

Deep Learning – HW 1

Model Architecture:

Layers structure:

1. Input layer (size 784),
2. hidden layer (size 75),
3. hidden layer (size 32),
4. output layer (size 10)

For each layer that isn't the output layer we used batch norm and dropout.

Num of epochs: 150 (with test based early stopping)

Optimizer: `torch.optim.Adam`

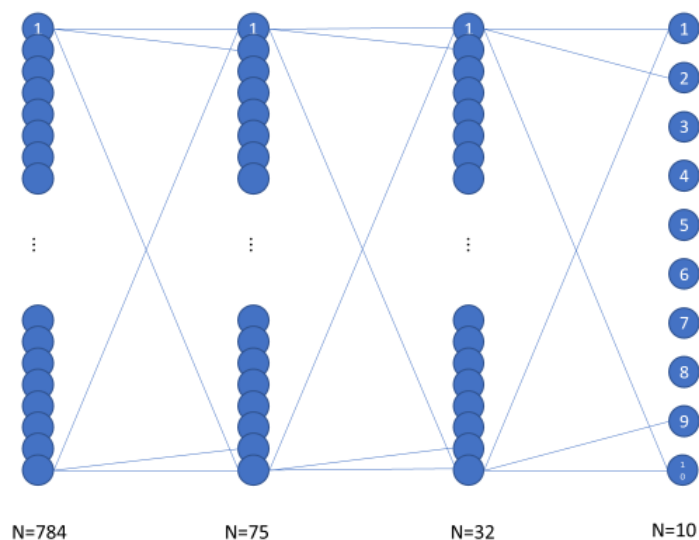
Loss function: `nn.CrossEntropyLoss`

Activation function: `nn.ReLU()`

Learning rate: 0.00005

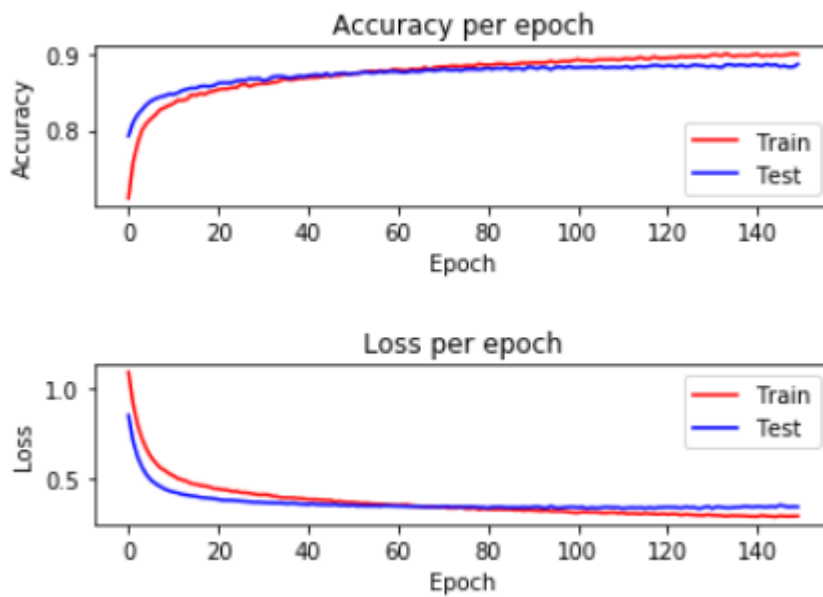
Batch size: 128

Dropout rate: 0.4



Fashion-MNIST fully connected network
(for readability purposed, not all connections are drawn)

Training progress:



Train accuracy: 0.8997

Test accuracy: 0.8871

Train loss: 0.3813

Test loss: 0.3433

Summary: We used multiple variations on each of the hyper parameters, using one and two layers. The greatest improvement was when switch from SGD optimizer to Adam. Most architectures gave similar accuracy, in the area of 0.85-0.87. We chose the best model we got, which proved to be with the parameters above.

It seems like there is some overfitting to the train data, so possibly a higher dropout rate could have helped, though we weren't able to get a model with higher performance for dropout 0.5.