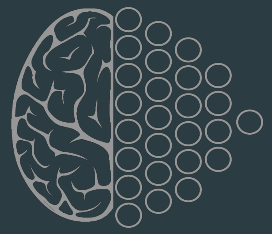
Lecture 5

INTRODUCTION AND APPLICATIONS OF MACHINE LEARNING

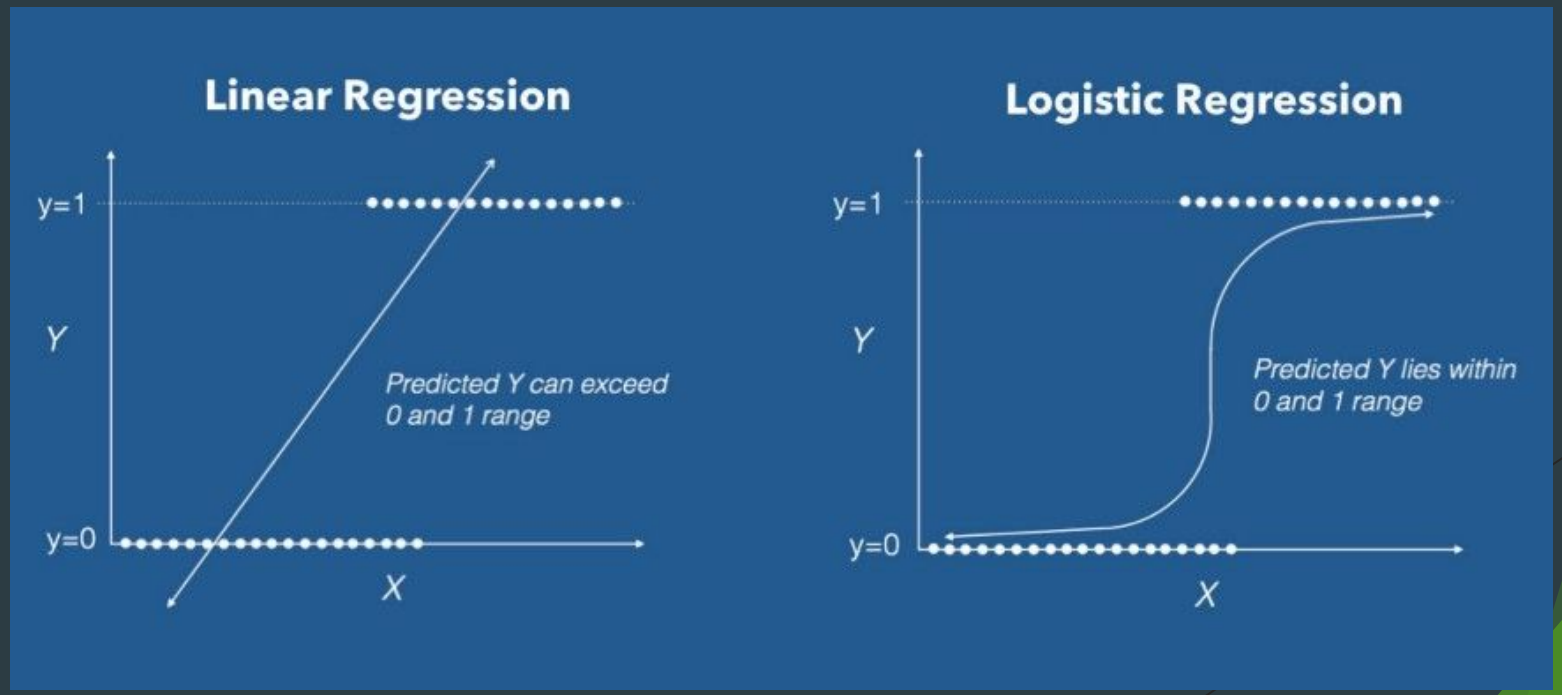


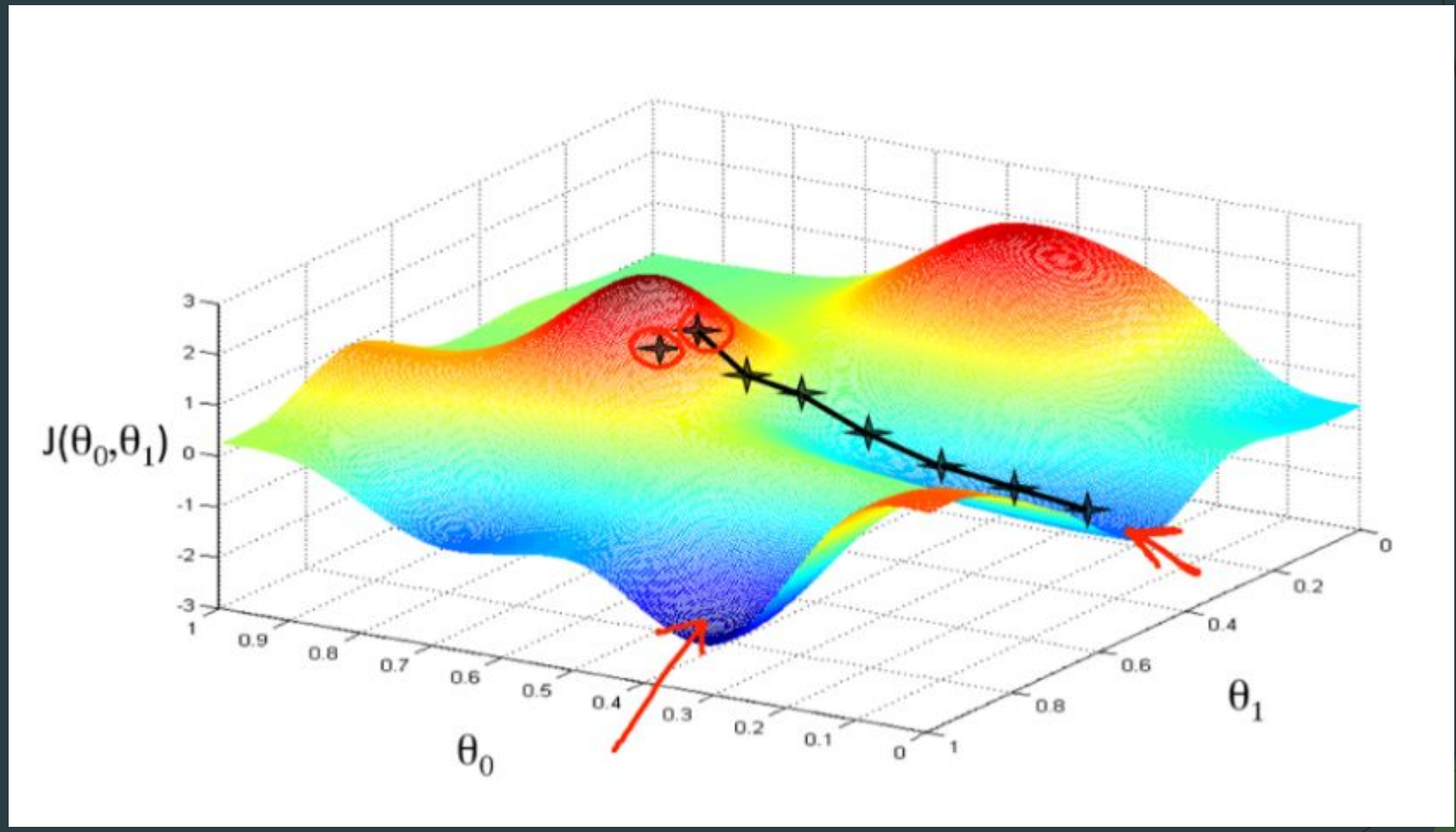
Roadmap

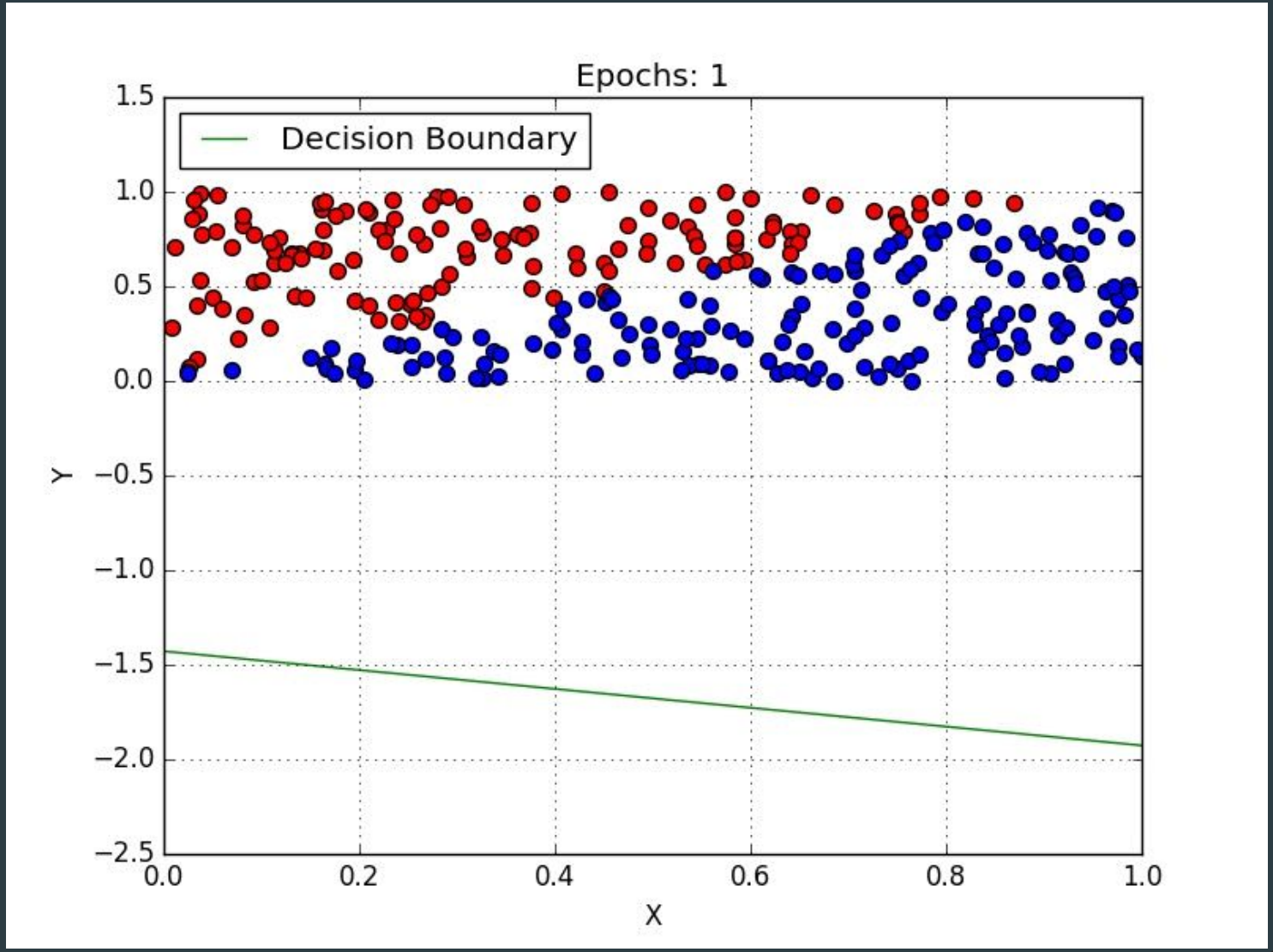
- Logistic regression
  - Hypothesis function
  - Loss function
  - Gradient descent update

Logistic regression

- Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Some of the examples of classification problems are Email spam or not spam, Online transactions Fraud or not Fraud, Tumor Malignant or Benign. Logistic regression transforms its output using the logistic sigmoid function to return a probability value.

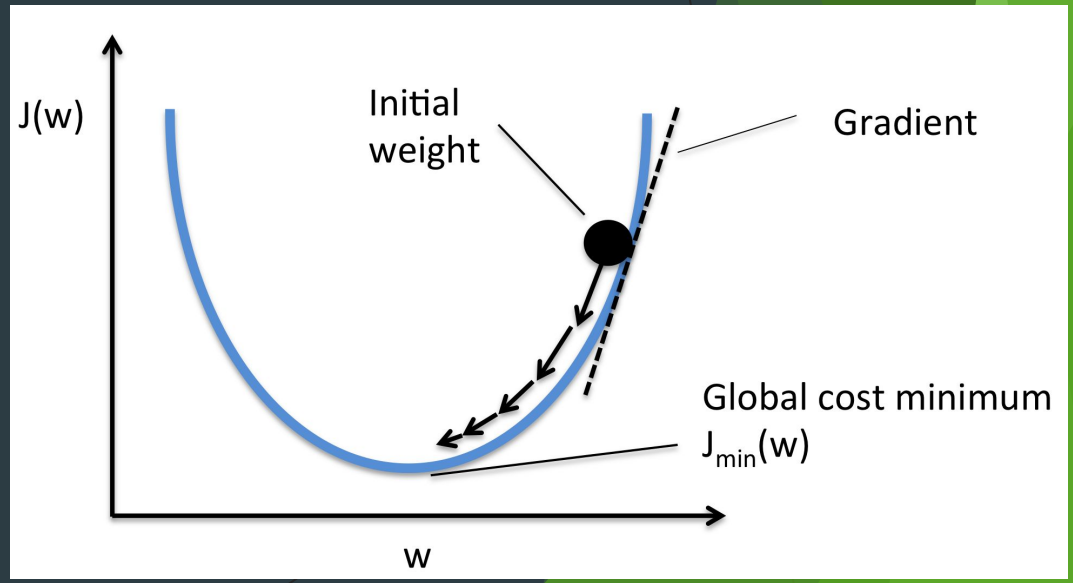



$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1

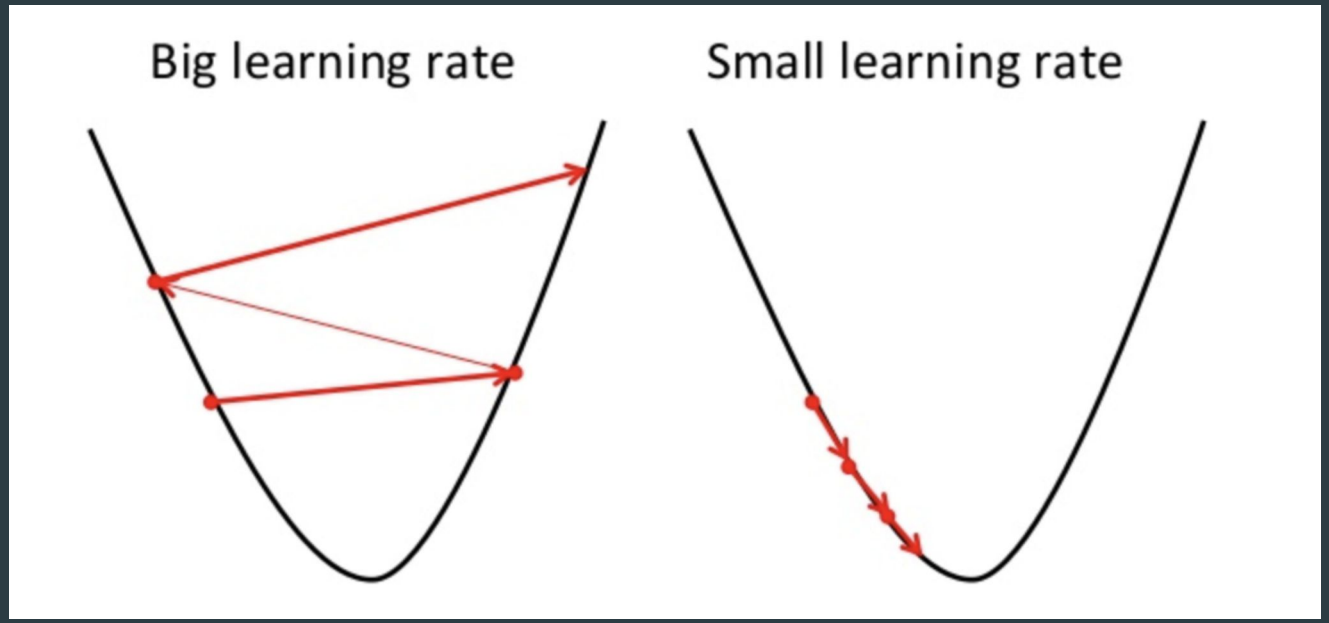

Epochs: 1
Decision Boundary
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
Y
0.0
0.2
0.4
0.6
0.8
1.0
X

Gradient Descent

- Optimization method to find minima in a function (here reduce the cost function MSE)
- It is an iterative process
- Start at any point and move towards the minima
- This depends on
  - Step size (η)
  - direction(determined by the negative of the gradient)



Gradient update

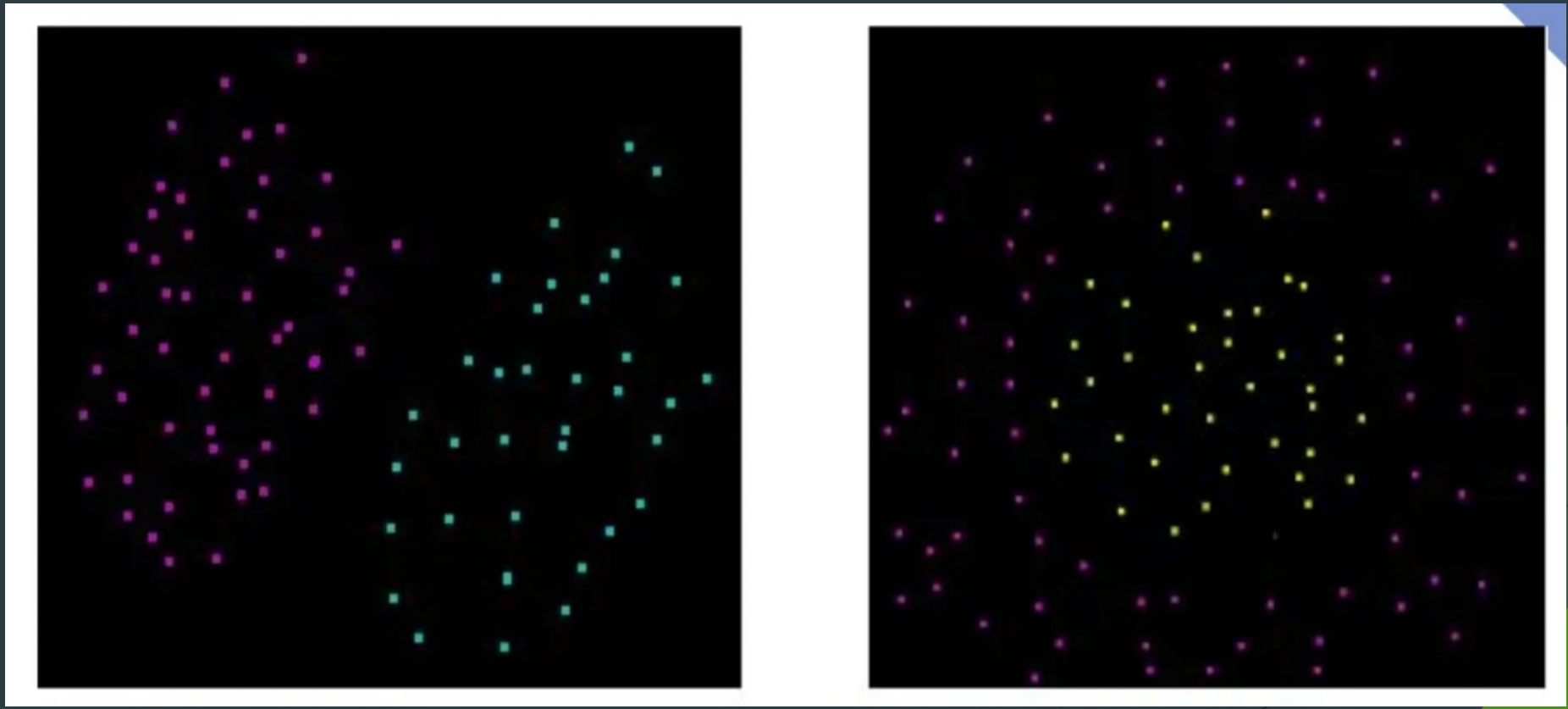


Support Vector Machines

- A support vector machine is a discriminative classifier which intakes training data, the algorithm learns an optimal hyperplane which categorizes new examples

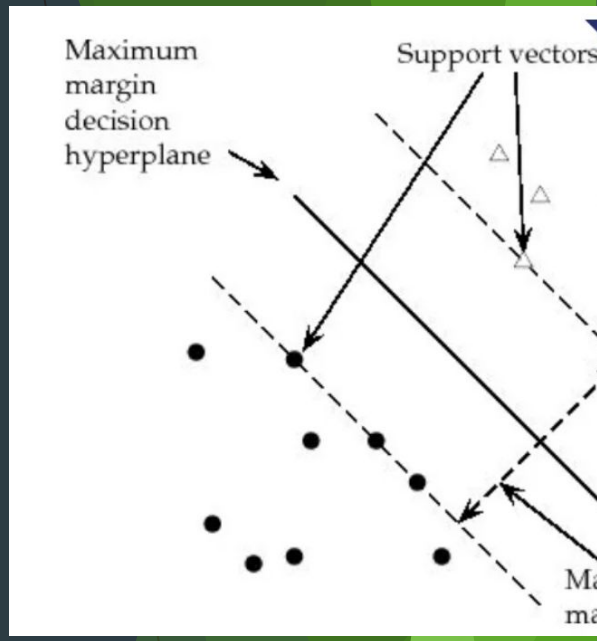
- supervised learning algorithm

Linearly-separable vs non



Margin

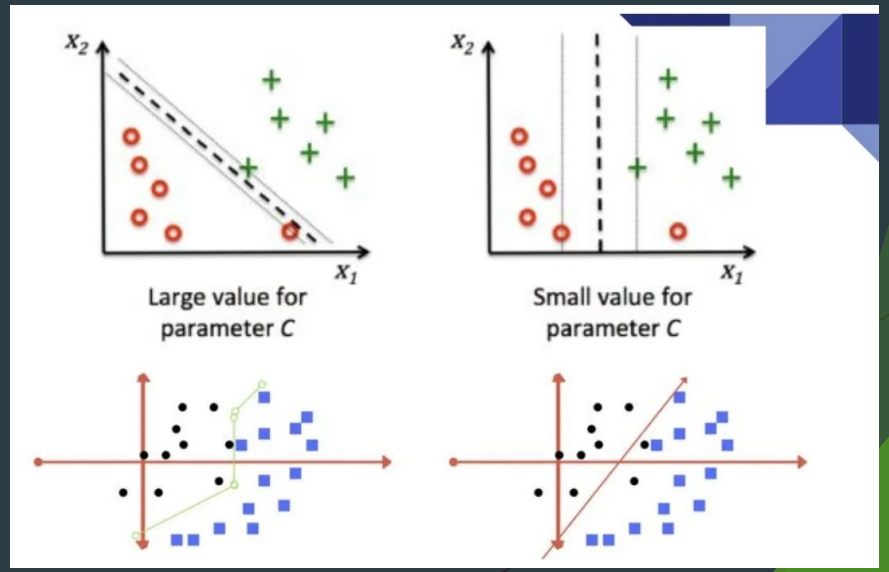
- Margin is the perpendicular distance between the closest data points and the hyperplane
- The best optimized line(hyperplane) with maximum margin is termed as Marginal Maximal hyperplane
- The closest points where the margin distance is calculated is known as the support vectors



Regularization(c)

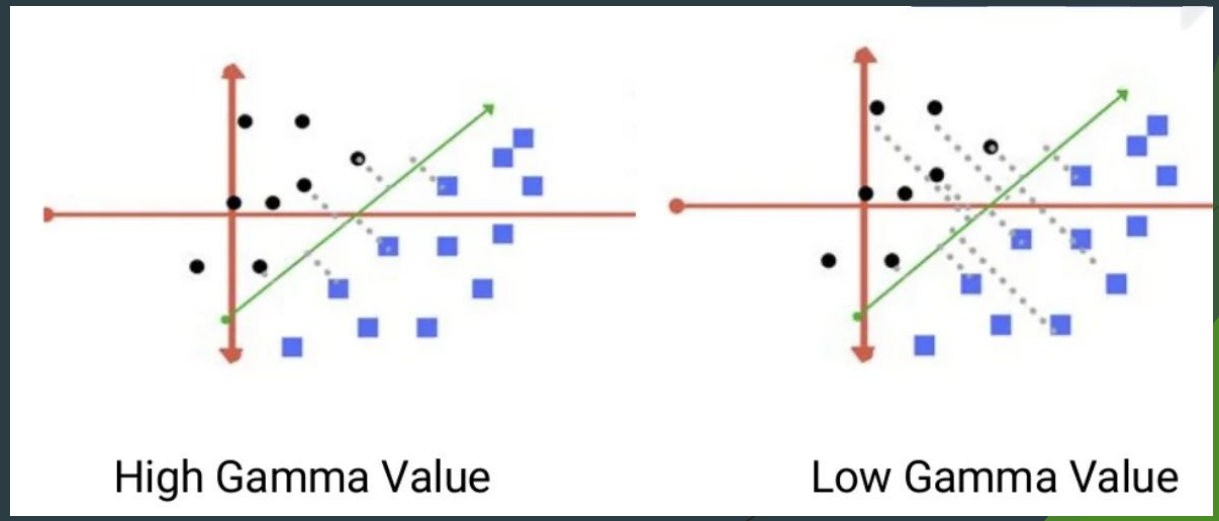
- Optimizes SVM classifier to avoid misclassifying

- C -> large and margin of hyperplane → small
- C -> small and margin of hyperplane → large



Gamma

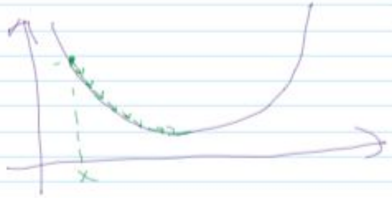
- Defines how far influences the calculation of plausible line of separation
- Low gamma -> points far from plausible line are considered for calculation
- High gamma -> points closer to plausible line are considered for calculation



Kernels

- Mathematical functions for transforming data
- Different SVM algorithms use different type of kernel functions
- Linear kernel
- Radial basis function
- Sigmoid
- Polynomial

Gradient descent algorithm



$y = f(x)$

$f(x) = (x-5)^2$

$$x = x - \eta \frac{\partial f(x)}{\partial x}$$

η: constant (learning rate)

$\frac{\partial f(x)}{\partial x}$: gradient/slope

1. θ init
2. $J_\theta = \frac{1}{m} \sum_{i=1}^{m} [\hat{y}^{(i)} - y^{(i)}_{actual}]^2$
3. update θ using GD

$J(\theta) = \frac{1}{m} \sum [\theta_0 + \theta_1 x]$

$$\theta = \theta - \eta \nabla J(\theta) \rightarrow \theta_0, \theta_1$$

$\Rightarrow$ $\theta_0 = \theta_0 - \eta \frac{\partial J(\theta)}{\partial \theta_0}$ $\quad \theta_1 = \theta_1 - \eta \frac{\partial J(\theta)}{\partial \theta_1}$

$$J(\theta) = \frac{1}{m} \sum_{i=0}^{m} [(\theta_0 + \theta_1 x^{(i)}) - y^{(i)}_{actual}]^2$$

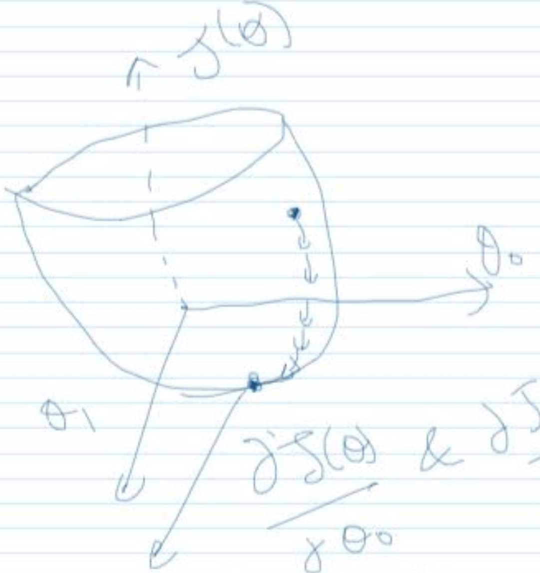
(1) $$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{2}{m} \sum_{i=0}^{m} [\theta_0 + \theta_1 x^{(i)} - y^{(i)}_{actual}] \cdot 1$$

(2) $$\frac{\partial J(\theta)}{\partial \theta_1} = \frac{2}{m} \sum_{i=0}^{m} [\theta_0 + \theta_1 x^{(i)} - y^{(i)}_{actual}] \cdot x^{(i)}$$

finally,

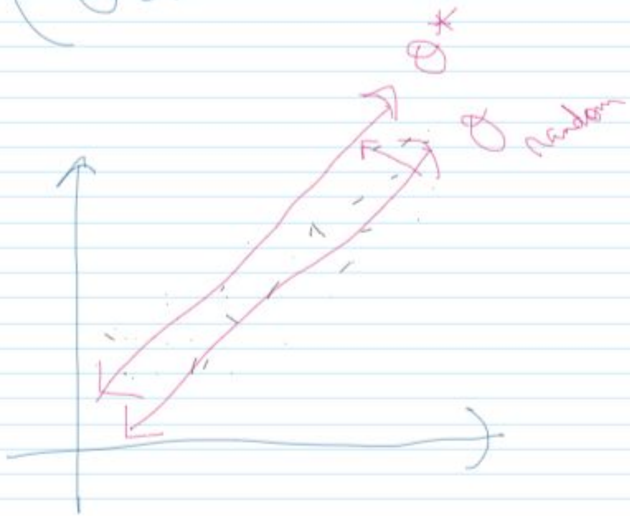
$$\theta_0 = \theta_0 - \eta \frac{1}{m} \sum_{i=1}^{m} [\hat{y}^{(i)} - y^{(i)}_{actual}] \cdot 1$$

$$\theta_1 = \theta_1 - \eta \frac{1}{m} \sum_{i=1}^{m} [\hat{y}^{(i)} - y^{(i)}_{actual}] x^{(i)}$$

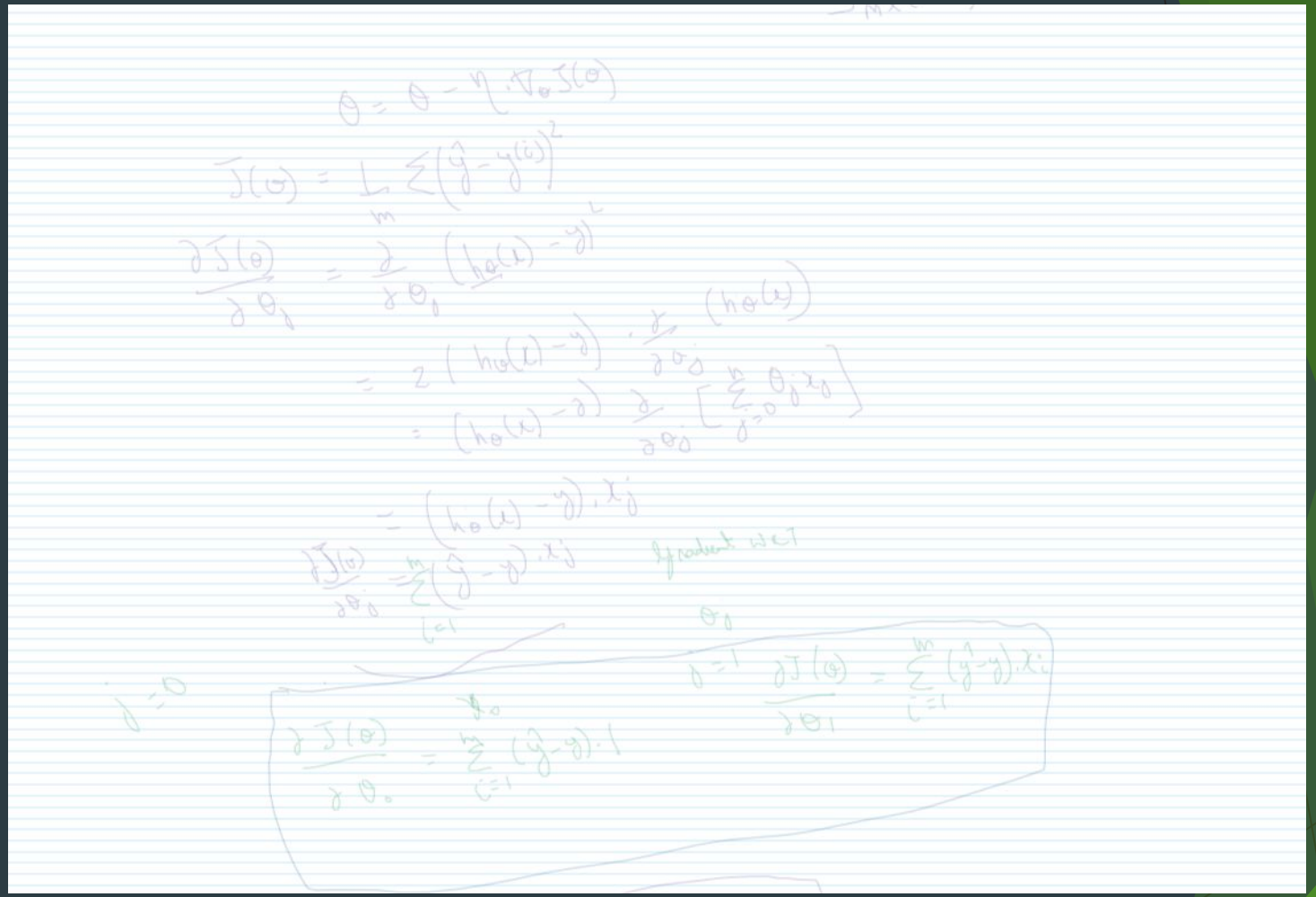


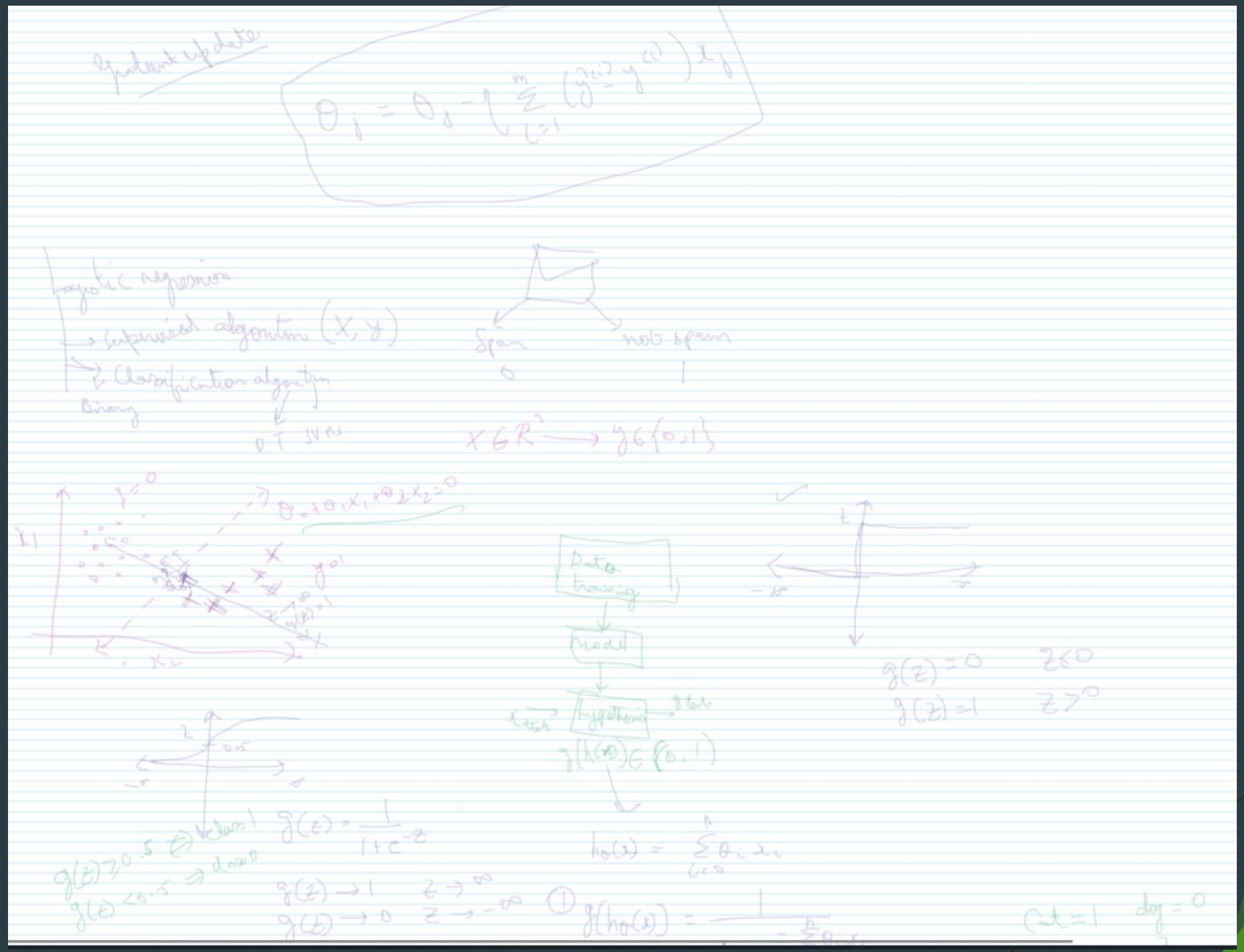
$\frac{\partial J(\theta)}{\partial \theta_0}$ & $\frac{\partial J(\theta)}{\partial \theta_1}$ =

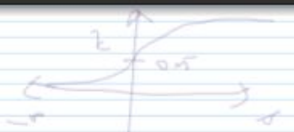
$(\theta_0, \theta_1) \rightarrow \theta^*$




2,00,000
1,00,000
* Bedrooms
3
4
* Area
2
2
* Location
300
400
2
100000
150000
X =
(m examples
x n features)
$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \cdots \theta_n x_n$
$= \theta_0 + \sum_{i=1}^{n} \theta_i x_i$
$x_0 = 1$
$= \theta_0 x_0$
$= \sum_{i=0}^{n} \theta_i x_i$
$\theta =$
$x =$
$X =$
n+1 parameters
$\theta_0 \cdots \theta_n$
m x (n+1)


gradient wrt


Gradient update
$\theta_j = \theta_j - \alpha \sum_{i=1}^{m} (y^{(i)} - y^{(i)}) x_j^{(i)}$
Logistic regression
→ Supervised algorithm (X, y)
→ Classification algorithm
Binary
0 or 1
Spam
0
not spam
1
$X \in R^2 \rightarrow y \in \{0,1\}$
$\theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$
Data training
Model
Hypothesis
$h_\theta(x) \in \{0,1\}$
$h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i$
$g(z) = \frac{1}{1+e^{-z}}$
$g(z) \geq 0.5$ when $z \geq 0$
$g(z) < 0.5$ when $z < 0$
$g(z) \rightarrow 1$ $z \rightarrow \infty$
$g(z) \rightarrow 0$ $z \rightarrow -\infty$
$g(z) = 0$ $z < 0$
$g(z) = 1$ $z \geq 0$
$g(h_\theta(x)) = \frac{1}{1 + e^{-\sum \theta_i x_i}}$
Cat = 1 dog = 0



$g(h(x)) \in (0, 1)$

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

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

$g(z) \geq 0.5 \Rightarrow$ class 1

$g(z) < 0.5 \Rightarrow$ class 0

$g(z) \rightarrow 1 \quad z \rightarrow \infty$

$g(z) \rightarrow 0 \quad z \rightarrow -\infty$

① $g(h_\theta(x)) = \frac{1}{1 + e^{-\sum_{i=0}^{n} \theta_i x_i}}$

Cat = 1 dog = 0

$(1 - 0.8)^2$

$+ (1 - 0.2)^2$

$+ \vdots$

$(0 - 0.6)^2$

} total error

① (0, 1)

② Assign diff value for diff points ↳ classifying

② Loss func

Avg. sq loss $= \frac{1}{m} \sum \left(\hat{y}^{(i)} - g(h_\theta(x^{(i)}))\right)^2$ ✗

③ Minimize the error

$\theta = \theta - \eta \frac{\partial L}{\partial \theta}$

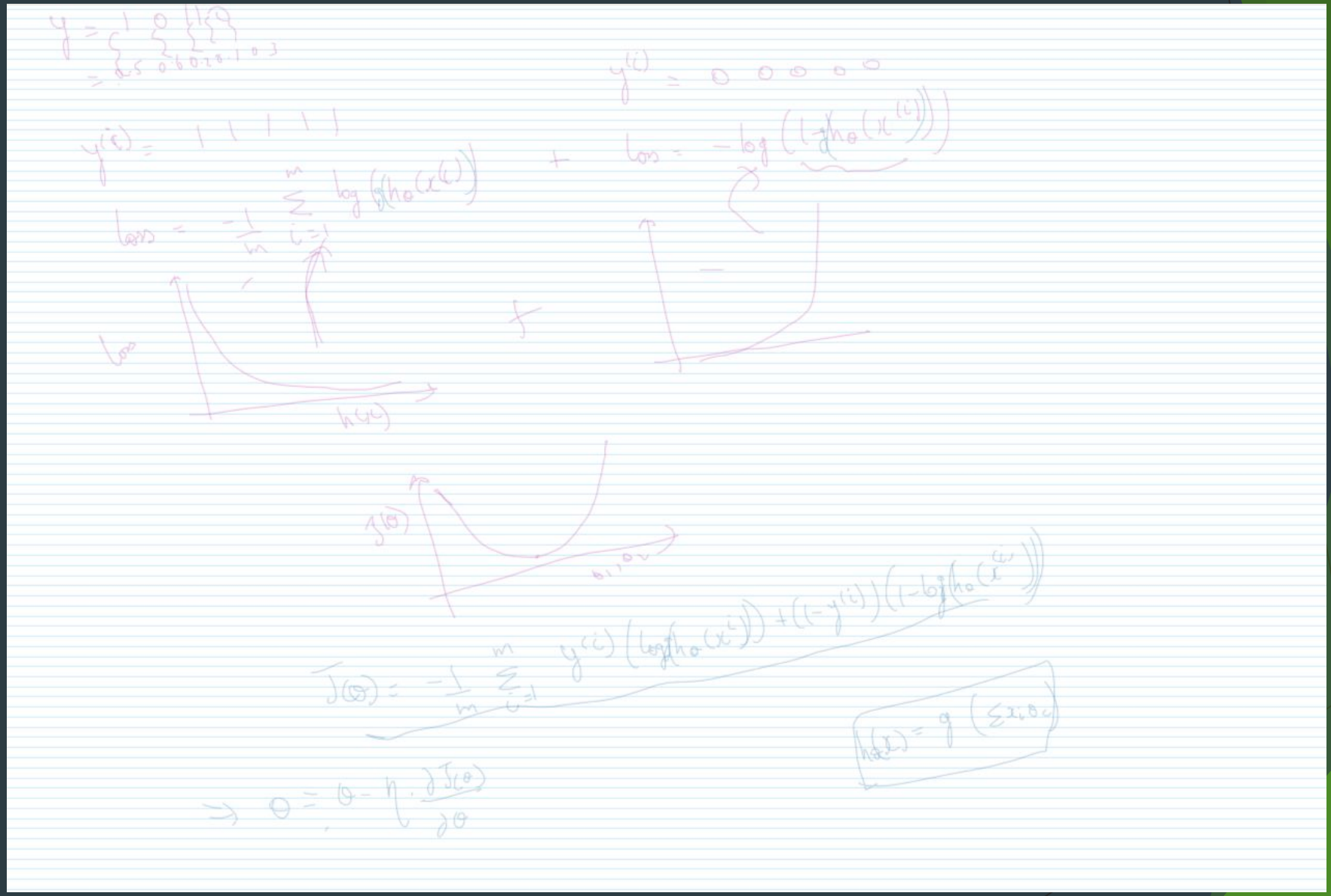
Log loss Binary Cross entropy $\hat{y} = h_\theta(x)$

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

$y = \{1 \; 0 \; 1 \; 1 \; 0\}$

$= \{0.5 \; 0.6 \; 0.2 \; 0.1 \; 0.3\}$

$y^{(i)} = 0 \; 0 \; 0 \; 0 \; 0$



THANK YOU!