

深度学习理论与实践

第二课：神经网络

主讲人 郑元春

中科院大数据挖掘与知识管理重点实验室
中科院虚拟经济与数据科学研究中心





知识引入



感知机



神经网络



反向传播



手写数字识别



1. 知识引入-极大似然估计

什么是极大似然估计？

利用已知的样本结果信息（观测值，输入特征），反推最具有可能（概率最大）导致这些样本结果出现的模型参数。

- 假设一：数据独立同分布
- 假设二：分布已知

极大似然估计：模型已经确定，参数未知；对模型的求解就是利用观测值来求解模型的参数。

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

预估参数

当参数求解出来，则整个模型是确定的



1. 知识引入-极大似然估计

什么是极大似然估计？



- 如果 θ 是已知的，则该函数叫做概率函数(Probability)，描述的是不同样本点 x 出现的概率；
- 如果 x 是已知的，则该函数叫做似然函数(Likelihood)，描述的是对于不同的模型参数，出现样本点 x 的概率。



1. 知识引入-极大似然估计

案例：估计白球比例

有一个罐子，里面有黑球和白球，其各自数目与比例不知。

想知道罐子中黑球和白球的比例，怎么做？



方案：从罐子中随机拿出一个球，记录其颜色并放回，这个抽样过程可以重复 N 次，通过 N 次实验记录的小球颜色来估计罐子中黑白球的比例。假如 $N = 100$ ，而且其中有70次是白色球，那么罐子中白球所占比例最有可能是多少？

直觉上罐子中白球比例最有可能是0.7
那么怎么证明呢？



1. 知识引入-极大似然估计

案例：估计白球比例

假设罐子中白球比例是 p ，那么黑球的比例就是 $1 - p$ ，由于每次是有放回的随机取出一个球，则每个拿出球的颜色这一事件 $\{x_1, x_2, \dots, x_{100}\}$ 服从**独立同分布假设**。将100次抽样中有70次白球的概率记做 $p(O|M)$ ，则有：

$$p(O|M) = C_{100}^{70} p^{70} (1 - p)^{30}$$

其中， M 表示模型； p 表示模型参数。

不同的 p 对应不同的分布，如何选择分布使得事件发生概率最大呢？



1. 知识引入-极大似然估计

案例：估计白球比例

$$p(O|M) = C_{100}^{70} p^{70} (1-p)^{30}$$

样本结果 O 出现的可能性最大，即函数 $p(O|M)$ 取极大值

对 $p(O|M)$ 中的参数 p 进行求导，令其导数为0，则可求出 $p(O|M)$ 的极大值。

$$\frac{\partial p(O|M)}{\partial p} = 0 \Rightarrow 70p^{69}(1-p)^{30} - 30p^{70}(1-p)^{29} = 0$$

$$\begin{aligned} &\text{即 } 70p^{69}(1-p)^{30} - 30p^{70}(1-p)^{29} \\ &= p^{69}(1-p)^{29}[70(1-p) - 30p] \\ &= 0 \end{aligned}$$

得 $p = 0, 1, 0.7$, 取 $p = 0.7$

思考： $p = 0$ 与 $p = 1$ 也是极值点，为什么不使用这两个点



1. 知识引入

作业题 I

有一个罐子，里面有黑球和白球，从罐子中随机拿出一个球，记录其颜色并放回，这个抽样过程重复100次，其中有70次是白色球，那么罐子中的白球所占比例最有可能是多少。我们对该问题进行建模时，目标函数为：

- A. 概率函数
- B. 似然函数

有一个罐子，里面有黑球和白球，白球占比70%，从罐子中随机拿出一个球，该球为白色球的概率是多少。我们对该问题进行建模时，目标函数为：

- A. 概率函数
- B. 似然函数





1. 知识引入-逻辑回归

逻辑回归

概述：一种分类算法，可以处理二元分类以及多元分类。

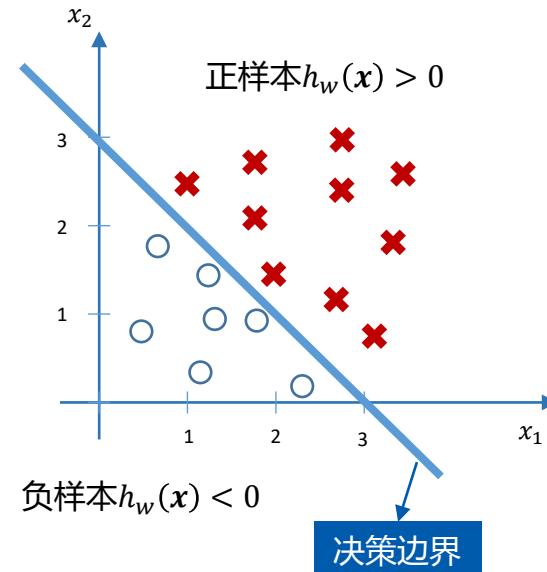
本质：假设数据服从一个分布，然后使用**极大似然估计**的方法估计该分布中的参数。

对于二分类问题，令分类界面（决策边界）为：

$$w_1 x_1 + w_2 x_2 + b = 0$$

若某样本点满足 $h_w(x) = w_1 x_1 + w_2 x_2 + b > 0$ ，

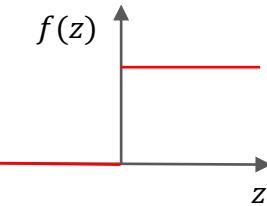
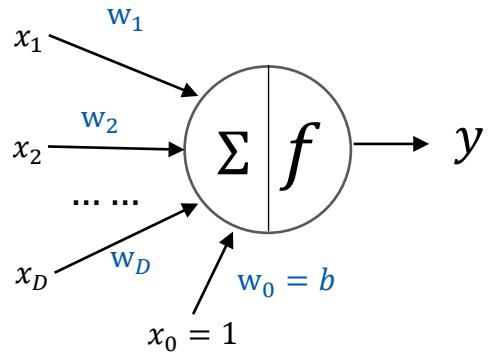
则该样本为正样本（类别为1）；相反，则亦然（类别为0）。





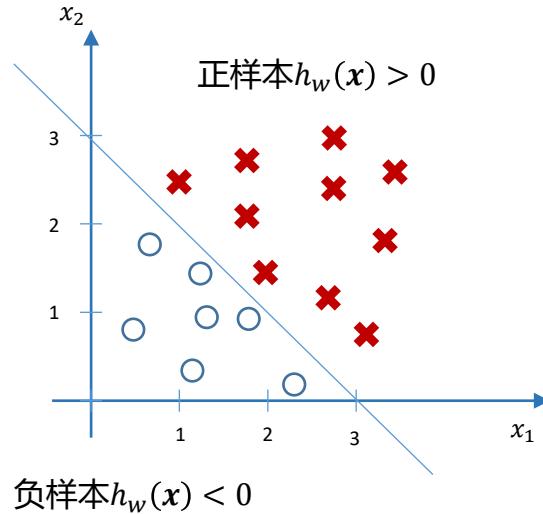
1. 知识引入-逻辑回归

逻辑回归



$$y = f\left(\sum_{i=0}^D w_i x_i\right)$$

感知机M-P模型



线性分类器

思考：感知机M-P模型与线性分类器，本质是否是一致的？



1. 知识引入

逻辑回归

目标：建立分类概率 $P(y = 1)$ 与输入向量 x 之间的直接关系，然后通过比较概率值来判别类别。

以二分类问题为例，给定数据集： $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$,

其中 $x_i \in \mathcal{R}^n, y_i \in \{0,1\}$

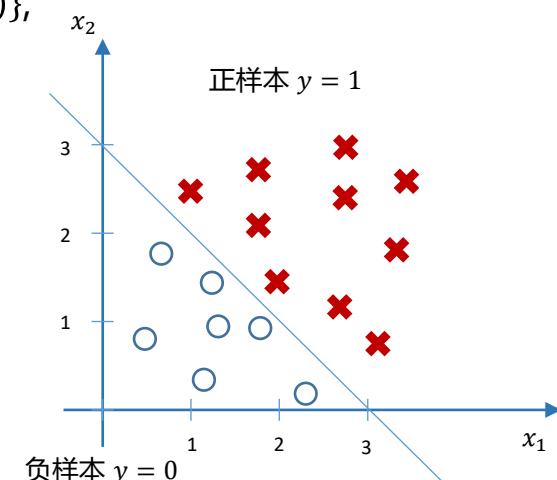
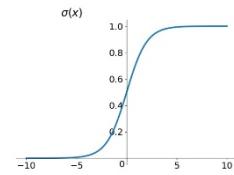
如何将决策边界 $w^T x$ 与分类概率建立联系？

$$P(y = 1|x) = \frac{1}{1 + e^{-(w^T x)}}$$

补充：Logistic函数/Sigmoid函数

函数形式： $\sigma(x) = \frac{1}{1+e^{-W^T x}}$

特点：可以将实数空间压缩到 $[0,1]$ 范围内。





1. 知识引入

逻辑回归

为了方便，将样本 x 是正/负样本概率分别表示为：

$$P(y = 1|x) = p(x)$$

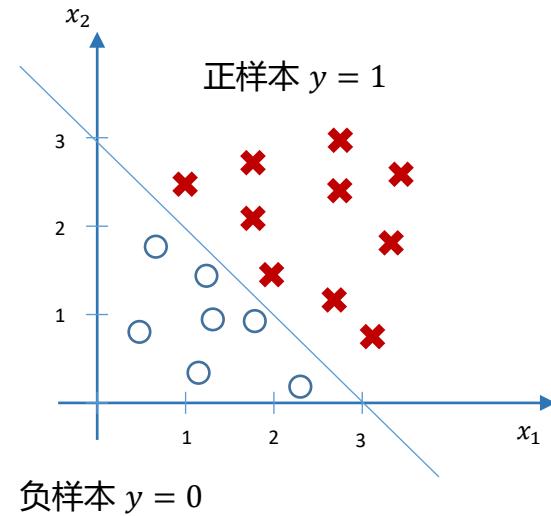
$$P(y = 0|x) = 1 - p(x)$$

则对于数据集 D 中，所有样本的似然函数为：

$$\prod [p(x_i)]^{y_i} [1 - p(x_i)]^{1-y_i}$$

为了求导方便，将连乘转化为累加的形式（取自然对数）：

$$L(w) = \sum [y_i \ln p(x_i) + (1 - y_i) \ln(1 - p(x_i))]$$





1. 知识引入

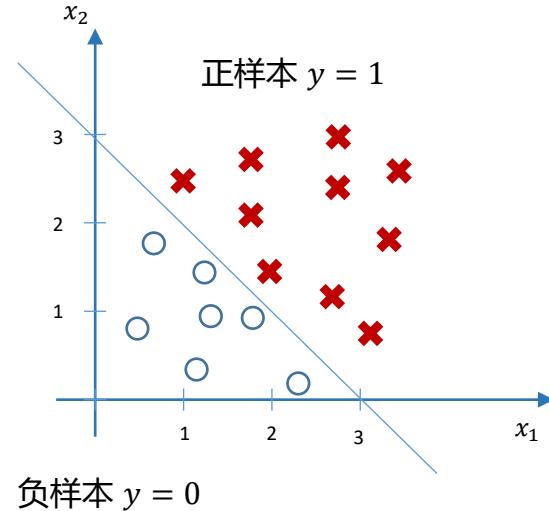
逻辑回归

对于数据集 D 中所有样本的对数似然函数为：

$$\begin{aligned}L(\mathbf{w}) &= \sum y_i \ln p(\mathbf{x}_i) + (1 - y_i) \ln(1 - p(\mathbf{x}_i)) \\&= \sum y_i \ln \frac{p(\mathbf{x}_i)}{1 - p(\mathbf{x}_i)} + \ln(1 - p(\mathbf{x}_i)) \\&= \sum (y_i - 1)\mathbf{w}^T \mathbf{x}_i - \ln(1 + e^{-\mathbf{w}^T \mathbf{x}_i})\end{aligned}$$

损失函数：对数似然函数取负号（求极小值）。

$$\begin{aligned}J(\mathbf{w}) &= -L(\mathbf{w}) \\&= -\sum y_i \ln p(\mathbf{x}_i) + (1 - y_i) \ln(1 - p(\mathbf{x}_i))\end{aligned}$$





1. 知识引入

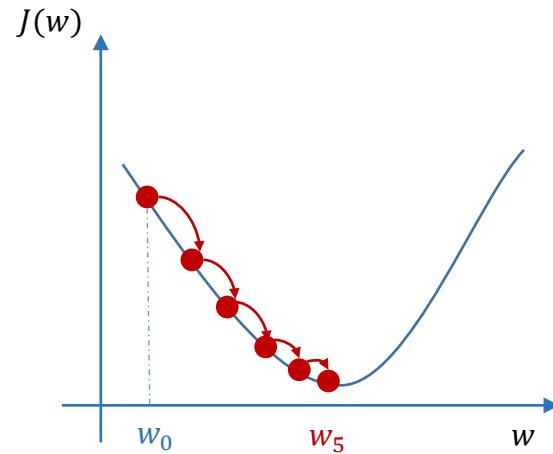
逻辑回归

随机梯度下降：通过一阶导数来寻找下降方向，迭代更新参数。

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = (p(\mathbf{x}_i) - y_i)\mathbf{x}_i$$

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \alpha \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}$$

其中， k 为算法的迭代次数， α 是更新步长（缩放系数）。



梯度下降演示



1. 知识引入

作业题 II

(1) 尝试推导逻辑回归的损失函数表达式：

$$J(\mathbf{w}) = - \sum y_i \ln p(\mathbf{x}_i) + (1 - y_i) \ln(1 - p(\mathbf{x}_i))$$

(2) 梯度下降法是常用的优化方法，负梯度方向为损失函数的下降方向，这也是逻辑回归损失函数中参数更新公式的由来：

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = (p(\mathbf{x}_i) - y_i) \mathbf{x}_i$$

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \alpha \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \mathbf{w}^k + \alpha \left(-\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} \right).$$

请证明：为什么负梯度方向是损失函数下降的方向？

提示：可以尝试将 $J(\mathbf{w}^{k+1}) = J(\mathbf{w}^k + \Delta \mathbf{w})$ 一阶泰勒展开。



知识引入



感知机



神经网络



反向传播



手写数字识别

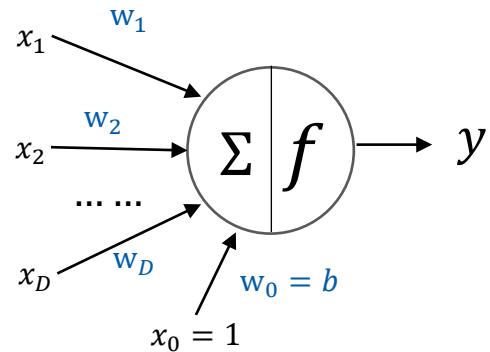


2. 感知机

感知机学习策略

假设数据集是线形可分的，感知机的学习目标是求得一个能够将训练集正样本和负样本完全分开的超平面（决策边界）。

为了找出这样的超平面（即确定感知机模型参数 w, b ），需要确定一个学习策略（即定义一个合理的损失函数，使得损失函数最小化时得到的超平面，能否将正负样本完全分开）。



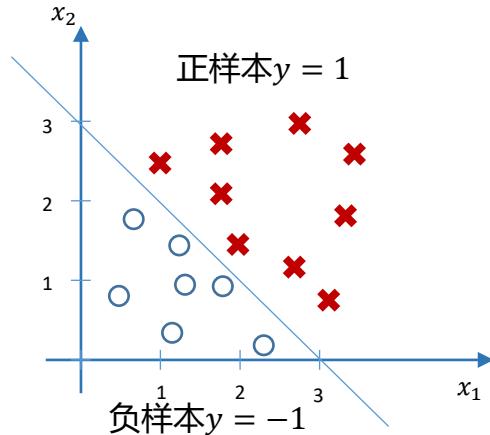
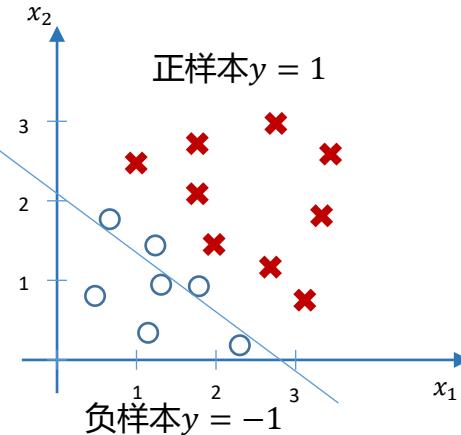
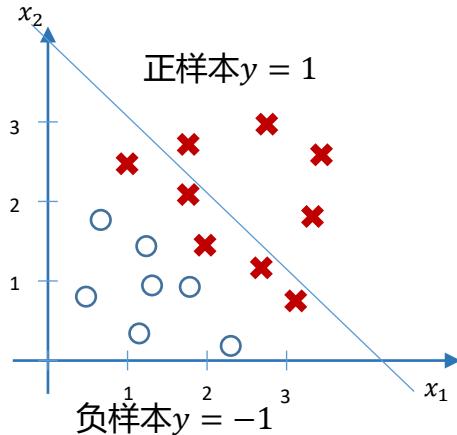


2. 感知机

感知机学习策略

思路1：误分类样本点的总数定义为损失函数。

有何缺点？



注意：逻辑回归中，正负样本y分别为1和0；感知机中，正负样本y分别为1和-1。只是为了方便。



2. 感知机

感知机学习策略

思路2：选择误分类样本点到超平面S的总距离作为损失函数。

超平面S的表达式： $\mathbf{w}^T \mathbf{x}_i + b = 0$

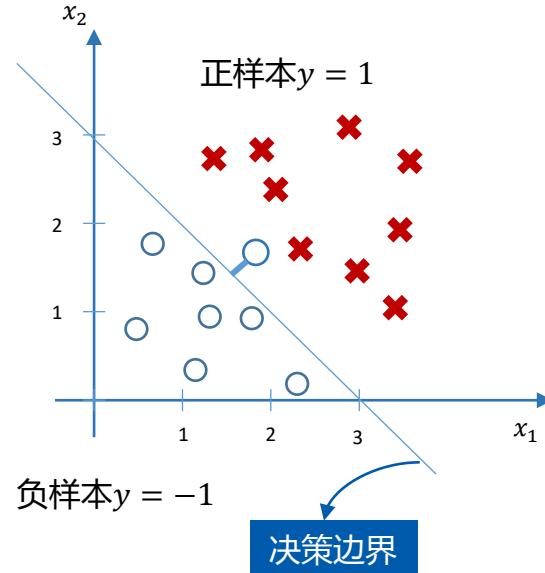
空间中任意一点 \mathbf{x}_i 到超平面S的距离： $\frac{1}{\|\mathbf{w}\|} |\mathbf{w}^T \mathbf{x}_i + b|$

当样本点被误分类时，其 $\mathbf{w}^T \mathbf{x}_i + b$ 与 y_i 异号，即

当 $\mathbf{w}^T \mathbf{x}_i + b > 0$ 时， $y_i = -1$ ，

当 $\mathbf{w}^T \mathbf{x}_i + b < 0$ 时， $y_i = +1$ ，

因此，距离又可以表示成： $-\frac{1}{\|\mathbf{w}\|} y_i (\mathbf{w}^T \mathbf{x}_i + b)$





2. 感知机

感知机学习策略

假设所有误分类样本点的集合是 M ，则总距离表示为：

$$-\frac{1}{\|\mathbf{w}\|} \sum_{x_i \in M} y_i (\mathbf{w}^T \mathbf{x}_i + b)$$

因此，给定训练集合 $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$ ，感知机学习的损失函数定义为：

$$L(\mathbf{w}, b) = - \sum_{x_i \in M} y_i (\mathbf{w}^T \mathbf{x}_i + b)$$

补充：

损失函数中忽略了 $\frac{1}{\|\mathbf{w}\|}$ ，对最终的结果是否会有影响呢？

没有影响，因为感知机的目标是求一个能够将训练集正样本和负样本完全分开的超平面（第17页），也就是说，最终目标是让所有样本被正确分类，此时总距离为0，而 $\frac{1}{\|\mathbf{w}\|}$ 为正，其分子为0即可。



2. 感知机

感知机学习策略

优化目标：求一组参数 \mathbf{w}, b ，使损失函数最小化。

$$\min_{\mathbf{w}, b} L(\mathbf{w}, b) = - \sum_{x_i \in M} y_i (\mathbf{w}^T \mathbf{x}_i + b)$$

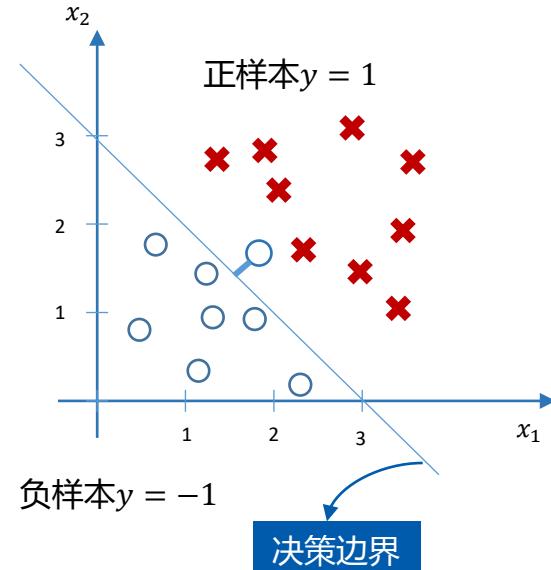
感知机学习算法是误分类驱动的，采用随机梯度下降算法：

$$\frac{\partial L(\mathbf{w}, b)}{\mathbf{w}} = - \sum_{x_i \in M} y_i \mathbf{x}_i \quad \mathbf{w} \leftarrow \mathbf{w} + \eta y_i \mathbf{x}_i$$

$$\frac{\partial L(\mathbf{w}, b)}{b} = - \sum_{x_i \in M} y_i \quad b \leftarrow b + \eta y_i$$

参数的梯度

参数更新策略





2. 感知机

感知机算法

输入：训练数据集合 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$; 学习率 η 。

输出：感知机模型参数 w, b , 以及感知机模型 $f(x) = \text{sign}(w^T x + b)$

1. 选取初值 w_0, b_0 ;
2. 在训练集中选取数据 (x_i, y_i) ;
3. 如果 $y_i(w^T x_i + b) \leq 0$, 更新参数 $w \leftarrow w + \eta y_i x_i \quad b \leftarrow b + \eta y_i$;
4. 跳转到第2步, 继续学习, 直到训练集中没有误分类的点。

思考：对于线性不可分的情形，感知机是否还有用？



2. 感知机

感知机实例

现有如下训练数据集，正样本点 $M_1(3,3)$ 和 $M_2(4,3)$ ，负样本点 $M_3(1,1)$ ，请利用感知机算法求感知机模型 $f(x) = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$ 。

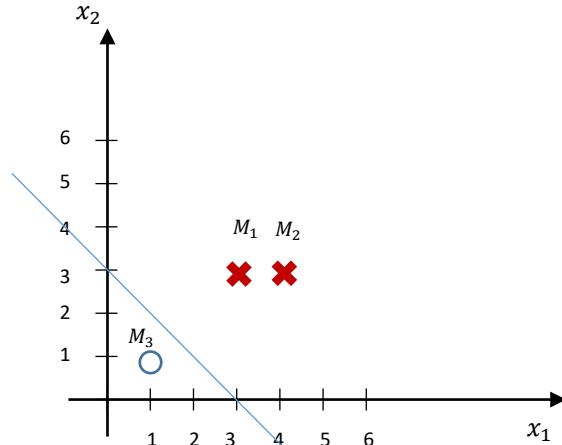
1. 取初值 $\mathbf{w}_0 = \mathbf{0}, b_0 = 0$
2. M_1 对应的感知机模型输出为0，未被正确分类，更新：

$$\mathbf{w}_1 = \mathbf{w}_0 + y_1 \mathbf{x}_{M1} = (3,3) \quad b_1 = b_0 + y_1 = 1$$

3. M_1 和 M_2 已经满足， M_3 对应感知机输出为1，未被正确分类，更新：

$$\mathbf{w}_2 = \mathbf{w}_1 + y_3 \mathbf{x}_{M3} = (2,2) \quad b_2 = b_1 + y_3 = 0$$

4. 重复更新步骤，直至没有误分类点。





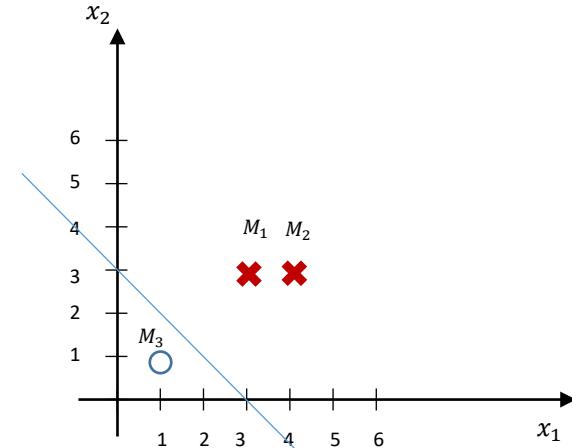
2. 感知机

感知机实例

迭代次数	误分类点	w	b	超平面
0		(0, 0)	0	0
1	M_1	(3, 3)	1	$3x_1 + 3x_2 + 1$
2	M_3	(2, 2)	0	$2x_1 + 2x_2$
3	M_3	(1, 1)	-1	$x_1 + x_2 - 1$
4	M_3	(0, 0)	-2	-2
5	M_1	(3, 3)	-1	$3x_1 + 3x_2 - 1$
6	M_3	(2, 2)	-2	$2x_1 + 2x_2 - 2$
7	M_3	(1, 1)	-3	$x_1 + x_2 - 3$
8	无	(1, 1)	-3	$x_1 + x_2 - 3$

假如在第5次迭代时候，选择的是 M_2 作为误分类点，
那么结果又是什么（见作业）

感知机模型： $f(x) = \text{sign}(x_1 + x_2 - 3)$





2. 感知机

作业题 III

现有如下训练数据集，正样本点 $M_1(3,3)$ 和 $M_2(4,3)$ ，负样本点 $M_3(1,1)$ ，请利用感知机算法求感知机模型 $f(x) = \text{sign}(w^T x + b)$ 。下方表格中已给出了前4步迭代的步骤，第5步迭代中选择 M_2 为误分类点，请补全后续详细的迭代过程。

迭代次数	误分类点	w	b	超平面
0		(0, 0)	0	0
1	M_1	(3, 3)	1	$3x_1 + 3x_2 + 1$
2	M_3	(2, 2)	0	$2 + 2x_2$
3	M_3	(1, 1)	-1	$x_1 + x_2 - 1$
4	M_3	(0, 0)	-2	-2
5	M_2			
...				
N	无			



知识引入



感知机



神经网络



反向传播

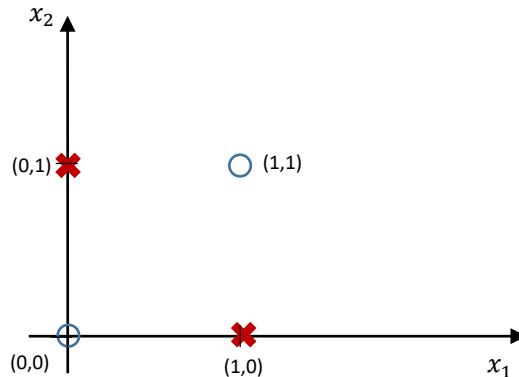


手写数字识别

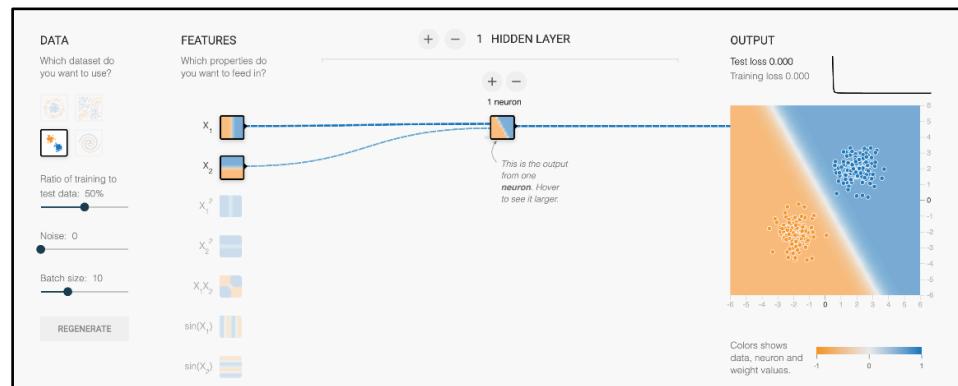


3. 神经网络

非线性问题：在“异或”问题上找不到一条直线能把○和×分开，即这是一个不能用直线分类的问题，该类问题叫做非线性问题。



异或问题在二维平面上的分布



网页演示

补充：异或的运算法则为 $0 \oplus 0 = 0$, $1 \oplus 0 = 1$, $0 \oplus 1 = 1$, $1 \oplus 1 = 0$ (同为0, 异为1)。



3. 神经网络

线性分类器如何解决非线性分类问题呢？

对于非线性分类问题，可以通过增加特征的方式来完成非线性的表达。

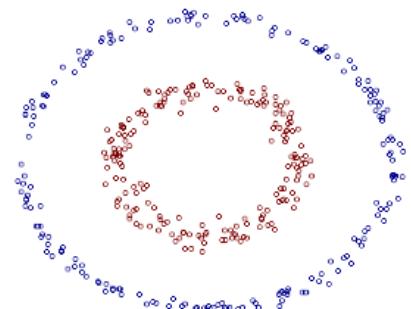
线性分类: $f(x) = g(w_0 + w_1x_1 + w_2x_2)$

非线性分类:

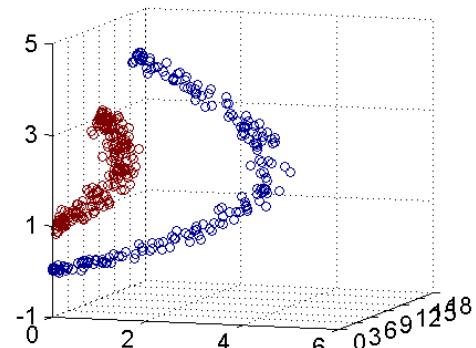
$f(x) = g(w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2 + w_4x_1^2x_2 + w_5x_1x_2^2 + \dots)$

假设有100维特征，那么使用“增加特征”的方式构造的特征，例如 $x_1x_2, x_1x_3, \dots, x_1x_{100}$ 和 $x_1^2x_2, x_1^2x_3, \dots, x_1^2x_{100}$ 等将会很快的达到 n^2 甚至是 n^3 个。

缺点：优化和计算难度极大增加。



低维线性不可分的案例



高维线性可分的案例

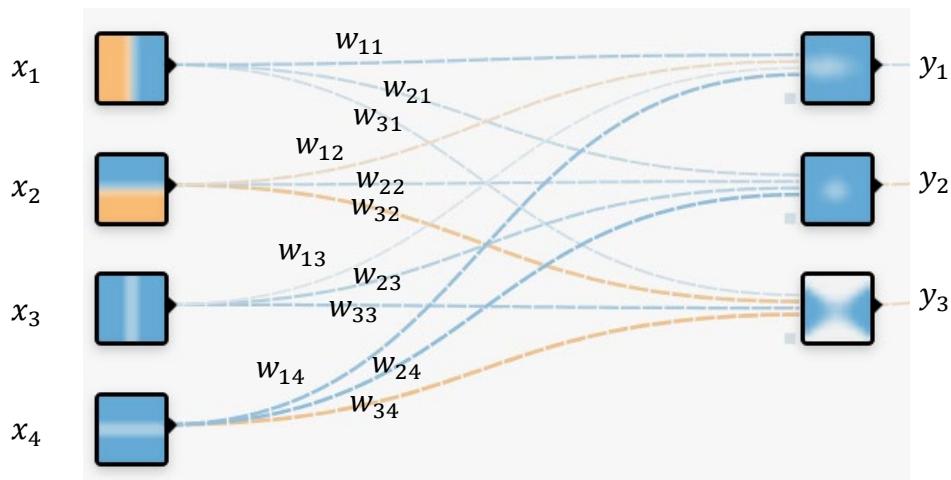


3. 神经网络

神经元的解释

神经网络是由一层一层神经元构建的，那么每一层究竟在干什么，或者是完成什么事情呢？

每层神经元理解： $y = f(Wx + b)$ ，其中 x 是输入向量， y 是输出向量， b 是偏移向量， W 是权重矩阵。每一层就相当于把输入 x 经过简单的变换得到 y 。



$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \end{pmatrix}$$



3. 神经网络

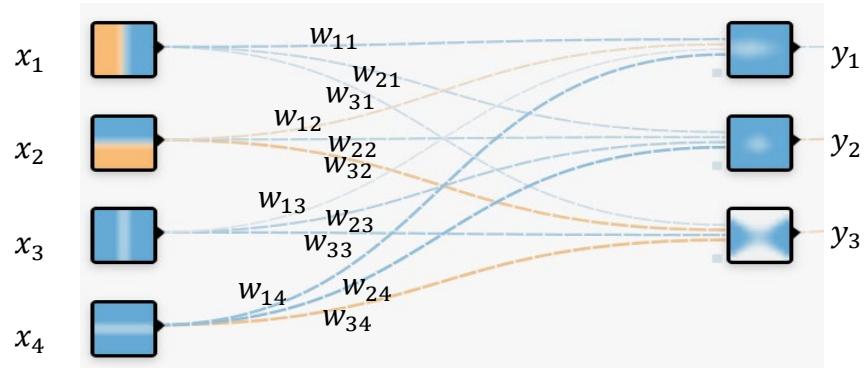
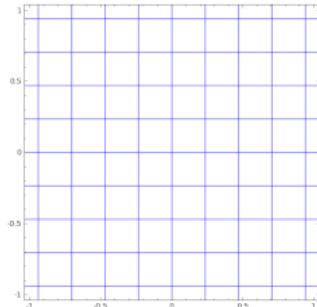
神经元的解释

$$y = f(Wx + b)$$

其中 x 是输入向量， y 是输出向量， b 是偏移向量， W 是权重矩阵。

神经元数学理解：通过如下5种变换，对输入空间（输入向量的集合）进行操作，完成输入空间->输出空间的变换。

- 降维/升维
- 放大/缩小
- 旋转
- 平移
- 弯曲

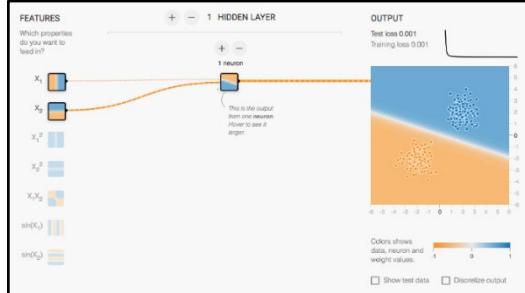




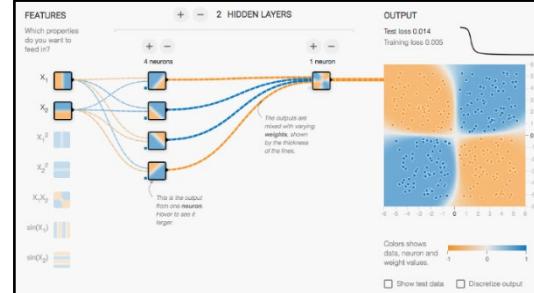
3. 神经网络

神经元的解释

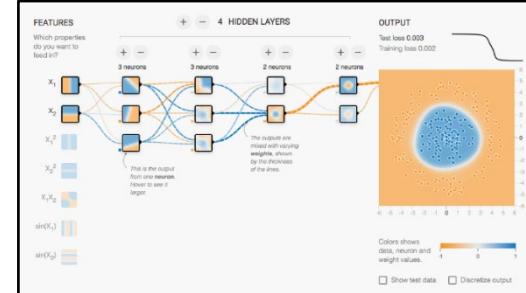
- 少数的神经元，只能做二分类问题；
- 增加神经元个数，可以做更加复杂的分类；
- 增加神经元个数，同时增加神经元的层数，完成更加复杂的非线性分类问题。



(a)单层神经元



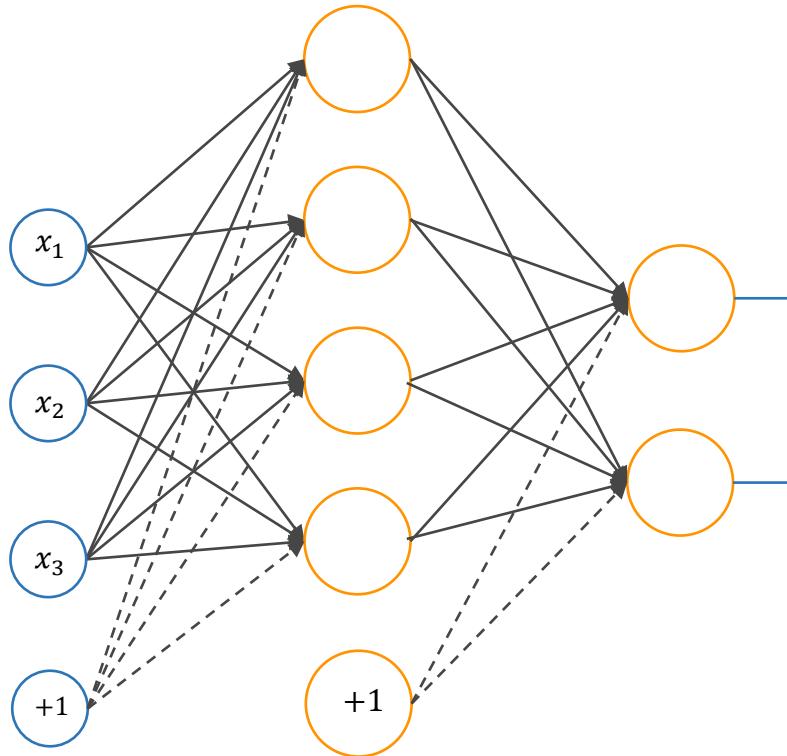
(b)单隐藏层多神经元



(c)多隐藏层多神经元



3. 神经网络—多层神经网络



输入层

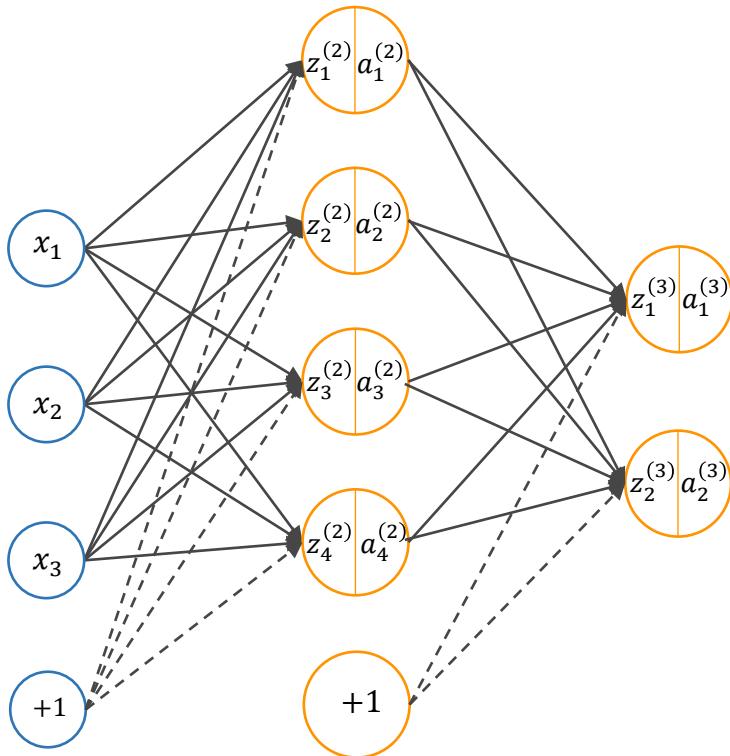
隐藏层

输出层





3. 神经网络—多层神经网络



输入层

隐藏层

输出层

定义符号：

n_l 第 l 层的神经元个数

$\theta_{ij}^{(l)}$ 从 $l-1$ 层第 j 个到 l 层第 i 个神经元连接的权重

$\Theta^{(l)}$ 从 $l-1$ 层到 l 层的权重参数组

$b_i^{(l)}$ 从 $l-1$ 层到 l 层第 i 个神经元连接的偏置

$\mathbf{b}^{(l)}$ 从 $l-1$ 层到 l 层的偏置参数组

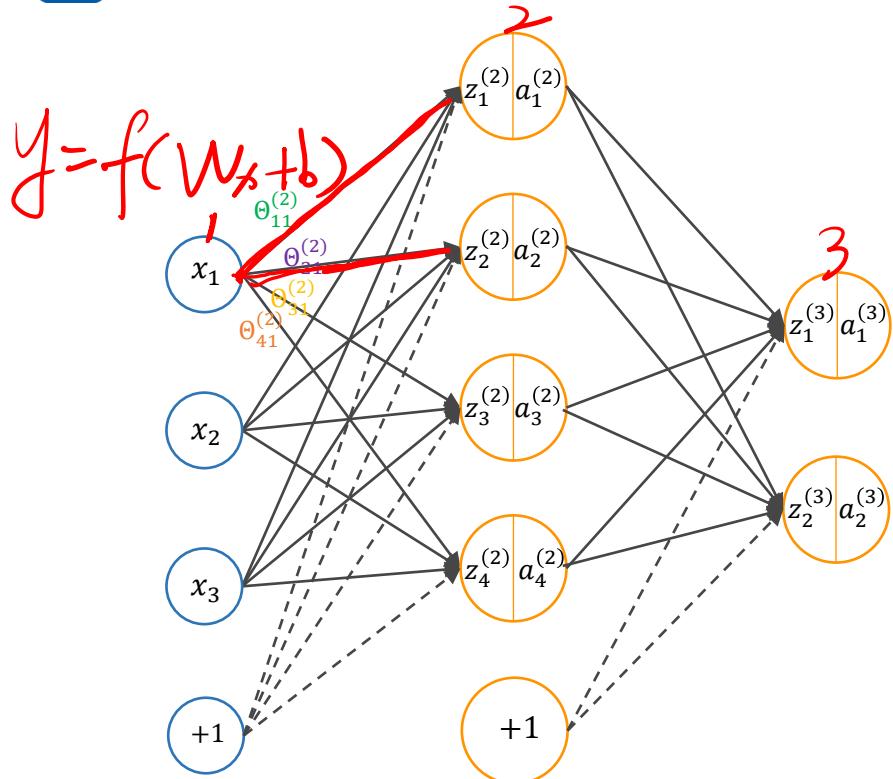
$f(\cdot)$ 神经元的激活函数

$z_i^{(l)}$ 第 l 层的第 i 个神经元的加权和

$a_i^{(l)}$ 第 l 层的第 i 个神经元的输出（经过激励函数）



3. 神经网络—多层神经网络



n_l 第 l 层的神经元个数

$$n_3 = 2, n_2 = 4, n_1 = 3$$

$z_i^{(l)}$ 第 l 层的第 i 个神经元的加权和

$a_i^{(l)}$ 第 l 层的第 i 个神经元的输出 (经过激励函数)

$\theta_{ij}^{(l)}$ 从 $l-1$ 层第 j 个到 l 层第 i 个神经元连接的权重

$f(\cdot)$ 神经元的激活函数 $a_i^{(l)} = f(z_i^{(l)})$

$$z_1^{(2)} = \theta_{11}^{(2)} x_1 + \theta_{12}^{(2)} x_2 + \theta_{13}^{(2)} x_3 + b_1^{(2)}$$

$$a_1^{(2)} = f(z_1^{(2)}) = f(\theta_{11}^{(2)} x_1 + \theta_{12}^{(2)} x_2 + \theta_{13}^{(2)} x_3 + b_1^{(2)})$$

$$z_2^{(2)} = \theta_{21}^{(2)} x_1 + \theta_{22}^{(2)} x_2 + \theta_{23}^{(2)} x_3 + b_2^{(2)}$$

$$a_2^{(2)} = f(z_2^{(2)}) = f(\theta_{21}^{(2)} x_1 + \theta_{22}^{(2)} x_2 + \theta_{23}^{(2)} x_3 + b_2^{(2)})$$

$$z_3^{(2)} = \theta_{31}^{(2)} x_1 + \theta_{32}^{(2)} x_2 + \theta_{33}^{(2)} x_3 + b_3^{(2)}$$

$$a_3^{(2)} = f(z_3^{(2)}) = f(\theta_{31}^{(2)} x_1 + \theta_{32}^{(2)} x_2 + \theta_{33}^{(2)} x_3 + b_3^{(2)})$$

$$z_4^{(2)} = \theta_{41}^{(2)} x_1 + \theta_{42}^{(2)} x_2 + \theta_{43}^{(2)} x_3 + b_4^{(2)}$$

$$a_4^{(2)} = f(z_4^{(2)}) = f(\theta_{41}^{(2)} x_1 + \theta_{42}^{(2)} x_2 + \theta_{43}^{(2)} x_3 + b_4^{(2)})$$

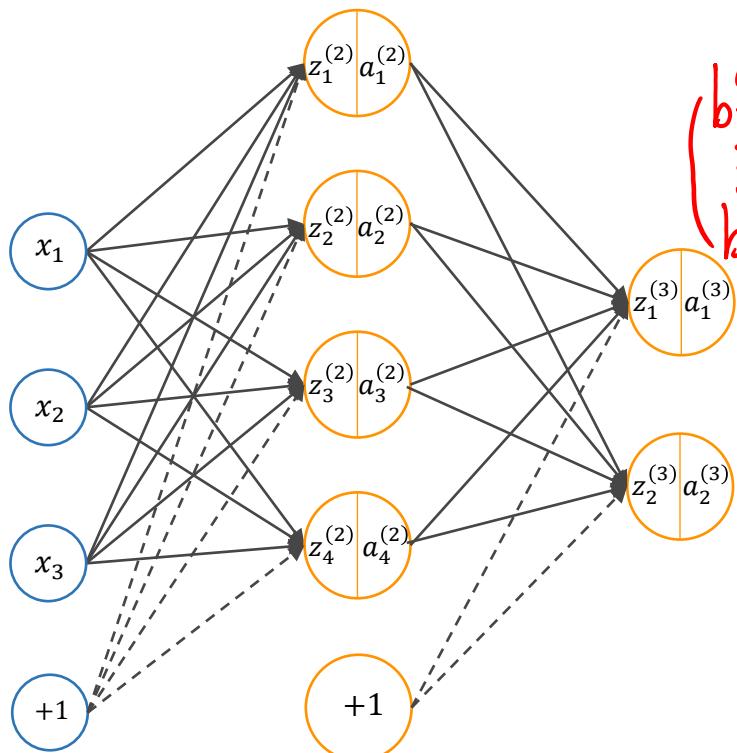
输入层

隐藏层

输出层



3. 神经网络—多层神经网络



$$z^{(l)} = \begin{pmatrix} z_1^{(l)} \\ z_2^{(l)} \\ \vdots \\ z_n^{(l)} \end{pmatrix}$$

第 l 层神经元的加权和
第 l 层神经元输出 (经过激励函数)
 $\theta^{(l)}$ 从 $l-1$ 层到 l 层的权重参数组 $n_l \ n_{l-1}$
 $b^{(l)}$ 从 $l-1$ 层到 l 层的偏置参数组

\mathbb{R}^{n_l}
 \mathbb{R}^{n_l}
 $\mathbb{R}^{n_l \times n_{l-1}}$
 \mathbb{R}^{n_l}

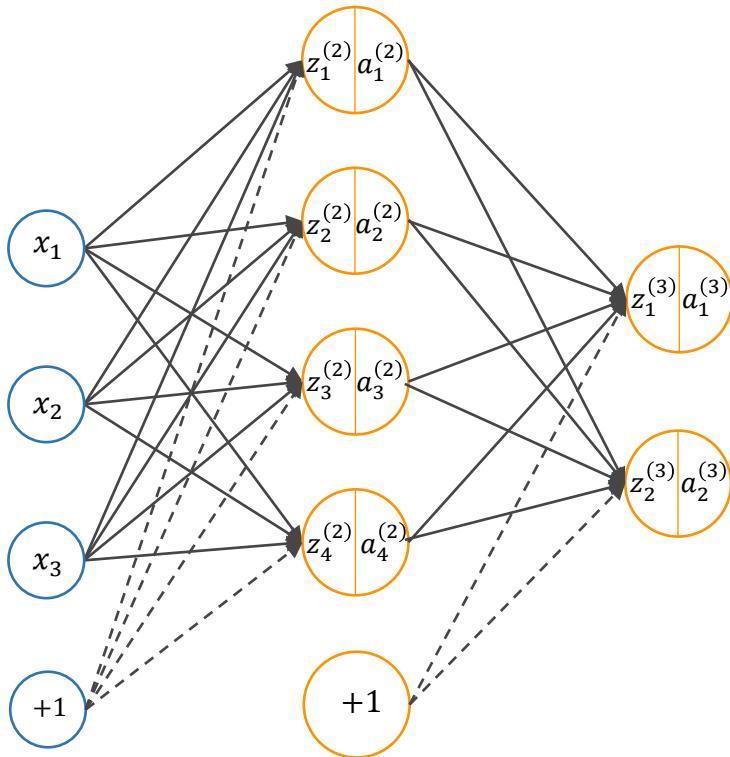
总结: 第 l ($2 \leq l \leq L$) 层神经元的状态及激活值的向量表示形式为

$$z^{(l)} = \underbrace{\theta^{(l)} a^{(l-1)} + b^{(l)}}_{l=2} \rightarrow \theta^{(l)} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + b^{(l)}$$

$$\left\{ \begin{array}{l} z_1^{(2)} = \theta_{11}^{(2)} x_1 + \theta_{12}^{(2)} x_2 + \theta_{13}^{(2)} x_3 + b_1^{(2)} \\ z_2^{(2)} = \theta_{21}^{(2)} x_1 + \theta_{22}^{(2)} x_2 + \theta_{23}^{(2)} x_3 + b_2^{(2)} \\ z_3^{(2)} = \theta_{31}^{(2)} x_1 + \theta_{32}^{(2)} x_2 + \theta_{33}^{(2)} x_3 + b_3^{(2)} \\ z_4^{(2)} = \theta_{41}^{(2)} x_1 + \theta_{42}^{(2)} x_2 + \theta_{43}^{(2)} x_3 + b_4^{(2)} \end{array} \right.$$



3. 神经网络—多层神经网络



$\mathbf{z}^{(l)}$ 第 l 层神经元的加权和

\mathbb{R}^{n_l}

$\mathbf{a}^{(l)}$ 第 l 层神经元输出（经过激励函数）

\mathbb{R}^{n_l}

$\boldsymbol{\theta}^{(l)}$ 从 $l-1$ 层到 l 层的权重参数组

$\mathbb{R}^{n_l \times n_{l-1}}$

$\mathbf{b}^{(l)}$ 从 $l-1$ 层到 l 层的偏置参数组

\mathbb{R}^{n_l}

总结

第 l ($2 \leq l \leq L$) 层神经元的状态及激活值的向量表示形式为：

$$\underline{\mathbf{z}^{(l)} = \boldsymbol{\theta}^{(l)} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}}$$

$$\mathbf{a}^{(l)} = f(\mathbf{z}^{(l)})$$

$$\underline{\mathbf{a}^{(1)} = \mathbf{x}}$$

前向传播过程：

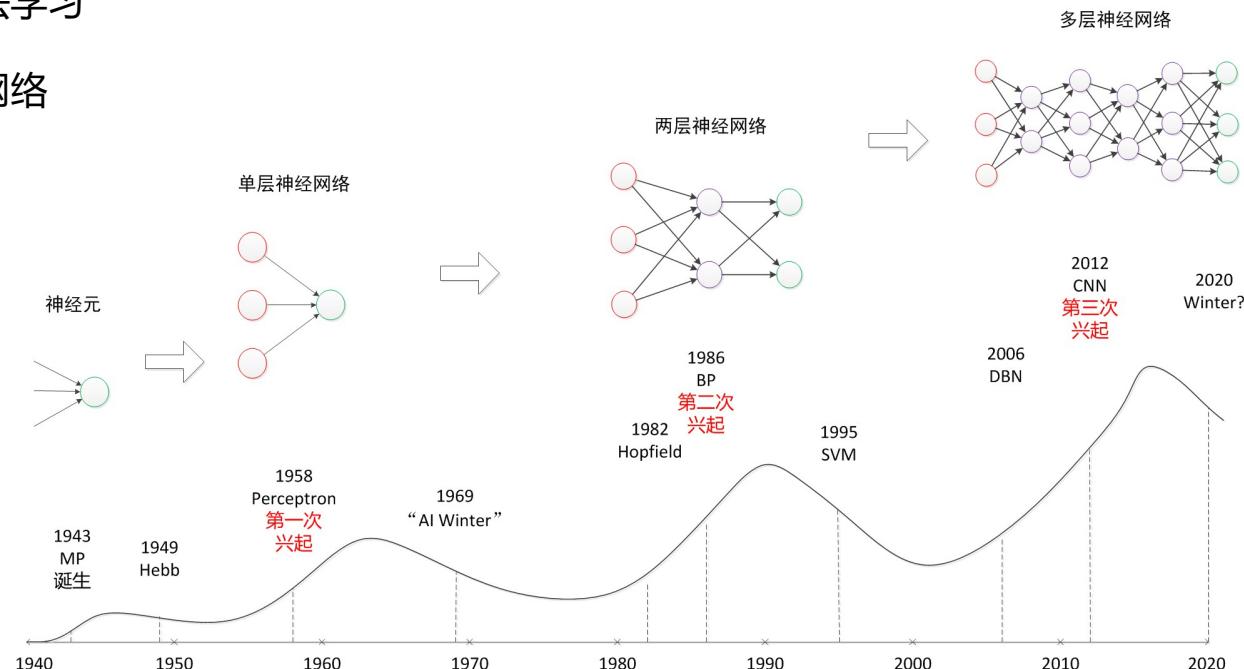
$$\underline{\mathbf{x}} = \underline{\mathbf{a}^{(1)}} \rightarrow \underline{\mathbf{z}^{(2)}} \rightarrow \underline{\mathbf{a}^{(2)}} \dots \rightarrow \underline{\mathbf{a}^{(L-1)}} \rightarrow \underline{\mathbf{z}^{(L)}} \rightarrow \underline{\mathbf{a}^{(L)}}$$



3. 神经网络

神经网络的发展与比较

- 从人工指定权值到算法学习
- 从感知机到多层神经网络
- 神经网络的起起落落

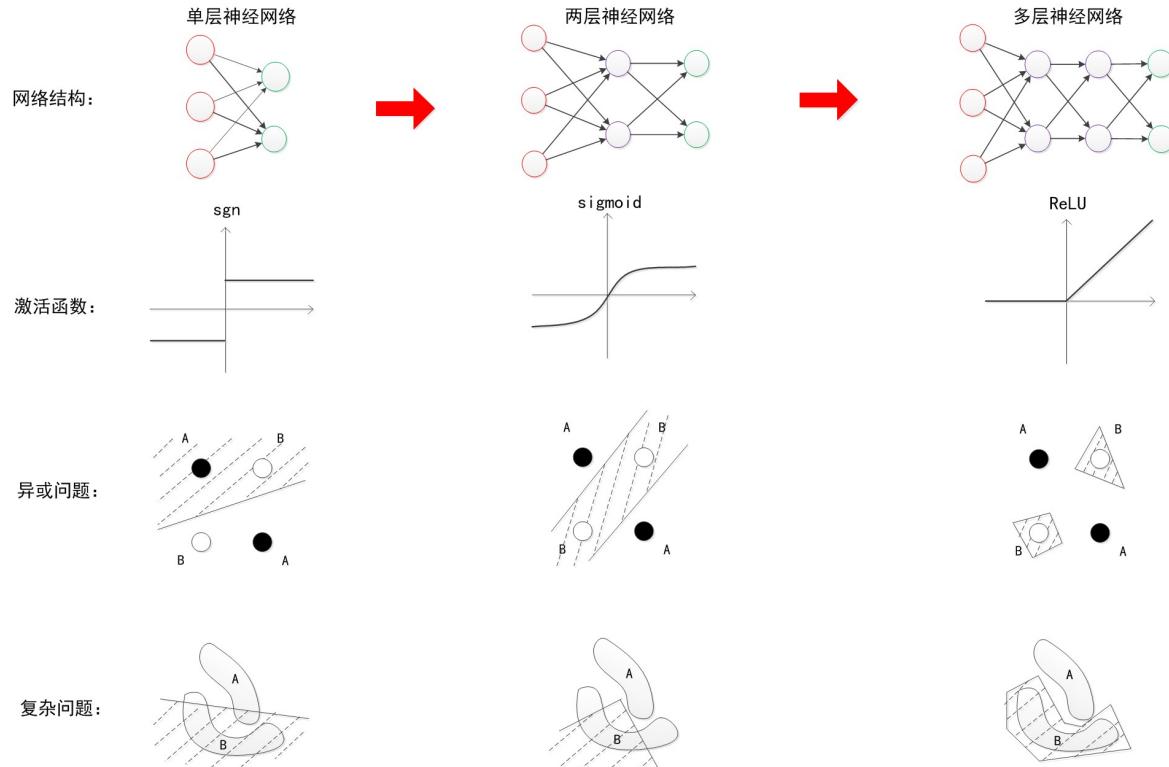




3. 神经网络

神经网络的发展与比较

- 激活函数的改变
- 决策边界的改变
- 解决问题能力改变





课后作业

作业 IV

为什么多层神经网络可以拟合任意函数？其与浅层宽网络的区别是什么？



知识引入



感知机



神经网络



反向传播



手写数字识别



4. 反向传播

梯度下降

前向传播算法对于未训练完全的网络，网络的输出值与真实值之间存在差异，把这个差异叫做网络的**误差**。误差的传播方向与前向传播方向正好相反。通过反向传播，利用梯度下降对每一个参数进行更新。

训练数据

对于第*i*个训练数据来说，其代价函数为：

$$\left\{ \left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)} \right), \left(\mathbf{x}^{(2)}, \mathbf{y}^{(2)} \right), \dots, \left(\mathbf{x}^{(i)}, \mathbf{y}^{(i)} \right), \dots, \left(\mathbf{x}^{(N)}, \mathbf{y}^{(N)} \right) \right\} \quad E = \frac{1}{2} \left\| \mathbf{y} - \mathbf{a}^{(L)} \right\|^2 = \frac{1}{2} \sum_{k=1}^{n_L} \left(y_k - a_k^{(L)} \right)^2$$

第*i*个训练样本的输入： $\mathbf{x}^{(i)} = \left(x_1^{(i)}, \dots, x_{n_1}^{(i)} \right)^T$

对于所有训练数据，其平均代价函数为：

第*i*个训练样本的输出： $\mathbf{y}^{(i)} = \left(y_1^{(i)}, \dots, y_{n_L}^{(i)} \right)^T$

$$E_{total} = \frac{1}{N} \sum_{i=1}^N E$$



4. 反向传播

梯度下降

梯度下降法（又称为“批量梯度下降法”）， $\theta^{(l)}$ 与 $b^{(l)}$ 的参数更新方式为：

$$\begin{aligned}\theta^{(l)} &= \theta^{(l)} - \mu \frac{\partial E_{total}}{\partial \theta^{(l)}} \\ &= \theta^{(l)} - \frac{\mu}{N} \sum_{i=1}^N \frac{\partial E}{\partial \theta^{(l)}}\end{aligned}$$

$$\begin{aligned}b^{(l)} &= b^{(l)} - \mu \frac{\partial E_{total}}{\partial b^{(l)}} \\ &= b^{(l)} - \frac{\mu}{N} \sum_{i=1}^N \frac{\partial E}{\partial b^{(l)}}\end{aligned}$$

注意：

- (1) $\theta^{(l)}$ 有可能是向量，也有可能是矩阵，主要取决于神经网络每层神经元的个数； $b^{(l)}$ 是向量。
- (2) $\theta^{(l)}$ 与 $b^{(l)}$ 的学习率可以不一样。

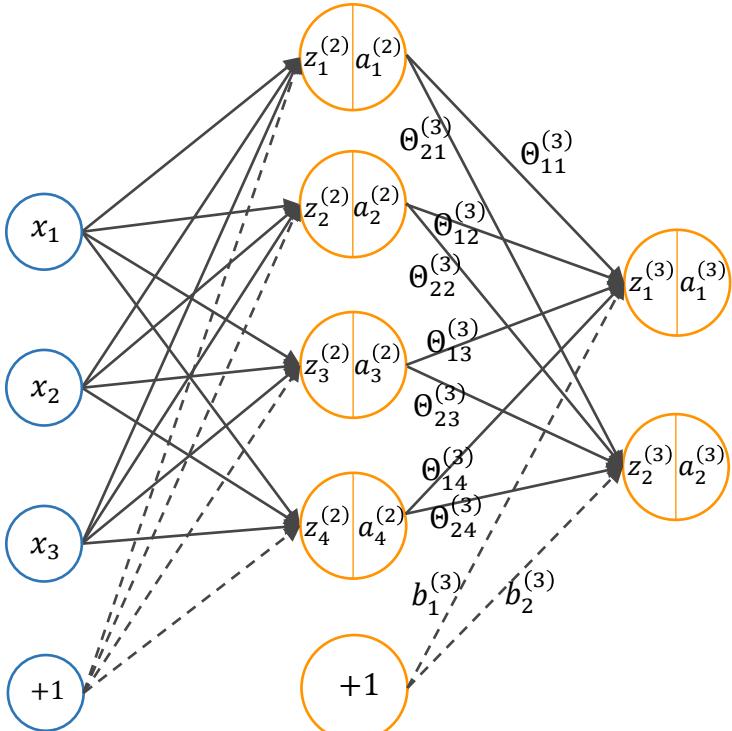


4. 反向传播

梯度下降

任意一个训练样本，将误差表示（代价函数）在输出层展开，则能得到：

$$\begin{aligned} E &= \frac{1}{2} \|\mathbf{y} - \mathbf{a}^{(3)}\|^2 = \frac{1}{2} \sum_{k=1}^2 (y_k - a_k^{(3)})^2 \\ &= \frac{1}{2} \left((y_1 - a_1^{(3)})^2 + (y_2 - a_2^{(3)})^2 \right) \\ &= \frac{1}{2} \left((y_1 - f(z_1^{(3)}))^2 + (y_2 - f(z_2^{(3)}))^2 \right) \\ &= \frac{1}{2} \left[\left(y_1 - f \left(\Theta_{11}^{(3)} a_1^{(2)} + \Theta_{12}^{(3)} a_2^{(2)} + \Theta_{13}^{(3)} a_3^{(2)} + \Theta_{14}^{(3)} a_4^{(2)} + b_1^{(3)} \right) \right)^2 + \left(y_2 - f \left(\Theta_{21}^{(3)} a_1^{(2)} + \Theta_{22}^{(3)} a_2^{(2)} + \Theta_{23}^{(3)} a_3^{(2)} + \Theta_{24}^{(3)} a_4^{(2)} + b_2^{(3)} \right) \right)^2 \right] \end{aligned}$$





4. 反向传播

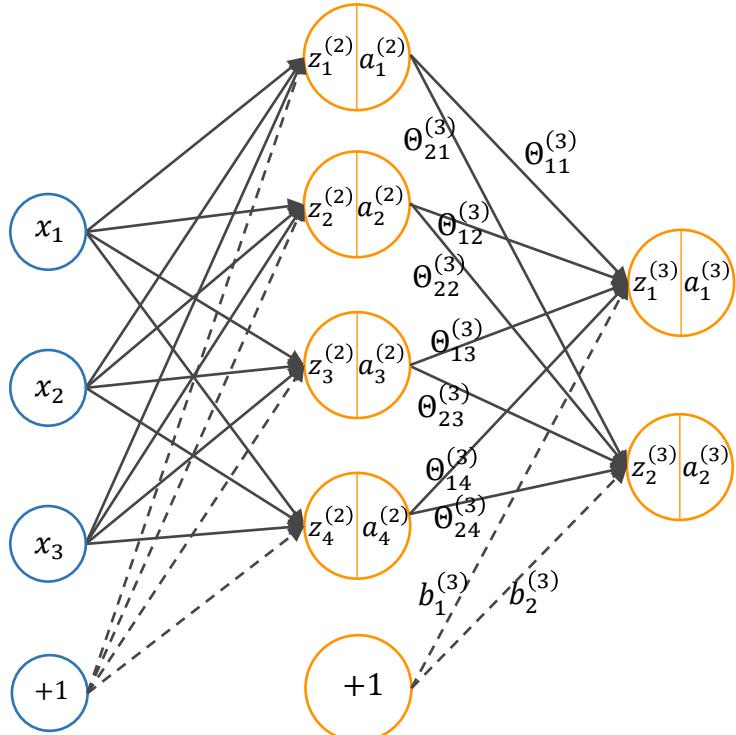
梯度下降-输出层权重更新

对输出层权重求偏导数，得到：

$$\begin{aligned}\frac{\partial E}{\partial \Theta_{11}^{(3)}} &= \frac{\partial E}{\partial a_1^{(3)}} \frac{\partial a_1^{(3)}}{\partial z_1^{(3)}} \frac{\partial z_1^{(3)}}{\partial \Theta_{11}^{(3)}} \\ &= \frac{1}{2} \cdot 2 \left(y_1 - a_1^{(3)} \right) \left(-\frac{\partial a_1^{(3)}}{\partial \Theta_{11}^{(3)}} \right) \\ &= -\left(y_1 - a_1^{(3)} \right) f' \left(z_1^{(3)} \right) \frac{\partial z_1^{(3)}}{\partial \Theta_{11}^{(3)}} \\ &= -\left(y_1 - a_1^{(3)} \right) f' \left(z_1^{(3)} \right) a_1^{(2)}\end{aligned}$$

令： $\delta_i^{(3)} = \frac{\partial E}{\partial a_i^{(3)}} \frac{\partial a_i^{(3)}}{\partial z_i^{(3)}}$ 则： $\frac{\partial E}{\partial \Theta_{11}^{(3)}} = \delta_1^{(3)} a_1^{(2)}$

$$\begin{aligned}E &= \frac{1}{2} \|y - a^{(3)}\|^2 = \frac{1}{2} \left((y_1 - a_1^{(3)})^2 + (y_2 - a_2^{(3)})^2 \right) \\ &= \frac{1}{2} \left[\left(y_1 - f \left(\Theta_{11}^{(3)} a_1^{(2)} + \Theta_{12}^{(3)} a_2^{(2)} + \Theta_{13}^{(3)} a_3^{(2)} + \Theta_{14}^{(3)} a_4^{(2)} + b_1^{(3)} \right) \right)^2 + \right. \\ &\quad \left. \left(y_2 - f \left(\Theta_{21}^{(3)} a_1^{(2)} + \Theta_{22}^{(3)} a_2^{(2)} + \Theta_{23}^{(3)} a_3^{(2)} + \Theta_{24}^{(3)} a_4^{(2)} + b_2^{(3)} \right) \right)^2 \right]\end{aligned}$$





4. 反向传播

梯度下降-输出层权重更新

对于输出层其他的权重参数，同样可得：

$$\delta_1^{(3)} = - \left(y_1 - a_1^{(3)} \right) f' \left(z_1^{(3)} \right)$$

$$\delta_2^{(3)} = - \left(y_2 - a_2^{(3)} \right) f' \left(z_2^{(3)} \right)$$

$$\frac{\partial E}{\partial \Theta_{11}^{(3)}} = \delta_1^{(3)} a_1^{(2)}$$

$$\frac{\partial E}{\partial \Theta_{21}^{(3)}} = \delta_2^{(3)} a_1^{(2)}$$

$$\frac{\partial E}{\partial \Theta_{12}^{(3)}} = \delta_1^{(3)} a_2^{(2)}$$

$$\frac{\partial E}{\partial \Theta_{22}^{(3)}} = \delta_2^{(3)} a_2^{(2)}$$

$$\frac{\partial E}{\partial \Theta_{13}^{(3)}} = \delta_1^{(3)} a_3^{(2)}$$

$$\frac{\partial E}{\partial \Theta_{23}^{(3)}} = \delta_2^{(3)} a_3^{(2)}$$

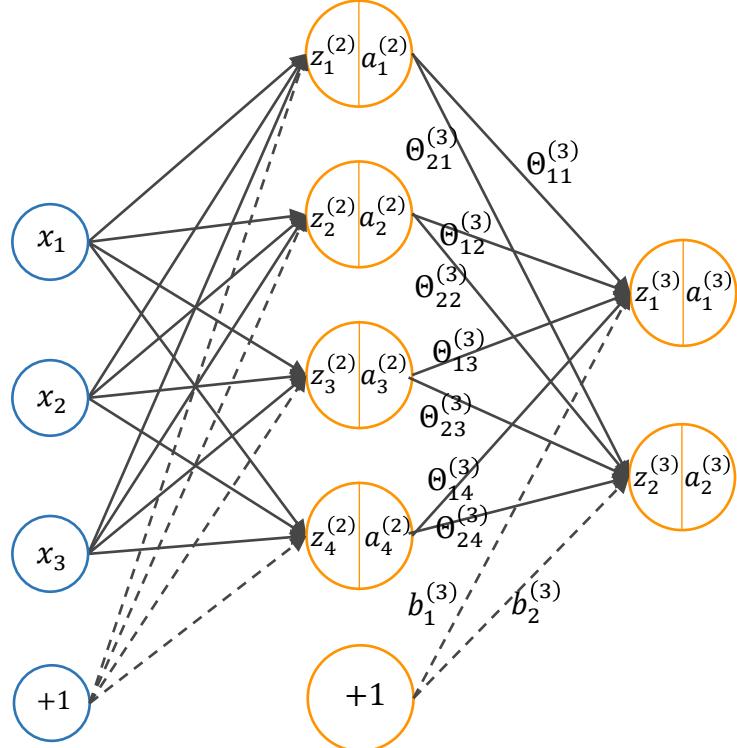
$$\frac{\partial E}{\partial \Theta_{14}^{(3)}} = \delta_1^{(3)} a_4^{(2)}$$

$$\frac{\partial E}{\partial \Theta_{24}^{(3)}} = \delta_2^{(3)} a_4^{(2)}$$

$$E = \frac{1}{2} \|y - a^{(3)}\|^2 = \frac{1}{2} \left((y_1 - a_1^{(3)})^2 + (y_2 - a_2^{(3)})^2 \right)$$

$$= \frac{1}{2} \left[\left(y_1 - f \left(\Theta_{11}^{(3)} a_1^{(2)} + \Theta_{12}^{(3)} a_2^{(2)} + \Theta_{13}^{(3)} a_3^{(2)} + \Theta_{14}^{(3)} a_4^{(2)} + b_1^{(3)} \right) \right)^2 + \right.$$

$$\left. \left(y_2 - f \left(\Theta_{21}^{(3)} a_1^{(2)} + \Theta_{22}^{(3)} a_2^{(2)} + \Theta_{23}^{(3)} a_3^{(2)} + \Theta_{24}^{(3)} a_4^{(2)} + b_2^{(3)} \right) \right)^2 \right]$$





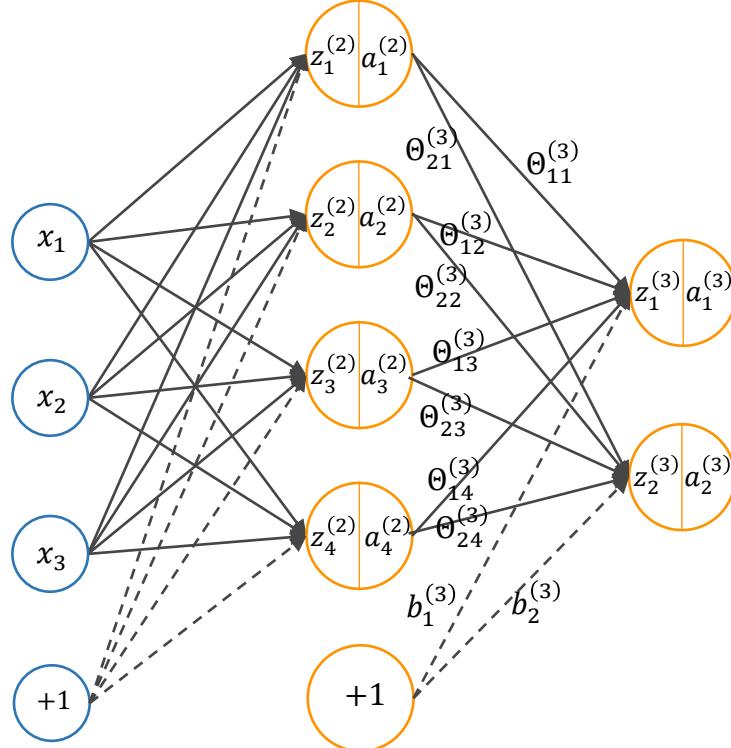
4. 反向传播

梯度下降-输出层权重更新

$$\delta_2^{(3)} = - \left(y_2 - a_2^{(3)} \right) f' \left(z_2^{(3)} \right)$$

手推 $\frac{\partial E}{\partial \Theta_{23}^{(3)}} = \delta_2^{(3)} a_3^{(2)}$

$$E = \frac{1}{2} \|y - a^{(3)}\|^2 = \frac{1}{2} \left((y_1 - a_1^{(3)})^2 + (y_2 - a_2^{(3)})^2 \right)$$
$$= \frac{1}{2} \left[\left(y_1 - f \left(\Theta_{11}^{(3)} a_1^{(2)} + \Theta_{12}^{(3)} a_2^{(2)} + \Theta_{13}^{(3)} a_3^{(2)} + \Theta_{14}^{(3)} a_4^{(2)} + b_1^{(3)} \right) \right)^2 + \right.$$
$$\left. \left(y_2 - f \left(\Theta_{21}^{(3)} a_1^{(2)} + \Theta_{22}^{(3)} a_2^{(2)} + \Theta_{23}^{(3)} a_3^{(2)} + \Theta_{24}^{(3)} a_4^{(2)} + b_2^{(3)} \right) \right)^2 \right]$$





4. 反向传播

梯度下降-输出层权重更新

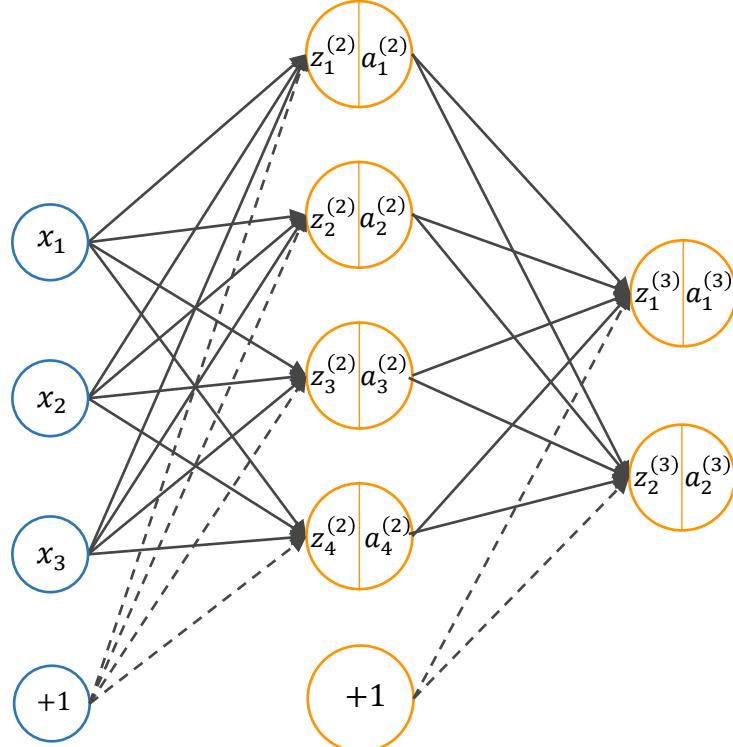
$$\delta_i^{(3)} = \frac{\partial E}{\partial z_i^{(3)}} = \frac{\partial E}{\partial a_i^{(3)}} \frac{\partial a_i^{(3)}}{\partial z_i^{(3)}}$$

$$\delta_i^{(2)} = \frac{\partial E}{\partial z_i^{(2)}} = \frac{\partial E}{\partial a_i^{(2)}} \frac{\partial a_i^{(2)}}{\partial z_i^{(2)}}$$

为什么引入 $\delta_i^{(l)}$?

- 简化 $\frac{\partial E_{(i)}}{\partial \Theta^{(l)}} \frac{\partial E_{(i)}}{\partial b^{(l)}}$ 的表达形式
- 通过 $\delta_i^{(l+1)}$ 来计算 $\delta_i^{(l)}$ (误差的反向传播)

$$\begin{aligned} E &= \frac{1}{2} \|y - a^{(3)}\|^2 = \frac{1}{2} \left((y_1 - a_1^{(3)})^2 + (y_2 - a_2^{(3)})^2 \right) \\ &= \frac{1}{2} \left[\left(y_1 - f \left(\Theta_{11}^{(3)} a_1^{(2)} + \Theta_{12}^{(3)} a_2^{(2)} + \Theta_{13}^{(3)} a_3^{(2)} + \Theta_{14}^{(3)} a_4^{(2)} + b_1^{(3)} \right) \right)^2 + \right. \\ &\quad \left. \left(y_2 - f \left(\Theta_{21}^{(3)} a_1^{(2)} + \Theta_{22}^{(3)} a_2^{(2)} + \Theta_{23}^{(3)} a_3^{(2)} + \Theta_{24}^{(3)} a_4^{(2)} + b_2^{(3)} \right) \right)^2 \right] \end{aligned}$$





4. 反向传播

梯度下降-输出层权重更新

$x = \mathbf{a}^{(1)} \rightarrow \mathbf{z}^{(2)} \rightarrow \mathbf{a}^{(2)} \dots \rightarrow \boxed{\mathbf{a}^{(L-1)}} \rightarrow \mathbf{z}^{(L)} \rightarrow \mathbf{a}^{(L)}$ 转化为向量形式：

推广到一般情况，假设网络有 L 层，则：

$$\delta_i^{(L)} = - \left(y_i - a_i^{(L)} \right) f' \left(z_i^{(L)} \right) \quad (1 \leq i \leq n_L)$$

$$\frac{\partial E}{\partial \Theta_{ij}^{(L)}} = \delta_i^{(L)} a_j^{(L-1)} \quad (1 \leq i \leq n_L, 1 \leq j \leq n_{L-1})$$

$$\boldsymbol{\delta}^{(L)} = - (\mathbf{y} - \mathbf{a}^{(L)}) \odot f' (\mathbf{z}^{(L)})$$

$$\nabla_{\Theta^{(L)}} E = \boldsymbol{\delta}^{(L)} (\mathbf{a}^{(L-1)})^T$$



4. 反向传播

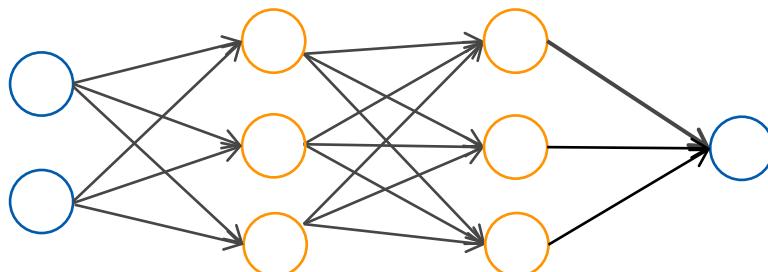
梯度下降-隐藏层权重更新

输出层误差

隐藏层误差

误差反向传播

$$\begin{aligned}\frac{\partial E}{\partial \theta_{ij}^{(l)}} &= \frac{\partial E}{\partial z_i^{(l)}} \frac{\partial z_i^{(l)}}{\partial \theta_{ij}^{(l)}} \\ &= \delta_i^{(l)} \frac{\partial z_i^{(l)}}{\partial \theta_{ij}^{(l)}} \\ &= \delta_i^{(l)} a_j^{(l-1)}\end{aligned}$$



$$\begin{aligned}\delta_i^{(l)} &\equiv \frac{\partial E}{\partial z_i^{(l)}} \\ &= \sum_{j=1}^{n_{l+1}} \frac{\partial E}{\partial z_j^{(l+1)}} \frac{\partial z_j^{(l+1)}}{\partial z_i^{(l)}} \\ &= \sum_{j=1}^{n_{l+1}} \delta_j^{(l+1)} \frac{\partial z_j^{(l+1)}}{\partial z_i^{(l)}}\end{aligned}$$



4. 反向传播

梯度下降-隐藏层权重更新

$$\begin{aligned}\delta_i^{(l)} &= \sum_{j=1}^{n_{l+1}} \delta_j^{(l+1)} \frac{\partial z_j^{(l+1)}}{\partial z_i^{(l)}} \\ &= \sum_{j=1}^{n_{l+1}} \delta_j^{(l+1)} \frac{\partial z_j^{(l+1)}}{\partial a_i^{(l)}} \frac{\partial a_i^{(l)}}{\partial z_i^{(l)}} \\ &= \sum_{j=1}^{n_{l+1}} \delta_j^{(l+1)} \Theta_{ji}^{(l+1)} \frac{\partial a_i^{(l)}}{\partial z_i^{(l)}} \\ &= \left(\sum_{j=1}^{n_{l+1}} \delta_j^{(l+1)} \Theta_{ji}^{(l+1)} \right) f'(z_i^{(l)})\end{aligned}$$

BP传播算法的核心公式：利用 $l + 1$ 层的 $\delta_i^{(l+1)}$ 来计算 l 层的 $\delta_i^{(l)}$ 。

$$\delta^{(l)} = \left((\boldsymbol{\theta}^{(l+1)})^T \boldsymbol{\delta}^{(l+1)} \right) \odot f'(\mathbf{z}^{(l)})$$



4. 反向传播

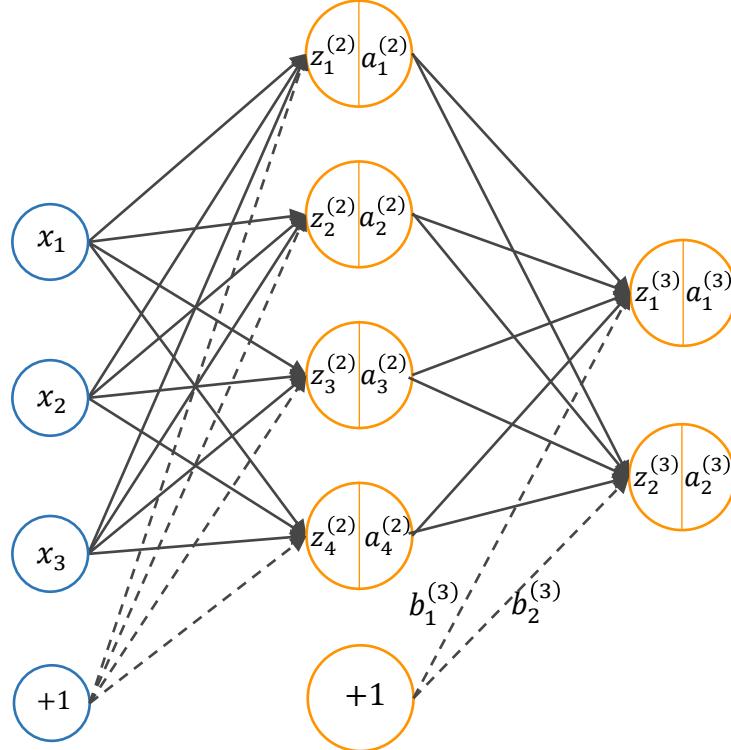
梯度下降-偏置更新

对输出层偏置求偏导数，得到：

$$\frac{\partial E}{\partial b_1^{(3)}} = \frac{\partial E}{\partial a_1^{(3)}} \frac{\partial a_1^{(3)}}{\partial z_1^{(3)}} \frac{\partial z_1^{(3)}}{\partial b_1^{(3)}} = \delta_1^{(3)}$$

$$\frac{\partial E}{\partial b_2^{(3)}} = \frac{\partial E}{\partial a_2^{(3)}} \frac{\partial a_2^{(3)}}{\partial z_2^{(3)}} \frac{\partial z_2^{(3)}}{\partial b_2^{(3)}} = \delta_2^{(3)}$$

$$\begin{aligned} E &= \frac{1}{2} \|y - a^{(3)}\|^2 = \frac{1}{2} \left((y_1 - a_1^{(3)})^2 + (y_2 - a_2^{(3)})^2 \right) \\ &= \frac{1}{2} \left[\left(y_1 - f \left(\Theta_{11}^{(3)} a_1^{(2)} + \Theta_{12}^{(3)} a_2^{(2)} + \Theta_{13}^{(3)} a_3^{(2)} + \Theta_{14}^{(3)} a_4^{(2)} + b_1^{(3)} \right) \right)^2 + \right. \\ &\quad \left. \left(y_2 - f \left(\Theta_{21}^{(3)} a_1^{(2)} + \Theta_{22}^{(3)} a_2^{(2)} + \Theta_{23}^{(3)} a_3^{(2)} + \Theta_{24}^{(3)} a_4^{(2)} + b_2^{(3)} \right) \right)^2 \right] \end{aligned}$$





4. 反向传播

梯度下降-偏置更新

对隐藏层偏置求偏导数，得到：

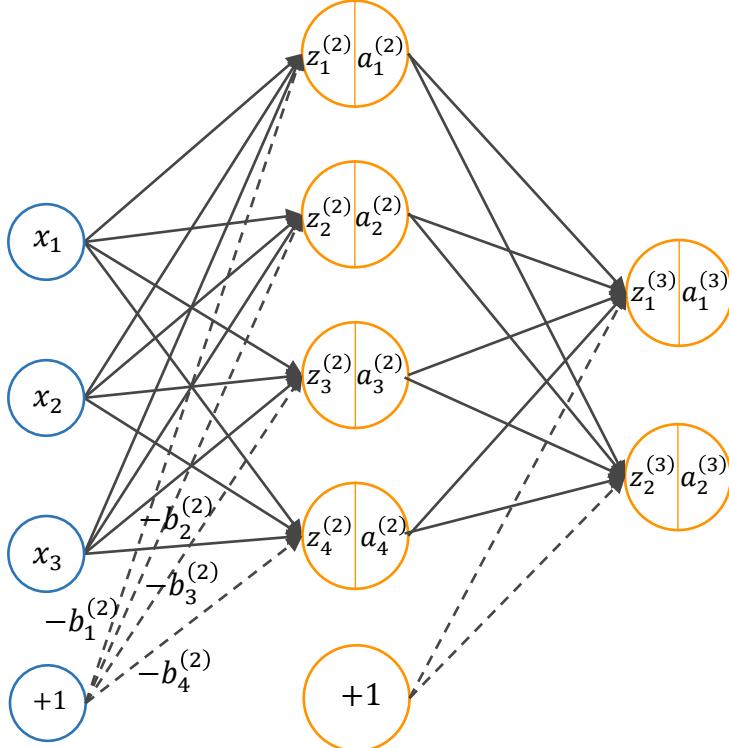
$$\frac{\partial E}{\partial b_1^{(2)}} = \frac{\partial E}{\partial a_1^{(3)}} \frac{\partial a_1^{(3)}}{\partial z_1^{(3)}} \frac{\partial z_1^{(3)}}{\partial z_1^{(2)}} \frac{z_1^{(2)}}{b_1^{(2)}} = \delta_1^{(2)}$$

$$\frac{\partial E}{\partial b_2^{(2)}} = \frac{\partial E}{\partial a_2^{(3)}} \frac{\partial a_2^{(3)}}{\partial z_2^{(3)}} \frac{\partial z_2^{(3)}}{\partial z_2^{(2)}} \frac{z_2^{(2)}}{b_2^{(2)}} = \delta_2^{(2)}$$

$$\frac{\partial E}{\partial b_3^{(2)}} = \frac{\partial E}{\partial a_3^{(3)}} \frac{\partial a_3^{(3)}}{\partial z_3^{(3)}} \frac{\partial z_3^{(3)}}{\partial z_3^{(2)}} \frac{z_3^{(2)}}{b_3^{(2)}} = \delta_3^{(2)}$$

$$\frac{\partial E}{\partial b_4^{(2)}} = \frac{\partial E}{\partial a_4^{(3)}} \frac{\partial a_4^{(3)}}{\partial z_4^{(3)}} \frac{\partial z_4^{(3)}}{\partial z_4^{(2)}} \frac{z_4^{(2)}}{b_4^{(2)}} = \delta_4^{(2)}$$

$$\begin{aligned} E &= \frac{1}{2} \|y - a^{(3)}\|^2 = \frac{1}{2} \left((y_1 - a_1^{(3)})^2 + (y_2 - a_2^{(3)})^2 \right) \\ &= \frac{1}{2} \left[\left(y_1 - f \left(\Theta_{11}^{(3)} a_1^{(2)} + \Theta_{12}^{(3)} a_2^{(2)} + \Theta_{13}^{(3)} a_3^{(2)} + \Theta_{14}^{(3)} a_4^{(2)} + b_1^{(3)} \right) \right)^2 + \right. \\ &\quad \left. \left(y_2 - f \left(\Theta_{21}^{(3)} a_1^{(2)} + \Theta_{22}^{(3)} a_2^{(2)} + \Theta_{23}^{(3)} a_3^{(2)} + \Theta_{24}^{(3)} a_4^{(2)} + b_2^{(3)} \right) \right)^2 \right] \end{aligned}$$





4. 反向传播

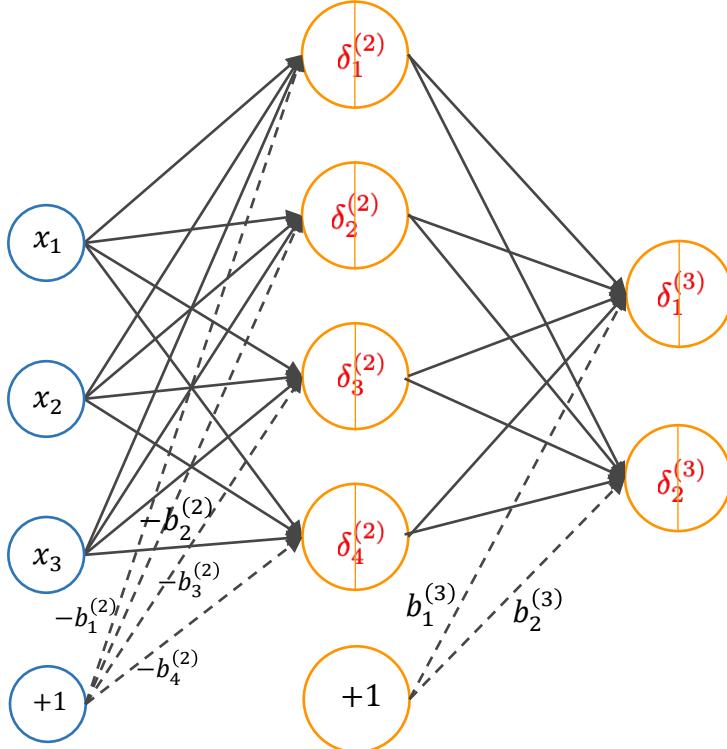
梯度下降-偏置更新

输出层和隐藏层的偏置为：

$$\begin{aligned}\frac{\partial E}{\partial b_i^{(l)}} &= \frac{\partial E}{\partial z_i^{(l)}} \frac{\partial z_i^{(l)}}{\partial b_i^{(l)}} \\ &= \delta_i^{(l)}\end{aligned}$$

向量形式为： $\nabla_{b^{(l)}} E = \boldsymbol{\delta}^l$

$$\begin{aligned}E &= \frac{1}{2} \|y - \mathbf{a}^{(3)}\|^2 = \frac{1}{2} \left((y_1 - a_1^{(3)})^2 + (y_2 - a_2^{(3)})^2 \right) \\ &= \frac{1}{2} \left[\left(y_1 - f \left(\Theta_{11}^{(3)} a_1^{(2)} + \Theta_{12}^{(3)} a_2^{(2)} + \Theta_{13}^{(3)} a_3^{(2)} + \Theta_{14}^{(3)} a_4^{(2)} + b_1^{(3)} \right) \right)^2 + \right. \\ &\quad \left. \left(y_2 - f \left(\Theta_{21}^{(3)} a_1^{(2)} + \Theta_{22}^{(3)} a_2^{(2)} + \Theta_{23}^{(3)} a_3^{(2)} + \Theta_{24}^{(3)} a_4^{(2)} + b_2^{(3)} \right) \right)^2 \right]\end{aligned}$$





4. 反向传播

BP算法的核心

反向传播算法的核心公式为：

$$\delta_i^{(L)} = - \left(y_i - a_i^{(L)} \right) f' \left(z_i^{(L)} \right) \quad \delta_i^{(l)} = \left(\sum_{j=1}^{n_{l+1}} \delta_j^{(l+1)} \Theta_{ji}^{(l+1)} \right) f' \left(z_i^{(l)} \right)$$

$$\frac{\partial E}{\partial \Theta_{ij}^{(l)}} = \delta_i^{(l)} a_j^{(l-1)}$$

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

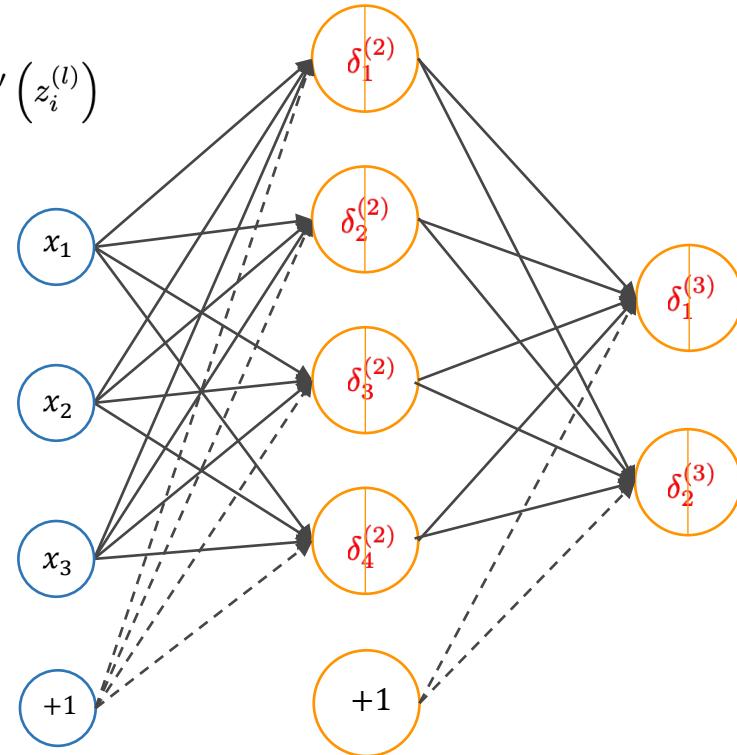
向量形式为：

$$\delta^{(L)} = - \left(y - a^{(L)} \right) \odot f' \left(z^{(L)} \right)$$

$$\delta^{(l)} = \left(\left(\Theta^{(l+1)} \right)^T \delta^{(l+1)} \right) \odot f' \left(z^{(l)} \right)$$

$$\nabla_{\Theta^{(l)}} E = \delta^{(l)} \left(a^{(l-1)} \right)^T$$

$$\nabla_{b^{(l)}} E = \delta^l$$





4. 反向传播

BP算法的核心

输入：训练数据集合 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$; 学习率 μ 。

输出：模型权重，偏置

1. 随机初始化输出层和隐藏层的权重、偏置；

2. 前向传播：

$$\mathbf{z}^{(l)} = \Theta^{(l)} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}$$

$$\mathbf{a}^{(l)} = f(\mathbf{z}^{(l)})$$

3. 计算每一层的误差：

$$\delta_i^{(L)} = - \left(y_i - a_i^{(L)} \right) f' \left(z_i^{(L)} \right), \quad (1 \leq i \leq n_L)$$

$$\delta_i^{(l)} = \left(\sum_{j=1}^{n_{l+1}} \delta_j^{(l+1)} \Theta_{ji}^{(l+1)} \right) f' \left(z_i^{(l)} \right), \quad (1 \leq i \leq n_l)$$

4. 求各层权重、偏置的偏导：

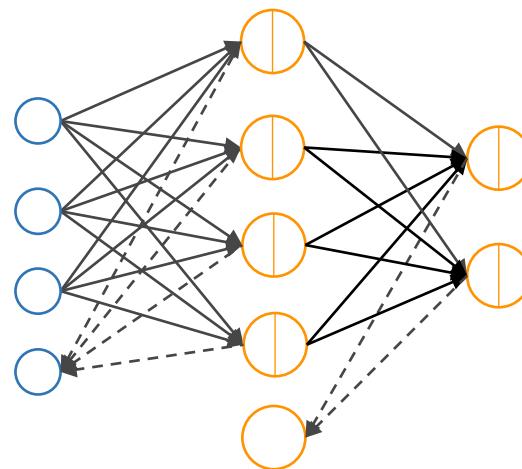
$$\frac{\partial E}{\partial \Theta_{ij}^{(l)}} = \delta_i^{(l)} a_j^{(l-1)}$$

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

5. 更新参数：

$$\Theta^{(l)} = \Theta^{(l)} - \frac{\mu}{N} \sum_{i=1}^N \frac{\partial E^{(i)}}{\partial \Theta^{(l)}}$$

$$\mathbf{b}^{(l)} = \mathbf{b}^{(l)} - \frac{\mu}{N} \sum_{i=1}^N \frac{\partial E^{(i)}}{\mathbf{b}^{(l)}}$$





知识引入



感知机



神经网络



反向传播



手写数字识别



课后作业

作业 V

我们的手写数字识别案例中，多层神经网络模型的参数总共有多少个？如果图片变成了 $256*256$ 大小，那么模型的参数将会发生如何变化？当模型参数过大的时候，有什么措施可以降低计算的复杂度或是提升计算的速度？