Neural Network Learning: Testing Bounds on Sample Complexity

Joaquim Marques de Sá[1], Fernando Sereno[2], Luís Alexandre[3]

INEB – Instituto de Engenharia Biomédica
[1] Faculdade de Engenharia da Universidade do Porto/DEEC, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
jmsa@fe.up.pt
[2] Escola Superior de Educação, Rua Dr Roberto Frias, 4200-465 Porto, Portugal
[3] Universidade da Beira Interior, Dep. Informática, Rua Marquês de Ávila e Bolama 6201-001 Covilhã, Portugal

Abstract. Several authors have theoretically determined distribution-free bounds on sample complexity. Formulas based on several learning paradigms have been presented. However, little is known on how these formulas perform and compare with each other in practice. To our knowledge, controlled experimental results using these formulas, and comparing of their behavior, have not so far been presented. The present paper represents a contribution to filling up this gap, providing experimentally controlled results on how simple perceptrons trained by gradient descent or by the support vector approach comply with these bounds in practice.

1 Introduction

Sample complexity formulas based on statistical learning theory and the probably approximately correct (PAC) learning paradigm can be found in [2], [3], [4]. These formulas assume a worst-case setting of the learning task. Other authors [6], on the contrary, presented a formula adequate to an average-learning setting. We are interested in assessing the quality of these formulas when using supervised classification with neural networks, based on the empirical error minimization (ERM) principle.

We thus assume that our NN represents a mapping $f: X \to T$ of an object space, X ($X \subseteq \Re^d$), into a target space T. For simplicity reasons, we only consider dichotomic decisions with $T = \{0, 1\}$. We denote by $f(x,w)$ the NN output using some weight vector $w \in W$, of a weight vector space W.

The NN learning task consists of the minimization of a risk functional:

$$R(w) = \int Q(z,w)\,dF(z), \quad w \in W \quad \text{with} \quad Q(z,w) = \begin{cases} 0 & t = f(x,w) \\ 1 & t \neq f(x,w) \end{cases}$$

for the dichotomic data classification task, where $z = (x,t)$ are data pairs and $F(z)$ is the data distribution. $R(w)$ is then the NN probability of error, $P_e(w)$.

In practice, the NN is designed so to minimize the empirical error, i.e., based on a training set $D_n = \{z_i = (x_i, t_i): x_i \in X, t_i \in T, i = 1,2,\ldots,n\}$, we pick up the mapping (by suitable NN weight adjustment) that minimizes

$$R_{\text{emp}}(w) = \frac{1}{n}\sum_{i=1}^{n} Q(z_i, w) \ .$$

The minimum empirical risk occurs for a weight vector w_n. $R_{\text{emp}}(w_n)$ is then the resubstitution estimate of the probability of error, $\hat{P}_e$, of the NN classifier.

Statistical Learning Theory [9] shows that the necessary and sufficient condition for the consistency of the ERM principle, independently of the probability distribution (independently of the problem to be solved), is:

$$\lim_{n\to\infty} \frac{G(n)}{n} = 0$$

where $G(n)$, called *growth function*, can be shown to either satisfy

$$G(n) = n \ln 2 \qquad \text{if } n \le h$$

or to be bounded by:

$$G(n) \le \ln\left(\sum_{i=0}^{h} \binom{n}{i}\right) \le h\left(1 + \ln\frac{n}{h}\right) \qquad \text{if } n > h.$$

The parameter h, called the *Vapnik-Chervonenkis dimension* (or *VC-dimension* for short) is the maximum number of vectors $z_1,\ldots, z_h$, which can be separated in all 2^h possible ways using the family of functions $f(z)$ implemented by the neural network (*shattered* by that family). As an example, consider a single perceptron with $x \in \Re^2$ that uses a step function as activation function. The perceptron implements a line in 2-dimensional space. Therefore, the determination of h amounts to determining the maximum cardinality of a set of points that can be separated in all 2^h possible ways by any line. In this simple case $h = 3$. For more complex situations there are no exact formulas of h and one has to be content with bounding formulas.

Let C be a class of possible dichotomies of the data inputs. The neural network attempts to learn a dichotomy $c \in C$ using a function from a family of functions F implemented by the neural network. Using an appropriate algorithm, the neural network learns $c \in C$ with an error probability (true error rate) e ($\equiv P_e$) with confidence d. A theorem presented in [3] establishes the PAC-learnability[1] of the class C only if its VC

[1] C is PAC-learnable by an algorithm using D_n and Φ, if for any data distribution and $\forall\ \varepsilon, \delta$, $0 < \varepsilon, \delta < 0.5$ the algorithm is able to determine a function $f \in \Phi$, such that $P(P_e(f) \le \varepsilon) \ge 1-\delta$, in polynomial time in $1/\varepsilon$, $1/\delta$, n and size(c). (See [1], [5], [7])

dimension, h, is finite. Moreover, it establishes the following sample complexity bounds:

1. Upper bound: For $0 < e < 1$ and sample size at least

$$n_u = \max\left[\frac{4}{e}\log_2\left(\frac{2}{d}\right), \frac{8h}{e}\log_2\left(\frac{13}{e}\right)\right], \tag{1}$$

any consistent algorithm is of PAC learning for C.

2. Lower bound: For $0 < e < ½$ and sample size less than

$$n_l = \max\left[\frac{1-e}{e}\ln\left(\frac{1}{d}\right), h(1-2(e(1-d)+d))\right], \tag{2}$$

no learning algorithm is of PAC learning for C.

Fig. 1 illustrates these bounds for several values of the VC-dimension, h, and the dimensionality of the input space, d, when $e = 0.05$ and $d = 0.01$. Notice the large gap between lower and upper bound for small values of h.

Another approach [6] consists on bounding the Average Generalization Error (AGE) of a learning machine, namely a MLP. This bound is obtained not in the context of PAC theory, but from hypothesis testing inequalities. As mentioned before, it is not a worst-case bound but an average-case bound. It has the following expression:

$$AGE < a + \frac{1}{2}\sqrt{\frac{d}{n}}, \tag{3}$$

where a is the training error, d represents the number of adjustable parameters (for an MLP, the number of weights) and m is the number of training points.

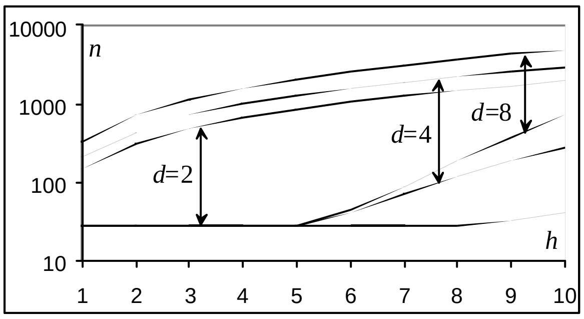

Fig. 1. Bounds of n for $e = 0.05$ and $d = 0.01$.

2 Experimental Setting

In order to test the previous bounds we must use an experimental setting where we have perfect control on the value of h and on the value of the (true) probability of error. These conditions are satisfied for instance for the data distribution depicted in Fig. 2, consisting of uniformly distributed points (x_2, x_1) in $[0, 1]^2$, linearly separated by the $x_2 = x_1$ straight line.

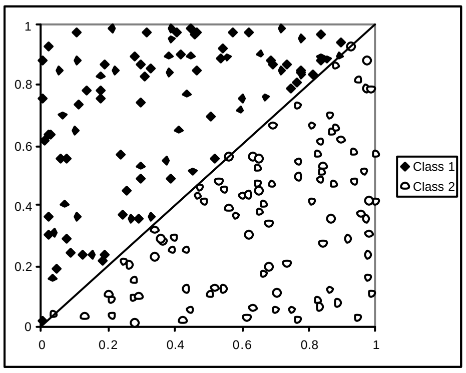

Fig. 2. Example of dataset used in the experiments.

We then know that h = 3, and for any perceptron using a step function - thus, implementing a straight line -, it is an easy task to determine the true error: just determine the areas corresponding to wrongly classified points (remember that the data distribution is uniform).

The experiments were performed as follows: a single 2-input-1-output perceptron with step activation function was trained on a random sample $D_n \subset [0, 1]^2$ until achieving perfect separation on D_n. This experiment was repeated a certain number of times in order to obtain, for each n, the d = 95% percentile of all computed probabilities of error.

Finally, using formula (2) we are able, by a table look-up procedure, to determine the value of e achieving the lower bound of n.

3 Results

For each value of n = 10, 20, ..., 150 we generated 25 random sets, D_n. The (true) error average and upper 95% percentile for the gradient descent training were determined, as shown in Fig. 3: “Average experimental error” and “95% experimental error” curves. Fig. 3 also shows the “95% Theoretical error” predicted by formula (2) and the AGE bound. The theoretical error predicted by formula (1) is largely pessimistic, yielding large values of e ($\approx$ 1) for the represented interval of n.

We repeated these same experiments for a support vector machine with a linear kernel and the same basic architecture (SVM2:1). The results obtained are shown in Fig. 4.

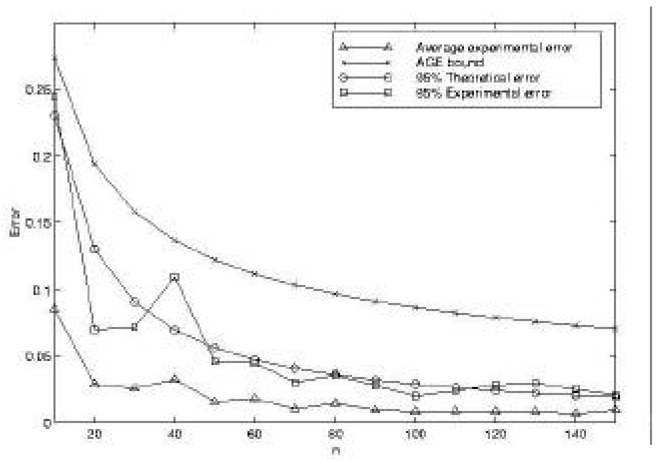

Fig. 3. Error curves for the perceptron, with the AGE bound.

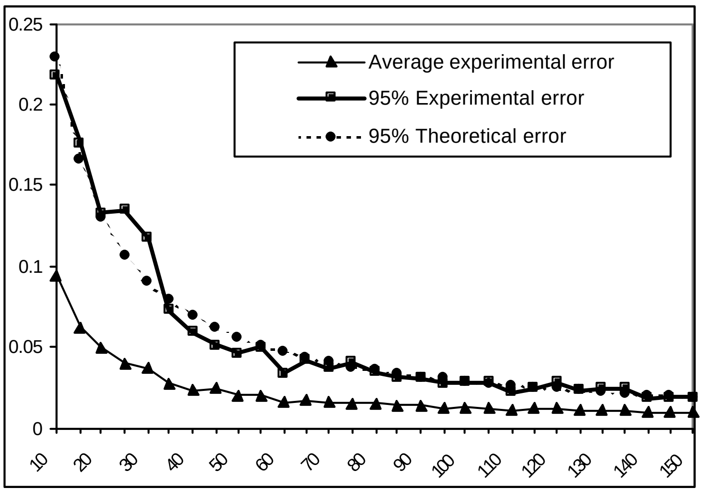

Fig. 3. Error curves for the SVM2:1.

4 Discussion

The results of the previous section show that the lower bound (2) represents a tight bound for the mentioned experiments. As a matter of fact, the "95% Experimental error" and the "95% Theoretical error" curves are very close to each other, namely for the larger values of *n*. The curves also illustrate the well-known *O*(1/*e*) behavior for this learning process [1]. Comparing the SVM approach with the gradient descent approach we see that the first one has smoother convergence. This is also to be expected given the decrease of the VC-dimension for the SVM approach.

The AGE bound in Figure 5 is quite loose if compared with the bound from formula (2). It also exhibits the $O(1/e)$ behavior of the learning process shown by the other bound and the theoretic and measured error rates. So, although the bound from expression (2) is derived in a worst-case scenario, it is in fact tighter than the bound from expression (3). This may not be a fair comparison since expression (2) works for a 95% confidence interval whereas expression (3) should always be valid, hence its larger margin from the average experimental error curve.

We are currently pursuing the experimental test of several sample complexity bounding formulas presented in the literature, using both artificial and real datasets of more complex nature. This raises two added difficulties: how to obtain good estimates of the VC-dimension for spaces of higher dimensionality than $\Re^2$ and using more complex functions than the step function; how to obtain good estimates of the true error of a classifier.

References

1. Anthony, M., Bartlett, P.L.: Neural Network Learning: Theoretical Foundations. Cambridge University Press (1999)
2. Baum, E.B., Haussler, D.: What Size Net Gives Valid Generalization? Neural Computation, 1 (1989) 151-160
3. Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M.K.: Learnability and the Vapnik-Chernovenkis Dimension. J Ass Comp Machinery, 36 (1989) 929-965
4. Ehrenfeucht, A., Haussler, D., Kearns, M., Valiant, L.: A General Lower Bound on the Number of Examples Needed for Learning. Information and Computation, 82 (1989) 247-261
5. Kearns, M.J., Vazirani, U.V.: An Introduction to Computational Learning Theory. The MIT Press (1997)
6. Gu, H., Takahashi, H.: Towards more practical average bounds on supervised learning. IEEE Trans. Neural Networks, 7 (1996) 953-968
7. Mitchell, T.M.: Machine Learning. McGraw Hill Book Co. (1997)
8. Marques-de-Sá, J.P.: Introduction to Statistical Learning Theory. Part I: Data Classification. http://www.fe.up.pt/nnig/ (2003)
9. Vapnik, V.N.: Statistical Learning Theory. John Wiley & Sons, Inc. (1998)