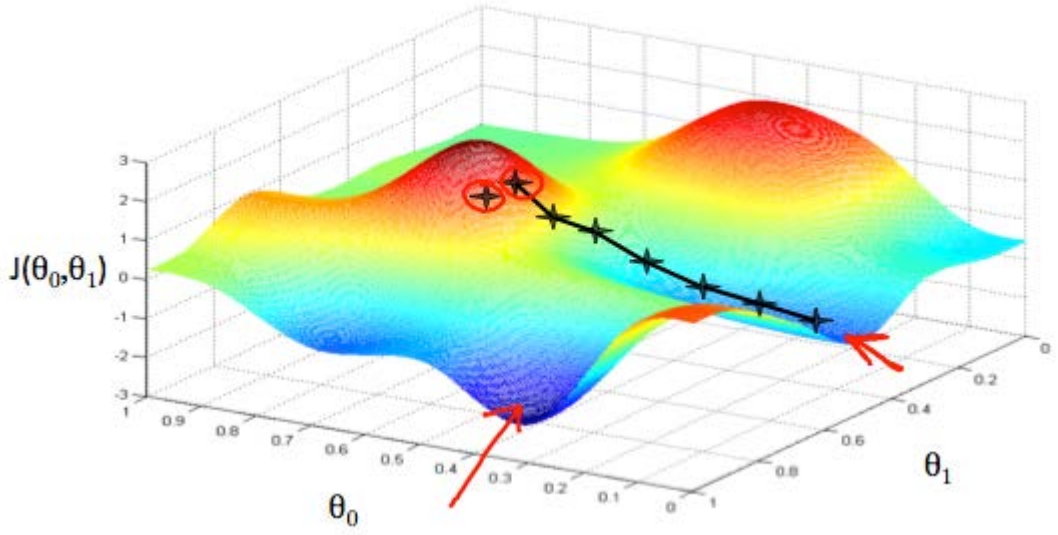
Gradient Descent



The engine behind almost
all ML algorithms

Gradient Descent

Have function: $J(\theta_1, \theta_2)$

Wants $\min_{\theta_1, \theta_2} J(\theta_1, \theta_2)$

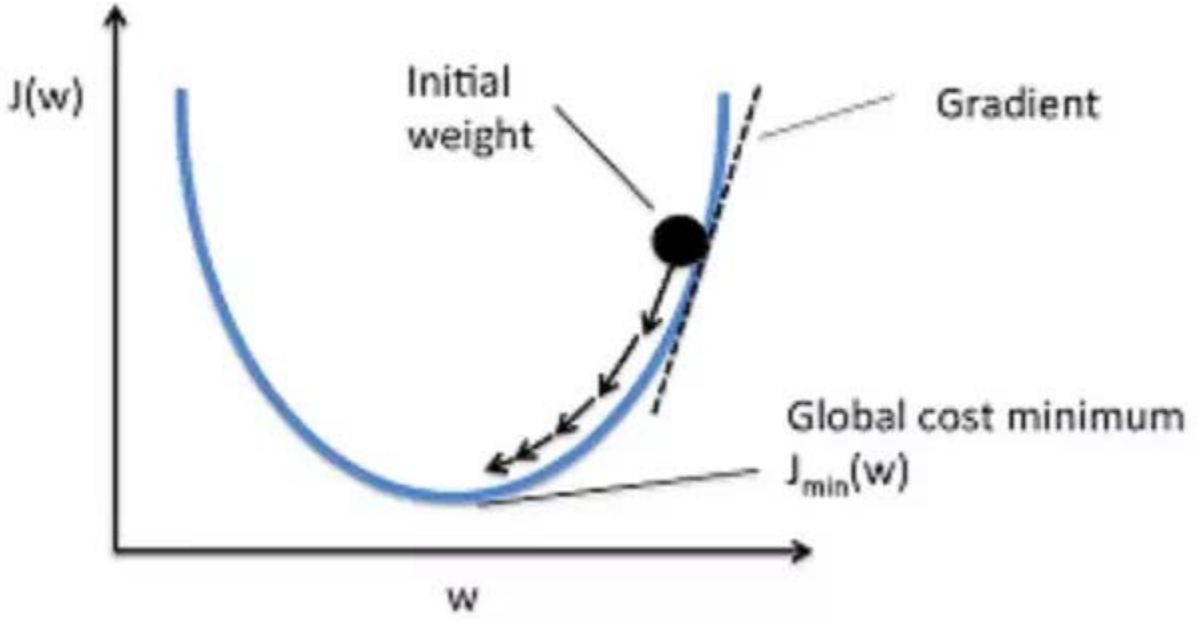
Algorithm:

- Start from some θ_1, θ_2
- Change θ_1, θ_2 $to\ reduce\ J(\theta_1, \theta_2)$ Until hopefully get to the minimum

Gradient Descent

$$\theta_1 = \theta_1 - \alpha * \frac{\partial J(\theta_1, \theta_2)}{\partial \theta_1}$$

$$\theta_2 = \theta_2 - \alpha * \frac{\partial J(\theta_1, \theta_2)}{\partial \theta_2}$$



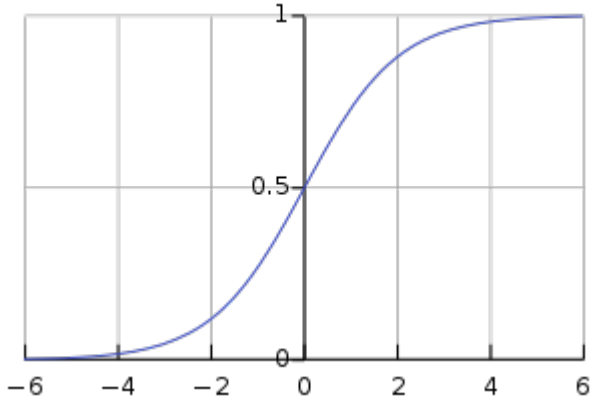
Logistic Regression

Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}) \dots (x^{(m)}, y^{(m)})\}$

m examples $x^{(i)} \in \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}^{(i)}$ $y \in \{0,1\}$

Prediction

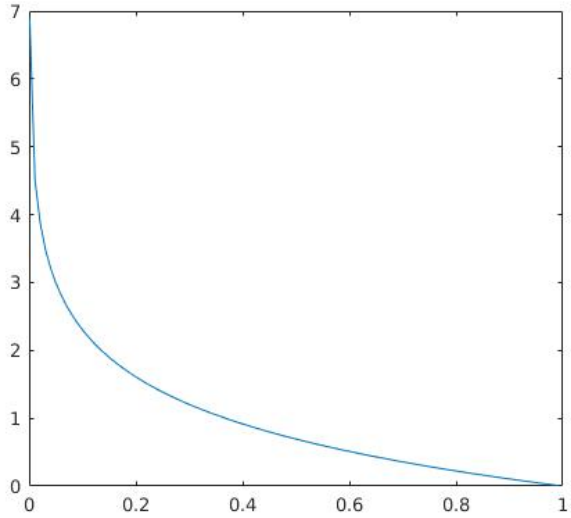
$$h_\theta(x_i) = \frac{1}{1 - e^{-\theta^T x_i}}$$



Logistic Regression cost function

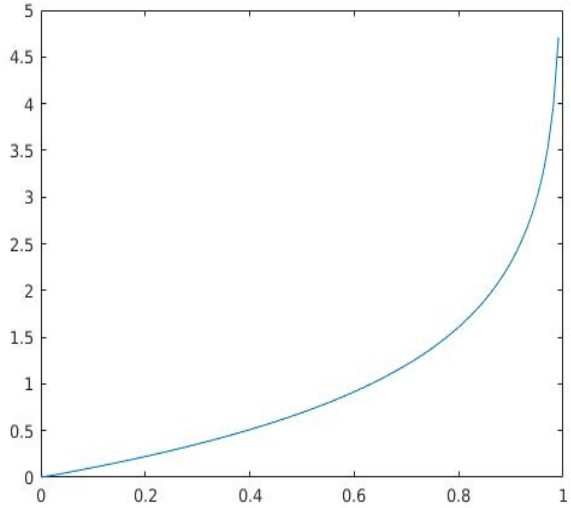
$$Cost(h_\theta(X), y) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} * \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) * \log(1 - h_\theta(x^{(i)}))$$

If y =1



$h_\theta(x)$

If y =0



$h_\theta(x)$

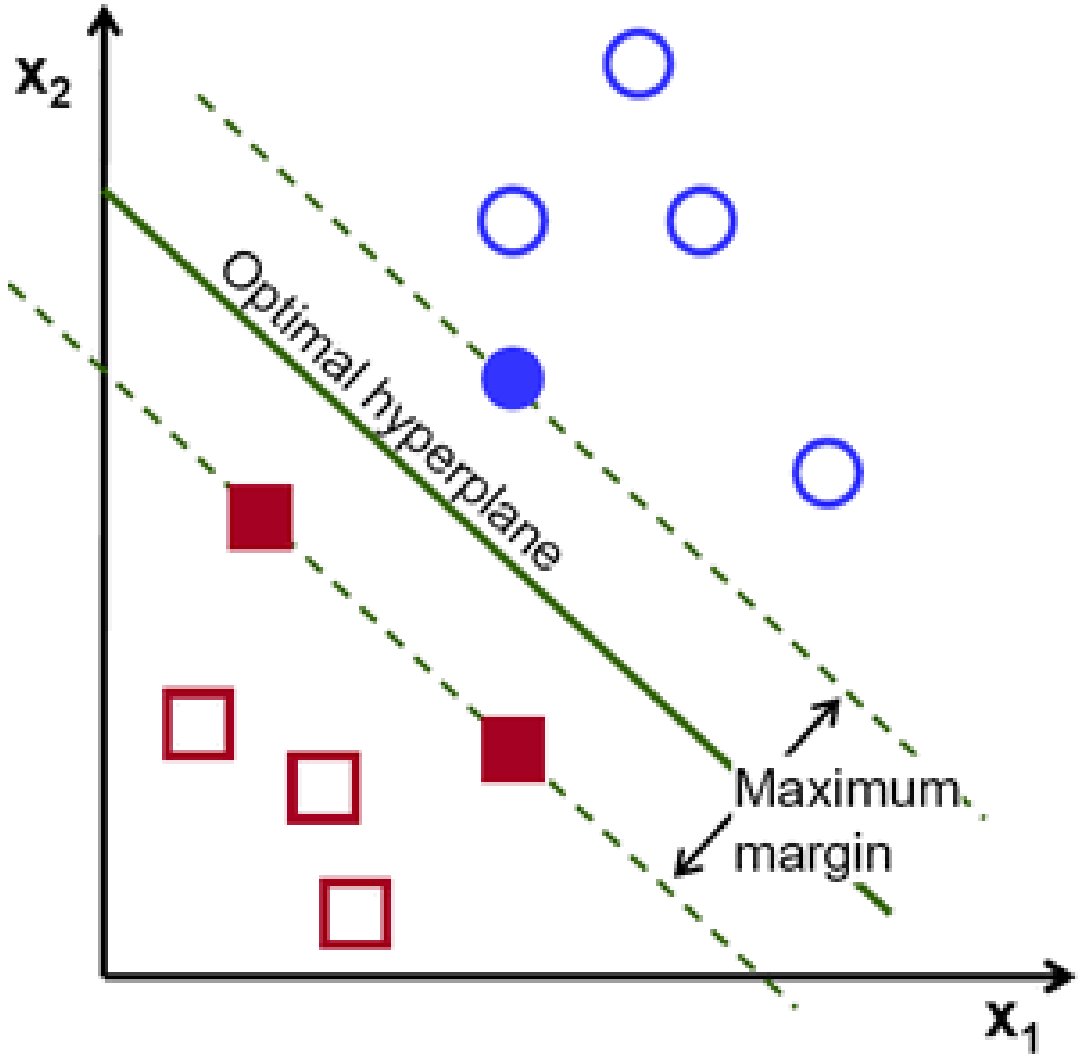
Logistic Regression
Gradient Descent

$$Cost(h_\theta(X), y) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} * \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) * \log(1 - h_\theta(x^{(i)}))$$

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

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

Support vector machine - SVM



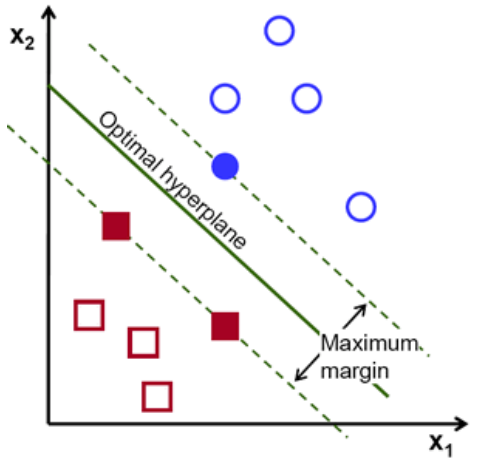
Support vector machine - SVM

Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}) \dots (x^{(m)}, y^{(m)})\}$

m examples $x^{(i)} \in \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}^{(i)}$ $y \in \{0,1\}$

Prediction

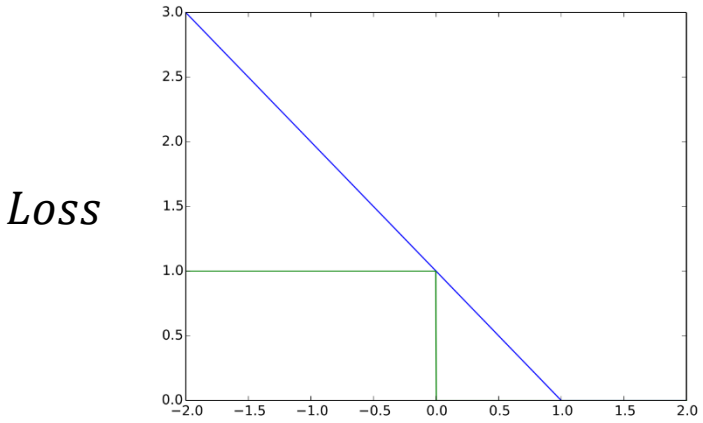
$$h_\theta(x, \theta, b) = \theta x + b$$



Support vector machine - SVM

$$Cost(h_\theta(X), Y) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j \neq y^{(i)}} \max(0\ , \theta_j^T x^{(i)}\ - \theta_{y^{(i)}}^T x^{(i)} + \Delta)$$

Hinge loss



SVM
Gradient Descent

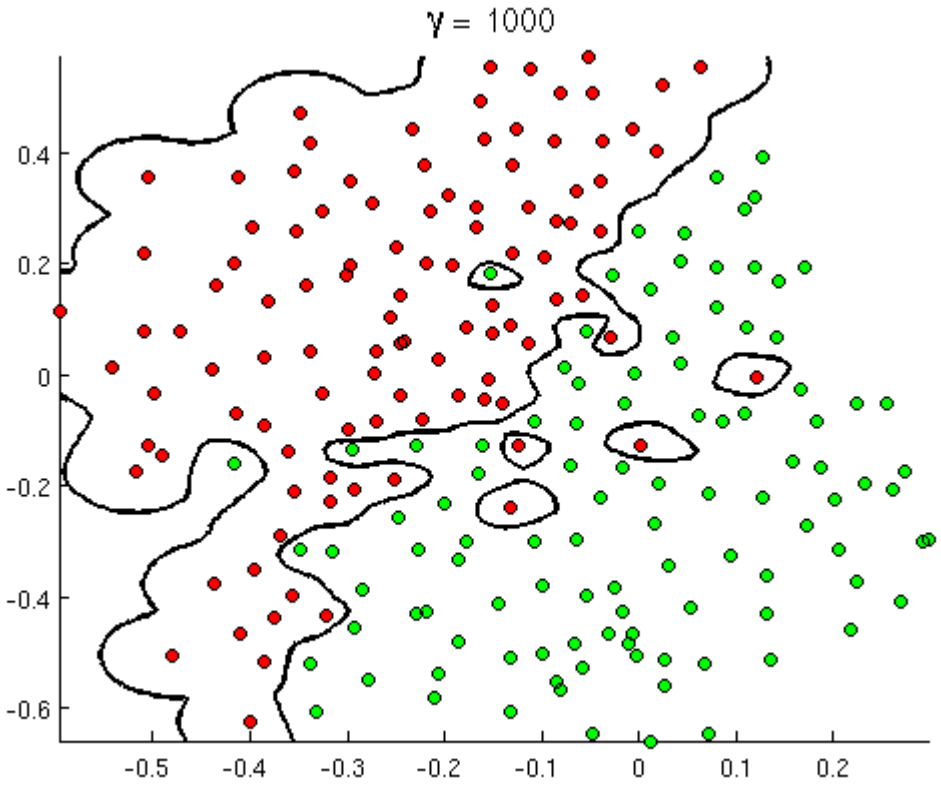
$$Cost(h_\theta(X), y) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j \neq y^{(i)}} \max(0\ , \theta_j^T x^{(i)} - \theta_{y^{(i)}}^T x^{(i)} + \Delta)$$

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

$$\theta_{y^{(i)}} = \theta_{y^{(i)}} - \alpha * -\frac{1}{m}\sum_{i=1}^{m}\sum_{j \neq y^{(i)}} 1(\left(\theta_j^T x^{(i)} - \theta_{y^{(i)}}^T x^{(i)} + \Delta\right) > 0) x^{(i)}$$

$$\theta_j = \theta_j - \alpha * \frac{1}{m}\sum_{i=1}^{m} 1(\left(\theta_j^T x^{(i)} - \theta_{y^{(i)}}^T x^{(i)} + \Delta\right) > 0) x^{(i)}$$

Gaussian - SVM



Regularization loss

$$Cost(h_\theta(X), y) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j \neq y_i} \max(0\ , \theta_j^T x_i\ - \theta_{y_i}^T x_i + \Delta) + \lambda R(\theta)$$

Regularization prevents the model of going wild

Exercise 1.

Seam Carving


Questions