

Deep Learning Project 1

20459219

Yi Yang

1 Data analysis

In this project, the dataset has 57 features, and the labels. Firstly, we can create a model by the train data, and use the python numpy library to do prepare work.

1.1 using the readline() and split() to get the feature[] from train data.

```
with open("traindata.csv") as fea:
```

```
    data = fea.readlines()

    for each in data:
        line = each[:-1].split(',')

        feature = []

        for i in line[0:]:
            feature.append(float(i))

        features.append(feature)
```

1.2 same as the 1.1, we can get the target[]

```
with open("trainlabel.csv") as la:
```

```
    data2 = la.readlines()

    for each in data2:
        line = each[:-1].split(',')

        tar = []

        for i in line[0:]:
            tar.append(float(i))

        target.append(tar)
```

2 classical classify

we have the sklearn library that invoked to do some model prediction.

```
model = LogisticRegression()
model.fit(features,target)

AccuracyL = np.mean(model.predict(featuresT) == target)

model = tree.DecisionTreeClassifier()

model = RandomForestClassifier(n_estimators=10)

model = GradientBoostingClassifier(n_estimators=100)
```

Such as the SVM it will be got 0.519263 accuracy;

Test logistic regression function, it will got 0.524724 accuracy;
 support vector machine, got 0.528061 accuracy;
 decision tree got 0.520524 accuracy.
 Gradient Boosting got 0.528061
 gaussianNB got 0.487983;

```
*****
LogisticRegression Accuracy:0.524724
*****
SupportVectotMachine Accuracy:0.519263
*****
GradientBoostingClassifier Accuracy:0.528061
*****
```

```
*****
RandomForestClassifier Accuracy:0.529881
*****
DecisionTree Accuracy:0.524420
*****
```

3 Neural Network

after used those classical classify, we can find the accuracy is so bad. Someone maybe want to improve this accuracy by boost, bagging and adaboost. But I have think maybe we can find the another power method to solve this problem. We have find this dataset has 57 feature, maybe it has too much attribute that meet some trouble in the loss function. So I use the NN to do that.

3.1 use tensorflow create the network, a deep learning model- BP neural net.

```
def add_layer(input, in_size, out_size, activation_function = None):
hidden_layer = add_layer(xs, 57, 45, activation_function = tf.nn.relu)
```

3.2 set the the input layer and hidden layer with some parameters(LR, learning rate, and training time)

```
loss = tf.reduce_mean(tf.reduce_sum(tf.square(ys - prediction), reduction_indices = [1]))
```

after NN-BP, The accuracy is 0. 0.984472049689441 almost 100%. I think that engouh.

```
0.0601093
0.0166937
0.0171383
0.0174222
0.0152493
0.024339
0.014181
[ 0.38927025]
0.984472049689441
1
```

3.3 write a sess.close() and file writing function with do the prediction value.

4 summary

After this project, I not only deeply know the old saying” no classify is best, just fit some dataset”. When I through this project know the trouble, I have been enjoy the deep learning knowledge and power.

