University of Wrocław
Faculty of Physics and Astronomy

Bartłomiej Domański

Implementation of associative memory models using neural networks

Realizacja modeli pamięci skojarzeniowej przy pomocy sieci neuronowych

Bachelor thesis carried out under the supervision of Prof. Krzysztof Graczyk at the UWr Institute of Theoretical Physics

Wrocław 2024

Contents

1 Introduction **5**

2 Classical Hopfield network **7**

2.1 Energy function 8

2.1.1 Crosstalk 9

2.2 Analogy to Ising's model 11

3 Dense associative memory **15**

3.1 Exclusive or - $S > N$ 16

4 Continous Hopfield network **19**

Chapter 1

Introduction

The origins of the development of artificial intelligence date back to the 1940s, when in 1943 Warren McCulloch and Walter Pitts proposed a model of an artificial neuron, called the perceptron. The perceptron was the first attempt at a mathematical model of a biological neuron to simulate human thought processes. The model involved summing the signals reaching the perceptron and, based on this, determining whether the perceptron was active or inactive. In the 1950s and 1960s, the limitations of the perceptron were demonstrated; it turned out that a single perceptron was incapable of solving nonlinear problems. This discovery, combined with a lack of adequate data and computing power, led to a decline in interest in neural networks.During this period, most research focused on other approaches to artificial intelligence.It was not until the 1980s, thanks to John Hopfield's work on recurrent networks, that interest in artificial neural networks was revived. Hopfield's networks brought new insights into the potential of artificial neural networks, laying the foundation for further research into machine learning and associative memory (addressed content). Today, artificial neural networks are considered the most important tool in the field of artificial intelligence, enabling significant advances in image recognition, natural language processing, robotics and many other fields. Thanks to advanced machine learning algorithms, they are now able to perform tasks of a complex nature that not long ago seemed impossible to automate. Despite intensive research on the subject, there are still many artificial intelligence mechanisms in a backward phase of development which suggests many new discoveries in the future.

The most important objective of this paper is a theoretical and practical comparison of models of associative memory in the form of artificial neural networks - the classical Hopfield network, dense associative memory and continuous Hopfield network. The basis of all three is the idea proposed by John Hopfield of describing an artificial neural network through an energy function. The essence of associative memory is the ability to reconstruct data based on parts of that data or damaged fragments. In the human brain, this mechanism is related to the ability to combine different elements of information and form associations between them, a process that allows one to recall complete memories or sequences of events based on a single element or piece of information. A key role in this process is played by connections between nerve cells. Similarly, in artificial neural networks, associative memory is simulated by creating connections between artificial neurons. In machine learning, models based on associative memory are capable of assigning weights to individual data elements and creating associations between them. This allows artificial systems to efficiently reconstruct from a portion of the available data.

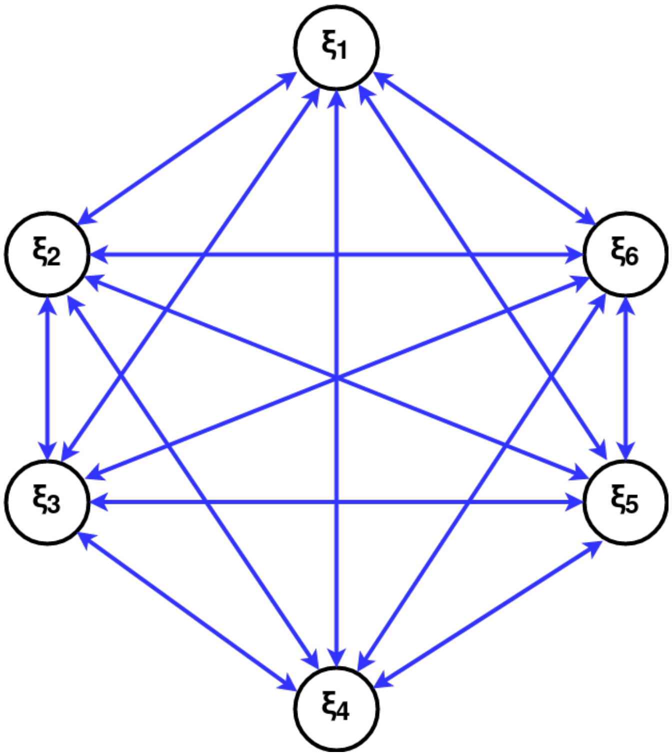
The thesis consists of 6 main chapters, of which the first three are theoretical and the others are research. The first one addresses the architecture and properties of the classical Hopfield network model. This part is a comprehensive mathematical description of all the processes considered and their connections to statistical physics. The second and third chapters are a theoretical study of the other two models - dense associative memory and continuous Hopfield network. The fifth chapter is followed by a direct practical comparison of the three considered models in the reproduction of image files, in addition to a description of the programmatic implementation. The sixth and seventh chapters are an approach to extending the use of the continuous Hopfield network beyond the issue of image file restoration. The author's method of dividing image files into classes is presented, as well as a technique for processing experimental data.

The theoretical part of the work is based on the literature in machine learning, mathematics and statistical physics. A particularly relevant source is the paper *Hopfield networks is all you need* by Hubert Ramsauer, Bernard Schafel and 14 other authors.

Chapter 2

Classical Hopfield network

The classical Hopfield network is a recurrent neural network model popularized in 1982 by John Hopfield. [3] It was first described by Little and Sawa in 1974 as a modification of the Ising model. The basic model consists of bilaterally connected perceptrons.



Each of the xi perceptrons is bipolar, that is, it accepts a value from the set -1, 1. Each xi receives a signal to all other neurons and can send a signal to all but itself. According to the perceptron model [2], a neuron is active (takes the value 1) or is inactive (takes the value -1) based on the equation

$$\xi_i = \Theta[\sum_j (W_{ij}\xi_j + b_i)] \tag{2.1}$$

where

$$\Theta(a) = \begin{cases} 1 & a >= 0 \\ -1 & a < 0 \end{cases} \tag{2.2}$$

In the equation 2.1 we consider the sum of signals arriving from all neurons multiplied by the corresponding weight W_{ij}. The weight determines how significant a particular incoming signal is; all weights are contained in the W matrix. The connections are symmetric [3], so

$$W_{ij} = W_{ji} \tag{2.3}$$

and, in addition, to handle the condition that the neuron is not connected to itself, it is introduced

$$W_{ii} = 0 \tag{2.4}$$

The additional word b_i is the threshold of the i-th neuron, the so-called bias. While the perceptron model was simple and abstract, Hopfield tried to create a model closer to the biology of the human brain by connecting perceptrons.

The classical Hopfield network with many neurons changes very dynamically due to the number of connections that modify the state of the network. We consider two ways of ordering state changes:

- Synchronous modification - all neurons calculate their activation (the argument of the $Theta$ function), and then simultaneously modify their state.
- Asynchronous modification - at any given time, a single neuron calculates its activation and modifies its state; subsequent neurons to perform this operation are either selected randomly or in a predetermined order.

In the context of associative memory, we want the network to be able to store certain data (patterns) and then recognize and reconstruct them from corrupted or incomplete data, just as the human brain does. A pattern is basically an N-element vector x with bipolar values {-1,1}, where N is the number of neurons xi in the network. This vector represents a particular state of the network, so the number of its elements must be equal to the number of neurons.

The basic method of learning a Hopfield network a given pattern is called Hebb's rule. According to it, the connection between two neurons strengthens if they activate simultaneously and weakens if they activate separately [5], this can be defined by the proportionality $W_{ij}\$W_{i_j}$. The generalized rule for multiple patterns can be represented as [2]

$$W_{ij} = \frac{1}{S}\sum_{\mu=1}^{S}\sum_{i,j}^{N} x_i^\mu x_j^\mu = \frac{1}{S}\sum_{\mu=1}^{S} x^\mu (x^\mu)^T \tag{2.5}$$

where S is number of patterns and N is number of neurons.

2.1 Energy function

Consider the product of the state of a neuron and the signal reaching it, xi_i is the current state, and xi_i^{new} is the state after the update. If the sign of both values is the same then the difference of the products is equal to 0.

$$\xi_i^{new} * \sum_j (W_{ij}\xi_j + b_i) - \xi_i * \sum_j (W_{ij}\xi_j + b_i) = 0 \tag{2.6}$$

When sign is different:

$$\xi_i^{new} * \sum_j (W_{ij}\xi_j + b_i) - \xi_i * \sum_j (W_{ij}\xi_j + b_i) = 2\xi_i^{new} * \sum_j (W_{ij}\xi_j + b_i) \tag{2.7}$$

In this case, the difference is positive, given both equations the conclusion is that the value of the product never decreases. Let the function D describe these products globally:

$$D(\xi_1, \xi_2, ..., \xi_N) = \sum_i \xi_i * \sum_j (W_{ij}\xi_j + b_i) = \sum_{i,j} W_{ij}\xi_i\xi_j + \sum_i b_i\xi_i \tag{2.8}$$

If a neuron changes its state, it will cause the value of D to increase. The function D has an upper limit:

$$D^{max} = \sum_{i,j} |w_{ij}| + \sum_i |b_i| \tag{2.9}$$

Each change in the state of the neuron leads to an increase in the value of D, so in a finite number of steps the function converges. After multiplying by -1 and normalizing the first component (due to the symmetry of the weights), we obtain:

$$E = -\frac{1}{2}\sum_{i,j}^{N} W_{ij}\xi_i\xi_j - \sum_i b_i\xi_i = -\frac{1}{2}\xi^T W \xi - \xi^T b \tag{2.10}$$

The above function is an energy function proposed by Hopfield to describe a neural network. It depends on the state of all neurons and the weights of the connections between them. The description is inspired by the Hamiltonian of the Ising model known from statistical physics.

$$E_{Ising} = -\frac{1}{2}\sum_{<i,j>} J_{ij}S_iS_j - \sum_i h_iS_i \tag{2.11}$$

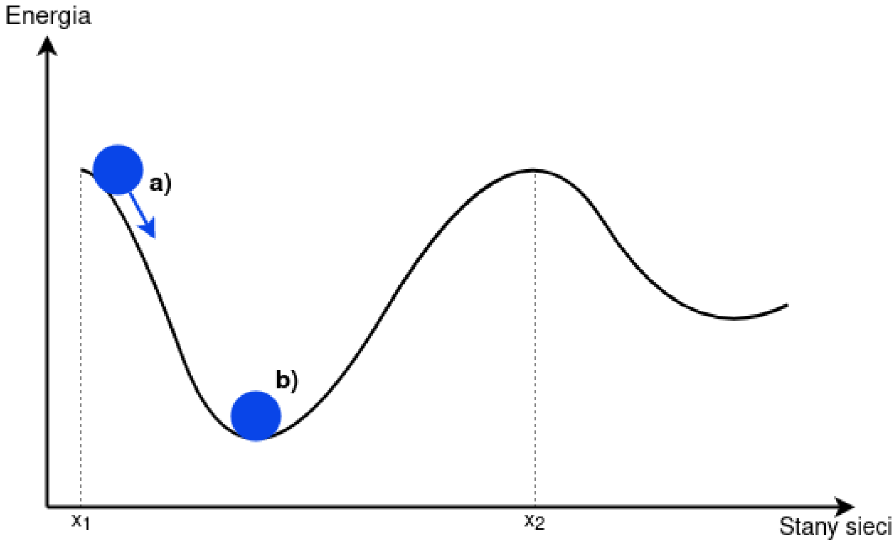


Figure 2.1: Schematic diagram of the Hopfield network energy. The horizontal axis represents specific network state scenarios (the state of all neurons). The points x_1 and x_2 represent the range of the attractor pool, that is, the range in which the network converges to a given local minimum of energy (attractor). The network starts in state (a) and, after successive updates, reaches state (b) corresponding to the local energy minimum.

Originally, the model describes the behavior of a ferromagnet through an arrangement of discrete spins arranged into network nodes. The equivalent S_i spins are xi_i neurons, while the role of the exchange integral J_{ij} is played by W_{ij} weights. In addition, the Ising model assumes interaction with an external magnetic field, at which point the Hopfield network introduces bias b_i.

The Hopfield network decreases the value of the energy function during successive updates until it finally reaches a local minimum and stops changing its state. The local minima of this function are called attractors, and the areas in which the function converges to a given minimum are the attractor pools [2], as in figure 2.1.

In practice, perfect learning of the network is difficult and the patterns we care about are not the only attractors that exist. When teaching the network a given pattern, it actually "remembers", in addition to the correct pattern, its opposite version with inverse values. Figuratively, if the network has an attractor corresponding to the vector (1, 1, 1, -1) it means that it also has an attractor corresponding to the vector (-1, -1, -1, 1). This is a direct result of the symmetry of the weights in the connections between neurons. In addition, as a result of learning errors, attractors may be formed in states that do not correspond to any of the selected patterns.

2.1.1 Crosstalk

Let's consider the stability of a particular pattern x^ν, under stability we expect that:

$$\Theta(h_i) = x_i^\nu \quad \forall i \tag{2.12}$$

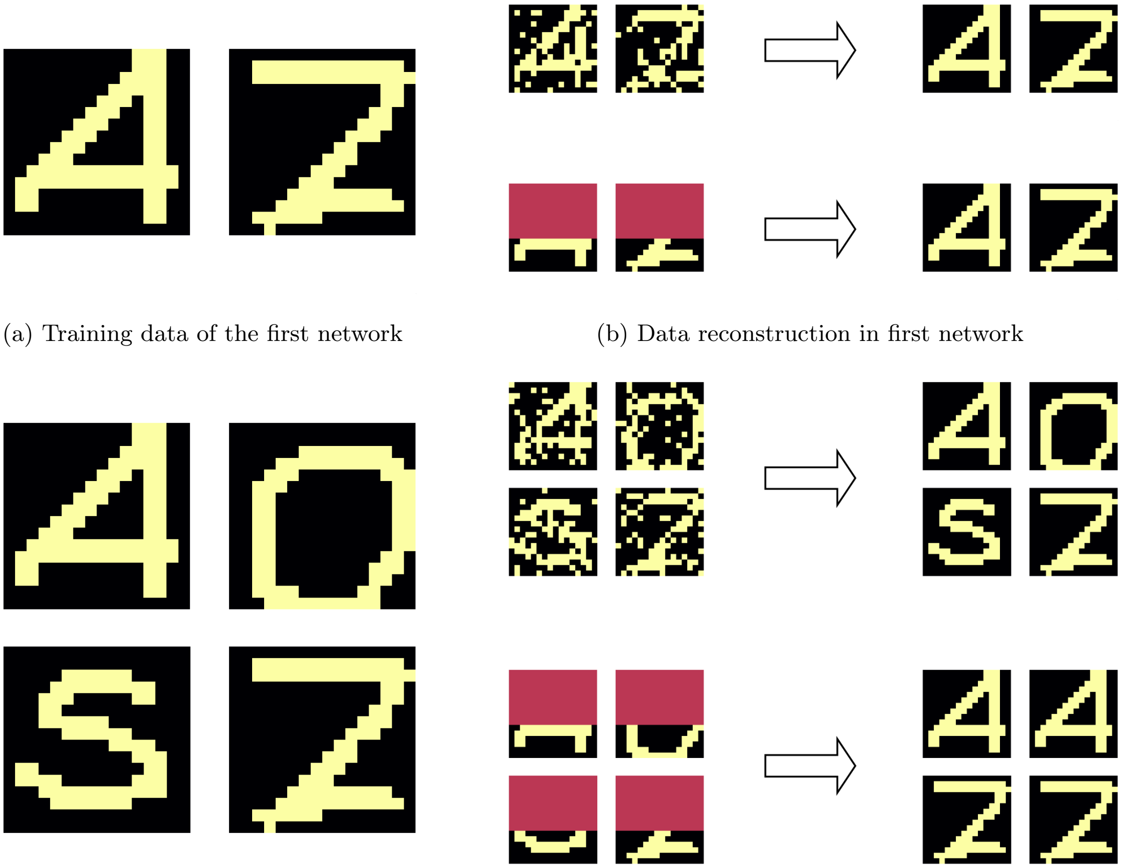
Where h_i is the network contribution to neuron i. Based on the 2.1 and 2.5 can be written:

$$h_i = \sum_j^N W_{ij} x_j^\mu = \frac{1}{N} \sum_j^N \sum_m^S x_i^m x_j^m x_j^\mu \tag{2.13}$$

The above form is an approach from another side to calculate the signal reaching neuron i. We consider the effect of all patterns m on the reproduction of a particular pattern μ. 2.13 can be split into two expressions, the part coming from the pattern sought and all others:

$$h_i = x_i^\mu + \frac{1}{N} \sum_j^N \sum_{m \neq \mu}^S x_i^m x_j^m x_j^\mu \tag{2.14}$$

The second component of the above sum is the so-called crosstalk, i.e. the component coming from other patterns, which makes it difficult to reproduce the x^μ pattern. If the crosstalk is negligibly small compared to the first component then it does not change the sign of h_i. The value at which crosstalk reverses about 1 percent



(a) Training data of the first network

(b) Data reconstruction in first network

(c) Training data of the second network

(d) Data reconstruction in second network

Figure 2.2: An example of how a classical Hopfield network works. 16x16 pixel images representing letters were used as training data, or patterns. Each pixel corresponds to one neuron of the network, so the network has 256 neurons. When the task is to reproduce a small number of patterns (here 2), the first network handles both noisy and partial data without problems. When increasing the number of patterns presented the second network failed to reconstruct the data, twice the network stopped at the wrong energy minimum.

of activation is the number of patterns $S_{max} = 0.138N$, above which there can be an exponential increase in the number of errors and permanent instability of the network [2]. Obtaining this result requires a detailed analysis of the relationship between the classical Hopfield network and the Ising model.

2.2 Analogy to Ising's model

Consider a model of a magnetized material with Hopfield lattice markings, each spin is affected by an external magnetic field.

$$b_i = \sum_j W_{ij}\xi_j + b^{ext} \tag{2.15}$$

The coefficient W_{ij} determines the influence of spin xi_j in the magnetic field on spin xi_i, these influences are symmetrical. At low temperature, the spins align according to the equation $\xi_i = \Theta(h_i)$, these changes occur asynchronously in a random order. The Hamiltonian defining the above assumptions:

$$H = -\frac{1}{2}\sum_{ij} W_{ij}\xi_i\xi_j - b^{ext}\sum_i \xi_i \tag{2.16}$$

When the temperature is not low, fluctuations occur that swap the values of the spins, which disturbs their alignment with the magnetic field. The higher the temperature, the greater the effect of the fluctuations on the behavior of the spins. [2]

Glauber dynamics mathematically describes the effect of fluctuations on the Ising model, the probability of setting a spin in a particular state is:

$$P(\xi_i := 1) = g(b_i) \tag{2.17}$$

$$P(\xi_i := -1) = 1 - g(b_i) \tag{2.18}$$

$$g(h) = \frac{1}{1 + exp(-2\beta b)} \tag{2.19}$$

where

$$\beta = \frac{1}{k_B T} \tag{2.20}$$

The function $g(b)$ (in this case sigmoid) depends on the temperature T prevailing in the system. As $1 - g(h) = g(-h)$ the dynamics of the spin change can be described as:

$$P(\xi_i := \pm 1) = g(\pm b_i) = \frac{1}{1 + exp(\mp 2\beta b_i)} \tag{2.21}$$

Temperature directly affects the shape of the sigmoid $g(b)$, at temperatures close to zero the sigmoid approaches the Θ function used in the classical Hopfield network.

For demonstration purposes, the above dynamics can be applied to describe a single spin, in which case we will obtain the average magnetization (average of one spin equivalent to one spin):

$$\xi_i =< \xi_i >= P(\xi_i = 1) * 1 + P(\xi_i = -1) * (-1) = \tag{2.22}$$

$$= \frac{1}{1 + exp(-2\beta b_i)} - \frac{1}{1 + exp(2\beta b_i)} = \frac{exp(\beta b_i) - exp(-\beta b_i)}{exp(\beta b_i) - exp(-\beta b_i)} = tanh(\beta b_i) \tag{2.23}$$

This result can also be applied to a system of N spins if they are affected by the same external magnetic field and the spins do not affect each other, such a system is a paramagnetic with $M = N < S >$ magnetization.

In the case where each spin is the source of a local magnetic field, an approximation should be used that proves useful in analyzing the Hopfield model. The mean-field theory involves replacing the "fluctuating" b_i by: [2]

$$< b_i >= \sum_j W_{ij} < \xi_j > + b^{ext} \tag{2.24}$$

then the equation of average magnetization 2.23 takes the form:

$$< \xi_i >= tanh(\beta < b_i >) = tanh(\beta \sum_j W_{ij} < \xi_j > + \beta b^{ext}) \tag{2.25}$$

The idea behind the theory is to select a single spin and replace all others by an average external magnetic field. Fluctuations of the other spins are not taken into account, even after changing the selected spin.

The actual temperature, of course, does not apply to artificial neural networks; in their case, one can define a pseudo-temperature expressed as $\beta = \frac{1}{T}$, so that one can still describe the change in shape of the sigmoid and hyperbolic tangent. Assuming a small number of memories in the network and combining the mean-field theory with the Hebb learning equation 2.5, a stochastic neural network can be described:

$$< \xi_i >= tanh(\frac{\beta}{N}\sum_{j,\mu} x_i^\mu x_j^\mu < \xi_j >) \tag{2.26}$$

The above equation does not have a simple solution, for simplicity we can assume that $< xi_i >$ is proportional to one of the memories in the network:

$$< \xi_i >= m x_i^\nu \tag{2.27}$$

$$m x_i^\nu = tanh(\frac{\beta}{N}\sum_{j,\mu} x_i^\mu x_j^\mu m x_j^\nu) \tag{2.28}$$

As in the classical Hopfield network, the hyperbolic tangent argument can be broken down into an expression proportional to the recollection and crosstalk. Under the current assumption of a small number of memories, crosstalk is negligible, therefore:

$$m x_i^\nu = tanh(\beta m x_j^\nu) \tag{2.29}$$

and as $tanh(-a) = -tanh(a)$:

$$m = tanh(\beta m) \tag{2.30}$$

For this form of the equation, the states of the lattice corresponding to memories are stable for a temperature less than or equal to 1. Adopting the nomenclature from the description of ferromagnets, the critical temperature for a stochastic lattice with a small number of memories is 1. Taking the equation of average magnetization 2.27, it can be written:

$$m = \frac{< \xi_i >}{x_i^\nu} = P(bit\ i\ is\ correct) - P(bit\ i\ is\ incorrect) \tag{2.31}$$

In addition, the average number of correct spins in the network response received is [2]

$$< N_{correct} >= \frac{1}{2}N(1+m) \tag{2.32}$$

Above the critical temperature $< N_{correct} >$ is $< N_{correct} >$ which is simply the number of correct units that can be expected with a random answer, $< N_{correct} >$ tends to N (all correct) for low temperatures.

The above analysis is valid for a small number of memories ($S \ll N$), in order to analyze the actual capacity of the classical Hopfield network I introduce a load parameter:

$$\gamma = \frac{S}{N} \tag{2.33}$$

The starting point is the equation combining the mean field theory with Hebb's theory 2.26, except that this time the crosstalk can't be ignored, we now focus on all memories, including mixed ones i.e. overlapping ones.

$$m_\nu = \frac{1}{N}\sum_i x_i^\nu < \xi_i > \tag{2.34}$$

We assume that we are investigating the recovery of pattern number 1. Then m_1 has the order of unity, while each of m_ν for $\nu \neq 1$ is small, of the order of $\frac{1}{\sqrt{N}}$ for random memories. The size of r, which is the mean square of the overlap of the system configuration with unrecovered memories, is of the order of unity.

$$r = \frac{1}{\gamma}\sum_{\nu\neq 1} m_\nu^2 \tag{2.35}$$

The equation $\frac{1}{gamma} = \frac{N}{S}$ makes r the average of the S (or $S-1$) squares of the overlap and cancels the expected m_ν relationship of $\frac{1}{\sqrt{N}}$. The equation of the mean field of 2.28 with a larger number of memories takes the form of

$$m_\nu = \frac{1}{N}\sum_i x_i^\nu tanh(\beta \sum_\mu x_i^\mu m_\mu) \tag{2.36}$$

and after dividing into factors with $\mu = 1$ and $\mu = \nu$

$$m_\nu = \frac{1}{N}\sum_i x_i^\nu x_i^1 tanh(\beta(m_1 + x_i^\nu x_i^1 m_\nu + \sum_{\mu\neq 1,\nu} x_i^\mu x_i^1 m_\mu)) \tag{2.37}$$

The first argument of the hyperbolic tangent has order 1, the third argument is also significant due to the greater number of S memories, but the second argument is negligible (order $\frac{1}{\sqrt{N}}$) therefore using the derivative of the hyperbolic tangent one can write [2]

$$m_\nu = \frac{1}{N}\sum_i x_i^\nu x_i^1 tanh(\beta(m_1 + \sum_{\mu\neq 1,\nu} x_i^\mu x_i^1 m_\mu)) + \frac{\beta}{N}\sum_i (1 - tanh^2(\beta(m_1 + \sum_{\mu\neq 1,\nu} x_i^\mu x_i^1 m_\mu)))m_\nu \tag{2.38}$$

A small overlap of m_μ, $\mu \neq 1$ can take on small negative or small positive values, hence they can be approximated as random variables with mean equal to zero and variance $\frac{\gamma r}{S}$. The component of $x_i^\mu x_i^1$ is random and independent of m_μ so according to the central limit theorem the whole sum is the average of Gaussian noise, this allows one to reduce the second half of the equation 2.38 and the whole above equation to the form:

$$m_\nu = \frac{1}{N}\sum_i x_i^\nu x_i^1 tanh(\beta(m_1 + \sum_{\mu\neq 1,\nu} x_i^\mu x_i^1 m_\mu)) + \beta m_\nu - \beta q m_\nu \tag{2.39}$$

or

$$m_\nu = \frac{N^{-1}\sum_i x_i^\nu x_i^1 tanh[\beta(m_1 + \sum_{\mu\neq 1,\nu} x_i^\mu x_i^1 m_\mu)]}{1 - \beta(1-q)} \tag{2.40}$$

where

$$q = \int \frac{dz}{\sqrt{2\pi}} exp(-\frac{z^2}{2}) tanh^2[\beta(m_1 + \sqrt{\gamma r} z)] \tag{2.41}$$

To now count r, we take the square of the equation 2.40:

$$m_\nu^2 = [\frac{1}{1-\beta(1-q)}]^2 \frac{1}{N^2}\sum_{ij} x_i^\nu x_i^1 x_j^\nu x_j^1 * tanh[\beta(m_1 + \sum_{\mu\neq 1,\nu} x_i^\mu x_i^1 m_\mu)] * tanh[\beta(m_1 + \sum_{\mu\neq 1,\nu} x_j^\mu x_j^1 m_\mu)] \tag{2.42}$$

and we calculate the average after all the patterns as in 2.35. Since the pattern ν does not appear in the arguments of the hyperbolic tangent, the averages of the expressions $x_i^\nu x_i^1 x_j^\nu x_j^1$ can be counted independently and only the factor $i = j$ survives. The remaining average of the hyperbolic tangents is independent of ν so:

$$r = \frac{q}{[1-\beta(1-q)]^2} \tag{2.43}$$

In the same way we can obtain an equation for m_1:

$$m_1 = \int \frac{dz}{\sqrt{2\pi}} exp(-\frac{z^2}{2}) tanh[\beta(m_1 + \sqrt{\gamma r} z)] \tag{2.44}$$

Now all three equations for q, m_1 and r can be calculated numerically, the best way is to analyze the case of temperature going to 0. In this limit, q goes to 1, but the equation $\beta(1-q)$ remains finite. This is where approximations are useful: [2]

$$\int \frac{dz}{\sqrt{2\pi}} exp(-\frac{z^2}{2})[1 - tanh^2[\beta(az+b)]] \approx \frac{1}{\sqrt{2\pi}} exp(-\frac{z^2}{2})|_{tanh^2\beta[az+b]=0} * \int dz(1 - tanh^2\beta[az+b]) \tag{2.45}$$

$$= \frac{1}{\sqrt{2\pi}} exp(-\frac{b^2}{2a^2}) \frac{1}{a\beta} \int dz \frac{\partial}{\partial z} tanh[\beta(az+b)] = \sqrt{\frac{2}{\pi}} \frac{1}{a\beta} exp(-\frac{b^2}{2a^2}) \tag{2.46}$$

oraz

$$\int \frac{dz}{\sqrt{2\pi}} exp(-\frac{z^2}{2}) tanh[\beta(az+b)] \overset{T\to 0}{\to} \int \frac{dz}{\sqrt{2\pi}} exp(-\frac{z^2}{2}) sgn[\beta(az+b)] \tag{2.47}$$

$$= 2\int_{-b/a}^{\infty} \frac{dz}{\sqrt{2\pi}} exp(-\frac{z^2}{2}) - 1 = erf(\frac{b}{\sqrt{2}a}) \tag{2.48}$$

Where erf is the error function. After using such approximations, three equations can be written:

$$C \equiv \beta(1-q) = \sqrt{\frac{2}{\pi\gamma r}} exp(-\frac{m^2}{2\gamma r}) \tag{2.49}$$

$$r = \frac{1}{(1-C)^2} \tag{2.50}$$

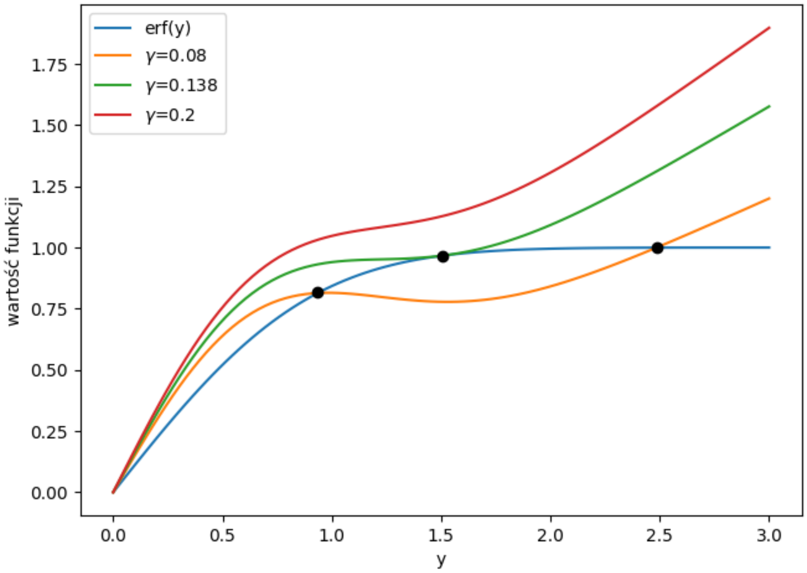


Figure 2.3: Graphical solution of the equation 2.52 for three selected values of γ. Non-trivial solutions with $m > 0$ are given by intersections (except for the one at $y = 0$). There is a critical value of γ above which non-trivial solutions $(m = 0)$ vanish.

$$m = erf(\frac{m}{\sqrt{2\gamma r}}) \tag{2.51}$$

By defining $y = \frac{m}{\sqrt{2\gamma r}}$, we obtain:

$$y(\sqrt{2\gamma} + \frac{2}{\sqrt{\pi}} exp(-y^2)) = erf(y) \tag{2.52}$$

By solving the equation 2.52, one can find the critical value of γ, at which non-trivial solutions $(m = 0)$ vanish. The numerical solution gives $\gamma \approx 0.138$ [2], this means the theoretical limit of the number of patterns $S_{max} \approx 0.138N$.

Chapter 3

Dense associative memory

As the numerical solution shows, the classical Hopfield model is effective when the number of stored patterns is much smaller than the number of neurons in the network. With too many patterns, some will begin to overlap, that is, generate contributions to the network of the same order. A solution that increases the capacity of the network is to modify the standard energy function to the form:

$$E = -\sum_{\mu=1}^{S} F[\sum_{i} x_i^{\mu} \xi_i] \tag{3.1}$$

The F-function (hereafter referred to as the interaction function) takes one of three forms

$$F(\alpha) = \alpha^n \tag{3.2}$$

$$F(\alpha) = \begin{cases} \alpha^n & \text{for } \alpha \geq 0 \\ 0 & \text{for } \alpha < 0 \end{cases} \tag{3.3}$$

or

$$F(\alpha) = exp(\alpha) \tag{3.4}$$

where n is postive integer [4] [1]. In the case of $n = 2$, the network reduces to a classic Hopfield model. With $n > 2$, more patterns can be placed in the same configuration space before they begin to overlap [4]. The idea behind the new update function is to find the energy difference between the state of the network with ith neuron in the active state and ith neuron in the inactive state.

$$\xi_i = \Theta[E(\xi_i = 1) - E(\xi_i = -1)] \tag{3.5}$$

When the energy equation is introduced directly, the update function takes the form

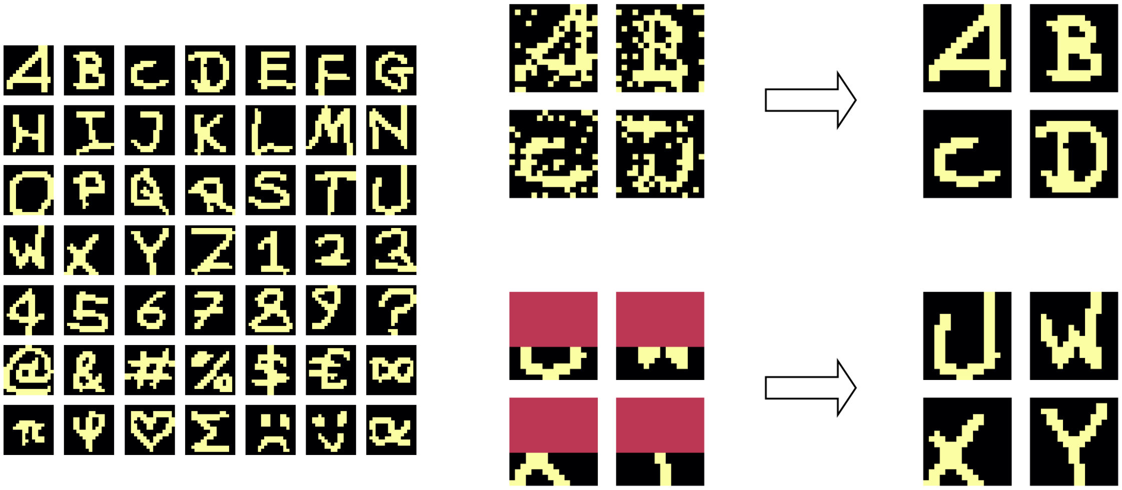
$$\xi_i = \Theta[\sum_{\mu=1}^{S} (F(1 * x_i^{\mu} + \sum_{j \neq i} x_j^{\mu} \xi_j) - F(-1 * x_i^{\mu} + \sum_{j \neq i} x_j^{\mu} \xi_j))] \tag{3.6}$$

$$\xi_i = \Theta[\sum_{\mu=1}^{S} (F(x_i^{\mu} + \sum_{j \neq i} x_j^{\mu} \xi_j) - F(-x_i^{\mu} + \sum_{j \neq i} x_j^{\mu} \xi_j))] \tag{3.7}$$

In other words, the value of a neuron is fixed at 1 or -1, depending on what is more favorable when reducing the energy of the whole system. As can be seen in this case, the change in the state of a neuron depends directly on the energy function, different from the basic model where energy was only a concept used to abstractly describe the algorithm. The new model described above is called Dense Associative Memory.

How does the above change affect the capacity of the network? Let's assume a set of patterns with random values from the set {-1,1} and a network initialized in the state $x = \xi^{\mu}$, that is, in the state corresponding to one of the patterns. The energy difference between the initial state and the state in which the i-th neuron is inverted at the activation function 3.2 can be written as:

$$\Delta E = \sum_{\nu=1}^{S} (x_i^{\nu} x_i^{\mu} + \sum_{j \neq i} x_j^{\nu} x^{\mu})^n - \sum_{\nu=1}^{S} (-x_i^{\nu} x_i^{\mu} + \sum_{j \neq i} x_j^{\nu} x^{\mu})^n \tag{3.8}$$



(a) Training data

(b) Data reconstruction

Figure 3.1: An example of how the dense associative memory model works. 49 16x16 pixel images representing various symbols were used as training data. The new model can handle more data without much trouble, reproducing both distorted and incomplete data. The increased capacity now makes it possible to better distinguish correlated patterns which is impossible with the traditional Hopfield network. The interaction function used is $F(\alpha) = \alpha^8$. With noisy data, full restoration always occurred after 1 update, for data with 176 pixels clipped, full restoration occurred between 1 and 3 updates.

Mean of this equation is: [4]:

$$< \Delta E >= N^n - (N-2)^2 \approx 2nN^{n-1} \tag{3.9}$$

This follows from the member $\nu = \mu$. The variance in the limit of large N is [4]:

$$\sigma^2 = 4n^2(2n-3)!!(S-1)N^{n-1} \tag{3.10}$$

If the magnitude of the fluctuation exceeds the energy gap ΔE and its sign is opposite to the sign of the energy gap then the i-th unit becomes unstable. Thus, with N and K large enough for the noise to follow a Gaussian distribution, the probability that the state of a single neuron is unstable is equal to [4]

$$P_{error} = \int_{<\Delta E>}^{\infty} \frac{1}{\sqrt{2\pi Var}} exp(-\frac{y^2}{2\sigma^2})\, dy \approx \sqrt{\frac{(2n-3)!!S}{2\pi N^{n-1}}} exp(-\frac{N^{n-1}}{2S(2n-3)!!}) \tag{3.11}$$

In order to achieve a probability less than a specific preset value, it is necessary to adopt an upper limit on the number of patterns that the network is able to store

$$S_{max} = \omega_n N^{n-1} \tag{3.12}$$

Where ω is a numerical constant, depending on the assumed probability threshold. The case $n = 2$ corresponds to the previously determined value of $S_{max} \approx 0.138N$, for ideal memory reconstruction (i.e., $P_{error} < 1/N$) the result is [4]

$$S_{max}^{noerrors} \approx \frac{N^{n-1}}{2(2n-3)!!ln(N)} \tag{3.13}$$

When increasing the power of n, the capacity increases in a non-linear manner, allowing the network to efficiently store and reproduce many more patterns than the number of neurons. At small n, many expressions overlap evenly and make it difficult to reproduce μ in the equation 3.7. In the limit of $n \to \infty$, the dominant contribution comes from the single pattern that is most similar to the initial state. It is computationally optimal to take $2 < n \ll \infty$.

3.1 Exclusive or - $S > N$

Using the analysis of a simple example of a Exclusive or (XOR), it can be demonstrated that as n increases, the computational capabilities of the network increase. In addition, the number of patterns will exceed the number of neurons, with a classical Hopfield network even approaching such a result would not be possible. We consider the case where the presentation of input data x and y returns z according to:

$$z = \begin{cases} -1 & \text{for } x = -1 \text{ i } y = -1 \\ 1 & \text{for } x = -1 \text{ i } y = 1 \\ 1 & \text{for } x = 1 \text{ i } y = -1 \\ -1 & \text{for } x = 1 \text{ i } y = 1 \end{cases} \tag{3.14}$$

The role of patterns will be taken by $[x, y, z]$ sets according to the above rules, we will thus obtain $S = 4$ patterns with $N = 3$ neurons. The energy equation 3.1 takes the form:

$$E_n(x, y, z) = -(-x - y - z)^n - (-x + y + z)^n - (x - y + z)^n - (x + y - z)^n \tag{3.15}$$

Taking first few integer n energy is:

$$E_1(x, y, z) = 0 \tag{3.16}$$

$$E_2(x, y, z) = -4(x^2 + y^2 + z^2) \tag{3.17}$$

$$E_3(x, y, z) = 24xyz \tag{3.18}$$

$$E_4(x, y, z) = -4(x^4 + y^4 + 6y^2z^2 + z^4 + 6x^2 + z^2) \tag{3.19}$$

$$E_5(x, y, z) = 80xyz(x^2 + y^2 + z^2) \tag{3.20}$$

In case of $n = 1$ function equals 0. For even n energy is even function, changing the value of any of the arguments to the opposite does not change the energy value (by en even power), so the desired energy change cannot be read. If n is odd and greater than 1, we are dealing with an odd function $E_n(x, y, -z) = E_n(x, y, z)$, generalizing can be written:

$$E_n(x, y, z) = \begin{cases} 0 & \text{for } n = 1 \\ A_n & \text{for } n = 2, 4, ... \\ A_n xyz & \text{for } n = 3, 5, ... \end{cases} \tag{3.21}$$

Where A_n is a numerical constant depending on n. Given an appropriate n (3, 5, ...), dense associative memory is able to take x and y as input and then reproduce z by minimizing the energy value. For example, for $n = 3$ the update rule 3.5 takes the form:

$$z = \Theta[E_3(x, y, -1) - E_3(x, y, +1)] \tag{3.22}$$

$$= \Theta[(-x-y-1)^3-(-x-y+1)^3-(x-y-1)^3+(x-y+1)^3-(-x+y-1)^3+(-x+y+1)^3+(x+y-1)^3-(x+y+1)^3] \tag{3.23}$$

$$= \Theta[-48xy] = \Theta[A_3xy] \tag{3.24}$$

With such an updated rule, the network is able to recreate the z element even though the number of patterns exceeds the number of neurons. For rectified polynomials (interaction function 3.3), a similar problem is solvable for any total $n > 2$.

Chapter 4

Continous Hopfield network

aaaa

Bibliography

[1] Mete Demircigil, Judith Heusel, Matthias Löwe, Sven Upgang, and Franck Vermet. On a model of associative memory with huge storage capacity. *Journal of Statistical Physics*, 168(2):288–299, May 2017.

[2] John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction To The Theory Of Neural Computation.* 1991.

[3] John Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79:2554–8, 05 1982.

[4] Dmitry Krotov and John J Hopfield. Dense associative memory for pattern recognition. 2016.

[5] David J. C. MacKay. *Information Theory, Inference and Learning Algorithms.* nvm, 2003.