Machine Learning assignment 4

Gulshan Jangid (2014CS50285)

(PL: python 2.7 for part 1a , python 3 for part 1b,part 2)

For generating the train and test dataset I have used a window of 7 frames before frame having reward of 1. In that window having last frame always chosen I randomly choose 4 frames out of remaining 6, this is done 3 times. After having dataset with reward 1, I choose twice the amount of data with same approach for reward 0 frames.

PART 1 (A):

Libraries used: sklearn, pandas, cPickle, numpy

Preprocessing: Cropping image to 179x160 i.e. removing score bar from the top

For Linear svm:

C=1, class_weight= {1:2.5, -1:1}, tolerance = 0.001
Train accuracy = 72%
Test accuracy = 69.15%
Test_F_score = [0.8019, 0.3031]

For Gaussian svm:

C=0.2, class_weight= {1:3, -1:1}, tolerance = 0.001, gamma = 0.02
Train accuracy = 65.67%
Test accuracy = 61.75%
Test_F_score = [0.7334, 0.3818]

Training data = 20k samples of 5 stacked grayscale images

Though gaussian svm increases the accuracy of the class 1 but svm model is still not good compared to CNN where accuracy and F score reach above 95%, because the data points are very close in feature space svm doesn't perform well.

PART 1 (B):

Libraries used: keras, sklearn, pandas, _pickle, numpy

Preprocessing: Cropping image to 157x160 i.e. removing score bar from the top and paddle at the bottom. Used grayscale images.

Train accuracy = 99.48%
Test accuracy = 96.48%
Test F_score = [0.9724, 0.9472]
Learning rate = 0.001
Batch size = 50
Epochs = 4

Training data = 60k samples of 5 stacked grayscale images

Changing the number of number of kernels doesn't affect the accuracy too much. Also adding more and more layers to the cnn starts to overfit the data.

PART 2:

Libraries used: keras, sklearn, pandas, _pickle, numpy
Architecture details:
Learning rate=0.001
Batch size = 50
Epochs = 6
Training data = 200k samples of 5 stacked grayscale images, taken 50k samples at a time

{1 CNN layer with 64 kernels with filter size of (6,6) and stride of 2}
--followed by--
{MAX pooling (of filter size (2,2) and stride = 2)}
--followed by--
{1 CNN layer with 64 kernels with filter size of (6,6) and stride of 2 }
--followed by--
{MAX pooling (of filter size (2,2) and stride = > 2)}
--followed by--
{1 CNN layer with 64 kernels with filter size of (6,6) and stride of 2 }
--followed by--
{MAX pooling (of filter size (2,2) and stride = > 2)}
--followed by--
{1 fully connected layer with 1024 units, ReLU activation}
--followed by--
{1 fully connected layer with 2048 units, ReLU activation}
--followed by--
{1 node output layer with sigmoid activation}.