COL774 Machine Learning

Pranav Bhagat - 2016CS10352

April 26, 2019

1 Problem 1

1.1

Library used for pca and Svm part :-

- cv2 and glob(for reading the image).
- sklearn's PCA and IncrementalPCA(for PCA)
- sklearn's SVC,LinearSVC(for Gaussian and linear svm)

Library used for Cnn part :-

- cv2 and glob(for reading the image).
- keras

Link for model: modelweights
It can be checked using check_cnn.py present in the directory.

1.2

For Pca and Svm:

- Input of the images was done in greyscale using cv2.
- Firstly calculate pca on the full data set using incremental pca with batch size 100 and top 50 dimensions accounted for around 82% of total variance.
- Now using this we transformed our training data and then genearted around 300k examples for our svm model.
- Both Linear Svm and Gaussian SVM didn't converge. So I ran them for fixed number of iterations.
- in Linear SVM upon running for 50000 iterations with C parameter = 1, It gave around 0.35 f1 score on the validation data. I tried varying the number of iteration like for 20000 iteration it gave around 0.27 f1 score.

- In case of Gaussian SVM with Gamma as reciprocal of number of features and C =1. I ran for two cases with iteration = 50000 and 100000 both resulting in minimal f1 score of 0.1.

1.3

For CNN:

- We generated around 400k examples for training our cnn model. And due to the large memory of training data we used generators for both validation and training data. Dimension of an example was (210x160x15) by stacking the sequence of 5 images.
- Architecture of the CNN model was same as described in the pdf. And for loss I used binary_crossentropy with adam otimizer.
- I ran it for 10 epochs. As the epoch increases training f1 score keep on increasing and for 10 epoch . But validation f1 score was increasing till 6-7 epochs after which validation f1 score starts decreasing suggesting about overfitting in later epochs.
- I ran the cnn network for two to three sets of data set each resulting in a similar range of validation f1 score.

1.4

For Competetive part:

- I used a similar sort of network from the cnn part with slight modification like adding dropout of 0.25 after each layer
- Learning rate was default 0.001 for adam otimizer and Batch size was 128.
- f1 score on validation data = 0.5366
- f1 score on test data(40%) = 0.5461