Something Really Interesting

Pujie Xin
tun29857@temple.edu

Introduction: We have 2 datasets, where one has two dimensions and the other has five dimensions. We need to use one Neural Network (NN) based method and one non-NN-based algorithm to classify the data based on training results. To start with, we need to have a look at the data set and observe what they look like and then we choose the proper methods to classify the data.

Algorithm MLP Description: Multi-layer Perception is a class of artificial feedforward neural network (ANN). It uses backpropagation method for training that learns a function $f: R^m \rightarrow R^o$, where m is the number of the input dimensions and o is the number of output dimensions. In our case, in input layer, 2D data has 2 input neurons and 5D data has 5 input neurons, and in output layer, both of the 2D data and 5D data have 2 output neurons. MLP is capable to learn non-linear models, however, MLP with hidden layers have a non-convex loss function where there is more than one local minimum, therefore different random weight initializations can lead to different validation accuracy. What's more, MLP is sensitive to feature scaling and has a requirement on tuning a number of hyperparameters such as the number of hidden neurons, layers, and iterations. In initialization, I used random seed method in pytorch. And similar to given code, I also set the number of hidden layer equal to 1. And the activation function is ReLU.

Algorithm KNN Description: KNN is a simple supervised machine learning algorithm that can be used to solve both classification and regression problems. This method is a non-parametric statistical pattern recognition technique. This algorithm assumes that if points are from same class, they should be close to each other. There are two parameters we need to tune in KNN in our case, which are the distance matrix and the number of neighbors K. I use Euclidean distance as the measurement of distance matrix and set K to different numbers to get the best performance.

Results:

MLP	Error rate(2D)	Error rate(5D)
Training	8.87%	37.82%
Dev	9.2%	38.83%
Eval	8.5%	37.1%
KNN	Error rate(2D)	Error rate(5D)
Training	7.69%	34.72%
Dev	7.85%	38.47%
Eval	8.5%	37.6%

Table 1. Results

MLP process: MLP shows better performance in 2D data than that in 5D data. However, it performs bad on 5D data.

KNN process: KNN shows better performance in 2D data than that in 5D data. However, it performs bad on 5D data. Generally, KNN performs better than MLP.

Conclusions:

Both of the methods perform bad on 5D data. In MLP, although I increase the number of nodes in hidden layer, the performance didn't improve a lot. And in KNN, K is range from $2^1\ to\ 2^{10}$, and I selected the best K in 2D and 5D dataset separately.

Generally speaking, Neural Network should have a better performance. However, when I observed the loss function, the loss was oscillating with a large variance. It means the training process failed, or not worked well.

For KNN, if the data points in same class are not clustered, it is expected to perform bad. I think KNN's performance is acceptable.