How Many Neurons? A Genetic Programming Answer

Leonardo Trujillo, Yuliana Martínez and Patricia Melin
Instituto Tecnológico de Tijuana, Tijuana, BC, México
leonardo.trujillo.ttl@gmail.com,ysaraimr@gmail.com,pmelin@tectijuana.edu.mx

ABSTRACT

The goal of this paper is to derive predictive models that take as input a description of a problem and produce as output an estimate of the optimal number of hidden nodes in an Artificial Neural Network (ANN). We call such computational tools Direct Estimators of Neural Network Topology (DENNT), an use Genetic Programming (GP) to evolve them. The evolved DENNTs take as input statistical and complexity descriptors of the problem data, and output an estimate of the optimal number of hidden neurons.

Categories and Subject Descriptors

I.2.2 [**Artificial Intelligence**]: Automatic Programming—*program synthesis*

General Terms

Algorithms, Experimentation, Performance

Keywords

Artificial Neural Networks, Genetic Programming

1. INTRODUCTION

Designing an ANN for a specific task is not a trivial endeavor and is normally carried out by trail and error. Consider the common Multilayer Perceptron (MLP), for instance, a system designer must choose the type of activation functions, the manner in which neurons are connected, the learning rule, the number of hidden layers and the number of nodes in each layer. But given a problem, it is difficult to infer how the ANN ought be constructed. In fact, even if we only focus on a MLP with a single hidden layer, with all other characteristics fixed, it is not easy to answer the following: How many hidden neurons are necessary to achieve optimal performance? Researchers have developed a variety of methods to answer the above question. A coarse taxonomy reveals two groups: manual methods and automatic methods. The first group is the most common, based on informal trial and error. Automatic methods attempt to relieve the system designer from having to specify the number of neurons himself; for instance, by using an exhaustive search. A subgroup are the iterative methods, which include learning-based approaches and meta-heuristic approaches. Learning-based approaches integrate an iterative optimization procedure with traditional learning algorithms, to progressively add or remove nodes [3, 1]. Other approaches use a meta-heuristic search, such as evolutionary computation [7, 4].

The above approaches have achieved good results, but are also limited in some ways. For example, exhaustive search incurs a high computational cost. On the other hand, for iterative methods a system designer needs to add, or develop, another layer of computational machinery to the system. Moreover, iterative methods can be slow. or might require extensive experimental tunning. We believe that an *ideal* method should be simple to implement, use, and reproduce. When confronted with a problem, such a method would take the data as input, or a description of the data, and produce as output an accurate estimation of the optimal number of neurons. We call such an approach a *Direct Estimator of Neural Network Topology* (DENNT).

GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

2. PROPOSAL

The proposed approach is to use GP to evolve DENNTs for a fully connected MLP with a single hidden layer, applied to real-valued data classification.

2.1 Problem statement

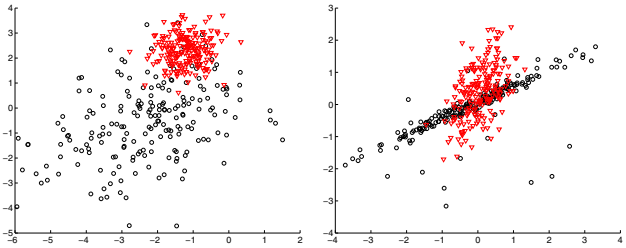
The above goal is formulated as a supervised learning and optimization problem. First, we build a set of synthetic 2-class multimodal problems, call this set P. Second, from each problem $p_i \in P$ we extract a vector of descriptive features β_i, with the goal of capturing problem specific characteristics. Third, we conduct an exhaustive search to determine the number of neurons α_i that achieves the best performance for each problem p_i. During the exhaustive search the number of neurons in the hidden layer is varied within the range $[1, 30]$. Afterwards, we have a set of problems P, where each $p_i \in P$ is described by a descriptive vector β_i, as well as the number of neurons α_i with which the MLP achieved the best results. Therefore, the problem we pose is that of finding an optimal DENNT K^o, such that

$$K^o = \arg\min_K \{Err[K(\beta_i), \alpha_i]\} \ \forall p_i \in P \quad (1)$$

where $Err[,]$ is the the root-mean-square error (RMSE).

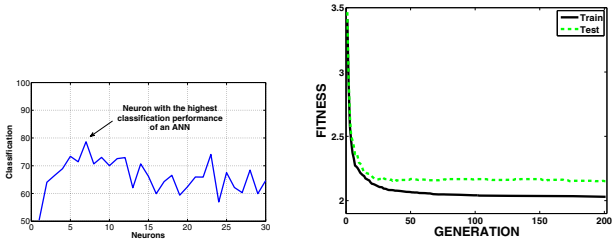
3. GP SYSTEM AND SYNTHETIC DATA

The terminal set T contains descriptive measures that can be extracted from the problem data, which are concatenated to construct vector β_i. We use two types of measures, statistical and complexity. The statistical measures are a subset of



(a) 3 Neurons (b) 11 Neurons

Figure 1: Scatter plots of two different problems.



(a) Signature (b) Evolution

Figure 2: (a) Performance signatures for the ANNs. (b) Median of the best training fitness and test fitness.

those proposed in [5]; these are: (1) the geometric mean ratio of the pooled standard deviations to standard deviations of the individual population (SD); (2) the average absolute correlation coefficients between two features (CORR) and its squared value (CORR2); (3) the average skewness of features (SKEW); (4) the average kurtosis of features (KURT); (5) the average entropy of features (HX) and its squared value (HX2).

The complexity measures directly consider the geometry of the distribution of data samples over the feature space. These are a subset of those proposed in [2], they are: (1) Fisher's discriminant ratio (FD); (2) volume of overlap region (VOR); (3) feature efficiency (FE); (4) class distance ratio (CDR); (5) classification error of a 1-NN classifier (E1NN). The function set F contains common primitive functions used for symbolic regression, $F = \left\{+, -, *, /, \sqrt{}, sin, cos, log, x^y\right\}$.

Fitness is posed as a cost function, using the RMSE computed on a set of n training samples, given by

$$f(K) = \sqrt{\frac{\sum_{i=1}^{n}((\lfloor K(\beta_i) \rfloor) - \alpha_i)^2}{n}}, \qquad (2)$$

where α_i is number of neurons of problem p_i.

To compute fitness, we randomly generate 1000 two-class classification problems using Gaussian mixture models, using a similar scheme to the one followed in [6], see Figure 1. To classify each problem we use a fully connected feedforward MLP with one hidden layer, hyperbolic tangent sigmoid transfer functions, and trained using backpropagation with adaptive learning rate and momentum. We test different network topologies in search of the one that achieves the best performance by varying the number of hidden neurons within the range of $[1, 30]$. In each case, at a different number of neurons, we determine the performance of the ANN classifier using 3-fold cross-validation, and perform 30 independent runs, a total of 90 performance estimates. For each topology we take the median performance as the representative value. This is depicted in Figure 2(a), where the classification accuracy achieved at each number of neurons is plotted. These graphs are performance signatures of the ANNs. To determine the optimal number of neurons for each problem we take the maximum value of this plot.

Table 1: GP parameters.

Parameter	Description
Population size	500 individuals
Generations	200 generations
Initialization	Ramped Half-and-Half, 10 levels max. depth
Operator probabilities	Crossover $p_c = 0.8$, Mutation $p_\mu = 0.2$
Bloat control	Dynamic depth
Maximum dynamic depth	11 levels
Hard maximum depth	30 levels
Selection	Lexicographic parsimony tournament
Survival	Keep best elitism
Runs	30

4. RESULTS AND CONCLUSIONS

Table 1 contains the parameters of the GP, a Koza-style GP with the dynamic depths method and lexicographic parsimony pressure for bloat control. The GP was executed 30 times, thus the results are statistics computed over all runs. In each run, 70% of the problems are used for training and the rest for testing. Figure 2(b) shows the evolution of the best individual fitness, and the fitness of the best individual computed with the test set, showing median values. The training process achieves similar training and testing fitness. Moreover, both reach low values, which corresponds with a low predictive error. The best median predictive error for the training set of problems is 2.03, while for the test set it is 2.15. The performance of the evolved DENNTs is encouraging, it suggest that is possible to predict the optimum number of neurons using only a description of the data. While the prediction is not yet perfect, we believe that the proposal can be extended, and refined, to derive more accurate predictive models. Such predictors could allow an autonomous system to configure its internal problem solving methods based on a description of the problem it needs to solve.

Acknowledgements

Second author supported by scholarship 298654, Consejo Nacional de Ciencia y Tecnología (CONACYT), México.

References

[1] M. Frean. The upstart algorithm: a method for constructing and training feedforward neural networks. *Neural Comput.*, 2:198–209, April 1990.

[2] T. K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:289–300, March 2002.

[3] Y. LeCun, J. S. Denker, and S. A. Solla. Optimal brain damage. In *NIPS'89*, pages 598–605, 1989.

[4] P. Melin and O. Castillo. *Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing: An Evolutionary Approach for Neural Networks and Fuzzy Systems (Studies in Fuzziness and Soft Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[5] S. Y. Sohn. Meta analysis of classification algorithms for pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21:1137–1144, November 1999.

[6] L. Trujillo, Y. Martínez, and P. Melin. Estimating classifier performance with genetic programming. In S. Silva et al., editor, *Proceedings of the 14th European Conference on Genetic Programming, EuroGP 2011*, volume 6621 of *LNCS*, pages 275–286, Turin, Italy, 2011. Springer Verlag.

[7] X. Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999.