基于前馈神经网络和 MINIST 数据集的手写体数字识别

自动化钱 001 李艺涵 2206123627

一. 背景介绍

本次作业利用前馈神经网络，基于 MINIST 数据集进行训练，最终实现对于手写数字体较为准确的识别。

前馈神经网络是一种基于神经元和层次结构的人工神经网络，由输入层，隐藏层，输出层组成，每一层由若干神经元组成。它的每个神经元都是由激活函数和输入的权重组成，输入的权重表示了神经元对输入信号的加权影响。前馈神经网络可以通过正向传播算法将输入信号传递到输出层，从而实现对输入信号的分类、回归等任务。

MINIST 数据集是常用的手写数字数据集，它包含了 60000 个训练样本和 10000 个测试样本，每个样本都是一个 28x28 像素的灰度图像。 MINIST 数据集已经被广泛应用于手写数字识别的研究中，它的规模适中，便于训练和测试。

二. 工具及相关设置

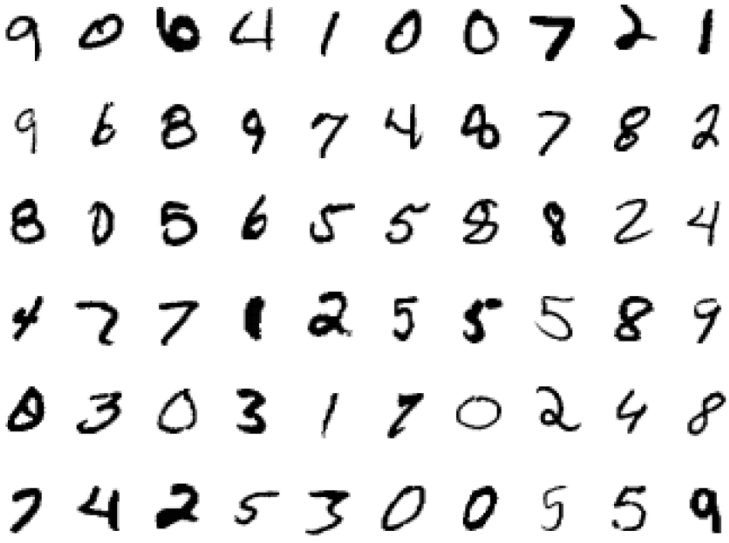
本次作业使用 Pytorch 框架完成，使用 anaconda 虚拟环境，所使用代码已上传至 GitHub，可在以下链接中查看：

https://github.com/YihanLi126/Handwriting-Rcognization

三. 训练过程

1. 数据收集及处理

利用 Pytorch 中的 datasets 工具获取 MINIST 数据集，提取样本图像中 28*28 的像素灰度值形成适用于 Pytorch 的张量(Torch Tensor)，并对张量进行标准化；分别将训练集和测试集数据加载为可读取的、标准的形式，设置每一个 batch 包含 64 个样本，以备训练和测试模型使用。为了加深对于数据集的直观感受，可将数据画出如下：

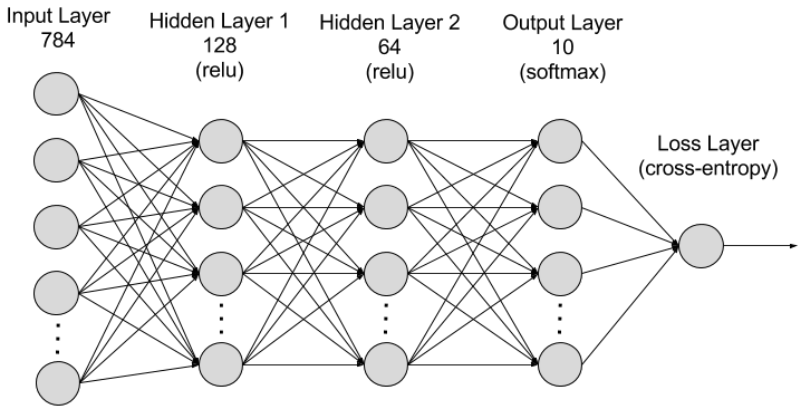


读取数据后，可以打印出数据集样本及标签的形状和大小：

```
(MyEnv) Yihans-MacBook-Air:Handwriting-Rcognization yihanli$ python img_show.py
Images Shape:  torch.Size([64, 1, 28, 28])
Labels Shape:  torch.Size([64])
```

可见每一个 batch 中有 64 张 28*28 大小的图片，相应地，每一个标签集里面有 64 个对应的标签。

2. 神经网络的搭建



（图片来自参考材料 1）

利用 Pytorch 搭建如上前馈神经网络模型：由于每张图片有 28*28=784 个像素点，将样本图片数据扁平化之后，将产生 784 个数据点，故将输入层设置为 784 个节点；隐层设置为两层，分别由有 128 和 64 个节点；将输入层和隐层的激励函数都设置为 ReLU 激励函数：

$$f(x) = x^+ = \max(0, x) = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases}$$

由于输出层为 0～9，将输出层设置为 10 个节点，并设置输出层激励为适用于分类问题的 LogSoftMax 激励：

$$\text{LogSoftmax}(x_i) = \log\left(\frac{\exp(x_i)}{\sum_j \exp(x_j)}\right)$$

最后再计算 NLL Loss, 使得 LogSoftMax 与 NLL Loss 共同作用，构成交叉熵损失，用于每次迭代后的权重调整。

3. 模型的训练

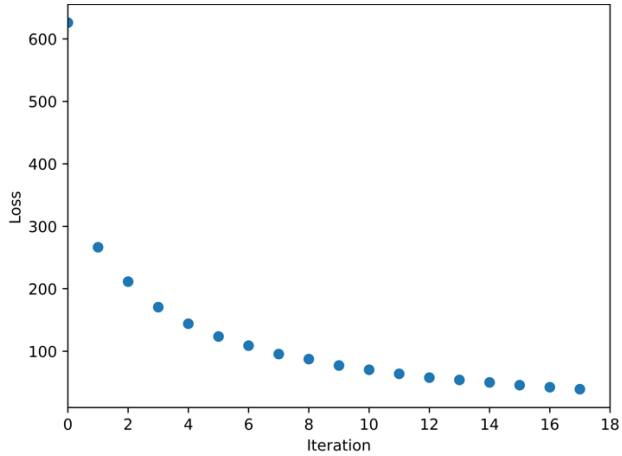
设置迭代次数为 18 次，记录历次迭代的 loss 值，用于后续的测试评估。训练完成后，将模型保存为.dat 文件，用于识别测试集图片。

4. 结果评估

读取并加载测试集数据，对样本图片逐个识别，记录下识别结果与对应的 label 中的真值并进行比对，对于识别正确和错误的个数进行记录，并计算出模型的正确率。

四．结果评估

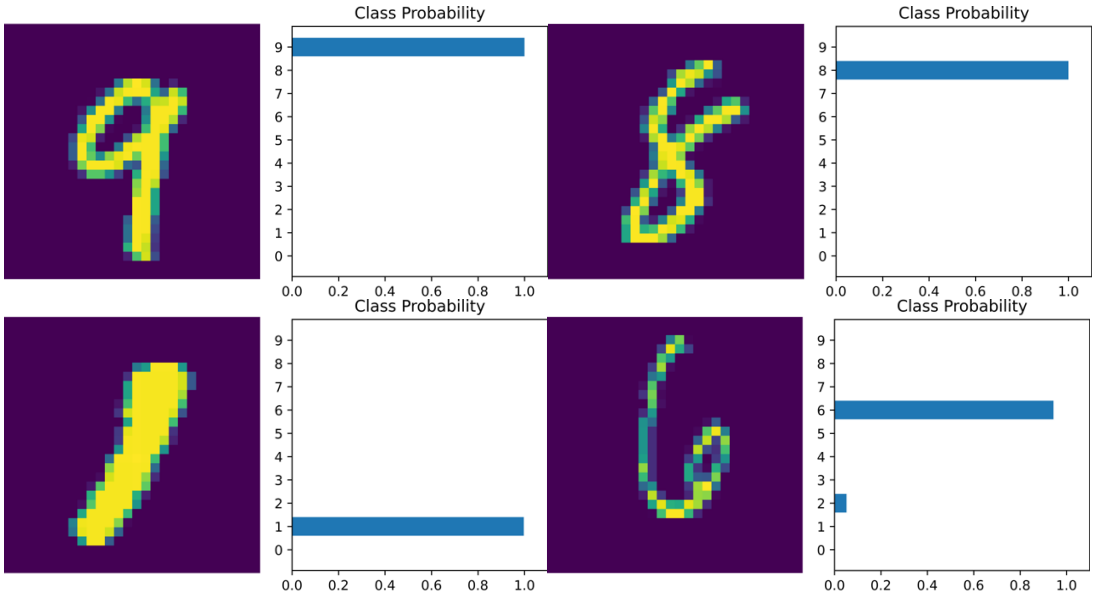
1. 训练过程 loss



作出迭代次数与模型 loss 的关系如上图。可以观察到，第一轮迭代后，loss 有显著的下降，在之后的多次迭代中，loss 的下降速率逐渐减缓，最终稳定在接近 0 的水平。

2. 识别结果可视化

下面给出几个模型识别结果样例：



3. 正确率评估

打印出模型正确率如下：

```
(MyEnv) Yihans-MacBook-Air:Handwriting-Rcognization yihanli$ python model_evaluation.py
Number Of Images Tested = 10000

Model Accuracy = 0.976
```

可知其识别准确率为 0.976，准确率较高。

五．结论

在本次作业中，我利用 Pytorch 框架搭建了一个较为基础的前馈神经网络模型，实现了准确率较高的手写体数字识别。在这个过程中，我对于数据集的获取、处理、神经网络的搭建、训练以及评估等过程有了较为基础的认知，为在相关方面的进一步学习创造了开端，有了很大的收获。

六．参考材料

1. http://yann.lecun.com/exdb/mnist/
2. https://towardsdatascience.com/handwritten-digit-mnist-pytorch-977b5338e627
3. https://medium.com/analytics-vidhya/training-mnist-handwritten-digit-data-using-pytorch-5513bf4614fb
4. https://en.wikipedia.org/wiki/Rectifier_(neural_networks)
5. https://pytorch.org/docs/stable/generated/torch.nn.LogSoftmax.html