

作业三报告

刘锦坤

2022013352

2024 年 6 月 24 日

1 Question1

```
命令提示符
Parameter conv1 weight 144
Parameter conv2 weight 4608
Parameter conv3 weight 18432
Parameter fc1 weight 800000
Parameter fc1 bias 500
Parameter fc2 weight 5000
Parameter fc2 bias 10
Total model parameter size: 829145
Train Epoch: 0 [0/938 (0%)] Loss: 2.3352 Accuracy: 9.38%
Train Epoch: 0 [50/938 (5%)] Loss: 0.2261 Accuracy: 95.31%
Train Epoch: 0 [100/938 (11%)] Loss: 0.1148 Accuracy: 96.88%
Train Epoch: 0 [150/938 (16%)] Loss: 0.0697 Accuracy: 98.44%
Train Epoch: 0 [200/938 (21%)] Loss: 0.4810 Accuracy: 95.31%
Train Epoch: 0 [250/938 (27%)] Loss: 0.1574 Accuracy: 93.75%
Train Epoch: 0 [300/938 (32%)] Loss: 0.0475 Accuracy: 98.44%
Train Epoch: 0 [350/938 (37%)] Loss: 0.0902 Accuracy: 96.88%
Train Epoch: 0 [400/938 (43%)] Loss: 0.0271 Accuracy: 100.00%
Train Epoch: 0 [450/938 (48%)] Loss: 0.0257 Accuracy: 98.44%
Train Epoch: 0 [500/938 (53%)] Loss: 0.0421 Accuracy: 98.44%
Train Epoch: 0 [550/938 (59%)] Loss: 0.1051 Accuracy: 95.31%
Train Epoch: 0 [600/938 (64%)] Loss: 0.0197 Accuracy: 100.00%
Train Epoch: 0 [650/938 (69%)] Loss: 0.0272 Accuracy: 98.44%
Train Epoch: 0 [700/938 (75%)] Loss: 0.1687 Accuracy: 96.88%
Train Epoch: 0 [750/938 (80%)] Loss: 0.0037 Accuracy: 100.00%
Train Epoch: 0 [800/938 (85%)] Loss: 0.0155 Accuracy: 100.00%
Train Epoch: 0 [850/938 (91%)] Loss: 0.0292 Accuracy: 98.44%
Train Epoch: 0 [900/938 (96%)] Loss: 0.0138 Accuracy: 100.00%
Your final test set accuracy is: 98.79%
*** PASS: check_digit_classification
```

图 1: BatchNorm Layer

```
命令提示符
Parameter bn3 num_batches_tracked 1
Parameter conv1 weight 144
Parameter conv2 weight 4608
Parameter conv3 weight 18432
Parameter fc1 weight 800000
Parameter fc1 bias 500
Parameter fc2 weight 5000
Parameter fc2 bias 10
Total model parameter size: 829145
Train Epoch: 0 [0/938 (0%)] Loss: 2.3027 Accuracy: 6.25%
Train Epoch: 0 [50/938 (5%)] Loss: 0.4269 Accuracy: 82.81%
Train Epoch: 0 [100/938 (11%)] Loss: 0.2753 Accuracy: 93.75%
Train Epoch: 0 [150/938 (16%)] Loss: 0.1216 Accuracy: 93.75%
Train Epoch: 0 [200/938 (21%)] Loss: 0.0655 Accuracy: 98.44%
Train Epoch: 0 [250/938 (27%)] Loss: 0.0383 Accuracy: 100.00%
Train Epoch: 0 [300/938 (32%)] Loss: 0.0142 Accuracy: 100.00%
Train Epoch: 0 [350/938 (37%)] Loss: 0.1462 Accuracy: 96.88%
Train Epoch: 0 [400/938 (43%)] Loss: 0.1329 Accuracy: 96.88%
Train Epoch: 0 [450/938 (48%)] Loss: 0.1655 Accuracy: 93.75%
Train Epoch: 0 [500/938 (53%)] Loss: 0.0733 Accuracy: 98.44%
Train Epoch: 0 [550/938 (59%)] Loss: 0.1889 Accuracy: 95.31%
Train Epoch: 0 [600/938 (64%)] Loss: 0.0519 Accuracy: 98.44%
Train Epoch: 0 [650/938 (69%)] Loss: 0.0272 Accuracy: 98.44%
Train Epoch: 0 [700/938 (75%)] Loss: 0.0231 Accuracy: 98.44%
Train Epoch: 0 [750/938 (80%)] Loss: 0.0804 Accuracy: 96.88%
Train Epoch: 0 [800/938 (85%)] Loss: 0.0436 Accuracy: 98.44%
Train Epoch: 0 [850/938 (91%)] Loss: 0.0863 Accuracy: 96.88%
Train Epoch: 0 [900/938 (96%)] Loss: 0.1337 Accuracy: 92.19%
Your final test set accuracy (97.78%) must be at least 98% to receive full points for this question
```

图 2: Without BatchNorm Layer

BatchNorm Layer 的重要作用是加速模型的收敛，BatchNorm Layer 将每个批量内的数据都进行了标准化处理，从而减少了由于前面的层输入参数变化导致的偏移量，进而使得模型收敛更迅速。

图 1 是加入 BatchNorm Layer 的模型，图 2 是没有加入 BatchNorm Layer 的模型，可以看到加入 BatchNorm Layer 的模型收敛速度更快，而且收敛到的结果更好。

2 Question2

图 3 展示了生成的对抗性样本，可以做如下分析：

根据 FGSM 的原理，对抗样本在原始图像上添加了扰动，具体表现在一些像素强度上的改变，可能某些像素强度变高或者变低，或者像最左边那些数字一样整体像素强度都变得很低。

这些对抗性样本能够欺骗 LeNet 的原理大致可以归于以下：

1. 通过反向的“梯度上升”，找到使得损失函数最大化的方向，从而生成对抗性样本，欺骗了模型。
2. 但是这也说明的 LeNet Model 本身对于微扰是非常敏感的，即使微小的变化也能通过网络层传播并放大，从而导致输出发生显著变化。
3. 但是最后的对抗性样例不能欺骗人眼，就说明这是 LeNet 泛化能力不足的体现，这些对抗样本事实上就利用了 LeNet 学习到的那些噪声特征进行欺骗。

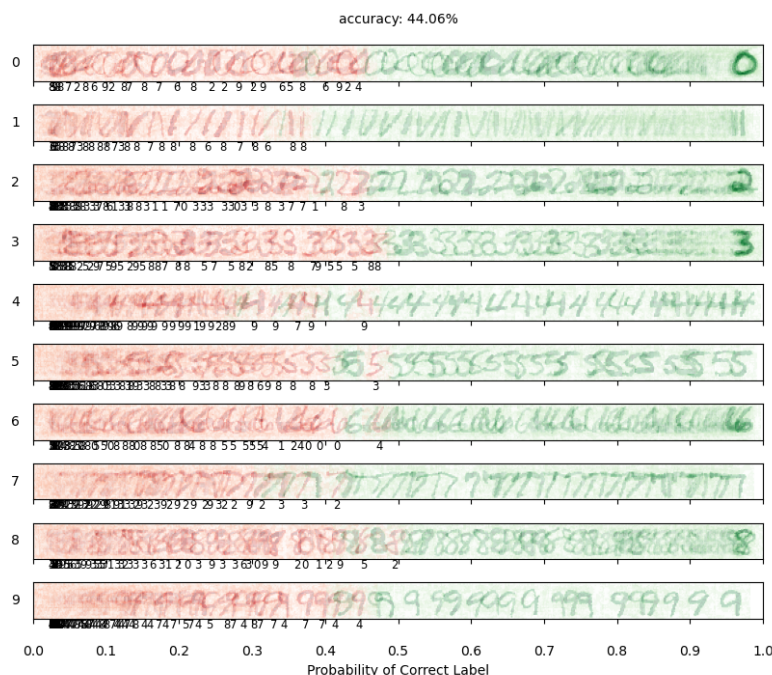


图 3: Adversarial Example

3 算法分析

Here is the modified code:

4 算法分析

这里对于本次作业中使用到的各个方法进行一个简要的分析：

4.1 Task1

这一部分主要使用 CNN Layer, BatchNorm Layer, ReLU, Pool Layer, FullConnected Layer 等方法进行手写数字集的识别，卷积层主要是对图像中的各个模态进行识别，批归一化层用以加速收敛和提高泛化能力，池化层将数据模糊化，也起到提高泛化能力的作用，最后利用全连接层作为感知机，对于前面提取到的特征进行分类。

4.2 Task2

这一部分就是用全连接层做一个函数图像的拟合，事实上存在神经网络的任意逼近定理，大致即为一个包含足够多隐含层神经元的多层前馈网络，能以任意精度逼近任意预定的连续函数，而这里的回归结果可以看成这个定理的一个体现。¹

4.3 Task3

这一部分内容在于对抗性样本的生成，这里使用了 FGSM 方法，即通过对原始图像添加一个扰动，使得模型的损失函数最大化，从而生成对抗性样本，这里的对抗性样本实际上就是利用了模型的一些特性，通过微扰来欺骗模型。

4.4 Task4

这一问使用 RNN 对于文本的语言进行了分类，RNN 是一种针对序列输入提出的神经网络，将状态和特征在序列的元素之间传递最终输出，这种方法的优势在于能够处理序列数据，比如文本，时间序列等，在面对序列数据时可以用较少的参数完成较好的提取特征。

¹参考 https://blog.csdn.net/qq_37983752/article/details/115055707