

# Minist 手写数字分类报告

---

ID: 21210980110 姓名: 刘家材

## 一. 模型与训练

---

### 1. 神经网络模型

在model.py中实现了一个二层神经网络: `class NN_Classfier`

激活函数: Relu

损失函数: 交叉熵. 见 `NN_Classfier. loss_fn()`

前向计算: 见 `NN_Classfier. forward()`

梯度计算: 见 `NN_Classfier. backward()`

梯度更新: 见 `NN_Classfier. step()`

保存模型: 见 `NN_Classfier. save_model()`

加载模型: 见 `NN_Classfier. load_model()`

### 2. 神经网络训练

train.py 中 函数Train() 负责以SGD为优化器训练, 原型为:

```
def Train(Network : NN_Classfier, train_epoch : int, batch_size : int, lr_rate : float, Lambda : float):
```

其中:

**Network** 是构建的2层神经网络NN\_Classfier的实例. `class NN_Classfier` 定义在 model.py

**train\_epoch** 是一共要训练的epoch个数

**batch\_size** 是SGD 训练时, 随机抽取的minibatch 的大小

**lr\_rate** 是模型参数学习率

**Lambda** 是正则化强度

训练过程中会保存最新训练的模型和最佳的模型,

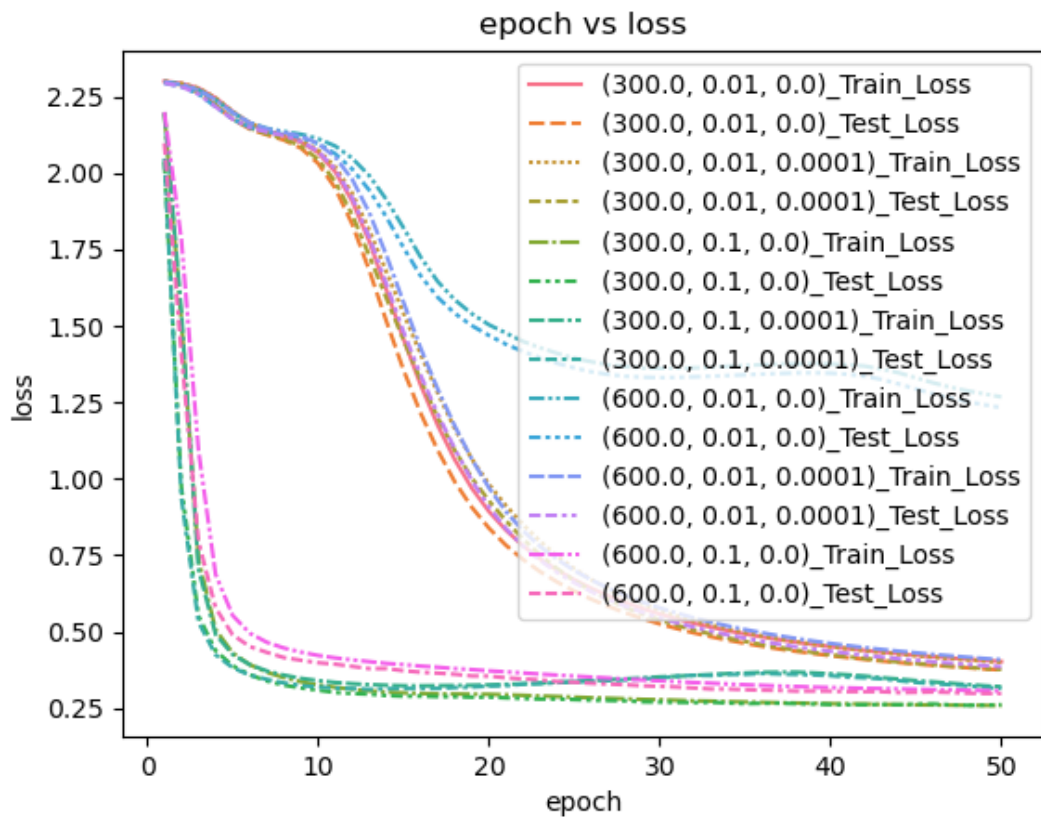
## 二. 参数查找

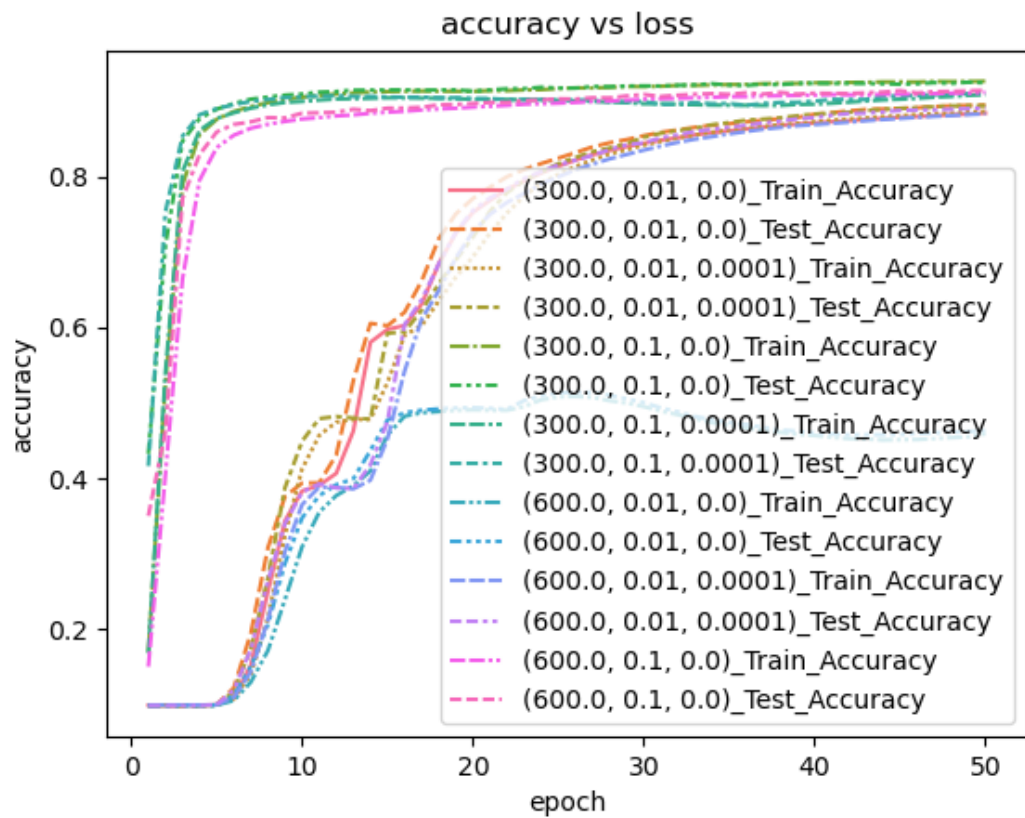
---

find\_parameter.py 中对比了以下7组参数的train 和 test performance. 我们统一使用50个epoch训练, batch\_size 设置为500

parameters
(hidden_size = 300, lr_rate = 1e-2, lambda = 0)
(hidden_size = 300, lr_rate = 1e-2, lambda = 1e-4)
(hidden_size = 300, lr_rate = 1e-1, lambda = 0)
(hidden_size = 300, lr_rate = 1e-1, lambda = 1e-4)
(hidden_size = 600, lr_rate = 1e-2, lambda = 0)
(hidden_size = 600, lr_rate = 1e-2, lambda = 1e-4)
(hidden_size = 600, lr_rate = 1e-1, lambda = 0)

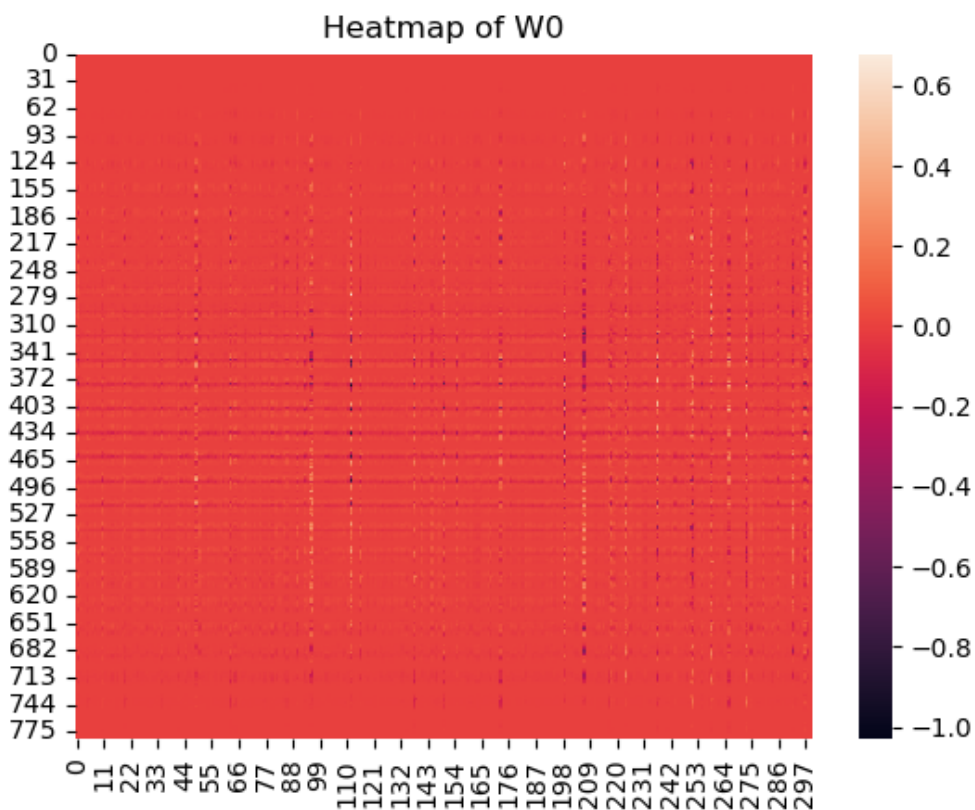
下图显示了各组参数在train 和 test 上的performance

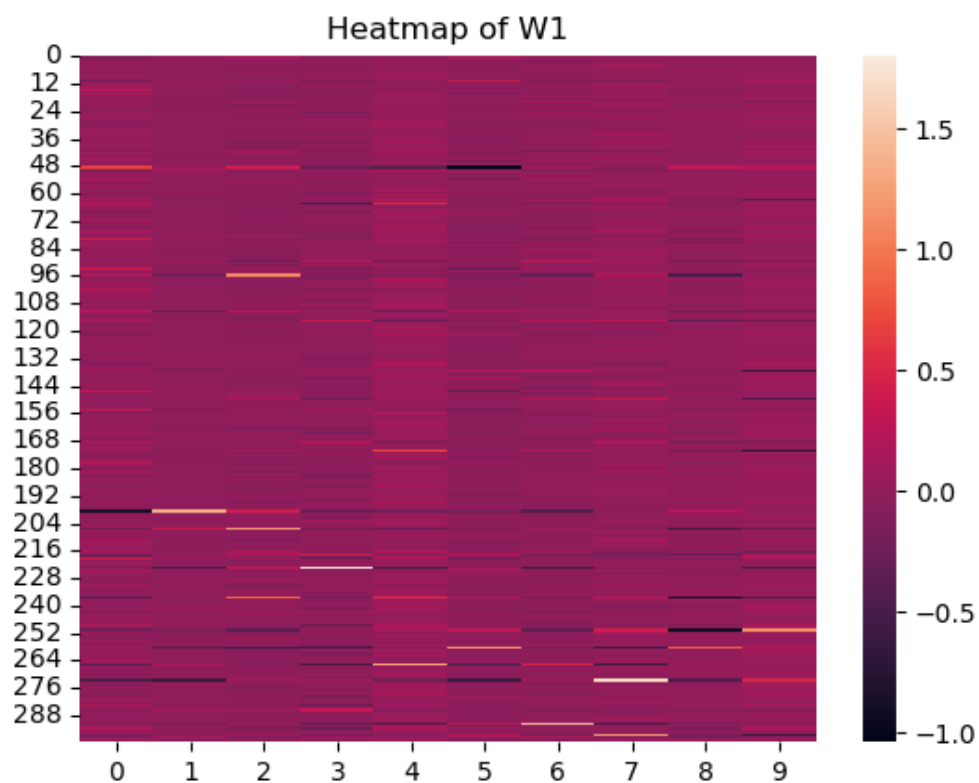




在这8组参数中，最后选定 (300,0.1,0) 为最佳模型

下面为(300,0.1,0)  $W_0$  与  $W_1$  的可视化





### 三. 测试

---

根据参数查找选择的模型，在训练集上有0.9272167准确率，在测试集上有 0.9251准确率

最佳参数模型文件在 models/Model\_300\_0.1000\_0.0000\_latest.m

### 四. 代码文件

---

已提交至: <https://github.com/skydownacai/NeuralNetworkAndDeepLearning-HW1.git>