IIAI30017

Artificial Intelligence

HW3: Neural Network

Instructor: YuanFu Yang

yfyangd@nycu.edu.tw

國立陽明交通大學
NATIONAL YANG MING CHIAO TUNG UNIVERSITY

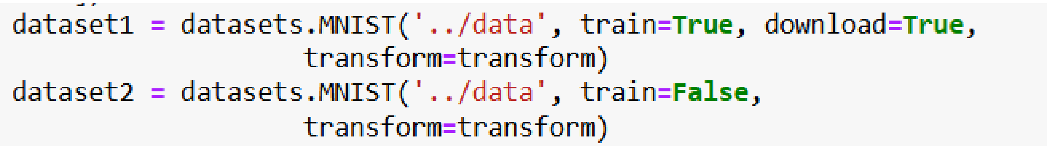
Neural Network

- Homework due: 12/4
- Late submissions will incur a penalty of one point for each day overdue.
- The assignment allows a maximum extension of 3 days (it will not be accepted if submitted later than 3 days).
- Submit files: code and report (4 questions/1 Bonus), and submit them in both **.py** and **PDF** file formats respectively.
- This assignment can be carried out using Colab or completed on your PC.

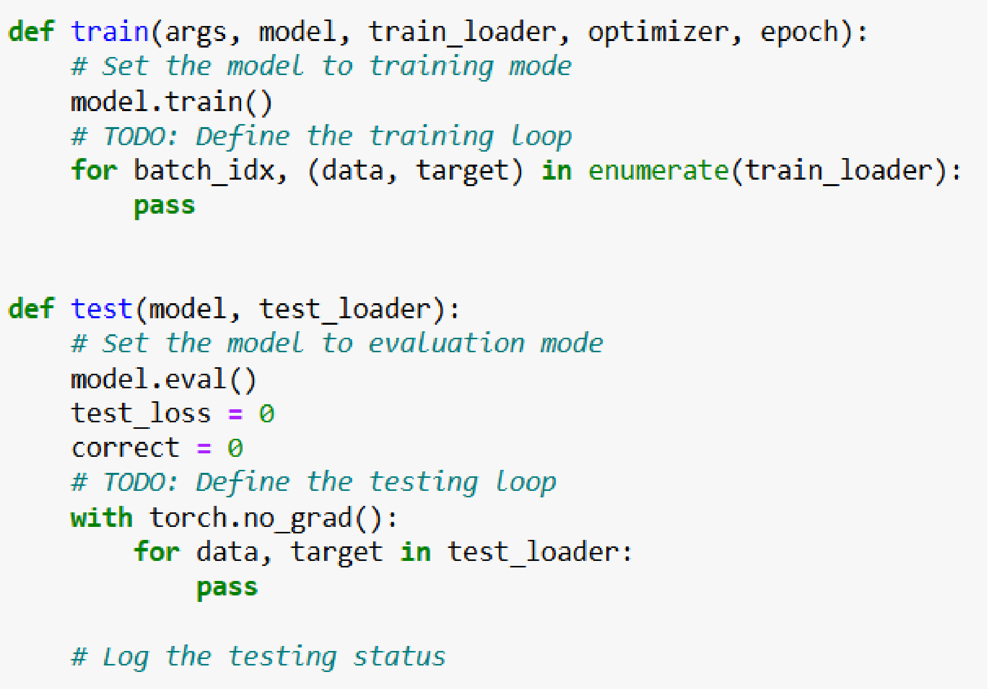
Neural Network

- This assignment focuses on performing image classification on the MNIST dataset. You need to complete all the TODOs in **main.py**, follow the instructions provided in the TODOs for coding, and carry out model training and testing.

main.py



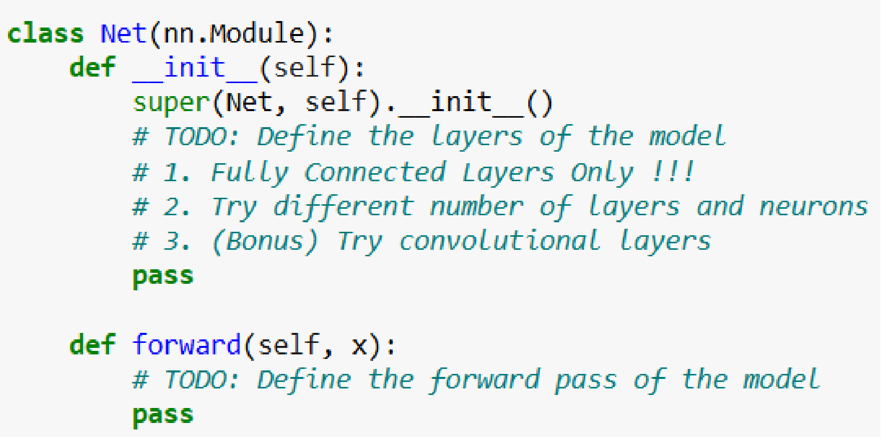
```python
dataset1 = datasets.MNIST('../data', train=True, download=True,
                          transform=transform)
dataset2 = datasets.MNIST('../data', train=False,
                          transform=transform)
```



```python
def train(args, model, train_loader, optimizer, epoch):
    # Set the model to training mode
    model.train()
    # TODO: Define the training loop
    for batch_idx, (data, target) in enumerate(train_loader):
        pass


def test(model, test_loader):
    # Set the model to evaluation mode
    model.eval()
    test_loss = 0
    correct = 0
    # TODO: Define the testing loop
    with torch.no_grad():
        for data, target in test_loader:
            pass

    # Log the testing status
```

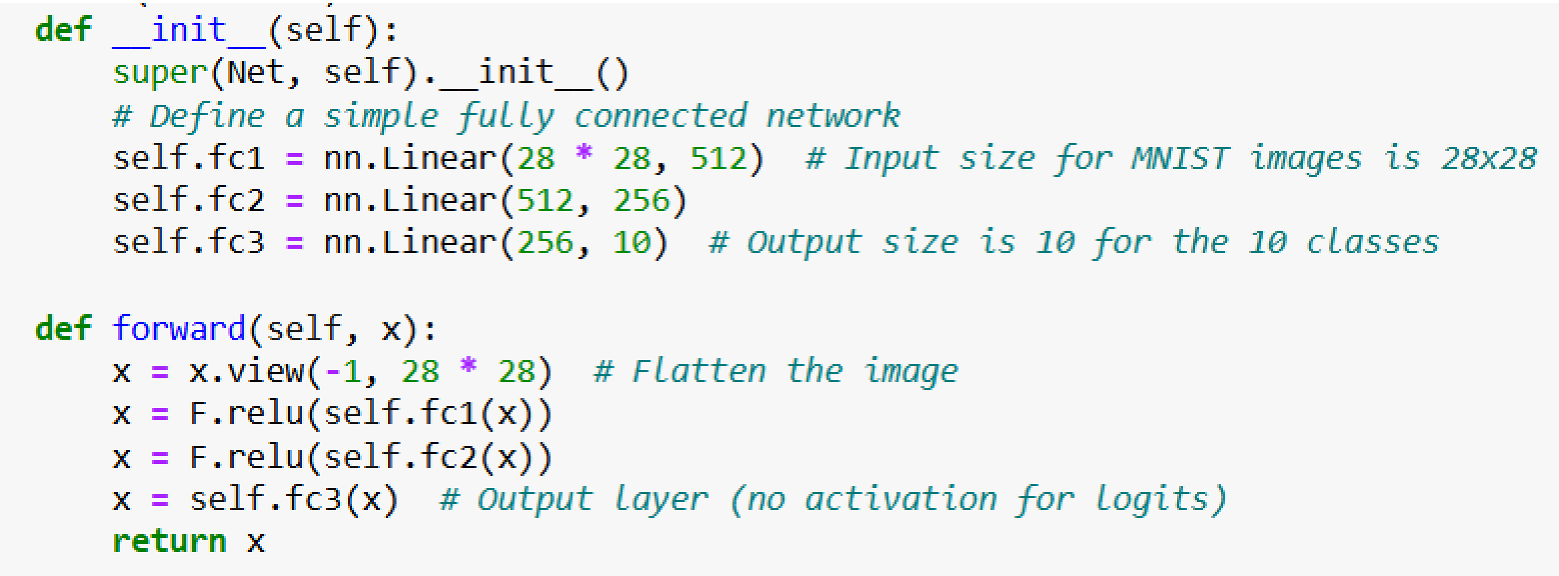


```python
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        # TODO: Define the layers of the model
        # 1. Fully Connected Layers Only !!!
        # 2. Try different number of layers and neurons
        # 3. (Bonus) Try convolutional layers
        pass

    def forward(self, x):
        # TODO: Define the forward pass of the model
        pass
```

Neural Network

- A simple NN architecture example

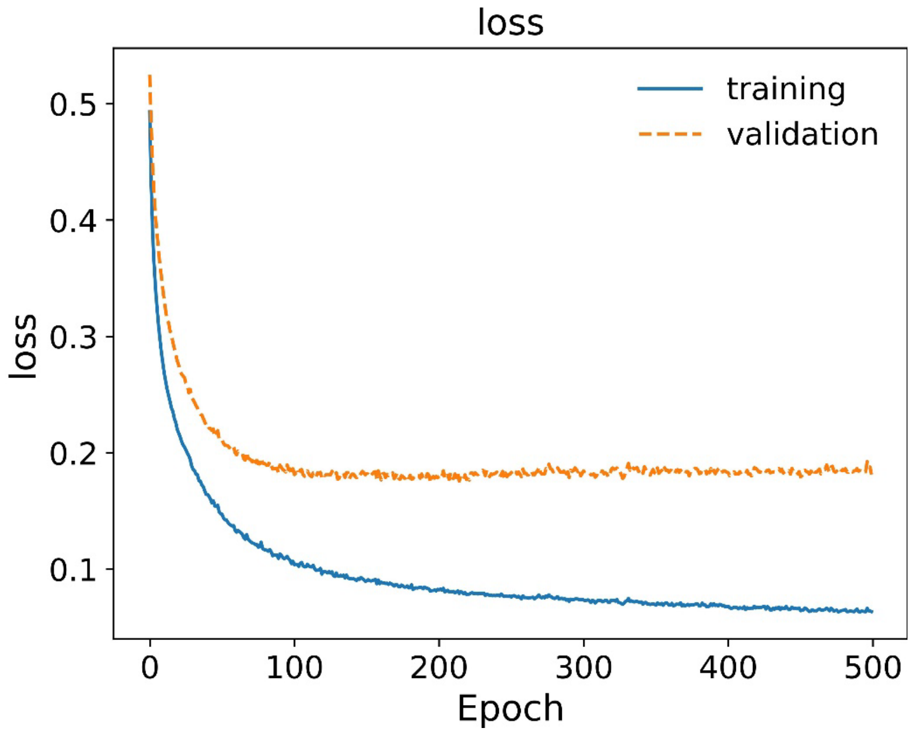


```
def __init__(self):
    super(Net, self).__init__()
    # Define a simple fully connected network
    self.fc1 = nn.Linear(28 * 28, 512)  # Input size for MNIST images is 28x28
    self.fc2 = nn.Linear(512, 256)
    self.fc3 = nn.Linear(256, 10)  # Output size is 10 for the 10 classes

def forward(self, x):
    x = x.view(-1, 28 * 28)  # Flatten the image
    x = F.relu(self.fc1(x))
    x = F.relu(self.fc2(x))
    x = self.fc3(x)  # Output layer (no activation for logits)
    return x
```

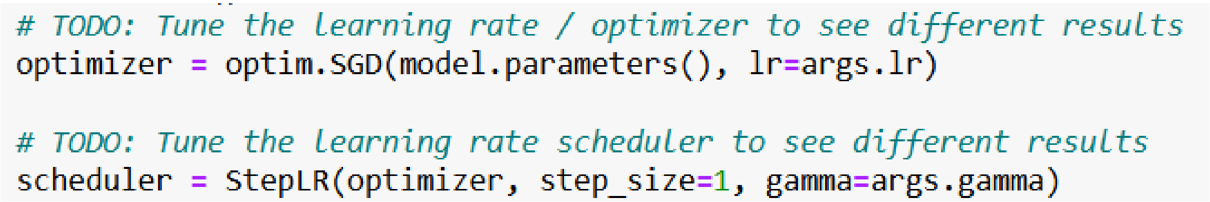
Neural Network

- HW3.1. Plot the Training & Validation Loss chart, with the X-axis representing Iteration or Epoch and the Y-axis representing Loss. The chart should include two lines: training loss and validation loss.



Neural Network

- HW3.2. Print the test accuracy. Additionally, explain your strategy for splitting the dataset into training, validation, and test sets.
- HW3.3. Discuss the Impact of Hyperparameter Tuning.-Explain how adjusting different hyperparameters (such as learning rate, batch size, and the number of epochs) affected the performance of your model. Provide examples of specific hyperparameter values you tried and the corresponding changes in training and test accuracy.
- HW3.4. Discuss Model Architecture Choices. Reflect on the architecture of your model and the rationale behind choosing the specific number of layers and neurons. What were the trade-offs you considered, and how did your design decisions impact the model's performance and generalization ability?



```
# TODO: Tune the learning rate / optimizer to see different results
optimizer = optim.SGD(model.parameters(), lr=args.lr)

# TODO: Tune the learning rate scheduler to see different results
scheduler = StepLR(optimizer, step_size=1, gamma=args.gamma)
```

Neural Network

- (Bonus) Use a CNN architecture for model training and validation, and compare it with the NN architecture. Discuss the differences between the two approaches.

IIAI30017

Artificial Intelligence

HW3: Neural Network

Q & A

Instructor: YuanFu Yang

yfyangd@nycu.edu.tw