ECE 532 Project Final Report

Github Repositary link: lubyant/ECE532_CourseProject

Abstract and Introduction:

Machine learning is a powerful approach to discover the statistical patterns of dataset. Machine learning can be implemented in different ways by applying sorts of algorithms. This leads to an interesting field that the same dataset but trained in different manner with different algorithm would cause completely different outputs. Applying the correct classifier or regression method is essential to achieve a promising training result.

The goal of this project is to investigate several machine learning algorithms by examining the performance of several classifiers trained by dataset of Fashion-MINIST. Throughout the process of researching this project, I wish to utilize the idea of the machine learning process and strengthen the understanding of the machine learning principle in a practical manner.

Data Source and Preprocessing

The training dataset is provided by Fashion-MNIST, an online public database produced by Zalando's article images. The total amount of samples data in the training dataset is 60000, and each sample consists 784 entities which was digitized based on a 28-pixel times 28-pixel greyscale picture images. It is a supervised dataset that each sample has been characterized and categorized as label 0-9, which represents the image shape such as T-shirt, trouser etc. Meanwhile, there is a 10000 samples test set is also provided, and it will be treated as hold on set for the final validation.

The dimension of this dataset is 60000*784 which is so large that it is time consuming for computing and space consuming for storing. In order to save runtime and storage, the dimension of dataset is reduced to only reflect the relatively importance feature by applying Principal Component Analysis. The figure 1 presents the shape of singular values computed by singular value decomposition. The threshold of the dimensional reduction is 95 percent of cumulative of summation of singular value – 187 out of 784. Therefore, by approach of PCA, approximately two-third of dimensions are removed so that it obviously accelerates the run time and save the storage.

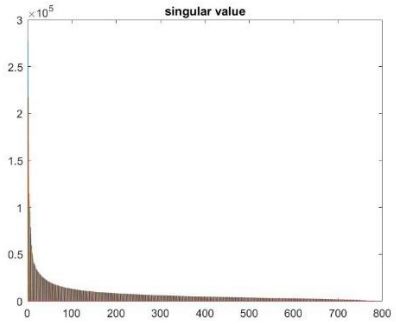

Figure 1. the singular value plots by the SVD

Method and Algorithm

- Linear classifications: The linear classifier is the most basic and simplest method in this project so that each input features are linearly combined to form the output. There are several algorithms apply here and they are organized in the following table 1. The support vector machine is the major classifier for training the dataset, as well as some extended reformation of the SVM. Since the labels are categorized into 10 groups from 0 – 9, here, I apply the comparison method – one-vs-one comparison. For One-vs-one comparison, only current class

is positive, another one compared class is negative, and the rest of classes are ignored. The advantage of this design is that it exhausts all combinations of class pair assignments. Since the SVM does not have the close form, Stochastic Gradient Descend is applied here to compute the approximation of classifier.

Name	Linear SVM	Quadratic Kernel SVM	Cubic Kernel SVM	Gaussian Kernel SVM
Multiclass method	One-vs-one	One-vs-one	One-vs-one	One-vs-one
Kernel function	none	$K(x,y) = (1 + x'y)^2$	$K(x,y) = (1 + x'y)^3$	$K(x,y) = \exp(-(x-y)^2)$
Learning method	SGD	SGD	SGD	SGD
CV	5-folds	5-folds	5-folds	5-folds

Table 1. the linear classifiers and their algorithms and parameters

- K nearest neighbors: The KNN method is another method of setting a classifier as it computes the distance between the sample of training and sampling data. For a k-nearest neighbor, we apply a range of value k to find out the shortest distance between a test set with number k of the training set. This k value can be iterated to investigate the pattern. In KNN, I have tried different types of distance function.

1. Euclidean Distance: the most common distance function used in KNN classfier. The reason to choose this function is that it is close to the l2 loss function in linear classifier
2. Cityblock Distance: it is a distance usually used in analyzing the geographic info which consider the distance between two points as two tangible line at corner.
3. Minkowski Distance: select it because it is between cityblock and Euclidean distance
4. Correlation Distance: this distance is defined as one minus the correlation between y and X.

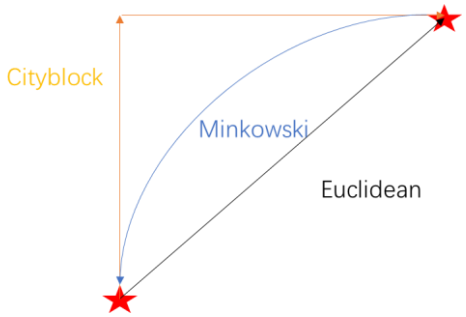

Figure 2. the three types of KNN distance function

- Artificial Neural Network: The artificial neural network mimics the biological idea that it builds up with several layers of data neuron: one inputting layer of training data, one output layer of target pattern, and several unseen hidden layers with neurons that computing from the previous layer. The structure of network system was presented in figure 2 schematic. In the input layer, there are 784 neurons for the adapting the all the features dimensions. In the hidden layer, initially, I set up 10 neurons for training the classifier with a sigmoid activation function. In the output layer, I set one output neuron for reporting the label. The algorithm for training the network is Levenberg-Marquardt backpropagation, which provided in MATLAB function.

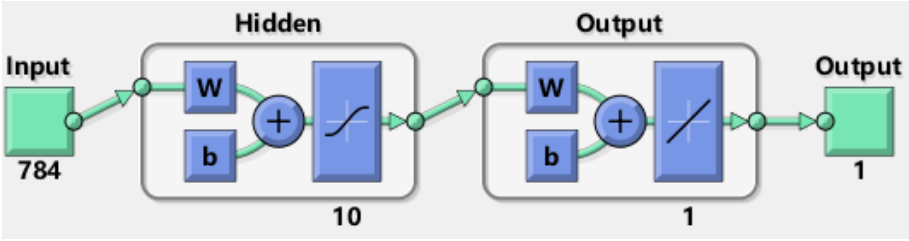

Figure 3. the setting of neural network

Result

1. Linear Classifier

Name	Linear SVM	Quadratic Kernel SVM	Cubic Kernel SVM	Gaussian Kernel SVM
Training Error Rate	13.4%	10.2%	10.3%	10.9%
Validation Error Rate	13.1%	9.9%	9.5%	9.9%
RMSE	1.34	1.20	1.19	1.21
Computing Time	770.16s	896.15s	976.99s	1364.2s

Table 2. result of 4 types of Kernel and linear Support Vector Machine

The results of several linear classifiers are presented in table 2. Generally, Linear SVM has fastest computing time with lowest accuracy and worst model performance. Other Kernel SVM classifiers will cost more running time for a higher performance of modeling. The error rate and root mean square certainly drop in a considerate manner compared to non-kernel SVM. Inside of Kernel SVMs, the different Kernel functions do not reflect a significant variation in terms of the model performance. Nevertheless, the computing time increases as I applied Gaussian function for Kernel SVM.

2. KNN

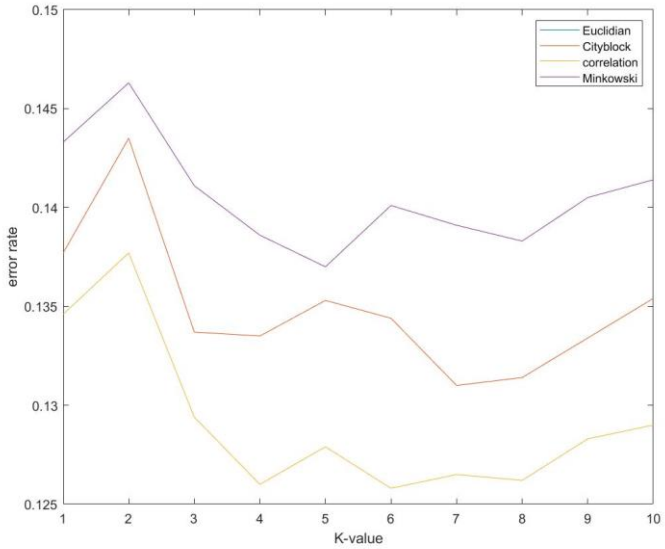

Figure 4. the error rate plots for different k-values and different distance functions

In figure 4, a series of K-value are iterated to find out the best performance KNN classifier. At the same time, different distance functions are also compared to each other in terms of their modeling performance. It turns out that the correlations distance function is considered to be superior to other three

distance function. Euclidean distance and Minkowsski distance have the same performance so that in figure that they are overlapped. City block distance has the decent performance between correlation and Euclidean distance. As for the k values, the optimized k -value is approximately around 3 to 7, depending on your selected distance function. For example, if you apply correlation distance function, the optimized k value will be 4. If you apply the Cityblock distance function, k -value becomes 7 to reach to local minimal.

3. ANN:

In figure 5, the neural network training performance is presented in plot forms in a series of iteration epochs compared to several dataset: the training dataset, the validation dataset and the testing dataset. The overall performance of network is good that root mean square is around 1.08 and error rate is at 8.9%. The total computation time is 45 minutes which is far longer than linear classifier and KNN method.

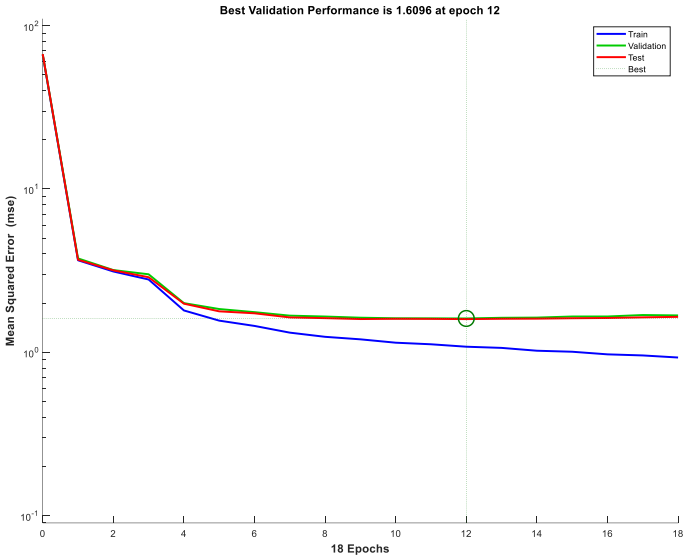

Figure 5. the performance of ANN with the epochs of iteration

Strength and Limitation among three classifiers:

The three type of classifier have their advantage and disadvantage in data training. For the linear classifier, it has the least training time, and its performance is quite decent and promising. The limitation of linear classifier is that it can deal with too complicate class boundary. For the nearest neighbor, the performance of trained model is slightly poor and running time is long since it iterates through different k values. For artificial neural network, its performance is superior to other classifier as well as its long running time. Meanwhile, ANN is also storage consuming for restoring the neural structure. In this case, I only have 10 neurons in one hidden layer. If the number of hidden layer or neurons increase, the cost of computing and storing will also increase.

Conclusion:

The output of three different classifier with different algorithm or parameter has been already shown above. It turns out that by applying different method, there are small variation for the modeling performance from approximate 12% error rate (the linear SVM) to 8% error rate (ANN). The pros and cons are evaluated among different classifier method that if your computer is good at computing and storage, choosing the ANN will be a good choose. The linear classifier and Kernel SVM could be a compromising option between the modeling performance and training cost. The kNN method is non-parametric method which is relatively easy to establish the training method.