反向传播（Backpropagation）算法的数学原理简单介绍

第一章 引言

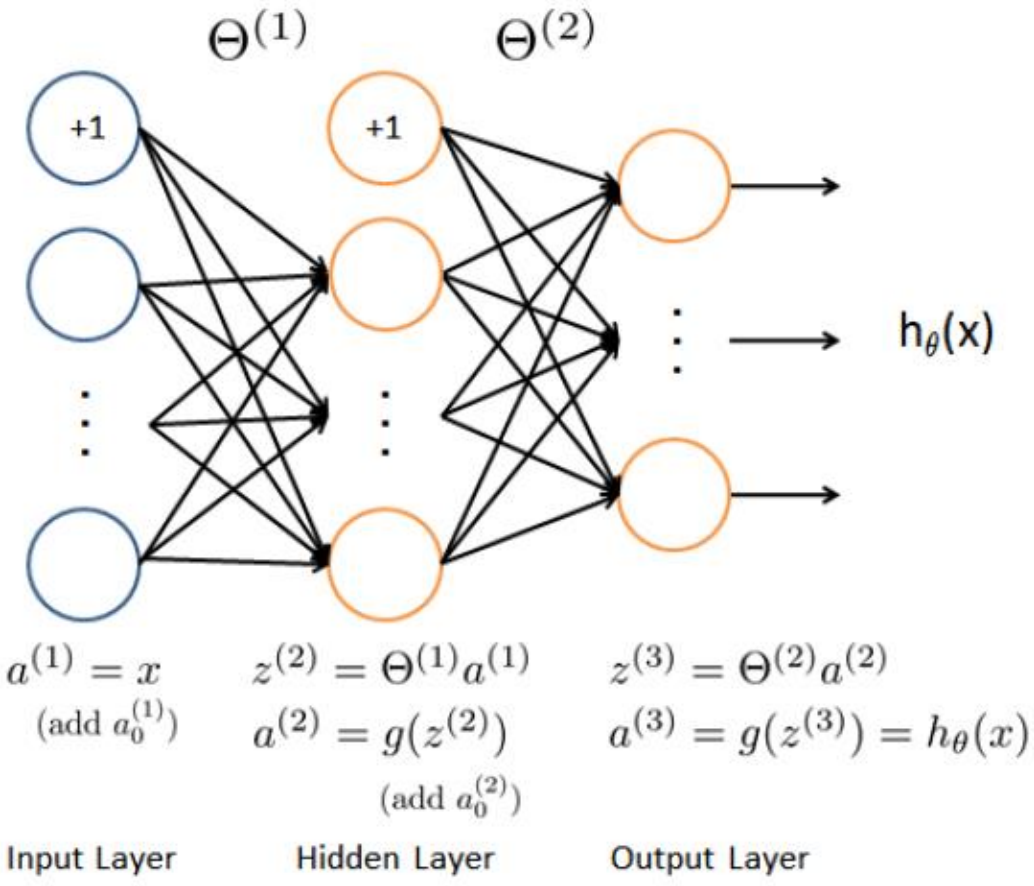

图 1：一个简单的人工神经网络

如图 1 所示，这是一个简单的人工神经网络，分为输入层，隐藏层和输出层。其中 $\Theta(1),\Theta(2)$ 分别是第 1 层，第 2 层向前传递数据时的线性组合参数，将 $\Theta(1),\Theta(2)$ 统称为 Θ。$z^{(2)},z^{(3)}$ 是线性组合的输出，作为函数 $g(z)$ 的输入，$g(z)$ 是一个数学变换（激活函数），最后一层的输出，即 $a(3)$ 或 $h_\theta(x)$ ，即人工神经网络的运算结果。

在训练时，将已经标记过的数据（标签可以表示为 y）输入网络，最开始的时候网络输出结果与标签结果 y 往往相距甚远，然后用优化算法（例如梯度下降法）不断修正 Θ ，使网络的计算结果越来越接近标准标签结果 y，直至达到要求的精度。

我们首先定义一个 **代价方程（Cost Function）**：

$$J(\Theta) = \left[\textit{计算结果 与 标准答案 之间的差距}\right]$$

我们暂时不关心 Cost Function 的具体形式，只需要理解它是对计算误差的一种衡量，并且它是 Θ 的函数。我们的目标就是使计算误差尽量小。以梯度下降法为代表的很多优化算法可以用来解决上述的最小化问题。在优化过程中，这些算法通常会用到 $J(\Theta)$ 对 Θ 中各元素的偏导数（通常 Θ 为一个矩阵），反向传播算法就包含计算偏导数的来更新 Θ。

需要注意到的一点是，在这个简单的人工神经网络中，除了最后一层以外，每一层都要在自己的输出向量前附加一个值为 $+1$ 的项（称为 Bias Term）。可以借用二维坐标系中的一条直线：$ax + b = 0$，Bias term 与对应参数相乘后的值就是直线的截距 b.

第二章 数学原理

2.1 定义

L ——神经网络的层数（从 1 开始编号，例如 layer1,layer2…）

S_l ——第 l 层所包含的单元数(不计入 Bias term)

(x,y) ——一条训练数据

$Z_i^{(l)}$ ——第 l 层中第 i 个单元的输入（$i = 1,2 \dots, S_l$）

$a_i^{(l)}$ ——第 l 层中第 i 个单元的输出（$i = 0,1,2 \dots, S_l$），$a_0^{(l)}$ = +1 即 bias term;

$$a_i^{(l)} = g\left(z_i^{(l)}\right) = \frac{1}{1 + e^{-z_i^{(l)}}}$$

$\Theta_{ij}^{(l)}$ ——对应 $a_i^{(l)}$ 的参数，用于计算 $Z_i^{(l+1)}$；其中:

$$j = 0,1,2 \dots, S_l;\ i = 1,2 \dots, S_{l+1}$$

$h_\Theta(x)$ ——基于 Θ，给定 x 后，人工神经网络的输出

$J(\Theta)$ ——代价方程（Cost Function）:

$$J(\Theta) = -y \log\left(h_\Theta(x)\right) - (1 - y) \log\left(1 - h_\Theta(x)\right)$$

$\delta_i^{(l)}$ —— 第 l 层第 i 个单元的计算误差：

$$\delta_i^{(l)} = \frac{\partial}{\partial z_i^{(l)}} J(\Theta) \qquad (i = 1,2 \dots, S_l)$$

2.2 数学推导

目标：对于所有的 l,i,j，计算 $\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta)$

过程：利用链式求导法则

$$\frac{\partial J}{\partial \Theta_{ij}^{(l)}} = \frac{\partial J}{\partial a_i^{(l+1)}} \cdot \frac{\partial a_i^{(l+1)}}{\partial z_i^{(l+1)}} \cdot \frac{\partial z_i^{(l+1)}}{\partial \Theta_{ij}^{(l)}}$$

①, ②, ③

$$① = \frac{\partial J}{\partial a_i^{(l+1)}} = \frac{\partial J}{\partial z_1^{(l+2)}} \cdot \frac{\partial z_1^{(l+2)}}{\partial a_i^{(l+1)}} + \frac{\partial J}{\partial z_2^{(l+2)}} \cdot \frac{\partial z_2^{(l+2)}}{\partial a_i^{(l+1)}} + \cdots + \frac{\partial J}{\partial z_{s_{l+2}}^{(l+2)}} \cdot \frac{\partial z_{s_{l+2}}^{(l+2)}}{\partial a_i^{(l+1)}}$$

$$= \sum_{j=1}^{s_{l+2}} \frac{\partial J}{\partial z_j^{(l+2)}} \cdot \frac{\partial z_j^{(l+2)}}{\partial a_i^{(l+1)}}$$

$$= \sum_{j=1}^{s_{l+2}} \delta_j^{(l+2)} \cdot \frac{\partial z_j^{(l+2)}}{\partial a_i^{(l+1)}}$$

$$= \sum_{j=1}^{s_{l+2}} \delta_j^{(l+2)} \cdot \frac{\partial}{\partial a_i^{(l+1)}} \left(\Theta_{j0}^{(l+1)} a_0^{(l+1)} + \Theta_{j1}^{(l+1)} a_1^{(l+1)} + \cdots + \Theta_{js_{l+1}}^{(l+1)} a_{s_{l+1}}^{(l+1)} \right)$$

$$= \sum_{j=1}^{s_{l+2}} \delta_j^{(l+2)} \cdot \Theta_{ji}^{(l+1)}$$

($i = 1,2 \ldots, S_l$；不包括 bias term)

$$② = \frac{\partial a_i^{(l+1)}}{\partial z_i^{(l+1)}}$$

$$= \frac{\partial}{\partial z_i^{(l+1)}} g\left(z_i^{(l+1)}\right)$$

$$= g\left(z_i^{(l+1)}\right) \cdot \left(1 - g\left(z_i^{(l+1)}\right)\right)$$

$$= a_i^{(l+1)} \left(1 - a_i^{(l+1)}\right)$$

($i = 1,2 \ldots, S_l$；不包括 bias term)

$$③ = \frac{\partial z_i^{(l+1)}}{\partial \Theta_{ij}^{(l)}}$$

$$= \frac{\partial}{\partial \Theta_{ij}^{(l)}} \left(\Theta_{i0}^{(l)} a_0^{(l)} + \Theta_{i1}^{(l)} a_1^{(l)} + \cdots + \Theta_{is_l}^{(l)} a_{s_l}^{(l)} \right)$$

$$= a_j^{(l)}$$

($j = 0,1,2 \ldots, S_l$；包括 bias term)

$$\therefore \frac{\partial J}{\partial \Theta_{ij}^{(l)}} = ① \times ② \times ③$$

$$= \sum_{k=1}^{s_{l+2}} \delta_k^{(l+2)} \Theta_{ki}^{(l+1)} \cdot a_i^{(l+1)} \left(1 - a_i^{(l+1)}\right) \cdot a_j^{(l)}$$

根据 δ 的定义：

$$\delta_i^{(l+1)} = ① \times ② = \sum_{k=1}^{S_{l+2}} \delta_k^{(l+2)} \Theta_{ki}^{(l+1)} \cdot a_i^{(l+1)} \left(1 - a_i^{(l+1)}\right)$$

$$(i = 1,2 \ldots, S_{l+1})$$

将第 $(l+1)$ 层的所有单元误差表示为 $\delta^{(l+1)}$，则 $\delta_i^{(l+1)}$ 是其第 i 个元素：

$$\therefore \delta^{(l+1)} = \left(\Theta^{(l+1)}\right)^T \delta^{(l+2)} \cdot * a^{(l+1)} \cdot * \left(1 - a^{(l+1)}\right) \quad (4)$$

(·∗表示矩阵对应元素之间两两相乘)

$$\therefore \frac{\partial J}{\partial \Theta^{(l)}} = \delta^{(l+1)} \left(\mathrm{a}^{(l)}\right)^T \quad (5)$$

式（4）（5）即 Prof. Ng 所给出的反向传播算法的公式

2.3 补充说明：

1. 公式（4）不能用来计算 $\delta^{(L)}$，因为 $\delta^{(L+1)}$ 未定义。$\delta^{(L)}$ 的计算如下：

$$J(\Theta) = -y \log\left(h_\Theta(x)\right) - (1-y) \log\left(1 - h_\Theta(x)\right)$$

$$= -y \log\left(a^{(L)}\right) - (1-y) \log\left(1 - a^{(L)}\right)$$

$$\delta_i^{(L)} = \frac{\partial J}{\partial z_i^{(L)}} = \frac{\partial J}{\partial a_i^{(L)}} \cdot \frac{\partial a_i^{(L)}}{\partial z_i^{(L)}}$$

$$= \frac{\partial}{\partial a_i^{(L)}} \left[-y_i \log\left(a_i^{(L)}\right) - (1-y_i) \log\left(1 - a_i^{(L)}\right)\right] \cdot \frac{\partial}{\partial z_i^{(L)}} g\left(\partial z_i^{(L)}\right)$$

$$= a_i^{(L)} - y_i$$

$$\therefore S^{(L)} = a^{(L)} - y$$

2. 除了计算 $\delta^{(L)}$，其它运算都未用到 $J(\Theta)$ 的具体表达式；
3. 由于 $Z_0^{(l)}$ 未定义，所以 $\delta_0^{(l)} = \frac{\partial J}{\partial z_0^{(l)}}$ 未定义，所以不要试图计算 bias term 项的误差。

参考文献

[1]word:

https://github.com/BasicCoder/NNBlog1/tree/master/%E5%8F%8D%E5%90%91%E4%BC%A0%E6%92%AD%E7%AE%97%E6%B3%95(BP)%E7%9A%84%E6%95%B0%E5%AD%A6%E5%8E%9F%E7%90%86%E7%AE%80%E5%8D%95%E4%BB%8B%E7%BB%8D

[2]PDF: