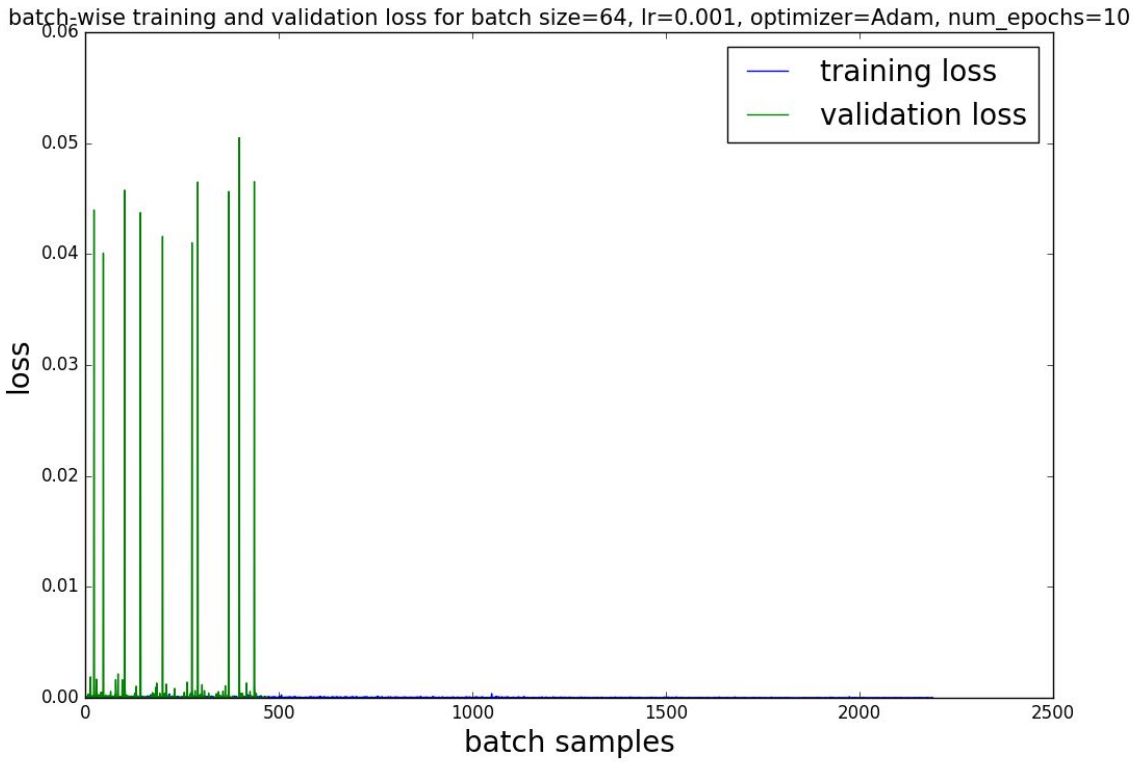
Introduction :

For this project Tejasvi and me collaborated to work on hand gestures recognition data taken from kaggle. I worked on pytorch to create a model to train, validate and test on hand gestures dataset.

Description of the individual work :

We developed a basic Neural Network architecture which worked on the given dataset but accuracy was not that good. Then I started working on improving the model by developing a script, which used multiple permutation of optimizers, batch sizes, epochs, learning rate. This script helped me determine the best combination of parameters. But I thought the model was overfitted as shown by this graph..

Description of portion of work in detail :



This graph showed me that the model was overfit. Then we used validation dataset, to do early stopping on our data. This prevented overfitting. Also, to choose best parameters I used following 54 different models to choose most optimum parameters.

| batch_size | learning_rate | optimizer_method | num_epochs | time | accuracy |
|---|---|---|---|---|---|
| 128 | 0.1 | Adam | 5 | 6.93 | 92.4 |
| 128 | 0.1 | Adam | 10 | 6.96 | 97.8 |
| 128 | 0.1 | Adam | 2 | 6.93 | 94.73 |
| 128 | 0.1 | Adadelta | 2 | 7.06 | 98.53 |
| 128 | 0.1 | Adadelta | 5 | 7.05 | 99.8 |
| 128 | 0.1 | Adadelta | 10 | 7.06 | 99.87 |
| 128 | 0.01 | Adam | 10 | 7.01 | 99.93 |
| 128 | 0.01 | Adam | 2 | 7.02 | 78.2 |
| 128 | 0.01 | Adam | 5 | 7 | 99.8 |
| 128 | 0.01 | Adadelta | 2 | 7.06 | 62.67 |
| 128 | 0.01 | Adadelta | 5 | 7.1 | 89.8 |
| 128 | 0.01 | Adadelta | 10 | 7.07 | 98.73 |
| 128 | 0.001 | Adam | 2 | 7.01 | 99.83 |
| 128 | 0.001 | Adam | 5 | 7.02 | 99.93 |
| 128 | 0.001 | Adam | 10 | 7.05 | 99.93 |
| 128 | 0.001 | Adadelta | 2 | 7.09 | 13.17 |
| 128 | 0.001 | Adadelta | 10 | 7.1 | 51.43 |
| 128 | 0.001 | Adadelta | 5 | 7.07 | 32.4 |
| 64 | 0.1 | Adam | 10 | 7.44 | 98.47 |
| 64 | 0.1 | Adam | 2 | 7.4 | 96.23 |

| | | | | | |
|---|---|---|---|---|---|
| 64 | 0.1 | Adam | 5 | 7.4 | 85.37 |
| 64 | 0.1 | Adadelta | 10 | 7.61 | 99.93 |
| 64 | 0.1 | Adadelta | 5 | 7.63 | 99.87 |
| 64 | 0.1 | Adadelta | 2 | 7.54 | 99.43 |
| 64 | 0.01 | Adam | 2 | 7.55 | 99.1 |
| 64 | 0.01 | Adam | 5 | 7.53 | 99.93 |
| 64 | 0.01 | Adam | 10 | 7.68 | 99.93 |
| 64 | 0.01 | Adadelta | 2 | 7.6 | 70.2 |
| 64 | 0.01 | Adadelta | 5 | 7.63 | 95.13 |
| 64 | 0.01 | Adadelta | 10 | 7.68 | 99.47 |
| 64 | 0.001 | Adam | 2 | 7.57 | 99.93 |
| 64 | 0.001 | Adam | 5 | 7.65 | 99.93 |
| 64 | 0.001 | Adam | 10 | 7.69 | 99.9 |
| 64 | 0.001 | Adadelta | 10 | 7.67 | 57.9 |
| 64 | 0.001 | Adadelta | 2 | 7.68 | 14.53 |
| 64 | 0.001 | Adadelta | 5 | 7.71 | 36.47 |
| 32 | 0.1 | Adam | 5 | 8.65 | 75.57 |
| 32 | 0.1 | Adam | 10 | 8.81 | 75.67 |
| 32 | 0.1 | Adam | 2 | 8.89 | 87.6 |
| 32 | 0.1 | Adadelta | 5 | 9.05 | 99.9 |
| 32 | 0.1 | Adadelta | 10 | 9.12 | 99.9 |
| 32 | 0.1 | Adadelta | 2 | 9.11 | 99.53 |
| 32 | 0.01 | Adam | 2 | 8.93 | 98.5 |
| 32 | 0.01 | Adam | 10 | 8.93 | 99.9 |
| 32 | 0.01 | Adam | 5 | 8.86 | 91.63 |

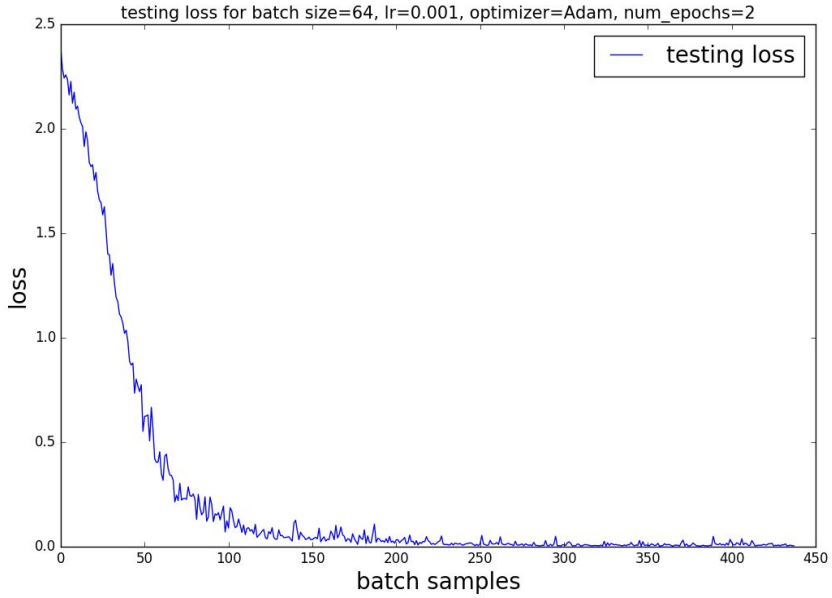
| | | | | | |
|---|---|---|---|---|---|
| 32 | 0.01 | Adadelta | 5 | 9.12 | 97.97 |
| 32 | 0.01 | Adadelta | 2 | 9.18 | 74.23 |
| 32 | 0.01 | Adadelta | 10 | 9.15 | 99.7 |
| 32 | 0.001 | Adam | 2 | 9.1 | 99.8 |
| 32 | 0.001 | Adam | 5 | 9.06 | 99.97 |
| 32 | 0.001 | Adam | 10 | 8.89 | 99.97 |
| 32 | 0.001 | Adadelta | 5 | 9.24 | 47.23 |
| 32 | 0.001 | Adadelta | 10 | 9.33 | 75.23 |
| 32 | 0.001 | Adadelta | 2 | 9.38 | 19.97 |

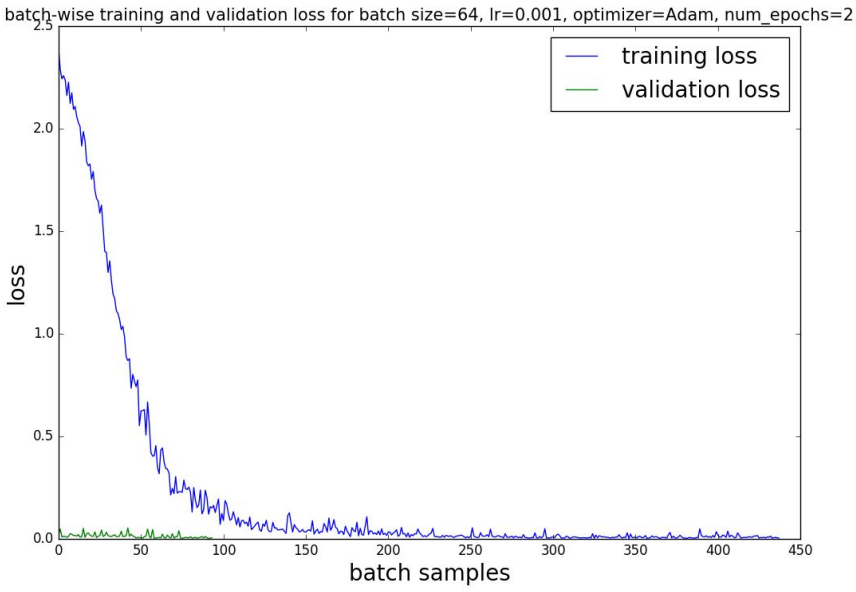
Next task was to create a docker image to upload to ECR. Tejasvi created the predict file, which takes a single image and classifies it. Creating docker images was difficult, as this was my first time working with docker image. Docker was exhausting, as I was getting error for dependencies, which took most of my time. After docker images was created, I uploaded it to ECR, from where tejasvi used that to create containers.

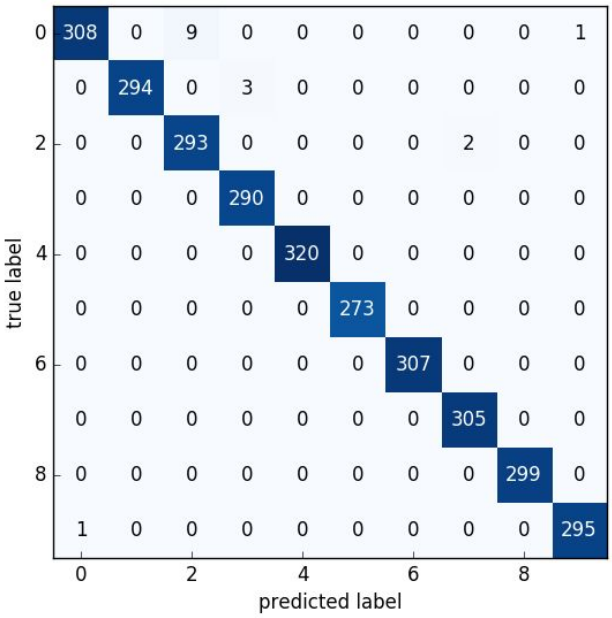
Results, summary and conclusions :

For our data set we have found best accuracy/ f-score for below parameters:

| batch size | Learning rate | optimizer method | num epochs | time((sec) | accuracy |
|---|---|---|---|---|---|
| 64 | 0.001 | Adam | 2 | 7.26 | 99.47 |


testing loss for batch size=64, lr=0.001, optimizer=Adam, num_epochs=2
testing loss
loss
batch samples
0.0
0.5
1.0
1.5
2.0
2.5
0
50
100
150
200
250
300
350
400
450


batch-wise training and validation loss for batch size=64, lr=0.001, optimizer=Adam, num_epochs=2
training loss
validation loss
loss
batch samples
0.0
0.5
1.0
1.5
2.0
2.5
0
50
100
150
200
250
300
350
400
450


0 308 0 9 0 0 0 0 0 0 1
0 294 0 3 0 0 0 0 0 0
2 0 0 293 0 0 0 0 2 0 0
0 0 0 290 0 0 0 0 0 0
4 0 0 0 0 320 0 0 0 0 0
0 0 0 0 0 273 0 0 0 0
6 0 0 0 0 0 0 307 0 0 0
0 0 0 0 0 0 0 305 0 0
8 0 0 0 0 0 0 0 0 299 0
1 0 0 0 0 0 0 0 0 0 295
0 2 4 6 8
true label
predicted label