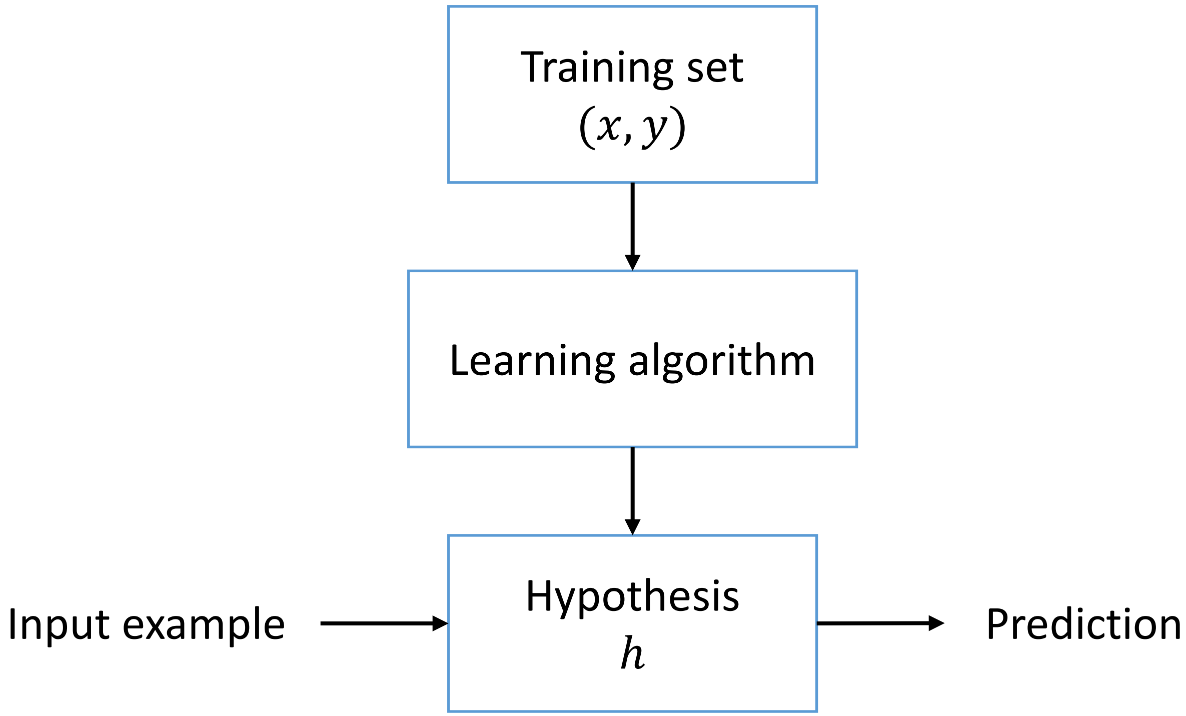
Linear Classification and Regression

Outline

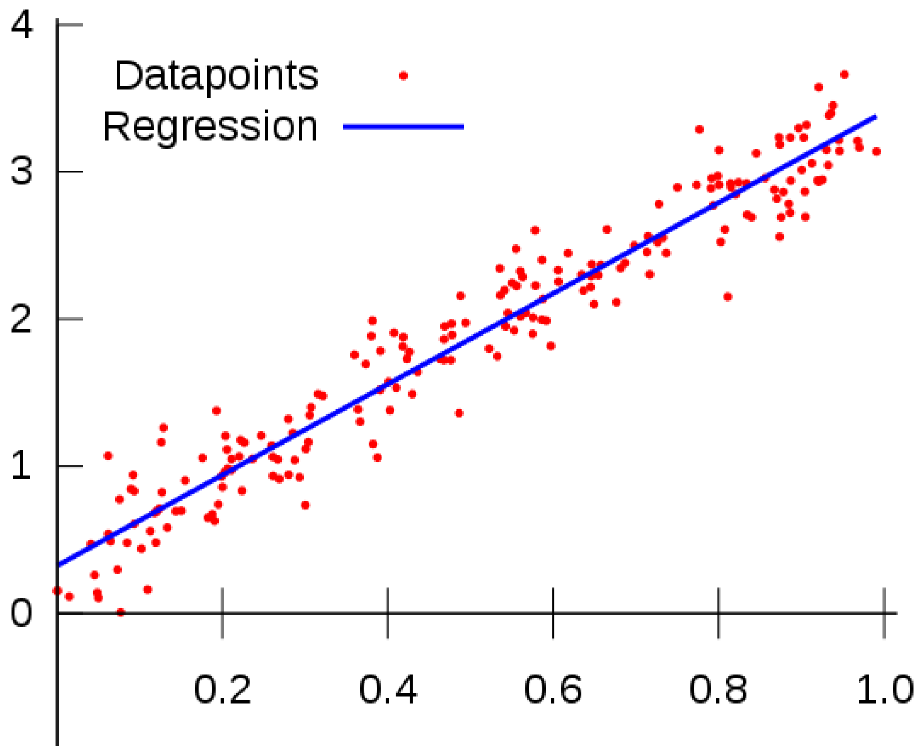
- Linear regression
  - Gradient descent
  - Closed form solution
- Locally weighted regression
- Probabilistic Interpretation of Linear Regression
- Maximum Likelihood estimator
- Logistic regression
- Perceptron
- Support Vector Machines (SVM)
  - Maximizing margin
  - Kernel trick
  - Soft margin

Based on a tutorial by Andrew Ng

Supervised Learning



Linear Regression



Notation

- m : number of training examples
- n : number of features
- x : input variables/features
- y : output variable/target variable
- $(x^{(i)}, y^{(i)})$: i-th training example

Linear Regression

- We assume that there is a linear relation between the output variable and the input features

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$$

- $\theta_1, \theta_2, \ldots, \theta_n$ define the slope of the line and θ_0 represent the bias.

- We can define $x_0 = 1$ for convenience

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

Linear Regression - Cost Function
Least Mean Square Algorithm (Widrow-Hoff)

- How to learn from the training set? How to find the parameters?
- Define the cost/loss function as the sum of squared error of predictions on training data

$$\mathcal{J}(\Theta) = \frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

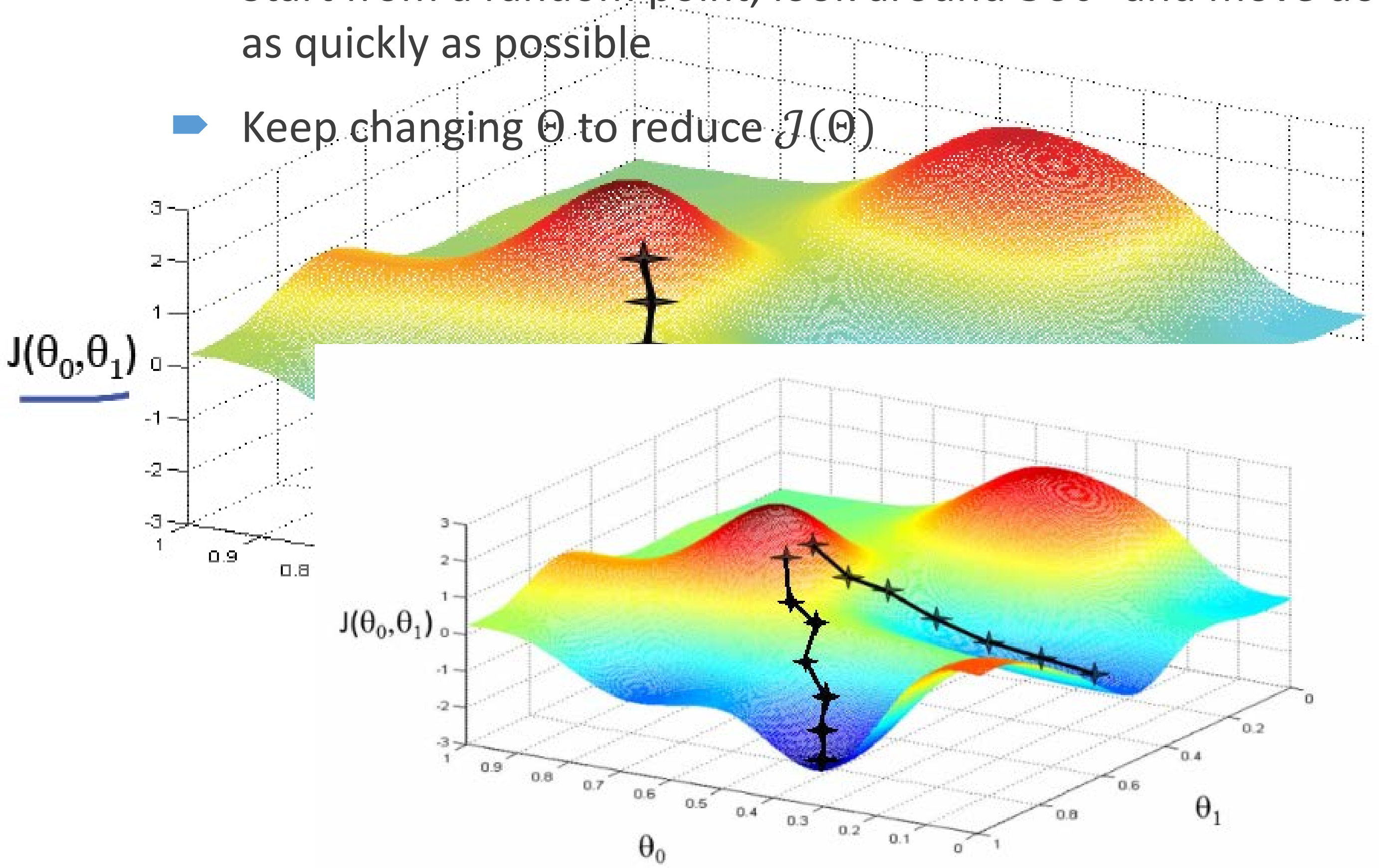
$$\mathcal{J}(\Theta) = \frac{1}{2}\sum_{i=1}^{m}\left(\Theta^T x^{(i)} - y^{(i)}\right)^2$$

- Minimize the cost/loss

$$\min_{\Theta} \mathcal{J}(\Theta)$$

Gradient Descent

- Start from a random point, look around 360° and move downhill as quickly as possible
- Keep changing Θ to reduce $\mathcal{J}(\Theta)$



Linear Regression - Gradient Descent

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} \mathcal{J}(\Theta)$$

$$\frac{\partial}{\partial \theta_j} \mathcal{J}(\Theta) = \frac{\partial}{\partial \theta_j} \left[\frac{1}{2} \sum_{i=1}^{m} \left(\Theta^T x^{(i)} - y^{(i)} \right)^2 \right]$$

$$\frac{\partial}{\partial \theta_j} \mathcal{J}(\Theta) = \sum_{i=1}^{m} \left(\Theta^T x^{(i)} - y^{(i)} \right) \frac{\partial}{\partial \theta_j} \left(\theta_0 + \theta_1 x_1^{(i)} + \cdots + \theta_n x_n^{(i)} - y^{(i)} \right)$$

$$\frac{\partial}{\partial \theta_j} \mathcal{J}(\Theta) = \sum_{i=1}^{m} \left(\Theta^T x^{(i)} - y^{(i)} \right) x_j^{(i)}$$

$$\theta_j = \theta_j - \alpha \sum_{i=1}^{m} \left(\Theta^T x^{(i)} - y^{(i)} \right) x_j^{(i)}$$

Batch Gradient Descent vs. Stochastic Gradient Descent

- Batch Gradient Descent

Repeat until convergence
{

$$\theta_j = \theta_j - \alpha \sum_{i=1}^{m} \left(\Theta^T x^{(i)} - y^{(i)}\right) x_j^{(i)}$$

}

For every example x_j

- Stochastic Gradient Descent

Repeat until convergence
{
for i =1 to m
{

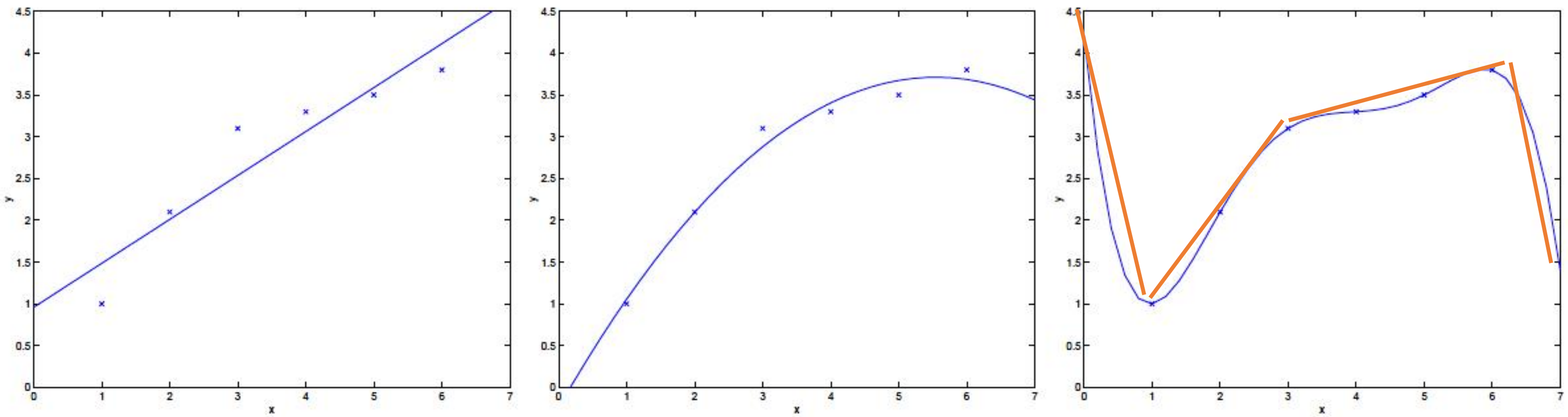
$$\theta_j = \theta_j - \alpha\left(\Theta^T x^{(i)} - y^{(i)}\right) x_j^{(i)}$$

}
}

For every example x_j

there exists a closed form for min $J(\theta)$

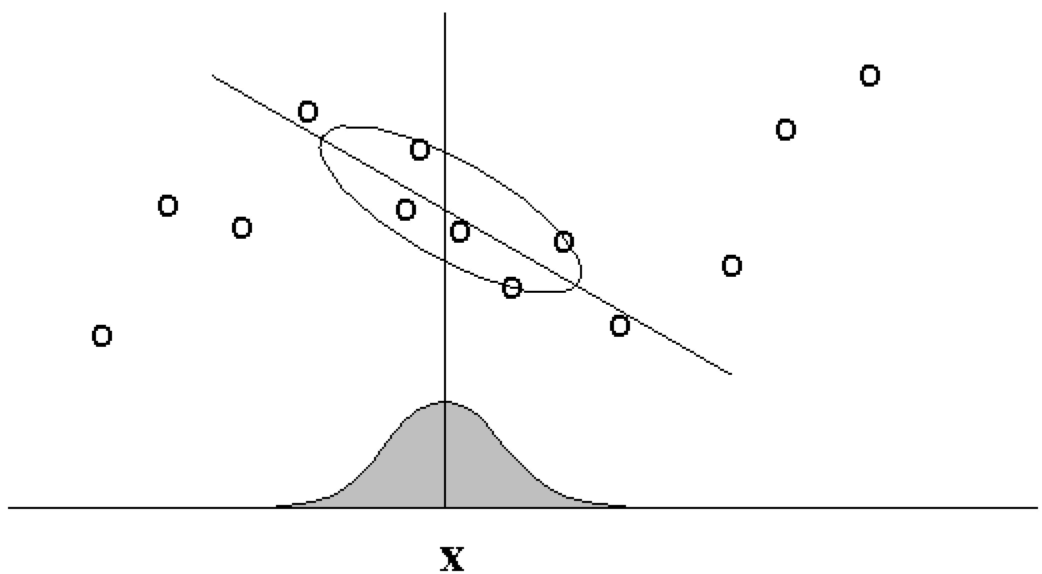
Locally Weighted Regression



Sometimes a simple linear model is not a good fit

Locally Weighted Regression

- LR we have seen is parametric (the θ's; data can be forgotten after training)
- LWR is a non-parametric model (data that needs to be kept to represent the hypothesis is O(m))



Locally Weighted Regression

- For a query point x, fit Θ to minimize

$$\mathcal{J}(\Theta) = \sum_{i=1}^{m} w^{(i)} \left(\Theta^T x^{(i)} - y^{(i)}\right)^2$$

Large error – small weight
Small error – weight unimportant

$$w^{(i)} = \exp\left(-\frac{\left\|x^{(i)} - x\right\|^2}{2\sigma^2}\right)$$

- σ is the bandwidth parameter
- It is computationally quite expensive if you have large training set
  - Improvements has been done using kd-trees, …

Probabilistic Interpretation of Linear Regression

- Lets assume

$$y^{(i)} = \Theta^T x^{(i)} + \epsilon^{(i)}$$

- $\epsilon^{(i)}$ is the error, IID
- Unmodeled effects (e.g. additional uncaptured features)
    - Random noise (uncertainty in the data)
- Assume

Independently Identically Distributed

$$\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$$

$$P\left(\epsilon^{(i)}\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(\epsilon^{(i)}\right)^2}{2\sigma^2}\right)$$

Probabilistic Interpretation of Linear Regression

$$y^{(i)} = \Theta^T x^{(i)} + \epsilon^{(i)}$$

$$P\left(y^{(i)} \middle| x^{(i)}; \Theta\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \Theta^T x^{(i)}\right)^2}{2\sigma^2}\right)$$

$$y^{(i)} | x^{(i)}; \Theta \sim \mathcal{N}\left(\Theta^T x^{(i)}, \sigma^2\right)$$

Probabilistic Interpretation of Linear Regression (Likelihood)

- $\epsilon^{(i)}$s are Independently Identically Distributed (IID)

$$L(\Theta) = P(Y|\boldsymbol{X};\Theta)$$

likelihood

$$L(\Theta) = \prod_{i=1}^{m} P\left(y^{(i)}\middle|x^{(i)};\Theta\right)$$

$$L(\Theta) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \Theta^T x^{(i)}\right)^2}{2\sigma^2}\right)$$

Maximum Likelihood Estimator

- Choose Θ to maximize $L(\Theta) = P(Y|\boldsymbol{X};\Theta)$
  - Choose the parameters to make the data as probable as possible

$$\ell(\Theta) = \log L(\Theta)$$

$$\ell(\Theta) = \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \Theta^T x^{(i)}\right)^2}{2\sigma^2}\right)$$

$$\ell(\Theta) = \sum_{i=1}^{m} \log\left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \Theta^T x^{(i)}\right)^2}{2\sigma^2}\right)\right]$$

$$\ell(\Theta) = m \log \frac{1}{\sqrt{2\pi}\sigma} + \boxed{\sum_{i=1}^{m} -\frac{\left(y^{(i)} - \Theta^T x^{(i)}\right)^2}{2\sigma^2}}$$

Maximum Likelihood Estimator

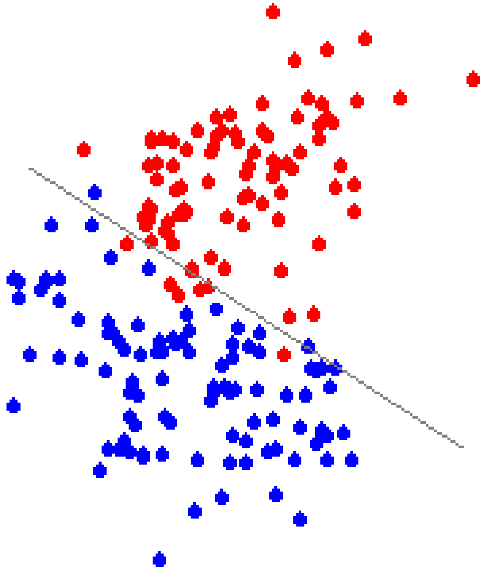
- Maximizing $\ell(\Theta)$ is the same as minimizing

$$\mathcal{J}(\Theta) = \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - \Theta^T x^{(i)}\right)^2$$

- Note that the value of σ doesn't matter in finding Θ

- The solution of the **Least Square** method that we used before is **exactly the same** as the **Maximum Likelihood** estimation of the parameters in the probabilistic setting assuming Gaussian error.

Linear Classification



Logistic Regression
Binary Classification

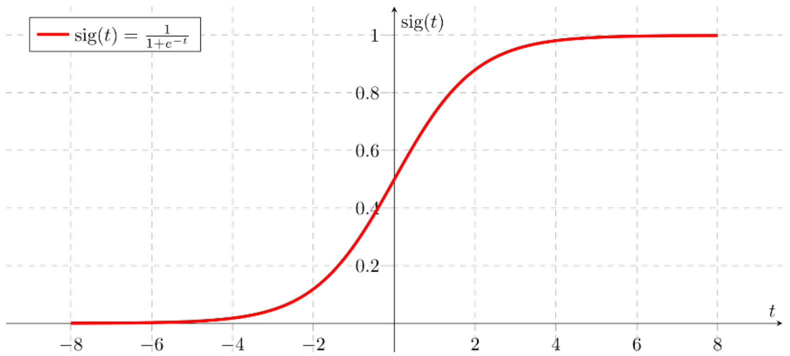
$$y \in \{0,1\}$$

$$h_\theta(x) \in [0, 1]$$

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

$$g(z) = \frac{1}{1 + e^{-z}}$$

Sigmoid function
Logistic function



Logistic Regression
Probabilistic Perspective

Lets design parameters
For the model and fit
them with Max
Likelihood

$$P(y = 1|x; \Theta) = h_\theta(x)$$

$$P(y = 0|x; \Theta) = 1 - h_\theta(x)$$

- Write it in a more compact way

$$P(y|x; \Theta) = (h_\theta(x))^y \big(1 - h_\theta(x)\big)^{1-y}$$

$$L(\Theta) = P(Y|\boldsymbol{X}; \Theta) = \prod_{i=1}^{m} P\left(y^{(i)} \middle| x^{(i)}; \Theta\right)$$

$$L(\Theta) = \prod_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right)\right)^{y^{(i)}} \left(1 - h_\theta\left(x^{(i)}\right)\right)^{1-y^{(i)}}$$

Logistic Regression
Maximum Likelihood Estimation

$$\ell(\Theta) = \log L(\Theta) = \log \prod_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right)\right)^{y^{(i)}} \left(1 - h_\theta\left(x^{(i)}\right)\right)^{1-y^{(i)}}$$

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

- Use Gradient Ascent to maximize the log likelihood

$$\Theta = \Theta + \alpha \nabla_\Theta \ell(\Theta)$$

Logistic Regression
ML – Gradient Ascent

$$\Theta = \Theta + \alpha \nabla_{\Theta} \ell(\Theta)$$

- We will skip the derivation but you will end up with the following

$$\frac{\partial}{\partial \theta_j} \ell(\Theta) = \sum_{i=1}^{m} \left(y^{(i)} - h_\theta\left(x^{(i)}\right)\right) x_j^{(i)}$$

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

$$\theta_j = \theta_j + \alpha \sum_{i=1}^{m} \left(y^{(i)} - \frac{1}{1 + e^{-\Theta^T x}} \right) x_j^{(i)}$$