a. Network 架構

1. FC(28*28, 60) + Leaky ReLU
2. FC(60, 50) + Leaky ReLU
3. FC(50, 40) + Leaky ReLU
4. FC(40, 10)
5. SoftmaxWithloss

b. Layer 細節

- FC：

```
def forward(self, input):
    self.input = input
    output = np.dot(input, self.weight) + self.bias
    return output

def backward(self, output_grad):
    input_grad = np.dot(output_grad, self.weight.T)
    self.weight_grad = np.dot(self.input.T, output_grad)
    self.bias_grad = np.sum(output_grad, axis=0, keepdims=True)
    return input_grad
```

- Leaky ReLU：

```
def forward(self, input):
    self.input = input
    output = np.maximum(0.01 * input, input)
    return output

def backward(self, output_grad):
    input_grad = output_grad * (self.input > 0) + 0.01 * output_grad * (self.input <= 0)
    return input_grad
```

- SoftmaxWithloss：

```
def forward(self, input, target):
    '''Softmax'''
    self.input = input
    self.target = target
    target = target.astype(int)
    target = np.argmax(target, axis=1)

    exp_input = np.exp(self.input - np.max(self.input, axis=1, keepdims=True))
    self.predict = exp_input / np.sum(exp_input, axis=1, keepdims=True)

    '''Average loss'''
    num_samples = input.shape[0]
    loss = -np.log(self.predict[range(num_samples), target])
    average_loss = np.mean(loss)

    return self.predict, average_loss

def backward(self):
    input_grad = self.predict - self.target
    return input_grad
```

c. 參數設定
   - Epoch = 100
   - Batch size = 50
   - Learning rate = 0.00001
   - Leaky ReLU 的斜率 (alpha) = 0.01

d. 預測結果
   - Loss：0.2 左右
   - Acc：90 左右

e. 遇到困難 & 解決方法
   1. **困難**：訓練到 Epoch = 10、11 時，loss 會出現 inf 或 nan，細看過程發現應該是梯度爆炸或消失，才會導致 loss 爆增。
      解決：無論 Learning rate 設置 0.01~0.00000001，皆無法解決此問題。後來我將先前加入的 data 正規化刪除後，便解決此問題。
   2. **困難**：loss 卡在 0.69 左右，acc 也只有 10，發現原因是 log(0.5) = -0.69，也就是預測出的機率皆趨近 0.5，同時有梯度消失的問題，因此 loss 不斷卡在 0.69 無法下降。
      解決：我發現 loss 的 backward 中，我沒有將 input_grad = self.predict - self.target，也就是梯度在更新時並沒有被改變，所以預測出的機率才會卡在 0.5。同時，我採用 Leaky ReLU，避免梯度消失，不過後來嘗試 ReLU 後，acc 也趨近 90，不會有明顯的梯度消失，因此我認為 ReLU 和 Leaky ReLU 在這個題目的效果似乎差不多。

f. 改進方向
   1. 我的 train & val Acc 皆 93 左右，不過 test Acc 卻只剩 89 左右，我認為可能有 overfitting 問題，我透過一些方式試圖改善，例如調整 Learning rate、Epoch、FC 層的數量，或加入 L1、L2 正規化，不過無法徹底解決，希望未來 Lab 可以找到方法解決這個問題。
   2. 我的 Learning rate 設置 0.01 時，loss 便會出現 nan，將它降低至 0.001 以下之後即可正常運作，不過我認為這樣還是存在一些問題，因為網路上許多人推薦的 Learning rate 經常是 0.1、0.01 等較大的數值，如此收斂速度會較快，也能避免卡在 local minimal 的區段。希望未來 Lab 也可以找到解決方法，將 model 的 Learning rate 設至較大的數值。