Product Unit Neural Networks with Constant Depth and Superlinear VC Dimension*

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. It has remained an open question whether there exist product unit networks with constant depth that have superlinear VC dimension. In this paper we give an answer by constructing two-hidden-layer networks with this property. We further show that the pseudo dimension of a single product unit is linear. These results bear witness to the cooperative effects on the computational capabilities of product unit networks as they are used in practice.

1 Introduction

Product units are formal neurons that are different from the most widely used neuron types in that they multiply their inputs instead of summing them. Furthermore, their weights operate as exponents and not as factors. Product units have been introduced by Durbin and Rumelhart [2] to allow neural networks to learn multiplicative interactions of arbitrary degree. Product unit networks have been proven computationally more powerful than sigmoidal networks in many learning applications by showing that they solve problems using less units than networks with summing units. Furthermore, the success of product unit networks has manifested itself in a rich collection of learning algorithms ranging from local ones, like gradient descent, to more global ones, such as simulated annealing and genetic algorithms [3, 5].

In this paper we investigate the Vapnik-Chervonenkis (VC) dimension of product unit networks. The VC dimension, and the related pseudo dimension, is well established as a measure for the computational diversity of analog computing mechanisms. It is further known to yield bounds for the complexity of learning in various well-studied models such as agnostic [1] and on-line learning [7]. For a large class of neuron types a superlinear VC dimension has been established for networks, while being linear for single units [4, 6, 8]. This fact gives evidence that the computational power of these neurons is considerably

* Work supported by the ESPRIT Working Group in Neural and Computational Learning II, NeuroCOLT2, No. 27150. This is a longer (9-page) version of a paper that appears in the Proceedings of the International Conference on Artificial Neural Networks ICANN 2001. Complete proofs also appear in [9].

enhanced when they cooperate in networks. Using methods due to Koiran and Sontag [4], who considered sigmoidal networks, it is not hard to show that there exist product unit networks with superlinear VC dimension. The drawback with this result is, however, that it requires the networks to have unbounded depth. Such architectures are rarely used in practical applications, where one or two hidden layers are almost always found to be satisfactory.

We show here that constant-depth product unit networks can indeed have superlinear VC dimension. In particular, we construct networks with two hidden layers of product and linear units that have VC dimension $\Omega(W \log k)$, where W is the number of weights and k the number of network nodes. The result is obtained by first establishing a superlinear lower bound for one-hidden-layer networks of a new, sigmoidal-type, summing unit. We contrast these lower bounds by showing that the pseudo dimension, and hence the VC dimension, of a single product unit is no more than linear.

2 Product Unit Networks and VC Dimension

A *product unit* has the form

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

with input variables $x_1, \ldots, x_p$ and real weights $w_1, \ldots, w_p$. The weights are the adjustable parameters of the product unit. (If $x_i < 0$, we require that w_i is an integer in order for the outputs to be real-valued.) In a monomial they are fixed positive integers. Thus, a product unit is computationally at least as powerful as any monomial. Moreover, divisive operations can be expressed using negative weights. What makes product units furthermore attractive is that the exponents are suitable for automatic adjustment by gradient-based and other learning methods.

We consider *feedforward networks* with a given number of input nodes and one output node. A network consisting solely of product units is equivalent to a single product unit. Therefore, product units are mainly used in networks where they occur together with other types of units. Here each non-input node may be a product or a linear unit (computing an affine combination of its inputs). We require that the networks have constant depth, where the *depth* is the length of the longest path from an input node to the output node.

VC dimension and pseudo dimension are defined as follows (see also [1]): A class $\mathcal{F}$ of $\{0,1\}$-valued functions in n variables is said to *shatter* a set of vectors $S \subseteq \mathbb{R}^n$ if $\mathcal{F}$ induces all dichotomies on S (that is, if for every partition of S into two disjoint subsets (S_0, S_1) there is some $f \in \mathcal{F}$ satisfying $f(S_0) \subseteq \{0\}$ and $f(S_1) \subseteq \{1\}$). The *Vapnik-Chervonenkis (VC) dimension* of $\mathcal{F}$ is the cardinality of the largest set shattered by $\mathcal{F}$. If $\mathcal{F}$ is a class of real-valued functions in n variables, the *pseudo dimension* of $\mathcal{F}$ is defined as the VC dimension of the class $\{g : \mathbb{R}^{n+1} \to \{0,1\} \mid \exists f \in \mathcal{F}\, \forall x \in \mathbb{R}^n \forall y \in \mathbb{R} : g(x,y) = \mathrm{sgn}(f(x) - y)\}$ (where $\mathrm{sgn}(z) = 1$ if $z \geq 0$, and 0 otherwise). The VC dimension (pseudo dimension) of a network is defined to be the VC dimension (pseudo dimension) of the set of functions computed by the network with all possible assignments of values to its adjustable parameters. For networks with real-valued output we assume that the output values are mapped to $\{0, 1\}$ using some (non-trivial) threshold. Clearly, the pseudo dimension of a network is not smaller than its VC dimension.

3 Superlinear Lower Bound for Product Unit Networks

We show in this section that product unit networks of constant depth can have a superlinear VC dimension. The following result is the major step in establishing this bound. We note that hidden layers are numbered here from the input nodes toward the output node.

Theorem 1. *Let n, k be natural numbers satisfying $k \leq 2^{n+2}$. There is a network $\mathcal{N}$ with the following properties: It has n input nodes, at most k hidden nodes arranged in two layers with product units in the first hidden layer and linear units in the second, and a product unit as output node; furthermore, $\mathcal{N}$ has $2n\lfloor k/4 \rfloor$ adjustable and $7\lfloor k/4 \rfloor$ fixed weights. The VC dimension of $\mathcal{N}$ is at least $(n - \lfloor \log(k/4) \rfloor) \cdot \lfloor k/8 \rfloor \cdot \lfloor \log(k/8) \rfloor$.*

For the proof we require the following definition: A set of m vectors in $\mathbb{R}^n$ is said to be *in general position* if every subset of at most n vectors is linearly independent. Obviously, sets in general position exist for any m and n. We also introduce a new type of summing unit that computes its output by applying the activation function $\tau(y) = 1 + 1/\cosh(y)$ to the weighted sum of its inputs. (The new unit can be considered as the standard sigmoidal unit with σ replaced by τ). We observe that τ has its maximum at $y = 0$ with $\tau(0) = 2$ and satisfies $\lim \tau(y) = 1$ for $y \to -\infty$ as well as for $y \to \infty$.

Lemma 2. *Let h, m, r be arbitrary natural numbers. Suppose $\mathcal{N}$ is a network with $m + r$ input nodes, one hidden layer of $h + 2^r$ nodes which are summing units with activation function $1 + 1/\cosh$, and a monomial as output node. Then there is a set of cardinality $h \cdot m \cdot r$ that is shattered by $\mathcal{N}$.*

Proof. We choose a set $\{s_1, \ldots, s_{h \cdot m}\} \subseteq \mathbb{R}^m$ in general position and let $e_1, \ldots, e_r$ be the unit vectors in $\mathbb{R}^r$, that is, they have a 1 in exactly one component and 0 elsewhere. Clearly then, the set

$$S = \{s_i : i = 1, \ldots, h \cdot m\} \times \{e_j : j = 1, \ldots, r\}$$

is a subset of $\mathbb{R}^{m+r}$ with cardinality $h \cdot m \cdot r$. We show that it can be shattered by the network $\mathcal{N}$ as claimed.

Assume that (S_0, S_1) is a dichotomy of S. Let $L_1, \ldots, L_{2^r}$ be an enumeration of all subsets of the set $\{1, \ldots, r\}$ and define the function $g : \{1, \ldots, h \cdot m\} \to \{1, \ldots, 2^r\}$ to satisfy

$$L_{g(i)} = \{j : s_i e_j \in S_1\} \ ,$$

where $s_i e_j$ denotes the concatenated vectors s_i and e_j. For $l = 1, \ldots, 2^r$ let $R_l \subseteq \{s_1, \ldots, s_{h \cdot m}\}$ be the set

$$R_l = \{s_i : g(i) = l\} \ .$$

For each R_l we use $\lceil |R_l|/m \rceil$ hidden nodes for which we define the weights as follows: We partition R_l into $\lceil |R_l|/m \rceil$ subsets $R_{l,p}, p = 1, \ldots, \lceil |R_l|/m \rceil$, each of which has cardinality m, except for possibly one set of cardinality less than m. For each subset $R_{l,p}$ there exist real numbers $w_{l,p,1}, \ldots, w_{l,p,m}, t_{l,p}$ such that every $s_i \in \{s_1, \ldots, s_{h \cdot m}\}$ satisfies

$$(w_{l,p,1}, \ldots, w_{l,p,m}) \cdot s_i - t_{l,p} = 0 \quad \text{if and only if} \quad s_i \in R_{l,p} \ . \tag{1}$$

This follows from the fact that the set $\{s_1, \ldots, s_{h \cdot m}\}$ is in general position. (In other words, $(w_{l,p,1}, \ldots, w_{l,p,m}, t_{l,p})$ represents the hyperplane passing through all points in $R_{l,p}$ and through none of the other points.) With subset $R_{l,p}$ we associate a hidden node with threshold $t_{l,p}$ and with weights $w_{l,p,1}, \ldots, w_{l,p,m}$ for the connections from the first m input nodes. Since among the subsets $R_{l,p}$ at most h have cardinality m and at most 2^r have cardinality less than m, this construction can be done with at most $h + 2^r$ hidden nodes.

Thus far, we have specified the weights for the connections outgoing from the first m input nodes. The connections from the remaining r input nodes are weighted as follows: Let $\varepsilon > 0$ be a real number such that for every $s_i \in \{s_1, \ldots, s_{h \cdot m}\}$ and every weight vector $(w_{l,p,1}, \ldots, w_{l,p,m}, t_{l,p})$:

$$\text{If} \quad s_i \notin R_{l,p} \quad \text{then} \quad |(w_{l,p,1}, \ldots, w_{l,p,m}) \cdot s_i - t_{l,p}| > \varepsilon \ .$$

According to the construction of the weight vectors in (1), such an ε clearly exists. We define the remaining weights $w_{l,p,m+1}, \ldots, w_{l,p,m+r}$ by

$$w_{l,p,m+j} = \begin{cases} 0 & \text{if } j \in L_l \ , \\ \varepsilon & \text{otherwise} \ . \end{cases} \tag{2}$$

We show that the hidden nodes thus defined satisfy the following:

Claim 3. *If $s_i e_j \in S_1$ then there is exactly one hidden node with output value 2; if $s_i e_j \in S_0$ then all hidden nodes yield an output value less than 2.*

According to (1) there is exactly one weight vector $(w_{l,p,1}, \ldots, w_{l,p,m}, t_{l,p})$, where $l = g(i)$, that yields 0 on s_i. If $s_i e_j \in S_1$ then $j \in L_{g(i)}$, which together with (2) implies that the weighted sum $(w_{l,p,m+1}, \ldots, w_{l,p,m+r}) \cdot e_j$ is equal to 0. Hence, this node gets the total weighted sum 0 and, applying $1 + 1/\cosh$, outputs 2. The input vector e_j changes the weighted sums of the other nodes by an amount of at most ε. Thus, the total weighted sums for these nodes remain different from 0 and, hence, the output values are less than 2.

On the other hand, if $s_i e_j \in S_0$ then $j \notin L_{g(i)}$ and the node that yields 0 on s_i receives an additional amount ε through weight $w_{l,p,m+j}$. This gives a total weighted sum different from 0 and an output value less than 2. All other nodes fail to receive 0 by an amount of more than ε and thus have total weighted sum different from 0 and, hence, an output value less than 2. Thus Claim 3 is proven.

Finally to complete the proof, we do one more modification with the weight vectors and define the weights for the ouptut node. Clearly, if we multiply all weights and thresholds defined thus far with any real number $\alpha > 0$, Claim 3 remains true. Since $\lim(1 + 1/\cosh(y)) = 1$ for $y \to -\infty$ and $y \to \infty$, we can find an α such that on every $s_i e_j \in S$ the output values of those hidden nodes that do not output 2 multiplied together yield a value as close to 1 as necessary. Further, since $1 + 1/\cosh(y) \geq 1$ for all y, this value is at least 1. If we employ for the output node a monomial with all exponents equal to 1, it follows from the reasoning above that the output value of the network is at least 2 if and only if $s_i e_j \in S_1$. This shows that S is shattered by $\mathcal{N}$. □

We now employ the previous result for the proof of Theorem 1.

Proof (Theorem 1). The idea is to take a set S' constructed as in Lemma 2 and, as shown there, shattered by a network $\mathcal{N}'$ with a monomial as output node and one hidden layer of summing units that use the activation function $1 + 1/\cosh$. Then S' is transformed into a set S and $\mathcal{N}'$ into a network $\mathcal{N}$ such that for every dichotomy (S'_0, S'_1) induced by $\mathcal{N}'$ on S' the network $\mathcal{N}$ induces the corresponding dichotomy (S_0, S_1) of S.

Assume that n and k are given as supposed and let S' be the set defined in Lemma 2 choosing $h = \lfloor k/8 \rfloor$, $m = n - \lfloor \log(k/4) \rfloor$, and $r = \lfloor \log(k/8) \rfloor$. Note that the assumption $k \leq 2^{n+2}$ ensures that $m \geq 0$. Then S' has cardinality

$$m \cdot h \cdot r = (n - \lfloor \log(k/4) \rfloor) \cdot \lfloor k/8 \rfloor \cdot \lfloor \log(k/8) \rfloor \ .$$

Furthermore, we have $m + r = n - 1$ and hence $S' \subseteq \mathrm{IR}^{n-1}$, and Lemma 2 implies that S' is shattered by a network $\mathcal{N}'$ with $n-1$ input nodes, a monomial as output node and one hidden layer of $h + 2^r \leq \lfloor k/4 \rfloor$ summing units with activation function $1 + 1/\cosh$. From S' we construct $S \subseteq \mathrm{IR}^n$ by defining

$$S = \{(e^{s'_1}, \ldots, e^{s'_{n-1}}, e) : (s'_1, \ldots, s'_{n-1}) \in S'\} \ .$$

In other words, S is obtained from S' by appending a component containing 1 to each vector and applying the function $y \mapsto \exp(y)$ to every component. On some input vector $s' \in S'$ a hidden node of $\mathcal{N}'$ with weight vector w and threshold t computes

$$1 + \frac{1}{\cosh(w \cdot s' - t)} = \frac{\exp(w \cdot s' - t) - \exp(-w \cdot s' + t) + 2}{\exp(w \cdot s' - t) + \exp(-w \cdot s' + t)} \ . \qquad (3)$$

If $s = (s_1, \ldots, s_n)$ is the vector in S obtained from the (unique) vector $s' = (s'_1, \ldots, s'_{n-1})$ in S' then according to the construction of S

$$(s'_1, \ldots, s'_{n-1}, 1) = (\ln(s_1), \ldots, \ln(s_{n-1}), \ln(s_n)) \ ,$$

which implies that an exponential in the right-hand side of (3) with weights w and threshold t yields on input vector s' the same output as a product unit with weights w, t on input vector s. (It is clear now that the reason for appending one component to the vectors in S' was to accommodate the threshold t as a weight in a product unit.) Therefore, the computation of a summing unit with activation function $1 + 1/\cosh$ on $s' \in S'$ can be simulated by feeding the vector $s \in S$ into a network with two hidden layers, where the first layer consists of two product units, the second layer has two linear units, and the output node computes a division. Furthermore, this network of 4 hidden nodes has $2n$ connections with adjustable weights and 7 connections with fixed weights (two for each linear unit, one for the threshold of the linear unit computing the numerator, and two for the division).

Replacing all $\lfloor k/4 \rfloor$ hidden nodes of $\mathcal{N}'$ in this way we obtain the network $\mathcal{N}$ which has at most k hidden nodes arranged in two layers where the first hidden layer consists of product units and the second of linear units. The output node has to compute a product of divisions which can be done by a single product unit. Further, $\mathcal{N}$ has $2n\lfloor k/4 \rfloor$ adjustable and $7\lfloor k/4 \rfloor$ fixed weights. Thus, $\mathcal{N}$ has the claimed properties and shatters S, which has the same cardinality as S'. □

From Theorem 1 we obtain the superlinear lower bound for constant depth product unit networks. (We omit its derivation due to lack of space.)

Corollary 4. *Let n, k be natural numbers where $16 \leq k \leq 2^{n/2+2}$. There is a network of product and linear units with n input units, at most k hidden nodes in two layers, and at most $W = nk$ weights that has VC dimension at least $(W/32)\log(k/16)$.*

4 Linear Upper Bound for Single Product Units

We now show that the pseudo dimension, and hence the VC dimension, of a single product unit is indeed at most linear.

Theorem 5. *The VC dimension and the pseudo dimension of a product unit with n input variables are both equal to n.*

Proof. That n is a lower bound easily follows from the fact that the class of monomials with n variables shatters the set of unit vectors from $\{0, 1\}^n$. We derive the upper bound by means of the pseudo dimension of a linear unit.

Let $(w, x) \mapsto f(w, x)$ be the function computed by a product unit, that is,

$$f(w, x) = x_1^{w_1} x_2^{w_2} \cdots x_n^{w_n} \quad ,$$

where $w_1, \ldots, w_n$ are parameters and $x_1, \ldots, x_n$ are input variables. Consider some arbitrary set $S = \{(s_1, u_1), \ldots, (s_m, u_m)\}$ where $s_i \in \mathbb{R}^n$ and $u_i \in \mathbb{R}$ for $i = 1, \ldots, n$. The pseudo dimension of a product unit is defined as the cardinality of the largest such S that is shattered by functions of the form

$$(w, x, y) \mapsto \operatorname{sgn}(f(w, x) - y)$$

with parameters $w_1, \ldots, w_n$ and input variables $x_1, \ldots, x_n, y$. According to the assumption on the parameters in Section 2, if some input value is negative, its weight may take on integer values only. The sole effect of changing the sign of an input value, therefore, is to possibly change the sign of the output of the product unit. Hence, if we consider the set $S' = \{(s'_1, u_1), \ldots, (s'_m, u_m)\}$, where s'_i arises from s_i by taking absolute values in all components, then the number of dichotomies induced on S is less or equal to the cardinality of the set $T \subseteq \{0,1\}^{2m}$ defined by

$$T = \{(\mathrm{sgn}(f(a, s'_1) - u_1), \ldots, \mathrm{sgn}(f(a, s'_m) - u_m), \\ \mathrm{sgn}(-f(a, s'_1) - u_1), \ldots, \mathrm{sgn}(-f(a, s'_m) - u_m)) : a \in \mathbb{R}^n\} \ .$$

Since we are interested in an upper bound for $|T|$, we may assume without loss of generality that no s'_i has some component equal to 0, because in that case the value $\mathrm{sgn}(f(a, s'_i) - u_i)$ is the same for all $a \in \mathbb{R}^n$. Thus, for inputs from S' the function f can be written as

$$f(w, x) = \exp(w_1 \ln x_1 + \cdots + w_n \ln x_n) \ .$$

Since $f(a, s'_i) > 0$ for every $a \in \mathbb{R}^n$ and $i = 1, \ldots, m$, we may suppose that each of $u_1, \ldots, u_m$ is different from 0. Now, depending on whether $u_i > 0$ or $u_i < 0$, exactly one of

$$\mathrm{sgn}(f(a, s'_i) - u_i)\,,\ \mathrm{sgn}(-f(a, s'_i) - u_i)$$

changes when a varies, while the other one remains constant. Hence, by defining

$$b_i = \begin{cases} 1 \text{ if } u_i > 0 \ , \\ -1 \text{ if } u_i < 0 \ , \end{cases}$$

we can select the varying components and obtain with

$$T' = \{(\mathrm{sgn}(b_1 f(a, s'_1) - u_1), \ldots, \mathrm{sgn}(b_m f(a, s'_m) - u_m)) : a \in \mathbb{R}^n\} \ .$$

a set $T' \subseteq \{0,1\}^m$ which has the same cardinality as T. Consider the function $(w, x) \mapsto g(w, x)$ defined for positive input vectors x as

$$g(w, x) = w_1 \ln x_1 + \cdots + w_n \ln x_n \ ,$$

that is, $g = \ln \circ f$. It is not hard to show now that

$$\mathrm{sgn}(b_i f(a, s'_i) - u_i) = \mathrm{sgn}(b_i g(a, s'_i) - b_i \ln(b_i u_i))$$

for every $a \in \mathbb{R}^n$ and $i = 1, \ldots, m$. (We omit the details due to lack of space.) From this we have that $|T'|$ is not larger than the number of dichotomies induced on the set

$$S'' = \{(b_1 \ln(s'_1), b_1 \ln(b_1 u_1)), \ldots, (b_m \ln(s'_m), b_m \ln(b_m u_m))\}$$

(where logarithms of vectors are taken component-wise) by functions of the form

$$(w, x, y) \mapsto \operatorname{sgn}(w_1 x_1 + \cdots + w_n x_n - y) \ . \tag{4}$$

To conclude the proof, assume that some set of cardinality $n + 1$ is shattered by a product unit with n input variables. Then from the reasoning above we may infer that some set of cardinality $n+1$ is shattered by functions of the form (4). This, however, contradicts the fact that the pseudo dimension of a linear unit with n parameters is at most n (see, e.g., [1, Theorem 11.6]). □

Corollary 6. *The VC dimension and the pseudo dimension of the class of monomials with n input variables are both equal to n.*

5 Conclusions

We have established a superlinear lower bound on the VC dimension of constant depth product unit networks. This result has implications in two directions: First, it gives theoretical evidence of the finding that product unit networks employed in practice are indeed powerful analog computing devices. Second, it has consequences for the complexity of learning using product unit networks. For several models of learnability it is known that the VC dimension yields lower bounds, e.g., on the size of the training set required for low generalization error or the number of correction steps for on-line learning. The result presented here can now directly applied to obtain such estimates. There are, however, models of learning for which better bounds can be obtained in terms of other dimensions, such as, e.g., the fat-shattering dimension and covering numbers. A topic for future research is therefore to determine good bounds for the generalization error of product unit networks in these learning models.

We have also presented here a superlinear lower bound for networks with one hidden layer of a new type of summing unit and have shown that VC and pseudo dimension of single product units are linear. This raises the interesting open problem to determine the VC dimension of product unit networks with one hidden layer. Furthermore, we do not know if the lower bounds given here are tight. Thus far, the best known upper bounds for networks with differentiable activation functions are low order polynomials (degree two for polynomials, four for exponentials). But they even hold for networks of unrestricted depth. The issue of obtaining tight upper bounds for product unit networks seems therefore to be closely related to a general open problem in the theory of neural networks.

References

1. M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations.* Cambridge University Press, Cambridge, 1999.
2. R. Durbin and D. Rumelhart. Product units: A computationally powerful and biologically plausible extension to backpropagation networks. *Neural Computation*, 1:133–142, 1989.

3. A. Ismail and A. P. Engelbrecht. Global optimization algorithms for training product unit neural networks. In *International Joint Conference on Neural Networks IJCNN'2000*, volume I, pages 132–137, IEEE Computer Society, Los Alamitos, CA, 2000.
4. P. Koiran and E. D. Sontag. Neural networks with quadratic VC dimension. *Journal of Computer and System Sciences*, 54:190–198, 1997.
5. L. R. Leerink, C. L. Giles, B. G. Horne, and M. A. Jabri. Learning with product units. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 537–544, MIT Press, Cambridge, MA, 1995.
6. W. Maass. Neural nets with super-linear VC-dimension. *Neural Computation*, 6:877–884, 1994.
7. W. Maass and G. Turán. Lower bound methods and separation results for on-line learning models. *Machine Learning*, 9:107–145, 1992.
8. A. Sakurai. Tighter bounds of the VC-dimension of three layer networks. In *Proceedings of the World Congress on Neural Networks*, volume 3, pages 540–543. Erlbaum, Hillsdale, NJ, 1993.
9. M. Schmitt. On the complexity of computing and learning with multiplicative neural networks. *Neural Computation*, to appear.