Using genetic algorithms to optimize multi-layer neural networks for the prediction of Diabetes Mellitus among the Pima Indians.

Kiarie Ndegwa
The Australian National University,
College of Computer Science and Information Technology(CSIT)
u4742829@anu.edu.au

May 22, 2016

Abstract

This project is concerned with generating a binary classifier based on a small data set containing 786 patients each with 8 features. This classifier is used to determine whether or a patient is susceptible to Diabetes Mellitus. These features are comprised of 8 medical readings taken throughout the years on patients from the Pima Indian community in America. This report uses a genetic algorithm to alter the topology of 2 multi-layer perceptrons(MLP) designed to correctly classify Diabetes Mellitus given these features. The first architecture is a cascade network comprised of 2 hidden layers each of which contain 4 and 2 neurons respectively. This architecture correctly achieved an error accuracy of 0.21 using back propagation against the above data set. Whilst the second architecture comprised of 4 and 3 neurons in its respective hidden layers, and achieved an error accuracy of 0.23. This report takes these same architectures and uses a genetic algorithm to optimize their respective weights. Consequently the cascade networks accuracy increases to an error of 0.4765, whilst the standard MLP algorithms accuracy error drops to 0.1818. A drop small but significant drop of of 0.0482 making the classifiers overall accuracy given the data set ∼81.2 %

keywords: Genetic algorithms, Multi-layer perceptron, Cascade Neural Network, PIMA Indians, Diabetes Mellitus

1 Design of the Evolutionary Computation Method.

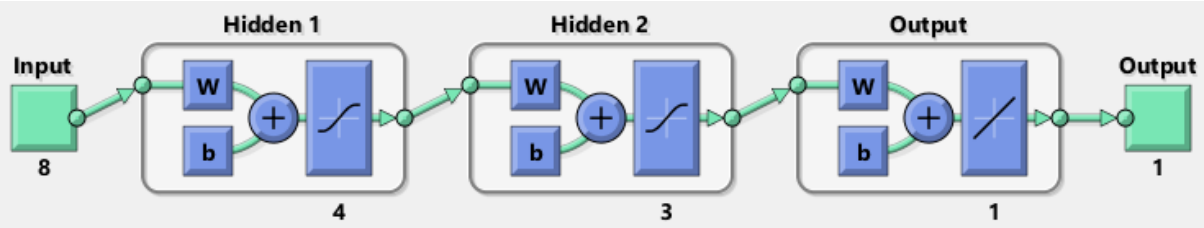
Given the limited features space of the Pima data set [1], using a genetic algorithm to alter an already limited feature space wouldn't allow for the proper exploration of optimization using the Genetic algorithm technique.

Genetic Algorithms have the added advantage of solving for the global minima given bound spaces with multiple local minima. Whereas backpropagation though more robust

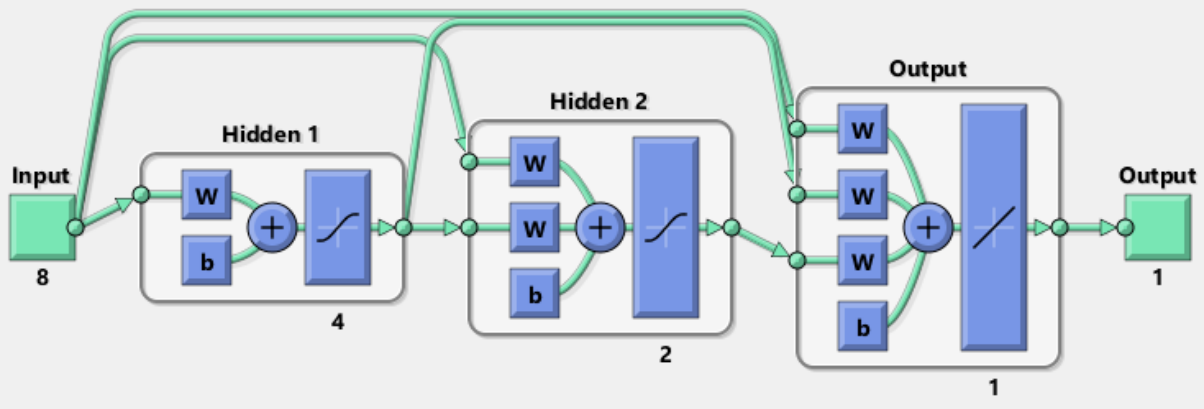
and computationally more efficient could at times fall into a local minima and limit the potential accuracy of the network. For this reason this report wishes to explore the effectiveness of genetic algorithms vs standard backpropagation with regards to learning effective weights and correctly classifying data.

Since we have data from the prior experiment detailing 2 different neural network topologies with optimal numbers of neurons in each layer, for the purpose of this report we will calculate the weights of the same topologies using a genetic algorithm. This weights will then be used to make new networks whose accuracy will be tested against test data set.

The architectures of the optimized topologies from the previous report are demonstrated in the schematics below.



Feed forward neural network optimized from previous task.



Cascade Neural Network generated and optimized in previous report

The first approach will take the multi-layer perceptron designed in the prior assignment, and try and optimize its topology. The second part of this report will be concerned with optimizing the weights of the cascade forward network.

2 Learning neural weight using of Genetic algorithms.

2.1 Over arching weight optimization algorithm using the genetic algorithm

The algorithm used to learn appropriate weights for the given networks is shown below:

1. Divide the data into a training and test data sets.
2. Run the GA feature selection algorithm on the training data set to produce weights.
3. Make a MLP using these weights.
4. Calculate the fitness function based on the mean squared error of the predicted values of this MLP with the learnt weights vs the target values of the training set.
5. Check stopping criteria

2.2 Genetic Algorithm implemented by Matlab code base.

The genetic algorithm used through this report, is that inbuilt into the Matlab code base. The algorithm begins by creating a random initial population. The algorithm then creates a sequence of new populations. At each step, the algorithm uses the individuals in the current generation to create the next population. To create the new population, the algorithm performs the following steps:

1. Scores each member of the current population by computing its fitness value.
2. Scales the raw fitness scores to convert them into a more usable range of values.
3. Selects members, called parents, based on their fitness.
4. Some of the individuals in the current population that have lower fitness are chosen as elite. These elite individuals are passed to the next population.
5. Produces children from the parents. Children are produced either by making random changes to a single parentmutationor by combining the vector entries of a pair of parentscrossover.
6. Replaces the current population with the children to form the next generation.
7. The algorithm stops when one of the stopping criteria is met.

2.3 Fitness function

The fitness function for the purposes of this report is generated by setting the learnt weights of a new neural net, and configuring it to classify test data. Accuracy of this network is then compared with target values and used to calculate the mean squared error (MSE) of the network. This MSE is used to guide the genetic algorithm.

2.4 Problem of over fitting

For the purpose of this experiment in order to see the strength of the generality properties of the GA. The data set is split into two subsets, i.e. 1 fold validation is used. In order to test the generality of this approach, different k fold test and training ratios are compared against results acquired by a network trained by back propagation.

The second part of this experiment is concerned with using 10 fold validation to get the actual classification accuracy of the tested networks.

2.5 Stopping criteria

The stopping criteria picked in this report is comprised of checking the minimum change in the generated best fitness function value. For the purposes of computational time, this is set to an inter-generational change of 0.001 in the Mean Squared Error.

This range was picked as it was found that optimizing neural networks with more weights resulted in longer training times that ranged anywhere between 1 hr to 2 hrs given an i7 Intel processor with 8 GB ram.

3 Results and Discussion.

As mention before for the purposes of classification accuracy, it was decided that in order to demonstrate the generalization efficiency of GAs that the 1-fold ratios influence on accuracy given the 2 networks be tested against results generated by the back propagation train neural nets of the previous report.

This resulted in the table below:

K-fold partition ratio	Multi-layer perceptron Error Accuracy	Cascade Neural Network Error Accuracy
50:50 (Test: 384, Train 384)	0.2708	0.3802
40:60 (Test: 304 Train 461)	0.2280	0.5277
30:70 (Test: 230, Train 538)	0.2870	0.5870
20:80 (Test: 154, Train 614)	0.2468	0.3636
10:90 (Test: 77, Train 691)	0.1818	0.4026

As can be seen in the table above the generalization ability of the genetic algorithm is demonstrated by its ability to learn generalized weights in the network rather quickly given a small training set, comprised of 60% of the available data set. Achieving a classification error of 0.2287, an accuracy higher than that achieved on the 2 layer MLP, which made use of the back propagation algorithm that made use of 10-fold validation. This is however found to be a local minima, and the genetic algorithm manages to find an even lower error minima when trained with more data, in the process achieving an error accuracy of 0.1668.

The feed forward network was comprised of considerably more weights than the simple multi-layer network. For this reason its training times using the genetic algorithm were very drawn out. This highlighted the major down side of GAs. Despite their flexibility with regards to search spaces, they require considerably more computational time than the standard back propagation algorithm that makes use of stochastic gradient descent.

Nevertheless, the cascade feed forward network as shown in the table above achieves its highest error accuracy of 0.3802. An error accuracy much lower than that of 0.23 achieved by its back propagated equivalent.

After carrying out 10 fold cross validation on the cascade network achieved an error accuracy of 0.4765 after a training time of 3.5 hrs.

For the MLP network under 10 fold cross validation, the error accuracy achieved was 0.1818 and took 1 hr to fully validate.

4 Conclusion and Future Work.

As was shown in the prior sections, Genetic algorithms though powerful optimization tools - are plagued by long training times and high computational cost that doesn't necessarily offset the benefits of replacing the backpropagation algorithm. This is demonstrated by the GA trained cascade neural network which had an overall decrease in estimation accuracy from 0.23 to 0.3802 despite a considerably long training time.

Conversely, should the correct set of variables be set up, the GA can sometimes result in excellent results. In this instance training the MLP topology, resulted in the highest classification accuracy yet achieved, given the Pima data set. This was an error accuracy of 0.1818 down from 0.23.

Given more time to further experiment with Matlabs GA package, the network topology in its entirety could have been designed from scratch using this technique. For this purpose it would have made sense to start this design process by having a single neuron evolve into a multi-layer perceptron.

Further usage of the genetic algorithm would then have involved training the weights of these resultant multi-layer perceptrons. This would fully demonstrate the power of genetic algorithms ability to create functional integrated bio-inspired useful algorithms and possibly an even more accurate classifier.

5 References

[1] Smith W. J., Everhart J.E., Dickson W.C., Knowler W.C., and DrPH R.S Johannes M.D. Using the adap learning algorithm to forecast the onset of diabetes mellitus. *The National Institute of Diabetes Digestive and Kidney Diseases: Epidemology and Data Systems programe and the Diabetes and Arthritis section and the John Hopkins University School of Medicine.*, 1988.