人工神经网络反向传播算法推导

$x = a^{(0)} \Rightarrow z^{(1)} = w^{(1)} a^{(0)} + b^{(1)} \Rightarrow a^{(1)} = \varphi(z^{(1)})$

$\Rightarrow z^{(2)} = w^{(2)} a^{(1)} + b^{(2)} \Rightarrow a^{(2)} = \varphi(z^{(2)}) \cdots$

$\cdots \Rightarrow z^{(m)} = w^{(m)} a^{(m-1)} + b^{(m)} \Rightarrow a^{(m)} = \varphi(z^{(m)}) \cdots$

$\cdots \Rightarrow z^{(l)} = w^{(l)} a^{(l-1)} + b^{(l)} \Rightarrow$

$y = a^{(l)} = \varphi(z^{(l)})$

说明：① 网络共有 l 层。

② $z^{(k)}$，$a^{(k)}$ $b^{(k)}$ 为向量，用 $z_i^{(k)}$，$a_i^{(k)}$，$b_i^{(k)}$ 表示其第 i 个分量。

③ 输出 y 可以是向量，用 y_i 表示其第 i 个分量。

先看输入一个向量 X，其目标为 Y。BP算法分为以下四步：

① 随机初始化所有的 W，b。

~~② 计算偏导。~~

~~Minimize: $E = \frac{1}{2}\|y - Y\|^2$~~

② 将 X 代入，求得所有 z，a，y。

③ 链式求导法则求偏导。

Minimize: $E = \frac{1}{2}\|y - Y\|^2$

设 $\delta_i^{(m)} = \frac{\partial E}{\partial z_i^{(m)}}$，则有：

$$\delta_i^{(l)} = \frac{\partial E}{\partial z_i^{(l)}} = \frac{\partial E}{\partial y_i} \cdot \frac{\partial y_i}{\partial z_i^{(l)}}$$

$$= (y_i - Y_i) \cdot \varphi'(z_i^{(l)})$$

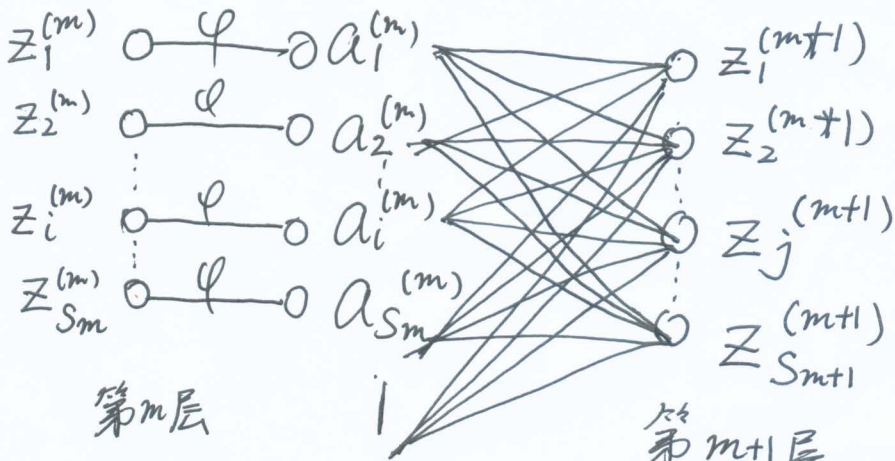
下面推导 $\delta^{(m)}$ 与 $\delta^{(m+1)}$ 关系。

$$\delta_i^{(m)} = \frac{\partial E}{\partial z_i^{(m)}} = \sum_{j=1}^{S_{m+1}} \frac{\partial E}{\partial z_j^{(m+1)}} \cdot \frac{\partial z_j^{(m+1)}}{\partial z_i^{(m)}}$$

(这里 S_{m+1} 为第 $m+1$ 层神经元个数。)

$$= \sum_{j=1}^{S_{m+1}} \delta_j^{(m+1)} \frac{\partial z_j^{(m+1)}}{\partial z_i^{(m)}}$$

为求第二项，我们画下图



因此：

$$\frac{\partial z_j^{(m+1)}}{\partial z_i^{(m)}} = \frac{\partial z_j^{(m+1)}}{\partial a_i^{(m)}} \frac{\partial a_i^{(m)}}{\partial z_i^{(m)}}$$

$$= W_{ji}^{(m+1)} \varphi'(z_i^{(m)})$$

因此：

$$\delta_i^{(m)} = \left[\sum_{j=1}^{S_{m+1}} \delta_j^{(m+1)} W_{ji}^{(m+1)}\right] \varphi'(z_i^{(m)})$$

因此，可由 $\delta_i^{(l)}$ 逐层向前递推 $\delta_i^{(m)}$。

获得所有 $\delta_i^{(l)}$ 后，容易得：

$$\frac{\partial E}{\partial W_{ij}^{(m)}} = \delta_j^{(m)} \cdot a_i^{(m-1)}$$

$$\frac{\partial E}{\partial b_i^{(m)}} = \delta_i^{(m)}$$

④ 更新。

对所有 W, b, 有：

$$W^{(new)} = W + \Delta W, \quad b^{(new)} = b + \Delta b$$

其中：$\Delta W = -\alpha \frac{\partial E}{\partial W}$, $\Delta b = -\alpha \frac{\partial E}{\partial b}$

⑤ 回到②，将 $W^{(new)}$, $b^{(new)}$ 替换原有 W, b 直至收敛。

注意事项：

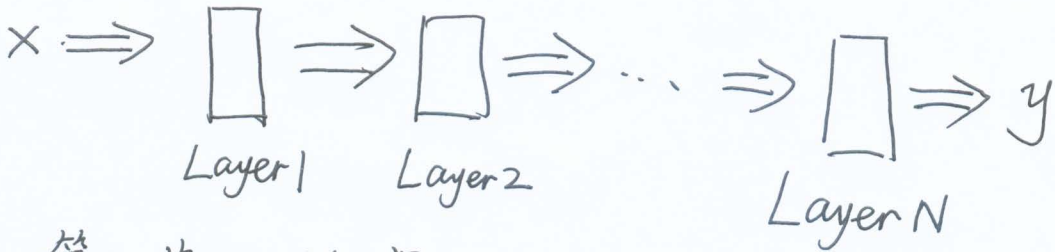
① 如果输入是 N个向量（Batch Size = N），则将每个向量都代入，都经历 1～4 步，将获得的 N个 ΔW, Δb 取平均，然后更新 W 和 b，回到 2步继续

② 停止原则（Stopping Criteria）：一般而言，将数据集分为 训练集（Training Set），验证集（Validation Set）和测试集（Testing Set）

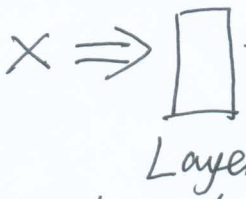
在训练集上训练、；在验证集上验证算法收敛性，若收敛退出训练；在测试集上测试

③ 初始化 W, b

可采用自编码器（Auto-encoder）；由 Hinton 2006年提出。

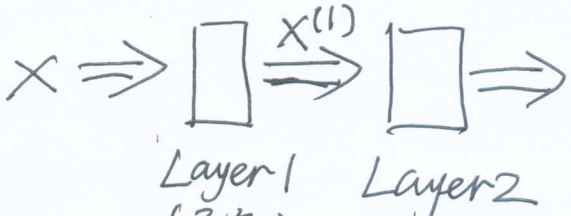


第一步，构建网络：



，用BP训练，获得Layer1初始参数

第二步，构建网络：



，用BP训练，获得Layer2初始参数

以此类推。获得每层初始参数。

最后对整个网络用BP微调（Tuning）