Classifiers Based Machine Learning and Deep Learning

Zhanteng Xie
Department of Mechanical Engineering, Temple University
zhanteng.xie@temple.edu

Introduction: Machine learning is a method of data analysis that automates analytical model building. It is very powerful and can solve a lot of problems such as classification problems and regression problems. We have learned so many machine learning algorithms in this semester. These algorithms includes maximum likelihood (ML) classification, principal component analysis (PCA), linear discriminant analysis (LDA), maximum likelihood estimation (MLE), Bayesian estimation (BE), K-nearest neighbor (KNN) algorithm, random forest (RF) algorithm, hidden Markov model (HMM) and K-Means clustering algorithm. We also learned how to use statistical significance to evaluate our system with the baseline system. At the end, we learned neural networks and deep learning algorithms, which include multilayer perceptron (MLP), convolutional neural network (CNN), recurrent neural network (RNN) and autoencoders.

In this report, we need to use a machine learning algorithm and a deep learning algorithm to classify a 2d data set and a 5d data set. How to select the suitable algorithms becomes our primary problem. In order to selecting the suitable algorithms, we need to analysis the data sets and the performance of these algorithms. 2d data set is the data set with two features and 5d data set is the data set with five features. From the previous experiments, we compared different machine algorithms to classify this kind of data sets and got the conclusion that KNN algorithm has a better performance than other machine learning algorithms for this kind of data sets. So we plan to use KNN algorithm as our machine learning algorithm to classify these two data sets. As for the deep learning algorithms, MLP is more suitable for this kind of data sets because CNN is more suitable for image classification, RNN is more suitable for speech recognition and autoencoders is more suitable for image segmentation. So we plan to use MLP algorithm as our deep learning algorithm to classify these two data sets.

Algorithm No. 1 Description: KNN algorithm is a simple and useful supervised machine learning algorithm. It can use to solve the classification and regression problems. In this report, we used it to solve the classification problems of 2d data set and 5d date set. There is the process of K-nearest neighbor algorithm. Firstly, we need to determine parameter value k which is the number of nearest neighbors. Secondly, we need to calculate the distance between the query-instance and all the train samples. Thirdly, we need to sort the distance and determine nearest neighbors based on the k-th minimum distance. Fourth, we need to gather the category of the nearest neighbors. Finally, we need to use a majority vote rule to get the prediction category of the query-instance.

In order to finding the optimal k parameter for particular data sets, we selected different k value to implement the KNN algorithm, whose range is from 1 to 1024 in powers of 2. We used the error rate to evaluate the performance of different k value. As a result, we found that 32 is the optimal k parameter for the 2d data set and 1024 is the optimal k parameter for the 5d data set. We used the fitcknn function of Matlab to implement these two KNN algorithms with the optimal k parameters.

Algorithm No. 2 Description: A MLP is a deep artificial neural network. There are at least three layers of nodes in a MLP: an input layer, a hidden layer and an output layer. Every node is a neuron using a nonlinear activation function except the input nodes. MLP uses backpropagation algorithm which is a supervised learning algorithm to train its model. Due to its nonlinear activation function, MLP is very suitable for the classification problems where the data are not linearly separable. Although MLP is powerful to nonlinear classification problems, training a MLP is not an easily thing. There is a basic process to implement a MLP neural network. Firstly, we need to design the network architecture for our MLP. We need to make sure how many layers in a MLP, how many nodes in a layer, and which

activation function we need to use. Then, we need to select a suitable backprogation algorithm and set relevant hyperparameters to train the MLP. There are many backprogation algorithms such as root mean square propagation (RMSProp), adaptive gradient algorithm (AdaGrad) and adaptive moment estimation (Adam). There are also many relevant hyperparameters such as the learning rate, number of epochs and batch size.

In order to finding the suitable architectures and hyperparameters, we did many different training experiments. We adjusted our architectures and hyperparameters according to the loss curve and the accuracy curve in the train data set and dev data set. As a result, we designed a 3 layers MLP for the 2d data set. The first layer is input layer with 2 nodes, the second layer is the hidden layer with 16 nodes and the third layer is the output layer with 2 nodes. We use rectified linear unit (ReLu) function as the activation function. We selected the Adam optimizer to do the backprogation training. The number of epochs is 1350. The initial learning rate is 6e-3 and decays with increasing number of epochs. The batch size is 360. The L2 regularization parameter is 1e-5. We designed a 4 layers MLP for the 5d data set. The first layer is input layer with 5 nodes, the second layer is the hidden layer with 10 nodes, the third layer is the hidden layer with 10 nodes, and the fourth layer is the output layer with 2 nodes. We use ReLu function as the activation function. We selected the Adam optimizer to do the backprogation training. The number of epochs is 2454. The initial learning rate is 1e-3 and decays with increasing number of epochs. The batch size is 360. The L2 regularization parameter is 1e-3. We initialized weight with a normal distribution using kaiming initialization method. Finally, we used Pytorch v0.4.1 package of Python to implement these two MLP networks with relevant suitable parameters.

Results: The error rate results are shown in table 1. Firstly, we look at the results of the 2d data set. We compare these error rates with the pytorch baseline error rates using statistical significance method. In the case of train data set, our results of KNN and MLP are statistically significant at 99% confidence level. In the case of dev data set, our results of KNN and MLP are statistically significant at 95% confidence level. In the case of eval data set, our results of KNN and MLP are statistically significant at 76% confidence level. Secondly, we look at the results of the 5d data set. We also compare these error rates with the pytorch baseline error rates using statistical significance method. In the case of train data set, our result of MLP is statistically significant at 75% confidence level and our result of KNN is statistically significant at 99% confidence level. In the case of dev data set, our result of MLP is statistically significant at 19% confidence level and our result of KNN is statistically significant at 6% confidence level. In the case of eval data set, our result of MLP is statistically significant at 18% confidence level and our result of KNN is statistically significant at 70% confidence level. From the statistical significance results of eval data set, we can see that our result of MLP is statistically significant better at 76% confidence level for 2d data set while our result of MLP is statistically significant better at 18% confidence level for 5d data set.

Conclusions: We used the machine learning algorithm KNN and deep learning algorithm MLP to classify 2d data set and 5d data set in this report. Through selecting the suitable parameters by doing different experiments, we got the suitable classifiers which have good classification performance in these two data sets. Compared with the pytorch baseline classifier system using statistical significance method, we can conclude that our best result of KNN is statistically significant better at 76% confidence level for 2d data set and is also statistically significant better at 70% confidence level for 5d data set. The performance in machine learning classifier is better than that of deep learning classifier in these two data sets.

	2D Data Set		
Algorithm	**Train**	**Dev**	**Eval**
KNN (k = 32)	8.03%	8.10%	8.05%
MLP (3 layers)	8.22%	8.05%	8.10%
	5D Data Set		
Algorithm	**Train**	**Dev**	**Eval**
KNN (k = 1024)	36.89%	37.08%	36.71%
MLP (4 layers)	37.22%	36.97%	37.26%

Table 1. The error rates of algorithms for 2D and 5D data set.