

数据拟合模型-使用神经网络进行昆虫分类

Xiaoma

2023 年 5 月 8 日

摘要

神经网络作为一种已被广泛使用的模型，已经被应用于各个领域，本文将使用全连接神经网络探索昆虫的分类问题，并对分类结果进行讨论。

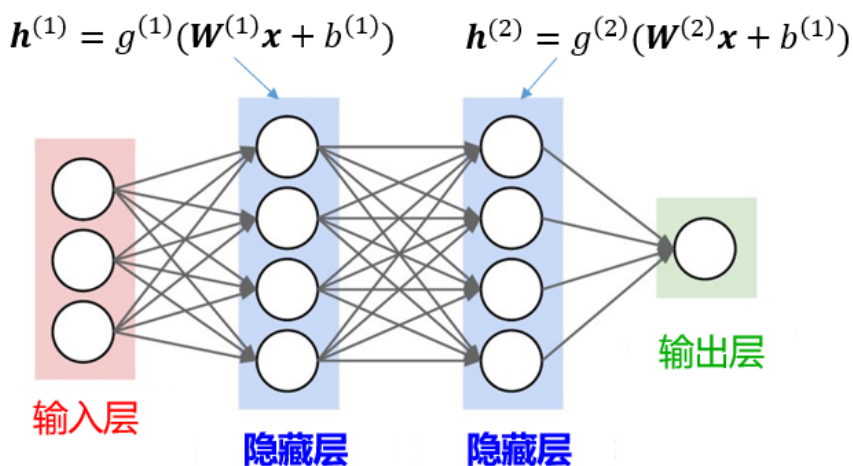
一、前言

人工神经网络（Artificial Neural Networks，简称为 ANNs）也简称为神经网络（NNs）或称作连接模型（Connection Model），它是一种模仿动物神经网络行为特征，进行分布式并行信息处理的算法数学模型。这种网络依靠系统的复杂程度，通过调整内部大量节点之间相互连接的关系，从而达到处理信息的目的。

神经网络的优势：

- 解决线性模型难以处理的线性不可分问题
- 线性模型泛化能力较弱
- 等等

全连接神经网络的结构图为



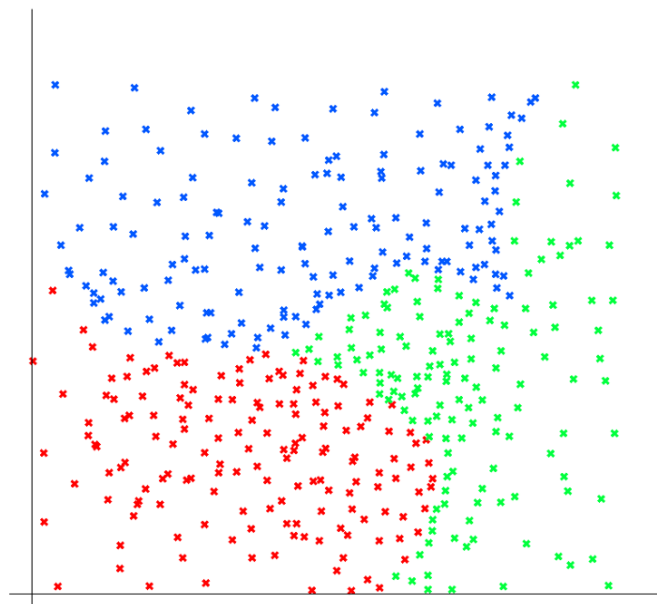
二、相关工作

1. 基于全连接神经网络实现昆虫分类问题
2. 比较不同网络结构对分类器性能的影响
3. 比较不同优化器，激活函数对分类器性能的影响

三、问题分析

3.1 分析

根据数据的聚类分布，我们发现昆虫的分类边界是非线性的，故使用线性模型很难得到较好的结果，故使用全连接神经网络模型。



四、建模的假设

4.2 假设 1

假设数据集中没有确实值

4.3 假设 2

假设每个数据对应的标签一定正确

五、符号说明

符号	说明
Model	全连接神经网络分类器
Net	全连接神经网络模型
Val_ACC	分类器的验证集精度
Train_ACC	分类器的训练集精度
Val_LOSS	分类器的验证集损失
Train_LOSS	分类器的训练集损失

六、数学模型建立

6.4 神经网络的结构

首先搭建一个神经网络模型，其结构为（初始结构不代表最终网络结构）

```
Net(  
    (net): Sequential(  
      (0): Linear(in_features=2, out_features=10, bias=True)  
      (1): ReLU()  
      (2): Linear(in_features=10, out_features=100, bias=True)  
      (3): ReLU()  
      (4): Linear(in_features=100, out_features=10, bias=True)  
      (5): ReLU()  
      (6): Linear(in_features=10, out_features=3, bias=True)  
    )  
  )  
)
```

6.5 模型调节-网络层数与神经元个数

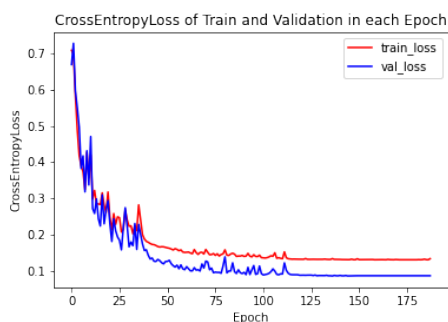
已知数据集样本个数与特征数目较少, 故首先假设该分类器不需要过大的网络深度与神经元个数, 分别尝试不同的网络层数与神经元个数, 结果为

网络层数	每层最大神经元数目	Train_ACC	Val_ACC
3	100	0.935	0.955
5	100	0.947	0.951
10	100	0.952	0.976
3	10	0.926	0.933
5	10	0.910	0.922
10	10	0.916	0.948

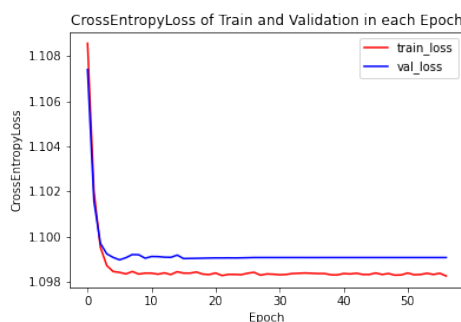
6.6 模型调节-优化器与激活函数

6.6.1 优化器选择

在上述参数调节过程中使用的优化器为 **optim.Adam**, 将其训练过程的 loss 曲线与 **optim.SGD** 进行比较



(a) LOSS – Adam

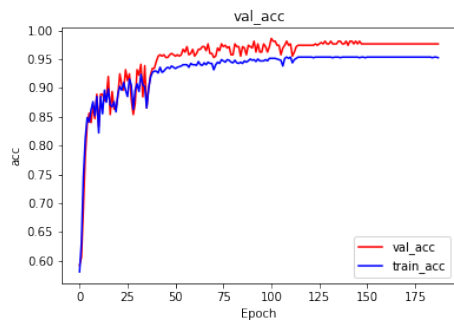


(b) ACC – SGD

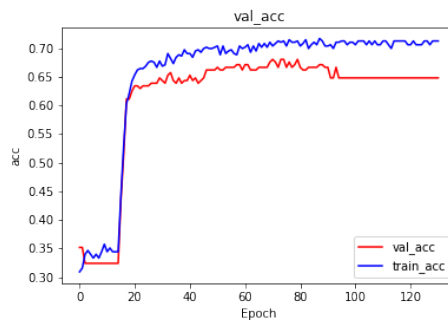
可以发现使用 **optim.Adam** 的分类器在训练过程中收敛速度和收敛值均优于使用 **optim.SGD**。

6.6.2 激活函数选择

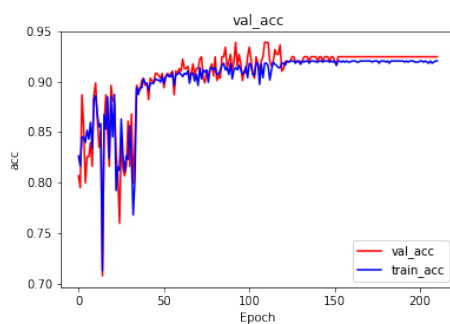
分别使用 **ReLU**, **Softmax**, **Tanh** 进行分类器训练, 并比较其精度的变化情况。
发现使用 **ReLU** 作为激活函数时, 分类器表现最优。



(c) $ACC - ReLU$



(d) $ACC - Softmax$

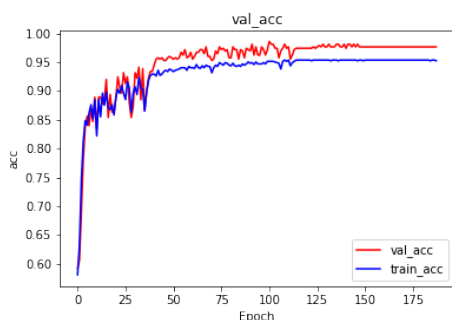


(e) $ACC - Tanh$

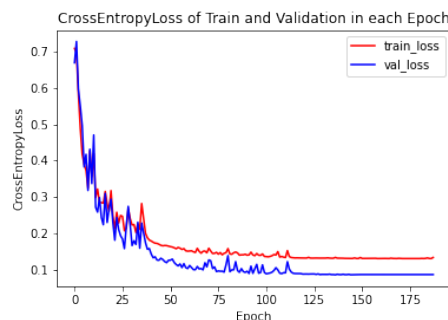
七、结果（与对比）

根据上述参数调节，我们得到的最优网络结构如下，使用其训练得到分类器并保存模型。

```
Net(  
    (net): Sequential(  
      (0): Linear(in_features=2, out_features=10, bias=True)  
      (1): ReLU()  
      (2): Linear(in_features=10, out_features=100, bias=True)  
      (3): ReLU()  
      (4): Linear(in_features=100, out_features=100, bias=True)  
      (5): ReLU()  
      (6): Linear(in_features=100, out_features=100, bias=True)  
      (7): ReLU()  
      (8): Linear(in_features=100, out_features=100, bias=True)  
      (9): ReLU()  
      (10): Linear(in_features=100, out_features=100, bias=True)  
      (11): ReLU()  
      (12): Linear(in_features=100, out_features=100, bias=True)  
      (13): ReLU()  
      (14): Linear(in_features=100, out_features=100, bias=True)  
      (15): ReLU()  
      (16): Linear(in_features=100, out_features=100, bias=True)  
      (17): ReLU()  
      (18): Linear(in_features=100, out_features=10, bias=True)  
      (19): ReLU()  
      (20): Linear(in_features=10, out_features=3, bias=True)  
    )  
)
```



(f) LOSS



(g) ACC

八、结论

1. 若激活函数与优化器合适，增大网络深度与宽度可以提升分类器性能
2. 若数据集较小且规整，网络结构的调整对分类器的性能影响较小
3. 对于本次实验数据集，Adam 优化比 SGD 优化更好
4. 对于本次实验数据集，激活函数选择 ReLU 性能更好

九、问题

参数调节过程为分步调参，二维交叉调参。

由于时间原因，并没有尝试其他神经网络框架进行分析。