COL 341
Assignment 2.1 Report

Ishaan Watts
2019PH10629

Part C

Performed hyper-parameter tuning on the given architecture using hpc. Trained each model for 270secs and plotted graphs as well as tables.

Architecture --> A1. [512, 256, 128, 64, 46]
A2. [256, 46]

Tuned parameters:

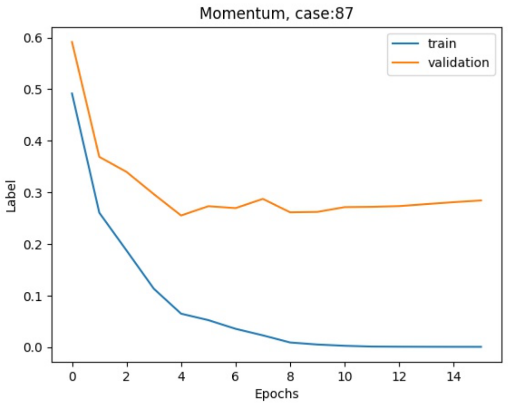
1. Batch Size = 32, 64, 128, 256
2. Learning Rate = 0.001, 0.01, 0.1, 1
3. Learning Rate Strategy = 0 (Constant), 1 (Adaptive)
4. Activation = 0 (log sigmoid), 1 (tanh), 2 (relu)
5. Optimizer = 0 (Momentum), 1 (Nesterov), 2 (RMSprop), 3 (Adam), 4 (Nadam)

Total 96 cases for each Optimizer i.e., 480 models trained.

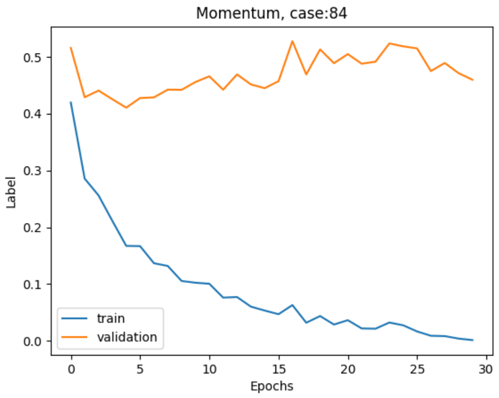
a) Momentum

Best params --> A1
Batch size = 256; Learning Rate = 0.1, Adaptive; Activation = relu, Epochs = 16
Test Accuracy = 95.0217, Train Loss = 0.000852, Validation Loss = 0.284479

Best params --> A2
Batch size = 128; Learning Rate = 0.1, Constant; Activation = relu, Epochs = 30
Test Accuracy = 93.6739, Train Loss = 0.001142, Validation Loss = 0.459926



A1



A2

b) Nesterov

Best params --> A1

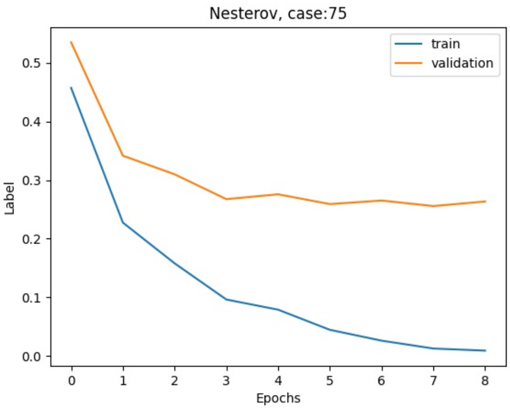
Batch size = 64; Learning Rate = 0.01, Adaptive; Activation = relu, Epochs = 9

Test Accuracy = 94.7609, Train Loss = 0.008799, Validation Loss = 0.263447

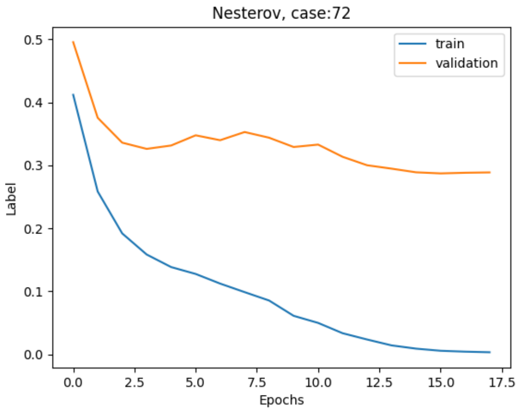
Best params --> A2

Batch size = 32; Learning Rate = 0.01, Constant; Activation = relu, Epochs = 18

Test Accuracy = 94.3913, Train Loss = 0.003292, Validation Loss = 0.288759



A1



A2

c) RMSprop

Best params --> A1

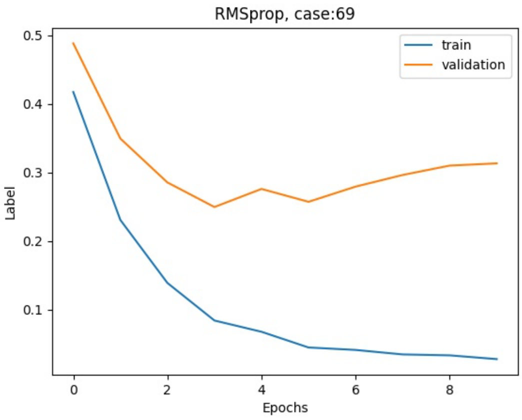
Batch size = 128; Learning Rate = 0.001, Adaptive; Activation = relu, Epoch = 10

Test Accuracy = 93.6552, Train Loss = 0.027702, Validation Loss = 0.313318

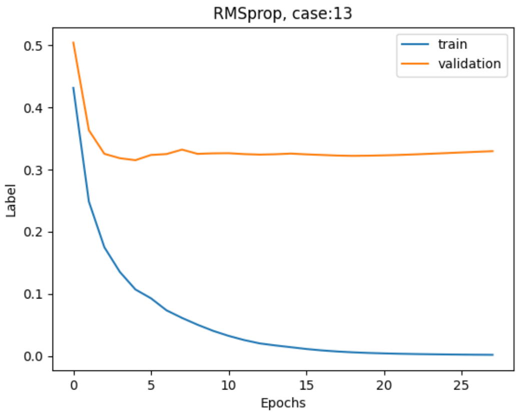
Best params --> A2

Batch size= 128; Learning Rate= 0.01, Adaptive; Activation= sigmoid, Epoch= 28

Test Accuracy = 92.5870, Train Loss = 0.002058, Validation Loss = 0.329633



A1



A2

d) Adam

Best params --> A1

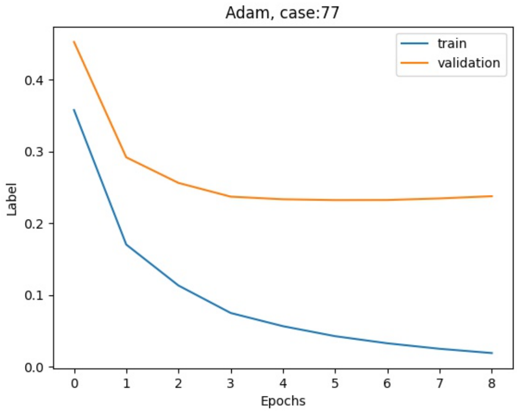
Batch size = 128; Learning Rate = 0.01, Adaptive; Activation = relu, Epochs = 9

Test Accuracy = 94.1739, Train Loss = 0.019306, Validation Loss = 0.237861

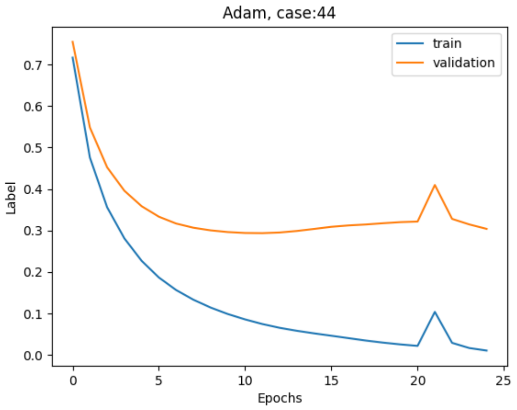
Best params --> A2

Batch size = 128; Learning Rate = 0.01, Constant; Activation = tanh, Epochs = 25

Test Accuracy = 92.2391, Train Loss = 0.010848, Validation Loss = 0.304060



A1



A2

e) Nadam

Best params --> A1

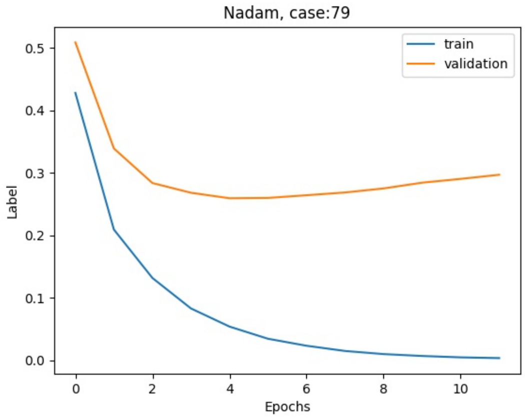
Batch size = 256; Learning Rate = 0.01, Adaptive; Activation = relu, Epochs = 12

Test Accuracy = 94.4783, Train Loss = 0.003425, Validation Loss = 0.296990

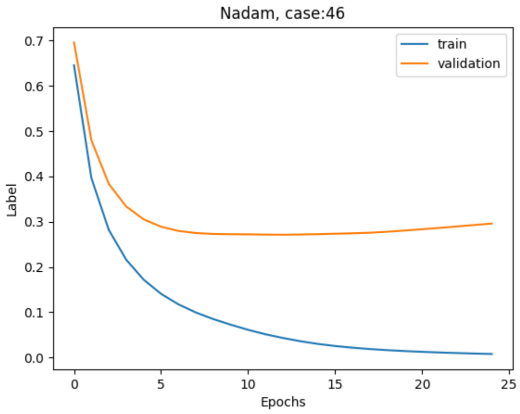
Best params --> A2

Batch size = 256; Learning Rate = 0.01, Adaptive; Activation = tanh, Epochs = 25

Test Accuracy = 93.6052, Train Loss = 0.008066, Validation Loss = 0.296199



A1



A2

Observations:

1. Adaptive learning strategy gave better results.
2. Ideal learning rate varies from 0.01-0.1 for Momentum, Nesterov; 0.001 for RMSprop and 0.01 for Adam; Nadam.
3. Batch sizes which gave good results: 128,256
4. Log sigmoid was stable while tanh, relu were little unstable but gave higher accuracy and lower loss.

Result:

Minimum Training Loss in Momentum

Final params:

A1 --> [512, 256, 128 ,64, 46]
Batch size = 256; Learning Rate = 0.1, Adaptive; Activation = relu, Epochs = 16
Test Accuracy = 95.0217, Train Loss = 0.000852, Validation Loss = 0.284479
Optimizer = Momentum

A2 --> [256, 46]
Batch size = 128; Learning Rate = 0.1, Constant; Activation = relu, Epochs = 30
Test Accuracy = 93.6739, Train Loss = 0.001142, Validation Loss = 0.459926
Optimizer = Momentum

Epochs in 270s may vary due to speed of computer

Part D

Performed hyper-parameter tuning on different architectures on hpc.
Trained each model for 270secs and plotted graphs as well as tables.
Architectures Used -->

a) [512, 256, 128, 64, 46]
b) [256, 46]
c) [512, 256, 46]
d) [256, 128, 46]
e) [512, 256, 128, 46]

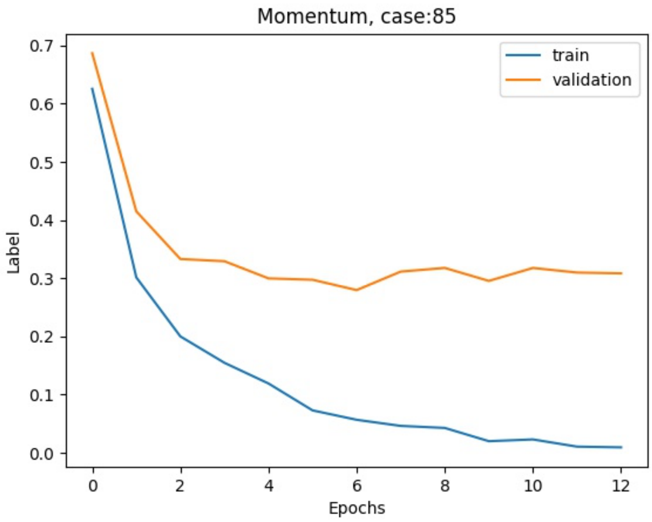
Tuned parameters:
1. Batch Size = 64, 128, 256
2. Learning Rate = 0.001, 0.01, 0.1
3. Learning Rate Strategy = 0 (Constant), 1 (Adaptive)
4. Activation = 0 (log sigmoid), 1 (tanh), 2 (relu)
5. Optimizer = 0 (Momentum), 1 (Nesterov), 2 (RMSprop), 3 (Adam), 4 (Nadam)

Total 210 cases for each Architecture i.e., 1050 models trained.

a) Architecture --> [512, 256, 128, 64, 46]

5 Layered architecture

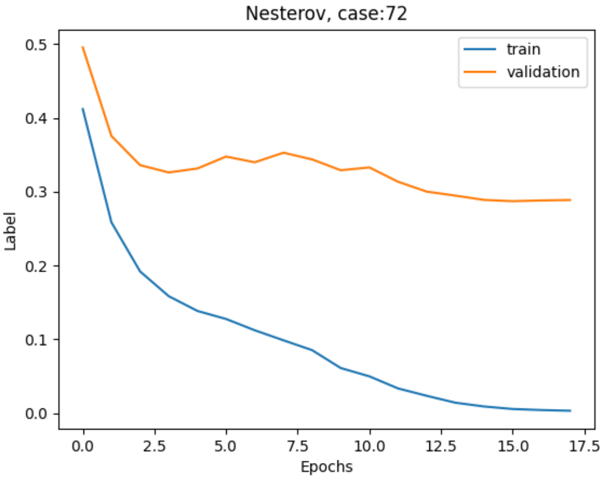
Best Params -->
Optimizer=Momentum; Batch=128;Learning Rate=0.1,Adaptive;Activation=relu
Epochs = 13; Test Accuracy = 95.0217



b) Architecture --> [256, 46]

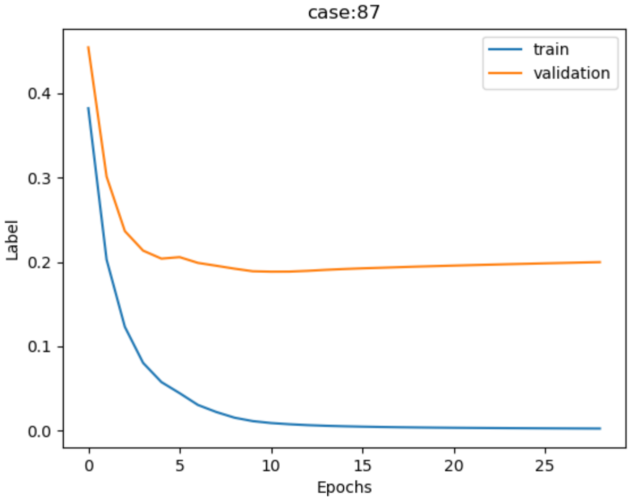
2 Layered architecture

Best Params -->
Optimizer = Nesterov; Batch = 32; Learning Rate = 0.01, Constant;
Activation = relu; Epochs = 18; Test Accuracy = 94.3913



c) Architecture --> [512, 256, 46]

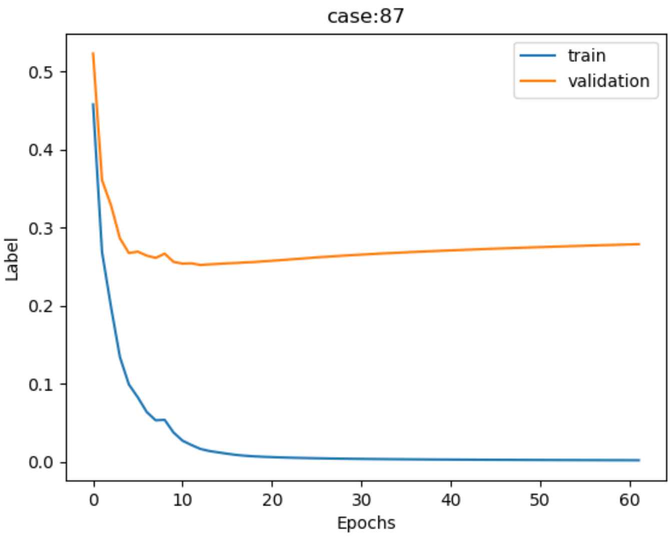
3 Layered architecture



Best Params -->
Optimizer = Momentum; Batch = 256; Learning Rate = 0.1, Adaptive;
Activation = relu; Epochs = 29; Test Accuracy = 95.6957

d) Architecture --> [256, 128, 46]

3 Layered architecture

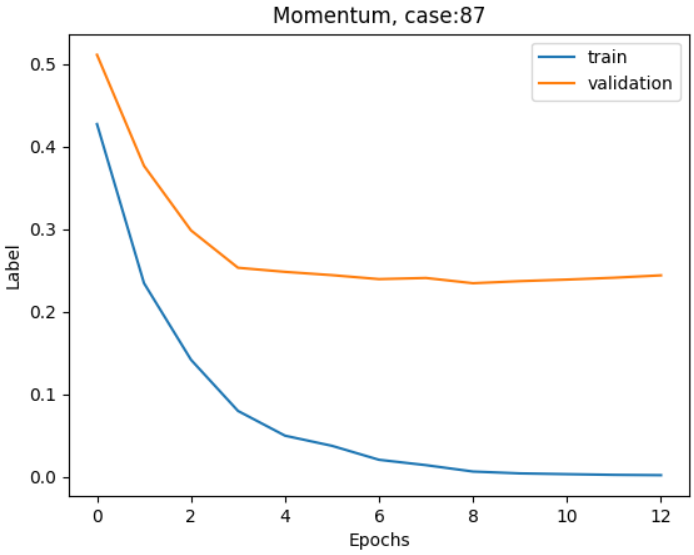


Best Params -->
Optimizer = Momentum; Batch = 256; Learning Rate = 0.1, Adaptive;
Activation = relu; Epochs = 62; Test Accuracy = 94.1957

e) Architecture --> [512, 256, 128, 46]

4 Layered architecture

Best Params -->
Optimizer = Momentum; Batch = 256; Learning Rate = 0.1, Adaptive;
Activation = relu; Epochs = 13; Test Accuracy = 95.3261



Observations

Effect of number of Layers:

2 Layered network gives the least test accuracy.
As we increase the number of layers the test accuracy increases.
However, there is decrease in accuracy in the 5 layered network due to overfitting.

Effect of number of neurons in Hidden layer:

Architecture c) and d) both are 3 layered with d) having double the number of neurons in hidden layer as that of c).
Increasing the number of neurons significantly increased the test accuracy in part d).

Final params:

3 Layered architecture --> [512, 256, 46]
Optimizer = Momentum
Batch = 256
Learning Rate = 0.1, Adaptive
Activation = relu
Epochs = 29 (May vary due to speed of computer)
Test Accuracy = 95.6957