MANFRED OPPER

Neural Computation Research Group
Aston University
Birmingham B4 7ET, United Kingdom

Theories that try to understand the ability of neural networks to generalize from learned examples are discussed. Also, an approach that is based on ideas from statistical physics which aims to model typical learning behavior is compared with a worst-case framework.

Learning to Generalize

Introduction

Neural networks learn from examples. This statement is obviously true for the brain, but also artificial networks (or neural networks), which have become a powerful new tool for many pattern-recognition problems, adapt their "synaptic" couplings to a set of examples. Neural nets usually consist of many simple computing units which are combined in an architecture which is often independent from the problem. The parameters which control the interaction among the units can be changed during the learning phase and these are often called *synaptic couplings.* After the learning phase, a network adopts some ability to generalize from the examples; it can make predictions about inputs which it has not seen before; it has begun to understand a rule. To what extent is it possible to understand the complexity of learning from examples by mathematical models and their solutions? This question is the focus of this article. I concentrate on the use of neural networks for classification. Here, one can take characteristic features (e.g., the pixels of an image) as an input pattern to the network. In the simplest case, it should decide whether a given pattern belongs (at least more likely) to a certain class of objects and respond with the output $+1$ or -1. To learn the underlying classification rule, the network is trained on a set of patterns together with the classification labels, which are provided by a trainer. A heuristic strategy for training is to tune the parameters of the machine (the couplings of the network) using a learning algorithm, in such a way that the errors made on the set of training examples are small, in

the hope that this helps to reduce the errors on new data. How well will the trained network be able to classify an input that it has not seen before? This performance on new data defines the generalization ability of the network. This ability will be affected by the problem of realizability: The network may not be sufficiently complex to learn the rule completely or there may be ambiguities in classification. Here, I concentrate on a second problem arising from the fact that learning will mostly not be exhaustive and the information about the rule contained in the examples is not complete. Hence, the performance of a network may vary from one training set to another. In order to treat the generalization ability in a quantitative way, a common model assumes that all input patterns, those from the training set and the new one on which the network is tested, have a preassigned probability distribution (which characterizes the feature that must be classified), and they are produced independently at random with the same probability distribution from the network's environment. Sometimes the probability distribution used to extract the examples and the classification of these examples is called the *rule.* The network's performance on novel data can now be quantified by the so-called *generalization error,* which is the probability of misclassifying the test input and can be measured by repeating the same learning experiment many times with different data.

Within such a probabilistic framework, neural networks are often viewed as statistical adaptive models which should give a likely explanation of the observed data. In this framework, the learning process becomes mathematically related to a statistical estimation problem for optimal network parameters. Hence, mathematical statistics seems to be a most appropriate candidate for studying a neural network's behavior. In fact, various statistical approaches have been applied to quantify the generalization performance. For example, expressions for the generalization error have been obtained in the limit, where the number of examples is large compared to the number of couplings (Seung *et al.,* 1992; Amari and Murata, 1993). In such a case, one can expect that learning is almost exhaustive, such that the statistical fluctuations of the parameters around their optimal values are small. However, in practice the number of parameters is often large so that the network can be flexible, and it is not clear how many examples are needed for the asymptotic theory to become valid. The asymptotic theory may actually miss interesting behavior of the so-called *learning curve,* which displays the progress of generalization ability with an increasing amount of training data.

A second important approach, which was introduced into mathematical statistics in the 1970s by Vapnik and Chervonenkis (VC) (Vapnik, 1982, 1995), provides exact bounds for the generalization error which are valid for any number of training examples. Moreover, they are entirely independent of the underlying distribution of inputs, and for the case of realizable rules they are also independent of the specific algorithm, as long as the training examples are perfectly learned. Because it is able to cover even bad situations which are unfavorable for improvement of the learning process, it is not surprising that this theory may in some cases provide too pessimistic results which are also too crude to reveal interesting behavior in the intermediate region of the learning curve.

In this article, I concentrate mainly on a different approach, which has its origin in statistical physics rather than in mathematical statistics, and compare its results with the worst-case results. This method aims at studying the typical rather than the worst-case behavior and often enables the exact calculations of the entire learning curve for models of simple networks which have many parameters. Since both biological and artificial neural networks are composed of many elements, it is hoped that such an approach may actually reveal some relevant and interesting structures.

At first, it may seem surprising that a problem should simplify when the number of its constituents becomes large. However, this phenomenon is well-known for macroscopic physical systems such as gases or liquids which consist of a huge number of molecules. Clearly, it is not possible to study the complete microscopic state of such a system, which is described by the rapidly fluctuating positions and velocities of all particles. On the other hand, macroscopic quantities such as density, temperature, and pressure are usually collective properties influenced by all elements. For such quantities, fluctuations are averaged out in the thermodynamic limit of a large number of particles and the collective properties become, to some extent, independent of the microstate. Similarly, the generalization ability of a neural network is a collective property of all the network parameters, and the techniques of statistical physics allow, at least for some simple but nontrivial models, for exact computations in the thermodynamic limit. Before explaining these ideas in detail, I provide a short description of feedforward neural networks.

Artificial Neural Networks

Based on highly idealized models of brain function, artificial neural networks are built from simple elementary computing units, which are sometimes termed neurons after their biological counterparts. Although hardware implementations have become an important research topic, neural nets are still simulated mostly on standard computers. Each computing unit of a neural net has a single output and several ingoing connections which receive the outputs of other units. To every ingoing connection (labeled by the index i) a real number is assigned, the synaptic weight w_i, which is the basic adjustable parameter of the network. To compute a unit's output, all incoming values x_i are multi-

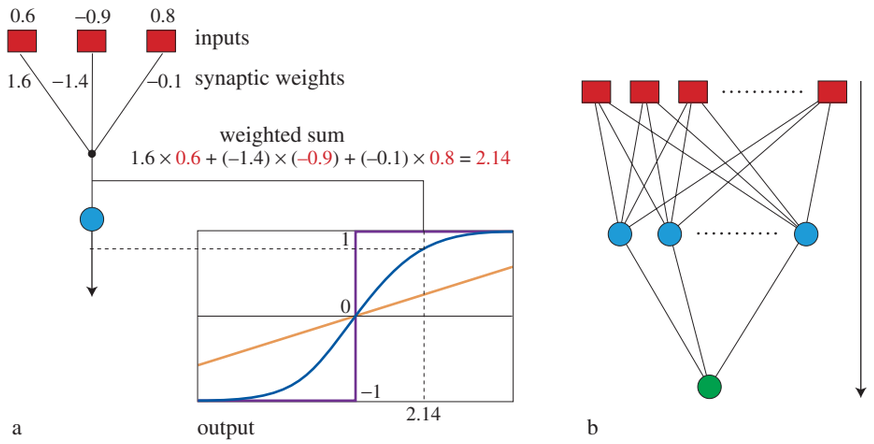

FIGURE 1 (a) Example of the computation of an elementary unit (neuron) in a neural network. The numerical values assumed by the incoming inputs to the neuron and the weights of the synapses by which the inputs reach the neuron are indicated. The weighted sum of the inputs corresponds to the value of the abscissa at which the value of the activation function is calculated (bottom graph). Three functions are shown: sigmoid, linear, and step. (b) Scheme of a feedforward network. The arrow indicates the direction of propagation of information.

plied by the weights w_i and then added. Figure 1a shows an example of such a computation with three couplings. Finally, the result, $\Sigma_i\, w_i x_i$, is passed through an activation function which is typically of the shape of the red curve in Fig. 1a (a *sigmoidal function*), which allows for a soft, ambiguous classification between -1 and $+1$. Other important cases are the step function (green curve) and the linear function (yellow curve; used in the output neuron for problems of fitting continuous functions). In the following, to keep matters simple, I restrict the discussion mainly to the step function. Such simple units can develop a remarkable computational power when connected in a suitable architecture. An important network type is the feedforward architecture shown in Fig. 1b, which has two layers of computing units and adjustable couplings. The input nodes (which do not compute) are coupled to the so-called hidden units, which feed their outputs into one or more output units. With such an architecture and sigmoidal activation functions, any continuous function of the inputs can be arbitrarily closely approximated when the number of hidden units is sufficiently large.

The Perceptron

The simplest type of network is the perceptron (Fig. 2a). There are N inputs, N synaptic couplings w_i, and the output is simply

$$\sum_{i=1}^{N} w_i x_i \qquad [1]$$

It has a single-layer architecture and the step function (green curve in Fig. 1a) as its activation function. Despite its simple structure, it can for many learning problems give a nontrivial generalization performance and may be used as a first step to an unknown classification task. As can be seen by comparing Figs. 2a and 1b, it is also a building block for the more complex multilayer networks. Hence, understanding its performance theoretically may also provide insight into the more complex machines. To learn a set of examples, a network must adjust its couplings appropriately (I often use the word couplings for their numerical strengths, the weights w_i, for $i = 1, \ldots, N$). Remarkably, for the perceptron there exists a simple learning algorithm which always enables the network to find those parameter values whenever the examples can be learnt by a perceptron. In Rosenblatt's algorithm, the input patterns are presented sequentially (e.g., in cycles) to the network and the

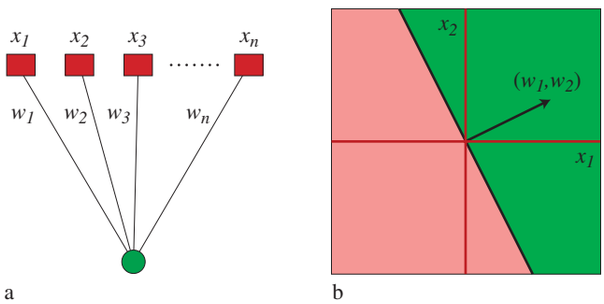

FIGURE 2 (a) The perceptron. (b) Classification of inputs by a perceptron with two inputs. The arrow indicates the vector composed of the weights of the network, and the line perpendicular to this vector is the boundary between the classes of input.

output is tested. Whenever a pattern is not classified correctly, all couplings are altered simultaneously. We increase by a fixed amount all weights for which the input unit and the correct value of the output neuron have the same sign but we decrease them for the opposite sign. This simple algorithm is reminiscent of the so-called *Hebbian learning rule,* a physiological model of a learning processes in the real brain. It assumes that synaptic weights are increased when two neurons are simultaneously active. Rosenblatt's theorem states that in cases in which there exists a choice of the w_i which classify correctly all of the examples (i.e., perfectly learnable perceptron), this algorithm finds a solution in a finite number of steps, which is at worst equal to $A\,N^3$, where A is an appropriate constant.

It is often useful to obtain an intuition of a perceptron's classification performance by thinking in terms of a geometric picture. We may view the numerical values of the inputs as the coordinates of a point in some (usually) high-dimensional space. The case of two dimensions is shown in Fig. 2b. A corresponding point is also constructed for the couplings w_i. The arrow which points from the origin of the coordinate system to this latter point is called the weight vector or coupling vector. An application of linear algebra to the computation of the network shows that the line which is perpendicular to the coupling vector is the boundary between inputs belonging to the two different classes. Input points which are on the same side as the coupling vector are classified as $+1$ (the green region in Fig. 2b) and those on the other side as -1 (red region in Fig. 2b).

Rosenblatt's algorithm aims to determine such a line when it is possible. This picture generalizes to higher dimensions, for which a hyperplane plays the same role of the line of the previous two-dimensional example. We can still obtain an intuitive picture by projecting on two-dimensional planes. In Fig. 3a, 200 input patterns with random coordinates (randomly labeled red and blue) in a 200-dimensional input space are projected on the plane spanned by two arbitrary coordinate axes. If we instead use a plane for projection which contains the coupling vector (determined from a variant of Rosenblatt's algorithm) we obtain the view shown in Fig. 3b, in which red and green points are clearly separated and there is even a gap between the two clouds.

It is evident that there are cases in which the two sets of points are too mixed and there is no line in two dimensions (or no hyperplane in higher dimensions which separates them). In these cases, the rule is too complex to be perfectly learned by a perceptron. If this happens, we must attempt to determine the choice of the coupling which minimizes the number of errors on a given set of examples. Here, Rosenblatt's algorithm does not work and the problem of finding the minimum is much more difficult from the algorithmic point. The training error, which is the number of errors made on the training set, is usually a nonsmooth function of the network couplings (i.e., it may have large variations for small changes of the couplings). Hence, in general, in addition to the perfectly learnable perceptron case in which the final error is zero, minimizing the training error is usually a difficult task which could take a large amount of computer time. However, in practice, iterative approaches, which are based on the minimization of other smooth cost functions, are used to train a neural network (Bishop, 1995).

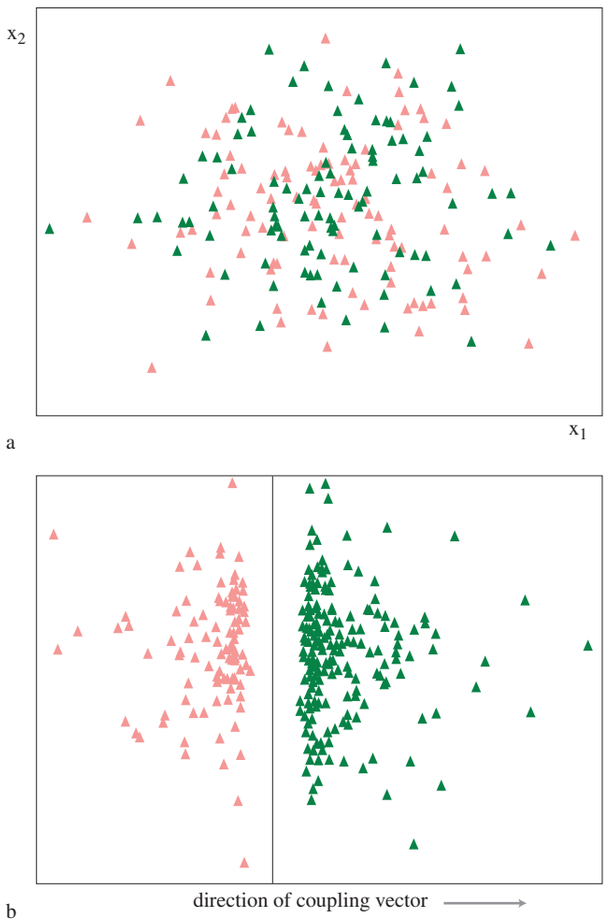

FIGURE 3 (a) Projection of 200 random points (with random labels) from a 200-dimensional space onto the first two coordinate axes (x_1 and x_2). (b) Projection of the same points onto a plane which contains the coupling vector of a perfectly trained perceptron.

Capacity, VC Dimension, and Worst-Case Generalization

As previously shown, perceptrons are only able to realize a very restricted type of classification rules, the so-called linearly separable ones. Hence, independently from the issue of finding the best algorithm to learn the rule, one may ask

the following question: In how many cases will the perceptron be able to learn a given set of training examples perfectly if the output labels are chosen arbitrarily? In order to answer this question in a quantitative way, it is convenient to introduce some concepts such as capacity, VC dimension, and worst-case generalization, which can be used in the case of the perceptron and have a more general meaning.

In the case of perceptrons, this question was answered in the 1960s by Cover (1965). He calculated for any set of input patterns, e.g., m, the fraction of all the 2^m possible mappings that can be linearly separated and are thus learnable by perceptrons. This fraction is shown in Fig. 4 as a function of the number of examples per coupling for different numbers of input nodes (couplings) N. Three regions can be distinguished:

Region in which $m/N \leq 1$: Simple linear algebra shows that it is always possible to learn all mappings when the number m of input patterns is less than or equal to the number N of couplings (there are simply enough adjustable parameters).

Region in which $m/N > 1$: For this region, there are examples of rules that cannot be learned. However, when the number of examples is less than twice the number of couplings ($m/N < 2$), if the network is large enough almost all mappings can be learned. If the output labels for each of the m inputs are chosen randomly $+1$ or -1 with equal probability, the probability of finding a nonrealizable coupling goes to zero exponentially when N goes to infinity at fixed ratio m/N.

Region in which $m/N > 2$: For $m/N > 2$ the probability for a mapping to be realizable by perceptrons decreases to zero rapidly and it goes to zero exponentially when N goes to infinity at fixed ratio m/N (it is proportional to $\exp[-Nf(m/N)]$, where the function $f(\alpha)$ vanishes for $\alpha < 2$ and it is positive for $\alpha > 2$. Such a threshold phenomenon is an example of a phase transition (i.e., a sharp change of behavior) which can occur in the thermodynamic limit of a large network size.

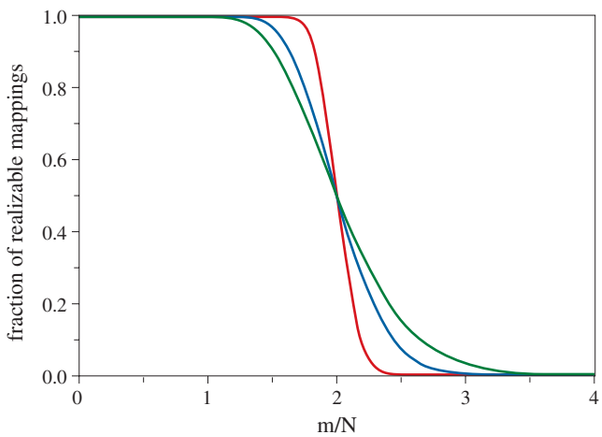

FIGURE 4 Fraction of all mappings of m input patterns which are learnable by perceptrons as a function of m/N for different numbers of couplings N: $N = 10$ (in green), $N = 20$ (in blue), and $N = 100$ (in red).

Generally, the point at which such a transition takes place defines the so-called capacity of the neural network. Although the capacity measures the ability of a network to learn random mappings of the inputs, it is also related to its ability to learn a rule (i.e., to generalize from examples). The question now is, how does the network perform on a new example after having been trained to learn m example on the training set?

To obtain an intuitive idea of the connection between capacity and ability to generalize, we assume a training set of size m and a single pattern for test. Suppose we define a possible rule by an arbitrary learnable mapping from inputs to outputs. If $m + 1$ is much larger than the capacity, then for most rules the labels on the m training patterns which the perceptron is able to recognize will nearly uniquely determine the couplings (and consequently the answer of the learning algorithm on the test pattern), and the rule can be perfectly understood from the examples. Below capacity, in most cases there are two different choices of couplings which give opposite answers for the test pattern. Hence, a correct classification will occur with probability 0.5 assuming all rules to be equally probable. Figure 5 displays the two types of situations for $m = 3$ and $N = 2$.

This intuitive connection can be sharpened. Vapnik and Chervonenkis established a relation between a capacity such as quantity and the generalization ability that is valid for general classifiers (Vapnik, 1982, 1995). The VC dimension is defined as the size of the largest set of inputs for which all mappings can be learned by the type of classifier. It equals N for the perceptron. Vapnik and Chervonenkis were able to show that for any training set of size m

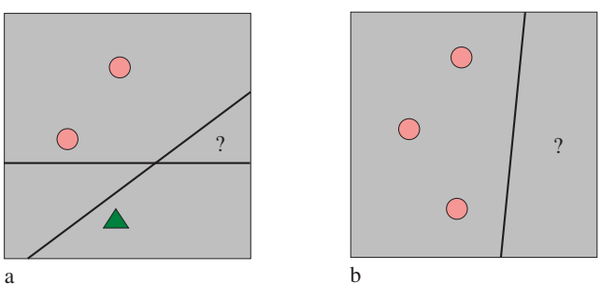

FIGURE 5 Classification rules for four patterns based on a perceptron. The patterns colored in red represent the training examples, and triangles and circles represent different class labels. The question mark is a test pattern. (a) There are two possible ways of classifying the test point consistent with the examples; (b) only one classification is possible.

larger than the VC dimension D_{VC}, the growth of the number of realizable mappings is bounded by an expression which grows much slower than 2^m (in fact, only like a polynomial in m).

They proved that a large difference between training error (i.e., the minimum percentage of errors that is done on the training set) and generalization error (i.e., the probability of producing an error on the test pattern after having learned the examples) of classifiers is highly improbable if the number of examples is well above D_{VC}. This theorem implies a small expected generalization error for perfect learning of the training set results. The expected generalization error is bounded by a quantity which increases proportionally to D_{VC} and decreases (neglecting logarithmic corrections in m) inversely proportional to m.

Conversely, one can construct a worst-case distribution of input patterns, for which a size of the training set larger than D_{VC} is also necessary for good generalization. The VC results should, in practice, enable us to select the network with the proper complexity which guarantees the smallest bound on the generalization error. For example, in order to find the proper size of the hidden layer of a network with two layers, one could train networks of different sizes on the same data.

The relation among these concepts can be better understood if we consider a family of networks of increasing complexity which have to learn the same rule. A qualitative picture of the results is shown in Fig. 6. As indicated by the blue curve in Fig. 6, the minimal training error will decrease for increasing complexity of the nets. On the other hand, the VC dimension and the complexity of the networks increase with the increasing number of hidden units, leading to an increasing expected difference (confidence interval) between training error and generalization error as indicated by the red curve. The sum of both (green curve) will have a minimum, giving the smallest bound on the generalization error. As discussed later, this procedure will in some cases lead to not very realistic estimates by the rather pessimistic bounds of the theory. In other words, the rigorous bounds, which are obtained from an arbitrary network and rule, are much larger than those determined from the results for most of the networks and rules.

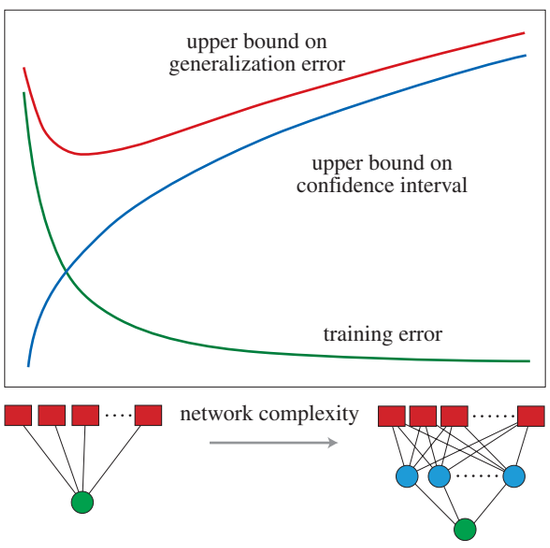

FIGURE 6 As the complexity of the network varies (i.e., of the number of hidden units, as shown schematically below), the generalization error (in red), calculated from the sum of the training error (in green) and the confidence interval (in blue) according to the theory of Vapnik–Chervonenkis, shows a minimum; this corresponds to the network with the best generalization ability.

Typical Scenario: The Approach of Statistical Physics

When the number of examples is comparable to the size of the network, which for a perceptron equals the VC dimension, the VC theory states that one can construct malicious situations which prevent generalizations. However, in general, we would not expect that the world acts as an adversary. Therefore, how should one model a typical situation? As a first step, one may construct rules and pattern distributions which act together in a nonadversarial way. The teacher–student paradigm has proven to be useful in such a situation. Here, the rule to be learned is modeled by a second network, the teacher network; in this case, if the teacher and the student have the same architecture and the same number of units, the rule is evidently realizable. The correct class labels for any inputs are given by the outputs of the teacher. Within this framework, it is often possible to obtain simple expressions for the generalization error. For a perceptron, we can use the geometric picture to visualize the generalization error. A misclassification of a new input vector by a student perceptron with coupling vector ST occurs only if the input pattern is between the separating planes (dashed region in Fig. 7) defined by ST and the vector of teacher couplings TE. If the inputs are drawn randomly from a uniform distribution, the generalization error is directly proportional to the angle between ST and TE. Hence, the generalization error is small when teacher and student vectors are close together and decreases to zero when both coincide.

In the limit, when the number of examples is very large all the students which learn the training examples perfectly will not differ very much from and their couplings will be close to those of the teacher. Such cases with a small generalization error have been successfully treated by asymptotic methods of statistics. On the other hand, when the number of examples is relatively small, there are many different students which are consistent with the teacher regarding the training examples, and the uncertainty about

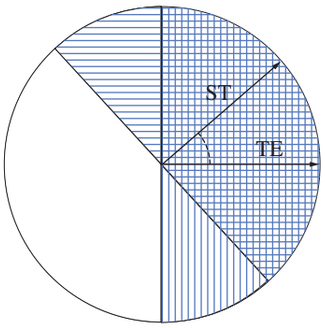

FIGURE 7 For a uniform distribution of patterns, the generalization error of a perceptron equals the area of the shaded region divided by the area of the entire circle. ST and TE represent the coupling vectors of the student and teacher, respectively.

the true couplings of the teacher is large. Possible generalization errors may range from zero (if, by chance, a learning algorithm converges to the teacher) to some worst-case value. We may say that the constraint which specifies the macrostate of the network (its training error) does not specify the microstate uniquely. Nevertheless, it makes sense to speak of a typical value for the generalization error, which is defined as the value which is realized by the majority of the students. In the thermodynamic limit known from statistical physics, in which the number of parameters of the network is taken to be large, we expect that in fact almost all students belong to this majority, provided the quantity of interest is a cooperative effect of all components of the system. As the geometric visualization for the generalization error of the perceptron shows, this is actually the case. The following approach, which was pioneered by Elizabeth Gardner (Gardner, 1988; Gardner and Derrida, 1989), is based on the calculation of $V(\varepsilon)$, the volume of the space of couplings which both perfectly implement m training examples and have a given generalization error ε. For an intuitive picture, consider that only discrete values for the couplings are allowed; then $V(\varepsilon)$ would be proportional to the number of students. The typical value of the generalization error is the value of ε, which maximizes $V(\varepsilon)$. It should be kept in mind that $V(\varepsilon)$ is a random number and fluctuates from one training set to another. A correct treatment of this randomness requires involved mathematical techniques (Mézard *et al.,* 1987). To obtain a picture which is quite often qualitatively correct, we may replace it by its average over many realizations of training sets. From elementary probability theory we see that this average number can be found by calculating the volume A of the space of all students with generalization error ε, irrespective of their behavior on the training set, and multiplying it by the probability B that a student with generalization error ε gives m times the correct answers on independent drawings of the input patterns. Since A increases exponentially with the number of couplings N (like typical volumes in N-dimensional spaces) and B decreases exponentially with m (because it becomes more improbable to be correct m times for any $\varepsilon \neq 0$), both factors can balance each other when m increases like $m = \alpha N$. α is an effective measure for the size of the training set when N goes to infinity. In order to have quantities which remain finite as $N \to \infty$, it is also useful to take the logarithm of $V(\varepsilon)$ and divide by N, which transforms the product into a sum of two terms. The first one (which is often called the *entropic term*) increases with increasing generalization error (green curve in Fig. 8). This is true because there are many networks which are not similar to the teacher, but there is only one network equal to the teacher. For almost all networks (remember, the entropic term does not include the effect of the training examples) $\varepsilon = 0.5$, i.e., they are correct half of the time by random guessing. On the other hand, the second term (red curve in Fig. 8) decreases with increasing generalization error because the probability of being correct on an input pattern increases when the student network becomes more similar to the teacher. It is often called the *energetic contribution* because it favors highly ordered (toward the teacher) network states, reminiscent of the states of physical systems at low energies. Hence, there will be a maximum (Fig. 8, arrow) of $V(\varepsilon)$ at some value of ε which by definition is the typical generalization error.

The development of the learning process as the number of examples αN increases can be understood as a competition between the entropic term, which favors disordered network configurations that are not similar to the teacher, and the energetic term. The latter term dominates when the number of examples is large. It will later be shown that such a competition can lead to a rich and interesting behavior as the number of examples is varied. The result for the learning curve (Györgyi and Tishby, 1990; Sompolinsky *et al.,*

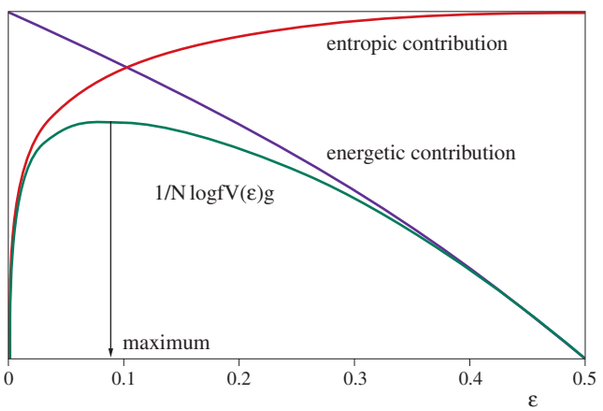

FIGURE 8 Logarithm of the average volume of students that have learned m examples and give ε generalization error (green curve). The blue and red curves represent the energetic and entropic contributions, respectively.

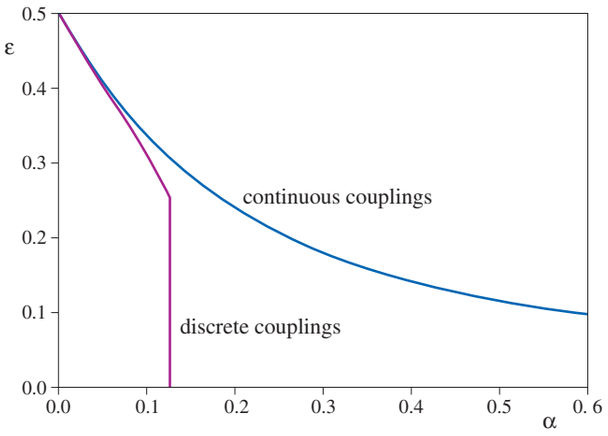

FIGURE 9 Learning curves for typical student perceptrons. $\alpha = m/N$ is the ratio between the number of examples and the coupling number.

1990) of a perceptron obtained by the statistical physics approach (treating the random sampling the proper way) is shown by the red curve of Fig. 9. In contrast to the worst-case predictions of the VC theory, it is possible to have some generalization ability below VC dimension or capacity. As we might have expected, the generalization error decreases monotonically, showing that the more that is learned, the more that is understood. Asymptotically, the error is proportional to N and inversely proportional to m, in agreement with the VC predictions. This may not be true for more complicated networks.

Query Learning

Soon after Gardner's pioneering work, it was realized that the approach of statistical physics is closely related to ideas in information theory and Bayesian statistics (Levin *et al.,* 1989; Györgyi and Tishby, 1990; Opper and Haussler, 1991), for which the reduction of an initial uncertainty about the true state of a system (teacher) by observing data is a central topic of interest. The logarithm of the volume of relevant microstates as defined in the previous section is a direct measure for such uncertainty. The moderate progress in generalization ability displayed by the red learning curve of Fig. 9 can be understood by the fact that as learning progresses less information about the teacher is gained from a new random example. Here, the information gain is defined as the reduction of the uncertainty when a new example is learned. The decrease in information gain is due to the increase in the generalization performance. This is plausible because inputs for which the majority of student networks give the correct answer are less informative than those for which a mistake is more likely. The situation changes if the student is free to ask the teacher questions, i.e., if the student can choose highly informative input patterns. For the simple perceptron a fruitful query strategy is to select a new input vector which is perpendicular to the current coupling vector of the student (Kinzel and Ruján, 1990). Such an input is a highly ambiguous pattern because small changes in the student couplings produce different classification answers. For more complicated networks it may be difficult to obtain similar ambiguous inputs by an explicit construction. A general algorithm has been proposed (Seung *et al.,* 1992a) which uses the principle of maximal disagreement in a committee of several students as a selection process for training patterns. Using an appropriate randomized training strategy, different students are generated which all learn the same set of examples. Next, any new input vector is only accepted for training when the disagreement of its classification between the students is maximal. For a committee of two students it can be shown that when the number of examples is large, the information gain does not decrease but reaches a positive constant. This results in a much faster decrease of the generalization error. Instead of being inversely proportional to the number of examples, the decrease is now exponentially fast.

Bad Students and Good Students

Although the typical student perceptron has a smooth, monotonically decreasing learning curve, the possibility that some concrete learning algorithm may result in a set of student couplings which are untypical in the sense of our theory cannot be ruled out. For bad students, even nonmonotic generalization behavior is possible. The problem of a concrete learning algorithm can be made to fit into the statistical physics framework if the algorithm minimizes a certain cost function. Treating the achieved values of the new cost function as a macroscopic constraint, the tools of statistical physics apply again.

As an example, it is convenient to consider a case in which the teacher and the student have a different architecture: In one of the simplest examples one tries to learn a classification problem by interpreting it as a regression problem, i.e., a problem of fitting a continuous function through data points. To be specific, we study the situation in which the teacher network is still given by a perceptron which computes binary valued outputs of the form $y = \Sigma_i w_i x_i, = \pm 1$, but as the student we choose a network with a linear transfer function (the yellow curve in Fig. 1a)

$$Y = \sum_i w_i x_i$$

and try to fit this linear expression to the binary labels of the teacher. If the number of couplings is sufficiently large (larger than the number of examples) the linear function

(unlike the sign) is perfectly able to fit arbitrary continuous output values. This linear fit is an attempt to explain the data in a more complicated way than necessary, and the couplings have to be finely tuned in order to achieve this goal. We find that the student trained in such a way does not generalize well (Opper and Kinzel, 1995). In order to compare the classifications of teacher and student on a new random input after training, we have finally converted the student's output into a classification label by taking the sign of its output. As shown in the red curve of Fig. 10, after an initial improvement of performance the generalization error increases again to the random guessing value $\varepsilon = 0.5$ at $\alpha = 1$ (Fig. 10, red curve). This phenomenon is called *overfitting*. For $\alpha > 1$ (i.e., for more data than parameters), it is no longer possible to have a perfect linear fit through the data, but a fit with a minimal deviation from a linear function leads to the second part of the learning curve. ε decreases again and approaches 0 asymptotically for $\alpha \to \infty$. This shows that when enough data are available, the details of the training algorithm are less important.

The dependence of the generalization performance on the complexity of the assumed data model is well-known. If function class is used that is too complex, data values can be perfectly fitted but the predicted function will be very sensitive to the variations of the data sample, leading to very unreliable predictions on novel inputs. On the other hand, functions that are too simple make the best fit almost insensitive to the data, which prevents us from learning enough from them.

It is also possible to calculate the worst-case generalization ability of perceptron students learning from a perceptron teacher. The largest generalization error is obtained (Fig. 7) when the angle between the coupling vectors of teacher and student is maximized under the constraint that the student learns all examples perfectly. Although it may not be easy to construct a learning algorithm which performs such a maximization in practice, the resulting generalization error can be calculated using the statistical physics approach (Engel and Van den Broeck, 1993). The result is in agreement with the VC theory: There is no prediction better than random guessing below the capacity.

Although the previous algorithms led to a behavior which is worse than the typical one, we now examine the opposite case of an algorithm which does better. Since the generalization ability of a neural network is related to the fact that similar input vectors are mapped onto the same output, one can assume that such a property can be enhanced if the separating gap between the two classes is maximized, which defines a new cost function for an algorithm. This optimal margin perceptron can be practically realized and when applied to a set of data leads to the projection of Fig. 11. As a remarkable result, it can be seen that there is a relatively large fraction of patterns which are located at the gap. These points are called *support vectors* (SVs). In order to understand their importance for the generalization ability, we make the following *gedankenexperiment* and assume that all the points which lie outside the gap (the nonsupport vectors) are eliminated from the training set of examples.

From the two-dimensional projection of Fig. 11, we may conjecture that by running the maximal margin algorithm on the remaining examples (the SVs) we cannot create a larger gap between the points. Hence, the algorithm will converge to the same separating hyperplane as before. This intuitive picture is actually correct. If the SVs of a training set were known beforehand (unfortunately, they are only identified after running the algorithm), the margin classifier would have to be trained only on the SVs. It would automatically classify the rest of the training inputs correctly.

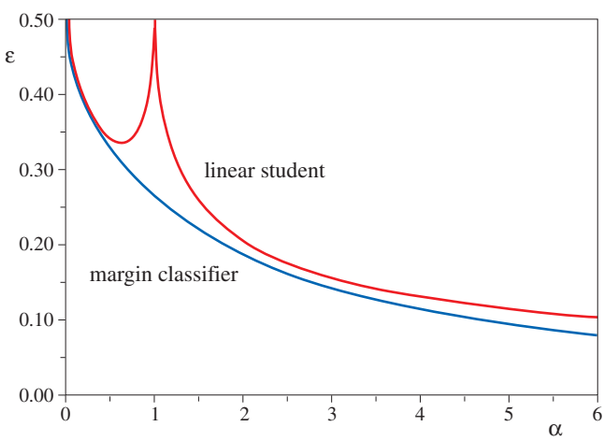

FIGURE 10 Learning curves for a linear student and for a margin classifier. $\alpha = m/N$.

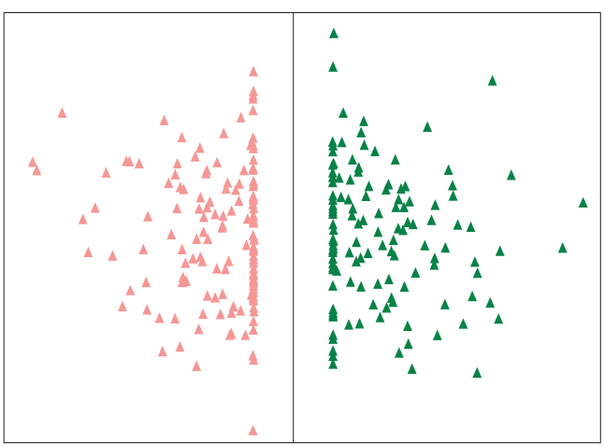

FIGURE 11 Learning with a margin classifier and $m = 300$ examples in an $N = 150$-dimensional space.

Hence, if in an actual classification experiment the number of SVs is small compared to the number of non-SVs, we may expect a good generalization ability.

The learning curve for a margin classifier (Opper and Kinzel, 1995) learning from a perceptron teacher (calculated by the statistical physics approach) is shown in Fig. 10 (blue curve). The concept of a margin classifier has recently been generalized to the so-called support vector machines (Vapnik, 1995), for which the inputs of a perceptron are replaced by suitable features which are cleverly chosen nonlinear functions of the original inputs. In this way, nonlinear separable rules can be learned, providing an interesting alternative to multilayer networks.

The Ising Perceptron

The approach of statistical physics can develop a specific predictive power in situations in which one would like to understand novel network models or architectures for which currently no efficient learning algorithm is known. As the simplest example, we consider a perceptron for which the couplings w_j are constrained to binary values $+1$ and -1 (Gardner and Derrida, 1989; Györgyi, 1990; Seung *et al.,* 1992b). For this so-called *Ising perceptron* (named after Ernst Ising, who studied coupled binary-valued elements as a model for a ferromagnet), perfect learning of examples is equivalent to a difficult combinatorial optimization problem (integer linear programming), which in the worst case is believed to require a learning time that increases exponentially with the number of couplings N.

To obtain the learning curve for the typical student, we can proceed as before, replacing $V(\varepsilon)$ by the number of student configurations that are consistent with the teacher which results in changing the entropic term appropriately. When the examples are provided by a teacher network of the same binary type, one can expect that the generalization error will decrease monotonically to zero as a function of α. The learning curve is shown as the blue curve in Fig. 9. For sufficiently small α, the discreteness of the couplings has almost no effect. However, in contrast to the continuous case, perfect generalization does not require infinitely many examples but is achieved already at a finite number $\alpha_c = 1.24$. This is not surprising because the teacher's couplings contain only a finite amount of information (one bit per coupling) and one would expect that it does not take much more than about N examples to learn them. The remarkable and unexpected result of the analysis is the fact that the transition to perfect generalization is discontinuous. The generalization error decreases immediately from a nonzero value to zero. This gives an impression about the complex structure of the space of all consistent students and also gives a hint as to why perfect learning in the Ising perceptron is a difficult task. For α slightly below α_c, the number of consistent students is small; nevertheless, the few remaining ones must still differ in a finite fraction of bits from each other and from the teacher so that perfect generalization is still impossible. For α slightly above α_c only the couplings of the teacher survive.

Learning with Errors

The example of the Ising perceptron teaches us that it will not always be simple to obtain zero training error. Moreover, an algorithm trying to achieve this goal may get stuck in local minima. Hence, the idea of allowing errors explicitly in the learning procedure, by introducing an appropriate noise, can make sense. An early analysis of such a stochastic training procedure and its generalization ability for the learning in so-called Boolean networks (with elementary computing units different from the ones used in neural networks) can be found in Carnevali and Patarnello (1987). A stochastic algorithm can be useful to escape local minima of the training error, enabling a better learning of the training set. Surprisingly, such a method can also lead to better generalization abilities if the classification rule is also corrupted by some degree of noise (Györgyi and Tishby, 1990). A stochastic training algorithm can be realized by the Monte Carlo metropolis method, which was invented to generate the effects of temperature in simulations of physical systems. Any changes of the network couplings which lead to a decrease of the training error during learning are allowed. However, with some probability that increases with the temperature, an increase of the training error is also accepted. Although in principle this algorithm may visit all the network's configurations, for a large system, with an overwhelming probability, only states close to some fixed training error will actually appear. The method of statistical physics applied to this situation shows that for sufficiently large temperatures (T) we often obtain a qualitatively correct picture if we repeat the approximate calculation for the noise-free case and replace the relative number of examples α by the effective number α/T. Hence, the learning curves become essentially stretched and good generalization ability is still possible at the price of an increase in necessary training examples.

Within the stochastic framework, learning (with errors) can now also be realized for the Ising perceptron, and it is interesting to study the number of relevant student configurations as a function of ε in more detail (Fig. 12). The green curve is obtained for a small value of α where a strong maximum with high generalization error exists. By increasing α, this maximum decreases until it is the same as the second maximum at $\varepsilon = 0.5$, indicating a transition like that of the blue learning curve in Fig. 9. For larger α, the state of perfect generalization should be the typical state. Nevertheless, if the stochastic algorithm starts with an initial state

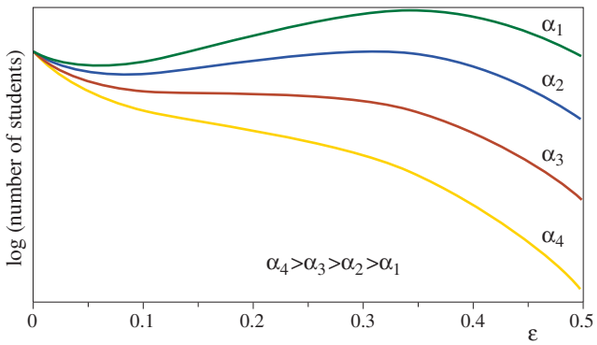

FIGURE 12 Logarithm of the number of relevant Ising students for different values of α.

which has no resemblance to the (unknown) teacher (i.e., with $\varepsilon = 0.5$), it will spend time that increases exponentially with N in the smaller local maximum, the metastable state. Hence, a sudden transition to perfect generalization will be observable only in examples which correspond to the blue curve of Fig. 12, where this metastable state disappears. For large vales of α (yellow curve), the stochastic algorithm will converge always to the state of perfect generalization. On the other hand, since the state with $\varepsilon = 0.5$ is always metastable, a stochastic algorithm which starts with the teacher's couplings will never drive the student out of the state of perfect generalization. It should be made clear that the sharp phase transitions are the result of the thermodynamic limit, where the macroscopic state is entirely dominated by the typical configurations. For simulations of any finite system a rounding and softening of the transitions will be observed.

More Sophisticated Computations Are Needed for Multilayer Networks

As a first step to understand the generalization performance of multilayer networks, one can study an architecture which is simpler than the fully connected one of Fig. 1b. The tree architecture of Fig. 13 has become a popular model. Here, each hidden unit is connected to a different set of the input nodes. A further simplification is the replacement of adaptive couplings from the hidden units to the output node by a prewired fixed function which maps the states of the hidden units to the output.

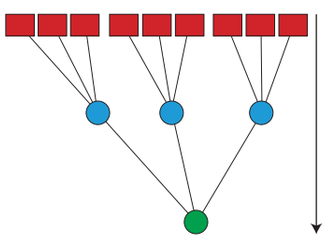

FIGURE 13 A two-layer network with tree architecture. The arrow indicates the direction of propagation of the information.

Two such functions have been studied in great detail. For the first one, the output gives just the majority vote of the hidden units — that is, if the majority of the hidden units is negative, then the total output is negative, and vice versa. This network is called a *committee machine.* For the second type of network, the *parity machine,* the output is the parity of the hidden outputs — that is, a minus results from an odd number of negative hidden units and a plus from an even number. For both types of networks, the capacity has been calculated in the thermodynamic limit of a large number N of (first layer) couplings (Barkai *et al.,* 1990; Monasson and Zecchina, 1995). By increasing the number of hidden units (but always keeping it much smaller than N), the capacity per coupling (and the VC dimension) can be made arbitrarily large. Hence, the VC theory predicts that the ability to generalize begins at a size of the training set which increases with the capacity. The learning curves of the typical parity machine (Fig. 14) being trained by a parity teacher for (from left to right) one, two, four, and six hidden units seem to partially support this prediction.

Below a certain number of examples, only memorization of the learned patterns occurs and not generalization. Then, a transition to nontrivial generalization takes place (Hansel *et al.,* 1992; Opper, 1994). Far beyond the transition, the decay of the learning curves becomes that of a simple perceptron (black curve in Fig. 14) independent of the number of hidden units, and this occurs much faster than for the bound given by VC theory. This shows that the typical learning curve can in fact be determined by more than one

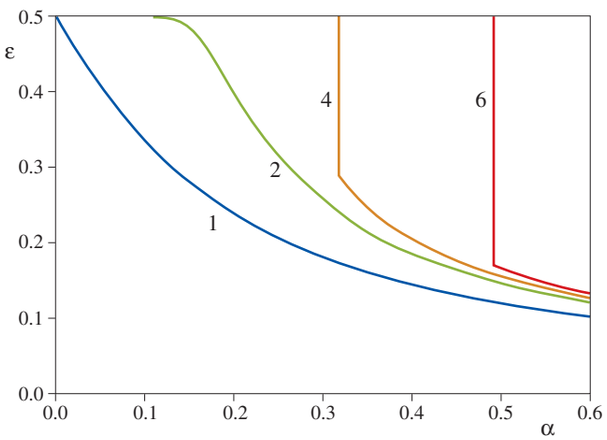

FIGURE 14 Learning curves for the parity machine with tree architecture. Each curve represents the generalization error ε as a function of α and is distinguished by the number of hidden units of the network.

complexity parameter. In contrast, the learning curve of the committee machine with the tree architecture of Fig. 13 (Schwarze and Hertz, 1992) is smooth and resembles that of the simple perceptron. As the number of hidden units is increased (keeping N fixed and very large), the generalization error increases, but despite the diverging VC dimension the curves converge to a limiting one having an asymptotic decay which is only twice as slow as that of the perceptron. This is an example for which typical and worst-case generalization behaviors are entirely different.

Recently, more light has been shed on the relation between average and worst-case scenarios of the tree committee. A reduced worst-case scenario, in which a tree committee teacher was to be learned from tree committee students under an input distribution, has been analyzed from a statistical physics perspective (Urbanczik, 1996). As expected, few students show a much worse generalization ability than the typical one. Moreover, such students may also be difficult to find by most reasonable learning algorithms because bad students require very fine tuning of their couplings. Calculation of the couplings with finite precision requires many bits per coupling that increases faster than exponentially with α and which for sufficiently large α will be beyond the capability of practical algorithms. Hence, it is expected that, in practice, a bad behavior will not be observed.

Transitions of the generalization error such as those observed for the tree parity machine are a characteristic feature of large systems which have a symmetry that can be spontaneously broken. To explain this, consider the simplest case of two hidden units. The output of this parity machine does not change if we simultaneously change the sign of all the couplings for both hidden units. Hence, if the teacher's couplings are all equal to $+1$, a student with all couplings equal to -1 acts exactly as the same classifier. If there are few examples in the training set, the entropic contribution will dominate the typical behavior and the typical students will display the same symmetry. Their coupling vectors will consist of positive and negative random numbers. Hence, there is no preference for the teacher or the reversed one and generalization is not possible. If the number of examples is large enough, the symmetry is broken and there are two possible types of typical students, one with more positive and the other one with more negative couplings. Hence, any of the typical students will show some similarity with the teacher (or it's negative image) and generalization occurs. A similar type of symmetry breaking also leads to a continuous phase transition in the fully connected committee machine. This can be viewed as a committee of perceptrons, one for each hidden unit, which share the same input nodes. Any permutation of these perceptrons obviously leaves the output invariant. Again, if few examples are learned, the typical state reflects the symmetry. Each student perceptron will show approximately the same similarity to every teacher perceptron. Although this symmetric state allows for some degree of generalization, it is not able to recover the teacher's rule completely. After a long plateau, the symmetry is broken and each of the student perceptrons specializes to one of the teacher perceptrons, and thus their similarity with the others is lost. This leads to a rapid (but continuous) decrease in the generalization error. Such types of learning curves with plateaus can actually be observed in applications of fully connected multilayer networks.

Outlook

The worst-case approach of the VC theory and the typical case approach of statistical physics are important theories for modeling and understanding the complexity of learning to generalize from examples. Although the VC approach plays an important role in a general theory of learnability, its practical applications for neural networks have been limited by the overall generality of the approach. Since only weak assumptions about probability distributions and machines are considered by the theory, the estimates for generalization errors have often been too pessimistic. Recent developments of the theory seem to overcome these problems. By using modified VC dimensions, which depend on the data that have actually occurred and which in favorable cases are much smaller than the general dimensions, more realistic results seem to be possible. For the support vector machines (Vapnik, 1995) (generalizations of the margin classifiers which allow for nonlinear boundaries that separate the two classes), Vapnik and collaborators have shown the effectiveness of the modified VC results for selecting the optimal type of model in practical applications.

The statistical physics approach, on the other hand, has revealed new and unexpected behavior of simple network models, such as a variety of phase transitions. Whether such transitions play a cognitive role in animal or human brains is an exciting topic. Recent developments of the theory aim to understand dynamical problems of learning. For example, online learning (Saad, 1998), in which the problems of learning and generalization are strongly mixed, has enabled the study of complex multilayer networks and has stimulated research on the development of optimized algorithms. In addition to an extension of the approach to more complicated networks, an understanding of the robustness of the typical behavior, and an interpolation to the other extreme, the worst-case scenario is an important subject of research.

Acknowledgments

I thank members of the Department of Physics of Complex Systems at the Weizmann Institute in Rehovot, Israel, where parts of this article were written, for their warm hospitality.

References Cited

AMARI, S., and MURATA, N. (1993). Statistical theory of learning curves under entropic loss. *Neural Comput.* **5,** 140.

BARKAI, E., HANSEL, D., and KANTER, I. (1990). Statistical mechanics of a multilayered neural network. *Phys. Rev. Lett.* **65,** 2312.

BISHOP, C. M. (1995). *Neural Networks for Pattern Recognition.* Clarendon/Oxford Univ. Press, Oxford/New York.

CARNEVALI, P., and PATARNELLO, S. (1987). Exhaustive thermodynamical analysis of Boolean learning networks. *Europhys. Lett.* **4,** 1199.

COVER, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Trans. El. Comp.* **14,** 326.

ENGEL, A., and VAN DEN BROECK, C. (1993). Systems that can learn from examples: Replica calculation of uniform convergence bound for the perceptron. *Phys. Rev. Lett.* **71,** 1772.

GARDNER, E. (1988). The space of interactions in neural networks. *J. Phys. A* **21,** 257.

GARDNER, E., and DERRIDA, B. (1989). Optimal storage properties of neural network models. *J. Phys. A* **21,** 271.

GYÖRGYI, G. (1990). First order transition to perfect generalization in a neural network with binary synapses. *Phys. Rev. A* **41,** 7097.

GYÖRGYI, G., and TISHBY, N. (1990). Statistical theory of learning a rule. In *Neural Networks and Spin Glasses: Proceedings of the STATPHYS 17 Workshop on Neural Networks and Spin Glasses* (W. K. Theumann and R. Koberle, Eds.). World Scientific, Singapore.

HANSEL, D., MATO, G., and MEUNIER, C. (1992). Memorization without generalization in a multilayered neural network. *Europhys. Lett.* **20,** 471.

KINZEL, W., and RUJÀN, P. (1990). Improving a network generalization ability by selecting examples. *Europhys. Lett.* **13,** 473.

LEVIN, E., TISHBY, N., and SOLLA, S. (1989). A statistical approach to learning and generalization in neural networks. In *Proceedings of the Second Workshop on Computational Learning Theory* (R. Rivest, D. Haussler, and M. Warmuth, Eds.). Morgan Kaufmann, San Mateo, CA.

MÉZARD, M., PARISI, G., and VIRASORO, M. A. (1987). Spin glass theory and beyond. In *Lecture Notes in Physics,* Vol. 9. World Scientific, Singapore.

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.

OPPER, M. (1994). Learning and generalization in a two-layer neural network: The role of the Vapnik–Chervonenkis dimension. *Phys. Rev. Lett.* **72,** 2113.

OPPER, M., and HAUSSLER, M. (1991). Generalization performance of Bayes optimal classification algorithm for learning a perceptron. *Phys. Rev. Lett.* **66,** 2677.

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.). Springer-Verlag, New York.

SAAD, D. (Ed.) (1998). *Online Learning in Neural Networks.* Cambridge Univ. Press, New York.

SCHWARZE, H., and HERTZ, J. (1992). Generalization in a large committee machine. *Europhys. Lett.* **20,** 375.

SCHWARZE, H., and HERTZ, J. (1993). Generalization in fully connected committee machines. *Europhys. Lett.* **21,** 785.

SEUNG, H. S., SOMPOLINSKY, H., and TISHBY, N. (1992a). Statistical mechanics of learning from examples. *Phys. Rev. A* **45,** 6056.

SEUNG, H. S., OPPER, M., and SOMPOLINSKY, H. (1992b). Query by committee. In *The Proceedings of the Vth Annual Workshop on Computational Learning Theory (COLT92),* p. 287. Association for Computing Machinery, New York.

SOMPOLINSKY, H., TISHBY, N., and SEUNG, H. S. (1990). Learning from examples in large neural networks. *Phys. Rev. Lett.* **65,** 1683.

URBANCZIK, R. (1996). Learning in a large committee machine: Worst case and average case. *Europhys. Lett.* **35,** 553.

VALLET, F., CAILTON, J., and REFREGIER, P. (1989). Linear and nonlinear extension of the pseudo-inverse solution for learning Boolean functions. *Europhys. Lett.* **9,** 315.

VAPNIK, V. N. (1982). *Estimation of Dependencies Based on Empirical Data.* Springer-Verlag, New York.

VAPNIK, V. N. (1995). *The Nature of Statistical Learning Theory.* Springer-Verlag, New York.

VAPNIK, V. N., and CHERVONENKIS, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probability Appl.* **16,** 254.

General References

ARBIB, M. A. (Ed.) (1995). *The Handbook of Brain Theory and Neural Networks.* MIT Press, Cambridge, MA.

BERGER, J. O. (1985). *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag, New York.

HERTZ, J. A., KROGH, A., and PALMER, R. G. (1991). *Introduction to the Theory of Neural Computation.* Addison-Wesley, Redwood City, CA.

MINSKY, M., and PAPERT, S. (1969). *Perceptrons.* MIT Press, Cambridge, MA.

WATKIN, T. L. H., RAU, A., and BIEHL, M. (1993). The statistical mechanics of learning a rule. *Rev. Modern Phys.* **65,** 499.