CS7015 DEEPLEARNING PROGRAMMING ASSIGNMENT 1

VIDYASREE VANKAM [IIITRKV 3rd year undergraduate ECE

21 September 2019

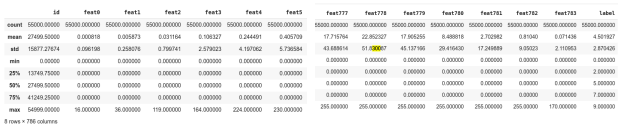
Contents

1 Kaggle Accuracy

Momentum optimization algorithm with lr=0.01,3hidden layers 300 neurons each with sigmoid activation crossentropy loss batch size 20 for 15 epochs give 0.85600 testscore for 50epochs 0.87733 testscore for 300 epochs give 0.87933 public score in kaggle. Momentum gives 87.933 accuracy in kaggle submission

2 Data Description

The dataset contains 786 entries(columns) out of which 2 columns are of id and label. Each image is of size 28*28 total 784 pixels of an image.Out of which 55000entries corresponding to train, 5000entries for validation and 10000entries for test data.



	id	feat0	feat1	feat2	feat3	feat4	feat5	feat777	feat778	feat779	feat780	feat781	feat782	feat783	label
count	55000.00000	55000.000000	55000.000000	55000.000000	55000.000000	55000.000000	55000.000000	55000.000000	55000.000000	55000.000000	55000.000000	55000.000000	55000.00000	55000.000000	55000.000000
mean	27499.50000	0.000818	0.005873	0.031164	0.106327	0.244491	0.405709	17.715764	22.852327	17.365255	8.488818	2.702982	0.81040	0.071436	4.501927
std	15877.27674	0.096198	0.258076	0.766741	2.579023	4.167062	5.736584	43.688614	51.[illegible]47	45.137166	29.416430	17.246869	9.05023	2.110663	2.879426
min	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
25%	13749.75000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	2.000000
50%	27499.50000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	5.000000
75%	41249.25000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	7.000000
max	54999.00000	16.000000	36.000000	119.000000	164.000000	224.000000	230.000000	255.000000	255.000000	255.000000	255.000000	255.000000	255.00000	170.000000	9.000000

8 rows × 786 columns

Figure 1: description of data

3 DataNormalization

DataNormalization is important as there are some features corresponding to larger value.So we normalize them by dividing every value by 255 to keep all the values in between 0 and 1 like min max normalization.

4 Hilbert Initialization

We multiply randomly initialized weights with hilbert initialization.

5 Experiment 1:Hidden layers performance on our data

Adam optimizer with learning rate 0.01 performs better as it is converging faster.Single hidden layer with 100 neurons give better training and validation accuracy with test score of 86.6 percent. But as the hidden layers increase we can learn more features.

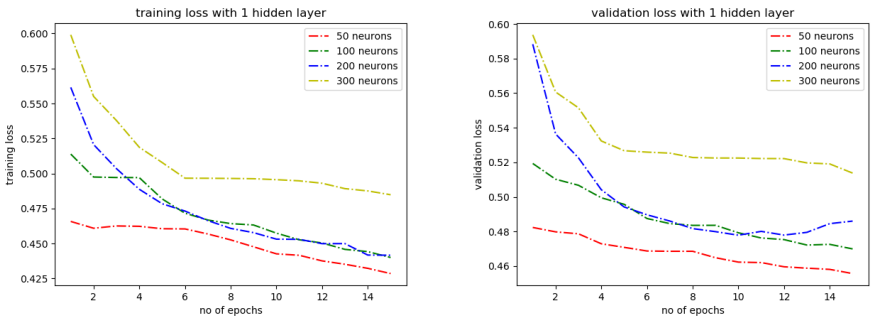


Figure 2: 1 hidden layer with various size of neurons depicting trainloss and validation loss respectively

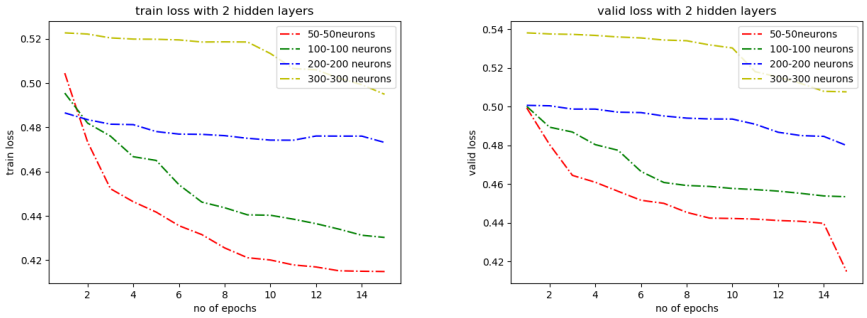


Figure 3: 2 hidden layers with various size of neurons depicting trainloss and validation loss respectively

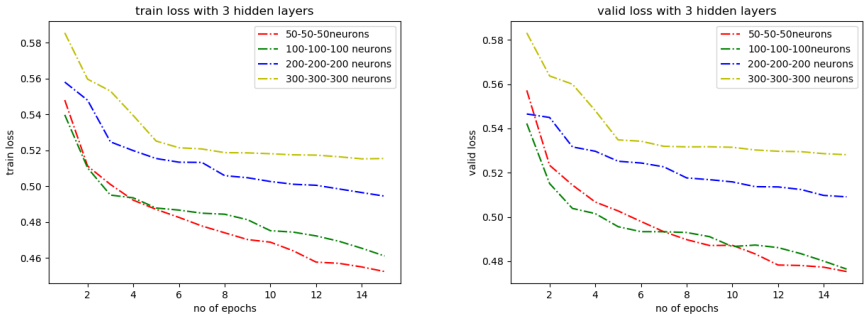


Figure 4: 3 hidden layers with various size of neurons depicting trainloss and validation loss respectively

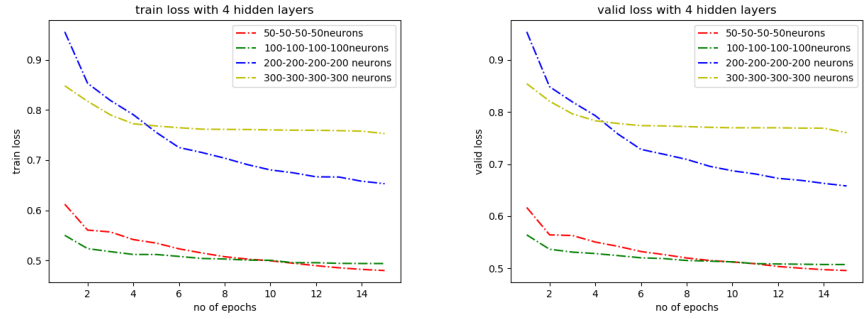


Figure 5: 4 hidden layers with various size of neurons depicting trainloss and validation loss respectively

6 Experiment 2:Performance of Various Optimization algorithms

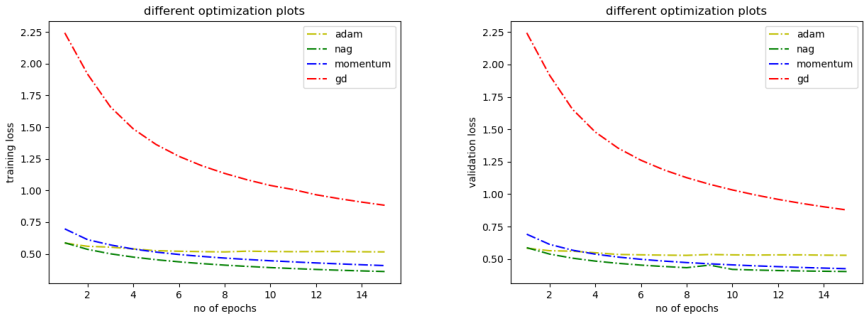


Figure 6: depicting trainloss and validation loss respectively for various optimization algorithms

NAG optimization algorithm gave validation accuracy better.After that momentum optimization algorithm give good validation accuracy of 86.76 percent for 15 epochs. For 50 epochs Momentum gave validation accuracy 87.2percent and for 300 epochs giving accuracy of 87.98 on valid and 98.85 percent of training accuracy. Adam optimizer gives better decay rate. GD optimizer has more loss compared to other optimization algorithms.

7 Experiment3:Different activation functions

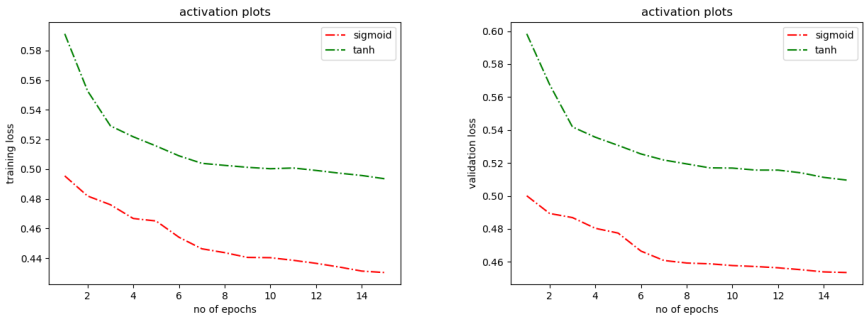


Figure 7: depicting trainloss and validation loss respectively for different activation functions

8 Experiment4:Comparison of Different loss functions

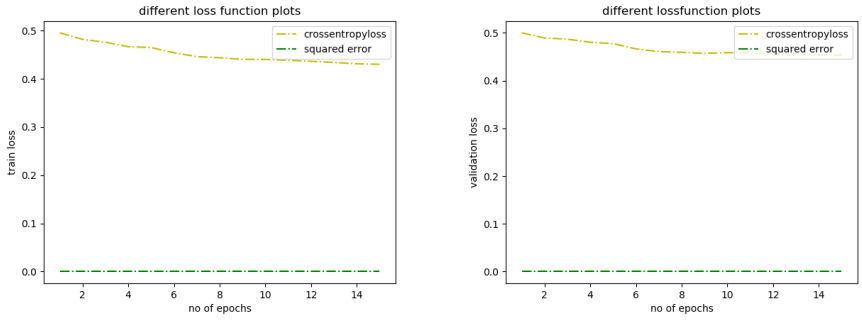


Figure 8: depicting trainloss and validation loss respectively for different loss functions

Cross entropy loss function gives better accuracy compared to squared error because here we want to classify image into 1 of some k classes.