Introduction of Machine Learning - lab4 report

111062332 朱誼學

1. Explain why ReLU is typically preferred over Sigmoid as the activation function in the convolutional block? (1%)

A: The Sigmoid function squashes input values into a range of [0, 1]. This can cause an issue if the input values are very large or very small. Since the gradient of sigmoid function will be very close to zero, which may lead to vanishing gradient. This problem will be more significant in deep neural networks. However, ReLU can avoid vanishing gradient. Moreover, ReLU has lower computational complexity, this makes the model learn more efficiently.

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%)

```
# My model design

model = Model()
model.add(Conv(filter_size=3, input_channel=1, output_channel=8, pad=1, stride=1))
model.add(Activation("relu", "cross_entropy"))
model.add(MaxPool(pool_size=2, stride=2))

model.add(Conv(filter_size=3, input_channel=8, output_channel=16, pad=1,
stride=1))
model.add(Activation("relu", "cross_entropy"))
model.add(MaxPool(pool_size=2, stride=2))

model.add(Flatten())
model.add(Dense(1024, 512))
model.add(Activation("relu", "cross_entropy"))
model.add(Dropout(rate=0.5))
model.add(Dense(512, 1))
model.add(Activation("sigmoid", "cross_entropy"))
```

A: I used 2 convolution layers and 2 fully-connected layers in my model.
For the convolution part,

I used filter_size=3, which is very common in modern CNN structure. Using a relatively small filter_size help us to reduce the computation cost.

I set the input and output channel of each conv layers to (1, 8) and (8, 16) respectively. Expanding to a larger number of channel helps model to learn more complex features.

padding is set to 1 in order to keep the input and output dimension the same.

stride = 1 indicates the model will capture the local features without skipping any part of the image.

For the fully-connected layers, these layers are used to do the classification task based on the output of convolution part.

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%)

A:
Learnable parameters in CNN model:

```
First conv: filter_size = 3, input_channel = 1, output_channel = 8
=> (3 * 3) * 1 * 8 = 72

Second conv: filter_size = 3, input_channel = 8, output_channel = 16
=> (3 * 3) * 8 * 16 = 1152

First fully-connected: input_size = 1024, output_size = 512
=> 1024 * 512 = 524,288

Second fully-connected; input_size = 512, output_size = 1
=> 512 * 1 = 512
```

Learnable parameters in NN model for binary classification in lab4:

```
layers_dims = [784, 100, 200, 200, 100, 1]
=> 784 * 100 + 100 * 200 + 200 * 200 + 200 * 100 + 100 * 1 = 158,500
```

Comparison:
There are 158.5k parameters in NN, this requires large compute cost.
As for CNN, the convolution part has only 72 + 1152 = 1222 parameters. If we simplified the fully-connected layers to only one (input_size=1024, output_size=1) layer, CNN can still do accurate classification.
The reason of why the total number of parameters in CNN is more than NN is that I want to make the CNN model more complex to increase the accuracy.
Theoretically, CNN can reduce the number of parameters with respect to NN.