Final Project for ECE 8527

Animesh Bala Ani
Department of Electrical and Computer Engineering, Temple University
animesh.ani@temple.edu

Introduction: In this project we were asked to make prediction on two datasets with two machine learning algorithms, one is from traditional machine learning approach and another is from deep learning approach. One of the provided datasets contains 2 features and another contains 5 features. To solve this problem, I chose Random Forest (RNF) algorithm for traditional machine learning approach considering its capability of achieving high accuracy on any kind of dataset, and Multilayer Perceptron (MLP) for deep learning approach considering its design simplicity. On the given unknown 5D eval dataset, the RNF achieved 36.48% error rate, which is the lowest error rate achieved from Python programming platform. Only better error rate is achieved on that dataset is 36.30% error, using Gaussian Mixture Model (GMM) on MATLAB platform. However, GMM achieved 37.13% error in Python platform, which is worse than the RNF approach. About the MLP approach, 39.94% error rate is achieved for that dataset.

Discussion on Datasets: The scatter plot of provided datasets are depicted below. For the 5D dataset, randomly 2 features are considered to obtain the scatter plot. However, the combination of different selections also provide similar overlaps, therefore presents no new information.

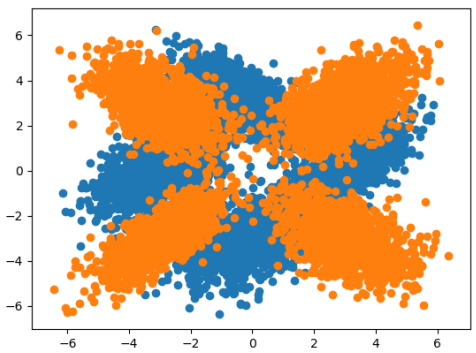

Figure 1: Scatter Plot of 2D dataset

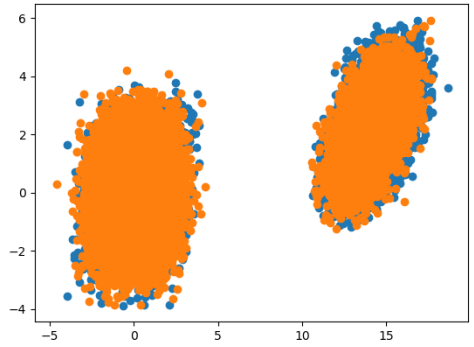

Figure 2: Scatter plot of 5D dataset.

It is obvious from the scatter plot that obtaining a decent amount of precision is easily possible for 2D dataset even after generalization, due to having small amount of overlap. However, this is highly unlikely for 5D dataset.

RNF Approach: To make prediction on both datasets, for traditional machine learning approach I used RNF algorithm.

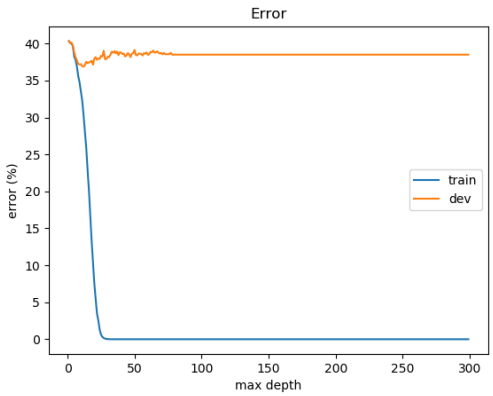

Figure 3: Error vs Complexity for RNF

Since RNF can expand its branches as much as possible and it can even achieve zero error putting all data points in different leaves, this approach is a good fit for highly overlapped dataset. Figure 3 depicts the error vs complexity (or maximum

depth of branches) curve on both training and evaluation dataset, which gives us the idea about how this algorithm performs. It seems by expanding branches, RNF can achieve zero error at around 25-30 branch depth for training dataset. However, the error rate starts increasing after 12 branch depth for the evaluation dataset. This proves that the generalization fails at 12 branch depth and the algorithm starts getting more and more overtrained. The performance on 2D dataset is also quite decent for RNF algorithm.

Deep Learning Approach: For deep learning approach, I used MLP for its design simplicity. I introduced two hidden layers for the MLP. To control reproducibility in CPU node, the random state is controlled with provided seeds. PyTorch is used for MLP development. With the developed MLP, a decent precision is easily achieved for the 2D dataset. However, for the 5D dataset, it performed quite badly. This is because it is very hard to find out proper combinations of nodes and layers, which requires a lot of trial and error procedure. While working with the MLP, it is noticed that, the batch size acts significantly to reduce error rate and 100 is the optimum size. Any deviation leads to some bad error rate, even 50% (which is equal to random guess). The random state also acts significantly. For the same layer-node combination, the MLP produces bad results for different provided seeds. This means the MLP easily gets stuck to local minima instead of moving near global minima. I think performing a smoothing approach may improve this situation. I used following layer combination to produce the MLP.

Input > Linear > ReLU > Linear > ReLU > Output

Results: Table 1 shows the achieved results from both algorithms for the provided datasets.

Table 1: Error rates for different experiments.

	2D Data Set		
Algorithm	**Train**	**Dev**	**Eval**
RNF	06.86%	08.90%	08.50%
MLP	08.07%	07.90%	08.40%
	5D Data Set		
RNF	30.27%	36.88%	36.48%
MLP	39.91%	39.93%	39.94%

The deviance between train, dev and eval dataset is considerably small, which means the algorithms are well generalized. Moreover, for 2D dataset all values are quite closer. Which means around 7% is the lowest error rate achievable for a generalized algorithm without overtraining. For 5D dataset, this value is around 36%. The performance of MLP came badly because it requires a lot of trials and error. However, RNF easily achieved good result by making generalization looking at dev dataset while training on train dataset avoiding overtraining.

Conclusion: For a small dataset of 100000 data points, RNF can be a good solution. It also takes decently small amount of time for training session. However, when number of features becomes larger and larger with a large amount of data points, deep learning approach might be an efficient solution. Although, to achieve a good result from deep learning approach, a lot of trial and error procedure requires to be followed.