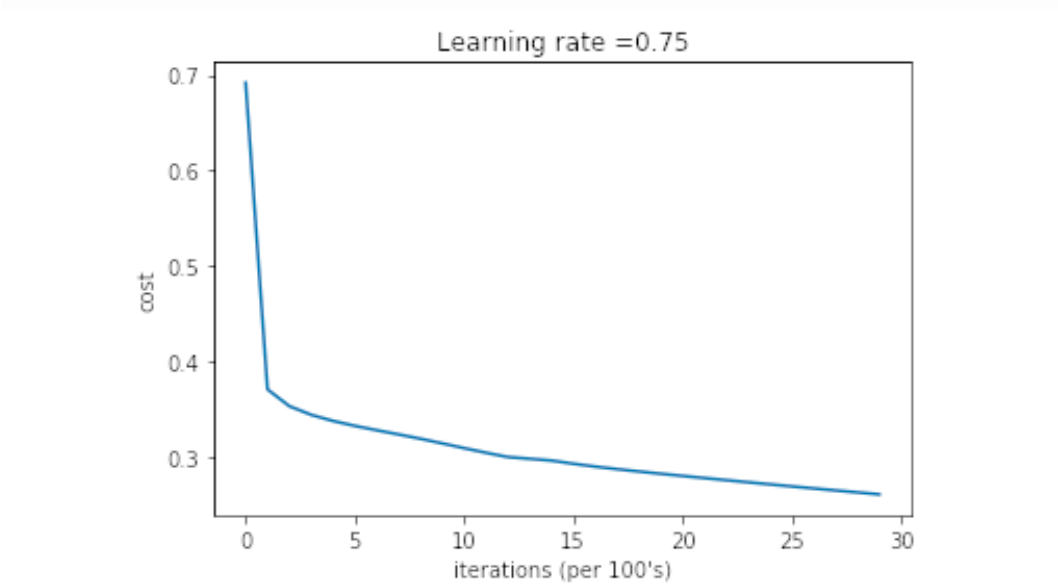
Assignment 3
(Neural Networks)
(Nikita Mehrotra, PhD18013)

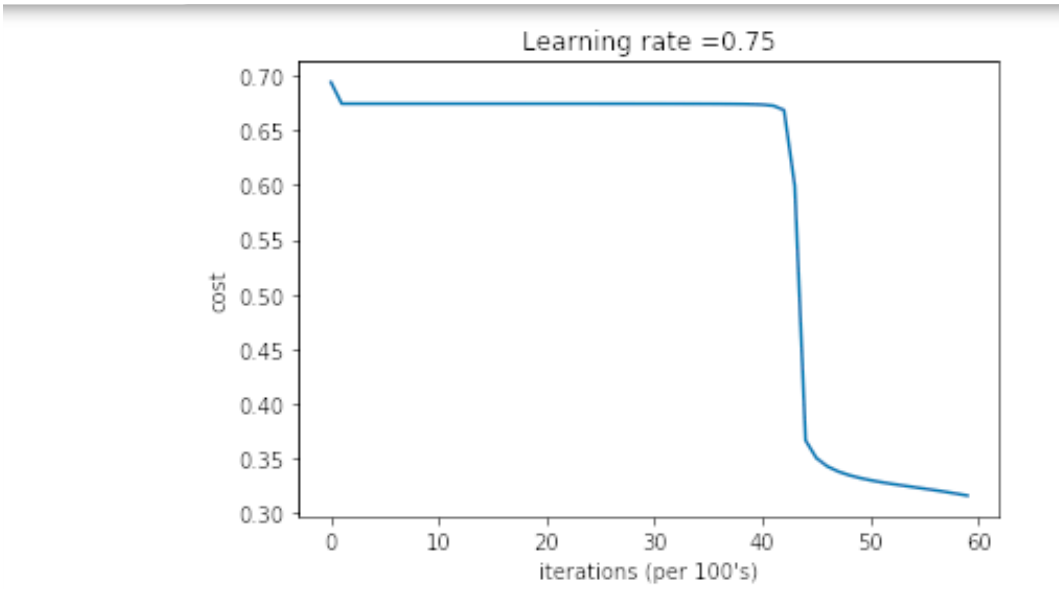
1. **NN from Scratch (using sigmoid)**
   1. **1 Hidden Layer**




```
Accuracy: 0.8903258145363407
Accuracy: 0.949485500467727
```

ACCURACY & LOSS CURVE

   2. **3 Hidden Layer**



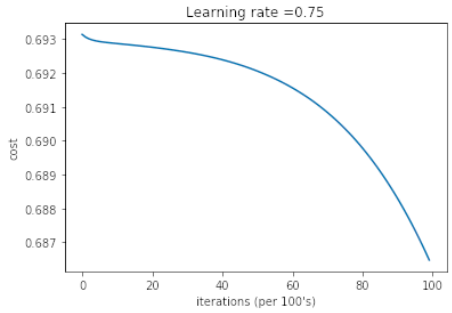

```
Accuracy: 0.8687719298245613
Accuracy: 0.9506548175865295
```

ACCURACY & LOSS CURVE

2. **Analysis for Overfitting (part 1):** Since the accuracy on training and test dataset is approximately same, so we conclude that our neural net does not overfits the data.

3. **NN from Scratch (using relu)**
   1. **1 Hidden Layer**

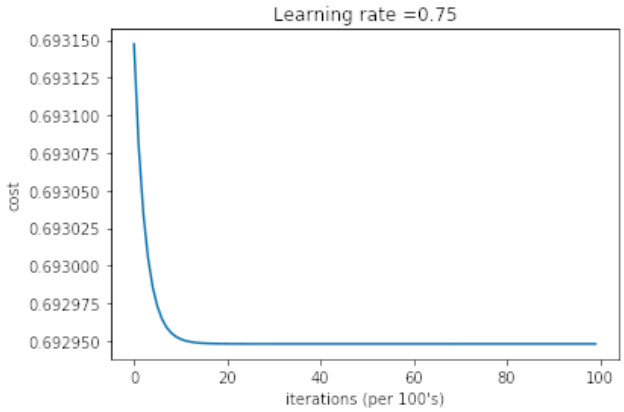
```
Cost after iteration 50: 0.692058
Cost after iteration 60: 0.691561
Cost after iteration 70: 0.690833
Cost after iteration 80: 0.689780
Cost after iteration 90: 0.688290
```




```
[[0. 0. 1. ... 0. 0. 0.]]
Accuracy: 0.57062656641604
[[0. 0. 0. ... 1. 1. 1.]]
Accuracy: 0.4964920486435921
```

ACCURACY AND LOSS CURVE

   2. **3 Hidden Layer**




```
[[0. 0. 0. ... 0. 0. 0.]]
Accuracy: 0.5099749373433584
[[0. 0. 0. ... 0. 0. 0.]]
Accuracy: 0.5159027128157156
```

ACCURACY AND LOSS CURVE

4. **Analysis for Overfitting (part 2):** Since the accuracy on training and test dataset is approximately same, so we conclude that our neural net does not overfits the data.

5. Scikit Learn Implementation

1. 3 Hidden layer

```
MLPClassifier(activation='logistic', alpha=1e-05, batch_size='auto',
       beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08,
       hidden_layer_sizes=[100, 50, 50], learning_rate='constant',
       learning_rate_init=0.001, max_iter=200, momentum=0.9,
       n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
       random_state=1, shuffle=True, solver='sgd', tol=0.0001,
       validation_fraction=0.1, verbose=False, warm_start=False)
training score with 1 hidden layer 0.9755639097744361
confusion metrics(Training):
 [[3978  112]
 [  83 3807]]
Classifictaion report               precision    recall  f1-score   supp

           7       0.98      0.97      0.98      4090
           9       0.97      0.98      0.98      3890

   micro avg       0.98      0.98      0.98      7980
   macro avg       0.98      0.98      0.98      7980
weighted avg       0.98      0.98      0.98      7980

Validation score with 1 hidden layer 0.9488721804511279
confusion metrics(Validation):
 [[945  52]
 [ 50 948]]
Classifictaion report               precision    recall  f1-score   supp

           7       0.96      0.95      0.96      2206
           9       0.95      0.96      0.95      2070

   micro avg       0.96      0.96      0.96      4276
   macro avg       0.96      0.96      0.96      4276
weighted avg       0.96      0.96      0.96      4276

Testing score with 1 hidden layer 0.9553320860617399
confusion metrics(Test):
 [[2100  106]
 [  85 1985]]
Classifictaion report               precision    recall  f1-score   supp

           7       0.95      0.95      0.95       997
           9       0.95      0.95      0.95       998

   micro avg       0.95      0.95      0.95      1995
   macro avg       0.95      0.95      0.95      1995
weighted avg       0.95      0.95      0.95      1995
```

2. 1 Hidden layer

```
MLPClassifier(activation='logistic', alpha=1e-05, batch_size='auto',
       beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08,
       hidden_layer_sizes=100, learning_rate='constant',
       learning_rate_init=0.001, max_iter=200, momentum=0.9,
       n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
       random_state=1, shuffle=True, solver='sgd', tol=0.0001,
       validation_fraction=0.1, verbose=False, warm_start=False)
training score with 1 hidden layer 0.9878446115288221
confusion metrics(Training):
 [[4037   53]
 [  44 3846]]
Classifictaion report               precision    recall  f1-score   support

           7       0.99      0.99      0.99      4090
           9       0.99      0.99      0.99      3890

   micro avg       0.99      0.99      0.99      7980
   macro avg       0.99      0.99      0.99      7980
weighted avg       0.99      0.99      0.99      7980

Validation score with 1 hidden layer 0.956390977443609
confusion metrics(Validation):
 [[950  47]
 [ 40 958]]
Classifictaion report               precision    recall  f1-score   support

           7       0.97      0.96      0.96      2206
           9       0.96      0.97      0.96      2070

   micro avg       0.96      0.96      0.96      4276
   macro avg       0.96      0.96      0.96      4276
weighted avg       0.96      0.96      0.96      4276

Testing score with 1 hidden layer 0.9618802619270346
confusion metrics(Test):
 [[2114   92]
 [  71 1999]]
Classifictaion report               precision    recall  f1-score   support

           7       0.96      0.95      0.96       997
           9       0.95      0.96      0.96       998

   micro avg       0.96      0.96      0.96      1995
   macro avg       0.96      0.96      0.96      1995
weighted avg       0.96      0.96      0.96      1995
```

3. CIFAR 10 Dataset

    1. Accuracy on Training Dataset: **0.9278**
    2. Accuracy on Test Dataset: **0.8965**

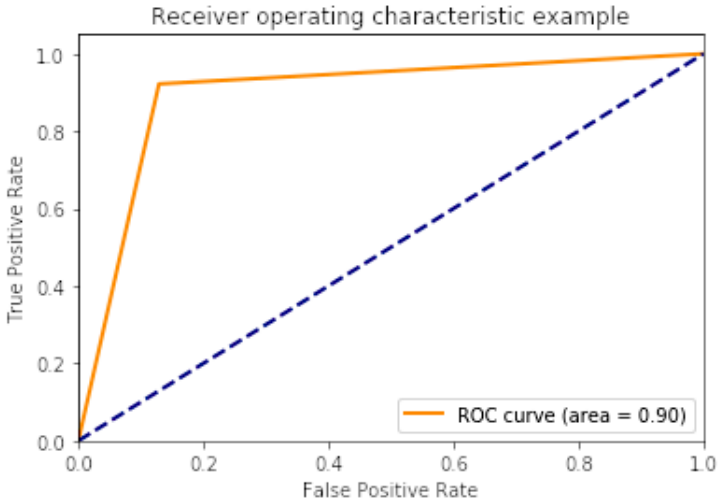
    Confusion Matrix

```
]: predicted_train=SVM.predict(extracted_features)
   # predicted_test=SVM.predict(test_X)

   #print('Confusion Matrix on Test Dataset':confusion_ma

]: print('Confusion Matrix on Training Dataset:\n',confus

   Confusion Matrix on Training Dataset:
    [[4534  466]
    [ 256 4744]]
```

ON TRAINING

```
]: print('Confusion Matrix on Test Dataset:\n',confusion_

   Confusion Matrix on Test Dataset:
    [[871 129]
    [ 78 922]]

]: from sklearn import metrics

   fpr, tpr, thresholds = metrics.roc_curve(test_Y, predi

]: import matplotlib.pyplot as plt
   roc_auc = metrics.auc(fpr, tpr)
```

TEST DATASET


Receiver operating characteristic example
True Positive Rate
False Positive Rate
ROC curve (area = 0.90)
0.0
0.2
0.4
0.6
0.8
1.0

ROC CURVE