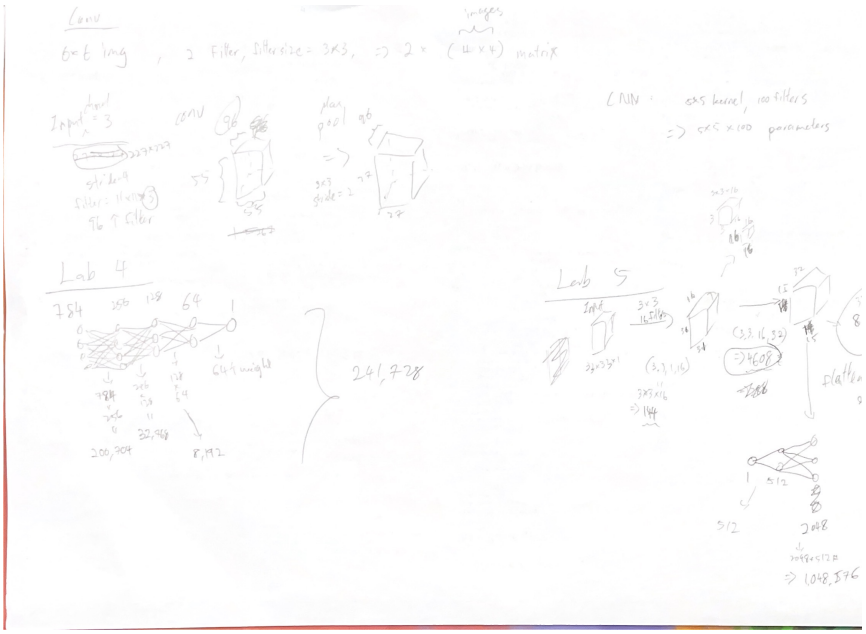
X1136010 黃偉祥

1. Explain why ReLU is typically preferred over Sigmoid as the activation function in the convolutional block? (1%)
    I. Because Sigmoid will lead to Vanishing Gradient problem, while ReLU is simpler and have gradient at any point (0 or 1).
2. Describe how you design the CNN architecture and your findings in choosing parameters such as filter_size and pool_size for each layer? (2%)
    I. I choose small filter size and pool size because I notice that our input data is only 32 * 32 dimensions, so I think big size may lead to ignoring some patterns
    II. Since I'm using small filter size and pool size, I will give only 1 padding and stride = 1 to filter and stride = 2 to pooling. Just to separate the patterns clearer in pooling layer.
    III. I'm using 2 conv and pool and 2 nn layers, 2 conv and pool layers just to make sure getting more patterns. And the 2 nn layers just to pass to sigmoid.
3. Calculate and compare the number of learnable parameters between the CNN model and the NN model you designed for binary classification in Lab4. For simplicity, omit the bias parameters and calculate only the weights. (2%)
    I. My Lab 4 architecture
        i. 784 -> 256 -> 128 -> 64 -> 1
        ii. Parameters = 784*256 + 256*128 + 128*64 + 64*1
            1. 200704 + 32768 + 8192 + 64
            2. Total = 241728
    II. My Lab 5 architecture

```
model = Model()
model.add(Conv(filter_size=3,input_channel=1,output_channel=16,pad=1,stride=1))
model.add(Activation("relu",None))
model.add(MaxPool(pool_size=2,stride=2))
model.add(Conv(filter_size=3,input_channel=16,output_channel=32,pad=1,stride=1))
model.add(Activation("relu",None))
model.add(MaxPool(pool_size=2,stride=2))
model.add(Flatten())
model.add(Dense(2048,512))
model.add(Activation("relu",None))
model.add(Dense(512,1))
model.add(Activation("sigmoid",None))
```

i. The first Conv have 3*3*16 parameters to update
ii. The second Conv have 3*3*16*32 parameters to update
iii. The first Dense have 2048*512 parameters to update
iv. The second Dense have 512*1 parameters to update
v. In total I have 144 + 4608 + 1048576 + 512 parameters
    1. Total = 1053840



Comparison

- My CNN model having 1053840 while NN model having 241728 parameters to update.
- But in CNN model, 1048576 out of 1053840 is coming from the Dense layer, so it could be only few thousand parameters, the only thing is that I get higher score in Kaggle with this large model, so I just keep it.
- In conclusion, Dense layer having full connected so it will generate more parameters to update but CNN can have much more fewer parameters to update.