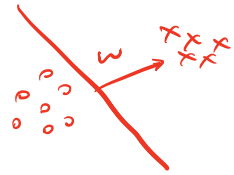
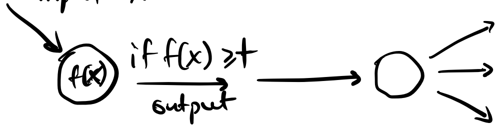
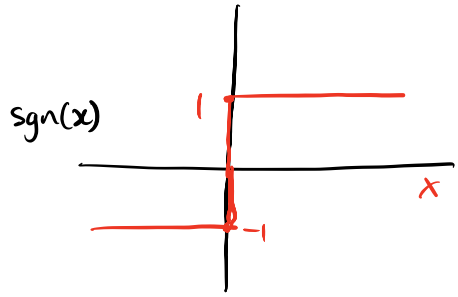
Neural Learning

- Neural Networks → Deep Learning (Multi-Layer Perceptron)
- Perceptron (Building block of NNs) (linear classifier)

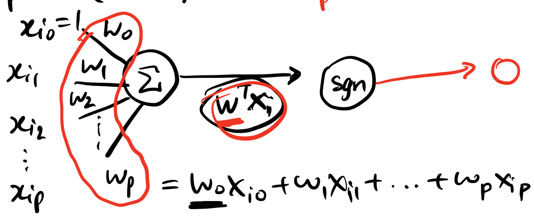






$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \end{cases}$$

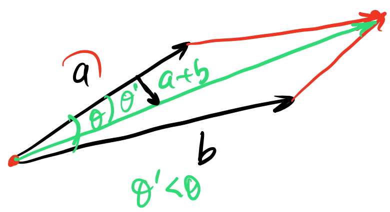
perceptron (classification) : parameter $w = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{pmatrix}$



$= w_0 x_{i0} + w_1 x_{i1} + \dots + w_p x_{ip}$

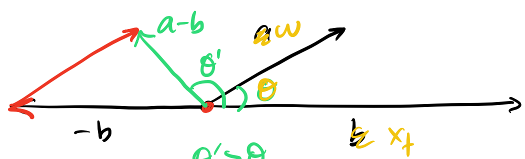
$D = \{(x_i, y_i)\} \quad y_i \in \{-1, 1\}$

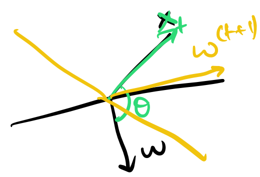
Vector addition



- $a + b$

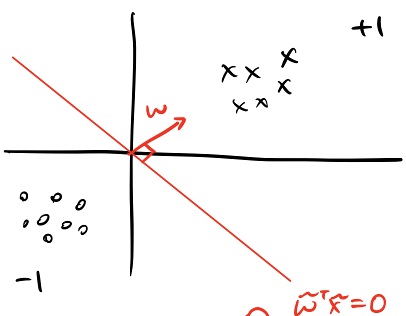
Vector subtraction

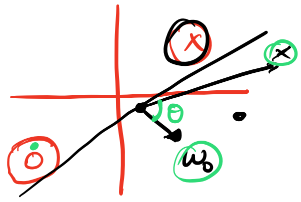




$a - b = a + (-b)$

$w^T x = w \cdot x = \langle w, x \rangle$





Perceptron:

- Init w randomly $(w^{(0)} = (0, 0, \dots, 0))$
- For $i = 1, \dots, T$:

$w^{(t)} x_t y_t \geq 0$ pick data point (x_t, y_t)

- if $w^{(t)} \cdot x_t \geq 0$ and $y_t = 1$: do nothing $w^{(t+1)} = w^{(t)}$
- if $w^{(t)} \cdot x_t < 0$ and $y_t = -1$: do nothing $w^{(t+1)} = w^{(t)}$
- if $w^{(t)} x_t \geq 0$ and $y_t = -1$: $w^{(t+1)} = w_t^{(t)} - x^t y_t$

$w^{(t)} x_t y_t < 0$

- if $w^{(t)} x_t < 0$ and $y_t = +1$: $w^{(t+1)} = w^{(t)} + x^t y_t$

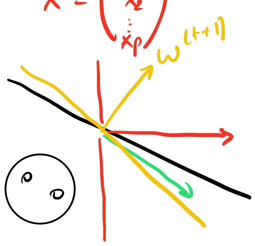
if $w^T x > 0$, θ is acute,

so x belongs to class +1

if $w^T x < 0$, θ obtuse,

so x belongs to -1 class

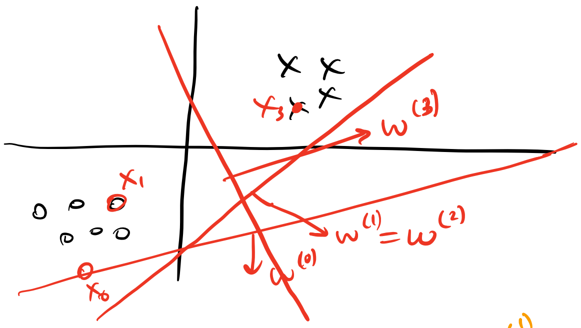
$\tilde{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{pmatrix} \quad \tilde{x} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix}$



Mistakes:

- if $w^T x_i > 0$ and $y_i = -1$

For $t = 1, \dots, T$:
receive (x_t, y_t)



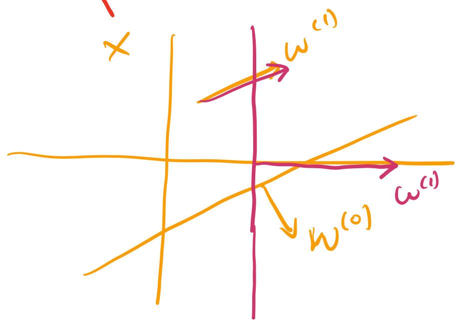
• if $w^T x_i < 0$ and $y_i = +1$

$\boxed{\text{if } y_t w^{(t)} \cdot x_t < 0: \; w^{(t+1)} = w^{(t)} + y_t x_t}$

$y_t \in \{-1, 1\}$

if $y_t w^{(t)} \cdot x_t < 0$: update "eta"

$w^{(t+1)} = w^{(t)} + \underbrace{y_t x_t}\eta$ ← learning rate

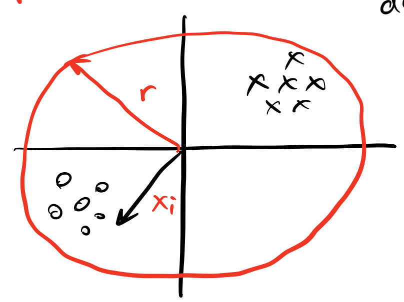


Perceptron Mistake Bound

Theorem: Assume: $\|x_i\|_2 < R$ (inputs are finite)

Assume: there exists w^* with $\underbrace{\|w^*\|_2 = 1}$ and $y_i w^* \cdot x_i \geq \gamma$ for all i data points.

margin



Gradient Descent (numerical technique)

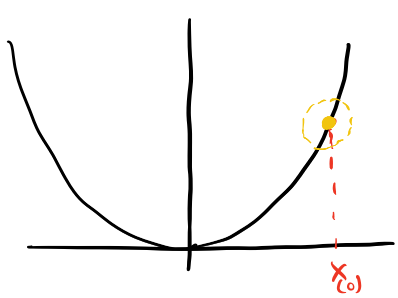
• $\min_x f(x)$

Least squares Reg: $\min_w L(w)$

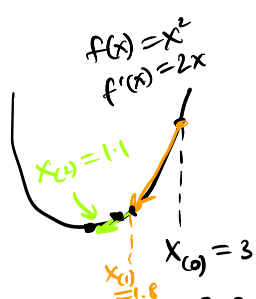
Ex: $f(x) = x^2 \longrightarrow \boxed{f'(x) = 0}$

$2x = 0$

$\hat{x} = 0$



minimiser has a closed form solution



$\alpha = 0.2$

$x_{(1)} = x_{(0)} - 0.2 \cdot 2 \cdot x_{(0)}$

$= 3 - 0.4 \cdot 3 = 1.8$

$x_{(2)} = 1.8 - 0.2 \cdot 2 \cdot 1.8$

$\approx 1.8 - 0.7$

$= 1.1$

$x_{(3)} = \text{----}$

$f(x)$
$f(1.8)$
$= (1.8)^2$
$f(1.1)$
$= (1.1)^2$

• Start with init value $x_{(0)}$

• direction of steepest descent $- f'(x_{(0)})$

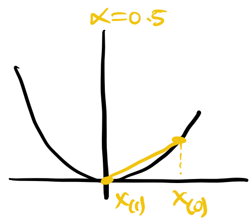
• $x_{(t)} = \underbrace{x_{(t-1)}}_{\text{current pos}} \underbrace{- \alpha f'(x_{(t-1)})}_{\text{step}}$

step-size

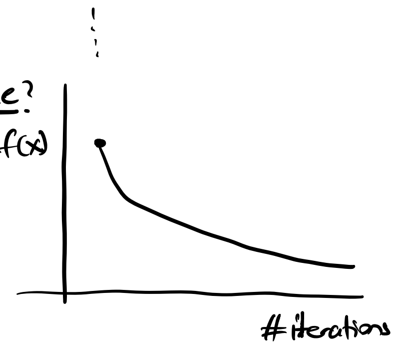
$x_{(0)} = 3 \qquad \alpha = 0.5$

$x_{(1)} = 3 - 0.5 \times 2 \times 3$

$= 0$



Convergence?



$\alpha = 1$

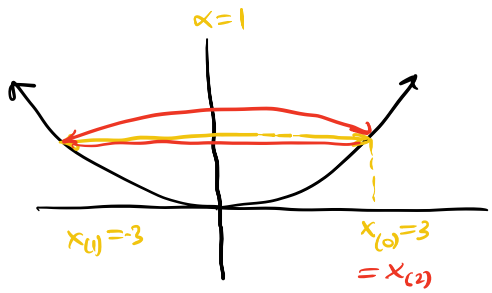
$x_{(1)} = 3 - 1 \times 2 \times 3$

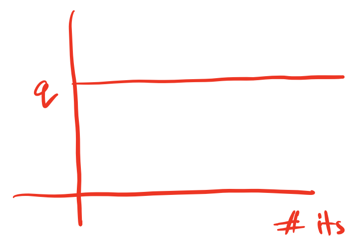
$= -3$

$x_{(2)} = -3 - \underbrace{1 \times 2 \times (-3)}$

$= -3 + 6$

$= 3$





Least Squares (simple):

$$\mathcal{L}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - (w_0 + w_1 x_i))^2$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = -\frac{2}{n} \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = -\frac{2}{n} \sum_{i=1}^{n} x_i (y_i - w_0 - w_1 x_i)$$

closed form sols:

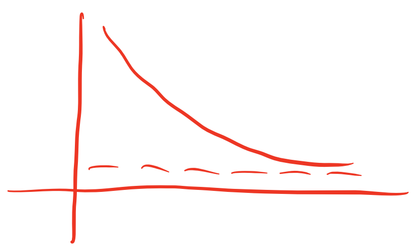
$$\boxed{\begin{aligned} w_0 &= \bar{y} - w_1 \bar{x} \\ w_1 &= \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2} \end{aligned}}$$

Gradient Descent

init $w_0^{(0)}, w_1^{(0)}$

$$w_0^{(t)} = w_0^{(t-1)} + \frac{2\alpha}{n} \sum_{i=1}^{n} (y_i - w_0^{(t-1)} - w_1^{(t-1)} x_i)$$

$$w_1^{(t)} = w_1^{(t-1)} + \frac{2\alpha}{n} \sum_{i=1}^{n} x_i (y_i - w_0^{(t)} - w_1^{(t-1)} x_i)$$



print $(\mathcal{L}(w_0^{(t)}, w_1^{(t)}))$

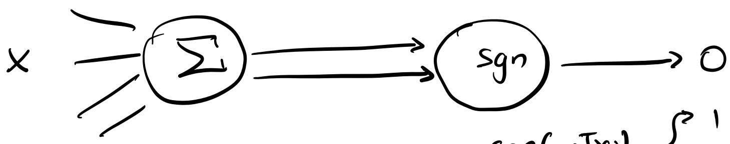
Least Squares

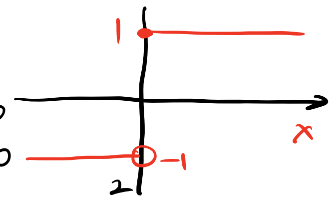
$$\mathcal{L}(\beta) = \|y - X\beta\|_2^2 \qquad \beta \in \mathbb{R}^{p+1}$$

$$\nabla_\beta \mathcal{L}(\beta) = -2X^T(y - X\beta) \longrightarrow \beta_{LS}^T = (X^TX)^{-1}X^Ty$$

$\beta^{(0)} = \underline{\text{init}}$

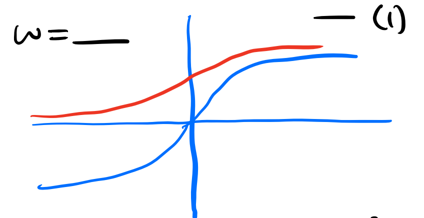
$$\beta^{(t)} = \beta^{(t-1)} + \alpha 2X^T(y - X\beta^{(t-1)})$$





$$sgn(w^T x_i) \begin{cases} 1 & \text{if } w^T x_i \geq 0 \\ -1 & \text{if } w^T x_i < 0 \end{cases}$$

$$\mathcal{L}(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(\overset{1}{y_i} - \overset{0.8}{sgn}(w^T x_i))^2$$



use different "sigmoidal"-function

logistic sigmoid: $\sigma(x) = \frac{1}{1+e^{-ax}}$

$$\frac{d\sigma(x)}{dx} = \sigma(x)(1-\sigma(x))$$

$y_i \in \{0,1\}$

$$\mathcal{L}(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \sigma(w^T x_i))^2$$

Grad-Desc

- init $w^{(0)}$
- repeat : $w^{(t)} = w^{(t-1)} - \alpha \nabla \mathcal{L}(w^{(t-1)})$
- stop when change in $\nabla(w^{(t)})$ is negligible

$$\mathcal{L}(w) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \sigma(w^T x_i))^2$$

$$\nabla_w \mathcal{L}(w) = \frac{1}{2}\sum_{i=1}^{n}\nabla_w (y_i - \sigma(w^T x_i))^2$$

$$= \sum_{i=1}^{n}\nabla_w (y_i - \sigma(w^T x_i))\left[y_i - \sigma(w^T x_i)\right)^{2-1}]$$

$$= -\sum_{i=1}^{n} x_i\, \sigma(w^T x_i)(1-\sigma(w^T x_i))(y_i - \sigma(w^T x_i))$$

$\nabla_w (y_i - \sigma(w^T x_i))$

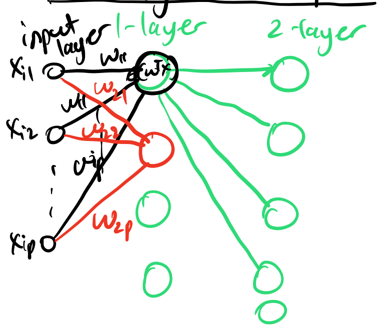
$- \nabla_w \sigma(w^T x_i)$

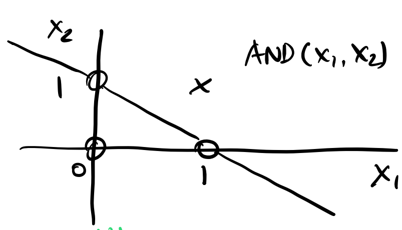
$- \nabla_w (w^T x_i) \cdot \sigma(w^T x_i)$

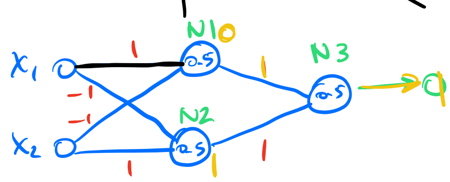
$- x_i \sigma(w^T x_i)(1-\sigma(w^T x_i))$

update: $w^{(t)} = w^{(t-1)} + \alpha \sum_{i=1}^{n} x_i\, \sigma(w^{(t-1)} \cdot x_i)(1-\sigma(w^{(t-1)} \cdot x_i))(y_i - \sigma(w^{(t-1)} \cdot x_i))$

Multilayer Perceptron







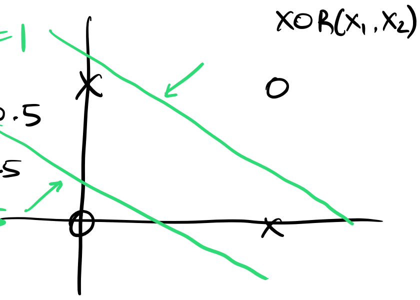
x_1	x_2	AND(x_1,x_2)
0	0	0
0	1	0
1	0	0
1	1	1

$x_1=0, x_2=0$ | $x_1=1, x_2=1$

N1: $0\times 1+0\times(-1)=0<0.5$

N2: $0\times(-1)+0\times 1=0<0.5$

N3: $0\times 1+0\times 1=0<0.5$



x_1	x_2	XOR(x_1, x_2)
0	0	0
0	1	1
1	0	1
1	1	0

$x_1=0, x_2=1$

N1: $0\times 1+1\times(-1)=-1<0.5$

N2: $0\times -1+1\times 1=1>0.5$

N3: $0\times 1+1\times 1=1>0.5$