A faster model selection criterion for OP-ELM and OP-KNN: Hannan-Quinn criterion

Yoan Miche[1,2] and Amaury Lendasse[1]

1- Helsinki University of Technology - ICS Lab.
Konemiehentie 2, 02015 TKK - Finland

2- INPG Grenoble - Gipsa-Lab, UMR 5216
961 rue de la Houille Blanche, Domaine Universitaire, 38402 GRENOBLE - France

Abstract. The Optimally Pruned Extreme Learning Machine (OP-ELM) and Optimally Pruned K-Nearest Neighbors (OP-KNN) algorithms use the a similar methodology based on random initialization (OP-ELM) or KNN initialization (OP-KNN) of a Feedforward Neural Network followed by ranking of the neurons; ranking is used to determine the best combination to retain. This is achieved by Leave-One-Out (LOO) cross-validation. In this article is proposed to use the Hannan-Quinn (HQ) Criterion as a model selection criterion, instead of LOO. It proved to be efficient and as good as the LOO one for both OP-ELM and OP-KNN, while decreasing computations by factors of four to five for OP-ELM and up to 24 for OP-KNN.

1 Introduction

Since data can be collected automatically from various and numerous sources, the global amount of information grows rapidly in most fields of science. Although this data most likely improves precision and details, it also raises many new challenges such as storage and processing. Among the most famous algorithms used for data processing through machine learning techniques, lies Feedforward neural networks. While multilayer feedforward neural networks have been proved to be universal approximators [1], they tend not to be used widely when processing important datasets because of the computational time it takes to actually train and build them: many parameters are required for a proper selection of the model structure and afterwards, the training.

In order to make model training and selection of single hidden layer feedforward neural networks faster, OP-ELM [2] (based on ELM [3]) and OP-KNN [4] have been proposed recently. In this paper, is proposed a different model structure selection criterion (inside the OP-ELM/KNN algorithm) to replace the previously used Leave-One-Out; it is just as efficient and faster for large datasets. The next section presents the OP-ELM/KNN shortly, while section 3 details the Hannan-Quinn criterion used for complexity selection. Experiments and results using this improved methodology are presented in section 4.

2 OP-ELM and OP-KNN

The Otimally Pruned Extreme Learning Machine [2] (OP-ELM, based on original ELM [3]) and Optimally Pruned KNN [4] (OP-KNN) are based on similar

first steps which Figure 1 summarizes.

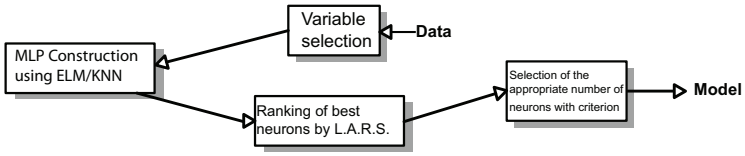

Fig. 1: OP-ELM/KNN methodology: first steps are similar. Last step of selection of neurons is performed using a criterion: Leave-One-Out (LOO) in the original algorithms.

A priori variable selection is first performed on the data. Then, the MultiLayer Perceptron (MLP), which is actually a single hidden layer feedforward network, is initialized, either by ELM (for OP-ELM) or by KNN (for OP-KNN). In the OP-ELM case, by a random initialization of the weights and biases of the MLP, while for OP-KNN, deterministic initialization using K-NN is used.

Neurons are then ranked using a MRSR [5] technique, which main idea is: Denote by $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ the $N \times M$ matrix of inputs, the MRSR adds each column of the regressor matrix one by one to the model $\hat{\mathbf{Y}}^k = \mathbf{X}\mathbf{W}^k$ where $\hat{\mathbf{Y}}^k = [\hat{\mathbf{y}}_1^k, \ldots, \hat{\mathbf{y}}_p^k]$ is the target approximation of the model. The $\mathbf{W}^k$ weight matrix has k nonzero rows at k-th step of the MRSR. With each new step a new nonzero row and a new column of the regressor matrix is added to the model. An important fact is that the obtained ranking by MRSR is exact in the case of a linear problem, as here since the neural network is linear between the hidden layer and output.

Finally, a criterion is used to decide which number of neurons will be retained (ranked neurons, so only the best ones are kept). This criterion is a Leave-One-Out, for the original OP-ELM/KNN.

The Leave-One-out (LOO) is usually a costly way of estimating a model's fit to the data, since it requires to go through all points of the data separately to estimate the model's output for each. In order to keep the OP-ELM/KNN fast, the PRESS Statistics [6] formula is used in order to compute this validation error, as in Eq. 1.

$$\epsilon_i^{\text{PRESS}} = \frac{y_i - \mathbf{x}_i \mathbf{b}}{1 - \mathbf{x}_i \mathbf{P} \mathbf{x}_i'}, \tag{1}$$

where $\mathbf{P} = (\mathbf{X}'\mathbf{X})^{-1}$, and $\mathbf{b}$ are the output weights of the MLP. While this formula makes it possible to evaluate the LOO error with simple matrix calculations, it still requires a matrix inversion and various matrix products, which can still be long. The goal of this paper is to present another criterion for the complexity selection (by the selection of the number of neurons to keep) which is much faster for it does not requires these matrix operations.

3 The Hannan-Quinn criterion

In order to perform complexity selection (by selecting the neurons to retain in the OP-ELM/KNN model), the classical LOO was used in the original versions of the two algorithms OP-ELM/KNN.

There are many possible criteria for complexity selection in machine learning. Typical examples are Akaike's information criterion (AIC) [7] or the Bayesian Information Criterion (BIC) [8]. Their expression is based on the residual sum of squares (Res) of the considered model (first term of the criterion) plus a penalty term (second term). Differences between criteria mostly occur on the penalty term. AIC penalizes only with the number of parameters p of the model (so that not too many free parameters are used to obtain a good fit by the model), Eq. 3; BIC takes into account the number of samples N used for the model training, in Eq. 2.

$$BIC = N \times \log\left(\frac{Res}{N}\right) + p \times \log N \tag{2}$$

$$AIC = N \times \left(\log\left(\frac{2\pi Res}{N}\right) + 1\right) + 2 \times p \tag{3}$$

The AIC is known to have consistency problems: while minimizing AIC, it is not guaranteed that complexity selection will converge toward an optima if the number of samples goes to infinity [9]. The main problem using such criteria is in trying to balance underfitting and overfitting knowing that convergence might never be achieved. One solution is through the penalty term, for example, by having a $\log N$ term in the penalty (with N the number of samples), which the BIC has. Unfortunately, for the experiments conducted in this paper, the BIC criterion did not give proper complexity selection (most likely due to the too fast increase of the penalty term with the number of samples).

The Hannan-Quinn Information Criterion [10] is close to these other criteria, as can be seen from the expressions of the AIC and BIC below (Eq. 2 and Eq. 3). The idea behind the design of this criterion is to provide a consistent criterion (regarding for example AIC which is not consistent in its standard definition) in which the second term (the penalty) $2 \times p \times \log\log N$ grows but at a very slow rate, regarding the number of samples.

$$HQ = N \times \log\left(\frac{Res}{N}\right) + 2 \times p \times \log\log N \tag{4}$$

From Figure 2, it can be seen that both criteria have very similar convergence regarding the number of neurons used for building the model. In this particular case, the HQ criterion is consistent since it enables a stable convergence. Hence, and from the following experiments, it can be considered that the HQ criterion is as good as the original LOO.

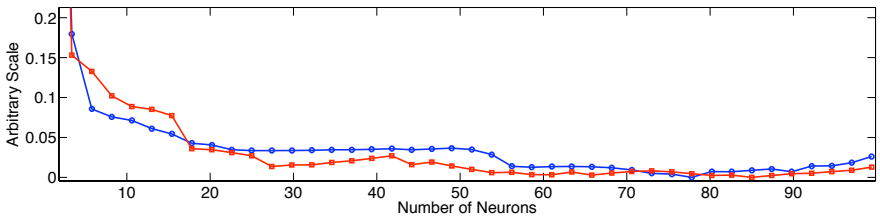

Fig. 2: Plot of the criterion value for both LOO and HQ versions of the OP-ELM (for the Bank dataset from UCI [11]): red squares for HQ and blue circles for LOO. Plots have been scaled to fit on same scale (HQ and LOO criteria have very different values). Convergence is very similar.

4 Experiments and results

Experiments for testing the effect of the HQ criterion on both OP-ELM and OP-KNN, have been conducted using seven different data sets from UCI machine learning repository [11]. The choice of these datasets has been made so that their variety in terms of number of samples and variables, covers usual "real life" datasets. Table 1 summarizes the characteristics of theses datasets.

		Samples	
Regression	# of Variables	Train	Test
Ailerons (D.A.)	5	4752	2377
Elevators (D.E.)	6	6344	3173
Auto price (A.P.)	15	106	53
Servo	4	111	56
Breast Cancer (B.C.)	32	129	65
Bank	8	2999	1500
Stocks	9	633	317

Table 1: Selected datasets: Number of variables and number of samples for training and testing (two thirds and one third respectively).

The datasets have been divided in two parts: training and test sets. Two thirds of the whole dataset for training and the remaining third for testing.

The original OP-ELM/KNN and their HQ modified versions have both been tested on these datasets, and results for test mean square errors and computational times are presented in Tables 2 and 3. It can be seen from Table 2 that the HQ version of the algorithms perform just as good, on average, as the original LOO-based version (or even slightly better). Results are within close range with SVM values (from LS-SVM [12]).

Computational times are highly reduced, when using the HQ criterion instead of the LOO, as expected (Table 3). A factor of four to five between the computational times, can be observed for OP-ELM, and up to 24 for the OP-

	A.P.	Bank	B.C.	D.A.	D.E.	Servo	Stocks
SVM	3.8E+06	2.2E-03	8.9E+02	2.6E-08	2.8E-06	4.2E-01	2.2E-01
OP-ELM	4.5E+06	1.1E-03	6.7E+02	2.7E-08	1.9E-06	5.8E-1	6.1E-1
OP-ELM-HQ	1.4E+06	1.1E-03	9.2E+02	2.6E-08	1.9E-06	5.7E-1	5.8E-1
# Neur. (HQ)	20 (50)	98 (98)	8 (4)	55 (95)	36 (26)	39 (100)	99 (99)
OP-KNN	2.7E+06	1.3E-03	1.1E+03	3.4E-08	2.5E-06	4.0E-01	4.8E-01
OP-KNN-HQ	3.1E+06	1.3E-03	1.1E+03	3.4E-08	2.4E-06	3.8E-01	4.8E-01
# Neur. (HQ)	46 (100)	45 (15)	2 (6)	23 (20)	17 (17)	59 (59)	11 (11)

Table 2: Test Mean Square errors comparisons for OP-ELM/KNN and HQ criterion version. Number of neurons for each are given: standard version in plain and HQ in parenthesis. 100 neurons used for OP-ELM and maximum 100-nearest neighbours for OP-KNN. SVM values for reference (using LS-SVM [12]).

KNN. It can also be seen that this difference is mostly noticeable when using large datasets (Ailerons, Elevators, Bank, here). While the difference is smaller for smaller datasets, it remains important enough to be considered when the OP-ELM/KNN is used many times, for variable selection with a Forward-Backward algorithm, for example. In these case, the small difference in computational time makes a clear difference on the many iterations.

	A.P.	Bank	B.C.	D.A.	D.E.	Servo	Stocks
SVM	492	6.5E+05	645	8.7E+04	7.7E+05	863	2188
OP-ELM	0.14	5.37	0.29	18.99	21.89	0.42	1.33
OP-ELM-HQ	0.13	1.88	0.24	4.59	5.42	0.40	0.84
Ratio	1.08	2.86	1.21	4.14	4.04	1.05	1.58
OP-KNN	1.6	17.7	0.09	43.16	67.04	0.08	0.91
OP-KNN-HQ	0.09	1.02	0.06	1.78	2.47	0.05	0.19
Ratio	17.78	17.35	1.5	24.25	27.14	1.6	4.79

Table 3: Computational times (seconds) when using OP-ELM/KNN and the HQ criterion version. Ratios between standard and HQ versions are given. Improvement with HQ criterion is visible when working with large datasets (matrix products for classical PRESS LOO need to be consequent for the HQ based version to take advantage). SVM values for reference (using LS-SVM [12]).

5 Conclusion

This paper presents a modification of the original OP-ELM/KNN algorithms in the place of the model structure selection criterion. The classical Leave-One-Out criterion is replaced by the Hannan-Quinn (HQ) criterion, which performances match the ones of the LOO (or perform actually slightly better, on average). The main advantage of this other criterion over the LOO one is to decrease the

computational times by a factor of four to five for OP-ELM and up to 24 for OP-KNN, for the conducted experiments. It seems very likely that for much larger datasets than the ones used for the experiments, the gain in computational time could be even higher.

References

[1] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[2] Y. Miche, P. Bas, C. Jutten, O. Simula, and A. Lendasse. A methodology for building regression models using extreme learning machine: OP-ELM. In *ESANN 2008, European Symposium on Artificial Neural Networks, Bruges, Belgium*, April 23-25 2008.

[3] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1–3):489–501, December 2006.

[4] Q. Yu, A. Sorjamaa, Y. Miche, A. Lendasse, A. Guillén, E. Séverin, and F. Mateo. Optimal pruned k-nearest neighbors: OP-KNN - application to financial modeling. In *HIS 2008, 8th International Conference on Hybrid Intelligent Systems*, September 10-12 2008.

[5] T. Similä and J. Tikka. Multiresponse sparse regression with application to multidimensional scaling. In *Artificial Neural Networks: Formal Models and Their Applications - ICANN 2005*, volume 3697/2005, pages 97–102. 2005.

[6] R.H. Myers. *Classical and Modern Regression with Applications, 2nd edition.* Duxbury, Pacific Grove, CA, USA, 1990.

[7] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, December 1974.

[8] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[9] R. J. Bhansali and D. Y. Downham. Some properties of the order of an autoregressive model selected by a generalization of akaike's epf criterion. *Biometrika*, 64(3):547–551, 1977.

[10] E. J. Hannan and B. G. Quinn. The determination of the order of an autoregression. *Journal of the Royal Statistical Society, B*, 41:190–195, 1979.

[11] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[12] Suykens J.A.K., Van Gestel T., De Brabanter J., B. De Moor B., and Vandewalle J. *Least Squares Support Vector Machines.* World Scientific, Singapore, 2002.