Vapnik-Chervonenkis Dimension of Neural Nets

Peter L. Bartlett
BIOwulf Technologies and
University of California at Berkeley
Department of Statistics
367 Evans Hall, CA 94720-3860, USA
bartlett@stat.berkeley.edu

Wolfgang Maass*
Institute for Theoretical Computer Science
Technische Universität Graz
A-8010 Graz, Austria
maass@igi.tu-graz.ac.at

February 26, 2002

*Wolfgang Maass acknowledges partial support by the Fonds zur Förderung der wissenschaftlichen Forschung (FWF), Austria, project P12153 and the research network NeuroCOLT II of the European Union.

1 Introduction

For any assignment of values to its internal parameters θ (weights, thresholds, etc.) a neural network $\mathcal{N}$ with binary outputs computes a function $x \mapsto \mathcal{N}(\theta, x)$ from D into $\{0,1\}$, where D is the domain of the network inputs x (e.g. $D = \mathbb{R}^n$). The Vapnik-Chervonenkis dimension (VC-dimension) of $\mathcal{N}$ is a number which may be viewed as a measure of the richness (or diversity) of the collection of all functions $x \mapsto \mathcal{N}(\theta, x)$ that can be computed by $\mathcal{N}$ for different values of its internal parameters θ. Not surprisingly, the VC-dimension of a neural network is related to the number of training examples that are needed in order to train $\mathcal{N}$ to compute—or approximate—a specific target function $h : D \to \{0,1\}$. We shall discuss a number of different types of neural networks, but typically the VC-dimension grows polynomially (in many cases, between linearly and quadratically) with the number of adjustable parameters of the neural network. In particular, if the number of training examples is large compared to the VC-dimension, the network's performance on training data is a reliable indication of its future performance on subsequent data.

The notion of the VC-dimension, which was introduced in [Vapnik and Chervonenkis, 1971], is not specific to neural networks. It applies to any parameterized class F of functions $x \mapsto f(\theta, x)$ from some domain D into $\{0,1\}$, where θ ranges over some given parameter space, for example $\mathbb{R}^w$. Related notions for the case of real-valued outputs will be discussed later. The largest possible richness of this class F of functions from D into $\{0,1\}$ is achieved if *every* function $h : D \to \{0,1\}$ can be computed by a function $x \mapsto f(\theta, x)$ in F. In this case one says that D is *shattered* by F, and the VC-dimension of F is equal to

$|D|$, the number of elements of the domain D. In the general case one defines the VC-dimension of F (VCdim(F)) as the size of the largest subset D' of its domain D so that D' is shattered by F (or more precisely: by the restrictions of the function $x \mapsto f(\theta, x)$ in F to inputs $x \in D'$). In other words: the VC-dimension of F is the size of the largest subset D' of its domain D for which every dichotomy h over D' (i.e., each of the $2^{|D'|}$ many functions $h : D' \mapsto \{0, 1\}$) can be computed by some function in F, or in mathematical notation:

$$\forall\, h : D' \rightarrow \{0,1\} \;\exists\, \theta \;\forall\, x \in D' \;(f(\theta, x) = h(x))\ .$$

Although the definition of the VC-dimension focuses on the shattering effect, it yields a remarkable bound that holds for *all* finite subsets X of the domain D: If d is the VC-dimension of F then at most $\Sigma_{i=0}^{d}\binom{|X|}{i} \leq |X|^d + 1$ functions from X into $\{0, 1\}$ can be computed by (restrictions of) functions in F. This estimate, which is commonly referred to as Sauer's Lemma, was independently discovered by several authors, including [Vapnik and Chervonenkis, 1971] (see [Anthony and Bartlett, 1999], Chapter 3 for a review). Results of this form provide the mathematical basis for bounding the number of training examples that are needed for learning functions in F in terms of the VC-dimension of F, as in the following theorem. (This theorem is a consequence of a slightly improved version, due to Talagrand, of a result from [Vapnik and Chervonenkis, 1971]; see [Anthony and Bartlett, 1999], Chapter 4 for related references.)

Theorem 1.1 *Suppose that F is a class of functions mapping from a domain X into $\{0, 1\}$, and suppose also that F has VC-dimension $d < \infty$. Let $((x_1, y_1), \ldots, (x_m, y_m))$ be a sequence of m randomly chosen labelled training* *examples from* $X \times \{0,1\}$. *Then with probability at least* $1-\delta$ *over this sequence, any function* $f \in F$ *has*

$$\Pr(f(x) \neq y) \leq \frac{1}{m} \left|\{1 \leq i \leq m : f(x_i) \neq y\}\right| + \epsilon,$$

provided that $m \geq c\left(d + \log(1/\delta)\right)/\epsilon^2$, *where* c *is a universal constant.*

In particular, if the sample size is large compared to the VC-dimension of the function class, the function from the class that minimizes the number of errors on a training sample will have near-minimal probability of misclassifying subsequent patterns. [**Insert here links to related articles on PAC-learning, etc.**].

The definition of the VC-dimension of a function class F immediately implies that $\mathrm{VCdim}(F) \leq \log_2 |F|$ if F is finite. Thus in particular if F is parameterized by w k-bit parameters, $\mathrm{VCdim}(F) \leq kw$. However, many infinite classes F also have a finite VC-dimension. Consider for example the class F_{T2} of functions from $\mathbb{R}^2$ into $\{0,1\}$ that can be computed by linear threshold gates (McCulloch-Pitts neurons) with two inputs:

$$F_{T2} = \{\langle x_1, x_2\rangle \mapsto \mathcal{H}(\theta_1 x_1 + \theta_2 x_2 - \theta_3) : \theta = \langle \theta_1, \theta_2, \theta_3\rangle \in \mathbb{R}^3\},$$

where $\mathcal{H}(x) = 1$ if $x \geq 0$, otherwise $\mathcal{H}(x) = 0$. (See Figure 1; the shaded region in each box corresponds to $h(x) = 1$.) Obviously the set $D' := \{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle\}$ can be shattered by F_{T2} (as illustrated by the eight dichotomies shown in Figure 1). On the other hand it is easy to see that the set $D' \cup \{\langle 1,1\rangle\}$ can *not* be shattered by F (since the dichotomy h that assumes the value 1 on the points $\langle 0,0\rangle$ and $\langle 1,1\rangle$ and the value 0 on the points $\langle 0,1\rangle$ and $\langle 1,0\rangle$ cannot be computed by any linear threshold

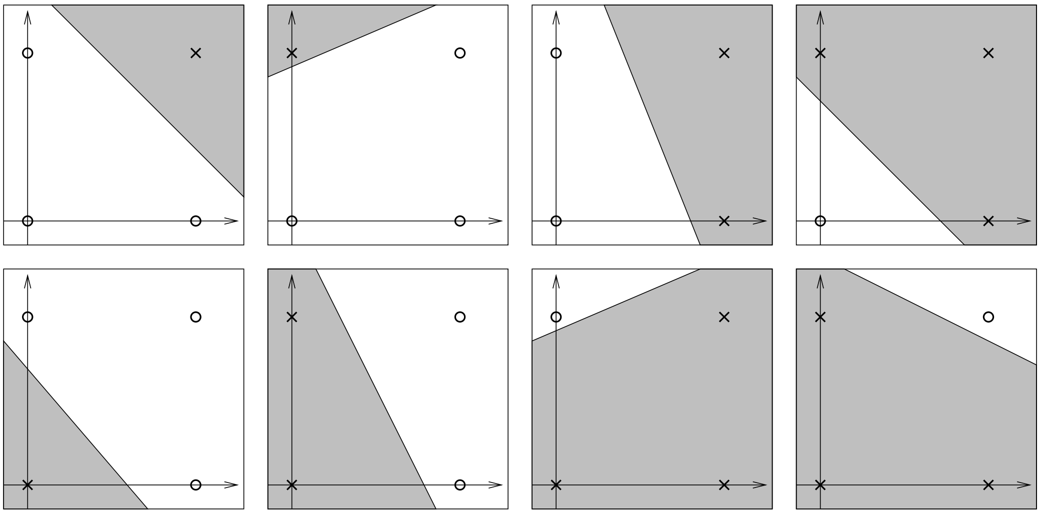

Figure 1: Eight dichotomies of four points in $\mathbb{R}^2$ computed by the class F_{T2} of linear threshold functions. For each of the eight functions $h \in F_{T2}$ illustrated, the shaded region represents the halfspace where $h(x) = 1$. When a point x satisfies $h(x) = 1$, it is marked as a cross; when it satisfies $h(x) = 0$ it is marked as a circle. The functions illustrated show that the set $\{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle\}$ is shattered by F_{T2}.

gate). Somewhat less obvious to see is that there exists *no* set $D' \subseteq \mathbb{R}^2$ consisting of 4 or more points which is shattered by F_{T2}, i.e. that 3 is in fact the VC-dimension of F_{T2}. This follows immediately from the following theorem.

Theorem 1.2 *(Wenocur and Dudley): Let F_{Tn} be the class of functions from $\mathbb{R}^n$ into $\{0,1\}$ that can be computed by a linear threshold gate, for any $n \in \mathbb{N}$. Then F_{Tn} has VC-dimension $n+1$.*

Sketch of the proof: One can easily verify that the set $S := \{\underline{0}\} \cup \{\underline{e}_i :$

$i \in \{1, \ldots, n\}\}$ is shattered by $\mathcal{N}$ (where $\underline{e}_i \in \{0,1\}^n$ denotes the ith unit vector). Hence $\mathrm{VCdim}(\mathcal{N}) \geq n+1$.

The upper bound follows from Radon's Theorem, which states that any set S of $\geq n+2$ points in $\mathbb{R}^n$ can be partitioned into sets S_0 and S_1 such that the convex hulls of S_0 and S_1 intersect. Obviously such sets S_0 and S_1 cannot be separated by any hyperplane, hence not by any linear threshold gate. ∎

2 Feedforward Neural Nets with Binary Outputs

Theorem 2.1 *(Cover, 1968; Baum and Haussler, 1989): Let $\mathcal{N}$ be an arbitrary feedforward neural net with w weights that consists of linear threshold gates. Then* $\mathrm{VCdim}(\mathcal{N}) = O(w \cdot \log w)$.

Sketch of the proof: Let S be some arbitrary set of m input-vectors for $\mathcal{N}$. By Theorem 1.2 and Sauer's Lemma, a gate g in $\mathcal{N}$ can compute at most $|X|^{\text{fan-in}(g)+1} + 1$ different functions from any finite set $X \subseteq \mathbb{R}^{\text{fan-in}(g)}$ into $\{0,1\}$, where fan-in(g) denotes the number of inputs of gate g. Hence $\mathcal{N}$ can compute at most $\prod_{g \text{ gate in } \mathcal{N}} (m^{\text{fan-in}(g)+1} + 1) \leq m^{2w}$ different functions from S into $\{0,1\}$. If S is shattered by $\mathcal{N}$ then $\mathcal{N}$ can compute all 2^m functions from S into $\{0,1\}$, which implies that $2^m \leq m^{2w}$, and hence $m \leq 2w \cdot \log m$. It follows that $\log m = O(\log w)$, thus $m = O(w \cdot \log w)$. ∎

It is tempting to conjecture that the VC-dimension of a neural net $\mathcal{N}$ cannot be larger than the total number of parameters in $\mathcal{N}$, which, in view of Theorem 1.2, is equal to the sum of the VC-dimensions of the individual gates in $\mathcal{N}$. This conjecture would imply that the $O(w \log w)$ upper bound of Theorem 2.1 can be improved to $O(w)$. However the following result (whose proof uses techniques from circuit complexity theory) shows that the superlinear upper bound of Theorem 2.1 is in fact asymptotically optimal. Hence with regard to the VC-dimension it is fair to say that a neural net can be "more than the sum of its parts."

Theorem 2.2 *(Maass, 1993): There exist neural networks $\mathcal{N}$ consisting of linear threshold gates whose VC-dimension scales proportional to $w \cdot \log w$, where w is the number of parameters of $\mathcal{N}$.*

This superlinear growth of the VC-dimension occurs already for feedforward neural nets with two hidden layers in the case of discrete network inputs. Sakurai [Sakurai, 1993] showed that for the case of continuous network inputs it may even occur with a single hidden layer.

Proving upper bounds for sigmoidal neural nets, whose computational units employ some smooth activation function instead of the Heaviside function $\mathcal{H}$, turns out to be quite challenging. For instance, there exists a feedforward neural net consisting of a linear threshold gate as output unit and two hidden units that employ as activation function a very smooth (real analytic) strictly increasing squashing function, which has an infinite VC-dimension. (See, for example, [Anthony and Bartlett, 1999]; the first result of this form was due to Sontag.) This shows that it is necessary to exploit more specific properties of a particular activation function, for example of the logistic sigmoid, in order to achieve a finite upper bound for the VC-dimension of a sigmoidal neural net. The following theorem [Goldberg and Jerrum, 1995] provides the key step in this direction.

Theorem 2.3 *(Goldberg and Jerrum, 1995): Consider the parameterized class*

$$F = \left\{ x \mapsto f(\theta, x) : \theta \in \mathbb{R}^d \right\},$$

for some $\{\pm 1\}$*-valued function* f*. Suppose that, for each input* $x \in \mathbb{R}^n$*, there is an algorithm that computes* $f(\theta, x)$ *and this computation takes no more than* t *operations of the following types:*

- *the arithmetic operations* $+$, $-$, $\times$, *and* $/$ *on real numbers,*
- *jumps conditioned on* $>$, $\geq$, $<$, $\leq$, $=$, *and* $\neq$ *comparisons of real numbers, and*
- *output* 0 *or* 1.

Then $\mathrm{VCdim}(H) \leq 4d(t + 2)$.

The proof involves counting cells in parameter space. Consider a single thresholded real-valued function, such as a neural network with a single real output that is thresholded at 0. Fix a set of n input patterns. To estimate the VC-dimension, we can estimate the number of distinct dichotomies of those patterns. Suppose two parameter values give distinct output labels for one of these patterns. Then in moving between these distinct values in parameter space, we must pass through a parameter value where the real output is zero in response to the pattern. Such values form the boundaries of cells in parameter space, and within a cell all classifications are identical.

Under appropriate conditions, counting the number of dichotomies reduces to counting the number of these cells. For well-behaved parameterizations, the number of cells defined by these zero sets is closely related to the number of distinct solutions of generic systems of equations. If the output of the network is polynomial in the parameters, classical results give bounds on the number of such solutions, and hence on the number of dichotomies. (Ben-David and Lindenbaumindependently obtained this proof and result in a paper that appeared at the same conference as Goldberg and Jerrum's paper.) The argument is essentially unchanged if the parameterized function class is a fixed boolean function of a number of thresholded functions that are each polynomial in the parameters. If the computation of $f(\theta, x)$ involves few operations, this implies f can be represented as a fixed boolean function of a small number of thresholded, low degree polynomials.

2.1 Piecewise polynomial activation functions

As an example of the application of Theorem 2.3, the output of a linear threshold net can be computed using $O(w)$ of the operations listed in the theorem, where w is the number of parameters, so the VC-dimension is $O(w^2)$. Theorem 2.1 shows that this bound can be improved to $\Theta(w \cdot \log w)$. Similarly, if the nonlinearity is a piecewise polynomial function with a fixed number of pieces of fixed degree, the number of operations is again $O(w)$, so the VC-dimension bound of $O(w^2)$ again applies. In some cases, this bound also can be improved, by applying Theorem 2.3 more carefully. This leads to the following bound [Bartlett et al, 1998] on the VC-dimension of a feedforward neural net of piecewise polynomial gates arranged in L layers (so that each gate has connections only from gates in earlier layers).

Theorem 2.4 *(Bartlett, Maiorov, Meir, 1998) Suppose $\mathcal{N}$ is a feed-forward network with w weights, l layers, and all non-output gates having a fixed piecewise-polynomial activation function with a fixed number of pieces. Then* $\text{VCdim}(\mathcal{N}) = O(wl \log w + wl^2)$.

Linear threshold gates have a piecewise polynomial activation function. Thus, Theorem 2.4, together with the lower bound for linear threshold nets (Theorem 2.2), show that the VC-dimension of piecewise polynomial networks with a fixed number of layers is also $\Theta(w \log w)$. Perhaps surprisingly, the transition from linear threshold gates to piecewise polynomial gates does not increase the rate of growth of the VC-dimension for networks with a fixed number of layers.

In contrast, if the number of layers is unbounded, the rate of growth of the VC-dimension can be faster for piecewise polynomial networks than for linear threshold networks. The following lower bound applies to networks of gates with an activation function satisfying two conditions: it has distinct left and right limits, and it has non-zero slope somewhere. This result is due to Koiran and Sontag [Koiran and Sontag, 1997]; the refinement to give the dependence on the depth was shown by Bartlett, Maiorov and Meir, and also by Sakurai.

Theorem 2.5 *Suppose the activation function $s : \mathbb{R} \to \mathbb{R}$ has the following properties:*

1. $\lim_{\alpha\to\infty} s(\alpha) \neq \lim_{\alpha\to-\infty} s(\alpha)$, *and*

2. *s is differentiable at some point $\alpha_0 \in \mathbb{R}$, with $s'(\alpha_0) \neq 0$.*

Then for any l and $w \geq 10l$, there is a neural network $\mathcal{N}$ with l layers and w parameters, where every gate but the output gate has activation function s, the output gate being a linear threshold gate, and for which the VC-dimension scales as lw. In particular, for $l = \Theta(w)$, there are such networks with $\mathrm{VCdim}(\mathcal{N}) = \Omega(w^2)$.

2.2 Sigmoidal activation functions

While the VC-dimension of networks with piecewise polynomial activation functions is well understood, most applications of neural networks use the logistic sigmoid function, or gaussian radial basis function. Unfortunately, it is not possible to compute such functions using a finite number of the arithmetic operations listed in Theorem 2.3. However, Karpinski and Macintyre [Karpinski and Macintyre, 1997] extended Theorem 2.3 to allow the computation of exponentials. The proof uses the same ideas, but the bound on the number of solutions of a system of equations is substantially more difficult.

Theorem 2.6 *Consider the parameterized class*

$$F = \left\{ x \mapsto f(\theta, x) : \theta \in \mathbb{R}^d \right\},$$

for some $\{\pm 1\}$-valued function f. Suppose that, for each input $x \in \mathbb{R}^n$, there is an algorithm that computes $f(\theta, x)$ and this computation takes no more than t operations of the following types:

- *the exponential function $\alpha \mapsto e^{\alpha}$ on real numbers, and*

- *all of the operations listed in Theorem 2.3.*

Then $\mathrm{VCdim}(F) = O(t^2 d^2)$.

We immediately obtain bounds for the VC-dimension of sigmoid networks and radial basis networks of the form $O(w^4)$, where w is the number of parameters. This upper bound is considerably larger than the $\Theta(w \log w)$ bound achieved for linear threshold networks or fixed depth piecewise polynomial networks. It remains open whether it is optimal. For fixed depth sigmoid networks, the best lower bounds are those implied by Theorems 2.2 and 2.5: $\Omega(w \log w)$ for networks of fixed depth, and $\Omega(w^2)$ for arbitrary depth.

If the inputs are restricted to a small set of integers, a simple parameter transformation allows the machinery of the piecewise polynomial case to be applied to two-layer sigmoid networks, giving the following result. See [Anthony and Bartlett, 1999] for a proof. A related result applies to gaussian radial basis networks.

Theorem 2.7 *Consider a two-layer feedforward network* $\mathcal{N}$ *with input domain* $X = \{-k, \ldots, k\}^n$ *(for* $k \in \mathbb{N}$*) and first-layer computation gates with the standard sigmoid activation function (the output gate being a linear threshold gate). Let* w *be the total number of parameters in the network. Then* $\mathrm{VCdim}(\mathcal{N}) = O(w \log(wk))$.

3 Feedforward Neural Nets with Real Outputs

All of the results presented so far apply to nets with binary-valued outputs. Neural networks with real outputs are also commonly used, for instance in regression problems. In such cases, the appropriate measure of complexity of the network is a scale-sensitive version of the VC-dimension, called the fat-shattering dimension.

Suppose that F is a set of functions mapping from a domain X to $\mathbb{R}$, $D = \{x_1, x_2, \dots, x_m\}$ is a subset of the domain X, and γ is a positive real number. Then we say that D is *γ-shattered* by F if there are real numbers $r_1, r_2, \dots, r_m$ such that for all $b \in \{0,1\}^m$ there is a function f_b in F with

$$\begin{aligned} f_b(x_i) &\geq r_i + \gamma \quad \text{if } b_i = 1, \\ f_b(x_i) &\leq r_i - \gamma \quad \text{if } b_i = 0 \end{aligned}$$

for $1 \leq i \leq m$. The *fat-shattering dimension* of F at scale γ, denoted $\mathrm{fat}_F(\gamma)$, is the size of the largest subset D of the domain X that is γ-shattered by F.

It is significant that this notion of complexity depends on a scale parameter γ. In a sense, the fat-shattering dimension ignores complex behaviour of the function class below a certain scale. If we are concerned with predicting a real value to some accuracy ϵ, then it seems that the behaviour of the function class on a scale much smaller than ϵ should not be relevant. The following result formalizes this intuition, by showing that the fat-shattering dimension is related to the number of training examples that are needed to solve a regression problem. Although the result is stated in terms of the squared prediction error, similar results apply to a broad class of loss functions. (The result relies on a generalization of Sauer's Lemma to the fat-shattering dimension from [Alon et al, 1997]. See, for example, [Anthony and Bartlett, 1999] for a proof.)

Theorem 3.1 *Suppose that F is a class of functions mapping from a domain X into the real interval $[0,1]$, and suppose also that F has finite fat-shattering dimension. Let $((x_1,y_1),\ldots,(x_m,y_m))$ be a sequence of m randomly chosen labelled training examples from $X \times [0,1]$. Then there are constants c_1, c_2 such that, with probability at least $1-\delta$, any function f^* that has the average over the sample of $(f^*(x)-y)^2$ within $1/\sqrt{m}$ of the minimum over F satisfies*

$$\mathbf{E}(f^*(x)-y)^2 \leq \inf_{g\in F} \mathbf{E}(g(x)-y)^2 + \epsilon, \tag{1}$$

provided that $m \geq c_1 \left(\mathrm{fat}_F(c_2\epsilon)\log^2(1/\epsilon) + \log(1/\delta)\right)/\epsilon^2$.

It is also known that for any learning algorithm to return a function f^* that satisfies (1) requires the amount of training data to grow at least as $\mathrm{fat}_F(\epsilon)$. This shows that the fat-shattering dimension is the right measure of complexity of a function class that is used for regression.

The fat-shattering dimension is also useful for pattern classification using thresholded real-valued functions, like neural networks. Many practical algorithms for such functions typically lead to solutions that have large *margins* on the training data, where the margin of a thresholded real-valued function is the amount by which the function is to the correct side of the threshold. The following result, from [Bartlett, 1998], shows that in these cases the fat-shattering dimension gives an upper bound on the error.

Theorem 3.2 *Consider a class F of real-valued functions. With probability at least $1-\delta$ over m independently generated examples $(x_1,y_1),\ldots,(x_m,y_m)$,* *for every function f in F, the classifier $\mathcal{H}(f)$ has misclassification probability no more than*

$$\frac{b}{m} + O\left(\sqrt{\frac{1}{m}\left(\mathrm{fat}_F(\gamma/16)\log^2 m + \log(1/\delta)\right)}\right),$$

where b is the number of labelled training examples with margin no more than γ.

The easiest way to obtain bounds on the fat-shattering dimension for neural networks is via VC-dimension bounds. The following theorem shows that the fat-shattering dimension of a network is no bigger than the VC-dimension of a slightly larger network with one additional input variable. The theorem is a trivial observation involving another combinatorial dimension, called the *pseudo-dimension*; see Chapters 11 and 14 of [Anthony and Bartlett, 1999] for details.

Theorem 3.3 *Let $\mathcal{N}$ be any neural network with a single real-valued output unit, and form a neural network $\mathcal{N}'$ as follows. The network $\mathcal{N}'$ has one extra real input and one extra computation unit. This additional computation unit is the output unit of $\mathcal{N}'$, and is a linear threshold unit receiving input only from the output unit of $\mathcal{N}$ and from the new input. For any $\gamma > 0$, $\mathrm{fat}_{\mathcal{N}}(\gamma) \leq \mathrm{VCdim}(\mathcal{N}')$.*

This result and the upper bounds of the previous section immediately imply upper bounds on the fat-shattering dimension of networks with linear threshold gates, with piecewise polynomial activation functions, and with logistic sigmoidal activation functions. These bounds are in terms of the number of parameters in the network, and, significantly, do not depend on the scale parameter γ. In some cases, bounds like this are very loose. For example, the following theorem [Bartlett, 1998] gives an upper bound on the fat-shattering dimension of a two-layer network with an arbitrary number of computation units (and hence parameters).

Theorem 3.4 *Suppose that $s : \mathbb{R} \to [-b, b]$ is a non-decreasing bounded function. For $v \geq 1$, suppose that F is the class of functions from $\mathbb{R}^n$ to $\mathbb{R}$ computed by two layer neural networks with an arbitrary number of first layer units, each with activation function s, and a linear output unit for which the sum of the magnitudes of the weights is bounded by v. Then*

$$\mathrm{fat}_F(\epsilon) = O\left(\frac{nv^2}{\epsilon^2} \ln\left(\frac{v}{\epsilon}\right)\right).$$

It follows that for regression and pattern classification (when the learning algorithm finds a network with large margins on the training data), it is not necessary to restrict the number of parameters in the network, provided the parameters are kept small. Bounds of this kind are also known for deeper networks; see [Anthony and Bartlett, 1999] for details.

4 Other Applications to Neural Networks

The VC-dimension of recurrent neural networks was analysed by DasGupta, Koiran and Sontag (see [Sontag, 1998] for a survey of results for feedforward and recurrent neural nets). In this case it is of interest to consider the case of a time series as the network input. The length k of the time series enters the bounds for the VC-dimension of the neural network as an additional parameter (in most bounds the number w of network parameters is multiplied by a factor of the form $\log k$ or k).

In models for biological neural circuits the transmission delays between neurons enter as additional parameters, which influence the VC-dimension of such circuits even more than the synaptic weights: the VC-dimension of a very simple mathematical model for a single spiking neuron grows superlinearly in the number d of adjustable delays, and the VC-dimension of a feedforward network of such neurons grows quadratically in d [Maass and Schmitt, 1999].

In [Koiran, 1996] a technique was introduced for using *upper* bounds on the VC-dimension of neural networks for proving *lower* bounds on the size of any sigmoidal neural net (with thresholded output) that is able to compute some concrete function. No other method for proving lower bounds on the size of sigmoidal neural nets is known at present. This technique can, for example, be used to show that there exist functions that can be computed with few spiking neurons, but if they are computed by a sigmoidal neural net, the number of neurons must grow linearly in the number of inputs. (see COMPUTATION WITH SPIKING NEURONS).

5 Discussion

The VC-dimension of a neural net with a binary output measures its "expressiveness". The related notion of the fat-shattering dimension provides a similar tool for the analysis of a neural net with a real-valued output. The derivation of bounds for the VC-dimension and the fat-shattering dimension of neural nets has turned out to be a rather challenging but quite interesting chapter in the mathematical investigation of neural nets. This work has brought a number of sophisticated mathematical tools into this research area, which have subsequently turned out to be also useful for the solution of a variety of other problems regarding the complexity of computing and learning on neural nets. More detailed information about all of the results in Sections 1 to 3 can be found in [Anthony and Bartlett, 1999].

Bounds for the VC-dimension (resp. fat-shattering dimension) of a neural net $\mathcal{N}$ provide estimates for the number of random examples that are needed to train $\mathcal{N}$ so that it has good generalization properties (i.e., so that the error of $\mathcal{N}$ on new examples from the same distribution is at most ε, with probability $\geq 1 - \delta$). From the point of view of a single application these bounds tend to be too large, since they provide such a generalization guarantee for *any* probability distribution on the examples and for *any* training algorithm that minimizes disagreement on the training examples. For some special distributions and specific training algorithms tighter bounds can be obtained, for instance with the help of heuristic arguments (replica techniques) from statistical physics.

References

[Alon et al, 1997] Alon, N., Ben-David, S., Cesa-Bianchi, N., Haussler, D. 1997. Scale-sensitive dimensions, uniform convergence, and learnability, *Journal of the ACM* 44(4):616–631.

[Anthony and Bartlett, 1999] Anthony, M., Bartlett, P. L. 1999. *Neural Network Learning: Theoretical Foundations*, Cambridge University Press. *

[Bartlett, 1998] Bartlett, P. L. 1998. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network, *IEEE Transactions on Information Theory*, 44(2):525–536.

[Bartlett et al, 1998] Bartlett, P. L., Maiorov, V., Meir, R. 1998. Almost linear VC-dimension bounds for piecewise polynomial networks, *Neural Computation*, 10:2159–2173.

[Baum and Haussler, 1989] Baum, E. B., Haussler, D. 1989. What size net gives valid generalization?, *Neural Computation*, 1:151-160.

[Cover, 1968] Cover, T. M. 1968. Capacity problems for linear machines, in *Pattern Recognition*, (L. Kanal ed.), Thompson Book Co., 283-289.

[Goldberg and Jerrum, 1995] Goldberg, P. W., Jerrum, M. R. 1995. Bounding the Vapnik-Chervonenkis dimension of concept classes parametrized by real numbers, *Machine Learning*, 18(2/3):131–148.

[Karpinski and Macintyre, 1997] Karpinski, M., Macintyre, A. J. 1997. Polynomial bounds for VC dimension of sigmoidal and general Pfaffian neural networks, *Journal of Computer and System Sciences*, 54:169–176.

[Koiran, 1996] Koiran, P. 1996. VC-dimension in circuit complexity , *Proc. of the 11th IEEE Conference on Computational Complexity*, 81-85.

[Koiran and Sontag, 1997] Koiran, P., Sontag, E. D. 1997. Neural networks with quadratic VC dimension, *Journal of Computer and System Sciences*, 54(1):190–198.

[Maass, 1993] Maass, W. 1993. Bounds for the computational power and learning complexity of analog neural nets, *Neural Computation*, 6:875-882.

[Maass and Schmitt, 1999] Maass, W., Schmitt, M. On the complexity of learning for spiking neurons with temporal coding, *Information and Computation*, 153:26-46.

[Sakurai, 1993] Sakurai, A. 1993. Tighter bounds of the VC-dimension of three-layer networks, *Proc. of WCNN '93*, 3:540-543. 45:20-48.

[Sontag, 1998] Sontag, E. D. 1998. VC-dimension of neural networks, in: *Neural Networks and Machine Learning* (C. M. Bishop, ed.), Springer Verlag, Berlin, 69-95. *

[Vapnik and Chervonenkis, 1971] Vapnik, V. N., Chervonenkis, A. Y. 1971. On the uniform convergence of relative frequencies of events to their probabilities, *Theory of Probability and its Applications*, 16:264-280.