Cost Function

* **Training Set:** m - examples $\{(x^{(1)}, y^{(1)}), \ldots \ldots, (x^{(m)}, y^{(m)})\}$

$$x \in \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad \begin{matrix} x_0 = 1 \\ y \in = \{0, 1\} \end{matrix}$$

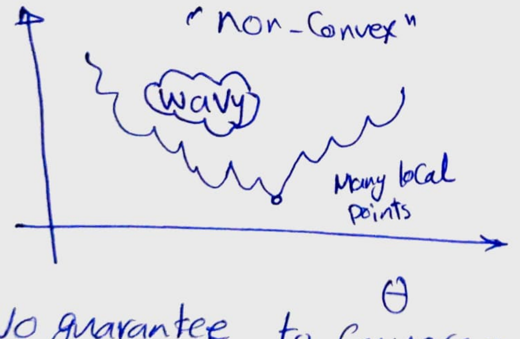
$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

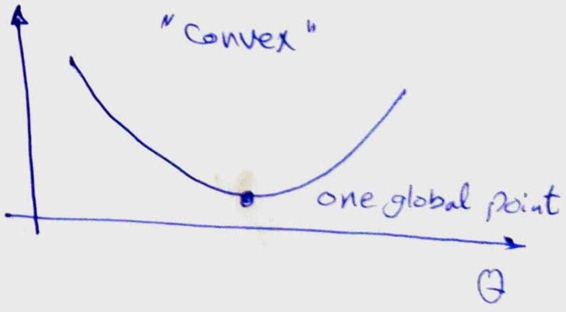
How to Choose Parameters θ ?

* For Linear Regression: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \underbrace{\frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2}_{Cost\,(h(x^{(i)}) - y^{(i)})}$ (loss)

Cost Function $= \frac{1}{2} [h_\theta(x) - y]^2$

If we apply this eqn to logistic Regression where $h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$ it will produce a non-Convex fn.

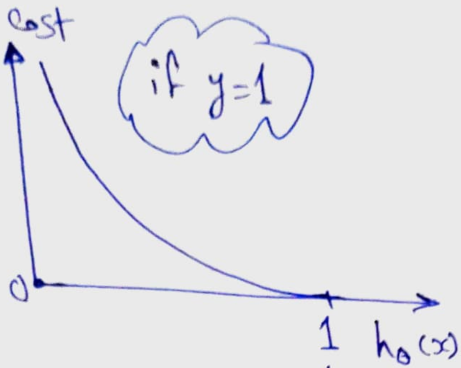


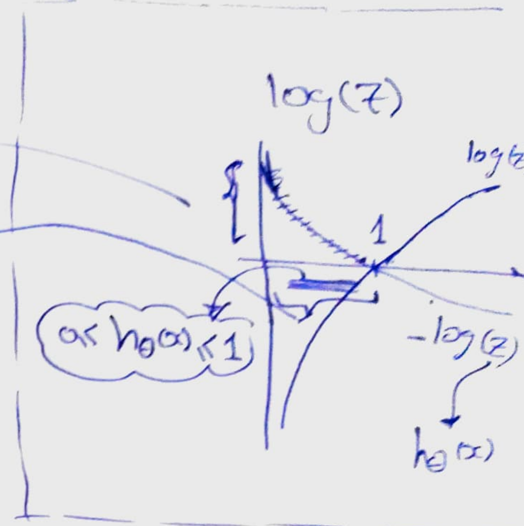


* No guarantee to Converge at min. point if we applied gradient decsent.

* Logistic Regression Cost Function:

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

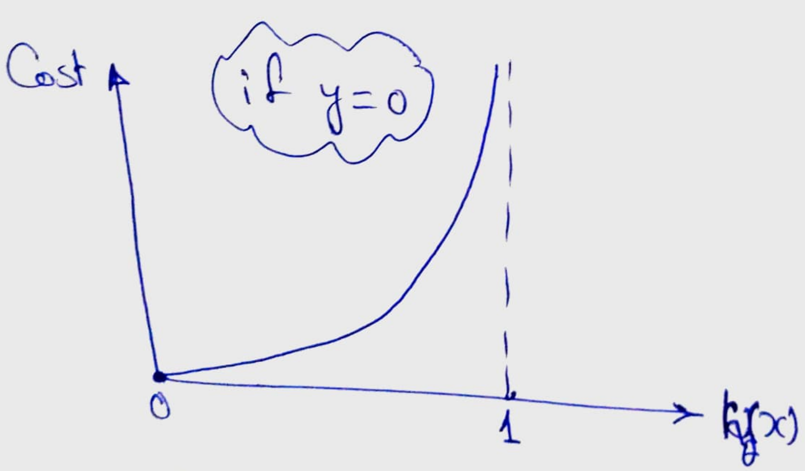


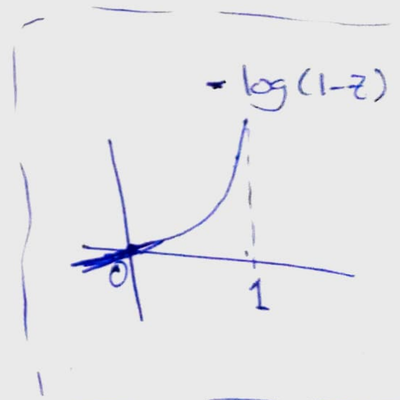


$y=1, \; h_\theta(x) = 1 \rightarrow \boxed{\text{Cost} = 0}$

<u>But</u>, as $h_\theta(x) \rightarrow 0 \; (P(y=1|_{x;\theta}) = 0)$
$\text{Cost} \rightarrow \infty$

Captures intuition that $h_\theta(x) = 0 \; (P(y=1|_{x;\theta}) = 0)$ but actually $\underline{y=1}$, So we penalize learning algorithm by a very large cost.





• <u>Notes</u>:

- If $h_\theta(x) = y \rightarrow \text{Cost} = 0$ (for $y=0$ and $y=1$)
- If $y=0$, then ($\text{Cost} \rightarrow \infty$ as $h_\theta(x) \rightarrow 1$)
- Regardless of whether $y=0$ or $y=1$, if $h_\theta(x) = 0.5$, then $\text{Cost} > 0$

Logistic Regression Cost function

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

for Single example

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

Compressing it in one eqn:

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

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

And why do we choose this particular function, while it may there Could other Cost functions. This Cost function can be derived from statistics using the principle of Maximum Likelihood Estimation which is an idea in statistics for how to efficiently find parameters data for different models. And it also has a nice property that it is Convex

So, we want to fit parameter θ to make $J(\theta)$ is minimum

to make a prediction given new x:

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

→ we will use $\boxed{\text{Gradient Descent}}$

Gradient Descent

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

} Simulataneously update all θ_j

* For Logistic Regression

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

} Simulataneously update all θ_j

* It's Actually the same algorithm of gradient Descent for Linear Regression, but:

* Linear Regression: $h_\theta(x) = \theta^T x$
* Logistic Regression: $h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$

→ A vectorized Implementation: (Run faster)

$h = g(X\theta)$

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

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

→ Also, Feature Scaling

↳ helps speed up gradient descent for the log-regression algorithm.