Predictive Diabetes Diagnosis using Backpropagation, Genetic Training, and Decision Tree Optimization

Matthew Pisano

Joseph Terranova

Project Overview

Project Motivation

- The motivation of this project originated from curiosity about the nature of neural networks and the efficiency of the different methods used to train them.

Objective

- The objective of the project is to test the methods of backpropagation, genetic algorithms and decision trees and see the different properties of the neural networks they yielded.
- For this project, we chose a diabetes diagnosis training set.

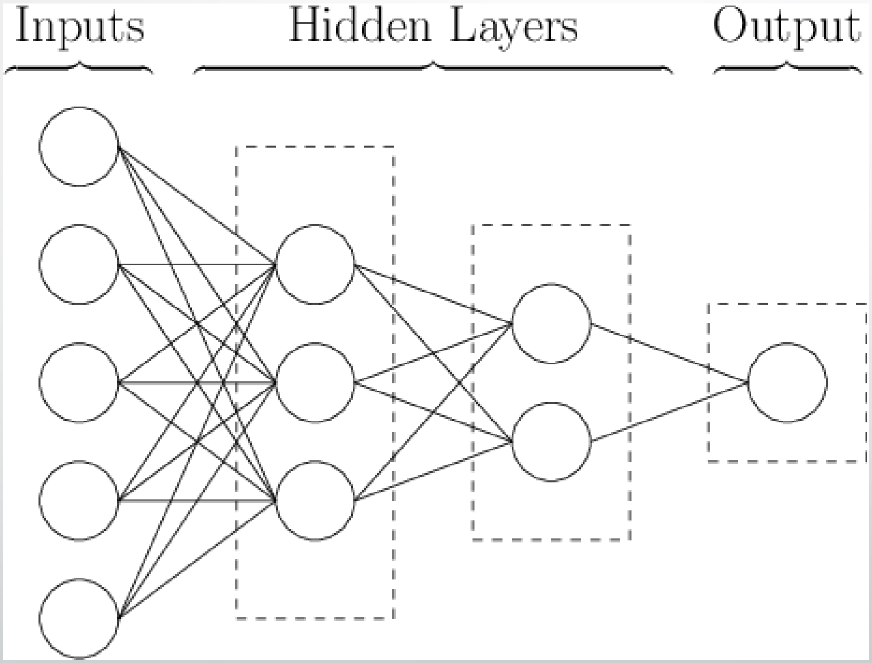
Background

- This project started out as only looking at neural networks that utilized decision trees.
- As the semester went on, we included neural networks that utilized backpropagation and those that used a genetic algorithm with each the backpropagation now optimized using a decision tree.

Background

- Neural networks produce results through the many interactions between its neurons. These neurons are organized in layers.
- The first layer is the input layer, where data is first given to the network and the last layer is the output layer, which takes input from other neurons and outputs the result of the network. Between these layers are zero or more hidden layers. They function the similarly to the output layer, taking input from the layer before it and producing an output.

Neural Network Diagram



History and Usage

- Since the introduction of the *Perceptron* (essentially a single output node) in 1958 and the many expansions upon its ideas since, many different methods and techniques have been developed to train Neural Networks. The purpose of this study is to investigate the different advantages and drawbacks of the three chosen techniques.
- Backpropagation is used for many applications including feature detection and handwriting recognition.
- Genetic algorithms are less computationally intensive than backpropagation but can take longer over their many generations. An advantage of genetic algorithms is their reliability to converge on local or global minima very quickly.
- Optimization with a decision tree adds extra efficiency to a neural network. By eliminating some input attributes, many calculations can be saved, especially with the matrix-heavy calculations of backpropagation.

Theory

Theory in Neural Networks

- The type of Neural Network used for this project is a feed-forward network. This means that information is inputted and it flows forward to the output nodes without any cycles.
- As information passes from each node to another, it is changed by that neuron.
- Each neuron takes input from every neuron in the previous layer and outputs to every neuron in the layer after it.
- Every input to a neuron is multiplied by a corresponding weight, summed together, has a threshold subtracted from it and then it is passed into an activation function.
- This is shown here with the sigmoid function $Y = \sigma[\sum_{i=1}^{n}(I_i W_i) - b]$

Theory in Backpropagation

- The main objective of backpropagation is for the network to directly learn from its errors.
- The difference between the desired output and the actual output is fed back into the network, starting at the output layer and ending at the first hidden layer.
- Regardless of the neuron, each weight is changed by $\Delta W_i = \delta \alpha Y_p$, where δ is the error gradient for that node.
- The error gradient is given by $\delta = \frac{d\sigma}{dx}[\sum_{i=1}^{n}(I_i W_i) - b]e$ for the output layer

and $\delta = \frac{d\sigma}{dx}[\sum_{i=1}^{n}(I_i W_i) - b] * \sum_{j=1}^{n_p}(\delta_j W_j)$ for the hidden layers.

Theory in Decision Trees

- Decision trees are based off the tree data structure.
- During execution, the tree analyzes the attributes of each sample point to predict what type of class a given patient falls into. Either diabetes or no diabetes in the case of this project.
- After the tree is built for execution, it can also be used to show which attributes have the least impact on the diagnosis of the patient. The lower down on the tree an attribute is, the less influential it is.
- We can use this to eliminate attributes from the neural network by only giving it those highest in the tree.

Building the Decision tree

- During the building of a tree, attributes are ordered by their Entropy and Information Gain.
- Entropy measures how well an attribute or the whole set predicts the class of the sample and the Information Gain of an attribute measures how much lower the entropy of an attribute is when compared to the entire set.
- As the tree is built, the highest info gain attributes are added as nodes, splitting the data set based on a pre-set threshold. That attribute is then eliminated from evaluation.
- The building stops once there are no more attributes left to split the data set on.

Decision Tree Calculations

- To chose what attribute is best to split on we must first get the entropy of the data set within the current node and the entropy of the available attributes to find the attribute with the most information gain
- Entropy for current data set is: entropy(D)= $-\sum_{j=1}^{|c|} \Pr(c_j) log_2 \Pr(c_j)$
- $\Pr(c_j)$ is the probability of the class c_j in the data set D
- Entropy for an attribute is: $entropy_{A_i}$(D) = $\sum_{j=1}^{v} \frac{|D_j|}{|D|}$ x entropy(D_j)
- And the information gain is entropy(D) - $entropy_{A_i}$(D)

Classifying Using Decision Tree

- During testing of a decision tree, individual patients and their attribute data is set to follow the path of the decision tree based on the thresholds and the individuals' attributes, HDL, BMI etc.
- Each individual is classified by the leaf node they reach at the bottom of the tree

Theory in Genetic Algorithms

- For making a genetic algorithm work with a neural network, many networks must compete against each other for the ability to pass their genomes onto the next generation.
- For this implementation, the genome was composed of the many weights within each network, with each gene representing the set of all weights (one from each node in layer *i*) relating to the connections of one node in layer *i* – *1*
- The fitness of each member network is measured by the inverse of its loss.

Implementation

Core Neural Network Implementation

- This project involved two implementations of a neural network over time.
- Both bodies of code were composed from scratch
- The first iteration was a classic graph data structure with the Neuron class serving as nodes and the NeuralNetwork class serving as the graph. The weights were stored in arrays.
- The second, more complex, implementation involved the NeuralNetwork class with the neurons being represented by matrices of weights and thresholds. This class used the NumPy library extensively for its very quick matrix multiplication abilities.

Training and Utilities

- In addition to the core neural network classes, the project also utilized several utility and helper classes.
- The Trainer class is an abstract class that is overridden by two sub classes, Backpropagator and Genetic. The neural network takes one of these two classes as a parameter and trains the neural network with the Backpropagation or Genetic algorithms, respectively.
- The DecisionTree class implements the decision tree optimization. It is executed before the neural network is trained and is used to cut out the less useful attributes in the training set. This eliminates some extra work the network would have to do otherwise.
- The Utils class contains many utility functions, from the sigmoid function to the plot function, which shows the results of the network over time.

Class Diagram

- Class Diagram

Results and Analysis

The Data Set

- The data set used for this project is from the Vanderbilt University Department of Biostatics and contains the testing parameters and results of 390 African American participants from central Virginia.
- The parameters of the dataset are comprised of the patient’s cholesterol, blood glucose levels, high-density lipoprotein (HDL) or ”good” cholesterol, the ratio of HDL cholesterol to total cholesterol, age, gender, height, weight, body mass index (BMI), systolic blood pressure, and diastolic blood pressure. The entries in the dataset are categorized into two groups, those that were diagnosed with diabetes and those who were not.
- Data Set

Preparing The Data Set

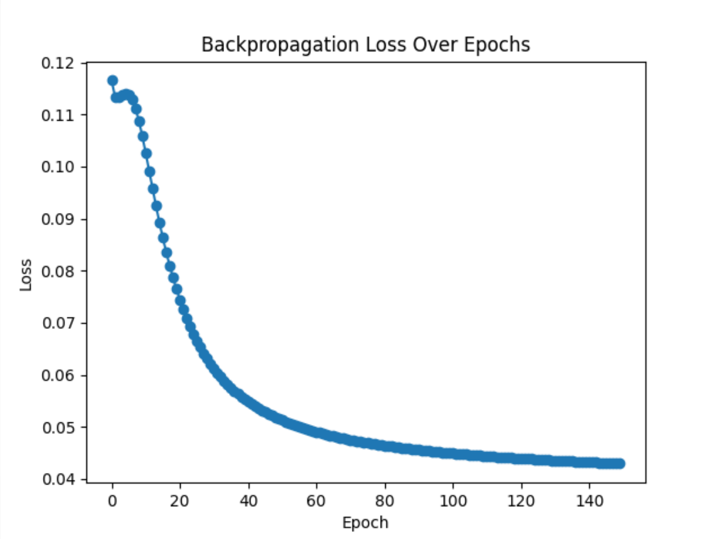
- To make the network as efficient as possible, some modification has to be made to the data set.
- Modifying Attributes – For the attributes that had text values, the gender attribute for example, the value of each attribute was assigned an integer, female being one and male being zero for that same example.
- Normalization – Preformed at execution time, each attribute had to be scaled between zero and one. By dividing by the largest value of each attribute, the attributes were properly scaled without any data loss.
- Prepared Data Set

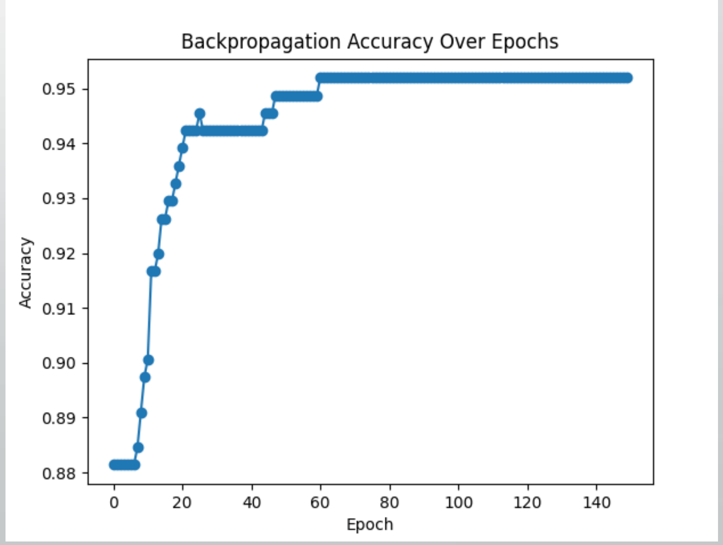
Training

- For training, each one of the three networks was given the same 80% proportion of the original data set and trained for 150 epochs and three independent trials. The loss function for each network is given by the sum of squared errors for every sample.
- The shape of the neural network we used was one with eleven input nodes before decision tree optimization and two output nodes, one for the network's confidence in a negative diagnosis and one for the confidence for a positive diagnosis.

Backpropagation Results

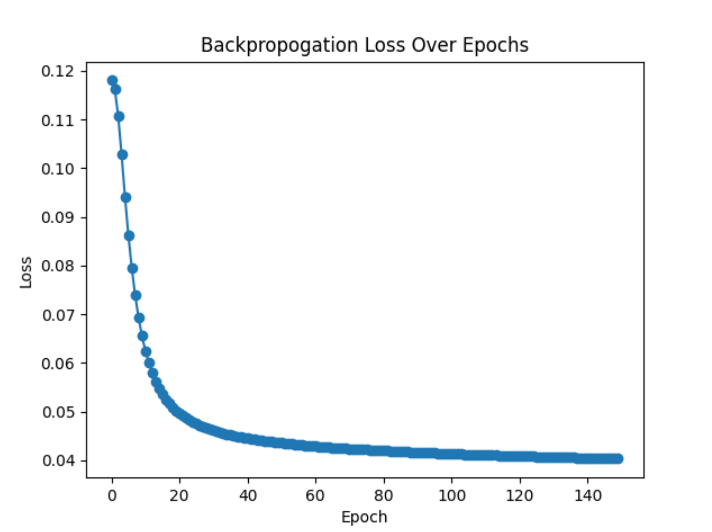
- For the backpropagation network, a learning rate (α) of 0.18 was used.
- Across three trials, training over the data set for the 150 epochs, the network trained after 0.82 seconds on average. The average loss for the testing set was 0.125, correctly predicting 85.9% of diagnoses in the testing set and 95.1% withing the training set, as shown below. Its overall score was 0.96.
- The training loss curve for backpropagation follows closely to a curve of 1/x . This signifies a good loss curve with minimal overfitting. The loss declines sharply, and the accuracy rises rapidly over successive epochs of training.

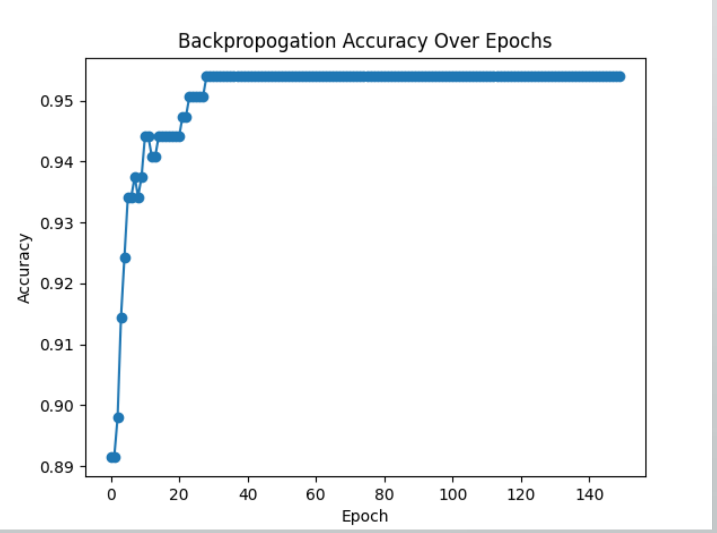




Decision Tree Results

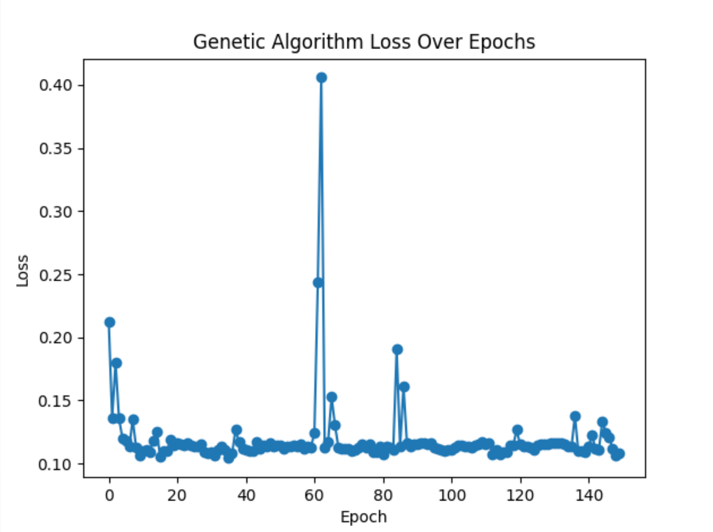
- For the backpropagation network with the decision tree optimization, a learning rate (α) of 0.18 was also used.
- The decision tree was able to eliminate the attributes of weight as least impactful and glucose levels as second to least impactful. This brought the total number of inputs down to nine input nodes.
- Across three trials, training over the data set for the 150 epochs, the network trained after 0.78 seconds on average. The average loss for the testing set was 0.125, correctly predicting 84.2% of diagnoses in the testing set and 95.0% withing the training set, as shown below. Its overall score was 1.05.
- The training loss curve for backpropagation follows closely to a curve of 1/x . Compared to raw backpropagation, this algorithm converges on an optimal solution much faster.

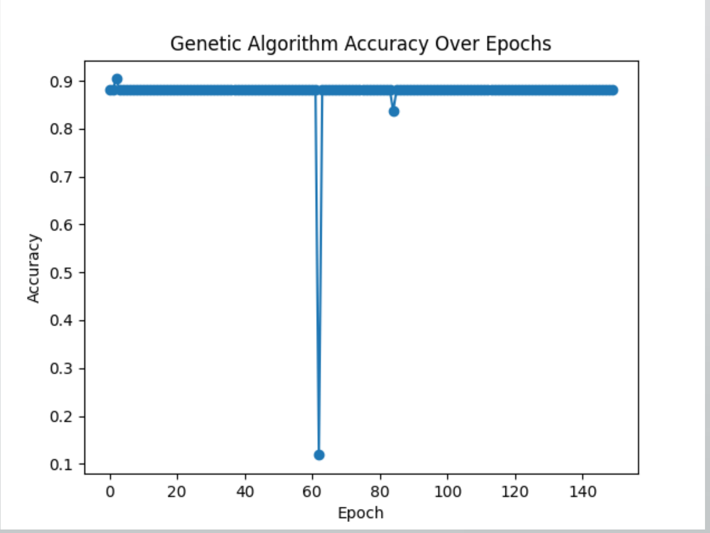




Genetic Algorithm Results

- For the genetic algorithm, a population size of 6, a crossover rate of 0.5, and a mutation rate of 0.01.
- Across three trials, the network took 4.95 seconds. The average loss for the testing set was 0.295, correctly identifying 70.4% of diagnoses in the testing set and 88.1% withing the training set, as shown below. Its overall score was 0.138.
- The training loss curve for the genetic algorithm shows a less pronounced curve, corresponding well with a high learning rate and a high change of over-fitting. The loss of the algorithm dropped sharply after the training started but leveled out higher than backpropagation did, resulting in a high accuracy, but it did not improve significantly over time.





Conclusions

- Across the three trials for the three different algorithms, backpropagation with the decision tree optimization seems to be the best fit for this particular data set. It completed quickly and with a high accuracy of 84.2% and score of 1.05 when predicting if a subject was diagnosed with diabetes given their medical attributes.
- A notable attribute of the genetic network is that it was able to converge on a solution much faster than wither of the two other implementations. This algorithm settles into a local minimum solution after five epochs, compared to the other two which settled at forty for raw backpropagation and twenty for optimized backpropagation. This result also deserves some credit for its utility.
- The decision tree optimization slightly decreased the execution time of the network. More importantly, it converged on an optimal solution much faster than raw backpropagation.

Output of Best Algorithm

- Network Output

References

- [1] MD Benjamin Wedro. Cholesterol management, Jul 2020.
- [2] Avik Dutta. Crossover diagram, Jun 2019.
- [3] Frank E Harrell. Diabetes data set, Dec 2002.
- [4] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Ŕıo, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. Nature, 585(7825):357–362, September 2020.
- [5] George V Jose. Useful plots to diagnose your neural network, Oct 2019.
- [6] Markus V. S. Lima. Neural network diagram, May 2018.
- [7] Michael Negnevitsky. Artificial Intelligence: A Guide to Intelligent Systems. Pearson Education Limited, second edition, 2005.
- [8] Guido Van Rossum and Fred L. Drake. Python 3 Reference Manual. CreateSpace, Scotts Valley, CA, 2009.