1 前馈神经网络：

考虑一个 $n+1$ 层的神经网络

$$Structure = \{L_0, L_1, \dots, L_n\}$$

其中 L_k 表示第 k 层网络结构，该层共记 N_k 个神经元； L_0 表示输入层，该层上的 N_0 个神经元代表输入数据的维度为 N_0 ；此外，作以下记号约定：

输入数据：

输入数据的规模为： $p \times d$，其中 $d = N_0$；

神经元的输入和输出：

$$x_k^{(i)}\text{: 第 } i \text{ 层第 } k \text{ 个神经元的输入；}$$

$$y_k^{(i)}\text{: 第 } i \text{ 层第 } k \text{ 个神经元的输出；}$$

这里： $i = 0,1,2,\dots,n$ ， $k = 1,2,\dots,N_i$；注意输入层的情况： $x_k^{(0)} = y_k^{(0)}$；

神经元对输入数据的权值与偏置：

$$w_{j,k}^{(i)}\text{: 第 } i \text{ 层第 } k \text{ 个神经元对第 } i-1 \text{ 层第 } j \text{ 个输出的权重；}$$

$$b_k^{(i)}\text{: 第 } i \text{ 层第 } k \text{ 个神经元的偏置；}$$

激活函数：

$$f_k^{(i)}\text{: 第 } i \text{ 层第 } k \text{ 个神经元的激活函数；}$$

一般情况下，每层激活函数是相同的；

学习率：

$$\eta = \left(\eta^{(1)}, \eta^{(2)}, \dots, \eta^{(n)}\right)$$

其中：

$$\eta^{(k)} = \begin{pmatrix} \eta_1^{(k)} \\ \eta_2^{(k)} \\ \vdots \\ \eta_{N_k}^{(k)} \end{pmatrix}$$

通常情况下，整个神经网络共用同样的学习率 η ，这里将每层每个神经元的学习率分开写，是为后面学习率的动态选择内容做铺垫。

1.1 前向传播：

考虑第 i 层第 k 个神经元上的数据传播情况：

输入：

$$x_k^{(i)} = \sum_{j=1}^{N_{i-1}} y_j^{(i-1)} \cdot w_{j.k}^{(i)} + b_k^{(i)}$$

输出为：

$$y_k^{(i)} = f_k^{(i)}\left(x_k^{(i)}\right)$$

其中：

$$i = 1,2, \dots, n$$

$$k = 1,2, \dots, N_i$$

记；

$$w_k^{(i)} = \begin{pmatrix} w_{1,k}^{(i)} \\ w_{2,k}^{(i)} \\ \vdots \\ w_{N_{i-1},k}^{(i)} \end{pmatrix}, \qquad N_{i-1} \times 1$$

$$x^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, \dots, x_{N_i}^{(i)}\right), \qquad 1 \times N_i$$

$$y^{(i)} = \left(y_1^{(i)}, y_2^{(i)}, \dots, y_{N_i}^{(i)}\right), \qquad 1 \times N_i$$

注意，这里将数据写为行向量而非常见的列向量的原因，是为了方便将后续的公式推广到输入数据为 $p \times d$ 的形式。

则可以得到对应的矩阵形式：

$$x_k^{(i)} = y^{(i-1)} \cdot w_k^{(i)} + b_k^{(i)}, \qquad 1 \times 1$$

$$y_k^{(i)} = f_k^{(i)}\left(x_k^{(i)}\right), \qquad 1 \times 1$$

若记：

$$w^{(i)} = \left(w_1^{(i)}, w_2^{(i)}, \ldots, w_{N_i}^{(i)}\right), \qquad N_{i-1} \times N_i$$

$$b^{(i)} = \left(b_1^{(i)}, b_2^{(i)}, \ldots, b_{N_i}^{(i)}\right), \qquad 1 \times N_i$$

$$F^{(i)}(\cdot) = \begin{pmatrix} f_1^{(i)}(\cdot) \\ f_2^{(i)}(\cdot) \\ \vdots \\ f_{N_i}^{(i)}(\cdot) \end{pmatrix}, \qquad 1 \times N_i$$

则进一步地有：

$$x^{(i)} = y^{(i-1)} \cdot w^{(i)} + b^{(i)}, \qquad 1 \times N_i \tag{1}$$

$$y^{(i)} = F^{(i)}\left(x^{(i)}\right), \qquad 1 \times N_i \tag{2}$$

这里， $i = 1,2, \ldots, n$

1.2 误差计算

神经网络的前向传播过程实际上是输入数据依据公式 (1)(2) 逐层递推的过程。当前向传播过程到达 $i = n$ 时，数据来到输出层；若记第 n 层第 k 个神经元的输出损失函数为：

$$l_k\left(y_k^{(n)}, y_k^*\right)$$

其中：$y^* = (y_1^*, y_2^*, \ldots, y_m^*)$ 是数据的 m 维 label，同时应当有 $m = N_n$；通常在神经网络输出数据为一维的时候，有：

$$e = l_1\left(y_1^{(n)}, y_1^*\right)$$

但某些如多分类的情况下，输出数据是多维度的,此时记某个维度上的误差为

$$e_k = l_k\left(y_k^{(n)}, y_k^*\right)$$

另记总的误差为：

$$E = g(e_1, e_2, \dots, e_m) \tag{3}$$

一般情况下 g 是一个线性函数，例如当我们考虑 MSE 误差时：

$$E = \frac{1}{2}\sum_{j=1}^{m}\left(y_k^{(n)} - y_k^*\right)^2$$

这里的线性关系也隐含着多个维度的输出之间相互独立不相关。之所以提出公式（3)这一更普遍的形式,是为后面解释更加广义的隐藏层误差反向传播做铺垫。

1.3 反向传播

考虑简单的损失函数形式：

$$E = \sum_{j=1}^{N_n} e_k \tag{4}$$

输出层：

考虑输出层前一层到输出层的正向传播过程：

$$e_k = l_k\left(y_k^{(n)}, y_k^*\right)$$

$$y_k^{(n)} = f_k^{(n)}\left(x_k^{(n)}\right)$$

$$x_k^{(n)} = y^{(n-1)} \cdot w_k^{(n)} + b_k^{(n)}$$

易得：

$$\frac{\partial e_k}{\partial w_k^{(n)}} = \frac{\partial l_k\left(y_k^{(n)}, y_k^*\right)}{\partial y_k^{(n)}} \cdot \frac{\partial f_k^{(n)}\left(x_k^{(n)}\right)}{\partial x_k^{(n)}} \cdot \left(y^{(n-1)}\right)^T, \qquad N_n \times 1$$

$$\frac{\partial e_k}{\partial b_k^{(n)}} = \frac{\partial l_k\left(y_k^{(n)}, y_k^*\right)}{\partial y_k^{(n)}} \cdot \frac{\partial f_k^{(n)}\left(x_k^{(n)}\right)}{\partial x_k^{(n)}}, \qquad 1 \times 1$$

进一步地考虑多维度输出的情况，记：

$$\frac{\partial E}{\partial w^{(n)}} = \left(\frac{\partial E}{\partial w_1^{(n)}}, \frac{\partial E}{\partial w_2^{(n)}}, \dots, \frac{\partial E}{\partial w_{N_n}^{(n)}}\right), \qquad N_{n-1} \times N_n$$

$$\frac{\partial E}{\partial w^{(n)}} = \left(\frac{\partial E}{\partial b_1^{(n)}}, \frac{\partial E}{\partial b_2^{(n)}}, \dots, \frac{\partial E}{\partial b_{N_n}^{(n)}}\right), \qquad 1 \times N_n$$

在考虑公式（4）的特殊情形时，有：

$$\frac{\partial E}{\partial w^{(n)}} = \left(\frac{\partial e_1^{(n)}}{\partial w_1^{(n)}}, \frac{\partial e_2^{(n)}}{\partial w_2^{(n)}}, \dots, \frac{\partial e_{N_n}^{(n)}}{\partial w_{N_n}^{(n)}}\right), \qquad N_{n-1} \times N_n$$

$$\frac{\partial E}{\partial b^{(n)}} = \left(\frac{\partial e_1^{(n)}}{\partial b_1^{(n)}}, \frac{\partial e_2^{(n)}}{\partial b_2^{(n)}}, \dots, \frac{\partial e_{N_n}^{(n)}}{\partial b_{N_n}^{(n)}}\right), \qquad 1 \times N_n$$

更新规则为：

$$w_k^{(n)} = w_k^{(n)} - \eta_k^{(n)} \cdot \frac{\partial e_k^{(n)}}{\partial w_k^{(n)}}, \qquad N_{n-1} \times 1$$

$$b_k^{(n)} = b_k^{(n)} - \eta_k^{(n)} \cdot \frac{\partial e_k^{(n)}}{\partial b_k^{(n)}}, \qquad 1 \times 1$$

为了以矩阵形式表述算法，做出以下符号约定：

$$\nabla E = \begin{pmatrix} \dfrac{\partial E}{\partial y_1^{(n)}} \\ \dfrac{\partial E}{\partial y_2^{(n)}} \\ \vdots \\ \dfrac{\partial E}{\partial y_{N_n}^{(n)}} \end{pmatrix}, \qquad N_n \times 1$$

$$\sigma'\left(x^{(n)}\right) = \begin{pmatrix} \dfrac{\partial f_1^{(n)}\left(x_1^{(n)}\right)}{\partial x_1^{(n)}} \\ \dfrac{\partial f_2^{(n)}\left(x_2^{(n)}\right)}{\partial x_2^{(n)}} \\ \vdots \\ \dfrac{\partial f_{N_n}^{(n)}\left(x_{N_n}^{(n)}\right)}{\partial x_{N_n}^{(n)}} \end{pmatrix}, \qquad N_n \times 1$$

定义：

$$\delta^{(n)} = \nabla E \odot \sigma'\left(x^{(n)}\right), \qquad N_n \times 1 \tag{5}$$

其中⊙ 表示点乘，即对应元素相乘；于是，可以得到：

$$\frac{\partial E}{\partial b^{(n)}} = \left(\delta^{(n)}\right)^T, \qquad 1 \times N_n \tag{6}$$

$$\frac{\partial E}{\partial w^{(n)}} = \left(y^{(n-1)}\right)^T \cdot \delta^{(n)}, \qquad N_{n-1} \times N_n \tag{7}$$

更新规则为：

$$b^{(n)} = b^{(n)} - \left(\eta^{(n)}\right)^T \odot \frac{\partial E}{\partial b^{(n)}}, \qquad 1 \times N_n \tag{8}$$

$$w^{(n)} = w^{(n)} - \begin{pmatrix} \left(\eta^{(n)}\right)^T \\ \left(\eta^{(n)}\right)^T \\ \vdots \\ \left(\eta^{(n)}\right)^T \end{pmatrix} \odot \frac{\partial E}{\partial w^{(n)}}, \qquad N_{n-1} \times N_n \tag{9}$$

隐藏层上：

首先考虑神经网络后 3 层的正向传播情况：

$$e_k = l_k\left(y_k^{(n)}, y_k^*\right)$$

$$y_k^{(n)} = f_k^{(n)}\left(x_k^{(n)}\right)$$

$$x_k^{(n)} = y^{(n-1)} \cdot w_k^{(n)} + b_k^{(n)}$$

$$y_p^{(n-1)} = f_p^{(n-1)}\left(x_p^{(n-1)}\right)$$

$$x_p^{(n-1)} = y^{(n-2)} \cdot w_p^{(n-1)} + b_p^{(n-1)}$$

注意到上述第三个公式涉及到 $y^{(n-1)}$，而该公式中取不同的 k 项，$y^{(n-1)}$ 均和 $w_p^{(n-1)}$ 有关联，

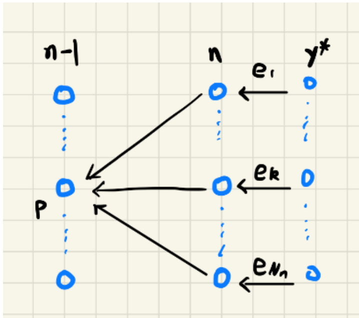

图 ：两层神经元间的误差传递

即第 $n-1$ 层第 p 个神经元上的误差与第 n 层上所有误差有关；当损失函数具有公式（4）的形式时，有：

$$\frac{\partial E}{\partial b_p^{(n-1)}} = \sum_{k=1}^{N_n} \frac{\partial e_k}{\partial y_k^{(n)}} \cdot \frac{\partial y_k^{(n)}}{\partial x_k^{(n)}} \cdot \frac{\partial x_k^{(n)}}{\partial y_p^{(n-1)}} \cdot \frac{\partial y_p^{(n-1)}}{\partial b_p^{(n-1)}}$$

$$\frac{\partial E}{\partial w_{l,p}^{(n-1)}} = \sum_{k=1}^{N_n} \frac{\partial e_k}{\partial y_k^{(n)}} \cdot \frac{\partial y_k^{(n)}}{\partial x_k^{(n)}} \cdot \frac{\partial x_k^{(n)}}{\partial y_p^{(n-1)}} \cdot \frac{\partial y_p^{(n-1)}}{\partial w_{l,p}^{(n-1)}}$$

写作矩阵形式，有：

$$\delta^{(n-1)} = \delta^{(n)} \times (w_k)^T \odot \sigma'\left(x^{(n-1)}\right) \tag{10}$$

$$\frac{\partial E}{\partial b^{(n-1)}} = \left(\delta^{(n-1)}\right)^T, \qquad 1 \times N_{n-1} \tag{11}$$

$$\frac{\partial E}{\partial w^{(n-1)}} = \left(y^{(n-2)}\right)^T \cdot \delta^{(n-1)}, \qquad N_{n-2} \times N_{n-1} \tag{12}$$

注意到，式（10）中 $\delta^{(n)} \times (w_k)^T$ 项意味着反向传播时，误差 $\delta^{(n-1)}$ 由 $\delta^{(n)}$ 经 w_k 加权得到。综合（5）-（12）,可以得到神经网络反向传播的形式为：

$$\delta^{(n)} = \nabla E \odot \sigma'\left(x^{(n)}\right)$$

$$\delta^{(k)} = \delta^{(k+1)} \times (w_{k+1})^T \odot \sigma'\left(x^{(k)}\right)$$

其中：$k = 1,2, \dots, n-1$

误差更新为：

$$\frac{\partial E}{\partial b^{(k)}} = \left(\delta^{(k)}\right)^T$$

$$\frac{\partial E}{\partial w^{(k)}} = \left(y^{(k-1)}\right)^T \cdot \delta^{(k)}$$

$$b^{(k)} = b^{(k)} - \left(\eta^{(k)}\right)^T \odot \frac{\partial E}{\partial b^{(k)}}$$

$$w^{(k)} = w^{(k)} - \begin{pmatrix} \left(\eta^{(k)}\right)^T \\ \left(\eta^{(k)}\right)^T \\ \vdots \\ \left(\eta^{(k)}\right)^T \end{pmatrix} \odot \frac{\partial E}{\partial w^{(k)}}$$