Statistical Mechanics of Learning : Generalization

Manfred Opper

Neural Computing Research Group

Aston University

Birmingham, United Kingdom

Short title: Statistical Mechanics of Generalization

Correspondence:

Manfred Opper

NCRG, Aston University, Aston Triangle, Birmingham B4 7ET

Phone:+44-121-333-4631

Fax:+44-121-4586

email: opperm@aston.ac.uk

1. INTRODUCTION

The theory of learning in artificial neural networks has benefited from various different fields of research. Among these, statistical physics has become an important tool to understand a neural network's ability to generalize from examples. It is the aim of this contribution to explain some of the basic principles and ideas of this approach.

In the following, we assume a feedforward network of N input nodes, receiving real valued inputs, summarized by the vector $\mathbf{x} = (x(1), \ldots, x(N))$. The configuration of the network is described by its weights and will be abbreviated by a vector of parameters $\mathbf{w}$. Using $\mathbf{w}$, the network computes a function $F_{\mathbf{w}}$ of the inputs $\mathbf{x}$ and returns $\sigma = F_{\mathbf{w}}(\mathbf{x})$ as its output.

In the simplest case, a neural network should learn a binary classification task. This means, it should decide, whether a given input $\mathbf{x}$ belongs to a certain class of objects and respond with the output: $F_{\mathbf{w}}(\mathbf{x}) = +1$ or, if not, it should answer with $\sigma = -1$ (the choice $\sigma = \pm 1$, rather than e.g. $0, 1$ is arbitrary and has no consequence for the learning curves). To learn the underlying classification rule, the network is trained on a set of m inputs $\mathbf{x}^m = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ together with the classification labels $\sigma^m = \{\sigma_1 \ldots, \sigma_m\}$, which are provided by a trainer or *teacher*. Using a *learning algorithm*, the network is adapted to this *training set* $D_m = (\sigma^m, \mathbf{x}^m)$ by adjusting its parameters $\mathbf{w}$, such that it responds correctly on the m examples.

How well will the trained network be able to classify an input that it has not seen before?

In order to give a quantitative answer to this question, a common model assumes that all inputs, those from the training set, and the new one, are produced independently at *random* with the *same* probability density from the network's environment. Fixing the training set for a moment, the *probability* that the network will make a *mistake* on the new input, defines the generalization error $\varepsilon(D_m)$. Its *average*, ε, over many realizations of the training set, as a function of the number of examples gives the so called *learning curve.* This will be the main quantity of our interest in the following.

Clearly, ε also depends on the specific algorithm that was used during the training. Thus, the calculation of ε requires the knowledge of the network weights generated by the learning process. In general, these weights will be complicated functions of the examples, and an explicit form will not be available in most cases.

The methods of statistical mechanics provide an approach to this problem, which often enables an *exact* calculation of learning curves in the limit of a very large network, i.e. for $N \to \infty$. It may seem surprising that a problem will simplify when the number of its parameters is increased. However, this phenomenon is well known for physical systems like gases or liquids which consist of a huge number of molecules. Clearly, there is no chance of estimating the complete *microscopic* state of the system, which is described by the rapidly fluctuating positions and velocities of all particles. On the other hand, the description of the *macroscopic* state of a gas requires only a few parameters like density, temperature and pressure. Such quantities can be calculated by suitably *averaging* over a whole ensemble of microscopic states that are compatible with macroscopic constraints.

Applying similar ideas to neural network learning, the problems which arise from specifying the details of a concrete learning algorithm can be avoided. In the statistical mechanics approach one studies the ensemble of *all* networks which implement the same set of input/output examples to a given accuracy. In this way the typical generalization behaviour of a neural network (in contrast to the worst or optimal behaviour) can be described.

2. THE PERCEPTRON

In this section I will explain this approach for one of the simplest types of networks, the *single layer perceptron* (see PERCEPTRONS, ADALINES, AND BACKPROPAGATION by Widrow & Lehr). A study of this network is not of purely academic interest, because the single layer architecture is a substructure of multilayer networks, and many of the steps in the subsequent calculations also appear in the analysis of more complex networks. Furthermore, by replacing the input vector $\mathbf{x}$ with a suitable vector of nonlinear features, the perceptron (equipped with a specific learning algorithm) becomes a *supportvector machine*, an extremely powerful learning device introduced by V. Vapnik and his collaborators (see SUPPORT VECTOR MACHINES by Schölkopf & Smola).

The adjustable parameters of the perceptron are the N weights $\mathbf{w} = (w(1), \ldots, w(N))$.

The output is a weighted sum

$$\sigma = F_{\mathbf{w}}(\mathbf{x}) = \mathrm{sign}\left(\sum_{i=1}^{N} w(i)\ x(i)\right) = \mathrm{sign}(\mathbf{w}\cdot\mathbf{x}) \tag{1}$$

of the input values. Since the length of $\mathbf{w}$ can be normalized without changing the performance, we choose $||\mathbf{w}||^2 = N$.

The input/output relation (1) has a simple geometric interpretation: Consider the *hyperplane* $\mathbf{w}\cdot\mathbf{x} = 0$ in the N–dimensional space of inputs. All inputs that are on the same side as $\mathbf{w}$ are mapped onto $+1$, those on the other side onto -1. Perceptrons realize *linearly separable* classification problems. In the following, we assume that the classification labels σ_k are generated by some other perceptron with weights $\mathbf{w}_t$, the "teacher" perceptron. A simple case of a student/teacher mismatch is discussed in section 5.

The geometric picture immediately gives us an expression for the generalization error. A misclassification of a new input $\mathbf{x}$ by a "student" perceptron $\mathbf{w}_s$ occurs only if $\mathbf{x}$ is between the separating planes defined by $\mathbf{w}_s$ and $\mathbf{w}_t$. If the inputs are drawn randomly from a spherical distribution, the generalization error is proportional to the angle between $\mathbf{w}_s$ and $\mathbf{w}_t$. We obtain

$$\varepsilon(D_m) = \frac{1}{\pi}\arccos\left(R\right), \tag{2}$$

where the "overlap" $R \doteq N^{-1}\mathbf{w}_s\cdot\mathbf{w}_t$ measures the similarity between student and teacher.

Following the pioneering work of Elizabeth Gardner (Gardner,1988), we assume that $\mathbf{w}_s$ was chosen *at random* with equal probability from all student perceptrons which are consistent with the training set, thereby avoiding the introduction of a concrete learning algorithm. In computational learning theory, the space of consistent vectors has been termed the *version space*. The corresponding probability density is $p(\mathbf{w}_s|D_m) = 1/V(D_m)$, if $\mathbf{w}_s$ is *in* the version space and $p(\mathbf{w}_s|D_m) = 0$ outside. $V(D_m)$ is the volume of the version space given by

$$V(D_m) = \int d\mathbf{w} \prod_{k=1}^{m} \Theta(\sigma_k \mathbf{w} \cdot \mathbf{x}_k). \tag{3}$$

Here, the Heaviside step function $\Theta(x)$ equals 1, if x is positive and zero else. Thus, only coupling vectors, for which the outputs σ_k are correct, i.e. $\sigma_k \mathbf{w}_s \cdot \mathbf{x}_k > 0$, contribute.

$V(D_m)$ is related to *Shannon's entropy* $\mathcal{S}$ of the distribution $p(\mathbf{w}|D_m)$ by $\mathcal{S} = -\int d\mathbf{w}\; p(\mathbf{w}|D_m) \ln p(\mathbf{w}|D_m) = \ln V(D_m)$. Similarly, in statistical mechanics the entropy measures the logarithm of the volume of microscopic states which are compatible with given values of macroscopic constraints, like the total energy of a system. In fact (see section 5), the constraint of perfect learning of all examples is equivalent to the condition of a minimal "training energy". The learning of an increasing number of examples reduces the set of consistent vectors $\mathbf{w}_s$ and leads to a decrease of the entropy $\mathcal{S}$, i.e. of our uncertainty about the unknown teacher $\mathbf{w}_t$. As we will see, by calculating the entropy one will get the generalization error ε for free.

3. ENTROPY AND REPLICA METHOD

Although $V(D_m)$ fluctuates with the random training set, for large N, the quantity $N^{-1}\mathcal{S}$ will be, with high probability, close to its *average* value $\mathcal{S}_{av}$. This results from the fact that the entropy is roughly additive in the degrees of freedom (which would be exactly true, if the components of the weight vector $\mathbf{w}$ were statistically independent). Hence, the fluctuations of $N^{-1}\mathcal{S}$ will be averaged out by the many additive random contributions. A similar argument applies to $R = N^{-1}\mathbf{w}_s \cdot \mathbf{w}_t$.

The calculation of $\mathcal{S}_{av}$ requires another tool of statistical physics, the *replica method.* It is based on the identity

$$\mathcal{S}_{av} = \langle\langle \ln\ V(D_m)\rangle\rangle = \lim_{n\to 0} n^{-1}(\langle\langle V^n(D_m)\rangle\rangle - 1)\ . \tag{4}$$

The brackets denote the average over the examples. Often, the average of $V^n(D_m)$, which is the volume of the version space of n perceptrons (replicas) $\mathbf{w}_a,\ a = 1, \ldots, n$, being trained on the same examples, can be calculated for *integers n.* At the end, an analytical continuation to real n is necessary. The calculation of the high–dimensional integral over weight vectors is enabled by two ideas: Since the labels are produced by the teacher perceptron $\mathbf{w}_t$, and the input distribution is *spherical*, the integrand will depend *only* on the angles between the vectors $\mathbf{w}_a,\ a = 1, \ldots, n$ and $\mathbf{w}_t$, ie. on the overlaps

$$q_{ab} = N^{-1}\mathbf{w}_a \cdot \mathbf{w}_b,\ a < b \tag{5}$$

$$R_a = N^{-1}\mathbf{w}_a \cdot \mathbf{w}_t.$$

The result can be written in the form

$$\langle\langle V^n(D_m)\rangle\rangle = \int \prod_a dR_a \prod_{a<b} dq_{ab} \exp[N\mathcal{G}(n,\{q_{ab}, R_a\})]. \tag{6}$$

The explicit form of $\mathcal{G}$ has been given in (Györgyi and Tishby,1990), see also (Engel and Van den Broeck, 1995).

The limit $N \to \infty$ (to get a nontrivial result, the number of examples are scaled like $m = \alpha N$, with α fixed) provides a second simplification: The integrals in (6) are dominated by values $R_a(n)$ and $q_{ab}(n)$, for which the exponent $\mathcal{G}(n, \{q_{ab}, R_a\})$ is maximal. Other values have an (in N) exponentially smaller weight. The continuation of these most probable values to noninteger $n \simeq 0$ is by far non-trivial: The symmetry of $\mathcal{G}(n, \{q_{ab}, R_a\})$ under permutation of indices a, b, suggests the *replica symmetric* ansatz $q_{ab}(n) = q(n)$ and $R_a(n) = R(n)$, which is correct for the present perceptron problem. A more complicated scheme called *replica symmetry breaking* (Mézard et al,1987) for continuing the matrices q_{ab} to noninteger dimensions can be necessary if the version space of the learning problem is e.g. disconnected or not convex (Monasson and Zecchina, 1995).

Within replica symmetry, the analytic continuation $R = R(n = 0)$ coincides with the average teacher–student overlap needed for the generalization error (2) and $q = q(n = 0)$ gives the average overlap between two random student vectors in the version space. The resulting learning curve ε as a function of the relative number of examples α, is shown as the solid line in Fig. 1. For a small size of the training set ($\alpha \to 0$), q and R are close to zero and $\varepsilon \approx \frac{1}{2}$, which is not better than random guessing. To ensure good generalization,

m, the size of the training set must significantly exceed N, the number of couplings. Finally, when the ratio $\alpha = \frac{m}{N}$ grows large, q and R approach 1, and the error decreases slowly to 0 like $\varepsilon \simeq 0.62\ \alpha^{-1}$.

The shrinking of the space of network couplings resembles a similar result obtained for the learning in attractor neural networks as presented in STATISTICAL MECHANICS OF LEARNING by Engel & Zippelius. For the latter case however, the output bits of the corresponding perceptron are completely random (given by the random patterns to be stored), instead of being defined by a teacher network. As the number of patterns grows, the volume of couplings decreases to zero already at a nonzero critical capacity $\alpha = 2$.

Sofar, we have discussed the *typical* generalization ability of a perceptron learning a linear separable rule. Is it possible to generalize faster, by using more sophisticated learning strategies? The answer is: Not much, if we are restricted to random training examples. Studies of the asymptotics for optimal Bayes classifiers (Opper and Haussler, 1995) yield a generic α^{-1} decay for broad classes of learning models. For linear separable rules and spherical input distribution the optimal decay is (Opper and Kinzel,1995, Watkin et al,1993) $\varepsilon \simeq 0.44\ \alpha^{-1}$, only slightly better than the typical error.

The situation changes if the learner is free to ask the teacher questions (Watkin et al,1993), i.e. if she can choose highly informative inputs. Then the decrease of the generalization error ε can be exponentially fast in α.

The approach of statistical mechanics is able to provide information about generalization abilities even in situations, where (at present) no efficient learning algorithms are known. Assume e.g., that the perceptron weights are constrained to binary values $w(j) \in \{+1, \ -1\}$. This may give a crude model for the effects of a finite weight precision in digital network implementations. For this binary perceptron, perfect learning is equivalent to a hard combinatorial optimization problem, which in worst case is believed to require a learning time that grows exponentially with N. Using the replica–method, the dotted learning curve in Fig. 1 is obtained. For sufficiently small α, the discreteness of the version space has only minor effects. However, since there is a minimal volume of the version space when only one coupling vector is left, the generalization error ε drops to zero at a finite value $\alpha_c = 1.24$. Remarkably, this transition is discontinuous. This means that for α slightly below α_c, the few coupling vectors $\mathbf{w}$ which are consistent with all examples typically differ in a finite fraction of bits from the teacher (and from each other).

4. PHASE TRANSITIONS AND SYMMETRY BREAKING

In statistical mechanics, the overlaps q and R are examples of *order parameters* which measure the degree of ordering of a system towards an external influence. In our case, the labels which are provided by the teacher make the typical student vector more and more aligned with the teacher. This is reflected by the increase of q and R from the minimal values 0 (when there are no examples) to their maximal values 1, when student and teacher align perfectly.

In statistical physics, a variety of systems show *phase transitions*, ie. ordering sets in *spontaneously* when a control parameter exceeds a critical value. Surprisingly, such phase transitions can also be observed for *multilayer networks*. I will illustrate such a behaviour for a simple toy multilayer network with a drastically simplified architecture, the *parity tree*. Here, the set of N input nodes is equally divided into K nonoverlapping groups, each of which is connected to one of K hidden units by N/K adjustable weights like in a perceptron. The output node computes the parity of the hidden outputs, i.e. a -1 output results from an odd number of negative hidden units and a $+1$ from an even number. For $K = 2$, we obtain the learning curve (Hansel et al, 1992) with the small dashes shown in Fig.1. Below a $\alpha = 1.2337$, the network perfectly memorizes the training examples without being able to generalize at all ($\varepsilon = \frac{1}{2}$). Above this critical value, generalization sets in and the generalization error decays smoothly. For $K > 2$, the transition is discontinuous (Engel and Van den Broeck, 2001).

This and similar phase transitions are related to *spontaneous symmetry breaking*: Although a statistical ensemble is invariant under a symmetry transformation, the typical, macroscopically observed state of the system may not show this symmetry. For the parity machine, the symmetry transformation is the inversion of the signs of the couplings corresponding to a pair of different hidden units. E.g., for $K = 2$, a network with couplings all equal to, say $+1$, implements exactly the same classification task as a network with all couplings equal to -1. For small α, the typical student networks trained by a "$+1$" teacher will reflect this ± symmetry. Their coupling vectors consist of an equal fraction of positive and negative weights without any preference for the teacher (or the reversed teacher) and generalization is impossible. If α exceeds the critical value, the symmetry is broken and we have *two* possible types of typical students, one with a majority of positive, the other one with a majority of negative couplings. Both types of the typical students display now some similarity with the teacher or its negative image and generalization sets in. A related type of *retarded generalization* can be found in models of *unsupervised learning*, where the learner has to infer an unknown symmetry axis of a high dimensional probability distribution (Engel and Van den Broeck, 2001). When the number of data is below a critical value, the estimated direction of the axis will be typically orthogonal to the true one.

Phase transitions for multilayer networks with fully connected architectures are related to the breaking of the permutation symmetry between hidden units (Schwarze and Hertz 1993, Engel and Van den Broeck, 2001). They can already be observed for *online learning* algorithms. Since the change of the weights depend on the most recently presented example only, and each example is seen only once, a detailed study of the dynamics of learning is possible (see STATISTICAL MECHANICS OF ON-LINE LEARNING AND GENERALIZATION by Biehl & Caticha).

5. ALGORITHMS AND OVERFITTING

The statistical mechanics approach can also be applied to the performance of concrete algorithms, if the algorithm minimizes a *training energy*, such as the quadratic deviation

$$E(\mathbf{w}_s|\sigma^m, \mathbf{x}^m) = \sum_{k=1}^{m} (\sigma_k - F_{\mathbf{w}_s}(\mathbf{x}_k))^2 \tag{7}$$

between the network's and the teacher's outputs (see PERCEPTRONS, ADALINES, AND BACKPROPAGATION by Widrow & Lehr).

The ensemble of students is now defined by all vectors $\mathbf{w}_s$ which achieve a certain accuracy in learning, i.e., which have a fixed training energy. We may as well fix the *average* energy, allowing for small fluctuations. Such fluctuations also occur for physical systems in thermal equilibrium, which can exchange energy with their environment. The *Gibbs–distribution*

$$p(\mathbf{w}_s|D_m) \propto e^{-\beta E(\mathbf{w}_s|D_m)} \tag{8}$$

provides the proper probability density for such a case, where the parameter β has to be adjusted such that the average energy achieves the desired value. In physics, $T = 1/\beta$ plays the role of the temperature. For the relation of this concept to the stochastic dynamics of networks, see STATISTICAL MECHANICS OF LEARNING by Engel & Zippelius.

For $\beta = \infty$, the distribution p is concentrated at the vector $\mathbf{w}_s$, for which E is minimal, corresponding to a learning algorithm which achieves the total minimum of the training energy. An application to the generalization performance of *supportvector machines* can be found in (Dietrich et al, 1999).

Let me briefly illustrate the results of this method for a single layer perceptron, where during the training phase, the student is replaced by a simple *linear* function

$$F_{\mathbf{w}_s}(\mathbf{x}) = \mathbf{w}_s \cdot \mathbf{x} \tag{9}$$

in (7). For a teacher of the same linear type, the classification rule is learnt completely with $m = N$ examples. A different behaviour occurs if the teacher is the *nonlinear* rule (1), and for generalization, also the student's output is given by (1). Although all examples are still perfectly learnt up to $\alpha = \frac{m}{N} = 1$, the generalization error increases to the random guessing value $\varepsilon = \frac{1}{2}$ (Fig.1, dashed line), a phenomenon termed *overfitting.*

If $m > N$, the minimal training error E is *greater than zero.* Nevertheless, ε decreases again and approaches 0 asymptotically for $\alpha \to \infty$. This shows that one can achieve good generalization with algorithms that allow for learning errors.

The introduction of a temperature is not only a formal trick. Stochastic learning with a nonzero temperature may be useful to escape from local minima of the training energy, enabling a better learning of the training set. Surprisingly, it can lead to *better generalization abilities* if the classification rule is not completely learnable by the net. In the simplest case, that happens when the rule contains a degree of noise (Györgyi and Tishby,1990, Opper and Kinzel,1995 and also SIMULATED ANNEALING AND BOLTZMANN MACHINES by Aarts & Korst).

The replica method enables us to study the properties of networks at their minimal training energy. However, a real learning algorithm may not reach such a state when local minima are present.

6. DISCUSSION

The statistical mechanics approach to learning and generalization allows to understand the typical generalization behaviour of large neural networks. Its major tool, the replica method, has been illustrated in this article for simple neural networks like the perceptron. Already for simple models, interesting and unexpected phenomena, like discontinuous and retarded generalization can be observed.

The statistical mechanics approach enables us to perform controlled analytical experiments on very large networks. In contrast to real experiments which will produce an enormous amount of microscopic data to be evaluated, this analytical method provides us with a small set of order parameters, which are directly interpretable in terms of the network's macroscopic performance. Concentrating on typical behaviour, the statistical mechanics analysis can complement other theoretical approaches to generalization, like the worst case PAC bounds, see (Engel and Van den Broeck,2001, Urbanczik,1996 and also LEARNING AND GENERALIZATION - THEORETICAL BOUNDS by Herbrich & Williamson).

Presently, the approach is both applied to novel learning concepts in neural computing, like support vector machines, as well as to a variety of other problems in information science, e.g. error correcting codes and hard combinatorial optimization problems (see e.g., Nishimori 2001). Siginificant progress has also been made in understanding *dynamical* properties of concrete algorithms, see (Heimel and Coolen,2001 and STATISTICAL MECHANCS OF ON-LINE LEARNING AND GENERALIZATION of Biehl & Caticha).

New and challenging applications are also found in other fields where learning and adaptation in large populations of entities plays a role, like in ecological systems and economical markets.

REFERENCES

Dietrich, R., Opper, M., and Sompolinsky, H., 1999, Statistical Mechanics of Support Vector Networks, Phys. Rev. Lett. 82: 2975.

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

Gardner, E., 1988, The space of interactions in neural network models, J. Phys. A: Math. Gen., 21:257-270.

Györgyi, G. and Tishby, N., 1990, Statistical Theory of Learning a Rule, in Neural Networks and Spin Glasses, (W. K. Theumann and R. Koeberle Eds.), Singapore: World Scientific, pp. 3-36.

Heimel, J.A.F., 2001 and Coolen, A.C.C., Supervised learning with restricted training sets: A generating functional analysis, J. Phys. A 34: 9009-9026.

Mézard, M., Parisi, G., and Virasoro, M. A., 1987, Spin Glass Theory and Beyond. Singapore: World Scientific.

Hansel, D., Mato, G., and Meunier, C., 1992, Memorization without generalization in a multilayered neural network, Europhys. Lett. 20: 471-476.

Monasson, R. and Zecchina, R., 1995, Weight Space Structure and Internal Representations: a direct Approach to Learning and Generalization in Multilayer Neural Networks Phys. Rev. Lett. 75: 2432.

* Nishimori, H., 2001, Statistical Physics of Spin Glasses and Information Processing New York: Oxford University Press.

Opper, M. and Kinzel, W., 1995, Statistical Mechanics of Generalization, in Physics of Neural Networks III, (J. L. van Hemmen, E. Domany and K. Schulten, Eds.), New York: Springer–Verlag.

Opper, M., and Haussler, D., 1995, Bounds for Predictive Errors in the Statistical Mechanics of Supervised Learning, Phys. Rev. Lett. 75: 3772.

Schwarze, H. and Hertz, J., 1993, Generalization in fully connected Committee Machines, Europhys. Lett. 21: 785.

Seung, H. S., Sompolinsky, H., and Tishby, N., 1992, Statistical mechanics of learning from examples. Phys. Rev. A: 45: 6056-6091.

T. L. H. Watkin, T. L. H., Rau, A., and Biehl, M., 1993, The statistical mechanics of learning a rule. Rev. Mod. Phys. 65: 499-556.

Urbanczik, R., 1996 Learning in a large Committee Machine: Worst Case and Average Case, Europhys. Lett. 35: 553.

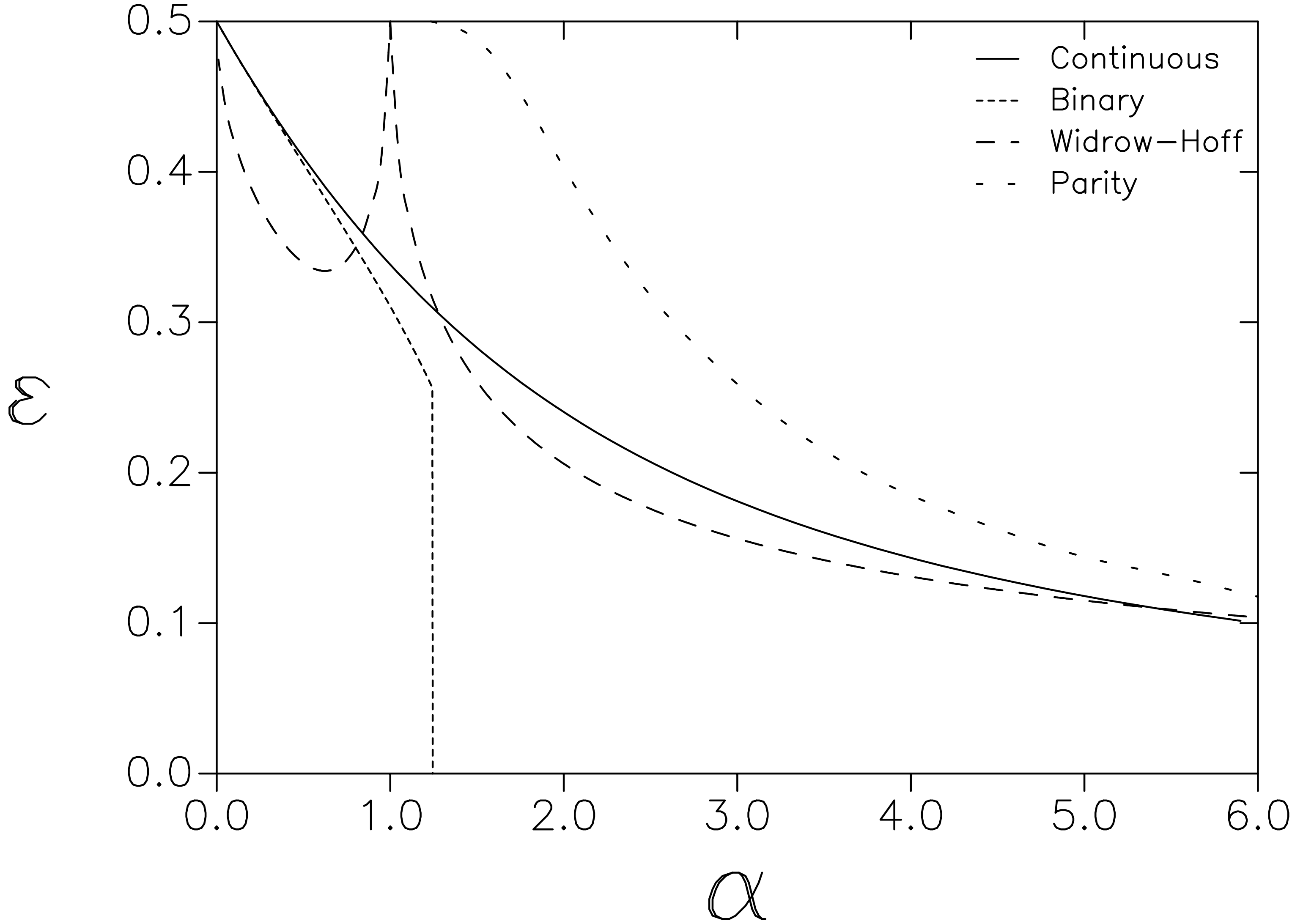

FIGURE CAPTIONS

Figure 1. Generalization errors ε for a typical continuous (solid curve) and a typical binary perceptron (dotted curve) as a function of the relative size $\alpha = \frac{m}{N}$ of the training set. For $\alpha = 1.24$, the generalization error drops discontinuously to zero. The dashed curve refers to the linear network (9). For $\alpha \approx 1$, the mismatch between the nonlinear teacher and the

linear student becomes apparent: Although all examples are perfectly learnt, generalization becomes impossible ($\varepsilon \approx \frac{1}{2}$). This overfitting phenomenon diappears for $\alpha > 1$, when the algorithm learns with training errors. The curve with small dashes shows the learning curve for a parity tree machine with two hidden units. Nontrivial generalization ($\varepsilon < \frac{1}{2}$) is impossible for $\alpha < 1.2337$.