COL774 Assignment 4

Vishal Bindal (2018CS50425)
Chirag Mohapatra (2018CS50403)

January 16, 2021

Non competitive part

Here all times denote the training times in seconds, and all f-1 scores denote the macro f-1 score multiplied by 100

a) **Vanilla neural network**
Learning rate: 0.1
Batch size: 100
SGD optimizer
epsilon 1e-4
max 100 epochs

Experimentation with hidden layer activation:
Relu
Time: 34.64810252189636
Train f-1: 98.14087685598051
Test f-1: 37.72851051012835

Leaky relu
Time: 34.34696388244629
Train f-1: 94.35404968309084
Test f-1: 36.583156746893685

Sigmoid
Time: 34.409926414489746
Train f-1: 97.19654384940632
Test f-1: 34.338805408298775

Tanh
Time: 34.564528942108154

Train f-1: 98.56884967633265
Test f-1: 33.695052358400076

Thus the relative performance of the model varies as *Relu* > *Leaky relu* > *Sigmoid* > *Tanh*

b) **Feature engineering**
Learning rate: 0.1
Batch size: 100
SGD optimizer
epsilon 1e-4
max 100 epochs
ReLU activation in hidden layer

Gabor
Time: 35.21726655960083
Train f-1: 98.81549927093614
Test f-1: 35.35701516651953

HOG
Time: 34.72723412513733
Train f-1: 66.02495393559022
Test f-1: 42.40706955644302

HOG improves the performance on test set, while Gabor degrades the performance.

c) **CNN**
Learning rate: 0.1
Batch size: 100
SGD optimizer
epsilon 1e-4
max 100 epochs

Time: 42.0122652053833
Train f-1: 98.93396900293293
Test f-1: 38.20937725900574

CNN, being a bigger model, takes some more time to train and predict compared to the vanilla neural network, but also gives slightly better performance.

Competitive part

We tried various models, including various variations of resnet, eigenfaces+svm, etc. Details of our best performing model are as follows:

- We used *resnext101_32x8d* (imported from *torchvision.models*) as it gave the best peformance relative to others. *pretrained* was set to True. The final FC layer was modified to accomodate 7 perceptrons in the output layer.

- We used SGD optimizer with learning rate 0.01. Batch size for training was set to 100

- We used data augmentation to increase training examples. For each example, we added a horizontally-flipped example to the train set. We also included the public test set for training in the final model.

- We also added rotation of images by 10 degrees to augment our dataset further.