ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

A fast learning algorithm for high dimensional problems: an application to microarrays

Oscar Fontenla-Romero, Bertha Guijarro-Berdiñas, Beatriz Pérez-Sánchez
David Martínez-Rego, Diego Rego-Fernández

Department of Computer Science, University of A Coruña
Faculty of Informatics. Campus de Elviña s/n. - A Coruña. Spain.
{ofontenla, diego.rego, cibertha, dmartinez, bperezs}@udc.es

Abstract. In this work, a new learning method for one-layer neural network based on a singular value decomposition is presented. The optimal parameters of the model can be obtained by means of a system of linear equations whose complexity depends on the number of samples. This approach provides a fast learning algorithm for huge dimensional problems where the number of inputs is higher than the number of data points. These kinds of situations appear, for example, in DNA microarrays scenarios. An experimental study over eleven microarray datasets shows that the proposed method is able to outperform other representative classifiers, in terms of CPU time, without significant loss of accuracy.

1 Introduction

The medical field has traditionally been an area for successful applications of Machine Learning (ML). With the fast and high increase of big databases some new problems have emerged in ML, one of them related with the scalability of algorithms. The lack of scalability is the reason that many of the classical and outstanding algorithms are unable to work properly, due either to their poor performance or because their computational needs are unapproachable. Most of the efforts of the Big Data community working on scalability have been centered on improving the scalability with regards to the number of training samples. However, there exists some situations in which the limiting condition is related to the number of input variables or features. This is the case of some medical data and, specifically, the case of deoxyribonucleic acid (DNA) microarrays. One of the most important applications of DNA microarrays is the classification and prediction of cancer. This type of data encloses a few number of samples while at the same time presents a large dimensionality of the feature space, commonly a number of samples below 100 versus many thousands of genes. These inherent characteristics seriously narrow the application of classical machine learning algorithms leading to diminished performance and increasing their computational complexity. Consequently, handling of microarray data emerges as a challenge for the research community.

In a previous research [1] we introduced a supervised learning method for single-layer feedforward neural networks. The novelty arises in the inclusion of a convex objective function, equivalent to the MSE, that avoids local minimal and obtains a global solution by means of a system of linear equations. The method

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

exhibits a fast operation with a quadratic complexity with respect to the number of input variables. This fact implies that in those contexts where the number of features is extremely large, e.g. microarray, the complexity of the method significantly increases. To overcome this problem and facilitate the handling of this kind of data, in this paper we present a modification of the method, without loss of its favourable characteristics, with a quadratic complexity with respect to the number of samples instead of input variables.

2 Background

As mentioned, the method presented in this paper is funded on a previous learning algorithm [1] that will be briefly described in this section for the sake of comprehension. Specifically, it is a supervised algorithm that obtains the optimal weights of a one-layer feedforward neural network (i.e., no hidden layers) with non linear output functions. In contrast to some other well-known algorithms it *backpropagates* the networks's *desired output* signal instead of the *error* between the desired and the real outputs. In Figure 1 this process is depicted graphically for only one output neuron in order to avoid a cumbersome derivation although solving the problem for J output neurons is straight by applying the same process for every neuron.

Let S be the number of training data samples, I the number of input variables (including the bias), $\mathbf{X} \in \mathbb{R}^{I \times S}$ be the input of the neural network; $\mathbf{d}$, $\mathbf{y} \in \mathbb{R}^S$ be the desired and real outputs; $\mathbf{w} \in \mathbb{R}^I$ be the weight vector connecting the I input neurons with the output neuron; and f ; f^{-1}; $f' : \mathbb{R} \to \mathbb{R}$ be the non linear function, its inverse and its derivative. If the desired output is backpropagated, i.e. $\bar{\mathbf{d}} = f^{-1}(\mathbf{d})$, then the optimal set of weights can be obtained by minimizing the MSE between $\mathbf{z} = \mathbf{X}^T\mathbf{w}$ and $\bar{\mathbf{d}}$ which leads to following system of linear equations:

$$\mathbf{A}\mathbf{w} = \mathbf{b} \tag{1}$$

where $\mathbf{A}$ and $\mathbf{b}$ are defined as [1]:

$$\mathbf{A} = \mathbf{X}\mathbf{F}\mathbf{F}\mathbf{X}^T \quad ; \quad \mathbf{b} = \mathbf{X}\mathbf{F}\mathbf{F}\bar{\mathbf{d}} \tag{2}$$

$\mathbf{F} = diag(f'(\bar{d_1}), f'(\bar{d_2}), \ldots, f'(\bar{d_S}))$ being a diagonal matrix.

This algorithm is very computationally efficient in the most frequent cases in ML problems where $S >> I$, as the size of the system of linear equations in (1) depends on I. However, in situations where $I >> S$, like DNA microarrays, the computational performance degrades as the size of the system is very large. In this last case, the calculation of matrix A in equation (2) and its inverse, necessary to obtain the optimal weights ($\mathbf{w}$) from equation (1), is highly computational demanding. In this work, we propose a transformation of the algorithm to be efficient also in this second scenario.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

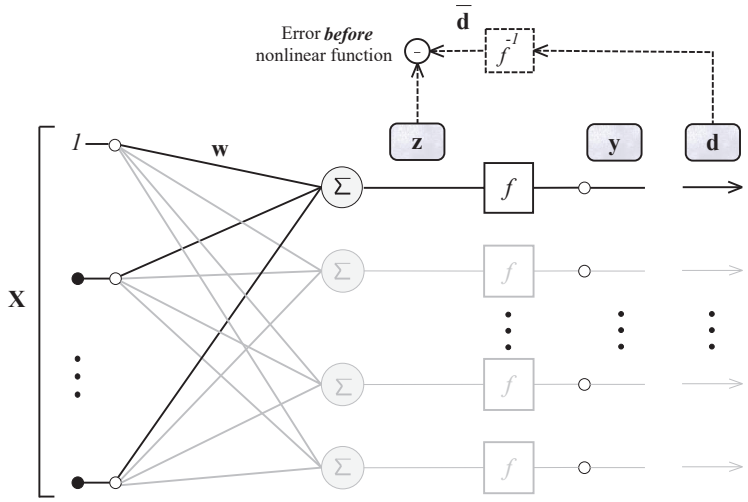

Figure 1: Architecture of a single-layer feedforward neural network.

3 Optimizing the learning algorithm by Singular Value Decomposition

Taking equations (1) and (2) as a starting point, a new equation system arises as follows:

$$\mathbf{XFFX}^T\mathbf{w} = \mathbf{XFF}\bar{\mathbf{d}}$$

By replacing $\mathbf{H} = \mathbf{XF}$, this system can be rewritten as:

$$\mathbf{HH}^T\mathbf{w} = \mathbf{HF}\bar{\mathbf{d}} \tag{3}$$

If a factorization of matrix $\mathbf{H}$ is performed by applying the Singular Value Decomposition (SVD) method then $\mathbf{H} = \mathbf{USV}^T$. In the full (standard) SVD $U \in \mathbb{R}^{I\times I}$, $S \in \mathbb{R}^{I\times S}$ and $V \in \mathbb{R}^{S\times S}$ but in practice the reduced (economy) SVD form can be computed. It contains all the information as the full SVD but is cheaper to compute. In that case, $S \in \mathbb{R}^{P\times P}$ is a square matrix where $P = min(I, S)$. Taking this into account and that $\mathbf{S}$ is a diagonal matrix, then $\mathbf{S} = \mathbf{S}^T$. Replacing $\mathbf{H}$ by its SVD in equation (3) we obtain:

$$\mathbf{USV}^T\mathbf{VSU}^T\mathbf{w} = \mathbf{USV}^T\mathbf{F}\bar{\mathbf{d}}$$

Removing $\mathbf{U}$ from both sides of the equation obtains:

$$\mathbf{SV}^T\mathbf{VSU}^T\mathbf{w} = \mathbf{SV}^T\mathbf{F}\bar{\mathbf{d}}$$

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

Considering $\mathbf{Z} = \mathbf{VS}$ and consequently $\mathbf{Z}^T = \mathbf{SV}^T$ the equation is transformed into:

$$\mathbf{Z}^T\mathbf{Z}\mathbf{U}^T\mathbf{w} = \mathbf{Z}^T\mathbf{F}\bar{\mathbf{d}}$$

that can be further transformed into:

$$\mathbf{U}^T\mathbf{w} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{F}\bar{\mathbf{d}}$$

Finally, considering the properties of the SVD, it holds that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and $(\mathbf{U}^T)^{-1} = (\mathbf{U}^{-1})^T = (\mathbf{U}^T)^T = \mathbf{U}$. Using these results, the optimal weights can be obtained using the following equation:

$$\mathbf{w} = \mathbf{U}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{F}\bar{\mathbf{d}}$$

As can be observed, with this new approach it is necessary to compute the inverse of $\mathbf{Z}^T\mathbf{Z}$, a $S \times S$ matrix, so a computational advantage is obtained for problems where $I >> S$.

4 Experimental results

The following experiments try to demonstrate how the formulation from section 3 behaves, with respect to accuracy and computational time, when dealing with microarray datasets, whose characteristics are a high number of features and small sample sizes. Eleven public datasets available at [2][3][4][5] and summarized in Table 1 are used to test the method. All datasets represent binary problems. The proposed method was compared against the original One Layer method (*OL*) explained in Section 2 together with other three representative classifiers: the classical Support Vector Machine (*SVM*) [6], a SVM using a Radial Basis Function as kernel function (*SVM (rbf)*), and the well-known Fisher linear discriminant (*Fisher*) [7]. Simulations were carried out using a Intel Xeon W3550 processor with 3.07GHz clock speed and 11,7Gb RAM.

Dataset	Instances	Features
Brain	21	12,625
CNS	60	7,129
Colon	62	2,000
Prostate	102	12,600
Breast	78	24,481
Lung	181	12,533
DLBCL	77	5,469
Leukemia	38	7,129
Ovarian	253	15,154
Gli85	85	22,283
Smk	187	19,993

Table 1: Description of datasets

Table 2 shows the mean Area Under the Curve (AUC) and standard deviation obtained by each method. In order to get significant results 10-fold cross-validation was employed and every experiment was repeated 30 times. As can be

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

seen, the proposed method (*OL-SVD*) and the original (*OL*) obtain exactly the same values, since these methods are mathematically equivalent, according to section 3. However, there are three cases (marked with an asterisk) in which the *OL* method is unable to obtain a solution as it exceeds the RAM memory when trying to compute the inverse of the $I \times I$ features matrix. Furthermore it can be concluded that, in terms of AUC, the proposed method performs at least as well as other conventional methods such as SVM or Fisher linear discriminant. It is important to note that some other methods, like the *logistic regression* method, were not included because of their very high CPU time demands (around 40 times the *OL-SVD* needs in the case of *logistic regression*).

	OL	OL-SVD	SVM	SVM (rbf)	Fisher
Brain	**0.54** ± 0.07	**0.54** ± 0.07	0.50 ± 0.00	0.50 ± 0.00	0.46 ± 0.02
CNS	**0.64** ± 0.03	**0.64** ± 0.03	0.50 ± 0.07	0.50 ± 0.03	0.50 ± 0.04
Colon	**0.83** ± 0.02	**0.83** ± 0.02	0.62 ± 0.07	0.78 ± 0.02	0.75 ± 0.02
Prostate	**0.90** ± 0.01	**0.90** ± 0.01	0.81 ± 0.05	0.78 ± 0.02	0.79 ± 0.02
Breast	*	**0.69** ± 0.02	0.65 ± 0.02	0.54 ± 0.03	0.66 ± 0.03
Lung	**1.00** ± 0.00	**1.00** ± 0.00	0.88 ± 0.03	0.99 ± 0.01	0.99 ± 0.01
DLBCL	**0.99** ± 0.01	**0.99** ± 0.01	0.79 ± 0.05	0.89 ± 0.01	0.93 ± 0.02
Leukemia	**1.00** ± 0.00	**1.00** ± 0.00	0.87 ± 0.05	0.82 ± 0.04	0.74 ± 0.02
Ovarian	**1.00** ± 0.00	**1.00** ± 0.00	**1.00** ± 0.00	0.98 ± 0.01	**1.00** ± 0.00
Gli85	*	**0.94** ± 0.01	0.50 ± 0.01	0.79 ± 0.02	0.82 ± 0.02
Smk	*	**0.80** ± 0.01	0.75 ± 0.02	0.72 ± 0.01	0.70 ± 0.02

Table 2: Mean AUC and standard deviation for the test set (%). Best results are boldfaced

A second comparison was made in terms of computational time. In this case, the CPU time is counted 10 times for each method training with *all* data in each dataset. These results are shown in Table 3. As can be observed, the new proposed method considerably improves OL method and also overcomes the other proved methods.

	OL	OL-SVD	SVM	SVM (rbf)	Fisher
Brain	1058.52 ± 10.86	**0.01 ± 0.01**	0.06 ± 0.11	0.05 ± 0.05	0.46 ± 0.15
CNS	183.70 ± 0.50	**0.04 ± 0.04**	2.04 ± 0.14	0.05 ± 0.00	0.24 ± 0.04
Colon	3.74 ± 0.18	**0.02 ± 0.02**	2.12 ± 0.15	0.03 ± 0.00	0.07 ± 0.00
Prostate	1230.39 ± 79.09	**0.17 ± 0.03**	3.08 ± 2.53	0.29 ± 0.03	0.74 ± 0.05
Breast	*	0.37 ± 0.04	**0.23 ± 0.03**	0.36 ± 0.02	2.19 ± 0.48
Lung	1338.19 ± 8.64	0.25 ± 0.03	5.33 ± 0.28	**0.23 ± 0.02**	0.88 ± 0.03
DLBCL	106.10 ± 1.84	**0.03 ± 0.01**	2.31 ± 0.14	0.05 ± 0.00	0.19 ± 0.02
Leukemia	232.58 ± 2.41	**0.01 ± 0.00**	1.00 ± 0.06	0.04 ± 0.00	0.21 ± 0.01
Ovarian	2086.65 ± 23.44	0.64 ± 0.05	**0.20 ± 0.02**	0.71 ± 0.07	1.62 ± 0.06
Gli85	*	**0.22 ± 0.02**	2.41 ± 0.12	0.23 ± 0.01	1.56 ± 0.07
Smk	*	**0.38 ± 0.02**	0.50 ± 0.02	0.83 ± 0.02	1.74 ± 0.06

Table 3: Mean CPU time (s) and standard deviation. Best results are boldfaced

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

5 Conclusions

OL is successful over datasets with more samples than features, but, when the number of features increases its computational performance falls. In this scenario, methods like *SVM* or *Fisher* are recommended, but with these algorithms the opposite occurs: neither *SVM* nor *Fisher* have a good computational performance when the number of samples is very high. In this paper, a modification based on *OL* called *OL-SVD* was introduced to avoid problems with higher number of features than samples, like microarray data sets. While *OL-SVD* does not represent a huge improvement over other methods like *SVM*, the advantage of *OL-SVD* is that, with a few modifications, it could automatically adapt to efficiently deal with both scenarios, more features than samples and vice versa. This new mechanism, that will be included in upcoming work, will suppose an improvement over other methods included in this work. Meanwhile, *OL* and *OL-SVD* are the perfect blend to deal with data sets with high variability of features and samples. Moreover it could be used as a constructive block for more complex paradigms like a layered learning algorithm, such as deep learning, or in a feature selection scenario as a fast classifier for a wrapper method. Finally, an additional line of research is the extension of the method by using some kind of regularization mechanism.

6 Acknowledgements

This work has been supported by the Secretaría de Estado de Investigación of the Spanish Government (Grant TIN2015-65069-C2-1-R), by the Xunta de Galicia (Grant GRC2014/035), and by the European Union FEDER funds.
Martínez-Rego acknowledges support of Xunta de Galicia under PostDoctoral Grant Code POS-A/2013/196.

References

[1] O. Fontenla-Romero, B. Guijarro-Berdiñas, B. Pérez-Sánchez, and A. Alonso-Betanzos. A new convex objective function for the supervised learning of single-layer neural networks. *Pattern Recognition*, 43(5):1984–1992, 2010.

[2] Broad Institute. Cancer Program Datasets. http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi, [Online; accessed November 2015].

[3] Kent Ridge Bio-medical Dataset. http://datam.i2r.a-star.edu.sg/datasets/krbd/, [Online; accessed November 2015].

[4] Feature Selection Datasets at Arizona State University. http://featureselection.asu.edu/datasets.php, [Online; accessed November 2015].

[5] A. Statnikov, C. F. Aliferis, and I. Tsamardinos. Gems: Gene Expression Model Selector. http://www.gems-system.org/, [Online; accessed November 2015].

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[7] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.