Angewandte Computer- und Biowissenschaften

HOCHSCHULE MITTWEIDA
University of Applied Sciences

BACHELOR THESIS

Ms
Tabea Heusel

Neuron models: Convergence and Stability Analyses of Hebb, Oja and BCM learning rules

2022

Faculty of **Applied Computer and Life Sciences**

BACHELOR THESIS

Neuron models: Convergence and Stability Analyses of Hebb, Oja and BCM learning rules

Author:
Tabea Heusel

Study Programme:
Applied Mathematics

Seminar Group:
MA18w1-B

First Referee:
Prof. Dr. rer. nat. habil. Thomas Villmann

Second Referee:
Dr. rer. nat. David Nebel

Mittweida, August 2022

Bibliographic Information

Heusel, Tabea: Neuron models: Convergence and Stability Analyses of Hebb, Oja and BCM learning rules, 45 pages, 3 figures, Hochschule Mittweida, University of Applied Sciences, Faculty of Applied Computer and Life Sciences

Bachelor Thesis, 2022

Abstract

This Bachelor thesis investigates the learning rules of the Hebbian, Oja and BCM neuron models for their convergence to, and the stability of, the fixed points. Existing research is presented in a structured manner using consistent notation. Hebbian learning is neither convergent nor stable. Oja learning converges to a stable fixed point, which is the eigenvector corresponding to the largest eigenvalue of the covariance matrix of the input data. BCM learning converges to a fixed point which is stable, when assuming a discrete distribution of orthogonal inputs that occur with equal probability. Hebbian learning can therefore not be used in further applications, where convergence to a stable fixed point is required. Furthermore, this Bachelor thesis came to the conclusion that determining the fixed points of the BCM learning rule explicitly involves extensive calculation and other methods for verifying the stability of possible fixed points should be considered.

I. Contents

II. List of Figures

III. List of Tables

IV. Preface

In order to successfully follow this Bachelor thesis basic knowledge of analysis, linear algebra, stochastics, as well as dynamical systems is required and assumed.

I would like to thank Prof. Thomas Villmann for providing me with the challenging and purposeful task of researching this topic and for the support and guidance during the writing of this thesis.

I look forward to using the knowledge gained throughout my bachelor's studies in my master's studies.

1 Introduction

A central problem in pattern recognition and artificial intelligence is how learning occurs. When investigating this issue biological neurons are mathematically modeled. While modeled neurons take the biological neurons in the brain as an inspiration, "they are generally not designed to be realistic models of brain function." [5] However, the basic principles of the biological neuron are used in artificial neurons. Biologically, a neuron receives input signals from other neurons via dendrites. Depending on whether the received signals are inhibitory or excitatory, and depending on the signal's strength, the neuron fires a signal towards the axon terminals or it does not. If the neuron fires, the signal gets transmitted towards the axon terminal which is then connected to another neuron's dendrites where the same process repeats itself. This process is illustrated in figure 1. [11]

In model neurons the learning is defined through the changing synaptic weights between distinct neurons, according to their specific learning rules. How to change the weights and thus defining a learning rule is the central problem of learning. The goal is to extract sensible information and have the data organize itself. Moreover, the neuron should function as a memory that remembers relevant information and forgets irrelevant information. [12]

In unsupervised learning, the focus of this Bachelor thesis, the change of the synaptic weights is solely based on the input to the neuron and the output of the neuron. New inputs that are more strongly correlated with the information contained in the neuron, cause larger output values than inputs that are weakly correlated. [12]

This Bachelor thesis creates a starting point to the hypothesis that the way the neuron is mathematically modeled within the neuromorphic hardware can determine the speed of the learning.

Neuromorphic hardware is a type of hardware that can be put in computer chips. The hardware is made up of interconnected silicon neurons. In common neural network models programmed to execute on a multi-use central processing unit (CPU), learning is done by stacking in the random access memory (RAM) in the computer. This is usually slow which is why graphic processing units with the possibility of parallelisation are used more frequently when training neural network models. Neuromorphic hardware takes this to another level. The learning is done through the neurons, physically contained in the hardware, which makes the learning process comparatively fast. [2]

However, it creates many restrictions for the model that is to be programmed. For example a chip with 256 interconnected neurons has a physically predefined number of neurons, that cannot be changed through programming. Nonetheless, a neuromorphic

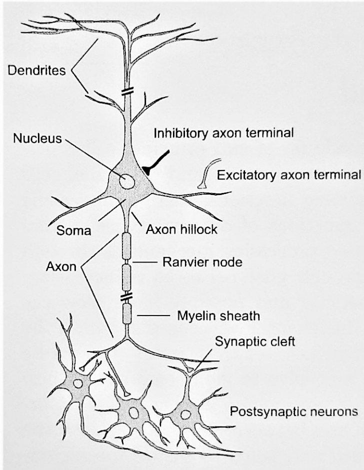

Figure 1.1: The anatomy of a neuron and its postsynaptic neurons. Taken from [11].

chip with silicon neurons already contained within it, and a well programmed model that fits this hardware is much more efficient than the usual way of programming on a multi-use CPU. [2]

Within this thesis, three ways of modeling a neuron are considered and explained: the neuron model according to the Hebbian principle, the Oja neuron model and the Bienenstock-Cooper-Munro (BCM) neuron model. Each of the neuron models has an associated learning rule. When wanting to implement a neuron model it is important to know that the learning rule will converge to a fixed point and that this fixed point is stable. This is why for each neuron model the convergence and stability are investigated. Both Oja and BCM are based on and extend the Hebbian model of neuron learning, which is why the Hebbian model will be discussed first.

1.1 Definitions and Notation

Throughout this Bachelor thesis a specific notation is used. This is done in order to create a common baseline for all neuron models and learning rules. When a theory describes the same parameter, it is immediately visible through the notation and terminology and parameter representation. As an example, the (synaptic) weights $\mathbf{w}$ will be referred to as such throughout the entire text and not as junction strengths ξ as they are called in [10].

Furthermore, the following general mathematical notation is valid throughout,

a	scalar, unless specified otherwise
$\mathbf{a}$	vector, unless specified otherwise
$\mathbf{A}$	matrix, unless specified otherwise
$\mathbf{0}$	null vector
$\langle \mathbf{a}, \mathbf{b} \rangle$	scalar product between $\mathbf{a}$ and $\mathbf{b}$
$\|\|\mathbf{a}\|\|$	euclidean norm of $\mathbf{a}$, unless specified otherwise
$\mathbb{R}^n$	real numbers, n-dimensional
$\mathbb{E}_X[\mathbf{a}]$	the expected value of $\mathbf{a}$
$\sum$	sum of the following parameters
$\mathbf{a}^T$ $\mathbf{A}^T$	transpose of the vector $\mathbf{a}$ / matrix $\mathbf{A}$
$\mathbf{F}(\dots)$	vector-valued function
$f(\dots)$	function
$\forall\, a$	the following statement is valid for all a
$\exists\, a$	for the following statement exists an a
$a \in b$	a is an element of b
$a \ll b$	a is much lower than b
$a \lll b$	a is very much lower than b
$a \to b$	a tends to b
$a \mapsto b$	a maps to b
$A \subset B$	A is a subset of B
$a \implies b$	a implies b

and the following notation records definitions of parameters specific to the topic of this thesis.

x_i	i^{th} input, presynaptic activity
$\mathbf{x}$	vectorized input, $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$
X	random variable associated with $\mathbf{x}$, not a matrix
w_i	i^{th} synaptic weight
$\mathbf{w}$	vectorized synaptic weight, $\mathbf{w} = (w_1, w_2, \dots, w_n)^T$
y	output, postsynaptic activity
η	learning rate
$\mathbf{w}(t)$	vector $\mathbf{w}$ at time step t
$\frac{\partial \mathbf{w}}{\partial t}$	change of $\mathbf{w}$ from one time increment to another
$\mathbf{I}$	Identity matrix
$\mathbf{C}$	covariance matrix, if not specified otherwise
$\mathbf{w}^*$	fixed point of the dynamical system
$\mathbf{e}_i$	i^{th} eigenvector, unless specified otherwise
λ_i	i^{th} eigenvalue, unless specified otherwise
θ	modification threshold of BCM learning rule

2 The Hebbian principle

The Hebbian principle was formulated by Hebb in 1949 to attempt to explain how learning occures within the brain. Hebb states that "when an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency as one of the cells firing B, is increased". [7] This leads to a dynamic strengthening or weakening of synapses over time, according to the input received by neuron B and sent by neuron A.

2.1 Simple neuron model

The signals that the neuron receives via the dendrites are modeled through an input vector $\mathbf{x}(t) \in \mathbb{R}^n$ that contains all inputs to the neuron at a specific time t. In the Hebbian model it is assumed that $x_i > 0 \ \forall i$, in order to remain biologically accurate. The output of the cell is given by the value $y \in \mathbb{R}_+$. This is then passed on to the next neurons. The artificial neuron's output is determined by a weighted sum of the input signals. The weights are contained in the weight vector $\mathbf{w} \in \mathbb{R}^n_+$. The following equation models the Hebbian principle mathematically according to the definitions given above.

$$y(t) = \sum_{i=1}^{n} w_i(t) x_i(t) \tag{2.1}$$

where n is the number of inputs to the neuron. [10] For a visualization of the equation see figure 2.1.

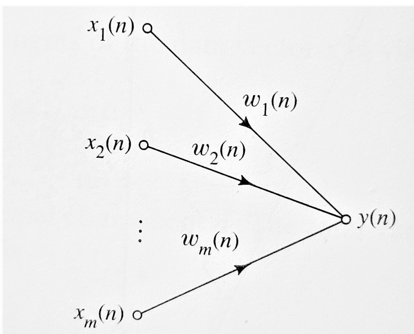

Figure 2.1: The Hebbian principle modeled mathematically. In this neuron model m is the number of inputs and n represents the parameter time step. Taken from [6].

2.2 Hebbian Learning

Given a steady stream of input vectors $\mathbf{x} \in \mathbb{R}^n_+$, where each input is received by the neuron at a separate time t, the learning of the neuron is determined by the changing weights $\mathbf{w} \in \mathbb{R}^n_+$. The neuron model can be rewritten in vectorized form as:

$$y(t) =< \langle \mathbf{w}(t), \mathbf{x}(t) \rangle = \langle \mathbf{x}(t), \mathbf{w}(t) \rangle \tag{2.2}$$

The final equality only holds if $\mathbf{w}$ and $\mathbf{x}$ are both real vectors. The Hebbian learning principle is consequently parametrized as

$$\frac{\partial \mathbf{w}}{\partial t} = \eta y \mathbf{x}(t) \ \ , t = 1, \ldots, m \tag{2.3}$$

where $\frac{\partial \mathbf{w}}{\partial t}$ denotes the change of the weight $\mathbf{w}$ at time t and m is the maximum time increment. The coefficient η is the learning rate, for which $0 < \eta \ll 1$ applies. It may be set as desired within these bounds, but it is typically chosen to be small or even decaying with time. This way the learning occurs slowly and $\mathbf{w}(t)$ only changes incrementally at one time t. The weight $\mathbf{w}$ is therefore determined by the equation

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta\, \mathbf{x}(t)\, \phi_H(y, \theta) = \mathbf{w}(t) + \eta y \mathbf{x}(t) \tag{2.4}$$

In this instance $\theta = 0$ and $\phi_H(y, \theta) = y$ meaning the function solely depends on y and there is no modification threshold. Nonetheless, the notation was chosen to make the Hebbian learning rule more easily comparable to the BCM learning rule presented in chapter 4. [12] [6]

2.3 The problem with Hebbian learning

According to the Hebbian principle the changes of the vector $\mathbf{w}$ at each time increment add up over time and settle on some values. [6] However, since the update will always be positive due to

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta \mathbf{x}(t) y = \mathbf{w}(t)(1 + \eta \mathbf{x}(t) \mathbf{x}(t)^T \mathbf{w}(t)) \tag{2.5}$$

where the vector $\mathbf{w}$ will increase in magnitude at every time step t. Therefore the weights do not stabilize as $t \to \infty$. [4] Especially for sequences of $\mathbf{x}(t)$, where $\mathbf{x}(t) = \mathbf{x}, \forall t$ it is clear that $||\mathbf{w}(t)|| \to \infty$ for $t \to \infty$. [10] As there is no term that causes synaptic decrease the synapses saturate leading to no information being stored in this neuron. [1]

To see that what is written above is indeed the case, the stability and convergence of Hebbian learning rule must be asymptotically analyzed. The following definitions, as presented in [6], are necessary for the analysis.

Definition 2.1:
A constant vector $\mathbf{w}^* \in \mathbb{R}^n$ is said to be a *fixed point* of the autonomous dynamical system $\frac{d}{dt}\mathbf{w}(t) = \mathbf{F}(\mathbf{w})$ if

$$\mathbf{F}(\mathbf{w}^*) = \mathbf{0}$$

where $\mathbf{0}$ is the null vector and $\mathbf{F} : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a differentiable vector valued function. The velocity vector $\frac{d\mathbf{w}}{dt}$ vanishes at the fixed point $\mathbf{w}^*$ and therefore the constant function $\mathbf{w}(t) = \mathbf{w}^*$ is a solution of $\frac{d}{dt}\mathbf{w}(t) = \mathbf{F}(\mathbf{w})$.

Definition 2.2:
A fixed point $\mathbf{w}^*$ of an autonomous dynamical system $\frac{d}{dt}\mathbf{w}(t) = \mathbf{F}(\mathbf{w})$ is *stable* if

$$\forall \varepsilon > 0 : \exists \delta > 0 : ||\mathbf{w}(0) - \mathbf{w}^*|| < \delta \implies ||\mathbf{w}(t) - \mathbf{w}^*|| < \varepsilon,\ \forall t > 0$$

Definition 2.3:
A fixed point $\mathbf{w}$ is *convergent* if

$$\exists\ \delta > 0 :\ ||\mathbf{w}(0) - \mathbf{w}^*|| < \delta \implies \mathbf{w}(t) \to \mathbf{w}^* \text{ for } t \to \infty$$

Definition 2.4:
A fixed point $\mathbf{w}^*$ is *asymptotically stable* if it is both stable and convergent. It is *globally asymptotically stable* if it is stable and all possible sequences of $\mathbf{w}(t)$ converge to a unique $\mathbf{w}^*$ as $t \to \infty$. The system will therefore ultimately settle down to the fixed point $\mathbf{w}^*$ for any choice of initial conditions.

Therefore, we look for those points for which holds that

$$\frac{\partial \mathbf{w}}{\partial t} = \mathbf{0} \tag{2.6}$$

In order to asymptotically analyze the stochastic method 2.5, it is assumed that the weights change significantly slower than new data is presented. [12] This is achieved via a small constant learning rate η. Together with further reasonable conditions it can be shown that the trajectories created in the stochastic method converge on compact input sets in probability in the direction of the trajectories that are created in the averaged dynamic. [8] [12]

Due to this, it is reasonable to average over the changes. Instead of calculating the new weight $\mathbf{w}(t+1)$ at every step only the change of the weight is investigated at each time step t. We examine the learning over a time average, such that

$$\frac{\partial \mathbf{w}}{\partial t} = \eta \sum_i w_{ij} \mathbb{E}[x_i x_j] = \eta \mathbf{C}\mathbf{w} \tag{2.7}$$

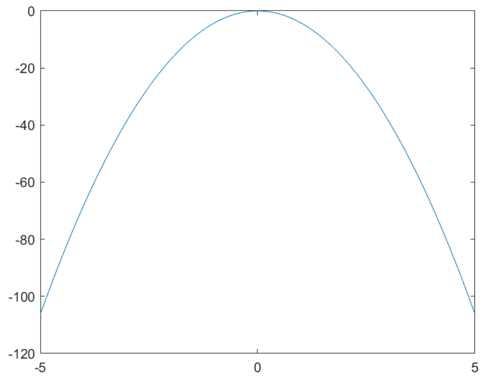

Figure 2.2: Function 2.9 for $dim(\mathbf{w}) = 1$.

where $\mathbf{C}$ is the covariance matrix. Therefore, if this method had fixed points, then

$$\frac{\partial \mathbf{w}}{\partial t} = \eta \mathbf{C}\mathbf{w} = \mathbf{0} = 0\mathbf{w} \tag{2.8}$$

would have to hold. Here $\mathbf{w}$ is an eigenvector of the matrix $\mathbf{C}$ with eigenvalue 0. The above method can also be displayed as a gradient descent with function

$$\mathbf{F}(\mathbf{w}) = -\frac{1}{2}\mathbf{w}^T\mathbf{C}\mathbf{w} \tag{2.9}$$

whose gradient is $-\mathbf{C}\mathbf{w}$. [12] Figure 2.3 visualizes this equation for $dim(\mathbf{w}) = 1$.

With the aid of the visualization, it is clear that $\mathbf{0}$ is not a stable fixed point but a local maximum of the function to be minimized. Furthermore, it is visible that the correlation of the input data is maximized in the direction $\mathbf{w}$ and these directions are possible fixed points. Nonetheless, the length of $\mathbf{w}$ is not limited meaning the length of it tends to infinity, because when

$$\mathbf{w}^T\mathbf{C}\mathbf{w} > 1 \text{ , then } (\lambda\mathbf{w})^T\mathbf{C}(\lambda\mathbf{w}) > 0 \text{ for } \lambda > 1$$

is even larger. Therefore, the Hebbian learning rule is unstable. [12]

Nonetheless, we may argue the following proposition, taken from [12]:

Proposition 1:
$\mathbf{w}^T\mathbf{C}\mathbf{w}$ is maximised by $\mathbf{e}_1$, as long as the vectors are of length 1. In the to $\mathbf{e}_1$ orthogonal room $\mathbf{e}_2$ maximises $\mathbf{w}^T\mathbf{C}\mathbf{w}$ and in the to $\mathbf{e}_2$ orthogonal room e_3 maximises $\mathbf{w}^T\mathbf{C}\mathbf{w}$.

Proof: When $\mathbf{w} = \sum_i a_i\mathbf{e}_i$, then

$$\mathbf{w}^T\mathbf{C}\mathbf{w} = (\sum_i a_i\mathbf{e}_i)^T\mathbf{C}(\sum_i a_i\mathbf{e}_i) = (\sum_i a_i\mathbf{e}_i)^T(\sum_i \lambda_i a_i\mathbf{e}_i)$$

$$= (a_1\mathbf{e}_1^T + \ldots + a_n\mathbf{e}_n^T)(\lambda_1 a_1 \mathbf{e}_1 + \ldots + \lambda_n a_n \mathbf{e}_n)$$

since $\langle \mathbf{e}_i, \mathbf{e}_j \rangle = 0$ for $i \neq j$ and $\langle \mathbf{e}_i, \mathbf{e}_j \rangle = 1$ for $i = j$, only the following terms remain

$$= \lambda_1 a_1^2 + \ldots + \lambda_n a_n^2$$

$$= \sum_i \lambda_i a_i^2$$

if one assumes that $\lambda_1 > \lambda_2 > \ldots > \lambda_n > 0$, and knowing that all but maximum one a_i are zero, because the coefficients of the eigenvectors are unique except for the sign, since they form a basis, we get

$$\leq \sum_i \lambda_1 a_i^2 = \lambda_1$$

Via a similar calculation one can then prove that the maximum λ_{i+1} can be reached through e_{i+1}. $\square$

To counteract the indefinite growth of vector $\mathbf{w}$ and the stability issues with the Hebbian learning rule another formulation is needed.

Both Oja and Bienenstock et al. have developed a solution for this problem. [10] [1] The Oja learning rule is presented in the next chapter and the BCM learning rule in the chapter after that.

3 The Oja neuron model

The Oja learning rule of a neuron was proposed by Oja in 1982. He used the simple neuron model and Hebbian learning as a starting point for his theory and found a way to prevent the previously described indefinite growth of the synaptic weights, as well as making the learning rule parallelizable to ensure faster calculation of the weights. [10]

The Oja neuron is mathematically the same as Hebb's neuron in Eq. 2.2 . [10] The input data $\mathbf{x}$ is a realisation of a random vector and is independently and identically distributed, according to some underlying probability P. It is assumed that $\mathbb{E}_X[\mathbf{x}] = 0$. [6]

3.1 Oja learning

The Oja learning theory takes the Hebbian learning scheme in equation 2.4 as a starting point. To prevent the indefinite growth of a synapse, the Hebbian learning scheme is normalized. This normalized learning equation is expressed componentwise as

$$w_i(t+1) = \frac{w_i(t) + \eta y(t) x_i(t)}{\left(\sum_{i=1}^{n} [w_i(t) + \eta y(t) x_i(t)]^2\right)^{1/2}} \tag{3.1}$$

whereas the vectorized notation is

$$\mathbf{w}(t+1) = \frac{\mathbf{w}(t) + \eta y(t)\mathbf{x}(t)}{||\mathbf{w}(t) + \eta y(t)\mathbf{x}(t)||} \tag{3.2}$$

where $\mathbf{w}(t)$ symbolizes a weight vector at time t. [10] The normalization ensures that $||\mathbf{w}(t)|| = 1, \ \forall t$. Furthermore, $\mathbf{x}(t) \in \mathbb{R}^n$, and $x_i(t) \in \mathbb{R}$, meaning we drop the positivity requirement stated in the previous section that $x_i > 0, \ \forall i$. This learning does not attempt to be biologically accurate anymore.

3.1.1 Convergence of Oja learning

The following propostition is taken from [12].

Proposition 2:
Equation 3.1 and equivalently 3.2 converges in the mean to $\mathbf{e}_1$ or $-\mathbf{e}_1$.

Proof: The expression $\mathbf{e}_i^T \mathbf{w}(t+1)$, where $\mathbf{w}(t+1)$ is the averaged vector is calculated as:

$$\mathbf{e}_i^T \mathbf{w}(t+1) = \mathbf{e}_i^T \frac{\mathbf{w}(t) + \eta \mathbf{C}\mathbf{w}(t)}{||\mathbf{w}(t) + \eta \mathbf{C}\mathbf{w}(t)||} = \mathbf{e}_i^T \mathbf{w}(t) \frac{1 + \eta \lambda_i}{||\mathbf{w}(t) + \eta \mathbf{C}\mathbf{w}(t)||}$$

$$\leq \mathbf{e}_i^T \mathbf{w}(t) \frac{1+\eta\lambda_i}{||\mathbf{w}(t)||\ ||1+\eta\lambda_1||} = \mathbf{e}_i^T \mathbf{w}(t) \frac{1+\eta\lambda_i}{1+\eta\lambda_1}$$

The last equation holds since $||w(t)|| = 1$ and $\lambda_1 > \lambda_2 > \ldots > \lambda_n > 0$.

The coefficients of the vector $\mathbf{w}$ are given by $\mathbf{e}_i^T \mathbf{w}$ regarding the basis $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n$. Therewith, the coordinates of $\mathbf{w}$ regarding the basis are given by

$$\left(\mathbf{w}(t)_1, \mathbf{w}(t)_2 \frac{1+\eta\lambda_2}{1+\eta\lambda_1}, \ldots, \mathbf{w}(t)_n \frac{1+\eta\lambda_n}{1+\eta\lambda_1} \right)^T \tag{3.3}$$

where $\mathbf{w}(t)_i$ are the old coefficients regarding the basis.

The second $0 < \frac{1+\eta\lambda_i}{1+\eta\lambda_1} \leq 1$ decreases as i increases. Therefore the coefficient in the direction of $\mathbf{e}_1$ is comparatively large. At every time step t the coefficients in the other directions decay while the one in the direction of $\mathbf{e}_1$ becomes larger in comparison.

There is a special case, when the coefficient in the direction of $\mathbf{e}_1$ is initialized as 0. Then equation 3.3 would converge to $\mathbf{0}$. If $\mathbf{w}(0)$ is initialized randomly, the probability of this event is 0. □

While this algorithm to determine $\mathbf{w}$ is convergent, it is also numerically unstable and not parallelizable. [12] Therefore another form must be found, which is the main result presented in [10].

3.1.2 Computationally efficient learning for parallel computers

Equation 3.1, and therewith also equation 2.2, requires global information to perform calculations, meaning all weights $w_i(t)$ are necessary to change a single weight $w_i(t+1)$. [12] Therefore, equation 3.2 is expanded as a power series at $\eta_0 = 0$. Oja learning reasonably assumes that the learning rate η is small which leads to the second and all higher powers of η being vanishing such that they can be neglected. [10]
We use $y = \mathbf{w}^T \mathbf{x}$ in the stochastic equation 3.2, such that

$$f(\eta) = \frac{\mathbf{w} + \eta \mathbf{w}^T \mathbf{x}\mathbf{x}}{||\mathbf{w} + \eta \mathbf{w}^T \mathbf{x}\mathbf{x}||} \tag{3.4}$$

and approximate this with the first 2 terms of the Taylor series:

$$f(\eta_0 = 0) \approx \frac{\mathbf{w}}{||\mathbf{w}||} + (\eta - \eta_0) \left(\frac{\mathbf{w} + \eta \mathbf{w}^T \mathbf{x}\mathbf{x}}{||\mathbf{w} + \eta \mathbf{w}^T \mathbf{x}\mathbf{x}||} \right)' (\eta_0 = 0) \tag{3.5}$$

In the next step, $\eta_0 = 0$ is substituted and the quotient rule $(\frac{f}{g})' = \frac{(f'g - g'f)}{g^2}$ and the chain rule $(f(g(x))' = f'(g(x))g'(x)$ are used.

$$= \frac{\mathbf{w}}{||\mathbf{w}||} + \eta \left(\frac{\mathbf{w}^T\mathbf{x}\mathbf{x}\,||\mathbf{w}+\eta\mathbf{w}^T\mathbf{x}\mathbf{x}|| - (\mathbf{w}+\eta\mathbf{w}^T\mathbf{x}\mathbf{x})\frac{(\mathbf{w}+\eta\mathbf{w}^T\mathbf{x}\mathbf{x})^T(\mathbf{w}^T\mathbf{x}\mathbf{x})}{||\mathbf{w}+\eta\mathbf{w}^T\mathbf{x}\mathbf{x}||}}{||\mathbf{w}+\eta\mathbf{w}^T\mathbf{x}\mathbf{x}||^2} \right) (\eta_0 = 0) \quad (3.6)$$

$$= \frac{\mathbf{w}}{||\mathbf{w}||} + \eta \left(\frac{\mathbf{w}^T\mathbf{x}\mathbf{x}}{||\mathbf{w}||} - \frac{\mathbf{w}\mathbf{w}^T\mathbf{w}^T\mathbf{x}\mathbf{x}}{||\mathbf{w}||^3} \right) \quad (3.7)$$

When considering $||w|| \approx 1$, we can simplify eq. 3.7 to

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta(\mathbf{w}(t)^T\mathbf{x}\mathbf{x} - (\mathbf{w}(t)^T\mathbf{x})^2\mathbf{w}(t)) \quad (3.8)$$

which is the stochastic version of the Oja rule. [12] It is a nonlinear stochastic difference equation. [10] Here the term $\mathbf{w}(t)^T\mathbf{x}\mathbf{x}$ represents the usual Hebbian synaptic modification and $-(\mathbf{w}(t)^T\mathbf{x})^2\mathbf{w}(t)$ signifies the forgetting of the neuron. [6] When averaging over the inputs, the stochastic version above can be turned into the averaged version of the learning rule such that

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta(\mathbf{C}\mathbf{w}(t) - (\mathbf{w}(t)^T\mathbf{C}\mathbf{w}(t))\mathbf{w}(t)) \quad (3.9)$$

applies. [12]

3.1.3 Stable fixed points of Oja's learning rule

The averaged learning rule is now investigated for stability, knowing that convergence is verified by proposition 2. This can be done via the Jacobian matrix, as stated in the following theorem that will be left without proof, taken from [13].

Theorem 3.1:
Let $U \subset \mathbb{R}^n$, $\mathbf{F} \in C^1(U, \mathbb{R}^n)$ and $\mathbf{w}^* \in U$ fulfills the fixed point equation $\mathbf{F}(\mathbf{w}^*) = \mathbf{0}$. If all eigenvalues of the Jacobian Matrix $\frac{\partial \mathbf{F}}{\partial \mathbf{w}}(\mathbf{w}^*)$ have a negative real part, then the fixed point $\mathbf{w}^*$ is asymptotically stable.

The following proposition follows the one presented in [12]:

Proposition 3:
Fixed points of the averaged Oja learning rule are the principal conponents, and the zero vector. The only stable fixed points are the vectors $\mathbf{e}_1$ and $-\mathbf{e}_1$, which are the principal components with the largest eigenvalue.

Proof: As the starting point we take the averaged learning rule 3.9 and recall that for fixed points we have that:

$$\frac{\partial \mathbf{w}}{\partial t} = \mathbf{0} \implies \eta(\mathbf{Cw} - (\mathbf{w}^T\mathbf{Cw})\mathbf{w}) = \mathbf{0} \implies \mathbf{Cw} = (\mathbf{w}^T\mathbf{Cw})\mathbf{w}$$

since $\eta \neq 0$. Therefore, $\mathbf{w}$ is the eigenvector of $\mathbf{C}$ with the eigenvalue $\lambda = \mathbf{w}^T\mathbf{Cw}$ or $\mathbf{w} = \mathbf{0}$. When $\mathbf{w} \neq \mathbf{0}$, then we can say that $\mathbf{w} = \alpha\mathbf{e}_i$ with $\alpha \neq 0$. Here, all e_i form a basis and e_i represents an arbitrary basis vector within this basis. Then:

$$\lambda\mathbf{w} = \lambda_i\alpha\mathbf{e}_i = \mathbf{C}\alpha\mathbf{e}_i = \mathbf{Cw} = (\mathbf{w}^T\mathbf{Cw})\mathbf{w} = (\alpha\mathbf{e}_i)^T\mathbf{C}(\alpha\mathbf{e}_i)\alpha\mathbf{e}_i = \alpha^3\lambda_i\mathbf{e}_i \implies \alpha = \pm 1$$

Therefore, only the vectors $\mathbf{e}_i$, $-\mathbf{e}_i$ and the zero vector are fixed points of the averaged learning rule.

The stability of the fixed points can be investigated via the Jacobian matrix of the function $\mathbf{w} \mapsto -(\mathbf{Cw} - (\mathbf{w}^T\mathbf{Cw})\mathbf{w})$. If the Jacobian matrix

$$J(\mathbf{w}) = -\mathbf{C} + (\mathbf{w}^T\mathbf{Cw})\mathbf{I} + 2\mathbf{w}\mathbf{w}^T\mathbf{C}$$

is positive definite at the fixed point the learning rule is stable. This is due to the fact that we are not looking for a local minimum but a maximum, as explained after equation 2.9 and visualized in figure 2.3. Therefore, we check if

$$\mathbf{e}_j^T J(\mathbf{e}_i)\mathbf{e}_k = \mathbf{e}_j^T(-\mathbf{C} + (\mathbf{e}_i^T\mathbf{C}\mathbf{e}_i)I + 2\mathbf{e}_i\mathbf{e}_i^T\mathbf{C})\mathbf{e}_k > 0$$

since $\mathbf{C}\mathbf{e}_k = \lambda_k\mathbf{e}_k$ and $\mathbf{e}_i^T\mathbf{C}\mathbf{e}_i = \lambda_i$ we get:

$$= -\lambda_k\mathbf{e}_j^T\mathbf{e}_k + \lambda_i\mathbf{e}_j^T\mathbf{e}_k + 2\mathbf{e}_j^T\mathbf{e}_i\lambda_k\mathbf{e}_i^T\mathbf{e}_k = \begin{cases} 0 & j \neq k \\ 2\lambda_i & i = k = j \\ \lambda_i - \lambda_j & i \neq j = k \end{cases}$$

Investigating the first case, it is immediately visible that for the eigenvalue 0 and therewith $\mathbf{w} = \mathbf{0}$ the Jacobian matrix is $-\mathbf{C}$, so negative definite, implying that the fixed point $\mathbf{0}$ is unstable.

Looking at the third case we assume that $j = k \neq i$ leading to $\lambda_i - \lambda_j > 0 \ \forall j \neq i$ if $i = 1$. This means that $\mathbf{e}_1$ or the negative, the eigenvectors of the eigenvalue λ_1, is a stable fixed point. Investigating the general case we look at whether any point converges to $\mathbf{e}_1$ or the negative via:

$$(\sum_k \alpha_k\mathbf{e}_k)^T J(\mathbf{e}_i)(\sum_k \alpha_k\mathbf{e}_k) = \sum_{k=1}^{n} \alpha_k^2\mathbf{e}_k^T J(\mathbf{e}_i)\mathbf{e}_k = \sum_{k=1}^{n} \mathbf{e}_k^T J(\mathbf{e}_i)\mathbf{e}_k$$

Any point therefore converges to the stable fixed point $\mathbf{e}_1$ or $-\mathbf{e}_1$.

The directions $\mathbf{e}_i$ with $i \neq 1$ are unstable when they are approached from directions $\mathbf{e}_j$ where $j > i$. Otherwise they are stable. $\square$

4 The BCM neuron model

The third neuron model to be analyzed is the Bienenstock-Cooper-Munro (BCM) neuron model. It constructs a model neuron model whose goal it is to maximize selectivity in this neuron. The BCM learning rule extends the Hebbian learning rule by finding a way to cause synaptic decrease. Whether the neuron inhibits or promotes an input pattern depends on whether the output exceeds a variable threshold. [1]

4.1 BCM neuron

The BCM neuron is similar to the simple neuron model based on the Hebbian principle. The Hebbian model assumes an input vector at a certain time t. All presynaptic signals from this vector arrive at the postsynaptic neuron at this time t. While the BCM neuron functions the same way after all assumptions have been made, their model describes a different way of how the inputs $\mathbf{x}$ come about. They assume that incoming spikes occur on each of incoming synapses. The input vector $\mathbf{x}(t)$ is then determined by averaging over the instantaneous inputs over a period τ which takes place between $t-1$ and t. [1]

The neuron takes the inputs and weights at time t and performs integration to determine the output at time t. The model is then simplified further by assuming that the integrative power of the neuron is a linear function. Then the neuron model can be parametrized just like the neruon in equation 2.2. In which exact way the neuron performs the integration may be changed. The "results remain unchanged if, for instance, [$y(t) = S(\langle \mathbf{w}(t), \mathbf{x}(t) \rangle$], with S being a positive-valued sigmoid shaped function". [1]

As mentioned before the neuron attempts to maximize selectivity. How to determine the selectivity is given by:

$$Sel_X(N) = 1 - \frac{\text{mean response of } N \text{ with respect to } X}{\text{maximum response of } N \text{ with respect to } X} \tag{4.1}$$

where N is the neuron and X is an R^n-valued random variable associated with the inputs $\mathbf{x}$. It represents a stationary stochastic process. At a time t a random input is given according to the time-invariant probability distribution of the random variable. The selectivity always lies between 0 and 1. When the selectivity of the neuron is high then $Sel_X(N)$ is close to 1 and vice versa. Applying these definitions to the neuron in state $\mathbf{w}$, we get

$$Sel_X(\mathbf{w}) = 1 - \frac{\mathbb{E}_X[\langle \mathbf{w}, X \rangle]}{ess\ sup[\langle \mathbf{w}, X \rangle]} \tag{4.2}$$

when parametrized. [1]

Due to the linearity of the expectation operator $\mathbb{E}$ we may transform this definition to read:

$$= 1 - \frac{\langle \mathbf{w}, \overline{\mathbf{x}} \rangle}{max(\langle \mathbf{w}, X \rangle)} = 1 - \frac{\overline{y}}{max(y)}$$

where $\overline{y} = \mathbb{E}_X[y]$, $\overline{\mathbf{x}} = \mathbb{E}_X[\mathbf{x}]$.
The $max(y)$ is not the same as $ess\ sup(y)$ but "is equivalent [...] in most common applications". [1]

The average $\overline{\mathbf{x}}$, and therewith also $\overline{y}$, is determined over a time period T, preceding t. It is determined over a much larger time period than the moving time averages are determined, such that $\tau \ll T$. The time step t is increased by one and the weights are recalculated each time a time period τ passes, and a time average of the input spikes has been determined. This causes $\overline{y}(t)$ to change at a much slower rate than $y(t)$. [1]

4.2 BCM learning

The following is the basic equation of learning proposed in [1]:

$$\frac{\partial w_i(t)}{\partial t} = \eta_w \phi_{BCM}(y(t), \theta) x_i(t) - \eta_0 w_i(t) \tag{4.3}$$

where $\phi_{BCM}(y(t), \theta)$ is a scalar function of postsynaptic activity that changes sign at a value θ, called the modification threshold.

$$\begin{aligned} \phi_{BCM}(y(t), \theta) < 0 \text{ for } y < \theta \\ \phi_{BCM}(y(t), \theta) > 0 \text{ for } y > \theta \end{aligned} \tag{4.4}$$

This learning equation is only dependent on the input to and the output out of the neuron, as is the case for unsupervised learning rules. The term $-\eta_0 w_i(t)$ causes a uniform decay of the weights. Due to $0 < \eta_0 \lll 1$, this does not affect the behavior of the neuron and therewith the dynamical system. This is why it is neglected form here on. The learning rate η_w is $0 < \eta_w \ll 1$. [1]

The vector $\mathbf{w}$ develops in the direction of X if the output is larger than the threshold θ and in the opposite direction of X if the output is smaller than θ. This is analogous to the Hebbian principle, where when $\mathbf{x} \in \mathbb{R}^n$,with $x_i > 0\ \ \forall i$ and the output y is large enough, the weight $\mathbf{w}$ increases. However, BCM learning features synaptic decrease through the function ϕ and when when $\mathbf{x} \in \mathbb{R}^n$,with $x_i < 0\ \ \forall i$ and the output y is not sufficiently large, the weight $\mathbf{w}$ decreases. This can be regarded as a form of competition between input patterns $\mathbf{x}$. [1]

4.2.1 Issues with a fixed threshold

The idea of the modification threshold in a learning scheme was introduced in [3] using a constant threshold. The definition of the threshold θ fundamentally influences the function of the neuron and must be chosen in such a way that the learning rule converges to a fixed point and is stable.

The constant threshold was proposed as a function that takes the output y as an input. The function modifies the output according to a postsynaptic firing threshold θ and the saturation limit of the neuron μ.

$$P(y) = \begin{cases} \mu & y > \mu \\ y & \theta \leq y < \mu \\ \theta & y < \theta \end{cases} \tag{4.5}$$

However, this "resulted in a certain lack of robustness of the system". [1]

Their mathematical results only showed weak asymptotic convergence of the weight vector. The limit of the expected value of the weight vector $\overline{\mathbf{w}}$ is "only an average limit in a large number of similar cells." When the learning rate is small the "actual limits tend to be close to $\overline{\mathbf{w}}$", however, the" non-zero variance in $\overline{\mathbf{w}}$ [causes] variations in asymptotic tuning in individual cells". Sometimes the "model neuron might even change its preferred pattern." [3]

While it shows only weak convergence and is unstable, it already displays properties that are important for modeling the biological neuron accurately. Given patterned input the response of the neuron increases in specificity, while when given noise-like input the neurons response decreases in specificity. Furthermore, if specificity was lost due to noise-like input, it can be regained through patterned input. "Even with relatively high noise levels, with signal to noise ratios considerably smaller than one," the neuron is still able to extract the patterned input and does this qualitatively similar to when there is no noise. Therefore, the learning algorithm already has good averaging properties. [3]

4.2.2 An appropriate choice of ϕ_{BCM} and θ

Due to the issues with the fixed threshold described above, a better choice for ϕ and θ must be found that ensures the learning algorithm converges and is stable. For this the threshold θ is made a function of time $\theta(t)$ and is modified at every time increment. The average output $\overline{y}$ is introduced to the function $\phi_{BCM}(y(t), \theta)$ that takes the output at time step t and the average output up until time step t into account. The use of $\overline{y}$ here ensures the "boundedness of the state and efficient threshold modification". Furthermore, stable fixed points exists, if they are of high selectivity. This is easy to see at a fixed point with

zero selectivity. Any input would drive it away from its current state to a state of higher selectivity. [1]
Including this and neglecting the uniform decay, Eq. 4.3 can be modified to read:

$$\frac{\partial \mathbf{w}}{\partial t} = \eta_w \phi_{BCM}(y(t), \theta)\mathbf{x}(t) \tag{4.6}$$

The function $\phi_{BCM}(y(t), \theta)$ changes sign depending on the crucial point θ according to 4.4. A simple choice of θ would be $\bar{y}$, the average postsynaptic activity. However, while this does provide the desired property of instability of low selectivity points, it is not bounded from the origin and infinity. [1] Hence, the final choice for $\phi_{BCM}(y, \theta)$ is this nonlinear function that provides both desired properties:

$$\begin{aligned} sign\phi_{BCM}(y,\bar{y}) &= sign\, y\left(y - \left(\frac{\bar{y}}{y(0)}\right)^p \bar{y}\right) \quad \text{for } y > 0 \\ \phi_{BCM}(0,\bar{y}) &= 0 \quad \text{for all } \bar{y} \end{aligned} \tag{4.7}$$

where $y(0)$ and p are distinct fixed positive constants. This means that the final choice for θ is:

$$\theta = \left(\frac{\bar{y}}{y(0)}\right)^p \bar{y}$$

Any function ϕ_{BCM} that fulfills these conditions is satisfactory and may be chosen instead of the one written here. Factors such as the maximal response of the neuron and the convergence speed depend on the the numerical values of $y(0)$ and p. [1]
In the following $y(0)$ and p are chosen to be $y(0) = p = 1$, making $\theta = \bar{y}^2$.

4.2.3 The final learning equation

BCM learning is proposed as follows:

$$\frac{\partial \mathbf{w}}{\partial t} = \eta_w \mathbf{x}\phi_{BCM}(y, \theta) = \eta_w \mathbf{x} y(y - \theta) \tag{4.8}$$

with θ being a sliding threshold that changes over time just like $\mathbf{w}$. The function

$$\phi_{BCM} = y(y - \theta)$$

takes the output y of the neuron and $\theta = \mathbb{E}_X[y^2]$ as an input and determines whether y is large enough to promote a future signal transferred inward via the i^{th} synapse or to inhibit it. The change over time of the threshold θ is proposed as:

$$\frac{\partial \theta}{\partial t} = \eta_\theta (y^2 - \theta) \tag{4.9}$$

Originally it was assumed that $\theta = \mathbb{E}_X^2[y]$. It has been shown that this can be well approximated by $\theta = \mathbb{E}_X[y^2]$. As this includes the variance of y it will always be positive. This change "ensures stability even when the average of the inputs is zero". [9]

4.3 Convergence and stability theorems and lemmas

The random variable X influences the behavior of the dynamical system. 4.8 In this thesis only discrete distributions are considered and the K possible inputs $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^n$ are assumed to have the same probability $\frac{1}{K}$. The stochastic process X is a jump process that randomly chooses an input at each time t. [1]

The following lemmas and theorems are stated in [1] and describe why the BCM learning rule converges and is stable. They are quoted using the notation defined in 1.1.

Lemma 4.1:
Let x_1 and x_2 be linearly independent and X satisfy $P[X = x_1] = P[X = x_2] = \frac{1}{2}$. Then for any value of ϕ_{BCM} satisfying 4.7, 4.6 admits exactly four fixed points, $\mathbf{w}^0, \mathbf{w}^1, \mathbf{w}^2 and \mathbf{w}^{1,2}$ with: $Sel_X(w^0) = Sel_X(w^{1,2}) = 0$ and $Sel_X(w^1) = Sel_{\mathbf{X}}(w^2) = \frac{1}{2}$. (Here the superscripts indicate which of the x_i are not orthogonal to $\mathbf{w}$. ($\mathbf{w}^0$ is the origin.) Thus, for instance, $\langle \mathbf{w}^1, x_1 \rangle > 0$, $\langle \mathbf{w}^1, x_2 \rangle = 0$.)

Theorem 4.2:
Assume that in addition to the conditions of lemma 4.1, $cos(x^1, x^2) \geq 0$. Then w^0 and $w^{1,2}$ are unstable, w^1 and w^2 are stable, and whatever its initial value, the state of the system converges almost surely (i.e., with probability 1) either to w^1 or w^2.

Theorem 4.3:
Under the same conditions as in theorem 4.2, there exists around $m^1(m^2)$ a region $F^1(F^2)$ such that, once the state enters $F^1(F^2)$, it converges almost surely to $m^1(m^2)$.

Lemma 4.4:
Let $x^1, x^2, \ldots, x^n$ be linearly independent and X satisfy $P[X = x^1] = \cdots = P[X = x^n] = \frac{1}{n}$. Then, for any function ϕ satisfying equation 4.7, 4.6 admits exactly 2^n fixed points with selectivities $0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n-1}{n}$. There are n fixed points $w^1, w^2, \ldots, w^n$ of selectivity $\frac{n-1}{n}$.

Theorem 4.5:
Assume, in addition to the conditions of lemma 4.4, that $x^1, x^2, \ldots, x^n$ are all mutually orthogonal or close to orthogonal. Then the n fixed points of maximum selectivity are stable, and whatever its initial value, the state of the system converges to one of them.

While this is valid according to [1], we may also investigate this learning rule in the same manner as the stability analysis in section 3.1.3.

4.4 Fixed points of the learning rule

The learning rule is analysed as noted in 4.8. The fixed points can be determined by setting the learning equation to zero, as stated in 2.1, such that

$$\mathbf{0} = \eta_w \mathbf{x} \phi_{BCM}(y, \theta)$$

The learning rate η_w is never 0, meaning we are only left with

$$\mathbf{0} = \mathbf{x} \phi_{BCM}(y, \theta)$$

We rewrite the learning rule as

$$\frac{\partial \mathbf{w}}{\partial t} = \mathbf{PC}\Phi \tag{4.10}$$

where $\mathbf{C} \in \mathbb{R}^{n \times n}$ is the covariance matrix of the input vectors $\mathbf{x}$ and $\mathbf{P} \in \mathbb{R}^{n \times n}$ contains the probabilities p_i of an input x_i occurring on the main diagonal and is otherwise 0. The vector $\Phi \in \mathbb{R}^n$ is made up of the function $\phi_i,\ i = 1, \ldots, n$, which returns a scalar when evaluated.

When we want that $\mathbf{PC}\Phi = \mathbf{0}$ and $\mathbf{PC}$ is not regular then Φ is in the kernel of it. To avoid this, we assume that $\mathbf{C}$ is regular and that it has eigenvectors $\mathbf{e}_i \in \mathbb{R}^n$. Any input $\mathbf{x}$ can then be represented by $\mathbf{x} = \sum_{k=1}^{n} \alpha_k \mathbf{e}_k$. We will only consider the eigenvectors $\mathbf{e}_i$ as inputs. This ensures that the we have a finite number of mutually orthogonal vectors.

Therefore, all possible outputs are

$$y_i = \mathbf{w}^T \mathbf{e}_i = \sum_{j=1}^{n} \mathbf{w}_j \mathbf{e}_{ij} \tag{4.11}$$

and

$$\theta = \mathbb{E}_X[y^2] = \sum_i p_i (\mathbf{w}^T \mathbf{e}_i)^2 = \sum_i p_i \sum_j (\mathbf{w}_j^T \mathbf{e}_{ij})^2 \tag{4.12}$$

We assume that $p_i > 0,\ \forall i$.

Using the information above, we attempt to find the fixed points of $\phi(y, \theta) = 0$.

First, we take $n = 1$. This way we get from 4.11 that $y_1 = we_1$ and from 4.12 that $\theta = p_1(we_1)^2$. When substituting this into $\phi(y, \theta) = y_1(y_1 - \theta)$ and setting it to zero to find the fixed points we get:

$$we_1(we_1 - p_1(we_1)^2) = 0$$

$$1 - p_1 we_1 = 0$$

$$1 = p_1 w e_1$$
$$\implies w^* = \frac{1}{p_1 e_1}$$

For $n = 2$, we have

$$y_1 = \mathbf{w}^T \mathbf{e}_1 = w_1 e_{11} + w_2 e_{12}, \;\; y_2 = \mathbf{w}^T \mathbf{e}_2 = w_1 e_{21} + w_2 e_{22}$$

and

$$\phi(y, \bar{y})_1 = y_1(y_1 - \theta), \;\; \phi(y, \bar{y})_2 = y_2(y_2 - \theta)$$

where $\theta = p_1(\mathbf{w}^T \mathbf{e}_1)^2 + p_2(\mathbf{w}^T \mathbf{e}_2)^2$ and $\mathbf{e}_1 = (e_{11}, e_{12})^T$ and $\mathbf{e}_2 = (e_{21}, e_{22})^T$.

We substitute this into

$$\phi_1 = y_1(y_1 - \theta) = 0 \;\; \textit{and} \quad \phi_2 = y_2(y_2 - \theta) = 0$$

and then first calculate w_1 and then w_2. The exact calculations are added in annex A. Here we just state the result.

$$w_{11,12} = -(e_{11} - p_1 2 w_2 e_{11} e_{12} - 2 p_2 w_2 e_{21} e_{22})$$
$$\pm \frac{\sqrt{(e_{11} - p_1 2 w_2 e_{11} e_{12} - 2 p_2 w_2 e_{21} e_{22})^2 - 4(-p_1 e_{11}^2 - p_2 e_{21}^2)(-p_2 w_2^2 e_{22}^2 + w_2 e_{12} - p_1 w_2^2 e_{12}^2)}}{2(-p_1 e_{11}^2 - p_2 e_{21}^2)}$$

where $a \neq 0$. w_{11} is the expression with a plus before the square root and w_{12} is the expression with a minus before the square root. For w_{11} the two possible second coordinates are:

$$w_{21,22} = \frac{-y \pm \sqrt{y^2 - 4xz}}{2x}$$

The exact for values of x, y and z can be found at the end of annex A. A similar calculation can be conducted in order to identify the second coordinates for the fixed points with w_{12} as the first coordinate.

We have therefore identified 4 possible fixed points. This is how many should exist according to 4.1 and 4.4. As the determination of the Jacobian matrix $J(\mathbf{w}^*)$ requires extensive calculation, it can be questioned, how sensible it is to determine fixed points in this manner to evaluate their stability, especially when $n > 2$.

5 Summary

After introducing three neuron models and their learning rules, as well as analyzing them with respect to their convergence and stable fixed points, we now compare how they model a neuron and determine a learning rule. Furthermore, how to determine fixed points and evaluate their stability is outlined to apply to other neuron models and learning rules.

5.1 Neuron models

The neuron model of the three models is the same in the sense that it can be noted by equation 2.2.

However, Hebb restricts the parameters biologically to be positive. [10]

Oja removes the positivity requirement and defines x to be a realization of a random vector that is independently and identically distributed according to an underlying probability. Furthermore, $\mathbb{E}_X[\mathbf{x}] = 0$. [10] [12]

BCM treats $\mathbf{x}$ at time t as an instantaneous variable, just like Oja and Hebb. However, how $\mathbf{x}$ is determined differs. It is an average taken over all input spikes in a time period τ lasting from $t-1$ to t. For the analysis later on, it is assumed that there is a finite amount of possible time averages $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{R}^n$ that occur at a time t with the same probability $\frac{1}{k}$. [1]

5.2 Learning rules

The three learning rules are:

Hebb: $\frac{\partial \mathbf{w}}{\partial t} = \eta\, \mathbf{x}(t)\, \phi_H(y, \theta) = \mathbf{w}(t) + \eta y \mathbf{x}(t)$

Oja: $\frac{\partial \mathbf{w}}{\partial t} = \eta\, (\mathbf{x}y - y^2 \mathbf{w}(t))$

BCM: $\frac{\partial \mathbf{w}}{\partial t} = \eta_w \mathbf{x} \phi_{BCM}(y, \theta) = \eta_w \mathbf{x} y (y - \theta) \;\;,\;\; \theta = \mathbb{E}_X[y^2].$

Hebbian learning does not have a way of causing synaptic decrease, leading to it not converging. It has no stable fixed points. [12]

Oja learning normalizes Hebbian learning and the learning equation written above has a specific term that causes the synaptic decrease. The stable fixed point of Oja's learning rule is an eigenvector of the covariance matrix. [10]

The BCM learning rule is characterized and stands out in comparison to the other rules by the fact that it has a threshold for determining when output signas are large enough to be excitory or when they should be inhibitory. This threshold ensure the stability and boundedness of the possible states $\mathbf{w}$. The learning rule has stable fixed points, that it converges to, when assuming that the inputs are mutually orthogonal, occur with the same probability, and are of maximal selectivity. [1]

5.3 Determining fixed points and their stability

The fixed points of a learning equation that fulfills the conditions listed in 2.1 can be determined via

$$\mathbf{F}(\mathbf{w}^*) = \mathbf{0}$$

Whether the determined fixed point(s) are stable can then be determined according to 3.1. To check this the Jacobian matrix $J(\mathbf{w}^*)$ of the fixed point is determined. Afterwards the eigenvalues of this matrix are determined, which must be lower than zero.

To make the generalization to n dimensions easier, one can start with $n = 1$ and $n = 2$, to see if the general case becomes clear this way.

6 Conclusion

Three neuron models were presented with consistent notation and explained. For each neuron model's learning rule the convergence to a fixed point and the stability thereof were analyzed.

As a result it is clear that the simple neuron model and therewith the learning rule according to the Hebbian principle must not be considered in further research, and can only be used as a starting point to understanding other learning rules. It is an intrinsically unstable algorithm that does not converge to a fixed point.

The Oja learning rule, which extracts the first principle component, is both convergent and has stable fixed points. Furthermore, the learning rule has already been optimized to be numerically stable and parallelizable. Therefore, further research may be conducted into how it can be efficiently implemented on different types of hardware.

The BCM learning rule attempts to maximize selectivity towards incoming signals. The learning rule is both convergent and has stable fixed points. These stable fixed points necessarily have a high selectivity. The stable fixed points cannot be efficiently determined using the method in this thesis. Further research may be conducted to determine if there exists a different approach. It should be researched further, too, to determine its use in neuromorphic hardware.

6.1 Outlook

The stable and convergent models presented in this thesis can be extended to networks of interacting neurons. Investigating the properties of the neuron model in a network of neurons ensures that the model will function the same way in a network as in isolation. Furthermore, it can be investigated how fast a network of interacting neurons converges to its final state. 1 The Oja learning rule has been extended by Sanger (Sanger, 1989) to a learning rule called generalized Hebbian algorithm which not only extracts the first principle component but all of them. This may or may not be necessary to research for networks of interacting neurons.

In terms of the BCM learning rule, ways to improve the learning rule have been suggested (Intrator& Cooper,1992; Law& Cooper,1994) since the original paper by Bienenstock et al([1]). These should be investigated for their convergence and the stability of their fixed points.

Another concept and way to model a neuron that should be looked into is spike-timing-dependent plasticity (STDP) and further its connection to BCM (Izhikevich et al., 2003).

Moreover, a method of maximization of information transmission for spiking neurons using a generalized BCM rule was developed (Toyoizumi et al., 2005), that should be considered when researching STDP in connection with BCM.

Lastly, there have been a multitude of papers on the topic of neuron models for neuromorphic hardware. Their aid in the verification of the hypothesis stated in chapter 1 should be investigated.

Moreover, while the concepts mentioned in this thesis are mathematically sound, it is not to say that they will indeed be computationally efficient. Only when knowledge about this has been gained, will the theoretical knowledge of the neuron model help with larger scale computations. The first implementations may be verified on a CPU. While this will help get a better understanding of how well each neuron model performs, as explained before, a CPU is not very efficient in its way of computing a result. Therefore, the different models should be implemented to be run on a GPU and further, if possible, on a quantum computer or neuromorphic hardware.

Appendix A: Calculation of fixed points of BCM learning for n=2

After substitution the calculation goes as follows:

$$
\begin{aligned}
\phi_1 &= 0 \\
y_1 - \theta &= 0 \\
\phi_1 = w_1 e_{11} + w_2 e_{12} - (p_1(\mathbf{w}^T \mathbf{e}_1)^2 + p_2(\mathbf{w}^T \mathbf{e}_2)^2) &= 0 \\
w_1 e_{11} + w_2 e_{12} - (p_1(w_1 e_{11} + w_2 e_{12})^2 + p_2(w_1 e_{21} + w_2 e_{22})^2) &= 0 \\
w_1 e_{11} + w_2 e_{12} - p_1 w_1^2 e_{11}^2 - p_1 2 w_1 w_2 e_{11} e_{12} & \\
-p_1 w_2^2 e_{12}^2 - p_2 {w_1}^2 e_{21}^2 - 2 p_2 w_1 w_2 e_{21} e_{22} - p_2 w_2^2 e_{22}^2 &= 0 \\
-p_1 w_1^2 e_{11}^2 - p_2 {w_1}^2 e_{21}^2 + w_1 e_{11} - p_1 2 w_1 w_2 e_{11} e_{12} & \\
-2 p_2 w_1 w_2 e_{21} e_{22} - p_2 w_2^2 e_{22}^2 + w_2 e_{12} - p_1 w_2^2 e_{12}^2 &= 0 \\
w_1^2 \underbrace{(-p_1 e_{11}^2 - p_2 e_{21}^2)}_{a} + w_1 \underbrace{(e_{11} - p_1 2 w_2 e_{11} e_{12} - 2 p_2 w_2 e_{21} e_{22})}_{b} \underbrace{- p_2 w_2^2 e_{22}^2 + w_2 e_{12} - p_1 w_2^2 e_{12}^2}_{c} &= 0
\end{aligned}
$$

We can solve this quadratic equation via the quadratic formula, by assuming $a \neq 0$:

$$
w_{11,12} = \frac{-b\sqrt{b^2 - 4ac}}{2a} = -(e_{11} - p_1 2 w_2 e_{11} e_{12} - 2 p_2 w_2 e_{21} e_{22})
$$

$$
\pm \frac{\sqrt{(e_{11} - p_1 2 w_2 e_{11} e_{12} - 2 p_2 w_2 e_{21} e_{22})^2 - 4(-p_1 e_{11}^2 - p_2 e_{21}^2)(-p_2 w_2^2 e_{22}^2 + w_2 e_{12} - p_1 w_2^2 e_{12}^2)}}{2(-p_1 e_{11}^2 - p_2 e_{21}^2)}
$$

w_{11} is the solution using the plus sign, and w_{12} is the solution using the minus sign.

Now we prepare $y_2 - \theta = 0$:

$$
\begin{aligned}
w_1e_{21}+w_2e_{22}-(p_1(\mathbf{w}^T\mathbf{e}_1)^2+p_2(\mathbf{w}^T\mathbf{e}_2)^2) &= 0\\
w_1e_{21}+w_2e_{22}-(p_1(w_1e_{11}+w_2e_{12})^2+p_2(w_1e_{21}+w_2e_{22})^2) &= 0\\
w_1e_{21}+w_2e_{22}-p_1w_1^2e_{11}^2-p_12w_1w_2e_{11}e_{12}-p_1w_2^2e_{12}^2-p_2w_1^2e_{21}^2-2p_2w_1w_2e_{21}e_{22}-p_2w_2^2e_{22}^2 &= 0\\
-p_1w_1^2e_{11}^2-p_2w_1^2e_{21}^2+w_1e_{21}-p_12w_1w_2e_{11}e_{12}-2p_2w_1w_2e_{21}e_{22}+w_2e_{22}-p_1w_2^2e_{12}^2-p_2w_2^2e_{22}^2 &= 0\\
w_1^2(-p_1e_{11}^2-p_2e_{21}^2)+w_1(e_{21}-p_12w_2e_{11}e_{12}-2p_2w_2e_{21}e_{22})+w_2e_{22}-p_1w_2^2e_{12}^2-p_2w_2^2e_{22}^2 &= 0
\end{aligned}
$$

Here we substitute $w_1 = w_{11}$:

$$
\begin{aligned}
&\frac{-(e_{11}-p_12w_2e_{11}e_{12}-2p_2w_2e_{21}e_{22})}{2(-p_1e_{11}^2-p_2e_{21}^2)}\\
&+\frac{\sqrt{(e_{11}-p_12w_2e_{11}e_{12}-2p_2w_2e_{21}e_{22})^2-4(-p_1e_{11}^2-p_2e_{21}^2)(-p_2w_2^2e_{22}^2+w_2e_{12}-p_1w_2^2e_{12}^2)}}{2(-p_1e_{11}^2-p_2e_{21}^2)}\\
&\qquad\cdot(e_{21}-p_12w_2e_{11}e_{12}-2p_2w_2e_{21}e_{22})\\
&\qquad+\left(\frac{-(e_{11}-p_12w_2e_{11}e_{12}-2p_2w_2e_{21}e_{22})}{2(-p_1e_{11}^2-p_2e_{21}^2)}\right.\\
&\left.+\frac{\sqrt{(e_{11}-p_12w_2e_{11}e_{12}-2p_2w_2e_{21}e_{22})^2-4(-p_1e_{11}^2-p_2e_{21}^2)(-p_2w_2^2e_{22}^2+w_2e_{12}-p_1w_2^2e_{12}^2)}}{2(-p_1e_{11}^2-p_2e_{21}^2)}\right)^2\\
&\qquad\cdot(-p_1e_{11}^2-p_2e_{21}^2)\\
&\qquad\qquad+w_2e_{22}-p_1w_2^2e_{12}^2-p_2w_2^2e_{22}^2=0
\end{aligned}
$$

First, we attempt to solve the first 2 terms of the array:

$$
\begin{aligned}
&\frac{-(e_{11}-p_12w_2e_{11}e_{12}-2p_2w_2e_{21}e_{22})}{2(-p_1e_{11}^2-p_2e_{21}^2)}\\
&+\frac{\sqrt{(e_{11}-p_12w_2e_{11}e_{12}-2p_2w_2e_{21}e_{22})^2-4(-p_1e_{11}^2-p_2e_{21}^2)(-p_2w_2^2e_{22}^2+w_2e_{12}-p_1w_2^2e_{12}^2)}}{2(-p_1e_{11}^2-p_2e_{21}^2)}\\
&\qquad\cdot(e_{21}-p_12w_2e_{11}e_{12}-2p_2w_2e_{21}e_{22})
\end{aligned}
$$

$$= \frac{-(e_{11} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})(e_{21} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})}{2(-p_1 e_{11}^2 - p_2 e_{21}^2)}$$
$$+ \frac{\sqrt{(e_{11} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})^2 - 4(-p_1 e_{11}^2 - p_2 e_{21}^2)(-p_2 w_2^2 e_{22}^2 + w_2 e_{12} - p_1 w_2^2 e_{12}^2)}}{2(-p_1 e_{11}^2 - p_2 e_{21}^2)}$$
$$\frac{\cdot (e_{21} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})}{2(-p_1 e_{11}^2 - p_2 e_{21}^2)}$$

Now we focus on the first term of the equation above:

$$\frac{-(e_{11} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})(e_{21} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})}{2(-p_1 e_{11}^2 - p_2 e_{21}^2)}$$

$$= \frac{(-e_{11} + p_1 2w_2 e_{11} e_{12} + 2p_2 w_2 e_{21} e_{22})(e_{21} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})}{2(-p_1 e_{11}^2 - p_2 e_{21}^2)}$$

$$= \frac{-e_{11}(e_{21} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})}{2a}$$
$$+ \frac{2p_1 w_2 e_{11} e_{12}(e_{21} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})}{2a}$$
$$+ \frac{2p_2 1w_2 e_{21} e_{22}(e_{21} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})}{2a}$$

$$= \frac{-e_{11} e_{21}}{2a} + \frac{p_1 2w_2 e_{11}^2 e_{12}}{2a} + \frac{e_{11} 2p_2 w_2 e_{21} e_{22}}{2a}$$
$$+ \frac{e_{21} 2p_1 w_2 e_{11} e_{12}}{2a} - \frac{4p_1^2 w_2^2 e_{11}^2 e_{12}^2}{2a} - \frac{4p_1 p_2 w_2^2 e_{11} e_{12} e_{21} e_{22}}{2a}$$
$$+ \frac{2p_2 w_2 e_{21} e_{22} e_{21}}{2a} - \frac{4p_1 p_2 w_2^2 e_{11} e_{12} e_{21} e_{22}}{2a} - \frac{4p_2^2 w_2^2 e_{21}^2 e_{22}^2}{2a}$$

Next, we rejoin all the resulting terms of the first 2 terms, which are:

$$
\begin{aligned}
&\frac{-e_{11}e_{21}}{2a}+\frac{p_1 2w_2 e_{11}^2 e_{12}}{2a}+\frac{2e_{11}p_2w_2e_{21}e_{22}}{2a}+\frac{2e_{21}p_1w_2e_{11}e_{12}}{2a}-\frac{4p_1^2w_2^2e_{11}^2e_{12}^2}{2a}\\
&-\frac{4p_1p_2w_2^2e_{11}e_{12}e_{21}e_{22}}{2a}+\frac{2p_2w_2e_{21}e_{22}e_{21}}{2a}-\frac{4p_1p_2w_2^2e_{11}e_{12}e_{21}e_{22}}{2a}-\frac{4p_2^2w_2^2e_{21}^2e_{22}^2}{2a}\\
&+\frac{\sqrt{(e_{11}-p_1 2w_2e_{11}e_{12}-2p_2w_2e_{21}e_{22})^2-4(-p_1e_{11}^2-p_2e_{21}^2)(-p_2w_2^2e_{22}^2+w_2e_{12}-p_1w_2^2e_{12}^2)}}{2a}\\
&\qquad\cdot\frac{(e_{21}-p_1 2w_2e_{11}e_{12}-2p_2w_2e_{21}e_{22})}{2a}
\end{aligned}
$$

As the next step we solve the third and fourth terms. For this we use the notation from when we found $w_{11,12}$ to keep a better overview:

$$
\begin{aligned}
&a\frac{2b^2-2b\sqrt{b^2-4ac}-4ac}{4a^2}\\
&=\frac{1}{2a}(b^2-b\sqrt{b^2-4ac}-2ac)
\end{aligned}
$$

Here we resubstitute the terms a, b, c:

$$
\begin{aligned}
=\frac{1}{2(-p_1e_{11}^2-p_2e_{21}^2)}\Bigg(&(e_{11}-p_1 2w_2e_{11}e_{12}-2p_2w_2e_{21}e_{22})^2\\
&-(e_{11}-p_1 2w_2e_{11}e_{12}-2p_2w_2e_{21}e_{22})\\
&\cdot\sqrt{(e_{11}-p_1 2w_2e_{11}e_{12}-2p_2w_2e_{21}e_{22})^2-4(-p_1e_{11}^2-p_2e_{21}^2)(-p_2w_2^2e_{22}^2+w_2e_{12}-p_1w_2^2e_{12}^2)}\\
&-2(-p_1e_{11}^2-p_2e_{21}^2)(-p_2w_2^2e_{22}^2+w_2e_{12}-p_1w_2^2e_{12}^2)\Bigg)
\end{aligned}
$$

Since a does not contain the term w_2, we will continue using it throughout for simplicity:

$$= \frac{(e_{11} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})^2}{2a} - \frac{(e_{11} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})}{2a}$$
$$\cdot \frac{\sqrt{(e_{11} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})^2 - 4(-p_1 e_{11}^2 - p_2 e_{21}^2)(-p_2 w_2^2 e_{22}^2 + w_2 e_{12} - p_1 w_2^2 e_{12}^2)}}{2a} - \frac{2a(-p_2 w_2^2 e_{22}^2 + w_2 e_{12} - p_1 w_2^2 e_{12}^2)}{2a}$$

$$= \frac{(e_{11} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})^2}{2a} - \frac{(e_{11} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})}{2a}$$
$$\cdot \frac{\sqrt{(e_{11} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})^2 - 4(-p_1 e_{11}^2 - p_2 e_{21}^2)(-p_2 w_2^2 e_{22}^2 + w_2 e_{12} - p_1 w_2^2 e_{12}^2)}}{2a} + p_2 w_2^2 e_{22}^2 - w_2 e_{12} + p_1 w_2^2 e_{12}^2$$

Again, we first focus on the first term and solve it:

$$\frac{(e_{11} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})(e_{11} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})}{2a}$$

$$= \frac{1}{2a}(e_{11}(e_{11} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22}) - p_1 2w_2 e_{11} e_{12}(e_{11} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22}) - 2p_2 w_2 e_{21} e_{22}(e_{11} - p_1 2w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22}))$$

$$= \frac{e_{11}^2}{2a} - \frac{2p_1 w_2 e_{11}^2 e_{12}}{2a} - \frac{2p_2 w_2 e_{11} e_{21} e_{22}}{2a} - \frac{2p_1 w_2 e_{11}^2 e_{12}}{2a} + \frac{4p_1^2 w_2^2 e_{11}^2 e_{12}^2}{2a} + \frac{4p_1 p_2 w_2^2 e_{11} e_{12} e_{21} e_{22}}{2a} - \frac{2p_2 w_2 e_{11} e_{21} e_{22}}{2a} + \frac{4p_1 p_2 w_2^2 e_{11} e_{12} e_{21} e_{22}}{2a} + \frac{4p_2^2 w_2^2 e_{21}^2 e_{22}^2}{2a}$$

Here, we rejoin all resulting terms of the third and fourth terms:

$$
\begin{aligned}
&\frac{e_{11}^2}{2a} - \frac{2p_1 w_2 e_{11}^2 e_{12}}{2a} - \frac{2p_2 w_2 e_{11} e_{21} e_{22}}{2a} \\
&\quad - \frac{2p_1 w_2 e_{11}^2 e_{12}}{2a} + \frac{4p_1^2 w_2^2 e_{11}^2 e_{12}^2}{2a} + \frac{4p_1 p_2 w_2^2 e_{11} e_{12} e_{21} e_{22}}{2a} \\
&\quad - \frac{2p_2 w_2 e_{11} e_{21} e_{22}}{2a} + \frac{4p_1 p_2 w_2^2 e_{11} e_{12} e_{21} e_{22}}{2a} + \frac{4p_2^2 w_2^2 e_{21}^2 e_{22}^2}{2a} \\
&\quad - \frac{(e_{11} - p_1 2 w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})}{2a} \\
&\cdot \frac{\sqrt{(e_{11} - p_1 2 w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})^2 - 4(-p_1 e_{11}^2 - p_2 e_{21}^2)(-p_2 w_2^2 e_{22}^2 + w_2 e_{12} - p_1 w_2^2 e_{12}^2)}}{2a} \\
&\quad + p_2 w_2^2 e_{22}^2 - w_2 e_{12} + p_1 w_2^2 e_{12}^2
\end{aligned}
$$

Lastly, we rejoin all terms resulting from the equation substituted on page 30:

$$
\begin{aligned}
&\frac{e_{11}^2}{2a} - \frac{2p_1 w_2 e_{11}^2 e_{12}}{2a} - \frac{2p_2 w_2 e_{11} e_{21} e_{22}}{2a} \\
&\quad - \frac{2p_1 w_2 e_{11}^2 e_{12}}{2a} + \frac{4p_1^2 w_2^2 e_{11}^2 e_{12}^2}{2a} + \frac{4p_1 p_2 w_2^2 e_{11} e_{12} e_{21} e_{22}}{2a} \\
&\quad - \frac{2p_2 w_2 e_{11} e_{21} e_{22}}{2a} + \frac{4p_1 p_2 w_2^2 e_{11} e_{12} e_{21} e_{22}}{2a} + \frac{4p_2^2 w_2^2 e_{21}^2 e_{22}^2}{2a} \\
&\quad - \frac{e_{11} e_{21}}{2a} + \frac{p_1 2 w_2 e_{11}^2 e_{12}}{2a} + \frac{2e_{11} p_2 w_2 e_{21} e_{22}}{2a} \\
&\quad + \frac{2e_{21} p_1 w_2 e_{11} e_{12}}{2a} - \frac{4p_1^2 w_2^2 e_{11}^2 e_{12}^2}{2a} - \frac{4p_1 p_2 w_2^2 e_{11} e_{12} e_{21} e_{22}}{2a} \\
&\quad + \frac{2p_2 w_2 e_{21} e_{22} e_{21}}{2a} - \frac{4p_1 p_2 w_2^2 e_{11} e_{12} e_{21} e_{22}}{2a} - \frac{4p_2^2 w_2^2 e_{21}^2 e_{22}^2}{2a} \\
&\quad - \frac{(e_{11} - p_1 2 w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})}{2a} \\
&\cdot \frac{\sqrt{(e_{11} - p_1 2 w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})^2 - 4(-p_1 e_{11}^2 - p_2 e_{21}^2)(-p_2 w_2^2 e_{22}^2 + w_2 e_{12} - p_1 w_2^2 e_{12}^2)}}{2a} \\
&+ \frac{\sqrt{(e_{11} - p_1 2 w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})^2 - 4(-p_1 e_{11}^2 - p_2 e_{21}^2)(-p_2 w_2^2 e_{22}^2 + w_2 e_{12} - p_1 w_2^2 e_{12}^2}}{2a} \\
&\quad \cdot \frac{(e_{21} - p_1 2 w_2 e_{11} e_{12} - 2p_2 w_2 e_{21} e_{22})}{2a} \\
&\quad + p_2 w_2^2 e_{22}^2 - w_2 e_{12} + p_1 w_2^2 e_{12}^2 \\
&\quad + w_2 e_{22} - p_1 w_2^2 e_{12}^2 - p_2 w_2^2 e_{22}^2 = 0
\end{aligned}
$$

In the next step, we removed terms that cancel out in the previous equation, to get:

$$
\begin{aligned}
&\frac{e_{11}^2}{2a} - \frac{e_{11}e_{21}}{2a} - \frac{2p_1w_2e_{11}^2e_{12}}{2a} + \frac{2p_1w_2e_{11}e_{12}e_{21}}{2a} \\
&\qquad - \frac{2p_2w_2e_{11}e_{21}e_{22}}{2a} + \frac{2p_2w_2e_{21}e_{22}e_{21}}{2a} - w_2e_{12} + w_2e_{22} \\
&\qquad - \frac{(e_{11} - p_1 2w_2e_{11}e_{12} - 2p_2w_2e_{21}e_{22})}{2a} \\
&\cdot \frac{\sqrt{(e_{11} - p_1 2w_2e_{11}e_{12} - 2p_2w_2e_{21}e_{22})^2 - 4(-p_1e_{11}^2 - p_2e_{21}^2)(-p_2w_2^2e_{22}^2 + w_2e_{12} - p_1w_2^2e_{12}^2)}}{2a} \\
&+ \frac{\sqrt{(e_{11} - p_1 2w_2e_{11}e_{12} - 2p_2w_2e_{21}e_{22})^2 - 4(-p_1e_{11}^2 - p_2e_{21}^2)(-p_2w_2^2e_{22}^2 + w_2e_{12} - p_1w_2^2e_{12}^2}}{2a} \\
&\qquad \cdot \frac{(e_{21} - p_1 2w_2e_{11}e_{12} - 2p_2w_2e_{21}e_{22})}{2a} = 0
\end{aligned}
$$

To minimize the the number of times that $2a$ is in the equation, we multiply everything by $2a$.

Furthermore, we join the last two terms by factoring out the square root term.

$$
\begin{aligned}
&e_{11}^2 - e_{11}e_{21} - 2p_1w_2e_{11}^2e_{12} + 2p_1w_2e_{11}e_{12}e_{21} - 2p_2w_2e_{11}e_{21}e_{22} + 2p_2w_2e_{21}e_{22}e_{21} - \frac{w_2e_{12}}{2a} + \frac{w_2e_{22}}{2a} \\
&+ \sqrt{(e_{11} - p_1 2w_2e_{11}e_{12} - 2p_2w_2e_{21}e_{22})^2 - 4(-p_1e_{11}^2 - p_2e_{21}^2)(-p_2w_2^2e_{22}^2 + w_2e_{12} - p_1w_2^2e_{12}^2)} \\
&\quad \cdot ((e_{21} - p_1 2w_2e_{11}e_{12} - 2p_2w_2e_{21}e_{22}) - (e_{11} - p_1 2w_2e_{11}e_{12} - 2p_2w_2e_{21}e_{22})) = 0
\end{aligned}
$$

We solve the third line of the previous equation

$$
\begin{aligned}
&e_{11}^2 - e_{11}e_{21} - 2p_1w_2e_{11}^2e_{12} + 2p_1w_2e_{11}e_{12}e_{21} - 2p_2w_2e_{11}e_{21}e_{22} + 2p_2w_2e_{21}e_{22}e_{21} - \frac{w_2e_{12}}{2a} + \frac{w_2e_{22}}{2a} \\
&+ \sqrt{(e_{11} - p_1 2w_2e_{11}e_{12} - 2p_2w_2e_{21}e_{22})^2 - 4(-p_1e_{11}^2 - p_2e_{21}^2)(-p_2w_2^2e_{22}^2 + w_2e_{12} - p_1w_2^2e_{12}^2)} \\
&\qquad \cdot (e_{21} - e_{11}) = 0
\end{aligned}
$$

Moreover, we take the square root term to the other side of the equation:

$$e_{11}^2 - e_{11}e_{21} - 2p_1w_2e_{11}^2e_{12} + 2p_1w_2e_{11}e_{12}e_{21}$$
$$-2p_2w_2e_{11}e_{21}e_{22} + 2p_2w_2e_{21}e_{22}e_{21} - \frac{w_2e_{12}}{2a} + \frac{w_2e_{22}}{2a}$$
$$=$$
$$\sqrt{(e_{11} - p_1 2w_2e_{11}e_{12} - 2p_2w_2e_{21}e_{22})^2 - 4a(-p_2w_2^2e_{22}^2 + w_2e_{12} - p_1w_2^2e_{12}^2)}$$
$$\cdot (e_{11} - e_{21})$$

We then square both sides to get rid of the square root:

$$(e_{11} - p_1 2w_2e_{11}e_{12} - 2p_2w_2e_{21}e_{22})^2$$
$$-4a(-p_2w_2^2e_{22}^2 + w_2e_{12} - p_1w_2^2e_{12}^2)(e_{11} - e_{21})^2$$
$$=$$
$$(e_{11}^2 - e_{11}e_{21} - 2p_1w_2e_{11}^2e_{12} + 2p_1w_2e_{11}e_{12}e_{21}$$
$$-2p_2w_2e_{11}e_{21}e_{22} + 2p_2w_2e_{21}e_{22}e_{21} - \frac{w_2e_{12}}{2a} + \frac{w_2e_{22}}{2a})^2$$

We first solve the left side of the equation in isolation

$$(e_{11} - p_1 2w_2e_{11}e_{12} - 2p_2w_2e_{21}e_{22})^2 - 4a(-p_2w_2^2e_{22}^2 + w_2e_{12} - p_1w_2^2e_{12}^2)(e_{11} - e_{21})^2$$

$$= e_{11}^2 - 2p_1w_2e_{11}^2e_{12} - 2p_2w_2e_{11}e_{21}e_{22} - 2p_1w_2e_{11}^2e_{12} + 4p_1^2w_2^2e_{11}^2e_{12}^2$$
$$+4p_1p_2w_2^2e_{11}e_{12}e_{21}e_{22} - 2p_2w_2e_{11}e_{21}e_{22} + 4p_1p_2w_2^2e_{11}e_{12}e_{21}e_{22} + 4p_2^2w_2^2e_{21}^2e_{22}^2$$
$$-(4ap_2w_2^2e_{22}^2 + 4aw_2e_{12} - 4ap_1w_2^2e_{12}^2)(e_{11}^2 - 2e_{11}e_{21} + e_{21}^2)$$

To keep the overview we now solve the third line of the previous equation in isolation

$$4ap_2w_2^2e_{22}^2(e_{11}^2 - 2e_{11}e_{21} + e_{21}^2)$$
$$-4aw_2e_{12}(e_{11}^2 - 2e_{11}e_{21} + e_{21}^2)$$
$$+4ap_1w_2^2e_{12}^2(e_{11}^2 - 2e_{11}e_{21} + e_{21}^2)$$

$$= 4ap_2w_2^2e_{11}^2e_{22}^2 - 8ap_2w_2^2e_{11}e_{21}e_{22}^2 - 4ap_2w_2^2e_{21}^2e_{22}^2 - 4aw_2e_{11}^2e_{12}$$
$$+ 8aw_2e_{11}e_{12}e_{21} - 4aw_2e_{12}e_{21}^2 + 4ap_1w_2^2e_{11}^2e_{12}^2 - 8ap_1w_2^2e_{11}e_{12}^2e_{21} + 4ap_1w_2^2e_{12}^2e_{21}^2$$

Then we rejoin all the terms of the left side of the equation:

$$e_{11}^2 - 2p_1w_2e_{11}^2e_{12} - 2p_2w_2e_{11}e_{21}e_{22} - 2p_1w_2e_{11}^2e_{12} + 4p_1^2w_2^2e_{11}^2e_{12}^2$$
$$+ 4p_1p_2w_2^2e_{11}e_{12}e_{21}e_{22} - 2p_2w_2e_{11}e_{21}e_{22} + 4p_1p_2w_2^2e_{11}e_{12}e_{21}e_{22} + 4p_2^2w_2^2e_{21}^2e_{22}^2$$
$$+ 4ap_2w_2^2e_{11}^2e_{22}^2 - 8ap_2w_2^2e_{11}e_{21}e_{22}^2 - 4ap_2w_2^2e_{21}^2e_{22}^2 - 4aw_2e_{11}^2e_{12}$$
$$+ 8aw_2e_{11}e_{12}e_{21} - 4aw_2e_{12}e_{21}^2 + 4ap_1w_2^2e_{11}^2e_{12}^2 - 8ap_1w_2^2e_{11}e_{12}^2e_{21} + 4ap_1w_2^2e_{12}^2e_{21}^2$$

Now we move on to solve the right side of the equation:

$$(e_{11}^2 - e_{11}e_{21} - 2p_1w_2e_{11}^2e_{12} + 2p_1w_2e_{11}e_{12}e_{21} - 2p_2w_2e_{11}e_{21}e_{22} + 2p_2w_2e_{21}e_{22}e_{21} - \frac{w_2e_{12}}{2a} + \frac{w_2e_{22}}{2a})^2$$

$$\begin{aligned}
= e_{11}^2(e_{11}^2 - e_{11}e_{21} - 2p_1w_2e_{11}^2e_{12} + 2p_1w_2e_{11}e_{12}e_{21} \\
- 2p_2w_2e_{11}e_{21}e_{22} + 2p_2w_2e_{21}e_{22}e_{21} - \frac{w_2e_{12}}{2a} + \frac{w_2e_{22}}{2a})
\end{aligned}$$

$$\begin{aligned}
- e_{11}e_{21}(e_{11}^2 - e_{11}e_{21} - 2p_1w_2e_{11}^2e_{12} + 2p_1w_2e_{11}e_{12}e_{21} \\
- 2p_2w_2e_{11}e_{21}e_{22} + 2p_2w_2e_{21}e_{22}e_{21} - \frac{w_2e_{12}}{2a} + \frac{w_2e_{22}}{2a})
\end{aligned}$$

$$\begin{aligned}
- 2p_1w_2e_{11}^2e_{12}(e_{11}^2 - e_{11}e_{21} - 2p_1w_2e_{11}^2e_{12} + 2p_1w_2e_{11}e_{12}e_{21} \\
- 2p_2w_2e_{11}e_{21}e_{22} + 2p_2w_2e_{21}e_{22}e_{21} - \frac{w_2e_{12}}{2a} + \frac{w_2e_{22}}{2a})
\end{aligned}$$

$$\begin{aligned}
+ 2p_1w_2e_{11}e_{12}e_{21}(e_{11}^2 - e_{11}e_{21} - 2p_1w_2e_{11}^2e_{12} + 2p_1w_2e_{11}e_{12}e_{21} \\
- 2p_2w_2e_{11}e_{21}e_{22} + 2p_2w_2e_{21}e_{22}e_{21} - \frac{w_2e_{12}}{2a} + \frac{w_2e_{22}}{2a})
\end{aligned}$$

$$\begin{aligned}
- 2p_2w_2e_{11}e_{21}e_{22}(e_{11}^2 - e_{11}e_{21} - 2p_1w_2e_{11}^2e_{12} + 2p_1w_2e_{11}e_{12}e_{21} \\
- 2p_2w_2e_{11}e_{21}e_{22} + 2p_2w_2e_{21}e_{22}e_{21} - \frac{w_2e_{12}}{2a} + \frac{w_2e_{22}}{2a})
\end{aligned}$$

$$\begin{aligned}
+ 2p_2w_2e_{21}e_{22}e_{21}(e_{11}^2 - e_{11}e_{21} - 2p_1w_2e_{11}^2e_{12} + 2p_1w_2e_{11}e_{12}e_{21} \\
- 2p_2w_2e_{11}e_{21}e_{22} + 2p_2w_2e_{21}e_{22}e_{21} - \frac{w_2e_{12}}{2a} + \frac{w_2e_{22}}{2a})
\end{aligned}$$

$$\begin{aligned}
- \frac{w_2e_{12}}{2a}(e_{11}^2 - e_{11}e_{21} - 2p_1w_2e_{11}^2e_{12} + 2p_1w_2e_{11}e_{12}e_{21} \\
- 2p_2w_2e_{11}e_{21}e_{22} + 2p_2w_2e_{21}e_{22}e_{21} - \frac{w_2e_{12}}{2a} + \frac{w_2e_{22}}{2a})
\end{aligned}$$

$$\begin{aligned}
+ \frac{w_2e_{22}}{2a}(e_{11}^2 - e_{11}e_{21} - 2p_1w_2e_{11}^2e_{12} + 2p_1w_2e_{11}e_{12}e_{21} \\
- 2p_2w_2e_{11}e_{21}e_{22} + 2p_2w_2e_{21}e_{22}e_{21} - \frac{w_2e_{12}}{2a} + \frac{w_2e_{22}}{2a})
\end{aligned}$$

$$
\begin{aligned}
&= e_{11}^4 - e_{11}^3 e_{21} - 2p_1 w_2 e_{11}^4 e_{12} + 2p_1 w_2 e_{11}^3 e_{12} e_{21} - 2p_2 w_2 e_{11}^3 e_{21} e_{22} \\
&\quad + 2p_2 w_2 e_{11}^2 e_{21} e_{22} e_{21} - \frac{w_2 e_{11}^2 e_{12}}{2a} + \frac{w_2 e_{11}^2 e_{22}}{2a} \\
&\quad - e_{21} e_{11}^3 + e_{11}^2 e_{21}^2 + 2p_1 w_2 e_{11}^3 e_{12} e_{21} - 2p_1 w_2 e_{11}^2 e_{12} e_{21}^2 \\
&\quad + 2p_2 w_2 e_{11}^2 e_{21}^2 e_{22} - 2p_2 w_2 e_{11} e_{21}^2 e_{22} e_{21} + \frac{w_2 e_{11} e_{12} e_{21}}{2a} - \frac{w_2 e_{11} e_{21} e_{22}}{2a} \\
&\quad - 2p_1 w_2 e_{11}^4 e_{12} + 2p_1 w_2 e_{11}^3 e_{12} e_{21} + 4p_1^2 w_2^2 e_{11}^2 e_{12}^2 e_{11}^2 - 4p_1^2 w_2^2 e_{11}^3 e_{12}^2 e_{21} \\
&\quad + 4p_1 p_2 w_2^2 e_{11}^3 e_{12} e_{21} e_{22} - 4p_1 p_2 w_2^2 e_{11}^2 e_{12} e_{21}^2 e_{22} + \frac{p_1 e_{11}^2 w_2^2 e_{12}^2}{a} - \frac{p_1 w_2^2 e_{11}^2 e_{12} e_{22}}{a} \\
&\quad + 2p_1 w_2 e_{11}^3 e_{12} e_{21} - 2p_1 w_2 e_{11}^2 e_{12} e_{21}^2 - 4p_1^2 w_2^2 e_{11}^3 e_{12}^2 e_{21} + 4p_1^2 w_2^2 e_{11}^2 e_{12}^2 e_{21}^2 \\
&\quad - 4p_1 p_2 w_2^2 e_{11}^2 e_{12} e_{21}^2 e_{22} + 4p_1 p_2 w_2^2 e_{11} e_{12} e_{21}^3 e_{22} - \frac{p_1 w_2^2 e_{11} e_{12}^2 e_{21}}{a} + \frac{p_1 w_2^2 e_{11} e_{12} e_{21} e_{22}}{a} \\
&\quad - 2p_2 w_2 e_{11}^3 e_{21} e_{22} + 2p_2 w_2 e_{11}^2 e_{21}^2 e_{22} + 4p_1 p_2 w_2^2 e_{11}^3 e_{12} e_{21} e_{22} - 4p_1 p_2 w_2^2 e_{11}^2 e_{12} e_{21}^2 e_{22} \\
&\quad + 4p_2^2 w_2^2 e_{11}^2 e_{21}^2 e_{22}^2 - 2p_2^2 w_2^2 e_{11} e_{21}^3 e_{22} + \frac{p_2 w_2^2 e_{11} e_{21} e_{22}}{a} - \frac{p_2 w_2^2 e_{11} e_{21} e_{22}^2}{a} \\
&\quad + 2p_2 w_2 e_{11}^2 e_{21} e_{22} e_{21} - 2p_2 w_2 e_{11} e_{21}^3 e_{22} - 4p_1 p_2 w_2^2 e_{11}^2 e_{12} e_{21}^2 e_{22} + 4p_1 p_2 w_2^2 e_{11} e_{12} e_{21}^3 e_{22} \\
&\quad - 4p_2^2 w_2^2 e_{11} e_{21}^3 e_{22}^2 + 4p_2^2 w_2^2 e_{21}^4 e_{22}^2 - \frac{p_2 w_2^2 e_{12} e_{21}^2 e_{22}}{a} + \frac{p_2 w_2^2 e_{21}^2 e_{22}^2}{a} \\
&\quad - \frac{w_2 e_{11}^2 e_{12}}{2a} + \frac{w_2 e_{11} e_{12} e_{21}}{2a} + \frac{p_1 w_2^2 e_{11}^2 e_{12}^2}{a} - \frac{p_1 w_2^2 e_{11} e_{12}^2 e_{21}}{a} \\
&\quad + \frac{p_2 w_2^2 e_{11} e_{12} e_{21} e_{22}}{a} - \frac{p_2 w_2^2 e_{12} e_{21}^2 e_{22}}{a} + \frac{w_2^2 e_{12}^2}{4a^2} - \frac{w_2^2 e_{12} e_{22}}{4a^2} \\
&\quad + \frac{w_2 e_{11}^2 e_{22}}{2a} - \frac{w_2 e_{11} e_{21} e_{22}}{2a} - \frac{p_1 w_2^2 e_{11}^2 e_{12} e_{22}}{a} + \frac{p_1 w_2^2 e_{11} e_{12} e_{21} e_{22}}{a} \\
&\quad - \frac{p_2 w_2^2 e_{11} e_{21} e_{22}^2}{a} + \frac{p_2 w_2^2 e_{21}^2 e_{22}^2}{a} - \frac{w_2^2 e_{12} e_{22}}{4a^2} + \frac{w_2^2 e_{22}^2}{4a^2}
\end{aligned}
$$

We join the right and left sides of the equation, such that

$$
\begin{gathered}
e_{11}^2 - 2p_1w_2e_{11}^2e_{12} - 2p_2w_2e_{11}e_{21}e_{22} - 2p_1w_2e_{11}^2e_{12} + 4p_1^2w_2^2e_{11}^2e_{12}^2 \\
+ 4p_1p_2w_2^2e_{11}e_{12}e_{21}e_{22} - 2p_2w_2e_{11}e_{21}e_{22} + 4p_1p_2w_2^2e_{11}e_{12}e_{21}e_{22} + 4p_2^2w_2^2e_{21}^2e_{22}^2 \\
+ 4ap_2w_2^2e_{11}^2e_{22}^2 - 8ap_2w_2^2e_{11}e_{21}e_{22}^2 - 4ap_2w_2^2e_{21}^2e_{22}^2 - 4aw_2e_{11}^2e_{12} \\
+ 8aw_2e_{11}e_{12}e_{21} - 4aw_2e_{12}e_{21}^2 + 4ap_1w_2^2e_{11}^2e_{12}^2 - 8ap_1w_2^2e_{11}e_{12}^2e_{21} + 4ap_1w_2^2e_{12}^2e_{21}^2 \\
= \\
e_{11}^4 - e_{11}^3e_{21} - 2p_1w_2e_{11}^4e_{12} + 2p_1w_2e_{11}^3e_{12}e_{21} \\
- 2p_2w_2e_{11}^3e_{21}e_{22} + 2p_2w_2e_{11}^2e_{21}e_{22}e_{21} - \frac{w_2e_{11}^2e_{12}}{2a} + \frac{w_2e_{11}^2e_{22}}{2a} \\
- e_{21}e_{11}^3 + e_{11}^2e_{21}^2 + 2p_1w_2e_{11}^3e_{12}e_{21} - 2p_1w_2e_{11}^2e_{12}e_{21}^2 \\
+ 2p_2w_2e_{11}^2e_{21}^2e_{22} - 2p_2w_2e_{11}e_{21}^2e_{22}e_{21} + \frac{w_2e_{11}e_{12}e_{21}}{2a} - \frac{w_2e_{11}e_{21}e_{22}}{2a} \\
- 2p_1w_2e_{11}^4e_{12} + 2p_1w_2e_{11}^3e_{12}e_{21} + 4p_1^2w_2^2e_{11}^2e_{12}^2e_{11}^2 - 4p_1^2w_2^2e_{11}^3e_{12}^2e_{21} \\
+ 4p_1p_2w_2^2e_{11}^3e_{12}e_{21}e_{22} - 4p_1p_2w_2^2e_{11}^2e_{12}e_{21}^2e_{22} + \frac{p_1e_{11}^2w_2^2e_{12}^2}{a} - \frac{p_1w_2^2e_{11}^2e_{12}e_{22}}{a} \\
+ 2p_1w_2e_{11}^3e_{12}e_{21} - 2p_1w_2e_{11}^2e_{12}e_{21}^2 - 4p_1^2w_2^2e_{11}^3e_{12}^2e_{21} + 4p_1^2w_2^2e_{11}^2e_{12}^2e_{21}^2 \\
- 4p_1p_2w_2^2e_{11}^2e_{12}e_{21}^2e_{22} + 4p_1p_2w_2^2e_{11}e_{12}e_{21}^3e_{22} - \frac{p_1w_2^2e_{11}e_{12}^2e_{21}}{a} + \frac{p_1w_2^2e_{11}e_{12}e_{21}e_{22}}{a} \\
- 2p_2w_2e_{11}^3e_{21}e_{22} + 2p_2w_2e_{11}^2e_{21}^2e_{22} + 4p_1p_2w_2^2e_{11}^3e_{12}e_{21}e_{22} - 4p_1p_2w_2^2e_{11}^2e_{12}e_{21}^2e_{22} \\
+ 4p_2^2w_2^2e_{11}^2e_{21}^2e_{22}^2 - 2p_2^2w_2^2e_{11}e_{21}^3e_{22} + \frac{p_2w_2^2e_{11}e_{21}e_{22}}{a} - \frac{p_2w_2^2e_{11}e_{21}e_{22}^2}{a} \\
+ 2p_2w_2e_{11}^2e_{21}e_{22}e_{21} - 2p_2w_2e_{11}e_{21}^3e_{22} - 4p_1p_2w_2^2e_{11}^2e_{12}e_{21}^2e_{22} + 4p_1p_2w_2^2e_{11}e_{12}e_{21}^3e_{22} \\
- 4p_2^2w_2^2e_{11}e_{21}^3e_{22}^2 + 4p_2^2w_2^2e_{21}^4e_{22}^2 - \frac{p_2w_2^2e_{12}e_{21}^2e_{22}}{a} + \frac{p_2w_2^2e_{21}^2e_{22}^2}{a} \\
- \frac{w_2e_{11}^2e_{12}}{2a} + \frac{w_2e_{11}e_{12}e_{21}}{2a} + \frac{p_1w_2^2e_{11}^2e_{12}^2}{a} - \frac{p_1w_2^2e_{11}e_{12}^2e_{21}}{a} \\
+ \frac{p_2w_2^2e_{11}e_{12}e_{21}e_{22}}{a} - \frac{p_2w_2^2e_{12}e_{21}^2e_{22}}{a} + \frac{w_2^2e_{12}^2}{4a^2} - \frac{w_2^2e_{12}e_{22}}{4a^2} \\
+ \frac{w_2e_{11}^2e_{22}}{2a} - \frac{w_2e_{11}e_{21}e_{22}}{2a} - \frac{p_1w_2^2e_{11}^2e_{12}e_{22}}{a} + \frac{p_1w_2^2e_{11}e_{12}e_{21}e_{22}}{a} \\
- \frac{p_2w_2^2e_{11}e_{21}e_{22}^2}{a} + \frac{p_2w_2^2e_{21}^2e_{22}^2}{a} - \frac{w_2^2e_{12}e_{22}}{4a^2} + \frac{w_2^2e_{22}^2}{4a^2}
\end{gathered}
$$

Moving all terms to one side, factoring out w_2^2 and w_2 and simplifying, we get

$$
\begin{aligned}
0 = {w_2}^2(&4p_1^2e_{11}^4e_{12}^2 - 8p_1^2e_{11}^3e_{12}^2e_{21} + 8p_1p_2e_{11}^3e_{12}e_{21}e_{22} - 16p_1p_2e_{11}^2e_{12}e_{21}^2e_{22} \\
&+2\frac{p_1e_{11}^2e_{12}^2}{a} - \frac{p_1e_{11}^2e_{12}e_{22}}{a} + 4p_1^2e_{11}^2e_{12}^2e_{21}^2 + 4p_1p_2e_{11}e_{12}e_{21}^3e_{22} \\
&-\frac{2p_1e_{11}e_{12}^2e_{21}}{a} + 2\frac{p_1e_{11}e_{12}e_{21}e_{22}}{a} + 4p_2^2e_{11}^2e_{21}^2e_{22}^2 - 2p_2^2e_{11}e_{21}^3e_{22} \\
&+\frac{p_2e_{11}e_{21}e_{22}}{a} - \frac{2p_2e_{11}e_{21}e_{22}^2}{a} + 4p_1p_2e_{11}e_{12}e_{21}^3e_{22} - 4p_2^2e_{11}e_{21}^3e_{22}^2 \\
&+4p_2^2e_{21}^4e_{22}^2 - 2\frac{p_2e_{12}e_{21}^2e_{22}}{a} + 2\frac{p_2e_{21}^2e_{22}^2}{a} + \frac{p_2e_{11}e_{12}e_{21}e_{22}}{a} \\
&+\frac{e_{12}^2}{4a^2} - \frac{2e_{12}e_{22}}{4a^2} - \frac{p_1e_{11}^2e_{12}e_{22}}{a} + \frac{e_{22}^2}{4a^2} - 4{p_1}^2e_{11}^2e_{12}^2 \\
&-8p_1p_2e_{11}e_{12}e_{21}e_{22} - 4p_2^2e_{21}^2e_{22}^2 - 4ap_2e_{11}^2e_{22}^2 + 8ap_2e_{11}e_{21}e_{22}^2 \\
&+4ap_2e_{21}^2e_{22}^2 - 4ap_1e_{11}^2e_{12}^2 + 8ap_1e_{11}e_{12}^2e_{21} - 4ap_1e_{12}^2e_{21}^2) \\
+w_2(&-4p_1e_{11}^4e_{12} + 2p_1e_{11}^3e_{12}e_{21} - 2p_2e_{11}^3e_{21}e_{22} + 2p_2e_{11}^2e_{21}e_{22}e_{21} - \frac{e_{11}^2e_{12}}{a} + \frac{e_{11}^2e_{22}}{a} \\
&+2p_1e_{11}^3e_{12}e_{21} - 2p_1e_{11}^2e_{12}e_{21}^2 + 6p_2e_{11}^2e_{21}^2e_{22} - 2p_2e_{11}e_{21}^2e_{22}e_{21} + \frac{e_{11}e_{12}e_{21}}{a} - \frac{e_{11}e_{21}e_{22}}{a} \\
&+4p_1e_{11}^3e_{12}e_{21} - 2p_1e_{11}^2e_{12}e_{21}^2 - 2p_2e_{11}^3e_{21}e_{22} - 2p_2e_{11}e_{21}^3e_{22} \\
&+2p_1e_{11}^2e_{12} + 4p_2e_{11}e_{21}e_{22} + 2p_1e_{11}^2e_{12} + 4ae_{11}^2e_{12} - 8ae_{11}e_{12}e_{21}) \\
&-2e_{21}e_{11}^3 + e_{11}^2e_{21}^2 + e_{11}^4 - e_{11}^2
\end{aligned}
$$

Lastly we may use the quadratic formula to solve this equation as well, assuming that $x \neq 0$, such that

$$w_{21,22} = \frac{-y \pm \sqrt{y^2 - 4xz}}{2x}$$

Here x is what is in brackets after w_2^2, y is what is in brackets after w_2 and z are the leftover terms.

Bibliography

[1] Elie L. Bienenstock, Leon N. Cooper, and Paul W. Munro. Theory for the Development of Neuron Selectivity: Orientation Specificity and Binocular Interaction in the Visual Cortex. *The Journal of Neuroscience*, 2(1):32–48, 1982.

[2] Dr.-Ing. Marco Breiling. Neuromorphic Hardware, 2022.

[3] Leon N. Cooper, Fishel Liberman, and Erkki Oja. A Theory for the Acquisition and Loss of Neuron Specificity in Visual Cortex. *Biological Cybernetics*, 33:9–28, 1979.

[4] Neil R. Euliano. *Neural and Adaptive Systems: Fundamentals Through Simulations*. Wiley, 2 edition, December 1999.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*, volume 1. Massachusetts Institute of Technology, 2016.

[6] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2. edition, 1998.

[7] D.O. Hebb. The Organization of Behaviour. *New York: Wiley & Sons*, 1949.

[8] Kurt Hornik. Approximation Capabilities of Multilayer Feedforward Networks. *Neural Networks*, 4:251–257, 1991.

[9] Nathan Intrator. Feature Extraction Using an Unsupervised Neural Network. *Neural Computation*, 4:98–107, 1992.

[10] Erkki Oja. A simplified Neuron Model as a Principal Component Analyzer. *Journal of Mathematical Biology*, 15(3):267–273, 1982.

[11] Thomas P. Trappenberg. *Fundamentals of Computational Neuroscience*. OUP Oxford, 2. edition, 2009.

[12] Thomas Villmann. *Computational Intelligence I - Neural Computation*.

[13] Thomas Villmann. *Mathematik fuer Molekulare Biotechnologie/Bioinformatik*. 3.02 edition, 2020.

Erklärung

Hiermit erkläre ich, dass ich meine Arbeit selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die Arbeit noch nicht anderweitig für Prüfungszwecke vorgelegt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

[redacted]

Mittweida, 03.08.2022