SCHEDULE

	MON	TUE	WED	THU	FRI

Important dates

Time Zones (US)

International

New York	1 p.m
Los Angeles	10 a.m
London	6 p.m
Moscow	9 p.m
Tokyo	3 a.m
Beijing	2 a.m
Rome	7 p.m
Honk Kong	2 a.m
Bangkok	1 a.m
Sydney	5 a.m
Paris	7 p.m
Cairo	8 p.m
Mumbai	12 a.m
Shenzen	2 a.m
Delhi	12 a.m
Istanbul	9 p.m
Dubai	10 p.m
Vienna	7 p.m

*Does not take US daylight savings into account

---

this is so thin, what the fuck

check check

Week 1 notes

$h_\theta(x)$ → prediction model

y → actual model

most commonly used

squared error function

$$\underset{\theta_0,\,\theta_1}{\text{minimize}} \; \frac{1}{2m} \sum_{i=1}^{m} \left[h_\theta(x^{(i)}) - y^{(i)} \right]^2$$

m → # of training set

$x^{(i)}$ → training set

$$h_\theta(x^{(i)}) = \theta_0 + \theta_1 \cdot x^{(i)}$$

$$\text{cost function} = J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left[h_\theta(x^{(i)}) - y^{(i)} \right]^2$$

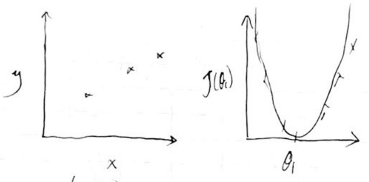
$$\underset{\theta_0,\,\theta_1}{\text{minimize}} \; J(\theta_0, \theta_1)$$

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

Parameters: θ_0, θ_1

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

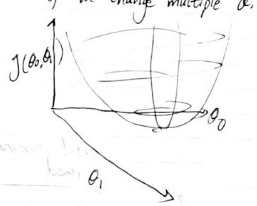
Goal: $\underset{\theta_0,\,\theta_1}{\text{minimize}} \; J(\theta_0, \theta_1)$



this is changing θ_1.

$h_\theta(x)$

if we change multiple θ, then we get high dimensional graphs



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

$temp0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

$temp1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

$\theta_0 := temp0$

$\theta_1 := temp1$

this reposition θ_0, θ_1, and give us a new value for $J(\theta_0, \theta_1)$

A' → transpose

$inv(A)$ → inverse

Week 2 Notes

multiple features n. $x^{(i)}$ = input (features) of i^{th} training example

$x_j^{(i)}$ = value of feature j

size x_1	# of rooms, x_2	floors, x_3	age of home, x_4	price y
2104	5	1	45	460
1416	3	2	40	232
1410				
852				

$x_1^{(4)} = 852$ → 4^{th} row of feature 1.

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

example $= 80 + 0.1 \cdot x_1 + 0.01 \cdot x_2 + 3 \cdot x_3 - 2 \cdot x_4$

$x_0^{(i)} = x_0 = 1$

$1 \times n{+}1 \times n{+}1 \times 1 = 1 \times 1$

if $x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}$, $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$ then $h_\theta(x) = \theta^T \cdot x$

$(n+1)\times(1)$ $(n+1)\times(1)$

$\boxed{\text{Hypothesis}}$ $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 \cdots \theta_n \cdot x_n$

$\boxed{\text{Parameters}}$: $\theta_0, \theta_1, \theta_2$ --- aka θ

$\boxed{\text{Cost function}}$: $J(\theta_0, \theta_1, \theta_2 \cdots \theta_n) = J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

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

$= \frac{1}{2m} \sum_{i=1}^{m} \left(\left(\sum_{j=0}^{n} \theta_j x_j^{i} \right) - y^{(i)} \right)^2$

$\boxed{\text{Gradient descent}}$

Repeat {

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

}

$\boxed{\text{Partial Derivative Demonstration}}$

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

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} \left[\frac{1}{2m} \sum_{i=1}^{m} \left(\left(\sum_{j=0}^{n} \theta_j x_j^{(i)} \right) - y^{(i)} \right)^2 \right]$$

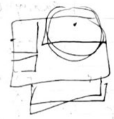
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} \left[\frac{1}{2m} \sum_{i=1}^{m} \left((\theta_0 x_0 + \theta_1 \cdot x_1 + \dots + \theta_n \cdot x_n) - y^{(i)} \right)^2 \right]$$

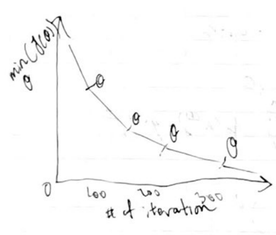
$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left((\theta_0 x_0 + \theta_1 \cdot x_1 + \dots + \theta_n \cdot x_n) - y^{(i)} \right) \cdot x_j^{(i)} \right]$$

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

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

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





bruh, this shit trash.

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

pinv(X' * X) * X' * Y

$\boxed{\text{Grdient Descent}}$

Σ, need α,

works well n large.

$\boxed{\text{Normal Equation}}$

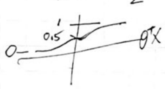
matrix, find θ.

no good if n large.

$\boxed{\text{Week 3 Notes}}$

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

$$\frac{1}{1+e^{-0}} = \frac{1}{1+1} = \frac{1}{2} = 0.5$$



$\boxed{\text{example:}}$

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

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

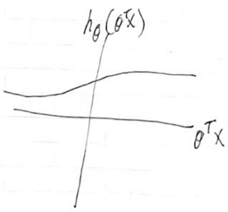
$\boxed{\text{example}}$

$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$

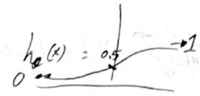
$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix}$ predict "y=1" if $\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 \geq 0$

$h_\theta(x) \geq 0.5 \to y = 1$

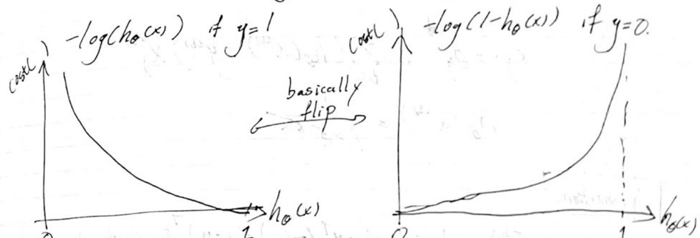
$h_\theta(x) < 0.5 \to y = 0$



$\boxed{\text{Logistic Regression Cost Function}}$



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



y=1, and $h_\theta(x)$ predicts 0. large penalty

y=1, and $h_\theta(x)$ predicts 1. 0 penalty.

flip

y=1, and $h_\theta(x)$ predicts 0, large penalty

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

no negative

negative $-y \cdot \log(h_\theta(x)) - (1-y) \cdot \log(1-h_\theta(x))$

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

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

$\min_{\theta} J(\theta)$ repeat $\{$

$$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta) \qquad \theta = \begin{bmatrix}\theta_0\\ \theta_1\\ \vdots\\ \theta_n\end{bmatrix}$$

$\}$

$\downarrow$

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

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

$\boxed{\text{Vectorization}}$

$$h = g(X\theta) \qquad J(\theta) = \frac{1}{m}\cdot\left(-y^T\log(h) - (1-y)^T\log(1-h)\right)$$

$$\theta := \theta - \frac{\alpha}{m}X^T(g(X\theta) - \vec{y})$$

$\boxed{\text{Regularization}}$

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

$$\left.\begin{aligned}&\theta_0 + \theta_1 x + \theta_2 x^2 + 0\cdot x^3 + 0\cdot x^4\\ &\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4\end{aligned}\right\}\text{ same format}$$

Repeat $\{$

$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})\cdot x_0^{(i)} \qquad j=0$$

$$\theta_j := \theta_j - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})\,x_j^{(i)} + \frac{\lambda}{m}\theta_j\right] \qquad j=1,2,3,\ldots,n$$

$\}$

$\downarrow$ can be rewritten as

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

$$X = \begin{bmatrix}(x^{(1)})^T\\ \vdots\\ (x^{(m)})^T\end{bmatrix} \longrightarrow \text{first data.}$$

$m \le n$

\# examples   \# features

$$\theta = (X^TX)^{-1}X^Ty$$

if $\lambda > 0$

$$\theta = \left(X^TX + \lambda\begin{bmatrix}0 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1\end{bmatrix}\right)^{-1}X^Ty$$

use this if no inverse: pinv vs inv

Logistic regression without regularization

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

With regularization

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

derivative:

$$\theta_0 \to \frac{1}{m}\sum_{i=1}^{m}(h_\theta(X^{(i)}) - y^{(i)})X_0^{(i)}$$

$$\theta_j \to \frac{1}{m}\sum_{i=1}^{m}(h_\theta(X^{(i)}) - y^{(i)})X_j^{(i)} + \frac{\lambda}{m}\theta_j$$

EX2 workspace

$$X = \begin{array}{c} \\ 100 \end{array}\begin{array}{c} 3 \\ \begin{bmatrix} 1 & EX1 & EX2 \\ \vdots & \vdots & \vdots \\ \vdots & & \end{bmatrix}\end{array} \quad Y = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \vdots \end{bmatrix}$$

$\boxed{EX2.m}$

$m \times n$
100×2

initial thetas zero (3, 1)
theta (3, 1)

h = sigmoid(X.theta);

$$h = \text{sigmoid}\begin{bmatrix} \theta_0 + EX1\cdot\theta_1 + EX2\cdot\theta_2 \\ \vdots \\ \vdots \end{bmatrix} \to h = \begin{bmatrix} \frac{1}{1+e^{-(\theta_0 + EX_1\cdot\theta_1 + EX_2\cdot\theta_2)}} \\ \vdots \end{bmatrix}$$

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \text{cost}(h_\theta(x^{(i)}), y^{(i)})$$

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

Regularization

$$\frac{\partial J(\theta)}{\partial\theta_j} \to \frac{1}{m}\sum_{i=1}^{m} h_\theta(x^{(i)}) - y^{(i)})\cdot x_j^{(i)}$$

h=sigmoid(X * theta)

octave code

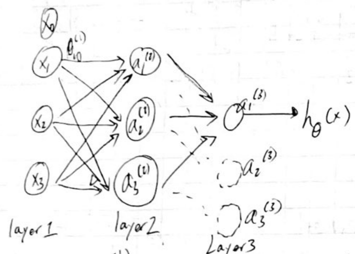
```
J(θ) = 1/m sum[-y.*log(h) - (1-y).*log(1-h)]

∂J(θ)/∂θj   (1/m) sum[(h-y).*X]
```

$$X.*Y = \begin{bmatrix} 5 \\ 6 \\ 6 \end{bmatrix} \qquad X.'*Y = 17$$

$$X = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad Y = \begin{bmatrix} 5 \\ 3 \\ 2 \end{bmatrix}$$

Week 4 Notes



$a_i^{(j)}$ = activation of unit i in layer j

$a_1^{(2)}$ = activation of unit 1 in layer 2.

$\Theta^{(j)}$ = matrix of weights controlling function mapping from layer j to layer $j+1$.

$$a_1^{(2)} = g(\Theta_{10}^{(1)} X_0 + \Theta_{11}^{(1)} X_1 + \Theta_{12}^{(1)} X_2 + \Theta_{13}^{(1)} X_3) \quad z_1^{(2)}$$

$$a_2^{(2)} = g(\Theta_{20}^{(1)} X_0 + \Theta_{21}^{(1)} X_1 + \Theta_{22}^{(1)} X_2 + \Theta_{23}^{(1)} X_3) \quad z_2^{(2)}$$

$$a_3^{(2)} = g(\Theta_{30}^{(1)} X_0 + \Theta_{31}^{(1)} X_1 + \Theta_{32}^{(1)} X_2 + \Theta_{33}^{(1)} X_3) \quad z_3^{(2)}$$

$$h_\theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)}) \quad z_1^{(3)}$$

$$a_1^{(2)} = g(z_1^{(2)}) \qquad z_1^{(2)} = \Theta_{10}^{(1)} X_0 + \Theta_{11}^{(1)} X_1 + \Theta_{12}^{(1)} X_2 + \Theta_{13}^{(1)} X_3$$

$$a_2^{(2)} = g(z_2^{(2)})$$

$$a_3^{(2)} = g(z_3^{(2)}) \qquad z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix} \qquad a^{(2)} = g(z^{(2)})$$

$$z^{(3)} = \Theta^{(2)} \cdot a^{(2)}$$

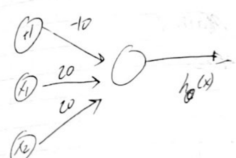
$$z_k^{(2)} = \Theta_{k,0}^{(1)} x_0 + \Theta_{k,1}^{(1)} x_1 + \dots + \Theta_{k,n}^{(1)} x_n$$

layer $j=2$ node k, variable z.

$$x = \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix} \quad z^{(j)} = \begin{bmatrix} z_1^{(j)} \\ z_2^{(j)} \\ \dots \\ z_n^{(j)} \end{bmatrix}$$

$$z_1^{(3)} = \Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)}$$

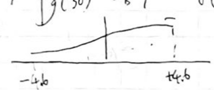
$$z^{(j)} = \Theta^{(j-1)} a^{(j-1)}$$



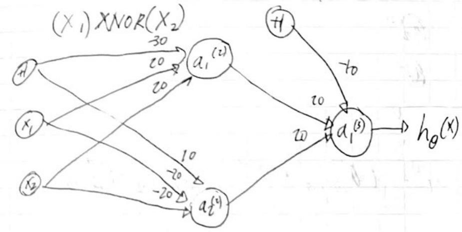
$x_1, x_2 \in \{0, 1\}$ $\quad g(-10 + 20x_1 + 20x_2)$

x_1	x_2	$h_\theta(x)$			
0	0	$g(-10) \to 0$		11	1
0	1	$g(10) \to 1$		10	0
1	0	$g(10) \to 1$		01	0
1	1	$g(30) \to 1$		00	0

$\Theta^1 = [-10, 20, 20]$



(x_1) XNOR (x_2)



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

EX3 Workspace

oneVsAll [X, y, num_labels, lambda)

Imagedata — whether it is X or y — 10 — 0.1

5000 × 400 — 5000 × 1 — $m=5000$, $n=400$

each image 20x20 Grayscale pixel

all_theta = zeroos Matrix 10 × 402

X = 5000 × 401

10 classes for 10 numbers

$a_1^{(2)}$, $a_2^{(2)}$, $a_3^{(2)}$, $a_4^{(2)}$

$$a_1^{(2)} = g(\theta_{10}^{(1)}X_0 + \theta_{11}^{(1)}X_1 + \theta_{12}^{(1)}X_2 + \theta_{13}^{(1)}X_3)$$

classifier — lrCostFunction (theta, X, y, lambda)

10 different classifier for each number

lrCostFunction (theta, X, y, lambda)

@t — t — X — y==c — lambda

$X = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$ — X(:) = 1 5 2 6 3 7 4 8

X(1,:) = 1, 2, 3, 4

X(2,:) = 5, 6, 7, 8

options = optimset('GradObj', 'on', 'MaxIter', 50);

max iteration 50

for c = 1: num_labels = for c = 1:10. all_theta = zeros(10, 401)

all_theta (c,:) = ... this means continue

fmincg(

401 θs for each pixel intensity

10 × 401

return the θ parameters for each of the 10 classes

for peepee = 1: size(X, 1)

X(PeePee, :)

1 × 401

all_theta 10×401

Predict One Vs All (all_theta , X)

$X = 5000 \times 400$

$\downarrow$

$X = 5000 \times 401 \rightarrow 1 \times 401$

all_theta = 10×401

$X = 5000 \times 400$

$m = 5000$

num_labels = 10

$p = 5000 \times 1$

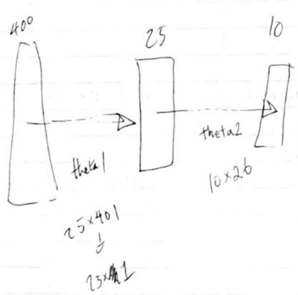
theta1 = 25×401

theta2 = 10×26

X ' all_theta '

p(1,:) = X(1,:) * (all_theta ')

gives you a 1x10 matrix of probabilites for the first image 20x20.



$25 \left[\text{sig}(\overset{401}{\downarrow}) \right]$

$\downarrow$

$\begin{array}{cc} & 1 \\ 25 & \left[\quad \right] \end{array} \rightarrow \begin{array}{cc} & 1 \\ 26 & \begin{bmatrix} \text{one ( )} \\ \vdots \end{bmatrix} \end{array}$

$\begin{array}{cc} & 401 \\ 5000 & \left[\quad \right] \end{array} \rightarrow \begin{array}{cc} & 401 \\ 1 & \left[\quad \right] \end{array}$

X X(biggpenis, :)

theta1 = 25×401.

$\text{sig}\left(\begin{array}{cc} & 1 \\ 25 & \left[\quad \right] \end{array} \right) \Rightarrow \begin{array}{cc} & 1 \\ 25 & \left[\quad \right] \end{array}$

sigmoid[25×401 , X(biggpenis,:) ']

$\downarrow$

theta1

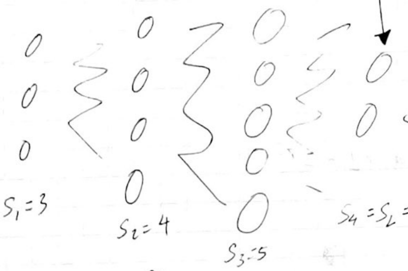
$\begin{array}{cc} & 1 \\ 26 & \left[\quad \right] \end{array}$ ~~26x1~~ biggerpenis

$\begin{array}{cc} & 26 \\ 10 & \left[\quad \right] \end{array}$ theta2

(Theta2 * biggerpenis) = 10×1 $\begin{array}{cc} & 1 \\ 10 & \left[\quad \right] \end{array}$

Week 5 Notes

$h_\theta(x)$



$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$

L = # of layers in network

S_l = # of units in a layer

$S_1 = 3$ $\quad S_2 = 4$ $\quad S_3 = 5$ $\quad S_4 = S_L = 2$

$L = 4$

Binary Classification

$y = 0$ or 1

$S_L = 1 \quad h_\theta(x) \in \mathbb{R}$

MultiClass classification (k classes)

$y \in \mathbb{R}^K \quad \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \begin{bmatrix}0\\1\\0\\0\end{bmatrix} \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$

$S_L = K \quad k \geq 3$

$K K \quad K_k$

Logistic Regression

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

Neural Network

$h_\theta(x) \in \mathbb{R}^K \quad (h_\theta(x))_i = i^{th}$ output

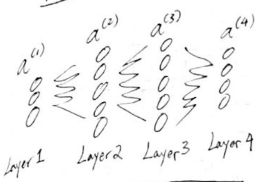
$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log(h_\theta(x^{(i)}))_k + (1-y_k^{(i)}) \log(1-(h_\theta(x^{(i)}))_k)\right]$$
$$+ \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{S_l}\sum_{j=1}^{S_{l+1}} (\theta_{ji}^{(l)})^2$$

Gradient computation:

$\min_\theta J(\theta)$ $\quad$ compute $J(\theta)$, $\frac{\partial}{\partial \theta_{ij}^{(l)}} J(\theta)$ $\quad \frac{\partial}{\partial \theta_{ij}^{(l)}} J(\theta)$

Given one training set (x, y)

Forward propagation



activation values

$a^{(1)} = x$

$z^{(2)} = \theta^{(1)} a^{(1)}$

$a^{(2)} = g(z^{(2)})$ (add $a_0^{(2)}$)

$z^{(3)} = \theta^{(2)} a^{(2)}$

$a^{(3)} = g(z^{(3)})$ (add $a_0^{(3)}$)

$z^{(4)} = \theta^{(3)} a^{(3)}$

$a^{(4)} = h_\theta(x) = g(z^{(4)})$

Backward Propagation

δ_j^{l} = the error of $a_j^{(l)}$ ← layer, ← node

$\delta_j^{(4)} = a_j^{(4)} - y_j \quad$ vectorized → $\delta^{(4)} = a^{(4)} - y$

$(h_\theta(x))_j$

$z^{(3)} = \begin{bmatrix} z_1^{(3)} \\ z_2^{(3)} \\ \vdots \\ z_n^{(3)} \end{bmatrix}$ → $\theta_{10}^{(1)} a_0^{(2)} + \theta_{11}^{(1)} a_1^{(2)} \ldots$

derivative

$\delta^{(3)} = (\theta^{(3)})^T \delta^{(4)} .* g'(z^{(3)})$

$\delta^{(2)} = (\theta^{(2)})^T \delta^{(3)} .* g'(z^{(2)})$

$a^{(3)} .* (1 - a^{(3)})$

$\frac{\partial}{\partial \theta_{ij}^{(l)}} J(\theta) = a_j^{(l)} \delta_i^{(l+1)}$ $\quad$ ignore λ; if $\lambda = 0$

$\Delta_{ij}^{(l)}$ ← capital δ, use to capture $\frac{\partial}{\partial \theta_{ij}^{(l)}} J(\theta)$

for $i = 1$ to m
- set $a^{(1)} = x^{(i)}$
- forward propagation to compute $a^{(l)}$ $l = 2, 3, \ldots, L$
- compute $\delta^{(L)} = a^{(L)} - y^{(i)}$ ← actual answer.
- then $\delta^{(L-1)}, \delta^{(L-2)}, \ldots, \delta^{(2)}$ ← going backwards.
- $\Delta_{ij}^{(l)} := \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$

$\boxed{\text{Two cases}}$

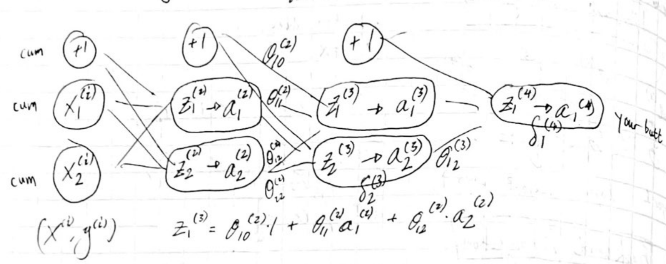
$D_{ij}^{(l)} := \frac{1}{m}\Delta_{ij}^{(l)} + \lambda \theta_{ij}^{(l)}$ if $j \neq 0$

$D_{ij}^{(l)} := \frac{1}{m}\Delta_{ij}^{(l)}$ if $j = 0$

complex math proof shows that $\frac{\partial}{\partial \theta_{ij}^{(l)}} J(\theta) = D_{ij}^{(l)}$

Forward Propagation

a = activation value



$(x^{(i)}, y^{(i)})$ $\qquad z_1^{(3)} = \theta_{10}^{(2)} \cdot 1 + \theta_{11}^{(2)} a_1^{(2)} + \theta_{12}^{(2)} \cdot a_2^{(2)}$

$\delta_j^{(l)}$ = "error" of cost for $a_j^{(l)}$ unit j layer l

$\delta_j^{(l)} = \frac{\partial}{\partial z_j^{(l)}} \cdot cost(i)$ where $cost(i) = y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log h_\theta(x^{(i)})$

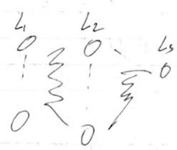
$\delta_1^{(4)} = y^{(i)} - a_1^{(4)}$ $\qquad \delta_2^{(3)} = \theta_{12}^{(3)} \cdot \delta_1^{(4)}$ $\qquad \delta_2^{(2)} = \theta_{12}^{(2)} \delta_1^{(3)} + \theta_{22}^{(2)} \cdot \delta_2^{(3)}$

whatever output you choose

$\delta_2^{(2)} = \delta_2^{(3)} + \theta_{22}^{(2)} \cdot \delta_2^{(3)}$

$\delta_2^{(2)} = \delta_2^{(3)}(1 + \theta_{22}^{(2)})$

(θ) parameter (D) gradients



$S_1 = 10 \quad S_2 = 10 \quad S_3 = 1$

$\theta^{(1)} \in \mathbb{R}^{10 \times 11} \quad \theta^{(2)} \in \mathbb{R}^{10 \times 11} \quad \theta^{(3)} \in \mathbb{R}^{1 \times 11}$

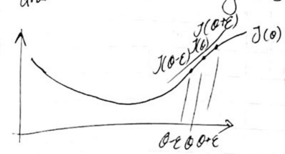
$S_2 \to 10$ nodes $S_1 \to 11$ nodes $+1$

$D^{(1)} \in \mathbb{R}^{10 \times 11} \quad D^{(2)} \in \mathbb{R}^{10 \times 11} \quad D^{(3)} \in \mathbb{R}^{1 \times 11}$

```
thetaVec = [ Theta1 (:); Theta2 (:); Theta3(:)];
DVec = [D1(:); D2(:); D3(:)];
Theta1 = reshape ( thetaVec (1:110), 10, 11);
Theta2 = reshape ( thetaVec (111:220), 10, 11);
Theta3 = reshape ( thetaVec (221:231), 1, 11);
```

← reshape back to original

Gradient could have a bug. So use gradient checking



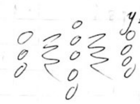
$$\frac{d}{d\theta} J(\theta) \approx \frac{J(\theta+\epsilon) - J(\theta-\epsilon)}{2\epsilon}$$

$$\text{gradApprox} = \frac{J(\text{theta} + \text{EPSILON}) - J(\text{theta} - \text{EPSILON})}{2 \cdot \text{EPSILON}}$$

$$\frac{\partial}{\partial \theta_j} J(\theta) \approx \frac{J(\theta_1 \ldots \theta_j + \epsilon, \ldots, \theta_n) - J(\theta_1 \ldots \theta_j - \epsilon \ldots \theta_n)}{2\epsilon}$$

gradApprox ≈ DVec

Putting it together



$y = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$