EDU
CSDN学院 IT实战派

机器学习工程师直通车

—— 深度学习部分

讲师：智亮

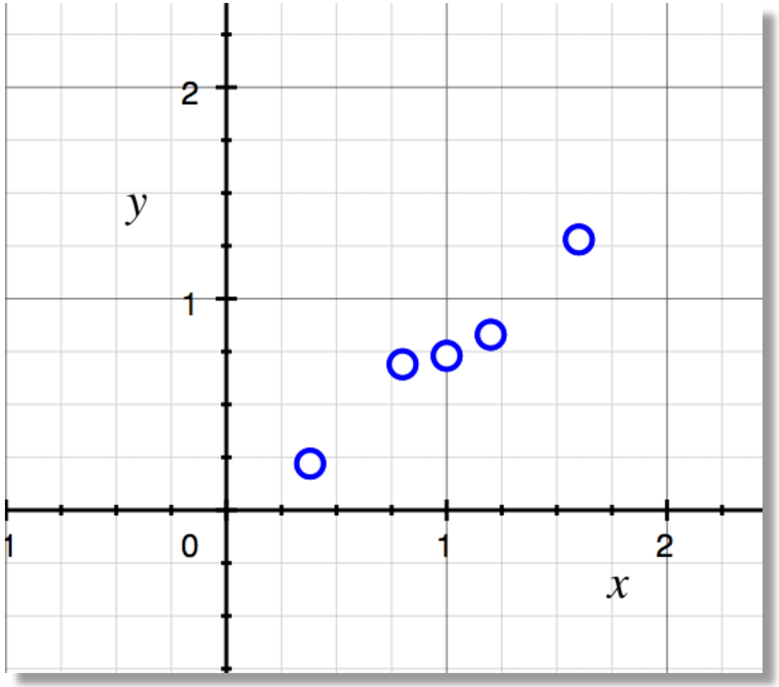
训练

- 回顾一下梯度下降算法

监督学习

$$y = \theta x$$

x(input)	y(ground truth)
1	0.73
0.38	0.22
1.2	0.83
1.6	1.28
0.8	0.69



名词解释：

- 损失（loss）：衡量神经网络的输出和Ground truth有多大差异的值。
- 损失函数（loss function）：计算损失的函数。又称为代价函数，误差函数。
- 梯度（gradient）：损失函数对于权重的导（函）数，用来衡量权重改变时，损失会如何变化。
- 梯度下降（gradient descent）：根据梯度来更新权重，使损失变小的方法。
- 随机梯度下降（stochastic gradient descent）：每次使用一条（或几条）数据（而不是全部数据），进行梯度下降的方法。
- 训练（training）：使用梯度下降更新一点点θ。不断重复这个动作，使loss下降，直到获得最小值的过程。
- 训练step：一次“计算梯度，更新权重”的动作。
- 数据集（dataset）：包含输入数据和ground truth，用来训练神经网络的数据。
- epoch：在随机梯度下降中，当数据集里面每条数据都用来训练了一次，叫一个epoch。

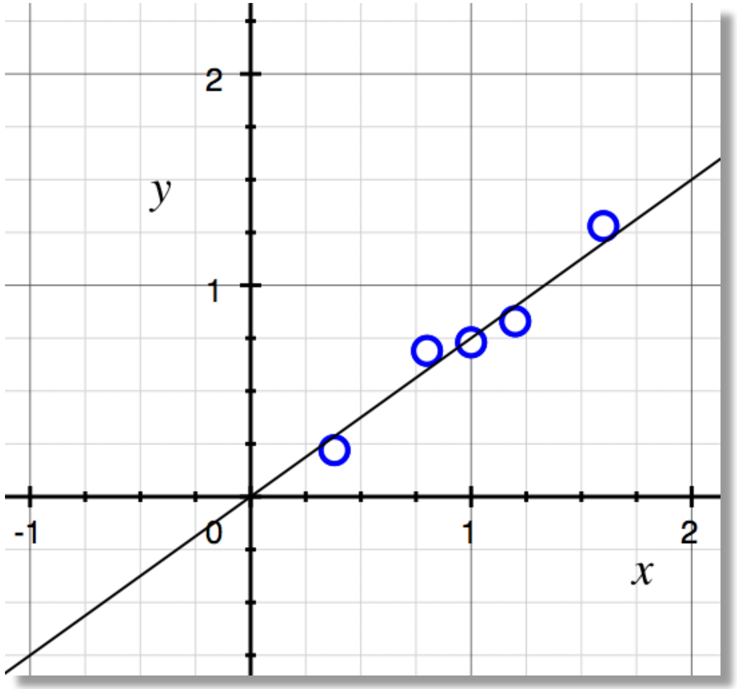
感知器的学习规则

- 回顾一下梯度下降算法

监督学习

$$y = \theta x$$

x(input)	y(ground truth)
1	0.73
0.38	0.22
1.2	0.83
1.6	1.28
0.8	0.69



误差和损失函数（loss/cost function）

$$loss = \frac{1}{2}(output - y)^2$$
$$= \frac{1}{2}(\theta x - y)^2$$

- 损失（loss）：衡量神经网络的输出和Ground truth有多大差异的值。
- 损失函数（loss function）：计算损失的函数。又称为代价函数，误差函数。

感知器的学习规则

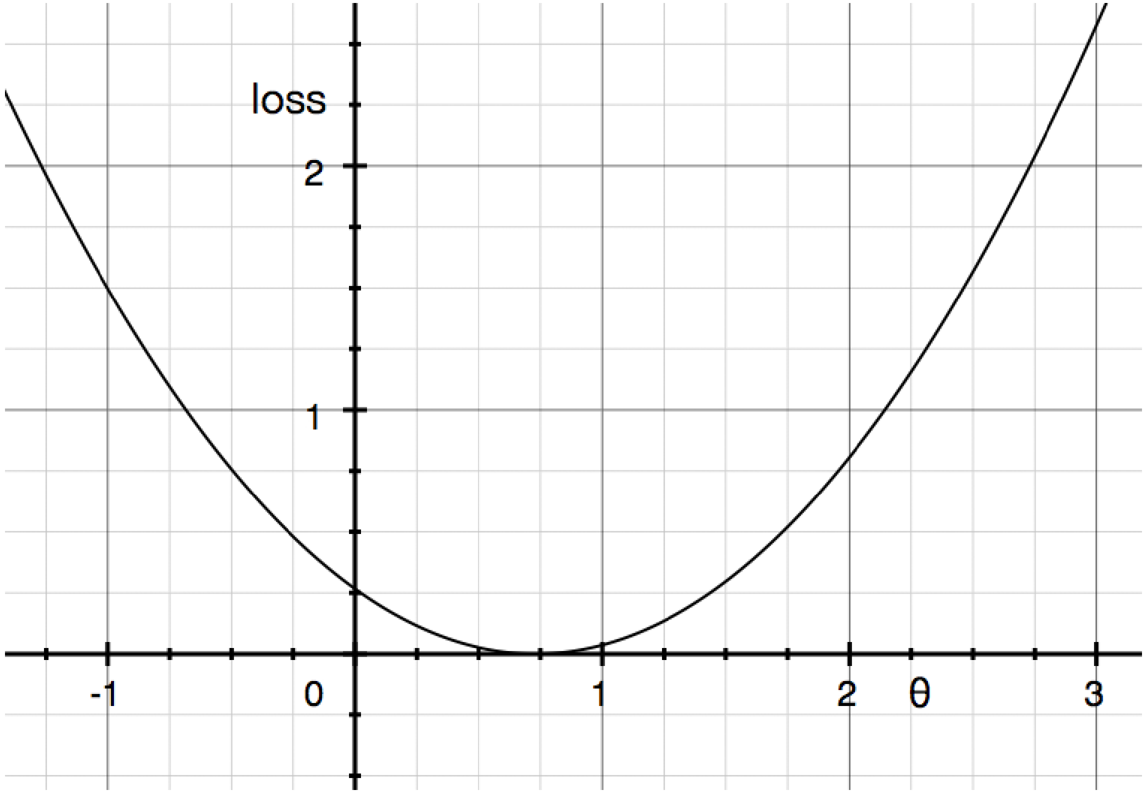
- 回顾一下梯度下降算法
  - 监督学习
  - 误差和损失函数（loss/cost function）

x(input)	y(ground truth)
1	0.73

$$y = \theta x \quad t = (ground \quad truth)$$

$$\begin{aligned} loss &= \frac{1}{2}(t-y)^2 \\ &= \frac{1}{2}t^2 - ty + \frac{1}{2}y^2 \\ &= \frac{1}{2}t^2 - t\theta x + \frac{1}{2}\theta^2 x^2 \end{aligned}$$

$$L(\theta) = \frac{1}{2}x^2 \cdot \theta^2 - tx \cdot \theta + \frac{1}{2}t^2$$



- 损失（loss）：衡量神经网络的输出和Ground truth有多大差异的值。
- 损失函数（loss function）：计算损失的函数。又称为代价函数，误差函数。

感知器的学习规则

EDU
CSDN学院 IT实战派

- 回顾一下梯度下降算法
  - 监督学习
  - 误差和损失函数（loss/cost function）

$$y = \theta x \quad t = (ground \quad truth)$$

$$\begin{aligned} loss &= \frac{1}{2}(t-y)^2 \\ &= \frac{1}{2}t^2 - ty + \frac{1}{2}y^2 \\ &= \frac{1}{2}t^2 - t\theta x + \frac{1}{2}\theta^2 x^2 \end{aligned}$$

$$L(\theta) = \frac{1}{2}x^2 \cdot \theta^2 - tx \cdot \theta + \frac{1}{2}t^2$$

损失函数的导函数称为梯度函数，代表着损失函数在θ轴各点的斜率

$$\begin{aligned} grad_\theta &= \frac{dL}{d\theta} \\ &= \frac{dL}{d(t-y)} \frac{d(t-y)}{d\theta} \\ &= (t-y) \cdot (-x) \\ &= (y-t)x \end{aligned}$$

$$\begin{aligned} grad_\theta &= \frac{dL}{d\theta} \\ &= x^2\theta - tx \\ &= x(\theta x - t) \\ &= (y-t)x \end{aligned}$$

$$\theta_{new} = \theta_{old} - \eta \cdot grad_\theta$$

- 梯度（gradient）：损失函数对于权重的导（函）数，用来衡量权重改变时，损失会如何变化。
- 梯度下降（gradient descent）：根据梯度来更新权重，使损失变小的方法。

感知器的学习规则

- 回顾一下梯度下降算法
  - 监督学习
  - 误差和损失函数（loss/cost function）

$$y = \theta x \quad t = (ground \quad truth)$$

$$loss = \frac{1}{2}(t-y)^2$$
$$= \frac{1}{2}t^2 - ty + \frac{1}{2}y^2$$
$$= \frac{1}{2}t^2 - t\theta x + \frac{1}{2}\theta^2 x^2$$

$$L(\theta) = \frac{1}{2}x^2 \cdot \theta^2 - tx \cdot \theta + \frac{1}{2}t^2$$

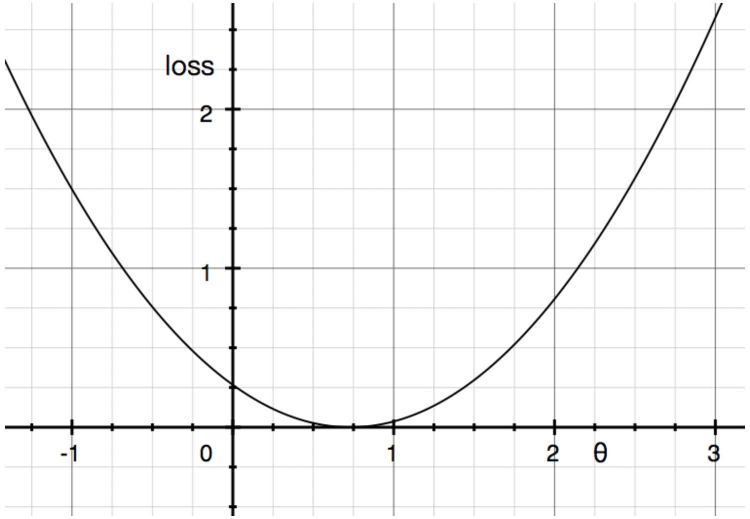
损失函数的导函数称为梯度函数，代表着损失函数在θ轴各点的斜率

$$grad_\theta = \frac{dL}{d\theta}$$
$$= \frac{dL}{d(t-y)} \frac{d(t-y)}{d\theta}$$
$$= (t-y) \cdot (-x)$$
$$= (y-t)x$$

$$grad_\theta = \frac{dL}{d\theta}$$
$$= x^2\theta - tx$$
$$= x(\theta x - t)$$
$$= (y-t)x$$

$$\theta_{new} = \theta_{old} - \eta \cdot grad_\theta$$

x(input)	y(ground truth)
1	0.73



- 梯度（gradient）：损失函数对于权重的导（函）数，用来衡量权重改变时，损失会如何变化。
- 梯度下降（gradient descent）：根据梯度来更新权重，使损失变小的方法。

学习率η

$$y = \theta x$$

$$loss = \frac{1}{2}(t-y)^2$$

$$grad_\theta = (y-t)x$$

$$\theta_{new} = \theta_{old} - \eta \cdot grad_\theta$$

若η=1

1)

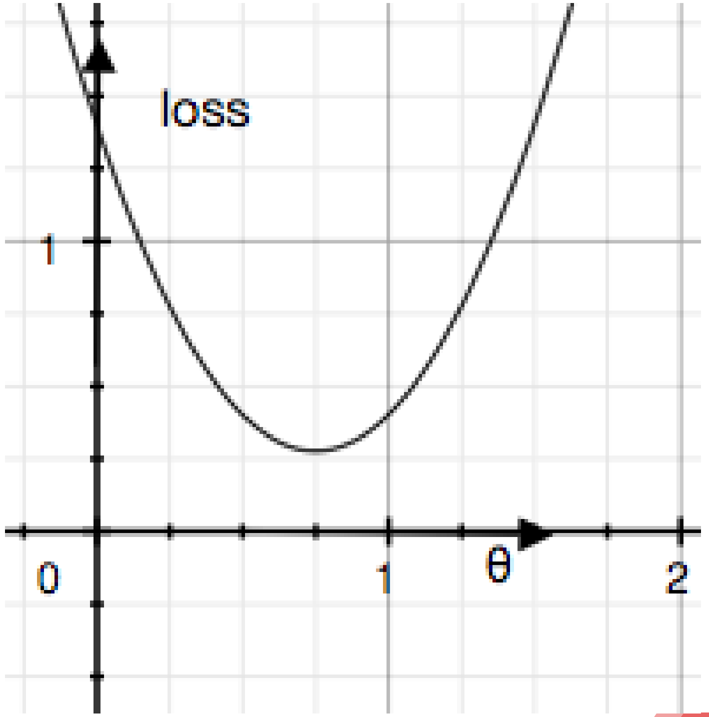
$$\theta = 3, x = 2, t = 8$$
$$y = 3 \times 2 = 6$$
$$loss = \frac{1}{2}(8-6)^2 = 2$$
$$grad_\theta = (6-8) \times 2 = -4$$
$$\theta_{new} = 3 - (-4) = 7$$

2)

$$\theta = 7, x = 2, t = 8$$
$$y = 7 \times 2 = 14$$
$$loss = \frac{1}{2}(8-14)^2 = 18$$
$$grad_\theta = (14-8) \times 2 = 12$$
$$\theta_{new} = 7 - 12 = -5$$

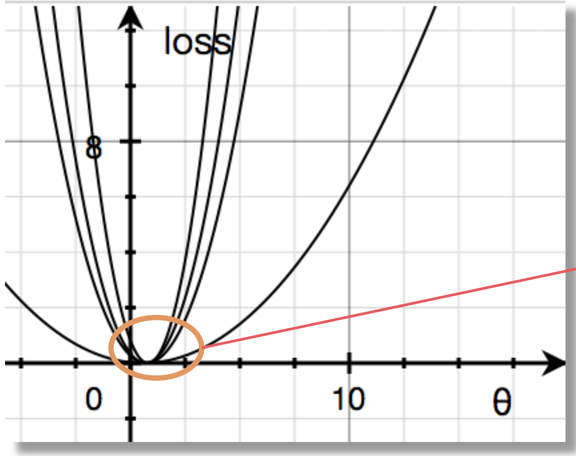
3)

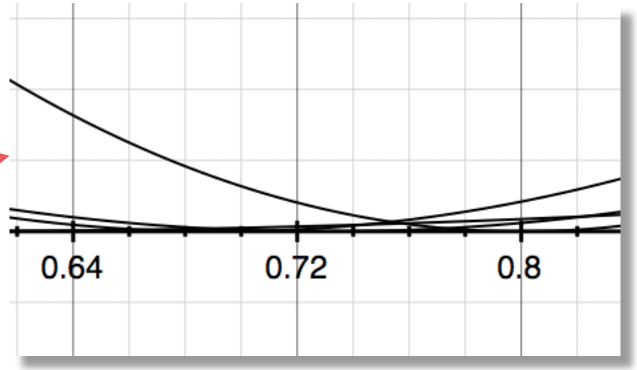
$$\theta = -5, x = 2, t = 8$$
$$y = -5 \times 2 = -10$$
$$loss = \frac{1}{2}(8-(-10))^2 = 162$$
$$grad_\theta = (-10-8) \times 2 = -36$$
$$\theta_{new} = -5 - (-36) = 31$$



梯度下降与训练

x(input)	y(ground truth)
1	0.73
0.38	0.22
1.2	0.83
1.6	1.28
0.8	0.69





- 梯度下降（gradient descent）：根据梯度来更新权重，使损失变小的方法。
- 随机梯度下降（stochastic gradient descent）：每次使用一条（或几条）数据（而不是全部数据），进行梯度下降的方法。
- 训练（training）：使用梯度下降更新一点点θ。不断重复这个动作，使loss下降，直到获得最小值的过程。
- 训练step：一次“计算梯度，更新权重”的动作。又称为iteration。
- 数据集（dataset）：包含输入数据和ground truth，用来训练神经网络的数据。
- epoch：在随机梯度下降中，当数据集里面每条数据都用来训练了一次，称为一个epoch。

感知器的学习规则

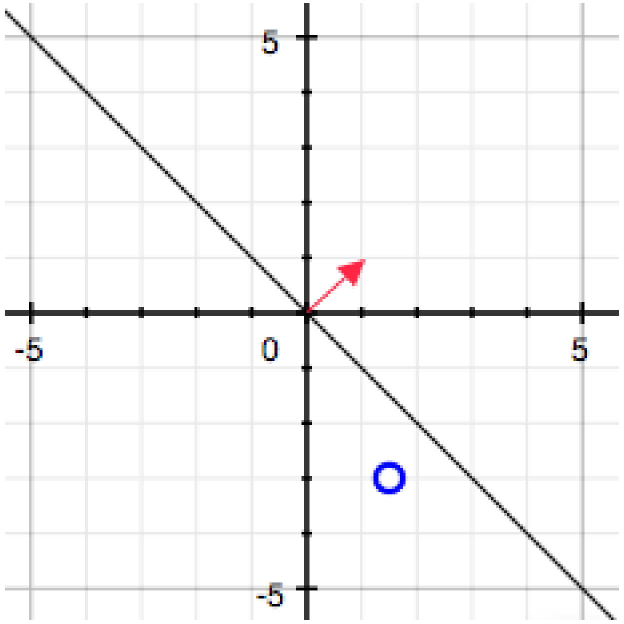
设 $x_1 = 1.5, \quad x_2 = -3$

$w_1 = 1, \quad w_2 = 1$

$b = 0$

此时
$$\begin{aligned} output &= f(w_1 x_1 + w_2 x_2 + b) \\ &= f(1 \times 1.5 + 1 \times -3 + 0) \\ &= f(-1.5) \\ &= 0 \end{aligned}$$

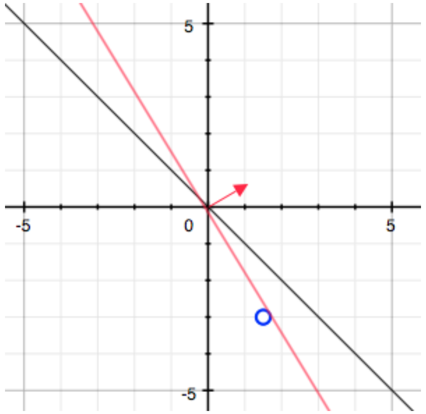
约定 $y = output$，若t=1， 那么
$$\begin{aligned} \Delta &= t - y \\ &= 1 - 0 \\ &= 1 \end{aligned}$$



感知器的学习规则

- 定义损失函数如下:

$$L(w,b) = -(t-y)(x \cdot w + b)$$



- 其对w和b的梯度为:

$$grad_w = \frac{dL}{dw} = -(t-y)x, \quad grad_b = \frac{dL}{db} = -(t-y)$$

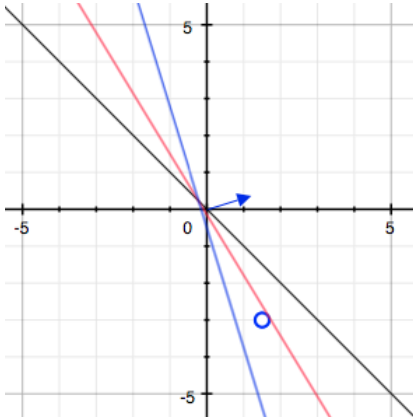
- 对权重更新时使用:

$$w_{new} = w_{old} + \eta(t-y)x, \quad b_{new} = b_{old} + \eta(t-y)$$

- 对之前的例子应用权重更新:

$$\begin{aligned} W &= W + \eta(t-y)x \\ &= \begin{bmatrix}1 & 1\end{bmatrix} + 0.1 \times (1-0) \times \begin{bmatrix}1.5 & -3\end{bmatrix} \\ &= \begin{bmatrix}1.15 & 0.7\end{bmatrix} \end{aligned}$$

$$\begin{aligned} b &= b + \eta(t-y) \\ &= 0 + 0.1 \times (1-0) \\ &= 0.1 \end{aligned}$$



- 更新后的

$$\begin{aligned} output &= f(Wx+b) \\ &= f(1.15 \times 1.5 + 0.7 \times (-3) + 0.1) \\ &= f(-0.274) \\ &= 0 \end{aligned}$$

- 再次更新后

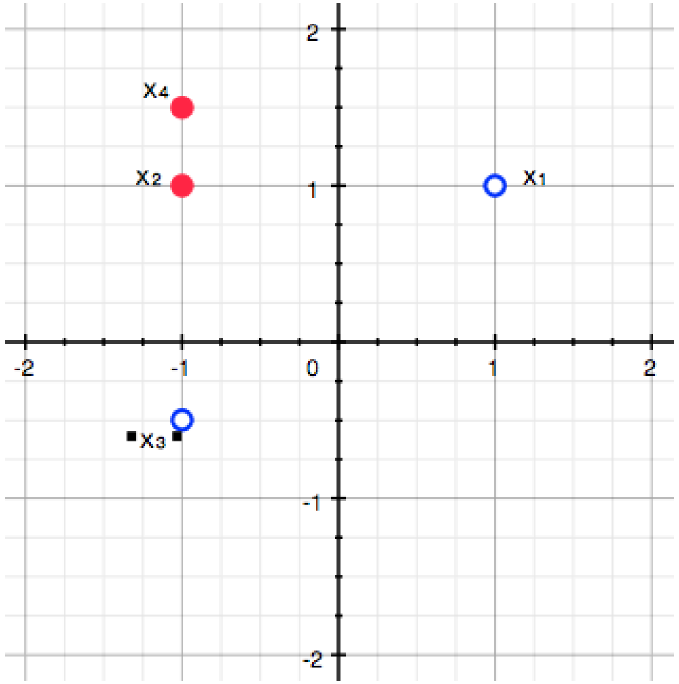
$$W = \begin{bmatrix}1.3, 0.4\end{bmatrix}, b = 0.2$$

$$\begin{aligned} output &= f(1.3.5 + 0.4 + 0.2) \\ &= f(0.95) \\ &= 1 \end{aligned}$$

感知器的学习规则

- 多条数据情况
- 假定我们有这样几条输入：

$$X = \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ -1 & -0.5 \\ -1 & 1.5 \end{bmatrix} \qquad t = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$



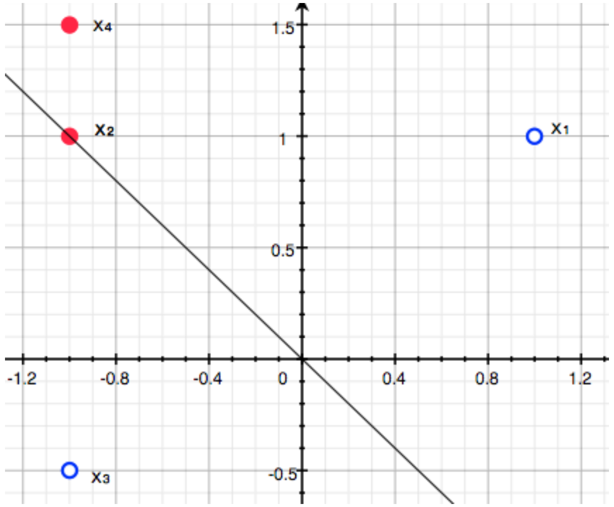
感知器的学习规则

初始化设置w=[1, 1]， b=0，则

$$logits = X \cdot w + b = \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ -1 & -0.5 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 0 = \begin{bmatrix} 1 \\ 0 \\ -1.5 \\ 0.5 \end{bmatrix}$$

$$output = f(logits) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \qquad f(x) = \begin{cases} 1, x > 0 \\ 0, x \leq 0 \end{cases}$$

现在， $t = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, y = output = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$



感知器的学习规则

$$grad_w = -\sum_i (t-y)x_i \qquad grad_b = -\sum (t-y)$$

$$= -\left(\begin{bmatrix}0\\1\\0\\1\end{bmatrix} - \begin{bmatrix}1\\0\\0\\1\end{bmatrix}\right)\cdot\begin{bmatrix}1 & 1\\-1 & 1\\-1 & -0.5\\-1 & 1\end{bmatrix} \qquad = -(-1+1+0+0)$$

$$= -\begin{bmatrix}-1\\1\\0\\0\end{bmatrix}\cdot\begin{bmatrix}1 & 1\\-1 & 1\\-1 & -0.5\\-1 & 1\end{bmatrix} \qquad = 0$$

$$= -\begin{bmatrix}-2 & 0\end{bmatrix}$$

设η=0.25，则 $W_{new} = W_{old} - \eta grad_w = \begin{bmatrix}0.5 & 1\end{bmatrix}$

$b = 0$

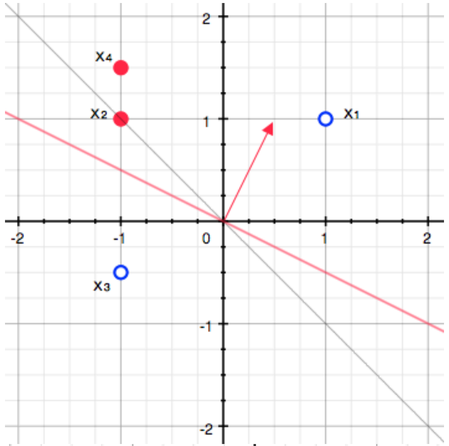
重复这个过程，得到新的

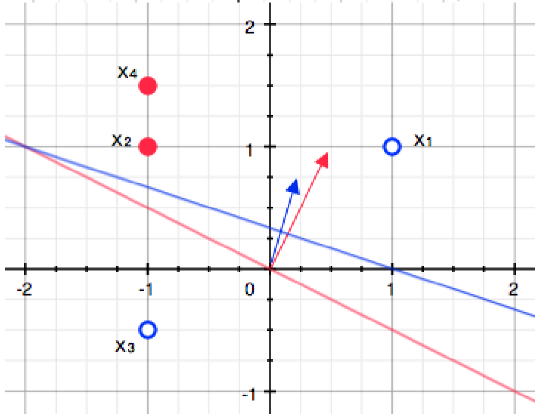
$$grad_w = \sum_i -(t-y)x_i \qquad grad_b = \sum -(t-y)$$

$$= -\begin{bmatrix}-1 & -1\end{bmatrix} \qquad = 1$$

$$W_{new} = W_{old} - \eta grad_w = \begin{bmatrix}0.25 & 0.75\end{bmatrix}$$

$$b = -0.25$$

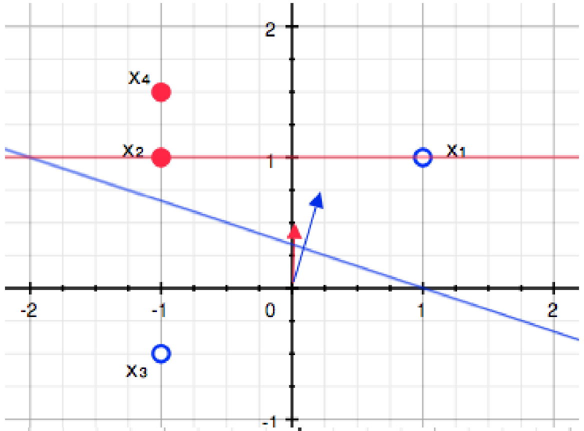




感知器的学习规则

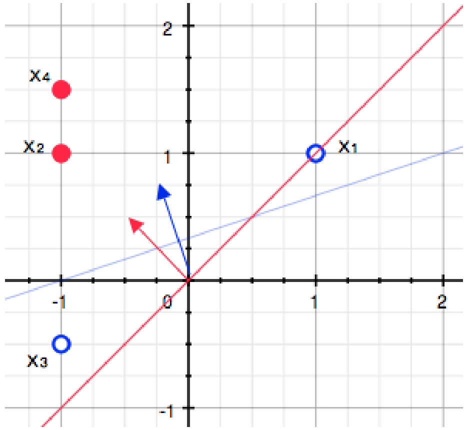
3） $W_{new} = W_{old} - \eta grad_w = \begin{bmatrix} 0 & 0.5 \end{bmatrix}$

$b = -0.5$

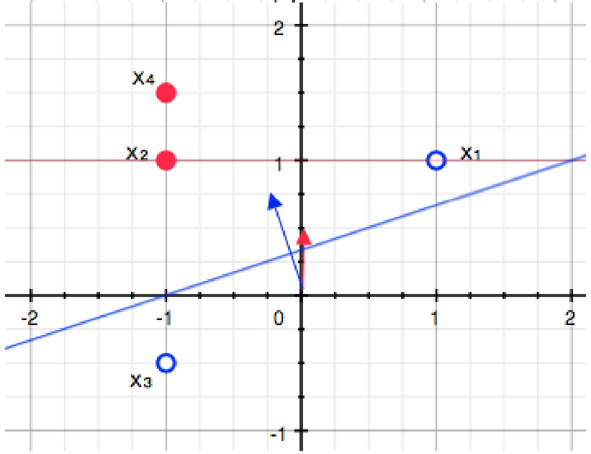


4） $W_{new} = W_{old} - \eta grad_w = \begin{bmatrix} 0.25 & 0.75 \end{bmatrix}$

$b = -0.25$



5）



6）