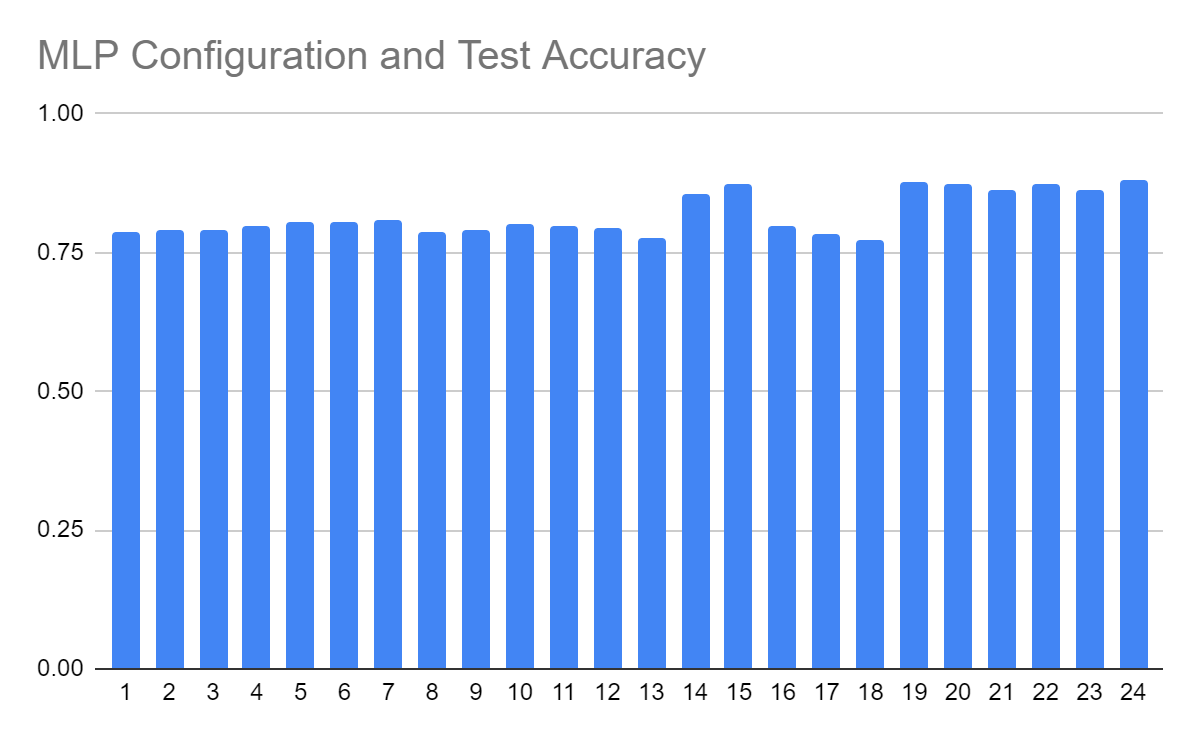
Discussion:

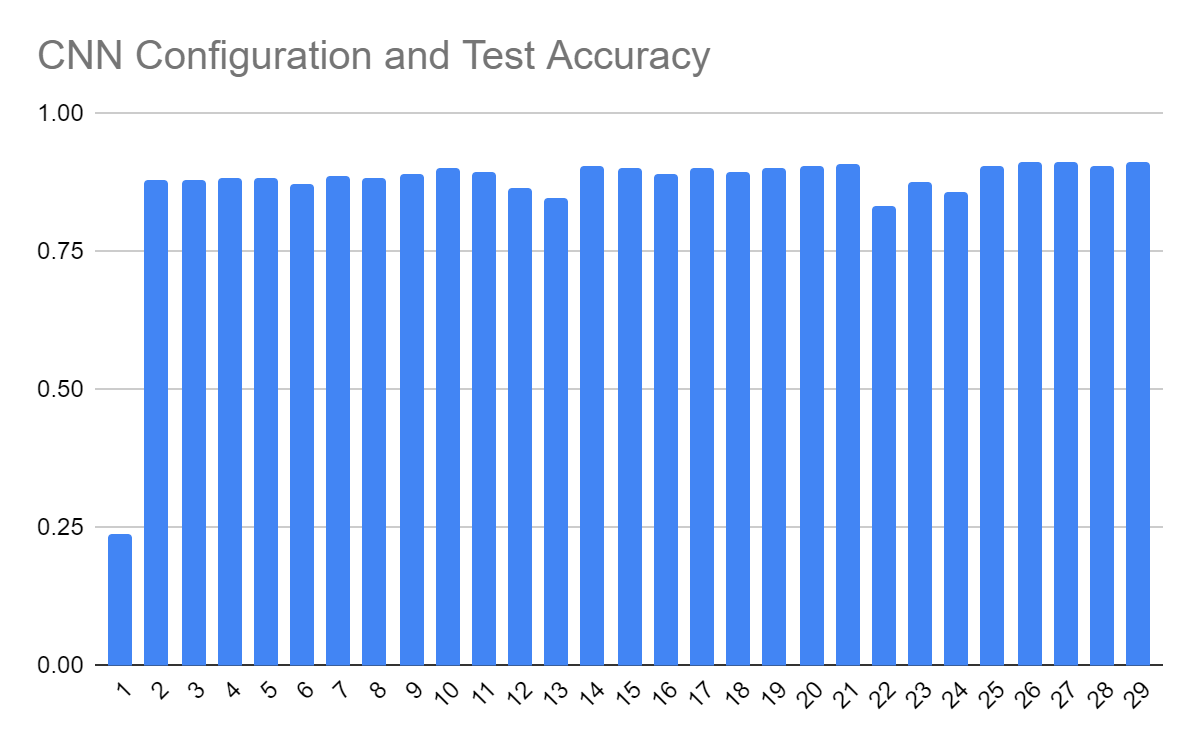
1. My intuition is that there is much greater intraclass variety in the FashionMNIST dataset. Distinguishing between different tops/dresses and different shoes (sneaker vs boot) looks much more difficult than digits.

| Layer | Activation Map  Dimensions | Number of Weights | Number of Biases |
| --- | --- | --- | --- |
| Input | 28x28x1 | 0 | 0 |
| Conv2D(64, (5, 5), input\_shape=(28, 28, 1)) | 24x24x64 | 1,600 | 64 |
| BatchNormalization() | 24x24x64 | 64 | 64 |
| Conv2D(64, (3, 3)) | 10x10x64 | 36,864 | 64 |
| BatchNormalization() | 10x10x64 | 64 | 64 |
| Dense(128) | 128x1 | 204,800 | 128 |
| BatchNormalization() | 128x1 | 128 | 128 |
| Dense(64) | 64x1 | 8,192 | 64 |
| BatchNormalization() | 64x1 | 64 | 64 |
| Dense(10, activation='softmax') | 10x1 | 640 | 10 |

1. Note: I did not initially think to measure training time for each model. I went back and measured training times for some notable models and added some configurations as a result of the time testing and thinking more about overfitting.



This chart shows the accuracy for the different MLP configurations as I experimented and tried to improve model performance.The best performing MLP model was Configuration 24, which had a test accuracy of 0.8814.



This chart shows the test accuracy for the different CNN configurations as I experimented and tried to improve model performance. You can see that switching the activation functions to ReLU in the second configuration was by far the most impactful change. Marginal improvements were made as changes in parameters and structure were experimented with. The best performing CNN model was Configuration 26, which had a test accuracy of 0.9144.

* 1. Table of notable models and training time/complexity

| Config | Training Time | Test Accuracy | Notes |
| --- | --- | --- | --- |
| MLP 15 | 268.45 | 0.8748 | This was initially thought to be a simple, fast to train, well performing model. |
| MLP 19 | 300.60 | 0.8775 | This was initially thought to be the best performing MLP model. |
| MLP 24 | 208.97 | 0.8814 | After thinking more about overfitting and training time, I tried MLP 19 with reduced epochs from 50 to 35 and this ended up being the best performing MLP model. Obviously, it was faster to train than the other two I measured as well. |
| CNN 26 | 332.22 | 0.9144 | This was my highest performing model. I noticed it had higher accuracy in training than testing though so I tried out Config CNN 29 below. |
| CNN 29 | 172.60 | 0.9141 | Same as CNN 26 but 13 epochs instead of 25. Basically equivalent performance, so prevented overfitting. |

1. My initial response to this question was:

In the MLP models, the best performing model I initially tried was 0.8775 test accuracy had:

* Adam
* two hidden layers of sizes 256 and 128
* learning rate of 0.0002

A model performed comparably well (0.8748 test accuracy) and had ~10% faster training time:

* SGD
* learning rate 0.01

So in this case, I do not think the added complexity/training time was worth it.

After thinking more about overfitting and complexity, as I explained in the table above, I tried the same Adam model above with 35 epochs instead of 50 and got slightly better performance than the other notable models. So within the context of the MLP models I tested, this was a well performing model and fine in terms of complexity and training time.

1. For the CNN’s, I ended up with a model that performed better than the MLP’s (0.9141 over 0.8814). It obviously was a more complex model, with more layers, different types of layers, etc., but the training time was actually shorter than the MLP. From the testing and configurations I tested, that makes it seem worth the use of a CNN. As I noted above, a more thorough investigation of training time could have been done, and a more rigorous testing of the models would have provided more insight (multiple trials per configuration). There was a limit to the amount of time I could spend on the project.

My opinion on whether the use of a CNN is justified could be changed if there was an MLP model that performed similarly or better than the ones I found and were much faster in training. If in practice, forward pass took much longer for the CNN than the MLP, it may be worth it to use an MLP. And if the use of augmentation, which could create a significantly larger dataset, exacerbated these effects while improving MLP performance relative to CNN, that would be a reason to use an MLP over the CNN.