Classify the Fashion MNIST dataset

I. Network structure

The network structure is as follows:

Output layer: Fully connected layer with 10 units and Softmax activation
Dropout layer with dropout rate 0.5
Fully connected layer with 128 units and ReLU activation
Dropout layer with dropout rate 0.5
Max Pooling layer with pooling size (2,2)
Convolutional layer with kernel size (3,3) and 64 filters and ReLU activation
Convolutional layer with kernel size (3,3) and 32 filters and ReLU activation
Input layer

The loss function is crossentropy because this network is doing classification tasks. For the convolutional layers, they are applied because the inputs are images and CNN is appropriate for extracting features in images. The max pooling layer down samples the outputs of the convolutional layer and extracts the representative features in a local area. The dropout layer is there to prevent the model from overfitting. Then, there is a fc layer with 128 units with ReLU activation. Another dropout layer after the first fc layer is added to enhance the performance on test data. Finally, a fc layer with 10 units is the output layer. The softmax activation function is chosen there because this represents the probability of belonging to each class well.

II. Representative weights

Conv1:

```
[ 9.58229676e-02, -2.76303321e-01, -2.76835024e-01,
 -3.00649166e-01, -2.30080396e-01,  1.50841936e-01,
 -1.20491594e-01, -8.39929432e-02, -1.88205406e-01,
 -1.02004476e-01,  3.42011489e-02,  3.12019885e-01,
 -4.51001860e-02, -2.82510761e-02,  5.05110584e-02,
  1.05094509e-02,  2.16750607e-01, -1.78140730e-01,
 -2.43342400e-01,  3.42196040e-02,  2.48639271e-01,
  3.95838544e-02, -1.14597432e-01, -1.74285229e-02,
 -1.25430122e-01,  6.95797428e-02,  3.38096023e-02,
  2.16826320e-01,  1.94446798e-02,  1.31910875e-01,
 -1.73470259e-01, -8.13847259e-02]
```

Conv2:

```
[[ 0.23381016, -0.07038157, -0.11812461, ...,  0.09645521,
    0.02171107, -0.0893659 ],
  [-0.0252474 ,  0.03616355,  0.04712946, ...,  0.00053328,
   -0.09191035, -0.03250504],
  [-0.05069688,  0.02082075, -0.06874983, ...,  0.00636286,
    0.00062092,  0.01343429],
  ...,
  [-0.08699358, -0.05453112, -0.06094761, ...,  0.07421406,
    0.01397604,  0.03686296],
  [-0.09596881,  0.06741914, -0.09490271, ..., -0.05343196,
   -0.22528072, -0.04189927],
  [-0.15448894,  0.01800303, -0.07656598, ...,  0.02334251,
   -0.0656969 , -0.14896145]]
```

FC1:

```
[-0.02831301,  0.00753535, -0.02492589, ..., -0.00466957,
 -0.0148673 , -0.03321284],
[-0.01772079,  0.00675977,  0.00334916, ...,  0.02399326,
  0.01983376,  0.00214701],
[-0.01366444,  0.00619417,  0.00608234, ...,  0.00271581,
 -0.00716715,  0.02072268],
...,
[ 0.03045873,  0.01113274,  0.04205231, ...,  0.01984649,
 -0.04663104,  0.01396745]
```

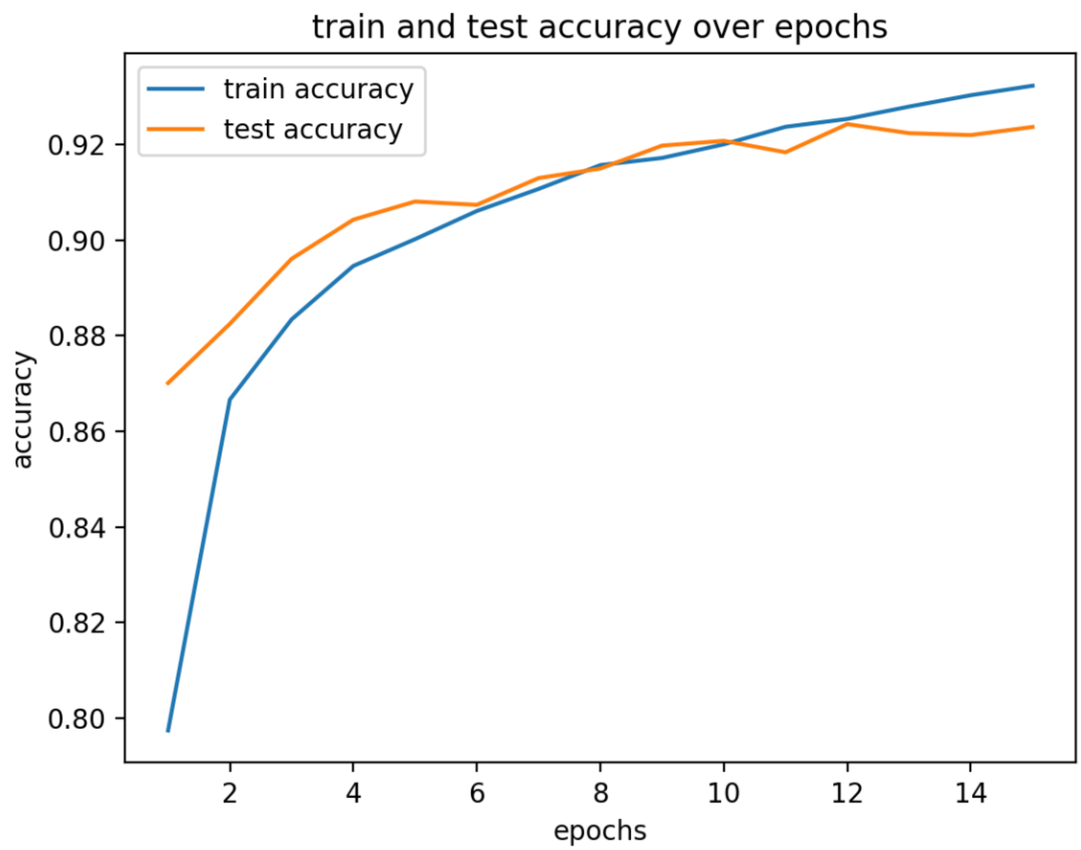
FC2:

```
[-0.2902708 ,  0.25061798, -0.26139852, ..., -0.35934612,
 -0.27775997, -0.40211487],
```

```
[ 0.06008262,  0.13959676,  0.13253024, ..., -0.47264102,
 -0.38761836,  0.01085826],
[ 0.18209827,  0.23196499, -0.21639547, ..., -0.3924969 ,
 -0.39228496, -0.12239689],
...,
[ 0.11513945,  0.01789363, -0.32515112, ..., -0.5328842 ,
 -0.10940735, -0.40739748]
```

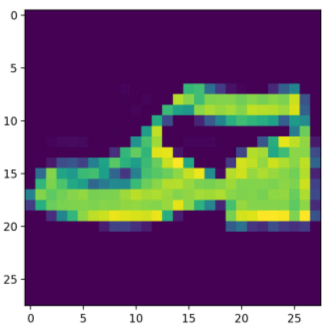
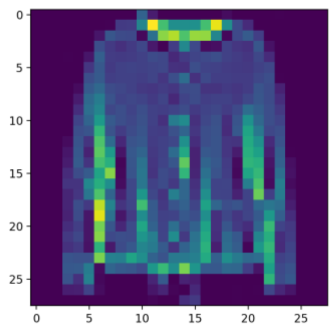
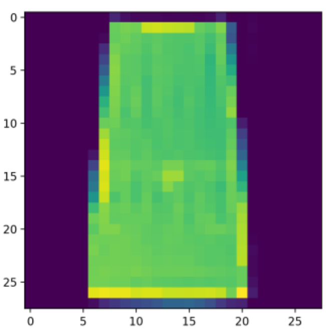
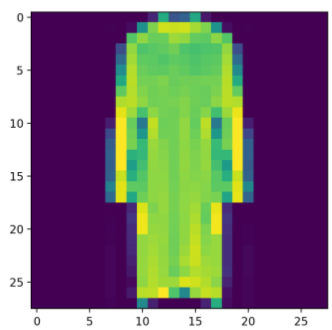
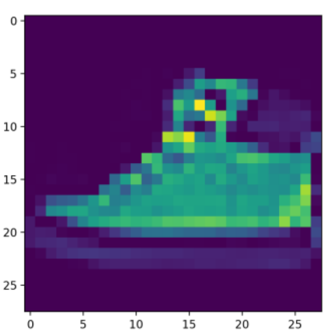
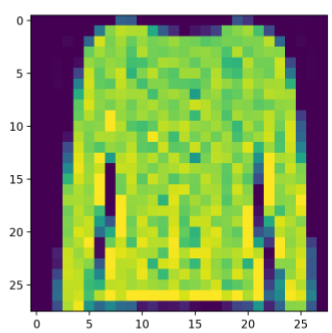
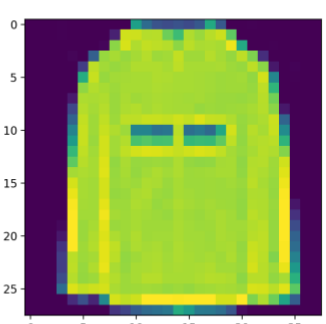
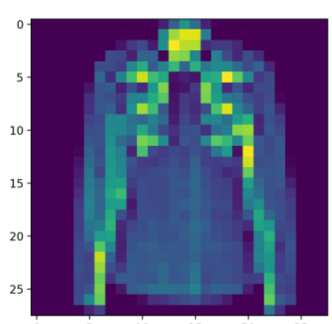
As we can see from the above prints, all of the weights are in the range [-1,1], which is a reasonable range. The weights are not tuned to be all zeros or identical, so they are likely to yield a good output.

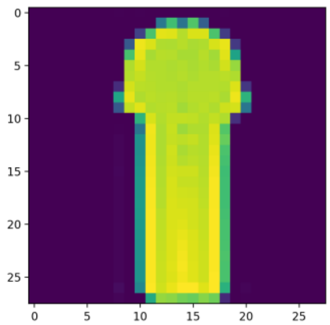
III. Training and testing accuracy vs epoch

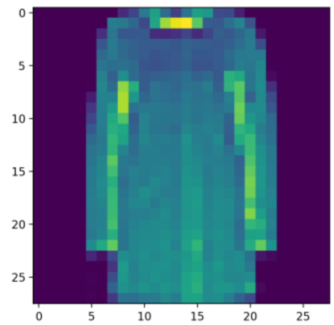


The training accuracy is increasing with the number of epochs, which is expected. Because of the dropouts, the model generalizes well so the test accuracy is higher than the training accuracy at first few epochs. However, the test accuracy cannot be improved in later epochs because of the overfitting.

IV. Data points that fail in the model


0 5 10 15 20 25
0 5 10 15 20 25

0 5 10 15 20 25
0 5 10 15 20 25

0 5 10 15 20 25
0 5 10 15 20 25

0 5 10 15 20 25
0 5 10 15 20 25

0 5 10 15 20 25
0 5 10 15 20 25

0 5 10 15 20 25
0 5 10 15 20 25

0 5 10 15 20 25
0 5 10 15 20 25

0 5 10 15 20 25
0 5 10 15 20 25





As we can see from the above misclassified images, some of them make no sense to be viewed as one of the clothing categories for humans, let alone the network. On the other hand, some of them lose pixel information in some areas and make it difficult for the network to correctly classify them.