1. **Overview**

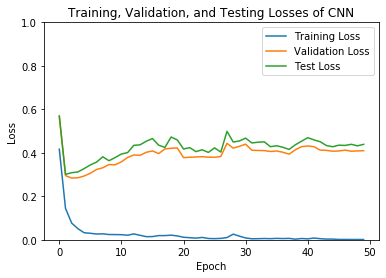
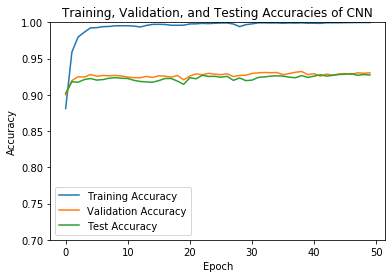
This section implements a convolutional neural network (CNN) for image recognition on the notMNIST dataset that we have been exploring with so far. This neural network is implemented with TensorFlow and Keras. The neural network architecture involves one convolutional layer with 32 3 x 3 filters. Each filter has a stride of 1, and passes its output through a ReLU activation function. This is followed by a 2 x 2 maxpooling layer, and 2 fully connected layers: one with 784 neurons, and the output layer which as 10 neurons that feed into a final softmax output function layer to give the prediction. Batch normalization is also used in the model following the convolutions. The accuracy and losses of each modified version is given below. All the derivations of the equations and theory involved are similar to those above with the exception of the convolutions, but that’s not within the scope of this project.

**2.1 – 2.2 Baseline Model**

We let the model train for 50 epochs on a batch size of 32 with the Adam optimizer for each of the tests. A constant learning rate of 10e-4 and cross entropy loss are used. The final training, validation, and test losses and accuracies are recorded below:

The Losses and Accuracies of the Baseline Model

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Loss | | | Accuracy (%) | | |
| Training | Validation | Test | Training (%) | Validation (%) | Test (%) |
| 0.0012 | 0.4099 | 0.4390 | 99.98 | 93.03 | 92.73 |

Figure 1: The loss and accuracy curves of the baseline model

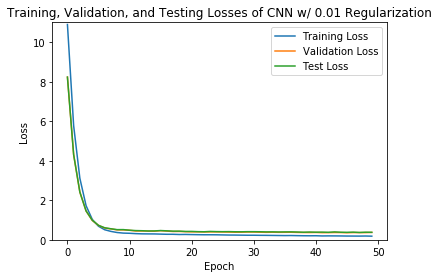
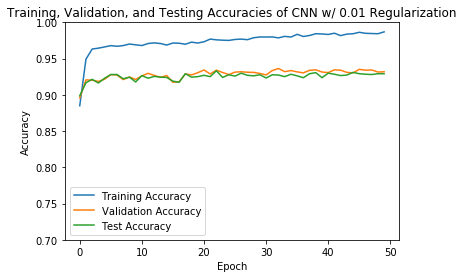
**2.3.1 Model with Regularization**

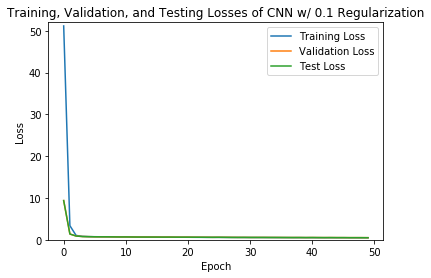
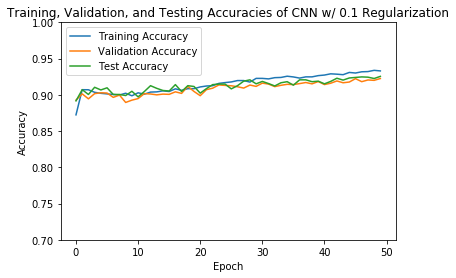
Regularization helps to prevent overfitting by applying a penalizing constant on a Euclidean norm of the weights. The regularization constants of 0.01, 0.1, and 0.5 were tested.

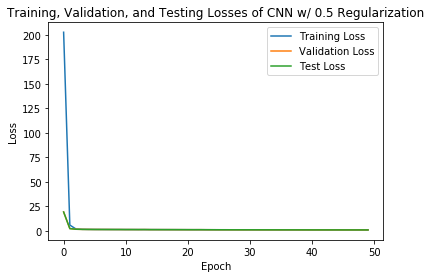
The accuracies of the model with regularization appear to converge faster than the baseline model (which did not have regularization). The validation and testing losses of the model appear to be significantly lower after training than the baseline. However, the accuracies for validation and test appear to be the very similar to the baseline for 0.01, indicating that the regularization did not have an effect on the model’s performance. However, the validation and test accuracies decreased as the regularization constant increased, indicating that the model started to overfit.

The Losses and Accuracies with Regularization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Regularization | Loss | | | Accuracy (%) | | |
| Training | Validation | Test | Training (%) | Validation (%) | Test (%) |
| 0.01 | 0.1834 | 0.3734 | 0.3787 | 98.69 | 93.18 | 92.91 |
| 0.1 | 0.4824 | 0.5054 | 0.5072 | 93.30 | 92.25 | 92.55 |
| 0.5 | 0.9095 | 0.9189 | 0.9050 | 89.57 | 88.57 | 89.50 |

Figure 2: The loss and accuracy curves of the model with 0.01 regularization

Figure 3: The loss and accuracy curves of the model with 0.1 regularization

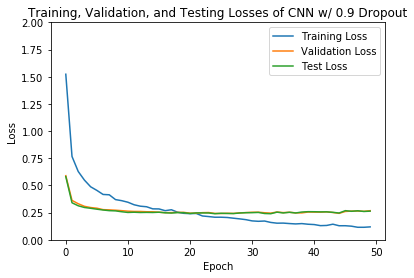
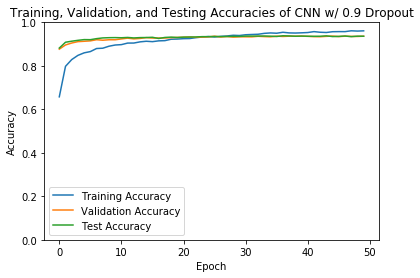
Figure 4: The loss and accuracy curves of the model with 0.5 regularization

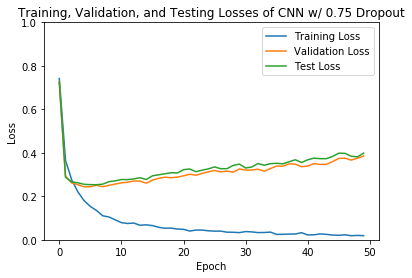
**2.3.2 Model with Dropout**

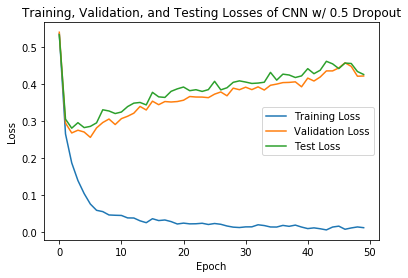
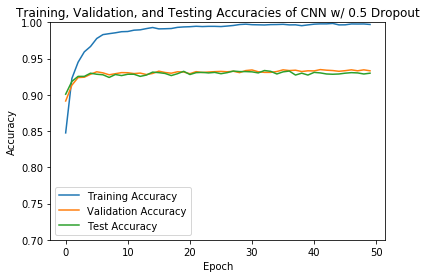
Dropout helps to prevent overfitting by applying a probability *p* of a neuron shutting down in a layer to reduce over examination of features. The dropout constants of 0.9, 0.75, and 0.5 were tested. The dropout constants allowed the model to converge faster than both the baseline and the model with regularization. However, it produced very similar validation and test accuracies as the baseline, indicating no effect.

The Losses and Accuracies with Dropout

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Dropout | Loss | | | Accuracy (%) | | |
| Training | Validation | Test | Training (%) | Validation (%) | Test (%) |
| 0.9 | 0.1188 | 0.2678 | 0.2635 | 96.10 | 93.63 | 93.61 |
| 0.75 | 0.0190 | 0.3851 | 0.3987 | 99.56 | 93.50 | 93.54 |
| 0.5 | 0.0119 | 0.4210 | 0.4251 | 99.71 | 93.32 | 92.99 |

Figure 5: The loss and accuracy curves of the model with 0.9 dropout

Figure 6: The loss and accuracy curves of the model with 0.75 dropout

Figure 7: The loss and accuracy curves of the model with 0.5 dropout