

DL Assignment 1 Report

Authors : Jaitesh CS17B021, Mooizz CS17B034

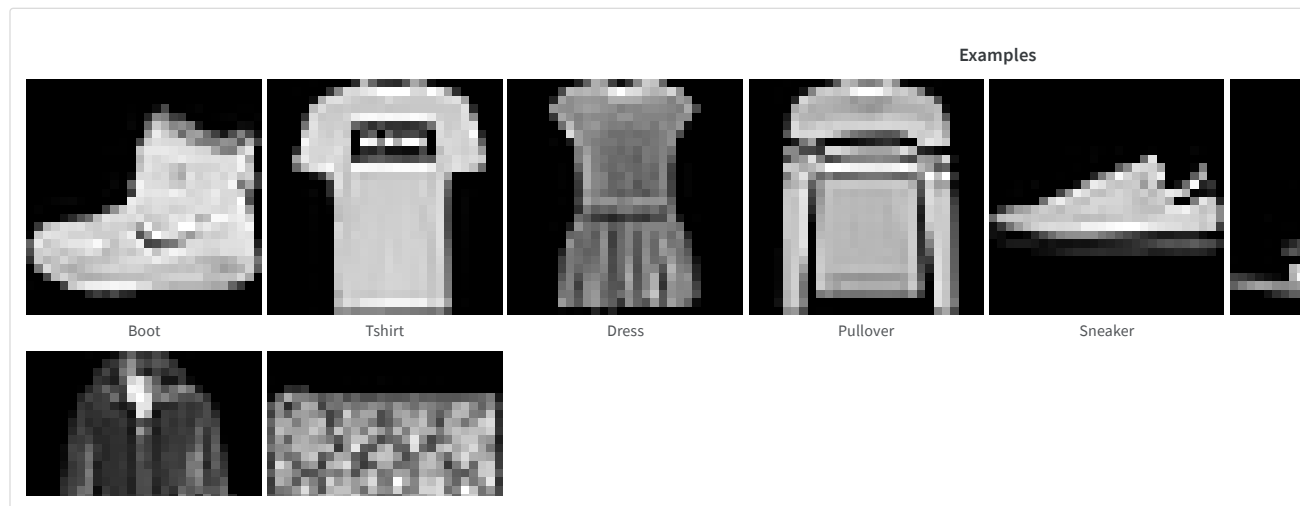
Mooizz A

please visit the below link to view the report clearly

Report Link : <https://wandb.ai/mooizz/feedforwardfashion/reports/DL-Assignment-1-Report--Vmlldzo1MzY0MjU>

Question 1

Sample Image for each class. Dataset was normalized for better training.

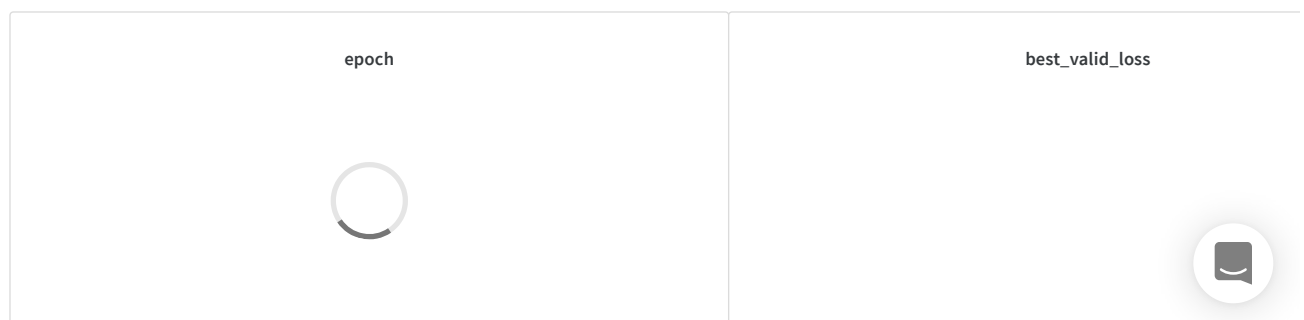


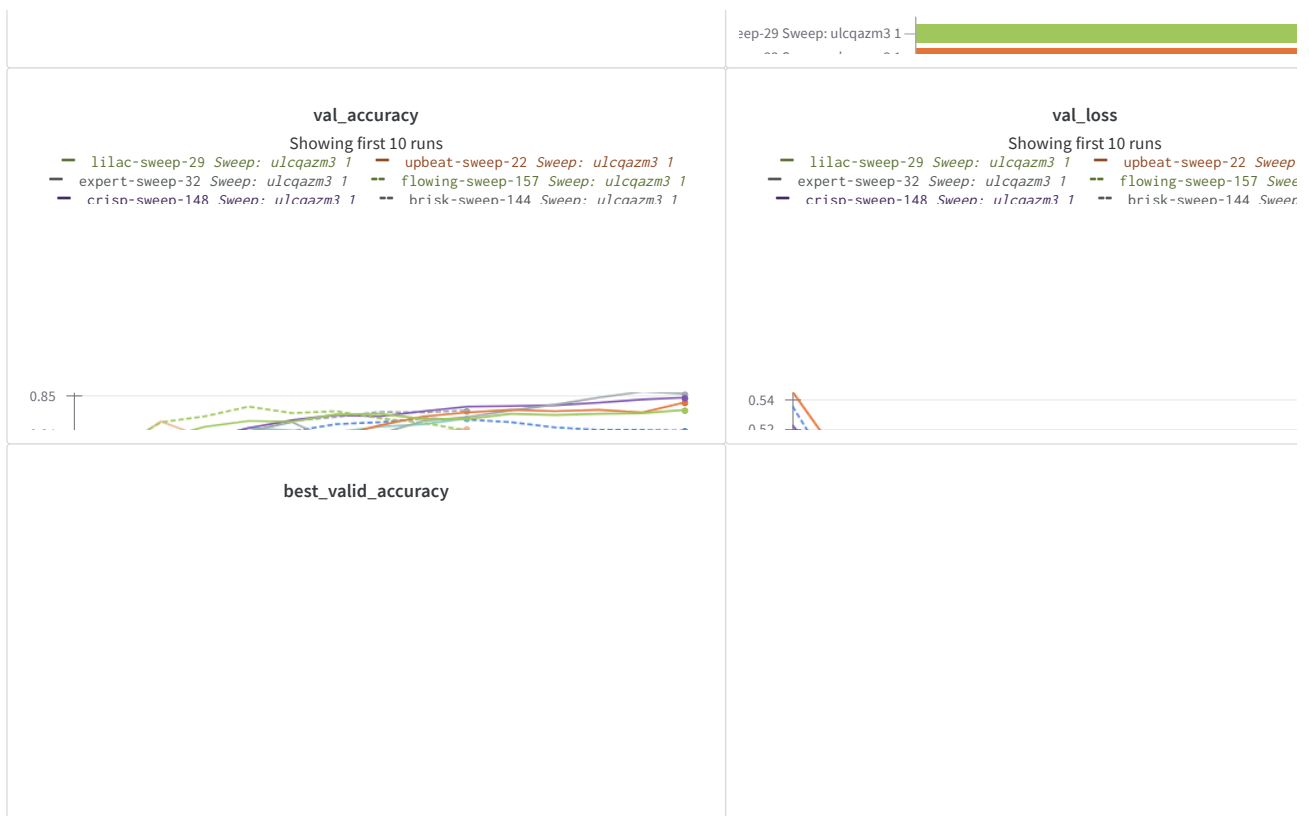
Question 4

We first tried random Hyper Parameter Search using wandb sweep. Realized that it is not feasible as there are a large number of Hyper Parameters

So, Bayesian Hyper Parameter Search was used (inbuilt wandb implementation). This method tries to model the relation between the Hyper Parameters and the performance metric (Here it's validation accuracy), based on the previous runs. Which allows the search to be much more efficient.

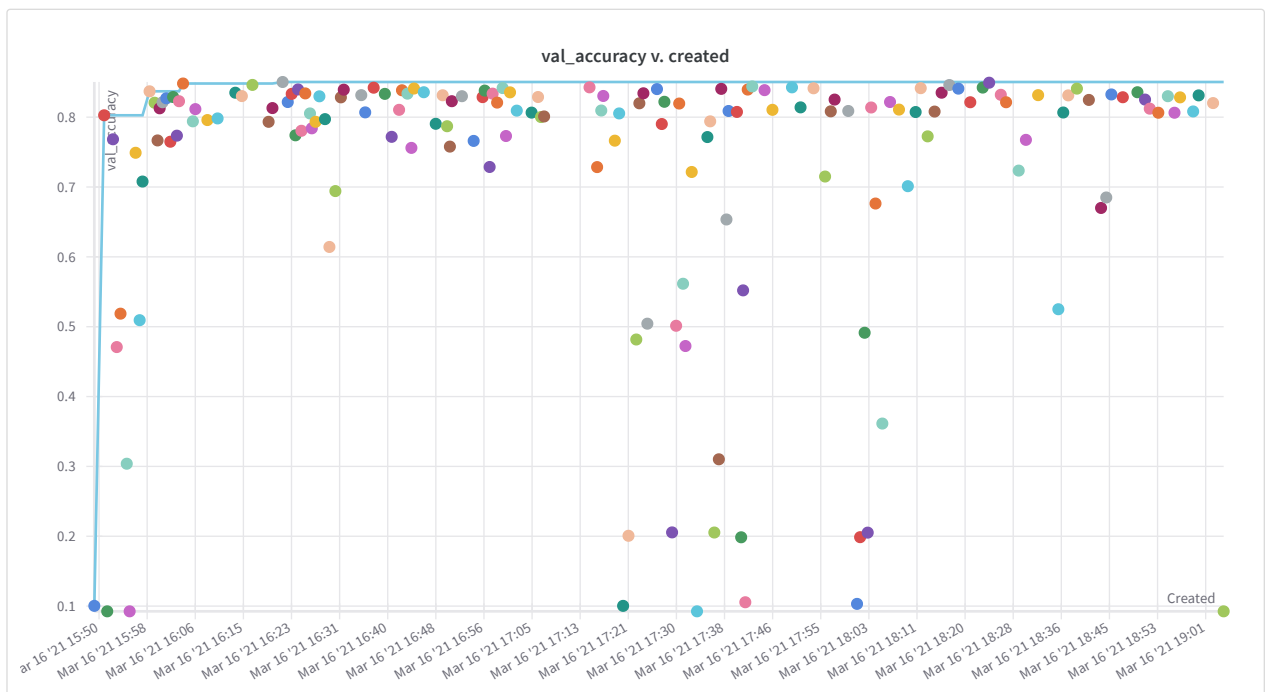
The sweep contains 170 runs and the best validation accuracy among all epochs for each run, along with the required metrics was recorded.



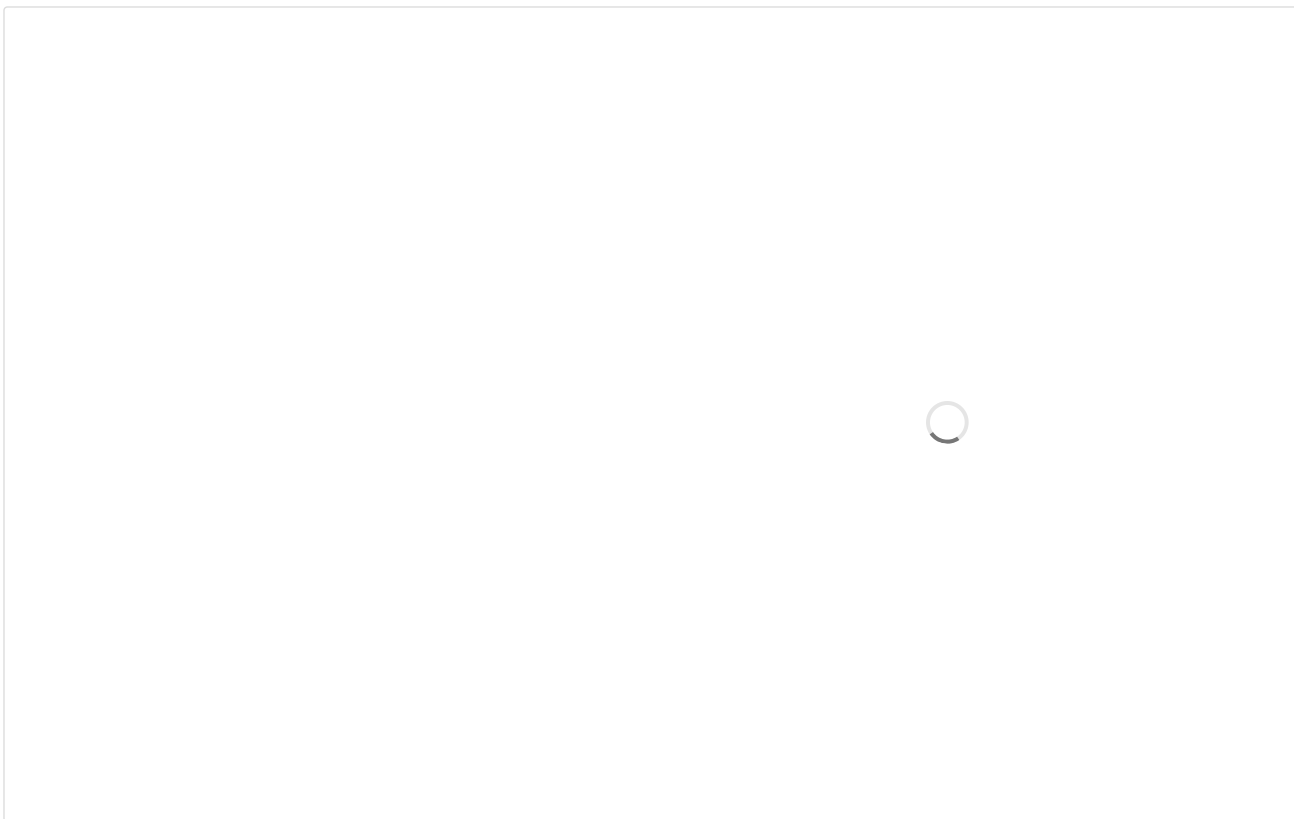
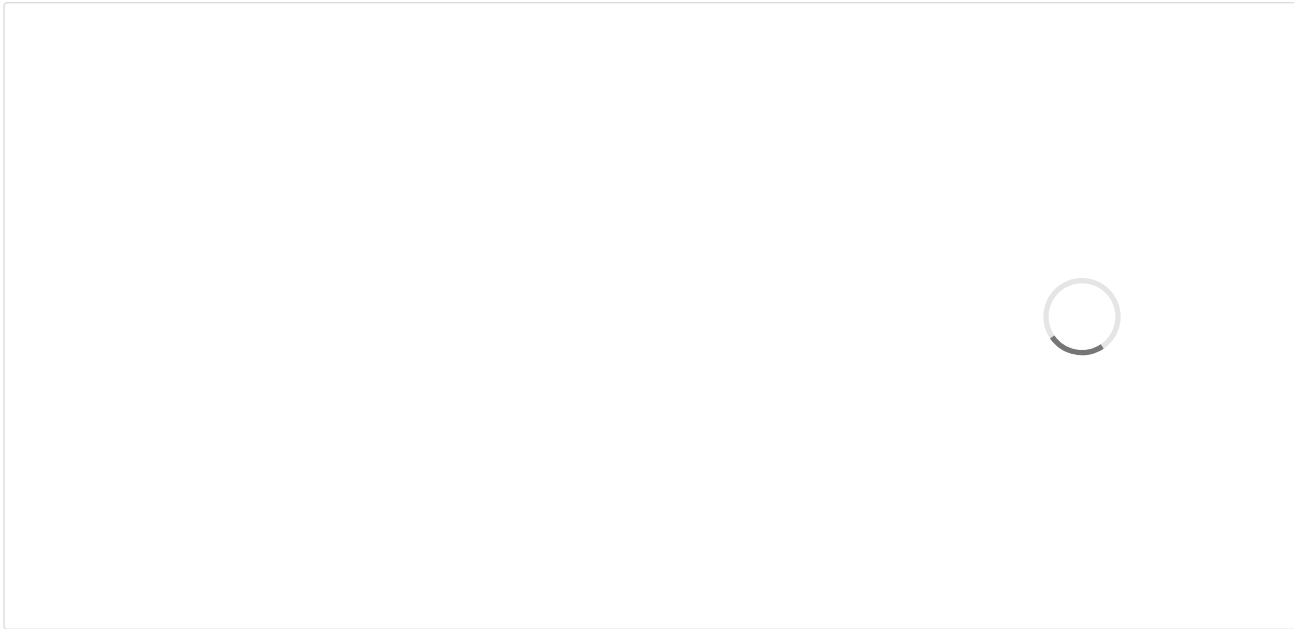


Question 5

Plot of accuracy on validation set for various runs(models)



Question 6



Inferences

- From the above Correlation Summary, its clear that **Xavier** weight initialization works much better than **random** weight initialization.



- Among the three activation functions tanh shows high correlation with validation accuracy. This is evident when we inspect the runs with accuracy greater than 80% in the scatter plot.
- The accuracy is observed to decrease when number of hidden layers is increased, this is probably due to overfitting.
- A significant improvement in accuracy is observed when hidden layer size is increased from 64 to 128 but when trained with hidden layer width of 256 there was no significant improvement in accuracy.
- Accuracy is low when the batch size is too small(16). This may be because smaller batch sizes lead to larger number of updates which sometimes result in convergence to poor local optima.
- **nadam, adam, rmsprop** optimizers seem to have better accuracy when compared to others

Sweep Configuration to find the best hyperparameter set is as follows:

```
epochs: [10, 15]
num_hidden_layers : [3]
hidden_layer_size: [128, 256]
weight_decay: [0, 0.5]
learning_rate: [1e-4]
optimizer: ['nadam', 'rmsprop', 'adam']
batch_size: [64]
weight_init: ['Xavier']
activation: ['tanh']
```

Parallel Plot for the above configuration

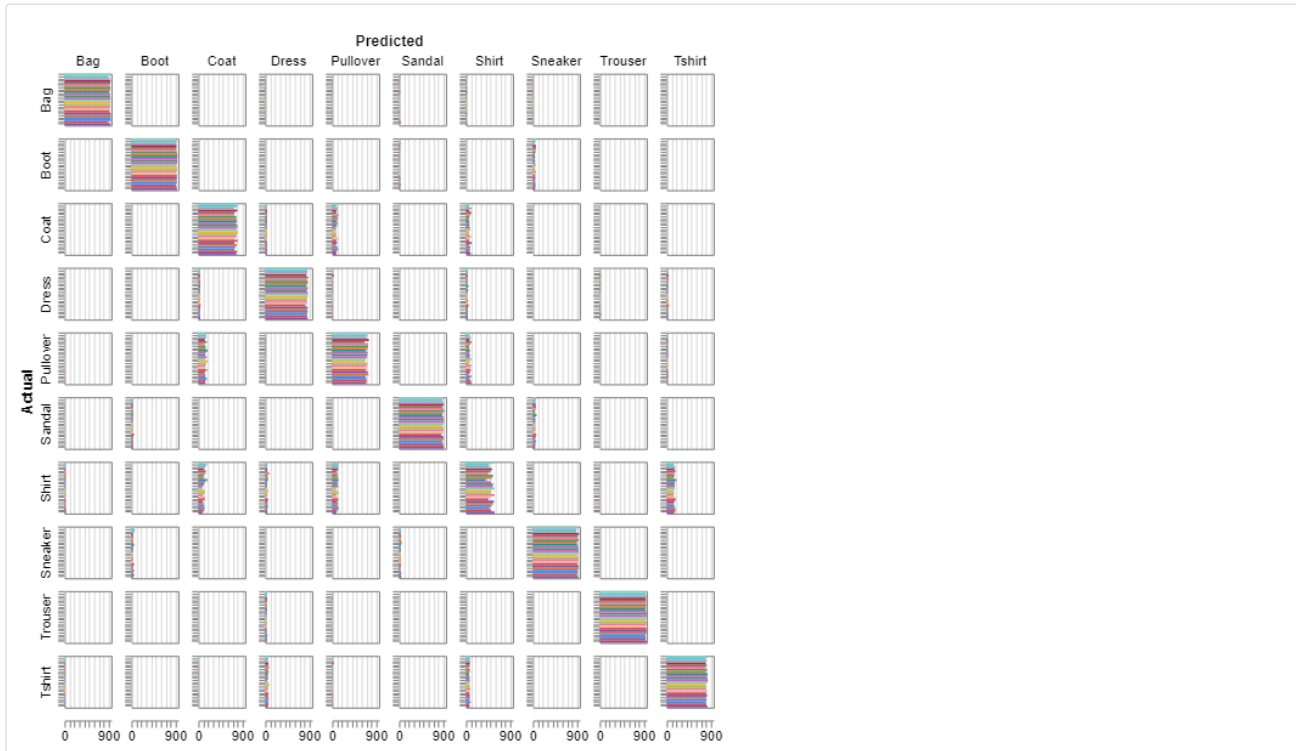


```
Best hyperparameter set from the above runs( stellar-sweep-6 )
epochs: 15
num_hidden_layers : 3
hidden_layer_size: 256
weight_decay: 0
learning_rate: 1e-4
optimizer: 'adam'
batch_size: 64
weight_init: 'Xavier'
activation: 'tanh'
```

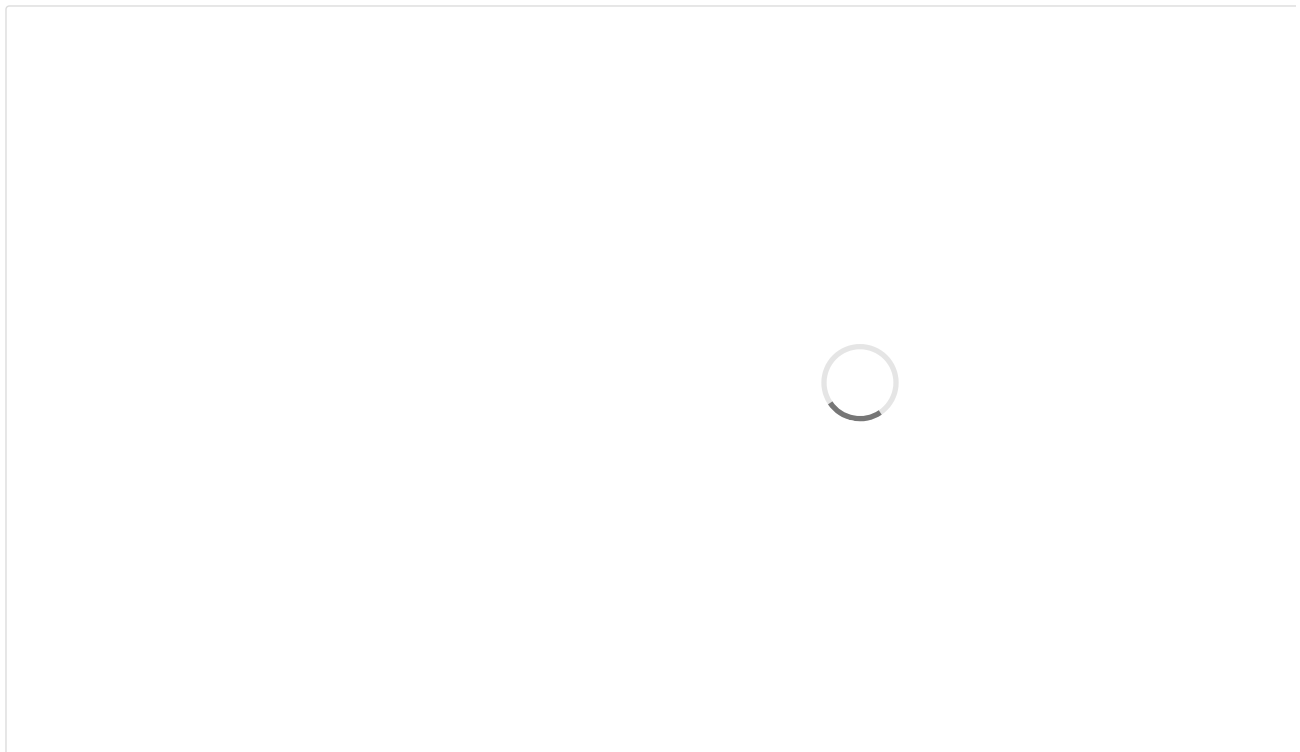
Question 7



Confusion matrix for various runs in the Sweep configuration mentioned above



Confusion matrix for the best hyperparameter set



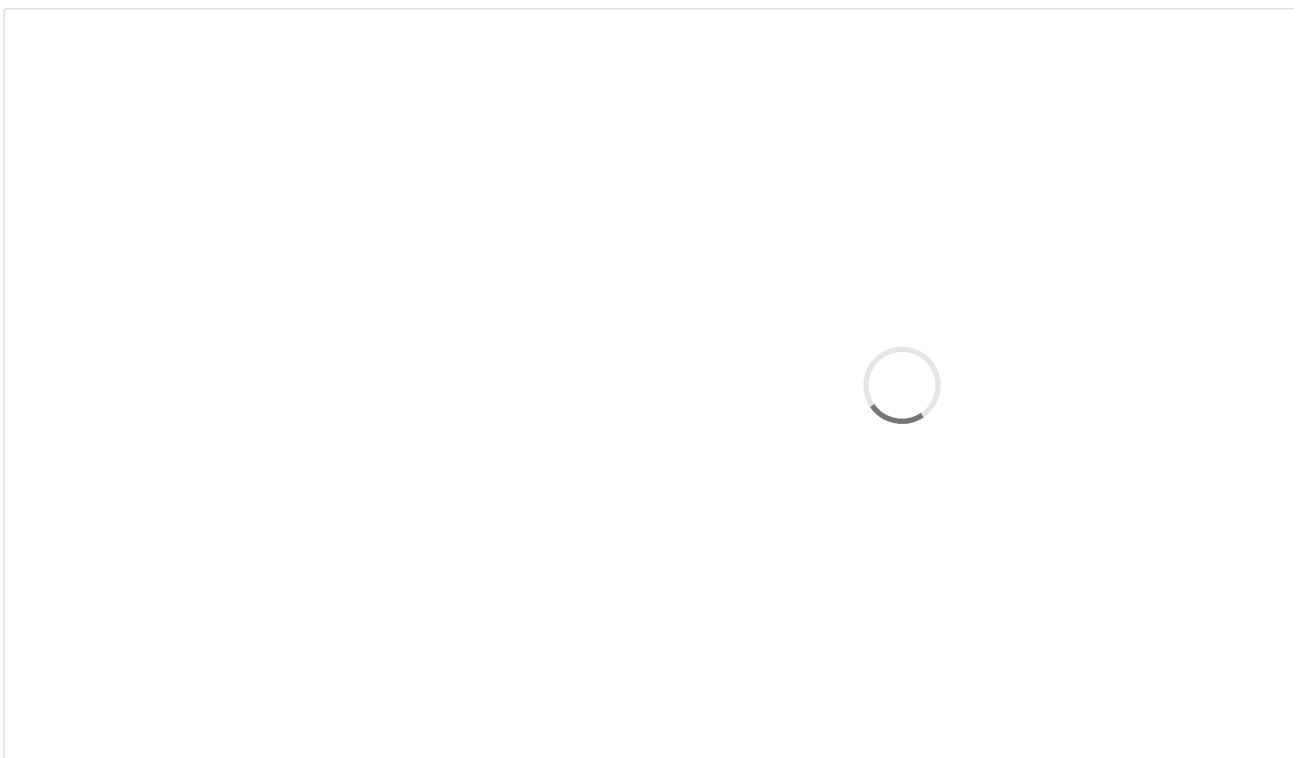
The best model gives 84.4% accuracy on test data.

Question 8





From the above plot , observe that the validation accuracy decreased as the model is trained for square loss function.



It is clear from the above confusion matrix there are more misclassifications in the runs(violet and green) where square loss function is used. So, we can infer that the cross entropy error is a better metric than Square error.

Question 9

Github link: <https://github.com/0-5-blood-prince/FeedForwardNetwork>

Question 10

From the above experimentation, It is clear that the adam optimizer, activations tanh and relu perform best and have high correlation. So we decided to try the below configuration where all the parameters are same as in the best hyperparameter set except the activation functions.



```
epochs: [10]
num_hidden_layers : [3]
hidden_layer_size: [128]
weight_decay: [0]
learning_rate: [1e-4]
optimizer :['adam']
batch_size: [64]
weight_init: ['Xavier']
activation: ['tanh','sigmoid','relu']
```

test_accuracy



Self Declaration

CS17B021


- Backward Propagation framework
- Hyperparameter tuning
- Inferences and sweep ulcqzm3, l934ae3m execution



- Readme file
- sweep an69q1wj execution, fixed errors

CS17B034

- Q1 (dataset.py)
- Structure of Neural Net(FeedForward Neural Net and fit)
- Optimizers
- sweep ulcqazm3 setup
- Inferences and sweep l934ae3m setup
- sweep bm5wopy3
- sweep an69q1wj setup

Created with  on Weights & Biases.

<https://wandb.ai/mooizz/feedforwardfashion/reports/DL-Assignment-1-Report--Vmldzo1MzY0MjU>

