CAMBRIDGE UNIVERSITY ENGINEERING DEPARTMENT
TRUMPINGTON STREET
CAMBRIDGE CB2 1PZ

On the Practical Applicability of VC Dimension Bounds

Sean B. Holden[1] and Mahesan Niranjan[2]

CUED/F-INFENG/TR.155 1994

October 12, 1994

To appear in *Neural Computation*

This report is available by anonymous ftp from svr-ftp.eng.cam.ac.uk in /pub/reports/holden_tr155.ps.Z.

[1]email: sbh@eng.cam.ac.uk
[2]email: niranjan@eng.cam.ac.uk

Abstract

This article addresses the question of whether some recent Vapnik-Chervonenkis (VC) dimension based bounds on sample complexity can be regarded as a practical design tool. Specifically, we are interested in bounds on the sample complexity for the problem of training a pattern classifier such that we can expect it to perform valid generalization. Early results using the VC dimension, while being extremely powerful, suffered from the fact that their sample complexity predictions were rather impractical. More recent results have begun to improve the situation by attempting to take specific account of the precise algorithm used to train the classifier. We perform a series of experiments based on a task involving the classification of sets of vowel formant frequencies. The results of these experiments indicate that the more recent theories provide sample complexity predictions that are significantly more applicable in practice than those provided by earlier theories; however, we also find that the recent theories still have significant shortcomings.

Contents

1 Introduction

Of the small number of existing, alternative theories that aim to model the phenomenon of generalization, one of the most widely studied is that based on computational learning theory (Anthony and Biggs (1992), Natarajan (1991)), which uses ideas originally introduced by Valiant (1984) and Blumer *et al.* (1989). It has become clear that a parameter of fundamental importance in this theory is the *Vapnik-Chervonenkis (VC) dimension*, which we define in full below. The VC dimension can be regarded as a measure of the *capacity* or *expressive power* of a connectionist network or other pattern classifier.

In this article we address the following question: do the VC dimension bounds available at present in any way constitute a practically applicable design tool, in the sense that they can be used in practice to guide the design of a pattern classifier? This type of question is not often asked by researchers in computational learning theory, where the emphasis tends to be on the production of powerful *theoretical* results. However, despite the significant intrinsic interest inspired by such results, the long-term aim of such studies must be to provide powerful and generally applicable tools for the design of machine learning systems, and, consequently, it is important that some attempt is made to assess the available theoretical results from this point of view.

The results presented in this article can be regarded as an extension of those obtained by Cohn and Tesauro (1992), who have made a detailed study of the average generalization performance of various networks applied to some simple problems, and compared the results with the worst-case bounds provided by some VC dimension based results. However there are three important differences. Firstly, all our experiments use types of networks for which either exact results or very good bounds on the VC dimension are known. This is advantageous for the reasons discussed in section 3; some, but not all of the experiments in Cohn and Tesauro (1992) used networks with this property. Secondly, our networks can be trained without the need to use the back-propagation algorithm; use of this algorithm leads, as discussed in Cohn and Tesauro (1992), to the need to be extremely careful in the control of possible associated random and systematic experimental errors. Finally, whereas in Cohn and Tesauro (1992) the experiments are based on synthetic data for rather unrealistic problems, namely the *'majority'*, *'real-valued threshold'*, *'majority-XOR'* and *'threshold-XOR'* problems, the experiments presented here are based on real data, namely a large set of formant frequencies for ten different vowels uttered by people of different age and gender; this data was introduced by Peterson and Barney (1952). Additionally, in this work we concentrate specifically on the investigation of recent bounds due to Haussler *et al.* (1990,1994), which were considered only quite momentarily in Cohn and Tesauro (1992), but which were found to perform better than earlier bounds in the situations considered. Finally, we discuss in detail the difficulties involved in applying these recent bounds in practice.

1.1 Why are VC Dimension Results Useful?

Results based on the VC dimension are useful because they tell us about the ability of a classifier to generalize after it has been trained. There is at present no single, complete theory of generalization that provides us with general and easily applied design guidelines; such a theory would obviously be highly desirable. Results based on the VC dimension have taken various different forms; the best known form (at least, in the connectionist network research community), which appears in the work of Blumer *et al.* (1989), Baum and Haussler (1989), Holden and Rayner (1994), Shawe-Taylor and Anthony (1991) and others, is as follows. Assume that the classifier of interest takes inputs in $\mathbf{R}^n$ and produces outputs in $\{0, 1\}$, and assume that training examples are generated independently according to some *arbitrary* distribution P on $\mathbf{R}^n \times \{0, 1\}$. Assume for the moment that our classifier is a connectionist network, and let the network of interest have architecture A (the network can be any type of feedforward network; we ignore the details for the time being). Finally, assume that we have a parameter $0 < \epsilon \leq 1/4$. Then there exists a value k, which is a function *only* of A and ϵ, such that if,

1. the network can learn at least a fraction $1 - \epsilon/2$ of k randomly drawn training examples and,

2. all future examples are also drawn according to P,

then there is a probability[3] close to 1 that the actual *generalization error* of the network is at most ϵ, where generalization error is defined as the probability that, for a random example $(\mathbf{x}, o)$ drawn according to P, the output of the trained network for the input $\mathbf{x}$ is not equal to o.

This sounds like, and indeed is, a very powerful result. It is completely independent of the actual distribution P that governs the way in which examples are generated and it is also independent of the actual algorithm used to train the network. The drawback is that all known upper bounds on the required value of k are rather large, in the sense that they lead to numbers of training examples that we would not in general expect to be able to load with the required accuracy on a network the size of A. This observation was verified experimentally in Cohn and Tesauro (1992). There are two main reasons for this (see Haussler *et al.* (1994)); unfortunately, the result is limited by precisely the characteristics that make it so powerful. First, the result is valid for all distributions P, even the ones that we would never expect to govern the occurrence of data in practice. The second reason is that the result is independent of the algorithm used to train the network, and the explanation here is rather more subtle. Assuming the structure of the network is fixed, then given a particular vector $\mathbf{w}$ of weights the network computes a function $f_{\mathbf{w}} : \mathbf{R}^n \longrightarrow \{0, 1\}$. We denote by $\mathcal{F}$ the class of all such functions, so,

$$\mathcal{F} = \{f_{\mathbf{w}} : \mathbf{w} \in \mathbf{R}^W\} \tag{1}$$

where W is the total number of weights used by the network and we assume that weights are real-valued. The result described above would apply even if we were able to use a training algorithm which always provides a function having acceptable error on the training examples (assuming that at least one such function exists in $\mathcal{F}$), but which in addition always provides the function that of all such functions is the one which provides the worst possible performance on future examples generated according to P.

Of the two reasons stated for the large size of the standard VC dimension bounds, we might intuitively expect that the first — the distribution independence — is the most significant. However, recent results obtained by Haussler *et al.* (1990,1994) suggest that by considering the precise training algorithm used it may be possible to maintain distribution independence and obtain quite practical bounds; note however that the model of machine learning used is different in some respects to that illustrated here, and is described in the next section. This is quite definitely an encouraging result; to say that we know something definite about the distribution governing the occurrence of data would in practice tend to mean that we would have a significant amount of *a priori* knowledge about the problem being addressed, and it is clearly desirable to maintain distribution independence if possible.

This article is organized as follows. In section 2 we briefly review some of the most recent theoretical bounds on the number of examples required when training a classifier or other system under specific conditions. In section 3 we describe the experiments used to investigate the quality

[3] The exact probability involved here can be quantified in terms of a further parameter δ; in this case k is a function of A, ϵ and δ. Further elaborations are also possible; we omit the full details here, and refer the reader to Blumer *et al.* (1989) and Shawe-Taylor and Anthony (1991).

of these bounds; the results of these experiments are described in section 4 and discussed in full in section 5, where we also discuss the general practical applicability of the relevant theory. Section 6 concludes the article.

2 Recent VC Dimension Bounds

In this section we provide a brief summary of some of the results in the articles by Haussler *et al.* (1990, 1994), which are investigated experimentally in the remainder of this article. Let X be an environment, which we identify with the set of all possible inputs to the system of interest. This is typically $\mathbf{R}^n$ or a subset such as $[0,1]^n$ where n is the number of inputs to the system; it can also be a set such as $\{0,1\}^n$. Given any class $\mathcal{F}$ of functions with domain X and range $\{0,1\}$ we define its VC dimension VCdim($\mathcal{F}$) in the usual manner. Given an arbitrary set of k points in X, each function $f \in \mathcal{F}$ induces a *dichotomy* or *two-colouring* on the set by dividing it into two disjoint subsets consisting of the points mapped to 1 and the points mapped to 0. Given such a set we can apply all the functions in $\mathcal{F}$ and count the total number of distinct dichotomies obtained. The VC dimension of $\mathcal{F}$ is defined as the size k of the largest subset of X for which we can obtain all 2^k possible dichotomies. For examples of VC dimensions for various relevant classes of functions see Anthony and Biggs (1992), Anthony and Holden (1994), Blumer *et al.* (1989), Bartlett (1992), Maass (1993), Sontag (1992), Holden (1992), Wenocur and Dudley (1981) and references therein.

The task of training a classifier to solve a given problem can be modelled as that of identifying some *target function* $g_T : X \longrightarrow \{0,1\}$ which is assumed to be a member of some class $\mathcal{G}$ of *target concepts*. We assume that a sequence T_k of k training examples is generated as follows. The sequence,

$$T_k = ((\mathbf{x}_1, o_1), (\mathbf{x}_2, o_2), \ldots, (\mathbf{x}_k, o_k)) \tag{2}$$

is formed by drawing k inputs $\mathbf{x}_i$ independently according to an *arbitrary* distribution P on X and forming each corresponding o_i such that $o_i = g_T(\mathbf{x}_i)$. Note that this is slightly different to the process described in the previous section. In the process described earlier both inputs $\mathbf{x}_i$ and outputs o_i are governed by a distribution P on $\mathbf{R}^n \times \{0,1\}$. In the process described here only inputs are generated according to a distribution, and outputs are then obtained using g_T. Note also that examples are in effect assumed to be noise free, and that we also assume that future examples are produced in the same manner after training.

Throughout this article we will denote by $\mathcal{F}$ the class of functions computed by a connectionist network or other pattern classifier (equation 1). Training the network involves adjusting its weights, on the basis of T_k, such that it computes a function $f_{\mathbf{w}} \in \mathcal{F}$ that is a 'good approximation' to $g_T \in \mathcal{G}$. In all of the following work we assume that the classifier learns to classify the examples in T_k correctly (that the classifier is *consistent*). We now ask the following question: if, under the conditions described, our classifier is trained using a sequence T_k of training examples, what is the probability that $f_{\mathbf{w}}(\mathbf{x}) \neq g_T(\mathbf{x})$ for new random inputs $\mathbf{x}$ generated according to P? We call this probability the *generalization error*, which we denote $\epsilon_g(k)$ for the remainder of this article. If we can answer this question then we clearly know something about the generalization ability of our network. We now describe some results that allow us to bound the expected value of $\epsilon_g(k)$.

2.1 A General Prediction Algorithm

Let us, for the moment, discard the assumption that we will necessarily use some function chosen from a specified class $\mathcal{F}$ in order to predict which output, 0 or 1, is associated with an input $\mathbf{x}$, chosen according to P. In Haussler *et al.* (1990) an algorithm, called the *randomized 1-inclusion graph prediction strategy*, is constructed which has the following property: if it is provided with a set T_k of examples generated as described above, along with a further input $\mathbf{x}_{k+1}$ drawn independently according to P, then the probability that its prediction is not in fact equal to $g_T(\mathbf{x}_{k+1})$ is at most $\text{VCdim}(\mathcal{G})/(k+1)$. The fact that an algorithm exists that is capable of providing this performance can be used to obtain a further result described in the next subsection.

Some degree of care needs to be taken in interpreting the results of Haussler *et al.* (1990). In particular, recall that the generalization error $\epsilon_g(k)$ denotes the probability of error for new inputs $\mathbf{x}$ generated according to P and classified using a classifier trained on a specific sequence T_k. This is *not* equivalent to the probability that a single new $\mathbf{x}_{k+1}$ generated according to P is misclassified by such a network. In the former case a single trial corresponds to the generation of a single input $\mathbf{x}$ according to P, whereas in the latter case a single trial corresponds to the generation of $(k+1)$ inputs according to P. Formally[4],

$$\epsilon_g(k) = P[\{\mathbf{x} : f(\mathbf{x}) \neq g_T(\mathbf{x})\}] \tag{3}$$

where f denotes the function computed by a classifier after training on some sequence T_k. The generalization error $\epsilon_g(k)$ is the standard measure of generalization performance used in practice.

2.2 Using a Bayes Optimal Classification Algorithm

We noted above that by producing results that are too powerful — by making them independent of the actual distributions or algorithms used — we can obtain results that are rather impractical. In the model of learning being used at present a further source of such problems has been introduced. Specifically, results must apply even if the actual target function g_T being used is highly unrealistic. In Haussler *et al.* (1994) this problem is addressed by introducing a probability distribution $\mathcal{P}$ on $\mathcal{G}$ that governs the way in which target functions appear. The article then considers the performance of a classifier that is optimum in the sense that it implements a *Bayes optimal classification algorithm* (Duda and Hart (1973)). In this case, it can be shown using the result given in the previous subsection that the expected generalization error is,

$$E[\epsilon_g(k)] \leq \frac{\text{VCdim}(\mathcal{G})}{k+1}, \tag{4}$$

where the expectation is taken over all k-element training sequences and all target functions, and the result holds regardless of the actual distributions P and $\mathcal{P}$.

The bound of equation 4 was proved in Haussler *et al.* (1994), in which it was also conjectured that it will be possible to obtain an improved bound of,

$$E[\epsilon_g(k)] \leq \frac{1}{2}\frac{\text{VCdim}(\mathcal{G})}{k+1}. \tag{5}$$

Two important points should be noted here. The first regards the use of a class $\mathcal{G}$ of target concepts and corresponding distribution $\mathcal{P}$. The use of a class $\mathcal{G}$ in the theory effectively models the fact that our classifier might, in practice, have to be applied to a selection of different problems. The distribution $\mathcal{P}$ can be thought of as encoding our prior beliefs about which function(s) will have to be learned. In this article we consider a single, specific problem (described in the next section). This specific problem corresponds to a *specific* $g_T \in \mathcal{G}$, and we can therefore assume that $\mathcal{P}$ assigns a probability of 1 to this particular g_T and a probability of 0 to all other

[4] We use the notation $P[\mathcal{E}]$ to denote the probability of the event $\mathcal{E}$ according to the distribution P.

target functions. As the results of equations 4 and 5 are independent of the actual distribution $\mathcal{P}$, they still apply.

The second point that should be noted is that the *Bayes optimal classification algorithm*, which is assumed in deriving equation 4, is distinct from the *Bayes classifier* (Duda and Hart (1973)) for a given problem. The Bayes optimal classification algorithm tells us an optimum way of predicting the output associated with a new input on the basis of a finite quantity of training data for the model of machine learning described above, whereas the Bayes classifier tells us how to classify new examples in order to obtain the smallest possible probability of error, given complete information about the statistics of a pattern classification problem. In fact, in the model of machine learning considered, a function exists — namely g_T — that classifies all examples correctly. The Bayes classifier therefore makes no errors for new examples and has an associated error probability of zero.

To end this section, it is relevant to mention some further attempts that have been made to obtain more realistic results than those obtained using the standard VC dimension theory. One such attempt has involved the introduction of the *effective VC dimension* (Guyon *et al.* (1992), Bottou *et al.* (1994)), and techniques based on statistical physics have also been used for this purpose. A comprehensive review of the latter work is given by Watkin *et al.* (1993). We will not discuss either of these alternative techniques further in this article.

3 Experiments Using the Peterson/Barney Data

3.1 The Peterson/Barney Data

The data used for the experiments was derived from a database containing the first four formant frequencies for ten different vowel sounds uttered by people of different age and gender; this database was originally due to Peterson and Barney (1952). For the purposes of this study, a two-class pattern classification problem was constructed in which we attempt to discriminate between the front vowels [i], [I], [e] and [ae] (class 1) and the mid vowels [a] and [o], and back vowels [U] and [u] (class 2). Figure 1 illustrates the entire set of available examples as it appears using only formants 2 and 3; in the following experiments all four formant frequencies were used as inputs to the networks. Class 1 contains a total of 600 examples, and class 2 a total of 594 examples. There were no conflicting examples in the complete set of 1194 examples, in the sense that no two examples exist with equal input vectors but conflicting classifications.

3.2 The Networks Used

The networks used in the experiments were specific examples of *Linearly Weighted Connectionist Networks*. These networks have been studied for many years; examples can be found in Nilsson (1965) and Cover (1965), and an extensive review can be found in Holden (1993). This class of networks computes functions of the general form,

$$f_{\mathbf{w}}(\mathbf{x}) = \mathcal{H}\left[\overline{f}_{\mathbf{w}}(\mathbf{x})\right] \tag{6}$$

where,

$$\overline{f}_{\mathbf{w}}(\mathbf{x}) = w_0 + \sum_{i=1}^{m} w_i \phi_i(\mathbf{x}). \tag{7}$$

In equations 6 and 7, $\mathbf{w}^T = [\ w_0 \quad w_1 \quad \cdots \quad w_m\ ] \in \mathbf{R}^{m+1}$ is a vector of $W = m+1$ real-valued weights, the $\phi_i : X \longrightarrow \mathbf{R}$ are m *fixed*, typically nonlinear *basis functions*, and $\mathcal{H}$ denotes the

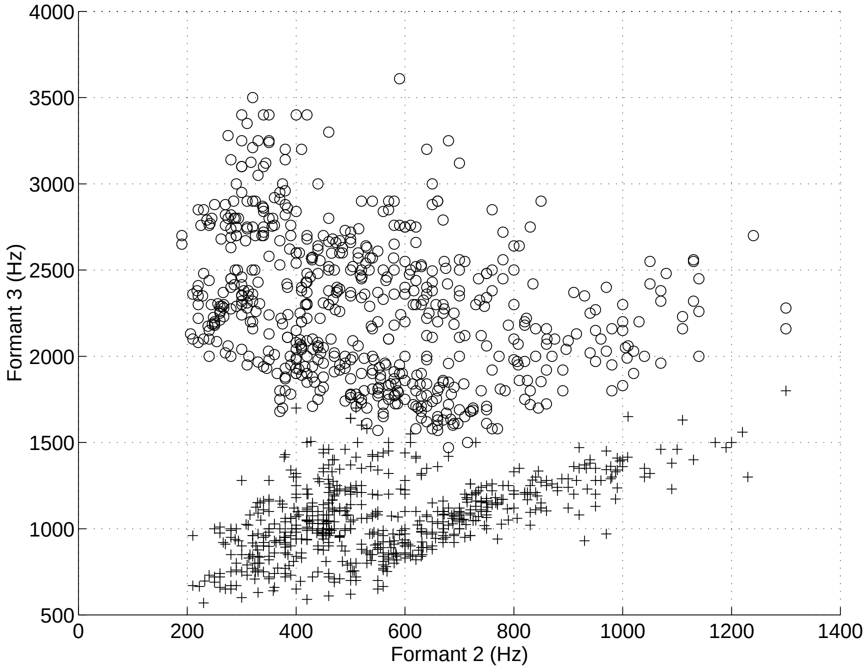

Figure 1: The Peterson/Barney data. Only two formants are shown in this figure. Examples in class 1 are displayed using 'o' and examples in class 2 using '+'.

step function,

$$\mathcal{H}(y) = \begin{cases} 1 & \text{if } y \geq 1/2 \\ 0 & \text{otherwise} \end{cases} . \tag{8}$$

Standard, linear perceptrons are clearly a specific case of this definition, but are not very useful for our purposes. In the following experiments we used two other network types, namely *polynomial networks*, and *radial basis function networks* having *fixed centres*. In the former case, the basis functions are products of elements of the input vector $\mathbf{x}^T = [\ x_1 \quad x_2 \quad \cdots \quad x_n\ ]$, for example $\phi_i(\mathbf{x}) = x_1 x_2^2 x_{10}^5$. If a network of this type has n inputs and uses basis functions corresponding to all possible products of up to d input elements then we call it an (n, d) *discriminator*. For example, an $(n, 2)$ discriminator computes functions of the form,

$$f_{\mathbf{w}}(\mathbf{x}) = \mathcal{H}\left[w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n}\sum_{j=i}^{n} w_{ij} x_i x_j\right]. \tag{9}$$

It is possible to show that an (n, d) discriminator has $W = \binom{n+d}{n}$ weights. In the case of radial basis functions we use *inverse multiquadric* basis functions of the form,

$$\phi_i(\mathbf{x}) = \frac{1}{\sqrt{\|\mathbf{x} - \mathbf{y}_i\|^2 + 1}} \tag{10}$$

where each $\mathbf{y}_i \in X$ is a fixed *centre* chosen according to the technique described below and $\|.\|$ is a suitable norm; in our case we assume that $X = \mathbf{R}^n$ and use the Euclidean norm.

There are two reasons for using these networks in preference to more usual alternatives, such as multilayer perceptrons or radial basis function networks with adapting centres. First, in both cases we have very good results for the VC dimension of the network. In the case of polynomial networks we have $\mathrm{VCdim}(\mathcal{F}) = W$ and in the case of radial basis function networks we have $W - 1 \leq \mathrm{VCdim}(\mathcal{F}) \leq W$; this is proved in Anthony and Holden (1994) (see also Anthony and Holden (1993)). In the following work we assume that $\mathrm{VCdim}(\mathcal{F}) = W - 1$ in the case of radial basis function networks. In the more usual cases mentioned the best that we can do at present is to bound the VC dimension for some specific cases, and it is not even known in general whether the bounds available are tight. Secondly, there is a technique available for training these networks that has significant advantages, when compared with the nonlinear optimization required for training the alternative network types, in that it allows us to significantly reduce the likelihood that various potential sources of random and systematic experimental error will affect our results.

It is well-known (see Wan (1990) and Gish (1990)) that when addressing a two-class pattern classification problem using a sufficiently powerful connectionist network with a single, real-valued output we can obtain an approximation to the posterior probability that a given input is in class 1 by minimizing the usual squared error,

$$\xi(\mathbf{w}) = \sum_{i=1}^{k} [o_i - \overline{f}_{\mathbf{w}}(\mathbf{x}_i)]^2 \tag{11}$$

for the examples in a training set T_k. We can therefore obtain an approximation to a Bayes classifier using a network of the form of equation 6. Of course, it is important to remember that we are unlikely to obtain the exact Bayes classifier, and consequently that the measured generalization errors obtained are in fact likely to be *worse* than those obtained using the true Bayes classifier. (The points raised above regarding the distinction between the Bayes classifier and the Bayes optimal classification algorithm should be recalled at this point.)

We must be rather careful to consider precisely how much experimental results obtained using classifiers trained by minimizing $\xi(\mathbf{w})$ can tell us about the quality of the bounds in equations 4 and 5. We use this approach as it is much closer to the types of technique used in practice than the Bayes optimal classification algorithm, and as the latter algorithm is in general likely to be extremely difficult to implement in full. The performance of the Bayes optimal classification algorithm depends on the value of k, as the algorithm only has access to a finite number of training examples. Although minimizing $\xi(\mathbf{w})$ can allow us to approximate the Bayes classifier under suitable conditions, it should be noted that it still corresponds to training a classifier using k examples. As the Bayes optimal classification algorithm is the optimal procedure for predicting outputs corresponding to new inputs within the model of machine learning described above, we should expect classifiers designed by minimizing $\xi(\mathbf{w})$ to perform *worse* in general than the Bayes optimal classification algorithm. This is discussed further in section 5.

The weight vector that minimizes $\xi(\mathbf{w})$ can be obtained easily as,

$$\mathbf{w} = \mathbf{P}^{+}\mathbf{o} \tag{12}$$

where,

$$\mathbf{P} = \begin{bmatrix} 1 & \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_m(\mathbf{x}_1) \\ 1 & \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_m(\mathbf{x}_2) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \phi_1(\mathbf{x}_k) & \phi_2(\mathbf{x}_k) & \cdots & \phi_m(\mathbf{x}_k) \end{bmatrix}, \tag{13}$$

$\mathbf{o}^T = [\ o_1 \quad o_2 \quad \cdots \quad o_k\ ]$ and $\mathbf{P}^{+}$ denotes the Moore-Penrose pseudoinverse of $\mathbf{P}$ (Golub and Van Loan (1989)). By training using this technique we obtain a unique, global minimum of $\xi(\mathbf{w})$. We therefore avoid the potential introduction of errors due to convergence to local minima, and in addition we avoid several other potential sources of error, as it is not necessary to choose initial weights, learning rates, momentum constants, training batch size, training cutoff time or order of pattern presentation as in many alternative techniques. A potential problem with this training technique is that it does not guarantee to find a weight vector that correctly classifies all the examples in T_k, even if such a weight vector exists. This is discussed in section 5.

3.3 The Experiments

In order to assess the bounds of equations 4 and 5 we conducted six experiments, three using polynomial networks and three using radial basis function networks. The polynomial networks were a $(4, 2)$ discriminator, a $(4, 4)$ discriminator, and a $(4, 5)$ discriminator, having 15, 70 and 126 weights respectively. The radial basis function networks again had 15, 70 and 126 weights, and the centres used were chosen at random such that they were uniformly distributed in the subset of $\mathbf{R}^4$ populated by available inputs[5]. For each radial basis function network the same set of centres was used throughout the relevant experiment. The networks were trained using the method described; all six networks are powerful enough to learn exactly the entire set of 1194 available examples illustrated in figure 1.

There is an important point that should be noted regarding the choice of networks and the interpretation of the results that are presented below. The actual bounds in equations 4 and 5 require that we know the VC dimension of the class $\mathcal{G}$ of possible target concepts; recall also that we must assume that the network can always learn the training examples exactly. The former point is a significant shortcoming of the current theory, as clearly we are unlikely in practice to be in a position to draw any conclusion about the VC dimension of $\mathcal{G}$. However, because we assume that our network can always learn the available training examples exactly we can assume that $\text{VCdim}(\mathcal{G}) \leq \text{VCdim}(\mathcal{F})$, and for the purposes of the following work we assume that $\text{VCdim}(\mathcal{G}) = \text{VCdim}(\mathcal{F})$. This issue is discussed in full in section 5.

In each experiment values of k in the range 50 to 790 were examined, using steps of size 20. For each value of k the relevant network was trained for 40 different, randomly selected sets of training examples. In each case the generalization error was estimated using a further (disjoint), randomly selected set of 350 test examples. This allowed us to obtain estimates of the expected and worst case generalization error. Sets of training and testing examples were generated by selecting examples uniformly at random, without replacement, from the entire set of available examples. Examples were selected without replacement in order to reflect the fact that, as real speech has a high degree of variability, it is unlikely that any real set of examples will ever contain two or more identical sets of formants. When training and testing the networks sets of examples were chosen such that there were equal numbers of examples from each of the two classes.

A potential problem exists in using this method for selecting examples in that it does not exactly reflect the process of generating examples that is assumed by the theory, in which inputs $\mathbf{x}_i$ are selected according to arbitrary P and outputs are formed as $g_T(\mathbf{x}_i)$. All the experiments were repeated using an alternative selection method in which training and testing examples were chosen uniformly at random *with* replacement and *without* forcing the sets to contain equal numbers of examples from each of the two classes. The training and testing sets were still forced to be disjoint. The results of the second set of experiments are given in appendix A; precisely the same conclusions can be drawn from either set of experiments.

A final point should be noted regarding the manner in which examples are selected. Use of a disjoint testing set reflects a standard experimental procedure, whereas the theory allows training and testing sets to have common elements. This suggests that our measured generalization errors might be higher than those obtained if we allowed training and testing sets that are not disjoint and hence correspond more exactly to the theory.

[5]This, of course, involves designing the networks having taken into account the characteristics of all the available data, and it is not clear whether the current theory strictly allows us to do this. We do not regard this as a problem in this case however, as enough is known about the characteristics of speech formants to allow good guesses for the relevant ranges in which to place centres to be made without making any reference to the actual data.

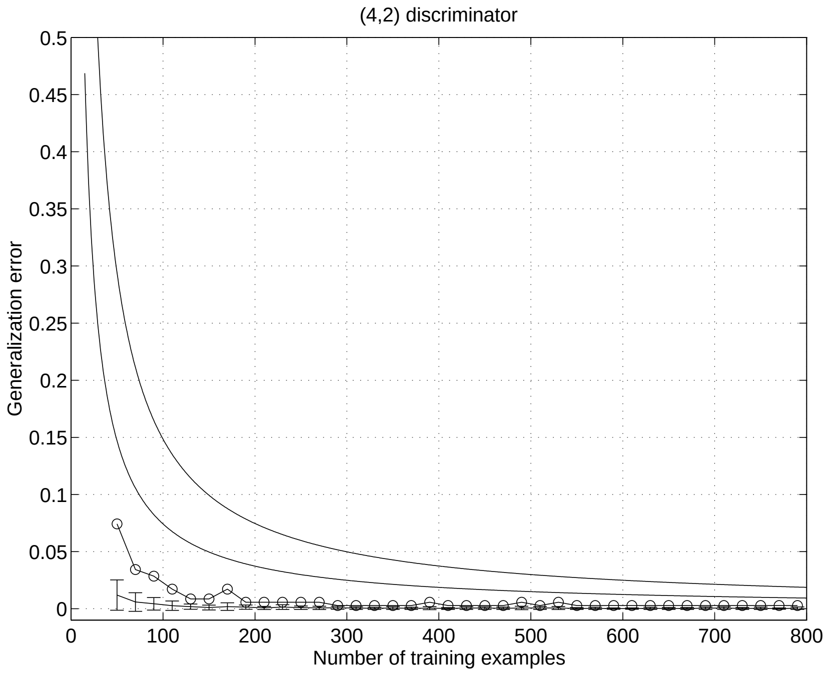

Figure 2: Results obtained for a $(4,2)$ discriminator. The two upper lines show the theoretical bounds of equations 4 and 5 respectively, assuming $\mathrm{VCdim}(\mathcal{G}) = \mathrm{VCdim}(\mathcal{F})$; for each value of k the bound on $E[\epsilon_g(k)]$ is shown. The line with points marked using 'o' shows the worst measured generalization error, and the final line shows the average generalization error along with error bars showing the standard deviation.

4 Experimental Results

Figures 2, 3, 4, 5, 6, and 7 show the results obtained using a $(4,2)$ discriminator, a $(4,4)$ discriminator, a $(4,5)$ discriminator, a radial basis function network with 15 weights, a radial basis function network with 70 weights, and a radial basis function network with 126 weights respectively.

Perhaps the most important observation that can be made here is that the fully proved bound of equation 4 in fact overestimates both the expected and worst case generalization error in these cases by a significant factor. Although this bound is a great improvement on those typically encountered using the earlier theories, it still provides significant overestimates. (Note however that we must be cautious in drawing the latter conclusion, for reasons discussed in the next section.)

The conjectured bound of equation 5 appears to be more realistic. In fact this bound also bounds the worst measured generalization errors in all the experiments conducted with only a very few exceptions such as, for example, in figure 11 in appendix A. Given that, as noted above, our networks cannot in general be expected to perform as well as the Bayes optimal classification algorithm, and also considering some other factors, discussed below, that lead us to expect that our measured generalization errors are worse than would be obtained if we were able to match *exactly* the conditions required by the theory, we conjecture that if it were possible to match exactly the required conditions then the worst measured generalization errors might also be bounded by equation 5 in the instances studied.

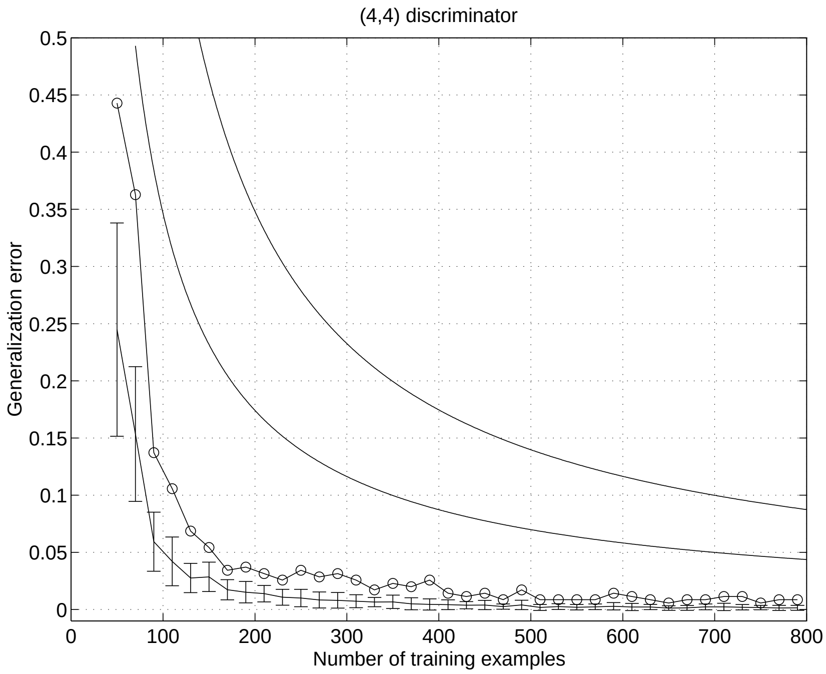

Figure 3: Results obtained for a $(4,4)$ discriminator. The plots are as described in figure 2.

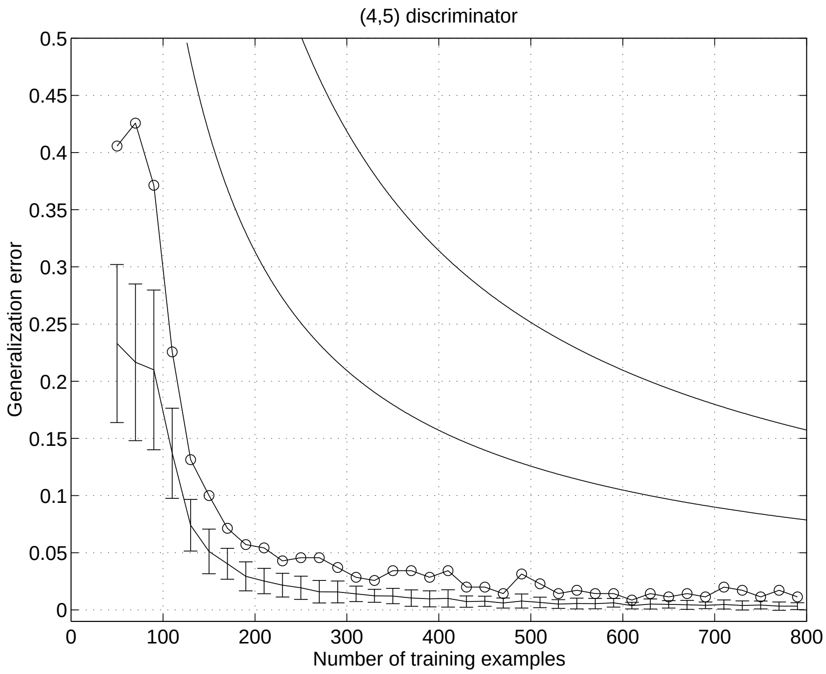

Figure 4: Results obtained for a $(4,5)$ discriminator. The plots are as described in figure 2.

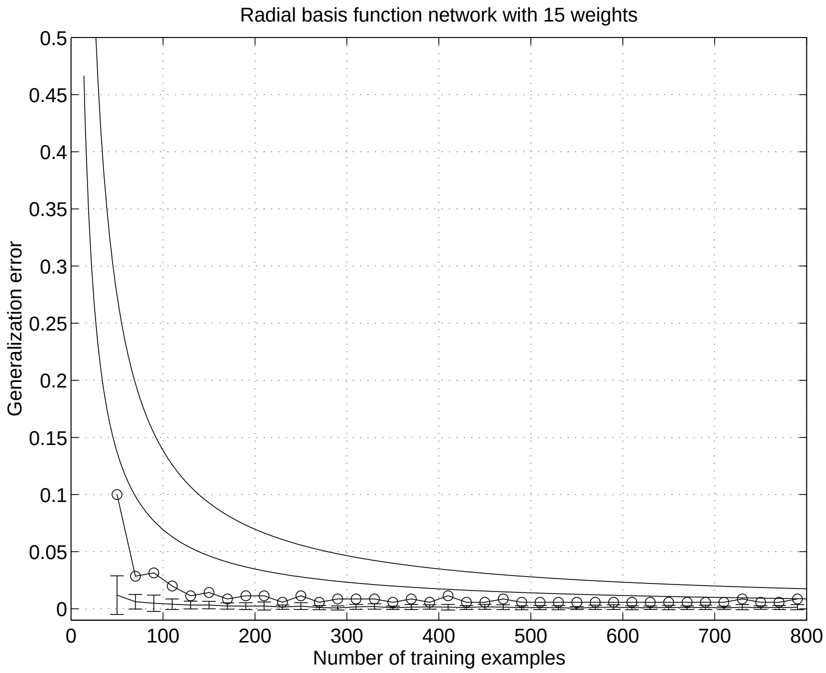

Figure 5: Results obtained for a radial basis function network with 15 weights. The plots are as described in figure 2.

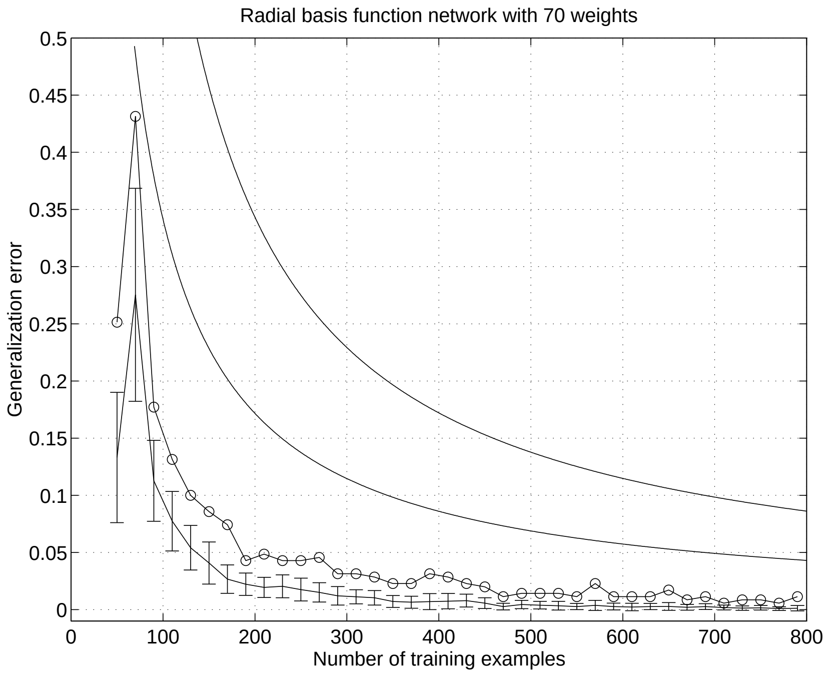

Figure 6: Results obtained for a radial basis function network with 70 weights. The plots are as described in figure 2.

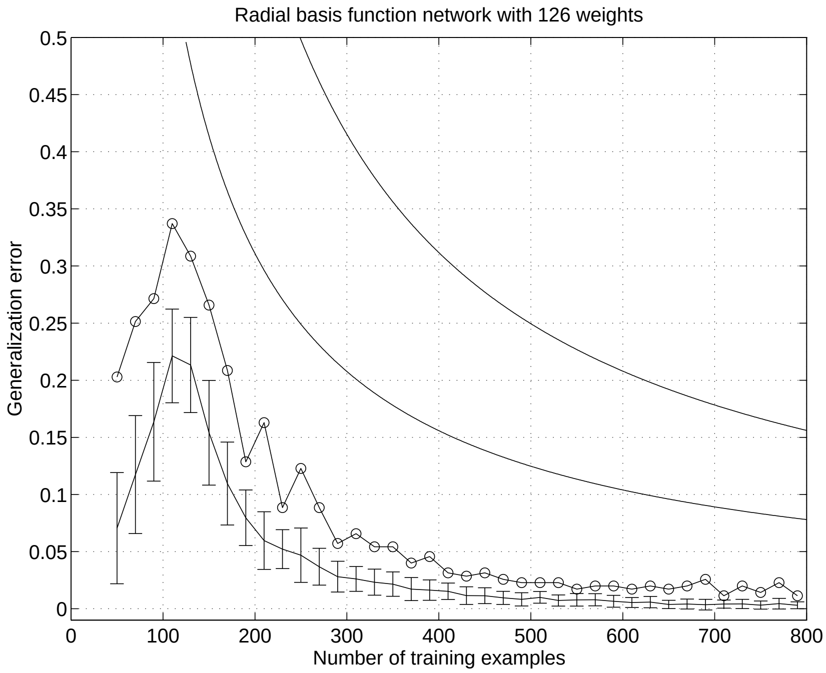

Figure 7: Results obtained for a radial basis function network with 126 weights. The plots are as described in figure 2.

5 Discussion

As a result of the specific assumptions involved, the theory described in section 2 appears to be quite difficult to interpret in any truly practical sense. In particular, the fact that we must assume that the classifier implements a Bayes optimal classification algorithm after training, that the available examples are noise free, and that VCdim($\mathcal{G}$) is known are all significant shortcomings of the present theory, and should be addressed.

5.1 Optimal Classification Algorithms and Noise Free Data

The first of these assumptions was mentioned above: it is unlikely in practice that it will be possible to implement exactly the Bayes optimal classification algorithm studied by Haussler *et al.* (1994). In our experiments we have attempted to solve this problem, and to use an approach more similar to that generally used in practice, by using a standard error minimization technique. As argued above, we consequently expect our measured generalization performances to be worse in general than those that could be obtained using the Bayes optimal classification algorithm.

The assumption that data is noise free is more problematic. It is highly unlikely to be a fully valid assumption in practice. Even in the case of the data used in the experiments described herein, which was collected with significant care, it is unlikely to be a completely valid assumption (Peterson and Barney (1952) and Nowlan (1994)). However, a simple intuitive argument regarding this problem is as follows: if we make the assumption when it is not in fact the case we are likely to overfit the data and consequently increase the generalization error obtained.

As a result of these two considerations we can therefore expect that the actual generalization errors measured are worse than those that would be obtained using a Bayes optimal classification algorithm with truly noise free data. This is important as the theoretical bounds of equations 4 and 5 nonetheless apply in all our experiments, and this suggests that these bounds may in fact overestimate expected generalization error to a greater extent than that suggested directly by our experimental results. (Note however that, as a result of considerations discussed in the next subsection, it is not certain that the results can be interpreted in this manner.)

There are also two further reasons for drawing this conclusion. First, as noted above, we force training and testing sets to be disjoint. Secondly, and again as noted above, our training technique does not guarantee to learn correctly all the examples in each T_k. If at any time this is the case then we obtain a measured generalization error corresponding to a network that learns *exactly* some subset of T_k.

5.2 Knowing the Target Class

The assumption that we have some knowledge of VCdim($\mathcal{G}$) is possibly the most important shortcoming of the current theory because, as noted above, it is highly unlikely to be a good assumption in practice. (This problem, and the related problem of choosing $\mathcal{P}$, are obviously quite similar to the ubiquitous problem of choosing a prior over weight vectors in the standard Bayesian treatments of learning, see for example Buntine and Weigend (1991).)

In fact, the assumption that in practice we will encounter target functions g_T drawn from a class $\mathcal{G}$ does not itself accurately model the actual situation that we generally encounter when designing a pattern classifier. Although the assumption that g_T is some member of a class $\mathcal{G}$ is a good one if we wish to consider *general* learning algorithms, that work in a variety of different circumstances, it is more usual that we approach a *specific* problem, that is, we wish only to learn some *specific* g_T. This is precisely the case in this article, and was discussed above. We might therefore expect that we can assume *any* $\mathcal{G}$ such that $g_T \in \mathcal{G}$ and further assume that $\mathcal{F} = \mathcal{G}$. Our experimental results suggest that this may in fact be a good strategy. The assumption that $\mathcal{F} = \mathcal{G}$ seems reasonable as the Bayes optimal classification algorithm must itself know $\mathcal{G}$ in order to make a prediction (although its hypothesis is not necessarily a member of $\mathcal{G}$), and the assumption that $g_T \in \mathcal{G} = \mathcal{F}$ also seems reasonable because all our classifiers can learn exactly *all* the data that is available to us. A theoretical result relevant to this problem can be found in Haussler *et al.* (1990) (theorem 4.1 of that article). This result upper bounds a particular measure of generalization performance for a consistent classifier using an expression that depends on the VC dimension of $\mathcal{F}$ and is independent of the characteristics of $\mathcal{G}$.

It is important to note that this problem has two main consequences in the context of this article. The first is that there is some uncertainty regarding how the theoretical bounds should be placed in relation to the experimental results. Our conclusion that these bounds are better than earlier ones is still highly likely to be sound, simply as a result of the *degree* of improvement observed (see Cohn and Tesauro (1992)). Also, this observation serves to accentuate the difficulty of applying this theory to practical classifier design.

5.3 Further Experiments

Further experiments would now be useful in order to investigate these bounds further. In particular, experiments using a larger set of data would be interesting, as well as useful in the sense that they would allow generalization errors to be calculated using a set of more than 350 test examples. Unfortunately, the requirement that all training examples are learnt exactly makes experiments using large sets of real data difficult. It would also be interesting, in the case of radial basis function networks with fixed centres, to examine the effect of using a different set of randomly chosen centres each time a network is trained, rather than using the same set for an entire experiment. We have not examined this approach as the time required to perform an experiment in this case is likely to become prohibitive. Finally, it would be interesting to investigate the extent to which the assumptions of the theory can be violated before the bounds become invalid. For example, how good are the bounds for cases where the training set cannot be learned perfectly?

6 Conclusion

In this article we have addressed the question of whether some recent bounds on the sample complexity of the task of training a pattern classifier such that it performs valid generalization can be used as a practical design tool. The bounds considered, although they are probably the most 'practical' available at present within the general framework of computational learning theory, require us to make several assumptions that will not in general be accurate in practice. In particular, it is necessary to assume that our classifier implements a Bayes-optimal classification algorithm, that all data is noise free, and that the VC dimension of the class $\mathcal{G}$ of target functions is known. The last of these assumptions forms at present the most important shortcoming of this theory. The need to make these assumptions makes it rather difficult to fully assess the bounds or to apply them in the design of practical pattern classifiers. At present, the only conclusion that can be drawn regarding the use of these bounds in practice is that they appear to provide an approximate, probably pessimistic guide to expected generalization error, and appear therefore to be applicable in certain circumstances as an initial aid to design. In the experiments performed the bounds were also found to be valid for worst case generalization error in most cases.

This conclusion is a rather pessimistic one. However we note, finally, that these bounds are still rather more practically applicable, although unfortunately less powerful, than earlier bounds obtained in computational learning theory, and that they therefore provide an excellent starting point for further research.

7 Acknowledgements

Thanks are due to Martin Anthony for his comments on the initial draft of this article, and for many useful discussions. Thanks are also due to the reviewers for various helpful comments. This research was supported by SERC research grant #GR/H16759.

A Experimental Results Obtained using the Alternative Example Selection Technique

Figures 8 to 13 are exactly analogous to figures 2 to 7, the only difference being that in producing these figures the alternative method for selecting examples was used. This method was described in section 3. The centres used by the radial basis function networks were identical to those used in the experiments described above.

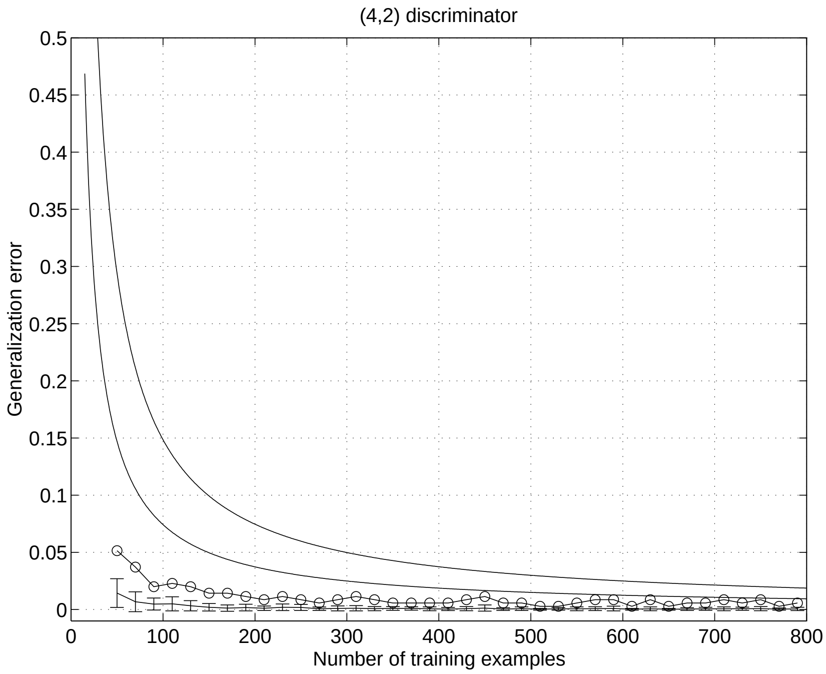

Figure 8: Results obtained for a $(4,2)$ discriminator. The plots are as described in figure 2.

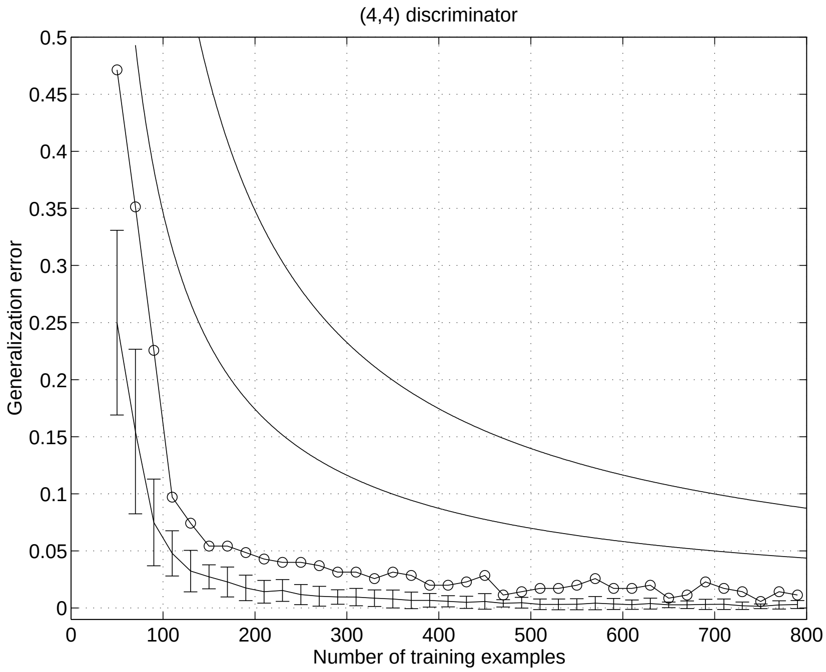

Figure 9: Results obtained for a $(4,4)$ discriminator. The plots are as described in figure 2.

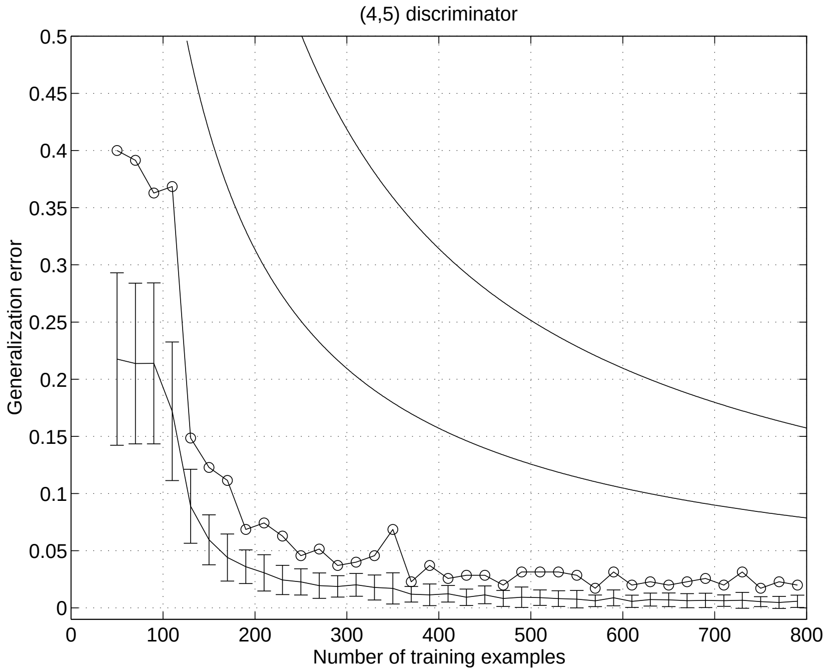

Figure 10: Results obtained for a $(4, 5)$ discriminator. The plots are as described in figure 2.

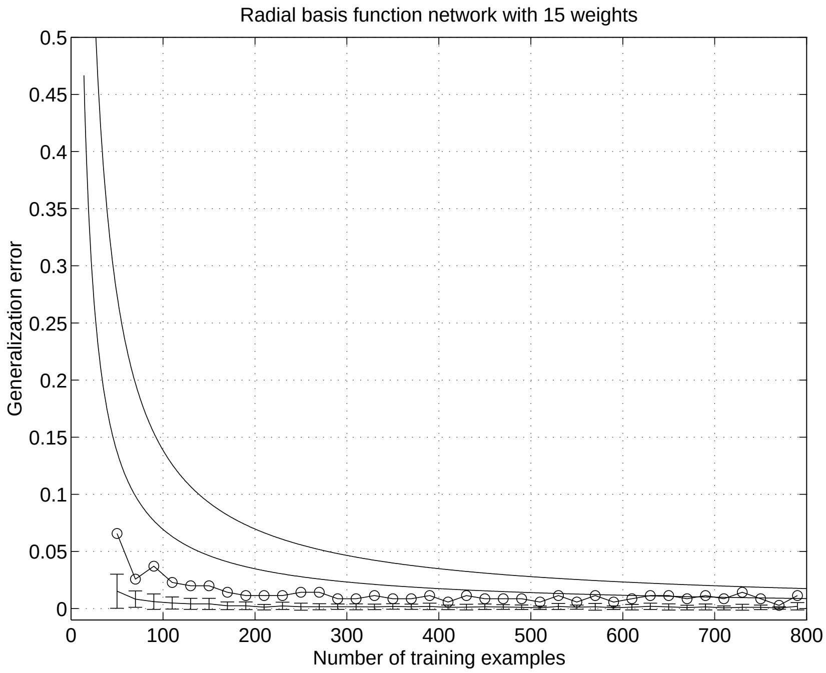

Figure 11: Results obtained for a radial basis function network with 15 weights. The plots are as described in figure 2.

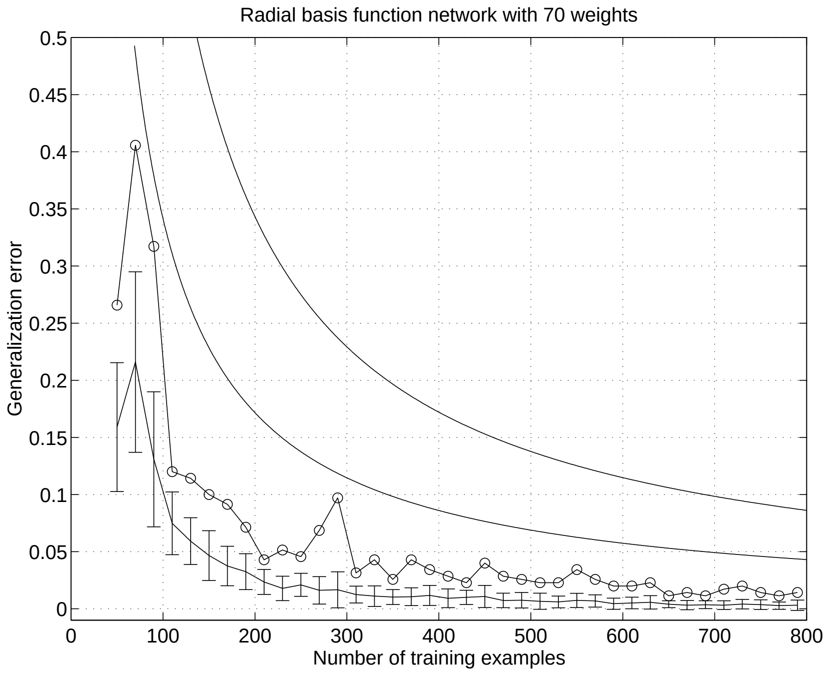

Figure 12: Results obtained for a radial basis function network with 70 weights. The plots are as described in figure 2.

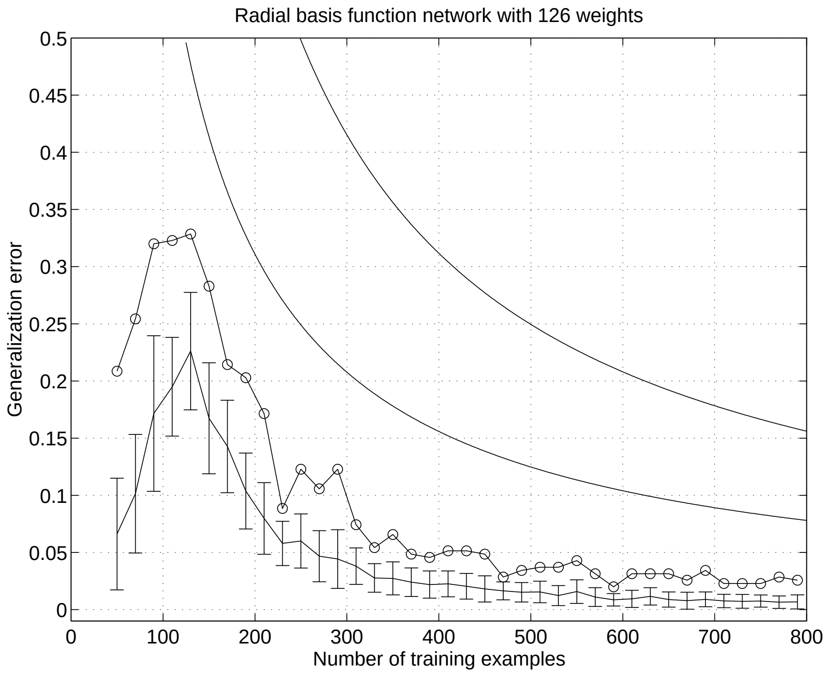

Figure 13: Results obtained for a radial basis function network with 126 weights. The plots are as described in figure 2.

References

[AB92] Martin Anthony and Norman Biggs. *Computational Learning Theory.* Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1992.

[AH93] Martin Anthony and Sean B. Holden. On the power of polynomial discriminators and radial basis function networks. In *Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory*, pages 158–164, July 1993.

[AH94] Martin Anthony and Sean B. Holden. Quantifying generalization in linearly weighted neural networks. *Complex Systems*, 1994. To appear.

[Bar92] Peter L. Bartlett. Lower bounds on the Vapnik-Chervonenkis dimension of multi-layer threshold nets. Technical Report IML92/3, University of Queensland, Department of Electrical Engineering, Intelligent Machines Laboratory, September 1992.

[BCV94] L. Bottou, C. Cortes, and V. Vapnik. On the effective VC dimension. Unpublished manuscript, February 1994.

[BEHW89] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36(4):929–965, October 1989.

[BH89] Eric B. Baum and David Haussler. What size net gives valid generalization? *Neural Computation*, 1:151–160, 1989.

[BW91] Wray L. Buntine and Andreas S. Weigend. Bayesian back-propagation. *Complex Systems*, 5:603–643, 1991.

[Cov65] Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14:326–334, 1965.

[CT92] David Cohn and Gerald Tesauro. How tight are the Vapnik-Chervonenkis bounds? *Neural Computation*, 4(2):249–269, March 1992.

[DH73] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.

[Gis90] Herbert Gish. A probabilistic approach to the understanding and training of neural network classifiers. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1361–1364, 1990.

[GL89] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* Johns Hopkins, second edition, 1989.

[GVB+92] I. Guyon, V. Vapnik, B. Boser, L. Bottou, and S. A. Solla. Structural risk minimization for character recognition. In *Advances in Neural Information Processing Systems*, volume 4, pages 471–479. Morgan Kaufmann Publishers, INC., 1992.

[HKS94] David Haussler, Michael Kearns, and Robert Schapire. Bounds on the sample complexity of Bayesian learning using information theory and the VC dimension. *Machine Learning*, 14:83–113, 1994.

[HLW90] D. Haussler, N. Littlestone, and M. K. Warmuth. Predicting $\{0,1\}$-functions on randomly drawn points. Technical Report UCSC-CRL-90-54, Computer Research Laboratory, Applied Sciences Building, University of California, Santa Cruz, Santa Cruz, CA 95064, December 1990.

[Hol92] Sean B. Holden. Neural networks and the VC dimension. In *Proceedings of the IMA International Conference on Mathematics in Signal Processing*, 1992.

[Hol93] Sean B. Holden. *On the Theory of Generalization and Self-Structuring in Linearly Weighted Connectionist Networks*. PhD thesis, Cambridge University Engineering Department, September 1993. Cambridge University Engineering Department Report number CUED/F-INFENG/TR.161.

[HR94] Sean B. Holden and Peter J. W. Rayner. Generalization and PAC learning: Some new results for the class of generalized single layer networks. *IEEE Transactions on Neural Networks*, 1994. To appear.

[Maa93] Wolfgang Maass. Bounds for the computational power and learning complexity of analog neural nets. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on the Theory of Computing*, pages 335–344, 1993.

[Nat91] Balas K. Natarajan. *Machine Learning: A Theoretical Approach*. Morgan Kaufmann Publishers INC, 1991.

[Nil65] Nils J. Nilsson. *Learning Machines. Foundations of Trainable Pattern-Classifying Systems*. McGraw-Hill, 1965.

[Now94] S. J. Nowlan, 1994. Private communication.

[PB52] G. E. Peterson and H. L. Barney. Control methods used in a study of the vowels. *Journal of the Acoustical Society of America*, 24:175–184, 1952.

[Son92] Eduardo D. Sontag. Feedforward nets for interpolation and classification. *Journal of Computer and System Sciences*, 45(1):20–48, August 1992.

[STA91] John Shawe-Taylor and Martin Anthony. Sample sizes for multiple-output threshold networks. *Network*, 2:107–117, 1991.

[Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.

[Wan90] Eric A. Wan. Neural network classification: A Bayesian interpretation. *IEEE Transactions on Neural Networks*, 1(4):303–305, December 1990.

[WD81] R. S. Wenocur and R. M. Dudley. Some special Vapnik-Chervonenkis classes. *Discrete Mathematics*, 33:313–318, 1981.

[WRB93] Timothy L. H. Watkin, Albrecht Rau, and Michael Biehl. The statistical mechanics of learning a rule. *Rev. Mod. Phys.*, 65(2):499–556, April 1993.