EE 565 Machine Learning Project 6 Report

Mehrdad Ghyabi

***Abstract*—This is a summarized report of the project five of the machine learning course. First half of this report is dedicated to decision trees and their performance in classifying random data drawn from double moon distribution and the effect of the parameter *d* on the shape of the resulting tree. In the second half of this report, a multi-layer perceptron (MLP) model is trained using back-propagation algorithm, to classify the MNIST handwritten dataset.**

***Index Terms*— Decision trees, Tree depth, Leaf nodes, Multi-layer perceptron, MNIST dataset**

I. Introduction

This report is consisted of four sections. After the introduction in section I, the performance of decision trees in random data points drawn randomly from double moon distribution with different *d* parameters is investigated in section II. In part III a neural network with one hidden layer is trained and tested with the MNIST handwritten digits dataset. The report is concluded by conclusions in section IV.

II. Problem 1: The Decision Trees

In this part of the project, a toolbox has been used to train decision trees. The training data set is 3000 two-dimensional data points randomly drawn from double moon distribution.

A. Toolbox

To solve this problem, the "sklearn" library in Python environment was chosen. This decision tree classifier many attributes and options. The splitting criterion can be set either "gini" or "entropy", and the "gini" was used as default. The splitter can be "random" or "best", the latter was used here. There is an attribute to limit the depth of the resulting tree. Also, a minimum number of samples in each leaf node can be assigned to the model. Moreover, a maximum number of leaf nodes can be defined. There are a few other optional parameters which are out of the scope of this report.

B. Double moon with d=0.5

A training data set consisting of 3000 datapoints randomly drawn from a double moon distribution with parameters $r = 1$ and $w = 0.6$ and $d = 0.5$ was used to train a decision tree classifier. The shape of the resulting tree is presented in the Figure 1. As it was anticipated the resulting tree divided the two-dimensional space with one split accurately. This tree has two nodes and both of them are leaf nodes. The decision boundary and the training dataset are illustrated in the Figure 2.

C. Double moon with d=-0.5

A training data set consisting of 3000 datapoints randomly drawn from a double moon distribution with parameters $r = 1$ and $w = 0.6$ and $d = -0.5$ was used to train a decision tree classifier. The shape of the resulting tree is presented in the Figure 3. The decision boundary is more complex than before and it is because of the fact that in this case, the two half-moons of the training dataset are not located separately from each other. This tree has 12 nodes and seven of which are leaf nodes. The decision boundary and the training dataset are illustrated in the Figure 4.

III. Problem 2: MLP trained with MNIST dataset

In this part, an MLP with one hidden layer is trained using the MNIST handwritten dataset. The MNIST dataset consists of 5000 images if handwritten digits in 10 classes (0-9). Each image has 400 pixels (200×200), and the handwritten digit is centered in it, taking the entire image. A randomly selected set of 4500 (90%) images of the dataset is selected as training set and the remaining 500 (10%) images are used as the test set. Figure 5 shows 100 randomly selected images of the training dataset.

The hidden layer of the network has 25 neurons in it. To train the network a back-propagation scheme is used with 50 epochs. In each epoch the whole training set is used to update the weights. A sigmoid function is selected as activation of the first layer. The output layer has 10 neurons and as a result the output of the network is in hot encoded format. The weights are randomly initialized in the beginning of the training process.

The weight space of this network has the size of 25×400. The weights are shown in the form of 25, 20×20 images in the Figure 6.

After training the network, the accuracy of network in classifying both training and test set is measured. The training accuracy is 96.24% and the test accuracy is 92.80%.

In this implementation of back-propagation, the "fmincg" toolbox is used in MATLAB. "fmincg" is an optimization toolbox which works properly in this case. The input data for this tool box is the cost function at each trial point in the weight space in addition to the first derivative of the cost function with respect to each weight. This toolbox works based on gradient descent.

IV. Conclusions

Decision trees classified training datasets with 100% accuracy. The size of the tree is a function of complexity of the training data. The more splits needed to accurately classify data means the deeper the tree gets.

A neural network with one hidden layer of 10 neurons was able to classify both training and test set of 20×20 handwritten digits with a good accuracy. The training was done with 50 epochs.

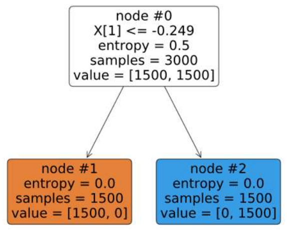

Fig. 1. Decision tree trained with d=0.5

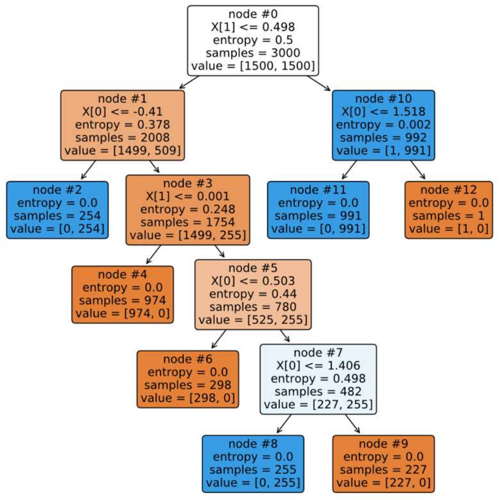

Fig. 3. Decision tree trained with d=-0.5

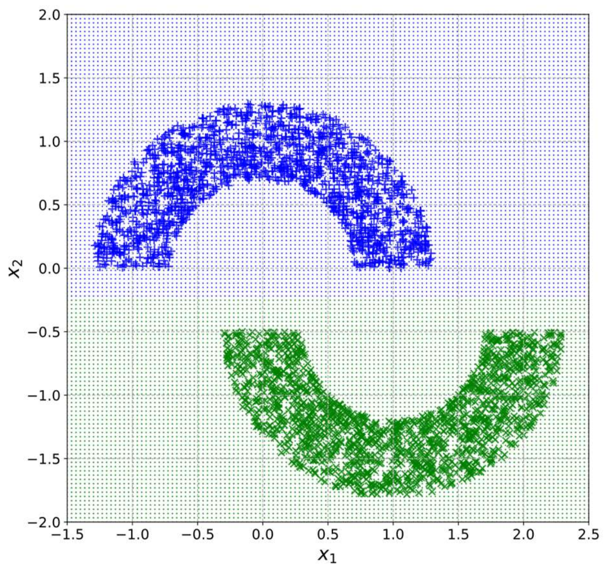

Fig. 2. Decision boundary of the tree trained with d=0.5

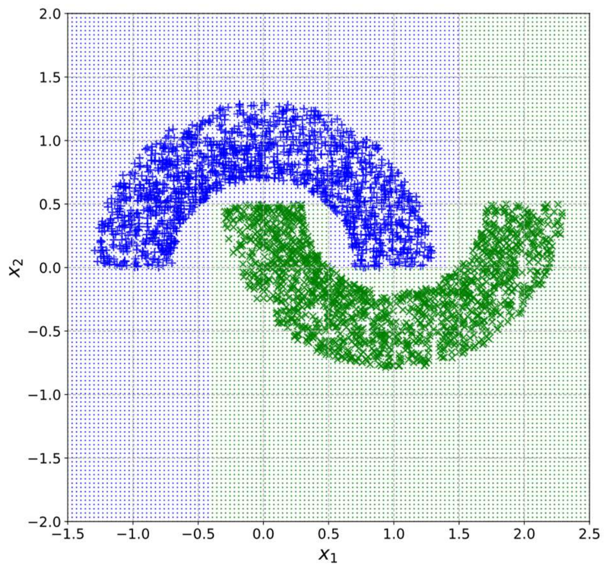

Fig. 4. Decision boundary of the tree trained with d=-0.5

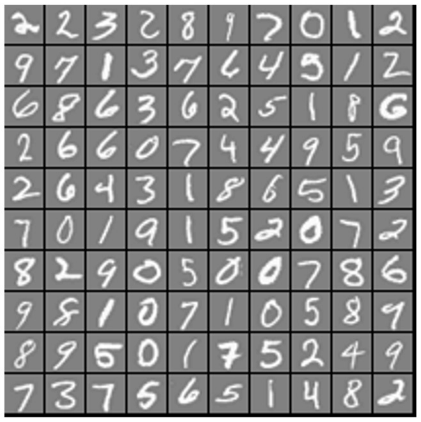

Fig. 5. A sample of training data

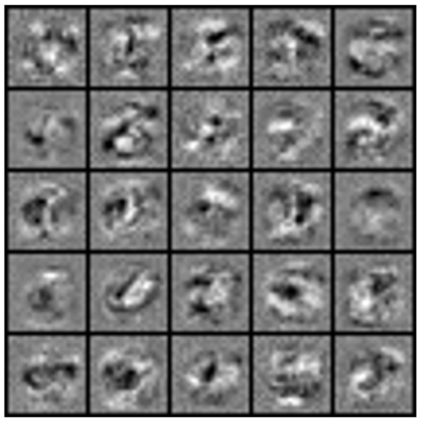

Fig. 6. Weight space visualization