university of groningen

Learning of single-layer neural networks: ReLU vs. sigmoidal activation

Author
Elisa C. OOSTWAL

Supervisors
Prof. Dr. Michael BIEHL
Michiel STRAAT, MSc

November, 2019

Contents

1 Introduction **2**

2 Statistical physics **2**

3 Training a network **3**
3.1 Description of the network . . . 3
3.2 Measure of performance: generalization error . . . 4
3.2.1 Sigmoidal activation . . . 5
3.2.2 ReLU activation . . . 5
3.3 Free energy . . . 5
3.4 Phase transitions . . . 6

4 Analysis **7**
4.1 Minimization of the free energy . . . 7
4.2 Validity of the solution . . . 7
4.3 Finding the phase transitions . . . 8

5 Results **8**
5.1 K = 1 . . . 8
5.2 K = 2 . . . 9
5.3 K = 3 . . . 10

6 Discussion **11**

7 Conclusion **11**

1 Introduction

In the late 1950's *neural networks* emerged as a new field of study within the Computing Science community. The earliest example of this is the *perceptron*, devised by Rosenblatt. This machine was able to recognize images, something which had not been accomplished before. Expectations were raised and its possibilities drew great attention, but the development of the study soon came to a halt, as the physical boundaries of computers did not allow for the complex computations involved. However, due to great advancements in hardware and the availability of large amounts of data, the field has regained interest from scientists [1–5]. This increase in popularity applies in particular to deep learning, in which neural networks with many hidden layers (*deep neural networks*) are trained [6].

Conventionally, sigmoidal activation functions have been used in the hidden units. Rectified Linear Unit (ReLU) activation has been proposed as a better alternative, showing an improved performance compared to sigmoidal activation [7–9]. These claims are however mainly based on empirical data. A theoretical approach is needed to understand the fundamental differences between sigmoidal activation and ReLU activation, if there are any at all.

Statistical physics has proved itself to be a fruitful approach in studying neural networks and providing a theoretical understanding of their underlying mechanisms [1] [10]. Statistical physics methods facilitate, for instance, the study of dynamical and equilibrium properties of randomized training processes in model situations [3]. At the same time, the approach inspires novel and efficient algorithms and facilitates interdisciplinary applications in a variety of scientific and technical disciplines. This approach has already been used in the analysis of sigmoidal activation [11]. More recently, the same approach was applied in a study which examines the differences between sigmoidal and ReLU activation, in the case of on-line learning [12].

In this study we will investigate why ReLU activation might perform better than sigmoidal units by researching their fundamental differences in the context of off-line learning. We will do this using a statistical physics approach. We will restrict ourselves to shallow networks with a single hidden layer as a first model system.

2 Statistical physics

In the field of physics, one may find themselves studying a system comprised of a very large number of particles (*macroscopic system*), for example in the case of a gas contained within a cylinder. The properties of the gas (*macroscopic properties*), such as its pressure, originate from the interactions between the individual molecules, as well as the interactions between the molecules and the walls of the cylinder. In principle, the laws of interaction between the particles are known, and hence one may try to derive the pressure from these laws. However, due to the colossal number of particles involved, it is impossible to solve the equations of motion [13]. Instead, a different framework is needed which allows for derivation of properties of such a macroscopic system.

Statistical physics has been developed for exactly this purpose. It knows two approaches to solving such problems. The first approach is the one of classical thermodynamics, which we will not elaborate on. The second approach is that of statistical mechanics. The objective here is to derive the properties of a macroscopic system from its microscopic properties, which is essentially done by averaging over the microscopic properties [13].

3 Training a network

We simulate the training of a neural network, which we will refer to as the student network. The student will be trained using off-line learning. We assume a teacher network exists: a network which, upon presentation of input data, can determine the outcome perfectly. The goal is to train the student in such a way that its performance becomes identical to the teacher. We will not perform the actual training of the student network, however, but instead wish to give a theoretical description of the training process.

In general, a neural network involves many parts: it includes a number of hidden layers, each consisting of several units. In turn, every unit has a weight vector (explained in section 3.1) that has the same number of elements as the input data. Collectively this amounts to a very large number of components that play a role in the process of learning. The discipline of statistical physics is specialized in dealing with such macroscopic systems, which makes it cut out for studying the underlying mechanics of a neural network. Moreover, we are not interested in the individual units (microscopic properties), but are concerned with the overall performance of the system (macroscopic properties): how accurate is the outcome of the system given a set of input data, how long does it take the system to learn the rule? This too makes statistical physics a suitable approach for answering these questions.

3.1 Description of the network

The student network has a single hidden layer comprised of K units. When the network is presented an N-dimensional input vector $\boldsymbol{\xi}$, each unit determines the inner field x_k to the input vector. To this end, each unit has a weight vector $\boldsymbol{w_k}$, which has the same length as the input vector:

$$x_k = \frac{1}{\sqrt{N}} \boldsymbol{w_k} \cdot \boldsymbol{\xi} \tag{1}$$

The total output of the student network for the given input $S(\boldsymbol{\xi})$ is then computed as:

$$S(\boldsymbol{\xi}) = \frac{1}{\sqrt{K}} \sum_{k=1}^{K} g(x_k), \tag{2}$$

where $g(x)$ is the activation function.

We assume the teacher network has the same architecture as the student network: a single hidden layer with K hidden units, each having a weight vector $\boldsymbol{w_m^*}$. The total output of the teacher network $\sigma(\boldsymbol{\xi})$ is determined in the same way as for the student network:

$$\sigma(\boldsymbol{\xi}) = \frac{1}{\sqrt{K}} \sum_{m=1}^{K} g(y_m), \text{ where } y_m = \frac{1}{\sqrt{N}} \boldsymbol{w_m^*} \cdot \boldsymbol{\xi} \tag{3}$$

Additionally, we assume that the components of the input vectors $\boldsymbol{\xi}$ are independent, identically distributed (i.i.d). This makes the central limit theorem apply to the inner fields x_i and y_j. Moreover, we assume that the weight vectors of the teacher $\boldsymbol{w_m^*}$ are normalized vectors of length $\sqrt{N}$ with i.i.d. random components. As a result, the weight vectors of the student $\boldsymbol{w_k}$ should be normalized vectors of length $\sqrt{N}$ as well. Finally, we assume that the teacher weight vectors are orthogonal with respect to one another.

The student networks "learns" the rule from the input data by verifying whether it has produced the same output as the teacher. If the result is not the same, the student updates its weight vectors accordingly. In our model we assume the network updates its weight vectors based on a fixed set of input data. This version of learning is referred to as *off-line learning*, as opposed to

on-line learning, in which data becomes available over time and in which the weight vectors are revised after every new input example.

The performance of a neural network relates directly to the ease or speed with which the student learns the rule (*training rate*). Roughly speaking, the training rate depends positively on the number of examples that are presented to the network P, and depends negatively on the number of parameters that are included in the input data $\boldsymbol{\xi}$ (or simply put the size of $\boldsymbol{\xi}$) N. There are two more parameters which affect the performance: the number of hidden units K, and the temperature β, a concept that is adopted from statistical physics. Their combined effect is encapsulated by the rescaled number of examples per student weight α, defined as:

$$\alpha = \frac{\beta P}{NK} \tag{4}$$

This parameter will be important in sections 3.3 and 3.4.

3.2 Measure of performance: generalization error

Once we have trained the student network, we would like to have a measure of performance. We will use the *generalization error* ϵ_g for this, which indicates how accurately our network determines the outcome for new data. It is defined as the average difference in outcome between the student- and teacher network for a random input with i.i.d. components $\boldsymbol{\xi}$:

$$\epsilon_g = \tfrac{1}{2}\Big\langle [S(\boldsymbol{\xi}) - \sigma(\boldsymbol{\xi})]^2 \Big\rangle \tag{5}$$

This is the standard definition which is applied when a network is being trained. However, we remind ourselves that we are not actually training a network, but simulate the scenario of training a network. This implies that we have an arbitrary student and teacher, of which we do not know the actual outcome. Instead, we would like to compute the performance of an arbitrary student that depends on the number of examples it has been presented with.

Saad et al. show that in the limit $N \to \infty$ the generalization error only depends on the *order parameters* T_{km}, R_{km}, and Q_{km} [14], where:

$$T_{km} = \langle y_k y_m \rangle = \tfrac{1}{N}\boldsymbol{w_k^*} \cdot \boldsymbol{w_m^*} \ , \ R_{km} = \langle x_k y_m \rangle = \tfrac{1}{N}\boldsymbol{w_k} \cdot \boldsymbol{w_m^*} \ , \ Q_{km} = \langle x_k x_m \rangle = \tfrac{1}{N}\boldsymbol{w_k} \cdot \boldsymbol{w_m} \tag{6}$$

R_{km} can be interpreted as the overlap between the weight vector of unit k in the student network and the weight vector of unit m in the teacher network. Q_{km} denotes the overlap between weight vectors of two student units k and m. Note that the overlap between two teacher weight vectors $T_{km} = \delta_{km}$, because the teacher weight vectors are assumed to be orthogonal (as mentioned in section 3.1). As a result, we do not need to consider T_{km} in the remainder of our analysis.

In our analysis we will assume that the student network has learned the rule when one of its weight vectors $\boldsymbol{w_k}$ aligns with one of the teacher weight vectors $\boldsymbol{w_m^*}$, and when the other student weight vectors show no overlap with this teacher weight vector. This is referred to as *site-symmetry*. Obviously, the system is invariant under hidden unit permutations, so we can restrict ourselves to one case. We therefore choose the case of matching indices $k = m$. As a result, the following holds for the order parameters:

$$R_{km} = \begin{cases} R & \text{if } k = m \\ S & \text{otherwise} \end{cases} \qquad Q_{km} = \begin{cases} 1 & \text{if } k = m \\ C & \text{otherwise} \end{cases} \tag{7}$$

The exact formula for the generalization error ϵ_g differs per activation function $g(x)$, as is discussed in the following sections.

3.2.1 Sigmoidal activation

Sigmoidal activation has been applied often and has proved itself as a steady activation function. The activation function is of the form

$$g(x) = \mathrm{erf}(x/\sqrt{2}) \tag{8}$$

For our analysis we have however opted for $g(x) = 1 + \mathrm{erf}(x/\sqrt{2})$, which yields the same ε_g. Saad et al. [14] derived the generalization error in the limit $N \to \infty$:

$$\epsilon_g = \frac{1}{6} + \frac{1}{K\pi} \sum_{i,j=1}^{K} \left[\arcsin\left(\frac{Q_{ij}}{2}\right) - 2\arcsin\left(\frac{R_{ij}}{2}\right) \right] \tag{9}$$

As a result of site symmetry (9) can be reduced to:

$$\epsilon_g = \frac{1}{3} + \frac{K-1}{\pi} \left[\arcsin\left(\frac{C}{2}\right) - 2\arcsin\left(\frac{S}{2}\right) \right] - \frac{2}{\pi} \arcsin\left(\frac{R}{2}\right) \tag{10}$$

3.2.2 ReLU activation

Rectified linear unit or ReLU is a relatively new activation function which has been shown to enable better training, in particular for networks with many hidden layers, referred to as deep networks [15]. It is nevertheless worthwhile to look into its effects on a network with only a single hidden layer. The function is described by

$$g(x) = \max(0, x) \tag{11}$$

The generalization error has been worked out by Straat et al. [12], and is given by:

$$\begin{aligned} \epsilon_g(\boldsymbol{J}) = & \frac{1}{2} + \frac{K-1}{4\pi} + \frac{1}{K} \left[\sum_{i=1}^{K} \sum_{j>i}^{K} \left(\frac{Q_{ij}}{4} + \frac{\sqrt{1-Q_{ij}^2}}{2\pi} + \frac{Q_{ij} \sin^{-1}(Q_{ij})}{2\pi} \right) \right. \\ & \left. - \sum_{i=1}^{K} \sum_{m=1}^{K} \left(\frac{R_{im}}{4} + \frac{\sqrt{1-R_{im}^2}}{2\pi} + \frac{R_{im} \sin^{-1}(R_{im}))}{2\pi} \right) \right] \end{aligned} \tag{12}$$

Assuming site-symmetry, this simplifies to:

$$\begin{aligned} \epsilon_g(\boldsymbol{J}) = & \frac{1}{2} + \frac{K-1}{4\pi} + \frac{K-1}{2} \left(\frac{C}{4} + \frac{\sqrt{1-C^2}}{2\pi} + \frac{C \sin^{-1}(C)}{2\pi} \right) \\ & - \frac{R}{4} - \frac{\sqrt{1-R^2}}{2\pi} - \frac{R \sin^{-1}(R)}{2\pi} \\ & - (K-1) \left(\frac{S}{4} + \frac{\sqrt{1-S^2}}{2\pi} + \frac{S \sin^{-1}(S)}{2\pi} \right) \end{aligned} \tag{13}$$

3.3 Free energy

In statistical physics a system is considered to be in equilibrium when its *free energy* is minimized. An important notion involved in this process is *entropy*. Roughly speaking, it is a measure of how disorganized a system is. Entropy lowers the energy of a system, and as a result entropy tends to maximize when a system is in a state of equilibrium. To illustrate this, consider the following example. Say, we take a chamber in which we place a slider that divides the room in two. In one of the two departments a gas is contained. If we now remove the slider, the system would be most organized if the gas would remain where it is. However, as we may expect, the gas will

distribute itself randomly over the entire chamber, effectively maximizing entropy [13].

This concept can be linked back to training a neural network. The process of training a student network with K hidden units can be be considered as minimization of the free energy of the network (system) with respect to the order parameters. We would want to obtain an organized system of K units, but entropy is a competing process which tries to randomize the network. The network gets better by training with rate $\alpha K\epsilon_g$, but entropy s counteracts this. The free energy f can thus be summarized by

$$f = \alpha K\epsilon_g - s \tag{14}$$

Entropy is not dependent on the activation function, and is hence the same for sigmoidal and ReLU. For large N, the general formula for entropy s is given by [11] [16]:

$$s = \frac{\ln(\det \mathcal{C})}{2} + \text{const} \tag{15}$$

where $\mathcal{C}$ is the covariance matrix of size $2K \times 2K$:

$$\mathcal{C} = \begin{pmatrix} Q_{11} & \cdots & Q_{1K} & R_{11} & \cdot & R_{1M} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ Q_{1K} & \cdots & Q_{KK} & R_{K1} & \cdot & R_{KM} \\ R_{11} & \cdots & R_{K1} & T_{11} & \cdot & T_{1M} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ R_{1M} & \cdots & R_{KM} & T_{M1} & \cdot & T_{MM} \end{pmatrix} \tag{16}$$

By assuming site-symmetry (15) reduces to [11] [16]:

$$s = \frac{1}{2}\ln[1 + (K-1)C - ((R-S) + KS)^2] + \frac{K-1}{2}\ln[1 - C - (R-S)^2] \tag{17}$$

3.4 Phase transitions

During the training process the student's accuracy will improve as it is presented with more examples. This dependency is not continuous, however [11]. Instead, there is a point α^* at which ϵ_g suddenly decreases with a faster rate, which shows as a discontinuity in the graph of ϵ_g as a function of α. At the point α^* the system specializes: it finds a "path" that leads to the configuration with minimal energy. Such a point is referred to as a *phase transition*, its name taken from the phenomenon in which matter changes from one phase (solid, liquid, gas, plasma) to another.

In earlier work by Biehl et al. [11] two types of phase transitions have been observed: a continuous (second order) transition and a discontinuous (first order) transition. A second order transition consists of one point α^* at which the global minimum of the free energy is replaced by two local minima, one of which becomes the new global minimum. The former global minimum is referred to as the unspecialized state, whereas the new global minimum is referred to as the specialized state. In a first order transition, the student goes through a meta-stable state before it specializes:

- $\alpha < \alpha_a$: the system is not yet specialized, only an unspecialized minimum exists.
- $\alpha_a \leq \alpha < \alpha_c$: in this region a specialized and unspecialized minimum co-exist.

  At some point within this region the free energy of the specialized and unspecialized branch are equal. This point is marked as α_b.
- $\alpha \geq \alpha_c$: the unspecialized minimum is replaced by a specialized state, in which the system will remain from this point on.

In a second order transition the competing minima emerge simultaneously and initially have the same free energy. This allows for a quick transition between the two minima. In a first order transition, the gap between the global minimum and local minimum is large, which impedes the specialization. This makes a second order specialization the preferred type. Phase transitions are hence an important indicator of the performance of an activation function. Moreover, if for the same number of hidden units, two different activation functions have a different type of phase transition, then this suggests that the two functions are fundamentally different. On the contrary, the value of α^* at which the phase transition occurs is not indicative of the training rate, as the α-values of two activation functions are not necessarily comparable.

4 Analysis

4.1 Minimization of the free energy

We wish to minimize the free energy function f (14) with respect to the order parameters $\{R, S, C\}$ for several α. We will do this for sigmoidal and ReLU activation, and for a varying number of hidden units K. We need to initialize the process by providing values for $\{R, S, C\}$ close to $\alpha = 0$. At the start of the minimization process we assume that the student units show little to no overlap with the teacher units, so $R = S = 0$. We also assume the student's weight vectors to be randomly initialized, meaning a C close to 0. We therefore initialize $\{R, S, C\}$ with $\{0, 0, 0\}$.

During the minimization process it is necessary to put constraints on the parameters, as certain combinations may lead to minima which are physically not possible. Consequently, we must perform constrained minimization. For this purpose MATLAB's built-in function `fmincon` was used. The constraints are derived from the definition of entropy: entropy cannot be negative ($s \geq 0$) and should be real (non-imaginary). From (15) it follows that the determinant of the covariance matrix $\mathcal{C}$ should be positive. Moreover, the determinants of the square submatrices of $\mathcal{C}$, denoted by $\mathcal{C}_i$, should be positive.

$$\mathcal{C}_i = \begin{pmatrix} C_{11} & C_{12} & \dots & C_{1i} \\ C_{21} & C_{22} & \dots & C_{2i} \\ \vdots & \vdots & \ddots & \vdots \\ C_{i1} & C_{i2} & \dots & C_{ii} \end{pmatrix} \tag{18}$$

In order to find the point(s) at which the phase transition occurs, we need to consider the specialized solution as well as the unspecialized solution. We therefore wish to be able to enforce either branch. We note that in the case of an unspecialized solution the student's weight vectors will show a small degree of overlap with every teacher weight vector. There will be no difference between the overlap of student unit i with teacher unit i (R) and the overlap of student unit i with teacher unit j (S). In other words, $R = S$, or $S - R = 0$. In the case of a specialized solution we expect that the weight vector of student unit i will show more overlap with the weight vector of teacher unit i than any other teacher unit j where $j \neq i$. Eventually, we hope to achieve a maximum overlap between student unit i and teacher unit i. Hence $R \neq S$ holds in this case. These facts allow us to add one more constraint which differs per branch. This lets us distinguish between the specialized and unspecialized solution.

4.2 Validity of the solution

It is essential to verify whether the solution that was found by the minimization process is a true minimum. As a first step, we should confirm that the gradient of the free energy function $\bigtriangledown f$ is equal to zero when we fill in the obtained values. A true minimum also entails that the solution should be stable. Stability is ensured when every eigenvalue of the Hessian matrix $\boldsymbol{H}$ is positive. The elements of the matrix are defined as

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}, \tag{19}$$

where x_{ij} is the ith element in the list of order parameters $\boldsymbol{x} = \{R_{ij}, Q_{ij}\}$. In the specialized case the solution should be stable for every α.

The test also has a second means: we can verify the soundness of the site-symmetry assumption. In order to test whether this assumption holds, we will confirm whether (7) indeed holds.

4.3 Finding the phase transitions

If we enforce the system to specialize, a phase transition will eventually occur, which can be identified as a discontinuity in the graph of the generalization error ϵ_g. The point at which this happens corresponds to α^* in the case of a second order phase transition, and to α_a in case of a first order phase transition. In order to find α_c we must enforce the system to remain unspecialized. Subsequently, we verify until which α the solution is stable. The first value of α for which the solution is no longer stable corresponds to α_c. If α_c coincides with α_a then we can confirm that the type of transition is second order. If the value of α_c differs from α_a then we have found a first order transition.

We may perform a second test that involves plotting the difference between R and S around the transition point to determine whether the phase transition is of first or second order. If the graph shows no discontinuities the transition type can be assumed to be second order. It the graph does show a discontinuity, it indicates that the phase transition is of first order. This test may prove to be useful in the case of small differences between α_a and α_c.

5 Results

5.1 K = 1

We first perform the analysis for a neural network comprising of a single hidden unit. This is equivalent to training a perceptron. We study the rate at which $R \to 1$, as well as the rate at which $\varepsilon_g \to 0$. From Figure 1 (left) we observe that the rate at which $R \to 1$ is marginally faster for ReLU than for sigmoidal activation. There are no clear differences between the activation functions when inspecting the rate at which $\varepsilon_g \to 0$ (refer to Figure 1, right).

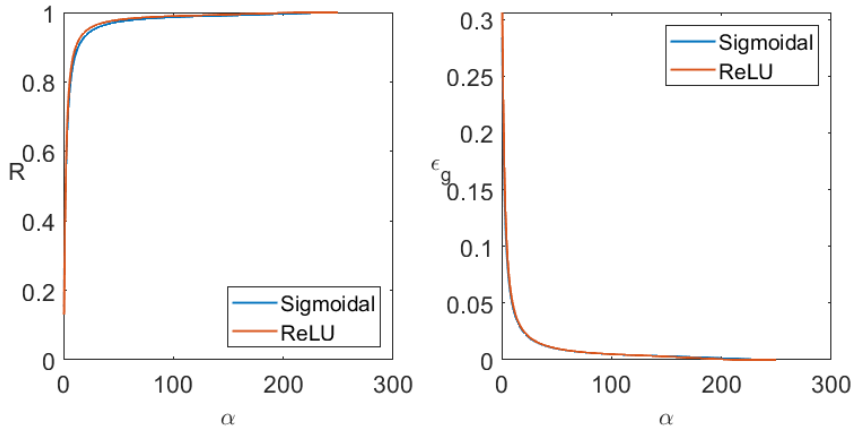

Figure 1: Learning rate of a single hidden unit, depicted in two ways. The behavior of the two activation functions does not differ significantly.

5.2 K = 2

We expand our network to two hidden units. In the student network using sigmoidal activation as well as in the student network using ReLU we now observe a specialization: after a certain value of α, R and S diverge. Two minima emerge, one in which $R \to 1$ and $S \to 0$, the other in which the opposite occurs. The two solutions have the same ε_g, and hence we have only plotted one of the two ($R > S$). We verify that this solution is the global minimum by comparing ε_g of the specialized solution ($R > S$) with that of the unspecialized solution ($R = S$). We find that indeed the specialized state yields a lower generalization error. We remark that the difference in ε_g between the specialized and unspecialized state is larger for ReLU than for sigmoidal activation.

In the case of sigmoidal activation we find a continuous (second order) transition with $\alpha^* = 23.6$. This agrees with the findings from Biehl et al. [11]. For ReLU we also find a second order transition, with $\alpha^* = 6.10$. We should note though that the graphs for $|R - S|$ are not as expected: we would expect $|R - S|$ to be equal to 0 up until α^*, followed by an increase of the form $(\alpha - \alpha^*)^{\frac{1}{2}}$. Instead, we observe values of $|R - S|$ in the order of 10^{-2} prior to α^* for both activation functions. We believe this is due to numerical issues, which can be fixed by requiring higher accuracy, for which we would have to use Mathematica instead of MATLAB.

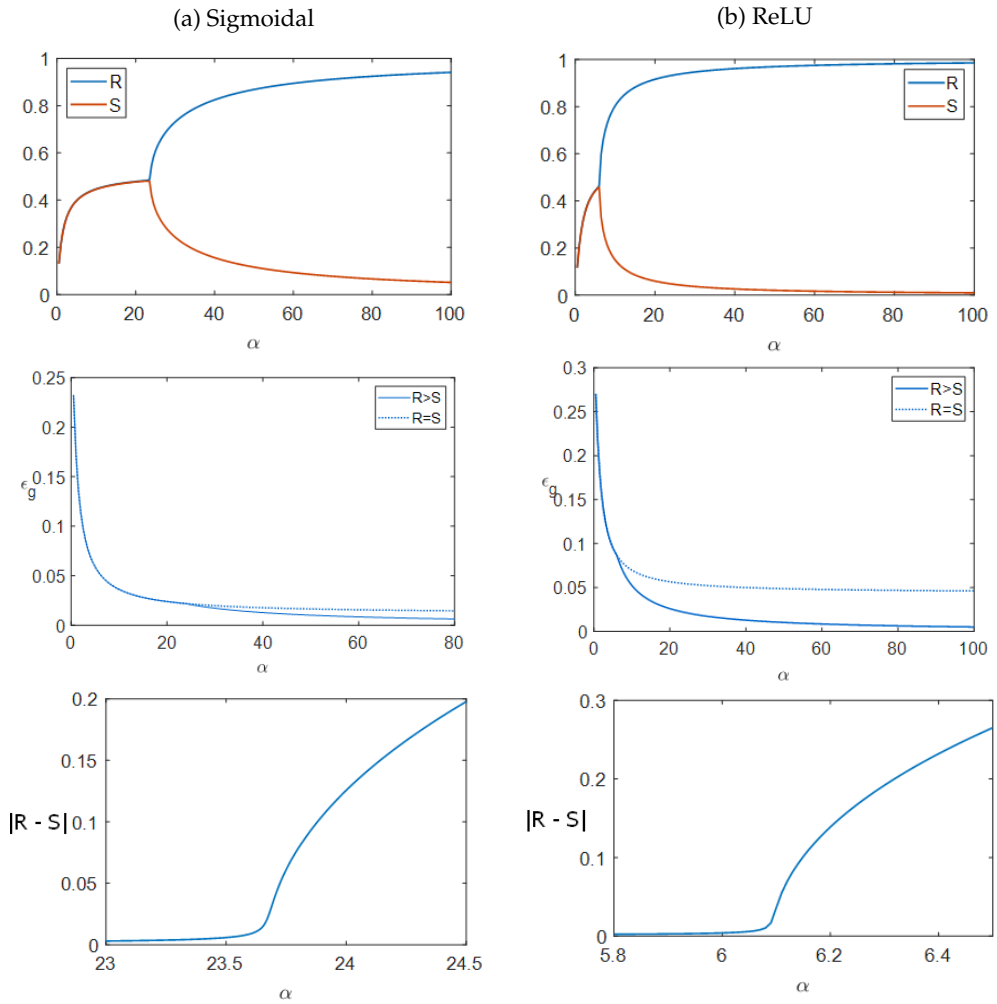

Figure 2: Specialization of a student network with $K = 2$ hidden units. Left: sigmoidal activation. Right: ReLU. Both networks undergo a second order phase transition.

5.3 K = 3

When we add an extra unit to our network the type of transition changes in the case of sigmoidal activation: we find a discontinuous (first order) transition with $\alpha_a = 34.0$ and $\alpha_c = 36.7$. Interestingly, ReLU maintains a second order transition, giving an indication of why ReLU may perform better than sigmoidal for $K = 3$ units. We also find an α^* which is hardly any larger than the previously observed value: $\alpha^* = 6.24$. Here we again have the issue of the graphs for $|R - S|$ not matching with our expectations. For ReLU we have the same problem as described for $K = 2$. For sigmoidal we would expect a vertical line at α_c, indicating a sudden jump in the value of $|R - S|$, instead of a line of finite slope. This can again be corrected for by switching to Mathematica.

Additionally, for both activation functions we find a local minimum in which $(R < S)$. We confirm that this is indeed a local minimum by comparing the values of ε_g of the three solutions. What is interesting though, is that for ReLU the values of ε_g for the solutions $(R > S)$ and $(R < S)$ are quite close, especially compared to sigmoidal activation.

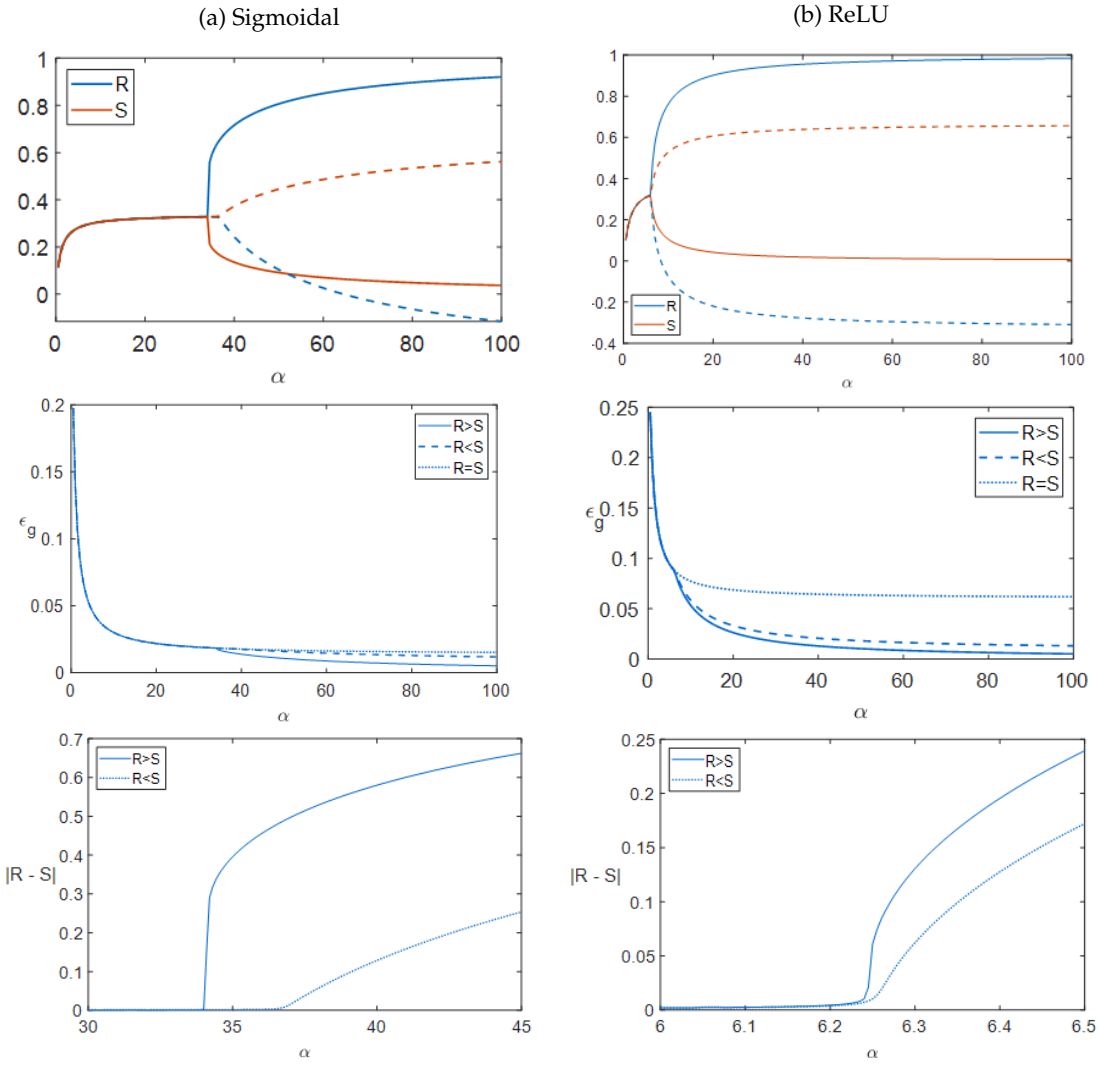

Figure 3: Specialization of a student network with $K = 3$ hidden units. Left: sigmoidal activation. Right: ReLU. While sigmoidal activation displays a first order transition, ReLU maintains a second order transition, which is beneficial for the learning.

6 Discussion

The findings presented in the previous section reveal that although ReLU shows similar behavior to sigmoidal activation for $K = 1, 2$ units, for $K = 3$ units it displays fundamentally different behavior: while sigmoidal activation exhibits a first order phase transition, ReLU experiences a second order phase transition. This might be a strong indication of why ReLU performs better than sigmoidal, at least for this number of hidden units.

What is also remarkable, is that for ReLU it appears the α at which the phase transition occurs barely changes with the number of units K. For sigmoidal activation the transition point clearly shifts with a larger number of hidden units, or in other words, α^* is dependent on K [11]. For ReLU we observe a trend as well, but the values of α^* are so close, it appears as though α^* might be only weakly dependent on K. We would have to inspect cases of larger K though, eventually going to the limit $K \to \infty$, in order to conclude something meaningful about this.

Additionally, we found a solution to the system which has not been reported before, namely a state in which $(R < S)$. For $K = 2$ this solution has a generalization error which is identical to the other specialized solution $(R > S)$. Although for $K = 3$ the $(R < S)$ solution is only a local minimum, the differences in the generalization error between this "anti-specialized" state and the specialized state $(R > S)$ are small for ReLU. It may be worthwhile to investigate whether this difference changes in the case of a larger number of hidden units.

The main weakness of our study would be the limited scope: our study would have been more meaningful if we could have expanded the networks even further. However, due to limited time we were not able to do so. Although a first attempt was made at computing the learning rates for $K = 4, 5$ units, our method proved to be inaccurate for these networks. Another flaw in our results is discrepancies in the course of $|R - S|$. Before specialization this value should be 0, but for the case sigmoidal $K = 2$ as well as for cases ReLU $K = 2, 3$ we find values in the order of 10^{-2}. Moreover, for sigmoidal $K = 3$ we should observe a sudden jump in the value of $|R - S|$, indicated by a vertical line, but instead the line has a finite steepness. These inaccuracies are most likely due to numerical issues stemming from MATLAB's `fmincon`. We therefore opt to use Mathematica in the future, which allows for higher accuracy.

7 Conclusion

Rectified Linear Unit activation exhibits a better performance than the widely used sigmoidal activation [7–9]. So far only empirical data existed to support this idea, but our theoretical analysis shows evidence for this as well. We found that ReLU still experiences a second order phase transition for $K = 3$ units, whereas sigmoidal undergoes a first order phase transition in this case. A second order transition is beneficial to the performance, since a first order phase transition goes through a meta-stable state in which both the specialized and unspecialized solution exist before specializing, as opposed to a second order phase transition in which the system immediately specializes. Another remarkable finding is that for ReLU the point of specialization barely shifts as the number of hidden units increases. This is in contrast to sigmoidal, which shows a clear correlation between the two. Further research is needed to confirm this idea though, in which we would investigate cases of even larger K, ultimately going towards $K \to \infty$. We suggest to use Mathematica in the future, as it appears to be more suitable for this purpose than MATLAB.

In our study we have been able to illustrate the differences between sigmoidal activation and ReLU activation, giving a first indication why ReLU might perform better than sigmoidal. We have however not yet been able to unravel the fundamental reason why ReLU specializes faster than sigmoidal, and why its behavior differs from sigmoidal activation in terms of phase transitions. A deeper analysis would be needed for this, which we hope to provide in the near future.

References

[1] J.A. Hertz, A.S. Krogh, and R.G. Palmer. *Introduction To The Theory Of Neural Computation*. Addison-Wesley, Reading, MA, USA, 1991.

[2] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, New York, NY, USA, 2001.

[3] A. Engel and C. Van den Broeck. *Statistical Mechanics of Learning*. Cambridge University Press, 2001.

[4] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995.

[5] C.M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, Heidelberg, Germany, 2006.

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[7] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. 30th ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[8] V. Nair and G.E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proc. 27th International Conference on Machine Learning (ICML)*, pages 807–814, USA, 2010. Omnipress.

[9] T. Villmann, J. Ravichandran, A. Villmann, D. Nebel, and M. Kaden. Investigation of Activation Functions for Generalized Learning Vector Quantization. In A. Vellido, K. Gibert, C. Angulo, and J. Martín Guerrero, editors, *Advances in Self-Organizing Maps, Learning Vector Quantization, Clustering and Data Visualization, WSOM 2019*, volume 976 of *Advances in Intelligent Systems and Computing*, pages 179–188, Cham, 2019. Springer.

[10] T. L. H. Watkin, A. Rau, and M. Biehl. The statistical mechanics of learning a rule. *Rev. Mod. Phys.*, 65(2):499–556, 1993.

[11] Biehl, M., Schlösser, E., and Ahr, M. Phase transitions in soft-committee machines. *Europhys. Lett.*, 44(2):261–267, 1998.

[12] M. Straat and M. Biehl. On-line learning dynamics of ReLU neural networks using statistical physics techniques. In M. Verleysen, editor, *27th Europ. Symp. on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 517–522. i6doc.com, 2019.

[13] F. Mandl. *Statistical Physics*. John Wiley And Sons Ltd, 1988.

[14] David Saad and Sara A. Solla. On-line learning in soft committee machines. *Phys. Rev. E*, 52:4225–4243, Oct 1995.

[15] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *AISTATS*, 2011.

[16] M. Ahr, M. Biehl, and R. Urbanczik. Statistical physics and practical training of soft-committee machines. *The European Physical Journal B-Condensed Matter and Complex Systems*, 10(3):583–588, 1999.