Introduction

I attempted to replicate the architecture of LeNet5, developed by Yann LeCun, trained and tested on the MNIST hand-written digit data set. I built two models, both with the same architecture. The first used the same optimizer and activation function as LeCun, but the second used more modern techniques. The difference in their performance was notable.

Models

Original: The architecture most closely resembles LeNet5. It uses the original activation functions and a similar optimizer.

Improved: This architecture is structurally identical to LeNet5, but it uses more modern activation functions and optimizer.

Model	Convolutional Layer Activation	Output Layer Activation	Optimizer	Epochs	Batch Size
Original	ReLU	Softmax	SGD (lr=0.001)	20	32
Improved	ReLU	Softmax	Adam (lr=0.001)	20	32

Results

Model	Training Accuracy	Testing Accuracy
Original	0.9706	0.9671
Improved	0.9982	0.9901

Notes

I chose to use a softmax function for both output layers to keep the results consistent, since I wanted to examine the effects of the optimizer used. I also used sigmoid activation functions in the original model (since that's what LeCun used), but the performance of that model suffered tremendously (accuracy never exceeded 11%).

Discussion

While both models developed relatively high accuracy, the one with the modern optimizer (Adam) learned much more quickly. Here's a table to show the effect:

Epoch	**Original Model Training Accuracy**	**Improved Model Training Accuracy**
1	0.4041	0.9511
2	0.8186	0.9819
3	0.8891	0.9874
4	0.9064	0.9899
5	0.9183	0.9923
6	0.9268	0.9954
7	0.9335	0.9960