机器学习（西瓜书）

注　解

（第 5 章　神经网络）

https://blog.csdn.net/jbb0523

前言

经常听人说南大周老师所著的《机器学习》（以下统称为西瓜书）是一本入门教材，是一本科普性质的教科书。在该书第十次印刷之际，周老师在“如何使用本书”中也提到“这是一本入门级教科书”。然而，本人读起来却感觉该书远不止“科普”“入门”那么简单，书中的很多公式需要思考良久方能推导，很多概念需要反复咀嚼才能消化。边读边想着是不是应该将自己学习时遇到的一些知识难点的解析分享出来，以帮助更多的人入门。自己的确也随手做过一些笔记，但由于怀疑这仅是自己的个别现象，毕竟读书期间，思考更多的是如何使用单片机、DSP、ARM、FGPA 等，而这些基本是不需要推导任何公式的，因此作罢。偶然间在周老师的新浪微博看到如下对话：

此时方知，可能“读不懂”并不是个别现象。因此决定写一本“西瓜书注解”或者称为“西瓜书读书笔记”，对自己研读西瓜书时遇到的“台阶”进行解释和推导，以帮助更多的人能够更快地进入到这个领域。另外，近期越来越强地意识到，扎扎实实地推导一些基础算法的公式，无论是对于理解算法本身机理还是进行学术研究，都是非常有必要的。

自己会根据个人学习进度和研究需要按章发布，不知道能不能坚持写完，加油！

毕竟自己也是一名初学者，所以可能一些概念解释并不完整、一些公式推导并不优美，甚至会存在错误，这是不可避免的，不接受漫骂，但欢迎将问题反馈给我，共同学习进步！

（网盘链接：https://pan.baidu.com/s/1QtEiNnk8jMzmbs0KPBN-_w）

第 5 章目录

第 5 章 神经网络

神经网络→深度神经网络→深度学习→人工智能，简直已经成为这个时代的符号之一。大胆猜测一下，会不会有人买西瓜书就是为了看一下本章的神经网络？

5.1 神经元模型

本节内容通俗易懂，几乎不需要什么注解。

本节第 2 段提到“阈[yù]值”(threshold)的概念时，特意边注到“亦称 bias，注意不是‘阀[fá]值’，虽然其含义的确类似于‘阀门’”；百度一下“阈值 阀值”，你就知道：之所有会有此注释，是因为有很多人不认识“阈”这个字，将其凭感觉读成“阀”，错的人越来越多，慢慢大家也就习惯了，后来官方也认可了“阀值”这个词汇（就像“铁骑[jì]”一样，现在被纠正为“铁骑[qí]”了）；坊间笑谈：“阀值是阈值的文盲版，阈值是阀值的学究版”。

神经网络(Neural Networks, NN)又称人工神经网络(Artificial NN, ANN)，最基本的成分是神经元(neuron)。图 5.1 中提到 M-P 神经元模型，这当然是 McCulloch 和 Pitts 的首字母简写；可能联想到的概念还有 BP 神经网络和 Hopfield 神经网络，两个概念分别在 5.3 节第 1 段(P101)和本章最后的小故事(P120)中提到。

5.2 感知机与多层网络

本节介绍感知机算法，进而引出多层网络的概念。有关感知机可参考李航《统计学习方法》的第 2 章（该书于 2019 年 5 月已出第二版），有更为详细的介绍。

1、图 5.3 的解释

注意两点：(I)函数f是图 5.2(a)中的阶跃函数$\mathrm{sgn}(x)$；(II)这里要实现逻辑与、或、非运算，因此输入x_1和x_2只能是 0 和 1；如此就可以理解从 98 页到 99 页的与、或、非例子了：

(1)“与”（$y=x_1 \wedge x_2$）

x_1	x_2	$h=1\cdot x_1+1\cdot x_2-2$	$y=f(h)$
0	0	-2	0
0	1	-1	0
1	0	-1	0
1	1	0	1

(2)“或”（$y=x_1 \vee x_2$）

x_1	x_2	$h=1\cdot x_1+1\cdot x_2-0.5$	$y=f(h)$
0	0	-0.5	0
0	1	0.5	1
1	0	0.5	1
1	1	1.5	1

(3)“非”（$y=\neg x_1$，即与x_2无关）

x_1	x_2	$h=-0.6\cdot x_1+0\cdot x_2+0.5$	$y=f(h)$
0	0	0.5	1
0	1	0.5	1
1	0	-0.1	0
1	1	-0.1	0

2、式(5.1)和式(5.2)的解释

这里主要就是理解为什么根据式(5.1)和式(5.2)调整权重w_i可以减小预测错误程度或纠正预测错误。注意这里讨论的仍是预测输出为 0 或 1 的二分类，边注中已解释说x_i是$\boldsymbol{x}$对应于第i个输入神经元的分量：

(1)当$\hat{y}=y$时（即预测正确），式(5.2)的$\Delta w_i=0$，所以对应的式(5.1)中的w_i没有影响；

(2)当$\hat{y}\neq y$时（即预测错误），式(5.2)的$\Delta w_i\neq 0$，对应的式(5.1)中的w_i将进行调整：

(a)若$y=1$，预测错误时$\hat{y}=0$，则$\Delta w_i=\eta x_i$；此时原先的预测值$w_i x_i<0$，根据式(5.1)调整权重后的预测值为$(w_i+\Delta w_i)x_i=(w_i+\eta x_i)x_i=w_i x_i+\eta x_i^2>w_i x_i$，即预测值将变大，更有可能大于 0（因为$y=1$，预测值大于 0 时阶跃函数输出 1）；

(b)若$y=0$，预测错误时$\hat{y}=1$，则$\Delta w_i=-\eta x_i$；此时原先的预测值$w_i x_i>0$，根据式(5.1)调整权重后的预测值为$(w_i+\Delta w_i)x_i=(w_i-\eta x_i)x_i=w_i x_i-\eta x_i^2<w_i x_i$，即预测值将变小，更有可能小于 0（因为$y=0$，预测值小于 0 时阶跃函数输出 0）；

综上所述，当预测错误时，根据式(5.1)和式(5.2)调整权重w_i将使错误程度减小。

3、图 5.4 的解释

注意这里讨论的仍是逻辑与、或、非运算，输入非 0 即 1；设每个样本为(x_1,x_2,y)，其中x_1和x_2为样本特征，y为样本类别；对于每幅子图，其所对应的数据集分别为：

(a) $D=\{(0,0,0),(0,1,0),(1,0,0),(1,1,1)\}$

(b) $D=\{(0,0,0),(0,1,1),(1,0,1),(1,1,1)\}$

(c) $D=\{(0,0,1),(0,1,1),(1,0,0),(1,1,0)\}$（即输出与特征$x_2$无关）

(d) $D=\{(0,0,0),(0,1,1),(1,0,1),(1,1,0)\}$

可以发现，正如图 5.4 所示，图(a)(b)(c)的样本可以用一条直线将其按类别分开，而图(d)的样本则不能使用一条直线将其按类别分开。

4、图 5.5 的解释

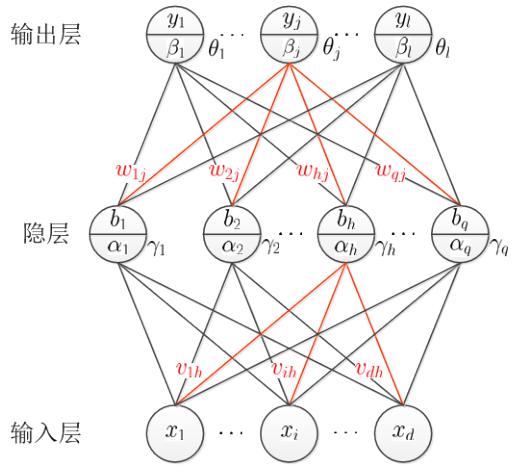
图 5.5 实现“异或”计算的过程如下：

x_1	x_2	$h_1=f(x_1+x_2-0.5)$	$h_2=f(-x_1-x_2+1.5)$	$y=f(h_1+h_2-1.5)$
0	0	$f(-0.5)=0$	$f(1.5)=1$	$f(-0.5)=0$
0	1	$f(0.5)=1$	$f(0.5)=1$	$f(0.5)=1$
1	0	$f(0.5)=1$	$f(0.5)=1$	$f(0.5)=1$
1	1	$f(1.5)=1$	$f(-0.5)=0$	$f(-0.5)=0$

5.3 误差逆传播算法

误差逆传播(BP)算法是训练多层神经网络的代表性算法。本节开篇第一段最后一句话提到术语“BP 网络”，一般是指用 BP 算法训练的多层前馈神经网络。本节以图 5.7 所示的单

隐层神经网络为例详细推导了 BP 算法。接下来，先将本节用到的符号进行如下的梳理。



输出层第j个神经元输出：$y_j = f(\beta_j - \theta_j)$, θ_j为阈值
输出层第j个神经元输入：$\beta_j = \sum_{h=1}^{q} w_{hj} b_h$
输出层神经元个数：l

隐层第h个神经元到输出层第j个神经元的连接权：w_{hj}

隐层第h个神经元输出：$b_h = f(\alpha_h - \gamma_h)$, γ_h为阈值
隐层第h个神经元输入：$\alpha_h = \sum_{i=1}^{d} v_{ih} x_i$
隐层神经元个数：q

输入层第i个神经元到隐层第h个神经元的连接权：v_{ih}

输入层第i个神经元输入：x_i
输入层神经元个数：d

输入层神经元个数：d
输入层第i个神经元输入：x_i

输入层第i个神经元 到 隐层第h个神经元的连接权：v_{ih}

隐层神经元个数：q

隐层第h个神经元输入：$\alpha_h = \sum_{i=1}^{d} v_{ih} x_i$

隐层第h个神经元输出：$b_h = f(\alpha_h - \gamma_h)$，$\gamma_h$为阈值

隐层第h个神经元 到 输出层第j个神经元的连接权：w_{hj}

输出层神经元个数：l
输出层第j个神经元输入：$\beta_j = \sum_{h=1}^{q} w_{hj} b_h$
输出层第j个神经元输出：$\hat{y}_j = f(\beta_j - \theta_j)$，$\theta_j$为阈值

训练集$D = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_m, \boldsymbol{y}_m)\}$，其中第$k$个样本$(\boldsymbol{x}_k, \boldsymbol{y}_k)$
$\boldsymbol{x}_k = (x_1^k, x_2^k, \ldots, x_d^k) \in \mathbb{R}^d$，$\boldsymbol{y}_k = (y_1^k, y_2^k, \ldots, y_l^k) \in \mathbb{R}^l$

接下来式(5.3)到式(5.15)推导的是标准 BP 算法，每次仅针对一个训练样例更新连接权和阈值，读取训练集一遍称为进行了一轮(one round，亦称 one epoch)学习。

待确定参数：输入层到隐层的$d \times q$个权值v_{ih} $(1 \le i \le d,\ 1 \le h \le q)$
隐层到输出层的$q \times l$个权值w_{hj} $(1 \le h \le q,\ 1 \le j \le l)$
隐层的q个神经元的阈值γ_h $(1 \le h \le q)$
输出层的l个神经元的阈值θ_j $(1 \le j \le l)$
另外，隐层的神经元个数q以及训练轮数 epoch 也都是需要确定的超参数

BP 算法首先在$(0, 1)$范围内随机初始化网络中所有连接权v_{ih}和w_{hj}，以及阈值γ_h和θ_j，在每一轮迭代中采用梯度下降策略逐步对权重进行更新，而标准 BP 算法在每一轮迭代中是依次仅针对一个训练样例更新连接权和阈值。

1、式(5.3)的解释

该式即网络输出层第j个神经元的输出表达式，其中$\beta_j = \sum_{h=1}^{q} w_{hj} b_h$是输出层的第$j$个神经元输入，$\theta_j$是相应的阈值，$f$是神经元的激活函数。通过该式，可以根据当前连接权和阈值计算出训练样本$(\boldsymbol{x}_k, \boldsymbol{y}_k)$的网络输出$\hat{\boldsymbol{y}}_k = (\hat{y}_1^k, \hat{y}_2^k, \ldots, \hat{y}_l^k)$。

2、式(5.4)的解释

该式就是针对训练样本$(\boldsymbol{x}_k, \boldsymbol{y}_k)$，求出网络输出$\hat{\boldsymbol{y}}_k = (\hat{y}_1^k, \hat{y}_2^k, \ldots, \hat{y}_l^k)$与真实值$\boldsymbol{y}_k$之间的均方误差（对应元素差的平方，再求和）；这里的二分之一是为了求导后可以使系数为 1，因为平方求导会多出来一个 2，正好与二分之一抵消掉。

3、式(5.7)的解释

均方误差E_k与权重w_{hj}的关系如下：

$$E_k = \frac{1}{2}\sum_{j=1}^{l}(\hat{y}_j^k - y_j^k)^2,\ \hat{y}_j^k = f(\beta_j - \theta_j),\ \beta_j = \sum_{h=1}^{q} w_{hj} b_h$$

4、式(5.8)的推导

由于$\beta_j = \sum_{h=1}^{q} w_{hj} b_h$，对$w_{hj}$求导显然只有第$h$项$w_{hj} b_h$包含$w_{hj}$，因此求导等于$b_h$。

5、式(5.9)的推导

对于图 5.2(b)中的 Sigmoid 函数，为表示方便，令$f(x) = \frac{1}{1+e^{-x}}$，求导，得

$$f'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1}{1+e^{-x}}\frac{e^{-x}}{1+e^{-x}} = \frac{1}{1+e^{-x}}\left(1 - \frac{1}{1+e^{-x}}\right) = f(x)(1-f(x))$$

上式求导过程中，使用了商的求导公式$(\frac{u}{v})' = \frac{u'v - uv'}{v^2}$；其中，对应到上式中，$v = 1 + e^{-x}$，$u = 1$，二者的导数分别为$u' = 0$，$v' = -e^{-x}$。

6、式(5.10)的推导

式中两部分的求导过程分别如下：

$$\frac{\partial E_k}{\partial \hat{y}_j^k} = \frac{\partial \frac{1}{2}\sum_{j=1}^{l}(\hat{y}_j^k - y_j^k)^2}{\partial \hat{y}_j^k} = (\hat{y}_j^k - y_j^k)$$

$$\frac{\partial \hat{y}_j^k}{\partial \beta_j} = \hat{y}_j^k(1 - \hat{y}_j^k)$$

第二部分是因为根据式(5.9)的结论，将式(5.3)中$\hat{y}_j^k = f(\beta_j - \theta_j)$的表达式代入，即

$$\frac{\partial \hat{y}_j^k}{\partial \beta_j} = \frac{\partial f(\beta_j - \theta_j)}{\partial \beta_j} = \frac{\partial f(\beta_j - \theta_j)}{\partial (\beta_j - \theta_j)}\frac{\partial (\beta_j - \theta_j)}{\partial \beta_j} = f(\beta_j - \theta_j)(1 - f(\beta_j - \theta_j)) = \hat{y}_j^k(1 - \hat{y}_j^k)$$

注意式(5.10)中第 2 个等号使用的是$(\hat{y}_j^k - y_j^k)$，第 3 个等号中则变成了$(y_j^k - \hat{y}_j^k)$，这里将第 2 个等号中的负号吸收了。

7、式(5.12)的推导

根据链式求导法则$\frac{\partial E_k}{\partial \theta_j} = \frac{\partial E_k}{\partial \hat{y}_j^k}\frac{\partial \hat{y}_j^k}{\partial \theta_j}$，其中$\frac{\partial \hat{y}_j^k}{\partial \theta_j} = \frac{\partial f(\beta_j - \theta_j)}{\partial \theta_j} = \frac{\partial f(\beta_j - \theta_j)}{\partial (\beta_j - \theta_j)}\frac{\partial (\beta_j - \theta_j)}{\partial \theta_j}$，而根据式

(5.9)的结论$\frac{\partial f(\beta_j-\theta_j)}{\partial(\beta_j-\theta_j)}=f(\beta_j-\theta_j)(1-f(\beta_j-\theta_j))$，并且易知$\frac{\partial(\beta_j-\theta_j)}{\partial\theta_j}=-1$，因此

$$\frac{\partial\hat{y}_j^k}{\partial\theta_j}=-f(\beta_j-\theta_j)(1-f(\beta_j-\theta_j))=-\hat{y}_j^k(1-\hat{y}_j^k)$$

而$\frac{\partial E_k}{\partial\hat{y}_j^k}=\frac{\partial\frac{1}{2}\sum_{j=1}^{l}(\hat{y}_j^k-y_j^k)^2}{\partial\hat{y}_j^k}=(\hat{y}_j^k-y_j^k)$，综合两部分求导结果，结合式(5.10)，得

$$\frac{\partial E_k}{\partial\theta_j}=-(\hat{y}_j^k-y_j^k)\hat{y}_j^k(1-\hat{y}_j^k)=\hat{y}_j^k(1-\hat{y}_j^k)(y_j^k-\hat{y}_j^k)=g_j$$

因此，类似于式(5.6)，$\Delta\theta_j=-\eta\frac{\partial E_k}{\partial\theta_j}=-\eta g_j$ 。

8、式(5.15)的推导

由于式(5.13)和式(5.14)都包含式(5.15)定义的变量e_h，因此先推导式(5.15)。

为了求$e_h=-\frac{\partial E_k}{\partial\alpha_h}$，先列出$E_k$和$\alpha_h$的联系：

$$①E_k=\frac{1}{2}\sum_{j=1}^{l}(\hat{y}_j^k-y_j^k)^2,\ ②\hat{y}_j^k=f(\beta_j-\theta_j),$$

$$③\beta_j=\sum_{h=1}^{q}w_{hj}b_h,\ ④b_h=f(\alpha_h-\gamma_h),$$

为了避免符号混淆，在接下来的推导中将$\beta_j=\sum_{h=1}^{q}w_{hj}b_h$中求和变量$h$换为$p$，而$h$特指$\alpha_h$中的$h$，即$\beta_j=\sum_{p=1}^{q}w_{pj}b_p$。

由于$\beta_j=\sum_{p=1}^{q}w_{pj}b_p$，即隐层第$h$个神经元的输出$b_h$与输出层任意神经元的输入$\beta_j$均有关，因此 $\frac{\partial E_k}{\partial b_h}=\frac{\partial E_k}{\partial\beta_1}\frac{\partial\beta_1}{\partial b_h}+\frac{\partial E_k}{\partial\beta_2}\frac{\partial\beta_2}{\partial b_h}+\ldots+\frac{\partial E_k}{\partial\beta_l}\frac{\partial\beta_l}{\partial b_h}=\sum_{j=1}^{l}\frac{\partial E_k}{\partial\beta_j}\frac{\partial\beta_j}{\partial b_h}$

进而，可得 $e_h=-\frac{\partial E_k}{\partial\alpha_h}=-\frac{\partial E_k}{\partial b_h}\frac{\partial b_h}{\partial\alpha_h}=-\sum_{j=1}^{l}\frac{\partial E_k}{\partial\beta_j}\frac{\partial\beta_j}{\partial b_h}\frac{\partial b_h}{\partial\alpha_h}$

根据式(5.10)定义可知$g_j=-\frac{\partial E_k}{\partial\beta_j}$，再根据式(5.9)的结论可知$\frac{\partial b_h}{\partial\alpha_h}=b_h(1-b_h)$，而

$$\frac{\partial\beta_j}{\partial b_h}=\frac{\partial\sum_{p=1}^{q}w_{pj}b_p}{\partial b_h}=w_{hj}$$

综上所述

$$e_h=-\sum_{j=1}^{l}\frac{\partial E_k}{\partial\beta_j}\frac{\partial\beta_j}{\partial b_h}\frac{\partial b_h}{\partial\alpha_h}=\sum_{j=1}^{l}g_jw_{hj}b_h(1-b_h)=b_h(1-b_h)\sum_{j=1}^{l}g_jw_{hj}$$

最后一个等号是由于b_h与求和变量j无关，故可以拿到求和号外面，到此推导完毕。

变量e_h是本节定义的第二个变量，第一个是式(5.10)的g_j；其中$g_j=-\frac{\partial E_k}{\partial\beta_j}$，即均方误差$E_k$对输出层第$j$个神经元的输入$\beta_j$的导数（不考虑负号）；而$e_h=-\frac{\partial E_k}{\partial\alpha_h}$，即均方误差$E_k$对隐层第$h$个神经元的输入$\alpha_h$的导数（不考虑负号）；因为图 5.7 的神经网络只有两层功能神经元，因此这里共定义了g_j和e_h，若有多个隐层，则为了表达方便需定义更多的变量，而定义的这些变量即体现了“误差逆传播”的含义，即误差E_k从后往前依次对各层功能神经元的输入求导。

9、式(5.13)的推导

类似于式(5.6)，并结合式(5.15)的定义$e_h = -\frac{\partial E_k}{\partial \alpha_h}$，得

$$\Delta v_{ih} = -\eta \frac{\partial E_k}{\partial v_{ih}} = -\eta \frac{\partial E_k}{\partial \alpha_h} \frac{\partial \alpha_h}{\partial v_{ih}} = \eta e_h x_i$$

其中，由$\alpha_h = \sum_{i=1}^{d} v_{ih} x_i$可得$\frac{\partial \alpha_h}{\partial v_{ih}} = \frac{\partial \sum_{i=1}^{d} v_{ih} x_i}{\partial v_{ih}} = x_i$，到此推导完毕！

10、式(5.14)的推导

类似于式(5.6)，得 $\Delta \gamma_h = -\eta \frac{\partial E_k}{\partial \gamma_h} = -\eta \frac{\partial E_k}{\partial b_h} \frac{\partial b_h}{\partial \gamma_h}$

而$b_h = f(\alpha_h - \gamma_h)$，故 $\frac{\partial b_h}{\partial \gamma_h} = \frac{\partial f(\alpha_h - \gamma_h)}{\partial \gamma_h} = \frac{\partial f(\alpha_h - \gamma_h)}{\partial (\alpha_h - \gamma_h)} \frac{\partial (\alpha_h - \gamma_h)}{\partial \gamma_h}$

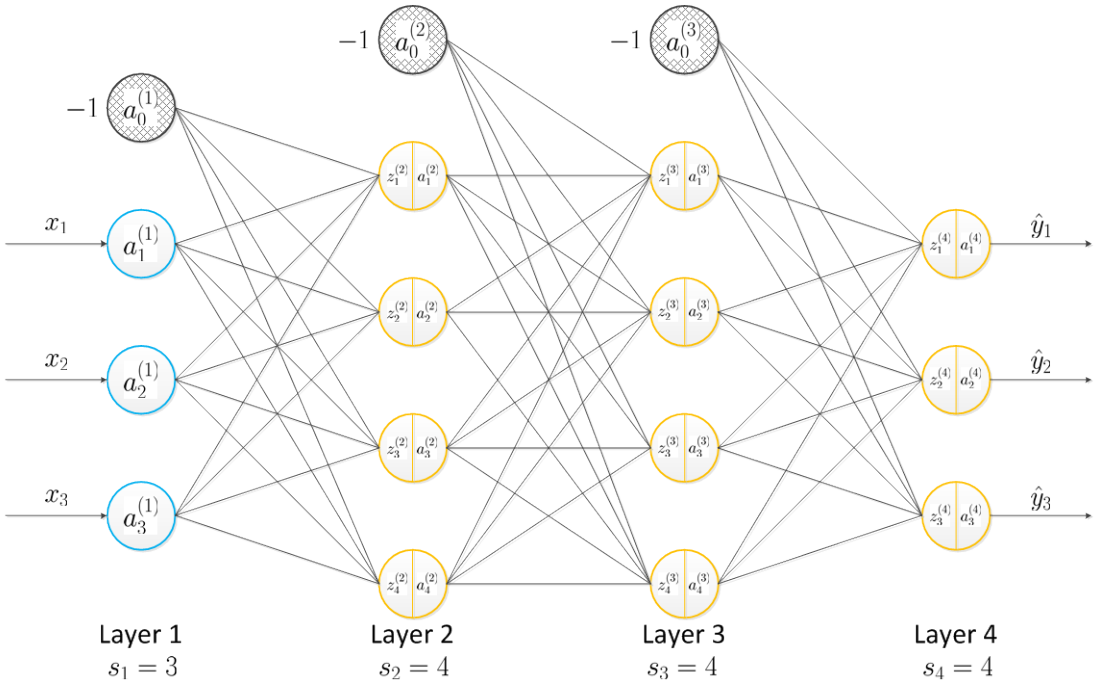
而 $\frac{\partial b_h}{\partial \alpha_h} = \frac{\partial f(\alpha_h - \gamma_h)}{\partial \alpha_h} = \frac{\partial f(\alpha_h - \gamma_h)}{\partial (\alpha_h - \gamma_h)} \frac{\partial (\alpha_h - \gamma_h)}{\partial \alpha_h}$

其中$\frac{\partial (\alpha_h - \gamma_h)}{\partial \gamma_h} = -1$，$\frac{\partial (\alpha_h - \gamma_h)}{\partial \alpha_h} = 1$，即$\frac{\partial b_h}{\partial \gamma_h} = -\frac{\partial b_h}{\partial \alpha_h}$，因此

$$\Delta \gamma_h = -\eta \frac{\partial E_k}{\partial \gamma_h} = -\eta \frac{\partial E_k}{\partial b_h} \frac{\partial b_h}{\partial \gamma_h} = \eta \frac{\partial E_k}{\partial b_h} \frac{\partial b_h}{\partial \alpha_h}$$

结合式(5.15)的定义$e_h = -\frac{\partial E_k}{\partial \alpha_h}$，因此$\Delta \gamma_h = \eta \frac{\partial E_k}{\partial b_h} \frac{\partial b_h}{\partial \alpha_h} = -\eta e_h$，到此推导完毕！

11、多隐层神经网络的 BP 算法推导



为了更进一步理解 BP 算法的推导，尤其是为将来推导深度神经网络的 BP 算法打下基础，接下来以包含两个隐层的神经网络（如上图所示）为例推导 BP 算法。

约定输入层为第 1 层，输入层神经元仅接受输入，不进行函数处理；隐层和输出层神经

元为功能神经元，对信号进行加工处理。除第 1 层外，第k层第i个神经元的输入为$z_i^{(k)}$，输出为$a_i^{(k)}$，其中$a_i^{(k)}=f\left(z_i^{(k)}\right)$，$f(\cdot)$为激活函数。第$k$层神经元个数为$s_k$；从第$k$层第$i$个神经元到第$l=k+1$层第$j$个神经元的连接权重记为$\theta_{ij}^{(kl)}$，第$k$层到第$l=k+1$层所有连接权重构成大小为$s_k\times s_l$矩阵$\boldsymbol{\Theta}^{(kl)}$：

$$\boldsymbol{\Theta}^{(12)}=\begin{bmatrix}\theta_{11}^{(12)} & \theta_{12}^{(12)} & \theta_{13}^{(12)} & \theta_{14}^{(12)}\\ \theta_{21}^{(12)} & \theta_{22}^{(12)} & \theta_{23}^{(12)} & \theta_{24}^{(12)}\\ \theta_{31}^{(12)} & \theta_{32}^{(12)} & \theta_{33}^{(12)} & \theta_{34}^{(12)}\end{bmatrix},\boldsymbol{\Theta}^{(23)}=\begin{bmatrix}\theta_{11}^{(23)} & \theta_{12}^{(23)} & \theta_{13}^{(23)} & \theta_{14}^{(23)}\\ \theta_{21}^{(23)} & \theta_{22}^{(23)} & \theta_{23}^{(23)} & \theta_{24}^{(23)}\\ \theta_{31}^{(23)} & \theta_{32}^{(23)} & \theta_{33}^{(23)} & \theta_{34}^{(23)}\\ \theta_{41}^{(23)} & \theta_{42}^{(23)} & \theta_{43}^{(23)} & \theta_{44}^{(23)}\end{bmatrix},\boldsymbol{\Theta}^{(34)}=\begin{bmatrix}\theta_{11}^{(34)} & \theta_{12}^{(34)} & \theta_{13}^{(34)}\\ \theta_{21}^{(34)} & \theta_{22}^{(34)} & \theta_{23}^{(34)}\\ \theta_{31}^{(34)} & \theta_{32}^{(34)} & \theta_{33}^{(34)}\\ \theta_{41}^{(34)} & \theta_{42}^{(34)} & \theta_{43}^{(34)}\end{bmatrix}$$

为了方便表示，除输出层外，每层增加取值为-1的第 0 个神经元，将第l层s_l个神经元的s_l个阈值统一表示为连接权重；记第k层到第$l=k+1$层包含阈值的所有连接权重构成的矩阵为$\tilde{\boldsymbol{\Theta}}^{(kl)}$，大小为$(s_k+1)\times s_l$：

$$\tilde{\boldsymbol{\Theta}}^{(12)}=\begin{bmatrix}\theta_{01}^{(12)} & \theta_{02}^{(12)} & \theta_{03}^{(12)} & \theta_{04}^{(12)}\\ \theta_{11}^{(12)} & \theta_{12}^{(12)} & \theta_{13}^{(12)} & \theta_{14}^{(12)}\\ \theta_{21}^{(12)} & \theta_{22}^{(12)} & \theta_{23}^{(12)} & \theta_{24}^{(12)}\\ \theta_{31}^{(12)} & \theta_{32}^{(12)} & \theta_{33}^{(12)} & \theta_{34}^{(12)}\end{bmatrix},\tilde{\boldsymbol{\Theta}}^{(23)}=\begin{bmatrix}\theta_{01}^{(23)} & \theta_{02}^{(23)} & \theta_{03}^{(23)} & \theta_{04}^{(23)}\\ \theta_{11}^{(23)} & \theta_{12}^{(23)} & \theta_{13}^{(23)} & \theta_{14}^{(23)}\\ \theta_{21}^{(23)} & \theta_{22}^{(23)} & \theta_{23}^{(23)} & \theta_{24}^{(23)}\\ \theta_{31}^{(23)} & \theta_{32}^{(23)} & \theta_{33}^{(23)} & \theta_{34}^{(23)}\\ \theta_{41}^{(23)} & \theta_{42}^{(23)} & \theta_{43}^{(23)} & \theta_{44}^{(23)}\end{bmatrix},\tilde{\boldsymbol{\Theta}}^{(34)}=\begin{bmatrix}\theta_{01}^{(34)} & \theta_{02}^{(34)} & \theta_{03}^{(34)}\\ \theta_{11}^{(34)} & \theta_{12}^{(34)} & \theta_{13}^{(34)}\\ \theta_{21}^{(34)} & \theta_{22}^{(34)} & \theta_{23}^{(34)}\\ \theta_{31}^{(34)} & \theta_{32}^{(34)} & \theta_{33}^{(34)}\\ \theta_{41}^{(34)} & \theta_{42}^{(34)} & \theta_{43}^{(34)}\end{bmatrix}$$

例如，第 3 层第 1 个神经元的输入为

$$z_1^{(3)}=\theta_{11}^{(23)}a_1^{(2)}+\theta_{21}^{(23)}a_2^{(2)}+\theta_{31}^{(23)}a_3^{(2)}+\theta_{41}^{(23)}a_4^{(2)}-\theta_{01}^{(23)}$$

各层神经元的输入和输出表示如下

输入：$\boldsymbol{x}=(x_1;x_2;x_3)$,

第 1 层：$\boldsymbol{a}^{(1)}=\left(a_1^{(1)};a_2^{(1)};a_3^{(1)}\right)=\boldsymbol{x}$

$$\tilde{\boldsymbol{a}}^{(1)}=\left(-1;a_1^{(1)};a_2^{(1)};a_3^{(1)}\right)=(-1;\boldsymbol{x})$$

第 2 层：$\boldsymbol{z}^{(2)}=\left(z_1^{(2)};z_2^{(2)};z_3^{(2)};z_4^{(2)}\right)=\left[\tilde{\boldsymbol{\Theta}}^{(12)}\right]^{\top}\tilde{\boldsymbol{a}}^{(1)}$

$$\boldsymbol{a}^{(2)}=\left(a_1^{(2)};a_2^{(2)};a_3^{(2)};a_4^{(2)}\right)=f\left(\boldsymbol{z}^{(2)}\right)$$

$$\tilde{\boldsymbol{a}}^{(2)}=\left(-1;a_1^{(2)};a_2^{(2)};a_3^{(2)};a_4^{(2)}\right)=\left(-1;\boldsymbol{a}^{(2)}\right)$$

第 3 层：$\boldsymbol{z}^{(3)}=\left(z_1^{(3)};z_2^{(3)};z_3^{(3)};z_4^{(3)}\right)=\left[\tilde{\boldsymbol{\Theta}}^{(23)}\right]^{\top}\tilde{\boldsymbol{a}}^{(2)}$

$$\boldsymbol{a}^{(3)}=\left(a_1^{(3)};a_2^{(3)};a_3^{(3)};a_4^{(3)}\right)=f\left(\boldsymbol{z}^{(3)}\right)$$

$$\tilde{\boldsymbol{a}}^{(3)}=\left(-1;a_1^{(3)};a_2^{(3)};a_3^{(3)};a_4^{(3)}\right)=\left(-1;\boldsymbol{a}^{(3)}\right)$$

第 4 层：$\boldsymbol{z}^{(4)}=\left(z_1^{(4)};z_2^{(4)};z_3^{(4)}\right)=\left[\tilde{\boldsymbol{\Theta}}^{(34)}\right]^{\top}\tilde{\boldsymbol{a}}^{(3)}$

$$\boldsymbol{a}^{(4)}=\left(a_1^{(4)};a_2^{(4)};a_3^{(4)}\right)=f\left(\boldsymbol{z}^{(4)}\right)$$

$$\hat{\boldsymbol{y}}=(\hat{y}_1;\hat{y}_2;\hat{y}_3)=\boldsymbol{a}^{(4)}$$

其中$\boldsymbol{v}=(v_1;v_2;\ldots;v_n)$表示$\boldsymbol{v}$为$n\times 1$的列向量，$f(\cdot)$为激活函数。

梯度下降法是将均方误差

$$E=\frac{1}{2}\left\|\hat{\boldsymbol{y}}-\boldsymbol{y}\right\|^2=\frac{1}{2}\left\|\boldsymbol{a}^{(4)}-\boldsymbol{y}\right\|^2=\frac{1}{2}\sum_{j=1}^{3}(a_j^{(4)}-y_j)^2$$

对$\tilde{\boldsymbol{\Theta}}^{(12)},\tilde{\boldsymbol{\Theta}}^{(23)},\tilde{\boldsymbol{\Theta}}^{(34)}$分别求导，求出$\Delta\tilde{\boldsymbol{\Theta}}^{(12)},\Delta\tilde{\boldsymbol{\Theta}}^{(23)},\Delta\tilde{\boldsymbol{\Theta}}^{(34)}$，然后按如下公式对连接权重更新：

$$\tilde{\boldsymbol{\Theta}}^{(12)}=\tilde{\boldsymbol{\Theta}}^{(12)}+\Delta\tilde{\boldsymbol{\Theta}}^{(12)},\ \tilde{\boldsymbol{\Theta}}^{(23)}=\tilde{\boldsymbol{\Theta}}^{(23)}+\Delta\tilde{\boldsymbol{\Theta}}^{(23)},\ \tilde{\boldsymbol{\Theta}}^{(34)}=\tilde{\boldsymbol{\Theta}}^{(34)}+\Delta\tilde{\boldsymbol{\Theta}}^{(34)}$$

其中$\Delta\tilde{\boldsymbol{\Theta}}^{(12)}=-\eta\frac{\partial E}{\partial\tilde{\boldsymbol{\Theta}}^{(12)}},\Delta\tilde{\boldsymbol{\Theta}}^{(23)}=-\eta\frac{\partial E}{\partial\tilde{\boldsymbol{\Theta}}^{(23)}},\Delta\tilde{\boldsymbol{\Theta}}^{(34)}=-\eta\frac{\partial E}{\partial\tilde{\boldsymbol{\Theta}}^{(34)}}$。

为了求$\Delta\tilde{\boldsymbol{\Theta}}^{(12)},\Delta\tilde{\boldsymbol{\Theta}}^{(23)},\Delta\tilde{\boldsymbol{\Theta}}^{(34)}$，先定义三个变量：

$$\boldsymbol{\delta}^{(4)}=\left(\delta_1^{(4)};\delta_2^{(4)};\delta_3^{(4)}\right)=\frac{\partial E}{\partial\boldsymbol{z}^{(4)}}=\left(\frac{\partial E}{\partial z_1^{(4)}};\frac{\partial E}{\partial z_2^{(4)}};\frac{\partial E}{\partial z_3^{(4)}}\right)=\frac{\partial E}{\partial\boldsymbol{a}^{(4)}}\odot f'\left(\boldsymbol{z}^{(4)}\right)$$

$$\boldsymbol{\delta}^{(3)}=\left(\delta_1^{(3)};\delta_2^{(3)};\delta_3^{(3)};\delta_4^{(3)}\right)=\frac{\partial E}{\partial\boldsymbol{z}^{(3)}}=\left(\frac{\partial E}{\partial z_1^{(3)}};\frac{\partial E}{\partial z_2^{(3)}};\frac{\partial E}{\partial z_3^{(3)}};\frac{\partial E}{\partial z_4^{(3)}}\right)=\frac{\partial E}{\partial\boldsymbol{a}^{(3)}}\odot f'\left(\boldsymbol{z}^{(3)}\right)$$

$$\boldsymbol{\delta}^{(2)}=\left(\delta_1^{(2)};\delta_2^{(2)};\delta_3^{(2)};\delta_4^{(2)}\right)=\frac{\partial E}{\partial\boldsymbol{z}^{(2)}}=\left(\frac{\partial E}{\partial z_1^{(2)}};\frac{\partial E}{\partial z_2^{(2)}};\frac{\partial E}{\partial z_3^{(2)}};\frac{\partial E}{\partial z_4^{(2)}}\right)=\frac{\partial E}{\partial\boldsymbol{a}^{(2)}}\odot f'\left(\boldsymbol{z}^{(2)}\right)$$

其中

$$f'\left(\boldsymbol{z}^{(4)}\right)=\left(\frac{\partial a_1^{(4)}}{\partial z_1^{(4)}};\frac{\partial a_2^{(4)}}{\partial z_2^{(4)}};\frac{\partial a_3^{(4)}}{\partial z_3^{(4)}}\right)$$

$$f'\left(\boldsymbol{z}^{(3)}\right)=\left(\frac{\partial a_1^{(3)}}{\partial z_1^{(3)}};\frac{\partial a_2^{(3)}}{\partial z_2^{(3)}};\frac{\partial a_3^{(3)}}{\partial z_3^{(3)}};\frac{\partial a_4^{(3)}}{\partial z_4^{(3)}}\right)$$

$$f'\left(\boldsymbol{z}^{(2)}\right)=\left(\frac{\partial a_1^{(2)}}{\partial z_1^{(2)}};\frac{\partial a_2^{(2)}}{\partial z_2^{(4)}};\frac{\partial a_3^{(2)}}{\partial z_3^{(2)}};\frac{\partial a_4^{(2)}}{\partial z_4^{(2)}}\right)$$

符号“$\odot$”表示矩阵哈达玛积，即两个大小相同的矩阵对应位置元素相乘。

对于$\boldsymbol{\delta}^{(4)}$来说，由于

$$\begin{aligned}\frac{\partial E}{\partial\boldsymbol{a}^{(4)}}&=\left(\frac{\partial E}{\partial a_1^{(4)}};\frac{\partial E}{\partial a_2^{(4)}};\frac{\partial E}{\partial a_3^{(4)}}\right)\\&=\left(a_1^{(4)}-y_1;a_2^{(4)}-y_2;a_3^{(4)}-y_3\right)=\boldsymbol{a}^{(4)}-\boldsymbol{y}\end{aligned}$$

因此$\boldsymbol{\delta}^{(4)}=(\boldsymbol{a}^{(4)}-\boldsymbol{y})\odot f'\left(\boldsymbol{z}^{(4)}\right)$；

对于$\boldsymbol{\delta}^{(3)}$来说，由于

$$\frac{\partial E}{\partial\boldsymbol{a}^{(3)}}=\left(\frac{\partial E}{\partial a_1^{(3)}};\frac{\partial E}{\partial a_2^{(3)}};\frac{\partial E}{\partial a_3^{(3)}};\frac{\partial E}{\partial a_4^{(3)}}\right)$$

接下来以$\frac{\partial E}{\partial a_1^{(3)}}$为例，详细推导$\frac{\partial E}{\partial\boldsymbol{a}^{(3)}}$中的每一项；已知

$$z_1^{(4)}=\theta_{01}^{(34)}a_0^{(3)}+\theta_{11}^{(34)}a_1^{(3)}+\theta_{21}^{(34)}a_2^{(3)}+\theta_{31}^{(34)}a_3^{(3)}+\theta_{41}^{(34)}a_4^{(3)}=\left[\tilde{\boldsymbol{\Theta}}_{:1}^{(34)}\right]^{\top}\tilde{\boldsymbol{a}}^{(3)}$$

$$z_2^{(4)}=\theta_{02}^{(34)}a_0^{(3)}+\theta_{12}^{(34)}a_1^{(3)}+\theta_{22}^{(34)}a_2^{(3)}+\theta_{32}^{(34)}a_3^{(3)}+\theta_{42}^{(34)}a_4^{(3)}=\left[\tilde{\boldsymbol{\Theta}}_{:2}^{(34)}\right]^{\top}\tilde{\boldsymbol{a}}^{(3)}$$

$$z_3^{(4)}=\theta_{03}^{(34)}a_0^{(3)}+\theta_{13}^{(34)}a_1^{(3)}+\theta_{23}^{(34)}a_2^{(3)}+\theta_{33}^{(34)}a_3^{(3)}+\theta_{43}^{(34)}a_4^{(3)}=\left[\tilde{\boldsymbol{\Theta}}_{:3}^{(34)}\right]^{\top}\tilde{\boldsymbol{a}}^{(3)}$$

其中$\mathbf{M}_{:i}$表示矩阵$\mathbf{M}$的第i列，则

$$\begin{aligned}\frac{\partial E}{\partial a_1^{(3)}}=&\frac{\partial E}{\partial z_1^{(4)}}\frac{\partial z_1^{(4)}}{\partial a_1^{(3)}}+\frac{\partial E}{\partial z_2^{(4)}}\frac{\partial z_2^{(4)}}{\partial a_1^{(3)}}+\frac{\partial E}{\partial z_3^{(4)}}\frac{\partial z_3^{(4)}}{\partial a_1^{(3)}}\\=&\frac{\partial E}{\partial z_1^{(4)}}\theta_{11}^{(34)}+\frac{\partial E}{\partial z_2^{(4)}}\theta_{12}^{(34)}+\frac{\partial E}{\partial z_3^{(4)}}\theta_{13}^{(34)}\\=&\left(\theta_{11}^{(34)},\theta_{12}^{(34)},\theta_{13}^{(34)}\right)\left(\frac{\partial E}{\partial z_1^{(4)}};\frac{\partial E}{\partial z_2^{(4)}};\frac{\partial E}{\partial z_3^{(4)}}\right)\\=&\boldsymbol{\Theta}_{1:}^{(34)}\frac{\partial E}{\partial \boldsymbol{z}^{(4)}}=\boldsymbol{\Theta}^{(34)}\boldsymbol{\delta}^{(4)}\end{aligned}$$

其中$\boldsymbol{v}=(v_1,v_2,\ldots,v_n)$表示$\boldsymbol{v}$为$1\times n$的行向量，$\mathbf{M}_{i:}$表示矩阵$\mathbf{M}$的第$i$行，进而有

$$\begin{aligned}\frac{\partial E}{\partial \boldsymbol{a}^{(3)}}=&\left(\boldsymbol{\Theta}_{1:}^{(34)}\boldsymbol{\delta}^{(4)};\boldsymbol{\Theta}_{2:}^{(34)}\boldsymbol{\delta}^{(4)};\boldsymbol{\Theta}_{3:}^{(34)}\boldsymbol{\delta}^{(4)};\boldsymbol{\Theta}_{4:}^{(34)}\boldsymbol{\delta}^{(4)}\right)\\=&\boldsymbol{\Theta}^{(34)}\boldsymbol{\delta}^{(4)}\end{aligned}$$

因此$\boldsymbol{\delta}^{(3)}=\left(\boldsymbol{\Theta}^{(34)}\boldsymbol{\delta}^{(4)}\right)\odot f'\left(\boldsymbol{z}^{(3)}\right)$；

同理，对于$\boldsymbol{\delta}^{(3)}$来说，由于

$$\frac{\partial E}{\partial \boldsymbol{a}^{(2)}}=\left(\frac{\partial E}{\partial a_1^{(2)}};\frac{\partial E}{\partial a_2^{(2)}};\frac{\partial E}{\partial a_3^{(2)}};\frac{\partial E}{\partial a_4^{(2)}}\right)$$

接下来以$\frac{\partial E}{\partial a_1^{(2)}}$为例，详细推导$\frac{\partial E}{\partial \boldsymbol{a}^{(2)}}$中的每一项；已知

$$z_1^{(3)}=\theta_{01}^{(23)}a_0^{(2)}+\theta_{11}^{(23)}a_1^{(2)}+\theta_{21}^{(23)}a_2^{(2)}+\theta_{31}^{(23)}a_3^{(2)}+\theta_{41}^{(23)}a_4^{(2)}=\left[\tilde{\boldsymbol{\Theta}}_{:1}^{(23)}\right]^{\top}\tilde{\boldsymbol{a}}^{(2)}$$

$$z_2^{(3)}=\theta_{02}^{(23)}a_0^{(2)}+\theta_{12}^{(23)}a_1^{(2)}+\theta_{22}^{(23)}a_2^{(2)}+\theta_{32}^{(23)}a_3^{(2)}+\theta_{42}^{(23)}a_4^{(2)}=\left[\tilde{\boldsymbol{\Theta}}_{:2}^{(23)}\right]^{\top}\tilde{\boldsymbol{a}}^{(2)}$$

$$z_3^{(3)}=\theta_{03}^{(23)}a_0^{(2)}+\theta_{13}^{(23)}a_1^{(2)}+\theta_{23}^{(23)}a_2^{(2)}+\theta_{33}^{(23)}a_3^{(2)}+\theta_{43}^{(23)}a_4^{(2)}=\left[\tilde{\boldsymbol{\Theta}}_{:3}^{(23)}\right]^{\top}\tilde{\boldsymbol{a}}^{(2)}$$

$$z_4^{(3)}=\theta_{04}^{(23)}a_0^{(2)}+\theta_{14}^{(23)}a_1^{(2)}+\theta_{24}^{(23)}a_2^{(2)}+\theta_{34}^{(23)}a_3^{(2)}+\theta_{44}^{(23)}a_4^{(2)}=\left[\tilde{\boldsymbol{\Theta}}_{:4}^{(23)}\right]^{\top}\tilde{\boldsymbol{a}}^{(2)}$$

则

$$\begin{aligned}\frac{\partial E}{\partial a_1^{(2)}}=&\frac{\partial E}{\partial z_1^{(3)}}\frac{\partial z_1^{(3)}}{\partial a_1^{(2)}}+\frac{\partial E}{\partial z_2^{(3)}}\frac{\partial z_2^{(3)}}{\partial a_1^{(2)}}+\frac{\partial E}{\partial z_3^{(3)}}\frac{\partial z_3^{(3)}}{\partial a_1^{(2)}}+\frac{\partial E}{\partial z_4^{(3)}}\frac{\partial z_4^{(3)}}{\partial a_1^{(2)}}\\=&\frac{\partial E}{\partial z_1^{(3)}}\theta_{11}^{(23)}+\frac{\partial E}{\partial z_2^{(3)}}\theta_{12}^{(23)}+\frac{\partial E}{\partial z_3^{(3)}}\theta_{13}^{(23)}+\frac{\partial E}{\partial z_4^{(3)}}\theta_{14}^{(23)}\\=&\left(\theta_{11}^{(23)},\theta_{12}^{(23)},\theta_{13}^{(23)},\theta_{14}^{(23)}\right)\left(\frac{\partial E}{\partial z_1^{(3)}};\frac{\partial E}{\partial z_2^{(3)}};\frac{\partial E}{\partial z_3^{(3)}};\frac{\partial E}{\partial z_4^{(3)}}\right)\\=&\boldsymbol{\Theta}_{1:}^{(23)}\frac{\partial E}{\partial \boldsymbol{z}^{(3)}}=\boldsymbol{\Theta}_{1:}^{(23)}\boldsymbol{\delta}^{(3)}\end{aligned}$$

进而有

$$\begin{aligned}\frac{\partial E}{\partial \boldsymbol{a}^{(2)}} =& \left(\boldsymbol{\Theta}_{1:}^{(23)}\boldsymbol{\delta}^{(3)}; \boldsymbol{\Theta}_{2:}^{(23)}\boldsymbol{\delta}^{(3)}; \boldsymbol{\Theta}_{3:}^{(23)}\boldsymbol{\delta}^{(3)}; \boldsymbol{\Theta}_{4:}^{(23)}\boldsymbol{\delta}^{(3)}\right) \\ =& \boldsymbol{\Theta}^{(23)}\boldsymbol{\delta}^{(3)}\end{aligned}$$

因此$\boldsymbol{\delta}^{(2)} = \left(\boldsymbol{\Theta}^{(23)}\boldsymbol{\delta}^{(3)}\right) \odot f'\left(\boldsymbol{z}^{(2)}\right)$；

总结一下：

$$\begin{aligned}\boldsymbol{\delta}^{(4)} &= \frac{\partial E}{\partial \boldsymbol{z}^{(4)}} = \frac{\partial E}{\partial \boldsymbol{a}^{(4)}} \odot f'\left(\boldsymbol{z}^{(4)}\right) = \left(\boldsymbol{a}^{(4)} - \boldsymbol{y}\right) \odot f'\left(\boldsymbol{z}^{(4)}\right) \\ \boldsymbol{\delta}^{(3)} &= \frac{\partial E}{\partial \boldsymbol{z}^{(3)}} = \frac{\partial E}{\partial \boldsymbol{a}^{(3)}} \odot f'\left(\boldsymbol{z}^{(3)}\right) = \left(\boldsymbol{\Theta}^{(34)}\boldsymbol{\delta}^{(4)}\right) \odot f'\left(\boldsymbol{z}^{(3)}\right) \\ \boldsymbol{\delta}^{(2)} &= \frac{\partial E}{\partial \boldsymbol{z}^{(2)}} = \frac{\partial E}{\partial \boldsymbol{a}^{(2)}} \odot f'\left(\boldsymbol{z}^{(2)}\right) = \left(\boldsymbol{\Theta}^{(23)}\boldsymbol{\delta}^{(3)}\right) \odot f'\left(\boldsymbol{z}^{(2)}\right)\end{aligned}$$

有了以上三个变量，就可以求出$\Delta\tilde{\boldsymbol{\Theta}}^{(12)}, \Delta\tilde{\boldsymbol{\Theta}}^{(23)}, \Delta\tilde{\boldsymbol{\Theta}}^{(34)}$了，具体如下：

对于$\tilde{\boldsymbol{\Theta}}^{(34)}$来说，第 1 列只与$z_1^{(4)}$有关，第 2 列只与$z_2^{(4)}$有关，第 3 列只与$z_3^{(4)}$有关，即：

$$z_1^{(4)} = \theta_{01}^{(34)}a_0^{(3)} + \theta_{11}^{(34)}a_1^{(3)} + \theta_{21}^{(34)}a_2^{(3)} + \theta_{31}^{(34)}a_3^{(3)} + \theta_{41}^{(34)}a_4^{(3)}$$

$$z_2^{(4)} = \theta_{02}^{(34)}a_0^{(3)} + \theta_{12}^{(34)}a_1^{(3)} + \theta_{22}^{(34)}a_2^{(3)} + \theta_{32}^{(34)}a_3^{(3)} + \theta_{42}^{(34)}a_4^{(3)}$$

$$z_3^{(4)} = \theta_{03}^{(34)}a_0^{(3)} + \theta_{13}^{(34)}a_1^{(3)} + \theta_{23}^{(34)}a_2^{(3)} + \theta_{33}^{(34)}a_3^{(3)} + \theta_{43}^{(34)}a_4^{(3)}$$

例如，$\frac{\partial E}{\partial \theta_{01}^{(34)}} = \frac{\partial E}{\partial z_1^{(4)}}\frac{\partial z_1^{(4)}}{\partial \theta_{01}^{(34)}} = \delta_1^{(4)}a_0^{(3)}$；通用的公式为

$$\frac{\partial E}{\partial \theta_{ij}^{(kl)}} = \frac{\partial E}{\partial z_j^{(l)}}\frac{\partial z_j^{(l)}}{\partial \theta_{ij}^{(kl)}} = \delta_j^{(l)}a_i^{(k)} \quad (0 \le i \le s_k,\ 1 \le j \le s_l,\ l = k+1)$$

$$\frac{\partial E}{\partial \tilde{\boldsymbol{\Theta}}^{(kl)}} = \tilde{\boldsymbol{a}}^{(k)}\left[\boldsymbol{\delta}^{(l)}\right]^\top \quad (1 \le k \le 3,\ 2 \le l \le 4)$$

进而有

$$\Delta\tilde{\boldsymbol{\Theta}}^{(12)} = -\eta\tilde{\boldsymbol{a}}^{(1)}\left[\boldsymbol{\delta}^{(2)}\right]^\top, \Delta\tilde{\boldsymbol{\Theta}}^{(23)} = -\eta\tilde{\boldsymbol{a}}^{(2)}\left[\boldsymbol{\delta}^{(3)}\right]^\top, \Delta\tilde{\boldsymbol{\Theta}}^{(34)} = -\eta\tilde{\boldsymbol{a}}^{(3)}\left[\boldsymbol{\delta}^{(4)}\right]^\top$$

到此，推导结束。以上推导过程可以轻易推广至更一般的情况，对于n层神经网络而言（1 个输入层，1 个输出层，$n-2$个隐层），网络在样本$(\boldsymbol{x}, \boldsymbol{y})$上的均方误差为：

$$E = \frac{1}{2}\left\|\hat{\boldsymbol{y}} - \boldsymbol{y}\right\|^2 = \frac{1}{2}\left\|\boldsymbol{a}^{(n)} - \boldsymbol{y}\right\|^2 = \frac{1}{2}\sum_{j=1}^{s_n}(a_j^{(n)} - y_j)^2$$

对于第 2 层至第n层，

$$\begin{aligned}&\boldsymbol{\delta}^{(n)} = \frac{\partial E}{\partial \boldsymbol{z}^{(n)}} = \frac{\partial E}{\partial \boldsymbol{a}^{(n)}} \odot f'\left(\boldsymbol{z}^{(n)}\right) = \left(\boldsymbol{a}^{(n)} - \boldsymbol{y}\right) \odot f'\left(\boldsymbol{z}^{(n)}\right) \\ &\cdots \\ &\boldsymbol{\delta}^{(k)} = \frac{\partial E}{\partial \boldsymbol{z}^{(k)}} = \frac{\partial E}{\partial \boldsymbol{a}^{(k)}} \odot f'\left(\boldsymbol{z}^{(k)}\right) = \left(\boldsymbol{\Theta}^{(kl)}\boldsymbol{\delta}^{(l)}\right) \odot f'\left(\boldsymbol{z}^{(k)}\right), k = l-1 \\ &\cdots \\ &\boldsymbol{\delta}^{(2)} = \frac{\partial E}{\partial \boldsymbol{z}^{(2)}} = \frac{\partial E}{\partial \boldsymbol{a}^{(2)}} \odot f'\left(\boldsymbol{z}^{(2)}\right) = \left(\boldsymbol{\Theta}^{(23)}\boldsymbol{\delta}^{(3)}\right) \odot f'\left(\boldsymbol{z}^{(2)}\right)\end{aligned}$$

第k层到第$l = k+1$层的权重更新为

$$\Delta\tilde{\boldsymbol{\Theta}}^{(kl)} = -\eta\tilde{\boldsymbol{a}}^{(k)}\left[\boldsymbol{\delta}^{(l)}\right]^\top \quad (1 \le k \le n-1,\ 2 \le l \le n)$$

$$\tilde{\boldsymbol{\Theta}}^{(kl)} = \tilde{\boldsymbol{\Theta}}^{(kl)} + \Delta\tilde{\boldsymbol{\Theta}}^{(kl)}$$

5.4 全局最小与局部极小

本节内容通俗易懂，几乎不需要什么注解。

由图 5.10 可以轻易理解局部极小和全局最小的概念，其余概念如模拟退火、遗传算法、随机梯度下降、启发式等，则需要专门查一些资料慢慢消化理解，此处不再赘述。

5.5 其他常见神经网络

本节内容高度概括，能看懂多少算多少吧。

1、RBF 网络

从式(5.18)可以看出，对于样本$\boldsymbol{x}$来说，RBF 网络的输出为q个$\rho(\boldsymbol{x}, \boldsymbol{c}_i)$的线性组合。若换个角度来看这个问题，将$q$个$\rho(\boldsymbol{x}, \boldsymbol{c}_i)$当作是将$d$维向量$\boldsymbol{x}$基于式(5.19)进行特征转换后所得的$q$维特征，即$\tilde{\boldsymbol{x}} = (\rho(\boldsymbol{x}, \boldsymbol{c}_1); \rho(\boldsymbol{x}, \boldsymbol{c}_2); \ldots; \rho(\boldsymbol{x}, \boldsymbol{c}_q))$，则式(5.18)求线性加权系数$w_i$相当于求解第 3.2 节的线性回归$f(\boldsymbol{x}) = \boldsymbol{w}^\top \tilde{\boldsymbol{x}} + b$，对于仅有的差别 bias 项$b$来说，你当然可以在式(5.18)中也加入 bias 项，很多人实际使用时就这么干的。

因此，RBF 网络在确定q个神经元中心$\boldsymbol{c}_i$之后（核心步骤，个人观点），接下来要做的就是线性回归，而这个过程与第六章的 SVM 有类似之处，详见式(6.24)的解释。

特别补充一句，式(5.19)称为高斯径向基函数，只是径向基函数(RBF)的一种！！！

西瓜书作者最近在 AAAI’18 上的工作“Dual-set multi-label learning”中的基分类器就是 RBF 网络，源码可以在论文第一作者的主页下载（https://chong-l.github.io/）。

2、增量学习和在线学习

西瓜书的优点就在于将很多零散的概念巧妙的囊括进来，此处即为一个例子，以下仅将第 109 页边注中的原文摘抄如下：

增量学习(incremental learning)是指在学得模型后，再接收到训练样例时，仅需根据新样例对模型进行更新，不必重新训练整个模型，并且先前学得的有效信息不会被“冲掉”；在线学习(online learning)是指每获得一个新样本就进行一次模型更新。显然，在线学习是增量学习的特例，而增量学习可视为“批量式”(batch-mode)的在线学习。

3、递归神经网络

第 5.5.5 节介绍的 Elman 网络是递归神经网络的一种。

递归神经网络即常听到的 RNN，全称是 Recurrent Neural Networks 或 Recursive Neural Networks；RNN 进一步演进就是长短时间记忆(Long Short-Term Memory, LSTM)网络。此类网络结构善长于处理时序信号，比如语音。

5.6 深度学习

本节用了两页零三行的篇幅介绍深度学习，一个当今火遍全世界的概念；虽然，业内常说深度学习并没有什么理论上的突破，但用如此篇幅显然无法让初学者理解深度学习的概念和技巧。如果说上一节是高度概括，那么相比之下，本节有过之而无不及；所以，还是上一节注解开篇的那句话：能看懂多少算多少吧。

接下来，介绍几个常见到的有关深度学习的概念。

1、什么是深度学习

在西瓜书作者前两年推广其提出的深度森林时，经常引用 SIAM News (Jun. 2017)中的一段话来解释深度学习的概念[slides]：

即深度学习就是指深度神经网络(Deep Neural Network, DNN)。

2、什么是端到端(end-to-end)的思想

提及深度学习，一般都会说它是端到端的学习。端到端学习的概念其实就是西瓜书本节内容最后两段所要表达的意思。概括地说，以前对图片进行分类时，首先要设计特征提取方法，比如常见的尺度不变特征变换(Scale-invariant feature transform, SIFT)，然后才能进行分类；分类结果的好坏很大程度上取决于所设计的特征。而对于深度学习来说，输入就是原始的图片像素值，输出就是分类结果，即端到端的学习。

3、什么是卷积神经网络

西瓜书第 114 页图 5.15 就是一个典型的卷积神经网络(Convolutional Neural Network, CNN)。至于 CNN 的原理，此处略去几万字，推荐两本书吧，看完之后一些常见的概念如卷积层、全连接层、pooling 层、ReLU、Dropout 等就都会有所了解：

(1) 汤晓鸥, 陈玉琨 主编. 人工智能基础(高中版). 华东师范大学出版社, 2018.

该书第 3 章详述了 CNN 的原理，图文并茂，简单易懂。

以下是网上搜到的两个分享链接，建议购买一本作为科普读物，三十元左右：

链接 1: https://pan.baidu.com/s/1mqUsIBHXAH2t4JHbJw5_6w 提取码: gr7f

链接 2: https://pan.baidu.com/s/1yYbmxxKbSX0uDm7k0HSizg 提取码: aqao

(2) 魏秀参 著. 解析深度学习：卷积神经网络原理与视觉实践. 电子工业出版社, 2018.

该书较为全面的介绍了 CNN 的方方面面，但个人感觉仍以科普为主，看完本书之后对 CNN 会有一个较为全面的了解。

作者魏秀参先后师从周老师和吴建鑫老师从事机器学习和计算机视觉方向研究，现任旷视(Face++)南京研究院负责人，作者早期曾将该书以 preprint 形式公开于网络，链接如下：

链接 1: http://lamda.nju.edu.cn/weixs/book/CNN_book.pdf

链接 2: https://pan.baidu.com/s/1pLcaFij

链接 3: https://drive.google.com/file/d/1sa1aSzYrNtGzXbegL02JtbYw3z3ZE13m/view?usp=sharing

另外，作者的微博和知乎：

微博 Wilson_NJUer: https://weibo.com/p/1005052618378195

知乎: https://www.zhihu.com/people/wei-xiu-shen/activities

4、什么是梯度爆炸和梯度消失

在 5.3 节注解的最后一条推导了多隐层神经网络的 BP 算法，除输入层外，每层定义的误差变量$\boldsymbol{\delta}^{(k)}$是由误差从输出层开始逐层求导（连乘）决定的，这其中包含了激活函数的导数以及初始化的权重。当神经网络层数较多时，可能得到的$\boldsymbol{\delta}^{(k)}$会非常大（梯度爆炸）或非常小（梯度消失），这直接影响权重更新量$\Delta\tilde{\boldsymbol{\Theta}}^{(kl)}$，这将导致深度神经网络无法训练。一种解决方法是将激活函数由图 5.2(b)的 Sigmoid 函数替换为 ReLU 函数（参见第 114 页的边注）。

5、什么是 ImageNet

了解深度学习，会接触到很多 xxxNet，如 AlexNet, VGGNet, GoogLeNet, ResNet 等等。

另外，还有一个 Net 必须知道，那就是 ImageNet。但是，ImageNet 并不是一种神经网络结构，而是斯坦福大学李飞飞负责标注一个大型数据集。该数据集就是以上提到的 AlexNet, VGGNet, GoogLeNet, ResNet 能够闪亮登场的伯乐，即大规模视觉识别挑战赛(Imagenet Large Scale Visual Recognition Challenge, ILSVRC)。李飞飞在 2018 年当选 ACM Fellow 时，ACM 官方给出的解释(https://awards.acm.org/award_winners/li_4289668)是“For contributions in building large knowledge bases for machine learning and visual understanding”，由此可见 ImageNet 的贡献。ImageNet 官方网站参见：http://www.image-net.org/

5.7 本章小节

本章讲解最为详细的一部分内容当属 5.3 节的 BP 算法，掌握 BP 算法是理解深度神经网络基础之一。当年 BP 算法的提出使多层网络的训练成为可能，掀起了神经网络研究的第二次高潮（参见西瓜书第 120 页的小故事），虽然近些年 BP 算法的作者 Hinton 曾说要将 BP 算法推翻（参见微信公众号机器之心 2017 年 9 月的两篇推送《Geoffrey Hinton：放弃反向传播，我们的人工智能需要重头再来》和《被 Geoffrey Hinton 抛弃，反向传播为何饱受质疑？（附 BP 推导）》），但在可预见的未来，BP 算法仍将是训练多层神经网络的主要工具。

深度学习无疑是当今最火的研究方向之一。当地时间 2019 年 3 月 27 日，美国计算机协会(ACM)宣布了今年的图灵奖获得者，深度学习三剑客 Yoshua Bengio、Geoffery Hinton 和 Yann LeCun 共同获此荣誉，这是三人学术贡献的肯定，更是对深度学习的肯定。

深度学习可理解为进行“表示学习”(representation learning，参见 5.6 节最后两段)，近几年新兴起的深度学习学术会议 ICLR(International Conference on Learning Representations)即以此命名。尽管 2019 年最新版的《中国计算机学会推荐国际学术会议和期刊目录》仍未收录该会议，但由深度学习三剑客之二的 Yoshua Bengio 和 Yann LeCun 牵头于 2013 年创办 ICLR 已经被学术研究者们广泛认可，被认为是「深度学习的顶级会议」。

自从 2012 年 Hinton 和他的学生 Alex Krizhevsky 设计的 AlexNet 在 ImageNet 竞赛中以大幅优势夺冠之后，“深度神经网络”深入人心，而且网络越做越深。为什么要将网络加深而不是加宽呢？实际上，“[Hornik et al., 1989]证明，只需一个包含足够多神经元的隐层，多层前馈网络就能以任意精度逼近任意复杂度的连续函数”（西瓜书第 105 页第 2 段），有关这个问题作者在 5.6 节第 2 段进行了解释，这里引用【周志华. 机器学习:发展与未来[J]. 中国计算机学会通讯, 2017, 13(1): 44-51.】中的一段话回答这个问题：

其实在机器学习理论里面，我们很早就知道，大致来说，如果你能够提升一个模型的复杂度，那么就可以提升其学习能力。比如说对神经网络这样的模型，我们怎么样提升它的复

杂度呢？很明显有两个办法：一个办法，是把网络加宽；另外一个办法，是把它加深。但是如果从提升复杂度的角度来说，加深会更有用。**因为加宽的话其实是增加了基函数的个数；加深的话，不只增加了函数个数，还增加了函数嵌套的层数，从泛函表达上它的能力会更好。**所以“加深”对增强模型的复杂度和学习能力更有用。

当然，也有学者研究将网络加宽……

澳门大学陈俊龙（https://www.fst.um.edu.mo/en/staff/pchen.html）于 2018 年发表了有关宽度学习系统(Broad Learning Systems)的论文：[Chen, C. P., & Liu, Z. (2018). Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. IEEE transactions on neural networks and learning systems, 29(1), 10-24.]，有关宽度学习系统还可以参见以下两个链接：

http://www.broadlearning.ai/

https://mp.weixin.qq.com/s/Zze1O83PZg9OBdZ7L7AQ7A

除了基于神经网络的深度学习，西瓜书作者还在考虑其它实现深度学习的方式，如作者在 IJCAI'17 上发表的 Deep Forest: Towards an Alternative to Deep Neural Networks 提出了深度森林模型，该文扩展后以“Deep Forest”为题发表于《国家科学评论》（National Science Review, NSR），链接: https://doi.org/10.1093/nsr/nwy108，arXiv: https://arxiv.org/abs/1702.08835。

若想更进一步学习掌握深度学习，西瓜书肯定是不够的，这时当然是要看 Ian Goodfellow、Yoshua Bengio 和 Aaron Courville 撰写的、号称 AI 圣经的花书《深度学习》：

英文版：http://www.deeplearningbook.org/

中文版：https://github.com/exacity/deeplearningbook-chinese

深度学习课程当然首推 Andrew Ng 的 deeplearning.ai 系列在线课程：

Coursera 网址：https://www.coursera.org/specializations/deep-learning

网易云课堂网址：https://mooc.study.163.com/university/deeplearning_ai#/c

课程笔记：https://github.com/fengdu78/deeplearning_ai_books（该笔记由中国海洋大学博士黄海广负责整理，他还专门建立了一个机器学习爱好者网站 http://www.ai-start.com/）

另外还有斯坦福大学的 CS230 Deep Learning 课程（http://cs230.stanford.edu/），以及台大的 Applied Deep Learning 课程（https://www.csie.ntu.edu.tw/~yvchen/f106-adl/index.html）。

Boltzmann 机和深度信念网络(Deep Belief Network, DBN)也是 Hinton 的代表性工作（分别参见 5.5.6 节和 5.6 节第 3 段），但现实中（读论文或开会听报告）暂时也没遇到，因此就先放一放吧。值得一提的是，DBN 也是动态贝叶斯网络(Dynamic Bayesian Network)的简写，DBN 可用于处理时序数据，而第 7 章 7.5 节介绍的贝叶斯网是静态贝叶斯网络。

期刊 Nature 和 Science 在学术界的地位不必多说，以下是与本章内容相关的几篇：

首先是三篇 Hinton 发表的有关神经网络/深度学习的 Nature/Science:

[1] Rumerlhar, D. E., **Hinton G. E.**, Williams R. J. (1986). Learning representation by back-propagating errors. ***Nature***, 323, 533-536.

[2] **Hinton, G. E.**, & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. ***Science***, 313(5786), 504-507.

[3] LeCun, Y., Bengio, Y., & **Hinton, G. E.** (2015). Deep learning. ***Nature***, 521(7553), 436-444.

还有一篇其他人发表的有关深度学习应用的 Nature：

[4] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. ***Nature***, 542(7639), 115-118.

除了以上四篇，还在网上搜到了 CMU 的 Tom M. Mitchell 发表的一篇 Science：

[5] Brynjolfsson, E. , & Mitchell, T. . (2017). What can machine learning do? workforce implications. ***Science***, 358(6370), 1530-1534.

最后，解释几个常听到的简写：

DNN(Deep Neural Network)，深度神经网络

CNN(Convolutional Neural Network)，卷积神经网络，善长于处理图像

RNN(Recurrent/Recursive Neural Networks)，递归神经网络，善长于处理时序信号

LSTM(Long Short Term Memory)，长短时间记忆，亦善长于处理时序信号

GAN(Generative Adversarial Networks)，生成对抗网络，之所以常将其归类为深度学习模型，是由于其两个组成部分（生成模型和判别模型）一般由深度神经网络构成

尽管当今连接主义大红大紫，但仍有人坚信统计主义和符号主义终究会回来的（序言中也有提到），让我们一起期待并努力吧……