Optimal Brain Surgeon Variants for Feature Selection

Mohammed Attik, Laurent Bougrain, Frédéric Alexandre

Optimal Brain Surgeon Variants For Feature Selection

Mohammed Attik
LORIA-INRIA
Campus Scientifique - BP 239
54506 Vandœuvre-lès-Nancy
France
E-mail: Mohammed.Attik@loria.fr

Laurent Bougrain
LORIA-INRIA
Campus Scientifique - BP 239
54506 Vandœuvre-lès-Nancy
France
E-mail: Laurent.Bougrain@loria.fr

Frédéric Alexandre
LORIA-INRIA
Campus Scientifique - BP 239
54506 Vandœuvre-lès-Nancy
France
E-mail: Frederic.Alexandre@loria.fr

***Abstract*— This paper presents three pruning algorithms based on Optimal Brain Surgeon (OBS) and Unit-Optimal Brain Surgeon (Unit-OBS). The first variant performs a backward selection by successively removing single weights from the input variables to the hidden units in a fully connected multilayer perceptron (MLP) for variable selection. The second one removes a subset of non-significant weights in one step. The last one combines the two properties presented above. Simulation results obtained on the Monk's problem illustrate the specificities of each method described in this paper according to the preserved variables and the preserved weights.**

I. Introduction

Pruning techniques for artificial neural networks have been first designed for optimization dealing with the overtraining problem ([1],[2]). This article is focused on a subpart of these methods where a connection is removed according to a relevance criterion often named the weight saliency (also termed sensitivity). More precisely, the weight with the smallest saliency will generate the smallest error variation if it is removed. Similarly, it is possible to obtain the saliency of a unit. Thus, these techniques are also used for variable selection ([3],[4],[5]).

The first motivation of this work is that the weight saliency distributions in the different layers of a MLP are not the same. It can be observed experimentally that the first layer is more stable, which explains that the saliency in the first layer is small as compared to the other ones. Accordingly, it can be interesting to selectively remove weights in the first layer.

The second motivation is to propose a novel way to select the weights in the Generalized Optimal Brain Surgeon method.

- In the next section, we review a mathematical formulation of the saliency analysis model developed by Hassibi for OBS ([2]), and by Stahlberger for Unit-OBS ([5]).
- In Section 3, we present in details our variants based on the mathematical formula presented in section 2.
- In Section 4, we present and comment the results of all variants on the first Monk's problem.
- Finally, in section 5, we conclude on this work.

II. Saliency Calculation

A. Weight saliencies in OBS

This development proposed by Hassibi([2],[6],[7]) considers the change in the training error E if the weight vector $\mathbf{w}$ is perturbed by a small variation $\delta\mathbf{w}$. This change of Error δE is a Taylor expansion of the second-order :

$$\delta E = \left(\frac{\partial E}{\partial \mathbf{w}}\right)^T \delta\mathbf{w} + \frac{1}{2}\delta\mathbf{w}^T.\mathbf{H}.\delta\mathbf{w} + O(||\delta\mathbf{w}||^3) \quad (1)$$

where $\mathbf{H} = \frac{\partial^2 E}{\partial \mathbf{w}^2}$ is the Hessian matrix. For a network trained to a local minimum in error, the first (linear) term vanishes; The error surface is assumed to be quadratic around the minimum, so the third and higher order terms can be ignored.

$$\delta E = \frac{1}{2}\delta\mathbf{w}^T.\mathbf{H}.\delta\mathbf{w} \quad (2)$$

The goal is to set one of the weights to zero which we call w_q to minimize the increase in error given by (Eq. 2).
Eliminating w_q can be expressed as :

$$w_q + \Delta w_q = 0 \text{ or } \quad \mathbf{e}_q^T\delta\mathbf{w} + w_q = 0 \quad (3)$$

where $\mathbf{e}_q$ is the unit vector in weight space corresponding to (scalar) weight w_q.
Solving this extremum problem of minimization with side condition (Eq. 3) using Lagrange method it is possible to update the magnitude of all the weights in the network by :

$$\delta\mathbf{w} = -\frac{w_q}{[\mathbf{H}^{-1}]_{qq}}\mathbf{H}^{-1}.\mathbf{e}_q \quad (4)$$

and the saliency of the weight q is given by :

$$L_q = \frac{1}{2}\frac{w_q^2}{[\mathbf{H}^{-1}]_{qq}} \quad (5)$$

B. Generalization

Stahlberger and Riedmiller ([5]) proposed to the OBS's users, a calculation, called Generalized Optimal Brain Surgeon (G-OBS), to obtain in a single step the update to apply to every weights when deleting a subset of m weights. As for OBS, the increase of the error is given by (Eq. 2), but the condition (Eq. 3) is replaced by the following generalized condition :

$$(\mathbf{w} + \Delta\mathbf{w})^T M = 0 \quad \text{with} \quad M = (\mathbf{e}_{q_1}, \mathbf{e}_{q_2}, \cdots, \mathbf{e}_{q_m}) \quad (6)$$

where M is the selection matrix and $q_1, q_2, \cdots, q_m$ are the indices of the weights that will be removed.
Solving this extremum problem with side condition (Eq. 6) using Lagrange method leads to the solution :

$$\Delta E = \frac{1}{2}\mathbf{w}^T M(M^T H^{-1} M)^{-1} M^T \mathbf{w} \quad (7)$$

$$\Delta \mathbf{w} = -H^{-1} M(M^T H^{-1} M)^{-1} M^T \mathbf{w} \quad (8)$$

The inverse matrix problem can be solved by decomposition techniques such as Singular Value Decomposition (SVD).

III. Algorithms

Stahlberger and Riedmiller ([5]) have defined a variable selection algorithm, called Unit-OBS, which computes, using the calculation G-OBS, which input unit will generate the smallest increase of error if it is removed (Eq. 7).

We propose a new algorithm for variable selection using the weight saliency computation defined in OBS. We call this new algorithm Flexible Optimal Brain Surgeon (F-OBS). Its particularity is to remove connections only between the input layer and the hidden layer (Fig. 1). This algorithm can produce two benefits :

- Reduce the number of weights between the variables and the hidden layer
- Reduce the number of variables

Compared to F-OBS, Unit-OBS is more constrained in the pruning of variables process. Pruning m weights which give a smallest saliency for Unit-OBS do not imply that all the weights are not significant.

Moreover, we propose another algorithm, called Generalized Optimal Brain Surgeon (G-OBS), for which the subset of connections to remove is defined by the smallest saliencies (Fig. 2).

Finally, a third algorithm, called Generalized Flexible Optimal Brain Surgeon (GF-OBS), is a combination of the F-OBS and G-OBS. Thus, this algorithm removes in one stage a subset of connections only between the input layer and the hidden layer (Fig. 3).

We introduce here the computational complexity of different methods. The complexities in time to calculate H^{-1}, ΔE (Eq. 7) and $\Delta \mathbf{w}$ (Eq. 8) are $O(n^2 p)$, $O(m^3)$ and $O(nm + m3)$ respectively.
Unit-OBS and the version of G-OBS proposed by Stahlberger and Riedmiller has a computational complexity of $\binom{n}{m} O(m^3)$ to find m weights giving the minimum of ΔE. Our variants (G-OBS, F-OBS, GF-OBS) neglect this computational complexity because they do not use (Eq. 7), allowing then to be faster for selections.

IV. Experiments

In this section, we evaluate the performance on the Monk's problem of our algorithms for variable selection and optimization, and we compare them to OBS and Unit-OBS.

1) *Train the network to minimum error.*
2) *Save the model (solution).*
3) *Compute* H^{-1}.
4) *Compute the saliencies for all the weights between the input layer and the hidden layer (Eq. 5).*
5) *Find the* w_q *corresponding to the smallest saliency.*
6) *Use the weight* w_q *to update all the weights in the network (Eq. 4).*
7) *Remove weight* w_q.
8) *Repeat steps 2 to 7 until a stopping criterion is reached.*

Fig. 1. Flexible Optimal Brain Surgeon (F-OBS) Algorithm

1) *Train the network to minimum error.*
2) *Save the model (solution).*
3) *Compute* H^{-1}.
4) *Compute the saliencies for all the weights in the network (Eq. 5).*
5) *Select the* m *smallest saliency weights.*
6) *For all selected* m *weights compute* M *using (Eq. 6) and* Δw *(Eq. 8).*
7) *Remove the* m *selected weights and use* Δw *to update all the weights in the network.*
8) *Repeat steps 2 to 7 until a stopping criterion is reached.*

Fig. 2. Generalized Optimal Brain Surgeon (G-OBS) Algorithm

A. First Monk's problem

To compare the algorithms we use the first Monk's problem. This well-known problem (See [8]) requires the learning agent to identify (true or false) friendly robots based on six nominal attributes. The attributes are head_shape (round, square, octagon), body_shape (round, square, octagon), is_smiling (yes, no), holding (sword, ballon, flag), jacket_color (red, yellow, green, blue) and has_tie (yes, no). The "true" concept for this problem is (head_shape = body_shape) or (jacket_color = red). The training dataset contains 124 examples and the validation dataset contains 432 examples.

B. Model description

To forecast the class according to the 17 input values (one per nominal value coded as 1 or -1 if the characteristic is true or false), the MLP starts with 3 hidden neurons containing a hyperbolic tangent activation function. This number of hidden neurons allows a satisfactory representation able to solve this discrimination problem.
The total number of weights for this fully connected network (including a bias) is 58. This value will have to be compared to the remaining weights after pruning.
After MLP training, the model is accepted if the mean of square error is ≤ 0.001 on both the training and the validation dataset, and the performance in classification are equal to 100% according to the confusion matrix.

1) *Train the network to minimum error.*
2) *Save the model (solution).*
3) *Compute* H^{-1}.
4) *Compute the saliencies for all the weights between the input layer and the hidden layer (Eq. 5).*
5) *Select the* m *smallest saliency weights.*
6) *For all selected* m *weights compute* M *using (Eq. 6) and* Δw *(Eq. 8).*
7) *Remove the* m *selected weights and use* Δw *to update all the weights in the network.*
8) *Repeat steps 2 to 7 until a stopping criterion is reached.*

Fig. 3. Generalized Flexible Optimal Brain Surgeon (GF-OBS) Algorithm

This stopping criterion is also used by the pruning methods. In this study, GF-OBS and G-OBS remove three weights at the same time.

C. Results

Comparing the variant methods is a difficult task. We select three values as measures of the performance for variable selection and optimization : the number of preserved weights, the number of pruned variables and the choice of pruned variables (rules respected) to see which variables are the most selected and which variables are rare or absent.

For each method, 300 different initializations were tested. The results are presented as histograms.

1) Variable Selection: In this section, we compare the following algorithms: Unit-OBS, F-OBS, GF-OBS, OBS and G-OBS for variable selection. The histograms (Fig. 4, Fig. 5) show the performance of Unit-OBS, F-OBD and GF-OBS for pruned variables and the histograms (Fig. 6, Fig. 7) show the performance of OBS and G-OBS methods.

The first remark is that not one of all OBS variants always reaches the same number of pruned variables. For every algorithm, the number of pruned variables differs from 2 to 12 according to the initialization. But the frequency depends on the algorithm. Unit-OBS presents some better results with high frequencies for a large number of pruned variables compared to the other methods. We can also see that F-OBS is better than OBS and GF-OBS is equivalent to OBS.

According to the first Monk rules (see section IV-A), the concept is true if the head shape is equal to the body shape i.e. if variable 1 = variable 4 and variable 2 = variable 5 and variable 3 = variable 6. Thus, a good method should preserve these variables and variable 12 (indeed, the concept is also true if variable 12 is true). Nevertheless, it is possible to obtain the desired performance eliminating some of the first six variables. In this case, a cleaver algorithm will prune couples of variables. For example, to prune variable 1 (head_shape=round) allows to prune variable 4 (body_shape=round). From this point of view, Unit-OBS presents a bad result compared to the others methods.

If we analyze the number of variables associated to a Monk rule by Unit-OBS and F-OBS methods, the ideal method is to eliminate 10 variables and to keep only 7 variables, the figure (Fig. 4) shows clearly that F-OBS overcomes Unit-OBS technique. In this way, we suggest to use F-OBS if we are interested to extract rules. It is also important to notice that, for the same number of variables removed by Unit-OBS, our method F-OBS allows to remove more weights (usually, twice more) which is also important for optimization purpose.

2) Model Optimization: In this section, we want to study the incidence to remove several weights on the capacity of the model (the number of preserved neurons as well as the number of weights).

The histograms in (Fig. 6, Fig. 7, Fig. 8) show the performance results of OBS and G-OBS for optimization.
The results show that G-OBS is able to obtain the same small number of selected variable and the same small number of preserved weights than OBS but the frequency of OBS is not considerably higher than G-OBS for this case.
Table I summarizes the variable selection and optimization analysis.

V. Conclusion

We have presented in this study new algorithms for variable selection and optimization in MLPs. We used statistic methods to compare empirical performances of these different variants. The first idea of this paper is to propose a new algorithm called F-OBS which focuses on eliminating the weights only between input and hidden layers of a 3-layer MLP. Unit-OBS presents some better results with high frequencies for a large number of pruned variables compared to F-OBS, but our algorithm is faster and keep better the variables which are associated to rules to extract.
The second idea is an implementation of G-OBS with a criterion to eliminate a subset of weights by selecting the weights with the smallest saliencies, which allows to make G-OBS faster. The results obtained are comparable to OBS, who allows to use G-OBS at a fast method for MLP topology optimization.
In aim to make F-OBS faster, we proposed GF-OBS which eliminates several weights at the same time.
Moreover, this paper presents a comparison between OBS and Unit-OBS more detailed that the previous works.

TABLE I
SUMMARY OF THE MODEL PERFORMANCES (1 IS ASSIGNED TO THE BEST METHOD, AND SO ON.)

Method	OBS	Unit-OBS	F-OBS	GF-OBS	G-OBS
Number of pruned variables	3	1	2	3	5
Choice of pruned variables	2	5	1	3	3
Number of preserved weights	1				2

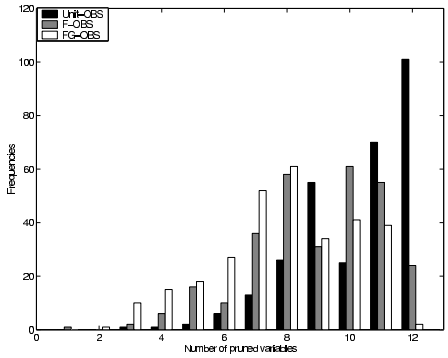

Fig. 4. Distribution of the number of pruned variables

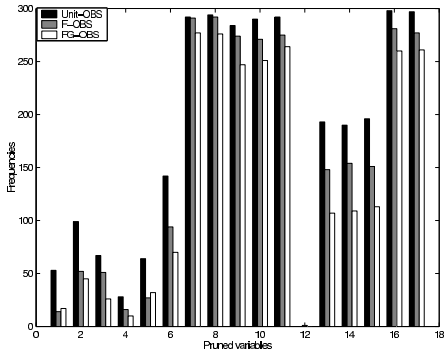

Fig. 5. Distribution of the pruned variables

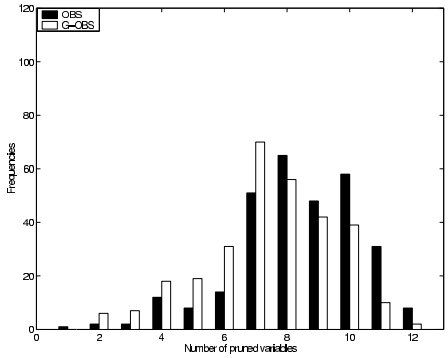

Fig. 6. Distribution of the number of pruned variables

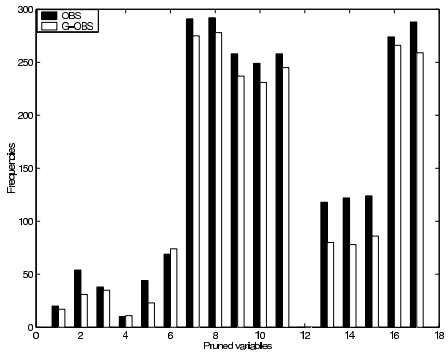

Fig. 7. Distribution of the pruned variables

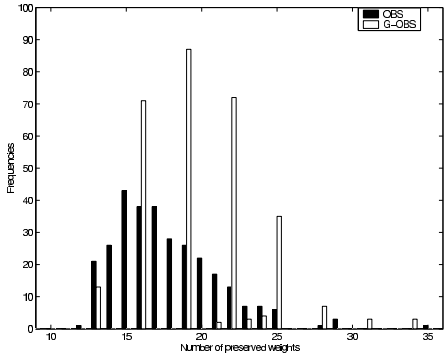

Fig. 8. Distribution of the number of preserved weights

ACKNOWLEDGMENT

This work was supported in part by the BRGM (Bureau de recherches géologiques et minières).

REFERENCES

[1] Y. Le Cun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems: Proceedings of the 1989 Conference*, D. S. Touretzky, Ed. San Mateo, CA: Morgan-Kaufmann, 1990, pp. 598–605.

[2] B. Hassibi and D. G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," in *Advances in Neural Information Processing Systems*, S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds., vol. 5. Morgan Kaufmann, San Mateo, CA, 1993, pp. 164–171. [Online]. Available: citeseer.nj.nec.com/hassibi93second.html

[3] T. Cibas, F. Soulie, P. Gallinari, and S. Raudys, "Variable selection with neural networks," 1996. [Online]. Available: citeseer.ist.psu.edu/cibas96variable.html

[4] T. Cibias, F. Souli, P. Gallinari, and S. Raudys, "Variable selection with optimal cell damage," 1994.

[5] A. Stahlberger and M. Riedmiller, "Fast network pruning and feature extraction by using the unit-OBS algorithm," in *Advances in Neural Information Processing Systems*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds., vol. 9. The MIT Press, 1997, p. 655.

[6] B. Hassibi, D. G. Stork, and G. Wolff, "Optimal brain surgeon: Extensions and performance comparison," in *Advances in Neural Information Processing Systems*, J. D. Cowan, G. Tesauro, and J. Alspector, Eds., vol. 6. Morgan Kaufmann Publishers, Inc., 1994, pp. 263–270.

[7] B. Hassibi, D. G. Stork, and G. J. Woff, "Optimal brain surgeon and general network pruning," in *Proceedings of 1993 IEEE International Conference on Neural Networks (Joint FUZZ-IEEE'93 and ICNN'93 [IJCNN93])*, vol. I. San Francisco, California: IEEE/INNS, Mar.-Apr. 1993, pp. 293–299, ricoh.

[8] S. B. Thrun, J. Bala, E. Bloedorn, I. Bratko, B. Cestnik, J. Cheng, K. D. Jong, S. Džeroski, S. E. Fahlman, D. Fisher, R. Hamann, K. Kaufman, S. Keller, I. Kononenko, J. Kreuziger, R. S. Michalski, T. Mitchell, P. Pachowicz, Y. Reich, H. Vafaie, W. V. de Welde, W. Wenzel, J. Wnek, and J. Zhang, "The MONK's problems: A performance comparison of different learning algorithms," Pittsburgh, PA, Tech. Rep. CS-91-197, 1991.