Designing Neural Networks for Tackling Hard Classification Problems

GEORGIOS LAPPAS
Technological Educational Institution of Western Macedonia
Kastoria Campus, PO Box 30, GR52100 Kastoria
GREECE
lappas@kastoria.teikoz.gr

Abstract: When designing neural networks for tackling hard classification problems researchers face the trivial problem of deciding the appropriate size of the neural network. The problem of optimizing the size of a neural network for obtaining high classification accuracy in datasets is indeed a hard problem in the literature. Existing studies provide theoretical upper bounds on the size of neural networks that are unrealistic to implement. Alternatively, optimizing empirically the neural network size may need a large number of experiments, which due to a considerable number of free parameters may become a real hard task in time and effort to accomplish. Hard classification problems are usually large in size datasets. Such datasets derive from collection of real world data like from multimedia content and are usually rich in training samples and rich in features that describe each collected sample. Working with neural networks and hard classification datasets will make even harder the task to optimize the neural network size. This work highlights on a mathematical formula for *a priori* calculating the size of a neural network for achieving high classification accuracy rate. The formula estimates neural networks size based only on the number of available training samples, resulting in sizes of neural networks that are realistic to implement. Using this formula in hard classification datasets aims to fix the size of an accurate neural network and allows researchers to concentrate on other aspects of their experiments. The focus on this approach turns to the number of available data for training the neural network, which is a new perspective in the neural network theory and the characteristics of this perspective are discussed in this article for designing neural networks for tackling hard classification problems.

Key–Words: Neural Networks, Circuit Complexity, Multimedia, Intelligent Multimedia, Pattern Classification.

1 Introduction

Neural networks are widely used in pattern classification. Optimizing the size of a neural network is a major task in machine learning. The number of neurons or circuit gates has to ensure a high generalization capability of the classifier on new unseen data. This problem has been theoretically and experimentally investigated over a long period of time [8, 35, 37, 38].

In order for neural networks to approximate a high classification rate, which is equivalent to low error rates, to a specific classification problem it usually requires a certain amount of adjustments, which are apparently unavoidable in the light of the No Free Lunch Theorems [51]. The need for tuning network parameters when the classification accuracy is very important may lead us to lots of experiments for establishing a high classification rate [5, 6]. A piori estimation of these parameters may help in reducing the time and effort to find an appropriate neural network for achieving high classification accuracy. An important parameter adjustment related to neural networks implementations is the number of hidden units (network size) that should be trained in order to approximate a function that best describes the training data. The problem of finding the smallest network that can realize an arbitrary function given a set of m vectors in n dimensions defines the *circuit complexity* problem [15]. The circuit complexity problem is of great importance in parallel processing for hardware implementations and tries to find tight bounds for the number of units used to realize an arbitrary function.

On non-seperable problems, finding the best function for minimizing the classification error is an NP-complete problem. Blum and Rivest [16] proved that even for a very small network to find weights and thresholds that learn any given set of training examples is an NP-complete problem. Höffgen and Simon [23] deal with the problem of learning a probably almost optimal weight vector for a neuron, finding that it is an NP-complete problem. Also finding an optimum network configuration for solving combinatorial optimization problems is not an easy task [53].

Thus, establishing an optimal weight configuration of threshold units cannot be executed in efficient time and only approximations of optimum solutions can be achieved.

Theoretical analysis on lower and upper bounds on VC-dimension [48] has been performed for some type of networks. Baum and Haussler [12, 13] show that in a feedforward network with W weights and N computational threshold units the VC-dimension is bounded $VC \leq 2 \cdot W \cdot log_2(e \cdot N)$. Maas [36] has obtained a lower bound $Wlog_2W$ for certain types of multilayer feedforward network of threshold units. Other tight bounds for specific type of functions have been obtained in [10, 11, 20, 27, 28, 46, 50]; see Anthony and Bartlett [9] for a review of the field.

Bounding VC-dimensions is a challenging task in mathematical investigations of neural networks which provides a number of sophisticated mathematical tools. The bounds, however, tend to be too large, since they provide such guarantees of generalisation for any probability distribution of training examples and for any training algorithm that minimizes the training error on the training examples. Therefore, the VC-dimension is a very general theoretical measure of pattern classification ability. The proposed bounds are usually unrealistic and with limited practical value for real world problems and for multimedia datasets.

Estimating the size of neural networks is important [32] as: a) it allows us to *a piori* set one of the difficult problem dependent parameters, and hence we can focus on the rest of parameters in a complex classification problem. b) it is important in parallel processing theory in terms of resources required to approximate best classification rates.

Hard classification datasets like multimedia datasets usually consist of large number of samples. Considering also a large number of desrciptive features for each sample it implies that optimizing the size of neural networks applied to multimedia datasets is a hard task in time and in number of experiments and only approximations of optimal solutions can be achieved [31]. Providing thus a method for an *a priori* estimation of the number of parallel processors needed for achieving accurate performance may be of critical value in hardware design and implementation of intelligent multimedia applications.

Other important applications may be in robotics [40] for building accurate machines that need to solve interface decision problems (automated pilots) or solving human-computer interface problems such as automated speech recognition, optical character recognition, camera images classification, and face recognition. Multimedia web mining as well as building multimedia intelligent agents for the web are also interesting application areas for accuracy in recognizing, classifying and processing multimedia data on the web, which is important for the semantic web and for e-commerce applications [45].

2 Estimating Neural Network Sizes

If we consider the asymptotically optimal upper bound of $2 \cdot \sqrt{2^n/n}$ linear threshold gates of unbounded fan-in for arbitrary Boolean functions $f(x_1, ..., x_n)$ proved in [34], then we obtain for the splice-junction gene sequences database (SJGSD) from the UCI Machine Learning Repository[1] with $n = 60 \cdot 2 = 120$ in binary encoding approximately 2.1×10^{17} gates. This number yields to an unrealistic neural network implementation.

In most cases of multimedia datasets and real world problems, the sample data provide only a tiny fraction of the theoretically possible number of function values. The SJGSD dataset for instance has 3190 samples. Each sample in binary encoding has length $n = 120$. A priori, we can argue that not all combinations of binary inputs are feasible (or even a small fraction only has indeed a valid interpretation). Only a fraction of binary inputs of length $n = 120$ is feasible in the given context, and, moreover, only $m_L = 3190 < 2^{12}$ vectors out of the hypothetical number of 2^{120} are provided by the SJGSD.

The 3190 vectors can be enumerated by $n_L = 12$ bits. If we now design a classification circuit of two main components, where the first component returns the n_L bits of the binary encoding of input vectors of length $n = 120$ and the second component produces the associated classification (we assume for a moment for one class only), then the size of the second component can be estimated by only 37 gates according to $2 \cdot \sqrt{2^{n_L}/n_L}$. Thus using the first component we can turn our focus from the number of sample features $n = 120$ to the number of bits to enumerate the 3190 available training samples. Therefore the network learning capacity is now related to the number of existing training samples instead of the number of existing features on the samples. In this way we can now use the existing theoretical upper bounds leading to have classification circuits (neural networks) with feasible to implement number of units. However, two issues have been neglected at this point: the size of the encoding circuit has to be added, and $2 \cdot \sqrt{2^{n_L}/n_L}$ is an asymptotic formula and yields for large values of n_L only. These problems have been addressed in [30], where the empirical hypothesis

$$8 \cdot \sqrt{2^{n_L}/n_L} \quad (1)$$

[1] http:// www.ics.uci.edu/∼mlearn/MLRepository.html

has been established for $n_L = \log(\text{size of sample set})$, which implies for the SJGSD-problem a gate number of 148 . For ourdays computers, designing neural networks that consist of 148 neurons (or else hidden units, or threshold gates, or threshold units or gates) is not a problem, leading indeed to feasible neural network implementations.

Equation (1) has been further analysed [2] and a further in-depth analysis of the complexity of the encoding circuit and the actual classification circuit established the following theorem in [3] for neural network sizes that yield to high classification accuracy :

Theorem 1 *Given a classification problem defined by m_L training samples and an overall size of sample data $m \leq 3 \cdot m_L/2$, then there exist unbounded fan-in threshold circuits with*

$$\lceil 1.25 \cdot S^R \rceil \text{ or with } , \lceil 0.07 \cdot S^L \rceil \qquad (2)$$

threshold gates that provide a high generalization rate on all m sample data, where $S^R := 2 \cdot \sqrt{2^{n_L}} + 3$ and $S^L := 34.8 \cdot \sqrt{2^{n_L}} + 14 \cdot n_L - 11 \cdot \log_2 n_L + 2$ for $n_L = \lceil \log_2 m_L \rceil$.

Readers will find all the in depth analysis for the above theorem in [3]. The resulting a priori estimation of the complexity of classification circuits has then been evaluated in [3] by an *a posteriori* analysis of best classification results published in the literature. The posteriori analysis showed that eq. (2) is applicable to various types of networks and various artificial neural structures. Notable that Eq. (2) is applicable to hard (non-separable) classification problems only, like real world problems and multimedia classification datasets. For "solvable problems" (linearly separable), Eq. (2) overestimates the number of threshold gates significantly as a "single perceptron" can solve the problem like in the Mushroom Dataset from the UCI Machine Learning Repository. As Baum [14] has shown if the sample set is linearly separable it is likely that the perceptron algorithm will find a highly accurate approximation of a solution vector in polynomial time.

In the rest of this work it will be presented a) the experimental background of the above formula, b) an *a posteriori* evaluation of the formula on best classification results published in the literature, c) an application of the formula for proposing neural network sizes for achieving high classification rates in famous benchmarking multimedia datasets d) a discussion of the different perspective offered by this work when working with neural network optimization.

3 Methodology

Threshold circuits consist of a single threshold gate at the root with AND gates at the next level. The basic computing unit is the *threshold logic unit* which was introduced by McCullogh and Pitts in 1943 [39]. The threshold unit forms the sum of the inner product between the input pattern, which represents $X_1, X_2, ..., X_n$ features, and the connection strengths $w_1, .., w_n$. The sum is compared to a threshold θ, and the threshold unit outputs only one of two values, which is either 0,1 or -1,+1. The threshold unit together with the perceptron learning algorithm can learn two classes, if they are linearly separable. Nonlinearly separable classes, which are the most frequent in real world problems, require more complex decision surfaces and can be solved by threshold circuits of a more complex nature.

The experimental analysis that led us to Eq. (2) was performed by using the LSA machine [4, 7]. LSA machine is a neural network that combines the classical perceptron algorithm [43] with a specific type of simulated annealing [21] as the stochastic local search procedure for finding threshold gates of a classification circuit. Simulated annealing is an established method for searching near oprimal solutions [33]. The simulated annealing procedure employs a logarithmic cooling schedule $c(k) = \Phi / \ln(k+2)$ search strategy where the "temperature" decreases at each step. The simulated annealing-based extension of the perceptron algorithm is activated when the number of misclassified examples increases for the new hypothesis compared to the previous one. In this case, a random decision is made according to the rules of simulated annealing. If the new hypothesis is rejected, a random choice is made among the misclassified examples for the calculation of the next hypothesis. A detailed description on the LSA machine can be found in [4, 7].

For our experiments we used depth-two circuits where depth-one consists of computational units P_t and depth-two of a voting function that decides the output class (Figure 1). The size of this neural network is $S = t + 1$. Following notations from the circuit complexity theory we don't count the input layer X as a depth-one layer. Therefore a depth-one threshold circuit is considered as a depth-two neural network with one hidden unit based on the neural model. Thus, (Figure 1) is a depth-3 neural network with t hidden units.

The output gate at the depth-2 calculates $|\{j : f_j(\vec{x}) = 1|/t$ where t is the number of threshold units at depth-1 and $|\{j : f_j(\vec{x}) = 1|$ denotes the total number of threshold units at depth-1 that decide for a positive example. Therefore, outputs at depth-1 are col-

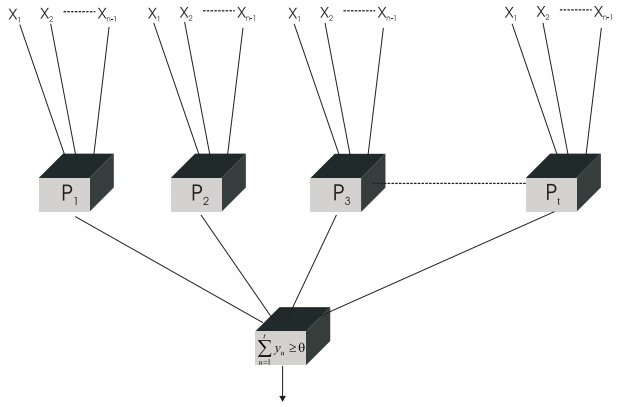

Figure 1: Depth 2 circuit with $t + 1$ threshold gates.

lected and the class decision is finally denoted by the voting function at depth-2.

Depth-four networks (Figure 2) with special procedures to train gates at larger depths have been also analyzed with the LSA machine leading approximately to the same size of networks for achieving low error rates [30]. The circle in (Figure 2) is the equivalent newtork as shown in (Figure 1) and has been drawn to compare the network sizes between Figure 1 and Figure 2. In these larger depth networks training is performed at odd depths (depth-1 and depth-3), whereas at even depths outputs generate new samples for training next depth gates. Details on the training procedure and the generation of new samples for training next depth gates is provided in [30].

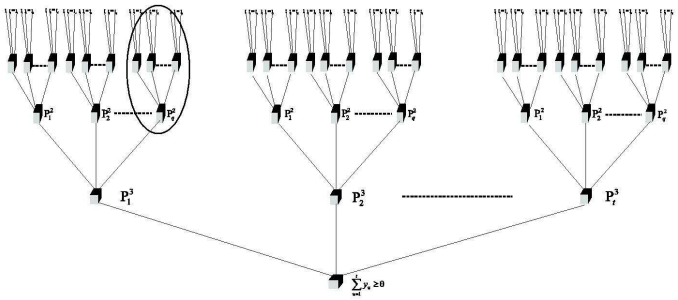

Figure 2: Regular structure of depth 4 circuits.

4 Experiments

For the experiments we chose two datasets from the UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/MLRepository.html), where one of the datasets induces three classes, leading us with a total of four datasets for our experiments. These datasets have been widely used as benchmark datasets.

a) Splice-junction Gene Sequences Database (SJGSD) Splice junctions are points on a DNA sequence at which "superfluous" DNA is removed during the process of protein creation in higher organisms. The problem posed in this dataset is to recognize, given a sequence of DNA, the boundaries between exons (the parts of the DNA sequence retained after splicing) and introns (the parts of the DNA sequence that are spliced out). This problem consists of two subtasks: recognizing exon/intron boundaries (referred to as EI sites), and recognizing intron/exon boundaries (IE sites). Additionally, a third class is introduced which is referred to as "Neither". Given a position in the middle of a sequence window, 60 DNA sequence elements are used to decide if this is an IE, EI, or "Neither" class. The database consists of 3190 vectors representing 60 attributes. The class distribution is: 25% for IE (767 instances); 25% for EI (768 instances); and 50% for "Neither" (1655 instances).

In order to discriminate between the three classes, we introduce three databases, each related to a single class as positive examples: the "IE database" consists of 767 positive examples (IE class) and 2423 negative examples (union of EI class and "Neither" class); the "EI database" consists of 768 positive examples and 2422 negative examples; the "Neither database" consists of 1655 positive examples and 1535 negative examples.

b) Wisconsin Breast Cancer Database (WBCD) The WBCD database is the result of efforts made at the University of Wisconsin Hospital for accurately diagnosing breast masses based solely on a Fine Needle Aspiration (FNA) test. WBCD is a binary classification problem where each vector represents 9 features. The output indicates either a benign case (positive example) or a malignant case (negative example). The data set consists of 699 samples, where 16 samples have missing values which are discarded in a preprocessing step. The remaining 683 data are divided into 444 benign and 239 malignant cases. Many researchers have tackled this dataset with reported results ranging from 96% to about 99%; cf. [42, 44, 52]. The combination of the classical perceptron algorithm with logarithm simulated annealing [4] results in $\approx$ 98.8% correct classification.

In our experiments the dataset m of a classification problem is divided into two disjoint sets of data: a) the training dataset $|m_L| = 2|m|/3$ for training the neurons and b) the testing dataset $|m_T| = |m|/3$ for evaluating the classification accuracy of the network.

The partition of the dataset into 2/3 of the data to be in the training set m_L, and the rest 1/3 of the data to be in the test set m_T for estimating the classification accuracy is a common pattern in machine learning

[41]. The same partition of data is used for the SJGSD datasets by Rampone in [41].

Each neuron at depth-one is trained by a sample set of size p randomly drawn from the available training data set m_L. In order to train a neuron j, i.e. to calculate a function of the type $f_j = \sum w_i \cdot x_i \geq \vartheta$, where $i = 1, .., n$ is the number of input gates in threshold unit j, and $j = 1, .., t$, we use a sample set m_L^j, which is randomly sampled from m_L, associated with threshold unit j. The sample size p is a parameter in our experiments. The parameter p is experimentally determined in our approach. The experimental determination of p is a usual method used by researchers for approximating values of p that provide high classification accuracy [47].

According to the No-Free-Lunch-Theorems [51], the performance of learning algorithms is problem-dependent. Four problem-dependent parameters, the network size, the sample size p, the length of inhomogeneous Markov chain k and the constant Φ of the simulated annealing cooling schedule, are required for our experiments with the LSA machine. A large number of experiments have been carried out to fine-tune the values of the problem-dependent parameters p, k and Φ for each dataset.

Parameter k determines the number of steps when searching for better solutions in the simulated annealing process. Experimental results from [1] suggest $k \leq (h/\delta)^{\Gamma}$ for the number of transitions in the local search procedure, where $1 - \delta$ is a confidence parameter, i.e. after k steps the probability to be in an optimum solution is at least $1 - \delta$, and h is the size of neighbourhoods. However, this might result to extremely high values for k, even for small values of Γ and thus is a parameter in our experiments. The constant Φ determines an escape from local minima in the simulated annnealing process and is expressed in terms of a percentage of the size p, i.e. $\Phi = G \cdot p$, where $G \in (0, 1)$.

As demonstrated in [30], the results on these datasets for various sets of parameters led to the conclusion that depth-two and depth-four classification circuits provide approximately the same generalization results if the number of neurons is approximately the same and close to $8 \cdot \sqrt{2^{n_L}/n_L}$ in both types of circuits. The reader may the experimental analysis for the upper bound $8 \cdot \sqrt{2^{n_L}/n_L}$ in [30].

We tested neural network sizes of $S = 8 \cdot \sqrt{2^{n_L}/n_L}$ against half and double size of S on the following datasets from the UCI Machine Learning Repository using the LSA Machine [4, 7]:

a) Hayes-Roth Datasets (Hayes): Barbara and Frederick Hayes-Roth [22] created this dataset for recognition and classification of exemplars. The Hayes-Roth datasets contains three classes Hayes-1, Hayes-2 and Hayes-3. The class Hayes-3 is linearly separable and therefore our focus is on Hayes-1 and Hayes-2. For each of the two classes, the data from the UCI Repository consist of 132 samples and 28 test samples. We merged the two files and used 2/3 of the data for training (i.e $m_L = 106$ and $n_L = 7$) in both Hayes-1 and Hayes-2, and 1/3 of the data for testing. The original 28 test samples were part of the test set.

b) Iris Plant Datasets (Iris): R.A. Fishers [6] Iris Plant Dataset is the oldest and perhaps the most frequently used dataset in machine learning with innumerable publications of results in the pattern classification literature. The set consists of 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two; the latter are not linearly separable from each other. We denote the three classification problems, as Iris-1, Iris-2 and Iris-3, and we will focus on the non-seperable Iris-2 and Iris-3 classification problems. In our experiment we used $m_L = 100$ randomly selected training examples and 50 examples for calculating the classification error.

c) US Congressional Voting Records Database (Votes): The US Congressional Voting Records (Votes) dataset includes votes for each of the U.S. House of Representatives Congressmen on 16 key issues (attributes) identified by the Congressional Quarterly Almanaco (CGA) in 1984. The dataset consists of 435 samples of two classes (267 Democrats, 168 Republicans).

d) Waveform Datasets (Wave): The Waveform Dataset consists of three classes, i.e. introducing three binary dataset (Waveform 1, Waveform 2, Waveform 3), where each class is a wave generated from a combination of 2 out of 3 base waves. Each of the three datasets contains 5, 000 samples representing 21 attributes with continuous values between 0.0 and 6.0. We emphasize that the binary encoding of continuous values does not affect the complexity bound, since the bound depends only on the enumeration of samples. The distribution of positive and negative examples is approximately 1:2 in all three datasets.

In Table 1 we compare the size $S = 8 \cdot \sqrt{2^{n_L}/n_L}$ to half of circuit sizes and double circuit sizes with respect to fine-tuned parameters p, k, Φ. The error on test data e_T is reported for all datasets.

The comparison shows that the upper bound $8 \cdot \sqrt{2^{n_L}/n_L}$ for estimating the circuit size S does indeed imply the highest classification rates. Except for the 'Hayes-Roth 2' dataset, where a 10.4 classification rate is obtained for double circuit size, the highest classification rate is obtained for $S = 8 \cdot \sqrt{2^{n_L}/n_L}$.

It is also notable that LSA machine is very competative to existing classification methods as almost

Dataset	e_T for S	e_T for $S/2$	e_T for $2 \cdot S$
Hayes-Roth 1	11.1%	11.9%	11.1%
Hayes-Roth 2	11.1%	11.9%	10.4%
Iris 2	0.7%	0.7%	0.7%
Iris 3	0.7%	0.7%	0.7%
Votes	3.2%	3.4%	3.3%
Waveform 1	14.4%	14.7%	14.8%
Waveform 2	11.2%	11.7%	11.4%
Waveform 3	11.4%	11.6%	11.4%

Table 1: Classification Errors e_T on the Test dataset for S, $S/2$, and $2 \cdot S$, (Depth-two Circuits).

all classification results on the above datasets are at least as good to those reported in the literature or even outperform them [30, 31].

5 Evaluation on Multimedia Datasets

An in-depth analysis by using existing theory and experimental results from all the above 12 datasets (3 SJGSD, WBCD, 3 Waveform, 2 Hayes-Roth, 2 Iris and Votes) led [3] to a more accurate result as described by Theorem 1 and equation (2). As shown in [3] both estimations $\lceil 1.25 \cdot S^R \rceil$ and $\lceil 0.07 \cdot S^L \rceil$ could be used to estimate *a priori* the size of neural networks that is sufficient to achieve a high rate of correct classification.

Evaluation of equation (2) has been *a posteriori* carried out by performing analysis of best classification results published in the literature. The following multimedia datasets where used:
a) Traffic Sign Recognition Dataset (TSRD). Used in [49] for the study of different classification techniques applied to the traffic sign recognition problem. The data consist of eight types of circular traffic signs well known from the international traffic code. The data have been collected by a video sequence and are initially recognised as images of signs called blob. There are 235 blobs of 3030 pixels each, where each pixel has information about three colours. Therefore each sample consists of 2,700 attributes. The authors provide results for gate numbers ranging from 10 to 82 in Multi-layer Perceptron Networks (MLPNs).
b) High Range Resolution Radar Datasets (HR3D): Used in [18] for evaluating various neural networks applied to the Automatic Target Recognition problem, where one has to classify patterns that belong to six classes of high range resolution radar profiles. Each of the six classes corresponds to a specific type of aircraft. The dataset has 1, 071 samples with 128 input variables for each aircraft, i.e. there are 6, 426 samples available. Results are provided by authors for MLPNs trained with the Levenberg-Marquardt algorithm and for Radial Basis Function (RBF) networks.
c) Detection of Neonatal Seizures (DNS) Used in [26] and consists of 240 feature vectors extracted from the same number of video recordings (taken altogether from 54 patients). The underlying model are RBF networks, with a modified training algorithm.
d) Focal Liver Lesions Detection (FLLD): The task is to detect focal liver lesions from computed tomography images [19]. The dataset consists of 147 images, and 115 images are used for training a neural network with 18 hidden units, i.e. we have $m_L = 115$ and $n_L = 7$.

Dataset	m_L^{exp}	n_L	S_{exp}	$\lceil 1.25 \cdot S^R \rceil$	$\lceil 0.07 \cdot S^L \rceil$
TSRD	157	8	42	44	45
HR3D (RBF)	1200	11	120	118	119
DNS	120	7	25	33	33
FLLD	115	7	18	33	33

Table 2: Comparing Results in the Literature with the Estimated Size of Neural Networks

Table 2 provides the comparison of best classification results S_{exp} achieved by the authors in the corresponding multimedia datasets, where m_L^{exp} is the size of the training data used by the authors in their experiments. Table 2 presents also the estimated by equation (2) neural network size by using the same m_L^{exp}.

Equation (2) estimates neural network sizes that are either very close to the size of best classifiers used by authors, or in-between the gate number for the best two results found by the authors. Equation (2) works impressively very well for TSRD and HR3D. The estimations on DNS are close to the number of 25 basic units used in [18], particularly if we take into account that the same classification error is reported in [18] for 30 basic units. On the FLLD dataset, we overestimate the number of basic units, but the test set consists of $\approx 22\%$ of the entire dataset only, i.e. for a larger test set one might expect a larger number of basic units necessary to achieve a better generalization rate [24].

6 Estimating Neural Networks Sizes for Tackling Multimedia Datasets

In this section we use equation (2) to calculate the size of neural networks in a number of well known multimedia datasets from the UCI Machine Learning Repository and proposing to use neural networks of sizes that are shown in Table 3. We denote that in this Table we use $|m_L| = 2|m|/3$. If the number of

Dataset	m_L	n_L	$\lceil 1.25 \cdot S^R \rceil$	$\lceil 0.07 \cdot S^L \rceil$
Arcene	600	10	84	86
Artificial Charact.	4000	12	164	166
Australian Sign Language signs	4433	13	231	231
Australian Sign Language signs HQ	1710	11	117	119
CMU Face Images	427	9	61	62
Covertype	387341	19	1814	1780
Gisete	9000	14	324	323
Image Segmentation	1540	11	117	119
Internet Advert.	2186	12	164	166
Japanese Vowels	427	9	61	62
Optical Recogn. Handwritten Digits	3747	12	164	166
Pen-based Recogn. Handwritten Digits	7328	13	231	231
Vehicle Silhouettes	631	10	84	86

Table 3: Estimated Size of Neural Networks in Multimedia Datasets for High Classification Accuracy

samples m_L used in the training phase of the network should follow a different distribution pattern, then new estimations of neural network sizes using equation (2) should be carried out.

Our study provides a new perspective when choosing neural network sizes for tackling hard classification problems. So far unrealistic bounds of network sizes exist in the literature for implementing neural networks. Rule of thumbs are used in [17] whereas the size complexity issue has a long discussion on related forums on the web due to the lack of guidelines in estimating realistic neural network sizes. So far in the literature the size is estimated according to the attributes $X_1, X_2, ..., X_n$ of the samples and/or the number of weights W resulting to upper bounds that lead to unrealistic to implement neural neworks. A distinguished feature of our approach is that the focus on estimating neural networks turns to the number of available training data and the required number of bits to enumerate them. Thus, to estimate the size of a neural network one only needs to know the number of samples that will train the network.

Figure 1 and Figure 2 illustrate the differences between the existing studies and our approach when tackling problems with 300 and 3000 sample data having 120 input features. Existing studies use the 120 features to calculate unrealistic size of neural networks, which sizes are the same for both 300 and 3000 datasets. Our study implies that *the learning capacity of the neural network is related to the number of available training data*, thus it will be different for the 300 and 3000 datasets. It seems feasible for the learning capacity of a neural network that the more data (samples) a dataset has, the larger neural network is

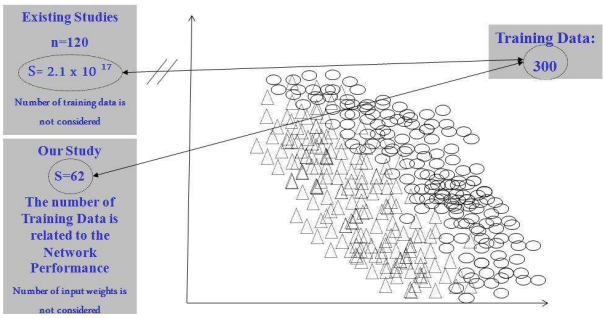

Figure 3: Learning capacity of Neural Networks with $m_L = 300$ samples and $n = 120$ features.

required, whereas existing studies do not consider the number of available data for determining the neural network size.

Our appoach works well with any data representation. The binary encoding of attributes (input vectors) does not affect the complexity bounds of Equation (2) since the bounds depend only on the enumeration of samples.

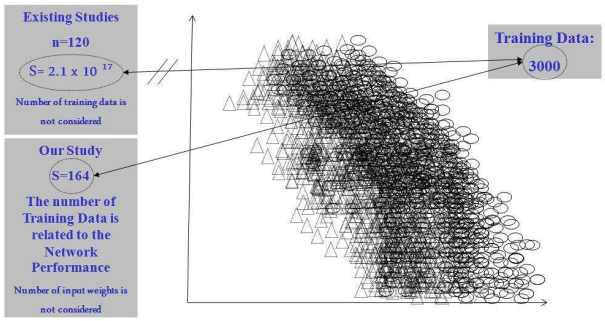

Figure 4: Learning capacity of Neural Networks with $m_L = 3000$ samples and $n = 120$ features.

Another important feature of our approach derived by the *a posteriori* evaluation of the complexity estimation. The evaluation was carried out for classifier systems consisting of different types of basic units, therefore our approach does not depend on any neural network architecture or training method used. Thus it can be used with any training method or any neural network architecture.

A third feature of our approach is related to the depth vs size problem, which is one of the hardest problems in theoretical computing, with very little process over the past decades. The debate about the relevance of Kolmogorov's theorem [29] to neural networks has been used to justify the focus on universal depth-three classifiers. Judd's work [25] gives also some preference to shallow depths, mainly due to simplicity in topology as moving to larger depths, a num-

ber of questions arise of how to construct higher layers, what data should be used, what learning algorithm should be used, and what will be the size of layers, which all increases the overall complexity of the depth vs size problem. In our approach we extended in depth the LSA machine and investigated the performance achieved by two types of LSA machines, a depth-two LSA machine and a depth-four LSA machine. To do so, a new learning approach was introduced for recursively creating and learning the training sets for further depths [30]. This novel approach gives future directions for how to treat higher levels of depth. Our approach showed that we approximate the same high classification accuracy by using the same number of training nodes for sizes of $8 \cdot \sqrt{2_L^n / n_L}$ threshold gates in depth-2 and depth-4 networks. Therefore what matters is the number of neurons used instead of what topology will be used, which may be considered as experimental evidence on preference to swallow depth neural networks.

Equation (2) may then assist researchers that are using the datasets of Table 3 to choose with confidence the estimated neural network size for their experiments and concentrate on other experimental issues.

7 Conclusion

Most of the work in the literature has related the size of neural networks with the number of attributes of the dataset and/or weight inputs, i.e. with the descriptive representation of the dataset. The distinguished feature of our approach is that the size of neural networks depends only on the number of available data for training the network and not on the number of weights. This means that the learning capacity of a neural network for achieving high classification accuracy is highly related to the number of available training data. This is a problem depended parameter and therefore there is no-free-lunch for using neural networks of constant sizes that perform well in any hard classification problem. Estimation of neural network sizes should be performed according to problem parameters of each dataset as the number of available training data used in our approach. Estimating the size of the neural network by using Equation (2) allows researchers to concentrate on other parameters to fine tune and optimize the performance of their classifier (learning algorithm parameters or type and structure of the network, data representation, other problem depended parameters). Equation (2) then provides us with an important tool for designing neural network sizes for accurate classification performance in real world and multimedia datasets.

References:

[1] A. Albrecht, On the Complexity to Approach Optimum Solutions by Inhomogeneous Markov Chains, in *Proc. Genetic and Evolutionary Computation Conference (GECCO'04)*, Springer-Verlag, LNCS series, 3102(2004):642-653.

[2] A.A. Albrecht, A.V. Chashkin, C.S. Iliopoulos, O.M. Kasim-Zade, G. Lappas, K. Steinhofel, A Priori Estimation of Classification Circuit Complexity, *Texts in Algorithmics*, 8:97-114, 2007.

[3] A.A. Albrecht, A.V. Chashkin, C.S. Iliopoulos, O.M. Kasim-Zade, G. Lappas, K. Steinhofel, A Note on a priori Estimations of Classification Circuit Complexity, *Fundamenta Informatique*, in press

[4] A. Albrecht, G. Lappas, Staal A. Vinterbo, C.K. Wong and Lucila Ohno-Machado, Two applications of the LSA machine, in *Proc. 9^{th} International Conference on Neural Information Processing (ICONIP'02)* (Singapore, 2002) pp. 184–189.

[5] A. Albrecht, G. Lappas (2004), On the Evaluation of Dividing Samples for Training an Extended Depth LSA Machine, *WSEAS Transactions on Systems*, 3(3), 1251-1257, 2004.

[6] A. Albrecht, G. Lappas (2004), Classification Improvement by an Extended Depth LSA Machine, *WSEAS Transactions on Systems*, 3(3), 1120-1125, 2004.

[7] A. Albrecht and C.K. Wong, Combining the perceptron algorithm with logarithmic simulated annealing, *Neural Processing Letters* **14**(2001) 75–83.

[8] M. Anthony, *Boolean Functions and Artificial Neural Networks*, CDAM Research Report LSE-CDAM-2003-01, Department of Mathematics and Centre for Discrete and Applicable Mathematics, The London School of Economics and Political Science, London, UK, 2003.

[9] M. Anthony, P.L. Bartlett. *Neural Network Learning*, Cambridge University Press, UK, 1999.

[10] P.L. Bartlett. 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(2)(1998):525-536.

[11] P.L. Bartlett, V. Maiorov, R. Meir. Almost Linear VC-dimension Bounds for Piecewise Polynomial Networks, *Neural Computation*, 10(1998):2159-2173.

[12] E.B. Baum. On the Capabilities of Multilayer Perceptrons, *Journal of Complexity*, 4(1988):193-215.

[13] E.B. Baum, D. Haussler. What Size Net Gives Valid Generalization?, *Neural Computation*, 1(1)(1989):151-160.

[14] E.B. Baum, The perceptron algorithm is fast for nonmalicious distributions, *Neural Computation* **2**(1990) 248–260.

[15] V. Beiu. Digital Integrated Circuit Implementations, in: E. Fiesler, R. Beale,(eds) *Handbook on Neural Computation*, Oxford University Press, 1997.

[16] A. Blum, R.L. Rivest. Training a 3-Node Neural Network is NP-Complete, *Neural Networks*, 5(1992):117-127.

[17] R.O. Duda, P.E. Hart, D.G. Stork. *Pattern Classification*, Wiley-Interscience, New York, 2001.

[18] R. Gil-Pita, P. Jarabo-Amores, R. Vicen-Bueno, M. Rosa-Zurera, Neural solutions for high range resolution radar classification. In *Proc 7th Int Work-Conf Artificial and Natural Neural Networks*, LNCS vol 2687 (2003), pp 1049-1056.

[19] M. Gletsos, SG. Mougiakakou, GK. Matsopoulos, KS. Nikita, AS. Nikita, D. Kelekis A computer-aided diagnostic system to characterize CT focal liver lesions: design and optimization of a neural network classifier, *IEEE Trans Inform Techn Biomed* 7:153-162, 2003.

[20] P.W. Goldberg, M.R. Jerrum. Bounding the Vapnik-Chervonenkis Dimension of Concept Classes Parametrised by Real Numbers, *Machine Learning*, 18(2/3)(1995):131-148.

[21] B. Hajek, Cooling schedules for optimal annealing, *Mathem. Operat. Res*. **13**(1988) 311–329.

[22] B. Hayes-Roth, F. Hayes-Roth, Concept Learning and the Recognition and Classification of Exemplars. *Journal of Verbal Learning and Verbal Behavior*, 16(1977):321-338.

[23] K.-U. Höffgen, H.-U. Simon. Robust Trainability of Single Neurons, *Proc.* 5^{th} *Annual ACM Workshop on Computational Learning Theory*, 1992, pp 428-439.

[24] S. Ingrassia, I. Morlini, *Neural network modeling for small datasets*, Technometrics 47(2005):297-311.

[25] J.S. Judd. On the Complexity of Loading Shallow 1-dimension Networks, *Journal of Complexity*, 4(1988):177-192.

[26] N. Karayiannis, Y. Xiong (2006) Training reformulated radial basis function neural networks capable of identifying uncertainty in data classification. *IEEE Transactions Neural Networks* 17:1222-1234.

[27] M. Karpinski, A.J. Macintyre. Polynomial Bounds for VC Dimension of Sigmoidal and General Pfaffian Neural Networks, *Journal of Computer and System Sciences*, 54(1)(1997):169-176.

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

[29] A.N. Kolmogorov. On the Representation of Continuous Functions of Several Variables by Superposition of Continuous Functions of one Variable and Addition, *Doklady Akademiia Nauk*, 114(5)(1957):953-956.

[30] G. Lappas, R.J. Frank and A.A. Albrecht, A computational study on circuit size vs. circuit depth, *International Journal of Artificial Intelligence Tools*, 15:143–162, 2006.

[31] G. Lappas, Combinatorial Optimization Algorithms Applied to Pattern Classification, *PhD Thesis*, University of Hertfordshire, UK.

[32] G. Lappas, Estimating the Size of Neural Networks from the Number of Available Training Data, in *Proceedings of the 17th International Conference on Artificial Neural Networks (ICANN07), Porto, Portugal, Lecture Notes on Computer Science*, Vol. 4668, pp. 68–77.

[33] Yu-Cheng Lin, Ming-Hua Lin, Hsin-Hsiung Huang, Lu-Yao Lee. A Simulated Annealing Approach for Social Utility Maximization in Dynamic Spectrum Management, in *Proceedings of the 8th WSEAS International Conference on Instrumentation, Measurement, Circuits and Systems*, Hangzhou, China, 20-22 May, 2009, pp. 244–247.

[34] O.B. Lupanov, On the design of circuits by threshold elements (in Russian), *Problemy Kibernetiki* 26:109–140, 1973.

[35] W. Maass, G. Schnitger, and E. Sontag, On the computational power of sigmoid versus boolean threshold circuits, in: *Proc. 32*$^{\rm nd}$ *Annual IEEE Symposium on Foundations of Computer Science*, pp. 767–776, 1991.

[36] W. Maass. On the complexity of learning on neural nets, in *Proc. Computational Learning Theory: EuroColt'93*, Oxford University Press, 1994, pp 1-17.

[37] W. Maass, Bounds on the computational power and learning complexity of analog neural nets, in: *Proc. 25*$^{\rm th}$ *Annual ACM Symp. on the Theory of Computing*, pp. 335–344, 1993.

[38] W. Maass, Bounds for the computational power and learning complexity of analog neural nets, *SIAM J. on Computing* 26:708–732, 1997.

[39] W.S. McCulloch, W.H. Pitts. A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, 5(1943):115-137.

[40] V. Mladenov, G. Tsenov, L. Economou, N. Harkiolakis, P. Karampelas. Neural Network Control of an Inverted Pendulum on a Cart, in *Proceedings of the 9th WSEAS International Conference on Robotics, Control and Manufacturing Technology*, Hangzhou, China, 20-22 May, 2009, pp. 112–120.

[41] S. Rampone, Recognition of Splice Junctions on DNA Sequences by BRAIN Learning algorithm, *Bioinformatics*, 14(1998):676-684.

[42] C.A. Pena-Reyes and M. Sipper, Fuzzy CoCo: A cooperative coevolutionary approach to fuzzy modeling, *IEEE Trans. on Fuzzy Systems* **9**(2001) 727–737.

[43] F. Rosenblatt, *Principles of Neurodynamics* (Spartan Books, New York, 1962).

[44] R. Setiono, Generating concise and accurate classification rules for breast cancer diagnosis. *Artificial Intelligence Medicine* **18**(2000) 205–217.

[45] B. Sion, F. Daniela, C.M. Titrade, C. Mihalcescu. Internet and eBusiness, in *Proceedings of the 8th WSEAS International Conference on Applied Computer and Applied Computational Science*, Hangzhou, China, 20-22 May, 2009, pp. 537–540.

[46] E.D. Sontag. VC-dimension of Neural Networks, in C.M. Bishop (ed) *Neural Networks and Machine Learning*, Springer Verlag, Berlin, 1998, pp 69-95.

[47] Hyontai Sug. An Experimental Decision of Samples for RBF Neural Networks, in *Proceedings of the 9th WSEAS International Conference on Multimedia Systems & Signal Processing*, Hangzhou, China, 20-22 May, 2009, pp. 249–252.

[48] V. Vapnik, A.Y. Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to their Probabilities. *Theory of Probability and Its Applications*, 16(2)(1971):264-280.

[49] R. Vicen-Bueno, R. Gil-Pita, M. Rosa-Zurera, M. Utrilla-Manso, F. Lopez-Ferreras. Multilayer perceptrons applied to traffic sign recognition tasks. In *Proc 8th Int Workshop on Artificial Neural Networks*, LNCS vol 3512, pp 865-872, 2005.

[50] R.S. Wenocur, R.M. Dudley. Some Special Vapnik-chervonenkis Classes, *Discrete Mathematics*, 33(1981):313-318.

[51] D.H. Wolpert and W.G. Macready, No free lunch theorems for optimization, *IEEE Trans. on Evolutionary Computation* **1**(1997) 67–82.

[52] W.H. Wolberg and O.L. Mangasarian, Multisurface method of pattern separation for medical diagnosis applied to breast cytology, *Proc. of the National Academy of Sciences, USA* **87**(1990) 9193–9196.

[53] M. Yannakakis. Computational Complexity, in E.H.L. Aarts, J.K. Lenstra (eds), *Local Search in Combinatorial Optimization*, Wiley & Sons, Chichester, 1998.