STAT665

Lab8

Bo Song

I. STL-10 Classification with VGG-19

Shape: the shape of X_train is 5000 * 4096, X_test is 8000 * 4096, Y_train is 5000 * 10, Y_test = 8000 * 10.

Since the last layer of VGG-19 model we used is a dense layer with 4096 nodes, so the X_train and X_test all have 4096 columns.

Dense neural network: 500 nodes in a single hidden layer, with 'relu' activation function and 0.5 dropout ratio. Accuracy is 0.93012

```
model = Sequential()
model.add(Dense(200, input_shape=(4096,)))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(10))
model.add(Activation('softmax'))

model.compile(loss='categorical_crossentropy',
              optimizer=RMSprop())
```

SVM: LinearSVC with default setting. Accuracy is 0.94150.

```
clf = svm.LinearSVC()
Y_train = np.dot(Y_train, np.array([0,1,2,3,4,5,6,7,8,9])) # convert one-hot encode to 1 column
Y_test = np.dot(Y_test, np.array([0,1,2,3,4,5,6,7,8,9])) # convert one-hot encode to 1 column
clf.fit(X_train, Y_train)
```

Lasso: alpha is set to 0.03: Accuracy is 0.76438

```
clf = linear_model.Lasso(alpha = 0.03, max_iter=1000)
Y_train = np.dot(Y_train, np.array([0,1,2,3,4,5,6,7,8,9])) # convert one-hot encode to 1 column
Y_test = np.dot(Y_test, np.array([0,1,2,3,4,5,6,7,8,9])) # convert one-hot encode to 1 column
clf.fit(X_train, Y_train)
```

It seems that a linear model is good enough to fit the data in this question, given the best performance of LinearSVC.

II. Chicago crime data and neural networks

I tried different neural network settings.

1000 * 500 * 500 gave me accuracy at 0.75418 which is best among other settings.

```
model = Sequential()

model.add(Dense(1000, input_shape = (X_train.shape[1],)))
model.add(Activation('relu'))
model.add(Dropout(0.2))

model.add(Dense(500))
model.add(Activation('relu'))
model.add(Dropout(0.2))

model.add(Dense(500))
model.add(Activation('relu'))
model.add(Dropout(0.2))

model.add(Dense(5))
model.add(Activation("softmax"))

model.compile(loss='categorical_crossentropy', optimizer=RMSprop())
```

My neural network selecting strategy is as follows.

First I tried the setting with 3 layers and each layer contains 1000 nodes. The cross validation accuracy is a little higher than test accuracy which indicates overfitting. Then I shrink the neural network size to 1000 * 500 * 500. It still contains 3 layers while the last two layers only contain 500 nodes each. The later network gave a better accuracy at 0.75418.

The reason I used 3 layers rather than 1 or 2 is that in pset3, we only use random forest model while it did not perform well. So I think we need to build a NN that is complex enough to handle this question. That’s why I chose a NN with 3 layers.