ANDREW NG - MACHINE LEARNING

WEEK 1

→ ML :
- learn from experience E
- task T
- performance measure P

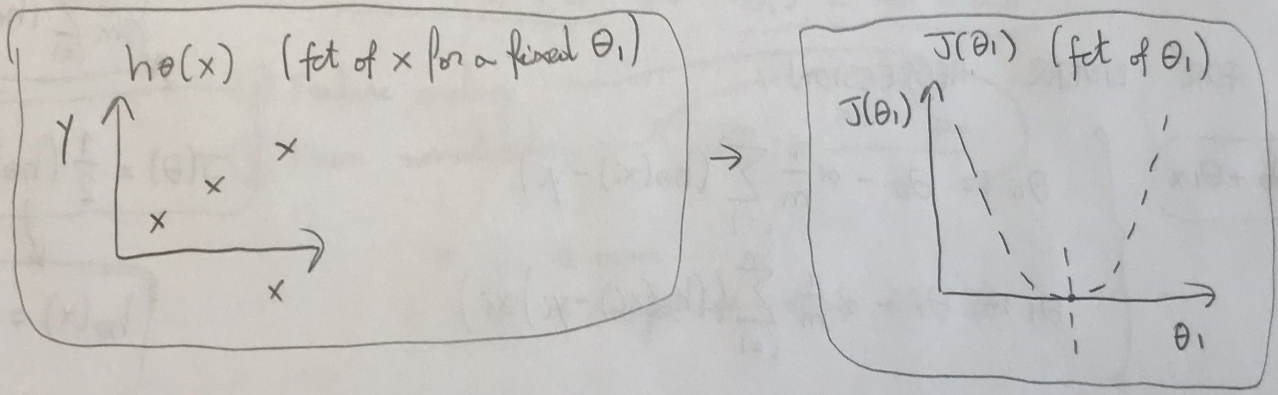
SUPERVISED
↳ regression
↳ classification

UNSUPERVISED

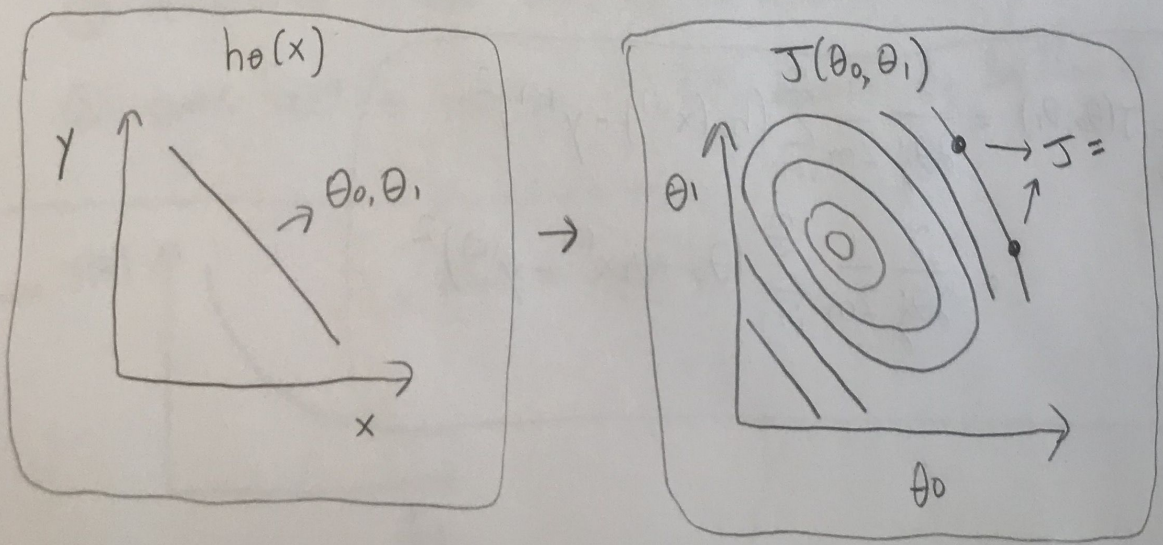
COST FUNCTION :

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

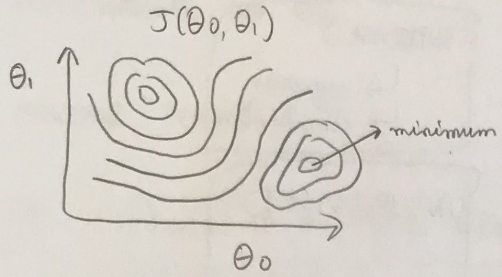
to minimize



CONTOUR PLOTS



GRADIENT DESCENT



$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

* simultaneously

α
- α too small → GD can be slow
- α too large → GD may miss the min, fail to converge or even diverge

* No need to decrease α over time because GD will automatically take smaller steps as we approach a local min.

FOR LINEAR REGRESSION :

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$\begin{cases} \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i) \\ \theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} ((h_\theta(x_i) - y_i) x_i) \end{cases}$$

$$\frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$J(\theta) = \frac{1}{2} (h_\theta(x) - y)^2$$

$$h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i$$

* BATCH GD → looks through all training set

NOTE :

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$= \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2$$

WEEK (2)

MULTIVARIATE LIN. REG.

MULTIPLE FEATURES :

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

$$h_\theta(x) = \begin{bmatrix} \theta_0 & \theta_1 & \dots & \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \theta^T x$$

GD :

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x_j^{(i)}$$

$$j := 0 \dots n$$

FEATURE SCALING :

→ Aim to have $-1 \leq x_{(i)} \leq 1$ OR $-0,5 \leq x_{(i)} \leq 0,5$

① Feature scaling
② Mean normalization

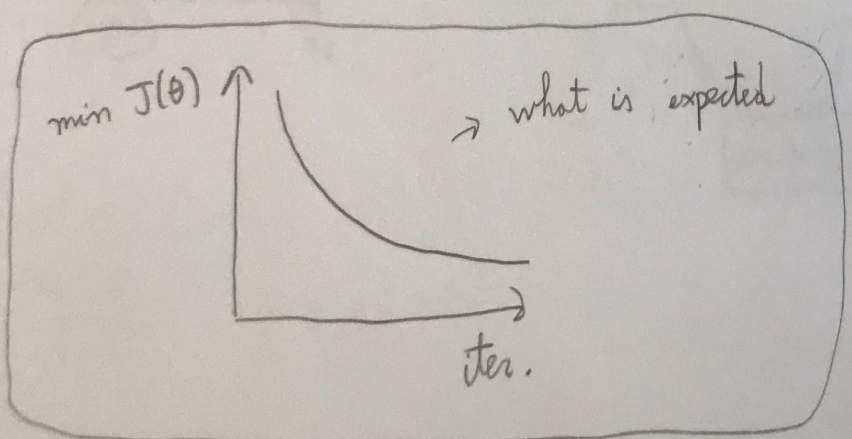
$$x_i := \frac{x_i - \mu_i}{s_i}$$

* μ_i → mean
s_i → range (max − min) OR std

GD TIPS :

① DEBUGGING → Plot $J(\theta)$ vs nbr of iteration

② CONV. TEST → Converges if $J(\theta)$ decreases by less than E (e.g. 10^{-3})



POLYNOMIAL REGRESSION :

→ CHANGE features depending on problem

$h_\theta(x) = \theta_0 + \theta_1 x_1 \longrightarrow h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

↳ $x_2 = x_1^2$

↳ $x_3 = x_1^3$

NORMAL EQUATION :

→ Find directly optimal θ without iterations

$$\theta = (X^T X)^{-1} X^T y$$

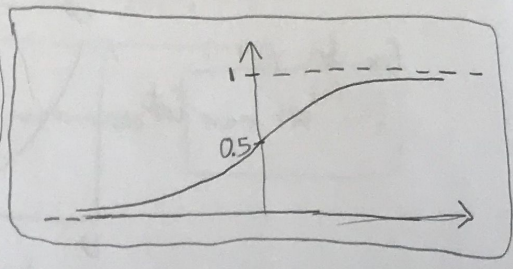
* No need to do feature scaling
If n too large (many features) → better with GD (if $n \geq 10000$)

(5) WEEK (3) - LOGISTIC REGRESSION

CLASSIFICATION

→ We want $0 \leq h_\theta(x) \leq 1$ → $h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$

↳ logistic fct
↳ sigmoid fct



→ $h_\theta(x)$ → Estimated probability that $y=1$ on input x

→ $h_\theta(x) = P(y=1 \mid x;\theta)$

→ $P(y=0 \mid x;\theta) + P(y=1 \mid x;\theta) = 1$

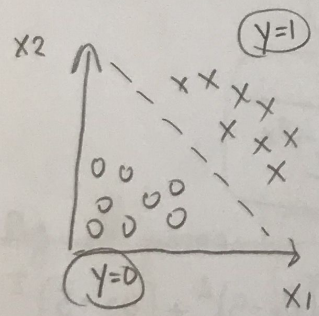
→ $h_\theta(x) = g(\theta^T x)$ & $g(z) = \frac{1}{1+e^{-z}}$

DECISION BOUNDARY

→ $y=1$ if $h_\theta(x) \geq 0.5$ → $g(z) \geq 0.5$ → $z \geq 0$ → $\theta^T x \geq 0$

→ $y=0$ if $h_\theta(x) < 0.5$ → $g(z) < 0.5$ → $z < 0$ → $\theta^T x < 0$

Ex : LINEAR DECISION BOUNDARY



$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

$\theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$

Predict $y=1$ if $-3 + x_1 + x_2 \geq 0$

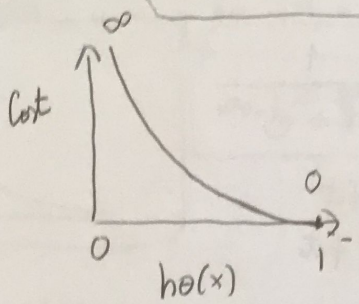
$\theta^T x$

$x_1 + x_2 \geq 3$

⑥

COST FUNCTION: $\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y=1 \\ -\log(1-h_\theta(x)) & \text{if } y=0 \end{cases}$

Ex for $y=1$:



Cost = 0 if $y=1$ and $h_\theta(x)=1$

Cost $\to \infty$ if $y=1$ and $h_\theta(x) \to 0$

$J(\theta) = -y \log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))\right]$$

↓ fct to minimize to fit θ

new prediction → $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ → $P(y=1 \mid x;\theta)$

GD: $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$

↳ $\theta_j := \theta_j - \alpha \sum_{j=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$ → identical to lin. reg.

$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

IMPLEMENTATION

Compute ① $J(\theta)$
② $\frac{\partial}{\partial \theta_j} J(\theta)$

EX: $\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$ $J(\theta) = (\theta_1 - 5)^2 + (\theta_2 - 5)^2$

$$\begin{cases} \frac{\partial}{\partial \theta_1} J(\theta) = 2(\theta_1 - 5) \\ \frac{\partial}{\partial \theta_2} J(\theta) = 2(\theta_2 - 5) \end{cases}$$

→ fminunc

7

MULTICLASS CLASSIFICATION

→ ONE VS. ALL

$$h_\theta^{(i)}(x) = P(y = i \mid x; \theta) \qquad (i = 1, 2, 3, \ldots n)$$

(n: nbr of classes)

↳ at prediction → Pick class i that maximizes $\max_i h_\theta^{(i)}(x)$

OVERFITTING

→ Underfitting = high bias

→ Overfitting = high variance

↳ too many features

↳ Fits the training set very well but fails to generalize to new samples

SOLUTIONS

→ ① Reduce number of features

→ ② Regularization

REGULARIZATION

Initial : $\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

↳ REG : $\min_\theta \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + 1000\theta_3^2 + 1000\theta_4^2$

↘ ↙

we need θ_3 and θ_4 very close to 0

→ Reg. all params. :

$$\min_\theta \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2$$

λ → reg. parameter

$\lambda \uparrow$ → underfitting

$\lambda \downarrow$ → overfitting (no reg)

REG. LIN. REG.

New GD:

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha \left[\left(\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \right) + \frac{\lambda}{m} \theta_j \right]$$

$j \in [1, 2, \ldots, n]$

$$\theta_j := \theta_j \left(1 - \alpha \frac{\lambda}{m}\right) - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

↳ $\left(1 - \frac{\alpha\lambda}{m}\right)$ always < 1

↳ reducing value of θ_j on every update

New Normal Eqn:

$$\theta = (X^T X + \lambda L)^{-1} X^T y$$

$$L = \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \end{bmatrix}_{n+1 \times n+1}$$

REG. LOG. REG.

New GD:

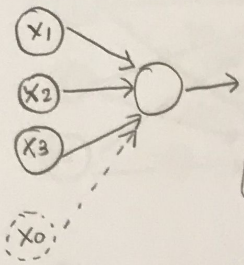
→ Same GD as LIN.REG.

→ But $h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$

↳ $J(\theta) = \left[-\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \log(h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}) \right] + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$

WEEK 4

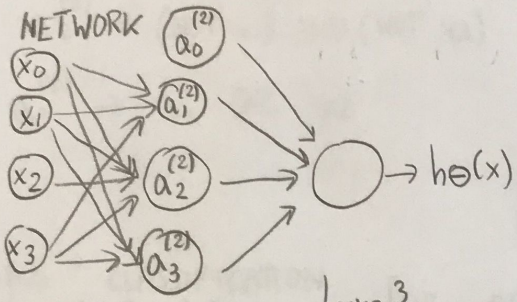
NEURON MODEL: LOGISTIC UNIT



$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ → sigmoid activation fct

NEURAL NETWORK



$\Theta^{(j)}$ → matrix of weights mapping from layer j to $j+1$

Ex: $a_2^{(2)} = g\left(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3\right)$

$h_\theta(x) = a_1^{(3)} = g\left(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)}\right)$

$\theta^{(1)} \in \mathbb{R}^{3\times 4}$

→ Network with s_j units in layer j and s_{j+1} units in layer $j+1$ → $\theta^{(j)}$ dim $s_{j+1} \times (s_j + 1)$

→ EX

$\theta^{(1)} \rightarrow 3\times 4$

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \Theta_{10} & \Theta_{11} & \Theta_{12} & \Theta_{13} \\ \Theta_{20} & \Theta_{21} & \Theta_{22} & \Theta_{23} \\ \Theta_{30} & \Theta_{31} & \Theta_{32} & \Theta_{33} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$3\times 1 \qquad 3\times 4 \qquad 4\times 1$

$\theta^{(2)} \rightarrow$

$$[a_1] = \begin{bmatrix} \Theta_{10} & \Theta_{11} & \Theta_{12} & \Theta_{13} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$1\times 1 \qquad 1\times 4 \qquad 4\times 1$

10

FORWARD PROPAGATION

$$\begin{cases} a_1^{(2)} = g(z_1^{(2)}) \\ a_2^{(2)} = g(z_2^{(2)}) \\ a_3^{(2)} = g(z_3^{(2)}) \end{cases} \rightarrow z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix} \rightarrow \begin{cases} z^{(2)} = \Theta^{(1)} x = \Theta^{(1)} a^{(1)} \\ a^{(2)} = g(z^{(2)}) \end{cases}$$

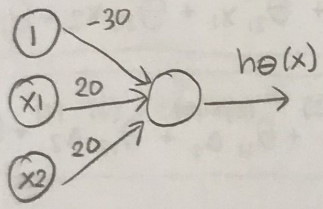
$$\downarrow$$

$$\begin{cases} z^{(3)} = \Theta^{(2)} a^{(2)} \\ h_\theta(x) = a^{(3)} = g(z^{(3)}) \end{cases}$$

→ NNs learn their own features

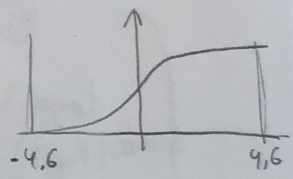
EXAMPLES:

LOGICAL AND



$\theta^{(1)} = [-30 \quad 20 \quad 20]$

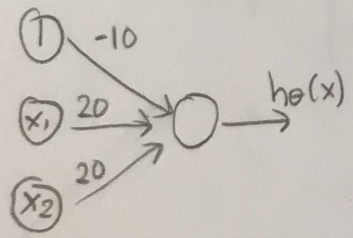
$g(z) \rightarrow$ logistic



x_1	x_2	$h_\theta(x) \approx$
0	0	0
0	1	0
1	0	0
1	1	1

$h_\theta(x) = g(-30 + 20x_1 + 20x_2)$

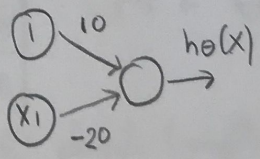
LOGICAL OR



x_1	x_2	$h_\theta(x) \approx$
0	0	0
0	1	1
1	0	1
1	1	1

$h_\theta(x) = g(-10 + 20x_1 + 20x_2)$

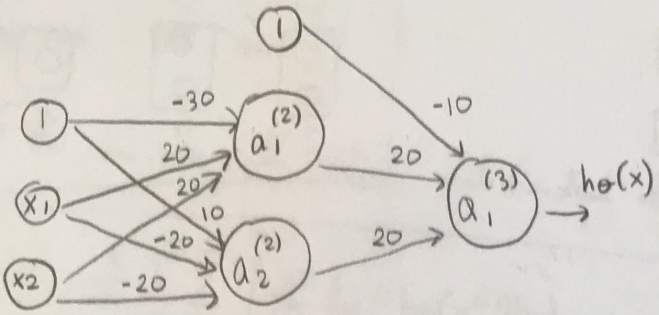
NOT



x_1	$h_\theta(x) \approx$
0	1
1	0

$h_\theta(x) = g(10 - 20x_1)$

XNOR

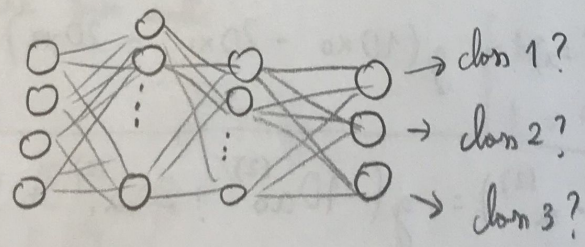


$$\begin{cases} a_1^{(2)} \rightarrow x_1 \text{ AND } x_2 \\ a_2^{(2)} \rightarrow (\text{NOT } x_1) \text{ AND } (\text{NOT } x_2) \\ a_1^{(3)} \rightarrow x_1 \text{ OR } x_2 \end{cases}$$

x_1	x_2	$a_1^{(2)}$	$a_2^{(2)}$	$h_\Theta(x) \approx$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

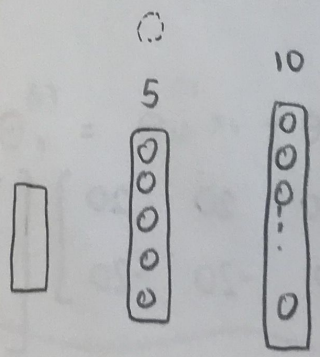
MULTICLASS CLASSIFICATION

3 classes



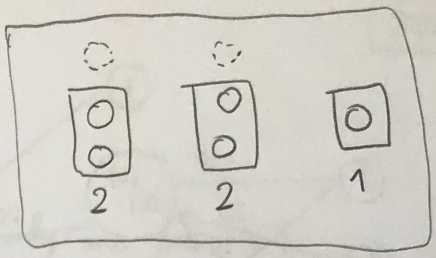
$$h_\Theta(x) \approx \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \rightarrow \text{class 2}$$

Example: { 5 hidden units; 10 classes (output) } →



$\Theta^{(2)} \rightarrow 10 \times (5+1) \rightarrow$ 60 elements (weights)

(12) XNOR IN DETAIL



$$\begin{cases} \text{AND} \rightarrow \Theta^{(1)} = [-30 \quad 20 \quad 20] \\ \text{NOR} \rightarrow \Theta^{(1)} = [10 \quad -20 \quad -20] \\ \text{OR} \rightarrow \Theta^{(1)} = [-10 \quad 20 \quad 20] \end{cases}$$

$\downarrow$

$$\text{XNOR}: \begin{cases} \Theta^{(1)} = \begin{bmatrix} -30 & 20 & 20 \\ 10 & -20 & -20 \end{bmatrix} \\ \Theta^{(2)} = [-10 \quad 20 \quad 20] \end{cases} \rightarrow \begin{cases} a^{(2)} = g(\Theta^{(1)} \cdot x) \\ a^{(3)} = g(\Theta^{(2)} \cdot a^{(2)}) \\ h_\Theta(x) = a^{(3)} \end{cases}$$

$$\rightarrow a_1^{(2)} = g(\Theta_1^{(1)} \cdot x) = g(-30x_0 + 20x_1 + 20x_2)$$

$$\rightarrow a_2^{(2)} = g(\Theta_2^{(1)} \cdot x) = g(10x_0 - 20x_1 - 20x_2)$$

$$\rightarrow \boxed{a_1^{(3)} = g(\Theta_1^{(2)} \cdot a^{(2)}) = g(-10a_0^{(2)} + 20a_1^{(2)} + 20a_2^{(2)})}$$

Ex:

$$\begin{cases} Z^{(2)} = \Theta^{(1)} x = \begin{bmatrix} -30 & 20 & 20 \\ 10 & -20 & -20 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \\ Z_1^{(2)} = -30 + 20x_1 + 20x_2 \rightarrow \boxed{a_1^{(2)} = g(Z_1^{(2)})} \end{cases}$$