Final Project Report

Jiawei Mo, Du Liang and Yihui Mao

June 12, 2019

Abstract

1 Data Processing

1.1 Normalization

Under what we observed, the first step what we have to deal with is the normalization of the feature. The interval of features is very large, which located at from 0.001 to 10000000, which cause the distorting differences in the ranges of values. We need to change to value to a common scale.
We introduce the function of log in normalization.

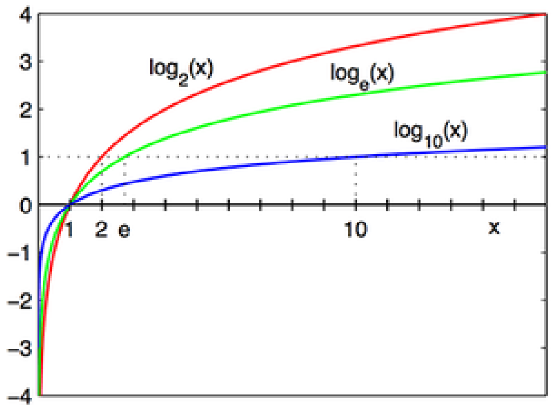

Figure 1: log function

Using the function, we can compress the interval of feature in a small range. For example, if we use the 10 base log function, we can compress the the range of y value from the 0 to 1.2. We use the 10 as the base number, which can guarantee the interval limited in 1 and also would not influence the monotonicity of values.

2 Learning algorithms

2.1 Algorithms/methods explored

- RNN

For sequence prediction, rnn may perform well, because rnn will continuue to fix last time prediction result, and make the last predict as final result. We used python library keras to simple build rnn structure,we set input shape is (103, 1) and recurrent it 10 time. Through test, we got about 90% accuracy of validation data.

- a) model=Sequential()

  b) model.add(LSTM((1),
  batch_ input_shape=(None, 103,1),
  return_sequences = False))

  c) model.compile(loss='mean_absolute$_error$',
  $optimizer$ ='$adam$',
  $metrics$ =['$accuracy$'])

- Neural Network

  Based on our analysis, we find that the features are with correlation among them and capacity of features is huge. Some algorithm, such as SVM , K mean and Bayesian Network would not deal with the complex case and so many data. We believe the Neural Network is more efficient and more powerful. We explored the Neural Network, and found which is better model to fit this case. We use the below architecture of Neural Network.

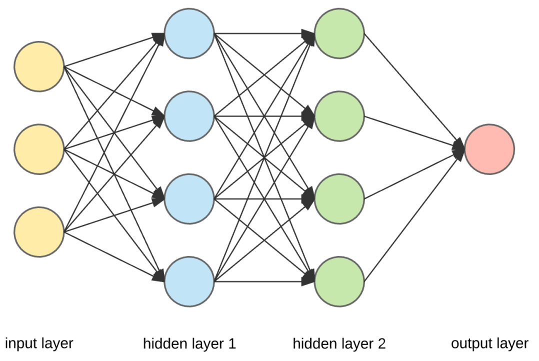

Figure 2: Architecture of Neural Network

We tried three kinds of Neural Network Architecture in feature103.txt

1
a) Input Layer =Dense(16, activation = tf.nn.relu))

b) Hidden Layer = Dense(32, activation = tf.nn.relu))

c) Outpuy Layer = Dense(1, activation = tf.nn.sigmoid))

2

a) Input Layer =Dense(128, activation = tf.nn.relu))

b) Hidden Layer = Dense(512, activation = tf.nn.relu))

c) Outpuy Layer = Dense(1, activation = tf.nn.sigmoid))

3 a) Input Layer =Dense(64, activation = tf.nn.relu))

b) Hidden Layer = Dense(256, activation = tf.nn.relu))

c) Hidden Layer = Dense(64, activation = tf.nn.relu))

d) Outpuy Layer = Dense(1, activation = tf.nn.sigmoid))

We tried three kinds of Neural Network Architecture in featureall.txt

1 a) Input Layer =Dense(512, activation = tf.nn.relu))

b) Hidden Layer = Dense(1024, activation = tf.nn.relu))

c) Outpuy Layer = Dense(1, activation = tf.nn.sigmoid))

2 a) Input Layer =Dense(16, activation = tf.nn.relu))

b) Hidden Layer = Dense(32, activation = tf.nn.relu))

c) Outpuy Layer = Dense(1, activation = tf.nn.sigmoid))

3 a) Input Layer =Dense(64, activation = tf.nn.relu))

b) Hidden Layer = Dense(256, activation = tf.nn.relu))

c) Hidden Layer = Dense(64, activation = tf.nn.relu))

d) Outpuy Layer = Dense(1, activation = tf.nn.sigmoid))

2.2 Final Model

We compared the above models, we find that the NN is with better performance. We decide to use the NN as our final model.Firstly,we applied the early stopping and the tensor board to monitor the curve of the accuracy and the loss. We find that in the above three NN model, the third one with more neural units and more layer levels has a better performance in training accuracy and a little better in validation accuracy. It is totally make sense, because more units and more level would lead to more complex models, which could raise the accuracy of the training. We decided to use the third option in NN model, then to tune parameters to adjust learning rate to increase the accuracy of training set and adjust regularization to decrease the over fitting.

3 Parameter Tuning and Model Selection

3.1 parameter Tuning

Firstly, We apply Adam as the optimizer, which can computes adaptive learning rates for each parameter and is also with the momentum to help model to converge more quickly. We tried 0.1, 0.01 and 0.001 as our learning rate. We find the 0.01 is a proper value to fit in our model, which can more easily raise our accuracy but without over fitting in the whole process. Based on the observation of curve, we find when the accuracy of training set of both files reaches 93.8 %, the model start to over fitting. The accuracy of training set still keep raise, but accuracy of validation is fluctuating. We tried to add the L1 and L2 regularization and dropout function. We find which lead to the fluctuation as the increase of accuracy, and decrease the gap between the accuracy of training set and validation. But it looks seems that once the accuracy of validation reached 94 %, the accuracy would not get raised. We tried three kinds of architecture in feature103.txt and featureall.txt. Then we will discuss which one has performance in the following part.

3.2 model selection

We would choose the model for feature103.txt is below

a) Input Layer =Dense(64, activation = tf.nn.relu))

b) Hidden Layer = Dense(256, activation = tf.nn.relu))

c) Hidden Layer = Dense(64, activation = tf.nn.relu))

d) Outpuy Layer = Dense(1, activation = tf.nn.sigmoid))

e) optimizer =Adam,loss = binary crossentropy, metrics =[accuracy])

We would choose the model for featureall.txt is below

a) Input Layer =Dense(64, activation = tf.nn.relu))

b) Hidden Layer = Dense(256, activation = tf.nn.relu))

c) Hidden Layer = Dense(64, activation = tf.nn.relu))

d) Outpuy Layer = Dense(1, activation = tf.nn.sigmoid))

e) optimizer =Adam,loss = binary crossentropy, metrics =[accuracy])

We use the hold out for model selection.We splitted the original training set into new training set and a validation set with ratio 4:1. We do not randomly form the folds. We pick one value in five values to form the folds. We use the auc and accuracy as the criterion. Compared to all result, I find the above model with better auc and validation accuracy. of evaluation.

4 result

4.1 result in In feature103

- model 1

  a) accuracy of training set: 0.9423
  b) accuracy of validation set: 0.9331
  c) auc of validation set: 0.791

- model 2:

  a) accuracy of training set:0.948
  b) accuracy of validation set: 0.9388
  c) auc of validation set:0.8069

- model 3:

  a) accuracy of training set: 0.953
  b) accuracy of validation set: 0.9402
  c) auc of validation set: 0.792

4.2 result in featureall

- model 1

  a) accuracy of training set: 0.9693
  b) accuracy of validation set: 0.9404
  c) auc of validation set: 0.8091

- model 2:

  a) accuracy of training set:0.96
  b) accuracy of validation set:0.94
  c) auc of validation set: 0.811

- model 3:

  a) accuracy of training set: 0.97
  b) accuracy of validation set: 0.938
  c) auc of validation set: 0.8324

The model what we choose in the part of model selection is the final model we choose, which is with better auc and a good accuracy in validation set.