ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 135-140

Influence of Weight-Decay Training in Input Selection Methods.

† Mercedes Fernández-Redondo - † Carlos Hernández-Espinosa.

† Universidad Jaume I, Campus de Riu Sec, Edificio TI, Departamento de Informática, 12080 Castellón, Spain. e-mail: redondo@inf.uji.es, espinosa@inf.uji.es

Abstract. We describe the results of a research on the effect of weight-decay (WD) in input selection methods based on the analysis of a trained multilayer feedforward network. It was proposed by some authors to train the network with WD before applying this type of methods. The influence of WD in sixteen different input selection methods is empirically analyzed with a total of seven classification problems. We show that the performance variation of the input selection methods by introducing WD depends on the particular method. But for some of them, the use of WD can deteriorate their efficiency. Furthermore, it seems that WD improves the efficiency of the worst methods and deteriorates the performance of the best ones. In that sense, it diminishes the differences among different methods. We think that the use of weight-decay with this type of input selection methods should be avoided because the results are not good and also the use of weight-decay supposes a complication of the procedure.

1. Introduction.

A supervised neural network is trained with a set of examples of resolution of one problem. So we can easily conclude that the selection of the training information is an important issue. Moreover, in real world problems there is usually a large amount of information that can be measured. But, not all the variables are equally useful, they may be noisy, irrelevant, or redundant. Their importance is unknown a priori.
The problem of input selection consists in selecting a smaller subset of features from a larger one of what we can call candidate features or inputs.
In the case of neural networks, by reducing the set of inputs we can reduce the size of the network, the amount of data to process, the training time and increase the generalization. In the bibliography, we can find several input selection methods for the case of multilayer feedforward networks (MF). One classification among them is:

- *Methods based in the analysis of a trained MF neural network [1-12].*
- *Methods based on the analysis of the training set, one example is [13].*
- *Methods based on the analysis of other neural network architectures [14].*

All the above methods allow obtaining an ordination of candidate inputs according to its importance. But they do not provide a final subset of inputs because we do not know how many inputs should be kept. Furthermore, the methods are different and the ordinations are usually different. Their performance varies.
In this paper, we will focus on the above first group of methods. In order to apply them, we should train at least one MF network with the full set of candidate inputs. After, we should analyze the trained network to determine an ordination of inputs.
These methods are based on the principle that during training the network should discover the important inputs and it should be reflected in the final network structure.

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 135-140

Some authors [3] have proposed to apply the input selection methods to a network trained with weight-decay (WD). At first sight, it seems reasonable because it is well known that the effect of WD is to prune irrelevant inputs. The objective of this research was to analyze the effect of WD training in input selection methods.
In the next section we present a brief description of some input selection methods and also a methodology to analyze their performance.

2. Theory and Methodology.

2.1 Input selection methods.

The methods based on the analysis of a trained MF network normally define the relevance of one input i, S_i. One input is considered more important if its relevance is larger. One way to calculate the relevance S_i is to define the relevance s_{ij} for every weight w_{ij} connecting the input unit i and hidden unit j. After that, the input relevance can be calculated by summing for all j.
In the following, we will group the methods according to their basic idea, a more complete description of them can be found in our previous paper [15].

First group.
Some criteria for defining weight relevance are based on direct weight magnitude. We have used two methods in this group, one was proposed by Belue [1] (named BL2 in the rest of the paper) and the other by Tetko [2] (named TEA).

Second group.
Other criteria of relevance are based on an estimation of the increment in the mean square error when pruning the weight: Cibas [3] (CIB) and Tetko [2] (TEE).

Third group.
Other methods define the relevance by using the variance of weights w_{ji} connected to an input i. A smaller variance means a lower relevance (the input behaves like a threshold). The methods are in [4] (named DEV) and in [5] (named DR3).

Fourth group.
Some other criterions evaluate the contribution of one input to the outputs. They try to take into account the values of the weights in the whole network structure. For example [2] (named TEB) and [6] (named TEK).

Fifth group.
There are several methods, which use the sensitivity of the outputs with respect to one input for defining its relevance. In this case, a higher sensitivity is assumed as a higher relevance. The methods were proposed by Belue [1] (named BL1), Cloete [7] (named CL), Priddy [8] (named PRI), Sano [9] (named SAN) and Deredy [5] (DR2)

Sixth group.
Others estimate the performance decrease when setting the input to a fix value. The input relevance is considered larger as the performance decrement increases. They were proposed by Utans [12] (UTA), Mao [10] (MAO) and LEE [11] (LEE).

2.2 Methodology.

In the following paragraphs we will explain several concepts which are useful to understand the experimental results. An additional explanation can be found in [15].
As pointed out before, the input selection methods provide an ordination of inputs

according to its presumed importance, but they do not provide the number of inputs that should be retained in the final subset.
Let's suppose, as an example, that we get the following ordination: {1,6,5,2,4,3,7}.
After that, we can obtain several subsets of inputs by successively removing the least important input. For example, the first subset is obtained by removing input number 1, {2,3,4,5,6,7}, the next subset is {2,3,4,5,7} and the last subset of one input is {7}.
For every subset, we can obtain the mean performance of a classifier to see how good the subset is. We apply the classifier to the problem using the inputs in the subset.
In our case, the classifier was MF, however its performance depends on the particular weight initialization, so it is necessary to train several networks to obtain a mean performance and an error in this mean. In our experiments, the minimum number of trained networks for one subset was 10 and we use the percentage correct in the test.
After obtaining the performance of all subsets, we can find an optimal subset for each method. The optimal subset should provide the best performance (percentage). However, in the cases of two subsets with indistinguishable percentage, the optimal subset is the one with lower number of inputs because the final network model is simpler. It is useful to perform t-tests to see if two percentages are distinguishable,.
We can consider that the performance of one method consists of the percentage and the number of inputs of its optimal subset.

3. Experimental results.

The main purpose of this research was to evaluate the effect of WD in the described input selection methods.
We have applied the methods to 7 different classification problems. They are from the UCI repository of machine learning databases: Liver Disorders (named BUPA in this paper), Credit Approval (CREDIT), Heart Disease (HEART), Pima Indians Diabetes (PIMA), Voting Records (VOTING) and Wisconsin Breast Cancer (WDBC). The complete data and a full description can be found in the repository.
In all the problems we have included a first useless input generated at random inside the interval [0,1]. We have also normalized the range of variability of every input into the interval [0,1]. This is very important because the range influences the final value of the input weights and therefore all the relevance measurements.
We need at least one trained network for each problem to apply the methods. Also for the objective of this research (evaluate the effect of WD) we should train one network with backpropagation and other with WD. We finally decided to train 10 networks for each training algorithm with different initializations. We have applied the methods to the 10 networks, obtained 10 different relevance measurements for each input and method and calculated a final value, which was the mean of the 10 measurements.
The reason for this procedure is that the relevance values depend on the final trained network and it can be biased by the initial weights if we only use one network.
The number of hidden units in the network and the weight-decay factor λ were obtained by a trial and error procedure for every problem.
Following the methodology described in the above section we have obtained the performance of every method (with and without WD), the results are in table 1.
In each row, we have the results of a method followed by the same method with the use of WD. For example, method BL1 and method BL1 with weight-decay are in the

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 135-140

first and second rows. In the columns we have the seven different databases, the columns with header "%" contain the performance of the optimal subsets (percentage) and the columns with the title "N" the number of inputs in the optimal subset.
Furthermore, with the shadows we tried to indicate the results of the comparison between one method with and without WD, the best results have a shadow in the cell. In the cases where the performances are indistinguishable (the percentages are indistinguishable and the number of inputs is the same) there is no shadow.

Table 1. Performance differences between using or not WD in input selection methods.

	BANDS		**BUPA**		**CREDIT**		**HEART**		**PIMA**		**VOTING**		**WDBC**	
Method	**%**	**N**	**%**	**N**	**%**	**N**	**%**	**N**	**%**	**N**	**%**	**N**	**%**	**N**
BL1	66.5±1.2	18	69.5±0.2	5	88±0	1	79.1±0.9	8	77.0±0.2	5	94.9±0.1	9	99.6±0.2	12
& WD	68.6±1.0	12	65±2	5	88±0	1	79.1±0.9	8	77.0±0.2	5	95.6±0.5	6	99.6±0.2	12
BL2	70.6±0.7	5	67.4±1.6	4	89.4±0.2	8	81.3±0.8	10	77.0±0.3	3	95.9±0.2	12	97.7±1.7	11
& WD	68.6±1.0	12	66±2	6	88±0	1	79.1±0.9	8	77.0±0.2	5	95.6±0.5	6	99.6±0.2	12
CL	66.4±1.3	15	69.5±0.2	5	88±0	1	80±2	6	77.0±0.2	5	94.7±0.4	10	99.6±0.2	12
& WD	68.6±1.0	12	65±2	5	88±0	1	79.1±0.9	8	77.0±0.2	5	95.6±0.5	6	99.6±0.2	12
PRI	66.6±1.2	18	69.5±0.2	5	88±0	1	79.1±0.9	8	77.0±0.2	5	94.9±0.1	9	99.6±0.2	12
& WD	68.6±1.0	12	65±2	5	88±0	1	79.1±0.9	8	77.0±0.2	5	95.6±0.5	6	99.6±0.2	12
DEV	69.4±0.5	6	69.5±0.2	5	88±0	1	81.3±0.8	10	77.0±0.3	3	95.9±0.2	12	99.2±0.1	17
& WD	68.6±1.0	12	67.4±1.6	4	88±0	1	79.1±0.9	8	77.0±0.2	5	95.6±0.5	6	99.6±0.2	12
LEE	70.8±0.9	22	60.1±1.3	6	76.9±0.4	12	76.9±1.8	11	69.7±0.7	7	92.1±0.4	11	98.4±0.3	21
& WD	71.6±0.7	24	69.5±0.2	5	76.7±0.5	3	72±2	12	70.8±0.9	4	92.2±0.6	12	98.2±0.4	25
TEK	67.9±1.1	10	69.1±0.3	6	88±0	1	79.9±0.8	5	76.2±0.2	6	94.9±0.3	16	99.5±0.2	19
& WD	68.2±0.9	15	60.1±1.3	6	88±0	1	80.8±0.3	11	77.0±0.2	5	94±0	3	99.4±0.2	18
SAN	67.4±0.9	21	69.5±0.2	5	88±0	1	78±3	7	76.8±0.2	6	94.9±0.1	9	99.6±0.2	12
& WD	68.6±1.0	12	65±2	5	88±0	1	79.1±0.9	8	77.0±0.2	5	95.6±0.5	6	99.6±0.2	12
TEA	69.4±1.8	4	69.5±0.2	5	89.3±0.2	9	81.3±0.8	10	77.0±0.3	3	95.9±0.2	12	99.1±0.2	9
& WD	72.0±0.9	6	66±2	6	88±0	1	79.1±0.9	8	77.3±0.2	3	95.6±0.5	6	99.6±0.2	12
TEB	69.0±0.6	6	67.4±1.6	4	89.0±0.2	10	79.4±1.8	5	77.0±0.3	3	95±0.3	9	99.6±0.2	12
& WD	68.6±1.0	12	66±2	6	88±0	1	79.1±0.9	8	77.0±0.2	5	95.6±0.5	6	99.6±0.2	12
DR2	67.2±0.5	6	67.4±1.6	4	88.2±0.1	5	82.9±0.5	5	77.0±0.2	5	95.4±0.2	6	98.9±0.1	10
& WD	69.8±1.1	2	66±2	6	88±0	1	77±3	5	77.0±0.2	5	95.6±0.5	6	98.6±0.2	16
DR3	68.8±0.3	3	63.5±1.7	6	88.0±0.2	14	74.8±0.5	5	76.3±0.3	8	92.2±1.5	13	98.4±0.5	27
& WD	74.0±1.4	29	69±2	6	88±0	1	72±3	10	78.5±0.4	8	93.4±0.5	9	99.3±0.1	26
MAO	73.7±0.5	8	69.1±0.3	6	88.0±0.4	14	79.3±1.1	8	76.2±0.2	6	95.5±0.2	6	99.0±0.1	9
& WD	70.7±1.5	4	60.6±0.8	4	88±0	2	76.3±0.6	12	75.5±0.1	3	95.2±0.3	7	99.0±0.1	9
UTA	73.1±1.0	4	69.5±0.2	5	88.3±0.1	6	76±2	5	77.3±0.2	3	96±0	5	98.9±0.2	11
& WD	72.5±0.1	2	65±2	5	88±0	1	76.4±0.5	2	77.3±0.2	3	95.6±0.5	6	99.4±0.2	23
CIB	73.7±0.5	8	69.1±0.3	6	87.7±0.2	12	79.3±1.1	8	76.2±0.2	5	95.5±0.2	7	99.2±0.1	18
& WD	74.6±0.5	6	63.5±1.7	6	87.8±0.4	12	77±3	5	77.0±0.2	5	95.7±0.3	5	99.2±0.1	18
TEE	---	--	63.1±1.2	5	---	--	76.2±0.5	7	70.8±0.9	5	93.7±0.2	16	99.5±0.1	29
& WD	---	--	68.3±0.4	6	---	--	78.7±0.5	11	75.8±0.1	2	94±0	3	98.5±0.2	21

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 135-140

The criterion to affirm that one performance is better is the one explained before. First we consider the percentage, and in the cases of indistinguishable percentage (taking into account the errors) a lower number of inputs means a better performance.
For example, in the results for the database BAND and the methods LEE and LEE with WD the values are "%=70.8±0.9, NI=22" and "%=71.6±0.7, NI=24". The percentages are not distinguishable (taking into account the errors), so we have considered that the performance of LEE (without WD) is better because the number of inputs (22) is lower, its cell is shadowed.
By observing the results in table 1, we can see that there is not a clear improvement by using WD, the results depend on the method and on the database. For instance, for CIB we got an improvement by using WD in 5 of the 7 databases and it worked worse in 1. However, for DR2 the situation is the opposite.
This is a serious drawback for the use of WD with input selection methods: it supposes an additional complexity (to select the weight-decay factor, λ) and the performance can be worse.
By analyzing carefully the results we were able to get an additional insight on what WD provides. In a previous paper [15], we presented a comparison among this input selection methods by using backpropagation as the training algorithm (without WD), we were able to obtain an ordination according to its performance, the result was:

UTA>TEA>BL2>DEV>TEB=DR2=MAO>CL>BL1>PRI=SAN>CIB=TEE>TEK>DR3>LEE

In table 2, we have presented the methods ordered as above, in the first column we have UTA (the best) and in the last LEE. In the rows, we can see the number of databases where the method performed better, equal or worse, by incorporating WD.

Furthermore, we have shadowed the cells where the use of WD supposes a clear modification in the efficiency of the method.

Table 2. Performance comparison between using or not WD, number of databases.

	UTA	TEA	BL2	DEV	TEB	DR2	MAO	CL	BL1	PRI	SAN	CIB	TEE	TEK	DR3	LEE
Better WD	3	3	2	2	1	1	1	2	2	2	3	4	4	4	6	3
Equal	1	1	1	2	1	2	1	3	4	4	2	2	0	0	0	0
Worse WD	3	3	4	3	5	4	5	2	1	1	2	1	1	3	1	4

In the first two methods UTA and TEA, the situation is balanced. In the followings BL2, TEB, DR2 and MAO there is a clear deterioration in the performance because of WD. In the next methods, CL, BL1 and PRI, the influence is not appreciable, the results with most of the databases are equal. Finally, in the worst methods (CIB, TEE, DR3), there is a clear improvement because of WD.
It seems that *the use of weight-decay improves the performance of the worst input selection methods, deteriorates the performance of the best ones and does not affect the rest.* In that sense, its use diminishes the differences among different methods.

4. Conclusions.

We have presented a research on the combination of weight-decay and input selection methods based on the analysis of a trained multilayer feedforward network. The influence of weight-decay in sixteen different input selection methods was empirically analyzed with a total of seven classification problems. We showed that

the use of weight-decay can deteriorate the efficiency of a method. Furthermore, it seems that weight-decay improves the performance of the worst input selection methods and deteriorate the performance of the best ones. In that sense, it diminishes the differences among different methods. We think that weight-decay should not be used for this particular task of input selection.

References.

1. Belue, L.M., Bauer, K.W.: "Determining input features for multilayer perceptrons". Neurocomputing, vol. 7, n. 2, pp. 111-121, 1995.
2. Tetko, I.V., et alt: Neural network studies. 2. Variable selection. Journal of Chemical Information and Computer Sciences, vol. 36, n. 4, pp. 794-803, 1996.
3. Cibas, T., Soulié, F.F., Gallinari, P., Raudys, S.: Variable selection with neural networks. Neurocomputing, vol. 12, pp. 223-248, 1996.
4. Devena, L.: Automatic selection of the most relevant features to recognize objects. Proceedings of the International Conference on Artificial Neural Networks, vol.2, pp.1113-1116, 1994.
5. El-Deredy, W., Branston, N.M.: Identification of relevant features in HMR tumor spectra using neural networks. Proceedings of the 4th International Conference on Artificial Neural Networks, pp. 454-458, 1995.
6. Tetko, I.V., Tanchuk, V.Y., Luik, A.I.: Simple heuristic methods for input parameter estimation in neural networks. Proceedings of the IEEE International Conference on Neural Networks, vol. 1, pp. 376-380, 1994.
7. Engelbrecht, AP., Cloete, I.: A sensitivity analysis algorithm for pruning feedforward neural networks. Proceedings of the International Conference on Neural Networks, vol. 2, pp. 1274-1277, 1996.
8. Priddy, K.L., et alt: Bayesian selection of important features for feedforward neural networks. Neurocomputing, vol. 5, n. 2&3, pp. 91-103, 1993.
9. Sano, H., et alt: A method of analyzing information represented in neural networks. Proceedings of 1993 International Joint Conference on Neural Networks, pp. 2719-2722, 1993.
10. Mao, J., Mohiuddin, K., Jain, A.K.: Parsimonious network design and feature selection through node pruning. Proceedings of the 12th IAPR International Conference on Pattern Recognition, vol. 2, pp. 622-624, 1994.
11. Lee, H., et alt: Selection procedures for redundant inputs in neural networks. Proc. of the World Congress on Neural Networks, vol. 1, pp. 300-303, 1993.
12. Utans, J., Moody, J., et alt: Input variable selection for neural networks: Application to predicting the U.S. business cycle. Proc. of IEEE/IAFE Computational Intelligence for Financial Engineering, pp. 118-122, 1995.
13. Battiti, R.: Using mutual information for selecting features in supervised neural net learning. IEEE Trans. on Neural Networks, vol. 5, n. 4, pp. 537-550, 1994.
14. Watzel, R., Meyer-Bäse, A., et alt: Identification of irrelevant features in phoneme recognition with radial basis classifiers. Proceedings of 1994 International Symposium on Artificial Neural Networks, pp. 507-512, 1994.
15. Fernández, M., Hernández, C.: Input selection by Multilayer Feedforward trained networks. Proceedings of the 1999 IEEE International Joint Conference on Neural Networks, pp. 1834-1839.