BP神经网络(反向传播)详细推导

2018-05-21 | 机器学习 | 56

849 | 3

这篇文章主要讨论神经网络的反向传播的细节，“误差”是如何反向传播的，我们又是如何利用梯度来优化参数的。

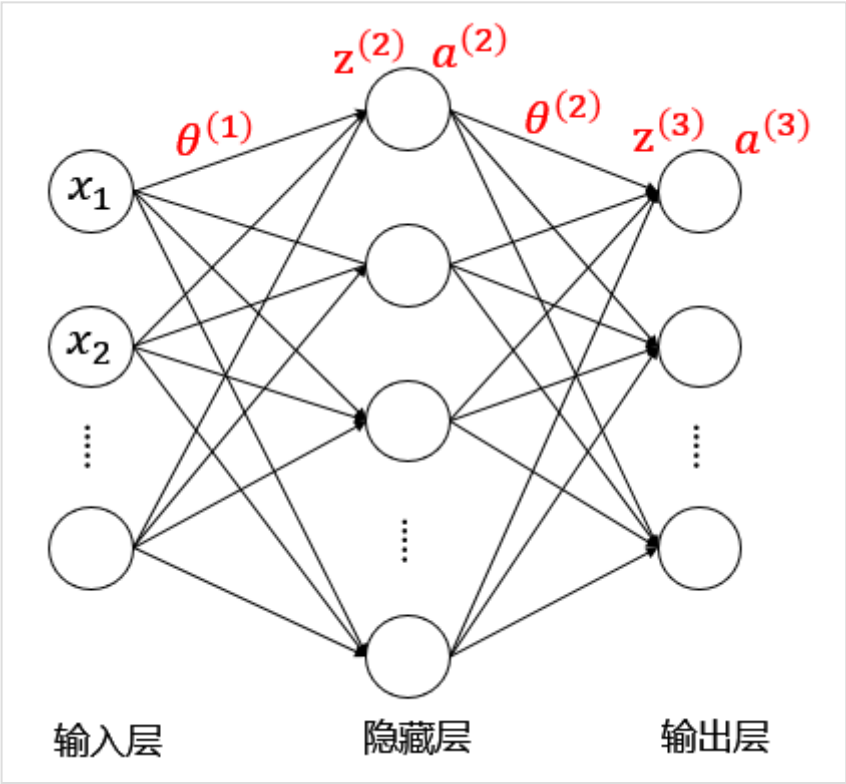
在学吴恩达机器学习视频的神经网络那节时，给出了许多公式，比如计算每层的误差，每层参数的梯度，但并没有给出推导过程，可能也是考虑入门级，大多人并不要知道其中含义就可以运用算法了。接下来我会给出详细的推导过程，帮助大家理解。

注意接下来所讲是未正则化的神经网络。

1 计算公式

1.1 正向传递

假设现在有一个三层的神经网络，如图：



参数含义：

- $\theta^{(i)}$ 第 i 层的参数矩阵
- $z^{(l)}$ 第 l 层的输入
- $a^{(l)}$ 第 l 层的输出

传递过程：

- $a^{(1)} = x$
- $z^{(2)} = \theta^{(1)} a^{(1)}$
- $a^{(2)} = g(z^{(2)})(add\ a_0^{(2)})$
- $z^{(3)} = \theta^{(2)} a^{(2)}$
- $h = a^{(3)} = g(z^{(3)})$

其中g 为sigmoid激活函数。

1.2 反向传播

我们用$\delta^{(l)}$ 表示每层的”误差“，y 为每个样本的标签，h 为每个样本的预测值。

先来从后往前计算每层的“误差“。注意到这里的误差用双引号括起来，因为并不是真正的误差。

- $\delta^{(3)} = h - y$ (1)
- $\delta^{(2)} = (\theta^{(2)})^T \delta^{(3)} g`(z^{(2)})$ (2)

注意第一层是没有误差的，因为是输入层。

吴恩达在课里面提到，” 误差 “的实质是 $\delta^{(l)} = \frac{\partial J}{\partial z^{(l)}}$ ，没错，后面详细说明。

然后来计算每层参数矩阵的梯度，用$\Delta^{(l)}$ 表示

- $\Delta^{(2)} = a^{(2)}\delta^{(3)}$ (3)
- $\Delta^{(1)} = a^{(1)}\delta^{(2)}$ (4)

最后网络的总梯度为：

- $D = \frac{1}{m}(\Delta^{(1)} + \Delta^{(2)})$ (5)

到这里反向传播就完成了，接着就可以利用梯度下降法或者更高级的优化算法来训练网络。

2 推导

这里只推导 δ 和 Δ 是怎么来的，其余的比较好理解。

首先明确我们要优化的参数有 $\theta^{(1)}$，$\theta^{(2)}$ ，利用梯度下降法的思想，我们只需要求解出代价函数对参数的梯度即可。

假设只有一个输入样本，则代价函数是：

$$J(\theta) = -ylogh(x) - (1-y)log(1-h)$$

回顾下正向传递的过程，理解其中函数的嵌套关系：

- $a^{(1)} = x$
- $z^{(2)} = \theta^{(1)}a^{(1)}$
- $a^{(2)} = g(z^{(2)})(add\ a_0^{(2)})$
- $z^{(3)} = \theta^{(2)}a^{(2)}$
- $h = a^{(3)} = g(z^{(3)})$

然后我们来求解代价函数对参数的梯度，$\frac{\partial}{\partial\theta^{(2)}}J(\theta)$ ，$\frac{\partial}{\partial\theta^{(1)}}J(\theta)$ 。

根据链式求导法则，可以计算得到：

$$\underbrace{\frac{\partial J}{\partial \theta^{(2)}}}_{\Delta^{(2)}} = \underbrace{\frac{\partial J}{\partial a^{(3)}}\frac{\partial a^{(3)}}{\partial z^{(3)}}}_{\delta^{(3)}}\frac{\partial z^{(3)}}{\partial \theta^{(2)}} = (h-y)a^{(2)}$$

把我画红线的地方令为$\delta^{(3)}$，是不是就得到了反向传播中的公式（1）？

把画绿线的部分令为$\Delta^{(2)}$，就得到了公式（3）。我们接着算：

$$\begin{aligned}\Delta^{(1)}\frac{\partial J}{\partial \theta^{(1)}} &= \frac{\partial J}{\partial a^{(3)}}\frac{\partial a^{(3)}}{\partial z^{(3)}}\frac{\partial z^{(3)}}{\partial a^{(2)}}\frac{\partial a^{(2)}}{\partial z^{(2)}}\frac{\partial z^{(2)}}{\partial \theta^{(1)}} \\ &= \delta^{(3)}\theta^{(2)}g'(z^{(2)})a^{(1)} \\ &= \delta^{(2)}a^{(1)}\end{aligned}$$

同样把红线部分令为$\delta^{(3)}$，紫色部分令为$\delta^{(2)}$，就得到了公式（2）。

绿线部分令为$\Delta^{(1)}$，就得到了公式（4）。

至此，推导完毕。得到这个规律后，便可以应用到深层次的网络中，计算反向传播时就很方便了。

上面的公式因为书写麻烦，便只写了结果。如果你用笔去慢慢推几分钟，会发现其实很简单。

欢迎订阅公众号，与我随时交流哦~

机器学习

‹ 吴恩达机器学习作业Python实现(三)：多类分类和前馈神经网络

吴恩达机器学习作业Python实现(四)：神经网络反向传播 ›