Implementation and Analysis of KNN, Decision Trees and ANN for the given problem statement.

Rahul.M.Koushik- PES1201700121, Akshay Kadam PES1201802417, Yash Kumar PES1201700772

Abstract—***Low Birth weight (LBW) acts as an indicator of sickness in newborn babies. LBW is closely associated with infant mortality as well as various health outcomes later in life. Various studies show strong correlation between maternal health during pregnancy and the child's birth weight. Exploiting machine learning techniques to gain useful information from health indicators of pregnant women for early detection of potential LBW cases. The forecasting problem should be reformulated as a classification problem between LBW and NOT-LBW classes.***

I. Introduction

The given problem statement has 10 columns of data , the last one consisting of the result that can be derived from the remaining columns.Here we implement multiple approaches to predict the same by splitting the data into 2 parts - Testing and Training data respectively

II. Implementation

A)Data Cleaning

First, we have to clean the given data in order to apply any methods and then draw conclusions from them, so we first fill all the empty cells with

i) Mode
ii) Mean

We notice the replaced value was fairly vital in the output noticed.We then removed the redundant column('Education') as it was the same value for all the given rows.

This data had to be further Normalized. After Normalization, we then used the sklearn to split the data into 2 parts - Test data and Train data.

B)K- Nearest Neighbors

kNN is a case-based learning method, which keeps all the training data for classification. Being a lazy learning method prohibits it in many applications such as dynamic web mining for a large repository. One way to improve its efficiency is to find some representatives to represent the whole training data for classification, viz. building an inductive learning model from the training dataset and using this model (representatives) for classification. There are many existing algorithms such as decision trees or neural networks initially designed to build such a model. One of the evaluation standards for different algorithms is their performance. As kNN is a simple but effective method for classification and it is convincing as one of the most effective methods on Reuters corpus of newswire stories in text categorization, it motivates us to build a model for kNN to improve its efficiency whilst preserving its classification accuracy as well. The training dataset including 100 data points with 8 classes is distributed in 8-dimensional data space.If we use Euclidean distance as our similarity measure, it is clear that many data points with the same class label are close to each other according to distance measure in many local areas. If we take these representatives as a model to represent the whole training dataset, it will significantly reduce the number of data points for classification, thereby to improve its efficiency. Obviously, if a new data point is covered by a representative it will be classified by the class label of this representative. If not, we calculate the distance of the new data point to each representative's nearest boundary and take each representative's nearest boundary as a data point, then classify the new data point in the spirit of KNN. In model construction process, each data point has its largest local neighbourhood which covers the maximal number of data points with the same class label. Based on these local neighbourhoods, the largest local neighbourhood (called largest global neighbourhood) can be obtained in each cycle. This largest global neighbourhood can be seen as a representative to represent all the data points covered by it. For data points not covered by any representatives, we repeat the above operation until all the data points have been covered by chosen representatives. We checked the accuracy for the model by choosing different values of K.We find out that the best value of K for the given dataset is 11. Further, since the dataset is relatively small consisting of only 100 rows, changing the test to train ratio severely affects the results as well.

We have implemented the Basic version of a KNN as well. We find the Average accuracy to be about 80% and the Maximum being 90.1% if the train and test data split is 80:21 respecively.

C) Decision Trees

A decision tree is a flowchart-like tree structure, where each internal node represents a test on an attribute, each branch represents an outcome of the test, class label is represented by each leaf node (or terminal node) . Given a tuple X, the attribute values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node which holds the class prediction for the tuple. It is easy to convert decision trees into classification rules. Decision tree learning uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. It is one of the predictive modelling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a finite set of values are called classification trees, in this tree structure, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.

Decision tree can be constructed relatively fast compared to other methods of classification. SQL statements can be constructed from tree that can be used to access databases efficiently.

In order to define information gain precisely, we begin by defining a measure commonly used in information theory, called **entropy** that characterizes the (im)purity of an arbitrary collection of examples.
For a binary classification problem-If all examples are positive or all are negative then entropy will be **zero** i.e, low.

This report studied decision tree algorithm. . The efficiency of decision tree algorithms can be analyzed based on their accuracy and time taken to derive the tree. **We find the Average accuracy to be about 73.33% and the Maximum being 87% if the train and test data split is 80:20 respecively.**

D)Neural-Network

An Artificial Neuron Network (ANN), popularly known as Neural Network is a computational model based on the structure and functions of biological neural networks. It is like an artificial human nervous system for receiving, processing, and transmitting information in terms of Computer Science.

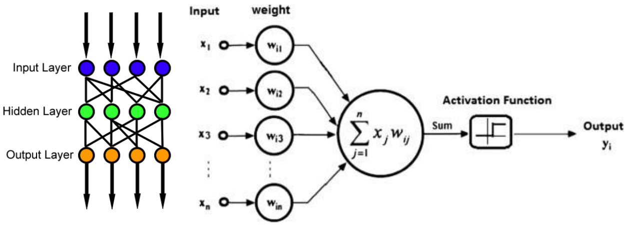
Basically, there are 3 different layers in a neural network :-

1. Input Layer (All the inputs are fed in the model through this layer)

2. Hidden Layers (There can be more than one hidden layers which are used for processing the inputs received from the input layers)

3. Output Layer (The data after processing is made available at the output layer)

Following is the manner in which these layers are laid



Input Layer

The Input layer communicates with the external environment that presents a pattern to the neural network. Its job is to deal with all the inputs only. This input gets transferred to the hidden layers which are explained below. The input layer should represent the condition for which we are training the neural network. Every input neuron should represent some independent variable that has an influence over the output of the neural network

We have used 8 input layers with their corresponding weights as redundant columns have been dropped.Also, on testing various activation function, we chose tanh to be the most accurate.

Hidden Layer

The hidden layer is the collection of neurons which has activation function applied on it and it is an intermediate layer found between the input layer and the output layer. Its job is to process the inputs obtained by its previous layer. So it is the layer which is responsible extracting the required features from the input data. Many researches has been made in evaluating the number of neurons in the hidden layer but still none of them was successful in finding the accurate result. Also there can be multiple hidden layers in a Neural Network. So you must be thinking that how many hidden layers have to be used for which kind of problem. Suppose that if we have a data which can be separated linearly, then there is no need to use hidden layer as the activation function can be implemented to input layer which can solve the problem. But in case of problems which deals with complex decisions, we can use 3 to 5 hidden layers based on the degree of complexity of the problem or the degree of accuracy required. That certainly not means that if we keep on increasing the number of layers, the neural network will give high accuracy! A stage comes when the accuracy becomes constant or falls if we add an extra layer! Also, we should also calculate the number of nuerons in each network. If the number of neurons are less as compared to the complexity of the problem data then there will be very few neurons in the hidden layers to adequately detect the signals in a complicated data set. If unnecessary more neurons are present in the network then Overfitting may occur. Several methods are used till now which do not provide the exact formula for calculating the number of hidden layer as well as number of neurons in each hidden layer.

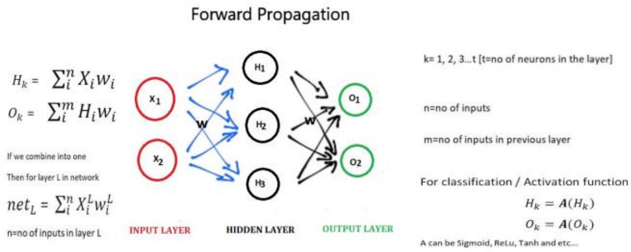
We have 10 nodes in the hidden layer with no bias and the activation function used here is sigmoid and then this is passed on to the output layer.

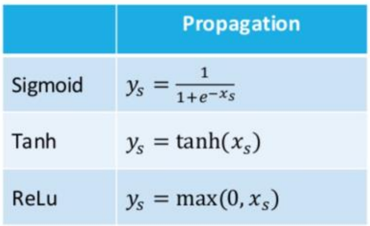
Output Layer

The output layer of the neural network collects and transmits the information accordingly in way it has been designed to give. The pattern presented by the output layer can be directly traced back to the input layer. The number of neurons in output layer should be directly related to the type of work that the neural network was performing. To determine the number of neurons in the output layer, first consider the intended use of the neural network.

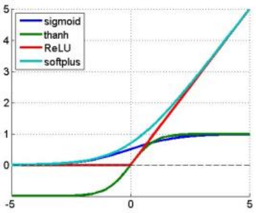
we have the input and we have some weights(parameters) we apply the dot product of these two vectors and produce the result (which would be a continuous value -infinity to + infinity).If we want to restrict the output values we use an Activation function.

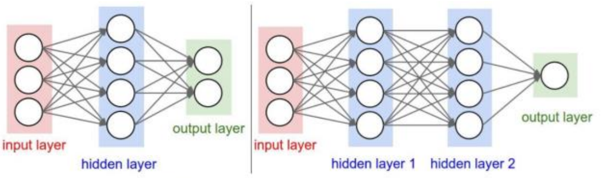
The activation function squashes the output value and produce a value within a rage (which is based on the type of activation function).


Forward Propagation
$H_k = \sum_i^n X_i w_i$
$O_k = \sum_i^m H_i w_i$
If we combine into one
Then for layer L in network
$net_L = \sum_i^n X_i^L w_i^L$
n=no of inputs in layer L
x_1
x_2
W
H_1
H_2
H_3
W
O_1
O_2
INPUT LAYER
HIDDEN LAYER
OUTPUT LAYER
k= 1, 2, 3...t [t=no of neurons in the layer]
n=no of inputs
m=no of inputs in previous layer
For classification / Activation function
$H_k = A(H_k)$
$O_k = A(O_k)$
A can be Sigmoid, ReLu, Tanh and etc...



	Propagation
Sigmoid	$y_s = \frac{1}{1+e^{-x_s}}$
Tanh	$y_s = \tanh(x_s)$
ReLu	$y_s = \max(0, x_s)$





Left: A 2-layer Neural Network (one hidden layer of 4 neurons (or units) and one output layer with 2 neurons), and three inputs. **Right:** A 3-layer neural network with three inputs, two hidden layers of 4 neurons each and one output layer. Notice that in both cases there are connections (synapses) between neurons across layers, but not within a layer.

There are 2 algorithms in Neural networks

1.Forward propagation.

2.Back propagation.

This is a simple process, we feed forward the inputs through each layer in the network , the outputs from the previous layer become the inputs to the next layer.(first we feed our data as the inputs)

The main goal of backpropagation is to update each of the weights in the network so that they cause the predicted output to be closer the target output, thereby minimizing the error for each output neuron and the network as a whole.So far we got the total error which is to be minimized.

Further, since the dataset is relatively small consisting of only 100 rows, changing the test to train ratio severely affects the results as well.

We find the Average accuracy to be about 89% and the Maximum being 92.1% if the train and test data split is 80:21 respecively.

E)Aggregation

The Above mentioned methods all give a certain accuracy ,however , the possibility to get a better classifier if we combine the output of each of those methods into a single output by assigning each of them a particular weight-age, is not viable as the ANN and KNN model far out-weigh the accuracy of the decision tree .This slightly decreases the accuracy of our model.

III.Conclusion

The aim of this paper has been to demonstrate that the technology for building decision trees , KNN and Neural Networks from examples is fairly robust. For the given problem statemntCurrent commercial systems are powerful tools that have achieved noteworthy successes. The groundwork has been done for advances that will permit such tools to deal even with noisy, incomplete data typical of advanced real-world applications. Work is continuing at several centers to improve the performance of the underlying algorithms.We would like to take the people responsible for this wonderful opportunity.