VC Dimension Bounds for Product Unit Networks*

Michael Schmitt

Lehrstuhl Mathematik und Informatik, Fakultät für Mathematik
Ruhr-Universität Bochum, D–44780 Bochum, Germany
http://www.ruhr-uni-bochum.de/lmi/mschmitt/
mschmitt@lmi.ruhr-uni-bochum.de

Abstract

A product unit is a formal neuron that multiplies its input values instead of summing them. Furthermore, it has weights acting as exponents instead of being factors. We investigate the complexity of learning for networks containing product units. We establish bounds on the Vapnik-Chervonenkis (VC) dimension that can be used to assess the generalization capabilities of these networks. In particular, we show that the VC dimension for these networks is not larger than the best known bound for sigmoidal networks. For higher-order networks we derive upper bounds that are independent of the degree of these networks. We also contrast these results with lower bounds.

1 Introduction

The most widely used type of neuron model in artificial neural networks is a summing unit which computes a weighted sum of its input values and applies an activation function that yields its output. Examples of this neuron type are the threshold and the sigmoidal unit. There is some agreement among neural network researchers that the summing operation only partially describes the way how biological neurons compute. A simple and obvious extension is to allow multiplicative interactions among neurons. There is sufficient evidence from neurobiology showing that multiplicative-like operations are an essential feature of single neuron computation (see, e.g., Koch and Poggio, 1992; Mel, 1994).

A neuron model that uses multiplication of input values as a basic operation is the higher-order neuron, also known as sigma-pi unit. Higher-order neural networks have been successfully used in many learning applications (see, e.g., Lee et al., 1986; Giles and Maxwell, 1987; Perantonis and Lisboa, 1992). A problem that is well known with higher-order neurons is the combinatorial explosion of higher-order terms. To overcome this deficiency Durbin and Rumelhart (1989) introduced a new neuron model that is able to learn higher-order terms: The product unit multiplies its weighted input values and the weights are variable exponents. Product unit networks were further studied by Leerink et al. (1995) and found to be computationally more powerful than sigmoidal networks in many learning applications.

Here we investigate the complexity of learning for networks containing product units. In particular, we derive bounds for the sample complexity in terms of the Vapnik-Chervonenkis (VC) dimension. The VC dimension is a combinatorial parameter that is well known to yield asymptotically tight bounds on the number of examples required for training neural networks to generalize well (see, e.g., Haussler, 1992; Maass, 1995; Anthony and Bartlett, 1999). We show for several types of networks containing product units that the VC dimension can be bounded by a small polynomial in terms of the number of weights and the number of units. Furthermore, we establish bounds for networks of higher-order neurons that do not involve any bound on the degree. Bounds previously shown for higher-order networks all require that the degree is restricted (Anthony, 1993; Bartlett et al., 1998; Goldberg and Jerrum, 1995). We also derive some lower bounds for product unit networks.

*Work supported in part by the ESPRIT Working Group in Neural and Computational Learning II, NeuroCOLT2, No. 27150.

The paper is organized as follows: In the next section we give formal definitions of product unit networks and the VC dimension. Section 3 contains the main results which are upper bounds for various types of networks containing product units. Finally, we present lower bounds and conclude with some remarks.

2 Neural Networks with Product Units

We consider feedforward networks containing product units, where a product unit has the form

$$x_1^{w_1} x_2^{w_2} \cdots x_p^{w_p}$$

Here $x_1, \ldots, x_p$ are the input variables and $w_1, \ldots, w_p$ are the weights of the unit. This type of unit has been introduced by Durbin and Rumelhart (1989) to take into account possible nonlinear interactions among neurons which are not modeled by standard summing units. Product units are used in networks where they occur together with other types of units, such as threshold or sigmoidal units. Both the threshold and the sigmoidal unit are summing units which calculate a weighted sum $w_0 + w_1 x_1 + \cdots w_p x_p$ of their inputs and apply a so-called activation function to this sum. The threshold unit uses the sign function with $\text{sign}(y) = 1$ if $y \geq 0$, and 0 otherwise, whereas the standard sigmoidal unit employs the logistic function $\sigma(y) = 1/(1 + e^{-y})$ as activation function. Another frequently used summing unit is the linear unit which simply outputs the weighted sum.

The standard architecture containing product units is a network with one hidden layer of product units and one sigmoidal output unit. Experiments by Durbin and Rumelhart (1989) and Leerink et al. (1995) showed that this architecture is sufficiently powerful to solve many well-studied problems using less neurons than networks with summing units. Theoretical results also show that networks with one hidden layer of product or sigmoidal units can approximate any continuous function arbitrarily well (Leshno et al., 1993). It is obvious that two subsequent layers of product units can be replaced by one layer. Similarly it can be seen that a network consisting solely of product units is not more powerful than a single product unit. There may be reasons, however, to use pure product unit networks instead of a single product unit, for instance when the degree of multiplicative interaction of the product units, i.e. their fan-in, is restricted by some value smaller than the number of inputs of the network. Here we consider networks that may consist of summing and product units and we do not impose any restriction on the fan-in of these units. Therefore, we may always suppose that a product unit feeds its outputs to some summing units. This allows it also to assume without loss of generality that each product unit has some additional weight by which its output value is multiplied. In summing units this weight is known as the bias or threshold.

We now give the definition of the VC dimension. We call a partition of a set $S \subseteq \mathbb{R}^n$ into two disjoint subsets (S_0, S_1) a *dichotomy*. The dichotomy (S_0, S_1) is said to be *induced* by a set $\mathcal{F}$ of functions that map $\mathbb{R}^n$ to $\{0, 1\}$ if there is some $f \in \mathcal{F}$ such that $f(S_0) \subseteq \{0\}$ and $f(S_1) \subseteq \{1\}$. Further, S is *shattered* by $\mathcal{F}$ if $\mathcal{F}$ induces all dichotomies of S. The *Vapnik-Chervonenkis (VC) dimension* of $\mathcal{F}$, denoted $\text{VCdim}(\mathcal{F})$ is the largest number m such that there exists a set of cardinality m that is shattered by $\mathcal{F}$.

Given some architecture with n input neurons and one output neuron we associate with it a set of functions mapping $\mathbb{R}^n$ to $\{0, 1\}$. This is the set of functions obtained by assigning all possible values to the parameters of the architecture. If the output neuron is a threshold unit, the functions are $\{0, 1\}$-valued as required in the definition of the VC dimension. To comply with the definition in the case that the output unit computes a real-valued function we require that the output of the network is thresholded at some fixed value, say $1/2$. According to known results about the VC dimension we may always assume without loss of generality that the output unit is a linear unit. We also investigate the VC dimension of sets of networks. In this case the corresponding set of functions is obtained simply by taking the union of the function sets which are associated with each particular network.

Some words are necessary concerning the input domain of product units. A negative number raised to some non-integral power yields a complex number and has no meaning in the reals. A method how to cope with this case was introduced by Durbin and Rumelhart (1989) and also employed by Leerink et al. (1995). They propose to discard the imaginary component and use only the real part for further processing. This, however, implies that the product unit becomes one that uses the cosine activation function which has tremendous consequences for the VC dimension. It is known that the VC dimension of a single neuron that uses the sine or cosine activation function is infinite (Sontag, 1992; Anthony and Bartlett, 1999). Therefore, no finite VC dimension bounds exist in general for networks containing such units. We show in the following that product unit networks that operate in the nonnegative domain of the reals nevertheless have a VC dimension that is comparable to networks of sigmoidal gates. Thus, if no additional precautions are taken to guarantee low VC dimension, product unit networks should operate in the nonnegative domain only, which can be achieved, e.g., by transforming the input vectors. In this paper we assume that the inputs to product units are all nonnegative, i.e. from the set $\mathbb{R}^+ = \{x \in \mathbb{R} : x \geq 0\}$.

3 Upper Bounds

First we consider the most widely used architecture, i.e. networks with one hidden layer.

Theorem 1. *Let $\mathcal{N}$ be a network with one hidden layer of h product units and W weights. Then the VC dimension of $\mathcal{N}$ restricted to $\mathbb{R}^+$ is at most $(hW)^2 + 8hW \log(3W)$.*

Proof. Let S be an arbitrary finite subset of $(\mathbb{R}^+)^n$ where n is the number of input units of $\mathcal{N}$. We proceed as follows: First we derive in terms of n, h and the cardinality of S an upper bound for the number of dichotomies that $\mathcal{N}$ induces on S. Then assuming that S is shattered we obtain the bound on the VC dimension. The main idea is to write the function of $\mathcal{N}$ as a polynomial involving exponentials of the network parameters and then to use a bound due to Karpinski and Macintyre (1997) on the number of connected components arising from sets of such polynomials.

Let m be the cardinality of S. When applied to some input vector $(s_1, \ldots, s_n) \in S$ the network $\mathcal{N}$ results in a function of its parameters that can be written as a sum of terms where each term has the form $v s_{i_1}^{w_{j_1}} \cdots s_{i_p}^{w_{j_p}}$. Here v is the corresponding weight of the output unit that is associated with the product unit. The term is 0 if one of the $s_{i_1}, \ldots, s_{i_p}$ is 0. We intend to write each non-zero term as an exponential of a linear function in the network parameters. To accomplish this, we use an idea from Karpinski and Macintyre (1997) and divide the parameter domain for each output unit weight v into three components corresponding to $v < 0, v = 0$, and $v > 0$. Thus we can switch to new parameters $v' = \ln(-v)$ if $v < 0$, $v' = v$ if $v = 0$, and $v' = \ln v$ if $v > 0$. Having this done for all output unit weights we obtain a partition of the parameter domain of $\mathcal{N}$ into 3^h components within each of which we are now able to write every product unit term as

$$e^{v' + w_{j_1} \ln s_{i_1} + \cdots + w_{j_p} \ln s_{i_p}}$$

if $v' \neq 0$ and $s_{i_1}, \ldots, s_{i_p} \neq 0$; otherwise the term is 0. The partition of the parameter domain into 3^h components obtained so far does not depend on the specific choice of $(s_1, \ldots, s_n) \in S$. In case that some s_i is 0, every product unit containing s_i outputs 0. This is taken into account by those components where the corresponding v is 0. Therefore, it suffices to consider the same 3^h components for all elements of S.

Each of these connected components may be further partitioned due to the fact that different dichotomies of S can be induced using suitable values for the remaining parameters. Anthony and Bartlett (1999) showed that if a set of functions $\mathcal{F}$ on l variables is closed under addition of constants and has the so-called solution set components bound B, then the number of dichotomies induced on a set of cardinality m is at most $B(em/l)^l$ for $m \geq l$. Note that the set of functions arising from $\mathcal{N}$ is closed under addition since the output unit is a linear unit with bias. According to Karpinski and Macintyre (1997) for polynomials involving exponentials that are linear in the parameters the bound B satisfies

$$B \leq 2^{ql(ql-1)/2} \cdot d^l \cdot (dl + l + 1)^l [l(dl + l + 1)]^{ql}$$

where q is the number of exponentials and d is the degree of the polynomials. For $\mathcal{N}$ we have $q = h$ and $d = 1$ so that we can simplify this to

$$B \leq 2^{hl(hl-1)/2} \cdot (2l + 1)^{(h+1)l} \cdot l^{hl}.$$

If S is shattered there must be at least 2^m connected components implying that $2^m \leq B(em/l)^l 3^h$. Using the above inequality for B and taking logarithms we have

$$m \leq hl(hl - 1)/2 + (h + 1)l \log(2l + 1) + hl \log l + l \log(em/l) + h \log 3.$$

Employing the well-known estimate $l \log m \leq m/2 + l \log(2l/(e \ln 2))$ (see, e.g., Anthony and Bartlett, 1999) we finally derive that $m \leq (hl)^2 + 8hl \log(3l)$. Since the number l of variables is equal to the number W of weights we have that the cardinality of S is at most $(hW)^2 + 8hW \log(3W)$ as stated. □

Since a network with n input units and one hidden layer of h product units has $h + 1$ weights at the output unit and hn weights at the hidden units we can now, as it is common for networks with one hidden layer, give a bound on the VC dimension also in terms of the number of input units.

Corollary 2. *Let $\mathcal{N}$ be a network with n input units restricted to $\mathbb{R}^+$ and one hidden layer of h product units. Then* $\mathrm{VCdim}(\mathcal{N}) \leq (h^2(n+1) + h)^2 + 8(h^2(n+1) + h) \log(3h(n+1) + 3)$.

In some learning applications it is common not to fix the number of neurons in advance but to let the networks grow. In this case there is no single but a variety of architectures resulting from the learning algorithm. It is possible to accomodate all these architectures in one large network so that the VC dimension can be bounded in terms of the latter network. However, taking into account the constraint underlying the growth of the network often leads to better bounds. In the following we assume that the growth is limited by a bound on the fan-out of the input units. Such a sparse connectivity has been suggested, e.g., by Lee et al. (1986).

Theorem 3. *Let $\mathcal{P}$ be a class of networks with n input units restricted to $\mathbb{R}^+$ and one hidden layer of product units such that every input unit has fan-out at most r. Then* $\mathrm{VCdim}(\mathcal{P}) \leq (rn(2rn+1))^2 + 8rn(2rn+2)\log(6rn+3)$.

Proof. (Sketch.) Each network in $\mathcal{P}$ has at most rn hidden units and $2rn + 1$ weights. Taking into account that class $\mathcal{P}$ contains at most $(rn)^{rn}$ networks we obtain the bound then by counting the number of dichotomies arising from these networks in analogy to Theorem 1. □

A higher-order sigmoidal unit computes by applying the standard sigmoidal function to a weighted sum of monomials. A monomial is a particular product unit where the exponents are restricted to the integers. Schmitt (1999) established bounds for classes of higher-order neurons with restricted input fan-out. The upper bound given there required a bound on the maximum exponent occurring in the monomials. The following result is worse in terms of r and n, but still polynomial; it shows, however, that the VC dimension is finite even when there is no bound on the exponents.

Theorem 4. *Let $\mathcal{H}$ be a class of higher-order sigmoidal units on n inputs in $\mathbb{R}$ such that each input unit has fan-out at most r. Then* $\mathrm{VCdim}(\mathcal{H}) \leq (rn(2rn+1))^2 + 8rn(2rn+2)\log(6rn+3)$.

Proof. (Sketch.) We have to take negative inputs into account. A negative input modifies the network function (in terms of its parameters) only if the exponent of the input variable is odd. In this case the sign of the product unit changes. In a higher-order sigmoidal unit with input fan-out at most r there are at most rn hidden units. Therefore, each element of S gives rise to at most 2^{rn} functions with different signs of the product units. Thus, using $m2^{rn}$ in place of m in the proof of Theorem 1 we get the claimed result. □

We finally consider arbitrary feedforward architectures. The following result shows that the bound for sigmoidal networks in Theorem 8.13 of Anthony and Bartlett (1999) remains valid if the network contains both sigmoidal and product units.

Theorem 5. *Let $\mathcal{N}$ be a neural network of k units where each unit is a sigmoidal or a product unit and let W be the number of weights. Suppose that the product units only receive values from $\mathbb{R}^+$. Then* $\mathrm{VCdim}(\mathcal{N}) \leq (kW)^2 + 11kW \log(18k^2W)$.

Proof. (Sketch.) We argue as in the proof of Theorem 8.13 of Anthony and Bartlett (1999) and use the fact that a product unit contributes to the number of dichotomies not more than a sigmoidal unit. □

From this result we obtain a bound for networks consisting of higher-order sigmoidal units. Here we use that the number of product units in such a network cannot be larger than the number of weights.

Corollary 6. *Let $\mathcal{N}$ be a network of k higher-order sigmoidal neurons and let W be the number of weights. Then the VC dimension of $\mathcal{N}$ restricted to $\mathbb{R}^+$ is at most $(kW + W^2)^2 + 11(k+W)W\log(18(k+W)^2W)$.*

4 Lower Bounds

For computations on $\mathbb{R}^+$ a product unit is at least as powerful as a monomial. Therefore, lower bounds on the VC dimension for networks of monomials or higher-order units imply lower bounds for product unit networks. Koiran and Sontag (1997) constructed networks consisting of linear and multiplication units with VC dimension quadratic in the number of weights. Hence, Corollary 1 of Koiran and Sontag (1997) implies the following:

Corollary 7. *For every W there is a network with $O(W)$ weights that consists only of linear and product units and has VC dimension W^2.*

Bartlett et al. (1998) generalized this result and obtained a lower bound for layered sigmoidal networks in terms of the number of weights and the number of layers. Using their method and the construction of Koiran and Sontag (1997) we get the following bound:

Corollary 8. *For every L and sufficiently large W there is a network with L layers and $O(W)$ weights that consists only of linear and product units and has VC dimension at least $\lfloor L/2 \rfloor \lfloor W/2 \rfloor$.*

Finally, we employ a bound established in Schmitt (1999) to show that there is a super-linear lower bound on the VC dimension for a class of product unit networks with one hidden layer.

Corollary 9. *The class of networks with one hidden layer of product units and n input units, where each input unit has fan-out at most 1, has VC dimension $\Omega(n \log n)$.*

5 Conclusions

We have studied the generalization capabilities of neural networks containing product units as computational elements. The results were given in terms of bounds for the VC dimension. These bounds can now be used to estimate the number of examples required to obtain low generalization errors in product unit networks. All upper bounds we presented are small polynomials in terms of the relevant parameters. In particular, the upper bound for networks of product and sigmoidal units is not larger than the best known bound for purely sigmoidal networks. Thus, from the point of view of the sample complexity there seem to be no disadvantages when replacing summing by product units. Although there has still much research to be done regarding algorithms for learning in product unit networks, the results shown here are an encouraging step forward.

References

Anthony, M. (1993). Classification by polynomial surfaces. *Discrete Applied Mathematics*, 61:91–103.

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

Bartlett, P. L., Maiorov, V., and Meir, R. (1998). Almost linear VC dimension bounds for piecewise polynomial networks. *Neural Computation*, 10:2159–2173.

Durbin, R. and Rumelhart, D. (1989). Product units: A computationally powerful and biologically plausible extension to backpropagation networks. *Neural Computation*, 1:133–142.

Giles, C. L. and Maxwell, T. (1987). Learning, invariance, and generalization in high-order neural networks. *Applied Optics*, 26:4972–4978.

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

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

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

Koch, C. and Poggio, T. (1992). Multiplying with synapses and neurons. In McKenna, T., Davis, J., and Zornetzer, S., editors, *Single Neuron Computation*, chapter 12, pages 315–345. Academic Press, Boston, Mass.

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

Lee, Y. C., Doolen, G., Chen, H. H., Sun, G. Z., Maxwell, T., Lee, H., and Giles, C. L. (1986). Machine learning using a higher order correlation network. *Physica D*, 22:276–306.

Leerink, L. R., Giles, C. L., Horne, B. G., and Jabri, M. A. (1995). Learning with product units. In Tesauro, G., Touretzky, D., and Leen, T., editors, *Advances in Neural Information Processing Systems 7*, pages 537–544, MIT Press, Cambridge, Mass. Extended version: Product unit learning. Technical Report UMIACS-TR-95-80, University of Maryland, 1995.

Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6:861–867.

Maass, W. (1995). Vapnik-Chervonenkis dimension of neural nets. In Arbib, M. A., editor, *The Handbook of Brain Theory and Neural Networks*, pages 1000–1003. MIT Press, Cambridge, Mass.

Mel, B. W. (1994). Information processing in dendritic trees. *Neural Computation*, 6:1031–1085.

Perantonis, S. J. and Lisboa, P. J. G. (1992). Translation, rotation, and scale invariant pattern recognition by high-order neural networks and moment classifiers. *IEEE Transactions on Neural Networks*, 3:241–251.

Schmitt, M. (1999). VC dimension bounds for higher-order neurons. In *Proceedings of the 9th International Conference on Artificial Neural Networks*, volume 2, pages 563–568, IEE Conference Publication No. 470, Institution of Electrical Engineers, London.

Sontag, E. (1992). Feedforward nets for interpolation and classification. *Journal of Computer and System Sciences*, 45:20–48.