Assignment 3

Avinash Bhutani (2016EE10427), Akanshu Gupta (2016EE10418),
and Devesh Joshi (2016EE10437)

1 Problems and Datasets

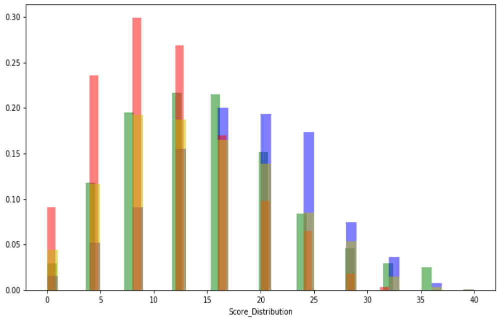
1.1 Performance Prediction for cognitive skills

Problem we chose to work on was proposed by an organization working on innovation and reforms in education system focusing mainly on school education. Data used is from results of first round of Olympiad conducted for over 2000 school students. 4 skills namely **'Quantitative aptitude', 'verbal reasoning', 'lateral thinking', 'analytical reasoning'** were tested in a 40 question paper (10 questions for each skill). From here onward we call them S1, S2, S3, S4 respectively. Aim of the Olympiad is to test kids on exclusive skills so as to provide them right content to improve. Skills chosen above in theory are 'different' from each other. We wish to formalize the degree of difference mathematically.

One of the team members is involved in the organization and hence data was accessible. Same is attached in submission, however identity of students is not revealed.

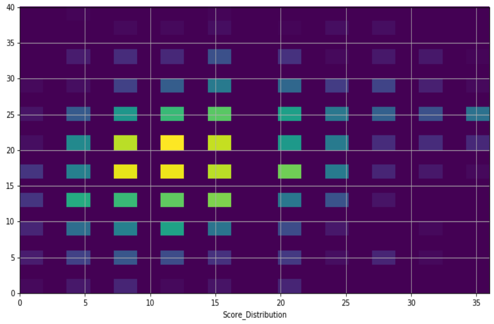
As a part of this assignment, we have implemented

Classification Models:
In this we use data of 3 of the skills for each student and try predicting if a student can pass (threshold taken as avg score) in the left out skill. For 2 out of 4 skills we show that results are nothing better than pure guessing. For 1 of the skill, however naive bayes could correctly predict if a student will pass in 4th skill in more than 79% cases. Histogram of score in Skills (training data):

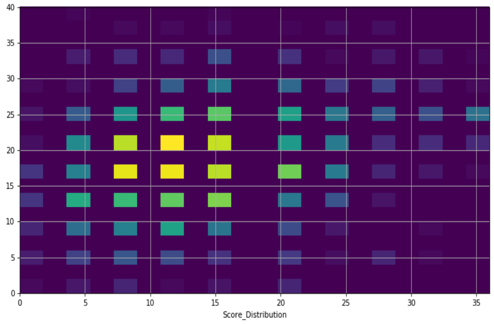


Further following 2D plots (made using 2 skills at a time) show that a 1 mode Gaussian will be a best fit on the data.

Histogram of score in S1 & S2 (training data):

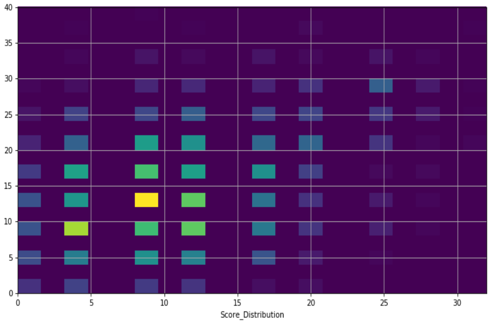


Histogram of score in S2 & S3 (training data):

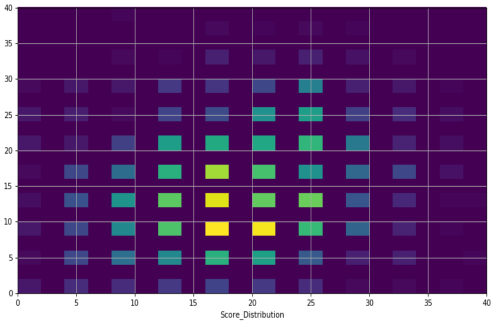


.

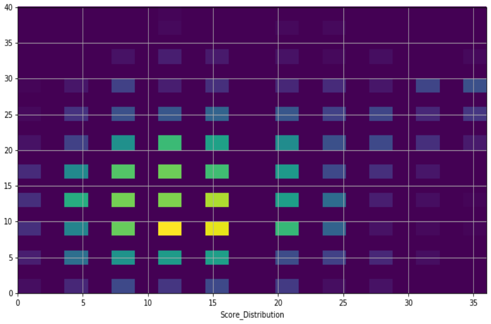
Histogram of score in S3 & S4 (training data):



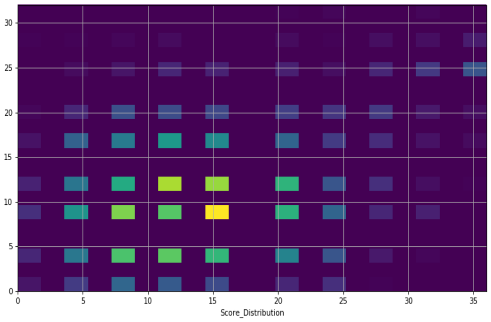
Histogram of score in S4 & S2 (training data):



Histogram of score in S4 & S1 (training data):

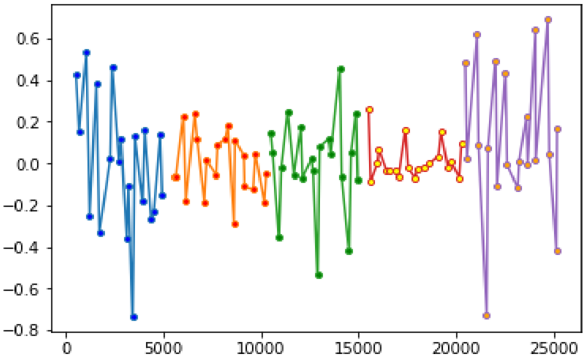


Histogram of score in S1 & S3(training data):



1.2 Detecting Pattern in Signals using LSTM

In this problem we were given a set of 30 signals and some points marked on it. From these signals we took 29 signals for training and 1 signal for testing. Then we used LSTM algorithm to predict the points. For LSTM we converted our single input data to a combination of 8 previous data, and also we converted our output in one-hot representation, in which 1′s are the desired points and 0′s are rest of the point. Parallel to this, as it was clearly seen from the signals that the points selected are the points where the amplitude of the signals changes drastically, so we also manually did the prediction, just by seeing the points of maximum amplitude change in order to compare the results of this algorithm to that of LSTM.



Combination of 5 signals with desired points marked on it

From the graph we can clearly see that it marked on points where amplitude changes

1.3 MNIST on Kaggle

We need to classify (in 10 classes) images for the ′new′ MNIST data set consisting of total 7948 images by implementing CNN or any other architecture keeping original MNIST training set for training and cross-validation.

2 Implementation and Results

2.1 Prediction of score in SX using Rest

A naive bayesian filter could predict pass fail in S3 accurately 80% of the times. This points out to one (or more) of following:

1. S3 overlaps with one or more out of rest skills. To find this prediction with this was

done using only 2 of them and then just 1.
2. Questions provided to students were not apt.

Results after using Bayesian filter and perceptron for classification. Perceptron seems to do bad on data (data is not linearly separable.)

Pred.	Skills used	Bayes	Perc.
S1	**S2, S3, S4**	56.72	55.09
S2	**S1, S3, S4**	60.42	59.01
S3	**S1, S2, S4**	79.72	64.11
S4	**S1, S2, S3**	60.81	55.09
S3	**S1, S2**	76.41	-
S3	**S2, S4**	65.69	-
S3	**S1, S4**	79.5	-
S3	**S1**	80.11	-
S3	**S2**	63.93	-
S3	**S4**	70.56	-

Results show that performance in S3 could be accurately 80% of the times by using S1 alone.

Future Prospects

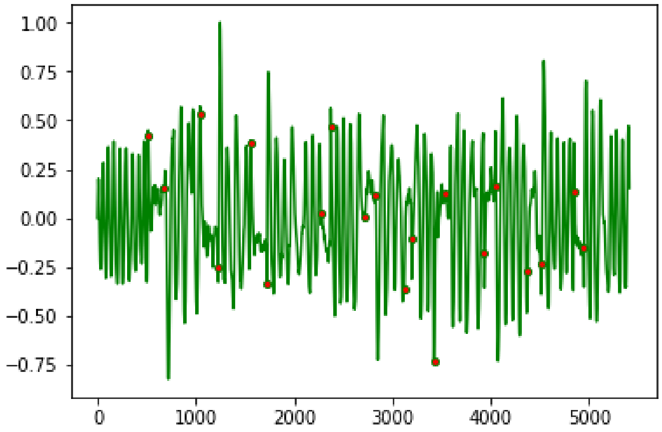
1. *The skill set this Olympiads tests contains 12 skills. As organization is in research phase, skill set can be improved by using such experimental results. Once the results of next rounds are out, we will extend this model to all 12 skills and improve on the set.*

2. *Once enough data is collected, such a model can help us make a recommendation system for students. Providing them with the content they are weak in can help them use their time effectively.*

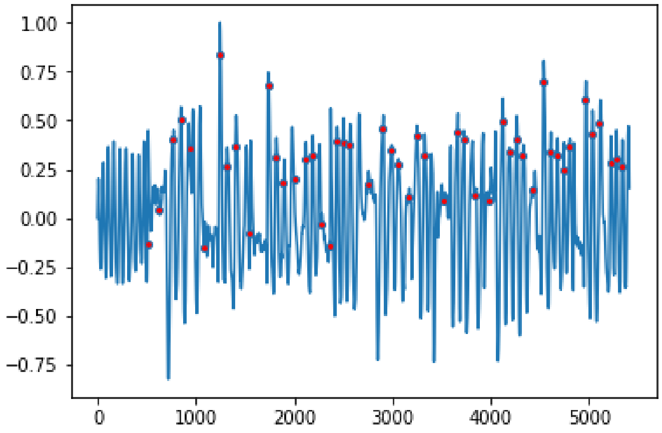
3. *Such experiments can help us get more insights in psychology, human behavior etc*

2.2 LSTM and other methods used on Signals

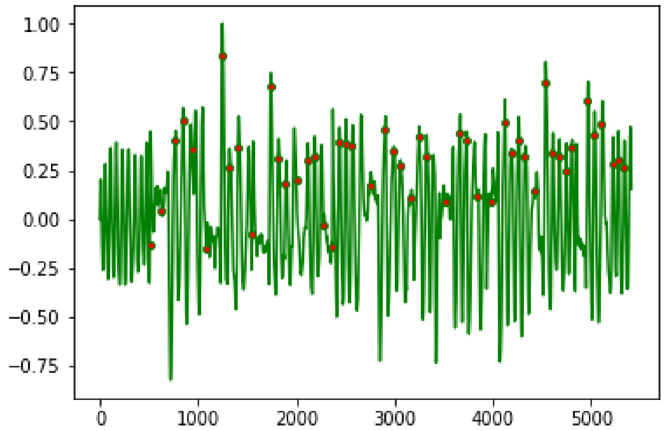
The first plot corresponds to the signal s1 with given output marked on it.
For the second plot, we marked the points where there was a sudden increase in magnitude although it gives the desired output(within 100 window) but it gave some false negative. It gave 30 outputs, where the number of required output was only 20 and 10 false alarms.
Third plot corresponds to the output after training a single layer of LSTM(Memory of 8 previous data) with 1 epoch on 29 signals, it took only 444 seconds, it gave 49 outputs including 20 desired one's(within 100 window) but 29 were false alarms unexpectedly greater than that in without applying any algorithm .
In the 4th plot we reduced our no. of input in LSTM from 8 to 5 and increased epochs to 7, this resulted in some better result, we got 35 points out of which 20 were desired one.



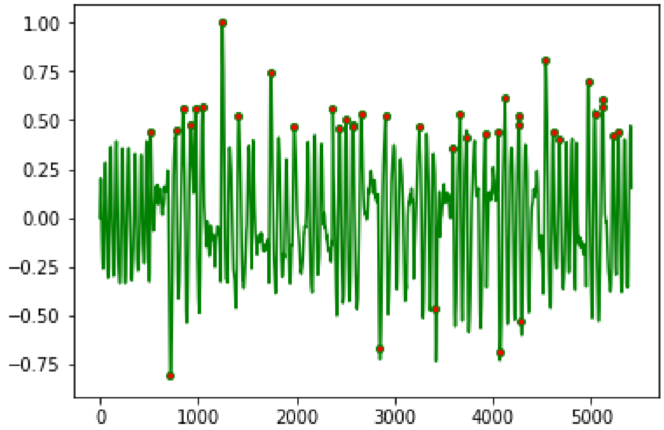
(Given) Signal 1 with output is plotted



Output points after magnitude calculation method calculations



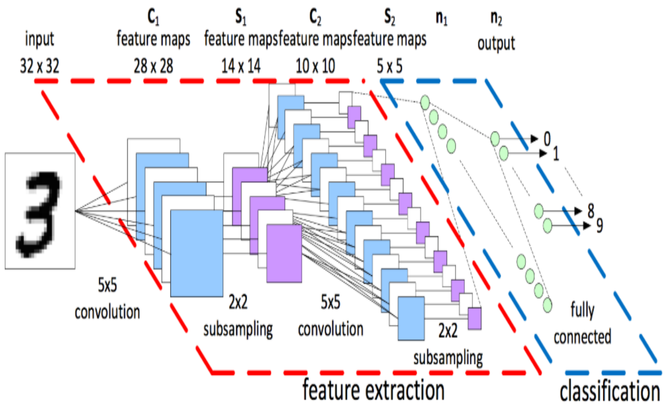
Output points after LSTM (Epoch 1)



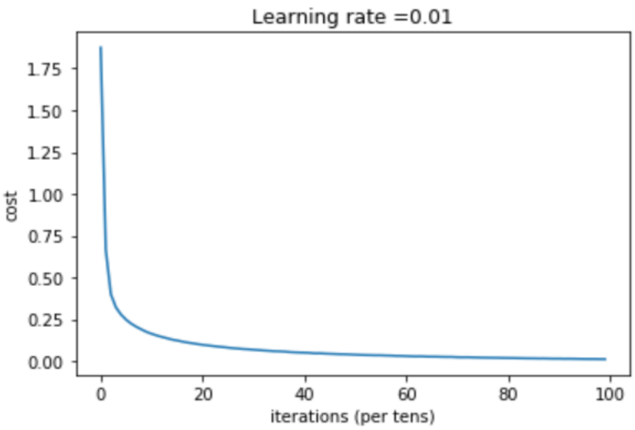
Output points after LSTM (Epoch 7)

2.3 MNIST on Kaggle

CNN based model was implemented to classify images, as simple neural network didn't result in good accuracy. Architecture for CNN model used is shown below:



Iteration vs value of cost function is shown:



An accuracy of 93.758 was achieved. Points which were kept in mind while implementing the same:

2.3.1

Model is trained for 100 epochs on a batch size of 128, 0.001 learning rate.

2.3.2

Layers used includes conv2d, maxpool, relu, flattening, fully connected layer at the end.