Perceptron Learning with Discrete Weights

Joaquim Marques de Sá[1,2] and Carlos A. S. Felgueiras[2] *

1- Universidade do Porto, Faculdade de Engenharia, DEEC
Porto - Portugal
jmsa@fe.up.pt

2- INEB – Instituto de Engenharia Biomédica
Porto - Portugal

Abstract. Perceptron learning bounds with real weights have been presented by several authors. In the present paper we study the perceptron learning task when using integer weights in $[-k, k]^{d+1}$. We present a sample complexity formula based on an exact counting result of the finite class of functions implemented by the perceptron, and show that this bound is less pessimistic than existing bounds for the discrete and, in certain conditions, also for the continuous weight cases.

1 Introduction

Learning bounds for general and restricted machine learning models have been presented by several authors. Surveys of these bounds can be found in [1], [2], [3] and [4]. Influential articles on this issue are [5] and [6]. Sample complexity formulas based on such bounds have also been presented for neural networks (NN), assuming either infinite or finite classes of functions implemented by the NN. In the first case, the bounds are based on the VC-dimension [3]; in the second case, simple class cardinality bounds are used [1].

Since any physical NN implementation is discrete and finite it seems worthwhile to study in detail the discrete and finite NN case. In the present paper we only consider a simple perceptron having as input a d-dimensional vector $\mathbf{x}$ plus a bias, and using a linear threshold as activation function at the output. We thus assume that our perceptron implements hyperplanes in $\mathbb{R}^d$ corresponding to a $\phi_{\mathbf{w}} : X \rightarrow T$ mapping from an object space, $X \subseteq \mathbb{R}^d$, into a dichotomic target space T, defined by a weight vector $\mathbf{w}$ from space W. We denote by $\phi(\mathbf{x}, \mathbf{w})$ the perceptron output using some weight vector $\mathbf{w} \in W$.

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

$$R(\phi) = \int Q(z, \mathbf{w}) dF(z),\ \mathbf{w} \in W$$

with

$$Q(z, \mathbf{w}) = \begin{cases} 0 & \text{if} \quad t = \phi(\mathbf{x}, \mathbf{w}) \\ 1 & \text{if} \quad t \neq \phi(\mathbf{x}, \mathbf{w}) \end{cases}$$

where $z = (\mathbf{x}, t)$ are data pairs and $F(z)$ is the data distribution.

*This work was supported by the Portuguese FCT-Fundação para a Ciência e a Tecnologia (project POSI/EIA/56918/2004).

We assume that the perceptron is designed to minimize the empirical error on an n-sized training set $D_n = \{z_i = (\mathbf{x}_i, t_i) : \mathbf{x}_i \in X,\, t_i \in T,\, i = 1, 2, \ldots, n\}$:

$$\hat{R}_n(\phi) = \frac{1}{n} \sum_{i=1}^{n} I_{\{\phi(\mathbf{x}_i) \neq t_i\}} (\mathbf{x}_i)$$

where $I(.)$ is the indicator function.

In the following sections we start by presenting some learning bounds for finite classes of classifiers and then apply these results to the perceptron, using an exact formula for the cardinality of discrete hyperplane spaces derived by one of us [7]. Finally, we derive new sample complexity formulas and compare their performance with competing formulas.

2 Learning Finite Classes

A well-known bound for finite class learning is based on Hoeffding's inequality and the union bound property [1]. For any probability measure P and positive ε and n, we have:

$$P\left(\max_{\phi \in C} \left|\hat{R}_n(\phi) - R(\phi)\right| > \varepsilon\right) \leq 2|C|e^{-2n\varepsilon^2} \tag{1}$$

where $|C|$ is the cardinality of a finite set of classifiers, C.

One can also derive relative learning bounds depending on the true classifier risk, $R(\phi)$, using Bernstein or Bennett inequalities [8]. Given n independent random variables $x_1, \ldots, x_n$ with $|x_i| \leq c$, zero mean and $E\left[x_i^2\right] = \sigma^2$, for any $\varepsilon > 0$ we have:

$$P\left(\frac{1}{n}\left|\sum_{i=1}^{n} x_i\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{n\varepsilon^2}{2\left(\sigma^2 + c\varepsilon/3\right)}\right) \tag{2}$$

$$P\left(\frac{1}{n}\left|\sum_{i=1}^{n} x_i\right| \geq \varepsilon\right) \leq \left(\left(1 + \frac{c\varepsilon}{\sigma^2}\right)^{-\frac{1+c\varepsilon/\sigma^2}{1+c^2/\sigma^2}} \left(1 - \frac{\varepsilon}{c}\right)^{-\frac{1-\varepsilon/c}{1+\sigma^2/c^2}}\right)^n \tag{3}$$

for the Bernstein (2) and Bennett (3) inequalities.

In order to apply this result we notice that:

$$\hat{R}_n(\phi) - R(\phi) = \frac{1}{n} \sum_{i=1}^{n} \left(I_{\{\phi(\mathbf{x}_i) \neq t_i\}} - R(\phi)\right)$$

The random variables $\chi_i = I_{\{\phi(\mathbf{x}_i) \neq t_i\}} - R(\phi)$ have a Bernoulli distribution with:

$$c = 1 - R(\phi); \quad \sigma^2 = R(\phi)\big(1 - R(\phi)\big)$$

Thus, using (2) and (3), the two-sided uniform convergence bound in (1) can be rewritten as

$$P\left(\max_{\phi \in C} \left|\hat{R}_n(\phi) - R(\phi)\right| > \varepsilon\right) \leq 2|C| \exp\left(-\frac{n\varepsilon^2}{2\big(1 - R(\phi)\big)\big(R(\phi) + \varepsilon/3\big)}\right) \tag{4}$$

$$P\left(\max_{\phi\in C}\left|\hat{R}_n(\phi)-R(\phi)\right|>\varepsilon\right)\leq 2|C|\left(\left(1+\frac{\varepsilon}{R}\right)^{-(R+\varepsilon)}\left(1-\frac{\varepsilon}{\tilde{R}}\right)^{\varepsilon-\tilde{R}}\right)^n \quad (5)$$

for the Bernstein and Bennett bounds, respectively. In (5) we have dropped the argument ϕ and defined $\tilde{R}(\phi)=1-R(\phi)$.

As we will see in the next section, the bound (4) performs better than (1), especially for low values of $R(\phi)$ and the bound (5) outperforms both bounds.

3 Discrete Perceptron Sample Complexity

The sample complexity $n_L(\varepsilon,\delta)$ of a learning algorithm, L, is defined as the smallest n such that for given $\varepsilon,\delta\in]0,1[$ (accuracy and confidence, respectively), and all training sets D_n, the algorithm yields a classifier $\phi_n=L(D_n)$ satisfying

$$P\left(R(\phi_n)-\min_{\phi\in C}R(\phi)>\varepsilon\right)\leq\delta$$

for every $n\geq n_L(\varepsilon,\delta)$. We use min(.) instead of inf(.) because we are dealing with a finite set of classifiers.

A Lemma due to Vapnik and Chervonenkis (1974) states that:

$$R(\phi_n^*)-\min_{\phi\in C}R(\phi)\leq 2\sup_{\phi\in C}\left|\hat{R}_n(\phi)-R(\phi)\right|$$

where $\phi_n^*=\arg\min_C\hat{R}_n(\phi)$, the empirical risk minimization (ERM) function. Using this result one can easily express the sample complexity in terms of the bounding n for (4) and (5).

Theorem 1. *Let C be the class of perceptrons with d inputs and integer weights in range $[-k,k]$. Denote the cardinality of C by $N_d(k)$. Then, for any P, ε and δ, the following sample complexity bounds hold:*

$$n_L(\varepsilon,\delta)=\frac{4\left(1-R(\phi_n^*)\right)\left(2R(\phi_n^*)+\varepsilon/3\right)}{\varepsilon^2}\log\left(\frac{2N_d(k)}{\delta}\right) \quad (6)$$

$$n_L(\varepsilon,\delta)=\frac{1}{\left(R+\frac{\varepsilon}{2}\right)\log\left(1+\frac{\varepsilon}{2R}\right)+\left(\tilde{R}-\frac{\varepsilon}{2}\right)\log\left(1-\frac{\varepsilon}{2\tilde{R}}\right)}\log\left(\frac{2N_d(k)}{\delta}\right) \quad (7)$$

using the Bernstein and Bennett's inequalities, respectively.

Proof. Using (4) and (5) one can derive a bounding n for the absolute difference between empirical and true errors. It is then straightforward to derive formula (6) and (7) taking into account the previous Lemma (amounting to a substitution of ε by $\varepsilon/2$). A formula for $N_d(k)$ is presented in the Appendix. □

We proceed to comparing this result with other sample complexity formulas found in the literature. For a perceptron with b-bit weights the following formula can be found in the literature (see e.g. [1]), based on Hoeffding's inequality and the following bound for the cardinality of C, $|C|\leq 2^{b(d+1)}$:

$$n_L(\varepsilon,\delta)\leq\frac{2}{\varepsilon^2}\left(b(d+1)\log 2+\log\left(\frac{2}{\delta}\right)\right)\text{ with }b=\lfloor\log_2(2k+1)\rfloor$$

Alternatively, using our $2N_d(k)$ formula

$$n_L(\varepsilon, \delta) = \frac{2}{\varepsilon^2} \log\left(\frac{2N_d(k)}{\delta}\right) \tag{8}$$

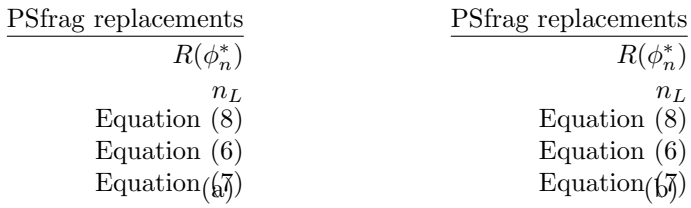


Fig. 1: Sample complexity for $\delta = 0.05$, $\varepsilon = 0.05$, $k = 4$ (a) and $k = 16$ (b)

Figure 1 shows the sample complexity given by formulas (6), (7) and (8) with respect to $R(\phi)$, for two-input ($d = 2$) perceptrons and two distinct values of k. We see that formula (6) yields a smaller bound than formula (8), especially for small values of $R(\phi)$, yet performing badly for large values of $R(\phi)$. The bound (7), based on Bennett's inequality, performs better for all values of $R(\phi)$.

Finally, we compare with the sample complexity bound for the continuous weight perceptron. For this purpose we use a result due to Vapnik and Chervonenkis. Let C be a class of classifiers defined on a set X. Then, for $n > 0$ and $1 > \varepsilon > 0$

$$P\left(\sup_{\phi \in C} \frac{R(\phi) - \hat{R}(\phi)}{\sqrt{R(\phi)}} > \varepsilon\right) < 4 \exp\left(h\left(1 + \log\left(\frac{2n}{h}\right)\right) - \frac{n\varepsilon^2}{4}\right) \tag{9}$$

where h is the VC-dimension of C.

Note that (9) holds simultaneously for all $\phi \in C$, namely ϕ_n^*, the ERM function.

Let $\phi_0 = \arg\inf_{\phi \in C} R(\phi)$. The additive Chernoff bound allows us to state that with probability at least $1 - \delta$ the following inequality holds true:

$$R(\phi_0) = \inf_{\phi \in C} R(\phi) \geq \hat{R}_n(\phi_0) - \sqrt{-\frac{\log \delta}{2n}}$$

In order to obtain sample complexity bounds in terms of $R(\phi_n^*)$ we use:

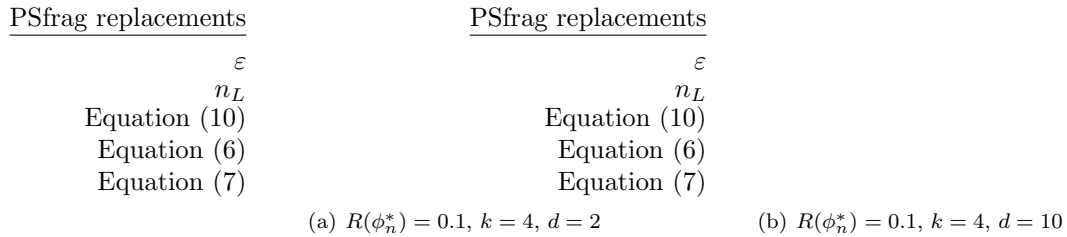
Let C be a class of classifiers with VC-dimension h. Then for any P, n and δ the following holds:

$$P\Big(R(\phi_n^*) - R(\phi_0) \geq \varepsilon(n, \delta, h)\Big) \leq \delta$$

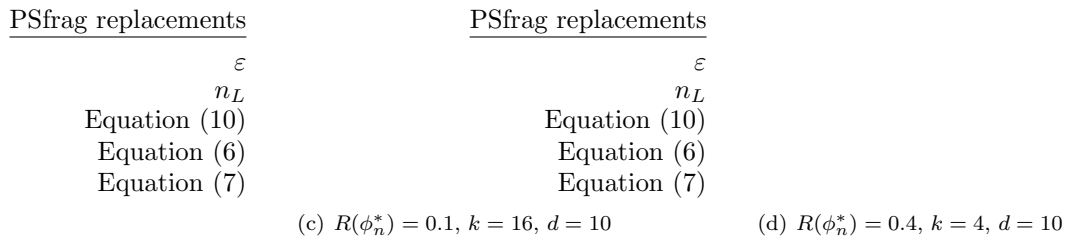
with

$$\varepsilon\left(n,\delta,h\right)=\sqrt{-\frac{\log\delta}{2n}}+2\sqrt{\frac{R(\phi_n^*)\Big(h\big(1+\log(2n/h)\big)-\log(\delta/8)\Big)}{n}} \tag{10}$$

Bounds for the perceptron can now be found since VC dimension is known in this case: $h=d+1$, see e.g. [1], [2], [3]. The sample complexity corresponding to (10) can be numerically obtained by computing the value of n that guarantees an estimation error $\varepsilon(n,\delta,h)$ below a given ε.



(a) $R(\phi_n^*)=0.1$, $k=4$, $d=2$ (b) $R(\phi_n^*)=0.1$, $k=4$, $d=10$



(c) $R(\phi_n^*)=0.1$, $k=16$, $d=10$ (d) $R(\phi_n^*)=0.4$, $k=4$, $d=10$

Fig. 2: Sample complexity for the discrete and continous weights perceptron ($\delta=0.05$).

Figure 2 shows the comparison of formulas (6) and (7) with the sample complexity computed from formula (10) for several values of $R(\phi_n^*)$, k and d.

Consider that for $d=2$, $\delta=0.05$, $\varepsilon=0.025$ and $R(\phi)=0.1$, we wanted to know at what value of k bound (6) is no better than the bound computed from (10). Formula (10) gives the value $n=45083$. Substituting in bound (6), we obtain $N_2(k)=5.58\times10^{16}$. Since $N_2(220000)=3.54\times10^{16}$ and $N_2(260000)=5.84\times10^{16}$ we need $b=19$ bits.

4 Conclusions

We presented sample complexity estimates of a perceptron with discrete weights based on a new and exact result of hyperplane counting. The presented bound performs much better than existing bounds, especially for low values of the number of discrete steps, $2k+1$, and of the true perceptron error.

Using the bound on the rate of relative uniform convergence we derived a sample complexity upper bound for the case of continuous weights, expressed in terms of the true error rate, and proceeded to compare it with our bound. We found that for not too high k (say, $\log_2 k \leq 19$ bits of precision for $d=2$) our bound performs better. As before, the performance improves especially for low values of the number of discrete weights, $2k+1$, and of the true perceptron error.

We are currently investigating the asymptotic properties of our bound and studying its generalization and practical application.

Appendix

The number of distinct hyperplanes in $\mathbb{R}^d$ with integer parameters in $[-k,k]^{d+1}$ is given by [7]:

$$N_d(k) = \frac{1}{2}\sum_{i=0}^{d}\binom{d+1}{i}2^{d+1-i}M_{d+1-i}(k) - 1$$

where $M_n(k)$ is the number of relatively prime $n-$tuples[1] which can be computed using the formula

$$M_n(k) = \sum_{j=1}^{k}\mu(j)\left\lfloor\frac{k}{j}\right\rfloor^n$$

where $\mu(j)$ is the Möbius function.

References

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

[2] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.

[3] Vladimir N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc, 1998.

[4] M. Vidyasagar. *Learning and Generalization*. Springer-Verlag, 2003.

[5] 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, 1989.

[6] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100:78–150, 1992.

[7] C. A. S. Felgueiras. Counting hyperplanes with discrete coefficients, 2004. http://www.fe.up.pt/∼casf/papers/counting.pdf.

[8] V. V. Petrov. *Limit Theorems of Probability Theory*. Oxford University Press, 1995.

[1] A $n-$tuple $(a_1,\ldots,a_n)$ is relatively prime iff $\gcd(a_1,\ldots,a_n)=1$