

Table 1: Results at End of Training for All Networks

Description	Training Loss	Testing Loss	Training Accuracy	Testing Accuracy
Original Network	0.924	0.914	68.3	69.6
Remove Dropout Layer	0.767	0.876	73.6	70.3
Additional Convolutional Layer	0.781	0.913	73.0	69.2
Adagrad Adaptive Scheduling	0.794	0.802	72.0	71.9
Leaky ReLU	0.928	0.922	68.1	68.7
Gaussian Dropout	0.930	0.910	68.3	69.3

HW3

Class: CS519 - Deep Learning

Name: Ammar Kothari

Date: 03/10/17

(1) Remove Dropout Layer

Reference Figure 1

Reference Figure 2

(2) Additional Fully Connected Layer with 512 filters

Reference Figure 3

Reference Figure 4

(3) Adaptive Schedule Learning Rate Tuning

I implemented Adagrad with the original network structure.

Reference Figure 5

Reference Figure 6

(4) Additional Tuning Methods

Method 1: Replace ReLU with Leaky ReLU

Leaky ReLU implements a small gradient value when the activation is zero. This can help with convergence.

Reference Figure 7

Reference Figure 8

Method 2: Replace Dropout with Gaussian Dropout

I replaced the dropout with Gaussian dropout layers. I changed the values from $[0.25, 0.25, 0.50]$ to $[0.5, 0.5, 0.75]$ for the layers.

Reference Figure 9

Reference Figure 10

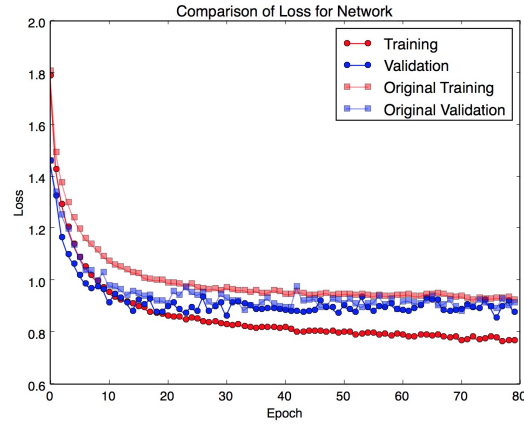


Figure 1: Loss During Training with Drop Out Layer Removed

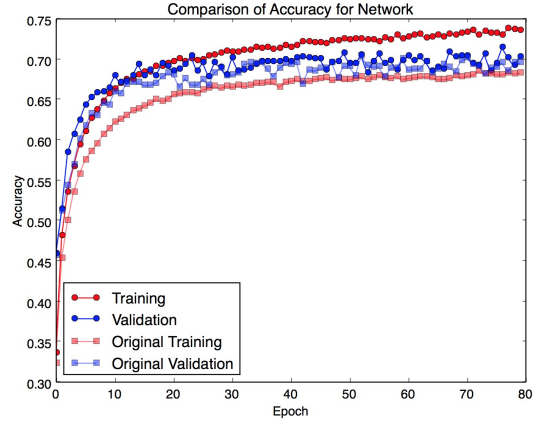


Figure 2: Accuracy During Training with Drop Out Layer Removed

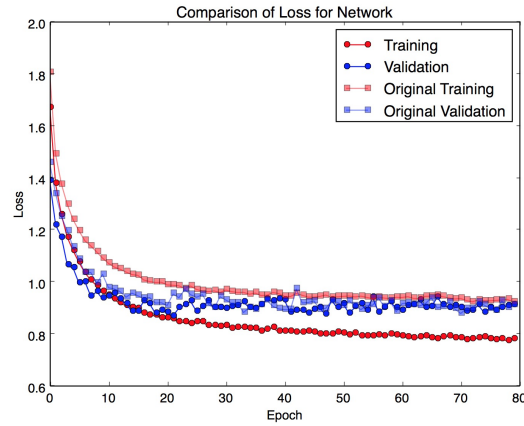


Figure 3: Loss During Training with Additional Convolutional Layer

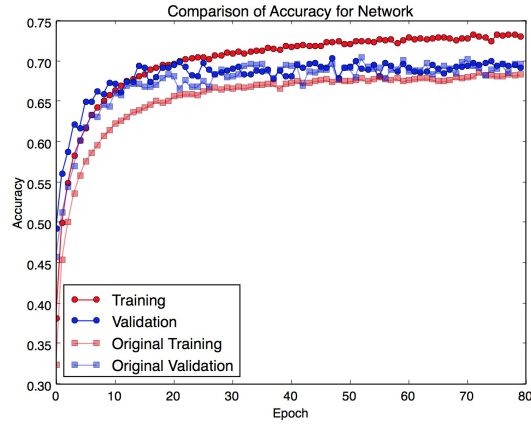


Figure 4: Accuracy During Training with Additional Convolutional Layer

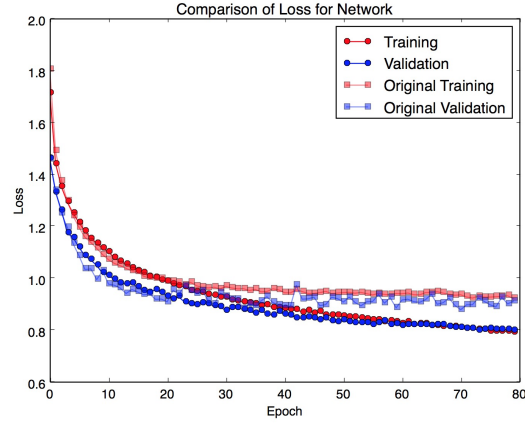


Figure 5: Loss During Training with Adaptive Scheduling to Tune Learning Rate

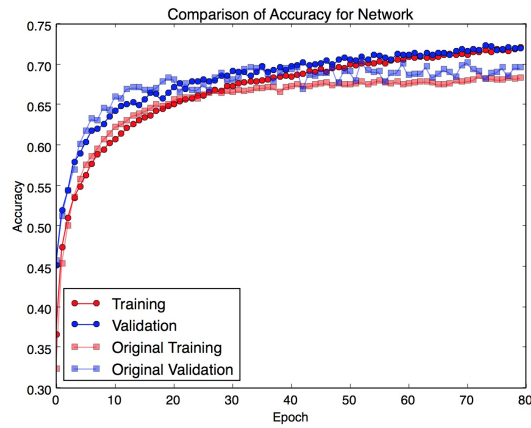


Figure 6: Accuracy During Training with Adaptive Scheduling to Tune Learning Rate

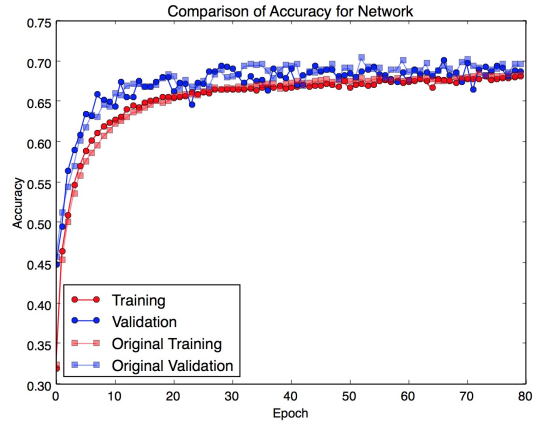


Figure 7: Loss During Training with Leaky ReLU

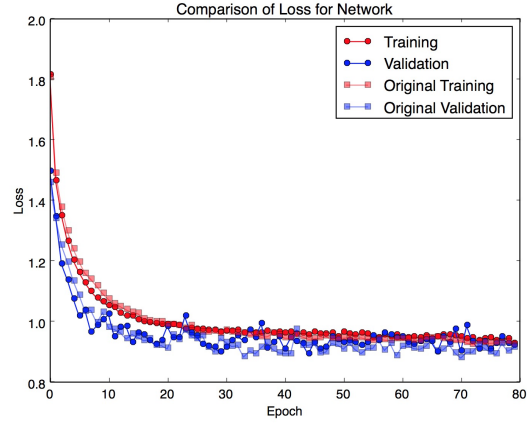


Figure 8: Accuracy During Training with Leaky ReLU

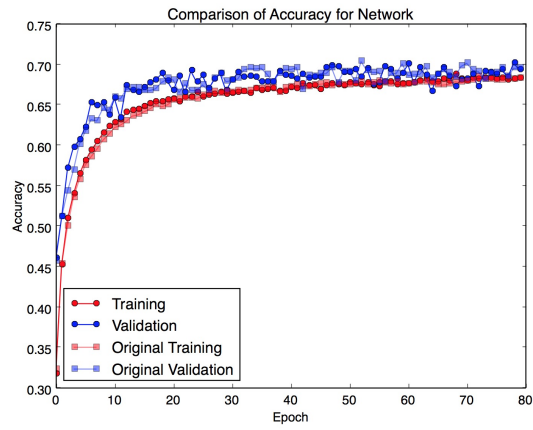


Figure 9: Loss During Training with Gaussian Dropout

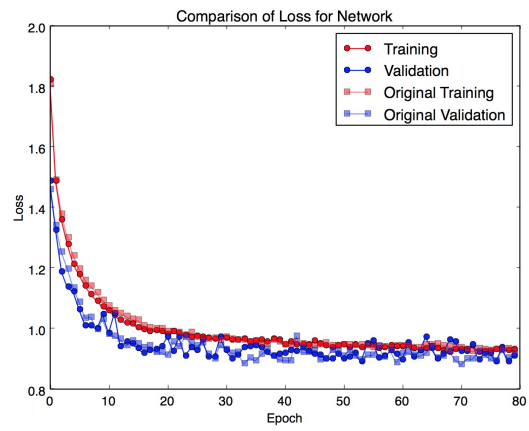


Figure 10: Accuracy During Training with Gaussian Dropout