Statistical Mechanics of Learning : Generalization

Manfred Opper

Physikalisches Institut

Julius Maximilians Universität

Würzburg, Germany

Short title: Statistical Mechanics of Generalization

Correspondence:

Manfred Opper

Physikalisches Institut, Julius Maximilians Universität, Am Hubland, D-97074 Würzburg

Phone:+49-931-888-5865

Fax:+49-931-888-5141

email: opper@physik.uni-wuerzburg.de

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 $\sigma = -1$. 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* $(\sigma^m, \mathbf{x}^m)$ 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 $\varepsilon(\sigma^m, \mathbf{x}^m)$. 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 $N \to \infty$. At first glance 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 consists 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. It was one of the major achievements of statistical mechanics to show that 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. It is believed that in this way the typical generalization behaviour of a neural network (in contrast to the worst or optimal behaviour) is described. From a less formal viewpoint, we may think that an ensemble is realized by a stochastic training algorithm.

2. THE PERCEPTRON

In this section I will explain this approach for one of the simplest types of networks, the *single layer perceptron* (Hertz et al,1991). This machine, for which a great variety of results has been obtained, is far from being a toy model. Since the single layer architecture is a substructure of multilayer networks, many of the steps in the subsequent calculations also appear in the analysis of more complex networks.

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 rule to be learnt belongs to this class so that it can be exactly implemented by a perceptron. We may think that the classification labels σ_k, which come from an ideal expert, are being generated by some other perceptron with weights $\mathbf{w}_t$, the "teacher" perceptron.

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

$$\varepsilon(\sigma^m, \mathbf{x}^m) = \frac{1}{\pi} \arccos\left(N^{-1}\mathbf{w}_s \cdot \mathbf{w}_t\right) \equiv \frac{1}{\pi} \arccos(R). \tag{2}$$

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

Following the pioneering work of Elizabeth Gardner (Gardner,1988), we will not concentrate on a specific $\mathbf{w}_s$, which is the result of a concrete learning algorithm. We will rather assume that $\mathbf{w}_s$ was chosen *at random* from the ensemble of all student perceptrons that are consistent with the training set. This space of consistent vectors is usually termed the *version space*. It turns out, that the generalization error for such a *typical* student can be

calculated from the volume V of the version space which is defined by

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

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.

The volume V is a measure for our uncertainty on the weights of the unknown teacher. In the language of statistical mechanics, such degree of uncertainty of the state of a physical system defines the *entropy* $\mathcal{S} = \ln V(\sigma^m, \mathbf{x}^m)$. The learning of an increasing number of labelled examples reduces the set of consistent vectors $\mathbf{w}_s$ and leads to a decrease of the volume (3). The decrease ΔS of the entropy equals the *amount of information* that is gained on the unknown teacher $\mathbf{w}_t$.

As we will see in the following, by calculating the entropy one will get the generalization error ε for free.

3. ENTROPY AND REPLICA METHOD

A priori, it is not clear how to get a useful estimate of the entropy $\mathcal{S}$: $V(\sigma^m, \mathbf{x}^m)$ is a *random variable* and fluctuates with the random training set, so an averaging process is necessary. As typical for a volume, V scales like $\simeq L^N$, where N is the dimension of the version space and L its typical diameter. Thus, for large N, its fluctuations will be over many orders of magnitude, and simply averaging V will put much weight on untypical events.

On the other hand, $\mathcal{S} = \ln(V) \simeq N \ln L$, grows linearly with N, which makes it plausible that the fluctuations of $N^{-1}\mathcal{S}$ will be averaged out by the additive, random contributions of very many degrees of freedom. The same argument applies to the overlap $R = N^{-1}\mathbf{w}_s \cdot \mathbf{w}_t$. We conclude that $N^{-1}\mathcal{S}$ and R are *self–averaging* quantities which can be safely replaced by their average values.

The calculation of the example averaged entropy is a nontrivial problem, which can be solved by a tool of statistical physics, the *replica method*. This is based on the identity

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

Here, the brackets denote the average over the examples. Often, the average of V^n, which is the phase space volume of the *n–fold replicated* system, can be calculated for *all integers* n. At the end of the calculation, an appropriate analytical continuation to real n is necessary. For integer n, we have

$$\langle\langle V^n \rangle\rangle = \int \prod_{a=1}^{n} d\mathbf{w}_a \exp\left[\alpha N \ln\left\langle\left\langle \prod_{a=1}^{n} \Theta(\sigma \mathbf{w}_a \cdot \mathbf{x})\right\rangle\right\rangle\right]. \tag{5}$$

Here, we have scaled the number of inputs like $m = \alpha N$, keeping α fixed to obtain a nontrivial limit for $N \to \infty$, and used the fact that all examples are *statistically independent* and identically distributed.

The calculation of the high–dimensional integrals in (5) is enabled by two ideas: The first utilizes the symmetry of the *spherical distribution*. Since the labels σ were produced by the teacher perceptron $\mathbf{w}_t$, the inner average in (5) will depend *only* on the $\frac{n(n+1)}{2}$ relative

angles between the vectors $\mathbf{w}_a,\ a = 1, \ldots, n$ and $\mathbf{w}_t$. Thus, $\ln\langle\langle \ldots \rangle\rangle = \mathcal{G}_1(n, \{q_{ab}, R_a\})$ is a function of the overlaps

$$q_{ab} = N^{-1}\mathbf{w}_a \cdot \mathbf{w}_b,\ a < b \tag{6}$$

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

By introducing the measure $e^{N\mathcal{G}_2(n,\{q_{ab},R_a\})}$, of all vectors $\mathbf{w}_a$ that are constrained to fixed overlaps $\{q_{ab}, R_a\}$, (5) can be converted into an expression that contains only integrations over R_a and q_{ab}

$$\langle\langle V^n \rangle\rangle = \int \prod_a dR_a \prod_{a<b} dq_{ab} \exp[N\mathcal{G}(n, \{q_{ab}, R_a\})], \tag{7}$$

with $\mathcal{G} = \alpha\mathcal{G}_1 + \mathcal{G}_2$. The explicit form of $\mathcal{G}$ has been given in (Györgyi and Tishby,1990).

The limit $N \to \infty$ provides a second simplification: The integrals in (7) 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.

It can be shown, that these most probable values can be continued to noninteger n and have limits $R = \lim_{n\to 0} R_a(n)$ and $q = \lim_{n\to 0} q_{ab}(n)$, where R coincides with the average teacher–student overlap needed for the generalization error (2). q gives the average overlap between two random student vectors in the version space. In statistical mechanics q and R are called *order parameters*. This name is justified when we look at the ordering of the student vectors, when more and more examples are learnt. When q and R are small, the student vectors typically point in arbitrary directions, showing little similarities with each

others and with the teacher. On the other hand, for a small version space, i.e. when many examples have been presented, all students closely resemble the teacher and q and R approach their maximal value 1.

The formal continuation of (7) to noninteger $n \simeq 0$ is by far non-trivial: The symmetry of $\mathcal{G}(n, \{q_{ab}, R_a\})$ under permutation of replica 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. However, a more complicated scheme for continuing the matrices q_{ab} to noninteger dimensions, which allows for a *replica symmetry breaking* (Mézard et al,1987), must be applied if the version space of the learning problem is sufficiently complex.

Within replica symmetry, R and q are obtained as solutions of the optimization problem

$$N^{-1}\mathcal{S}_{av} = extr_{\{q,R\}} \lim_{n\to 0} n^{-1}(\mathcal{G}(n, q, R) - 1). \tag{8}$$

This yields, from (2), the desired learning curve ε as a function of the relative number of examples α, which is shown in Fig. 1 (solid line). For a small size of the training set ($\alpha \to 0$), q and R are close to zero and the generalization error $\varepsilon \approx \frac{1}{2}$, which is not better than a random guessing of the output. 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 the contribution 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 finite 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 input examples. Using the replica approach, the learning curve of an optimal Bayes classifier (Opper and Kinzel,1995, Watkin et al,1993) for linear separable rules has been calculated, yielding $\varepsilon \simeq 0.44\ \alpha^{-1}$. Studies of multilayer networks indicate that the asymptotic α^{-1} decay is generic for networks with continuous weights and learnable problems.

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.

4. DISCONTINUOUS LEARNING

The method of the last section will even provide information about generalization abilities, when (at present) no efficient learning algorithm is known. This is the case, when the network weights are constrained to binary values $w(j) \epsilon \{+1, -1\}$. Such a choice may give a crude model for the effects of a finite weight precision in digital network implementations.

For a binary perceptron, perfect learning is equivalent to a hard combinatorial optimization problem (integer linear programming), 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.

The discreteness of couplings is not the only source of phase transitions in neural networks. Nonsmooth learning curves will occur for continuous weights in multilayer nets, if the architecture allows for symmetries, that can be spontaneously broken. This includes the permutation symmetry between different hidden units in the *committee machine*, or the inversion symmetry for a *parity machine*.

5. ALGORITHMS AND OVERFITTING

The statistical ensemble approach can in many cases also be applied to the performance of concrete algorithms, if the task of the algorithm is to minimize a *training energy*. A famous example for such a learning strategy is *backpropagation* (Hertz et al,1991), where the quadratic deviation

$$E(\mathbf{w}_s|\sigma^m, \mathbf{x}^m) = \sum_{k=1}^{m} (\sigma_k - F_{\mathbf{w}_s}(\mathbf{x}_k))^2 \tag{9}$$

between the network's and the teacher's outputs are minimized via a gradient descent of E.

Generalizing the method of the previous sections, we will abandon the assumption that the student network is able to implement the learning task perfectly. 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. For actual calculations, it is simpler to fix the *average* energy, allowing for small fluctuations, which can be neglected for a large network. Statistical mechanics provides a solution to this problem by a probability density p in the space of networks which weights each student according to

$$p(\mathbf{w}_s|\sigma^m, \mathbf{x}^m) \propto e^{-\beta E(\mathbf{w}_s|\sigma^m, \mathbf{x}^m)}. \tag{10}$$

The parameter β has to be adjusted such that the average energy achieves the desired value. This so called *Gibbs–distribution* has its origin in the the theory of physical systems which are in thermal equilibrium with their environment. There, $T = 1/\beta$ plays the role of the temperature.

Of special interest is the value $\beta = \infty$, i.e. zero temperature. Here the probability distribution p is entirely concentrated at the vector $\mathbf{w}_s$, for which E is minimal. This is the one which is produced by a learning algorithm that finds the total minimum of the training energy. Using the replica method in a suitable way, the generalization error can be calculated.

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

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

in (9). The backpropagation algorithm reduces then to the so called *Widrow–Hoff* (Hertz et al,1991) rule. For a teacher of the same linear type, the calssification rule is learnt completely with $m = N$ examples. A rather 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 into learning theory 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 randomness or noise (Györgyi and Tishby,1990, Opper and Kinzel,1995).

6. DISCUSSION

The statistical mechanics approach to learning allows to understand the typical generalization behaviour in large neural networks. Its major tool, the replica method, has been illustrated for the single layer perceptron. Already for this simple network, interesting phenomena, like discontinuous learning and overfitting can be observed. Since the field is rapidly developing, these topics are only a small selection of learning problems that have recently been attacked by statistical mechanics.

I would like to finish this contribution by mentioning a few of them: The more complex problem of learning in multilayer networks is of current interest. Besides the replica method, approximations like the annealed theory and the high temperature limit enable a calculation of the rich structured learning curves.

Also for perceptron learning, new problems, which aim to bring the theory closer to reality, have been investigated recently. These include time dependent rules, the influence of input distributions, intelligent pruning of network weigts and unsupervised learning.

Finally, a unification of the statistical physics methods with ideas coming from other

fields of research like Mathematical Statistics (see the contribution LEARNING AND GENERALIZATION by V. Vapnik) is a challenging problem.

A more detailed review of the techniques, models and further references can be found in the longer review articles and books of the following bibliography.

REFERENCES

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.

* Hertz, J. A., Krogh, A., and Palmer, R. G., 1991, Introduction to the theory of neural computation, Redwood City: Addison-Wesley.

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

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

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.

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 version space of the binary perceptron consists of the teacher only and the generalization error drops discontinuously to zero. The dashed curve refers to a perceptron trained with the Widrow–Hoff algorithm. 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 Widrow algorithm learns with training errors.