Driven Forward Features Selection: a comparative study on Neural Networks

Vincent Lemaire and Raphael Féraud

France Télécom R&D Lannion,
vincent.lemaire@orange-ft.com

Abstract. In the field of neural networks, feature selection has been studied for the last ten years and classical as well as original methods have been employed. This paper reviews the efficiency of four approaches to do a driven forward features selection on neural networks . We assess the efficiency of these methods compare to the simple Pearson criterion in case of a regression problem.

1 Introduction

Up to 1997, when a special issue on relevance including several papers on variable and feature selection was published, few domains explored more than 40 features. The situation has changed considerably in the past few years, notably in the field of data-mining with the availability of ever more powerful data warehousing environments. A recent special issue of JMLR [1] gives a large overview of techniques devoted to variable selection and an introduction to variable and feature selection can be found in this special issue [2]. A challenge on feature selection has been organized during the NIPS 2003 conference to share techniques and methods on databases with up to 100000 features. This challenge lead to provide an interesting and exhaustive book [3].

The objective of variable selection is three-fold: improve the prediction performance of the predictors, provide faster and more cost-effective predictors, and allow a better understanding of the underlying process that generated data. Among techniques devoted to variable selection, we find filter methods, which select variables without using a model (for example by ranking them with correlation coefficients), and subset selection methods, which assess subsets of variables according to their usefulness to a given model. Wrapper methods [4] use the elaborated model as a black box to score subsets of variables according to their usefulness for the modeling task. In practice, one needs to define: (i) how to search the space of all possible variable subsets; (ii) how to assess the prediction performance of a model to guide the search and halt it; (iii) how to select the predictor to use.

We discuss in this paper the problem of feature selection and review four methods which have been developed in this field. The main idea is to compare four popular techniques in sense of methods which are integrated in data mining software (Clementine, SAS, Statistica Data Miner...). This paper presents the comparison specifically for neural networks (NN) therefore point (iii) listed above is fixed.

The remainder of the document is organized as follows. Next section deals with classical ingredients which are required in feature selection methods (1) a feature evaluation criterion to compare variable subsets (2) a search procedure, to explore (sub)space of possible variable combinations (3) a stop criterion or a model selection strategy. The section 3 presents the driven forward strategy and four methods to do variable selection with neural networks. Section 4 proceeds with an experimental evaluation on each method on the driven forward strategy for a regression problem.

2 Basic ingredients of feature selection methods

For all methods in this paper, the notations employed are **(1)** about data distribution: J the number of variables in the full set; I the number of examples in the training set; V_j the variable for which we look for the importance; V_{ij} the realization of the variable V_j for the example i; I_m the input vector part of the example m with n components; $P_{V_j}(u)$ the probability distribution of the variable V_j; $P_I(\nu)$ the probability distribution of examples I; and **(2)** about neural network: OL the output layer; HL the hidden layer; IL the input layer; w_{wz} a weight between a neuron w and a neuron z; f the predictive model (here a neural network); Y_m the output vector part of the example m; and $f_j(a;b) = f_j(a_1,...,a_n;b) = f(a_1,...,a_{j-1},b,a_{j+1},...,a_n)$ where a_p is the p^{th} component of the vector a. Finally we note $S(V_j|f)$ as being the importance of the variable V_j using the predictive model f. Note that all methods are presented for an output vector which has only one component but extension to many component is straightforward.

2.1 Features evaluation

Several evaluation criteria, based either on statistical grounds or heuristics, have been proposed for measuring the importance of a variable subset. For regression, classical candidates are prediction error measures. We will use the mean squared error to compare results in section 4. A survey of classical statistical methods may be found in [5] for regression, [6] for classification, [3] for both; and [7] for neural networks .

2.2 Search strategy

In general, since evaluation criteria are non monotonous, comparison of feature subsets amounts to a combinatorial problem which rapidly becomes computationally unfeasible. Most algorithms are based upon heuristic performance measures for the evaluation and sub-optimal search. Most sub-optimal search methods follow one of the following sequential search techniques [8]: (a) start with an empty set of variables and add variables to the already selected variable set (forward methods); (b) start with the full set of variables and eliminate variables from the selected variable set (backward methods); (c) start with an empty set and alternate forward and backward steps (stepwise methods). In this paper we will compare criteria only with a driven forward strategy described below.

2.3 Driven Forward Selection.

In this paper we define a driven forward selection strategy such as: 1) compute the variable importance using a criterion; 2) rank the variables using the result of the first step; 3) train models where variables are added more and more using the ranking of the variable importance computed in the second step; 4) observe the results versus the number of variables used. This strategy is driven since the first ranking is not questioned and therefore one have at most J model to train.

A simple driven forward strategy uses, for example, the Pearson correlation coefficient which is adapted for linear dependencies[1] and which is not model oriented (it does not take into account the regression model during selection):

$$S(V_j|f) = S(V_j) = \frac{\sum_{i=1}^{I} \left(V_{ij} - \overline{V_j}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_{i=1}^{I} \left(V_{ij} - \overline{V_j}\right)^2 \sum_{i=1}^{I} \left(Y_i - \overline{Y}\right)^2}} \tag{1}$$

For Pearson criterion the driven strategy described is clear since this criterion does not need to use a model in the first step ($S(V_j|f) = S(V_j)$). However any wrapper criterion which allows to measure variable importance could be use in the same way. In this case there is a preliminary step which is to train a model which uses the full set. Then the first step compute the variable importance using this model ($S(V_j|f)$). Others step are not changed. What we can except is that all criteria studied in this paper can achieved better results than using Pearson criterion.

2.4 Stopping criterion

No stopping criterion has been used in this paper. The performance obtained by each variable selection method has been memorized to be able to plot all results on all selected variables subset with all criteria.

3 Features Selection Methods with Neural Networks Compared

3.1 A Feature Selection Method based on Empirical Data Probability

The method described here [9] combines the definition of the ‘variable importance’ as given in Féraud et al. [10] with an extension of Breiman’s idea [11].

[1] To capture non linear dependencies, the mutual information is more appropriate but it needs estimates of the marginal and joint densities which are hard to obtain for continuous variables. This method has not been tested in this paper.

This new definition of variable importance both takes into account the probability distribution of the studied variable and the probability distribution of the examples. The importance of an input variable is a function of examples I probability distribution and of the probability distribution of the considered variable (V_j). This method is tested for the first time in this paper on a regression problem.

The importance of the variable V_j is the sum of the measured variation of the predictive model output when examples are perturbed according to the probability distribution of the variable V_j. The perturbed output of the model f, for an example I_i is the model output for this example but having exchanged the $\mathrm{j^{th}}$ component of this example with the $\mathrm{j^{th}}$ component of another example, k. The measured variation, for the example I_i is then the difference between the 'true output' $f_j(I_i;V_{ij})$ and the 'perturbed output' $f_j(I_i;V_{kj})$ of the model. The importance of the variable V_j is computed on both the examples probability distribution and the probability distribution of the variable V_j. The importance of the variable V_j for the model f is then:

$$S(V_j|f) = \iint P_{V_j}(u)du P_I(v)dv \left| f_j\left(I_i;V_{ij}\right) - f_j\left(I_i;V_{kj}\right) \right| \tag{2}$$

Approximating the distributions by the empirical distributions, the computation of the average of $S(V_j|f)$ would require to use all the possible values of the variable V_j for all examples available such as:

$$S(V_j|f) = \frac{1}{I} \sum_{i \in I} \sum_{k \in I} \left| f_j\left(I_i;V_{ij}\right) - f_j\left(I_i;V_{kj}\right) \right| \tag{3}$$

As the variable probability distribution can be approximated using representative examples (P) of an ordered statistic:

$$S(V_j|f) = \frac{1}{I} \sum_{i \in I} \sum_{p \in P} \left| f_j\left(I_i;V_{ij}\right) - f_j\left(I_i;v_p\right) \right| \mathrm{Prob}(v_p) \tag{4}$$

This method is especially useful when V_j takes only discrete values since the inner sum is exact and not an approximation. View the size of the database used for comparison section 4 P has been fixed to 10 (the deciles are used). For all deciles we chose to used their median as representative values. This approximation allows to speed up the computation and prevents errors which are due to outliers or pathological values.

3.2 A Features Selection Method based on Neural networks weights

This method uses only the network parameter values. Although this is not sound for non linear models, there have been some attempts for using the input weight values in the computation of variable relevance. The weight value in the input layer[2], IL, can provide information about variable importance. The variable importance based on neural networks weights is:

$$S(V_j|f) = \frac{\sum_{z \in HL} \|w_{zj}\|}{\sum_{z \in HL} \sum_{w \in IL} \|w_{zw}\|} \tag{5}$$

3.3 A Features Selection Method based on saliency

Several methods propose to evaluate the relevance of a variable by the derivative of the error or of the output with respect to this variable. These evaluation criteria are easy to compute, most of them lead to very similar results. These derivatives measure the local change in the outputs with respect of a given input, the other inputs being fixed. Since these derivatives are not constant as in linear models, they must be averaged over the training set. For these measures to be fully meaningful inputs should be independent and since these measures average local sensitivity values, the training set should be representative of the input space (which is a minimum assumption).

The Saliency Based Pruning method [13] uses as evaluation criterion the variation of the learning error when a variable V_j is replaced by its empirical mean $\overline{V_j}$ (zero if variables are assumed centered). The saliency is:

$$S(V_j|f) = \frac{1}{I}\left(\sum_{i=1}^{I} \left\|f(I_i; V_{ij}) - y_i\right\|^2\right) - \frac{1}{I}\left(\left\|\sum_{i=1}^{I} f(I_i; \overline{V_j}) - y_i\right\|^2\right) \tag{6}$$

This is a direct measure of the usefulness of the variable for computing the output. Changes in MSE are not ambiguous only when inputs are not correlated. Variable relevance being computed once here, this method does not take into account possible correlations between variables.

3.4 A Features Selection Method based on output derivatives

Several authors have proposed to measure the sensitivity of the network transfer function with respect to input V_j by computing the mean value of outputs derivative with respect to V_j over the whole training set. Most measures use average squared or absolute derivatives [14–16]. The variable importance is: $S(V_j|f) = \frac{1}{I}\sum_{i=1}^{I}(\partial f/\partial V_j(V_{ij}))$. These measures being very sensitive to the input space representativeness of the sample set, several authors have proposed to use a subset of the sample in order to increase the significance of their relevance measure. In order to obtain robust methods, "non-pathological" training examples should be discarded. A parameter, here ϵ, is needed to adjust the

[2] A more sophisticated heuristic, but very close to the one above in case of a single output neuron, has been proposed by Yacoub and Bennani [12], it exploits both the weight values and the network structure of a multilayer perceptron.

range variation over V_j given an example (V_{ij}). In this paper we choose to use the definition:

$$S(V_j|f) = \frac{1}{I}\sum_{i=1}^{I} |f_j(I_i, Vij - \epsilon) - f_j(I_i, Vij + \epsilon)| \tag{7}$$

4 Experimental results on Orange Juice Database

4.1 Experimental conditions and results presentations

Database: The database has been provided by Prof. Marc Meurens, Université Catholique de Louvain, BNUT unit. The goal is to estimate the level of saccharose of an orange juice from its observed near-infrared spectrum. The training set is constituted of 150 examples described by 700 features (variables) and the test set is constituted of 68 examples described also by 700 features. There is no missing value and variables are continuous but note that the number of training examples (150) is more of four times as small as the number of features (700). Nothing else is known about this database (see `http://www.ucl.ac.be/mlg/index.php?page=DataBases`). The preprocessing used for input variable as well as for output variable is only a min-max standardization. All the results presented below (the mean squared error) are computed on the standardized output.

Cross Validation: For all experimental conditions, 25 trainings are performed with different initialization of the weights and different training, validation set as follow: we have drawn a training set (100 examples) from the training set available on the web site (among 150) and the others example of the training set has been used as a validation set. Each training is stopped when the cost (the mean squared error) on the validation set does not decrease since 200 iterations. At the end of each training, the global mean squared error on the test set is computed for comparison purposes. In the driven forward strategy the variables importance are not questioned. So, when one gives results over 25 training there are results over 25 forward procedures (for a given step, a given number of variables, the variables chosen are not necessary the same to compute the mean errors presented in Figure 7).

Neural network topology and training parameters: A single multilayer perceptron with 1 hidden layer, tangent hyperbolic activation function and stochastic back-propagation of the squared error as training algorithm has been used. Using full set of variables the learning rate has been determined to be α=0.001 and the number of hidden unit has been determined to be HL=15. Again, these parameters has been evaluated over 25 training from a range variation of α from 0.0001 up to 0.1 and HL from 1 up to 30.

Regularization: The orange juice database is constituted of 700 variables which are very correlated to the output target (see Figure 1, coefficients between normalized input variables and the normalized output). Methods presented above test the importance of all variables one by one so a successful

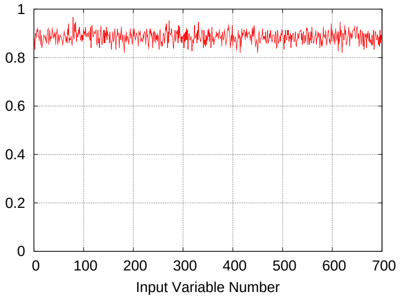

Fig. 1. Absolute Pearson coefficient.

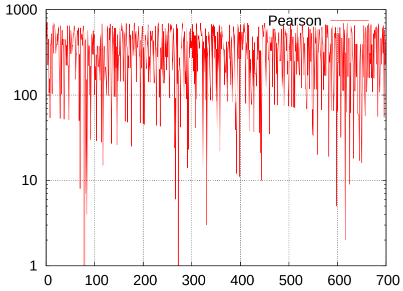

Fig. 2. Ranking of Pearson coefficient.

regularization method has to be employed. We added a regularization term active only on directions in weight space which are orthogonal to the training update [17]. This regularization prevents correlation effects between input variables without learning degradations. The regularization term (in batch procedure for it) has been always 10^{-3} of the learning rate..

4.2 Comparison using the full set and a same neural network

Figures 3,4,5,6 show variable importance found using the five criterion described above (except Pearson criterion for which one can see this representation in Figure 2) and computed with the same neural networks trained with the full set of variables. Figure 3, Figure 4, Figure 5, Figure 6 show respectively versus the number of the variables the “Norm Importance” obtained using equation 5, “Saliency Importance” obtained using equation 6, “Local Importance” obtained using equation 7 and “Global Importance” obtained using equation 4 . On all sub figure horizontal axis represents the number of the variables and vertical axis represents (in log-scale to focus on first important variables) the ranking of the variables from 1 (the most useful) to 700 (the less useful). This representation identifies clearly first important variables for all criterion using the same neural network and allows to compare behaviors.

The four criteria Norm, Saliency, Local and Global do not agree with Pearson criterion (see Figure 2). For criteria Norm, Local and Global important variables are near the six hundredth variable. Saliency criterion selects variables near the 130th. Global criterion ranks this group after the group near the six hundredth variable. Among group near the 600th variable Global criterion does not order variables as Norm and Local criteria (the 562th before the 592th). Norm and Local criteria very agree on this regression problem. Results presented in next section with the driven forward procedure will give more results elements.

4.3 Results with the driven forward strategy

Whatever is the neural network trained the results obtained using Pearson criterion will be the same since this criterion does not use the model to compute

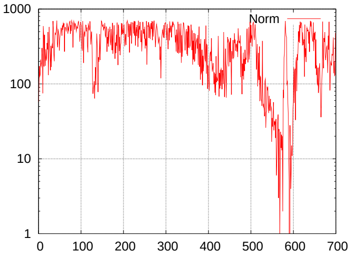

Fig. 3. 'Norm Importance'

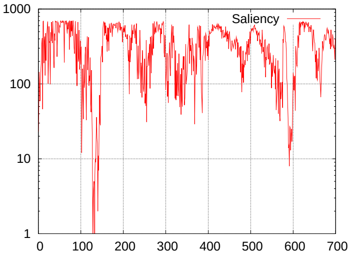

Fig. 4. 'Saliency Importance'

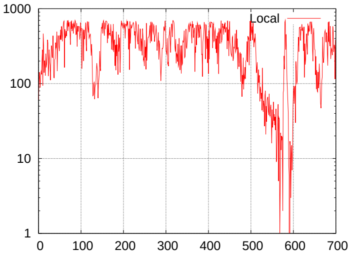

Fig. 5. 'Local Importance'

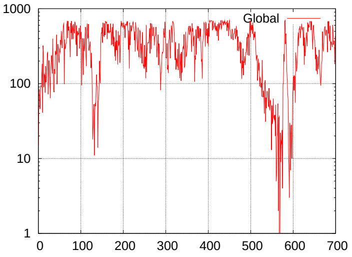

Fig. 6. 'Global Importance'

variable importance. But it is not the case for others criteria described above. The ranking obtained can depend on the neural network trained and therefore of its initialization, the order to present examples, etc... For all criteria 20 neural networks ($k = 20$) have been trained using the full set of variables. The mean value of the criterion has been computed on all neural networks such as: $\overline{S(V_j|f)} = 1/k \sum_k S(V_j|f_k)$. Using this mean value on all variables a ranking has been determined. Table 1 presents this ranking. Then this ranking has not been questioned. It is used to train neural networks which used one, two or more important variables. Experimentations have been made twenty times to obtained mean results using one, two or more important variables on all criteria.

Table 1. The ten more important variables.

Pearson	80	273	85	332	617	71	83	268	599	118
Norm	595	596	592	593	590	594	591	570	597	598
Saliency	595	131	1	2	129	3	130	592	6	593
Local	595	592	596	593	590	594	591	599	597	598
Global	570	595	592	596	590	593	594	572	569	571

The Figure 7 presents results obtained with the four methods and Pearson criterion which is a baseline results. Results after 100 variables are not presented since they are the same for all criteria and are the same than using the full set. Each plot represents the mean results of the mean squared error on the normalized output through 20 forward procedures. The standard deviation is not represented for reading reasons and a figure which is not overloaded. The standard deviation is ± 0.003 for all points. For example, for the Pearson criterion and using ten variables, the result is therefore 0.035 ± 0.003.

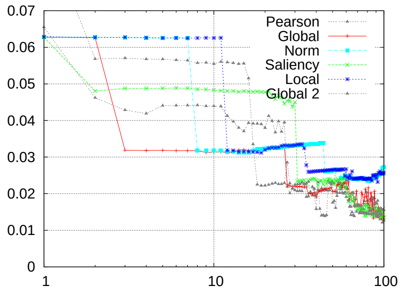

Fig. 7. Mean squared error using driven forward strategy versus the number of variables used.

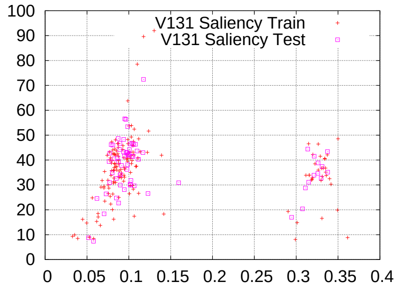

Fig. 8. Neural network outputs versus the ordered values of the 131th variable.

On this regression problem, which is compose of a full set of 700 variables and few examples for training, we observe the following ranking of criteria (from the best to the least): (1) Global, (2) Saliency (3) Local and Norm, (3) Pearson. With less than 100 variables criteria Global, Saliency and Pearson obtain the same results than using all variables (0.011 ± 0.003). To obtain this performance Norm and Local criteria need 150 variables. Significant degradations on results appear under 60 variables on all criteria. The Global criterion gives excellent and best results: better performances of the neural network trained are always obtained before others (until all criteria allow to obtained same results).

To analysis more in depth the difference in term of performances we focus on the 131th variable since there is a disagreement between criteria for this variable. We plot on Figure 8 ordered values of the 131th variable on horizontal axis and the estimated output on the vertical axis (using the same neural network as in section 4.2). Clearly for this variable, which constituted by two groups of values, it is not relevant to measure its importance with saliency: its mean is out of the data distribution. This discontinuity explains the overestimation of the variable importance using Saliency criterion. On the other hand, Local criterion does not rank this 131th variable in the ten most important variables since derivatives importance is not adapted to bimodal distribution. The Global criterion where data distribution is used is able to take into account bimodal distribution. It ranks this variable as an important variable. This type of difference in behaviors explains the difference in performances.

5 Conclusion

These comparisons show that, on this real application, it is possible to obtain excellent performances with the four criteria with a large preference for the Global criterion; knowing that the database used is a particular database with very correlated variables and few examples compare to the number of the full set of variables. Future work should address experiments on larger data sets[3].

References

1. JMLR, ed.: Special Issue on Variable and Feature Selection. Volume 3(Mar). Journal of Machine Learning Research (2003)
2. Guyon, I., Elisseef, A.: An introduction to variable and feature selection. JMLR **3(Mar)** (2003) 1157–1182
3. Guyon, I.: To appear - Feature extraction, foundations and applications. - (2006)
4. Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intelligence **97(1-2)** (1997)
5. Thomson, M.L.: Selection of variables in multiple regression part i: A review and evaluation and part ii: Chosen procedures, computations and examples. International Statistical Review **46:1-19 and 46:129-146** (1978)
6. McLachlan, G.: Discriminant Analysis and Statistical Pattern Recognition. Wiley-Interscience publication (1992)
7. Leray, P., Gallinari, P.: Feature selection with neural networks. Technical report, LIP6 (1998)
8. Miller, A.J.: Subset Selection in Regression. Chapman and Hall (1990)
9. Lemaire, V., Clérot, C.: An input variable importance definition based on empirical data probability and its use in variable selection. In: International Joint Conference on Neural Networks IJCNN. (2004)
10. Féraud, R., Clérot, F.: A methodology to explain neural network classification. Neural Networks **15** (2002) 237–246
11. Breiman, L.: Random forest. Machine Learning **45** (2001)
12. Yacoub, M., Bennani, Y.: Hvs: A heuristic for variable selection in multilayer artificial neural network classifier. In: ANNIE. (1997) 527–532
13. Moody, J.: Prediction Risk and Architecture Selection for Neural Networks. From Statistics to Neural Networks-Theory and Pattern Recognition. Springer-Verlag (1994)
14. Ruck, D.W., Rogers, S.K., Kabrisky, M.: Feature selection using a multilayer perceptron. J. Neural Network Comput. **2**(2) (1990) 40–48
15. Réfénes, A.N., Zapranis, A., Utans, J.: Stock performance using neural networks: A comparative study with regression models. Neural Network **7** (1994) 375–388
16. Refenes, A., Zapranis, A., Utans, J.: Neural model identification, variable selection and model adequacy. In: Neural Networks in Financial Engineering, Proceedings of NnCM-96. (1996)
17. Burkitt, A.N.: Refined pruning techniques for feed-forward neural networks. Complex System **6** (1992) 479–494

[3] as for example `http://theoval.cmp.uea.ac.uk/~gcc/competition/`