

1、概念： 人工神经网络是由一系列的单元相互紧密联系构成的，每个单元由一定的实数输入和唯一的实数输出。神经网络的一个重要用途就是接受和处理传感器产生的复杂输入并进行自适应的学习。人工神经网络算法模拟生物神经网络，是一种模式匹配算法，通常用于解决分类和回归问题。

2、感知机模型： 感知机是一种线性分类器，它将每一个实例分类为正类（+1）和负类（-1）感知机的物理意义是：它将输入空间（特征空间）划分为正负两类的分离超平面。

感知机的学习策略就是损失函数最小化。

3、从感知机到神经网络： 神经网络中最基本的成分是神经元。

每个神经元与其他神经元相连。

感知机只有输出层神经元进行激活函数处理，即只拥有一层功能神经元。输入层并不进行激活函数处理。

感知机只有一层功能神经元，他只能处理线性可分问题。

4、多层功能神经元： 每层神经元与下一层神经元全部互连。

同层神经元不存在互连

跨层神经元也不存在互连

这样的神经网络结构通常称为多层前馈神经网络。

神经网络的学习就是根据训练数据集来调整神经元之间的连接权重，以及每个神经元的阈值。

多层前馈神经网络的学习通常采用误差逆传播算法（BP算法）

BP算法从原理上就是普通的梯度下降法求最小值的问题。它关键的地方在一下两个：

导数的链式法则

sigmoid激活函数的性质;sigmoid函数求导的结果

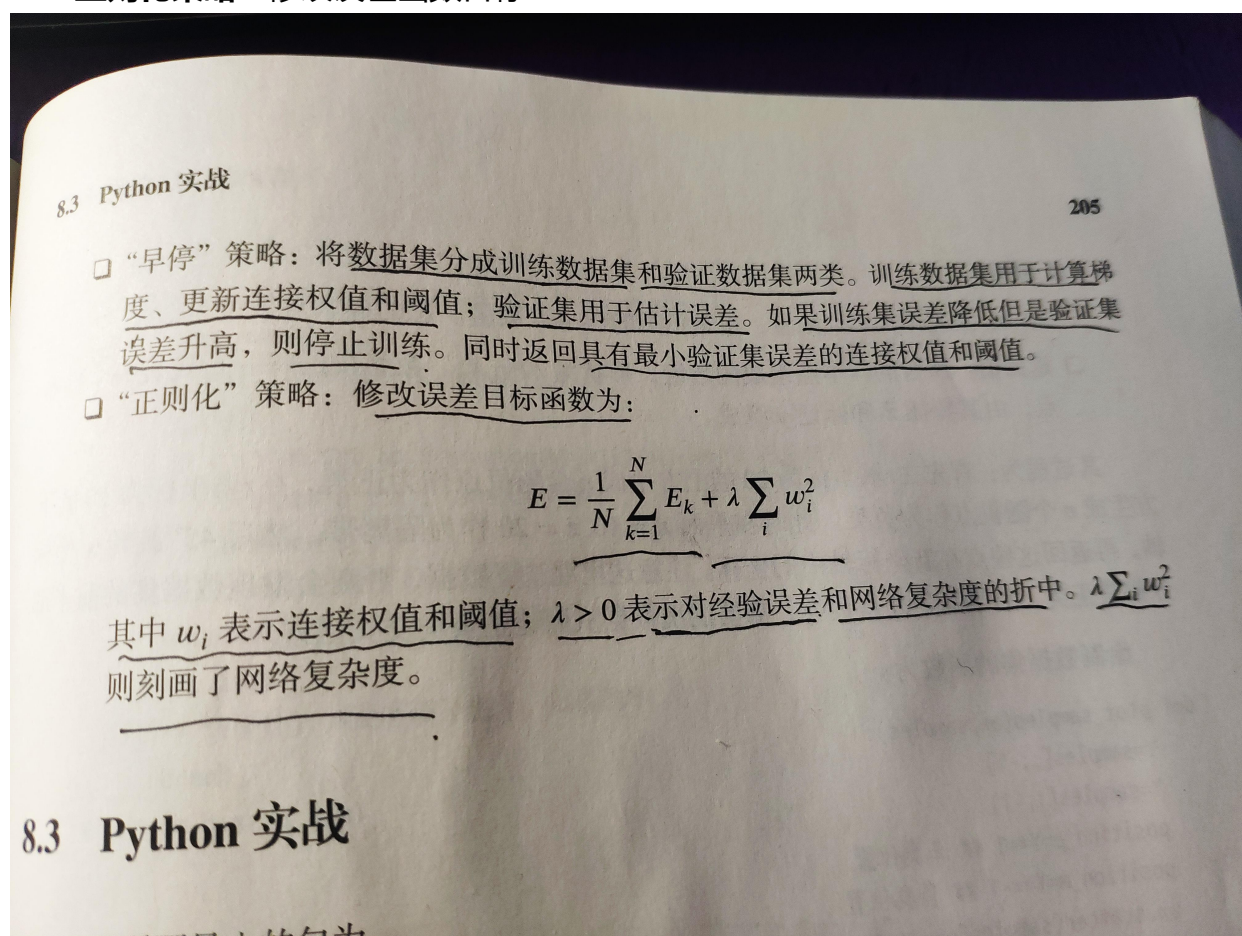
等于自变量的乘积形式。

BP神经网络表达能力非常强大，因此可能遇到过拟合：其训练误差降低，但是测试误差上升。

有两种方法来缓解过拟合。

早停策略：将数据集分成训练数据集和验证数据集两类。训练数据集用于计算梯度、更新连接权值和阈值；验证集用于估计误差。如果训练误差降低但是验证集误差升高则停止训练。同时返回具有最小验证集误差的连接权值和阈值。

正则化策略：修改误差函数目标



5、感知机学习算法只能用于线性可分数据集

6、多层神经网络：

```
from sklearn.neural_network import MLPClassifier
```

重要参数（其余参数参考文档）：

hidden_layer_sizes（如何设置隐含层神经元的个数理论上尚未解决，实际中靠试错法来调整）：一个元组。元组指定了隐含层结构。元组长度为隐含层的层数，元组的元素则指定了每一层隐含层中功能神经元的数量。默认参数只有一层，该层有100个功能神经元。

activation：是个字符串。该参数指定了激活函数的类型。可以为：

‘logistic’：激活函数为sigmoid函数

‘tanh’：激活函数为tanh函数

‘relu’：激活函数为修正线性单元。

algorithm：是个字符串。该参数指定了最优化算法的类型。可以为下列值：

'l-bfgs' :一种伪牛顿算法。对于小规模数据集，这种算法效果较好。

'sgd' : 随机梯度下降算法。

'adam' :对于相对较大规模的数据集，这种算法的效果相当好。

learning_rate_init:一个小数。该参数为初始学习率。只有当使用sgd算法或者adam算法时，该参数才有意义。