EARIN Lab 5 Report

Krzysztof Rudnicki, 307585
Jakub Kliszko, 303866

May 16, 2023

1 Exercise Variant 2

Use MIST dataset. Evaluate at least 3 different numbers/values/types of:

- learning rate

- mini-batch size (including batch containing only 1 example)

- number of hidden layers (including 0 hidden layers - linear model)

- width (number of neurons in hidden layers)

- optimizer type (e.g., SGD, SGD with momentum, Adam)

2 Implementation

Program can be ran by installing python, moving to project directory and issuing command:

`python main.py`

Results will be displayed on three 2d scatter plots.
Plots Title is filled with parameters used abbreviated for space sake
Abbreviatons meaning:

- lr - Learning Rate

- bs - Batch Size

- hl - Number of Hidden Layers

- w - Width

- Adam or SGD or SGD_Momentum - Optimizer type

Plot types:

- Loss value for learning step

- Train Accuracy for epoch

- Validation Accuracy for epoch

Figure 1: Exemplary plot of loss for learning Step
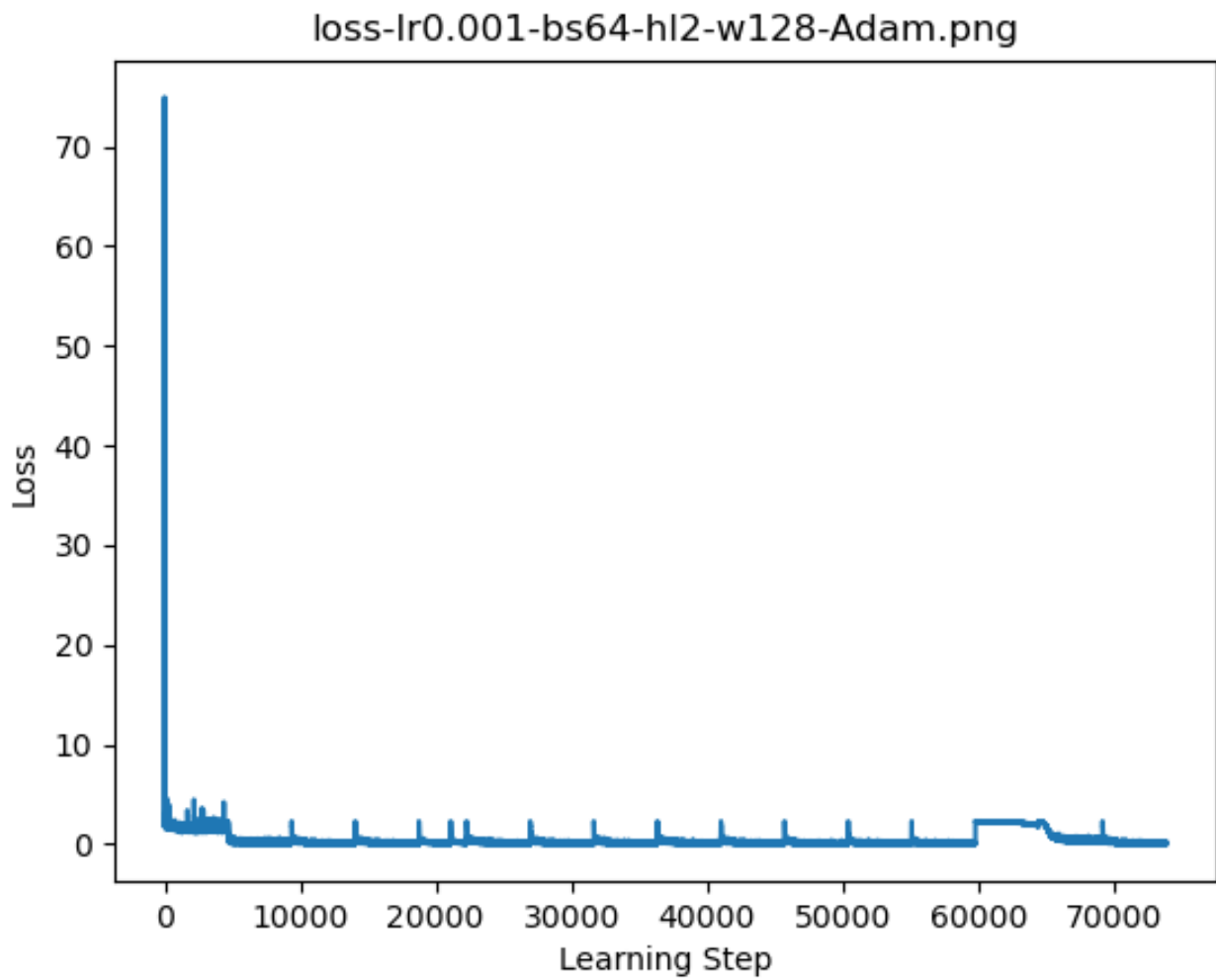lr0.001-bs64-hl1-w128-Adam

loss-lr0.001-bs64-hl2-w128-Adam.png

Figure 2: Exemplary plot of Validation Accuracy for epoch
lr0.001-bs64-hl1-w128-Adam

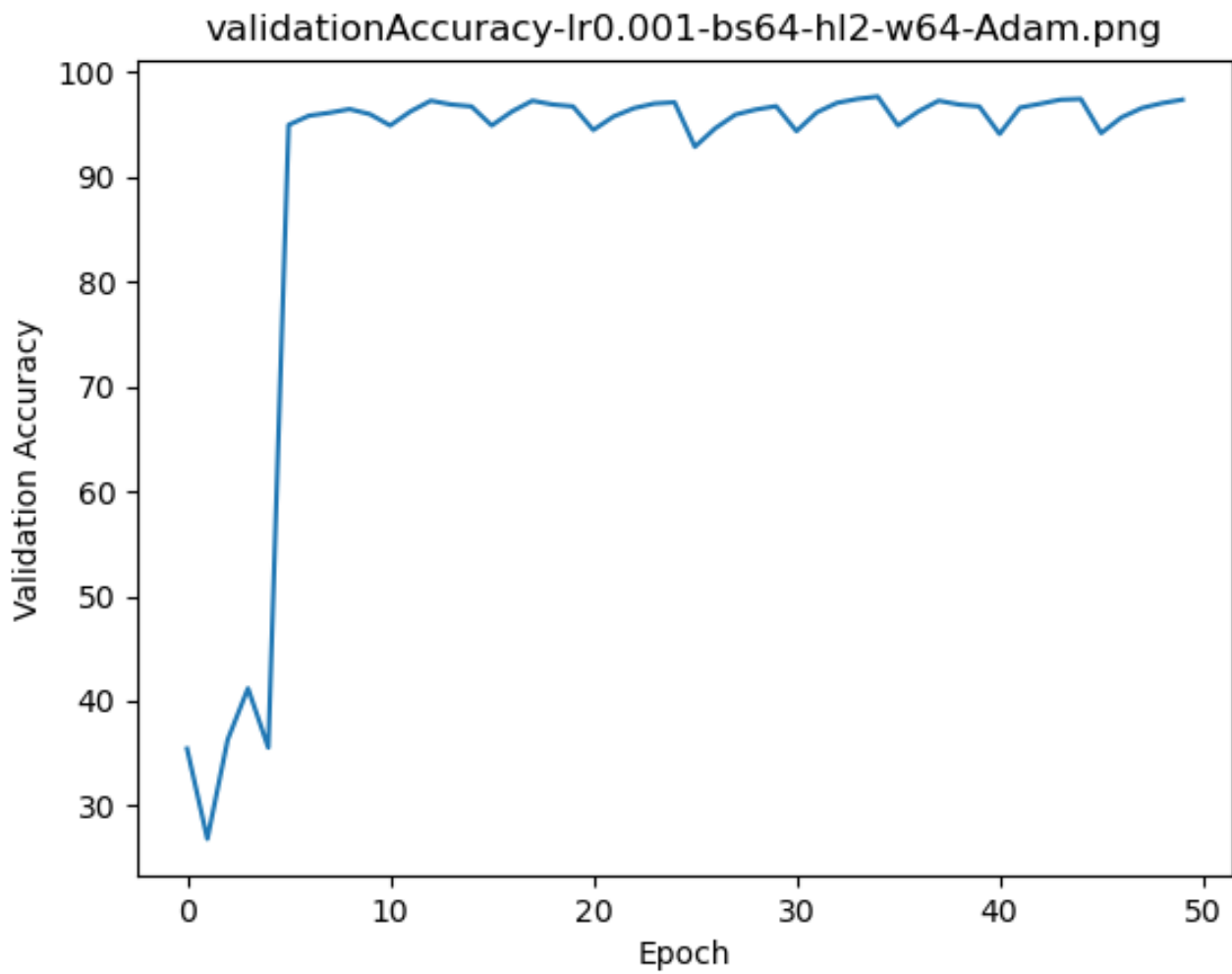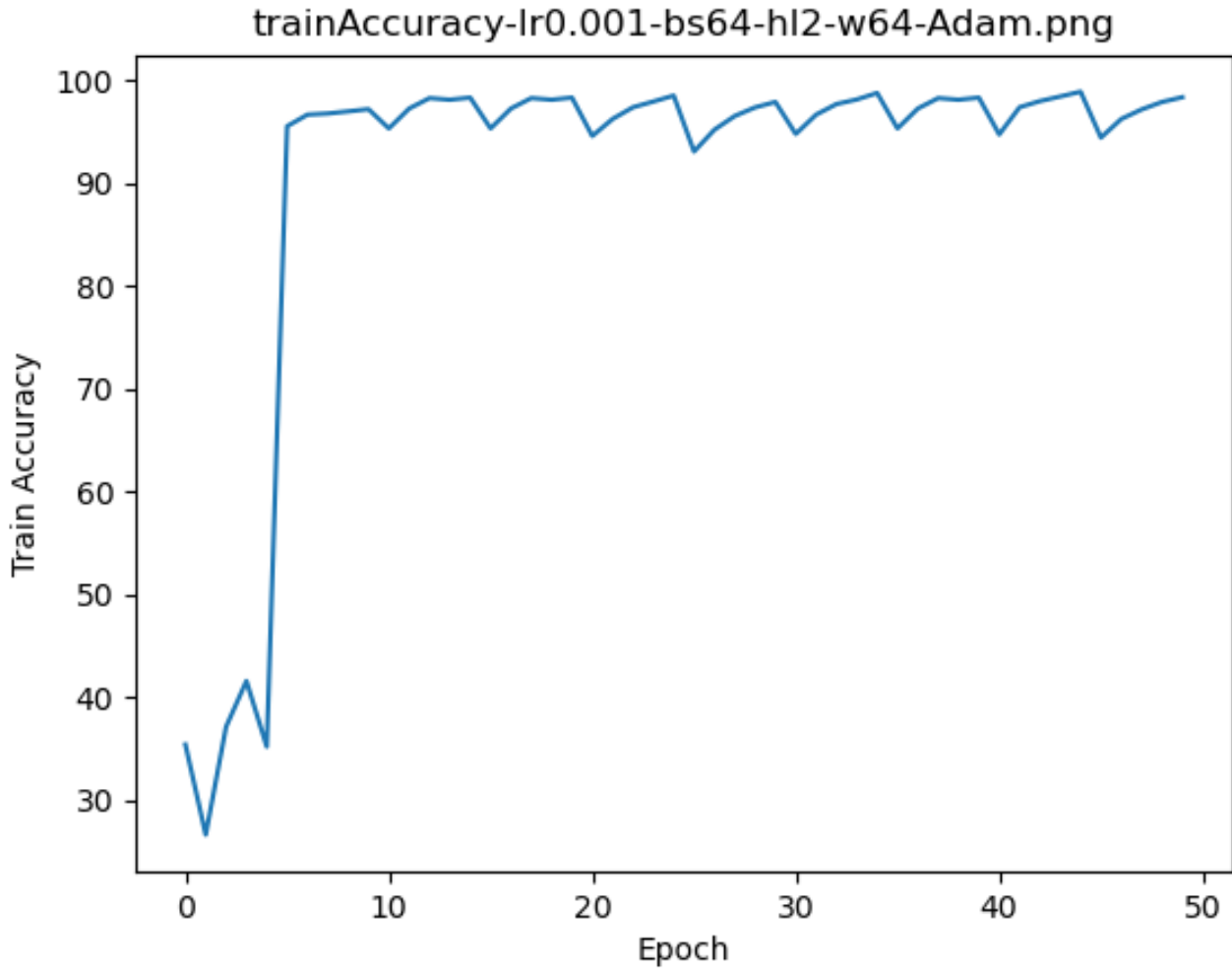validationAccuracy-lr0.001-bs64-hl2-w64-Adam.png

Figure 3: Exemplary plot of Train Accuracy for epoch
lr0.001-bs64-hl1-w128-Adam

Results will be displayed and saved in the same folder as code directory for further inspection, with file name containing information about input parameters

Additionaly speed it took for neural network to run will be saved to results.txt along with what parameters were used

We decided to run 19 tests in total:

1. learning rate [0.1, 0.01, 0.001] (3 tests)

2. mini-batch size [1, 64, 128, 256] (4 tests)

3. number of hidden layers [0, 1, 2, 3] (4 tests)

4. width [64, 128, 256, 512, 1024] (5 tests)

5. optimizer type [SGD, SGD_Momentum, Adam] (3 tests)

3 Results

We have successfully implemented network analyzing MNIST dataset

3.1 Loss Graphs

3.1.1 Learning Rate
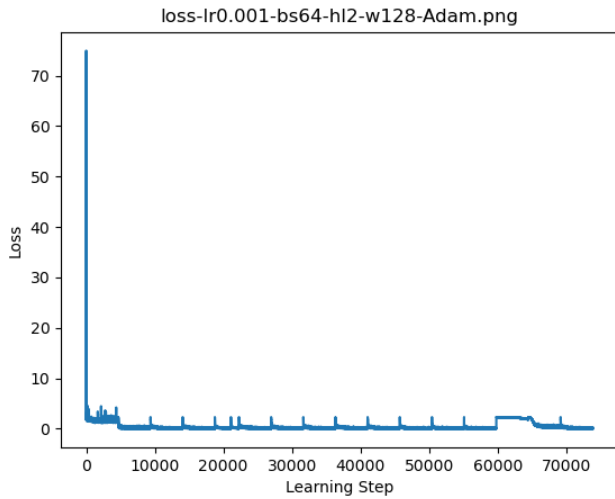
Figure 4: Default settings + learning rate = 0.001
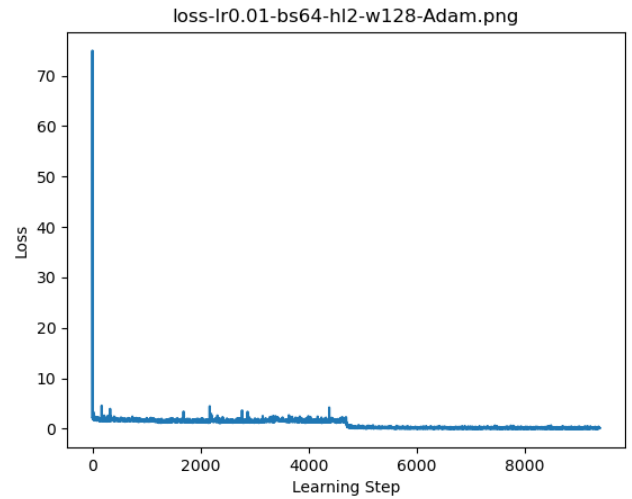
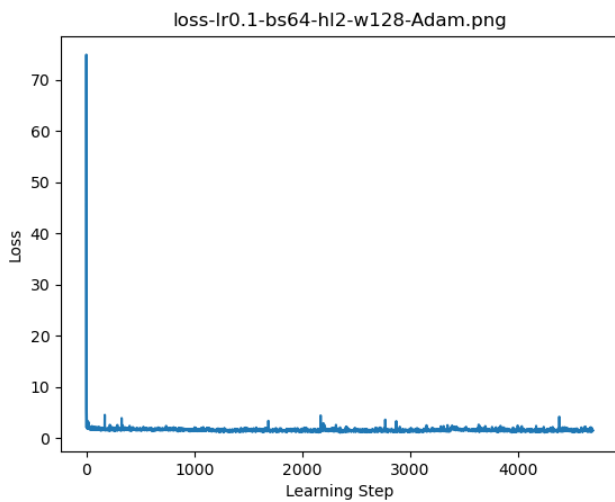Figure 5: Default settings + learning rate = 0.01

Figure 6: Default settings + learning rate = 0.1

Analysis Smaller learning rate causes losses to drop near zero level faster (notice how there is a vertical drop for lower learning step for leaerning rate 0.001, later for learning rate 0.01 and never for learnig rate 0.1)

3.1.2 Mini-Batch size
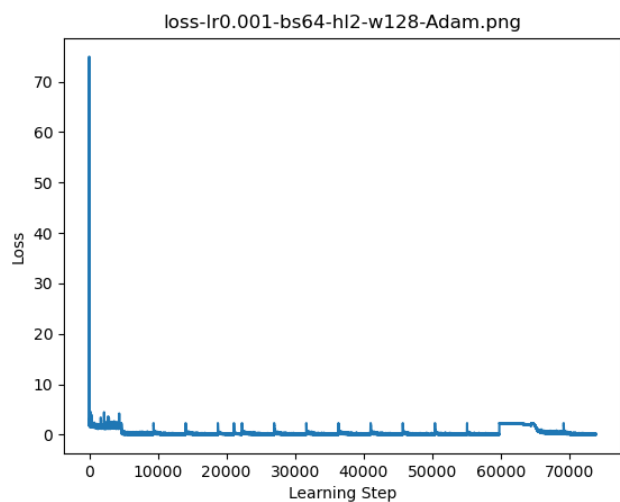
Figure 7: Default settings + batching size = 64
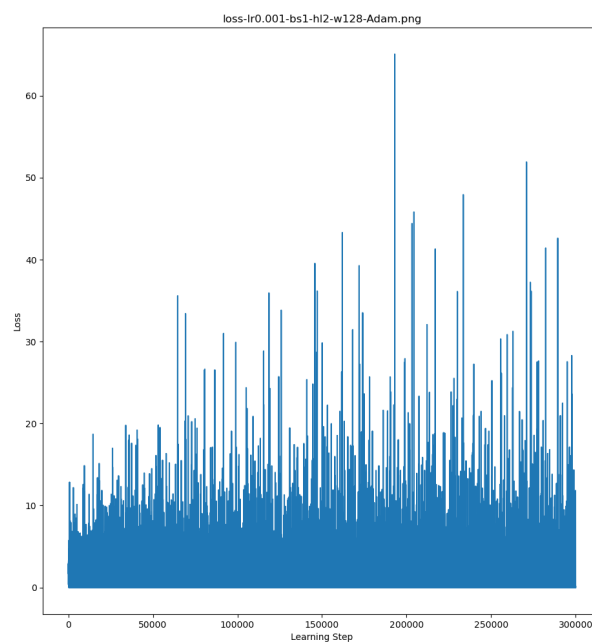
Figure 8: Default settings + batching size = 1
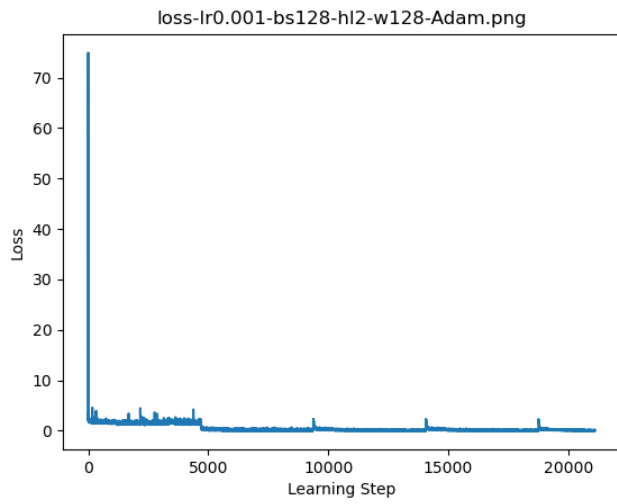
Figure 9: Default settings + batching size = 128
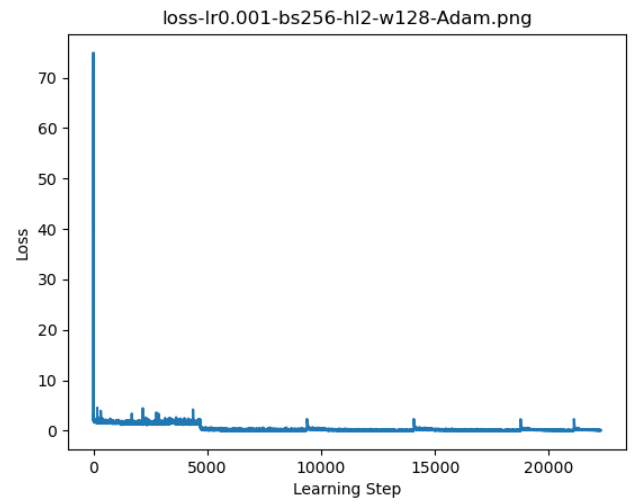
Figure 10: Default settings + batching size = 256

Analysis Batching size equal to 1 makes losses seem pretty much random, increasing batch sizes increases 'flat' (smaller) loss periods between local peaks

3.1.3 Number of Hidden Layers
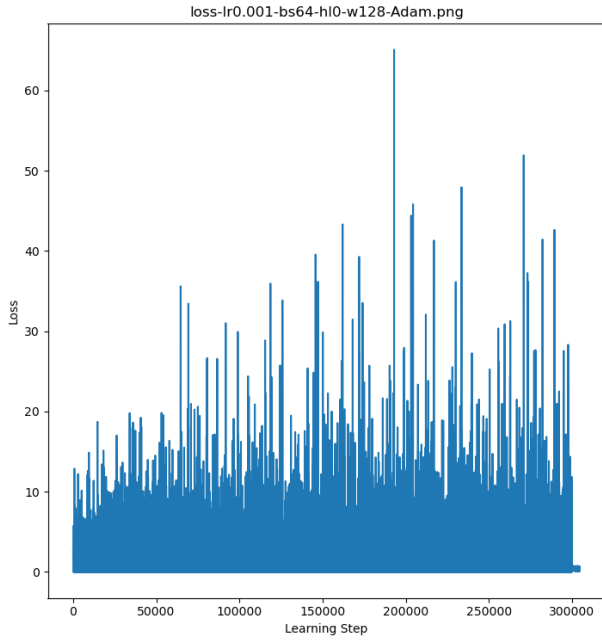
Figure 11: Default settings + hidden layers = 0
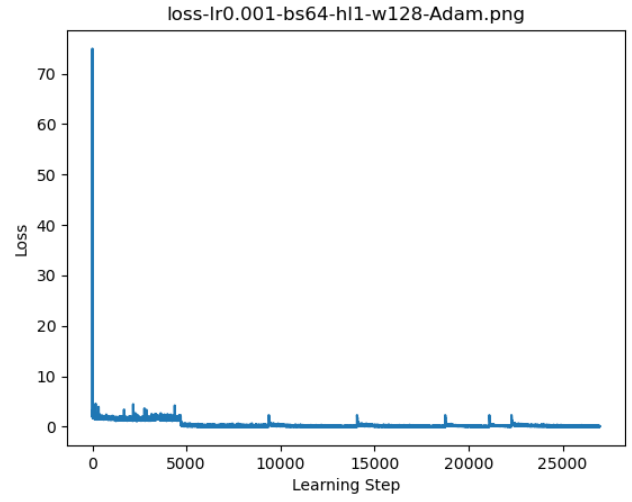
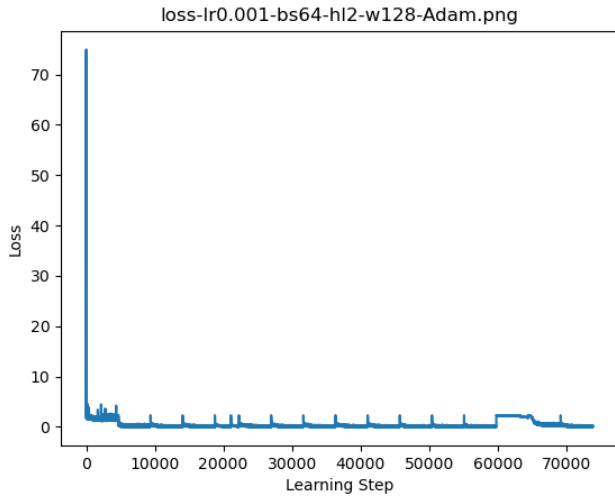Figure 12: Default settings + hidden layers = 1
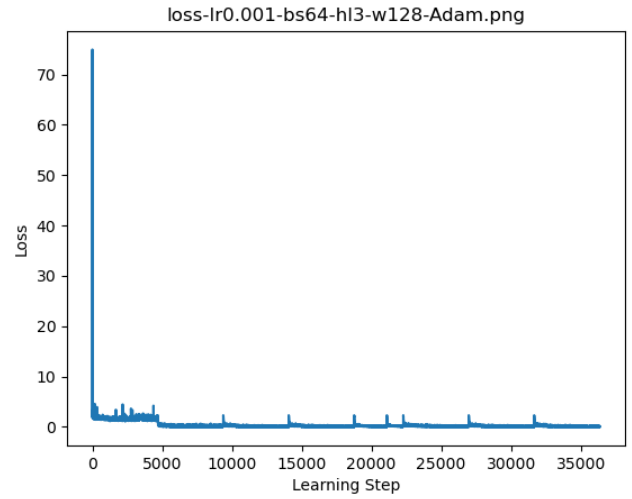

Figure 13: Default settings + hidden layers = 2

Figure 14: Default settings + hidden layers = 3

8

Analysis   Setting hidden layers to 0 again makes losses value seem random, hidden layer set to 1 provides us with later drop but longer flat periods, number of hidden layers set to 2 offeres the opposite and number of layers set to 3 seems to be a compromise between those two

3.1.4   Width
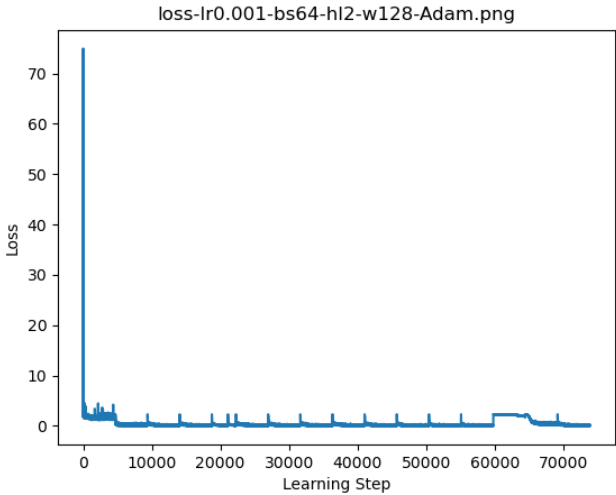
Figure 15: Default settings + width = 128
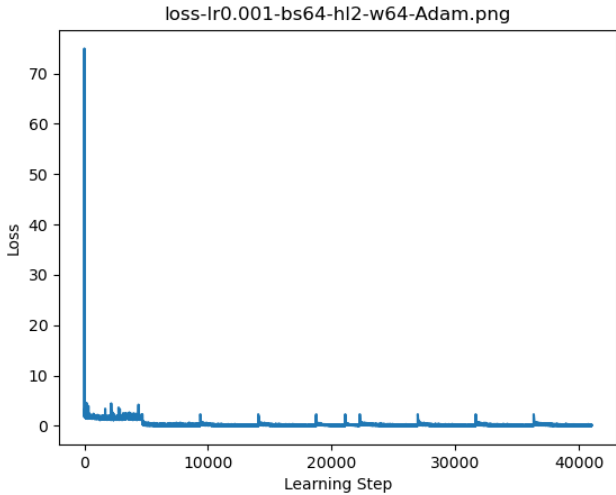
Figure 16: Default settings + width = 64
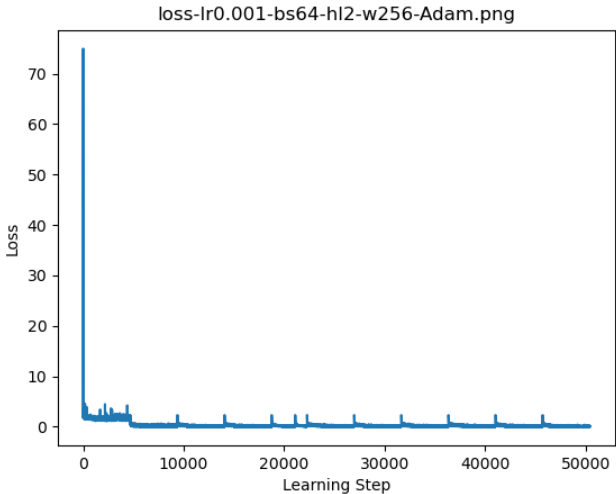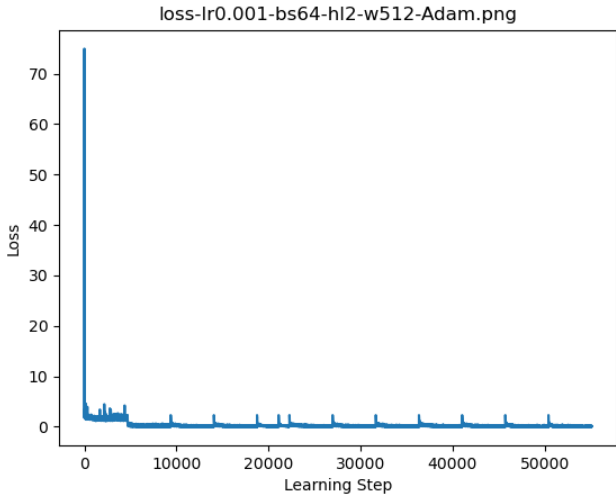

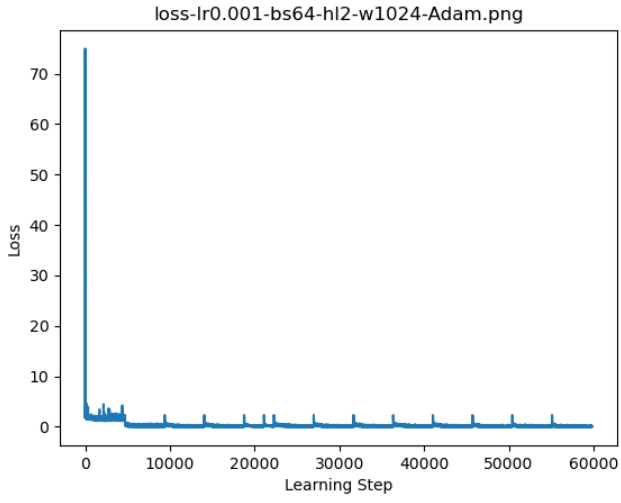Figure 17: Default settings + width = 256

Figure 18: Default settings + width = 512

9

Figure 19: Default settings + width = 1024

Analysis Changing width does not seem to affect loss greatly

3.1.5 Optimizer Type

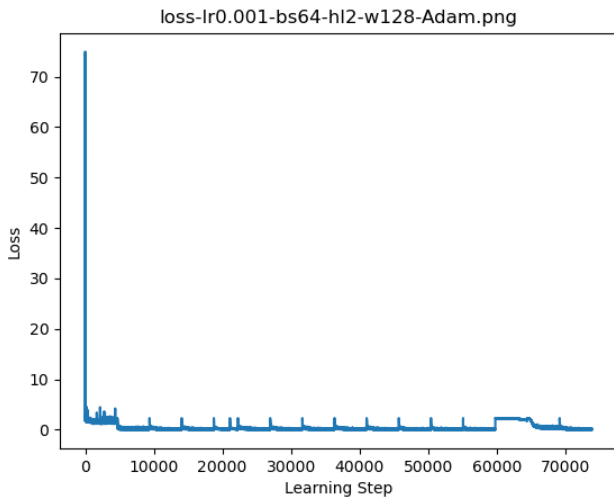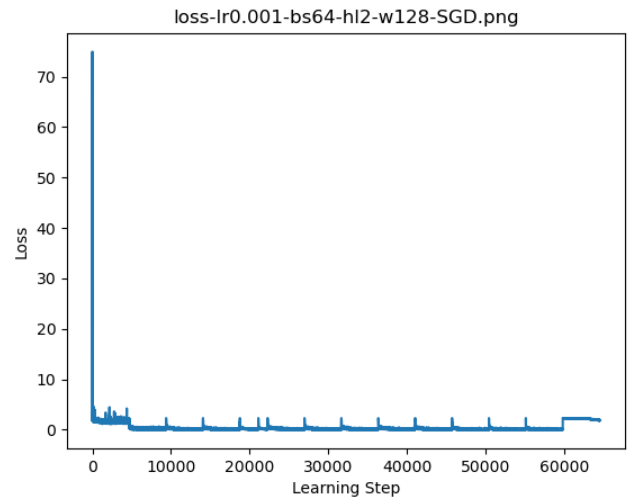
Figure 20: Default settings + Adam optimizer

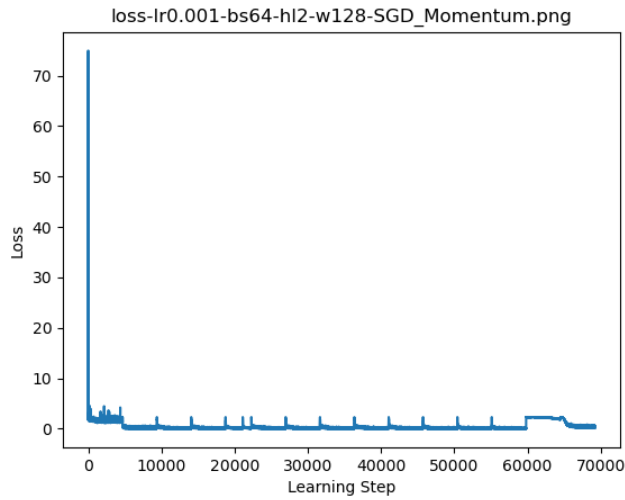Figure 21: Default settings + learning rate = 0.01

Figure 22: Default settings + SGD_Momentum optimizer

Analysis   Changing optimizers does not seem to affect loss greatly

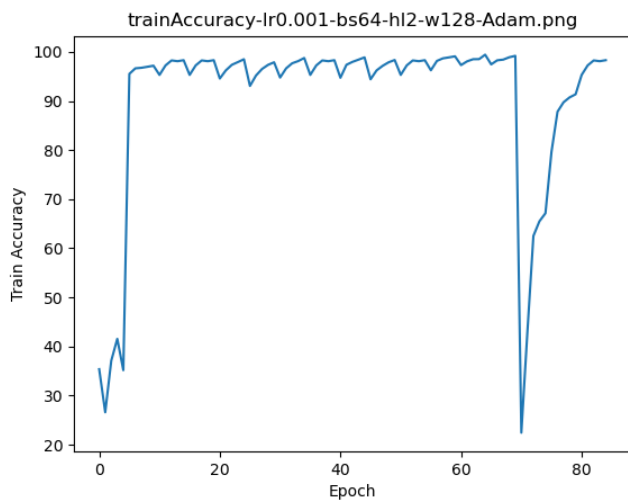## 3.2   Train Accuracy Graphs

3.2.1   Learning Rate

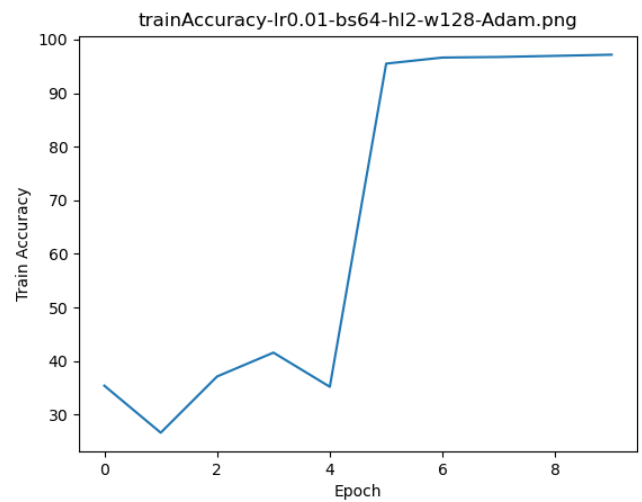Figure 23: Default settings + learning rate = 0.001

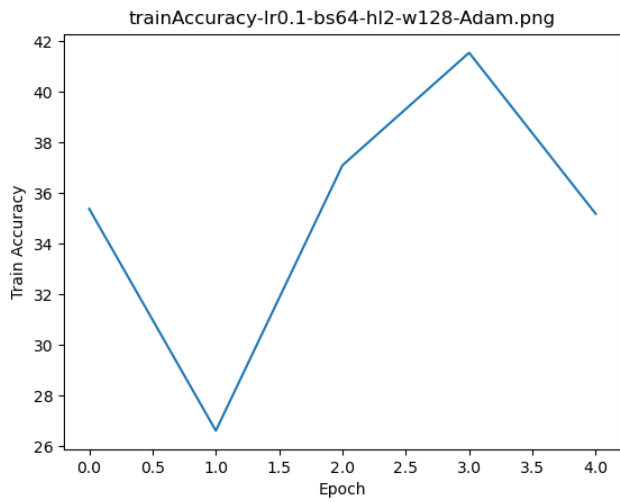Figure 24: Default settings + learning rate = 0.01

11

Figure 25: Default settings + learning rate = 0.1

Analysis Making training accuracy smaller than 0.01 seems to be unnecessary for this dataset as it offers little to no improve in accuracy. Setting it to 0.1 on the other hand provides very low accuracy of neural network
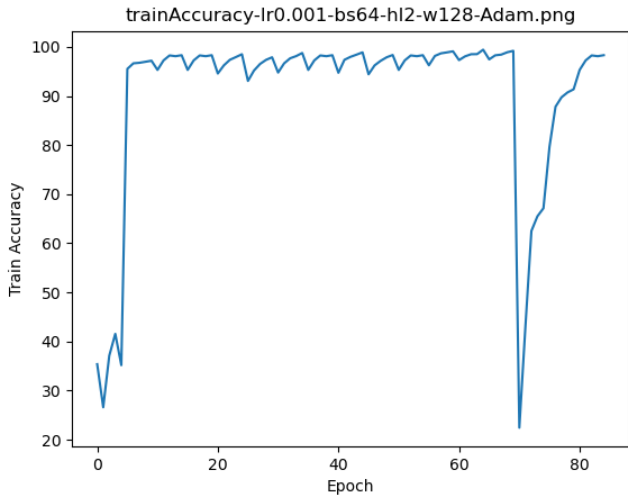
3.2.2 Mini-Batch size

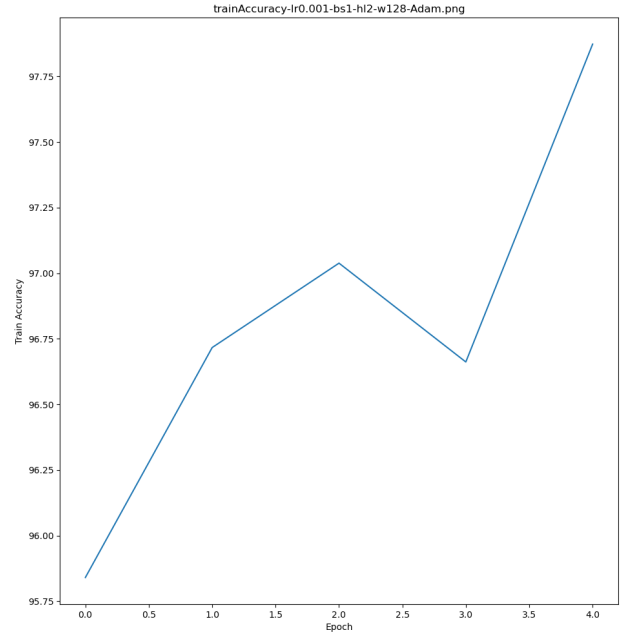Figure 26: Default settings + batching size = 64
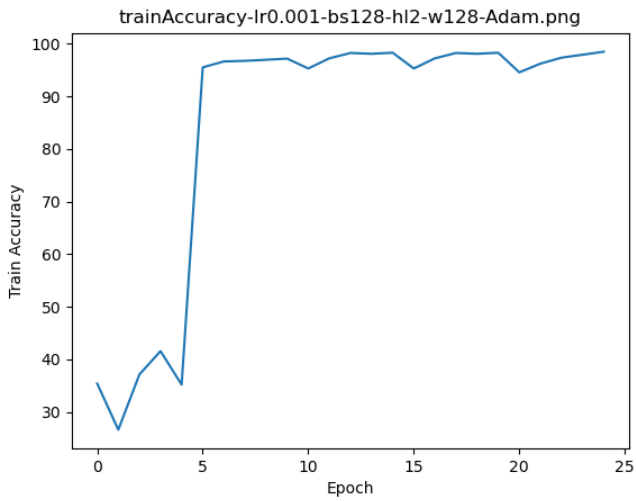

Figure 27: Default settings + batching size = 1
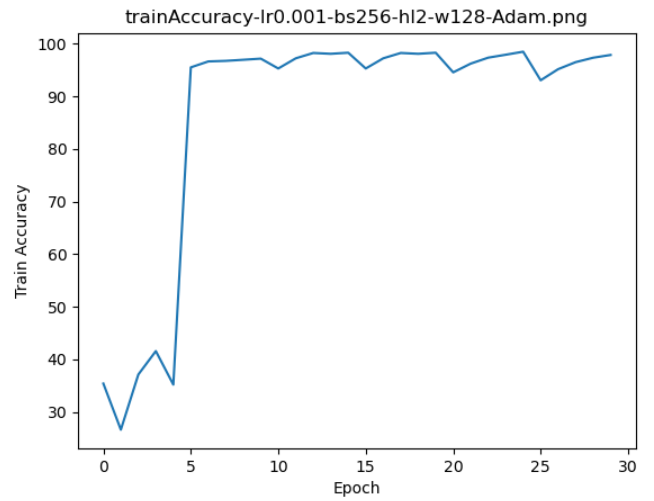

Figure 28: Default settings + batching size = 128

Figure 29: Default settings + batching size = 256

13

Analysis Increasing batching size does not seem to make drastic change for train accuracy, even setting batching size to 1 offers very good (over 95 %) accuracy
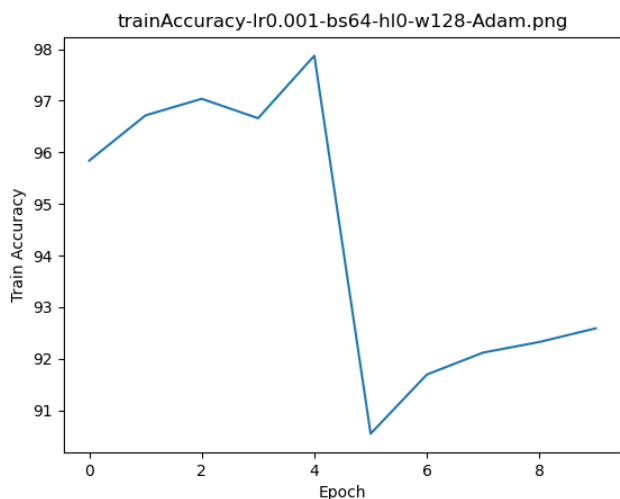
3.2.3 Number of Hidden Layers

Figure 30: Default settings + hidden layers = 0
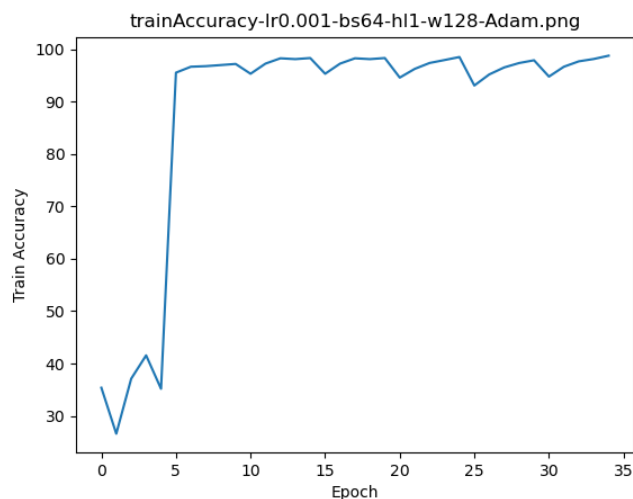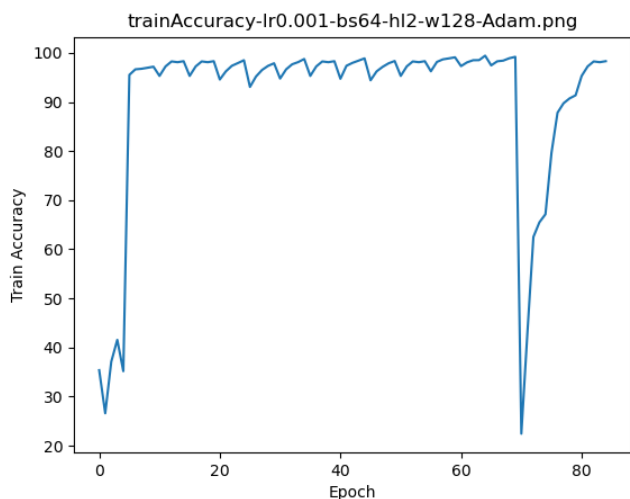

Figure 31: Default settings + hidden layers = 1

Figure 32: Default settings + hidden layers = 2
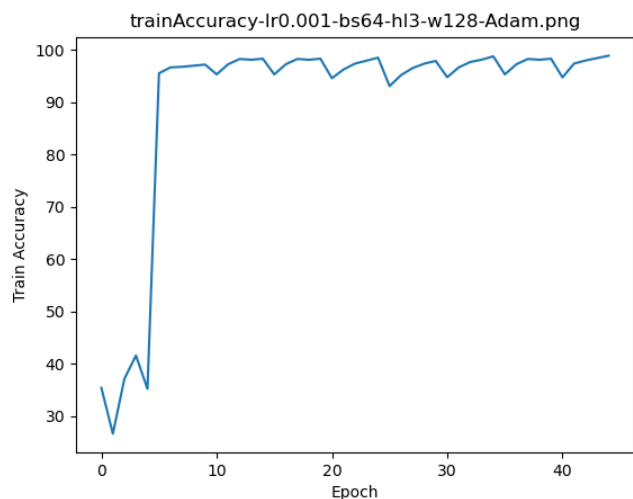

Figure 33: Default settings + hidden layers = 3

Analysis Increasing hidden layers number reduces variety of train accuracy results, with hidden layers set to 0 having accuracy vary between 91 and 98 and with hidden layers equal to 3 showing much smoother graph
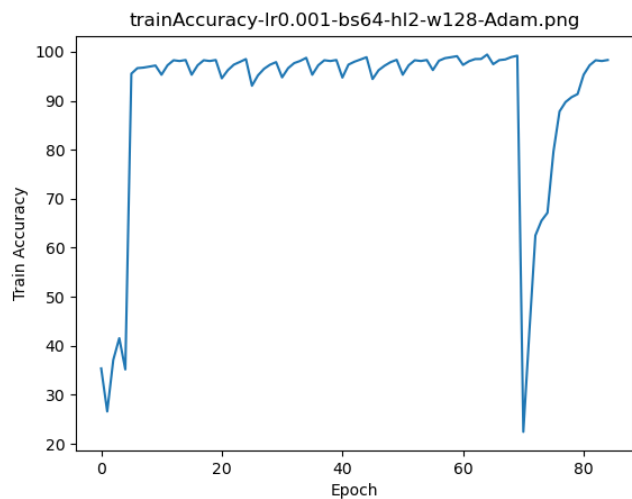
14

3.2.4 Width

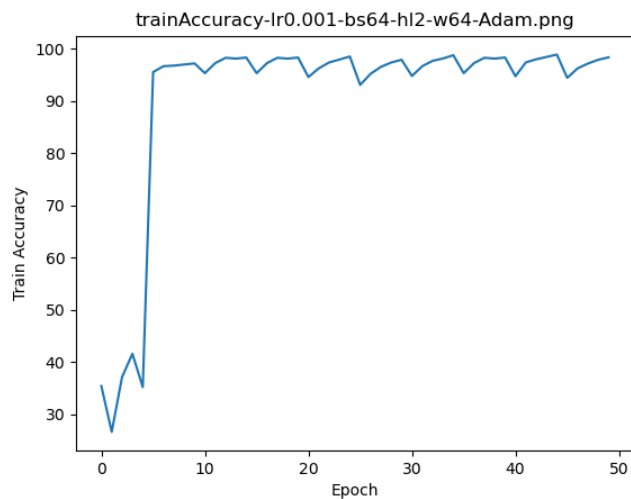Figure 34: Default settings + width = 128

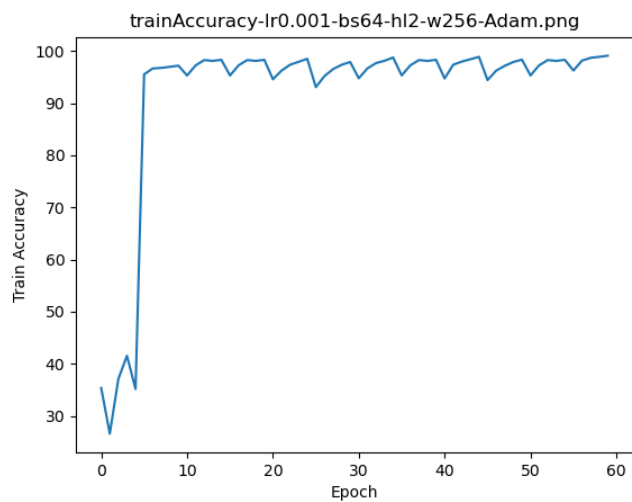Figure 35: Default settings + width = 64

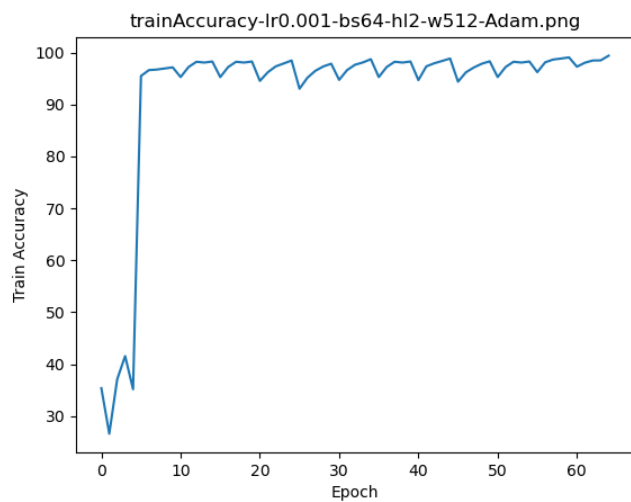Figure 36: Default settings + width = 256

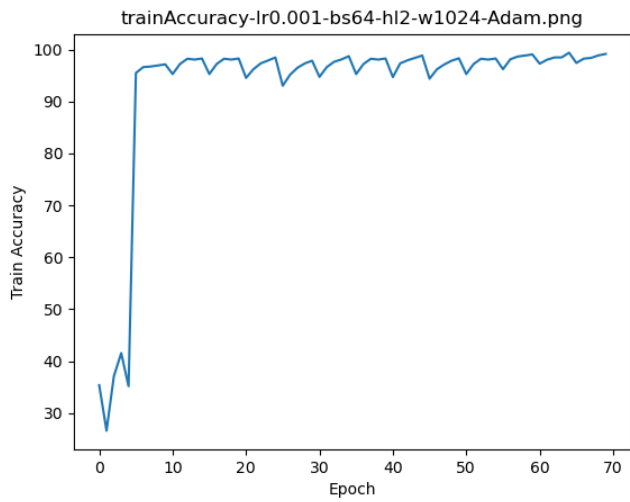Figure 37: Default settings + width = 512

Figure 38: Default settings + width = 1024

Analysis   Changing width does not seem to influence accuracy much
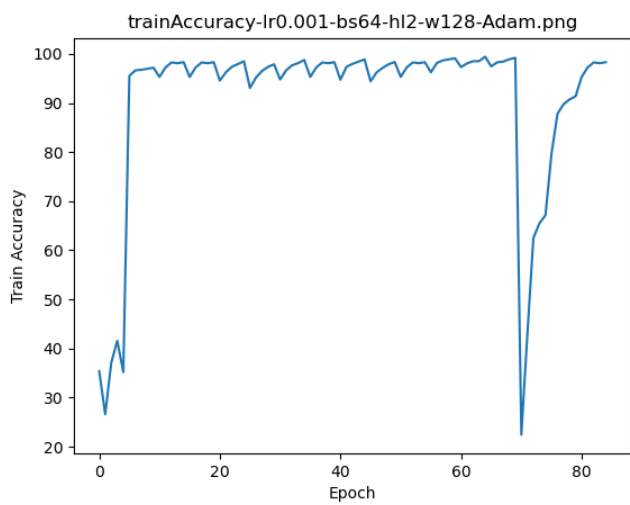
3.2.5 Optimizer Type
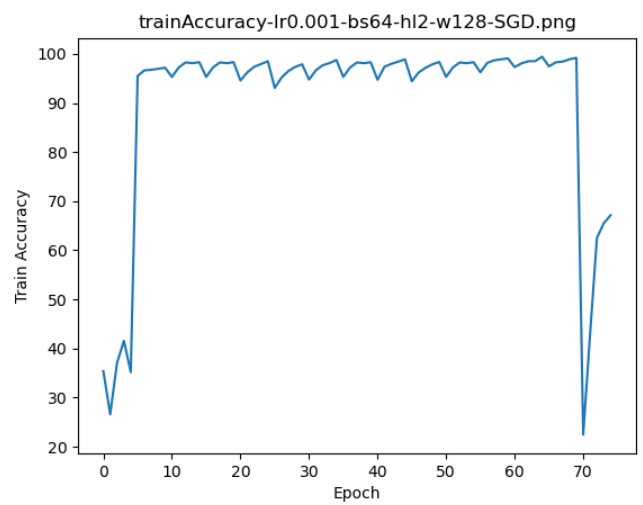

Figure 39: Default settings + Adam optimizer

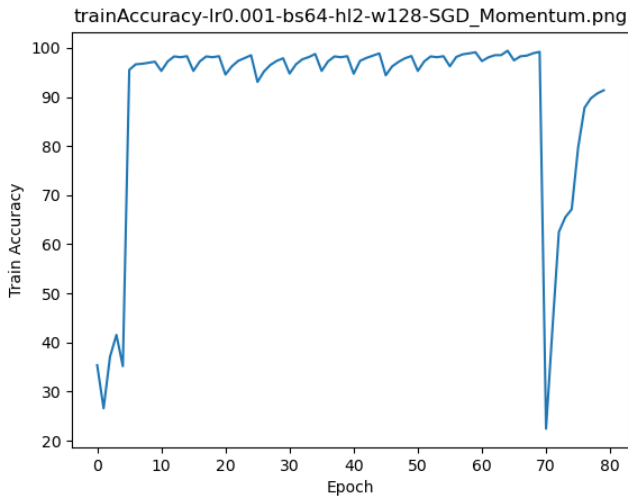Figure 40: Default settings + learning rate = 0.01

Figure 41: Default settings + SGD_Momentum optimizer

Analysis   Changing optimizer type does not seem to influence accuracy much

3.3 Validation Accuracy Graphs

Validation Accuracy Graphs look similar to Train accuracy so in order not to repeat ourselves we ommited analysis of them
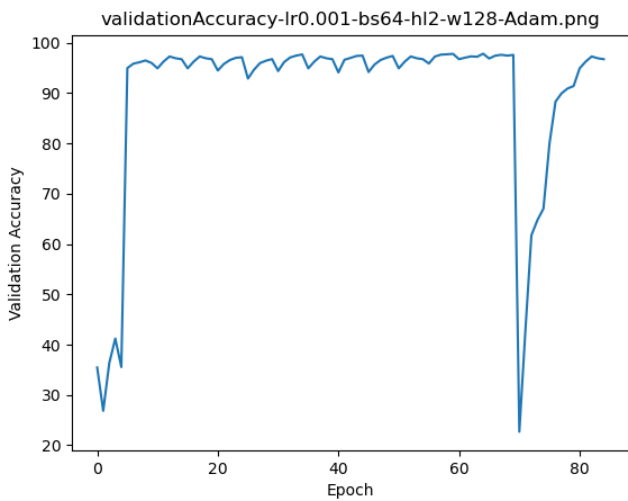
3.3.1 Learning Rate

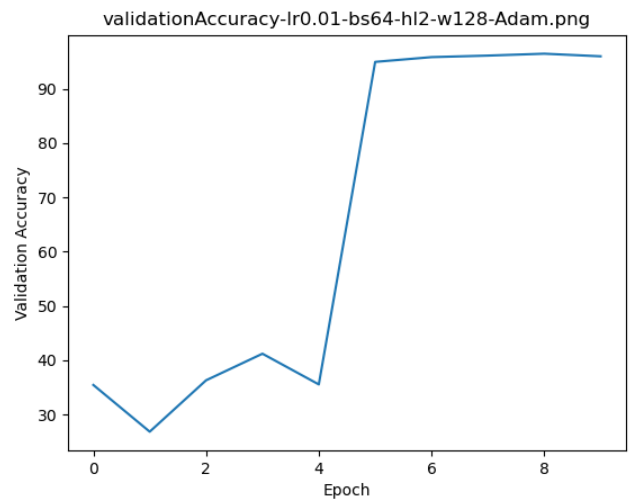
Figure 42: Default settings + learning rate = 0.001

Figure 43: Default settings + learning rate = 0.01
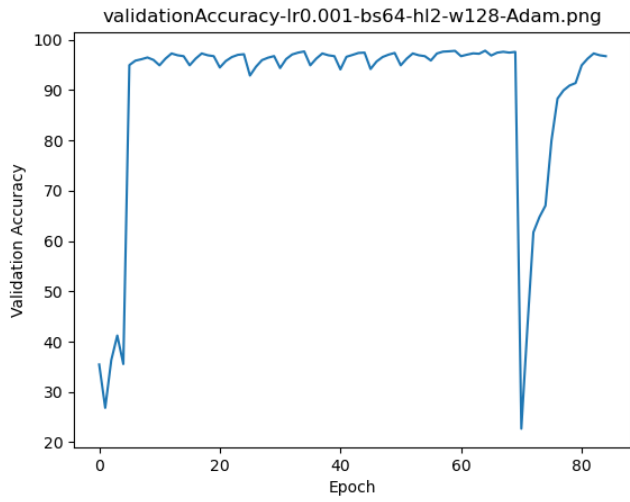
17

Figure 44: Default settings + learning rate = 0.1

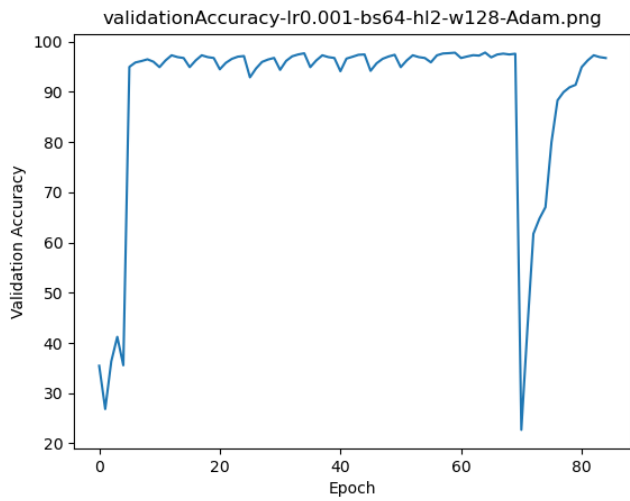### 3.3.2 Mini-Batch size
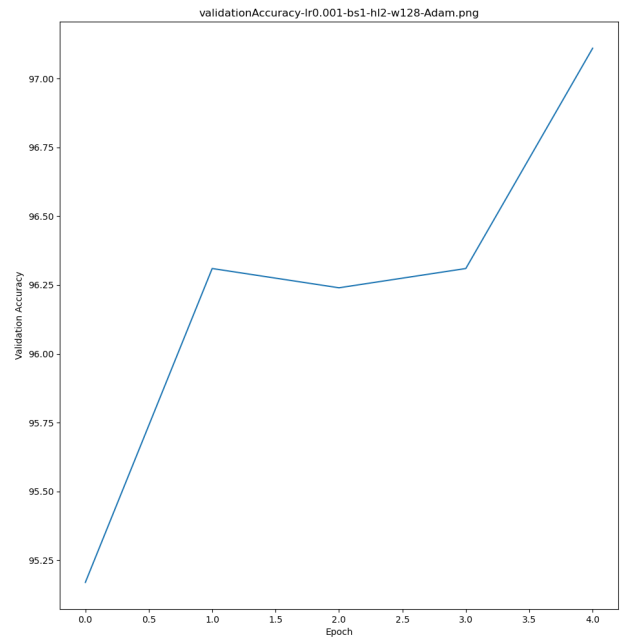

Figure 45: Default settings + batching size = 64

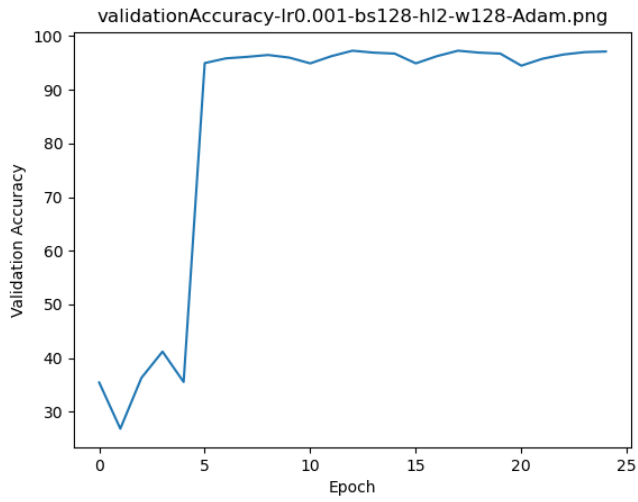Figure 46: Default settings + batching size = 1
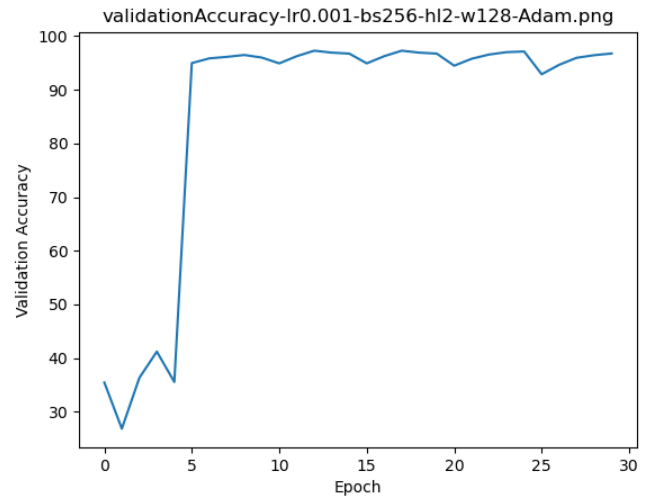
Figure 47: Default settings + batching size = 128

Figure 48: Default settings + batching size = 256
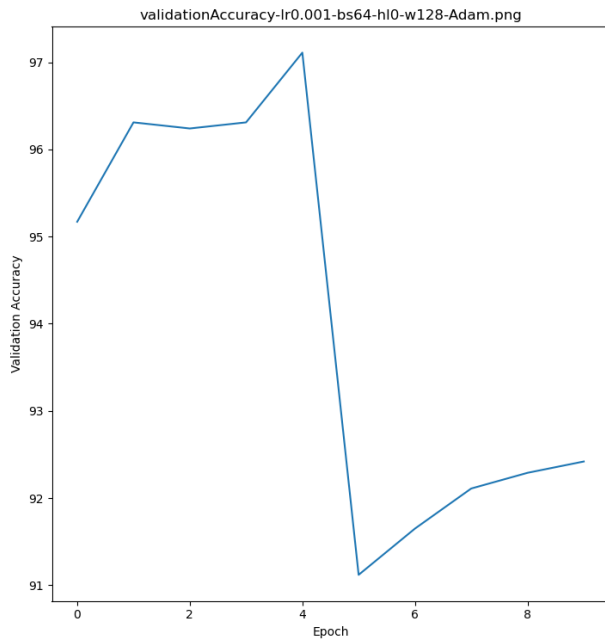
3.3.3 Number of Hidden Layers

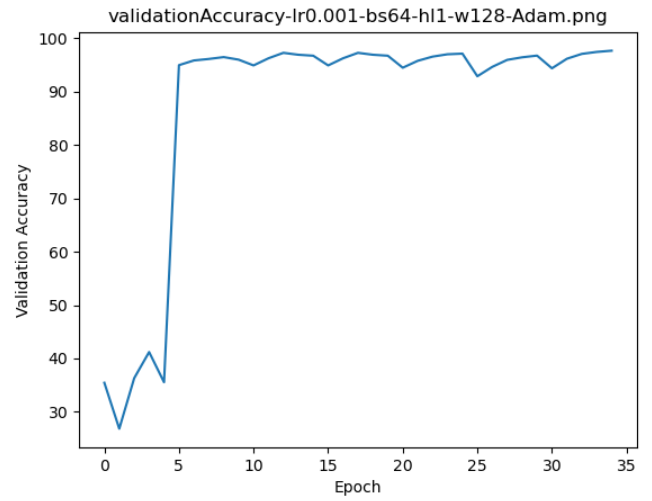
Figure 50: Default settings + hidden layers = 1

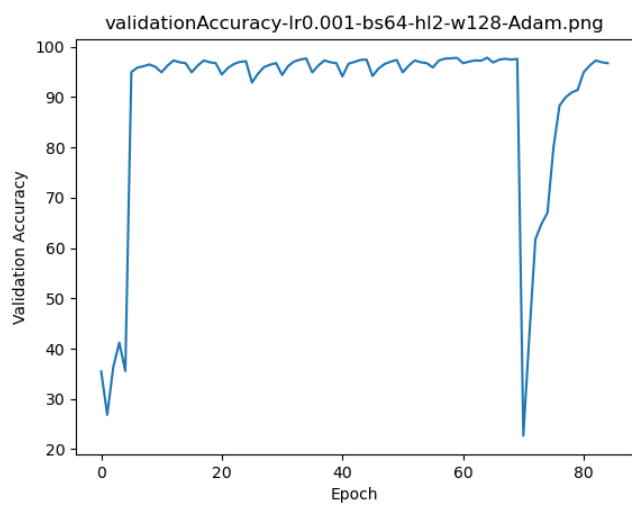Figure 49: Default settings + hidden layers = 0

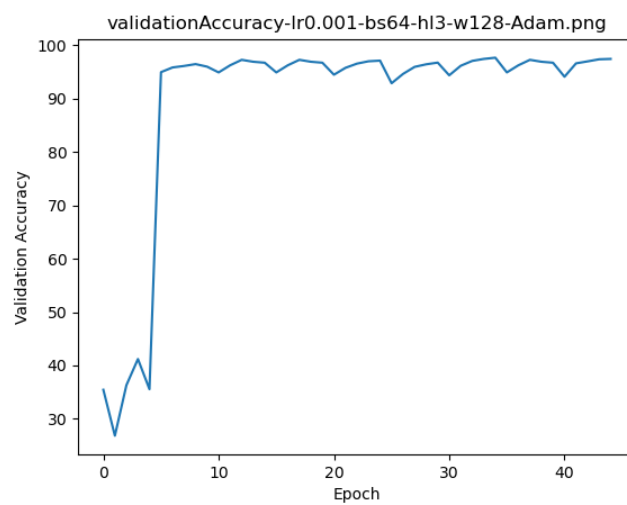Figure 51: Default settings + hidden layers = 2

Figure 52: Default settings + hidden layers = 3
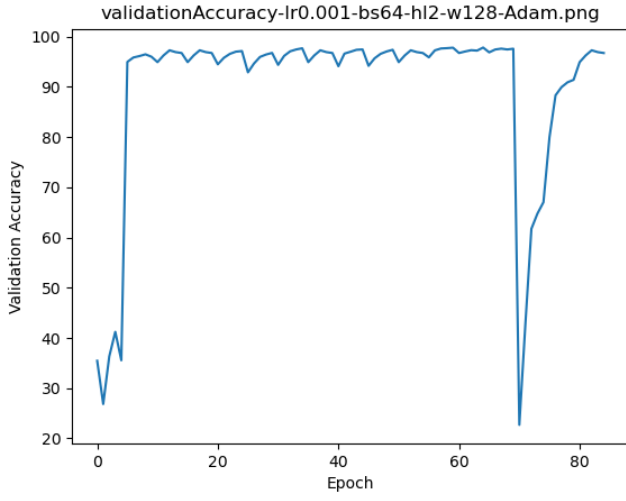
3.3.4 Width

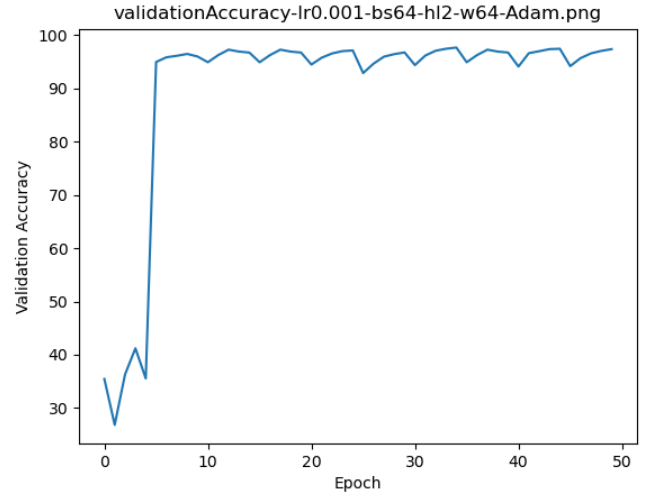Figure 53: Default settings + width = 128

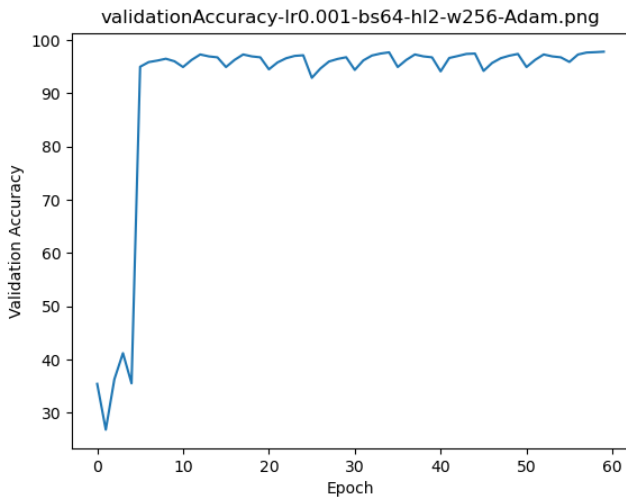
Figure 54: Default settings + width = 64

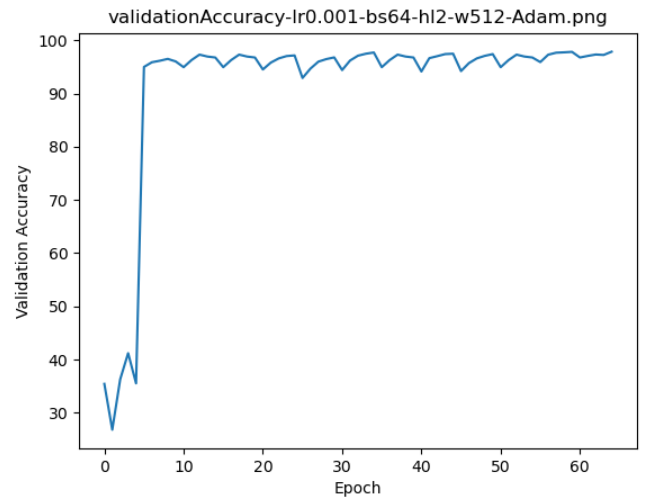Figure 55: Default settings + width = 256

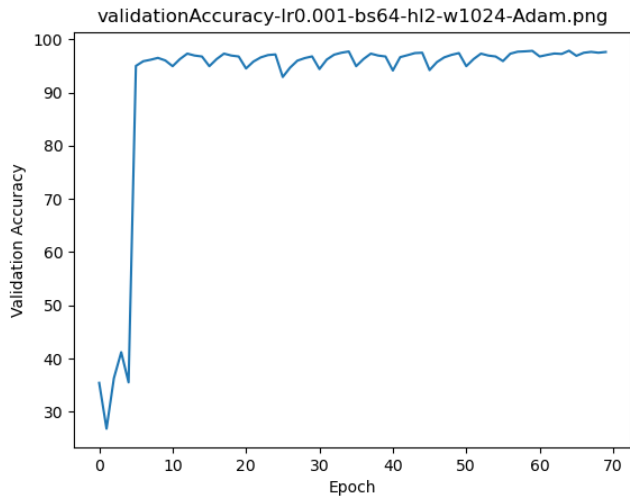Figure 56: Default settings + width = 512

21

Figure 57: Default settings + width = 1024

3.3.5 Optimizer Type

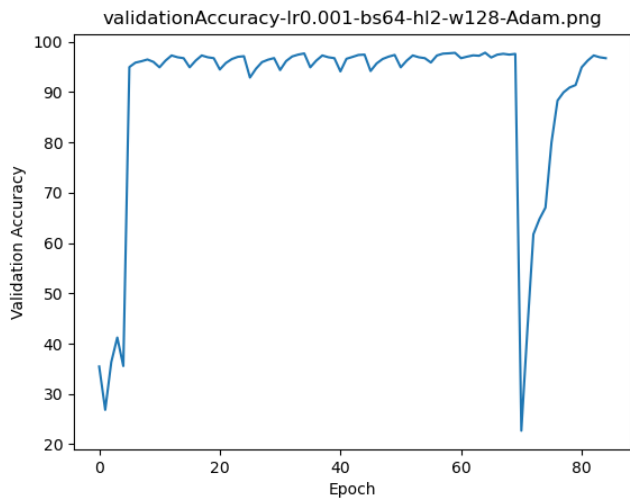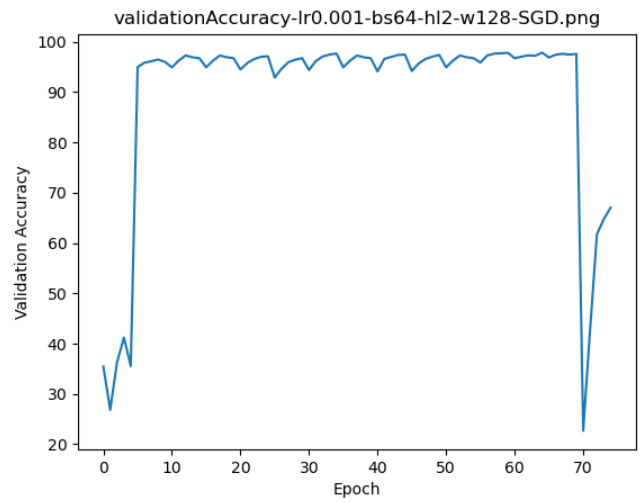Figure 58: Default settings + Adam optimizer

Figure 59: Default settings + learning rate = 0.01
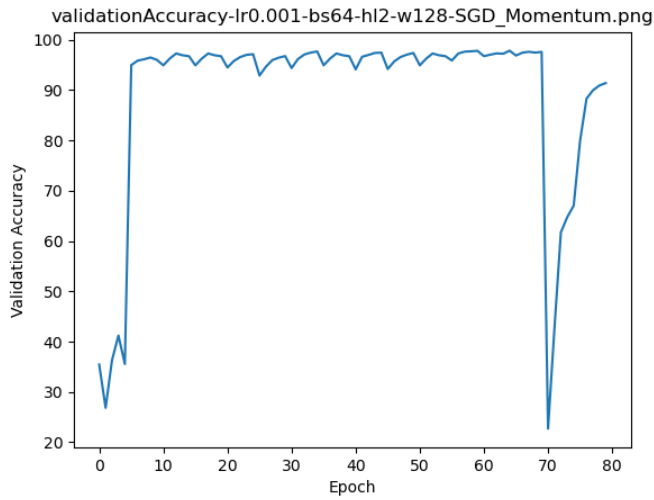
Figure 60: Default settings + SGD_Momentum optimizer

3.4 Speed

We also checked how different parameters impact speed of the neural network learning

Changing learning rate

Learning rate	Execution Time [s]
0.001 (default)	49.3
0.01	53.9
0.1	50.1

Changing batch size

Batch Size	Execution Time [s]
1	567.2 (!)
64 (default)	49.3
128	48.1
256	42.8

Changing number of hidden layers

Hidden layers	Execution Time [s]
0	44.5
1	48.8
2 (default)	49.3
3	48.7

Changing width

23

Width	Execution Time [s]
64	46.7
128 (default)	49.3
256	49.0
512	52.1
1024	73.9

Changing optimizer type

Learning rate	Execution Time [s]
Adam (default)	49.3
SGD	45.1
SGD Momentum	45.5

Analysis   Two parameters that had the biggest impact on execution time was setting batch size to extremely low value (1) and increasing width (jump between 512 width and 1024 is equal to 20 seconds more execution time)