Limiting Network Size within Finite Bounds for Optimization

Linu Pinto*, Dr.Sasi Gopalanb

*Research Scholar, Department of Mathematics, Cochin University of Science and Technology, Kochi-682022, Kerala, India  
bProfessor in Mathematics, School of Engineering, Cochin University of Sience and Technology, Kochi-682022, Kerala, India

Keywords:  
Feed Forward Neural Network  
Back propagation Algorithm  
VC Dimension  
Sigmoid Networks  
Dissimilarity Measure

ABSTRACT

Largest theoretical contribution to Neural Networks comes from VC Dimension which characterizes the sample complexity of classification model in a probabilistic view and are widely used to study the generalization error. So far in the literature the VC Dimension has only been used to approximate the generalization error bounds on different NN architectures. VC Dimension has not yet been implicitly or explicitly stated to fix the network size which is important as the wrong configuration could lead to high computation effort in training and leads to overfitting. So there is a need to bound these units so that task can be computed with only sufficient number of parameters. For binary classification tasks shallow networks are used as they have universal approximation property and it is enough to size the hidden layer width for such networks. The paper brings out a theoretical justification on required attribute size and its corresponding hidden layer dimension for a given sample set that gives an optimal binary classification results with minimum training complexity in a single layered feed forward network framework. The paper also establishes proof on the existence of bounds on the width of the hidden layer and its range subjected to certain conditions. Findings in this paper are experimentally analyzed on three different datasets using Mathlab 2018 (b) software.

1. INTRODUCTION

Deep neural Network have successfully made its application in many pattern recognition problems [1], [2] and hence its theoretical properties especially the expressive power in terms of network complexity are currently an active area of research. Even though studies [3] shows that deep architectures are more compact in representing complex functions compared to shallow networks. There are empirical results [4] that shows the ability of shallow networks in identifying some complex functions as deep networks with same number of parameters. But the theoretical analysis complimenting these results are still an open question that yet need to be studied. Although topological justification [5],[6],[7],[8] has been done on the theoretical analysis but they demand an exponential increase in the number of parameters.

It has long been known from the studies [9],[10] that shallow networks is sufficient to compute and approximate any complex function on a compact domain upto a desired level of accuracy and are well known to have universal approximation property. But theorems supporting their claim do not give an estimate on the network size and assume infinite number of network units. So far studies were done on the reduction of this network complexity by use of many regularization techniques like weight decay [11]. The approximation capabilities and generalization error that controlled by measures from statistical learning theory provides a good theoretical explanation on the network complexity in terms of sample size and its distribution. This technique was quantified by a concept VC Dimension by Vapnik and Chervonenkis in 1971 [12]. This concept was widely used to compute generalization error bounds on the architecture on the deep and shallow networks and are not explicitly or implicitly applied to configure the network complexity.

In this paper we investigate the adaptability of VC Dimension to structure shallow networks architecture by limiting the network size to a finite value thereby balancing the dimensions of the sample space.

1.1 Literature Review

Arguments on the suitable depth and width of a network is a widely discussed open problem [12], [14], [15], [16], [17], [18], [19], [20], [21]. Many of these recent papers demonstrate that increasing the depth of a network leads to the reduction in the dimension of each layer. A Feed forward network with a single hidden layer could approximate any continuous function or a measurable function on a compact set with a continuous sigmoidal activation function [22], [23], [24]. Later many results on Feed Forward Neural Network were proved with various methods in [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35] establishing the existence of an approximation to any complex continuous function up to a desired level of accuracy (say 𝜖) and generalization abilities of Neural Network. Main problem is related to the size of the hidden layer in attaining the desired accuracy. In 1993 Barron [27] gives a result on the number of neurons needed for the function approximation based on Fourier transform where each hidden neuron is activated with sigmoidal function by imposing certain conditions. Similar results were developed later in [28], [29], [30] related to the complexity of approximation and shows the importance of sigmoid activation functions.

*Corresponding author. E-mail address: linupinto671@gmail.com
The approximation capabilities and generalization error that controlled by measures from statistical learning theory provides a good theoretical explanation on the expressive power of Neural Network. Sample complexity is a measure from statistical learning theory that relates with sample size and its distribution. The generalization capabilities of NN should be independent with the sample complexity after successful learning. This sample complexity is quantified based on a concept VC Dimension by Vapnik and Chervonenkis in 1971 [12]. The VC Dimension was later studied by Baum and Haussler in 1989; Maas in 1994, Sakurai in 1999 in [36],[37],[38] giving the lower and upper bounds for FNN with binary outputs and binary activation functions. Goldberg and Jerum in 1995 [41] analysed the bounds on VC Dimension with piecewise Polynomial activation functions. This was further improved by Koiran and Sontag in 1997 [39]. In 1997 Karpinski and Macintyre in [40] gives these bounds on more general activation functions like Tanh, logsig or atan. In 2009 Anthony and Barlett [29] gives a detailed explanation on VC Dimension with bounds on FNN with commonly used activation functions. All the related studies in the bounds points the theoretical explanation on the expressive power of Neural Network. Apart from VC Dimension concept there are several ways to analyze the expressive power of Neural Network. Delallau and Bengio in 2011 [15] showed that for certain classes of polynomials sum-product networks can match approximately good with deep networks. In 2014 Montufar et al. [16] showed that in network topology the number of linear regions could be estimated and increases exponentially in proportion to the number of layers. In 2014 Baimchini and Scarcelli [17] characterises the topological properties of network functions by giving bounds on betti numbers. In 2015 and later in 2016 Telgarsky [18], [19] gives experimental results on the efficiency of Deep Networks for classification problems. In 2015 Eldan and Shamir [20] proved that to approximate a function the dimension of the input space should be increased exponentially for a two layered feed forward network.

1.2 Overview of the paper

Section 2 first explains the network structure studied in the paper by giving a detailed explanation on basic notations and definitions related to single output classification problem with a single hidden layered Feed Forward Neural Network. Secondly it explains a new perspective taken in the development of the work that is the VC Dimension approach. The approach is systematically grooved with real algebraic definitions to fix the network size within a finite range. The results are supported with theorems. This section also explains the respective algorithm selected for the optimization and the proximity measures to evaluate the performance of the network framework proposed. Section 3 gives the experimentation results on the proposed work and section 4 gives a detailed discussion on the results with a suggestion on the required dimension on the sample space for the optimization with shallow networks. Section 4 gives the conclusion along with future study and limitations of the work.

2. Method

2.1 Background

This section summaries single layered feed forward neural network model used in the proposed study on a binary classification problem. This basic structure comprises three main constituent called layers that act as the processing tiers to form a regression curve for a mapping between the input vector and the targeted scalar. The first layer receive the inputs , the middle layer called the hidden layer activate the inputs with sigmoid activation functions to transform it into a more affluent domain by consuming connection weights and thresholds and the third layer called the output layer gives the corresponding output by the activation of linear threshold functions. The calculations in the hidden layer forms the underlying structure for the regression curve. This structure could further be tuned with weights and thresholds in the connections.

The notations relating the proposed network model consists of sigmoid activation functions \((σ)\) as the collection of functions in the hidden layer and \((φ)\) is the linear threshold activation using in the output layer. \(w_{ik}^j\) denotes the weight that strengthens the connectivity between the \(k^{th}\) neuron in the \(l^{th}\) layer and \(j^{th}\) neuron in the \(l+1^{st}\) layer. \(x_i^l\) is the \(i^{th}\) input vector \(x_i^l=⟨x_{i1},x_{i2},x_{i3},...,x_{in}⟩\) for \(i=1,2,3,...,r\) samples collection \(E\) from the population \(X\). Let \(S\) denotes a subset of \(E\) and each \(S \subseteq E\) represents the state ‘\(u\)’ of the network. Hence there are a total of \(2^n\) states for the network. \(s_j^l\) denotes the weighted sum at \(j^{th}\) node of \(l^{th}\) layer with thresholder \(t_j^{l-1}\). Weighted sum \(s_j^l\) together with the threshold \(t_j^{l-1}\) are activated with the activation function \(f_j^l\). \(W_{r}\) be the total weights and threshold in the network. \(C_p\) be the total computational units in the network with \(n\) input nodes and \(m\) hidden nodes for the network shown in Fig. 1. Depth of the network is defined as the number of hidden layers in the network and width is defined as the number of nodes in each layer. For a single layered Feed forward network as shown in Fig. 1 depth is one and the width is the number of nodes \(m\) in the hidden layer.

![Fig. 1: Two Layered Feed Forward Sigmoid Network with Single Output](image-url)

The functions represented by the processing state ‘\(u\)’ corresponding to each set \(S \subseteq E \subseteq X\) is denoted as
\( f_u : X \times \Omega_w \rightarrow Y \) \hspace{1cm} (1)

\( f_u \) defined in (1) and shown in Fig. 1 is a well defined function such that

\[
\begin{align*}
\hat{f}_u(\tilde{X}, w_u) &= W^T \sigma(W^0 \tilde{X} + T^0) + T^1 \\
\end{align*}
\]  \hspace{1cm} (2)

where \( W^0 = \begin{bmatrix}
w_{01}^0 & w_{02}^0 & \cdots & w_{0n}^0 \\
\vdots & \vdots & \ddots & \vdots \\
w_{m1}^0 & w_{m2}^0 & \cdots & w_{mn}^0
\end{bmatrix} \) is an \( m \times n \) matrix.

\[
T_0 = \begin{bmatrix}
t_1^0 \\
\vdots \\
t_m^0
\end{bmatrix}, 
T^1 = \begin{bmatrix}
w_{11}^1 & w_{12}^1 & \cdots & w_{1n}^1 \\
\vdots & \ddots & \vdots & \vdots \\
w_{m1}^1 & w_{m2}^1 & \cdots & w_{mn}^1
\end{bmatrix}, 
\tilde{X} \in \mathbb{R}^n, \Omega_w, \text{ collection of weight matrices} \ w_u(n+3) \times m \text{ with each entry} \ w \in \mathbb{R} \ , X \in \mathbb{R}^n \text{ and} \ Y \in [0,1]. \text{ For a Two layered feedforward neural network as shown in Fig. 1 with activation functions fixed of Neural Network } \mathcal{M} \text{ for a state } u \text{ could be represented in a } (n + 3) \times m \text{ weight matrix } w_u \in \Omega_w \text{ as}
\]

\[
w_u = \begin{bmatrix}
w_{11}^0 & w_{12}^0 & \cdots & w_{1n}^0 & t_1^0 & t_1^1 \\
\vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
w_{m1}^0 & w_{m2}^0 & \cdots & w_{mn}^0 & t_m^0 & t_m^1 \\
\end{bmatrix} 
\] \hspace{1cm} (3)

2.2 VC Dimension based Approach

Literature review shows that single layered feed forward neural network with sigmoid activation functions has established its importance in the approximation of any measurable functions upto a desired level of accuracy. Main problem concerning this approximation is related to the size of the hidden layer. The concept of VC Dimension can address this problem to some extent but its probabilistic nature is limited to the computation of generalization error and is inappropriate in giving a higher judgement degree to identify a state ‘u’ accurately. Hence an improbabilistic definition in terms of continuity of functions designated as the \( \epsilon \)-identifiable is applied to connect the concept with network size.

The function \( f_u \) corresponding to the state ‘u’ as defined in (2) is said to be \( \epsilon \)-identifiable on a set \( S \subseteq E \), if for a predefined \( \epsilon \) value ( considerably small and positive ) there exist \( w_u \in \Omega_w \) such that

\[
f_u(\tilde{x}, w_u) = \begin{bmatrix}
\theta^1 \in B_{\epsilon}(1) \\
\theta^0 \in B_{\epsilon}(0)
\end{bmatrix} \quad \forall \tilde{x} \in S \subseteq E \quad \forall \theta^0 \in S^C \subseteq E 
\] \hspace{1cm} (4)

where \( B_{\epsilon}(1) = \{ \theta^1 \in R : 1 - \epsilon < \theta^1 \leq 1 \} \) , \( B_{\epsilon}(0) = \{ \theta^0 \in R : 0 \leq \theta^0 < \epsilon \} \) \( \text{ and} \ B_{\epsilon}(1) \cap B_{\epsilon}(0) = \emptyset \). Clearly the function denoted as \( f_{\epsilon}^u \) is well defined.

The function (2) is non-identifiable within an \( \epsilon \) range on a set \( S \subseteq E \), if there exist no \( w_u \in \Omega_w \) such that (4) satisfies.

The definition becomes more acceptable by giving a criteria to the chosen epsilon. If the chosen epsilon is least as possible then proximity could be well defined and the function turns out to be \( \epsilon \)-identifiable function.

The function \( f_u \) corresponding to the state ‘u’ is said to be least \( \epsilon \)-identifiable function on a set \( S \subseteq E \), if for a predefined \( \epsilon \) value there exist a sequence of values \( \epsilon_0 < \epsilon_1 < \cdots < \epsilon_n = \epsilon \) up to a desired level and \( w_u \in \Omega_w \) such that

\[
f_u(\tilde{x}, w_u) = \begin{bmatrix}
B_{\epsilon_0}(1) < B_{\epsilon_1}(1) < \cdots < B_{\epsilon_n}(1) \\
B_{\epsilon_0}(0) < B_{\epsilon_1}(0) < \cdots < B_{\epsilon_n}(0)
\end{bmatrix} \quad \forall \tilde{x} \in S \subseteq E
\]

(5)

where \( B_{\epsilon_0}(1) = \{ r \in R : 1 - \epsilon_i < r \leq 1 \} \) and \( B_{\epsilon_i}(0) = \{ r \in R : 0 \leq r < \epsilon_i \} \)

The advantage of this setting is that a state ‘u’ can be identified by many \( \epsilon \)-identifiable functions whereas the least \( \epsilon \)-identifiable function is one with lowest \( \epsilon \) value that could be determined and corresponding \( w_u \) is the optimal weight matrix that could be identified with respect to the state ‘u’.

The collection of least \( \epsilon \)-identifiable functions for all \( s \in E \subseteq X \) computable by a neural network \( \mathcal{M} \) forms a class on a sample set \( E \) called the least \( \epsilon \)-identifiable function class and is denoted as \( H_M \). This act as the concept class for defining VC Dimension interms of \( \epsilon \) definitions.

\[
H_M = \left\{ f_u : \exists w_u \in \Omega_w \text{ and } f_u(\tilde{x}, w_u) = \begin{bmatrix}
B_{\epsilon_0}(1) \\
B_{\epsilon_i}(0)
\end{bmatrix} \quad \forall \tilde{x} \in S \subseteq E, \forall S \subseteq C \subseteq E \right\}
\]

(6)

Index of the \( \epsilon \)-identifiable function class \( H_M \) is the cardinality of the least \( \epsilon \)-identifiable function class \( H_M \) denoted as \( I(H_M) \).

The following result shows the properties of the index \( H_M \) in identifying a state ‘u’ by the neural network \( \mathcal{M} \).

Result 3.1:

For a sample set \( E \) with cardinality ‘r’, Index of the \( \epsilon \)-identifiable function class \( H_M \) has the following properties:

i. \( I(H_M) \leq 2^r \ \forall r \)

ii. If \( I(H_M) = 2^r \) then \( \forall S \subseteq C \subseteq E \) there exist \( f_u \) satisfying (11).

Proof:

If a state ‘u’ can be identified by neural network \( \mathcal{M} \) then there exist \( f_u \) corresponding to the state ‘u’. For a binary classification problem there exist a total of \( 2^r \) states for a sample set \( E \) with cardinality ‘r’. Hence maximum cardinality of least \( \epsilon \)-identifiable function class \( H_M \) is \( 2^r \).

Therefore \( I(H_M) \leq 2^r \ \forall r \).

If \( I(H_M) = 2^r \) then every state ‘u’ could be identified by \( \mathcal{M} \).

Hence there exist a least \( \epsilon \)-identifiable function \( f_u \) \( \forall S \subseteq C \subseteq E \).

For the sample population with cardinality \( |X| = M, \epsilon \) be the collection of subsets of \( X \) with cardinality \( r < M \).
\[ \epsilon = \{ E_p \subset X, |E_p| = r \forall p = 1, 2, \ldots, MG_r \} \]  

Vapnik and Chervonenkis, 1971, [12] proposed a general and standard tactic in depicting the complexity of a classification problem with respect to the sample space called VC Dimension. In terms of Index of the \( \epsilon \)-identifiable function class \( H_{\epsilon} \), the concept of VC Dimension could be illustrated as the dimension of the \( \epsilon \)-identifiable function space \( eL_M \) and is the largest value of \( \epsilon \) for which \( I(H_M) = 2^\epsilon \). It could be denoted as \( D(eL_M) \).

The difference in this setting is that the application of \( \epsilon \) concept brings more optimal approximation to the concept of VC Dimension.

Hence mathematically expressed as

\[ D(eL_M) = \max \{ r : I(H_M) = 2^r \text{ for some } E_p \in \epsilon \} \]  

(8)

Certainty of Neural Network \( \mathcal{M} \) in identifying a function \( f_u \) corresponding to a state with respect to sample set \( E \) lies in the dimension of the \( \epsilon \)-identifiable function space \( eL_M \) that can be proved by the following theorem.

**Theorem 3.2**

For a sample set \( E \) with cardinality \( r \) on an \( n \) dimensional space \( X = R^n \), if the dimension of the \( \epsilon \)-identifiable function space \( eL_M \) of Neural Network \( \mathcal{M} \) is \( D(eL_M) = r \) then for each state \( u \) corresponding to a sample set \( E \) there exist a function \( f_u^\epsilon \) and weight matrix \( w_u \in \Omega_w \) such that

\[ f_u (\tilde{x}, w_u) = \begin{cases} b_u(1) & \tilde{x} \in S \subset E \\ b_u(0) & \tilde{x} \in S^c \subset E \end{cases} \]  

(9)

\[ f_u^\epsilon (\tilde{x}, w_u) = \begin{cases} b_u(1) & \tilde{x} \in S \subset E \\ b_u(0) & \tilde{x} \in S^c \subset E \end{cases} \]  

(10)

\[ D(eL_{\epsilon}) \geq \frac{nC_p}{8} \log_2 \frac{C_p}{4} \]  

(11)

Combining the studies of Macintyre and Sonntag (1993), [43], Karpinski and Macintyre (1997), [40] and Goldberg and Jerrum (1995), [41] the upper bound on the VC Dimension for a feed forward Neural Network \( \mathcal{M} \) with \( W_T \) parameters, \( C_p \) computational units, linear threshold unit in the output layer and sigmoid functions in the hidden layer is fixed as

\[ D(eL_{\epsilon}) \leq (W_TC_p)^2 + 11W_TC_p \log_2(18W_TC_p^2) \]  

(12)

With the results obtained from the literature in (10) and (11) along with the explicit definition of VC Dimension as the dimension of the \( \epsilon \)-identifiable function space \( eL_M \), following inequality on the bounds on VC Dimension is obtained which is valid for a binary classification problem with real inputs that uses feed forward Neural Network topology with sigmoid activation functions in the hidden layer and linear function in the output layer.

\[ \frac{nC_p}{8} \log_2 \frac{C_p}{4} \leq D(eL_{\epsilon}) \leq D(eL_{\epsilon}) \leq (W_TC_p)^2 + 11W_TC_p \log_2(18W_TC_p^2) \]  

(13)

The next approach is to find the approximate bounds on the width of the network based on the inequality obtained in (12). Before finding the approximate bounds on the width of the network it is important to analyses whether such bounds exists and finite. The following theorems give a detailed proof on this regard.

**Theorem 3.3**

For a nonlinear classification with a two layered feedforward neural network \( \mathcal{M} \) as shown in Fig1 with total weights and threshold \( W_T \), total computational units \( C_p \) input nodes and \( m \) hidden nodes there exist a lower bound \( l_m \in R^+ \) over the width of the network \( \epsilon \) such that (12) holds and satisfies the condition \( C_p \leq 2^{\frac{n}{2}} \forall n \geq 4 \).

**Proof:**

By Result 3.1 of Sec 2.2, the certainty of Neural Network \( \mathcal{M} \) in identifying a function corresponding to a state \( u \) for a sample set with cardinality \( r \) is given by \( D(eL_{\epsilon}) \leq r \leq D(eL_{\epsilon}) \).

\[ \frac{n}{8} \log_2 \frac{C_p}{4} \leq r \leq Q(m)Q(m) + 11 \log_2(18(m + 1)Q(m)) \]  

(15)

Q(m) is a quadratic polynomial of the form

\[ Q(m) = (n + 2)m^2 + (n + 3)m + 1 \]  

(16)
Consider $r \leq Q(m)[Q(m) + 11 \log_2(18(m + 1)Q(m))]$
implies
\[ r \leq Q(m)[Q(m) + 11 \log_2(18Q(m)) + 11 \log_2(m + 1)] \]
Applying the condition $C_p = (m + 1) \leq 2^{n/2}$
implies $r \leq Q(m)[Q(m) + 11 \log_2(18Q(m)) + 11(n/2 - 2)]$
implies $199Q(m)^2 + 11\left(\frac{n}{2} - 2\right)Q(m) - r \geq 0 \tag{17}$
Consider $199Q(m)^2 + 11\left(\frac{n}{2} - 2\right)Q(m) - r = 0 \tag{18}$
Now Discriminant,\[ D = [11\left(\frac{n}{2} - 2\right)]^2 + 4 \times 199r > 0, \forall r > 0, n > 4 \tag{19} \]
Suppose $|\sqrt{D}| < 11\left(\frac{n}{2} - 2\right)$ then
\[ Q(m) = \frac{-11(n/2-2) + \sqrt{D}}{2 \times 199} < 0 \tag{20} \]
is a contraction since by (16) $Q(m)$ is positive. Hence $|\sqrt{D}| > 11\left(\frac{n}{2} - 2\right)$ and there exist a positive real root $\beta$
for $(18) \forall r > 0, n > 4$. Hence (17) implies $Q(m) \geq \beta \tag{21}$
Choose a real number $\gamma$ such that $Q(m) \geq \gamma > \beta$ and $\gamma > 1$.
Consider $Q(m) \geq \gamma \tag{22}$
Implies $(n + 2)m^2 + (n + 3)m - 1 - \gamma \geq 0 \tag{23}$
$(n + 2)m^2 + (n + 3)m + 1 - \gamma = 0 \tag{24}$
Discriminant of (23) is given by $d = (n + 3)^2 + 4(n + 2)(\gamma - 1) > 0$
Hence the roots are real and \[ m = \frac{-(n + 3) \pm \sqrt{d}}{2(n + 2)} \]
Suppose $|\sqrt{d}| < n + 3$
Implies $\sqrt{(n + 3)^2 + 4(n + 2)(\gamma - 1)} < n + 3$, contradiction to the assumption $\gamma > 1$. Hence $|\sqrt{d}| > n + 3$ and there exist a positive real root $l_m$ on the solution set of (24) and is a lower bound for the solution satisfying (23) and hence (17).

**Theorem 3.4**

For a nonlinearity function with a two layered feedforward neural network $\mathcal{M}$ as shown in Fig.1 with total weights and threshold $W_T$, total computational units $C_p$, $n$ input nodes and $m$ hidden nodes there exist an upper bound $L_m \in R^+$ over the width of the network $\mathcal{M}$ with respect to the size $r$ and dimension $n$ of sample space such that (12) holds and $L_m = \min{[k_1, k_2]}$

where $k_1 = \frac{16r}{n(n - 8)} - 1, k_2 = \frac{n}{2} - 2 - 1$ and $C_p \leq \frac{n}{2} - 2 \forall n > 8$. Also the sufficient condition for the existence of least upper bound $L_m^* \in R^+$ is given by $r = \frac{n}{2} - 6(n - 8)$.

**Proof:**

By (15) $\frac{n(m + 1)}{8} \log_2(\frac{m + 1}{4}) \leq r \tag{25}$
Also $m + 1 \leq 2^{n/2} \forall n > 8 \tag{26}$
Hence there exist a $N \in \mathbb{Z}^+$ such that $n2^{n/2}\left(\frac{n}{2} - 4\right) \leq 8r, \forall n \leq N$
implies $2^{n/2} - 1 \leq \frac{16r(n-n)}{n(n-8)} - 1 \forall n \leq N \tag{28}$
implies $m \leq \frac{16r(n-n)}{n(n-8)} - 1 \forall n \leq N \tag{29}$

From (26) and (29) $L_m = \min{\left\{\frac{16r}{n(n-8)} - 1, 2^{n/2} - 1\right\}} \forall n \leq N, n > 8 \tag{30}$

If (26) and (29) coincides we get a lower upper bound for the width of the network and is given by $r = 2^{n/6}n(n-8), n > 8 \tag{31}$
is a sufficient condition for the existence of least upper bound $L_m \in R^+$ on the width of the network with $n$ input nodes and $r$ samples.

Based on the conditions on theorem 3.3 and theorem 3.4 the bounds on the width of the network could be structured.

**Result 3.5**

The width of a single layered feed forward sigmoid network should be $l_m < m < L_m$ such that $C_p \leq 2^{n/2} \forall n > 8$ where $l_m$ is the least width possible that satisfies the relation $199Q(m)^2 + 11\left(\frac{n}{2} - 2\right)Q(m) - r \geq 0$. $Q(m) = (n + 2)m^2 + (n + 3)m + 1 - \gamma = 0$. $\gamma$ is given by $d = (n + 3)^2 + 4(n + 2)(\gamma - 1) > 0$.

Hence the roots are real and $m = \frac{-(n + 3) \pm \sqrt{d}}{2(n + 2)}$.
Suppose $|\sqrt{d}| < n + 3$
Implies $\sqrt{(n + 3)^2 + 4(n + 2)(\gamma - 1)} < n + 3$, contradiction to the assumption $\gamma > 1$. Hence $|\sqrt{d}| > n + 3$ and there exist a positive real root $l_m$ on the solution set of (24) and is a lower bound for the solution satisfying (23) and hence (17).

After fixing the network topology as explained in section 2.3 the optimization is carried out with backpropagation algorithm. Backpropagation gives a thorough perceptions on the overall behavior corresponding to the change of parameters of the network. The hidden layer activation function is log sigmoid and the output activation function is linear. Hence the function representing the forward pass of a fully connected single hidden layered network with an output node could be represented by (2). Performance of the algorithm is based on a measure called error function, mathematically expressed as $E(w_u) = \frac{1}{2r} \sum_j ||y_u(x) - f_u(w_u, x)||^2 \tag{32}$

Since there is only a single output node (32) could be further represented by $E(w_u) = \frac{1}{2r} \sum_j (y_u(x) - \sum_{k=1}^m w_{uk}^1 \sigma(x_k^1) + t_1)^2 \tag{33}$

Here $y_u(x)$ denotes the targeted output and the summation is taken over r samples. The data is processed by backpropagation algorithm to determine the optimal weights that reduces the error based on the error function using a gradient descent method. The success of backpropagation algorithm depends on
finding the direction to which the error function attains its global minimum.

The gradient of error with respect to the weights between the layers is given by

$$\frac{\partial E}{\partial w_{jk}} = -(1 + e^{-\sum_{i=1}^{M} w_{kj}x_i + t_k}) - 1$$

(34)

$$\frac{\partial E}{\partial x_i} = -\sum_{j=1}^{M} w_{kj} \frac{\partial E}{\partial w_{kj}}$$

(35)

$$\frac{\partial E}{\partial w_{kj}} = -w_{1k} \frac{e^{-\sum_{i=1}^{M} w_{ki}x_i + t_k}}{(1 + e^{-\sum_{i=1}^{M} w_{ki}x_i + t_k})^2} x_j$$

(36)

$$\frac{\partial E}{\partial t_k} = -w_{1k} \frac{e^{-\sum_{i=1}^{M} w_{ki}x_i + t_k}}{(1 + e^{-\sum_{i=1}^{M} w_{ki}x_i + t_k})^2}$$

(37)

The gradient of errors is used to update the weight of the network by gradient descent algorithm

$$w_{1k}^{t+1} = w_{1k}^t - \frac{n}{r} \sum_{i=1}^{N} \frac{\partial E}{\partial w_{1k}}$$

(38)

### 2.5 Dataset

The Network structured in section 2.3 is experimentally studied with three datasets that includes type of glass dataset, thyroid dataset and wine vintage dataset. The scope of the proposed study is limited to binary classification and hence the target attributes are designed accordingly. Description on each dataset and its attributes used for the study is shown in Table 1.

| Dataset          | Data Description                  | Data Inputs/Attributes                                                                 | Data Targets   | Cardinality of the Sample set |
|------------------|----------------------------------|---------------------------------------------------------------------------------------|----------------|-----------------------------|
| Types of Glass   | Classifies Glasses based on glass Chemistry | Refractive Index, Sodium, Magnesium, Aluminum, Silicon, Potassium, Calcium, Barium, Iron | Window Glass, Non Window Glass | 214                         |
| Thyroid Dataset  | Classify patients according to thyroid functioning based on clinical records | There are 21 data inputs based on clinical records with 15 binary and 6 continuous patient attributes | Normal, Abnormal | 7200                        |
| Wine Vintage     | Classifies wines from three wineries in Italy based on constituents found through chemical analysis | Alcohol, Malic Acid, Ash, Alkalinity of Ash, Magnesium, Total phenols, Flavanoids, Non Flavanoid Phenols, Color intensity, Hue, OD280/OD315 of diluted wines, Proline | Vineyard 1, Others | 178                         |

Source: UCI Machine learning Repository (http://mlearn.ics.uci.edu/MLRepository.html)

### 2.6 Performance Evaluation and Measures on Proximity of a Network \( \mathcal{M} \)

With the data sets explained in section 2.5 and the optimal algorithm decided as per section 2.4, the network structured in section 2.3 is evaluated experimentally with the measures on proximity. This paper analyses the proximity of the network \( \mathcal{M} \) based on two measures, Dissimilarity Measure and Mean Squared Error.

Let \( g_{1i}, g_{2i}, g_{3i} \) be a real valued function defined on discrete values \( i = 1, 2, 3, \ldots, M \). Then

$$d_M(g_{1i}, g_{2i}, g_{3i}) = \frac{1}{100} \left( \sum_{i=1}^{M} \sum_{j=1}^{3} |g_{ij} - g_{kj}| \right)$$

(42)

is the dissimilarity measure on a network \( \mathcal{M} \) if the following conditions hold.

i. \( d_M(g_{1i}, g_{2i}, g_{3i}) \geq 0 \)

ii. \( d_M(g_{1i}, g_{2i}, g_{3i}) = 0 \) iff \( g_{1i} = g_{2i} = g_{3i} \)
iii. $d_M(g_1, g_2, g_3) = d_M(g_2, g_1, g_3) = d_M(g_3, g_1, g_2) = d_M(g_1, g_2, g_3)$

Let $h_i, t_j$ be real valued functions defined on discrete values $i = 1, 2, 3, ..., r$. Then Mean Squared Error on a network $\mathcal{M}$ is defined as

$$E_r(\mathcal{M}) = \sum_{i=1}^{r} (h_i - t_i)^2$$

(43)

3. RESULTS

At first, suitable attribute dimension for a given sample size was studied on each dataset based on the theoretical analysis as per theorem 3.4 and the results are plotted on table 2. Application of theorem 3.3 gives the lower bound on the dimension of hidden layer and corresponding upper bound is obtained from table 2 for each dataset. The results are plotted separately on table 3.

| TABLE 2 | TABULAR RESULTS ON THEOREM 3.4 |
|---------|-------------------------------|
| Type of Glass | Lower Bound | Upper Bound |
| Types of Glass | 1 | 5 |
| Thyroid | 1 | 361 |
| Wine Vintage | 1 | 21 |

Next study was done experimentally with nntool box on Mathlab 2018 (b) software and the Network is trained by changing the hidden layer size within and outside the range specified in table 3 to see the best performance plot. In the case of thyroid dataset as the upper bound is large and network training till that range is computationally complex, the best performance is plotted up to a possible hidden layer size. The network training is done till the mean squared error is minimum and the training testing and the validation results obtained with the number of epochs in the x-axis and the mean squared error corresponding to each epoch against the y-axis coincides. The proximity of the graphs is measured based on a measure called measure of proximity of network $\mathcal{M}$ or Dissimilarity measure ($d_M$) as defined in Sec. 2.6. The Optimal Network is selected with Minimum $d_M$ and least Mean squared error. The best training performance on each width size is recorded in table 4 and its analysis is done by plotting bar graph (Fig 2, Fig 3, Fig 4) corresponding to the network width against the performance measures for three different datasets.
## TABLE 4
PERFORMANCE EVALUATION ON EACH DATASET

| Dataset               | Attribute | Network Width | MSE   | Dissimilarity Measure | Network Width | MSE   | Dissimilarity Measure | Network Width | MSE   | Dissimilarity Measure |
|-----------------------|-----------|---------------|-------|-----------------------|---------------|-------|-----------------------|---------------|-------|-----------------------|
| Types of Glass        | 9         | 1             | 0.072046 | 0.034059 | 4              | 0.18393 | 0.161554 | 7              | 0.081343 | 0.404684 |
|                       | 2         | 0.18006       | 0.15373 | 5              | 0.1836 | 0.132167 | 8              | 0.10097 | 0.17574 |
|                       | 3         | 0.11571       | 0.491896 | 6              | 0.076018 | 0.184104 | 9              | 0.089524 | 0.266313 |
| Thyroid Dataset       | 21        | 1             | 0.022802 | 0.012865 | 15             | 0.02713 | 0.018116 | 62             | 0.041717 | 0.120191 |
|                       |           | 2             | 0.022306 | 0.035519 | 18             | 0.027175 | 0.028797 | 65             | 0.047965 | 0.038426 |
|                       |           | 3             | 0.02755 | 0.007137 | 21             | 0.034139 | 0.12296 | 69             | 0.044806 | 0.048954 |
|                       |           | 4             | 0.021166 | 0.038158 | 27             | 0.028771 | 0.039152 | 70             | 0.038365 | 0.113186 |
|                       |           | 5             | 0.029079 | 0.022980 | 32             | 0.033619 | 0.081230 | 72             | 0.0407  | 0.062029 |
|                       |           | 6             | 0.025959 | 0.009491 | 36             | 0.037501 | 0.110415 | 75             | 0.034194 | 0.104610 |
|                       |           | 9             | 0.031908 | 0.040902 | 44             | 0.037868 | 0.062041 | 81             | 0.041241 | 0.064936 |
|                       |           | 11            | 0.026016 | 0.057034 | 50             | 0.039362 | 0.145853 | 95             | 0.041506 | 0.082120 |
|                       |           | 12            | 0.028535 | 0.055387 | 57             | 0.03251  | 0.123610 | 99             | 0.04137  | 0.062851 |
| Wine Vintage Dataset  | 13        | 1             | 0.18412 | 0.168229 | 8              | 0.075268 | 0.218679 | 18             | 0.089226 | 0.450785 |
|                       |           | 2             | 0.22897 | 0.111716 | 10             | 0.14376 | 0.141477 | 19             | 0.10944 | 0.473913 |
|                       |           | 3             | 0.21279 | 0.257802 | 11             | 0.099735 | 0.231491 | 20             | 0.072471 | 0.400476 |
|                       |           | 4             | 0.17069 | 0.178478 | 13             | 0.079918 | 0.194489 | 21             | 0.088083 | 0.359499 |
|                       |           | 5             | 0.20205 | 0.17549  | 15             | 0.078167 | 0.916329 | 24             | 0.089768 | 0.173866 |
|                       |           | 6             | 0.10006 | 0.257095 | 16             | 0.066525 | 0.301176 | 32             | 0.033028 | 0.448941 |
|                       |           | 7             | 0.091456 | 0.125131 | 17             | 0.087048 | 0.380771 | 45             | 0.048058 | 1.007583 |

![Fig 2: Performance of Neural network on Types of Glass Dataset](types-of-glass-dataset.png)
4. DISCUSSION

The studies done by [8],[9],[10] related to network size do not provide an explicit result on the size of the network and assume unlimited number of network units. Many common networks with a single hidden layer have universal approximation property and can perform highly well than deep ones. The theoretical (Table 2) as well as the experimental results (Fig 2,3,4) in this paper shows the approximate width is proportional to the dimension of dataset. The condition $C_p \leq 2^{\frac{n}{2}} \forall n \geq 8$ implies the appropriate attribute size is when $k_1 < k_2$. The upper bound $L_m$ on the dimension of the hidden layer is $L_m = \min(k_1, k_2)$. By the
application of theorem 3.4 minimum number of nodes in the hidden layer for each data set is at least one. As per the results obtained on table 2 the attribute size and the suggested range on the dimension of hidden layer for each dataset is shown in table 5. Types of Glass dataset requires at least 14 attributes with hidden layer width range as [1,31]. A much more suitable sample dimension will be 21 attributes with hidden width range as [1,11]. In the case of thyroid dataset 22 attributes is needed but it corresponds to a high width range [1,373]. This is computationally expensive and hence more suitable structure of thyroid dataset will be attribute size 45 against the width range [1,93]. 15 attributes is the minimum requirement for wine vintage dataset with hidden layer width range as [1,26]. A more appropriate one with lesser training complexity will be 20 against the range [1,10]. Hence more attributes needed to be identified depending on the sample size for reducing training complexity. Since increasing the attribute size is beyond the scope of this paper, it is sufficient to find the hidden layer width range corresponding to the dataset dimension given in table 1. Table 3 gives the network width range on three given datasets. Experiment analysis done with Mathlab 2018(b) software to analyze whether the width of the hidden layer of the network that shows best performance falls within the range specified in table 3. This is experimentally studied on the basis of the performance plot obtained on each training. Bars with low MSE and Low Dissimilarity measure are considered to be optimal plot. The results obtained in table 4 are analysed with bar graphs shown in fig 2, fig 3, fig 4 and optimal hidden layer width is recorded in table 6.

### Table 5

| Attribute Size | Hidden Layer Width |
|----------------|--------------------|
| Types of Glass | [1,31]             |
| Types of Glass | [1,33]             |
| Types of Glass | [1,25]             |
| Types of Glass | [1,21]             |
| Types of Glass | [1,24]             |
| Types of Glass | [1,29]             |
| Types of Glass | [1,20]             |
| Types of Glass | [1,13]             |
| Types of Glass | [1,11]             |
| Types of Glass | [1,10]             |
| Types of Glass | [1,9]              |
| Types of Glass | [1,8]              |
| Types of Glass | [1,7]              |
| Types of Glass | [1,6]              |
| Types of Glass | [1,5]              |
| Types of Glass | [1,4]              |
| Types of Glass | [1,3]              |
| Types of Glass | [1,2]              |
| Types of Glass | [1,1]              |
| Types of Glass | [1,0]              |
| Types of Glass | [0,9]              |
| Types of Glass | [0,8]              |
| Types of Glass | [0,7]              |
| Types of Glass | [0,6]              |
| Types of Glass | [0,5]              |
| Types of Glass | [0,4]              |
| Types of Glass | [0,3]              |
| Types of Glass | [0,2]              |
| Types of Glass | [0,1]              |
| Types of Glass | [0,0]              |

### Table 6

| Dataset        | Input Size | Sample Size | Optimal Network Width |
|----------------|------------|-------------|-----------------------|
| Types of Glass | 9          | 214         | 1                     |
| Thyroid        | 21         | 7200        | 1,3,6                 |
| Wine Vintage   | 13         | 178         | 7                     |

### 5. Conclusion

VC Dimension is the most widely used and acceptable measure that can quantify the complexity of functions in recognizing patterns computable by neural network. Since this concept is probabilistic in nature it lacks high intellectual decisions. If this concept is approached in an analytical way it can be applied theoretically to study the complexity of functions and structure the network accordingly. The paper analyses the VC Dimension in an ε—identifiable function space thereby creating a way to size the network according to sample dimension. Theorems formulated in this paper provides conditions on the size of the sample space to have a minimum degree of training complexity with sufficient number of parameters. This eliminates the unnecessary computations due to large number of units in the hidden layer by limiting the hidden layer size within a finite bounds. This could be further extended by the normalization of inputs and regularization of weights which provides a fine tuning to the optimization process. This technique can be generalized to any binary classification problem irrespective of sample distribution and size. The work is limited in its network architecture, binary classification problem and activation functions. Also the experimental analysis is limited to three datasets and more datasets with varying dimensions need to be experimented in different software techniques which is beyond the scope of this paper.
REFERENCES

[1] Schmidhuber,Jürgen(2015). Deep learning in neural networks: an overview. Neural Networks, 61, 85–117.

[2] LeCun, Yann, Bengio, Yoshua, & Hinton, Geoffrey (2015). Deep learning. Nature, 521(7553), 436–444.

[3] Bengio Y, LeCun Y (2007) Scaling learning algorithms towards AI. In: Large-scale kernel machines. MIT Press, Cambridge.

[4] Ba LJ, Caruana R (2014) Do deep networks really need to be deep? In: Ghahramani Z (ed) Advances in neural information processing systems, vol 27. MIT Press, Cambridge, pp 1–9.

[5] Bianchini M, Scarselli F (2014) On the complexity of neural network classifiers: a comparison between shallow and deep architectures. IEEE Trans Neural Netw Learn Syst 25:1553–1565.

[6] Mhaskar H, Liao Q, Poggio T (2016) Learning functions: when is deep better than shallow. CBMM Memo No. 045, May 31, 2016. https://arxiv.org/pdf/1603.00988v4.pdf.

[7] Mhaskar H, Liao Q, Poggio T (2016) Learning real and Boolean functions: when is deep better than shallow. CBMM Memo No.45, March 4, 2016. https://arxiv.org/pdf/1603.00988v1.pdf.

[8] Poggio T, Mhaskar H, Rosasco L, Miranda B, Liao Q (2017) Why and when can deep—but not shallow—networks avoid the curse of dimensionality: a review. Int J Autom Comput. https://doi.org/10.1007/s11633-017-1054-2.

[9] Ito Y (1992) Finite mapping by neural networks and truth functions. Math Sci 17:69–77.

[10] Pinkus A (1999) Approximation theory of the MLP model in neural networks. Acta Numer 8:151–166.

[11] Fine TL (1999) Feedforward neural network methodology. Springer, Berlin.

[12] V. N. Vapnik and A. Ya. Chervonenkis (1971). On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability & Its Applications, 16(2):264–280, doi:10.1137/1116025.

[13] Kitaev, A. Yu., Shen, A. H., & Vyalyi, M. N. (2002). Classical and quantum computation. ISBN: 0821832298, Boston, MA, USA: American Mathematical Society.

[14] Dawson, C. M., & Nielsen, M. A. (2006). The Solomonoff-Kitaev algorithm. Quantum Information & Computation, 6(1), 81–95.

[15] Delalleau, Olivier, & Bengio, Yoshua (2011). Shallow vs. deep sum-product networks. Advances in Neural Information Processing Systems. (pp.666–674).

[16] Montufar,GuidoF.,Pascanu,Razvan,Cho,Kyunghyun,&Bengio,Yoshua (2014). On the number of linear regions of deep neural networks. InAdvances in Neural Information Processing Systems (pp.2924–2932).

[17] Bianchini,Monica,&Scarselli, Franco(2014). On the complexity of neural network classifiers: a comparison between shallow and deep architectures. IEEE Transactions on Neural Networks and Learning Systems, 25(8),1553–1565.

[18] Telgarsky, Matus (2015). Representation Benefits of Deep Feedforward Networks. ArXivPreprintArXiv:1509.08101.

[19] Telgarsky,Matus(2016). Benefits of depth in neural networks. ArXivPreprintArXiv:1602.04485. V.N. Vapnik “The Nature of Statistical Machine”, Springer-Verlag Press, 1995.

[20] Eldan R and Shamir O (2015). The power of depth for feedforward neural networks. arXiv preprint arXiv:1512.03965.

[21] Raghu, Maithra, Poole, Ben, Kleinberg, Jon, Ganguli, Surya, & Sohl-Dickstein, Jascha (2016). On the expressive power of deep neural networks. ArXivPreprintArXiv:1606.05336.

[22] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems, 2, 303–314.

[23] Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. Neural Networks, 2, 183–192.

[24] Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 2, 359–366.

[25] Hornik K (1991). Approximation capabilities of multilayer feedforwardnetworks. Neural networks.

[26] Chui C.K. and Li. X. (1992). Approximation by ridge functions and neural networks with one hidden layer. Journal of Approximation Theory.

[27] Barron, A. R. (1993) Universal approximation bounds for superpositions of a sigmoidal function. IEEE Transactions on Information theory.

[28] Bartlett, P. L. (1998). The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. IEEE Transactions on Information Theory, 44, 525–536.

[29] Anthony, M. and Bartlett, P. L. (2009). Neural Network Learning: Theoretical Foundations. Cambridge University Press.

[30] Poggio T., Rosasco L., Shashua A., Cohen N., and Anselmi. F. (2015). Notes on hierarchical splines, dclns and i-theory. Technical report, Center for Brains, Minds and Machines (CBMM).

[31] Dinh, L., Pascanu, R., Bengio, S. and Bengio, Y. (2017). Sharp minima can generalize for deep nets. arXiv preprint arXiv:1703.04933.

[32] Bartlett, P. L., Foster, D. J. and Telgarsky, M. J. (2017). Spectrally-normalized margin bounds for neural networks. In Advances in Neural Information Processing Systems.

[33] Golowich,N.,Rakhlin,A.andShamir,O.(2017). Size-independent sample complexity of neural networks. arXiv preprint arXiv:1712.06541.

[34] Neyshabur, B., Tomioka, R. and Srebro, N. (2015). Norm-based capacity control in neural networks. In Conference on Learning Theory.

[35] Neyshabur, B., Bhojanapalli, S., McAllester, D. and Srebro, N. (2017). A pac-bayesian approach to spectrally-
normalized margin bounds for neural networks. arXiv preprint arXiv:1707.09564.

[36] Baum E. B, and Haussler D (1989). What size net gives valid generalization? Neural Computation, 1(1):151–160.

[37] Maass W. (1994) Neural nets with superlinear VC-dimension. Neural Computation, 6(5):877–884.

[38] Sakurai, Akito (1999). Tight Bounds for the VC-Dimension of Piecewise Polynomial Networks. In Advances in Neural Information Processing Systems (pp. 323–329).

[39] Koiran P. and Sontag E.D. (1997). Neural Networks with Quadratic VC Dimension. Journal of Computer and System Sciences, 54(1):190–198.

[40] Karpinski, M., & Macintyre, A. (1997). Polynomial bounds for vc dimension of sigmoidal and general pfaffian neural networks. Journal of Computer and System Sciences, 54(1), 169–176.

[41] Goldberg P.W. and Jerrum M.R. (1995) Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers. Machine Learning, 18(2):131–148. (Conference version in COLT’93).

[42] Sontag E.D., (1992). Feedforward nets for interpolation and classification. J. Comput. System Sci., 45(1):20-48, 1992.

[43] Macintyre A. and Sontag E.D (1993). Finiteness results for sigmoidal neural networks. In STOC ‘93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing, New York, NY, USA, pages 325-334, 1993. ACM Press.