

# 建立全连接神经网络

58121124 张博彦

2023 年 5 月 28 日

## 1 手动实现单隐层全连接神经网络

### 1.1 训练过程中训练集上损失随 epoch 的变化曲线图

自己手动实现的 FCNN 训练所得 loss 变化如图 1

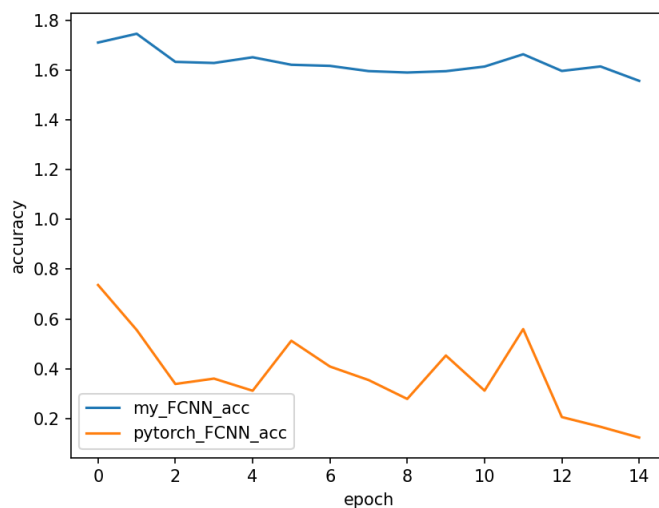


图 1: loss

### 1.2 性能评估

自己手动实现 FCNN 的训练过程如图 2

训练完成后，对测试集进行预测，所得的准确率如图 3

自己手动实现FCNN的训练过程如下

Epoch0损失为 1.709956169128418

Epoch1损失为 1.7452867031097412

Epoch2损失为 1.6324107646942139

Epoch3损失为 1.627949595451355

Epoch4损失为 1.6508899927139282

Epoch5损失为 1.6208078861236572

Epoch6损失为 1.6163491010665894

Epoch7损失为 1.5953319072723389

Epoch8损失为 1.5896148681640625

Epoch9损失为 1.5950055122375488

Epoch10损失为 1.6135588884353638

Epoch11损失为 1.6629339456558228

Epoch12损失为 1.595992922782898

Epoch13损失为 1.6139957904815674

Epoch14损失为 1.556248664855957

自己手动实现的FCNN模型训练完成

图 2: 自己手动实现 FCNN

自己手动实现的FCNN进行预测

自己手动实现下，测试集上的正确率为：**0.8847155448717948**

图 3: my\_FCNN\_acc

### 1.3 参数分析实验

1. 训练周期 (epoch)：当训练周期低于 12 时，会出现欠拟合情况，导致在训练集和测试集上的性能都较差。当训练周期高于 20 时，会出现过拟合情况，导致在训练集上有较好的性能，但在测试集上表现较差。故选择训练周期为 15，使得模型能够较好的拟合。
2. 隐藏层神经元个数 (num\_hiddens)：当个数设置为 64 时，准确率降低了 0.005。当个数设置为 512 时，准确率降低了 0.004。而当个数设置为 128 或 256 时，正确率在 0.885 左右。由此可见，隐藏层神经元个数在本次实验中对准确率影响较小，但选择 256 以达到最高准确率。
3. 学习率 (learning rate)：当学习率低于 0.07 时，会使模型收敛过慢，导致在训练了 15 个轮次之后仍旧无法达到或接近局部最小值，而当学习率高于 0.4 时，会使模型不收敛，导致模型无论如何训练都无法达到或接近局部最小值。故选择学习率为 0.1，使得模型能够以较快且合适的速度收敛至局部最小值附近。

#### 1.3.1 结果分析

自己手动实现的 FCNN 模型在测试集上的准确率在 0.885，取得了较好的预测结果。如需继续提升正确率，可能需要对原始数据进行预处理，并在自己手动搭建的 FCNN 中增加隐藏层层数，从而继续提高改模型性能

## 2 使用 PyTorch 库简洁实现全连接神经网络

### 2.1 训练过程中训练集上损失随 epoch 的变化曲线

调用 pytorch 库实现的 FCNN 训练所得 loss 变化如图 1

## 2.2 性能评估结果

使用 pythorch 实现 FCNN 的训练过程如图 4

```
调用pytorch库实现FCNN的训练过程如下
Epoch0损失为 0.7364422082901001
Epoch1损失为 0.5559237003326416
Epoch2损失为 0.33907440304756165
Epoch3损失为 0.3606775403022766
Epoch4损失为 0.3119339942932129
Epoch5损失为 0.5126023888587952
Epoch6损失为 0.4091530740261078
Epoch7损失为 0.35488376021385193
Epoch8损失为 0.2789960503578186
Epoch9损失为 0.4533287286758423
Epoch10损失为 0.31239134073257446
Epoch11损失为 0.5594244003295898
Epoch12损失为 0.20622463524341583
Epoch13损失为 0.16725631058216095
Epoch14损失为 0.12391434609889984
调用pytorch库实现的FCNN模型训练完成
```

Figure 4: Training process of FCNN using pytorch

图 4: pytorch 实现 FCNN

训练完成后，对测试集进行预测，所得的准确率如图 5

调用pytorch库实现的FCNN进行预测  
pytorch方法实现下，测试集上的正确率为： 0.8819

图 5: pytorch\_FCNN\_acc

## 2.3 参数分析实验

### 2.3.1 参数设置

1. 训练周期 (epoch)：当训练周期低于 12 时，会出现欠拟合情况，导致在训练集和测试集上的性能都较差。当训练周期高于 20 时，会出现过拟合情况，导致在训练集上有较好的性能，但在测试集上表现较差。故选择训练周期为 15，使得模型能够较好的拟合。
2. 学习率 (learning rate)：当学习率低于 0.07 时，会使模型收敛过慢，导致在 15 个轮次之后仍旧无法达到局部最小处，而当学习率高于 0.4 时，会使模型不收敛，导致模型无论如何训练都无法达到或接近局部最小值。故选择学习率为 0.1，使得模型能够以较快且合适的速度收敛至局部最小处附近。
3. 激活函数 (activate\_function)：当使用 sigmoid 函数作为激活函数时，正确率约为 0.84，相较于原激活函数降低了 0.03 左右，可见该模型适合使用 ReLU 函作为激活函数

### 2.3.2 结果分析

调用 pytorch 库实现的 FCNN 的模型在测试集上的准确率在 0.881 左右，取得了较好的预测结果。如要继续提升正确率，可能需要优化数据的预处理方式以及增加神经网络的层数才能够继续提高该模型的性能。