Comparison of Artificial Neural Networks; and training an Extreme Learning Machine

Miquel Perelló, E-mail: miquel.perello.nieto@est.fib.upc.edu

Abstract—In this work, I analysed different types of Artificial Neural Networks with two software. There is a brief explanation of basic points of the six neural networks used for the experiment. After comparing and analysing the results, there is a parameter adjustment section for an Extreme Learning Machine Neural Network with the objective of outperform the previous results.

Keywords—Artificial Neural Network, Extreme Learning Machine.

INTRODUCTION

THIS work shows the performance of some Artificial Neural Networks (ANN) incrementing the number of hidden units. These ANN are executed in two different software explained above. After that, I adjusted different parameters of an ELM trying to outperform the best results.

0.1 Software

One of the programs is in Matlab. It uses the PRTools libraries. Some of the ANN we can train with that program are an Extreme Learning Machine (ELM) and Levenberg Marquardt Neural Network (LM). On the other hand, we have a C++ program developed by Prof. Lluís Belanche from Soft Computing department (SOCO) of Universitat Politécnica de Catalunya (UPC). With this software we can train and test an Heterogeneous Neural Network (HNN), a Multilayer Perceptron (MLP) and a Radial Basis Function Neural Network (RBF).

0.2 Dataset

For all comparisons I used the Annealing dataset. This dataset and more are available from the University of California, Irvine (UCI) webpage (http://archive.ics.uci.edu/ml/).

It is not an easy classification problem. In fact, the number of instances and the six possible classes do not give much instances for training. A part of that, there are twenty-nine nominal , three integer and six continuous attributes.

- **Instances :** 798
- **Attributes :** 38
- **Classes :** 6

Prior to this work, there has been a feature selection and finally the number of attributes have been reduced to twenty-eight.

In the case of HNN all attributes are preprocessed by the same program. For other NN, nominal attributes have been preprocessed and expanded to eighty-eight real valued attributes.

1 ANN COMPARISON

This section contains the results of different performance tests. First of all, I selected the same error function for ANN of the HNN program. Once that is done, I compare the performance of all mentioned NN increasing number of hidden units.

1.1 ANN types

A brief description about ANN used in this work are :

Multilayer Perceptron : is a feedforward ANN that uses backpropagation technique for training. It usually uses a sigmoid, hyperbolic tangent or logistic activation function.

Radial Basis Function Network in that case is similar to the MLP, but it uses a radial basis function for activate the neurons.:

Heterogeneous Neural Network : are able to deal with nominal and ordinal features without preprocessing phase. In that case, it uses non-derivative algorithms to train the network like breeder genetic algorithm (BGA).

Extreme Learning Machine (ELM) is a feedforward ANN that uses random weights for input layer, also random bias for hidden units. Otherwise, output weights are computed with Moore–Penrose (MP) generalised inverse. This technique tends to need larger number of neurons.:

Levenberg Marquardt Neural Network (LM) is a feedforward ANN that uses a numerical solution for the minimising problem of training the network.:

1.2 Error function selection

HNN program let us select two different error functions, Mean Squared Error (MSE) and Cross-entropy (CE) function. For performing the final test with a minimum number of ANN combinations, I selected the best error function in average.

Data has been divided in $50\% - 25\% - 25\%$ for training, validation and test [1]. It is executed ten times for each NN and combination. Finally the test results are averaged. Figure 1 shows the difference of MSE and CE error functions with test accuracy values.

We can see that almost all the values are positive, it means that MSE performs better in most of the cases. On the other side, RBF is almost equal except two cases. I selected MSE function to perform all next comparisons.

1.3 ANN performance

Here the performance of all ANN is computed with the same dataset division $50\%-25\%-25\%$. The parameters of LM are 10 epoch while training and hyperbolic tangent sigmoid transfer function. For ELM a gaussian activation function, temperature of 0.001, random initial weights and fifty maxim iterations.

The best results are from HNN and MLP with twenty-four neurons 76.4889% and 76.2667% of accuracy respectively. LM and ELM are so deep in the figure 2, for that reason there is another figure 3.

1. Although I am using a test set, this values are only for model selection, but in no way it is intended to reflect a real error estimation.

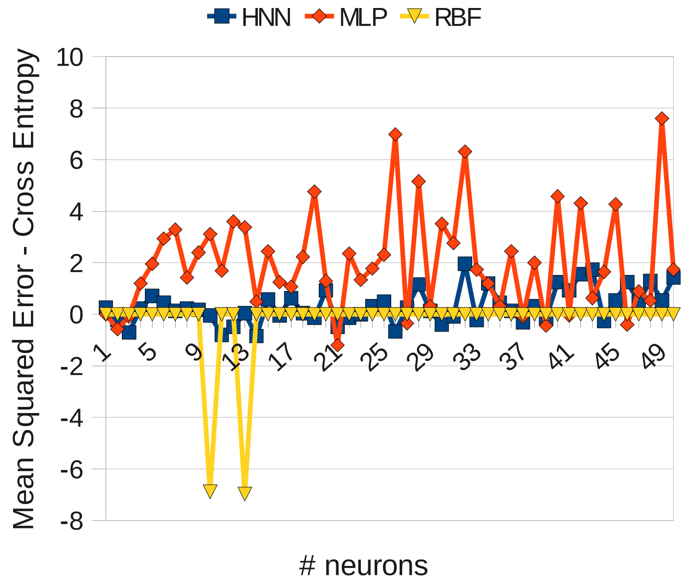

Fig. 1. Difference in accuracy of Mean Squared Error and Cross Entropy

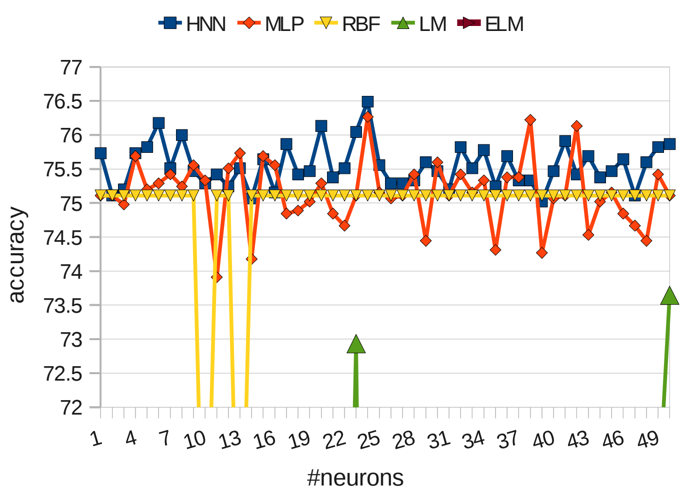

Fig. 2. Accuracy of different kinds of neural networks

2 Extreme Learning Machine parameters

This algorithm for single-hidden layer feedforward neural netwoks (SLFNs) provide good generalisation performance at extremely fast learning speed [1, Huang et al.]. To achieve a good parametrisation I created some new test to improve the performance for this dataset.

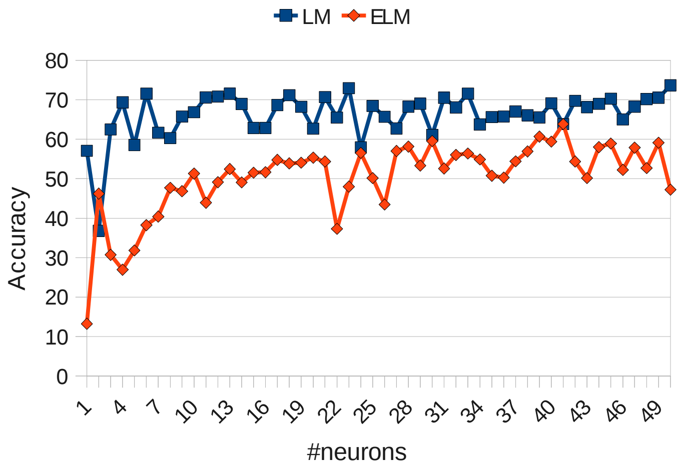

Fig. 3. Accuracy of Extreme learning machines and Levenberg Marquardt Neural Networks

In that case, I do not divided the data for test set. Instead of that I performed a 10-fold cross-validation because I am not interested in the out-sample error.

I tested different activation functions for the neurons (Gaussian, sinusoidal and sigmoid), and finally I selected the sigmoid function.

Then I selected the best number of neurons for that problem. It seems that does not need so many neurons to achieve good results. In fact, with more than twenty neurons, accuracy begins to decrease. On optimal situation there is an accuracy of 76.172% (see figure 4.

To test temperature values I fixed the number of neurons to 5 because there is no apparently reason to select more than that. The results on incrementing the temperature do not seem to look better. Quite the opposite, there is a range where this value remains stable $(0.0005-0.0085)$, but it starts to decay on average (see figure 5).

Max value for that parameter is the same as before (76.172%). This performance is the fourth of the best seen before. Only with a difference of 0.3169%. These are a very good results for a so fast algorithm.

3 Conclusion

We have compared the performance between various NN for classifying one dataset. Therefore, the result is not generalisable. Instead of that, it will be interesting to implement all

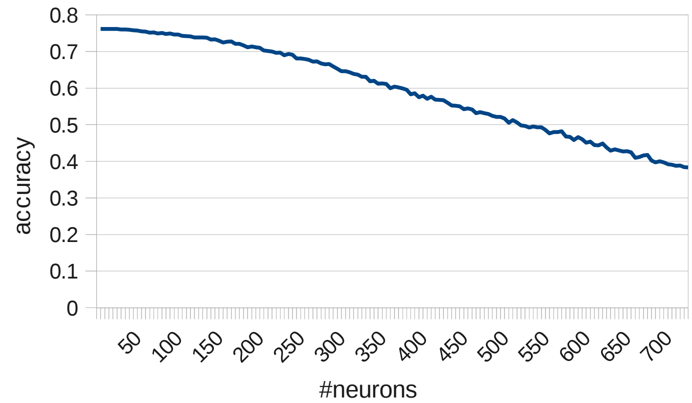

Fig. 4. Accuracy of Extreme learning machines and Levenberg Marquardt Neural Networks

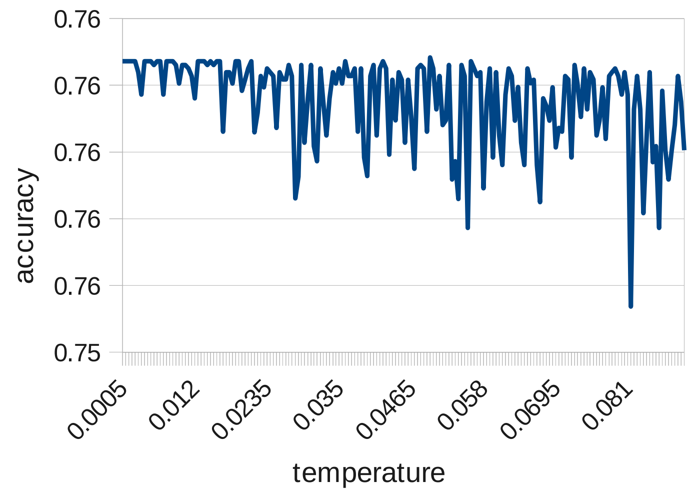

Fig. 5. Accuracy of Extreme learning machines and Levenberg Marquardt Neural Networks

the algorithms in a same software platform, and create a test with more than one dataset. Furthermore, it is needed to perform a good parametrisation for all algorithms, and it would be interesting to perform cross-validation.

References

[1] Huang, Guang-Bin, Qin-Yu Zhu, and Chee-Kheong Siew. ”Extreme learning machine: theory and applications.” Neurocomputing 70.1 (2006): 489-501.