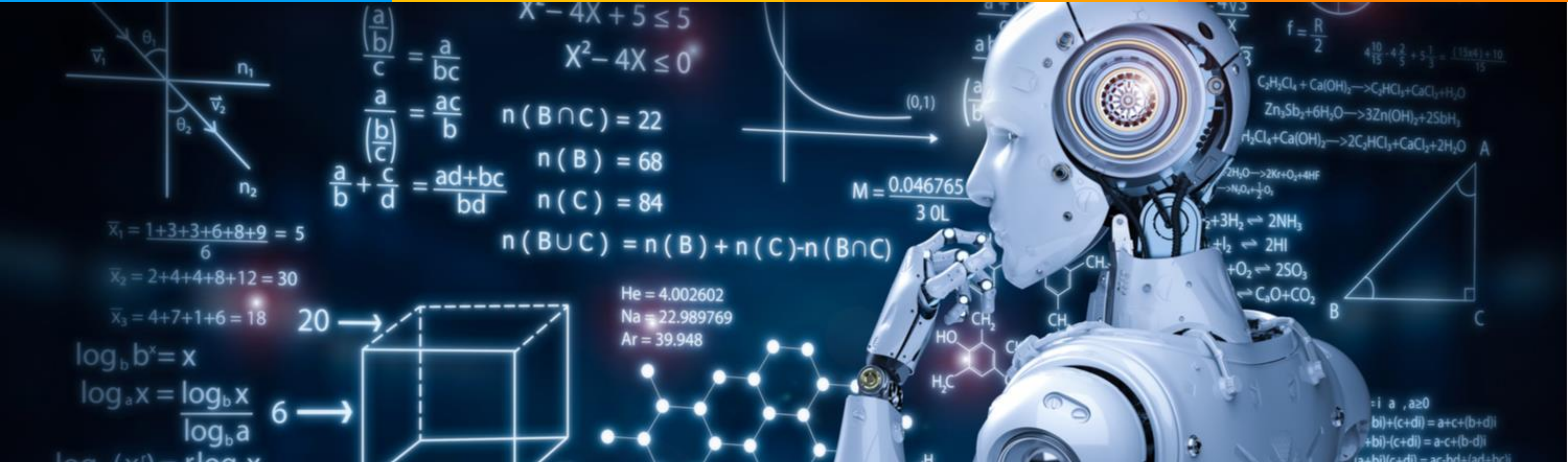
Duration: 60 min | Srikanth Shenoy | AML 5102 | M.E. AI & ML



Lecture 32: Logistic Regression

Recap - SVM

- Margin, hyperplane, support vectors
- Hard Margin
- Soft Margin
- Primal and dual
- Kernel

Sigmoid & Log of odds

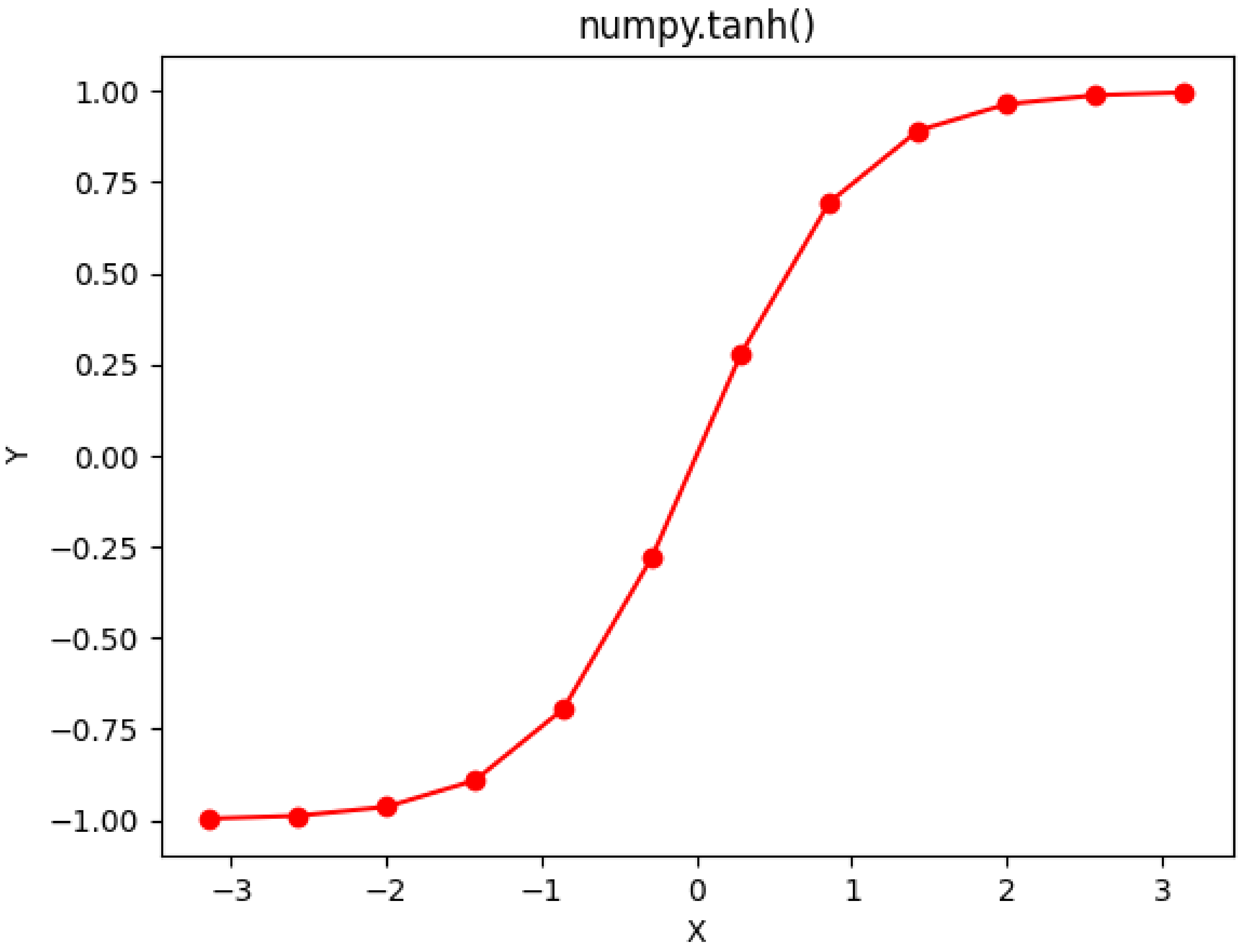
tanh

$$tanh(x) = \frac{sinh(x)}{cosh(x)}$$

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

$$x \in (-\infty, +\infty)$$

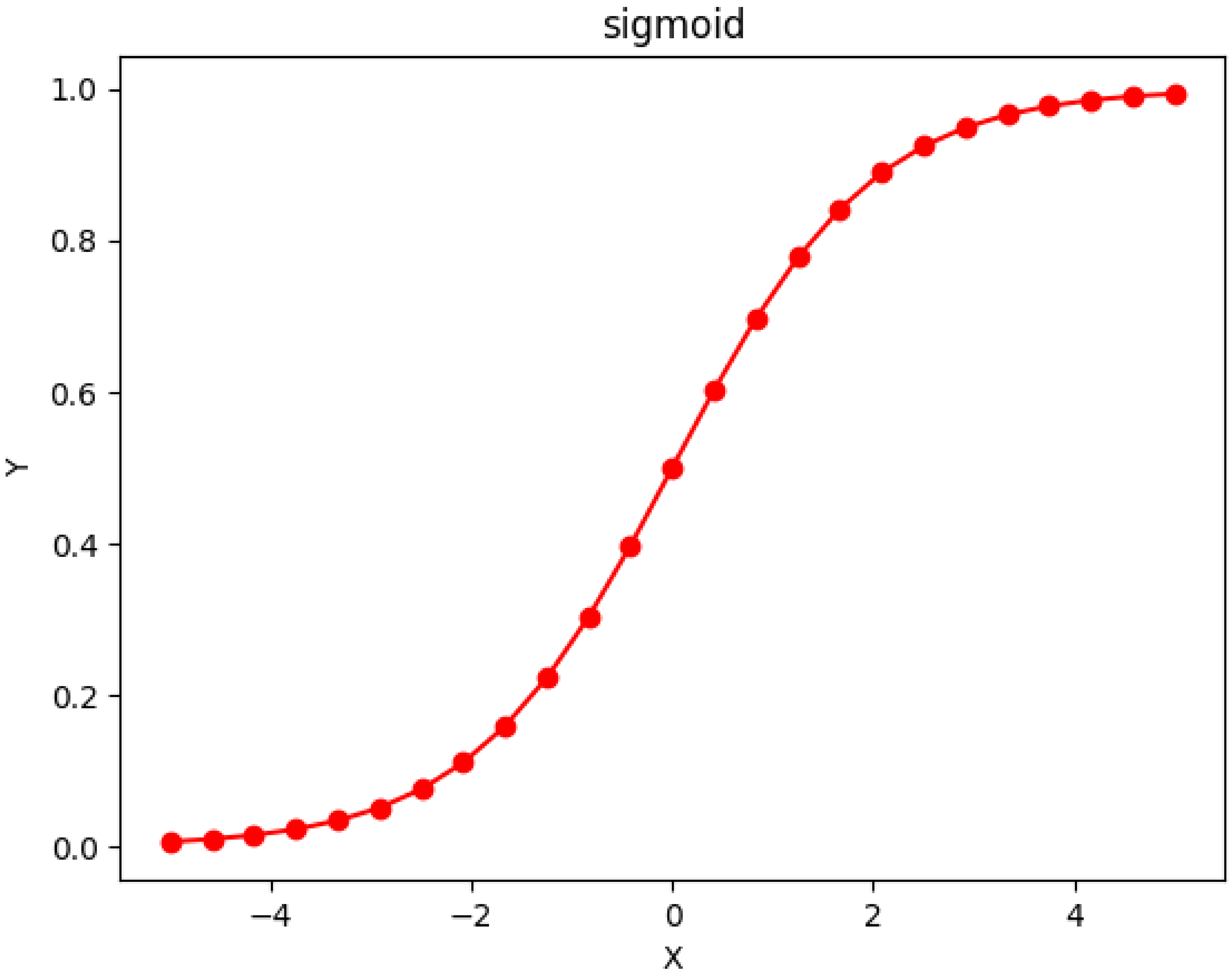
$$y \in [-1, 1]$$



Sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$x \in (-\infty, +\infty)$$
$$y \in [0, 1]$$

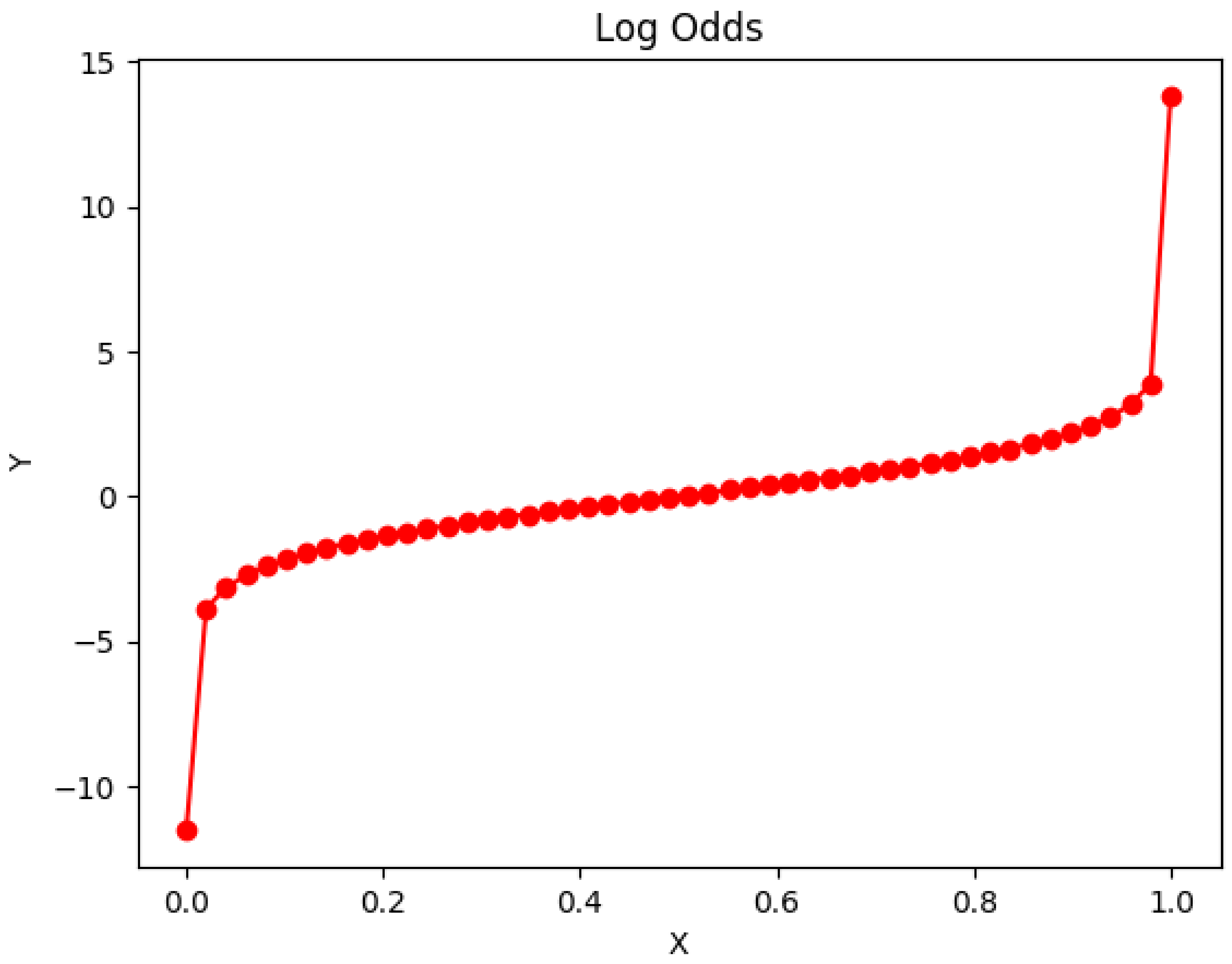


Log of Odds

$$log\left(\frac{p}{1-p}\right) = x$$

$$p = \frac{1}{1+e^{-x}}$$

$$p = \sigma(x)$$



Dot product to Sigmoid

$$log\left(\frac{p}{1-p}\right) = w_1x_1 + w_2x_2 + \ldots w_nx_n + b = w^Tx + b$$

$$p = \frac{1}{1+e^{-(w^Tx+b)}}$$

w are coefficients of the n features in dataset

Can this be considered loss function and its gradient calculated?

Can we take average probability for a given w, b?

Not quite

Formulate loss function with Maximum Likelihood Estimate (MLE)

Maximum Likelihood Estimate (MLE)

Coin Toss

- Single Coin toss
  - Heads or Tails
- Heads = 1, Tails = 0
- Bernoulli Trial
- P(Heads) = P(X=1) = p
- P(Tails) = P(X=0) = 1-p
- Combined Probability Mass Function

$$p(x) = p^x (1-p)^{(1-x)}$$

Step 1: Likelihood function definition

name	sex	age	sibsp	parch	survived
Allen, Miss. Elisabeth Walton	female	29.0000	0.0	0.0	1
Allison, Master. Hudson Trevor	male	0.9167	1.0	2.0	1
Allison, Miss. Helen Loraine	female	2.0000	1.0	2.0	0
Allison, Mr. Hudson Joshua Creighton	male	30.0000	1.0	2.0	0
Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1.0	2.0	0

- Maximize $P(sample1 \in y^{(1)} AND sample2 \in y^{(2)}$

$$AND...samplem \in y^{(m)})$$

Step 1: Likelihood function definition

- Maximize

$$P(sample1 \in y^{(1)} AND sample2 \in y^{(2)} AND ... samplem \in y^{(m)})$$

- For IID data

$$P(sample1 \in y^{(1)}) P(sample2 \in y^{(2)}) ... P(samplem \in y^{(m)})$$

$$P(y^{(1)} | \mathbf{X}^{(1)}) \times P(y^{(2)} | \mathbf{X}^{(2)}) ... \times P(y^{(m)} | \mathbf{X}^{(m)})$$

$$\prod_{i=1}^{m} P(y^{(i)} | \mathbf{X}^{(i)})$$

Step 1: Likelihood function definition (contd)

- Maximize $\prod_{i=1}^{m} P(y^{(i)}|\mathbf{X}^{(i)})$
- Maximize Likelihood function

$$\mathcal{L}(Data|\theta) = P(Data|\theta) = \prod_{i=1}^{m} P(y^{(i)}|\mathbf{X}^{(i)}) = \prod_{i=1}^{m} p_i^{y_i}(1-p_i)^{(1-y_i)}$$

- This is not a good format. Why?
  - Product of probabilities approaches 0. Computationally not robust
  - Gradient calc needs product rule: very cumbersome
  - Hard to do minibatch with gradient calc using product rule

Step 2: Convert to Log Likelihood function

- Maximize Likelihood function

$$\mathcal{L}(Data|\theta) = P(Data|\theta) = \prod_{i=1}^{m} P(y^{(i)}|\mathbf{X}^{(i)}) = \prod_{i=1}^{m} p_i^{y_i}(1-p_i)^{(1-y_i)}$$

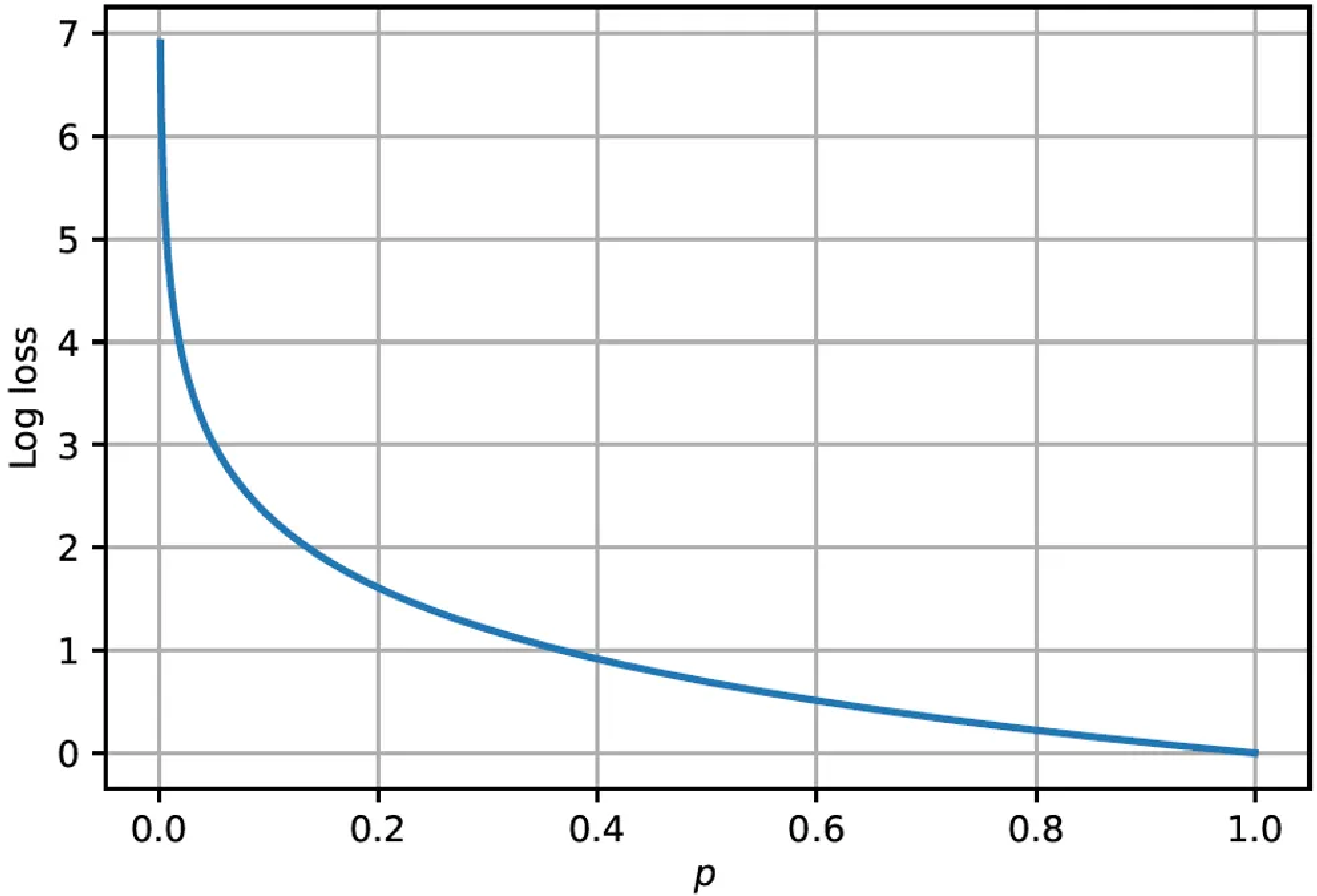
- Taking log on both sides, maximize log likelihood function

$$log(\mathcal{L}(Data|\theta)) = \boldsymbol{\ell}(Data|\theta) = \sum_{i=1}^{m} log(p_i^{y_i}) + \sum_{i=1}^{m} log(1-p_i)^{(1-y_i)})$$

- Maximize

$$\boldsymbol{\ell}(Data|\theta) = \sum_{i=1}^{m} y_i log(p_i) + \sum_{i=1}^{m} (1-y_i) log(1-p_i)$$

$$\boldsymbol{\ell}(Data|\theta) = \sum_{i=1}^{m} y_i log(p_i) + \sum_{i=1}^{m} (1 - y_i) log(1 - p_i)$$



Step 3: Negative Log Likelihood function

- Maximize Log Likelihood function

$$\ell(Data|\theta) = \sum_{i=1}^{m} y_i log(p_i) + \sum_{i=1}^{m} (1 - y_i) log(1 - p_i)$$

- Minimize neg log likelihood function

$$-\sum_{i=1}^{m} y_i log(p_i) - \sum_{i=1}^{m} (1 - y_i) log(1 - p_i)$$

- This is our cost function to minimize
- Also known as binary cross entropy loss function

Step 4: Use sigmoid in neg log likelihood function

- Minimize neg log likelihood function

$$\mathcal{J} = -\sum_{i=1}^{m} y_i log(p_i) - \sum_{i=1}^{m}(1-y_i)log((1-p_i)$$

You have to remember formulas on this slide

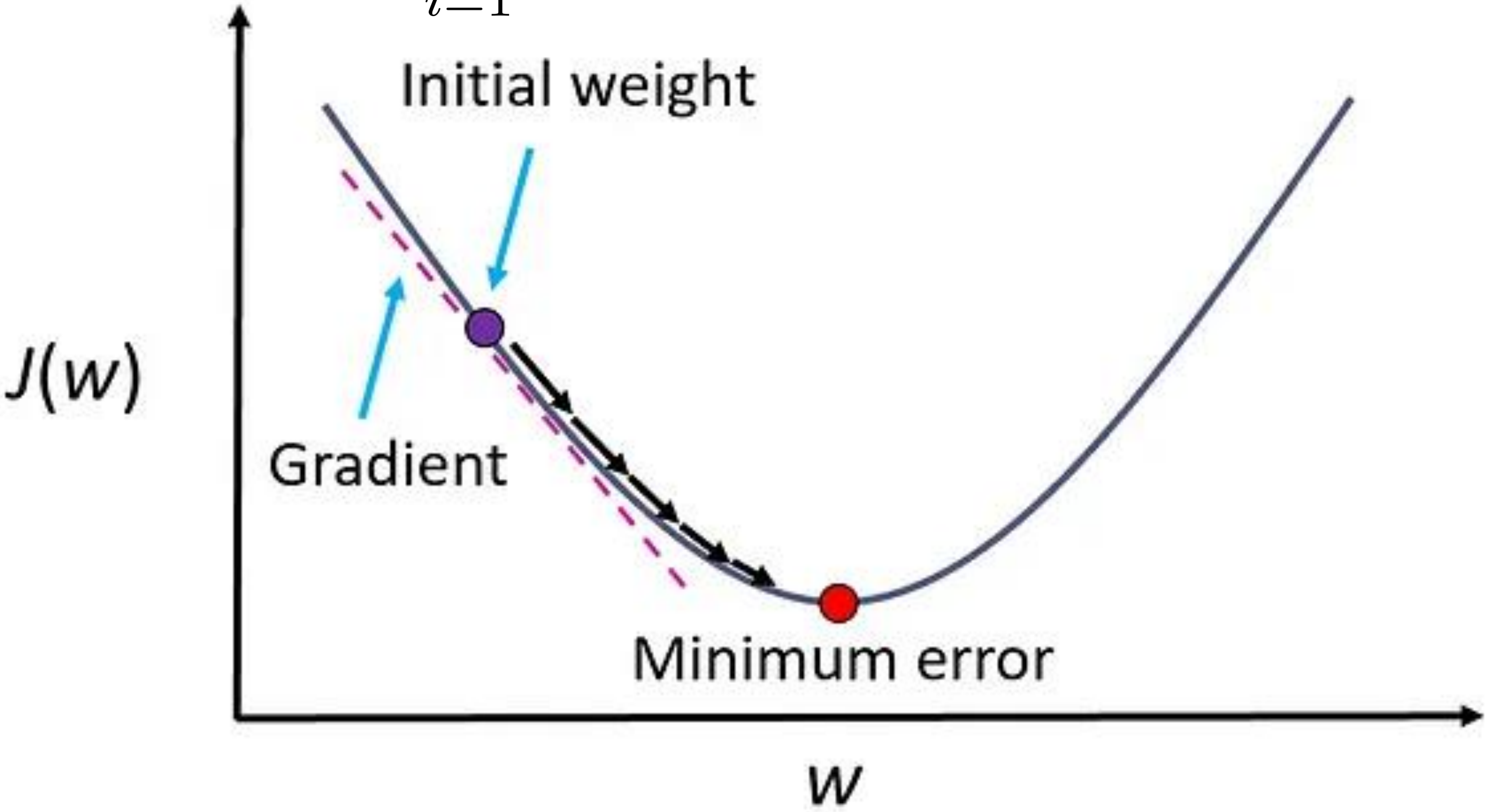
$$p = \frac{1}{1+e^{-(w^Tx+b)}}$$

$$\arg\min_{w,b} \mathcal{J}(x;w,b) = -\sum_{i=1}^{m} y_i log\left(\frac{1}{1+e^{-(w^Tx_i+b)}}\right)$$

$$-\sum_{i=1}^{m}(1-y_i)log\left(1-\frac{1}{1+e^{-(w^Tx_i+b)}}\right)$$

$$\arg\min_{w,b} \mathcal{J}(x; w, b) = -\sum_{i=1}^{m} y_i log\left(\frac{1}{1 + e^{-(w^T x_i + b)}}\right)$$

$$-\sum_{i=1}^{m}(1 - y_i)log\left(1 - \frac{1}{1 + e^{-(w^T x_i + b)}}\right)$$



Step 5: Neg log likelihood function vectorized

- Minimize neg log likelihood function

$$\mathcal{J} = -\sum_{i=1}^{m} y_i log(p_i) - \sum_{i=1}^{m} (1 - y_i) log((1 - p_i)$$

$$p = \frac{1}{1 + e^{-(w^T x + b)}}$$

$$J(\mathbf{w}) = -\frac{1}{n}\left(\mathbf{y}^T \log \mathbf{p} + (\mathbf{1} - \mathbf{y}^T) \log(\mathbf{1} - \mathbf{p})\right)$$

This is different from the vectorized form you obtained in Sudarsan sir's Logistic Regression class

Because in this class we will always treat data matrix rows as records and columns as features

Step 6: Calculate Gradient (plain & vectorized)

- Minimize neg log likelihood function

$$J(\mathbf{w}) = -\frac{1}{n}\left(\mathbf{y}^T \log \mathbf{p} + (\mathbf{1} - \mathbf{y}^T) \log(\mathbf{1} - \mathbf{p})\right)$$

- Non vectorized gradient

$$\frac{\partial}{\partial w_j} J(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} [(p_i - y_i) x_{ij}]$$

- Vectorized gradient

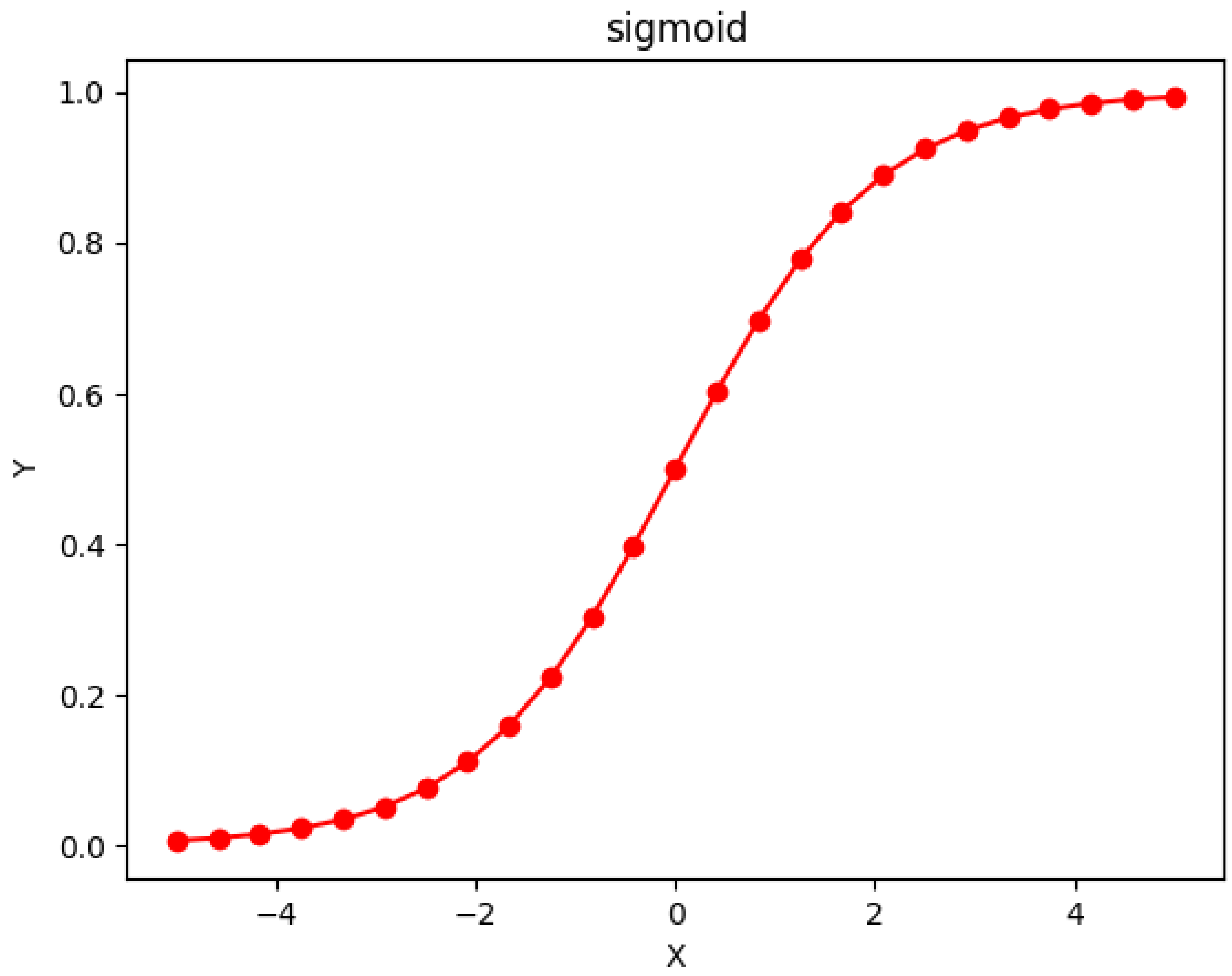
$$\nabla_{\mathbf{w}} J(\mathbf{w}) = \frac{1}{n} X^T (\mathbf{p} - \mathbf{y})$$

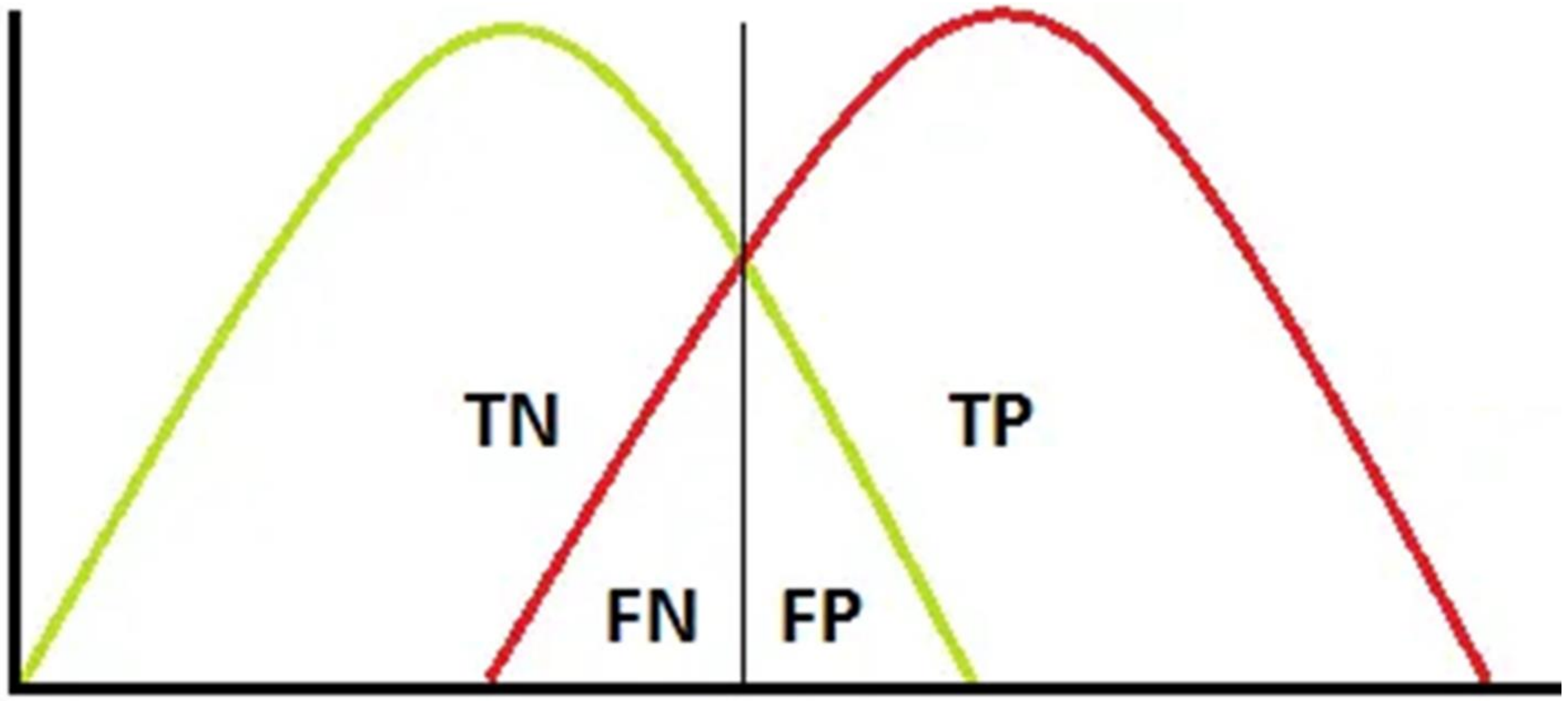
- Gradient descent $\mathbf{w} \leftarrow \mathbf{w} - \alpha \nabla_{\mathbf{w}} J(\mathbf{w})$

Regularization in Logistic Regression

- Same as linear regression

Adjusting threshold






QUESTIONS