Tight Bounds for the VC-Dimension of Feedforward and Recurrent Networks of Piecewise Polynomial Activation Function Units

Akito SAKURAI
Advanced Research Laboratory, Hitachi, Ltd.
Hatoyama, Saitama 350-03, JAPAN
E-mail: sakurai@harℓ.hitachi.co.jp

Abstract. We consider the VC-dimension of a set of the neural networks of depth s with w adjustable parameters that has h hidden units of activation functions of at most q segments of degree at most d polynomials. When $d \geq 2$ and $q \geq 2$, the VC-dimension is $O(ws(s \log d + \log(qh/s)))$, $O(ws((h/s)\log q) + \log d)$, and $\Omega(ws \log(dqh/s))$. When $d \geq 2$ and $q = 1$, it is $\Theta(ws \log d)$. When $d = 1$ and $q \geq 2$, it is $\Theta(ws \log(qh/s))$. When $d = 0$ and $q > 2$, it is $O(ws \log(qh/s))$ and $\Omega(w \log h)$. When $d = 0$ and $q = 2$, it is $\Theta(w \log h)$. The recurrent networks we consider is the feedforward network equiped with feedback connections with unit time delay. Let r be the number of repetitions along the feedbacks. Then the VC-dimension is $O(wrs(\log r + s \log d + \log(qh/s)))$, $O(wrs((h/s)\log q) + \log rd)$, and $\Omega(wrs \log(dqh/s))$ for $d \geq 2$ and $q \geq 2$; $\Theta(wrs \log d)$ for $d \geq 2$ and $q = 1$; $\Theta(wrs \log(qh/s))$ for $d = 1$ and $q \geq 2$; $O(wrs \log(qh/s))$ and $\Omega(wr \log h)$ for $d = 0$ and $q > 2$. $O(wr \log(rh))$, $O(wh)$ (for any r), $\Omega(wr)$ (for $r \leq h$), and $\Omega(wh)$ (for any r) for $d = 0$ and $q = 2$.

1. Introduction

When Vapnik and Chervonenkis developed in [20] the concept that later became called Vapnik-Chervonenkis dimension, or VC-dimension, they dared not dream that it would become as common as is now. After Blumer et al. showed in [3] that the number of samples needed to accomplish certain learning task and that the expected error for unseen data are bounded from above by using the VC-dimension of a hypothesis space, many researchers, including those in neural network fields, have recognized the importance of the VC-dimension.

The VC-dimension is an index to measure the expressive capability of hypothesis space or, in neural network field, that of functional space whose member is a specific neural network with fixed weights and connections. The VC-dimension is roughly the maximum number of input data, most favorably arranged for the networks in the input data space, such that any of the 0–1 valued functions defined on these data (considered as points in the input data space) can be realized by a network with appropriately chosen weights. For networks that output continuous or analog value, the value is thresholded to 1 or 0 before considering the VC-dimension. That is, the network is assumed to be used for classification tasks. Still, the VC-dimension is known to relate to interpolation capability of the original networks.

We had been unable to obtain VC-dimension values for practical types of networks, until fairly tight upper and lower bounds were obtained ([15], [16], and [17]) for linear threshold element networks in which all elements perform a threshold function on weighted sum of inputs. Roughly, the lower bound for the networks is $(1/2)w \log h$ and the upper bound is $w \log h$ where h is the number of hidden elements and w is the number of connecting weights (for one-hidden-layer case $w \approx nh$ where n is the input dimension of the network).

In many applications, though, sigmoidal functions, specifically a typical sigmoid function $1/(1+\exp(-x))$, or for economy of calculation piecewise linear functions, are used instead of the threshold function. This is mainly because the differentiability of the functions is needed to perform backpropagation or other learning algorithms. Unfortunately, we were yet to learn explicit bounds on the VC-dimension for this type of networks, although many related results were getting available in the literature. W. Maass obtained a result for piecewise linear functions networks, for which the VC-dimension is $O(w^2 \log q)$, where q is the number of linear pieces of the function ([12]). Macintyre and Sontag proved that the VC-dimension of neural networks with the sigmoid functions (and other broad types of functions) is finite ([13]). Bartlett and Williamson

proved that $8w\log(11wD)$ is an upper bound on the VC-dimension of the two-layer (*i.e.*, one hidden layer) sigmoid networks with inputs restricted in $\{-D, -(D-1), \ldots, D-1, D\}$, where D is an integer ([2]).

Recently Koiran and Sontag [10] proved a lower bound $\Omega(w^2)$ for the piecewise polynomial case and he claimed that an open problem if there is a mathichg w^2 lower bound for the type of networks after Maass's proof of $O(w^2)$. But we still have something to do, since he showed it only for the case $w = \Theta(h)$ and the number of hidden layers being unbounded; also $O(w^2)$ bound has room to improve.

We in this paper improve the bounds obtained by Maass and Koiran and consequently show the role of polynomials, which can not be played by linear functions, and the role of the constant functions that could appear for piecewise polynomial case, which cannot be played by poynomial functions.

We here summerize our results. For the network with polynomial activation functions, the bound is $\Theta(ws\log d)$ (see [10] for $\Omega(w^2)$ where $s = \Theta(h)$ and $w = \Theta(h)$ are assumed and [12] and [6] for $O(w^2)$ where $s = \Theta(h)$ is assumed). Our bounds hold either for $w = \Theta(h)$, $w = \Theta(h^2)$ or inbetween.. For the network with piecewise polynomial activation functions, the upper bound is $O(ws(s\log d + \log(dqh/s)))$ and $O(ws((h/s)\log q) + \log(dqh/s))$ (see [12] for $O(w^2\log qd)$). More precisely $ws((s/2)\log d + \log(qh/s))$ is an asymptotic upper bound when $(s/2)\log d < (h/s)\log q$ and $ws((h/s)\log q + \log(dqh/s))$ is the one otherwise. The best lower bound we obtained so far is $\Omega(ws\log(dqh/s))$. The bound holds either for $w = \Theta(h)$, $w = \Theta(h^2)$ or inbetween.

The piecewise polynomial activation function case and the threshold activation function case differs because the output of each unit in the linear threshold case can convey only two values, whereas in the polynomial case, it can encode as many valued as needed into the intermediate values which in later layers will be decoded to get output values. The difference between the constant depth and the unbounded depth case is due to the fact that the amount of information that can be decoded from the one obtained from the preceding layers is proportional to the number of layers in the later layers.

The feedback connections in the recurrent networks have an effect of repeting hidden layers. Therefore the VC-dimension of recurrent networks, in many cases, is multiplied by the number of repetitions r. For example, for the polynomial networks the VC-dimension is $\Theta(wrs\log d)$, for the piecewise polynomial networks it is $O(wrs(\log r + s\log d + \log(qh/s)))$, $O(wrs(\log rd + (h.s)\log q))$, and $\Omega(wrs\log(dqh/s))$. The feedforward networks are just special cases for $r = 1$.

The threshold networks are though exceptions. The VC-dimension is, when $r < h$, $O(wr\log rh)$ and $\Omega(wr)$; when $r \geq h$ it is $\Theta(wh)$.

2. Terminology

A *neuron, neuronal unit*, or simply *unit* is a one-output real-valued function which has a special representation as $\alpha(f(\mathbf{w};\mathbf{x}))$, where f is the *input* function and α is the *activation* function. The input function in this paper is the weighted sum of inputs, *i.e.* the inner-product of the weight vector and the input; except for the cases noted otherwise. The activation function is a threshold, a polynomial, or a piecewise polynomial function. We will call the units with these activation functions as threshold, polynomial, or piecewise polynomial units respectively. A *neural network* is a feed-forward network (or circuit) composed of neural units.

The inputs to the network or the output from the network are called *external inputs* or *external output* to distinguish them from intra-network inputs and outputs. The neural networks we deal with in this paper have only one output. The network output value is considered to be thresholded to 1 if the intrinsic output value is positive and 0 otherwise.

The *depth* of a network is the length of the longest path from its external inputs to its external output, where the length is the number of units on the path. Likewise we can assign a *depth* to each unit in a network as the length of the longest path from the external input to the output of the unit, where the length is the number of units on the path. The depth of the external output unit is equal to the depth of the network. A *hidden layer* is a set of units with the same depth other than the depth of the network. Therefore a depth L network has $L-1$ hidden layers.

In many cases $\mathbf{w}$ will stand for a vector composed of all the connection weights in the network (including

threshold values for the threshold units) and w is the length of $\mathbf{w}$. The number of units in the network, excluding "input units," will be denoted by h; in other words, the number of hidden units plus one, or sometimes just the number of hidden units.

A function whose range is $\{0, 1\}$ (a set of 0 and 1) is called a *Boolean-valued function*.

An *N-pointset* is a pointset whose cardinality is N. The cardinality of a set P is represented by $\#(P)$.

Let $\mathcal{F}$ be a set of Boolean-valued functions, and S be an N-pointset in $\mathcal{R}^n$. Set $\Pi_{\mathcal{F}}(S) \stackrel{\text{def}}{=} \{D : D \subset S, \exists f \in \mathcal{F}\ [\forall p \in D\ f(p) = 1 \text{ and } \forall p \in S - D\ f(p) = 0]\}$. If $\Pi_{\mathcal{F}}(S) = 2^S$, that is, if for any dichotomy $S_0 \cup S_1\ (S_0 \cap S_1 = \{\})$ of S there exists some $f \in \mathcal{F}$ such that $f(S_0) = \{0\}$ and $f(S_1) = \{1\}$, S is *shattered by* $\mathcal{F}$. The *VC-dimension* VC-dim($\mathcal{F}$) of a class $\mathcal{F}$ of Boolean-valued functions is the maximum cardinality of the sets that are shattered by $\mathcal{F}$. If we define $\Pi_{\mathcal{F}}(m) = \max(\#(\Pi_{\mathcal{F}}(S)))$ for a positive integer m where S in the expression varies in all possible sets in $\mathcal{R}^n$ with cardinality m, VC-dim($\mathcal{F}$) is the maximum of d which satisfies $\Pi_{\mathcal{F}}(d) = 2^d$. The VC-dimension of a set of networks is the VC-dimension of the set of functions represented by the networks.

When $m \geq d$, $\Phi_d(m) \stackrel{\text{def}}{=} \sum_{i=0}^{d} \binom{m}{i}$, otherwise $\Phi_d(m) \stackrel{\text{def}}{=} 2^m$, assuming $d \geq 0$ and $m \geq 0$. Then we have ([3],[4]):

(i) If VC-dim(H) = d, for any $m \geq 0$, $\Pi_H(m) \leq \Phi_d(m)$.

(ii) For any $m \geq d \geq 1$, $\Phi_d(m) \leq 2(m^d/d!) \leq (em/d)^d$.

(iii) In particular, if T_n is a set of Boolean-valued functions realizable by an n-input threshold unit, its VC-dimension is $n + 1$ and for any $m \geq 0$, $\Pi_{T_n}(m) = 2\Phi_n(m-1) \leq \Phi_{n+1}(m)$.

Throughout this paper we use $\log x$ to stand for $\lfloor \log_2 x \rfloor$.

3. Determination of the VC-dimensions

To determine the VC-dimension, basically we need to calculate the number of implementable Boolean-valued functions on N points with various values of N and geometrical arrangements of the points. Although we want to find the maximum of N as defined in Sec. 2, the geometrical arrangement of points that maximalizes N is in general very hard to obtain. A practical way to circumvent the difficulty is by bounding the VC-dimension from above and below as tightly as possible.

When all the units in the networks of the set output real values, we use a *region counting argument*, developed by Goldberg and Jerrum [6], that goes as follows. Suppose that $f_G(\mathbf{w}; \mathbf{x})$ is the network output function when $\mathbf{x}$ is applied to the network input and $\mathbf{w}$ is the connection weights in the network. From the definition, the VC-dimension of the network, that is, the VC-dimension of the function set $\{f_G(\mathbf{w}; \cdot) \mid \mathbf{w} \in \mathcal{R}^w\}$, is

$$\max\{N \mid 2^N \leq \max_{\mathbf{x}_1, \cdots, \mathbf{x}_N} \#(\{\langle \mathcal{H}(f_G(\mathbf{w}; \mathbf{x}_1)), \ldots, \mathcal{H}(f_G(\mathbf{w}; \mathbf{x}_N)) \rangle \mid \mathbf{w} \in \mathcal{R}^w\}), \tag{3.1}$$

where $\langle\ \rangle$ is an N-tuple, $\mathcal{H}$ is the Heaviside function, and $\#(S)$ is the cardinality of a set S. Since

$$\begin{aligned} &\#(\{\langle \mathcal{H}(f_G(\mathbf{w}; \mathbf{x}_1)), \ldots, \mathcal{H}(f_G(\mathbf{w}; \mathbf{x}_N)) \rangle \mid \mathbf{w} \in \mathcal{R}^w\}) \\ &\qquad \leq \text{ the number of connected regions in } \mathcal{R}^w - \textstyle\bigcup_{i=1}^{N} \{\mathbf{w} \mid f_G(\mathbf{w}; \mathbf{x}_i) = 0\}, \end{aligned} \tag{3.2}$$

except for rare cases such as when there exists $\mathbf{w}_0$ such that $f_G(\mathbf{w}_0; \mathbf{x}_{i_j}) = 0$ $(1 \leq j \leq l)$ and for any $\mathbf{w}$ in any neighborhood of $\mathbf{w}_0$ there exists $1 \leq j \leq l$ such that $f_G(\mathbf{w}; \mathbf{x}_{i_j}) > 0$, exploration of an upper bound of the VC-dimension is reduced to bounding from above the number of connected regions. The problem is further reduced to a simpler one that counts the number of connected components of $\bigcap_{i=1}^{m} \{\mathbf{w} \mid f(\mathbf{w}; \mathbf{x}_i) = 0\}$ by using Warren's theorem [22].

Theorem 3.1. *Let M be a connected topological n-manifold, and let $M_1, \ldots, M_b$ be connected $(n-1)$-manifolds embedded in M so that:*

(1) the M_i are topologically closed and locally flat in M;

(2) the intersection of any given j of the M_i, $1 \leq j \leq n$, is either empty or is an $(n-j)$-manifold that has a finite number of components and is locally flat at the intersection of any $j-1$ of the given M_i;

(3) any intersection of more than n of the M_i is empty.

Let b_j be the number of connected-components among all intersections of any j of the M_i with M.

Under these hypotheses, the set $M - \bigcup_{i=1}^{m} M_i$ *has at most* $\sum_{j=0}^{n} b_j$ *connected components.*

Note that to apply the theorem, the regularity conditionsw (1)–(3) should be satisfied which may not hold in many cases. In the case of polynomials and piecewise polynomials (the number of pieces should be finite), since we have only finite number of continuous components, (1)–(3) are always satisfied by a little purturbation of coefficients appeared in polynomials.

We use the notations $\mathcal{N}(f)$ for the set $\{\mathbf{w} \mid f(\mathbf{w}) = 0\}$.

Theorem 3.2. *The number of connected components* $N_{cc}(\cdot)$ *of* $\mathcal{R}^w - \bigcup_{i=1}^{m} \mathcal{N}(f_i)$ *is bounded from above by*

$$\sum_{j=1}^{w} \binom{m}{j} 2^j \max_{i_k\text{'s}} \{N_{cc}(\bigcap_{k=1}^{j} \mathcal{N}(p_{i_k})) \mid p_{i_k} \text{ is either } f_{i_k} + \alpha_{i_k} \text{ or } f_{i_k} - \alpha_{i_k}\}.$$

This theorem is an almost direct consequence of Theorem 3.1 and the fact that $\mathcal{N}(f_i+\alpha) \cap \mathcal{N}(f_i-\alpha) = \phi$.

Constructive arguments are used to obtain lower bounds. For a certain N, which usually depends on w and h, if we propose a concrete method or prove the existence of a method that can designate the network that implements all the Boolean-valued functions defined on carefully chosen N points; then the N is a lower bound of the VC-dimension of the set.

For the problems tha we deal with in the paper, upperbounds impose no rstrictions on the configuration or the architecture. For example, the upper bounds are independent of the input dimension n and the relation between the number of weights w and the number of hidden units h or between w and n. The input dimension may be constant, w may be $\Theta(h)$, $\Theta(h^2)$, or say $\Theta(h^{1.5})$. When we prove the lower bounds we try to attain them for $n = o(h)$ and for w to be any value between $\Theta(h)$ and $\Theta(h^2)$.

4. UpperbBounds of the VC-dimension of feedforward networks

For polynomial functions we can use the following theorems [22], [14],

Theorem 4.1. *Let* $f_G(\mathbf{w}; \mathbf{x}_i)$ $(1 \le i \le N)$ *be real polynomials in* $\mathbf{w}$*, each of degree* d *or less. The number of connected components of the set* $\bigcap_{i=1}^{m} \{\mathbf{w} \mid f_G(\mathbf{w}; \mathbf{x}_i) = 0\}$ *is bounded from above by* $2(2d)^w$ *where* w *is the length of* $\mathbf{w}$*.*

Lemma. *Let* $f_G(\mathbf{w}; \mathbf{x}_i)$ $(1 \le i \le N)$ *be real polynomials in* $\mathbf{w}$*, each of degree* d *or less. The number of connected regions in* $\mathcal{R}^w - \bigcup_{i=1}^{m} \{\mathbf{w} \mid f_G(\mathbf{w}; \mathbf{x}_i) = 0\}$ *is bounded from above by* $(4emd/w)^w$ *where* w *is the length of* $\mathbf{w}$*.*

The following lemma is convenient to calculate the bounds when Theorem 3.2 applies.

Lemma. *If* $m \ge w(\log C + \log\log C + 2)$*, then* $2^m > \sum_{j=1}^{w} 2^j \binom{m}{j} \cdot C^w$*.*

Proof. Using a formula ([3]): "for any $m \ge w \ge 1$, $\sum_{i=0}^{w} \binom{m}{i} \le 2(m^w/w!) \le (em/w)^w$," we get $\sum_{j=1}^{w} 2^j \binom{m}{j} \cdot C^w < (2emC/w)^w$. Let $m = w(\log C + t + 1)$. Then $m - (w \log(2emC/w)) > w[1 + t - \log(2e/w) - \log(\log C + 1)] > 0$ for $t \ge \log\log C + 1$. □

4.1. Polynomial network case

In this section we consider the polynomial activation function case.

Theorem 4.2. *Suppose that* $q = 1$*, i.e., the activation function is a polynomial of degree at most* d*.* $O(ws \log d)$ *is an upper bound of the VC-dimension for the networks with depth* s*. When* $s = \Theta(h)$ *the bound is* $O(wh \log d)$*. More precisely* $ws(\log d + \log\log d + 2)$ *is an asymptotic upper bound. Note that if we allow a polynomial as the input function,* $d_1 d_2$ *will replace* d *above where* d_1 *is the maximum degree of the input functions and* d_2 *is that of the activation functions.*

The theorem is clear from the facts that the network funtion (f_G in (3.1)) is a polynomial of degree at most $d^s + d^{s-1} + \cdots + d$, Theorem 4.1 and the lemma.

Interestingly, the bound $O(ws \log d)$ is immune to the number of units h except for indirect effects through w, which shows a sharp constrast to $\Theta(w \log h)$ for the linear threshold element case. Let us consider the reason for a while.

Suppose that the functions realized by a set of networks is represented by a function parametrized by connection weights as $f_G(\mathbf{w};\mathbf{x})$. Since roughly speaking the VC-dimension reflects the complexity of these functions with weights as variables, the above bound seems to imply that the complexity of polynomial network is independ of the number of units but dependent only on the number of layers. This comes from the following two facts. One is that in the proof the complexity of polynomial is measured by the degree of the polynomial and not by the number of terms in the polynomial. The other is that increasing the number of layers causes further composition of polynomials and thus increases the degree of the function but increasing the number of units in one layer increases number of terms but not the degree and thus does not increase the complexity measure of the function.

This seems to be an artifact, since we know that at least for the single variable case, the complexity (say the number of roots or the number of optimas) depend on the number of terms and not on the degree.

To eliminate the artifact we need to know the bound that is expressed in terms of the number of monomials that appear in polynomial equations, which will be used to estimate the complexity of the formula representing the function of the set of networks. We have a good estimation by Khovanskii ([9]) but the bound is still not good enough to get satisfactry tight bounds for polynomial cases. By the way, Khovanskii's results are general enough to be applied to standard sigmoidal cases and yield explicit bounds of the VC-dimension of them ([8], [19]).

4.2. Piecewise polynomial networks

For the piecewise polynomial networks, we have two types of bounds. The first one is suitable for bounded depth cases (*i.e.* $s = o(h)$) and the second one for the unbounded depth case (*i.e.* $s = \Theta(h)$).

Theorem 4.3. *Suppose that the activation functions are piecewise polynomials with at most q segments of polynomials degree at most d. $O(ws(s\log d + \log(qh/s)))$ and $O(ws((h/s)\log q + \log d))$ are upper bounds for the VC-dimension, where s is the depth of the network. More precisely, $ws((s/2)\log d + \log(qh/s))$ and $ws((h/s)\log q + \log d)$ are asymptotic upper bounds. Note that if we allow a polynomial as the input function then $d_1 d_2$ will replace d above where d_1 is the maximum degree of the input functions and d_2 is that of the activation functions.*

Note that when $s\log d < (h/s)\log q$ asymptotically the former is better bound and otherwise the latter is since $O(ws(s\log d + \log(qh/s))) = O(ws(s\log d + \log(dqh/s)))$ and $O(ws((h/s)\log q + \log d)) = O(ws((h/s)\log q + \log(dqh/s)))$.

Lemma 4.5. *$N_{cc}(\bigcap_{i=1}^{m}\mathcal{N}(f_G(\mathbf{w};\mathbf{x}_i)))$ is bounded from above by $(4emd^{(s+3)/2}qh/ws)^{ws}$ and $(4emd^s q^h/w)^w$ where d is the maximum of the degree of the polynomials and q is the maximum of the number of the polynomial segments.*

Proof. We have two different ways to calculate the bounds. First

$$
\begin{aligned}
&N_{cc}\left(\mathcal{R}^w - \bigcup\mathcal{N}(f_G(\mathbf{w};\mathbf{x}_i))\right)\\
&\quad\le N_{cc}\left(\mathcal{R}^{w_1} - \bigcup\mathcal{N}(f_{G_1}(\mathbf{w}_1;\mathbf{x}_i))\right)\\
&\qquad\cdot \max N_{cc}\left(\mathcal{R}^{w_1+w_2} - \bigcup\mathcal{N}(f_{G_2}(\mathbf{w}_1\circ\mathbf{w}_2;\mathbf{x}_i))\right)\cdots\max N_{cc}\left(\mathcal{R}^{w_1+\cdots+w_s} - \bigcup\mathcal{N}(f_{G_s}(\mathbf{w}_1\circ\cdots\circ\mathbf{w}_s;\mathbf{x}_i))\right)\\
&\quad\le \left(\frac{4emqh_1 d}{w_1}\right)^{w_1}\cdot\left(\frac{4emqh_2(d+1)d}{w_1+w_2}\right)^{w_1+w_2}\cdots\left(\frac{4emqh_s(d^{s-1}+\cdots+d+1)d}{w_1+\cdots+w_s}\right)^{w_1+\cdots+w_s}\\
&\quad\le \left(\frac{4emqd^s(h/s)}{w}\right)^{ws}.
\end{aligned}
$$

The last inequality comes from the following ones :

$$
\begin{aligned}
d^{w_1}\cdot((d+1)d)^{w_1+w_2}\cdot((d^{s-1}+\cdots+d+1)d)^{w_1+\cdots+w_s}&\\
\le d^{sw_1+(s-1)w_2+\cdots+w_s}\cdot d^{(1+2+\cdots+s)w_1+(2+\cdots+s)w_2+\cdots+sw_s}&\\
\le d^{sw}\cdot d^{(s(s+1)/2)w} = d^{ws(s+3)/2}\qquad\text{for } s\ge 3&\\
(C/w_1)^{w_1}\cdot(C/(w_1+w_2))^{w_1+w_2}\cdot(C/(w_1+\cdots+w_s))^{w_1+\cdots+w_s}&\\
\le (C/w)^{ws}\qquad\text{where } C = 4emq > ew \text{ since } m\ge w&
\end{aligned}
$$

$$h_1^{w_1} \cdot h_2^{w_1+w_2} \cdots h_s^{w_1+\cdots+w_s} \leq (h_1 \cdots h_s)^w \leq (h/s)^{ws}$$

where the property that for $x \geq e$, $x^{1/x}$ is greater than 1 and monotonically decreasing is used.

Secondly

$$N_{cc}\left(\mathcal{R}^w - \bigcup_{i=1}^{w} \mathcal{N}(f_G(\mathbf{w}; \mathbf{x}_i))\right) \leq N_{cc}\left(\mathcal{R}^w - \bigcup_{i=1}^{w}\bigcup_{j=1}^{q^h} \mathcal{N}(f_{G,j}(\mathbf{w}; \mathbf{x}_i))\right) \leq \left(\frac{4emq^h d^s}{w}\right)^w$$

where $f_{G,j}$ is one of the network functions realized by specifying and fixing a polynomial segment for each unit. □

5. Lower bounds for polynomial networks

Theorem 5.1 *Let us consider the case that the activation function are polynomials of degree at most d. $\Omega(ws \log d)$ is a lower bound of the VC-dimension for the networks with depth s. When $s = \Theta(h)$ the bound is $\Omega(wh \log d)$. More precisely, $w(s-4)\log d$ is a lower bound where d is the degree of activation functions and is a power of two. When $d = 2$, the bound improves to $w(s-2)\log d$. In both cases, a few fixed weights are used other than w adjustable ones.*

The proof consists of several lemmas. The network we are constructing will have two parts: an encoder and a decoder. We deliberately fix the N input points. The decoder part has fixed underlying architecture but also fixed connecting weights whereas the encoder part has varied weights so that for any given binary outputs for the N input points such that the decoder outputs the specified value from the codes encoded by the encoder.

First we consider the decoder, which has two real inputs and one real output. One of the two inputs y holds a code of a binary sequence $b_1, b_2, \ldots, b_m$ and the other x holds a code of a binary sequence $c_1, c_2, \ldots, c_m$. The elements of the latter sequence are all 0's except for $c_j = 1$, where $c_j = 1$ orders the decoder to output b_j from it and cosequently from the network.

We show two types of networks; one consists of units with polynomial activation fucntios of degree at most two and has the VC-dimension $w(s-1)$ and the other consists of units with polynomial activation functions of degree d a power of two and has the VC-dimension $w(s-5)\log d$.

We use for convenience two functions $\mathcal{H}_\theta(x) = 1$ if $x \geq \theta$ and 0 otherwize and $\mathcal{H}_{\theta,\phi}(x) = 1$ if $x \geq \phi$, 0 if $x \leq \theta$, and undefined otherwise. Throughout this section we will use a simple logistic function $\rho(x) = (16/3)x(1-x)$ with the following property.

Lemma 5.2. *For any binary sequence $b_1, b_2, \ldots, b_m$, there exists an interval $[x_1, x_2]$ such that $b_i = \mathcal{H}_{1/4,3/4}(\rho^i(x))$ and $0 \leq \rho^i(x) \leq 1$ for any $x \in [x_1, x_2]$.*

The next lemmas are easily proven.

Lemma 5.3. *For any binary sequence $c_1, c_2, \ldots, c_m$ where all but $c_j = 1$ are all 0's, there exists x_0 such that $c_i = \mathcal{H}_{1/4,3/4}(\rho^i(x_0))$. Specifically we will take $x_0 = \rho_L^{-(j-1)}(1/4)$, where $\rho_L^{-1}(x)$ is the inverse of $\rho(x)$ on $[0, 1/2]$. Then $\rho^{j-1}(x_0) = 1/4$, $\rho^j(x_0) = 1$, $\rho^i(x_0) = 0$ for all $i > j$, and $\rho^{j-i}(x_0) \leq (1/4)^i$ for all positive $i \leq j$.*

Proof. Clear from the fact that $\rho(x) \geq 4x$ on $[0, 1/4]$. □

Lemma 5.4. *For any binary sequence $b_1, b_2, \ldots, b_m$, take y such that $b_i = \mathcal{H}_{1/4,3/4}(\rho^i(y))$ and $0 \leq \rho^i(y) \leq 1$ for all i and $x_0 = \rho_L^{-(j-1)}(1/4)$, then $\mathcal{H}_{7/12,3/4}\left(\sum_{i=1}^m \rho^i(x_0)\rho^i(y)\right) = b_j$, i.e. $\mathcal{H}_0\left(\sum_{i=1}^m \rho^i(x_0)\rho^i(y) - 2/3\right) = b_j$.*

Proof. If $b_j = 0$, $\sum_{i=1}^m \rho^i(x_0)\rho^i(y) = \sum_{i=1}^j \rho^i(x_0)\rho^i(y) \leq \rho^j(y) + \sum_{i=1}^{j-1}(1/4)^i < \rho^j(y) + (1/3) \leq 7/12$. If $b_j = 1$, $\sum_{i=1}^m \rho^i(x_0)\rho^i(y) > \rho^j(x_0)\rho^j(y) \geq 3/4$. □

By the above lemmas, the decoder network shown in Figure 1 realizes the following function:

- Suppose that a binary sequence $b_1, \ldots, b_m$ and an integer j is given. Then we can choose y that depends only on $b_1, \ldots, b_m$ and x_0 that depends only on j such that b_j is output from the decoder.

Note that in the network we use $(x+y)^2 - (x-y)^2 = 4xy$ to implement multiplication units by polynomial units.

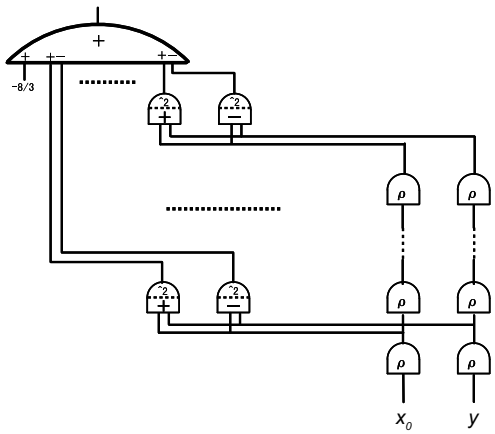

Fig.1. The network architecture of units with polynomial activation functions of degree two.

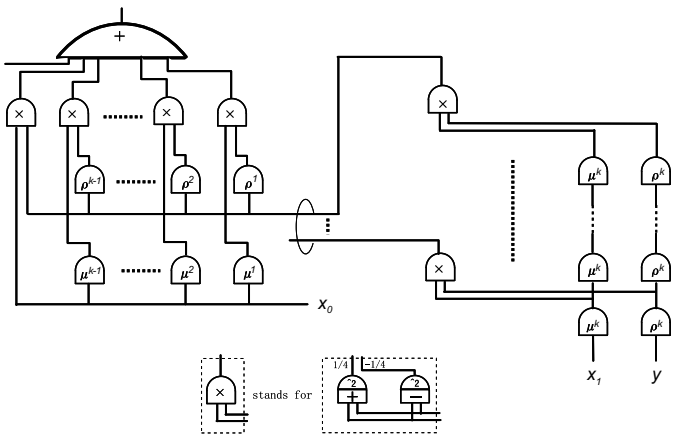

Fig.2. The network architecture of units with polynomial activation functions of degree of powers of two.

For the case of degree of higher than two we have to construct a bit more complicated one by using another simple logistic function $\mu(x) = (36/5)x(1-x)$. The following lemmas hold.

Lemma 5.5. *If* $0 < \rho^i(x) < 1$ *for any* $0 < i \leq l$, *take an* ϵ *such that* $(16/3)^l\epsilon < 1/4$. *Then* $\rho^l(x) - (16/3)^l\epsilon < \rho^l(x+\epsilon) < \rho^l(x) + (16/3)^l\epsilon$.

Proof. There are four cases depending on whether $\rho^{l-1}(x+\epsilon)$ is on the uphill or downhill of ρ and whether x is on the uphill or downhill of ρ^{l-1}. The proofs are done by induction.

First suppose that the two are on the uphills. Then $\rho^l(x+\epsilon) = \rho(\rho^{l-1}(x+\epsilon)) < \rho(\rho^{l-1}(x) + (16/3)^{l-1}\epsilon)) < \rho^l(x) + (16/3)^l\epsilon$. Secondly suppose that $\rho^{l-1}(x+\epsilon)$ is on the uphill but x is on the downhill. Then $\rho^l(x+\epsilon) = \rho(\rho^{l-1}(x+\epsilon)) > \rho(\rho^{l-1}(x) - (16/3)^{l-1}\epsilon)) > \rho^l(x) - (16/3)^l\epsilon$. The other two cases are similar. □

Lemma 5.6. *Take* $x_0 = \mu_L^{-(j-1)}(1/6)$, *where* $\mu_L^{-1}(x)$ *is the inverse of* $\mu(x)$ *on* $[0, 1/2]$. *Then* $\mu^{j-1}(x_0) = 1/6$, $\mu^j(x_0) = 1$, $\mu^i(x_0) = 0$ *for all* $i > j$, *and* $\mu^{j-i}(x_0) = (1/6)^i$ *for all* $i > 0$ *and* $\leq j$.

Proof. Clear from the fact that $\mu(x) \geq 6x$ on $[0, 1/6]$. □

Lemma 5.7. *Suppose that* $k > 0$, $m > 0$, $1 \leq j \leq m$, $1 \leq l \leq k$, *and a binary sequence* $b_1, b_2, \ldots, b_k$, $b_{k+1}, b_{k+2}, \ldots, b_{2k}, \ldots, b_{(m-1)k+1}, \ldots, b_{mk}$ *are given. Take* y *such that* $b_i = \mathcal{H}_{1/4,3/4}(\rho^i(y))$ *and* $0 \leq \rho^i(y) \leq 1$ *for all* i, $x_1 = \mu_L^{-(j-1)}(1/6)$, *and* $x_0 = \mu_L^{-(l-1)}(1/6^k)$. *Then for* $z = \sum_{i=1}^m \rho^{ik}(y)\mu^{ik}(x_1)$, $\mathcal{H}_0\left(\sum_{i=0}^{k-1} \rho^i(z)\mu^i(x_0) - (1/2)\right) = b_{kj+l}$ *holds.*

Proof. We will show that the difference between $\rho^{jk+l}(y)$ and $\sum_{i=0}^{k-1} \rho^i(z)\mu^i(x_0)$ is sufficiently small. Clearly $z = \sum_{i=1}^m \mu^{ik}(x_1)\rho^{ik}(y) = \sum_{i=1}^j \mu^{ik}(x_1)\rho^{ik}(y) \leq \rho^{jk}(y) + \sum_{i=1}^{j-1}(1/6^k)^i < \rho^{jk}(y) + 1/(6^k - 1)$ and $\rho^{jk}(y) < z$. If z is on the uphill of ρ^l then by using the above lemma, we get $\sum_{i=0}^{k-1} \rho^i(z)\mu^i(x_0) = \sum_{i=0}^{l} \rho^i(z)\mu^i(x_0) < \rho^l(z) + 1/(6^k - 1) < \rho^{jk+l}(y) + (1 + (16/3)^l)(1/(6^k - 1)) < \rho^{jk+l}(y) + 1/4$ (note that $l \leq k-1$ and $k \geq 2$). If z is on the downhill of ρ^l then by using the above lemma, we get $\sum_{i=0}^{k-1} \rho^i(z)\mu^i(x_0) = \sum_{i=0}^{l} \rho^i(z)\mu^i(x_0) > \rho^l(z) > \rho^l(\rho^{jk}(y)) - (16/3)^l(1/(6^k - 1)) > \rho^{jk+l}(y) - 1/4$. □

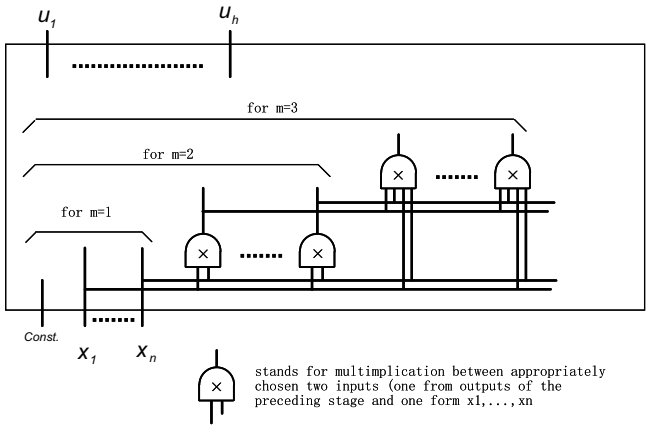

Fig.3. The encoder architecture for polynomial activation functions of degree two or higer.

Next we show the encoding scheme we adopted. We will show the case $w = \Theta(h^2)$ since other cases $w = \Theta(h^\alpha)$ (where $1 \leq \alpha < 2$) are easily obtained from this.

Theorem 5.8 *There is a network $f(\mathbf{w};\mathbf{x})$ with $2n$-dimensional inputs $\mathbf{x}$, $2h$ hidden units, and h^2 weights $\mathbf{w}$ and h^2 sets of input values $\mathbf{x}_1,\ldots,\mathbf{x}_{h^2}$ such that for any set of values $y_1,\ldots,y_{h^2}$ we can choose $\mathbf{w}$ to satisfy $y_i = f(\mathbf{w};\mathbf{x}_i)$.*

Proof. We utilize the fact that monomials obtained by choosing at most m variables from n variables with repetition allowed (say $x_1^2x_2x_6$) are all linearly independent ([1]). Note that the number of monomials thus formed is $\binom{n+m}{m}$. If we choose exactly m variables, then the number is $\binom{n+m-1}{m}$.

Suppose for simplicity that we have $2n$ inputs and $2h$ main hidden units (we have other hidden units too), and $h = \binom{n+m}{m}$. By using multiplication units (in fact each is a composite of two squaring units and the outputs are supposed to be summed up as in Figure 2), we can form $h = \binom{n+m}{m}$ linearly independent monomials composed of varibles $x_1,\ldots,x_n$ by using at most h multiplication units (or no multiplication units are required for $m = 1$). In the same way, we can form h linearly independent monomials composed of varibles $x_{n+1},\ldots,x_{2n}$.

Let us denote the monomials by $u_1,\ldots,u_h$ and $v_1,\ldots,v_h$. We form a subnetwork to calculate $\sum_{j=1}^{h}(\sum_{i=1}^{h} w_{i,j}u_i)v_j$ by using h multiplication units. Clearly the calculated result y is the weighted sum of monomials described above where the weights are $w_{i,j}$ for $1 \leq i,j \leq h$.

Since $y = f(\mathbf{w};\mathbf{x})$ is a linear combination of linearly independent terms, if we choose appropriately h^2 sets of values $\mathbf{x}_1,\ldots,\mathbf{x}_{h^2}$ for $\mathbf{x} = (x_1,\ldots,x_{2n})$, then for any assignment of h^2 values $y_1,\ldots,y_{h^2}$ to y we have a set of weights $\mathbf{w}$ such that $y_i = f(\mathbf{x}_i,\mathbf{w})$. □

Proof of Theorem 5.1. The whole network consists of the decoder and the encoder. The h^2s input points are the Cartesian product of the above $\mathbf{x}_1,\ldots,\mathbf{x}_{h^2}$ and $\{1/4,\rho^{-1}(1/4),\ldots,\rho^{-(s-1)}(1/4)\}$ (or $\{1/6,\mu^{-1}(1/6),\ldots,\mu^{-(s-1)}(1/6)\}$ for the $d > 2$ case) (see Lemma 5.4 and Lemma 5.7). Let y_{h^2j+i} be the desired output on the point $(\mathbf{x}_{i+1},\rho^{-j}(1/4))$ (or $(\mathbf{x}_{i+1},\mu^{-j}(1/6))$) $\{x_0$ for $b_j = 1 \mid 1 \leq j \leq s\}$. $\mathbf{w}$ is decided based upon y_i's and $\mathbf{x}_1,\ldots,\mathbf{x}_{h^2}$ by Theorem 5.8. Then by Lemma 5.4 (or by Lemma 5.7) the whole network consisting of the encoder in Theorem 5.8 and the decoder in Figure 1 (or Figure 2 respectively) outputs y_{h^2j+i} (after being thresholded) for the input $(\mathbf{x}_{i+1},\rho^{-j}(1/4))$ (or $(\mathbf{x}_{i+1},\mu^{-j}(1/6))$ respectively).

Let the depth of the decoder be s. The number of units used for the decoder is $4s+1$ (for the degree 2 case which can decode at most s bits) or $4(s-2)+4(k-1)+1$ (for the degree 2^k case which can decode at most $(s-2)k$ bits). The number of units used for the encoder is less than $5h$. Note that we have constraints on s (which dominates the depth of the network) and h (which dominates the number of units in the network) that $h \leq \binom{n+m}{m}$ and $m = O(s)$ or roughly $\log h = O(s)$ be satisfied.

When $s = o(h)$, by a network of depth s with h units of degree at most 2, we can attain the VC-dimension ws where $w = ((h-s)/5)^2$ approximately; by units of degree at most $d = 2^k$, we can attain the VC-dimension

$ws \log d$ where $w \approx ((h-s-k)/5)^2$; For the $s = \Theta(h)$ case, let us set $s = h/6$ for networks with h units. Then we can attain the VC-dimension $wh/6$ where $w \approx (h/6)^2$ by units of degree at most 2; or we can attain the VC-dimension $w(h-k)\log d/6$ where $w \leq ((h-k)/6)^2$ approximately. □

6. Lower bounds for piecewise polynomial networks

We will show two ways to attain the same lower bound. One is to replace "ρ^k unit" and "μ^k unit" in Figure 2 by a composition of piecewise polynomial function. The other is to combine an architecture for piecewise constant functions and that for polynomial functions. First we will show the former one.

Theorem 6.1. *Let us consider a set of networks of units with linear input functions and piecewise polynomial (with q polynomial segments) activation functions. $\Omega(ws\log(dqh/s))$ is a lower bound of the VC-dimension, where s is the depth of the network and d is the maximum degree of the activation functions. More precisely, $w(s-2)(\log d + \log(h/s) + \log q)$ is a lower bound.*

Our proof is based on that of the polynomial networks. We will use h units with activation function of $q \geq 2$ polynomial segments of degree at most d in place of each of "ρ^k unit" in the decoder of polynomial networks, which give the ability of decoding $\log dqh$ bits in one layer and $s\log dqh$ bits in total by $\Theta(sh)$ units in total. If h designates the total number of units, the number of the decodable bits is represented as $\log(dqh/s)$.

In the following for simplicity we suppose that dqh is a power of 2.

Let $\rho^k(x)$ be the k-th composition of $\rho(x)$ as usual *i.e.* $\rho^k(x) = \rho(\rho^{k-1}(x))$ and $\rho^1(x) = \rho(x)$.

Let $\rho^{\log d,l}(x) = \rho^{\log d}(\lambda^l(x))$, where $\lambda(x) = 4x$ if $x \leq 1/2$ and $4-4x$ otherwise, which by the way has 2^l polynomial segments.

Now the "ρ^k unit" in the polynomial case is replaced by the array $\rho^{\log d,\log q,\log h}(x)$ of h units that is defined as follows:

(i) $\rho^{\log d,\log q,1}(x)$ is an array of two units; one is $\rho^{\log d,\log q}(\lambda^+(x))$ where $\lambda^+(x) = 4x$ if $x \leq 1/2$ and 0 otherwise and the other is $\rho^{\log d,\log q}(\lambda^-(x))$ where $\lambda^-(x) = 0$ if $x \leq 1/2$ and $4-4x$ otherwise.

(ii) $\rho^{\log d,\log q,m}(x)$ is the array of 2^m units, each with one of the functions $\rho^{\log d,\log q}(\lambda^{\pm}(\cdots(\lambda^{\pm}(x))\cdots))$ where $\lambda^{\pm}(\cdots(\lambda^{\pm}(x))\cdots)$ is the m-th composition of $\lambda^+(x)$ or $\lambda^-(x)$. Note that $\lambda^{\pm}(\cdots(\lambda^{\pm}(x))\cdots)$ has at most three linear segments (one is linear and the others are constant 0) and that the sum of 2^m possible combinations $f(\lambda^{\pm}(\cdots(\lambda^{\pm}(x))\cdots))$ is equal to $f(\lambda^m(x))$ for any function f such that $f(0) = 0$.

Because of (ii) above we abuse the notation $\rho^{\log d,\log q,\log h}(x)$ to mean the sum of the outputs of the units in the array,*i.e.*, $\rho^{\log d}(\lambda^{\log q}(\lambda^{\log h}(x))) = \rho^{\log d}(\lambda^{\log qh}(x))$.

Instead of μ^k in the polynomial case, we use $\mu^{\log d,\log q,\log h}(x)$ which is defined in the same way as $\rho^{\log d,\log q,\log h}(x)$ except that:

- $\lambda^+(x) = 6x$ if $x \leq 1/2$ and 0 otherwise,
- $\lambda^-(x) = 0$ if $x \leq 1/2$ and $6-6x$ otherwise. and
- $\lambda(x) = 6x$ if $x \leq 1/2$ and $6-6x$ otherwise.

Unless there would be difficulty in infering context, we will simply use $\lambda^+(x)$, $\lambda^-(x)$, $\lambda(x)$, and $\lambda^{\pm}(x)$. When necessary we would use subscripts as in $\lambda_\rho(x)$ and $\lambda_\mu(x)$.

Then lemmas similar to the ones in the polynomial case follow. Let us define $\rho^{(i)}$ that has similar property as ρ^i. In parallel let us define $\mu^{(i)}$ as follows. $\rho^{(i)}(x)$ is defined as:

- $\rho^k \circ \lambda^l \circ \lambda^m \circ (\rho^{\log d,\log q,\log h})^j$ where $i = j(\log dqh) + p$ $(0 \leq p < \log dqh)$ and $k = \min\{p, \log d\}$, $l = \min\{\max\{0, p-\log d\}, \log q\}$, $m = \min\{\max\{0, p-\log dq\}, \log h\} = \max\{0, p-\log dq\}$.

Let us define $\mu^{(i)}(x)$ and its inverse.

- $\mu^{(i)}(x) \stackrel{\text{def}}{=} \mu^k \circ \lambda^l \circ \lambda^m \circ (\mu^{\log d,\log q,\log h})^j$ where j, k, l, and m are the same as in the above.
- $\mu^{(-i)}(x) \stackrel{\text{def}}{=} (\mu^{\log d,\log q,\log h})^{-j} \circ \lambda^{-m} \circ \lambda^{-l} \circ \mu^{-k}$ where j, k, l, and m are the same as in the above, $\mu^{-k}(x)$ is the k-th composition of the inverse of $\mu(x)$ on $[0, 1/2]$, $\lambda^{-m}(x)$ is the m-th composition of the inverse of $\lambda(x)$ on $[0, 1/2]$, and $(\mu^{\log d,\log q,\log h})^{-j}$ is the j-th composition of $\lambda^{-\log h} \circ \lambda^{-\log q} \circ \mu^{-\log d}$.

Lemma 6.2. *For any binary sequence $b_1, b_2, \ldots, b_s$, there exists an interval $[x_1, x_2]$ such that $b_i = \mathcal{H}_{1/4,3/4}(\rho^{(i)}(x))$ for any $x \in [x_1, x_2]$.*

Proof. Clear from the fact that for any $x \in [0,1]$ we can find x_0 and x_1 such that $x = \rho(x_0) = \rho(x_1)$ (and $x = \lambda(x_0) = \lambda(x_1)$), $x_0 \in [0, 1/4]$, and $x_1 \in [3/4, 1]$.

Lemma 6.3. *If* $0 < \rho^{(i)}(x) < 1$ *for any* $0 < i \leq l$ *and* $(16/3)^l\epsilon < 1/4$, $\rho^{(l)}(x) - (16/3)^l\epsilon < \rho^{(l)}(x+\epsilon) < \rho^{(l)}(x) + (16/3)^l\epsilon$.

Lemma 6.4. *Take* $x_0 = \mu^{(-(j-1))}(1/6)$. *Then* $\mu^{(j-1)}(x_0) = 1/6$, $\mu^{(j)}(x_0) = 1$, $\mu^{(i)}(x_0) = 0$ *for all* $i > j$, *and* $\mu^{(j-i)}(x_0) \leq (1/6)^i$ *for all* $i > 0$ *and* $\leq j$.

Proof. Clear from the fact that $\mu(x) \geq 6x$ on $[0, 1/6]$.

Lemma 6.5. *For any binary sequence* $b_1, b_2, \ldots, b_k, b_{k+1}, b_{k+2}, \ldots, b_{2k}, \ldots, b_{(s-1)k+1}, \ldots, b_{sk}$ *where* $k = \log dqh$, *take* y *such that* $b_i = \mathcal{H}_{1/4,3/4}(\rho^{(i)}(y))$ *and* $0 \leq \rho^{(i)}(y) \leq 1$ *for all* i , $x_1 = \mu^{(-(jk-1))}(1/6)$, *and* $x_0 = \mu^{(-(l-1))}(1/6)$. *Then for* $z = \sum_{i=1}^{m} \rho^{(ik)}(y)\mu^{(ik)}(x_1)$, $\mathcal{H}_0\left(\sum_{i=0}^{k-1} \rho^{(i)}(z)\mu^{(i)}(x_0) - (1/2)\right) = b_{kj+l}$ *holds.*

Since the proofs are similar to those for the polynomial case with only notational differences between ρ^l and $\rho^{(l)}$ and between μ^l and $\mu^{(l)}$, they are omitted.

If we alocate h/s units to each layer of the decoder, the number of units used for the decoder is $2(s-4) + 2(s-4)(h/s) + 4(\log(dqh/s) - 1) + 1 + 2(h/s)$ (the decoder can decode at most $(s-4)\log(dqh/s)$ bits), since $\mu^{(k-1)}$ consists of $(1/2)(h/s)$ units, $\mu^{(k-2)}$ consists of $(1/4)(h/s)$ units,..., $\mu^{(1)}$ consists of $(1/2^{k-1})(h/s)$ units so that $\mu^{(1)}, \ldots, \mu^{(k-1)}$ consists of at most $k + (h/s)$ units.

The number of units used for the encoder is less than $4h$; we though have constraints on s (which dominates the depth of the network) and h (which dominates the number of units in the network) that $h \leq \binom{n+m}{m}$ and $m = O(s)$ or roughly $\log h = O(s)$ be satisfied.

When $s = o(h)$, by units of degree at most d, we can attain the VC-dimension $ws\log(dqh/s)$ where $w \approx (4h/9)^2$; For the depth $h/5$ networks with h units, we can attain the VC-dimension $wh\log(dq/5)$ where $w = (2h/9)^2$ approximately.

In the following, we will show the different architecuture for $d = 1$ that use piecewise constant functions and a few linear functions, which may be used in combination with polynomial networks to yield networks achieving the VC-dimension lower bound of piecewise polynomial networks.

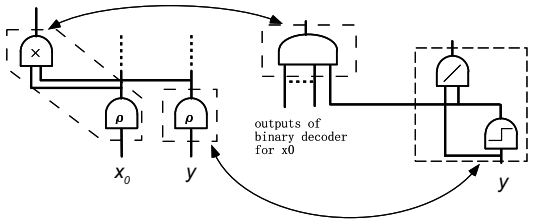

Fig.4. The correspondence between polynomial network and piecewise one ($d = 2$).

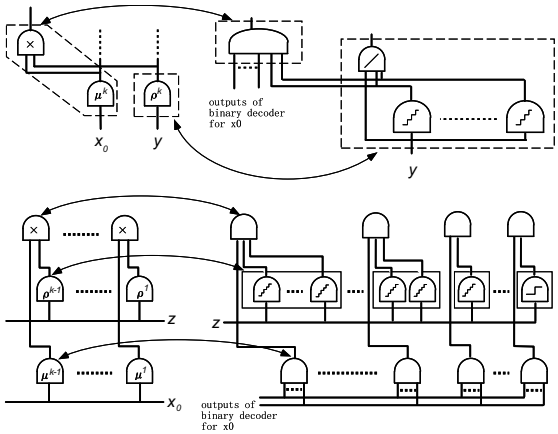

Fig.5. The correspondence between polynomial network and piecewise one ($d \geq 2$).

The basic scheme of the decoder for the case $d = 1$, $q = 2$, and $h = 1$ is similar to Figure 1. "ρ unit" is replaced by two units; one is the Heaviside ($\mathcal{H}_0(x)$) unit and the other is the linear unit, *i.e.*,

- $\mathcal{H}_0(x)(x - 1 + 0.5)$, and
- $(x - \mathcal{H}_0(x)(x - 1 + 0.5)) \times 2$.

for input $x < 2$. The multiplication unit in Figure 1 is replaced by a simple AND which is realized by $\mathcal{H}_0(x + y - 1.5)$.

The first output of the above "ρ unit" gives the most significant bit of x and the second output gives the tailing bits mulitplied by 2. Therefore the sequence of the above "ρ units" gives a sequence of the binary

expantion of x.

Note that the linear units in the above architecture are easily eliminated by using

- $\mathcal{H}_0(x)(x - 2^{-(i-1)} + 2^{-i})$,
- $(x - \mathcal{H}_0(x)(x - 2^{-(i-1)} + 2^{-i})) \times 2$, and
- $\mathcal{H}_0(x + y - 1.5 \cdot 2^{-(i-1)})$.

The basic scheme of the decoder for the case $d = 1$, $q \leq 2$, and $h \leq 2$ is similar to Figure 2. "ρ^k unit" is replaced by two units; one is a step function and the other is linear, *i.e.*,

- $\mathcal{S}_0(x - qi + 0.5)$, and
- $(x - \mathcal{S}_0(x - qi + 0.5)) \times qh$,

where $\mathcal{S}_0(x) = \lceil x \rceil$ for $0 \leq x \leq q$, 0 for $x < 0$, and $q - 1$ for $q < x$. The multiplication is $\alpha(M(\mathcal{H}_0(y) - 1) + x)$ where $\alpha(x)$ here is a lamp function, *i.e.*, $\alpha(x) = x$ if $x \geq 0$ and 0 otherwise, and M is a large constant.

The modification of the left part of the decode in Figure 2 is a bit complicated. $\rho^1, \ldots, \rho^{\log q}$ are replaced by:

- $\alpha_1(x) = \mathcal{H}_0(x - qh/2)$,
- $\alpha_2(x) = \sum_{i=1}^{3} (-1)^{i-1} \mathcal{H}_0(x - (qh/4)i)$,

$$\vdots$$

- $\alpha_{\log q}(x) = \sum_{i=1}^{q-1} (-1)^{i-1} \mathcal{H}_0(x - (qh/2^{\log q})i)$,

and $\rho^{\log q+1}, \ldots, \rho^{\log qh}$ are replaced by:

- $\alpha_{\log q+1}(x) = \sum_{i=1}^{q-1} (-1)^{i-1} \mathcal{H}_0(x - (h/2)i) + (-1)\mathcal{H}_0(x - (h/2)q) + \sum_{i=1}^{q-1} (-1)^{i-1} \mathcal{H}_0(x - (h/2)q - (h/2)i)$,

$$\vdots$$

- $\alpha_{\log qh}(x) = \sum_{i=1}^{q-1} (-1)^{i-1} \mathcal{H}_0(x - (h/2^{\log h})i) + (-1)\mathcal{H}_0(x - q)$
  $+ \sum_{i=1}^{q-1} (-1)^{i-1} \mathcal{H}_0(x - q - (h/2^{\log h})i) + (-1)\mathcal{H}_0(x - 2q) + \cdots$
  $+ \sum_{i=1}^{q-1} (-1)^{i-1} \mathcal{H}_0(x - (q-1) - (h/2^{\log h})i)$,

where each summation and $(-1)\mathcal{H}_0(\ \cdot\ )$ term corresponds to a unit with at most q constant segments. $\mu^1, \ldots, \mu^{\log qh}$ are replaced by:

- $\mathcal{H}_0(x - 2^0 - 0.5) - \mathcal{H}_0(x - 2^0 + 0.5)$,
- $\mathcal{H}_0(x - 2^1 - 0.5) - \mathcal{H}_0(x - 2^1 + 0.5)$,

$$\vdots$$

- $\mathcal{H}_0(x - 2^{\log qh-1} - 0.5) - \mathcal{H}_0(x - 2^{\log qh-1} + 0.5)$,

where the property that at most one bit is 1 among the binary expantion of the input is assumed. Note that the total number of units used in the part is at most $\log q + 4h + \log qh$.

The elimination of the linear units requires more units this time than the case for $q = 1$. The left part of the deocder is used in stead of the above right part of the decoder. Then in total $s(\log q + 4h)$ units are used in the decoder. The multipliation is $\log qh$ bitwise AND which requires $s \log qh$ units in total.

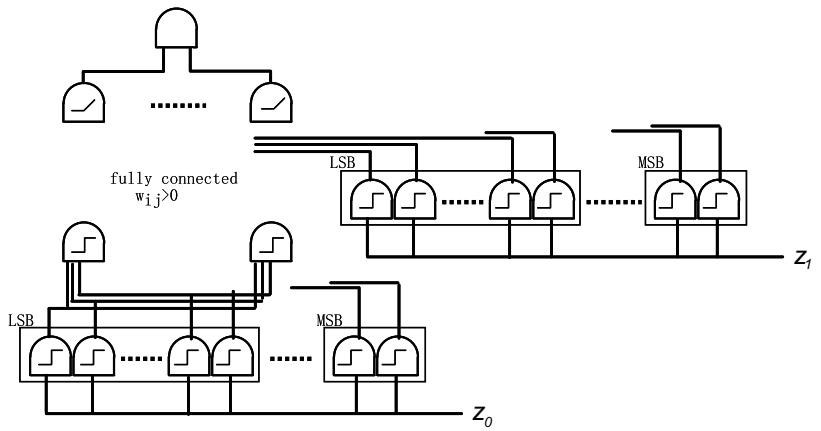

Fig.6. The encoder network by threshold and lamp units.

The main part of the encoder consists of two layers that has inputs from two binary decoders; the two integer external inputs z_0 and z_1 are fed to its own binary decoders.

The first layer of the main part consists of h threshold units and the second consists of h lamp units whose activation function $\alpha(x)$ is x if $x \geq 0$ and 0 otherwise.

The i-th unit $g_{1,i}$ in the first layer performs $g_{1,i}(\mathbf{x}) = 1$ if $\sum x_k 2^{k-1} = i$ and 0 otherwise, which is realized by $\mathcal{H}_0(M(\sum x_k i_k - \sum i_k) + 0.5)$ where $i = \sum i_k 2^{k-1}$ where M is a large constant.

The j-th unit $g_{2,j}$ in the second layer performs $g_{2,j}(\mathbf{x}, y) = y$ if $\sum x_k 2^{k-1} = j$ and 0 otherwise, which is realize by $\alpha(M(\sum x_k j_k - \sum j_k) + y)$ where $j = \sum j_k 2^{k-1}$ where $\alpha(x)$ is a lamp function defined by $\alpha(x) = x$ if $x \geq 0$ and 0 otherwise, and M is a large constant.

7. The VC-dimension of recurrent networks

Koiran and Sontag have done a pioneering work in the field. We approach it from a different dirrection and give a better and perspective view.

The recurrent network we consider has the form:

$$\mathbf{y}(t) = f(\mathbf{u}(t), \mathbf{x}(t), \mathbf{w}_1), \quad \mathbf{x}(t+1) = \mathbf{y}(t), \quad z(t) = g(\mathbf{x}(t), \mathbf{w}_2).$$

where f and g are standard feedforward networks, z is the network output, $\mathbf{x}(0)$ is predefined constant, and $\mathbf{u}$ is the network input. The depth of the recurrent networks is the depth of the network for f.

As in the feedforward case, the output $z(t)$ (real value) will be fed to the Heaviside function and converted to 0 or 1 depending on the value being negative or positive.

In the following we consider two cases for the input, *i,e.*, $\mathbf{u}(t)$ is constant in time or varying in time. The network output after prespecifed time units r is the output for the input $\mathbf{u}(t)$. The output value need not be stable and constant after r time units, but in fact in the network we constract for the proof of lower bounds for constant input case, the output value is stable after the time.

The results are summarized as follows.

Theorem 7.1. *For the polynomial recurrent networks, the VC-dimension is* $\Theta(wrs \log d)$*, which is* $\Theta(wrh \log d)$ *for* $s = \Theta(h)$*, where* d *is the maximum of the degree of the activation functions and* s *is the depth of the network without the feedback loops.*

Theorem 7.2. *For the piecewise polynomial recurrent networks, the VC-dimension is* $O(wrs(\log r + s \log d + \log(qh/s))$ *and* $O(wrs(\log rd + (h/s)\log q))$ *where* s *is the depth of the feedforward part of the network. Moreover we have* $\Omega(wrs \log(dqh/s))$ *bound. As in the feedforward caes, we have* $\Theta(wrh \log dq)$ *for* $s = \Omega(h)$*.*

Theorem 7.3. *For the thrshold recurrent networks, the VC-dimension is* $\Theta(wh)$ *for* $r \geq h$ *where the network depth is unbounded. More precisely it is* $O(wr \log rh)$*,* $O(wh)$*, and* $\Omega(wr)$ *for* $r \leq h$*.*

Corollary 7.4. *For the standard sigmoidal recurrent networks, the VC-dimension is* $O(w^2h^2r^2)$ *where* s *is the depth of the feedfoward part of the network. Moreover we have* $\Omega(wrs)$ *bound. As in the feedforward caes, we have* $\Theta(wrh)$ *for* $s = \Omega(h)$*.*

8. Upper bounds for recurrent networks

The proofs for the two theorems below are obtained just by "unfolding" the r repetitions of "feedbacked" networks.

Theorem 8.1. *For the polynomial recurrent networks case, the VC-dimension is* $O(wrs \log d)$*, which is* $O(wrh \log d)$ *for* $s = \Theta(h)$*. The bound holds for the constant* $\mathbf{u}$ *and time-varying* $\mathbf{u}(t)$*.*

Proof. Similar to the proof of Theorem 4.2. The number of functions realizable by the networks is upper bounded by $(4eNd^{rs}/w)^w$. Hence the VC-dimension is $O(wrs \log d)$. □

Theorem 8.2. *For the piecewise polynomial recurrent network case, the VC-dimension is* $O(wrs(\log r + s \log d + \log(qh/s))$ *and* $O(wrs(\log rd + (h/s)\log q))$ *where* s *is the depth of the network disregarding the feedback connections. As in the feedforward caes, we have* $O(wrh \log rdq)$ *for* $s = \Theta(h)$ *or the unbounded depth case. The bound is valid for the constant* $\mathbf{u}$ *and time-varying* $\mathbf{u}(t)$*.*

Proof. In the same way as in the proof of Theorem 4.5, the first bound is:

$$
\begin{aligned}
&N_{cc}\left(\mathcal{R}^w - \bigcup \mathcal{N}(f_G(\mathbf{w};\mathbf{x}_i))\right) \\
&\le \left(\frac{4emqh_1 d}{w_1}\right)^{w_1} \cdot \cdots \left(\frac{4emqh_s(d^{s-1}+\cdots+d+1)d}{w_1+\cdots+w_s}\right)^{w_1+\cdots+w_s} \\
&\qquad \left(\frac{4emqh_1(d^s+\cdots+1)d}{w_1+\cdots+w_s}\right)^{w_1+\cdots+w_s} \cdots \left(\frac{4emqh_s(d^{2s-1}+\cdots+1)d}{w_1+\cdots+w_s}\right)^{w_1+\cdots+w_s} \\
&\qquad\qquad \cdots\cdots \left(\frac{4emqh_s(d^{rs-1}+\cdots+1)d}{w_1+\cdots+w_s}\right)^{w_1+\cdots+w_s} \\
&\le \left(\frac{4emqd^s(h/s)}{w}\right)^{wrs},
\end{aligned}
$$

and the second bound is:

$$
N_{cc}\left(\mathcal{R}^w - \bigcup_{i=1}^{w}\mathcal{N}(f_G(\mathbf{w};\mathbf{x}_i))\right) \le N_{cc}\left(\mathcal{R}^w - \bigcup_{i=1}^{w}\bigcup_{j=1}^{q^{rh}}\mathcal{N}(f_{G,j}(\mathbf{w};\mathbf{x}_i))\right) \le \left(\frac{4emq^{rh}d^{rs}}{w}\right)^{w}
$$

□

Theorem 8.3. *For the thrshold recurrent networks, the VC-dimension is $O(wr\log rh)$.*

The VC-dimension seems to increase infinitely with respect to r, but it is an illusion.

Theorem 8.4. *For the thrshold recurrent networks, the VC-dimension is $O(wh)$ where the network depth is unbounded.*

Proof. We just count an upper bound for the number of functions realized at the last (or r-th) stage of the computation. The stage has two types inputs: one is external inputs and the other is the one from the previous stage of computations. Let N be the number of the possible external inputs. Since the inputs from the previous stage is represented by some point in $\{0,1\}^h$, the total number of points to be input to the last stage is at most $N2^h$. The number of realizable functions defined on $N2^h$ points is upper bounded by $\prod_{i=1}^{h}\left(eN2^h/n_i\right)^{n_i} \le \left(eN2^h\right)^w (h/w)^w$. Hence we get the bound. □

Clearly the bound is true even when the network input varies in time.

If we have a linear gate, the result of Theorem 8.4 may not hold. See the next section for the confirmation.

9. Lower bounds for the recurrent networks

Theorem 9.1. *For the polynomial recurrent networks, the VC-dimension is $\Omega(wrs\log d)$, which is $\Omega(wrh\log d)$ for $s=\Theta(h)$.*

Theorem 9.2. *For the piecewise polynomial recurrent networks, the VC-dimension is $\Omega(wrs\log(dqh/s))$. As in the feedforward caes, we have $\Omega(wrh\log dq)$ for $s=\Theta(h)$ or the unbounded depth case.*

We get proofs for the above two cases by modifing the corresponding feedforward network cases. Note that when we use in the decoding subnetwork any of the pairs, ρ and ρ, μ^k and ρ^k, or $\mu^{\log d,\log q,\log h}$ and $\rho^{\log d,\log q,\log h}$, we use them in repetitive stages. The basic idea is therefore to include them in the feedback loop as a main decoding part.

Theorem 9.3. *For the thrshold recurrent network case, the VC-dimension is $\Omega(wr)$ for $r \le h$.*

Proof. We construct a network with a seemingly different structure. For simplicity we suppose that the network inputs are three $\log h$-bits inputs $\mathbf{u}_1$, $\mathbf{u}_2$, and $\mathbf{u}_3$, $\mathbf{u}_1$ is used to select a unit in the first layer in the encoder, thereby only one unit outputs 1 and the others output 0. $\mathbf{u}_2$ is used to select a unit in the second layer in the encoder, by which only one connecting weight among the weights between the two layers affect the network output.

There is a kind of A-D converter (analog to digital converter) whose output is fed to the second layer mentioned above. The comparison is done in the selected unit and the result is fed back to the AD converter. In the AD converter, the bit pattern of at most h bit length that is interpreted as the binary expansion of the selected connecting weight is calculated. $\mathbf{u}_3$ specifies the bit output from the weight or the result of the AD conversion.

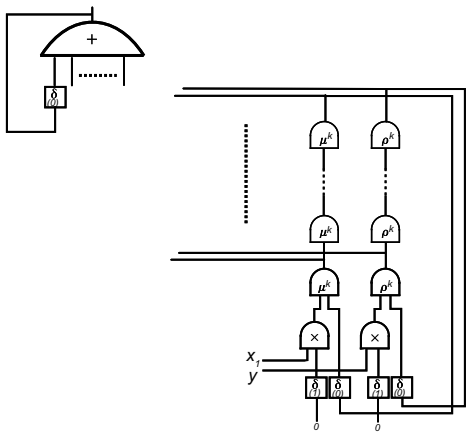

Fig.7. The recurrent network for $q = 1$ and $d \geq 2$.

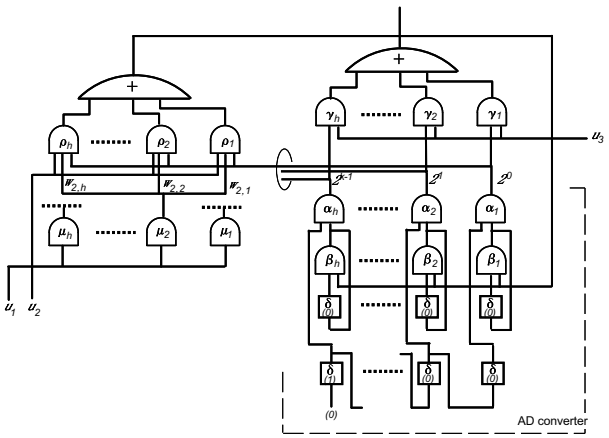

Fig.8. The recurrent network for $q \geq 2$ and $d \geq 2$.

The detail of the network follows. In the following, the weight $w_{i,j}$ is an h bit integer.

(1) $\mathbf{u}_1$, $\mathbf{u}_2$, and $\mathbf{u}_3$ are the vectors of $+1$ or -1 and all the possible combinations may be fed.

(2) $\mu_i = \sigma_0(\mathbf{i} \cdot \mathbf{u}_1 - \log h + 1/2)$ where $\mathbf{i}$ is an array of $+1$ and -1 in which $+1$ corresponds to 1 and -1 to 0 of the binary representation of i.

(3) $\rho_i = \sigma_0 \left(2^h(\mathbf{i} \cdot \mathbf{u}_2 - \log h + \epsilon) + \sum_{j=1}^{h} w_{i,j}\mu_j - \sum_{j=1}^{h} 2^{j-1}\beta_j\right)$.

(4) $y = \sigma_0 \left(\sum_{j=1}^{h} \rho_j\right)$

(5) $\alpha_j = \sigma_0 \left(\delta(\alpha_{j-1})\right)$

(6) $\beta_j = \sigma_0 \left(\delta(\beta_j) + \alpha_j + y\right)$

(7) $\gamma_i = \sigma_0 \left(\mathbf{i} \cdot \mathbf{u}_3 - \log h + \beta_i - 1/2\right)$

(8) $y = \sigma_0 \left(\sum_{j=1}^{h} \gamma_j\right)$

When $\mathbf{u}_1$ and $\mathbf{u}_2$ correspond to the binary representations of i and j respectively, only μ_i and possibly ρ_j output 1. Let $t = 0$ be the start time of calculation. At time t only α_{h-t} among all the α's is 1. At the end of computation at time t, β_h to $\beta_{h-(t-1)}$ correctly represent the leading t bits of the weight $w_{i,j}$, since at the start of computation time t, y is 1 if and only if the t-th leftmost bit of the weight $w_{i,j}$ is 1. □

Theorem 9.4. *There is a set of recurrent networks consisting of a linear unit and threshold units that has the VC-dimension* $\Omega(wr \log h)$.

10. References

[1] Anthony, M: Classfication by polynomial surfaces, *NeuroCOLT Technical Report Series*, NC-TR-95-011 (1995).

[2] Bartlett, P. L. & R. C. Williamson: The VC-dimension and pseudodimension of two-layer neural networks with discrete inputs, *preprint* (1993).

[3] Blumer, A., A. Ehrenfeucht, D. Haussler, and M. K. Warmuth: Learnability and the Vapnik-Chervonenkis Dimension, *Journal of the ACM*, **36**(4), 929-965 (1989).

[4] Cover, T. M.: Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, *IEEE Transactions on Electronic Computers*, vol. 14, 326–334 (1965).

[5] Devaney, R.L.: *An Introduction to Chaotic Dynamical Systems*, Benjamin/Cummings Publishing Compnay, 1986.

[6] Goldberg, P. and M. Jerrum: Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers, *Proc. Sixth Annual ACM Conference on Computational Learning Theory*, 361–369 (1993).

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

[8] Karpinski, M. and A. Macintyre: Quadratic bounds for VC dimension of sigmoidal neural networks, *manuscript* (1994).
[9] Khovanskii, A. G. : *Fewnomials*, Translations of Mathematical Monographs 88, AMS (1991).
[10] Koiran, P. and E. Sontag: Neural networks with quadratic VC dimension, *NeuroCOLT Technical Report series*, NC-TR-95-044 (1995).
[11] Koiran, P. and E. Sontag: Vapnik-Chervonenkis dimension of recurrent neural networks, *DIMACS Technical Report*, 97-56 (1996).
[12] Maass, W. G.: Bounds for the computational power and leaning complexity of analog neural nets, *Proc. 25th Annual Symposium of the Theory of Computing*, 335-344 (1993).
[13] Macintyre, A. and E. D. Sontag: Finiteness results for sigmoidal "neural" networks, *Proc. 25th Annual Symposium of the Theory of Computing*, 325–334 (1993).
[14] Milnor, J.: On the Betti numbers of real varieties, *Proc. of the AMS*, **15**, 275–280 (1964).
[15] Sakurai, A.: Tighter Bounds of the VC-Dimension of Three-layer Networks, *Proc. WCNN'93*, III, 540–543 (1993).
[16] Sakurai, A.: On the VC-dimension of depth four threshold circuits and the complexity of Boolean-valued functions, *Proc. ALT93 (LNAI 744)*, 251–264 (1993); refined version is in *Theoretical Computer Science*, **137**, 109-127 (1995).
[17] Sakurai, A.: On the VC-dimension of neural networks with a large number of hidden layers, *Proceedings of NOLTA'93*, IEICE, 239–242 (1993).
[18] Sakurai, A.: On the VC-dimension of depth four threshold circuits and the complexity of Boolean-valued functions, *Theoretical Computer Science*, vol.137, no.1, 109–127 (1995).
[19] Sakurai, A.: Polynomial bounds for the VC-dimension of sigmoidal, radial basis function, and sigma-pi networks, *Proceedings of WCNN'95*, vol.I, 58–63 (1995).
[20] Vapnik, V., and A. Y. Chervonenkis : On the uniform convergence of relative frequencies of events to their probabilities, *Theory of Probability and Its Applications*, **16**, 264-280 (1971).
[21] Vapnik, V. : *Estimation of Dependences Based on Empirical Data*, Springer-Verlag (1982).
[22] Warren, H. E.: Lower bounds for approximation by nonlinear manifolds, *Trans. AMS*, **133**, 167–178, (1968).