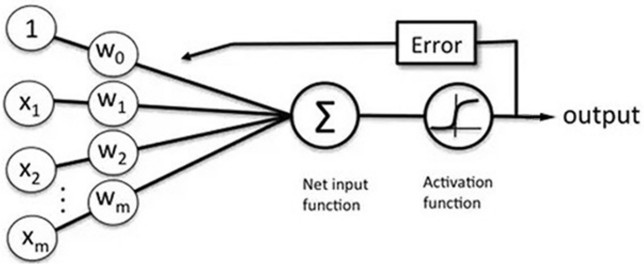
UTRECHT UNIVERSITY
FACULTY OF HUMANITIES
BACHELOR ARTIFICIAL INTELLIGENCE

Modelling the Human Brain:
Feed-Forward Neural Networks

A Comparison

BSc THESIS

SANDER EMILE BOS



Advisor: Dr. Ben Harvey ,
Co-advisor: Prof. dr. Michael Moortgat

ECTS Credits: 7.5

Utrecht, 2018

Utrecht University

Utrecht University
Faculty of Humanities
Bachelor Artificial Intelligence

Modelling the Human Brain: Feed-Forward Neural Networks

A Comparison

BSc Thesis

SANDER EMILE BOS

Advisor: Dr. Ben Harvey ,
Co-advisor: Prof. dr. Michael Moortgat

Approved by Date:.

(Signature) *(Signature)* *(Signature)*

. . . .
Dr. Ben Harvey , Prof. dr. Michael Moortgat

Utrecht, 2018

Abstract

Artificial neural networks are increasingly being used in tasks that would traditionally require a human intervention. These computational models form a mathematical abstraction of biological neural networks. McCulloch and Pitts proposed such a model to connect chains of neural events with inferences between logical propositions of the mind. Frank Rosenblatt, consequently, extended this idea and proposed the Perceptron in order to represent more complex functions. This model is more similar to the biological neural network, Kunihiko Fukushima further explored this concept and applied it to the mammalian visual system. In this paper we discuss these so called feed-forward models to the extent of the shared similarities between artificial and biological neural networks. We provide the necessary background knowledge regarding the basic artificial and biological neuronal computation units. Additionally, we give an overview of the hierarchical nature of the visual system and discuss this in the context of convolutional neural networks. From this comparative analysis it follows that although artificial neural networks function according to certain biological principles, some principles are still not accounted for and might prove essential for particular aspects of neurological processing. Highlighting the shortcomings of current models should serve as a waypost for future research, such that it may add to our understanding of both computational models as well as biological models.

Keywords

Artificial Neural Networks, Visual Processing, Convolutional Neural Networks, Neurons, Hebbian Learning, Backpropagation, Hierarchical Processing

To the loving memory of my grandfather,
who did not live to see this work come to fruition.

Contents

Chapter 1

Introduction

Artificial intelligence (AI), as the name suggests, is the scientific discipline that concerns itself with the understanding and replication of intelligence. Russell and Norvig differentiate between several distinct approaches which are divided in several different categories [1], specifically: Acting Humanly—studies agents from a behaviouristic standpoint. It concerns itself with the question: 'When can we consider an agent to be intelligent'. A notable example is the Turing test; Acting Rationally—concerns itself with an agents ability of perceiving its environment and its capability of reaching certain goals within that environment; Thinking Humanly—this approach tries to model human cognition, be it through introspection or through experimental observation. Cognitive science, for example, is one such approach that integrates experiment with computer models; Thinking Rationally—instead of understanding 'how' humans think, this approach tries to model how we 'should' think, by utilizing symbolic logic and definitive reasoning patterns.

The brain is widely regarded as the locus of cognition. Artificial neural networks (ANNs) are computational models of the brain that aim to capture, and possibly transcend, the complexities of human cognition. Therefore this pursuit typically falls into the 'thinking humanly' category, but as we shall see with McCulloch and Pitts who modelled thoughts or 'ideas' as logical propositions, it is not uncommon for these approaches to draw from other categories as well. Historically, most of these approaches have been closely linked to the field of neuroscience, but recent years have fostered a dwindling in their collaboration. As such, artificial neural networks are currently being studied increasingly segregated from neuroscience, while insights from neuroscience could in fact be used as a guidance for the development of new computational models. Conversely, new artificial network algorithms might shed light on input-output mappings of certain brain areas. Additionally, an understanding of ANNs may add to the interpretability of cortical processing.

Nevertheless, current ANNs fall short of the complexities exhibited by the human brain. They are working models that abstract away from reality. In this article we aim to bring some of these abstractions to light, such that it may benefit the development of novel algorithms. By doing so, we hope that, ultimately, the value of sustaining close relations between neuroscience and AI is re-emphasized. It should be noted that this article is not intended to be a rigorous mathematical discussion about ANNs. The covered topics are only discussed to the extend that the reader can comprehend the comparisons we are trying to make. Consequently, the reader might be inclined to feel that on some topics the mathematics are glossed over. These sections will have a referral to sources that cover them in more detail.

We will come to understand neural networks by starting at the simplest level and progressively working our way up, alternating between the biological and the artificial paradigms. In this way their differences will naturally present themselves. The article is organized as follows. Chapter two will discuss the basic elements that make up a biological neural network: the nerve cell. We cover its anatomy, the cell to cell communication and what is meant by learning at this micro level of analysis. Chapter three will cover the transition from nerve cells to artificial neurons by looking at some of the earliest models. This section aims to give the reader a feel for the behavior of the machinery such that the following sections seem more intuitive. Chapter four shifts back to the realm of biology and will explain the neural network of the visual system. Chapter five will give an account on convolutional neural networks (CNNs) and how they are derived from the visual system. The last chapter is the discussion, which will give a quick overview of what has been discussed.

Chapter 2

Background

When a comparison is drawn between the workings of a biological and an artificial neural network, it is important to first have a clear understanding of the basic elements that make up such networks. For biological networks this basic element is the nerve cell, which we shall discuss in the following section. Later we shall see that the basic element of an artificial network, called a node, simulates several primary properties of a biological neuron. There is a remarkable diversity of nerve cells, interconnected in a wide variety of structures. In order for us to draw a concise comparison, we therefore need to abstract away from this complexity on several different levels. For the structural level of individual units we will only consider those aspects that are common to most neurons. For the network level we will be focusing primarily on the neocortex, since its functioning is roughly homogeneous throughout [2].

2.1 The Nerve Cell

In the human central nervous system we distinguish between two types of cells: nerve cells and glial cells [3]. Haug estimated in 1986 that humans have around $70 - 80 \times 10^9$ nerve cells and $40 - 50 \times 10^9$ glial cells [4]. However these estimates are far from accurate, since they are based on mean density measurements of glial cells and of nerve cells in the cerebellar cortex. More recent accounts estimate that we have around 10^{11} neurons and that there are at least twice as many glial cells [3]. Although glial cells support nerve cells in a variety of ways: some types modulate the K^+ concentration in the synaptic cleft, while others insulate the electrical properties of nerve cells, none are directly involved in the propagation of electrical signals throughout the nervous system. For this reason we will not take them into consideration for our network analysis. A myriad of different nerve cells can be found in the neocortex, ranging from the excitatory pyramidal cells to the inhibitory large basket cells [5] [3]. As stated before, we shall only consider those properties that are most common to all neurons.

2.1.1 Basic Structure

Nerve cells are typically comprised of the following components: a cell body (soma), axons, dendrites and synaptic terminals [3]. All these contribute in some way to the main function of the nerve cell: the propagation of electrical signals, or more formally: the propagation of action potentials. The soma is the main component of a nerve cell. This is where most organelles are situated, including the nucleus, which holds the genetic material. It is also to considered to be the main factory of the cell, since this is where most proteins are synthesized. Branching off the soma is a network of nerve fiber, known as dendrites. Together with the soma they form the input module of the nerve cell. The soma has a segment called the axon hillock, which is the foundation of a typically long strand of fiber called the axon. The axon, which branches out (arborizes) towards the end, forms the output segment of the nerve cell. The tips of these branches (presynaptic terminals) meet with other nerve cells at junctions known as synapses. Electrical signals cross between nerve cells at these synapses. Nerve cells receiving such a signal are called postsynaptic cells and the nerve cell from which the signal is received is called the presynaptic cell. Most nerve cells do not actually physically connect to other nerve cells. Rather, a small gap separates the two connecting terminals. This gap is known as the synaptic cleft. Most electrical signals are thus not transferred physically from nerve cell to nerve cell, but chemically. See Figure 1 for a schematic drawing, we shall refer to this as the basic structure of a nerve cell.

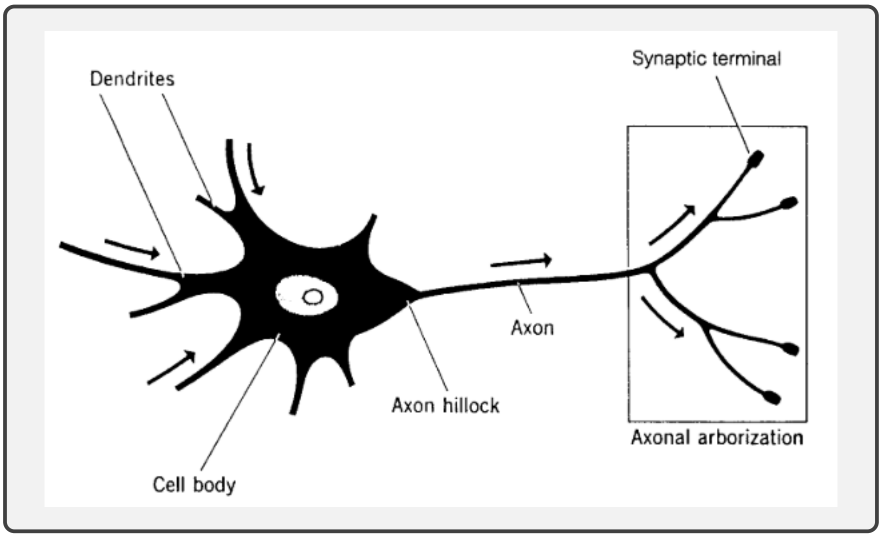


Figure 1: Schematic of a typical neuron. The axon and its axonal aborization constitute the output segment of the cell, while the dendrites and cell body form the input segment. The axon hillock is the base of the axon from which action potentials are generated. The arrows indicate the flow of electrical current. As cited in Arbib, 2003 [5].

2.2 Neuronal Communication

In 1891, Ramón y Cajal and Arthur van Gehuchten independently showed that the dendrites of the postsynaptic neurons are polarized after which the polarization flows from the dendrites to the soma and to the axon next thereafter. Ramón y Cajal called this the principle of dynamic polarization. Interestingly, they did not have any recordings of activity, they could only look at their anatomy. More recent accounts show that dendritic processing is not strictly unidirectional, but this level of complexity will not be considered in our model [5].

Electrical signaling in nerve cells comes in the form of ions. The most important ions for neuronal functioning are potassium (K^+), sodium (Na^+), calcium (Ca^{2+}) and chloride (Cl^-)[1]. The nerve cell is enveloped in a fluid containing these ions. Concentration difference between intracellular and extracellular ions gives rise to a potential difference along the membrane of the cell, which is termed the membrane potential. We distinguish between two forces that can act on these ions: a concentration gradient and an electrical gradient. The passing of ions into and out of the cell is regulated by a variety of selective proteins inside the cell membrane. Passive potassium leak channels (PLCs) and active sodium-potassium pumps (SPPs) are involved when the neuron is at rest, or more accurately: when the neuron is not spiking. The SPPs consume one ATP molecule for every three Na^+) ions they pump out of the cell and every two K^+ ions they pump into the cell. This yields a concentration gradient for K^+, which will then flow out of the cell through the passive leak channels, until the concentration gradient reaches equilibrium with the electrical gradient. Equilibrium sets in when the membrane potential approaches roughly $-70mV$. The negative sign derives from the potential outside the cell being defined as neutral.

Another protein that regulates the flow of ions is a neurotransmitter receptor, which is a type of ligand-gated ion channels. The event of a ligand[2] attaching itself to the receptor, causes the gate to open and the ions it selects upon, to flow in. We call such a binding event the input of a nerve cell. The neurotransmitter receptors are situated on the membrane of the dendrites and the soma only, which explains why these segments are considered to be the input surface of the neuron. We distinguish between excitatory and inhibitory inputs (or transmitters) to a neuron. Excitatory inputs allow cations to pass the cell membrane, raising the membrane potential locally. This is known as depolarization. On the other hand, inhibitory inputs allow anions to pass, lowering the membrane potential locally. This is known as hyperpolarization. That is, until the ligand detaches from the receptor, upon which the ion channel closes and the membrane potential plateaus. The resulting concentration of ions on the inside of the neuron quickly diffuse to areas of lower concentration inside the cell, but also to areas outside the cell through the leak channels. Consequently, the diffusion of ions causes the local membrane potential to propagate, weakening as it spreads. Thus the membrane potential subsides over distance and over time, which we will call the fading property.

Another fundamental property of neurons is the all-or-none law, which states that only when the membrane potential at the axon hillock reaches a certain threshold (around $-55mV$), an action potential is generated. If the threshold is not reached, the neuron remains at rest. Additionally, from the fading property it follows that the closer the inputs are to the axon hillock, the larger their effect will be on the generation of an action potential. Singular binding events have only a limited impact on the change in membrane potential. However, the effects of an input can stack additively if the effects of previous inputs have not yet subsided. This can happen when the frequency of incoming inputs is sufficiently high enough or when the inputs occur sufficiently close in space. If the summation of excitatory and inhibitory inputs at the axon hillock nets a potential that exceeds the threshold value, an action potential is generated and successively propagated all the way down the length of the axon. When stimuli are not collectively sufficient to bring

---

[1] Positively charged ions are called cations, whereas negatively charged ions are called anions.

[2] in this case a neurotransmitter.

the membrane potential across the threshold value, the cell refrains from firing. These events are referred to as failed initiations.

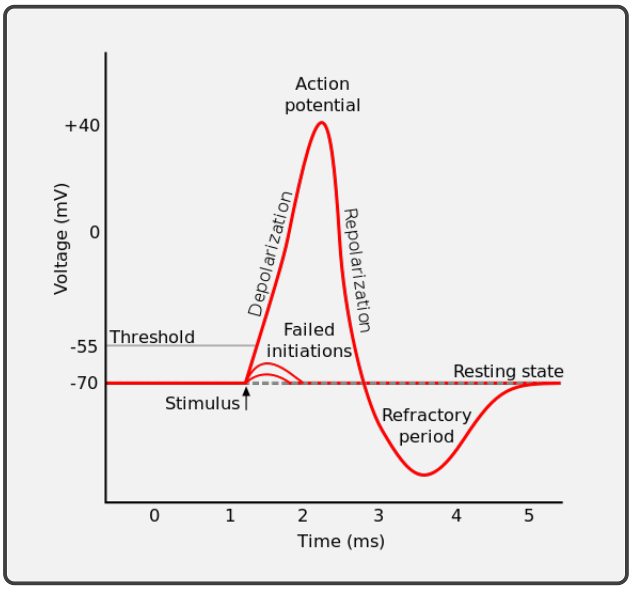


Figure 2: The characteristic shape of an action potential. When a stimulus reaches the threshold value, the cell becomes depolarised. Weak stimuli fail initiate an action potential. In the repolarisation stage the cell is brought to its restarting state; fewer positively charged ions flow into the cell. After the repolarisation, the cell enters the refractorary period, where it is generally not possible to initiate a new action potential until the membrane potential has recovered to the resting potential. [Online image] Retrieval date: June 5, 2018 from https://courses.lumenlearning.com/boundless-psychology/chapter/neurons/

Action potentials have a distinct shape, characterized by different stages, see Figure 2. First, a sequence of inputs depolarize the membrane until a threshold is reached. This causes the voltage-gated sodium channels near the axon hillock to open, allowing sodium ions to flow inwards, causing an additional increase in membrane potential. The local depolarization activates the next voltage-gated sodium channel to open, which in turn causes even more channels to open. This series of voltage-gated sodium channels opening is propagated in waves across the length of the axon to the presynaptic terminals. Local depolarization continues until the gated ion channels become saturated, which is at a membrane potential of approximately +40mV. At this stage the intracellular environment is much more positively charged than the extracellular fluid. The voltage-gated sodium channels close and potassium ions start to flow outward through the potassium leak channels and through the voltage-gated potassium channels, causing the membrane potential to drop towards the resting potential. This is known as the repolarization stage. However, the closing of the voltage-gated potassium channels is a slow process, causing the membrane potential to overshoot the resting potential, resulting in hyperpolarization of the membrane. Now the membrane potential will slowly recover to its resting potential by effectuating the potassium leak channels and the sodium pumps. We call this the refractory stage and during its first phase no new action potential can be initiated. In its second stage it is possible to have another action potential occur, however this will require a large excitatory stimulus. The refractory stage ensures that the action potential cannot reverse its course in the opposite direction down the axon. When the action potential reaches the presynaptic terminals, the membrane potential is depolarized, triggering an influx of Ca^{2+}. These ions successively come into contact with certain neurotransmitter containing structures, called vesicles. When this happens, the vesicles fuse to the membrane in such a way that the neurotransmitter molecules can enter the synaptic cleft and bind to the receptor molecules on the dendrites of the postsynaptic terminal.

2.3 Neuronal Learning

On a system level we can see that our performance on certain tasks improves based on previous experience. An important aspect of biological neural networks is that the connections between individual nerve cells is not fixed, allowing connections between presynaptic and postsynaptic cells to be strengthened the more they are simultaneously active. Moreover, strengthened connections are not temporary but are generally preserved. This principle is called ‘*Hebbian learning*’ after the man who first conceptualized it in 1949: Donald Hebb. An often used phrase that captures the essence of this principle is: ‘*cells that fire together, wire together.*’

Chapter 3

From Biological to Formal Models

During the first half of the twentieth century there was still a dichotomy between explanations of physical events and explanations of mental events. '*A Logical Calculus of the Ideas Immanent in Nervous Activity*', an article co-authored by W. McCulloch and W. Pitts, was initially written to bring these two seemingly distant topics closer together. With the advancements in both neuroanatomy; ushered in by Ramón y Cajal and neurodynamics; initiated by Rashevsky, McCulloch reckoned that a logical atomism would be able to connect thought and action, which he would call a 'leap from the psyche to soma' [6]. For this, he defined a basic mental unit[1] which could be modelled with propositional logic. These propositions are unambiguous statements which have either one of two truth values: True (denoted as: $\mathbb{T}$ or 1) or False (denoted as: $\mathbb{F}$ or 0). This way, chains of neural events would correspond to inferences between the logical propositions of the mind.

Today one would argue that this does not describe the true state of affairs, but the first models of artificial neurons were initially conceived with the intent of being overly simplistic [6]. The idea was to start with only the most essential features, study them, and continuously revise those models, until we end up with more realistic ones. This section aims to outline what the earliest models looked like and what abstractions were made.

3.1 Assumptions

The formalization of artificial neurons as conceptualized by McCulloch & Pitts rests on several key assumptions:

(i) Each neuron's activity hinges on a simple threshold law.
(ii) There is no variance in the excitability of neurons.
(iii) The activity of neurons occurs within discrete time intervals.
(iv) A neuron remains idle when it receives one or more inhibitory signals.
(v) The structure of neurons does not change with time nor with experience.

The justification for (i) comes from the 'all-or-none' property of biological neurons. As for the others, certain abstractions had to be made. For instance, during the first part of the refractory

---

[1] A mental event with temporal properties, causally connected to other such mental events.

period a neuron is incapable of firing. (iii) implies that neurons fire synchronously, whereas we have seen that incoming signals vary in time of arrival as well as in arrival distance from the axon-hillock. Moreover, biological neurons can receive a multitude of inhibitory and excitatory input signals. If those signals net a sufficiently large change at the axon-hillock to cross a boundary value, the cell fires. One instance of a inhibitory signal does not bring the summation of signals to a halt. Additionally, the inputs of biological neurons are restricted to a finite amount of space, whereas in this model the restriction does not apply. Lastly, connections between biological neurons strengthen proportional to the frequency of those neurons firing together, which is not possible in the model as it currently stands.

3.2 A Logical Calculus

McCulloch and Pitts distinguished between two types of neurons: those that are capable of receiving incoming signals and those that are not. The latter neurons are termed 'peripheral afferents' which correspond to the sensory neurons and are inherently part of networks that lack circularity. Figure 3 shows in what way functions in boolean logic are equivalent to sequences of neural events [7]. Here, excitatory and inhibitory synapses are denoted by closed and open circles respectively. A neuron only fires if it interfaces with at least two excitatory synapses and no inhibitory synapses[2]. Since we have established that neural signaling is equivalent to propositions with truth values, any neural network (comprised of such neurons) that represents a Boolean function can be expressed as a syntax tree constructed from the leafs up to the root. This model derives its strength from being able to simulate conjunctions, disjunctions and negations. From these operations any Boolean logic function can be constructed. Note that negation has not been separately defined. Negation can be simulated by a neuron with an inhibitory synapse and a threshold value of 0.

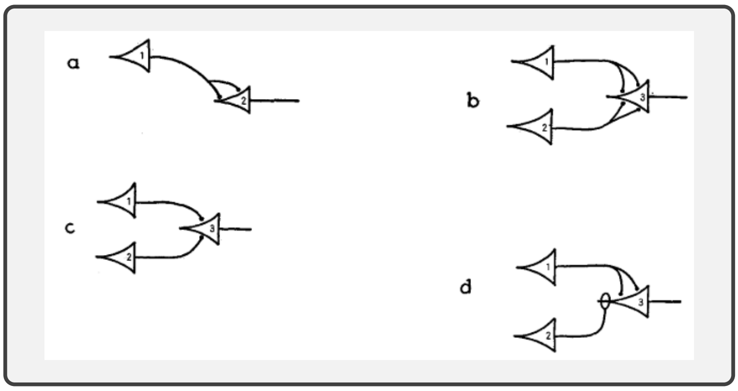
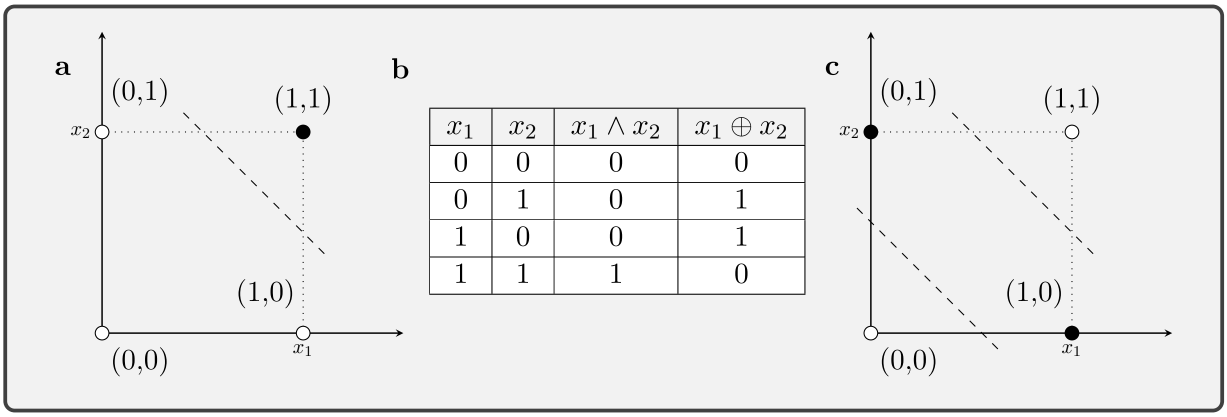


Figure 3: Example neural networks. Excitatory and inhibitory synapses are depicted as closed dots and open circles respectively. A neuron fires if it receives two or more incoming excitatory signals. Each depicted network represents a logic gate. (a) Temporal predecessor. (b) Disjunction. (c) Conjunction. (d) Conjoined negation. As in McCulloch & Pitts, 1943 [7].

The four examples depicted in Figure 3 form the elementary units from which any network may

---

[2] This corresponds to a threshold value of 2.



x_1	x_2	$x_1 \wedge x_2$	$x_1 \oplus x_2$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Figure 4: (a) Linear separability for the AND problem. (b) AND and XOR truth tables. (c) Linear inseperability of the XOR problem.

be constructed: Figure 3-a temporal predecessor, Figure 3-b disjunction, Figure 3-c conjunction, Figure 3-d conjoined negation ($A \wedge \neg B$). McCulloch and Pitts developed their own language to represent such relations, they can be expressed as follows:

(a) $\mathbf{N_2(t)} \Leftrightarrow \mathbf{N_1(t-1)}$, neuron 2 is active at time t if and only if neuron 1 is active at time $t-1$.

(b) $\mathbf{N_3(t)} \leftrightarrow \mathbf{N_1(t-1)} \vee \mathbf{N_2(t-1)}$, neuron 3 is active at t iff neuron 1 or (inclusive) neuron 2 is active at $t-1$.

(c) $\mathbf{N_3(t)} \leftrightarrow \mathbf{N_1(t-1)} \wedge \mathbf{N_2(t-1)}$, neuron 3 is active at t iff neuron 1 and neuron 2 are active at$t-1$.

(d) $\mathbf{N_3(t)} \leftrightarrow \mathbf{N_1(t-1)} \wedge \neg\mathbf{N_2(t-1)}$, neuron 3 is active at time t iff neuron 1 is active at $t-1$ and neuron 2 is idle at $t-1$.

This model presupposes absolute inhibition, meaning a single inhibitory signal is sufficient for a cell not to fire. However it should be noted that these networks are equivalent to those that employ relative inhibition. That is to say, inhibitory signals are to be subtracted from the excitatory signals. The neuron only fires if the net result of these two inputs crosses a threshold value. In this way we can define level of excitation as the strength of a connection, or in more modern terms: as the 'weight' of the connection. I.e. excitation would be a weight greater than zero, whereas inhibition would be a weight less than zero. The weights are fixed in a network composed of logical units, meaning there is no way for the output to adapt based on previous performance. In other words: the network is incapable of learning.

3.2.1 Linear Separability

McCulloch and Pitts neurons represent Boolean functions, that is, functions that map binary input to a single binary output: $f : \{0,1\}^N \rightarrow \{0,1\}$. Herein N is the dimensionality of the input space. It should be noted that such neurons represent only those Boolean functions that are linearly separable. That is, functions for which a hyperplane exists that separates the input space in two half-spaces, such that all True valued inputs are separated from the False valued inputs. Figure 4-a shows what this would look like for the AND function with $N = 2$. One such problem that cannot be solved by a perceptron is the XOR ($\oplus$) problem, since its representation in input space is cannot be linearly separated by a single hyperplane, see Figure 4-c. Fortunately, problems such as this one can be solved if we interlink multiple units according to a corresponding syntactic tree. This shall be referred to as an artificial neural network. We can define an ANN as follows:

Definition 3.1. *An ANN is a directed graph from a non-empty set of input units to a set of hidden computing units and ending in a non-empty set of output computing units, such that no parallel links exists between units of the same set.*

A network is functionally divided into three layers: the input layer, a hidden layer and the output layer, each harboring the corresponding set of elementary units. The units in the input layer receive their inputs from some input matrix and pass their information on to the next layer, they do not compute anything themselves. The units in remaining layers are like the conjunction neuron depicted in Figure 5, which we shall discuss momentarily. Moreover, the remaining units are functionally equivalent, the only difference being a hidden unit passes its output on to either an output unit or a subsequent hidden unit, whereas an output unit computes the output of the network. Therefore, we can refer to both a hidden and an output unit as a 'computing unit' (CU), which we define as follows:

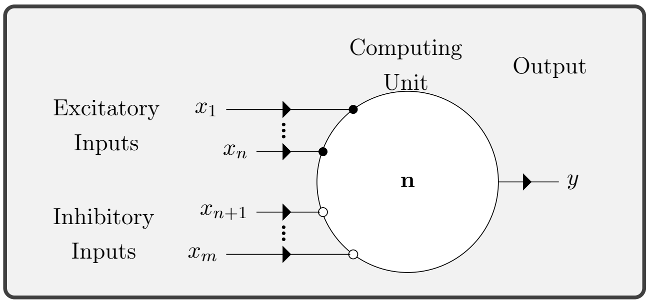


Figure 5: McCulloch and Pitts neuron capable of representing a conjunction of literals, $y = 1$ if only the excitatory inputs $x_1 \wedge x_2 \wedge ...x_n$ and none of its inhibitory inputs$x_{n+1} \wedge x_{n+2} \wedge ...x_m$ are active. Adapted from Pospichal & Kvasnicka, 2015 [8]

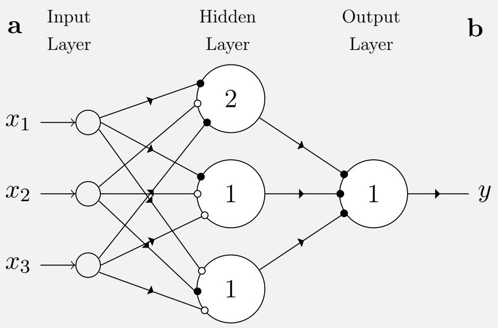
Definition 3.2. *A computing unit is a node in an ANN with the following I/O-behavior:*

1. *Binary signals are delivered to the CU through links connecting to antecedent input units or other antecedent CUs.*
2. *Such signals are either excitatory:* $x_1, x_2 ... x_n = \mathbb{T}$ *or inhibitory:* $\neg x_{n+1}, \neg x_{n+2} ... \neg x_m = \mathbb{F}$.
3. *Incoming signals are weighed according to a valuation function (f) and will output* $y = 1$ *just in case:* $x_1 \wedge x_2 ... \wedge x_n \wedge \neg x_{n+1} \wedge \neg x_{n+2} ... \wedge \neg x_m \geq n$ *and otherwise:* $y = 0$.

If we let excitatory signals be positive literals (i.g. $x_1, x_2 ... x_n$) and inhibitory signals be negative literals (i.g. $\neg x_{n+1}, \neg x_{n+2} ... \neg x_m$), then it is clear to see that a perceptron can simulate a conjunctive clause such as depicted in Figure 5. Since such input sequences only differ in their ordering, We can express Boolean functions in disjunctive normal form (DNF), that is: as a disjunction of conjunctions, e.g. $(x_1 \wedge x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge x_3)$ [9]. To make this more clear, consider the arbitrary Boolean function depicted in Figure 6. We can write the output of our example network as follows:

$$y = f(x_1, x_2, x_3) = (x_1 \wedge \neg x_2 \wedge x_3) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3)$$

Herein each disjunct represents a CU in the hidden layer and each output unit corresponds with a disjunctive clause, i.g.: the top hidden node corresponds to the disjunct: $(x_1 \wedge \neg x_2 \wedge x_3)$, the middle hidden node corresponds to the disjunct: $(x_1 \wedge \neg x_2 \wedge \neg x_3)$ and the bottom hidden node corresponds to the disjunct: $(\neg x_1 \wedge x_2 \wedge \neg x_3)$.



b

Row	x_1	x_2	x_3	$y = f(x_1, x_2, x_3)$
(i)	0	0	0	0
(ii)	0	0	1	0
(iii)	0	1	0	1
(iv)	0	1	1	0
(v)	1	0	0	1
(vi)	1	0	1	1
(vii)	1	1	0	0
(viii)	1	1	1	0

Figure 6: An arbitrary Boolean function. (a) A neural network representing the function. (b) The corresponding truth table.

In general it is possible to reduce any network such as the one in Figure 6 to a simpler form by eliminating any tautologies, e.g.:

$$\begin{aligned} y &= f(x_1, x_2, x_3) \\ &= (x_1 \wedge \neg x_2 \wedge x_3) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3) \\ &= (x_1 \wedge \neg x_2 \wedge (x_3 \vee \neg x_3)) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3) \\ &= (x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3) \end{aligned}$$

Which gives us the simplified network depicted in Figure 7-a. This method enables us to construct multilayered neural networks. That is, networks with an input and output layer and at least one hidden layer. Multilayer networks can approximate any Boolean function arbitrarily well. In general, we say that a multilayered neural network is a universal approximator with respect to the domain of Boolean functions. To give an example, consider the XOR problem we discussed earlier. A single layered neural network was incapable of solving this problem due to linear inseparability in input space. However, a network with just one hidden layer can solve it if we express the XOR truth table in DNF: $(\neg x_1 \wedge x_2) \vee (x_1 \wedge x_2)$, see Figure 7-b.

Before we move on to the next section, it should be mentioned that this logical calculus McCulloch and Pitts developed is not a calculus in the traditional sense of the word. That is, a system comprised of a logical grammar and a deductive system. As we have seen, a logical grammar is accounted for, but a deductive system was not formulated in ‘A Logical Calculus of the Ideas Immanent in Nervous Activity’. Instead, they assumed some key features of neuron behavior and then created a system of one-to-one mappings from classes of logical expressions to certain neuron configurations, such that chains of neural events corresponds to logical relations. For a formal definition of a logical calculus see Fitch, 1944 [10].

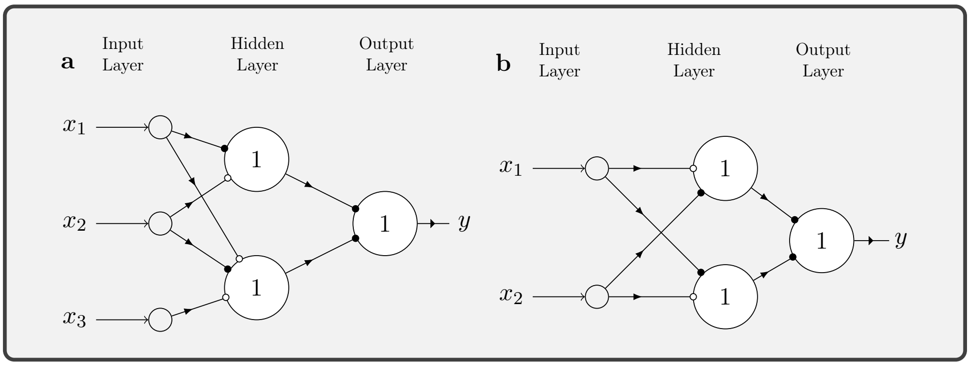


Figure 7: (a) A simplified neural network. (b) A neural network solving the XOR problem. The top hidden node corresponds to the disjunct: $(\neg x_1 \wedge x_2)$, the bottom hidden node corresponds to the disjunct: $(x_1 \wedge \neg x_2)$. Right figure adapted from Pospichal & Kvasnicka, 2015 [8]

.

3.3 Intermediate Comparison

Let us now swing back to our definition of a neural network and its computing units, and compare this to what we know about networks of biological nerve cells. A Receptor cell receives some stimulus which is then passed along down a designated neural network comprised of more typical nerve cells until it reaches an effector cell, which in turn gives some response. In our neural network, a stimulus corresponds to the network input, receptor cells to the input layer, nerve cells to CUs, effectors to the output layer and finally the response corresponds to the output of the network. If we look more closely at the anatomy of a nerve cell, then we can see that the soma and its dendrites correspond to the evaluation of conjunctions of inputs, the axon-hillock corresponds to the valuation function f and the output corresponds to the output of a node.

The synapses, however, are not yet accounted for. These correspond to the weights between subsequent units, which we shall cover in a following section. Additionally, what has not been noted is the fact that biological nerve cells are capable of forming connections with antecedent nerve cells, or in other words: capable of forming cycles. This is called ‘recurrence’ and allows the computations of CUs to be stored for a certain amount of time. Something that does not fall under the umbrella of definition 3.1. Our definition allows propagation of information in one direction only. For this reason such neural networks are commonly referred to as ‘feed-forward’ neural networks. From an ANN point of view, the result computed by the network is stochastic when that network contains cycles. In other words, the connection pattern does not uniquely determine the output, whereas this is the case in a feed-forward network.

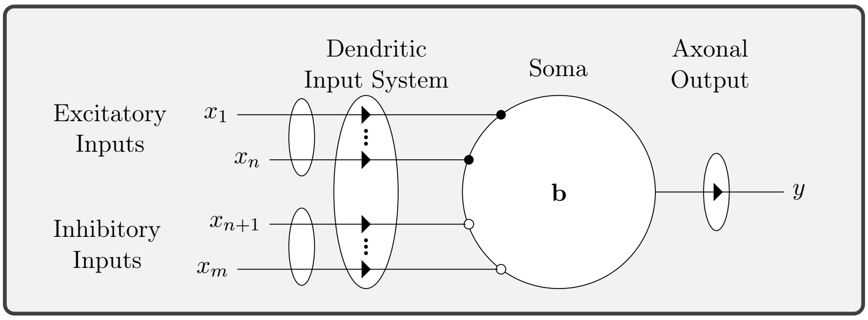


Figure 8: Biological interpretation of an artificial neuron. It receives excitatory and inhibitory inputs x through links which together make up the dendritic system. The node itself represents the soma, which will only propagate a signal y through its axon if its threshold value b is crossed. Adapted from Pospichal & Kvasnicka, 2015 [8].

3.4 The Perceptron

There is another way to think about the computations of an artificial neuron, which is in terms of addition and subtraction. I.e. a CU with threshold value b[3], receives a range of inputs and will only output $y = 1$ if:

$$\sum_{i=1}^{n} x_i - \sum_{j=n+1}^{m} x_j \geq b$$

In other words, the unit will only output 1 if the difference between its excitatory inputs and its inhibitory inputs is at least b. See Figure 8 for an illustration of such an artificial neuron with its biological interpretation. With that said, we can highlight a prominent problem of the McCulloch and Pitts model, which is its scalability. In order to represent more complex functions, more of the same units needed to be interconnected similar to adding logic gates in a circuit. In his seminal paper: '*The perceptron: a probabilistic model for information storage and organization in the brain*', Frank Rosenblatt recognized this downside and proposed a more general model that minimizes network growth through the introduction of real valued weights [11]. This way, smaller networks could simulate the same functions a McCulloch and Pitts neuron could, while greatly reducing its complexity.

The proposed model is known as the 'perceptron'. The most basic feed-forward neural network that is capable of solving those functions whose classification pattern is linearly separable. The model is based on Rosenblatt's understanding of the visual system. Each input represents some information unit presented to the organ of sight; each output represents its perception. For simplicity, let us assume the perceptron classifies objects into two classes, which means the output is a binary value: inactive or 0 for one class, active or 1 for the other. We can then describe the perceptron's input-output behavior as:

$$y = \Theta\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{3.1}$$

---

[3] Also known as the bias term.

In which, w_i a real-valued weight, y is a binary output, and $\Theta(\Phi)$ is the step function defined as follows:

$$\Theta(\Phi) = \begin{cases} 1 & \Phi \geq 0 \\ 0 & \Phi < 0 \end{cases}$$

This way we can understand a network as a function with the weights and biases as parameters. Our example features the step function, but for modern networks there is a plethora of functions that can be used.

3.4.1 The Perceptron Learning Rule

Just like the artificial neurons proposed by McCulloch and Pitts, Rosenblatt initially envisioned his perceptron to be a simplified version of reality. There is a great variety of perceptrons, but the one most commonly referred to is called an alpha perceptron. It distinguishes between three types of nodes: sensory (S), associative (A) and Response (R). The sensory inputs project binary values to some associative or computing units, which in turn then map to an output or response unit. The links between the S and A units are fixed, while the links between A and R units are adaptive and selected according to some random distribution. The idea is to have some input-output pairs that you want the perceptron to classify without error. If it cannot do this correctly, then we need to continuously change the adaptive weights until it does. Those inputs or features that are important to the pattern should have a heavier weight associated with them then those that are of less importance.

The learning rule was originally developed by Rosenblatt, which he called the ‘reinforcement system’. Rosenblatt’s learning rule was conceived to mimic Hebbian learning:

> ‘Hebb’s philosophy of approach seems close to our own, and his work has been a source of inspiration for much of what has been proposed here.’ [11]

He differentiated between three systems. For illustrative purposes, two of these are: an α-system and a γ-system. In the former system just the active links are changed, while inactive links remain unaltered. The latter system changes the weights of active and inactive links proportionally with respect to each other, meaning the numerical value of the total weight remains unchanged. In modern terms we can formalize the perceptron learning rule as a sequence of steps which are as follows:

(i) Choose a learning rate $\eta > 0$.

(ii) Initialize the weights and the bias to some random values.

(iii) Present an input pattern to the perceptron and calculate the output.

(iv) If the output matches the target value t, terminate.

(v) Otherwise, update the weights according to: $w_i = w_i + \eta(t - y)x_i$

(vi) Update the bias according to: $b = b + \eta(t - y)$

(vii) Repeat steps *iv-vii*.

The learning rate is a constant value, which is carefully picked by the engineer or programmer based on his knowledge of the problem domain. In essence it determines the rate at which we adjust the weights. The general rule is that a small learning rate will eventually find good weights but converges slowly to a local or global minimum, while a large learning rate may show 'bouncing' between points in weight space; ignoring the minimum [12]. It is however unnecessary to have a separate learning rule for the bias, since the bias is equivalent to having another weight with an input link that always has an activity of 1. Therefore, we can treat the bias as a weight w_0 and treat all weights as a vector $\mathbf{w}$ and all inputs as a separate transposed vector $\mathbf{x}^{\mathrm{T}}$, with an added $x_0 = 1$ [12]. When we say we want a network to classify its inputs correctly, we mean we want to minimize the difference between a networks current output y and the target value t. We call this difference the error, E, and it is represented as a quadratic function such that an algorithm like '*steepest descent*' might find the point in weight space where it is smallest, be it locally or globally.

In 1957 Kolmogorov showed that there exists a mapping from n dimensional continuous functions to functions of a single dimension [13]. In terms of neural networks this means that any three layer feed-forward network is computationally universal with respect to continuous functions. However Braunn and Griebel noted that this proof was not constructive, i.e. there is no straightforward way to establish what weights among hidden nodes are needed in order to represent the function [14]. Backpropagation is an algorithm that solves this problem to a certain extent by taking the error and propagating it backwards through the network. For this, the nodes need to use a differentiable function such as the sigmoid function instead of the step function. For a more rigorous mathematical discussion about the error and backpropagation we refer to Russel & Norvig, 2010 [1].

3.5 Concluding Comparison

We have already seen couple of ways in which artificial neurons do not do justice to their biological counterparts. However more is left to be discussed. First and foremost, we have seen that many kinds of neurons make up the central nervous system. Artificial neurons only represent a basic neuron. Additionally, biological neurons change their behavior based on a diverse range of incoming neurotransmitters and based on a variety of hormones, each having its own unique effect. Moreover, the amount of hormones and neurotransmitters also varies greatly. Artificial neurons only distinguish between just two inputs: excitatory and inhibitory. A vast underrepresentation of the biological paradigm. We have also discussed learning based on a target value. It is however highly unlikely that biological neurons have target values at all. We can add to this that there are no mishaps in the firing of artificial neurons, which seems improbable for real neurons. A very important abstraction is that artificial neurons do not account for different firing rates of nerve cells, they assume a fixed value for this. Neither do they account for the asynchronous firing we see in biological networks. Our current models assume linear summation. It is however probable that real neurons can be more closely imitated by non-linear summations. In chapter 5 one such kind will be briefly discussed.

Chapter 4

Visual Processing and Neural hierarchy

In this section we discuss the visual system, with a specific focus on the early stages of visual processing. We intent to highlight its hierarchical structure and the way in which representation of the input images are increasing in complexity as you move up in the hierarchy. This section is specifically aimed at capturing the interplay of specific groups of neurons, for this will form the basis of a different type of neural network, known as a convolutional neural network, which we shall cover in the following chapter.

4.1 The Retina

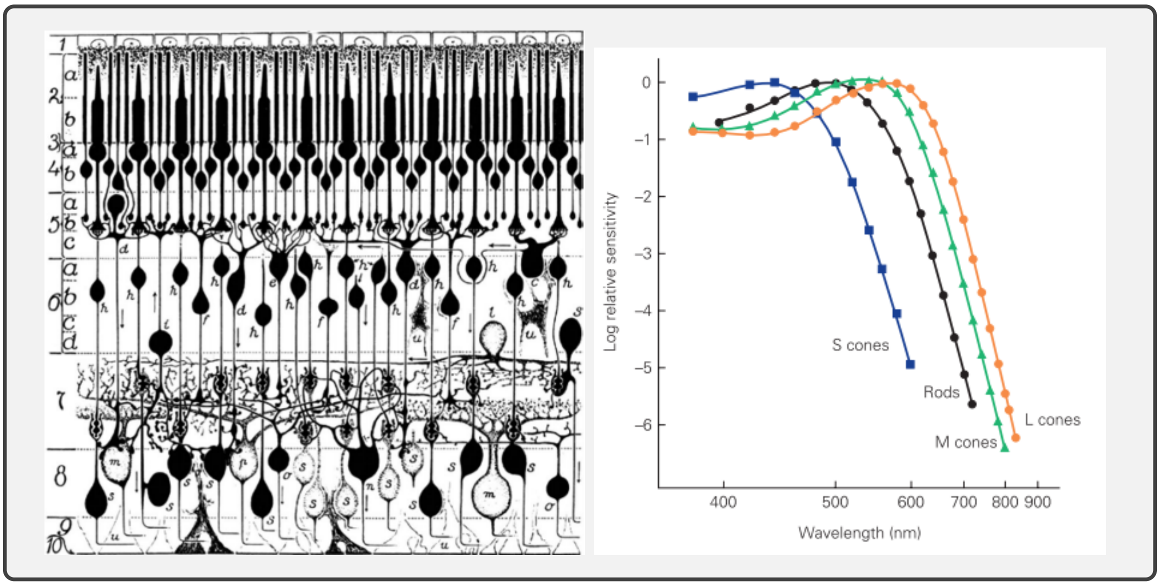


Figure 9: Left: Structural layering of the primate retina. The top layers, 2a through 5b, harbor the rods (thin) and cones (thick). Ganglion cells are situated at the bottom under labels: m,n,o,p and s. As in Polnyak, 1941 [15]. Right: light absorption properties across a range of wavelengths for S, M and L-cones and rods. As cited in Kandel et al., 2012 [3].

At the back of the eyes is a coating of nerve tissue called the retina. It consists of a variety of nerve cells layered on top of each other, see Figure 9. Starting at the outer most layer, closest to the eyeball, are the photoreceptors that absorb incoming light and convert them into electrical signals. These signals are then propagated to other neurons in layers down the line, chief among which are the ganglion cells, whose axons form the first part of the connection between the retina and the brain areas associated with higher level processing. This connection is called the optic nerve. Photoreceptors in primates are divided into two functionally different types: rods and cones. Rods have a slimmer structure than cones and they are extremely sensitive to low light intensities, however their response quickly becomes saturated as light intensity increases. Cones, on the other hand, require more light before they become active. Therefore, rods are primarily responsible for our vision at night, while cones make up our vision during daylight hours. Primates have several different types of cones, each sensitive to different wavelengths. The cones are subdivided into: S-cones, which respond to short wavelengths, M-cones which respond to medium wavelengths and L-cones which respond to long wavelengths. These wavelengths correspond to blue, green and red light respectively, see Figure 9. Rods on the other hand cannot cause color percepts. The center of vision or central fovea has a high cone density, resulting in high visual acuity in that area. As you move away from the center of vision, cone density quickly tapers off while rod density increases; first peaking before gradually decreasing again. Since there are less axons in the optic nerve then there are photoreceptors suggests that there is some form low-level data compression in this area of the nervous system.

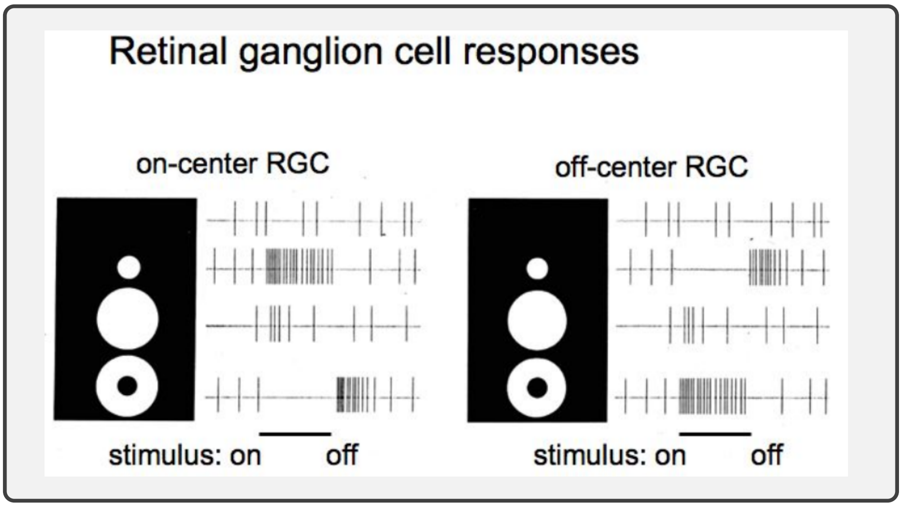


Figure 10: Retinal ganglion cell (RGC) responses from different stimuli in their receptive fields. The bar density corresponds to the firing rate of the cell when presented with the stimulus on its left. The higher the bar density, the higher the firing rate. [Online image] Retrieval date: July 4, 2018 from http://www.cns.nyu.edu/ david/courses/perception/lecturenotes/ganglion/-ganglion.html

The area on the retina to which a specific neuron is receptive is called the *receptive field*. For illustrative purposes we will discuss the receptive field of a ganglion cell, which is characterized by a center-surround (or mexican hat) profile. That is, the center is an excitatory zone, while the periphery is an inhibitory zone. Ganglion cells are also divided into two groups based on their change in firing rate when illumination changes. We distinguish between ON cells that increase their firing when illumination is increased and OFF cells that increase their firing when illumination is decreased. For example, the firing rate of ON cells peaks when presented with a white pattern (in its center) on a black background (the periphery), while the firing rate of

OFF cells peaks when presented with a black pattern on a white background. Patterns that cover both the periphery and the center elicit a much weaker response than when the pattern covers only the center, suggesting that processing at this stage locally enhances spatial contrast boundaries, see Figure 10. The visual system is specialized in detecting temporal changes, or '*features*', rather than the perception of a constant image. Eye saccades are one way to ensure that we keep on detecting these changes and it prevents the image from fading. It should be noted that the receptive fields of different ganglion cells overlap to a certain degree. Therefore a cluster of ganglion cells produce an output that corresponds with a map of those features that occur in multiple regions. We call this map a '*feature map*'. At this stage of visual processing the contrast feature map is not the only existing map, we also distinguish between: lightness, brightness, color, depth and motion. The color feature map, for example, is jointly formed with the contrast feature map, but we shall not cover this here. The important thing to take away is that these filtering operations are not homogeneous across the retina and they create a higher level representation of the retinal image. For a more detailed discussion of feature maps at the early stages of visual processing we refer to Spillman, 2014 [16].

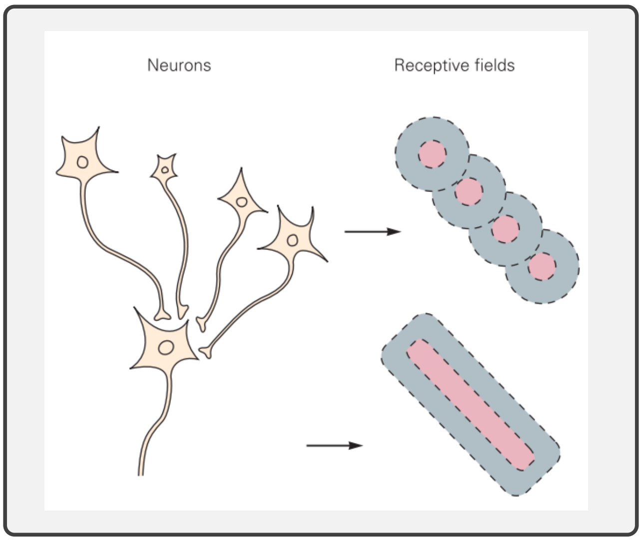


Figure 11: Depicted are a group of simple cells connected to a complex cell. Aligned simple cells have overlapping and aligned receptive fields. The complex cell that receives their input, has a receptive field that encompasses those of simple cells. As in Kandel, 2015 [3].

4.2 Hierarchical Visual Processing

Different types of retinal ganglion cells pass their signals to different cell layers in the dorsal lateral (DLGN) geniculate nucleus of the thalamus. The first type is the magnocellular layer, responsible for processing changes in stimuli. The second type, called the parvocellular layer, is responsible for processing information about the brightness, the color and the spatial detail of the image. They are not responsive to orientation of a stimulus. These two cell layers relay their information to the primary visual cortex (V1). Hubel and Wiesel showed through single-unit recordings in cats that in the visual cortex there is a group of neurons whose firing rate peaks when presented with a specific orientation of a bar [17]. Other orientations showed a much weaker or even absent response. Seeing that these cells receive their inputs from cells in the DLGN whose receptive fields are aligned and partially overlap, we can extrapolate that the preference of simple cells derives from the spatial alignment of DLGN neurons. Hence we say the simple cells of V1 carry an increased level of representation over these DLGN neurons.

In turn, simple cells connect to complex cells which are responsive, not only to orientation (albeit to lesser extent), but to movement as well. Complex cells have a larger receptive field than simple cells have and they are, contrary to simple cells, indifferent to the spatial position of the stimulus. Hubel and Wiesel also showed that the receptive field of complex neurons encompasses those of simple cells. Moreover, complex cells receive inputs from clusters of simple cells and as soon as one of them fires (receives a stimulus with the right orientation and spatial alignment), the complex cell fires [17]. To put this in terms of the logical neurons in the McCulloch and Pitts network: simple cells correspond to nodes simulating the AND function, since they only fire if all antecedent neurons are sufficiently spatially arranged; while complex cells correspond to nodes simulating the OR function, since they fire whenever one or more of its antecedent simple cells fires. On a final note, when a stimulus is presented to a specific area on the retina, adjacent neurons that consequently become active illicit a response from adjacent neurons in the DLGN. This principle then extends to the visual cortex, suggesting the existence of a topographical map. This map ensures that the spatial layout of the image is preserved after many layers of computation. See Figure 11 for the receptive fields of simple and complex cells.

Chapter 5

Convolutional Neural Networks

In this section we again cover a transition from the realm of biology to the realm of the artificial. We have seen that early parts of the visual system represent small features of the image that is presented to the retina. As you move away from the eyes to the cortex, an increasing amount of those features will be represented by single cells. For instance, cones in the retina represent only a couple of photons, cells in the primary visual cortex might represent an edge and cells in the fusiform face area can respond to entire faces. In this section we will discuss how we can model such a hierarchical system with a special kind artificial neural networks: convolutional neural networks (CNNs).

5.1 Layering

One of the paradigms CNNs are used for is the classification of objects in images. A naturally hard task since there are many disparities in the image such as occlusion. A regular ANN will have a hard time classifying objects correctly as soon as the viewing angle is changed. To solve this problem we require a network that is insensitive to certain features, while being responsive to others. In this respect CNNs seek to improve on their ANN counterpart. As we have seen in chapter 4, simple cells and complex cells correspond to artificial nodes simulating conjunctions and disjunctions respectively. Recall that conjunction nodes only fire when all of their inputs are active. In other words: if all of the features are accounted for. Also recall that disjunction nodes fire when one or more of its inputs are active. What we have here is one way to discriminate between those the network needs to be insensitive to and those that it needs to be responsive to. If we alternate layers of these types of nodes in a network we will eventually converge to nodes that recognizes objects.

Kunihiko Fukushima was the first to tackle visual pattern recognition based on Hubel and Wiesel's hierarchical model of simple and complex cells. His model, called the neocognitron, is a form of feed-forward neural nets which employed the structure of alternating layers [18]. Follow-up work by Yann LeCun showed for handwritten digits the backpropagation and error minimization techniques could be used to improve on classification accuracy [19].

5.2 The Operations

A CNN is characterized by a specific set of operations that is carried out in sequence. All convolutional neural networks carry out the same sequence, but may vary slightly in their interpretation of the operation. The operations are as follows:

1. *Filtering*
2. *Thresholding*
3. *Pooling*
4. *Normalization*

When a two-dimensional image is presented to a CNN, the first layer it encounters is a filtering layer. An image is nothing more than a grid, or matrix, of pixels; each pixel having its own RGB-value associated with it. A smaller grid, or set of weights also known as '*filter*', is then used to slide over the original image, calculating the dot product between the original image and the filter at each step of the way. This yields an output value for every area the filter 'scans'. The biological counterpart of a filter is the receptive field. Each filter, then, creates a map of a specific feature. Considering each image has a color value, we will thus end up with three different matrices: one for green, one for red and one for blue. These maps are then fed as inputs to the subsequent layer. Since filters are sets of weights, we can see that an optimal filter can be learned by backpropagation. As for the step size, there is some freedom in the magnitude by which the image can slide. This magnitude is known as 'stride'. Typically the filtering operation is followed by a non-linear activation as opposed to what we have seen with the McCulloch and Pitts model. The most commonly used non-linear activation function is the Rectified Linear Unit, or ReLu for short. This function has the property that it transforms all negative values into zero. In general terms this means that the activation remains zero if the feature is not detected. considers This ensures that the feature map is only passed on to the next layer if it reaches a certain threshold.

Pooling, as it is most frequently used, is nothing but a max function over specific areas of the input feature map, although the literature indicates that the mean and median can also be used. It takes the highest value of each respective area and joins them to form a new feature map. The goal of this operation is to ensure that only the most important features are carried over to the next stage of processing. This way less computations are necessary than when we pass on the entire feature map. For instance, if we have a grid with a dimensionality of $4x4$ and we take the the top left area of size $2x2$ from which we extract the highest value before we move on to the top right area of size $2x2$. From this area we also take the highest value and we continue this process until we have covered the entire $4x4$ grid. We then create a new, smaller, grid of size $2x2$ which only has the highest values, or most important features, of the original grid. These alternating filtering and pooling layers thus create high level representations of the original image. This representation is then reduced in dimensionality. The new representation is then passed on to a structure that is just like the ones we have seen in regular ANNs, i.e. a fully connected structure. This means that there are connections from each set of high level features to a classification output node.

These networks might be a lot more computationally capable than the McCulloch and Pitts network is. But still some aspects of biological networks are missed. For instance, neurons in the visual cortex might die, while artificial neurons in these networks do not drop out. Additionally there exists top down connections from the cortex to the DLGN, sometimes in even greater amounts than the forward connections [5]. These models are not capable of mimicking this aspect. Furthermore, the ReLu operation has no upperbound, whereas biological neurons do have a maximum firing rate.

Chapter 6

Discussion

Our goal was to highlight which abstractions artificial neural network models make of their biological counterparts. First we outlined the properties of a nerve cell in chapter 2, after which we discussed the inception of the first ANN models, their workings and the assumptions they were based on. We have seen that the first shortcomings directly resulted from the assumptions made by McCulloch and Pitts. Some of these were solved by the model Rosenblatt put forward, such as the weights that could represent the strength of a connection. However, many aspects of the biological paradigm, have still been missed. Future ANN models should seek to include some of these missed features, such that neuroscientists might gain insight into what their effects are when introduced in isolation into existing models, cateris paribus.

Furthermore, we have seen that processing in the visual system works hierarchically. We have seen that we can ground a special kind of neural networks on the principles of the early stages of this system, called convolutional neural networks. While these resemble a the biological paradigm more closely than a McCulloch and Pitts network, still certain shortcomings persist. For instance, filters in artificial neural networks are all homogeneous, whereas different ganglion cells in the retina make up for a variety of different filters.

Acknowledgements

This has been a tremendously difficult year for me, seeing that four family members have come to pass. In light of this I would like to thank dr. Janneke van Lith, dr. Ruben van Doorn and dr. Rosalie Iemhoff for their extensive support and guidance. Special thanks goes out to my thesis advisor, dr. Ben Harvey who has been more than willing to cooperate and to help see this thesis come to its fruition. It has been an honor to work under his supervision. Last, but not least, I would like to thank BSc. Bryan Cardenas and BSc. Vivian van Ooijen for reviewing this thesis.

Utrecht, 2018

Sander Emile Bos

Bibliography

[1] S. Russell and P. Norvig, *Artificial Intelligence_ A Modern Approach (3rd Edition)*, 3rd ed. Cranbury, NJ, USA: Pearson Education, 2010.

[2] O. D. Creutzfeldt, “Generality of the functional structure of the neocortex.” *Die Naturwissenschaften*, vol. 64, no. 10, pp. 507–517, 1977.

[3] E. R. Kandel, J. H. Schwartz, T. M. Jessel, S. A. Siegelbaum, and A. Hudspeth, *Principles of Neural Science . Fifth Edition. Edited by Eric R. Kandel, James H. Schwartz, Thomas M. Jessell, Steven A. Siegelbaum, and A. J. Hudspeth; Art Editor:, Sarah Mack. New York: McGraw-Hill. $135.00. l + 1709 p.; ill.; index. ISBN: 978-0-07-13901*, 5th ed., S. Mack, Ed. McGraw-Hill Education / Medical, 2012.

[4] H. Haug, “History of neuromorphometry,” *Journal of Neuroscience Methods*, vol. 18, no. 1-2, pp. 1–17, 1986.

[5] M. A. Arbib, “The handbook of brain theory and neural networks,” *Journal of the Indian Institute of Science*, p. 1290, 2003.

[6] G. Piccinini, “The first computational theory of mind and brain: A close look at McCulloch and Pitts's "logical calculus of ideas immanent in nervous activity",” *Synthese*, vol. 141, no. 2, pp. 175–215, 2004.

[7] W. S. McCulloch and W. Pitts, “A Logical Calculus of the Idea Immanent in Nervous Activity,” *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943. [Online]. Available: http://www.cse.chalmers.se/~coquand/AUTOMATA/mcp.pdf

[8] J. Pospichal and V. Kvasnicka, “70th anniversary of publication: Warren McCulloch & walter pitts - a logical calculus of the ideas immanent in nervous activity,” *Advances in Intelligent Systems and Computing*, vol. 316, pp. 1–10, 2015.

[9] S. C. Kleene, “Representation of Events in Nerve Nets and Finite Automata,” *Rand Project Air Force Santa Monica Ca*, no. June, p. 103, 1951. [Online]. Available: http://courses.cs.bilgi.edu.tr/file.php/291/Kleen1956-Representation_of_events_in_nerve_nets_and_finite_automata.pdf

[10] F. B. Fitch, “A minimum calculus for logic,” *The Journal of Symbolic Logic*, vol. 9, no. 04, pp. 89–94, 12 1944. [Online]. Available: https://www.cambridge.org/core/product/identifier/S0022481200061879/type/journal_article

[11] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in . . . ,” *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958. [Online]. Available: http://psycnet.apa.org/journals/rev/65/6/386.pdf%5Cnpapers://c53d1644-cd41-40df-912d-ee195b4a4c2b/Paper/p15420

[12] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, “Learning From Data,” *Learning from Data*, vol. 21, no. 4, pp. 479–481, 2010.

[13] A. N. Kolmogorov and T. b. V. Tikhomirov, “On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables,” *Doklady Akademii Nauk SSSR*, vol. 114, pp. 369–373, 1957.

[14] J. Braun and M. Griebel, “On a constructive proof of Kolmogorov’s superposition theorem,” *Constructive Approximation*, vol. 30, no. 3, pp. 653–675, 2009.

[15] S.L.Polyak, *The Retina.* Chicago, IL: University of Chicago Press, 1941.

[16] L. Spillmann, “Receptive fields of visual neurons: The early years,” *Perception*, vol. 43, no. 11, pp. 1145–1176, 2014.

[17] D. H. T. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,” *The Journal of Physiology*, vol. 160, p. 106–154, 1962.

[18] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.