SGD vs Adam: Training with Limited Epochs

Mehmet Aytuğ Yürük

January 1, 2026

Abstract

I wanted to see how SGD and Adam work when training time is short. I trained a simple MLP on MNIST with different initialization and scheduler settings. I ran 24 experiments in total. This is not a big study. I just wanted to see the difference between these two popular optimizers with my own eyes.

1 Introduction

When training neural networks, the optimizer choice matters a lot. I was curious: SGD or Adam? Which one works better when I have only a few epochs?

This is a small study. I do not propose anything new. I just wanted to run a simple experiment to see how SGD and Adam are different. I also changed initialization and scheduler settings to check if the results stay the same.

2 Experiment Setup

2.1 Model and Data

I used a simple MLP: 784 inputs (28×28 pixels) → 128 hidden neurons (ReLU) → 10 outputs. I chose the classic MNIST dataset.

Since MNIST does not provide a predefined validation split, the dataset was manually divided into 90% training and 10% validation subsets. A fixed random seed was used across all experiments to ensure that the data split remained identical. Consequently, all experiments were conducted on the same data partitions.

2.2 Training Protocol

I tried two different epoch budgets: 5 and 30 epochs. After each epoch, I saved the training and validation loss.

To find the best model, I looked at the validation F1 score. I saved the weights from the epoch with the highest F1 score. I used these weights for the final test, not the weights from the last epoch.

2.3 Experiment Variables

The main thing I wanted to compare is the optimizer:

- SGD (learning rate = 0.001)
- Adam (learning rate = 0.001)

But just changing the optimizer is not enough. So I also changed other things:

- Weight initialization: Xavier and He
- Learning rate scheduler: Step, Cosine, Exponential
- Epoch budget: 5 and 30

In total, I ran $2 \times 2 \times 3 \times 2 = 24$ experiments. In each comparison, only the optimizer is different. Everything else is the same.

3 Results

3.1 Optimizer Comparison with Short Budget

Figure 1 shows the loss curves for SGD and Adam with a 5 epoch budget. Both use the same initialization and scheduler.

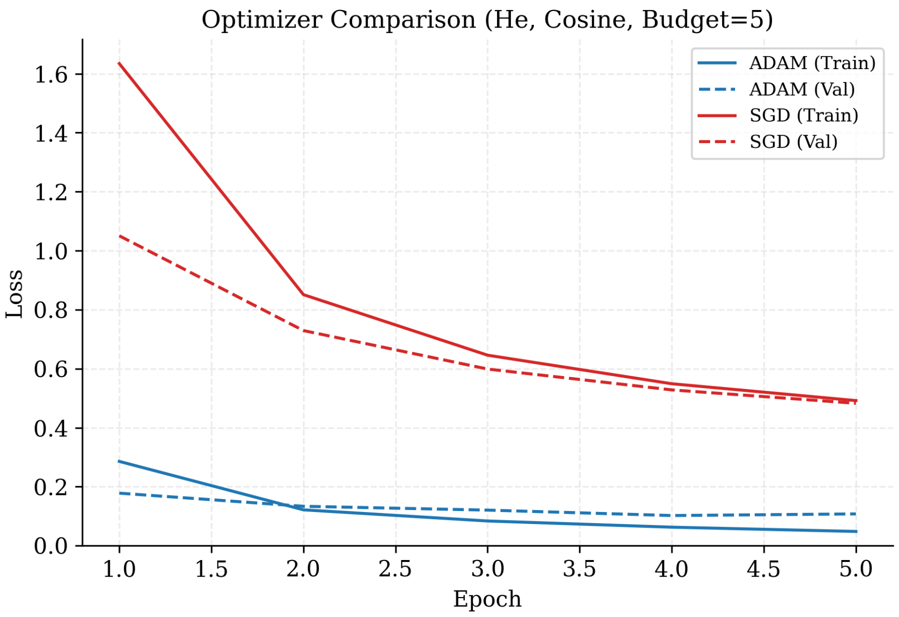


Figure 1: Training and validation loss for SGD and Adam (He init, cosine scheduler, 5 epochs).

As you can see, Adam drops the loss very fast from the first epochs. SGD is slower but more steady. With only 5 epochs, this difference is very clear.

3.2 Generalization Gap

Figure 2 shows the generalization gap. This is simply validation loss minus training loss. A positive value means the model fits the training data better than the validation data.

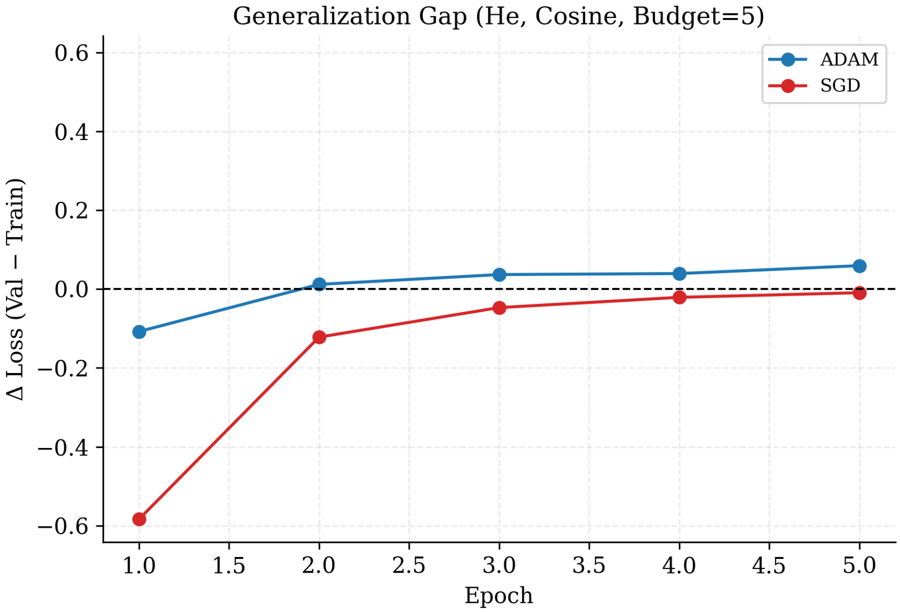


Figure 2: Generalization gap for SGD and Adam (5 epochs).

Adam's gap gets bigger as epochs go on. It fits the training data faster, but the gap with validation grows. SGD's gap stays more stable.

3.3 Test Performance

Figure 3 shows the macro F1 scores on the test set.

Both optimizers give high F1 scores. But the final score is not the interesting part. How they get there is more interesting. The loss curves and generalization gap tell us more about how the optimizers work.

3.4 Are Results Consistent?

The results I showed above are not just for one setting. As you can see in the Appendix, the same pattern appears in all initialization and scheduler combinations. Adam is always fast, SGD is always more stable. This is how these optimizers work.

4 Discussion

What I learned from these experiments is simple: optimizer choice matters, especially when you train with few epochs.

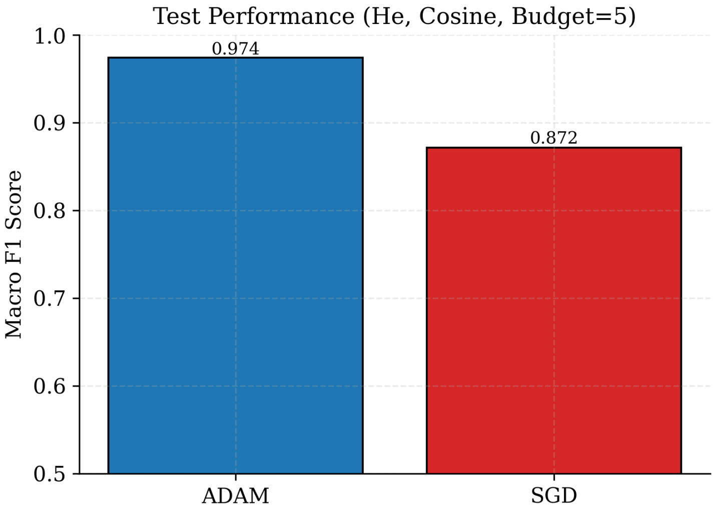


Figure 3: Test F1 scores for SGD and Adam (5 epochs).

Adam is fast. It drops the loss a lot in the first few epochs. But this speed has a cost: the generalization gap grows.

SGD is slow but more stable. It cannot reach a loss as low as Adam in few epochs, but the gap between training and validation is smaller.

In short: fast optimization does not always mean better generalization.

5 Conclusion

In this study, I answered the question I was curious about: how do SGD and Adam behave with limited epochs?

Adam converges fast but has a big generalization gap. SGD is slow but more balanced. Both have their place. Which one to choose depends on your training budget and what you need.

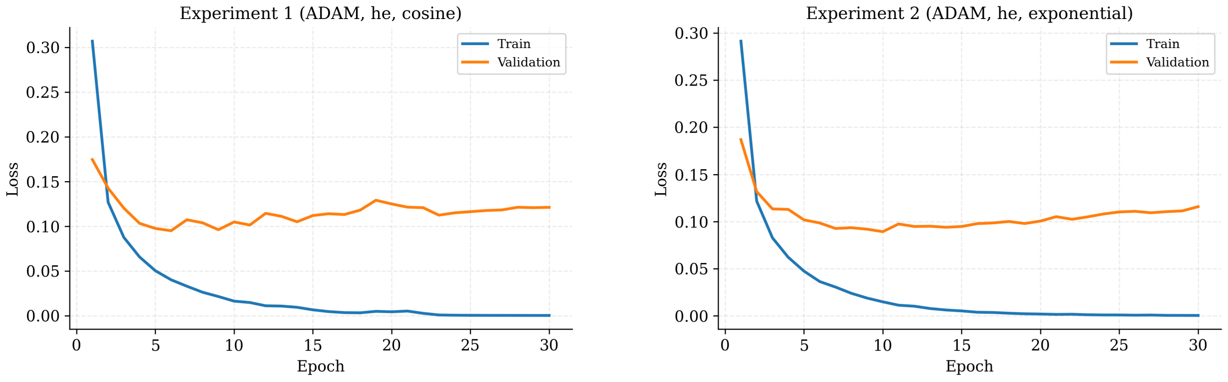
This was a simple experiment, but I learned a lot from it.

Code Availability

The full source code and experiment configurations used in this study are available at: `https://github.com/aytugyuruk/SGD-vs-Adam-Training-with-Limited-Epochs.git`

Appendix: Loss Curves for All Experiments

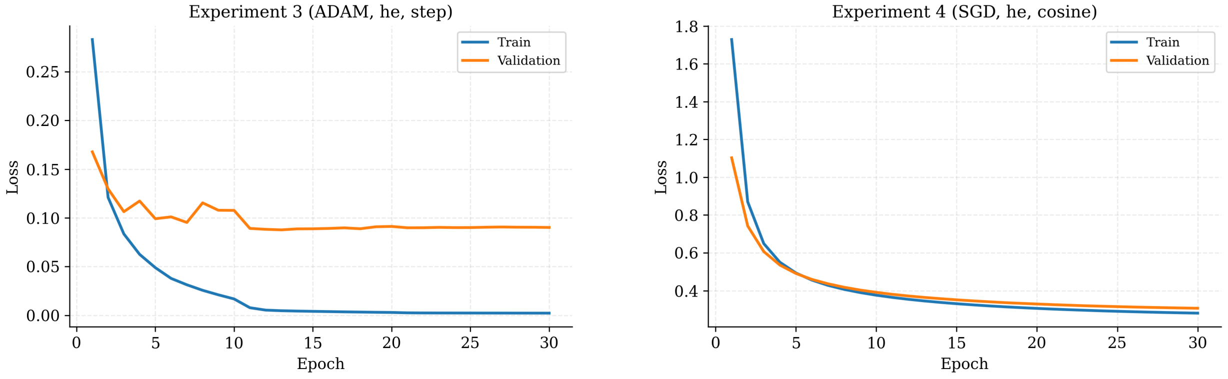
In this appendix, you can see the training and validation loss curves for all 24 experiments. I added this to show that the same pattern appears in all initialization and scheduler combinations.



(a) Experiment 1

(b) Experiment 2

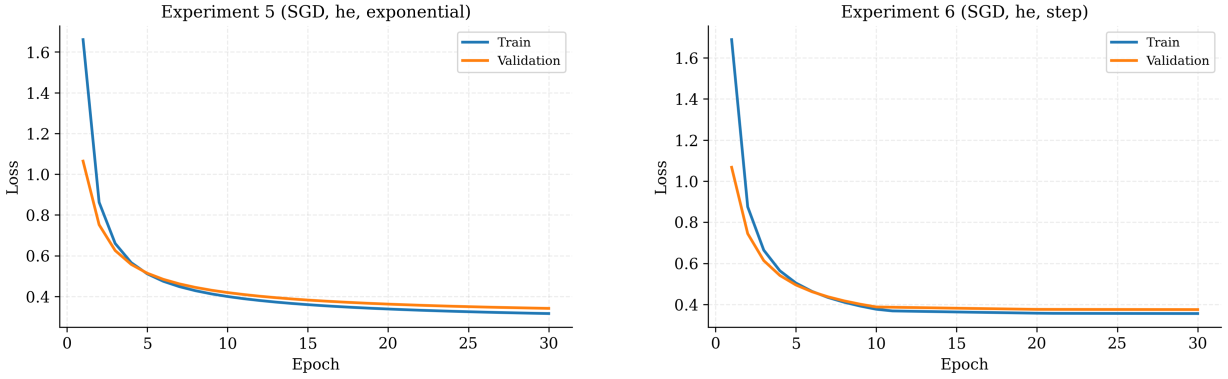
Figure 4: Training and validation loss curves for Experiments 1 and 2.



(a) Experiment 3

(b) Experiment 4

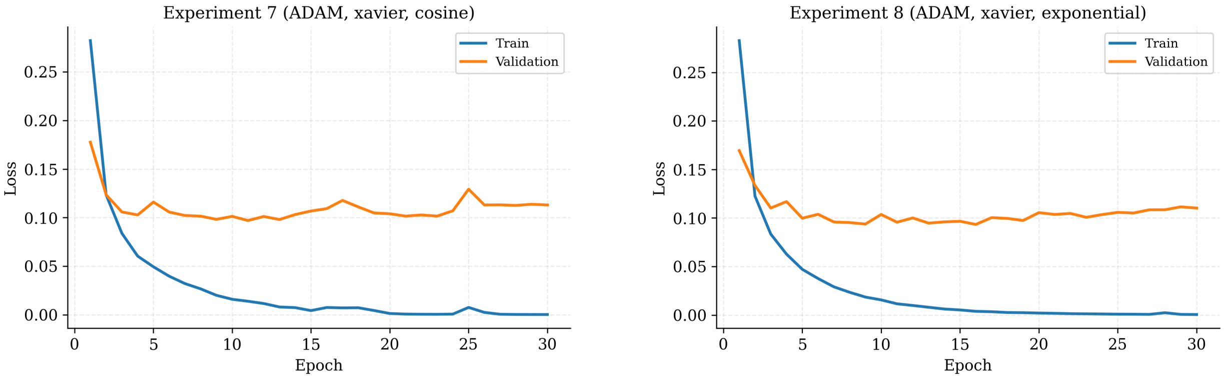
Figure 5: Training and validation loss curves for Experiments 3 and 4.



(a) Experiment 5

(b) Experiment 6

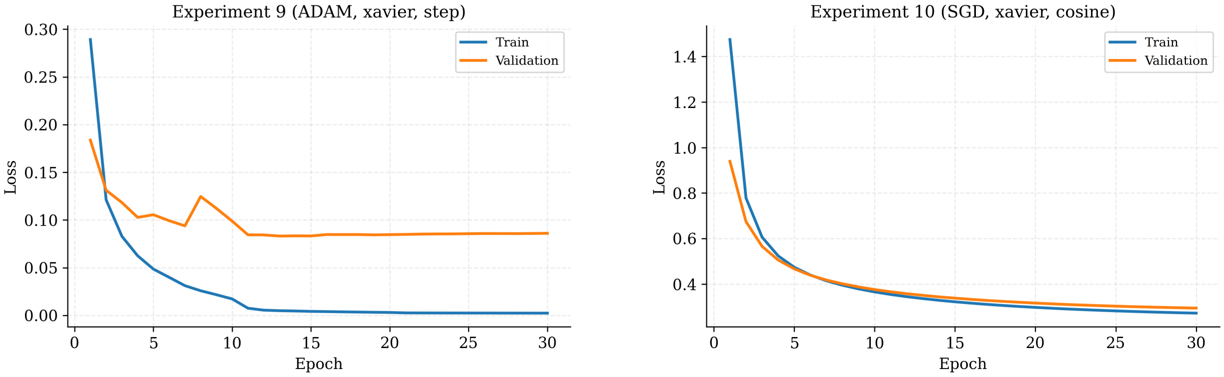
Figure 6: Training and validation loss curves for Experiments 5 and 6.



(a) Experiment 7

(b) Experiment 8

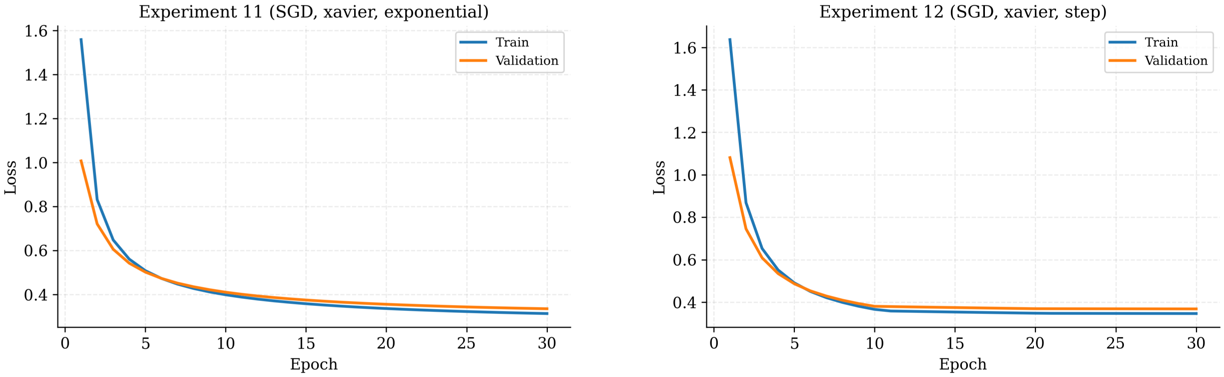
Figure 7: Training and validation loss curves for Experiments 7 and 8.



(a) Experiment 9

(b) Experiment 10

Figure 8: Training and validation loss curves for Experiments 9 and 10.



(a) Experiment 11

(b) Experiment 12

Figure 9: Training and validation loss curves for Experiments 11 and 12.