Performance of robust training algorithms for neural networks

1995 J. Phys. A: Math. Gen. 28 5485

(http://iopscience.iop.org/0305-4470/28/19/006)

J. Phys. A: Math. Gen. 28 (1995) 5485–5493. Printed in the UK

Performance of robust training algorithms for neural networks

Andreas Wendemuth†

Department of Physics, Theoretical Physics, Oxford University, 1 Keble Road, Oxford, OX1 3NP, UK

Received 5 April 1995

Abstract. The performance of three recently presented training algorithms in neural networks is investigated. These algorithms are robust to infeasible problems, in which case an appropriate error function is minimized. In the infeasibility regime, simulations are performed and compared to recently published analytical work in one-step replica symmetry broken theory. A careful analysis explains insufficiencies in these analytic results. A new stability result in the infeasibility regime is derived and shown to match simulation data.

1. Introduction

To date, a number of training algorithms (e.g. [2, 7, 13]) for neural networks have been published. However, these algorithms focus on learnable problems in which case convergence can be proved. In the case of unlearnable problems, these algorithms are not robust in the sense that they minimize an appropriate cost function.

However, from the point of view of cost functions rather than algorithms, statistical mechanics techniques may be used to investigate properties of the network under certain cost functions even in the infeasiblity regime. In this context, Majer *et al* [8] have recently calculated the distribution of stabilities in perceptrons beyond saturation. The validity of these results cannot, however, be verified as long as no robust algorithms are at hand for simulations in the infeasibility regime.

This gap can now be bridged. Recently, three algorithms have been presented which are robust versions of the Rosenblatt, minover and Gardner–Derrida algorithms [14]. In this paper they are used to train perceptrons with infeasible problems. The obtained stability distributions are analysed and compared to the results of Majer *et al*. Discrepancies are observed which can now be explained in detail from the geometric interpretation of the algorithmic performance. Additional theoretical work is presented which explains the stability distributions for maximum stability scenarios.

2. Problem and robustness

The given problem is to find an N-dimensional perceptron vector J such that for a given set of N-dimensional patterns $\{\xi_1, \ldots, \xi_p\}$ and corresponding outputs ('targets') $\{\tau_1, \ldots, \tau_p\}$

† Present address: Philips Research Laboratory, Weisshausstrasse 2, 52066 Aachen, Germany. E-mail address: Wendemu@pfa.philips.de

0305-4470/95/195485+09$19.50

($\xi_\mu \in \mathbb{R}^N$; $|\xi_\mu|^2 = N\ \forall\mu$; $\tau_\mu \in \{\pm 1\}$) the equations

$$\tau_\mu = \mathrm{sign}(J \cdot \xi_\mu) \qquad \mu = 1, \ldots, p \tag{1}$$

are satisfied if the problem is feasible or, otherwise, the given error functionals of (1) are minimized (see below).

In a geometric context, equations (1) requires a plane through the origin separating the points ξ_μ into two classes with positive and negative output. The minimum distance of any ξ_μ to the plane is the measure of stability

$$\Delta =: \min_\mu \Delta_\mu =: \min_\mu \left(\frac{J \cdot \tau_\mu \xi_\mu}{|J|} \right). \tag{2}$$

Thus, the perceptron of optimal stability is (J^*) such that

$$\Delta_{\mathrm{opt}} = \min_\mu \left(\frac{J^* \cdot \tau_\mu \xi_\mu}{|J^*|} \right) = \max_{\{J\}} \min_\mu \left(\frac{J \cdot \tau_\mu \xi_\mu}{|J|} \right). \tag{3}$$

Defining $\sigma_\mu = \tau_\mu \xi_\mu$, training will be performed with the robust algorithm presented in [14]:

$$J^{(t+1)} = J^{(t)} + \frac{1}{N} \left(\sigma^{(t)} + \frac{N - J \cdot \sigma^{(t)}}{(J^{(t)})^2 - J^{(t)} \cdot \sigma^{(t)}} J^{(t)} \Theta(-J \cdot \sigma^{(t)}) \right). \tag{4}$$

3. A high number of stabilized patterns (Gardner–Derrida)

The Gardner–Derrida cost function represents the maximized number of stabilized patterns. Finding the maximum number of stabilized patterns in the infeasible case has been shown [1] to be non-deterministic polynomial complete (NP-complete). This tells us that, in the worst case, we are facing a very difficult problem. However, here we investigate algorithms achieving locally optimal solutions which in the average case are acceptable, as will be seen below. Indeed, various approaches for maximizing the number of unstabilized patterns in unlearnable problems have been reported. Amongst them are the works by Rujan [11], Frean [4] and Gallant [5], all of which are briefly discussed in the context of the present work in [14]. However, these algorithms include time-expensive schemes like annealing procedures or the storage of intermediate sets of weights, which are not needed in the three algorithms used here.

A pattern is regarded as stabilized if $(J^{(t)} \cdot \sigma_\mu)/|J^{(t)}| > \kappa$. Maximizing the number of stabilized patterns will, for example, be useful if one is 'only' interested in learning, or if the patterns are noisy and therefore unreliable. For learnable problems, all patterns will be stabilized but the results will be otherwise suboptimal. In this case, one may prefer the third algorithm presented here which yields local maximum stability.

For unlearnable problems, maximizing the number of stabilized patterns means that a subset of patterns will be allowed to have very low (negative) stabilities. One does not aim at using these patterns for training. Instead, update steps are taken with the pattern which is easiest to stabilize at the current iteration. It was shown in [14] that, in this spirit, a locally maximized number of stabilized patterns will be reached if one uses at any iteration step the one incorrectly mapped pattern with stability *closest to* κ, i.e. update steps are performed according to equation (4) where $\sigma^{(t)}$ is given by

$$J^{(t)} \cdot \sigma^{(t)} = \max_\mu \{ J^{(t)} \cdot \sigma_\mu \mid J^{(t)} \cdot \sigma_\mu / |J^{(t)}| < \kappa \}.$$

The stability distribution, and consequently the rate of errors of the robust Gardner–Derrida algorithm, can be compared to theoretical results given by Majer *et al* [8]. As an

example of their results, the authors investigate the distribution of stabilities at $\alpha = 1$, $\kappa = 1$ which is an unlearnable problem. They obtain a gap below κ in the stability distribution. Going from replica symmetric (RS) theory to one-step replica symmetry broken (RSB) theory, the gap narrows, the unstabilized branch of the distribution becomes flatter, the stabilized branch becomes increasingly peaked, and the weight of the δ-function at $\Delta = \kappa$ decreases. It is unclear from the theoretical investigation whether these effects are becoming more severe with further steps of RSB. The one-step RSB results are plotted in figure 1.

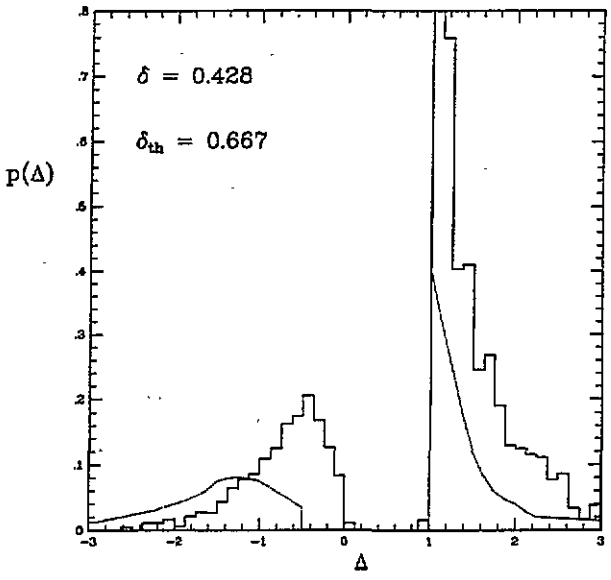

Figure 1. Stability distribution $p(\Delta)$ for the Gardner–Derrida cost function algorithm with desired stability $\kappa = 1$ at $\alpha = 1$: dotted curves, analytic result in one-step RSB [8], exhibiting a δ-function at $\Delta = 1$ with weight 0.667; histogram, simulations with $N = 500$ and 100 000 steps, averaged over 100 pattern instances. The weight accumulated in $1 \leqslant \Delta < 1.125$ is 0.428.

These results are compared to simulation data obtained with the robust Gardner–Derrida algorithm. It has already been shown in [14] that this algorithm will produce a gap below κ in the stability distribution, in accordance with [8]. In simulations, 100 000 update steps have been taken in networks with patterns of 500 randomly chosen binary inputs, again for $\alpha = 1$ and $\kappa = 1$. This has been repeated for 100 independently chosen sets of patterns, the stability distributions have been sampled and averaged. They are shown as a histogram in figure 1. The simulation data verify the *tendency* of the theoretical results, they exhibit the predicted features such as the gap, the two branches and the δ-function. However, the tendency which was observed by Majer *et al* [8] in the first step of RSB continues: the gap becomes even narrower, the unstabilized branch of the distribution keeps its Gaussian shape but is shifted, the stabilized branch becomes increasingly peaked, and the weight of the δ-function at $\Delta = \kappa$ decreases.

With these observations, it must be questioned whether one-step RSB already gives sufficiently accurate results. The claim of Majer *et al* [8] that 'the effects of RSB appear to be rather small' could not be verified.

The rate of errors in this case is $\simeq 0.14$ in one-step RSB and 0.18 in the simulations. This again indicates an error-rate increase in further steps of RSB which is a continuation of the increase observed by Majer *et al* [8] at the first RSB step. The same effect has also

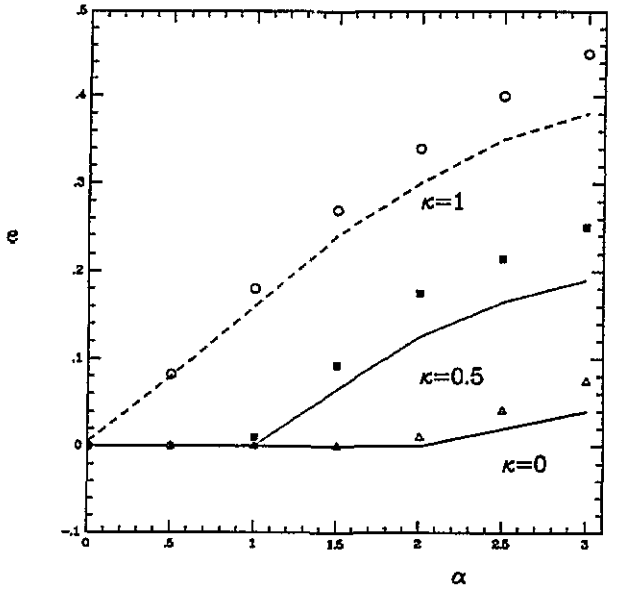

Figure 2. Error rate e of the Gardner–Derrida cost-function algorithm for three values of the desired stability κ at various α: curves, analytic result in one-step RSB [8]; points, simulations with $N = 100$ and 50 000 steps, averaged over 100 pattern instances. The standard errors of the mean are smaller than the size of the symbols.

been observed for other values of κ and α, as shown in figure 2.

4. Robust Rosenblatt perceptron

The second cost function and respective algorithm, in the spirit of Rosenblatt's perceptron learning algorithm, aims at learning any unstabilized pattern. The desired minimum stability is again chosen to be κ.

For learnable problems, all patterns will be stabilized but the results will be otherwise suboptimal. In this case, one may again prefer the (third) maximum stability solution.

For unlearnable problems, all unstabilized patterns will be used in random order for the updates which leads to a steady state in the stability distribution. In particular, patterns which have been stabilized previously will, in the steady state, be located on either side of the stability threshold κ.

In [14] it was shown that one can perform updates according to equation (4), where according to the spirit of the original perceptron algorithm [10], $\sigma^{(t)}$ is taken *randomly* out of the set of patterns with $\Delta_\mu < \kappa$. The difference to the previous section is that this algorithm does not aim at maximizing the number of stabilized patterns, instead it treats any unstabilized pattern equally.

Theoretical results for the perceptron algorithm have again been obtained by Majer *et al* [8]. As an example, the authors have chosen $\alpha = 4$, $\kappa = 0.5$. The theoretical results and a histogram displaying the simulation results are shown in figure 3. Theory and simulation coincide well in the tails of the distribution which changed only incrementally when going from RS to one-step RSB. This suggests that further steps of RSB will not produce significant alterations.

The effect of an RSB decline of the weight of the δ-function at $\Delta = \kappa$ with a simultaneous

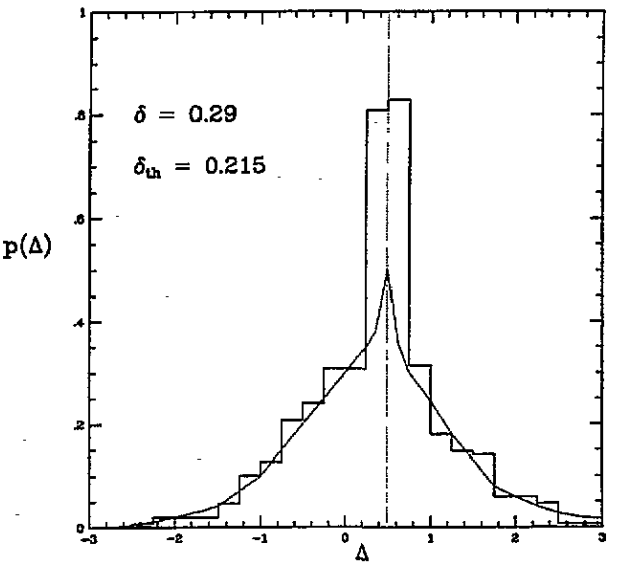

Figure 3. Stability distribution $p(\Delta)$ for the Rosenblatt cost-function algorithm with desired stability $\kappa = 0.5$ at $\alpha = 4$: dotted curves, analytic result in one-step RSB [8], exhibiting a δ-function at $\Delta = 0.5$ with weight 0.215, and a 'tip' at the same location; histogram, simulations with $N = 150$ and 100 000 steps. The weight accumulated in $0.25 \leqslant \Delta < 0.75$ is 0.29.

development of a 'tip' in the tails demands a more detailed investigation. The simulation data indeed verify the explanation of a steady state in the stability distribution where patterns will be located on either side of the stability threshold κ. This can be seen clearly from the simulation data which exhibit a tip with equal weight on either side of κ. As already shown in [14], J will always perform macroscopic jumps away from patterns at the boundary $\Delta_\mu = \kappa$. Further, stability relies on very long training sequences leading to a very large $|J|$ to ensure quasi-continuity of the movements taken by J and a quasi-homogenous frequency of non-stabilized patterns used for updating. This was well reflected in the simulations which needed $\geqslant 100\,000$ steps for the network under consideration, and in the need to average the results over a large number ($\geqslant 20\,000$) of final training steps in order to obtain the steady state results.

Majer *et al* [8] have attributed the full weight of their δ-function to the *stabilized* patterns; under this ongoing discussion, this appears to be incorrect. Further simulation results in figure 4 give the error rate. These results show that the difference between the measured and the predicted error rate is about one half of the weight of the δ-function. This clearly supports that, in the Rosenblatt solution, the patterns at the stability boundary $\Delta = \kappa$ alternate between stability and instability.

5. Maximum stability

Here, the gap size between the convex hulls of the two output clusters is taken to be maximal, even if the gap is negative. In the feasibility regime, maximum stability is often chosen since it leads to good generalization abilities and fault tolerance [9]. It is noted right away that in the negative stability regime, the *number* of patterns stabilized under this concept is far from maximal. If a high number of stabilized patterns is desired, the

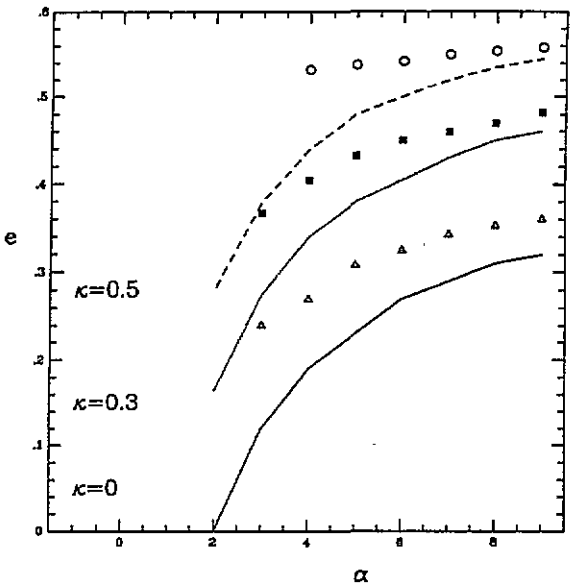

Figure 4. Error rate e of the Rosenblatt cost-function algorithm for three values of the desired stability κ at various α: curves, analytic result in one-step RSB [8]; points, simulations with $N = 100$ and 50 000 steps.

Gardner–Derrida solution will be useful.

In order to attain maximum stability, it has been shown in [14] that one can perform updates after equation (4) where $\sigma^{(t)}$ is given by $J^{(t)} \cdot \sigma^{(t)} = \min_\mu \{J^{(t)} \cdot \sigma_\mu\}$. In contrast to the previous sections, this is a *worst case update* where one aims only to improve the stability of the least highly stabilized pattern.

Let us now compare data obtained by this algorithm to analytically calculated stability distributions. In a statistical mechanics type calculation, the maximum stability $\Delta^*(\alpha)$ is found implicitly by an analytic continuation of the Gardner result [6]:

$$\frac{1}{\alpha} = \int_{-\Delta^*}^{\infty} \mathrm{D}z (z + \Delta^*)^2 \tag{5}$$

where $\mathrm{D}z = (2\pi)^{-1/2} \exp(-z^2/2)$ is the Gaussian measure. This result is confirmed by simulation data, see figure 5 as an example. It is also plausible that this result shall hold since the error measure of the corresponding algorithm is locally convex. Assuming equi-partitioning of these convex regions, replica symmetry can be expected to be unbroken. The stability distribution for feasible problems has been calculated in [12]:

$$p(\Delta) = \delta(\Delta - \Delta^*) \int_{-\Delta^*}^{\infty} \mathrm{D}z + \Theta(\Delta - \Delta^*) \frac{1}{\sqrt{2\pi}} \exp[-\Delta^2/2]. \tag{6}$$

Taking equation (5) into account, however, in this equation more than N patterns contribute to the weight of the δ-function for $\Delta^* < 0$ ($\alpha > 2$). This is in contradiction to results from optimization theory [3] which state that J^* will be given as a pseudo-inverse constructed from a set $\{\sigma_\mu^*\}$ of at most N patterns with equal overlap $J_\mu^* \cdot \sigma_\mu / |J^*| = \Delta^*$ to J^* (hence the δ-function).

It is as well confirmed by simulations which clearly show that for $\alpha > 2$, exactly N patterns constitute the δ-function, i.e. its weight is $1/\alpha$. Since the other patterns do not

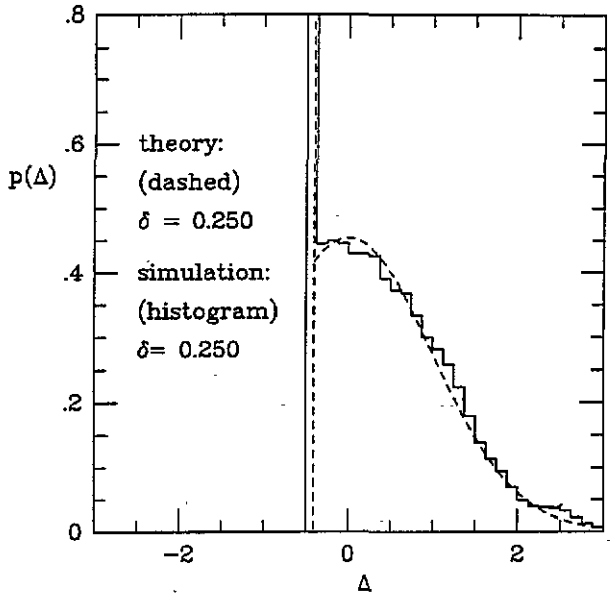

Figure 5. Stability distribution $p(\Delta)$ for the maximum stability cost-function algorithm at $\alpha = 4$: broken curves, analytic result, equation (13), exhibiting a δ-function at $\Delta = -0.41$ with weight 0.25; histogram, simulations with $N = 200$ and 200 000 steps. The weight accumulated in $0.5 \leqslant \Delta < 0.25$ is 0.250.

contribute to the result, it follows from geometric considerations that their distribution must be Gaussian with width one. This can be seen as follows.

Let $V(\Delta)$ be the volume of the surface on the N-dimensional ξ-shell with the condition that the stability is Δ, i.e.

$$V(\Delta) = \int \mathrm{d}\xi\, \delta(\xi^2 - N)\delta\left(\frac{1}{|J|}\boldsymbol{J}\cdot\boldsymbol{\xi} - \Delta\right). \tag{7}$$

Further, let $\rho(\Delta)$ be the density of the *free* patterns not contributing to the δ-function. Then, for randomly distributed patterns, from the ratio of volumes available for the patterns we find

$$\frac{\rho(\Delta_2)}{\rho(\Delta_1)} = \frac{V(\Delta_2)}{V(\Delta_1)}. \tag{8}$$

In order to compute this ratio we can choose the coordinate system such that $\boldsymbol{J} = (|J|, 0, \ldots, 0)$. Integrating over the first dimension, we obtain

$$\frac{V(\Delta_2)}{V(\Delta_1)} = \frac{\displaystyle\int \prod_{i=2}^{N} \mathrm{d}\xi_i\, \delta\left(N - \Delta_2^2 - \sum_{i=2}^{N}\xi_i^2\right)}{\displaystyle\int \prod_{i=2}^{N} \mathrm{d}\xi_i\, \delta\left(N - \Delta_1^2 - \sum_{i=2}^{N}\xi_i^2\right)}. \tag{9}$$

The numerator and denominator are now equal to the surface area of two $(N-1)$-dimensional spheres. Their ratio then reduces to the ratios of radii, taken to the power $(N-2)$. We obtain

$$\frac{\rho(\Delta_2)}{\rho(\Delta_1)} = \frac{V(\Delta_2)}{V(\Delta_1)} = \left(\frac{N - \Delta_2^2}{N - \Delta_1^2}\right)^{-1+N/2}. \tag{10}$$

For $N \gg \Delta_2, \Delta_1$, this equation gives

$$\rho(\Delta_2)\exp(\Delta_2^2/2) = \rho(\Delta_1)\exp(\Delta_1^2/2) \tag{11}$$

which is readily solved by

$$\rho(\Delta) = \text{constant} \times \exp(-\Delta^2/2). \tag{12}$$

This leaves as the only possible solution the following *rescaling* of the *full* stability distribution for $\alpha > 2$:

$$p(\Delta) = \delta(\Delta - \delta^*)\frac{1}{\alpha} + \Theta(\Delta - \Delta^*)\frac{1 - 1/\alpha}{\int_{\Delta^*}^{\infty} \mathrm{D}z}\frac{1}{\sqrt{2\pi}}\exp(-\Delta^2/2). \tag{13}$$

As an example, maximum stability has been investigated for $\alpha = 4$. The theoretical curve from equation (13) is plotted together with simulation data in figure 5, showing good agreement. In particular, the δ-function is exactly at the position predicted by equation (5) and has the weight $0.25 = 1/\alpha$, as demanded by optimization theory.

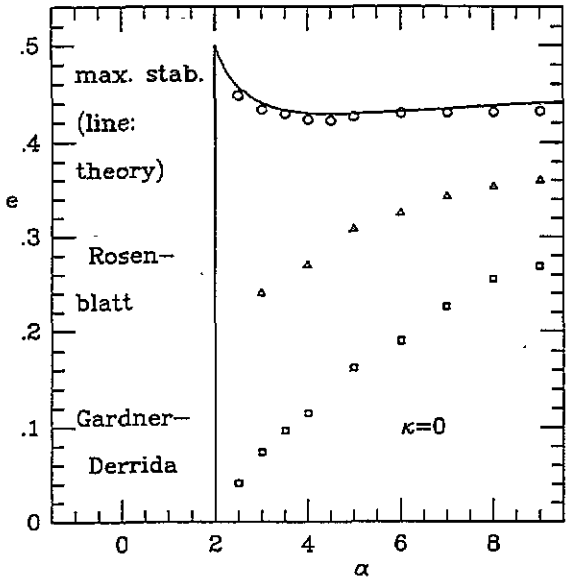

Figure 6. Error rate e of all three cost-functions algorithms for the desired stability $\kappa = 0$ at various α. Curve: analytic result for the minover solution, equation (15). Points: simulations with $N = 100$, 50 000 steps. (Top: maximum stability; centre: Rosenblatt; bottom: Gardner–Derrida.)

Further, from equation (13), the error rate $e(\kappa)$ is given by the fraction of patterns which have stability $\Delta < \kappa$, i.e. for $\kappa > \Delta^*$:

$$e(\kappa) = 1 - \left(1 - \frac{1}{\alpha}\right)\frac{\int_{\kappa}^{\infty} \mathrm{D}z}{\int_{\Delta^*}^{\infty} \mathrm{D}z}. \tag{14}$$

Simply asking for the fraction of wrongly mapped patterns (disobeying equation (1)) corresponds to the error rate $e(0)$ which is

$$e(0) = 1 - \left(1 - \frac{1}{\alpha}\right)\frac{1}{2\int_{\Delta^*}^{\infty} \mathrm{D}z} < \frac{1}{2}. \tag{15}$$

This is in contrast to the acclaimed belief that $e(0) = 0.5$ for the perceptron of maximum stability beyond saturation (e.g. in [8]). For $\kappa = 0$, figure 6 shows the theoretical curve after equation (15) (with Δ^* after equation (5)) and simulation data. The results agree well, confirming that the perceptron of maximum stability is not useless beyond saturation. Figure 6 also compares the error rate (15) to the error rates $e(0)$ of the Rosenblatt and Gardner–Derrida solutions. As expected, the Gardner–Derrida solution has the lowest error rate.

6. Conclusions

We have investigated the performance of three recently presented robust training algorithms for neural networks trained with infeasible problems. For the Gardner–Derrida and Rosenblatt solutions, the stability distributions generated by the robust algorithms explain insufficiencies in previous results from one-step replica symmetry broken theory. For the maximum stability solution, analytic results were presented which match the algorithmic results and which, in particular, show that this solution produces error rates less than $\frac{1}{2}$.

Stabilities and error rates with robust algorithms have therefore been shown to be theoretically predictable, using not only statistical mechanics but also insight from geometric explanations and from optimization theory. The presented material ensures that these algorithms provide a reliable tool in the understanding of neural networks beyond saturation, and in the approach to appropriate robust training of multilayer networks. Together, they pave a solid basis for both analytic and algorithmic understanding of robust neural network training of infeasible problems.

Acknowledgments

I would like to to thank O Winther, M Biehl and W Whyte for inspiring discussions. Support by the Friedrich-Naumann-Stiftung and the European Community under contract no ERB4001GT922302 is gratefully acknowledged. It is a pleasure to acknowledge support by CONNECT/ The Niels Bohr Institute, Copenhagen, where part of this work was performed.

References

[1] Amaldi E 1991 On the complexity of training Perceptrons *Artificial Neural Networks* ed T Kohonen, K Mäkisara, O Simula and J Kangas (Amsterdam: North-Holland)
[2] Anlauf J K and Biehl M 1989 The AdaTron algorithm *Europhys. Lett.* **10** 687
[3] Fletcher R 1987 *Practical Methods of Optimization* (New York: Wiley)
[4] Frean M 1992 A thermal perceptron learning rule *Neural Comput.* 4 946
[5] Gallant S I 1990 Perceptron-based learning algorithms *IEEE Trans. Neural Networks* **1** 179
[6] Gardner E 1988 The space of interactions in neural network models *J. Phys. A: Math. Gen.* **21** 257
[7] Krauth W and Mezard M 1987 *J. Phys. A: Math. Gen.* **20** L745
[8] Majer P, Engel A and Zippelius A 1993 Perceptrons beyond saturation *J. Phys. A: Math. Gen.* **26** 7405
[9] Opper M, Kinzel W, Klein H and Nehl R 1990 On the ability of the optimal perceptron to generalize *J. Phys. A: Math. Gen.* **23** L581-6
[10] Rosenblatt F 1961 *Principles of Neurodynamics—Perceptrons and the Theory of Brain* (Washington, DC: Spartan)
[11] Rujan P 1988 Searching for optimal configurations by simulated tunneling *Z. Phys.* B 73 391
[12] Wendemuth A, Opper M and Kinzel W 1993 The effect of correlations in neural networks *J. Phys. A: Math. Gen.* **26** 3165
[13] Wendemuth A 1994 Training of optimal cluster separation networks *J. Phys. A: Math. Gen.* **27** L387
[14] Wendemuth A 1995 Learning the unlearnable *J. Phys. A: Math. Gen.* in press