

# 机器学习第 7 次实验报告

李宜霖苏家兴窦志成

2021 年 06 月 16 日

## 目录

<b>1</b>	<b>概述</b>	<b>3</b>
<b>2</b>	<b>BP 算法推导</b>	<b>3</b>
2.1	权重调整 . . . . .	4
2.2	阈值调整 . . . . .	4

## 1 概述

这里是概述

- 第一条
- 第二条
- 第三条

Code 1: 这里是代码标题

```
1 def convolution_forward(X, W, b, stride = 1, pad = 0):
2     cal_X = im2col_forward(X, 5, 5, stride, pad)
3     result = np.dot(cal_X[0], W)
4     result += b
5     result = sigmoid(result)
6     return result
```

## 2 BP 算法推导

反向传播算法使用梯度下降策略对神经网络参数进行调整。以全连接神经网络为例，如图1，该网络有三层：

1. 含有  $d$  个神经元的输入层
2. 含有  $q$  个神经元的隐层
3. 含有  $l$  个神经元的输出层

假设网络的激活函数都是 Sigmoid 函数，即  $f(x) = \frac{1}{1+e^{-x}}$ ，每个神经元接受的输入如图1所示，

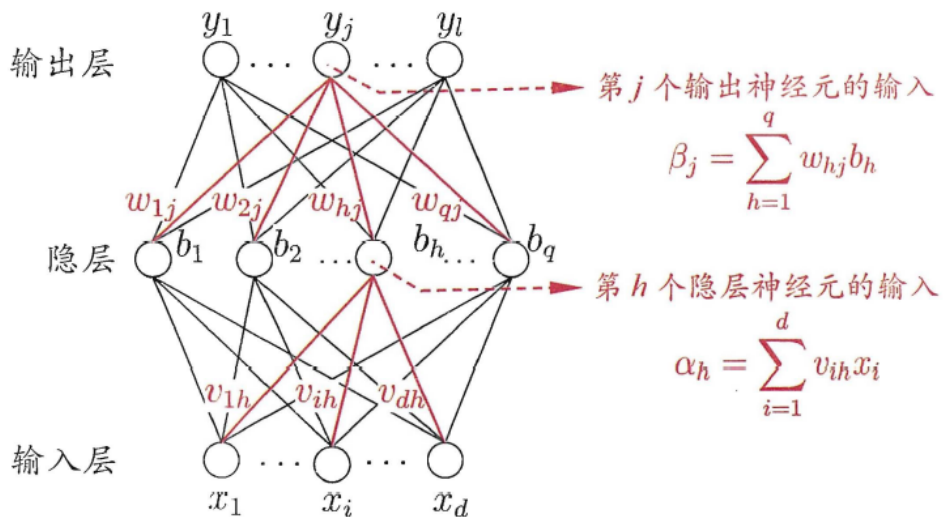


图 1: 含有单层隐含层的三层神经网络

定义误差函数为  $E = \frac{\sum_{i=1}^l (\hat{y}_i - y_i)^2}{2}$ ，即均方误差函数。以隐层到输出层的参数调整为例。

## 2.1 权重调整

权重  $w_{ij}$  的调整方式如公式1。

$$\begin{aligned}\Delta w_{ij} &= -\eta \frac{\partial E}{\partial w_{ij}} \\ &= -\eta \frac{\partial E}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial \hat{\beta}_j} \frac{\partial \hat{\beta}_j}{\partial w_{ij}}\end{aligned}\tag{1}$$

由于  $\frac{\partial E}{\partial \hat{y}_j} = \hat{y}_j - y_j$ ，Sigmoid 函数满足  $f'(x) = f(x)(1 - f(x))$ ， $\frac{\partial \hat{\beta}_j}{\partial w_{ij}} = b_i$ 。因此公式1可以变换为如公式2形式。

$$\begin{aligned}\Delta w_{ij} &= -\eta(\hat{y}_j - y_j)(\hat{y}_j(1 - \hat{y}_j))(b_i) \\ &= -\eta b_i \hat{y}_j(\hat{y}_j - y_j)(1 - \hat{y}_j)\end{aligned}\tag{2}$$

## 2.2 阈值调整

类似于权重调整，也对  $E$  求阈值  $\theta$  的导数。

$$\Delta \theta_j = -\eta \frac{\partial E}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial \theta_j}\tag{3}$$

由于 Sigmoid 函数满足  $f'(x) = f(x)(1 - f(x))$ ，因此公式3可以变换为如公式4形式。

$$\begin{aligned}\Delta \theta_j &= -\eta(\hat{y}_j - y_j)(-1)(\hat{y}_j(1 - \hat{y}_j)) \\ &= \eta \hat{y}_j(\hat{y}_j - y_j)(1 - \hat{y}_j)\end{aligned}\tag{4}$$

每次训练后，朝误差减少的方向调整参数。其中  $\eta$  是学习率。