Binary Classification by SVM Based Tree Type Neural Networks

Submitted by: Ritvik Kapila, Gauri Gupta

- We have built a constructive algorithm for addressing the binary nonlinear classification problem by using a Tree-Structured Neural Network in Multi-Dimensional input space.

- In this approach, we first try to solve the classification problem by starting from a single neuron (small network). For correcting the misclassifications, we add neurons with suitable biases to our parent neuron. We thus, keep increasing the size and complexity of our network to gradually form an optimum classifier.

To construct our neural network, we first formulated a linear programming framework and found an equivalent optimization problem. We have implemented a recursive algorithm for the given optimization problem to build a tree structured neural network starting from a single neuron.

Objectives:

- According to the given algorithm, we expected the training accuracy to increase progressively with each new child layer and ultimately rise to 1.

- We also expected the testing accuracy to increase as we go down the tree deeper but fall after a peak is reached. This is when we start overfitting the data to achieve a 100% training accuracy.

Advantages of the Model:

- No hyperparameter tuning is required to train the model, unlike the training of neural networks, where we have to tweak several hyperparameters to obtain the optimum results. This reduces the training time of our model.

- 100% training accuracy can be obtained on the dataset, as the network can keep increasing in size and complexity until it classifies all the data-points correctly.

Since the previous project update, we have made several implementations and the major ones are:

1. Removed trivial solution:

   While implementing the linprog library for solving the optimize function, we were ending up with a trivial solution which led to all the weights of the neuron to 0. So, in order to obtain a non-trivial solution, we further added some conditions on the slack variables while solving the LPP.

2. Class Imbalance:

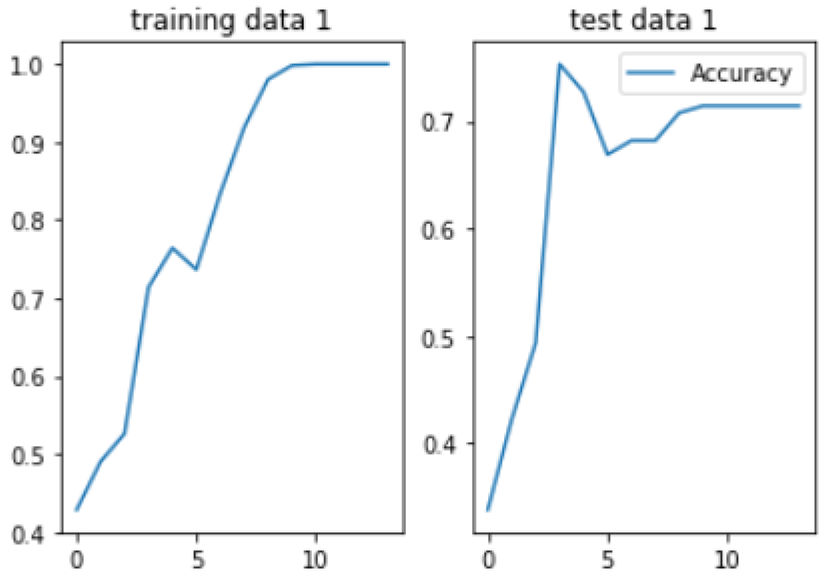
   The next challenge we faced was the problem of class imbalance, which occurred when we had a set of classes such that the samples in either of the correctly classified classes C1 or C2 were 0. This led to a recurring loop in our optimization problem and the network would start to grow in one of the directions indefinitely. To solve this problem, we used Twin Support Vector Machines [2] to solve for the network whenever we encountered the class imbalance. Using the Twin SVM classifier along with the previously built neuron structure, we tried to build the tree structure that tackled the problem in an efficient manner.

Results:

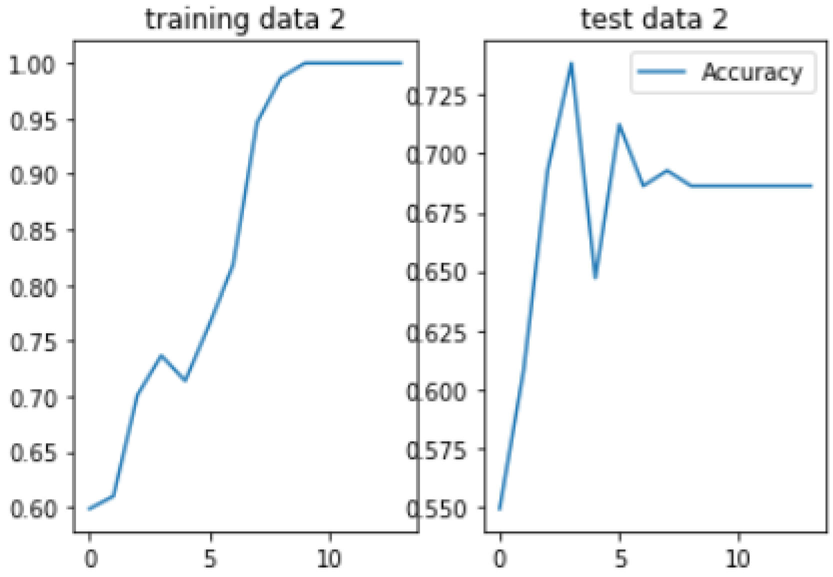
The following results obtained were in coherence with our expectations.

- We used the *UCI* dataset to train our model and performed 5-fold cross validation for finding the training and testing accuracies.
- We calculated the accuracies with respect to the number of layers of the model, which effectively is a measure of the size of the training dataset.

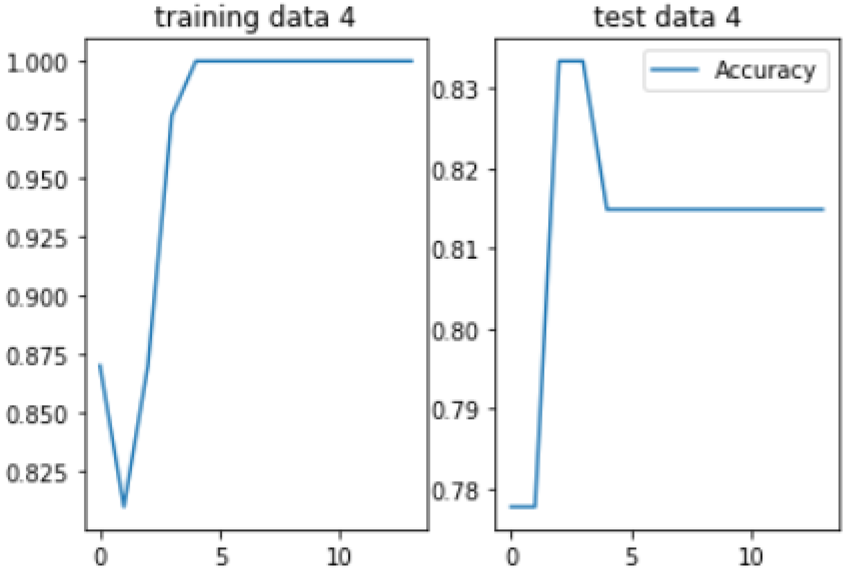
- As we had expected, the training accuracy rises to a 100% as can be seen in the figure below.



- The testing accuracy increases initially until a certain maximum value, and then starts decreasing when the model grows further. This is a clear indication of overfitting of the dataset.



- We stop the growth of the model when the network starts to overfit the data-points, thus achieving maximum test error using a bottom up approach.



We are grateful to Prof. Jayadeva for giving us this opportunity to work with him. His guidance, advice and educative discussions were a source of great motivation for us.

References:

[1] Dr, Jayadeva & Deb, Alok & Chandra, Suresh. (2002). Binary classification by SVM based tree type neural networks.

[2] Dr, Jayadeva & Khemchandani, Reshma & Chandra, Suresh. (2017). Twin Support Vector Machines.