

深度神经网络的传播算法

前向传播

1. 初始化

输入层: $A^{[0]} = X$ (大小为 $n^{[0]} \times m$)

2. 从第1层到第 $L - 1$ 层 (隐藏层)

对于每一层 $l = 1$ 到 $L - 1$:

权重矩阵: $W^{[l]}$ (大小为 $n^{[l]} \times n^{[l-1]}$)

偏置向量: $b^{[l]}$ (大小为 $n^{[l]} \times 1$)

线性部分: $Z^{[l]} = W^{[l]}A^{[l-1]} + b^{[l]}$ (大小为 $n^{[l]} \times m$)

激活函数: $A^{[l]} = g^{[l]}(Z^{[l]})$ (大小为 $n^{[l]} \times m$)

3. 输出层 (第 L 层)

权重矩阵: $W^{[L]}$ (大小为 $n^{[L]} \times n^{[L-1]}$)

偏置向量: $b^{[L]}$ (大小为 $n^{[L]} \times 1$)

线性部分: $Z^{[L]} = W^{[L]}A^{[L-1]} + b^{[L]}$ (大小为 $n^{[L]} \times m$)

激活函数: $A^{[L]} = g^{[L]}(Z^{[L]})$ (大小为 $n^{[L]} \times m$)

4. 损失函数

$J(A^{[L]}, Y)$ (标量, 大小为 1)

反向传播

1. 初始化梯度

输出层梯度: $dA^{[L]} = \frac{\partial J(A^{[L]}, Y)}{\partial A^{[L]}}$ (大小为 $n^{[L]} \times m$)

2. 从第 L 层到第2层

对于每一层 $l = L$ 到 2:

激活函数的导数: $dZ^{[l]} = dA^{[l]} \odot g'^{[l]}(Z^{[l]})$ (大小为 $n^{[l]} \times m$)

权重梯度: $dW^{[l]} = \frac{1}{m} dZ^{[l]} (A^{[l-1]})^T$ (大小为 $n^{[l]} \times n^{[l-1]}$)

偏置梯度: $db^{[l]} = \frac{1}{m} np.sum(dZ^{[l]}, axis = 1, keepdims = True)$ (大小为 $n^{[l]} \times 1$)

上一层梯度: $dA^{[l-1]} = (W^{[l]})^T dZ^{[l]}$ (大小为 $n^{[l-1]} \times m$)

3. 第1层

激活函数的导数: $dZ^{[1]} = dA^{[1]} \odot g'^{[1]}(Z^{[1]})$ (大小为 $n^{[1]} \times m$)

权重梯度: $dW^{[1]} = \frac{1}{m} dZ^{[1]} (A^{[0]})^T$ (大小为 $n^{[1]} \times n^{[0]}$)

偏置梯度: $db^{[1]} = \frac{1}{m} np.sum(dZ^{[1]}, axis = 1, keepdims = True)$ (大小为 $n^{[1]} \times 1$)

参数更新

对于每一层 $l = 1$ 到 L :

$W^{[l]} := W^{[l]} - \alpha \cdot dW^{[l]}$

$b^{[l]} := b^{[l]} - \alpha \cdot db^{[l]}$

这里, α 是学习率。