Final Report

Ya Xiao
Department of Electrical and Computer Engineering, Temple University
tuf99262@temple.edu

Introduction: Machine learning and deep learning are two subsets of artificial intelligence. Machine learning involves with creation of algorithms that can modify itself without human intervention to produce desired output by improving itself through structured data. There are many machine learning algorithms such as k-Nearest Neighbor (KNN), Gaussian Naïve Bayes, Support Vector Machines (SVM), and Random Forest (RNF). Neural networks is a beautiful biologically-inspired programming paradigm which enables a computer to learn from observation data. It is a series of algorithms that try to recognize underlying relationships in a set of data through a process that mimics the way of the human brain operation. Neural networks and deep learning currently act a good performance to solve many problems such as the image recognition, speech recognition, and natural language processing. Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), and Recurrent Neural Networks (RNNs) are different deep learning algorithms.

In this assignment, we have two-classes train, dev and eval datasets with two dimensions and five dimensions features. However, the labels in 5-D eval sets are not included. We can only predict the labels. Then, we need to pick up one deep learning algorithm and one machine learning algorithm to deal with those datasets. We use Multilayer Perceptron (MLP), and Random Forest (RNF) from the Scikit-Learn and PyTorch libraries as the algorithms to train the datasets. By changing the different parameters in different algorithms, we have fair results of error rates based on different datasets.

Algorithm No. 1 Description: A multilayer perceptron (MLP) is a class of feedforward artificial neural network. It is a logistic regressor and consists of at least three layers of nodes which are an input layer, a hidden layer and an output layer. Since MLPs are connected, each node in one layer connects with a certain weight to every node in the following layer.

We use PyTorch v1.3.1 which is an open source machine leaning library based on the Torch library to extract the multilayer perceptron (MLP) algorithm. For the 2-D datasets, according to the train.py, we change the parameter NUM_ARGS to 2, NUM_EPOCHS to 4300 and BATCH_SIZE to 200. We also change the optimizer parameters to different values. The parameter LEARNING_RAGE is 0.0048. The BETAS is from 0.99 to 0.999. The EPS is 1e-8 and the WEIGHT_DECAY is 0.00009. In the model.py, we assign the parameter SEED1 as 5000. The DEF_NUM_FEATS is 2. The NUM__NODE is 50 and The NUM_CLASSES is 2. In run.sh, we change DL_NUM_FEATS to 2. By trying to change different values of those parameters, the numbers mentions before make best performance on the 2-D datasets. In addition, for the 5-D datasets, we use the same values that we mentioned before because those values can produce the best performance in the system.

Algorithm No. 2 Description: The Random Forest (RNF) consists of a large number of individual decision trees that operates as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes the model's prediction. RNF algorithm can be used for both classification and regression problems with large larger datasets. It also helps identify most significant variables from thousands of input variables.

We use Scikit-Learn which is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms such as RNF and k-Means. For the 2-D datasets, we draw a plot of error rate and the number of the tree from 1 to 1024 based on the train sets. Then, we select that the number of trees related to minimum error rate is 126. For the 5-D datasets,

we use the same progress of dealing with 2-D sets. However, the number of trees related to minimum error rate is 256.

Results:

Data Set	2D			5D		
Algorithm	Train	Dev	Eval	Train	Dev	Eval
Scikit-Learn: Random Forests (RNF)	00.00%	8.25%	9.10%	00.00%	38.38%	38.54%
PyTorch: Multilayer Perceptron (MLP)	8.09%	8.25%	8.30%	40.24%	40.36%	40.25%

Table 1. the error rates of two-classes datasets with two dimensional and five dimensional features

The Table 1 shows the error rate of two-classes train, dev, and eval datasets with two dimensions features and two-classes train, dev, and eval datasets with five dimensions features using Random Forests (RNF) and Multilayer Perceptron (MLP) algorithms. When we use the RNF algorithm, we observe that the error rate of 2D train sets and 5D sets both are 00.00%. This is because the train sets are over-trained. We set the number of decision trees as 126 for the 2D train set and as 256 for the 5D train set. It is very surprising to get 00.00% of error rate of the both train sets with such a relatively low complexity because it can memorize 100,000 patterns with few trees and each of the trees must be very big. This shows RNF is very good at memorizing specific patterns. RNF is highly scalable to any number of dimensions and has generally quite acceptable performance. Compared the results of 2D sets with different algorithms, we observe that the MLP algorithm has a better performance exception for the train set. However, the RNF algorithm does better performance on the 5D datasets. The eval set is 38.54% after testing by the professor. In addition, the speeds of running these two algorithms are distinct. The speed of running RNF algorithm is faster than the speed of running MLP using the same datasets, which means that RNF is very powerful algorithm in this case.

Conclusions: Overall, we use the Multilayer Perceptron (MLP) and the Random Forests (RNF) algorithms to predict the labels based on two-classes train, dev, and eval sets with two dimensional and five dimensional features. We find that different algorithms applying to same datasets produce different results of error rates. The reasons why the performances are different are because there are many factors affected the results of error rates such as the parameters in the algorithms, selected features, the amount of the data and the selected algorithms. Enhancing a model performance can be challenging at times. For example, the selected features matter the algorithm. Sometimes, the selected the features are suitable for one particular algorithm. For example, the MLP algorithm has better performance on the 2D sets. However, the performance is worse on the 5D datasets. Therefore, hitting at the right machine learning algorithm is the ideal approach to achieve higher accuracy. Some algorithms are better suited to a particular type of datasets than others.

In addition, we find that there are many parameters in different algorithms. Setting right values for these parameter is not easy. Every little change may cause a big difference on the system. When we set different values to the parameters, we only change 0.0001 difference on the parameters based on 5D datasets using MLP. The error rate of dev set increases dramatically from 40.36% to 50.00%. This means that the change of that is statistically significant. Because the parameters can also effect the result of error rate, it is hard to adjust those parameters suitably. Therefore, these factors of machine learning are worthy to think about. We need to have a clear picture of the data before choosing different machine learning algorithms. Some algorithms can work with smaller sample sets while others require tons of samples.