Machine Learning Final Project

Omidreza Ahmadzadeh
Department of Mechanical Engineering, Temple University
omidreza.ahmadzadeh@temple.edu

Introduction: In this research, there are two datasets of two-dimensional and five-dimensional data that were generated with a similar distribution. We analyze the behavior of a 2D data set to find the best classifier. The data will be classified based on two different types of machine learning algorithms. The first one is from the traditional approaches (non-neural network), and the second is from the neural network. Figure 1 depicts the train data of 2D data, and probably it was generated with a Gaussian distribution, moreover, the data and the neighbors of the data are from the same class, therefore, we choose K-nearest neighbors (kNN) algorithm for classifying the data, for non-neural network approach. For neural network classifier, Multi-Layer Perceptron (MLP) was implemented. Since we can modify the number of layers and neurons in each layer easily. The 2D data set consists of 10,000 train data points, 2,000 development data points for validation, and 2,000 evaluation data points. Similarly, the 5D data set has 100,000 train data points and 10,000 data points for evaluation and development datasets.

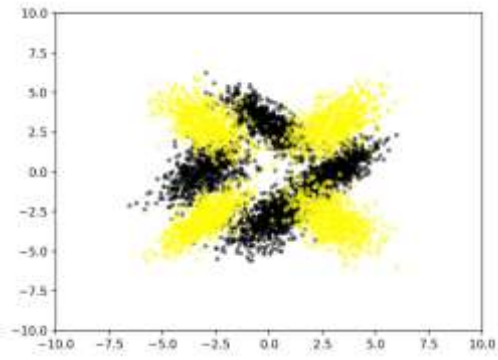

Figure 1 Scatter plot of the 2D train data

Algorithm No. 1 kNN: K nearest neighbors is a simple and supervised machine learning algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already at the beginning of the 1970's as a non-parametric technique. Classification Learner from MATLAB software was used for learning. The train and development datasets were collected in a single file, and cross-validation with 5 folds was implemented, and normalization was used in both 2D and 5D data sets. In 2D data the minimum error was achieved with K equals 35, and Euclidean distance was used as a metric. However, 5D data has a better performance with the number of neighbors equal to 351, and Mahalanobis distance was used as a metric. If we used another value for K for both data set, the error would be increased.

Algorithm No. 2 MLP: An MLP consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. MLPs are useful in research for their ability to solve problems stochastically, which often allows approximate solutions for extremely complex problems like fitness approximation. MLPs were a popular machine learning solution in the 1980s, finding applications in diverse fields such as speech recognition, image recognition, and machine translation software. Several training architectures were implemented to find the best performance for classification data. We got the minimum error with three hidden layers for 2D data, therefore, we assumed three hidden layers are suited for the 5D data set. If we increase the number of layers, the error rates on train data were decreased; however, the error rates on development data were increased. Since it causes overfitting on train data and losing generalization. The MLP was implemented from the Keras library in Python. By try and error and searching, I used 1024, 2048, and 512 neurons for hidden layers, respectively with 10 epochs. Also, the sigmoid function has a better result, so it was used for activation function. 10 epochs mean that the algorithm runs 10 times to find the best results. Since MLP highly depends on initial conditions, and they are random, each time we reach the different error rates for our network.

Results: The kNN algorithm was run on a 2 GHz Core I7- third generation with 6 GB Ram and Matlab. For the MLP algorithm, since I used a high number of neurons, I used Google Colab with 12 GB Ram and Python that ran very fast. The results were shown in the following table.

	2Data Set			5Data Set		
Algorithm	**Train**	**Dev Test**	**Eval**	**Train**	**Dev Test**	**Eval**
kNN	7.81%	7.85%	8.05%	36.71%	36.70%	37.01%
MLP	8.13%	7.85%	8.95%	36.40%	37.11%	36.58%

Table 1. Error rates for both 2D, and 5D data sets, based on KNN and MLP algorithms.

From Table 1, it was shown that for 2D data set kNN has better results since this data is simpler and kNN can produce a good result. Moreover, from Figure1 it is clear that the data the neighbors of a point probably have the same label as the point. On the contrary, for the 5D data set MLP algorithm has better performance since 5D data has more features and more complicated than 2D data. By changing the number of layers or neurons our error rates will be increased in MLP approaches for both data. Finally, we can mention that for evaluation 2D data, it is better to use the kNN approach, and for evaluation 5D data we can use the MLP algorithm.

Conclusions: In this project we used non-neural and neural network algorithms to predict the class of the 2D and 5D data. We can say that both approaches with some tuning have a similar result. For example, we tuned the number of layers in MLP to avoid overfitting. In our learning method, we did not consider the evaluation data set. We only used train data and development data for training and testing the system, and the system with a minimum error on development data was used for classifying the evaluation data set. For kNN, we used cross-correlation, and it caused a better performance of the system, since instead of using only development data set as a validation data, the whole train and development data were partitioned to 5 groups, and each time we use 1 group for validation (20 percent of the data). Similarly, for MLP we use 10 epochs to run our Network 10 times to find the best performance, if we run it more than 10 times, the error rates slightly increase, and if we consider the cost of the time, therefore 10 epochs are enough. As we mentioned before both results have similar performance, so we choose any of them based on the application and resources.