

# Homework 4

---

Jin Xu, Rutgers University

## Implementation

---

My model:

```
def __init__(self):
    super(LeNet, self).__init__()
    self.conv1 = nn.Sequential(
        nn.Conv2d(3, 6, 5),
        nn.ReLU(),
        nn.MaxPool2d(kernel_size=(2, 2), stride=2)
    )
    self.conv2 = nn.Sequential(
        nn.Conv2d(6, 16, 5),
        # nn.BatchNorm2d(16),
        nn.ReLU(),
        nn.MaxPool2d(kernel_size=(2, 2), stride=2)
    )
    self.fc1 = nn.Sequential(
        nn.Linear(16*5*5, 120),
        nn.ReLU(),
        # nn.Dropout(0.5)
    )
    self.fc2 = nn.Sequential(
        nn.Linear(120, 84),
        nn.ReLU()
    )
    self.fc3 = nn.Linear(84, 10)
```

My forward:

```
def forward(self, x):
    out = self.conv1(x)
    out = self.conv2(out)
    out = out.view(out.size(0), -1) # flatten conv layer to fc layer
    out = self.fc1(out)
    out = self.fc2(out)
    out = self.fc3(out)
    return out
```

## Result

---

I run the model training on a GPU(RTX 2080) using the default setting in the code.

Without BN/dropout:

```
Test set: Average loss: -3.6597, Accuracy: 6016/10000 (60%)

Traning and Testing total excution time is: 561.9021873474121 seconds
```

With a batch normalization layer between **conv2** and **relu2**:

```
Test set: Average loss: -4.1582, Accuracy: 6993/10000 (70%)  
Traning and Testing total excution time is: 580.124659538269 seconds
```

With a dropout layer after **fc1**(&relu3):

```
Test set: Average loss: -2.7200, Accuracy: 5754/10000 (58%)  
Traning and Testing total excution time is: 564.5865948200226 seconds
```