

# 人工神经网络

## 第一次作业

### MNIST Digits Classification with CNN

计 55 张卡尔 2015011025

2017 年 10 月 25 日

## 实验内容

本实验将会使用卷积神经网络实现 MNIST 手写数字识别分类，对不同的网络结构以及不同的参数进行讨论，与 MLP 进行对比，探索优化方法及策略。

## 网络结构设计

本实验主要使用的网络结构如下：

- 输入层 ( $N * 28 * 28$ )
- 卷基层 ( $N * 4 * 28 * 28$ ，卷积核大小  $3 * 3$ )
- Relu 层
- 池化层 ( $N * 4 * 14 * 14$ )
- 卷基层 ( $N * 4 * 14 * 14$ ，卷积核大小  $3 * 3$ )
- Relu 层
- 池化层 ( $N * 4 * 7 * 7$ )
- 全连接层 ( $N * 196 * 10$ )
- Softmax 交叉熵损失层

## 训练参数探究

### 损失函数函数

使用相同参数、相同结构的上述网络，分别使用 Euclidean 和 SoftmaxCrossEntropy 两种损失函数训练 100 个 epoch，对比结果如下。

策略	训练集 loss	测试集 loss	训练集 acc	测试集 acc	训练时间	参数个数
Euclidean	0.087	0.088	96.75%	96.78%	233 分钟	2158
SoftmaxCrossEntropy	0.046	0.043	98.56%	98.65%	233 分钟	2158%

Table 1: 损失函数对比表

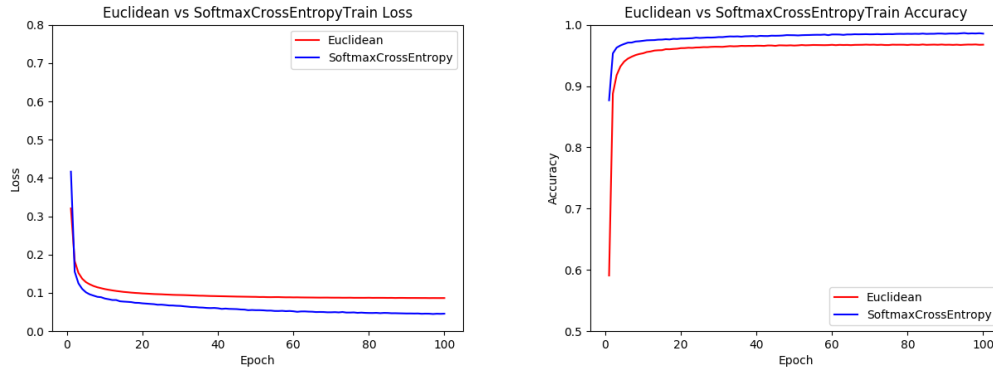


Figure 1: 损失函数对比图

通过对比，可以发现在使用相同的参数和网络结构对同一任务进行训练时，在训练相同数量的 epoch 的情况下，Relu 比 Sigmoid 收敛速度更快且最终精确度更高，损失值更小。分析其原因，我认为是 Sigmoid 函数本身的特点导致的。Sigmoid 函数在中间部分的梯度较高，在两侧梯度很低，导致训练到一定程度时，梯度越来越小，从而训练速度十分缓慢。而 Relu 函数在  $[0, \infty)$  上的梯度为常数，不会出现梯度消失的问题，因而收敛速度更快，且能够在更少数量的 epoch 上达到比较高水平的精准度。

在训练时间方面，Sigmoid 网络花费了 51'53"，Relu 网络花费了 57'26"，相差不大。

## Learning Rate

### 初始权重

策略	训练集 loss	测试集 loss	训练集 acc	测试集 acc	训练时间	参数个数
init_std=0.1	0.045	0.047	98.66%	98.52%	233 分钟	2158
init_std=1	0.046	0.043	98.56%	98.65%	233 分钟	2158

Table 2: 初始权重对比

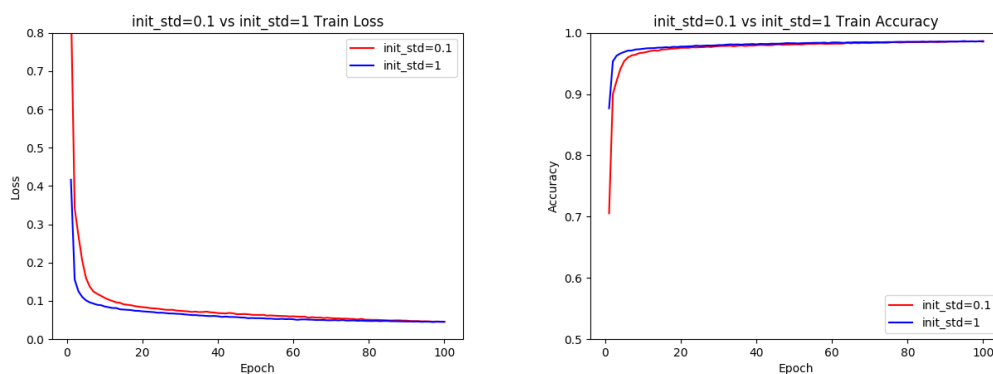


Figure 2: 初始权重对比图

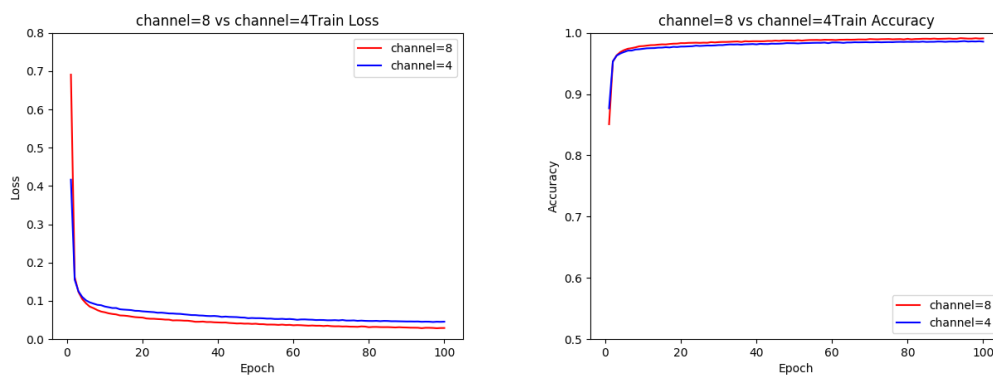


Figure 3: 通道数对比图

策略	训练集 loss	测试集 loss	训练集 acc	测试集 acc	训练时间	参数个数
8 通道	0.029	0.028	99.08%	99.10%	540 分钟	4584
4 通道	0.046	0.043	98.56%	98.65%	233 分钟	2158

Table 3: 通道数对比表

## 通道数

## 其他参数

由于本次实验时间有限，之前几个参数的调整过程占用时间过长，其他参数的优化问题在本次实验中没有深入研究。batch\_size 使用的是常规的 100，weight\_decay 由于在其他参数尚未优化之前效果不明显，故暂取为 0。

## 实验总结

本次实验占用了我很多的时间，收获也是十分丰富的。我走了许多弯路，例如在一开始在寻找正常的高准确度上浪费了许多时间，由于有多个参数未调，因此有时候会同时修改了多个参数，导致找到了一个相对较优的参数就开始进行优化算法的实现。但其实后来才发现把 learning rate 调大以后甚至比现在做过优化的算法效果还好。总而言之，通过这次实验我真实体会到了网络结构、激励函数、参数这些对网络效果的影响。只有真正理解了每一个数值的具体含义和其对网络的影响，才能进行合理的调参。