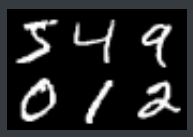
Report

A short description of the overall project in your own words. (200 words or less)

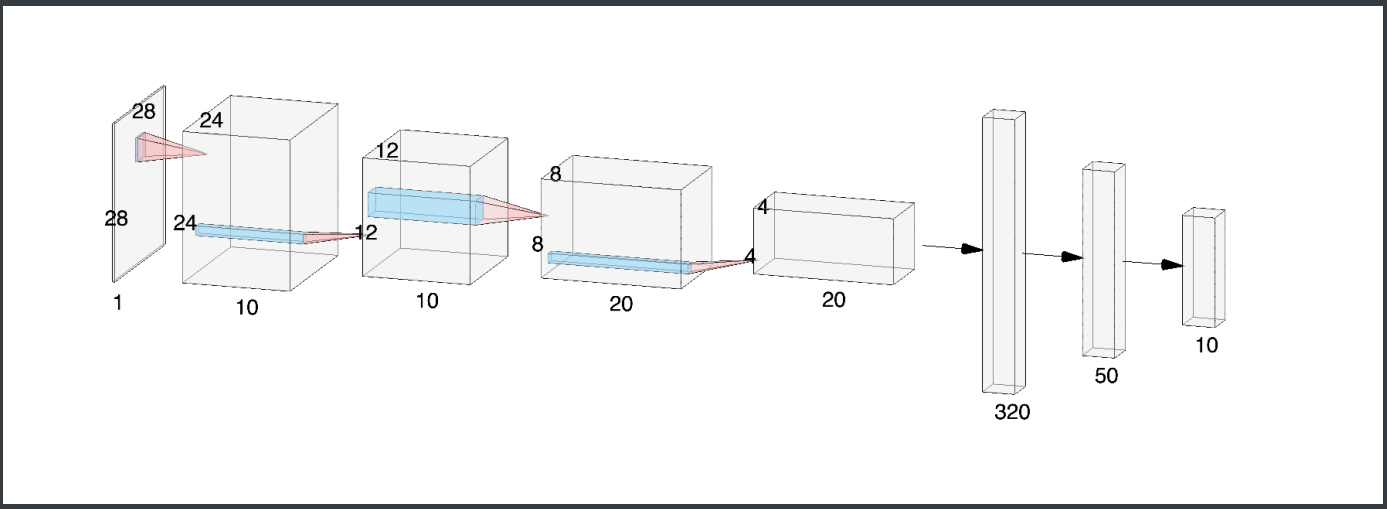
We build our first deep neural network according to instructions. Then we learned the process of training the model. After that, we examined our model. In addition, we did transfer learning to let our model recognize greek letters. After that, we carete our own datasets which are handwriten digits for letting the model to recognize. Finally, we did some experiments about the neural networks about dropout rates of the Dropout layer, activation function for each layer, number of epochs of training, batch size on the FASHION-MNIST.

Any required images along with a short description of the meaning of the image.

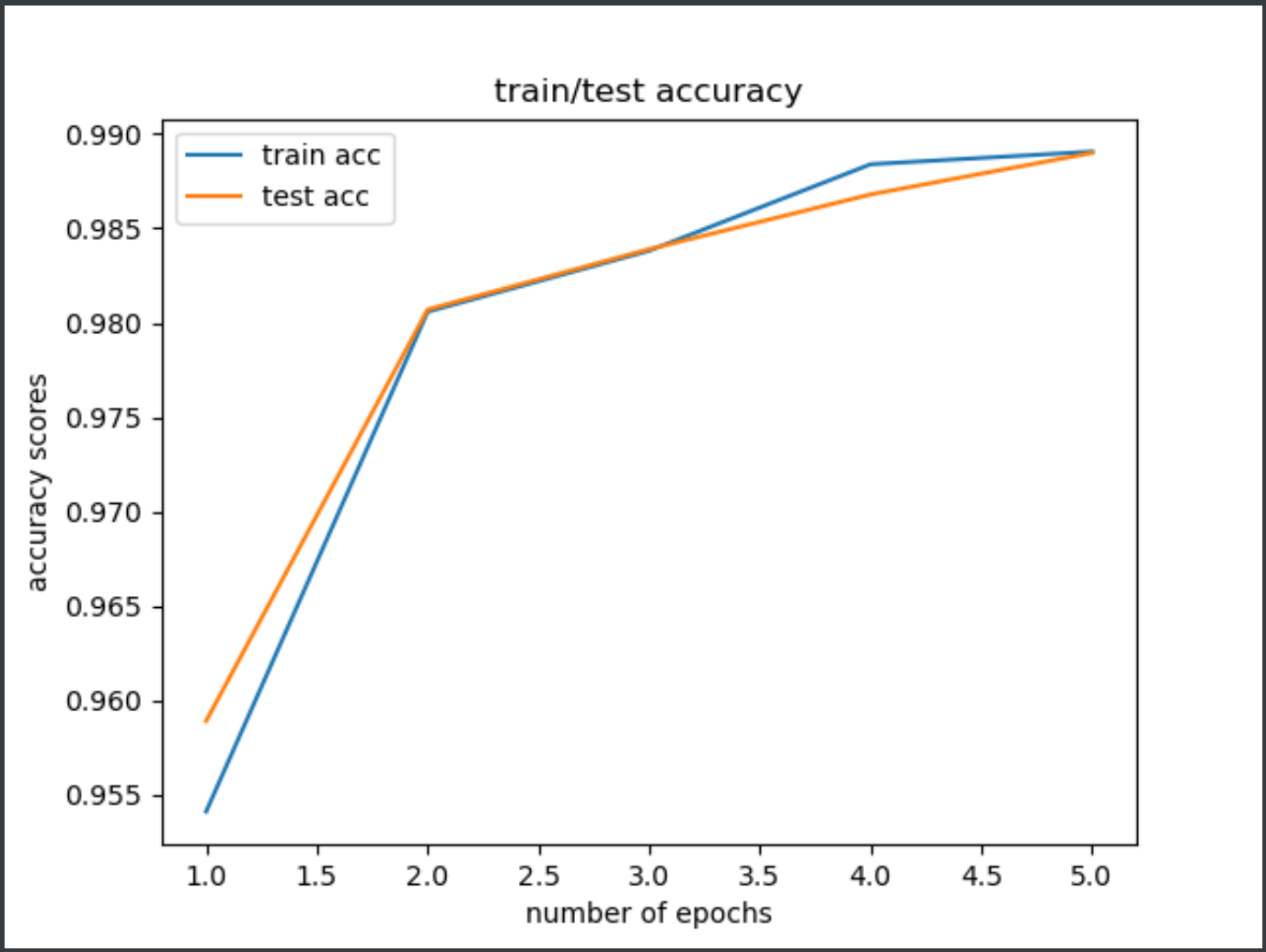
- task1-a

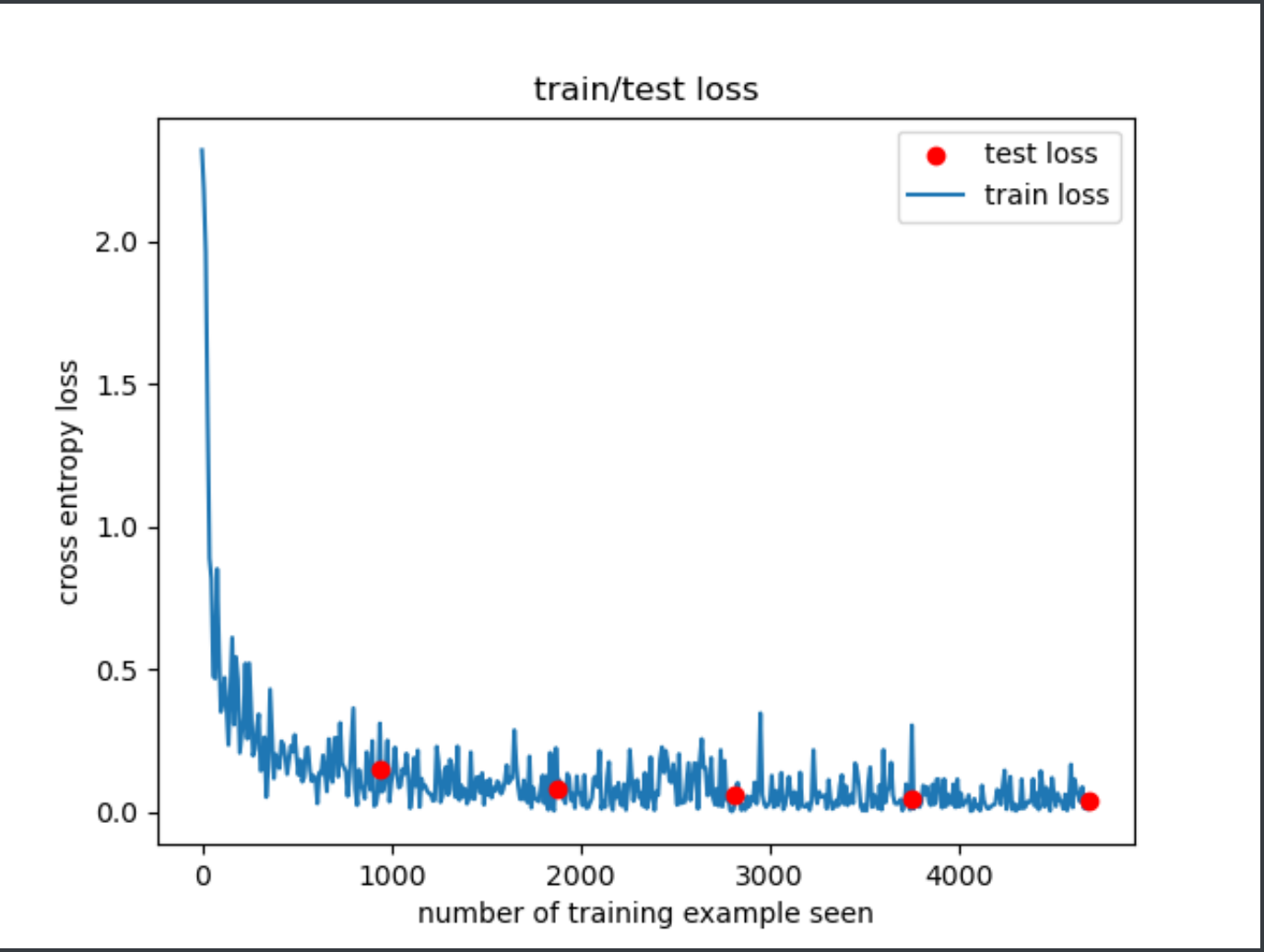


- task1-c


28
28
1
24
24
10
12
12
10
8
8
20
4
4
20
320
50
10

- task1-d

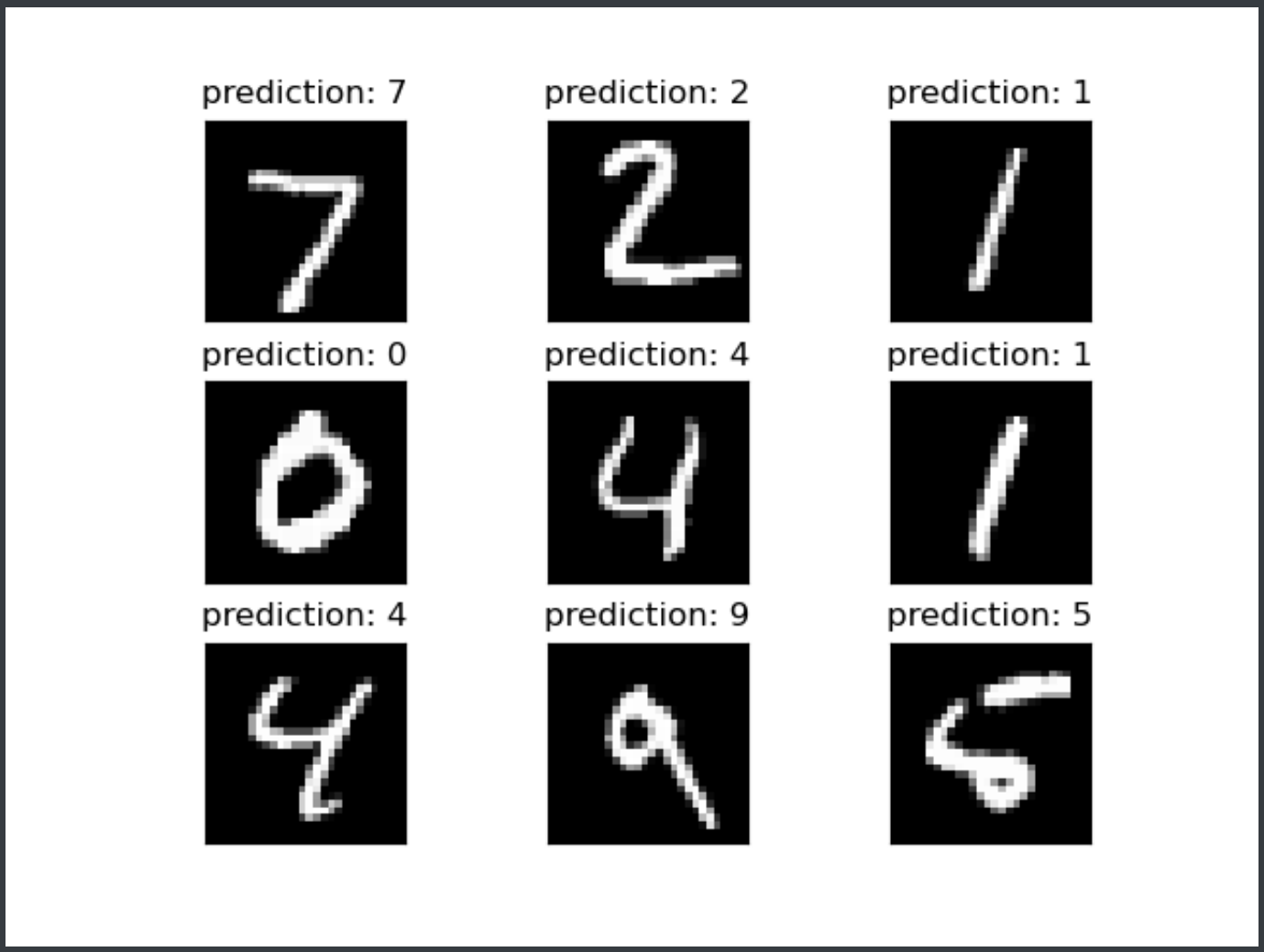

train/test accuracy
train acc
test acc
accuracy scores
0.990
0.985
0.980
0.975
0.970
0.965
0.960
0.955
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
number of epochs



- task1-f

  We use the MNIST training dataset for model prediction

```
************************ Example 1 ************************
output values:       -12.37 -10.33 -8.59 -10.02 -15.96 -13.42 -24.47 -0.0 -13.78 -11.11
max output index:    7
correct label:       7
************************ Example 2 ************************
output values:       -7.2 -9.29 -0.0 -12.34 -11.78 -13.41 -11.43 -15.07 -10.13 -19.29
max output index:    2
correct label:       2
************************ Example 3 ************************
output values:       -9.68 -0.0 -10.05 -12.36 -6.76 -9.43 -10.15 -7.32 -9.22 -9.64
max output index:    1
correct label:       1
************************ Example 4 ************************
output values:       -0.0 -16.46 -10.66 -17.0 -13.5 -9.31 -8.36 -11.35 -12.27 -11.89
max output index:    0
correct label:       0
************************ Example 5 ************************
output values:       -10.05 -13.49 -11.46 -12.95 -0.01 -10.35 -11.47 -9.87 -10.29 -4.95
max output index:    4
correct label:       4
************************ Example 6 ************************
output values:       -11.27 -0.0 -11.6 -14.67 -7.59 -11.98 -13.0 -8.22 -10.46 -11.72
max output index:    1
correct label:       1
************************ Example 7 ************************
output values:       -16.36 -10.43 -9.17 -12.88 -0.01 -10.35 -13.96 -8.43 -4.71 -6.38
max output index:    4
correct label:       4
************************ Example 8 ************************
output values:       -12.09 -11.23 -6.95 -7.3 -2.64 -4.92 -15.06 -7.78 -5.38 -0.09
max output index:    9
correct label:       9
************************ Example 9 ************************
output values:       -12.24 -15.44 -9.6 -13.3 -10.9 -0.01 -5.59 -11.39 -5.04 -8.62
max output index:    5
correct label:       5
************************ Example 10 ************************
output values:       -14.29 -15.99 -10.12 -9.44 -6.25 -10.45 -19.7 -4.91 -6.28 -0.01
max output index:    9
correct label:       9
```



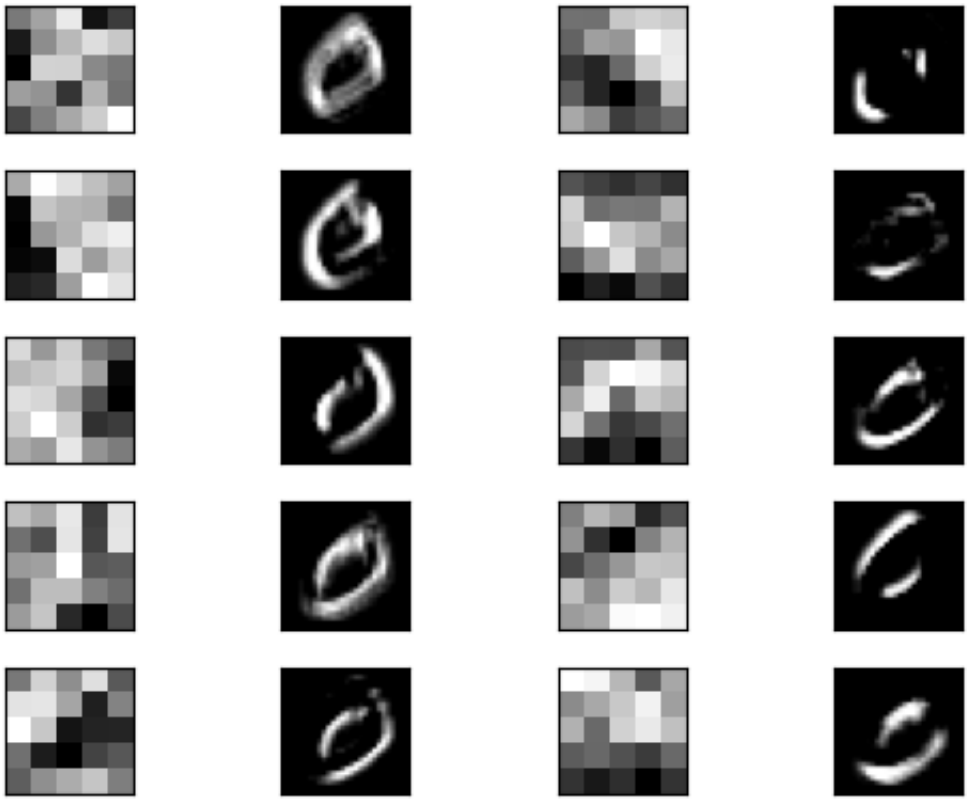
- task1-g

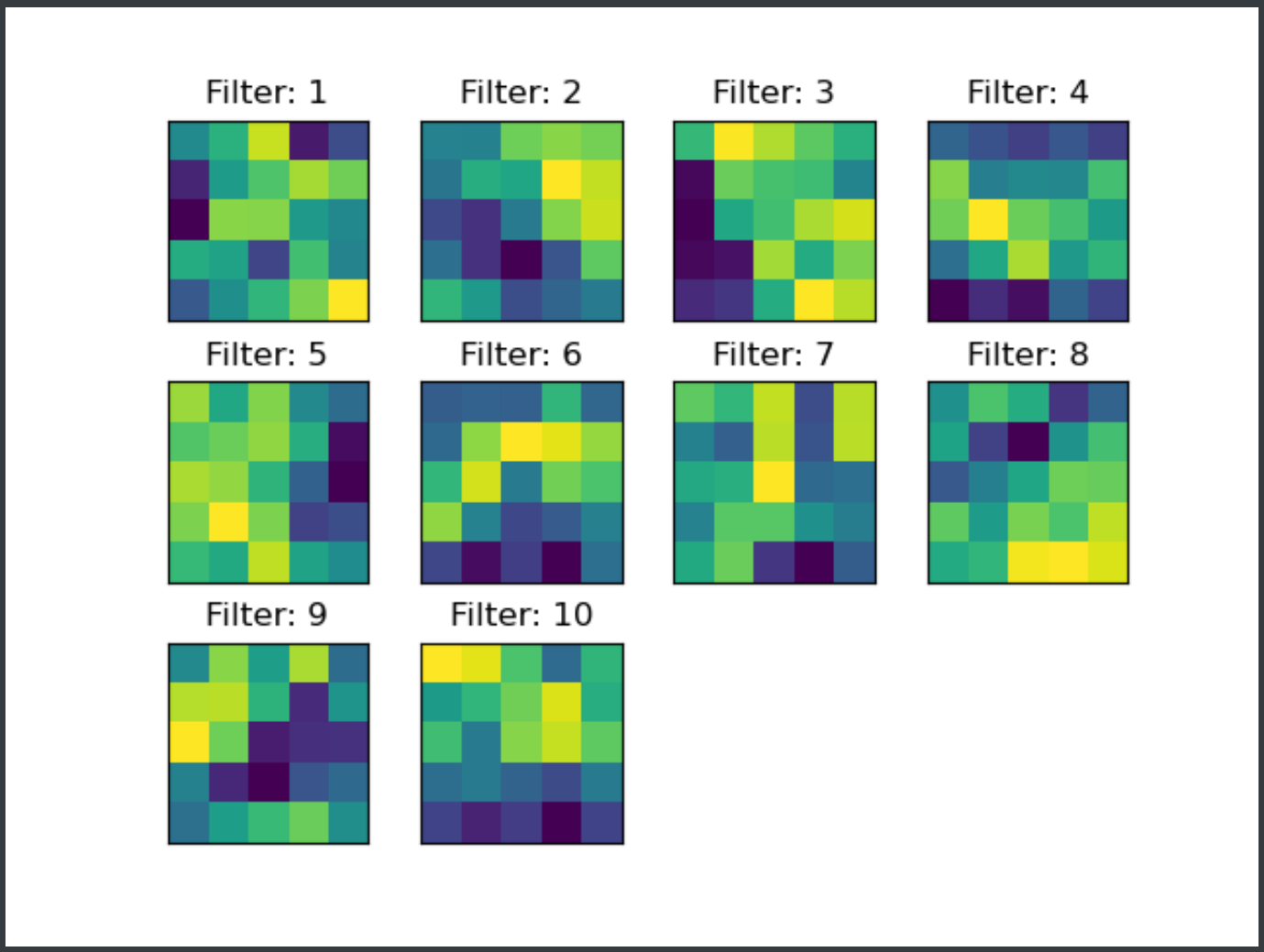
  The results of prediction of our handwritten inputs

```
data label:  [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
pred result:  [9, 1, 2, 3, 4, 3, 5, 7, 8, 9]
pred accuracy: 70.0%
```

- task2

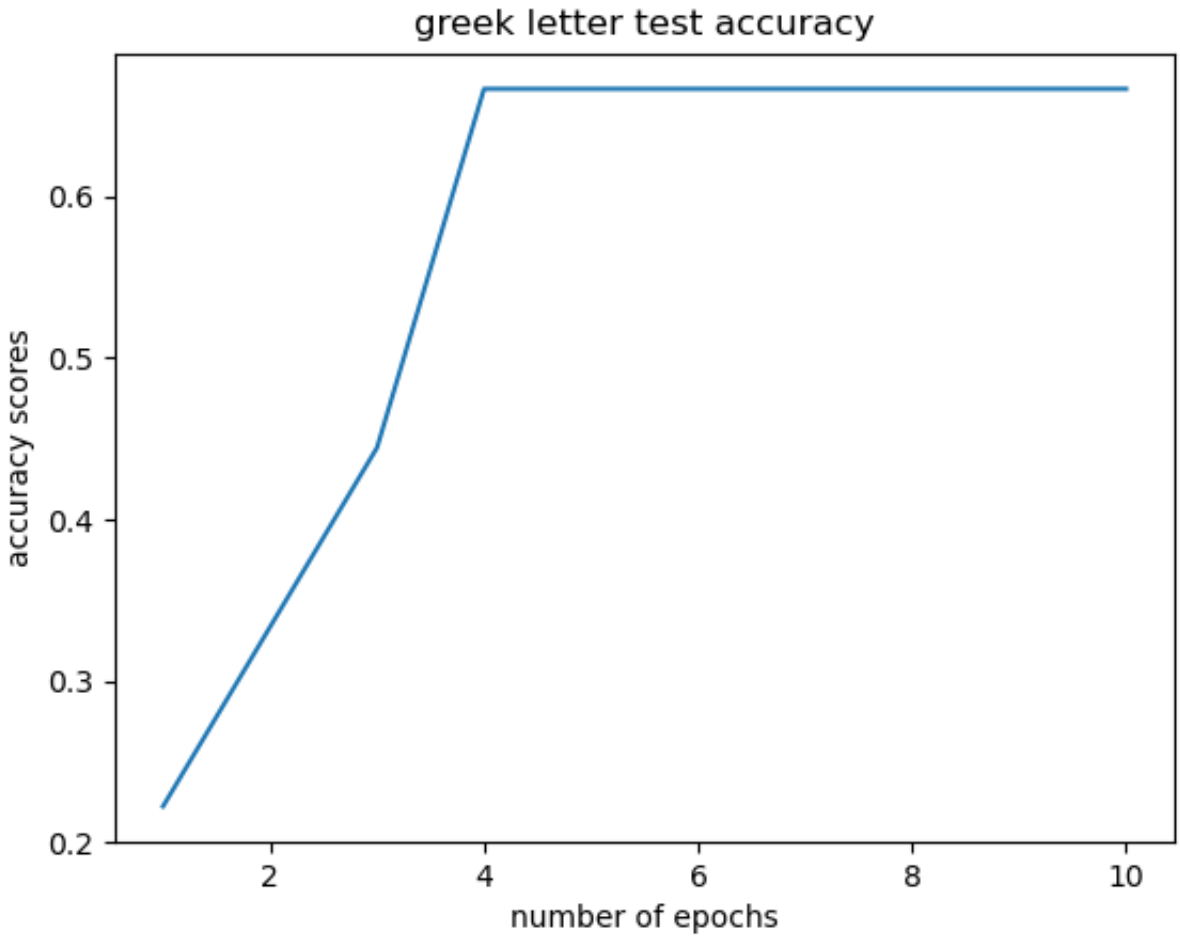
  that's the filter effect and layer weights. They all make sense

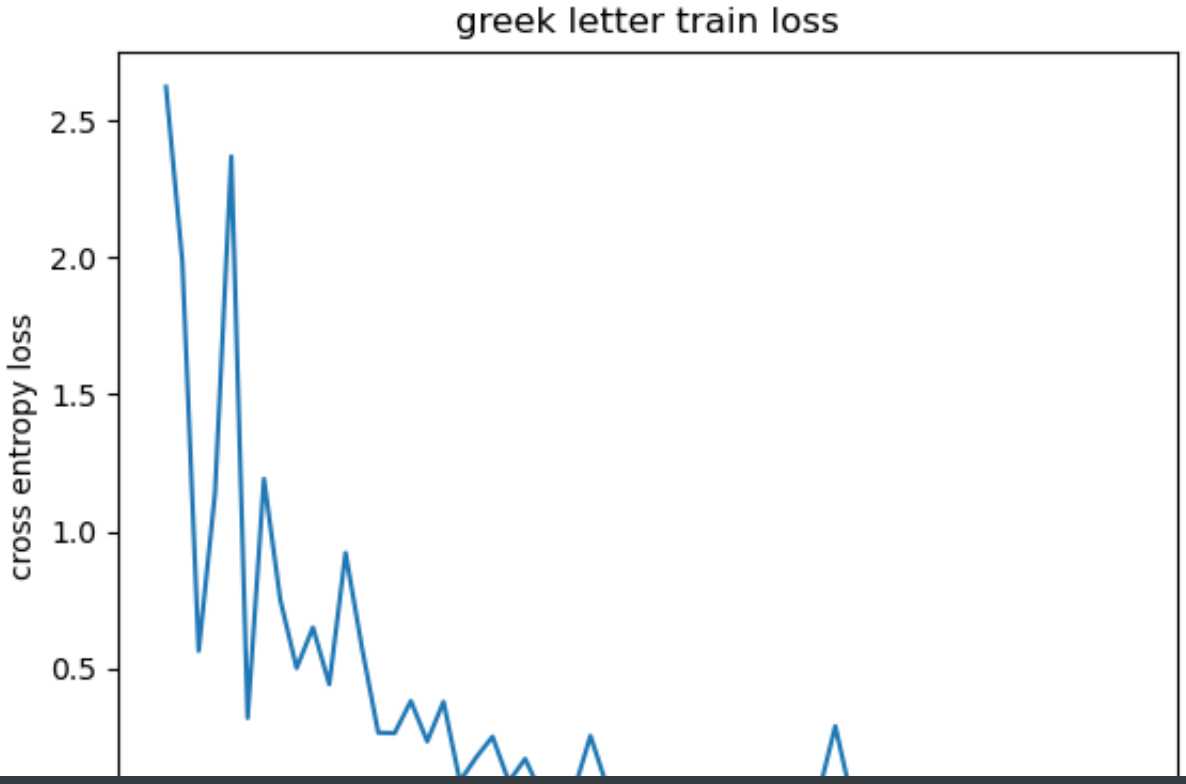


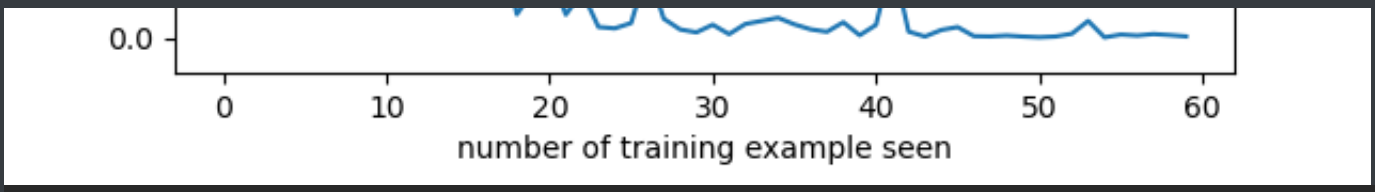


- task3

  These are results from greek letters transfer learning. The additional data are in data folder. It only takes 2 ephochs to let the model 100% recognize them.


greek letter test accuracy
accuracy scores
0.2
0.3
0.4
0.5
0.6
2
4
6
8
10
number of epochs


greek letter train loss
cross entropy loss
0.5
1.0
1.5
2.0
2.5

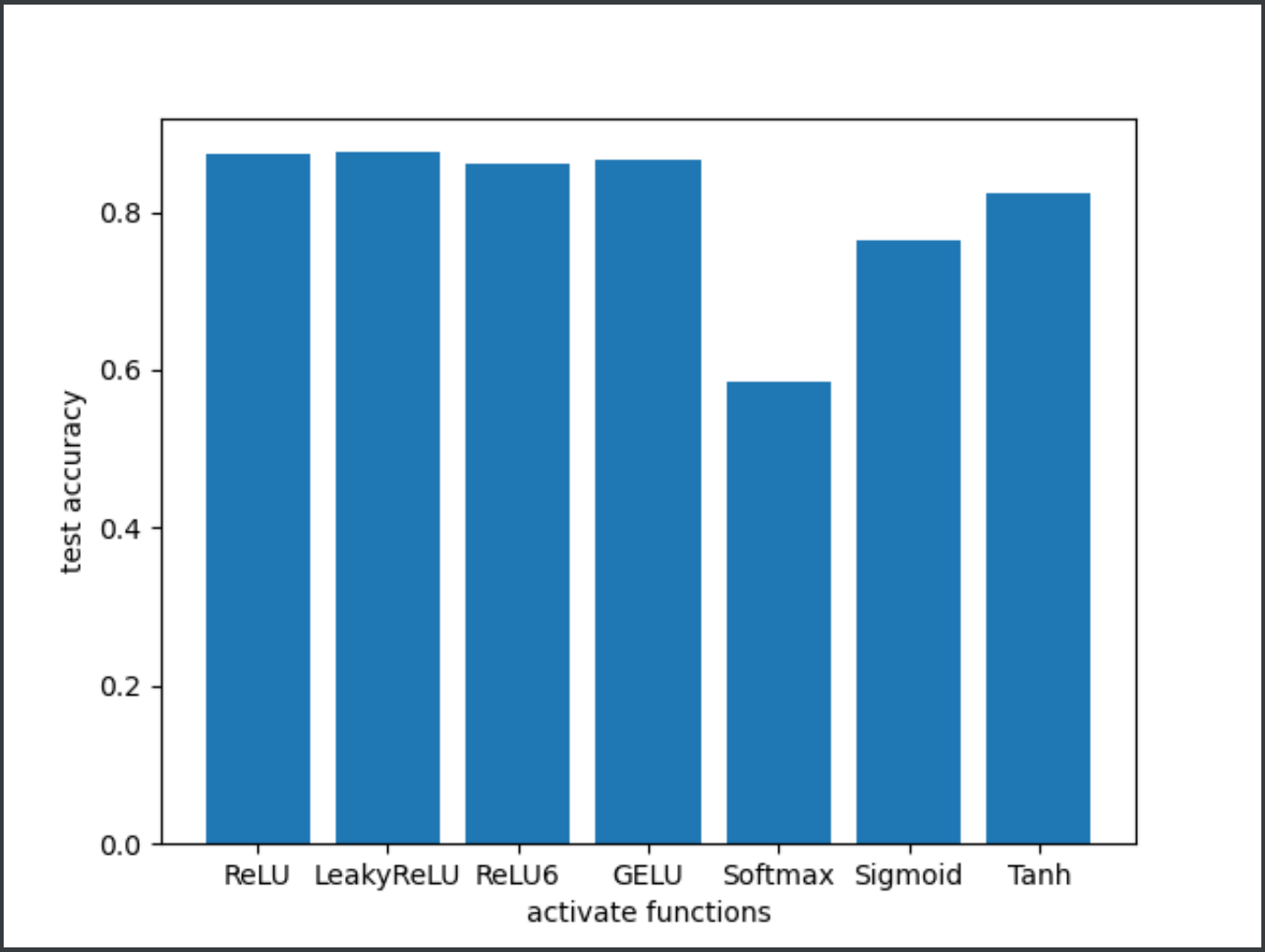



```
MyNetwork(
  (layers): Sequential(
    (0): Conv2d(1, 10, kernel_size=(5, 5), stride=(1, 1))
    (1): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=0, dilation=1, ceil_mode=False)
    (2): ReLU()
    (3): Conv2d(10, 20, kernel_size=(5, 5), stride=(1, 1))
    (4): Dropout(p=0.5, inplace=False)
    (5): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=0, dilation=1, ceil_mode=False)
    (6): ReLU()
    (7): Flatten(start_dim=1, end_dim=-1)
    (8): Linear(in_features=320, out_features=50, bias=True)
    (9): ReLU()
    (10): Linear(in_features=50, out_features=3, bias=True)
    (11): LogSoftmax(dim=1)
  )
)
```

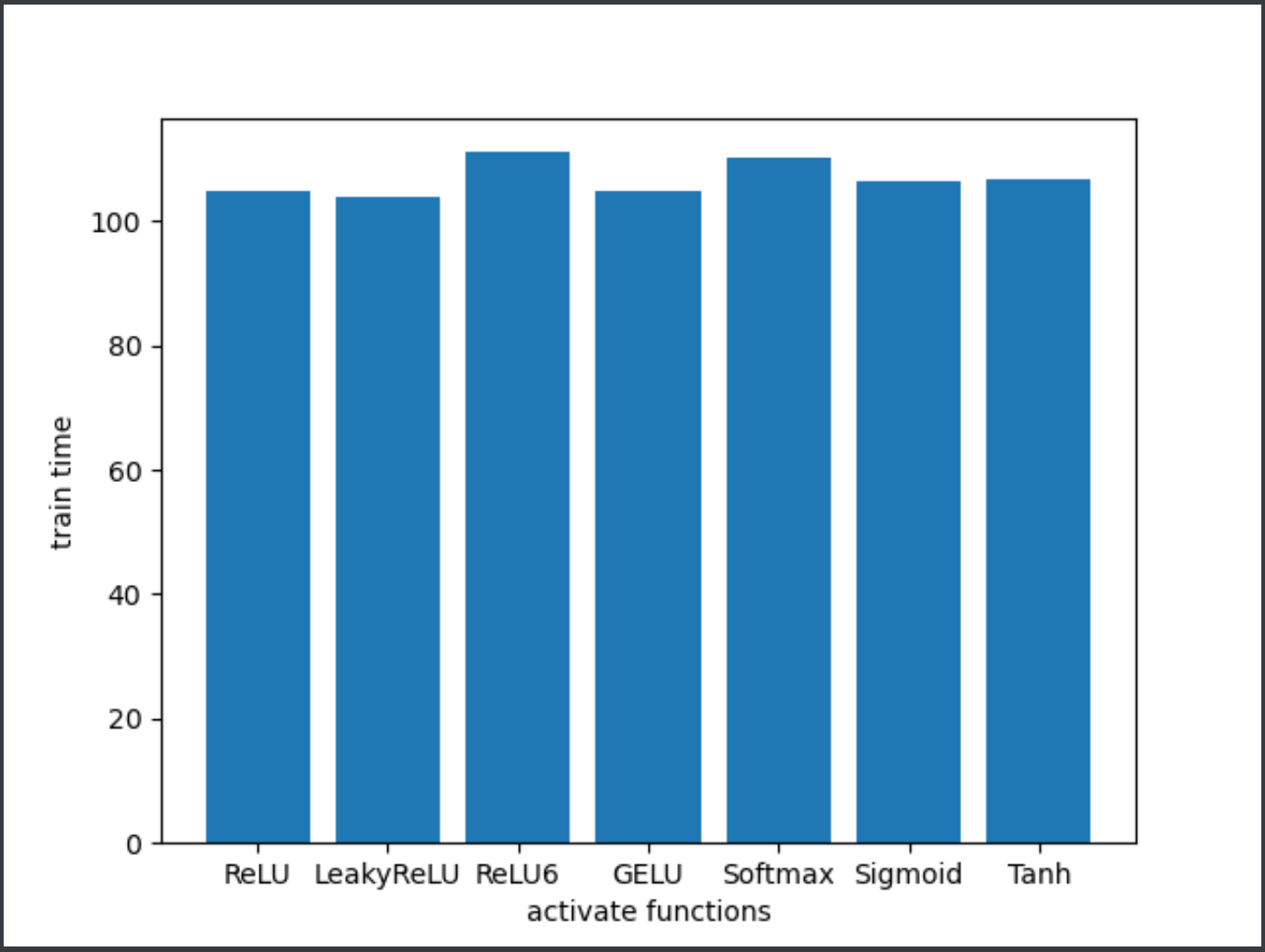
- task4

For the experimentation, we changed activation function, batch size, droupout rate, epoch number of the model.

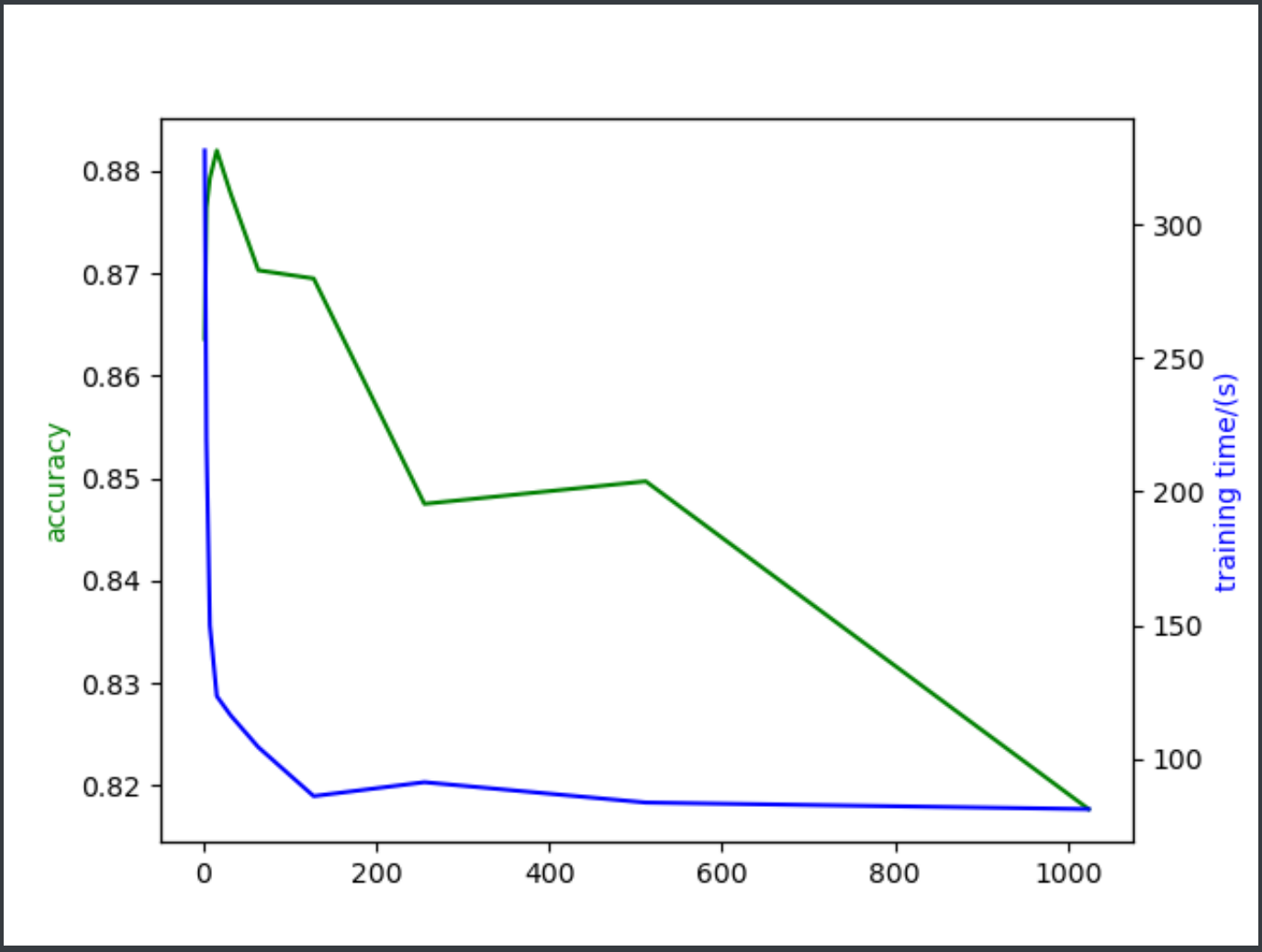
For activation functions test accuracy, RELU, GELU are good mainstream ones, so we predict they have high test accuracy. Softmax, Sigmoid, Tanh are not so common, so as predicted they are not as good as activation functions above.



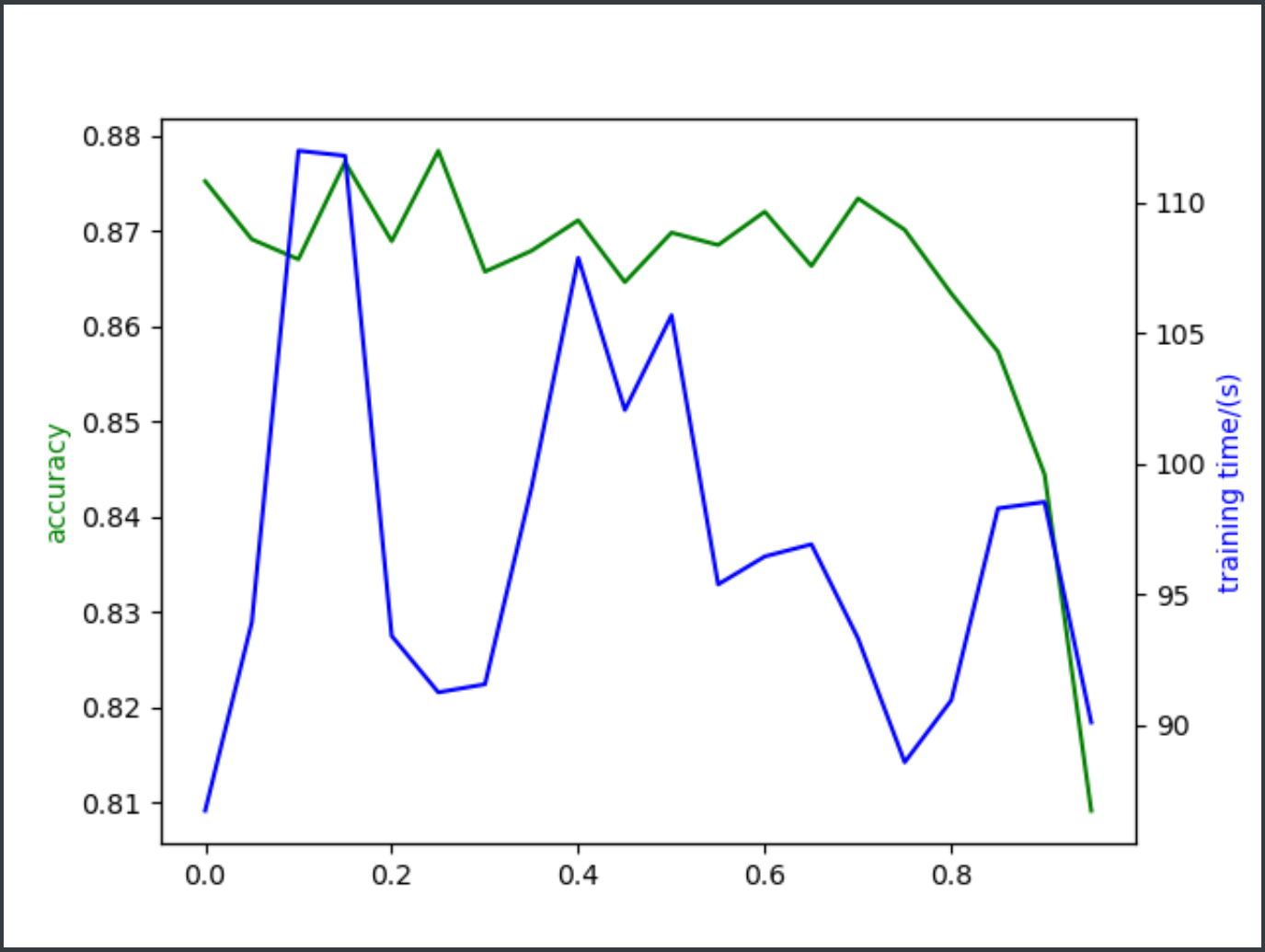
For activation functions train time, we expect they have similiar times.



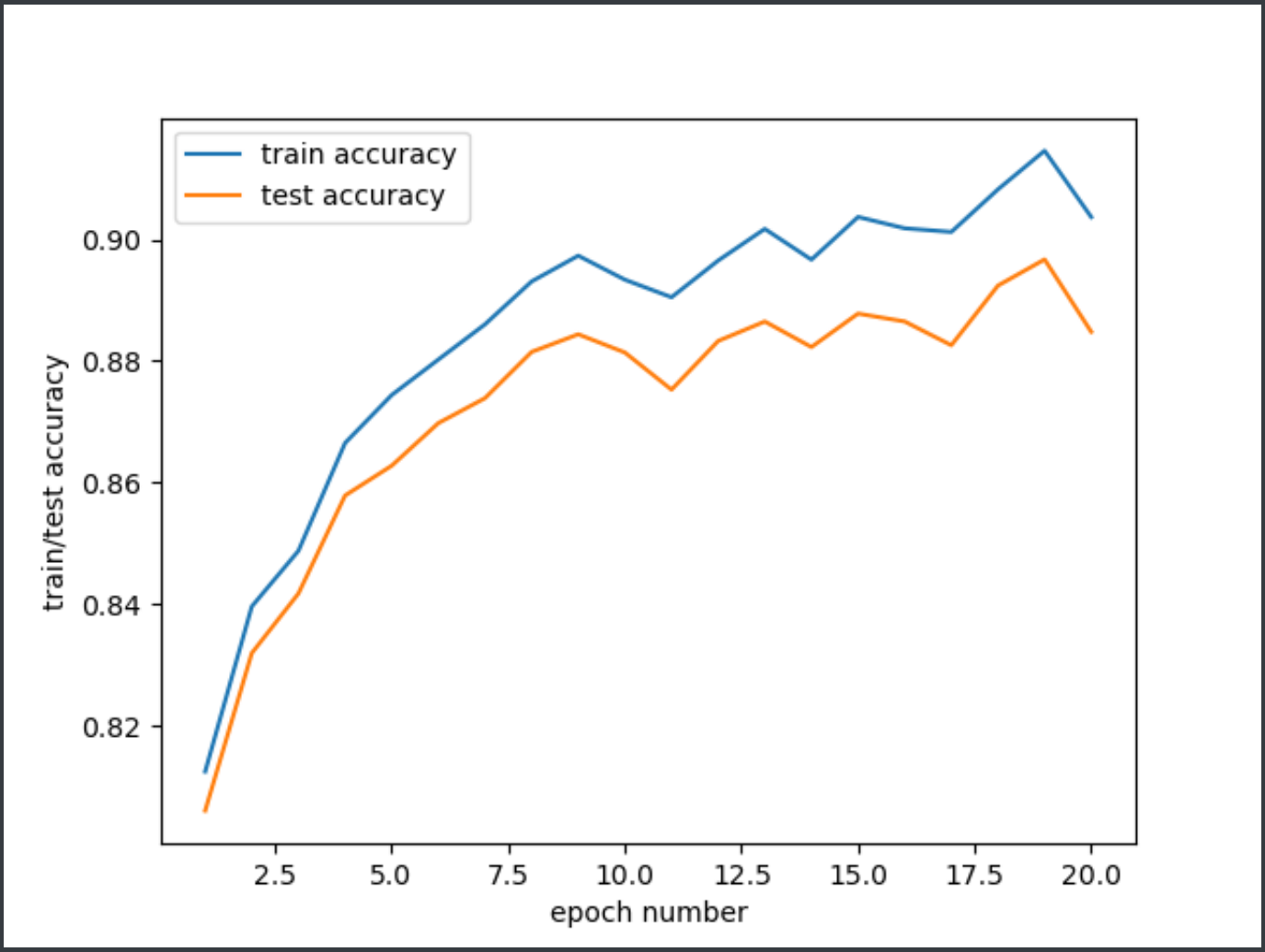
For the batchsize, it makes sense that the training time reduces as we increase the batch that we feed to the model. However, the accuracy drops with increased batchsize. That's the opposite of our prediction results.



The dropout rate accuracy should reach the local maxima in the middle, and drop significantly in the end. Unfortunately, we don't see the local maxima. The training times on the other hand fluctuate up and down due to the fluctuation of computer calculation power.

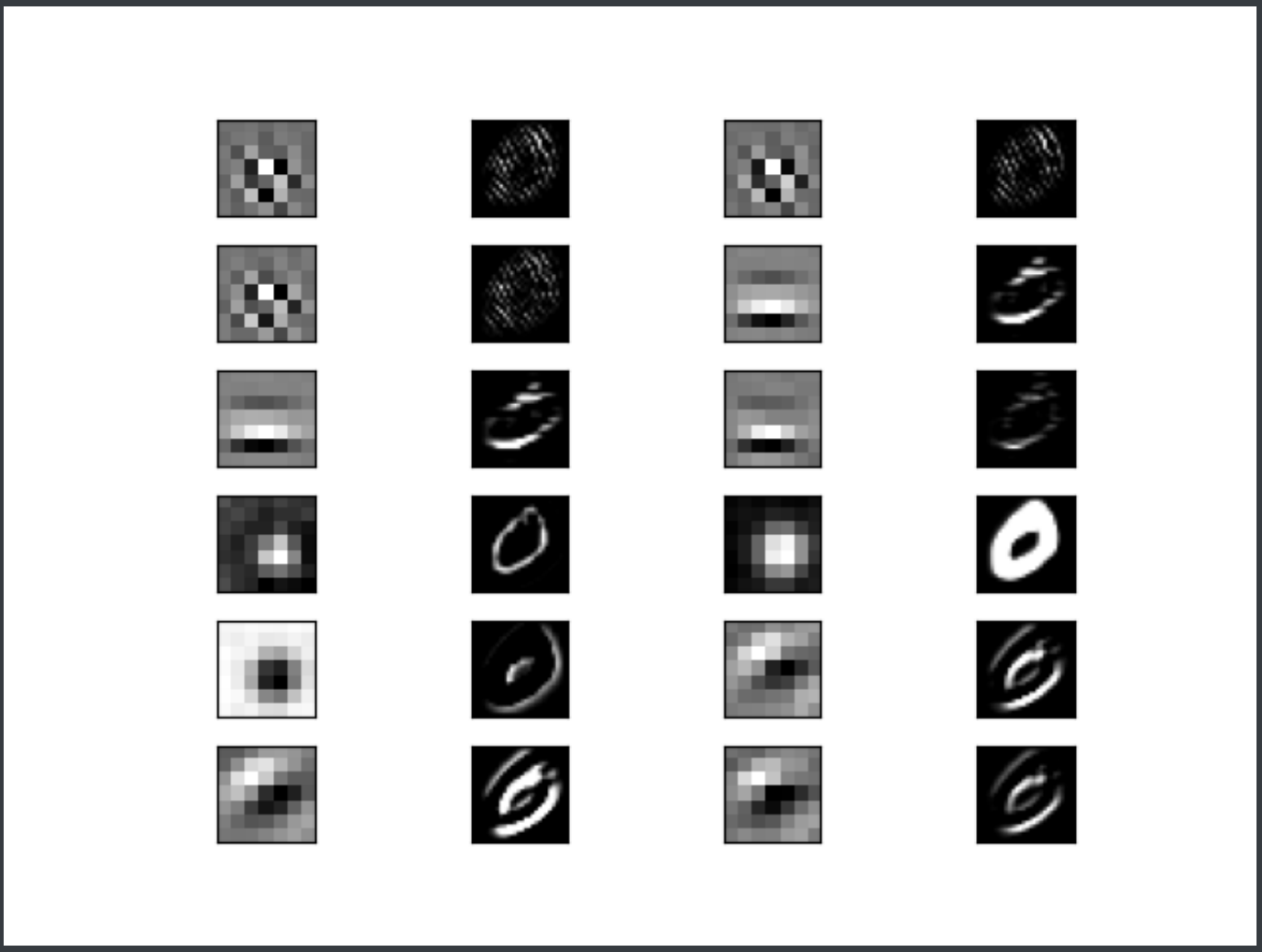


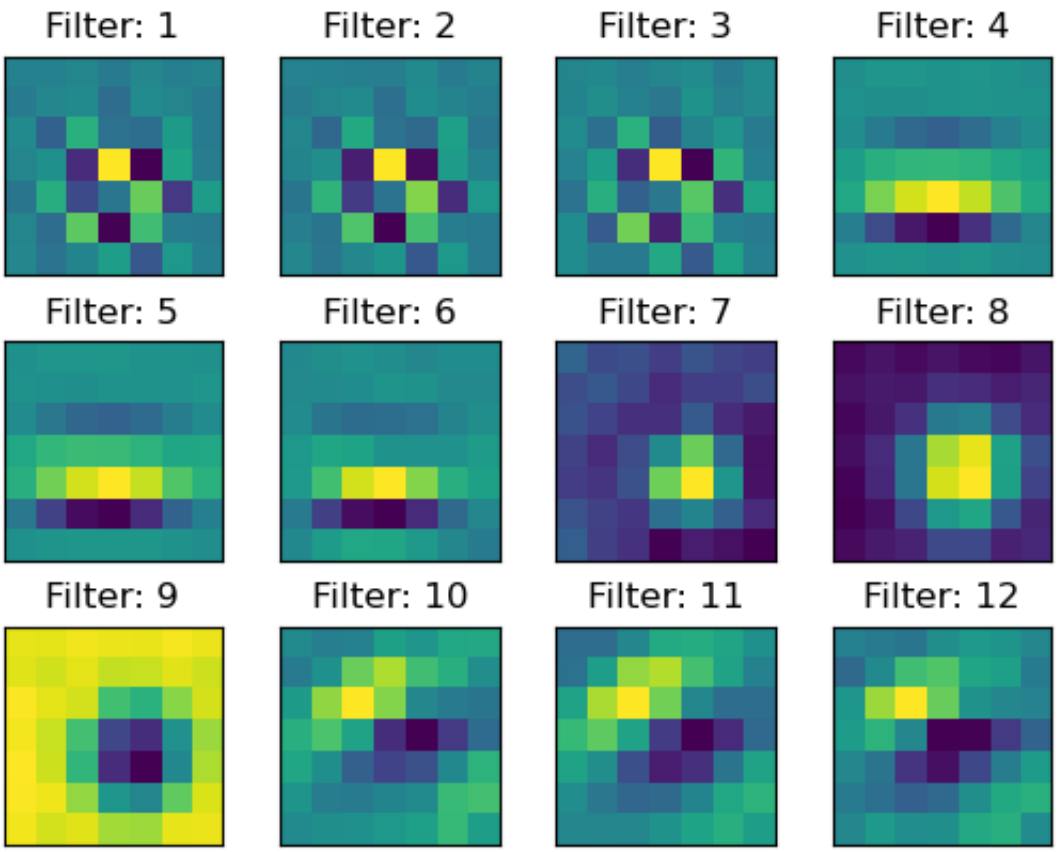
For epoch number, both train and test accuracy increase as more epoch number we use. In the end, the differences between the train and test accuracy are improved due to model overfitting.



A description and example images of any extensions.

This is our extension. We downloaded Resnet model from pytorch, and extracted first convolution layer' 12 filters. Finally, we applied the filter onto the number 0 and displayed in the graph.





A short reflection of what you learned.

- We learned the method to build a functional neural network. Call pytorch's API, create our own data.

Acknowledgement of any materials or people you consulted for the assignment.

- We mainly consulted pytorch website (https://pytorch.org/)