymbol{\sigma}$ is the Pauli vector describing the incoming current, and J_{sd} is a constant that parametrize this interaction. Consider first an incident current toward a two domain system. A domain is an area in the material where all $\boldsymbol{m}$ point in the same direction. As the current has traversed the first domain, all electrons will have spins aligned with the first domain. As the electrons enters the next domain, the spins will start to turn and align with the new magnetization direction. The change in spin of the current electrons is opposite of the change in the local electrons. This torque is called the *Slonczewski torque* [16]

$$\boldsymbol{\tau}_s = -\frac{\gamma \hbar}{2eM_sV} \boldsymbol{m} \times (\boldsymbol{m} \times \boldsymbol{I}_s). \tag{1.13}$$

where $\boldsymbol{I}_s$ is the spin current. A symmetry argument lead to the possibility of another torque

$$\boldsymbol{\tau}_{s\beta} = -\frac{\gamma\hbar}{2eM_sV}\beta_s\boldsymbol{m}\times\boldsymbol{I}_s. \tag{1.14}$$

The coefficient β_s is a material dependent parameter and determines the strength of $\boldsymbol{\tau}_{s\beta}$ relative to $\boldsymbol{\tau}$.

If the magnetization is more continuously varying we say that the structure has a magnetic texture. Figure 1.2 illustrates this. In this setup, which can be considered the continuous limit of the two domain system, the spin-transfer torque takes another form, and is called the Berger spin-transfer torque

$$\boldsymbol{\tau}_B = \frac{\gamma\hbar}{2eM_s}P(\boldsymbol{j}\cdot\nabla)\boldsymbol{m}. \tag{1.15}$$

Here P is the spin polarization and $\boldsymbol{j}$ is the current density [16].

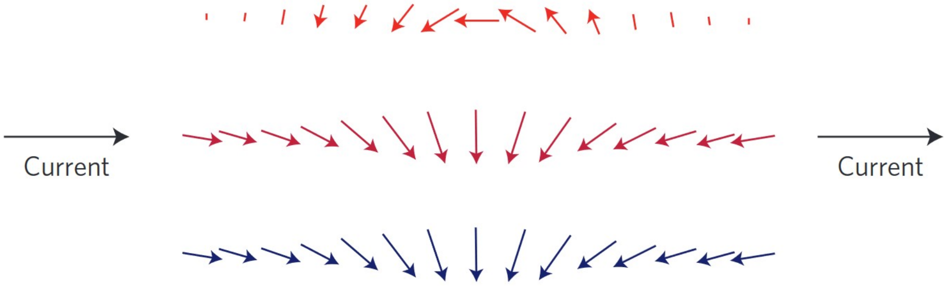


Figure 1.2: Current incident on a ferromagnet with texture indicated by purple arrows. Blue arrows show the spin of the incident electrons that change as they move through the material. Red arrows illustrate the spin-transfer torque. Figure courtesy of [16].

Similarly to $\boldsymbol{\tau}_{s\beta}$ there can exist a torque perpendicular to $\boldsymbol{\tau}_B$ which takes the form

$$\boldsymbol{\tau}_{B\beta} = \frac{\gamma\hbar}{2eM_s}\beta_B P\boldsymbol{m}\times(\boldsymbol{j}\cdot\nabla)\boldsymbol{m}. \tag{1.16}$$

Spin orbit coupling

The next interaction between a current and a ferromagnetic system to be discussed is the spin-orbit interaction. For the hydrogen atom the Hamiltonian for this interaction can in general be written [14]

$$H_{\text{so}} = -\frac{\hbar \boldsymbol{\sigma} \cdot \boldsymbol{p} \times \nabla V}{4m_e^2 c^2}. \tag{1.17}$$

Here $\boldsymbol{p}$ and m_e is the momentum operator and mass of the electron, c is speed of light. V is the Coulomb potential from the interaction between electron and proton. For a bulk material the Hamiltonian becomes more complicated as V should contain all electric potentials. It is possible to split the Hamiltonian in two parts, intrinsic and extrinsic, depending on the origin of the potential. Potentials stemming from the electronic band structure of the material will be intrinsic, while potentials due to impurities, boundaries and external field will be extrinsic. The two Hamiltonians H_{int} and H_{ext} can be written as

$$H_{\text{int}} = -\frac{1}{2}\boldsymbol{\sigma} \cdot \boldsymbol{b}(\boldsymbol{k}) \tag{1.18}$$

$$H_{\text{ext}} = -\lambda \boldsymbol{\sigma} \cdot \boldsymbol{k} \times \nabla V_{\text{ext}} \tag{1.19}$$

where $\boldsymbol{b}(\boldsymbol{k})$ is a spin-orbit field due to the band structure as a function of the wave vector $\boldsymbol{k}$. λ is constant that is proportional to the atomic number Z. A symmetry argument leads to the conclusion that $\boldsymbol{b}(\boldsymbol{k})$ must vanish if there is spacial inversion and time reversal symmetry in the system. The two symmetries can be denoted by operators $\hat{I}$ for spacial inversion, and $\hat{T}$ for time reversal

$$\hat{I} = \begin{cases} \boldsymbol{k} \rightarrow -\boldsymbol{k} \\ \boldsymbol{r} \rightarrow -\boldsymbol{r} \\ \boldsymbol{\sigma} \rightarrow \boldsymbol{\sigma} \end{cases} \qquad \hat{T} = \begin{cases} \boldsymbol{k} \rightarrow -\boldsymbol{k} \\ \boldsymbol{r} \rightarrow \boldsymbol{r} \\ \boldsymbol{\sigma} \rightarrow -\boldsymbol{\sigma} \end{cases} \tag{1.20}$$

The argument goes as follows: if spacial inversion symmetry is present, then $H_{\text{int}} = \hat{I}(H_{\text{int}})$ and similarly for time reversal $H_{\text{int}} = \hat{T}(H_{\text{int}})$. However, if one considers the right hand side of Eq. (1.18) then $\hat{I}(\boldsymbol{\sigma} \cdot \boldsymbol{b}(\boldsymbol{k})) = \boldsymbol{\sigma} \cdot \boldsymbol{b}(\boldsymbol{k})$ only if $\boldsymbol{b}(\boldsymbol{k})$ is a even function of $\boldsymbol{k}$. Furthermore, $\hat{T}(\boldsymbol{\sigma} \cdot \boldsymbol{b}(\boldsymbol{k})) = \boldsymbol{\sigma} \cdot \boldsymbol{b}(\boldsymbol{k})$ only if $\boldsymbol{b}(\boldsymbol{k})$ is an odd function of $\boldsymbol{k}$. This contradiction leads to the conclusion that $\boldsymbol{b}(\boldsymbol{k}) = 0$ unless at least one of the two symmetries is broken.

To illustrate how the spin orbit coupling leads to a torque, we shall consider a two-dimensional system where spacial inversion symmetry is broken along the z-axis. This can for instance be achieved by applying an external field along $\hat{z}$-direction. The resulting Hamiltonian is called the *Rashba Hamiltonian* can be written

$$H_{\text{Rashba}} = -\alpha_R \boldsymbol{\sigma} \cdot (\boldsymbol{p} \times \hat{z}) \tag{1.21}$$

where α_R parameterize this effect [17]. Upon comparison with the qunatum mechanical Hamiltonian for the Zeeman effect, $H_Z \propto \boldsymbol{\sigma} \cdot \boldsymbol{B}$, we see that the terms $(\boldsymbol{p} \times \nabla V)$ and $\boldsymbol{p} \times \hat{z}$ in Eqs. (1.17) and (1.21) can be though of as effective momentum dependent fields.

The Rashba Hamiltonian will result in a torque

$$\boldsymbol{\tau}_{\text{so}} = \boldsymbol{m} \times \boldsymbol{\mathcal{H}}_{\text{Rashba}} \tag{1.22}$$

where

$$\boldsymbol{\mathcal{H}}_{\text{so}} = -\frac{\delta H_{\text{so}}}{\delta \boldsymbol{\sigma}} = \alpha_R(\boldsymbol{p} \times \hat{z}) \tag{1.23}$$

By the same symmetry reasons as before there can also exist a torque

$$\boldsymbol{\tau}_{\text{so}\beta} = \boldsymbol{m} \times \boldsymbol{\tau}_{\text{so}}. \tag{1.24}$$

1.2.3 Calculation of the effective field, $\boldsymbol{\mathcal{H}}_{eff}$

The effective field $\boldsymbol{\mathcal{H}}_{eff}$ remains to be calculated in Eq. (1.9). The functional derivative of the right hand side can be written

$$\frac{\delta F[\boldsymbol{m}(\boldsymbol{r},t)]}{\delta \boldsymbol{m}} = \frac{\partial f(\boldsymbol{m}, \nabla \boldsymbol{m}, \boldsymbol{r}, t)}{\partial \boldsymbol{m}} - \nabla \cdot \frac{\partial f(\boldsymbol{m}, \nabla \boldsymbol{m}, \boldsymbol{r}, t)}{\partial(\nabla \boldsymbol{m})} \tag{1.25}$$

By substituting the expression for f in Eq. (1.7) and doing the derivatives one obtains

$$\boldsymbol{\mathcal{H}}_{eff} = \frac{1}{\mu_0 M_s}\left[2A\nabla^2 \boldsymbol{m} - 2K_n(\boldsymbol{m}\cdot\hat{n}) - 2D\nabla \times \boldsymbol{m}\right] - \mu \boldsymbol{h}^{\text{ext}}. \tag{1.26}$$

Together with the LLG Eq. (1.8) one can now solve the motion of $\boldsymbol{m}$.

1.2.4 Temperature in the LLG equation

The LLG Eq. (1.8) can be expanded to include the effect of temperature in the magnetic system. This is achieved by adding a new field, $\boldsymbol{\mathcal{H}}_{th}$ to the already existing field $\boldsymbol{\mathcal{H}}_{eff}$. The thermal field is not an physical magnetic field, but it encodes the fluctuations that would appear if the temperature is above zero. Due to the stochastic fluctuations of $\boldsymbol{\mathcal{H}}_{eff}$ the new and updated equation is called *stochastic-Landau-Lifshitz-Gilbert* (sLLG) and reads

$$\frac{\partial \boldsymbol{m}}{\partial t} = -\gamma \boldsymbol{m} \times (\boldsymbol{\mathcal{H}}_{eff} + \boldsymbol{\mathcal{H}}_{th}) + \alpha \boldsymbol{m} \times \frac{\partial \boldsymbol{m}}{\partial t} + \boldsymbol{\tau}. \tag{1.27}$$

The first term describes the interaction with an effective magnetic field and a thermal field $\boldsymbol{\mathcal{H}}_{th}$ [18, 19]. The thermal field must meet the following restrictions [20]

$$\langle \boldsymbol{\mathcal{H}}_{th}(\boldsymbol{r}, t) \boldsymbol{\mathcal{H}}_{th}(\boldsymbol{r}, t') \rangle = 2\xi_H \delta(\boldsymbol{r} - \boldsymbol{r'}) \delta(t - t') \tag{1.28}$$

and

$$\langle \boldsymbol{\mathcal{H}}_{th}(\boldsymbol{r}, t) \rangle = 0 \tag{1.29}$$

where $\xi_H = \alpha k_B T / (\gamma \mu_s)$.

1.3 Electrical currents in magnetic materials and anisotropic magnetoresistance

When an electrical current is applied to a ferromagnetic material, it can exhibit a physical phenomenon called *magnetoresistance.* Magnetoresistance occurs when the electrical resistivity of a material ρ is dependent on the magnetization $\boldsymbol{m}$ of the material. The charge current density $\boldsymbol{j}$ as a function of the electrical field $\boldsymbol{E}$ is then given by

$$\boldsymbol{j}(\boldsymbol{r}) = \overset{\leftrightarrow}{\boldsymbol{\sigma}}[\boldsymbol{m}] \boldsymbol{E}(\boldsymbol{r}). \tag{1.30}$$

Here $\overset{\leftrightarrow}{\boldsymbol{\sigma}}[\boldsymbol{m}]$ is the conductivity tensor, dependent on $\boldsymbol{m}$ [10]. In essence this equation is a generalization of Ohm's law. Under the assumption that the relaxation of the current happens much faster than the magnetization, the electrical field is given by $\boldsymbol{E} = \nabla \Phi$, where Φ is the electric potential. An equation for Φ can be found by taking the divergence of Eq. (1.30)

$$\nabla \cdot \boldsymbol{j} = \nabla \cdot (\overset{\leftrightarrow}{\boldsymbol{\sigma}} \boldsymbol{E}). \tag{1.31}$$

Conservation of charge gives $\nabla \cdot \boldsymbol{j} = 0$, and using the notation $\partial_x = \partial / \partial x$ the right hand side can be written

$$\sum_i \partial_i \sum_j \sigma_{ij} E_j = \sum_i \sum_j (E_j \partial_i \sigma_{ij} + \sigma_{ij} \partial_i E_j) = (\nabla \cdot \overset{\leftrightarrow}{\boldsymbol{\sigma}}) \cdot \boldsymbol{E} + \overset{\leftrightarrow}{\boldsymbol{\sigma}} : \nabla \boldsymbol{E} \tag{1.32}$$

where the notation $\boldsymbol{A} : \boldsymbol{B} \equiv \mathrm{Tr}(\boldsymbol{A}\boldsymbol{B}^\dagger)$ and is also known as the *Frobenius inner product* [21]. Substituting $E = \nabla \Phi$ we get the equation for the electric potential

$$(\nabla \cdot \overset{\leftrightarrow}{\boldsymbol{\sigma}}) \cdot \nabla \Phi + \overset{\leftrightarrow}{\boldsymbol{\sigma}} : \nabla (\nabla \Phi) = 0, \quad \Phi^i_{\text{electrode}} = V_i. \tag{1.33}$$

The boundary conditions state that the potential $\Phi^i_{\text{electrode}}$ at electrode i must be equal to the applied voltage V_i for that electrode.

$\overleftrightarrow{\boldsymbol{\sigma}}[\boldsymbol{m}]$ can be found by inverting the resistivity tensor $\overleftrightarrow{\boldsymbol{\rho}}[\boldsymbol{m}]$. We are going to assume that the local resistivity at a given point can be split into two parts, one isotropic part that is constant throughout the material, and one part caused by magnetoresistance. This allows us to write the total local resistivity as [10]

$$\rho_{\text{tot}} = \rho_\perp + (\rho_\parallel - \rho_\perp)\cos^2\theta, \quad \theta = \boldsymbol{m} \cdot \hat{\boldsymbol{j}}. \tag{1.34}$$

Here, $\rho_\perp$ and $\rho_\parallel$ is the resistivity of current flowing perpendicular ($\theta = 90°$) and parallel ($\theta = 0°$) to $\boldsymbol{m}$ respectively. The resistivity tensor can then be written $\overleftrightarrow{\boldsymbol{\rho}}[\boldsymbol{m}] = \rho_\perp \mathbb{I} + (\rho_\parallel - \rho_\perp)\tilde{\boldsymbol{P}}[\boldsymbol{m}]$, where $\tilde{\boldsymbol{P}}[\boldsymbol{m}] = \boldsymbol{m} \otimes \boldsymbol{m}$ is the projection operator. Making use of the relation $\overleftrightarrow{\boldsymbol{\sigma}} = (\overleftrightarrow{\boldsymbol{\rho}})^{-1}$ gives

$$\overleftrightarrow{\boldsymbol{\sigma}}[\boldsymbol{m}] = \sigma_0 \mathbb{I} - \sigma_0 \frac{a}{1+a} \begin{bmatrix} m_x^2 & m_x m_y & m_x m_z \\ m_y m_x & m_y^2 & m_y m_z \\ m_z m_x & m_z m_y & m_z^2 \end{bmatrix} \tag{1.35}$$

where $\sigma_0 = 1/\rho_\parallel$ and $a = (\rho_\parallel - \rho_\perp)/\rho_\perp$[1]. The first term is the isotropic conductivity, while the second term is due to magnetoresistance. The matrix terms can be split up into two separate effects called anisotropic magnetoresistance (AMR) and *planar Hall effect* (PHE). The AMR is encoded in the diagonal terms, while the off diagonal constitute the PHE. The PHE will cause a current density $\boldsymbol{j}_{\text{PHE}} \sim \boldsymbol{m}(\boldsymbol{m} \cdot \boldsymbol{E})$ [23]. However, it is an other effect called the *anomalous Hall effect* (AHE) that is not included in Eq. (1.35). The AHE will cause a current $\boldsymbol{j}_{\text{AHE}} \sim \boldsymbol{m} \times \boldsymbol{E}$. To include this, a conductivity tensor $\overleftrightarrow{\boldsymbol{\sigma}}_{\text{AHE}}$ on the following form should be added to Eq. (1.35)

$$-\sigma_{\text{AHE}} \begin{bmatrix} 0 & m_z & -m_y \\ -m_z & 0 & m_x \\ m_y & -m_x & 0 \end{bmatrix} \tag{1.36}$$

where σ_{AHE} parameterize the AHE.

In the presence of electrical currents, the LLG Eq. (1.11) needs to be solved self consistently with Eq. (1.30), Eq. (1.33) and Eq. (1.35). This is because $\boldsymbol{\tau}$ in the LLG Eq. will rotate $\boldsymbol{m}$ and thus change $\overleftrightarrow{\boldsymbol{\sigma}}[\boldsymbol{m}]$. Subsequently

---

[1] Other definitions σ_0 and a is also used. One convention uses $\sigma_0 = (1/\rho_\parallel + 2/\rho_\perp)/3$ and $a = 2(\rho_\parallel - \rho_\perp)/(\rho_\parallel + \rho_\perp)$ [1, 22].

this changes $\boldsymbol{j}$ and Φ.

The software *Boris Computational Spintronics* is applied for the simulations [2], a slightly different convention is used. The conductivity is defined as

$$\sigma = \frac{\sigma_0}{1 + a\cos^2(\theta)}, \quad \theta = \boldsymbol{m} \cdot \hat{\boldsymbol{j}}. \tag{1.37}$$

This is just then the inverse of Eq. (1.34). With this simplification $\boldsymbol{j}$ will always be parallel with $\boldsymbol{E}$. This preserves the effect of anisotropic magnetoresistance. However, it neglects the Hall effect. This may have an impact on results, but it is argued that the PHE and AHE in maze structures is small [22]. Furthermore, it expedites the solving of Eqs. (1.30) and (1.33) as σ now is a scalar, as opposed to a tensor.

1.4 Stripe skyrmions

The ground state of a magnetic system is decided by the minimum of the free energy. The relative strengths of the different interaction terms determines what configurations that are stable. As previously mentioned, the DMI favors a tilting of the spins. This interaction can stabilize magnetic quasiparticles called skyrmions [24] in the ground state of a system. The skyrmions in are topological stable particles characterized by a continuously changing magnetization. The magnetization in the center of the skyrmion and at the rim points in opposite directions. There are a lot of different types of skyrmions [24]. One type is *stripe skyrmions*. Stripe skyrmions are characterized by a maze like magnetization profile, as depicted in figures 1.3 and 1.4. The qualitative connection between the DMI strength D and the width of the stripes is that the larger D becomes, the thinner the stripes get [25]. Consequently this leads to denser packed stripes in a given area. Figure 1.3 shows an image of $\mathrm{Fe_3Sn_2}$, a material with a large DMI coefficient, where stripe skyrmions appear. Considering figure 1.4 it is interesting to see that the in-plane magnetization, indicated by the colors, on the sides of a domain pointing in the $\hat{z}$ direction, indicated in black and white, always point in the opposite directions. This is a reflection of the continuously changing magnetization that characterizes skyrmions.

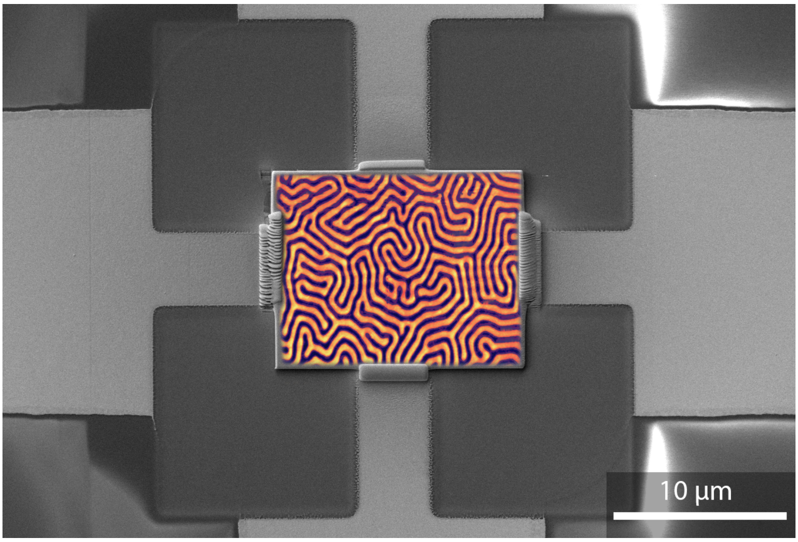


Figure 1.3: Magnetic maze textures of Fe_3Sn_2 captured by magnetic force microscopy. Credit to: Erik Roede & Dennis Meier.

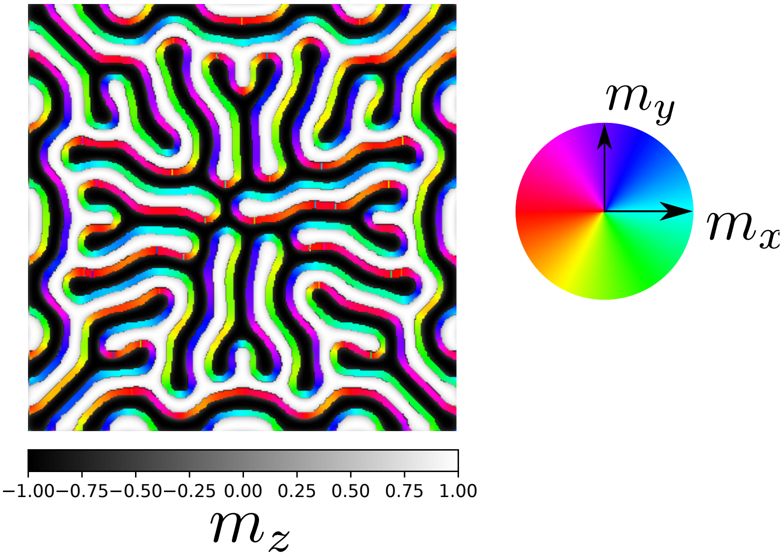


Figure 1.4: Magnetic maze structure or stripe skyrmions. The colors indicate the angle of the magnetization vector $\boldsymbol{m}$ in the xy-plane, together with the z-component of $\boldsymbol{m}$ in gray tones. $\hat{z}$-direction is out of plane. The result is obtained using *Boris Computational Spintronics* [2].

2 Introduction to neural networks

This section will explore the how the human brain processes information, especially how it can associate an input to a stored memory. Furthermore, the brain and a conventional computer will be compared. The three disciplines artificial neural networks, statistical physics and neuroscience will be introduced and connected together through the Hopfield neural network model. The Hopfield model will be investigated in detail, before presenting some simple examples where associative memory problems have been solved, using the prementioned model.

2.1 The neuron model

This section will explore the neuron, and give a simple mathematical description of it. A distinction between how the brain and a computer operates is highlighted.

2.1.1 The neuron anatomy and biological functionality

The neuron is an electrically excitable cell type, and essential to the nervous tissue of almost all animals [26]. It is a fundamental processor of information, and interconnects with other neurons to create vast networks. In the human brain there are several billion neurons [27]. The neuron consist of four main parts: the *soma*, *dendrites*, *the axon* and *the synapse*. Figure 2.1 shows a schematic picture of the components. The soma, or cell body, and takes care of the main biological tasks necessary for a cell to live. The axon carries the electrical signal, and splits into several branches. The dendrites connects to the axons, collecting the signals. In the interconnection between dendrite and axon lies the synapse. The synapse interprets the input from the axon, and sends a output to the dendrites depending on some function that is adaptable for each synapse. A simple model will be discussed in the following to see how this functions.

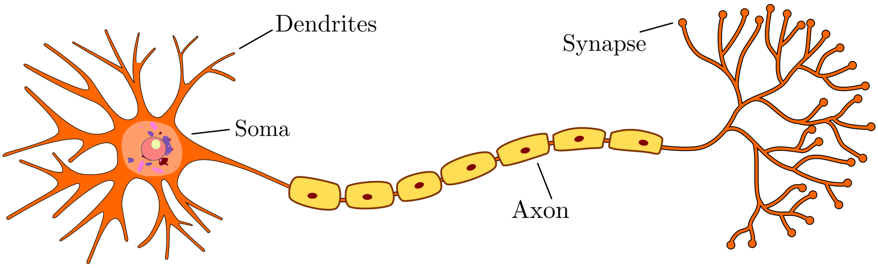


Figure 2.1: Schematic of a single neuron. Figure from [28], adapted to add labels.

2.1.2 Mathematical description of the neurons

The following is based on the works of Geszti in *Physical Models of Neural Networks* [26]. Information that travels through the neurons is primarily coded in the firing frequency of the neuron. The mathematical model to be introduced assumes that information is solely encoded as frequency.

The frequency code requires that the neurons have a interval where they fire evenly. This comes at the cost of processing speed, but makes the architecture more resilient to noise-errors than if information was coded binary in individual pulses. A neuron gets stimulated by other neurons though the dendrites, and is the sum of excitatory and inhibitory signals that decides the output frequency of a neuron.

The time-averaged potential in a neuron z_i is dependent on the input frequencies x_j from all the j-neurons connected to the i-th neuron

$$z_i = z_0 + \sum_j J_{ij} x_j \tag{2.1}$$

where J_{ij} is the synaptic weights that make up the teachable parameters crucial for any neural network. If $J_{ij} < 0$ then the synapse is inhibitory, and if $J_{ij} > 0$ then the synapse is excitatory. z_0 is the average potential in the neuron when no stimulus is present. The potential level z_i is compared to a threshold level $\tilde{U}_i$, and filtered before it decides the output firing frequency y_i

$$y_i = f(z_i - \tilde{U}_i) = f(\sum_j J_{ij} x_j - U_i) \tag{2.2}$$

where $f(z)$ is a non-linear function, that maps its input to a limited interval. This equation models how neurons connect to each other in a simple way. However, it is hard to envision how it could model thought processes where time plays a significant role as there is no time dependence in the equation. Furthermore, it does not account for the event of the output y_i connecting back into x_j and it induce the need for specific output neurons. By letting neuron y_i send information back into x_j, and thus removing the output, one can write the equation of the system as

$$V_i(t+1) = f(\sum_{j=1}^{N} J_{ij} V_j(t) - U_i). \tag{2.3}$$

Here V_i is the state of neuron i, and all neurons comprise to the total state of the network, $\boldsymbol{V}$, which is given by the set of all neurons $\boldsymbol{V} = \{V_i\}$.

The next step in the development in the neuron model is to constrain each neuron to being either firing or silent. The two possible states will be encoded as either $V_i = +1$ for firing and $V_i = -1$ for silent. The neurons is then said to be ***bipolar***. The activation function is taken to be the sign function $\mathrm{sign}(z) = z/|z|$, such that the neuron at a later time also in constrained to be bipolar.

The Eq. (2.3) is deterministic, but can be made probabilistic by choosing an appropriate activation function that maps the input to an interval between 0 and 1, and interpret this as the probability of a neuron firing. To ensure that a large input in the activation function leads to a large probability of firing, and vice versa for small inputs, it is natural to choose an activation function

$$P(V_i) = \frac{e^{-V_i H_i \beta}}{Z}, \quad \text{where} \quad Z = \sum_{V_i=\{1,-1\}} e^{-V_i H_i \beta} \quad \text{and} \quad H_i = \sum_{j=1}^{N} J_{ij} V_j(t) - U_i \tag{2.4}$$

where Z is known as the partition function and ensures normalization of the probability. β is a parameter for the probability distribution, that can be chosen freely. In the limit $\beta \to \infty$ the probability reduces to $1/2$ for $V_i = 1$

regardless of H_i. This is of course also through for $V_i = -1$ as the probabilities $P(V_i = 1)$ and $P(V_i = -1)$ must sum to one. Furthermore in the limit $\beta \to 0$ the model becomes deterministic, and we end up at the starting Eq. (2.3)

$$V_i(t+1) = \text{sign}(\sum_{j=1}^{N} J_{ij} V_j(t) + h_i). \tag{2.5}$$

2.1.3 The Von Neumann bottleneck

In this simple model, it is the synapse that does the multiplication $J_{ij}x_j$ in Eq. (2.2), while the soma does the summation and function transformation. The weight J_{ij} is stored within the synapse, where the calculation takes place. This is a crucial distinction from how a conventional computer operates. In 1945 Von Neumann defined some rules for how a computer could be built, inspired by how the human brain functions [29]. This is today called the *Von Neumann architecture*, and conventional computers is built on many of these principles. Von Neumann identified the main components of a computer, among others memory, M, and a control arithmetic unit, CA. Furthermore he identified that data would have to be transferred from M to CA and back to M for each computation. Conventional computers today still has separated memory and logic units, as opposed to the human brain. This has the consequence that if the Eq. (2.2) where to be computed, it would be necessary to transfer the weights J_{ij} from the memory to the logic unit where the multiplication takes place. Then the result would have to be transferred back to the memory unit where it is stored. This transfer of data sets a cap on computational performance and is called the *Von Neumann bottleneck* [30]. Implementing hardware that is not limited by the Von Neumann bottleneck is a crucial problem in neuromorphic computing. This will be addressed in chapter 3.

2.2 Spin glass and the Ising model

We can draw a strong parallel between the neuron model in the previous section into the world of magnetism. Consider the free energy for a many spin system in Eq. (1.5), and let us neglect the DMI interaction. If we imagine a large enough K_n-value, all spins will align either parallel or anti parallel to the easy axis $\hat{n}$, making the spins bipolar. The system is now fully described along one axis, and allows us to drop the vector notation and just indicate direction as either being positive or negative. If one allows not only nearest neighbor interaction between spins, but also long range interaction the free energy for a single spin is

$$F_i = -m_i(\sum_{ij} J_i j m_j + M_s \mu_s h_i^{\text{ext}} + K_n). \tag{2.6}$$

With only nearest neighbor interaction this is known as the Ising model, and with long range interaction as the Spin glass model [31]. The probability of a spin pointing along $\hat{n}$ in a material at temperature T is given by [32]

$$P(m_i > 0) = \frac{e^{-F_i \beta}}{Z}, \quad \text{where} \quad \beta \equiv \frac{1}{k_B T} \tag{2.7}$$

where k_B is a constant known as the Boltzmann factor. By comparison with Eq. (2.4) we see that the two models are identical. In this model electron spin (or their magnetic moments), effectively act as neurons. This is a important observation. It connects statistical physics with neuroscience. The spin glass model is studied comprehensibly in physics, and the understanding can be carried directly over to the neuron model. The next section will connect artificial neural networks in the context of computer science to the neuron model.

2.3 Artificial neural networks

This section will provide a short introduction to artificial neural networks in the context of computer science and machine learning. Parallels will be draw between the machine learning algorithms and the previously discussed neuron model.

2.3.1 The general machine learning algorithm

Any machine learning algorithm is composed of four main components: a dataset, a *cost function*, an optimization method and a model. The following provides an general introduction to a supervised machine learning algorithm based on [33].

Consider a conditional probability distribution $P_{\text{data}}(\mathbf{y} \mid \mathbf{x})$. Our dataset consists of observations of inputs $\boldsymbol{x}$ and outputs $\boldsymbol{y}$ of the random variables $\mathbf{x}$ and $\mathbf{y}$ governed by P_{data}. The model is set of functions $P_{\text{model}}(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta})$. Here

$\boldsymbol{\theta}$ is a set of parameters that shapes the probability distribution. The goal of machine learning is to learn the true data generating distribution P_{data}. This is achieved by first guessing what kind of set of functions P_{model} should be, and then tweaking the parameters $\boldsymbol{\theta}$ so that it mimics P_{data}.

Let the set of observations be denoted $\{\boldsymbol{x}^{(i)}\}$ for inputs and $\{\boldsymbol{y}^{(i)}\}$ for outputs. Choosing $\boldsymbol{\theta}$ is commonly done through a maximum likelihood principle. This implies choosing the parameters such that the probability of observing the output given the input is maximized. It can be written

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} P_{\text{model}}(\{\boldsymbol{y}^{(i)}\} \mid \{\boldsymbol{x}^{(i)}\}; \boldsymbol{\theta}). \tag{2.8}$$

Assuming that we have $m-$observations in the dataset that are independent and identically distributed allows us to write (2.8) as

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} P_{\text{model}}(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}). \tag{2.9}$$

This maximization problem does not change if one takes the logarithm and divides by the factor m. Doing this makes the problem a bit more convenient, as it can be expressed in terms of minimizing the cross entropy between the empirical probability distribution, $\hat{P}_{\text{data}}$, obtained from the dataset

$$\begin{aligned}\boldsymbol{\theta}_{ML} &= \arg\max_{\boldsymbol{\theta}} \frac{1}{m}\sum_{i=1}^{m} \ln P_{\text{model}}(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{y},\mathbf{x}\sim\hat{P}_{\text{data}}} \ln P_{\text{model}}(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta}) \\ &= \arg\min_{\boldsymbol{\theta}} \mathcal{S}(\hat{P}_{\text{data}}, P_{\text{model}}).\end{aligned} \tag{2.10}$$

The notation $\mathbb{E}_{\mathrm{x}\sim P}[f(x)]$ can be read as the expectation value of $f(x)$ with respect to $P(x)$. The cross entropy between two probability distributions $P(x)$ and $Q(x)$ is defined to be $\mathcal{S}(P,Q) \equiv -\mathbb{E}_{\mathrm{x}\sim P} \ln Q(x)$.

The cost function, typically denoted $J(\boldsymbol{\theta})$, could simply be the cross entropy, $J(\boldsymbol{\theta}) = \mathcal{S}(\hat{P}_{\text{data}}, P_{\text{model}})$. However, it is possible and common to add terms to this, for instance regularization terms that decrease the variance of P_{model}.

2.3.2 Feedforward neural networks

When learning nonlinear functions it is difficult to guess what family of functions P_{model} should be. The *feedforward neural network* (FNN) provides a solution to this problem. The feedforward neural network generally consists of 3 types of layers of neurons, the input layer, hidden layers and output layers. Connections are drawn between each neuron in each layer. Figure 2.2 provides an overview of the network layout. When describing the FNN it is preferable to adopt a vector notation by letting each layer be described as $\boldsymbol{x} = \{x_i\}$, where $\{x_i\}$ is the set of all neurons in the layer. Doing this for hidden and output layers makes equations more compact. The machine learning as described in the previous subsection takes place by feeding the input data $\boldsymbol{x}$ into the input layer. The information flow from input to output is illustrated by figure 2.3.

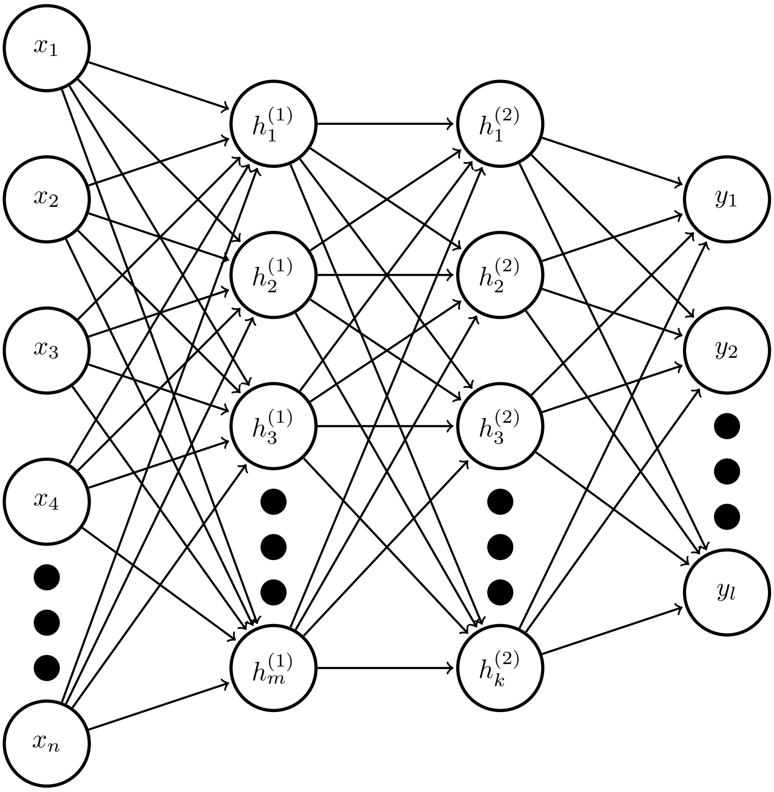


Figure 2.2: Neural network with two hidden layers, and arbitrary depth in each layer.

The neurons in the first hidden layer is calculated from the input by a weighted sum of all neurons from the input layer in addition to some input independent bias, and then transformed by a nonlinear transformation called

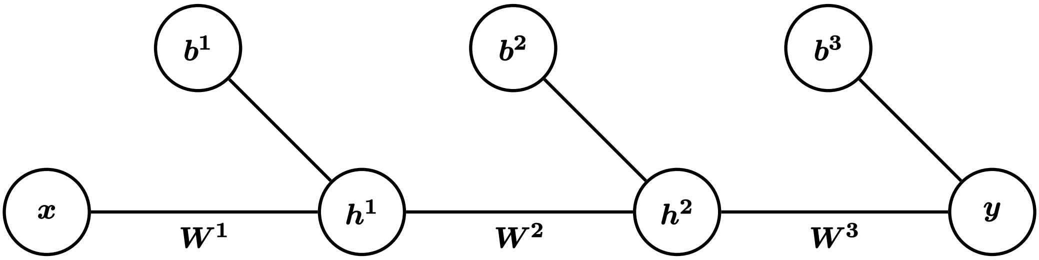


Figure 2.3: A general schematic of a feedforward neural network with vector notation. Biases $\boldsymbol{b}$ and weight matrices $\boldsymbol{W}$ indicated between each layer.

the activation function f^1

$$\boldsymbol{h^1} = f^1(\boldsymbol{W^1 x} + \boldsymbol{b^1}), \tag{2.11}$$

where $\boldsymbol{b^1}$ is the biases, and $\boldsymbol{W^1}$ is the weight matrix connecting $\boldsymbol{x}$ to $\boldsymbol{h^1}$. The next layer of neurons, $\boldsymbol{h^2}$ is then computed based on the results of Eq. (2.11), in a similar way. This procedure is reiterated all the way through to the output layer

$$\boldsymbol{y} = f^3(\boldsymbol{W^3 h^2} + \boldsymbol{b^3}). \tag{2.12}$$

A typical example of an activation function is the *sigmoid function*

$$\sigma(x) = \frac{1}{1+\exp(-x)}. \tag{2.13}$$

Common for activation functions is that they map the output to a given interval, setting a limit on how large, either negative og positive, the state of a neuron can be. The sigmoid function maps all real numbers to an interval between 0 and 1. It slightly resembles the sign-function that maps all real numbers to either -1 or 1. To have such non linear activation functions is essential for machine learning with artificial neural networks.

The machine learning as described in the previous section takes place by initiating the network with a randomly selected weights and biases. The network architecture consist of the number of hidden layers and their depth, and is together with activation functions chosen before the learning procedure starts. Then $\boldsymbol{y}_{\text{model}}$ is calculated based on the input $\boldsymbol{x}_{\text{data}}$. Then the cross entropy between $\boldsymbol{y}_{\text{model}}$ and the empirically observed $\boldsymbol{y}_{\text{data}}$ is then minimized by tweaking the parameters $\boldsymbol{\theta}$ which in this case consists of both $\boldsymbol{W}$ and $\boldsymbol{b}$. The selection of activation functions and network architecture might seem like choosing the family of functions P_{model} which was the problem we were trying to avoid. However, this is not the case as stated in the *universal approximation theorem* [34, 35]. The theorems interpretation is that a FNN with at least one hidden layer with enough units and an activation function like the sigmoid function, can represent any Borel measurable function [33] from one finite dimensional space to another. A definition of a Borel measurable function is given in [34], but it suffices to say that any continuous function on a closed and bounded subset of $\mathbb{R}^n$ is Borel measurable, where $\mathbb{R}^n$ is the n dimensional set of real numbers. The term represent means in this case approximate with any desired nonzero amount of error. Here within lays the true power of the FNN.

Compare the FNN with the time independent neuron model in Eq. (2.2). It is clear that the Eq. indeed is a FNN without hidden layers. Of course a extension of the neuron model to feature hidden layers is not far-fetched. However, the the time dependent neuron model in Eq. (2.3) can not be described as a FNN. There is information flowing both ways through layers, and no output units. Another type of neural networks is needed to describe this model.

2.3.3 Fully connected recurrent neural networks

By letting a layer in a FNN connect back into a predecessing layer one has created a *recurrent neural network* (RNN). Suppose that as previously mentioned the FNN can describe a data generating probability distribution $P(\mathbf{y}|\mathbf{x})$. By connecting all neurons together in an FNN one gets a fully connected RNN called a *Hopfield neural network*. The Hopfield neural network describes a probability distribution $P(\mathbf{x})$. As it is no output data $\boldsymbol{y}$ in this problem, this is an example of unsupervised learning. The goal is to deduce what relations between the components off $\boldsymbol{x}$ translates to probable or improbable configurations of $\boldsymbol{x}$. Note that the graph connecting neurons is now undirected, as information is free to flow both ways between two neurons, as opposed to the directed graph for the FNN, where information flows in only one direction through layers. The requirement that each possible configuration of $\boldsymbol{x}$ should have a nonzero probability can be enforced by using an *energy-based model*

$$P_{\text{model}}(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{Z}\exp(-E(\boldsymbol{x};\boldsymbol{\theta})) \tag{2.14}$$

where Z is called the partition function, ensuring that the probability is normalized. The connection between neurons decides how E may look. In a network described by a fully connected graph, E should be dependent on x_i and the connections between neurons $x_i x_j$. This is the case that will be considered in the following. The energy function E is given as

$$E(\boldsymbol{x}) = -\boldsymbol{x}^\top \boldsymbol{W} \boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x} \tag{2.15}$$

where $\boldsymbol{W}$ is a matrix that describes the connections between neurons, and $\boldsymbol{b}$ is a set of biases for the neurons. Togheter they make up the parameters $\boldsymbol{\theta}$ that needs to be fund. Finding the weights $\boldsymbol{W}$ and $\boldsymbol{b}$ is done by a maximum likelihood approach similar to that for the FNN. Consider a set of observed data $\{\boldsymbol{x}^{(i)}\}$ from a data generating distribution P_{data} that we wish to find. The maximum likelihood estimate for the parameters is given by

$$\begin{aligned}\boldsymbol{\theta}_{ML} &= \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} P(\boldsymbol{x}^{(i)} | \boldsymbol{x}^{(j)}; \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathrm{x}_i | \mathbf{x} \sim \hat{P}_{\text{data}}} \ln P_{\text{model}}(x_i \mid \boldsymbol{x}; \boldsymbol{\theta}) \\ &= \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\mathrm{x}_i | \mathbf{x} \sim \hat{P}_{\text{data}}} (E(\boldsymbol{x}) + Z)\end{aligned} \tag{2.16}$$

This can be interpreted as choosing the parameters $\boldsymbol{\theta}$ to minimize $E(\boldsymbol{x})$ with the constraint Z using the observed data.

Drawing samples from P_{model} might be difficult when the number of neurons in $\boldsymbol{x}$ grow large. *Gibbs sampling* is an example of a *Markov Chain Monte Carlo* algorithm for sampling probability distributions. The algorithm is initiated by setting all $\{x_i\}$ to random values, which is the state of the network at $t = 0$. Then a random neuron x_i is selected an calculated based on the conditional probability of all other neurons that connect to the selected one

$$x_i(t+1) \sim P_{\text{model}}(x_i | \boldsymbol{x}(t)). \tag{2.17}$$

The new state $\boldsymbol{x}(t+1)$ is now all neurons as they where at $t = 0$ with the exception of the updated x_i. Then another neuron x_j is selected and its value is updated as

$$x_j(t+2) \sim P_{\text{model}}(x_j | \boldsymbol{x}(t+1)) \tag{2.18}$$

Repeating this process over many time steps t will yield a correctly sampled state from the original distribution P_{model}. By comparing the energy based model for the probability distribution, and the Gibbs sampling method with Eq. (2.5) we see that the two models are almost equivalent. This is another important point. It connects the neuron model to artificial neural networks. As have already been discussed, it exists a connection between statistical physics and the neuron model. Now we can add artificial neural networks to this.

2.4 Hebbian Learning

This section will deal with how memories can be stored in the context of the neuron model based on [26]. Section 2.1 explained how the neuron firing pattern changes in time. However, it is possible to attain a firing patterns which is stable in time. A stable pattern will be written as $\boldsymbol{V}(t) = \boldsymbol{\xi}$. This pattern will then not evolve any further, i.e. $\boldsymbol{V}(t+1) = \boldsymbol{\xi}$, when considering the deterministic evolution equation (2.5). The vector $\boldsymbol{\xi}$ is then considered an attractor in the system and any input that leads to this configuration of the system is called the corresponding *basin of attraction.* There can be more than one stable pattern in such a neural network. Different basins may lead to different attractors. An attractor is considered stable, as small perturbations from $\boldsymbol{\xi}$ may just lead the system back into the steady state. Attractors is essential for associative memory, as information is coded in the firing pattern. The brain accomplishes this by changing the weights J_{ij} between neurons. When an external pattern is subject to memorization, the weights J_{ij} is changed such that the pattern remains stable after the input is removed. *Hebb's rule* determines how the weights should be updated for a pattern $\boldsymbol{\xi}$ to be memorized

$$J_{ij} \rightarrow J_{ij} + \lambda \xi_i \xi_j \tag{2.19}$$

where λ is a parameter that determines the speed of memorization. This is of course an extreme simplification of how the brain actually function, but it is a rule that works when modeling memory in connective dynamical networks. This is a remarkable simple rule as the weights between two neurons is only dependent on the state of each of the two neurons. A network can store several patterns but careful consideration needs to be taking into the changing of J_{ij} as learning one pattern must not delete previous learned patterns. In a general network this is only possible with a more complicated updating algorithm than Hebb's rule. For instance is the learning rule for a FNN dependent on the state of all other neurons. However, there exists one type of model where Hebb's rule accomplishes this. The next section will provide some insight into a model that can learn memories by the simple Hebb's rule, and what happens when the weights J_{ij} becomes overloaded with memories.

2.5 The Hopfield model

This section is based on [26]. In 1982 Hopfield theorized that compering the mechanizes of associative memory to the Ising model could give considerable insight, although this is a simplification of both the brain and magnetic systems. The Hopfield model is defined in the following:

Serial dynamics Neurons should be updated according to Eq. (2.4). Instead of updating all spins from t to $t+1$ at the same time, which is called *parallel dynamics*, each spin should be updated one at a time. This is called *serial dynamics*.

Hebb's rule Generate p different firing patterns $\{\boldsymbol{\xi}^p\}$ and fix the weights J_{ij} according to Hebb's rule (2.19).

By using Hebb's rule with zero initial weights one obtains

$$J_{ij} = \lambda \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu \quad (i \neq j). \tag{2.20}$$

It is easy to see that the weights now become symmetric: $J_{ij} = J_{ji}$. To gain some insight into how many memories a Hopfield neural network can store, the self-coupling terms will be set to zero, $J_{ii} = 0$ and the networks energy function can be written

$$E(\boldsymbol{V}) = -\sum_{\substack{i,j \\ i \neq j}} J_{ij} V_i V_j - \sum_i h_i^{ext} S_i. \tag{2.21}$$

This energy equation defines a hyperplane over the configuration space containing all possible states $\boldsymbol{V}$ can be in. In the $T = 0$ regime modeled by Eq. (2.5), this energy equation will monotonously decrease. Minima of the function will act as attractors that the spin system will evolve towards. The stored patterns will be encoded as the minima energy function. All the starting configurations that decrease monotonously towards the minima will be the corresponding basin of attraction. Figure 2.4 illustrates the energy function of the Hopfield network after being subject to Hebbian learning.

There are two types of noise in this model:

Temperature noise A consequence of the probabilistic evolution equation, parameterized by β.

Synaptic noise This noise is related to the weights J_{ij} not being to encode

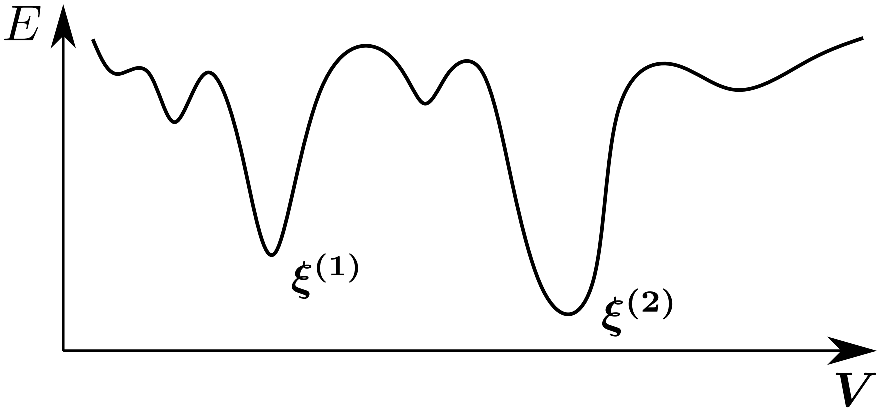


Figure 2.4: Energy function of the Hopfield model after Hebbian learning. The two memories $\boldsymbol{\xi}^{(1)}$ and $\boldsymbol{\xi}^{(2)}$ is encoded as a valley in the energy function. The smaller local minima is spurious memories, a result of synaptic noise, or overlap between the two encoded memories. To minimize the chance of the system retrieving a spurious memory, a fictitious temperature T might allow for the possibility for $\boldsymbol{V}$ to travel to states with higher energy with a small probability, and thus climbing out of local minima. Selecting the fictitious temperature correctly, or slowly letting $T \to 0$ gives a high probability of the system $\boldsymbol{V}$ settling in a memory state.

all $\{\boldsymbol{\xi}^p\}$ without any overlap. The accuracy of a retrieval pattern is influenced by the other stored patterns.

To analyze the noise in the system, it is convenient to define the overlap, m_μ, between the system configuration $\boldsymbol{V}$ and a stored pattern $\boldsymbol{\xi}^\mu$ as

$$m_\mu = \frac{1}{N}\sum_{i=1}^{N} V_i \xi_i^\mu. \tag{2.22}$$

For a system with $h^{ext} = 0$ then the energy function can be written

$$E = -\frac{\lambda}{2}N^2\sum_{\mu=1}^{p}(m_\mu)^2 + \frac{\lambda}{2}Np. \tag{2.23}$$

If a pattern $\boldsymbol{\xi}^\nu$ is (almost) perfectly retrieved, i.e. $\boldsymbol{V} \to \boldsymbol{\xi}^\nu$, then the overlap $(m_\nu)^2 = \mathcal{O}(1)$. To make the analysis easier, the following constraint is demanded from the stored patterns $\{\boldsymbol{\xi}^p\}$

$$q_{\mu\nu} = \langle \boldsymbol{\xi}^\mu \boldsymbol{\xi}^\nu \rangle = \frac{1}{N}\sum_{i=1}^{N} \xi_i^\mu \xi_i^\nu \to 0 \quad \text{as} \quad N \to \infty. \tag{2.24}$$

In other words, the overlap between memories should vanish for large N. This is the case if memories consist of random states ξ_i, and we shall consider this to be the case.

To further investigate the effect of synaptic noise, we are going to consider the deterministic regime, where $T = 0$, i.e. there is no noise present due to temperature. This is modeled by Eq. (2.5). An errorless retrieval of a stored pattern $\boldsymbol{\xi}^\nu$ is possible and achieved if $\xi_i^\nu \sum_{j\neq i} J_{ij}\xi_j^\nu \geq 0$ for all i.

$$\Delta_i^\nu \equiv \frac{\xi_i^\nu \sum_{j\neq i} J_{ij}\xi_i^\nu}{\sqrt{\sum_{j\neq i} J_{ij}^2}} \tag{2.25}$$

is called the stability of pattern $\boldsymbol{\xi}^\nu$ at the i-th neuron. For retrieval of the memory without errors, one should have $\Delta_i^\nu > 0$ everywhere. It is possible to give an upper bound of how many patterns one can store while this remains the case, as long as the set of stored patterns $\{\boldsymbol{\xi}^p\}$ consists of random spins. By approximating the denominator in Eq. (2.25) by its mean, $\hat{E}(\sqrt{\sum_{j\neq i} J_{ij}^2}) = \lambda\sqrt{p(N-1)}$, and using the definition of J_{ij} from Eq. (2.20) we can write

$$\begin{aligned}\Delta_i^\nu &= \frac{\xi_i^\nu \sum_{j\neq i} \lambda \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu \xi_j^\nu}{\lambda\sqrt{p(N-1)}} \\ &= \frac{\xi_i^\nu \sum_{j\neq i} \xi_i^\nu \xi_j^\nu \xi_j^\nu}{\sqrt{p(N-1)}} + \frac{\xi_i^\nu \sum_{j\neq i} \sum_{\mu\neq\nu}^{p} \xi_i^\mu \xi_j^\mu \xi_j^\nu}{\sqrt{p(N-1)}} \\ &= \sqrt{\frac{N-1}{p}} + \frac{\xi_i^\nu \sum_{j\neq i} \sum_{\mu\neq\nu}^{p} \xi_i^\mu \xi_j^\mu \xi_j^\nu}{\sqrt{p(N-1)}}.\end{aligned} \tag{2.26}$$

The remaining sum is the sum of random spins ±1. For N and p large, this becomes a Gaussian distributed variable with zero mean and standard deviation $\sqrt{(N-1)(p-1)}$, such that the probability distribution of Δ_i^ν becomes $\mathcal{P}(\Delta_i^\nu) = \mathcal{N}[\sqrt{\frac{N-1}{p}}, (N-1)(p-1)]$. The probability of $\Delta_i^\nu < 0$ is then

$$P(\Delta_i^\nu < 0) = \int_{-\infty}^{0} \mathcal{P}(\Delta_i^\nu) d\Delta_i^\nu = \frac{1}{2}\mathrm{erfc}\sqrt{\frac{N}{2p}} \tag{2.27}$$

where $\mathrm{erfc}(x)$ is the error function. Perfect retrieval is expected when $\frac{N}{p} \gg 1$. In this limit, the probability becomes

$$\varepsilon \equiv P(\Delta_i^\nu < 0)_{\frac{N}{p}\gg 1} = \sqrt{\frac{p}{2\pi N}} e^{-N/2p} \tag{2.28}$$

and the i-th neuron is retrieved correctly. The probability for the signal as a whole to be retrieved perfectly, P_0 becomes

$$P_0 = (1 - \varepsilon)^N. \tag{2.29}$$

From this it is possible to obtain an upper bound for p that ensures perfect retrieval

$$p \leq \frac{N}{2\ln(N)}. \tag{2.30}$$

This result is valid for when $\frac{N}{p} \gg 1$. However, for smaller fractions, the calculation becomes difficult. Experiments show that the network has a sharp breakdown when $\frac{N}{p} = 0.14$. This limit exhaust the synaptic weights, destroying the function of associative memory in the network.

2.6 Maximum entropy distribution

We shall see that the Ising model can be derived from a maximum entropy principle. The idea is that the probability distribution that describes the network is the one that maximizes the entropy of the network. A general derivation of the maximum entropy distribution is given in the following, based on [36].

We define the entropy of a system $\mathcal{S}[P(x)]$ to be

$$\mathcal{S}[P(x)] \equiv -\sum_x P(x) \ln P(x). \tag{2.31}$$

Here $P(x)$ is the probability distribution of the system that we seek to find. If the system subject to constraints, where the constraints can be written as

$$\sum_x f_j(x) P(x) = a_j \tag{2.32}$$

where a_j is the constraint. Using the method of *Lagrange Multipliers* we can maximize $\mathcal{S}[P(x)]$ under the constraints by introducing the functional

$$\tilde{\mathcal{S}}[P(x)] = -\sum_x P(x) \ln P(x) - \sum_j \lambda_j (\sum_x f_j(x) P(x) - a_j). \tag{2.33}$$

Now taking the functional derivative we obtain

$$\frac{\delta \tilde{\mathcal{S}}}{\delta P(x)} = -\ln P(x) - 1 - \sum_j \lambda_j f_j(x) = 0 \tag{2.34}$$

Solving for $P(x)$, and ensuring that the probability is normalized by dividing with the constant Z gives the maximum entropy distribution

$$P(x) = \frac{1}{Z}\exp(-\sum_j \lambda_j f_j(x)), \quad \text{where} \quad Z = \sum_x \exp(-\sum_j \lambda_j f_j(x)). \tag{2.35}$$

This is a generalization of the Boltzmann distribution, and one can obtain it by letting $\lambda = 1/k_B T$ and $f_j(x)$ be the Hamiltonian of the system [32].

For the Hopfield and spin glass model we denote the system as a set of neurons that either fires or is silent, or spins that is either up or down, $\boldsymbol{V} = \{V_i\}$. There are constraints on each individual spin V_i and the correlation between two spins $V_i V_j$. Making the transitions $x \rightarrow \boldsymbol{V}$, we have the following constraints

$$\langle V_i \rangle_{\text{model}} \equiv \sum_{\boldsymbol{V}} V_i P(\boldsymbol{V}) = \langle V_i \rangle_{\text{data}} \quad \text{and} \quad \langle V_i V_j \rangle_{\text{model}} \equiv \sum_{\boldsymbol{V}} V_i V_j P(\boldsymbol{V}) = \langle V_i V_j \rangle_{\text{data}}. \tag{2.36}$$

The constraints demands that the model probability distribution must have the same expectation value and variance as the observed data. All higher moments are not constrained. Fulfilling these constraints leads to selecting the appropriate multipliers h_i and J_{ij}

$$P(\boldsymbol{V}) = \frac{1}{Z}\exp(\sum_i h_i V_i + \sum_{i \neq j} J_{ij} V_i V_j) = \frac{1}{Z}\exp(\boldsymbol{h}^\top \boldsymbol{V} + \boldsymbol{V}^\top \boldsymbol{J} \boldsymbol{V}). \tag{2.37}$$

This is the probability distribution for the spin glass and Hopfield models.

2.7 A Hopfield model example for associative memory

Two simple examples will give further insight to the Hopfield model and its use. For this, one further simplification is made to the model. Another simple example highlights how the energy surface E behaves when memories are stored in the Hopfield network. A further simplification to the Hopfield model will be done by allowing self connections $J_{ii} \neq 0$ but turning of the field h_i. The Hebbian learning is then given by Eq. (2.19) without the restriction $i \neq j$. This ensures that $J_{ij} \geq 0$. as $J_{ii} = \lambda \sum_{\mu=1}^{p} (\xi_i^\mu)^2$. A weight J_{ij} says something about the desire of neuron i and j has to be equal to each other. If

$J_{ij} > 0$ then neuron i and j will strive towards being in the same state, while the opposite is true if $J_{ij} < 0$. J_{ii} then says something about the stability of each neuron or the desire to be equal to its previous state. The energy is then given by

$$E(\boldsymbol{x}) = -\boldsymbol{x}^\top \boldsymbol{J}\boldsymbol{x} \tag{2.38}$$

where $\boldsymbol{x}$ again is a bipolar pattern describing the states of the neurons in the network, and $\boldsymbol{J}$ is that the system becomes invariant for a sign flip off all states x_i. In other words $\boldsymbol{x}$ and $-\boldsymbol{x}$ describes the same states, as is seen by Eq. (2.38). $\boldsymbol{J}$ only encodes relations between neurons in $\{x_i\}$ and not anything about their sign. Figure 2.5 illustrates a fully connected Hopfield network with symmetric weights.

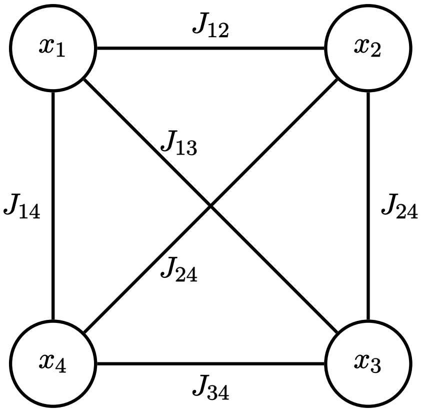
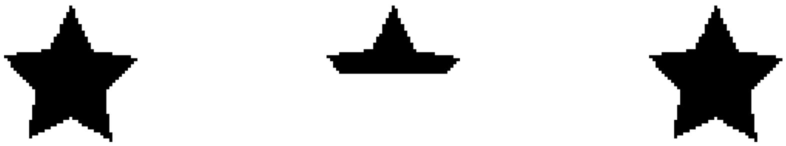


Figure 2.5: Simple schematic of a 4 neuron Hopfield neural network.

To illustrate the usage of a Hopfield neural network for associative memory, a simple model has been implemented. 3600 fully connected neurons was initiated and pictures 2.6a and 2.6d was simultaneously stored in the network according to the Hebbian learning rule. Figure 2.6 indicates that the stored patterns indeed are attractors in the system, as both a noise image and a image where half is missing, both converges to the correct memory. We can thus also conclude that each guess is in the corresponding basin of attraction of the memories. Although only two patterns are stored in this network, it could hold many more. For the worst possible scenario where all memories are orthogonal, this network could store up to $3600 \cdot 0.14 = 504$ patterns. It is worth noting that the computational demanding part is not choosing the weights to store the patterns, but rather deducing the output from an input. This is opposite of what one might suspect from working with feedforward neural networks.

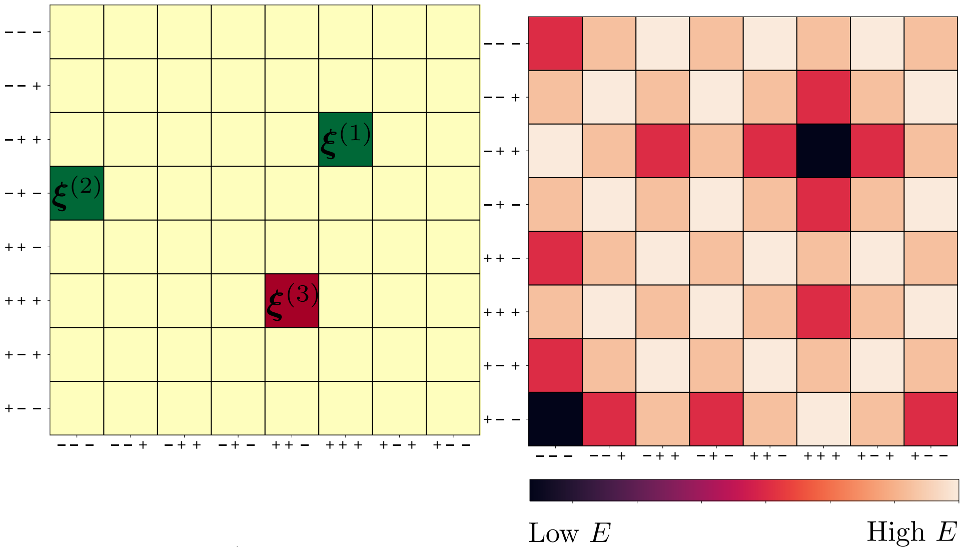


(a) Train 1 **(b)** Input 1 **(c)** Output 1

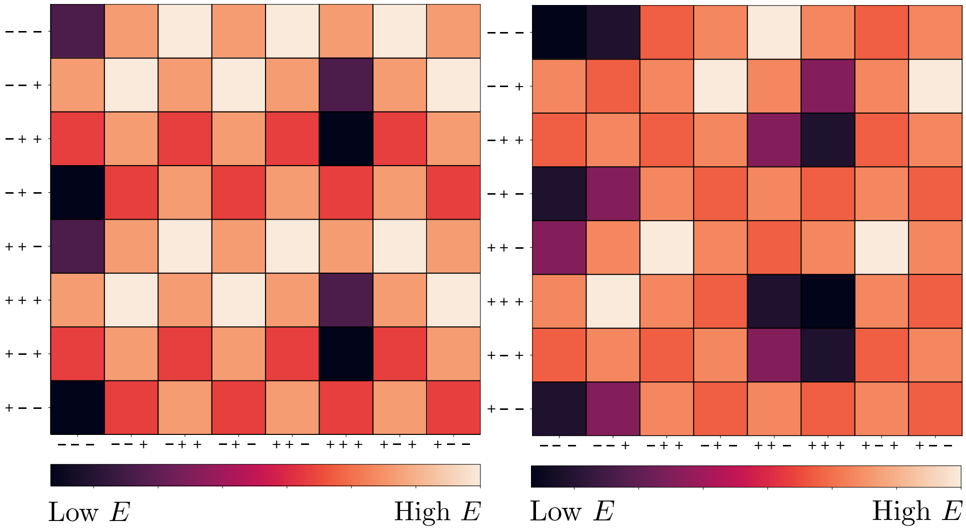
(d) Train 2 **(e)** Input 2 **(f)** Output 2

Figure 2.6: The results of a simple Hopfield network for associative memory. **(a)** and **(d)** are stored simultaneously in the network by Hebbian learning. **(b)** an initial input where half the memory is lost and **(e)** an initial input of a stored memory with noise. **(c)** and **(f)** is the corresponding outputs of inputs **(b)** and **(e)**.

Another simulation of a 6 neuron Hopfield network provides some intuition that Hebbian learning has on the energy function. Figure 2.7 shows the energy surface over the an 6-dimensional binary configuration space, projected down onto a 2-dimensional surface. The resulting 8×8 matrix gives all the $2^6 = 64$ possible configurations a 6-bit vector can be in. The state of a vector is denoted by a sequence of $+$ and $-$, equivalent of each node in the network being either spin up or down. The rows of the matrix indicates the first 3 bits while the column represents the last 3. Figure 2.7a shows the different memories that the network will try to store, numbered by the order that they are added to the network. The first pattern to be learned is $-+++++$. 2.7b shows that Hebbian learning indeed result in the network having a global energy minima for the desired pattern $\boldsymbol{\xi}^{(1)}$. However it is also another global minima with the same energy for configuration $+-----$. This is the mirror image of the the learned pattern. This is of course due to the fact that Eq. (2.38) is a quadratic function such that $E(-\boldsymbol{x}) = E(\boldsymbol{x})$. This gives an energy function that is symmetric. The inferring process can



(a) Inputs $(\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}, \boldsymbol{\xi}^{(3)})$ **(b)** One input $(\boldsymbol{\xi}^{(1)})$



(c) Two inputs $(\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)})$ **(d)** Three inputs $(\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}, \boldsymbol{\xi}^{(3)})$

Figure 2.7: The energy function of a 6 neuron Hopfield network. Patterns $\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}, \boldsymbol{\xi}^{(3)}$ is added one at the time, and the energy is calculated over the total configuration space. The network has a breakdown and can not store all 3 memories at the same time.

be thought of as traversing the energy configuration where one only jumps to a neighboring configuration if this results in a decrease in E. In this sense, state vectors $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ is neighbors if they only differs in one bit. As an example consider the configuration in the top right corner, $---+--$. The states that is located directly to the left and beneath is indeed only one bit-flip away. However, the total number of neighbors must be 6. The matrix can be though of to be periodic such that the top left, and bottom right corners is also neighbors. The final two neighbors is $---++-$ and $-+-+--$. These two are not directly touching the top right corner in 2.7b but this is the loss of doing the projection from 6 dimensional space down to 2 dimensions. One might suspect that $-++-++$ is a local minima of E and thus a spurious memory. However, this is not the case as indeed it is a neighbor of the global minima $-+++++$. Figure 2.7c shows E when both the first and second pattern is stored. There are only global minima at the stored configurations, and their mirrors. Finally the network breaks down when the third pattern is added in figure 2.7d. E now has only 4 minima as opposed to the desired 6. Furthermore, 2 of these minima are for the wrong configurations. The mirror image of $\boldsymbol{\xi}^{(2)}$ interferes with $\boldsymbol{\xi}^{(3)}$ (and vice versa) giving global minima at $------$ and $++++++$.

A final note on the Hopfield model is that it is relevant in solving other problems than associative memory. It is shown that it can give ”good enough” solutions to some NP hard problems [31], such as the traveling salesman problem. Furthermore, in 2021, Ramsauer et al. [37] introduced a continuous Hopfield neural network that delivered state-of-the art benchmarks on some classification task. We now turn our attention to simulating a physical realization of a Hopfield neural network capable of associative memory.

3 A Hopfield neural network with natural learning

This section will explore how one can realize a physical Hopfield neural network by applying voltages to films with magnetic maze texture. The discussion will be based on the paper *Hopfield neural network in magnetic textures with intrinsic Hebbian learning* [1]. The purpose is to show a proof of concept; that one can build an artificial neural network (ANN) capable of associative memory using magnetic films and concepts from spintronics.

3.1 Magnetic films for natural learning

The main idea of the paper is that the non-linearity and plasticity of magnetic films with magnetization texture can be used to store and learn the weights of the network. The magnetic films is modeled by the LLG equation where the torque considered is limited to the Berger spin-transfer torque $\boldsymbol{\tau}_B$ in equation (1.15). The equation of motion of $\boldsymbol{m}$ then reads

$$\frac{\partial \boldsymbol{m}}{\partial t} = -\gamma \boldsymbol{m} \times \boldsymbol{\mathcal{H}}_{eff} + \alpha \boldsymbol{m} \times \frac{\partial \boldsymbol{m}}{\partial t} + \boldsymbol{\tau}_B \tag{3.1}$$

where in this case

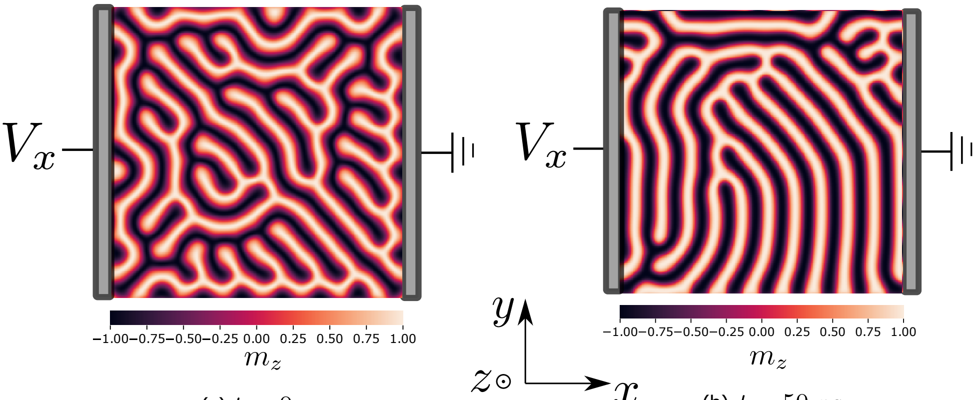
$$\boldsymbol{\mathcal{H}}_{eff} = \frac{2A}{\mu_0 M_s} \nabla^2 \boldsymbol{m} + \frac{2K}{\mu_0 M_s} \boldsymbol{m} \cdot \hat{\boldsymbol{z}} - \frac{2D}{\mu_0 M_s} \nabla \times \boldsymbol{m}. \tag{3.2}$$

In other words this is a system where there is no temperature fluctuations. Furthermore, there is an easy axis along the $\hat{\boldsymbol{z}}$ direction and no external magnetic field $\boldsymbol{h}$. The magnetic texture creates a maze-like configuration as can be seen in figure. For this type of magnetization to be stable, it is important with a strong DMI-interaction as compared to the other terms in Eq. (3.2). More specifically this means that $D > 4\sqrt{AK}/\pi$. There is an infinite configuration space spanned by the different maze structures possible, which is almost energy degenerate. One can easily change magnetization structure through the spin-transfer torque by applying an external current into the film.

The following simulations solves Eqs. (1.15), (3.1), (3.2), (1.30), (1.37) and (1.33) using *Boris Computational Spintronics* [2].

3.1.1 Conductance as a function of maze structure

A simple simulation highlights what happens when voltages and currents is applied to magnetic maze structures. This is done to gain insight in the affects the spin transfer torque have on the conductance of a magnetic thin film with maze structures. Initially a 400 nm × 400 nm × 2 nm chip is wedged between two electrodes. The setup and result is shown in figure 3.1. The initial configuration of the maze structure is obtained by a uniformly magnetized film in the z-direction, and then relaxed toward one of the many degenerate ground states of Eqs. (3.1) and (3.2) with $\boldsymbol{\tau}_B = 0$. This is the starting maze structure at $t = 0\,\text{ns}$. Due to the symmetric initialization of the structure, there is no reason why the film should exhibit different properties in the y or x direction.



(a) $t = 0$ ns

(b) $t = 50$ ns

Figure 3.1: 400 nm × 400 nm × 2 nm chip with magnetic maze structure depicted by the z-component of the magnetization vector $\boldsymbol{m}$ (out of plane). Electrodes are attached to the vertical sides of the film. **(a)** is the initial random configuration at $t = 0\,\text{ns}$ before a voltage V_x is applied. **(b)** shows how the structure has changed as a result of the current flowing through the chip for 50 ns.

The resulting pattern after applying voltages for $t = 50\,\mathrm{ns}$ no longer appears to be random, but vertical lines normal to the current direction has appeared. This can be understood as an attempt to minimize the resistance through the current experiences through the chip. From equation 1.37 we see that the local conductivity is the largest for $\boldsymbol{m} \cdot \hat{\boldsymbol{j}} = 0$ i.e. when the current and magnetization are normal to each other. The spin transfer torque between the local electrons in the film and the free electrons in the current will try to align $\boldsymbol{j}$ and $\boldsymbol{m}$. As this is the case, $\boldsymbol{m}$ should be continuously rotating around the x-axis to ensure that $\boldsymbol{j}$ never manages to catch up with $\boldsymbol{m}$. This is achieved when the stripes align normal to the x-axis. Figure 3.2 displays how the conductance changes in the film during the 50 ns simulation. Conductance is defined as

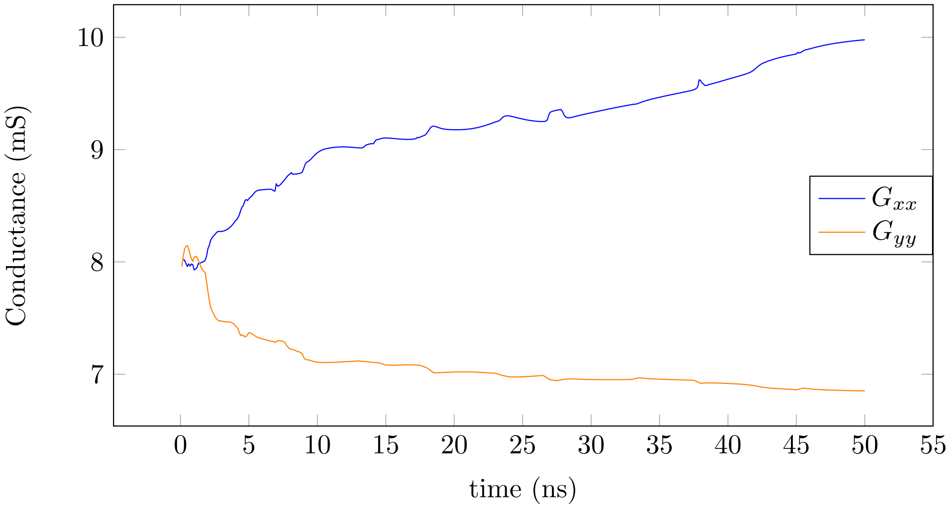


Figure 3.2: Conductance G_{xx} and G_{yy} for the film in figure 3.1 as a function of time.

$$G_{ij} = \frac{I_i}{V_j} \tag{3.3}$$

and is the extensive version of Eq. (1.30). G_{xx} is steadily increasing, but most rapidly in the first 10 ns. At one point (not shown in the figure) the conductance will reach a upper threshold, when all stripes are straightened and normal to the applied current direction. An important point is that the conductance G_{yy} is decreasing as G_{xx} is increasing. No currents have been applied in the y direction. However, the movement of the domain walls to decrease the resistance in the x direction comes at the cost of a higher resistance in the y direction. This will be an important point in the next section, when a physical Hopfield network is discussed.

3.2 Physical realization of the Hopfield network

This section will describe the framework on which the physical Hopfield network will be built, and then provide the results of the numerical simulations.

3.2.1 Physical neurons and weights

The idea is to realize a physical Hopfield neural network by attaching a set of electrodes to a thin film that exhibits stripe skyrmions. The neurons of the network will be the voltage of each electrode, and the state if the network will be described by to describe the state of the network. Now the voltage is taken to be either $+V_0$ or $-V_0$, so that each neuron . The currents flowing into or out of each electrode is governed by Kirchoff's law

$$I_i = \sum_j G_{ij}[\boldsymbol{m}]V_j = (\boldsymbol{GV})_i \tag{3.4}$$

where $G_{ij}[\boldsymbol{m}]$ is the conductance matrix of the film, that is dependent on the magnetization $\boldsymbol{m}$ due to the AMR. The network can be trained to remember a set of binary values by simply applying a pattern of voltages $\boldsymbol{V} = \{V_i\}$. Note that indices i, j here do not refer to spacial coordinates, but rather to specific electrodes. Currents will start to flow in the chip from electrodes with $V_i = +V_0$ to electrodes with $V_j = -V_0$ as long as not all electrodes are kept at either $+V_0$ or $-V_0$. As we have seen in figure 3.1 the structure will start to evolve such that the conductance of the film increases in those directions where currents are flowing. This will simultaneously lead to an decrease in conductance in the areas where currents are not flowing. By measuring the elements of $\boldsymbol{G}$ before and after the training procedure one can obtain which pattern that has been encoded in the film. $G_{ij} = I_i/V_j$ can be measured by simply setting all $V_{k\neq j} = 0$ and $V_j = +V_0$ and measuring the current I_i. The sign convention used for the for the current is positive I_i when flowing out of the electrode. It is worth noting that when this convention is used, the off diagonal terms will be negative $G_{i\neq j} < 0$ while $G_{ii} < 0$. This can be seen from a current conservation principle. When all $V_k = 0$ except $V_i = +V_0$ current will flow out of electrode i and into all the others. Since we must have $I_i = \sum_{k \neq i} I_k$, where I_k must be negative, we get by inserting Eq. (3.4)

$$G_{ii} = -\sum_{k \neq i} G_{ki} > 0 \quad \text{where} \quad G_{ki} < 0 \quad \text{for} \quad i \neq j. \tag{3.5}$$

The memory that is trained into the network is captured in $G'_{ij} = G_{ij} - G^0_{ij}$. This can now be connected to the Hopfield neural network model, by letting the electrodes be the neurons in the model, $\text{sign}(V_i)$ be the state of the neurons, and finally G'_{ij} be the weights between each neuron. It is important to make the distinction between G'_{ij} and G_{ij}. This is because the starting configuration of the texture may already have some patterns preferred over others. As a matter of fact, since $\boldsymbol{G}$ is inversely proportional to the distance between electrodes the connections between for example neuron 1 and 2 will be larger in absolute value than 1 and 3, $|G_{12}| > |G_{13}|$ since electrodes 1 and 2 is closer together physically on the chip than 1 and 3. The network should have no bias for some patterns before training, and this is accounted for by choosing G'_{ij} to be the weights. In [1, 22] it is suggested to use the uniformly magnetized film as $\boldsymbol{G}^0$. Granted this will remove the bias towards electrodes that are close together, but it does not account for the case where the initial configuration of the maze structure just happens to prefer some neuron connections than others. This requires extra steps before training to ensure that the domain walls is not aligned by chance to reduce resistance between some electrodes rather than others. Therefore, I suggest it is better to choose the starting conductance that the film has *with* texture. This has it drawback as well, as one would have to measure $\boldsymbol{G}^0$ for each new chip one is training.

The power consumption of the chip, E, is related to the energy function of the physical Hopfield network, E', in the following way

$$E = -\sum_i I_i V_i = -\sum_i \sum_j (G'_{ij} + G^0_{ij}) V_j V_i = -\boldsymbol{V}^\top \boldsymbol{G}' \boldsymbol{V} - \boldsymbol{V}^\top \boldsymbol{G}^0 \boldsymbol{V} = E' + E^0. \tag{3.6}$$

Here the last term $E^0 = -\boldsymbol{V}^\top \boldsymbol{G}^0 \boldsymbol{V}$ is a constant, while $E' = -\boldsymbol{V}^\top \boldsymbol{G}' \boldsymbol{V}$ holds all the information about the memories, and is the energy function of the network as G'_{ij} is chosen to be the weights between neurons, and not G_{ij}.

3.2.2 Simulating memory learning on a chip

Four input electrodes is connected to a circular thin film as shown in figure 3.3. Note that the electrodes is now situated on top of the chip, instead of on the sides as in figure 3.1. The film exhibits a maze structure, due to a large DMI coefficient. The diameter of the film is 1 µm and has a thickness of 2 nm. The electrodes are 100 nm × 100 nm squares and placed 600 nm apart in a square. The initial state of the film is the relaxed state from a uniformly magnetized film in the z direction, governed by Eqs. (3.1) with no external torques $\boldsymbol{\tau} = 0$. The figure shows the initial state with the electrodes numbered in a clockwise manner. The memories $\boldsymbol{\xi}^{(1)} = ++-+$, $\boldsymbol{\xi}^{(2)} = +--+$ and $\boldsymbol{\xi}^{(3)} = +-+-$ have been trained into three different chips. The notation $+-+-$ as before refers to the state of the network being $+$ in the first neuron, $-$ in the second neuron and so forth. It is important to emphasize that there is one separate chip and simulation for each memory. However, all started with the same maze structure. Coefficients used in simulations is given in table 3.1. From the maze pattern of the magnetization it is possible to see that for especially the $++-+$ and $+-+-$ memories in the top right and bottom left chips follow the principle of straitening the domain walls out normal to the direction of the current flow $\boldsymbol{j}$.

It is interesting that in the computer programmable version discussed in section 2.5 the weights is obtained by summing over the outer products of the memories $\boldsymbol{J} = \lambda \sum_i^p \boldsymbol{\xi}\boldsymbol{\xi}^\top$ while on this hardware version, the training is achieved by simply keeping the network in the state of the memory one is trying to read in for a long time.

To see if this procedure actually ends up at the memory patterns being stored in the chips, it is natural to inspect the energy function E' and look for minima. Figure 3.4 shows E_0, E' and E for all three chips, where the $\boldsymbol{V}$ space have been projected down to 1 dimension, in a similar fashion as in figure 2.4. All patterns $\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}, \boldsymbol{\xi}^{(3)}$ have been encoded in their respective chips energy function as the global minima. If one for instance examines the middle plot where the memory $\boldsymbol{\xi}^{(2)} = +--+$ have been trained, it is evident that the energy function E' indeed has a minima for the desired pattern. Furthermore, the energy of all other configurations has increased. This is in correspondence with the results shown in figure 3.2, i.e. that increase in conductance in one direction can come at the expense of decrease in another direction. However, for memories $\boldsymbol{\xi}^{(1)}$ and $\boldsymbol{\xi}^{(3)}$ have led to a decrease in E' for other patterns as well. For the bottom graph in figure 3.4 all patterns $\boldsymbol{V}$ has decreased, with the exception of the trivial pattern $++++$ where no currents are flowing. This can be understood by looking at the corresponding magnetization configuration in figure 3.3. Every pattern except the trivial one will experience increased conductance for at least on set of neighboring electrodes.

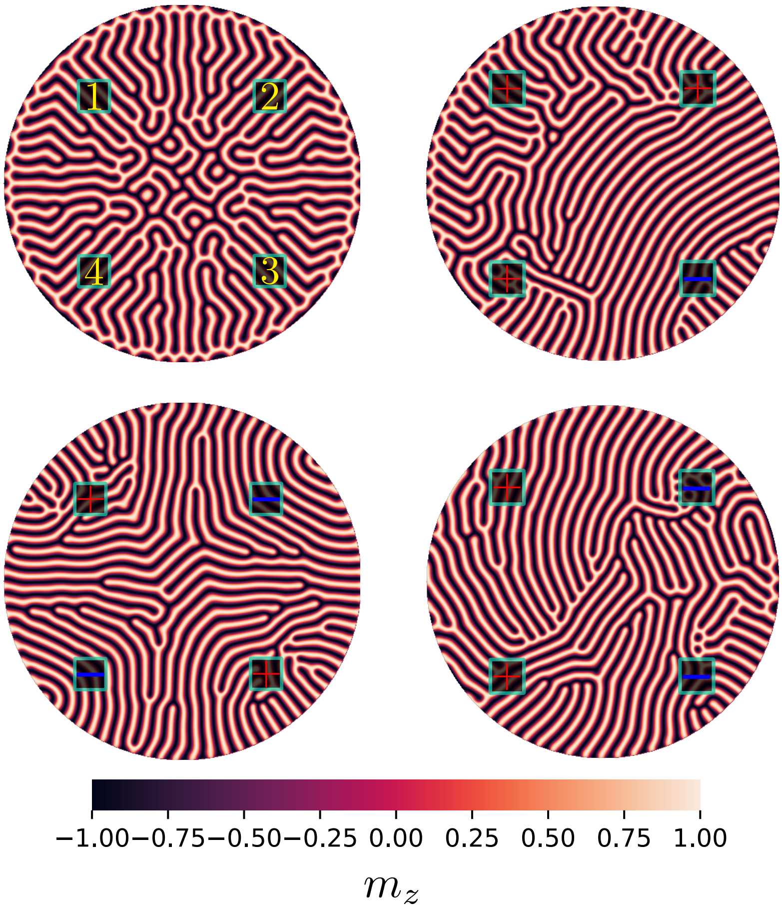


Figure 3.3: Figure shows the initial maze structure before training (top left) with electrodes numbering convention. The rest of the images show the resulting maze structures after applying the indicated voltages (+ or −) for 100 ns. z direction is out of plane.

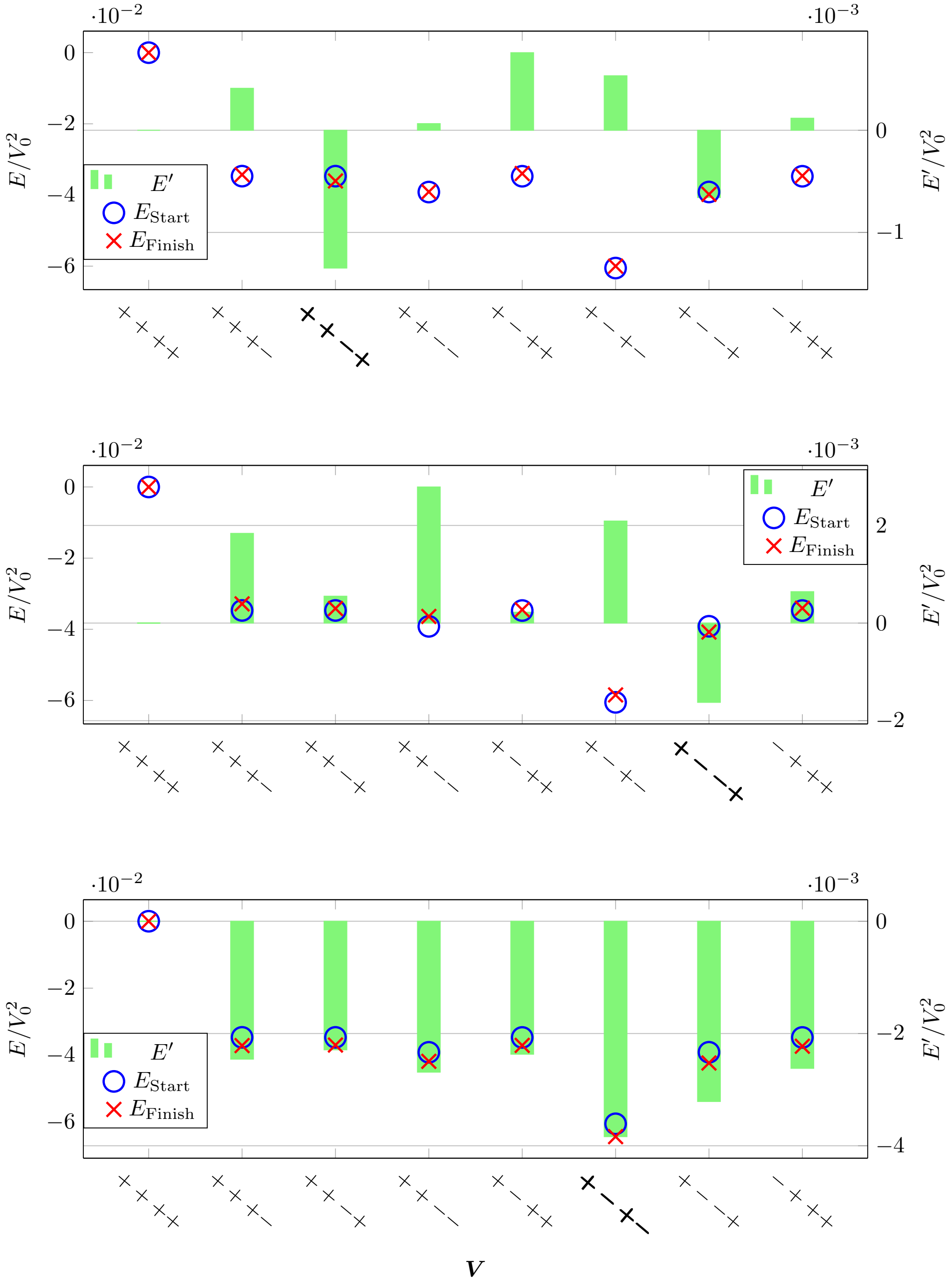


Figure 3.4: The energy function, E', together with the total energy before and after training. $E' = E_{\text{Finish}} - E_{\text{Start}}$. The x-axis indicates the state of the network $\boldsymbol{V}$ excluding the mirrored states. Memorized patterns in bold.

The third chip in the bottom figure of 3.4 the state $\boldsymbol{V} = +--+$ is a local minima and thus a spurious memory. There is no state one can reach from $+--+$ with a single bit flip with given energy landscape that has less energy. Furthermore, it is worth noting that in the top plot of figure 3.4 training pattern $\boldsymbol{\xi}^{(1)} = ++-+$ has led to an decease in E' for $\boldsymbol{V} = +--+$ while training pattern $\boldsymbol{\xi}^{(1)} = +--+$ has led to an increase in E' for pattern $\boldsymbol{V} = ++-+$.

Name	Parameter	Value	Unit
Exchange constant	A	6.0005×10^{-12}	$\mathrm{J\,m^{-1}}$
Easy-axis anisotropy	K^z	$1.499\,97 \times 10^{5}$	$\mathrm{J/m^3}$
Saturation magnetization	M_s	4.9×10^{5}	$\mathrm{A\,m^{-1}}$
Gilbert damping	α	0.3	1
Field-like torque coefficient	β	0	1
DMI coefficient	D	$2.001\,19 \times 10^{-3}$	$\mathrm{J/m^2}$
Isotropic conductivity	σ_0	7×10^{6}	$\mathrm{S\,m^{-1}}$
Spin polarization	P	0.8	1
AMR coefficient	a	1.5	1
Writing voltage	V_0^W	0.05	V
Reading voltage	V_0^R	0.001	V

Table 3.1: Coefficients used in simulations. Note that $\sigma_0 = 1/\rho_\perp$ and $a = (\rho_\parallel - \rho_\perp)/\rho_\perp$. The material parameters are typical for Pt/CoFe/MgO multilayers [22], except for the AMR coefficient which is chosen to be larger than natural to speed up the simulations.

3.2.3 Inferring process

The last subsection considered reading in a memory into the magnetic films. The energy function indicates that the memory indeed have been stored. To complete the formulation of the physical Hopfield network the memory retrieval process needs to be considered as well. It is possible to calculate the energy function for all possible $\boldsymbol{V}$ and read out the minima similar to what has been done in figure 3.4. Computationally this method is not preferable as it involves calculating the energy of all $2^{(N-1)}$ possible states, excluding the mirror states. The iterating procedure discussed in section 2.1 should be implemented for this problem. This is achieved by letting the voltages at electrodes evolve from a random initial pattern $\boldsymbol{V}(t=0)$ as

$$V_i(t+1) = \text{sign}(I_i'(t))V_0 \tag{3.7}$$

where $I_i' = (\boldsymbol{G'V})_i = I_i - I_i^0$, and is a similar separation that has been done with energy $E' = E - E^0$ and conductance $\boldsymbol{G'} = \boldsymbol{G} - \boldsymbol{G^0}$. The updating of states V_i could either be done in serial or parallel. Increased memory retrieval accuracy of the system is accomplished by turning the process into a Boltzmann machine by selecting a neuron i at random, and flipping the state $V_i(t+1) = -V_i(t)$ with probability

$$P = \frac{1}{1+\exp(\Delta E/T)}. \quad \text{where} \quad \Delta E = E(t+1) - E(t). \tag{3.8}$$

T here is a fictitious temperature that should be slowly decreased towards 0 to minimize the chance of retrieving a spurious memory. The Boltzmann machine does not require calculating $\boldsymbol{G'}$ or E' because $\Delta E = \Delta E'$ as E^0 is a constant. Albeit this comes at the cost of having to calculate the change in E for each change in $\boldsymbol{V}$ by Eq. (2.38) in addition to actually calculating the probability in Eq. (3.8). Although it may be natural to consider energies in the Boltzmann machine, it can be simplified in the following way. By rewriting ΔE as

$$\begin{aligned}
\Delta E &= E(t+1) - E(t) \\
&= -\sum_i I_i(t+1)V_i(t+1) + \sum_i I_i(t)V_i \\
&= -I_j(t+1)V_j(t+1) - \sum_{i\neq j} I_i(t+1)V_i(t+1) + I_j(t)V_j(t) + \sum_{i\neq j} I_i(t)V_i(t) \\
&= (I_j(t+1) + I_j(t))V_j(t).
\end{aligned} \tag{3.9}$$

One can vastly reduce the number of computations needed in each iteration of the Boltzmann machine. The Boltzmann machine requires a temporary flip of $V_j(t) \to V_j(t+1)$ to make it possible to measure $I_j(t+1)$. If a random number between 0 and 1 is less than or equal to the probability in Eq. (3.8), than the flip is made permanent. If not, V_j is returned back to its original state.

4 Summary and outlook

This chapter will summarize the findings in the report. The results and its meaning will be discussed, while natural further work is indicated.

As we have seen, simulations indicate that it is possible to realize a physical Hopfield neural network for associative memory tasks. However, simple 4 node networks would have little practical use. The true power of the Hopfield network is first exploited when the number of nodes grow large. It is natural to consider how one would scale up such systems. It has been proposed to simply scale up the 2×2 electrode design above into an $N \times N$ [22]. This would however cause some drawbacks. For a Hopfield model to be fully connected, the weights between each neuron should be independent. It is the current between electrodes than creates the foundation for the weights between neurons. If one considered the two neurons at opposite corners in a large $N \times N$ network, it is hard to imagine that the current between these two will be independent of what happens locally to each neuron. Nevertheless, they would be able to interact with distant neurons indirectly through its neighbors. In other words this will be a locally connected Hopfield network. This may have an impact the performance of the networks memory storing capabilities.

Another measure to further this work could be to add terms to the Hamiltonian of the system. For instance temperature is neglected in the simulations. By the stochastic-Landau-Lifshitz-Gilbert equation (1.27) a large enough temperature will distort the systems. One could even envision that a temperature might diminish the effect of spurious memories. In addition one could consider the other torques presented in section 1.2.2. The effects of this is not obvious and should be investigated to see if it either improves or deteriorate the performance of the network.

When one or several memories are encoded in a chip, one could retrieve the $\boldsymbol{G'}$ matrix and run the energy function minimization procedure discussed in section 3.2.3 with a conventional computer program. This would involve solving Eq. (3.4) for each iteration. Considering a 4 neuron network this is not a problem. However, when the number of neurons grow large, this might not be an effective way to retrieve memories and would not exploit the advantage of intrinsic computation in these films. The Eq. (3.4) is of course naturally solved in the film when applying voltages, and obtaining the currents is just a matter of doing a measurement at each electrode. However, one would still have to implement an electronic system capable of flipping the voltage V_i depending on sign(I') and not the measured current I. This would have to work on top of the magnetic films, controlling the read out process and retrieving the network state when a stable pattern is achieved. Considering that the read out process is more computationally expensive than the writing, it is natural to consider the latter of these approaches as the most preferable for large networks.

Increasing the chip size and adding more electrodes, while increasing the number of terms in the Hamiltonian will escalate the simulation time. These simple simulations running on an NVIDIA GeForce GTX 970 GPU had simulation time in the order of days, even as *Boris Computational Spintronics* incorporates parallel computing. It is difficult to imagine that a simulation of a large network of this kind with the inclusions of all effective fields and torques discussed in section 1.2 on a conventional computer would be feasible. Physical experiments should be considered as a more practical approach than numerical simulations when the number of neurons grow large.

In this report, the basic theoretical introduction to spintronics and Hopfield neural networks have been given. This has then been used to numerically simulate simple 4-neuron Hopfield networks. The simulations is a proof of concept that it is possible to realize hardware components for neuromorphic computing capable of associative memory. The non-linearity and plasticity of magnetic thin films with stripe skyrmions has been used to train specific memories. The training procedure happens naturally within the film, as the free energy of the film is minimized with external currents present. The plasticity of the stripe skyrmions ensure that memories encoded are not volatile. The read out process is done searching for the minimum power consumption of the film. The computation is done naturally within in the film by Ohm's law. The magnetic films thus stores the weights where the computation takes place. Retrieving memories is thus reduced to a matter of measuring a current and flipping voltages. The hardware architecture is fundamentally different from the Von Neumann architecture implemented in classical computers, and is not limited by the Von Neumann bottleneck. The simulations show promising results for memory storing and retrieval, inspiring further work in spintronics based neuromorphic computing.

Bibliography

[1] W. Yu, J. Xiao, and G. E. W. Bauer. "Hopfield neural network in magnetic textures with intrinsic Hebbian learning". In: *Physical Review B* 104.18 (Nov. 16, 2021). Publisher: American Physical Society, p. L180405. URL: https://link.aps.org/doi/10.1103/PhysRevB.104.L180405 (visited on 12/03/2021).

[2] S. Lepadatu. "Boris computational spintronics—High performance multi-mesh magnetic and spin transport modeling software". In: *Journal of Applied Physics* 128.24 (Dec. 28, 2020). Publisher: American Institute of Physics, p. 243902. ISSN: 0021-8979. URL: https://aip.scitation.org/doi/full/10.1063/5.0024382 (visited on 11/08/2021).

[3] *COMSOL Multiphysics®*. Version 5.6. Stockholm, Sweden. URL: www.comsol.com.

[4] W. Yu. *Micromagnetic Simulation with COMSOL Multiphysics®*. COMSOL. URL: https://www.comsol.com/blogs/micromagnetic-simulation-with-comsol-multiphysics/ (visited on 12/20/2021).

[5] G. Moore. "Cramming More Components Onto Integrated Circuits". In: *Proceedings of the IEEE* 86.1 (Jan. 1998). Reprinted from Gordon E. Moore, "Cramming More Components onto Integrated Circuits," Electronics, pp. 114–117, April 19, 1965., pp. 82–85. ISSN: 1558-2256.

[6] R. S. Williams. "What's Next? [The end of Moore's law]". In: *Computing in Science Engineering* 19.2 (Mar. 2017). Conference Name: Computing in Science Engineering, pp. 7–13. ISSN: 1558-366X.

[7] *AlphaGo: The story so far*. Deepmind. URL: https://deepmind.com/research/case-studies/alphago-the-story-so-far#what_is_go_ (visited on 12/17/2021).

[8] E. Goi et al. "Perspective on photonic memristive neuromorphic computing". In: *PhotoniX* 1.1 (Mar. 3, 2020), p. 3. ISSN: 2662-1991. URL: https://doi.org/10.1186/s43074-020-0001-6 (visited on 12/17/2021).

[9] J. Grollier et al. “Spintronics for neuromorphic computing”. In: *arXiv:2007.06092 [cond-mat, physics:physics]* (July 12, 2020). arXiv: 2007.06092. URL: http://arxiv.org/abs/2007.06092 (visited on 12/17/2021).

[10] D. Prychynenko et al. “Magnetic Skyrmion as a Nonlinear Resistive Element: A Potential Building Block for Reservoir Computing”. In: *Physical Review Applied* 9.1 (Jan. 31, 2018). Publisher: American Physical Society, p. 014034. URL: https://link.aps.org/doi/10.1103/PhysRevApplied.9.014034 (visited on 11/15/2021).

[11] P. C. Hemmer. *Kvantemekanikk*. 5. utg. Book Title: Kvantemekanikk. Trondheim: Tapir akademisk forl, 2005. 398 pp. ISBN: 978-82-519-2028-5. URL: https://www.nb.no/search?q=oaiid:”oai:nb.bibsys.no:990505369154702202”&mediatype=bÃÿker (visited on 12/20/2021).

[12] S. Chikazumi and C. D. Graham Jr. *Physics of Ferromagnetism*. 2nd ed. Vol. 94. International Series of Monographs on Physics. Oxford University Press Inc., New York, 1997. ISBN: 978-0-19-956481-1.

[13] K. Akashdeep. *Magnetism and Spintronics: A quick introduction*. Dec. 9, 2021.

[14] L. A. Kristiansen. “Spin-orbit-induced dynamics of magnetic textures”. PhD thesis. Norwegian University of Science and Technology, Jan. 2017.

[15] H. Yu, J. Xiao, and H. Schultheiss. “Magnetic texture based magnonics”. In: *Physics Reports*. Magnetic texture based magnonics 905 (Apr. 15, 2021), pp. 1–59. ISSN: 0370-1573. URL: https://www.sciencedirect.com/science/article/pii/S0370157320304270 (visited on 10/19/2021).

[16] A. Brataas, A. D. Kent, and H. Ohno. “Current-induced torques in magnetic materials”. In: *Nature Materials* 11.5 (2012). Num Pages: 10 Publisher: Nature Publishing Group, The Macmillan Building London N1 9XW United Kingdom, pp. 372–381. ISSN: 1476-1122. URL: https://www.proquest.com/docview/1031325250?pq-origsite=primo& (visited on 10/20/2021).

[17] A. Qaiumzadeh, R. A. Duine, and M. Titov. “Spin-Orbit Torques in Two-Dimensional Rashba Ferromagnets”. In: *Physical Review B* 92.1 (July 6, 2015), p. 014402. ISSN: 1098-0121, 1550-235X. arXiv: 1503.06872. URL: http://arxiv.org/abs/1503.06872 (visited on 10/04/2021).

[18] J. C. Slonczewski. “Current-driven excitation of magnetic multilayers”. In: *Journal of Magnetism and Magnetic Materials* 159.1 (1996), pp. L1–L7. ISSN: 0304-8853. URL: https://www.proquest.com/docview/26094235?pq-origsite=primo (visited on 10/20/2021).

[19] S. Ament et al. “Solving the stochastic Landau-Lifshitz-Gilbert-Slonczewski equation for monodomain nanomagnets : A survey and analysis of numerical techniques”. In: *arXiv:1607.04596 [cs]* (Jan. 16, 2017). arXiv: 1607.04596. URL: http://arxiv.org/abs/1607.04596 (visited on 10/20/2021).

[20] R. Khoshlahni et al. “Ultrafast generation and dynamics of isolated skyrmions in antiferromagnetic insulators”. In: *Physical Review B* 99.5 (Feb. 21, 2019). Publisher: American Physical Society, p. 054423. URL: https://link.aps.org/doi/10.1103/PhysRevB.99.054423 (visited on 10/20/2021).

[21] T. H. Pate. “Hölder type matrix inequalities of Pate, Blakley, and Roy extended to the inner product of Frobenius”. In: *Linear Algebra and its Applications*. Special Issue in Honor of José Perdigão Dias da Silva 436.6 (Mar. 15, 2012), pp. 1763–1769. ISSN: 0024-3795. URL: https://www.sciencedirect.com/science/article/pii/S0024379511006525 (visited on 12/15/2021).

[22] W. Yu, J. Xiao, and G. E. W. Bauer. “Supplemental Materials: A Hopfield neural network in magnetic textures with intrinsic Hebbian learning”. In: *Physical Review B* (2021), p. 12.

[23] T. Taniguchi, J. Grollier, and M. Stiles. “Spin-Transfer Torques Generated by the Anomalous Hall Effect and Anisotropic Magnetoresistance”. In: *Physical Review Applied* 3.4 (Apr. 6, 2015). Publisher: American Physical Society, p. 044001. URL: https://link.aps.org/doi/10.1103/PhysRevApplied.3.044001 (visited on 12/04/2021).

[24] B. Göbel, I. Mertig, and O. A. Tretiakov. “Beyond skyrmions: Review and perspectives of alternative magnetic quasiparticles”. In: *Physics Reports*. Beyond skyrmions: Review and perspectives of alternative magnetic quasiparticles 895 (Feb. 12, 2021), pp. 1–28. ISSN: 0370-1573. URL: https://www.sciencedirect.com/science/article/pii/S0370157320303525 (visited on 11/08/2021).

[25] S. Woo and K. Litzius. “Observation of room-temperature magnetic skyrmions and their current-driven dynamics in ultrathin metallic ferromagnets”. In: *Nature Materials* 15.5 (May 2016), pp. 501–506. ISSN: 1476-4660. URL: https://www.nature.com/articles/nmat4593 (visited on 11/08/2021).

[26] T. Geszti. *Physical Models of Neural Networks*. Budapest: World Scientific Publishing Co. Pte. Ltd., 1990. ISBN: 981-02-0012-9.

[27] D. A. Woodruff. *What is a neuron?* Nov. 22, 2016. URL: https://qbi.uq.edu.au/brain/brain-anatomy/what-neuron (visited on 12/19/2021).

[28] S. N. *English: Structure of Neuron (unlabeled).* Feb. 23, 2018. URL: https://commons.wikimedia.org/wiki/File:Structure_of_Neuron.png (visited on 12/19/2021).

[29] J. von Neumann. “First draft of a report on the EDVAC”. In: *IEEE Annals of the History of Computing* 15.4 (1993). Conference Name: IEEE Annals of the History of Computing, pp. 27–75. ISSN: 1934-1547.

[30] F. Peper. “The End of Moore’s Law: Opportunities for Natural Computing?” In: *New Generation Computing* 35.3 (July 1, 2017), pp. 253–269. ISSN: 1882-7055. URL: https://doi.org/10.1007/s00354-017-0020-4 (visited on 12/19/2021).

[31] Daniel L. Stein and Charles M. Newman. *Spin Glasses and Complexity.* Primers in Complex Systems. Princeton: Princeton University Press, 2013. ISBN: 978-0-691-14733-8. URL: https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=484814&site=ehost-live (visited on 11/01/2021).

[32] J. O. Andersen. *Introduction to Statistical Mechanics.* First Edition. Trondheim: Akademika forlag, 2012. ISBN: 978-82-321-0105-4.

[33] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning.* MIT Press, 2016. URL: https://www.deeplearningbook.org/ (visited on 10/04/2021).

[34] K. Hornik, M. Stinchcombe, and H. White. “Multilayer feedforward networks are universal approximators”. In: *Neural networks* 2.5 (1989). Place: OXFORD Publisher: Elsevier Ltd, pp. 359–366. ISSN: 0893-6080. URL: https://search.proquest.com/docview/25289495?pq-origsite=primo (visited on 12/20/2021).

[35] G. Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of control, signals, and systems* 2.4 (1989), pp. 303–314. ISSN: 0932-4194.

[36] E. T. Jaynes. “Information Theory and Statistical Mechanics”. In: *Physical Review* 106.4 (May 15, 1957). Publisher: American Physical Society, pp. 620–630. URL: https://link.aps.org/doi/10.1103/PhysRev.106.620 (visited on 12/19/2021).

[37] H. Ramsauer et al. “Hopfield Networks is All You Need”. In: *arXiv:2008.02217 [cs, stat]* (Apr. 28, 2021). arXiv: 2008.02217. URL: http://arxiv.org/abs/2008.02217 (visited on 12/20/2021).

Austefjord, J.W.

NTNU
Norwegian University of
Science and Technology