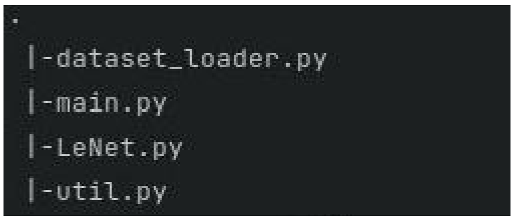
Assignment 4

You are required to implement a CNN with pytorch library. The code templates are included in the folder Assignment4_template. The files are organized as shown in Figure 1.



```
.
|-dataset_loader.py
|-main.py
|-LeNet.py
|-util.py
```

Figure 1. The tree structure of folder Assignment1_template

image_loder.py: Define the function to load the MNIST dataset
LeNet.py: Define the LeNet with layers from pytorch lib
util.py: Define some functions you could use. You can build your own functions if that is more convenient for you.
main.py: main file to run to print out the required results.

In this homework, you need to
1. build the modified LeNet with a structure shown in Table 1 to classify the MNIST dataset. Please use cross-entropy as the loss function and SGD as the optimizer.
https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html
https://pytorch.org/docs/stable/generated/torch.optim.SGD.html

Table 1. The LeNet CNN structure

layer	size	stride	Activation
Conv	5x5x1x6	1	ReLU
Avgpool	2x2	2	
Conv	5x5x6x16	1	ReLU
Avgpool	2x2	2	
Conv	5x5x6x120	1	ReLU
FC	120x84		ReLU
FC	84x10		

2. Train and test the LeNet with floating-point operations
3. Apply the dynamic quantization to the LeNet you have trained, print out your results as required
4. Apply the static quantization to the LeNet you have trained, print out your results as required.

Please refer to the tutorial for quantization in pytorch
https://pytorch.org/blog/introduction-to-quantization-on-pytorch/

Submission:

- You are required to submit a zip file of the finished code on Canvas and a report with following questions answered (with screen shot).
  1. What is the accuracy/size/execution_time_of_test of your network in floating-point operation?
  2. What is the accuracy/size/execution_time_of_test of your network after dynamic quantization? Are the results changed? Why?
  3. What is the accuracy/size/execution_time_of_test of your network after static quantization? Are the results changed compared to previous two cases? Why?