Task 4: Multi-class classification using a Fully Connected Neural Network on the MNIST Dataset

1 Objective

The objective of this experiment is to compare the performance of different optimization algorithms on the MNIST digit classification task using neural networks of varying depth (3, 4, and 5 hidden layers). The optimizers evaluated are SGD, Batch Gradient Descent, Momentum, Nesterov Accelerated Gradient (NAG), RMSprop, and Adam.

2 Dataset

- Dataset: MNIST
- Classes used: Digits 0–4
- Training samples (fast mode): 1000
- Test samples: 500
- Input size: 784 (28×28 images)
- Output classes: 5

3 Model Architectures

Architecture	Hidden Layers
3 Layers	$128 \rightarrow 64 \rightarrow 32$
4 Layers	$128 \rightarrow 64 \rightarrow 32 \rightarrow 16$
5 Layers	$128 \rightarrow 64 \rightarrow 32 \rightarrow 16 \rightarrow 8$

4 Optimizers Used

Optimizer	Key Parameters
SGD	lr = 0.001
Batch GD	Full batch updates
Momentum	momentum = 0.9
NAG	momentum = 0.9, nesterov = True
RMSprop	alpha = 0.99
Adam	betas = (0.9, 0.999)

5 Training Loss Curves

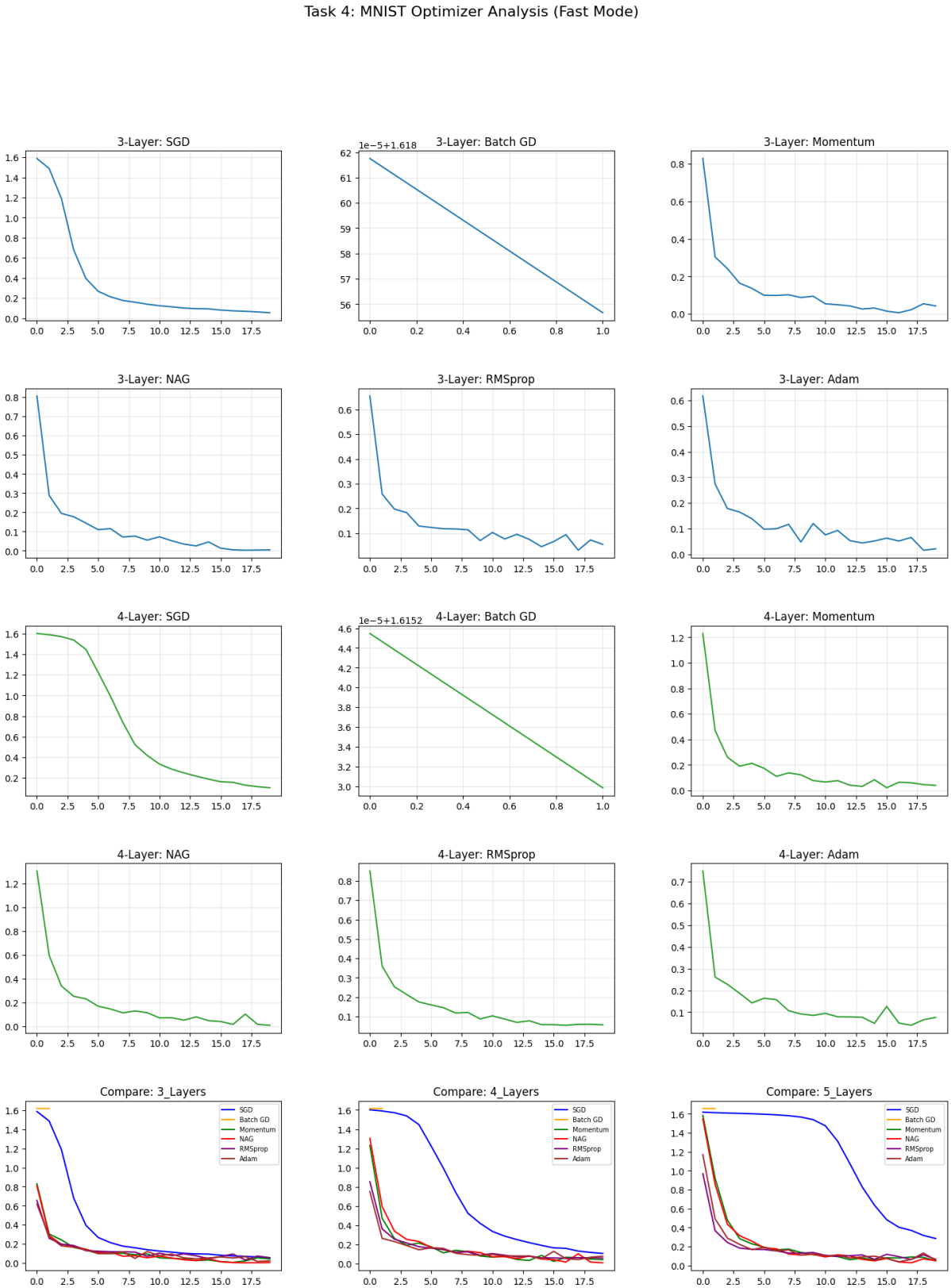


Figure 1: Training loss curves for different optimizers and architectures

6 Training Accuracy (%)

Architecture	SGD	Batch GD	Momentum	NAG	RMSprop	Adam
3 Layers	99.0	15.7	99.7	100	99.4	100
4 Layers	97.9	19.4	99.8	99.8	77.0	93.8
5 Layers	92.2	18.9	98.6	99.2	99.3	99.4

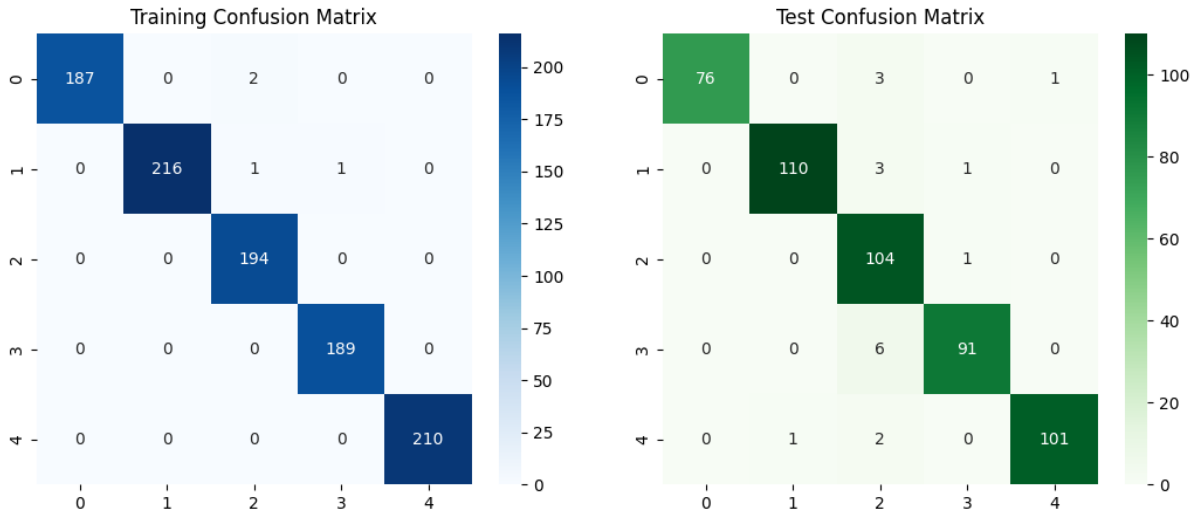
7 Validation Accuracy (%)

Architecture	SGD	Batch GD	Momentum	NAG	RMSprop	Adam
3 Layers	96.4	18.6	96.8	96.8	96.2	**97.4**
4 Layers	95.2	21.0	97.0	96.6	72.2	89.2
5 Layers	87.2	19.4	95.0	94.6	96.2	97.0

8 Best Model Selection

The best model is the **3-layer network with Adam optimizer**, achieving a validation accuracy of **97.4%**.

9 Confusion Matrices



(a) Training Confusion Matrix (b) Test Confusion Matrix

Figure 2: Confusion matrices of the best performing model

10 Observations

- Batch Gradient Descent performs worst due to slow convergence.

- SGD performs well but converges slower than adaptive methods.
- Momentum and NAG improve convergence speed and stability.
- RMSprop performs well for shallow networks but is unstable for deeper ones.
- Adam provides the best balance of speed, stability, and accuracy.

11 Effect of Network Depth

- 3 Layers: Best generalization performance.
- 4 Layers: Slight overfitting.
- 5 Layers: Harder to train and more prone to overfitting.

12 Conclusion

Adaptive optimizers, especially Adam, outperform traditional SGD methods. A medium-depth neural network (3 layers) provides the best trade-off between model complexity and generalization. Increasing network depth does not necessarily improve performance.