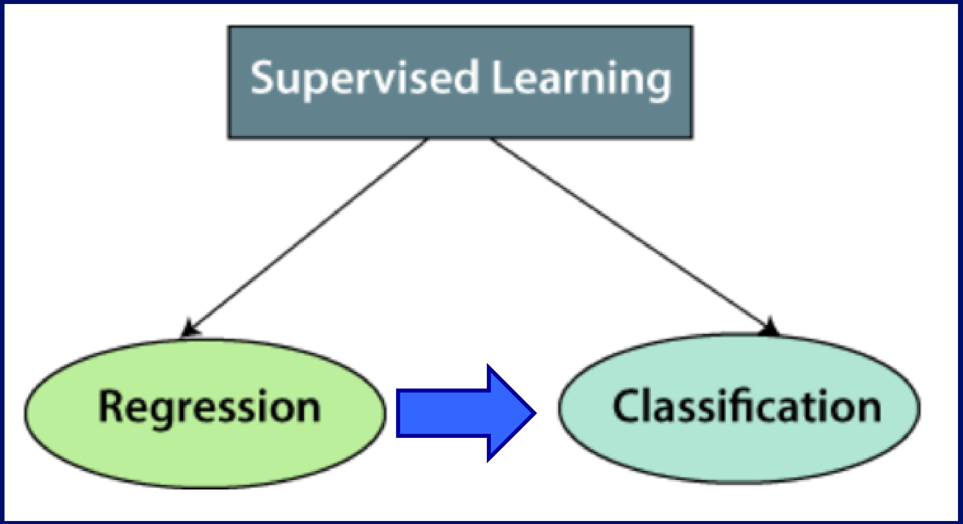
Regression and Classification (II)

Classification is a special case of regression



Output:
Continuous numeric

Output:
Discrete categorical

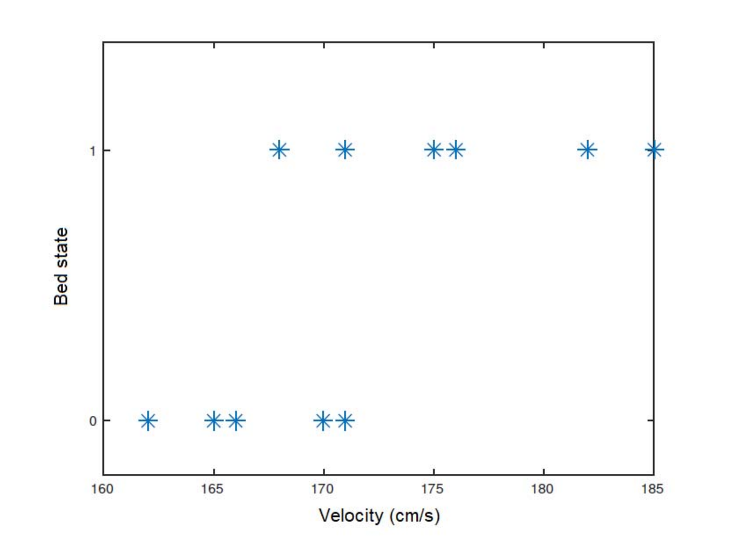
Method:
Linear regression
…

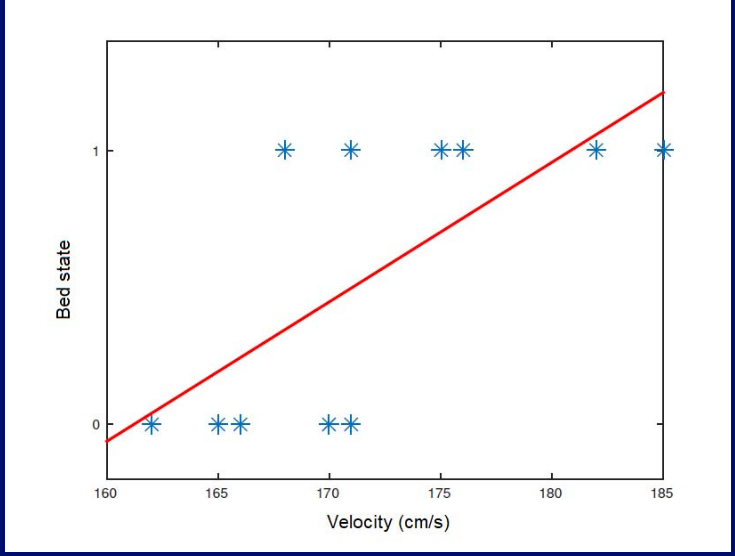
Method:
Logistic regression
…

Motivating example

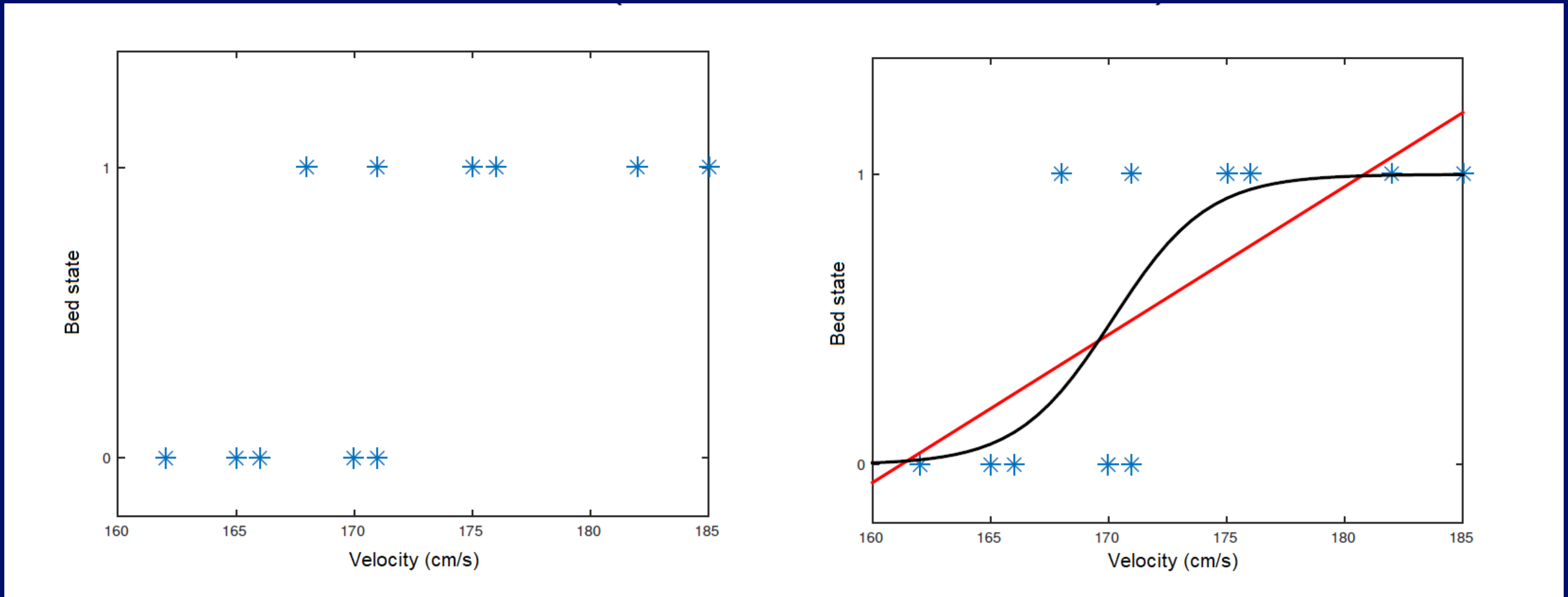
Consider a specific example of the binary classification problem with only one feature: sediment transport in a river reach

- Input: mean velocity of water flow
- Output: state of the river bed (0 = no bed load, 1 = bed in motion)





Simple linear regression is obviously not appropriate in this case.

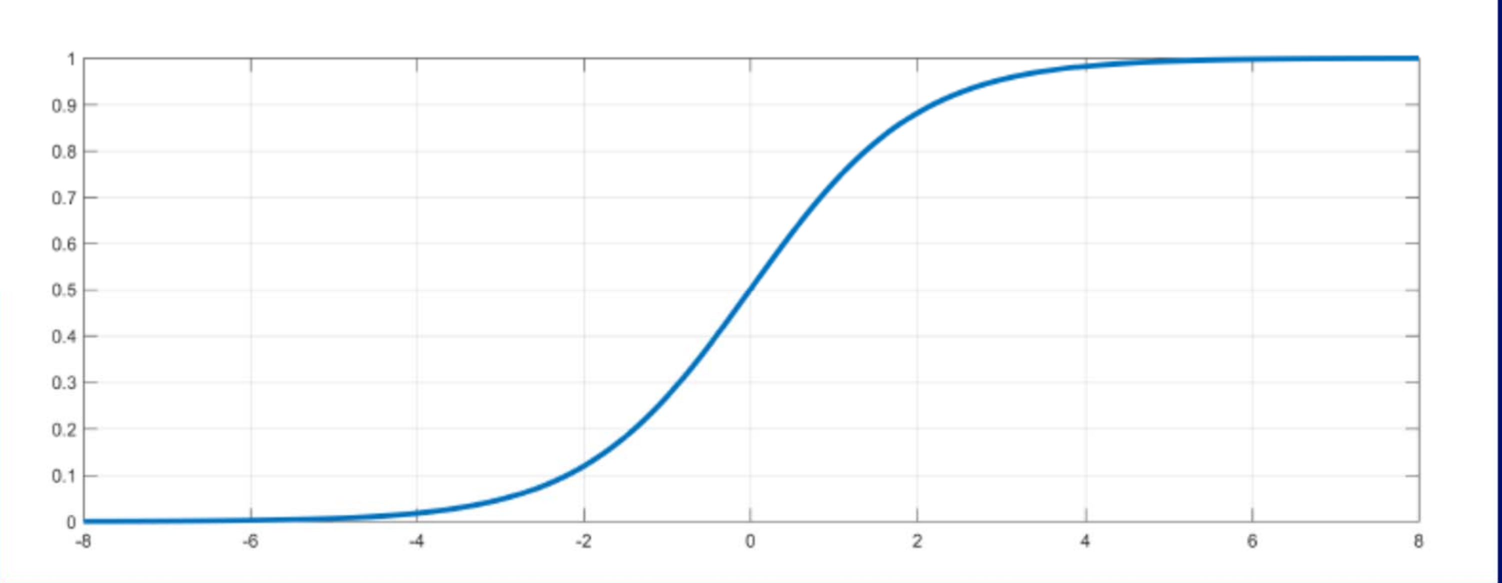


A better option is to use an S-shaped curve to fit the data.

Which functions have such shapes?

An example is the logistic/sigmoid function:

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

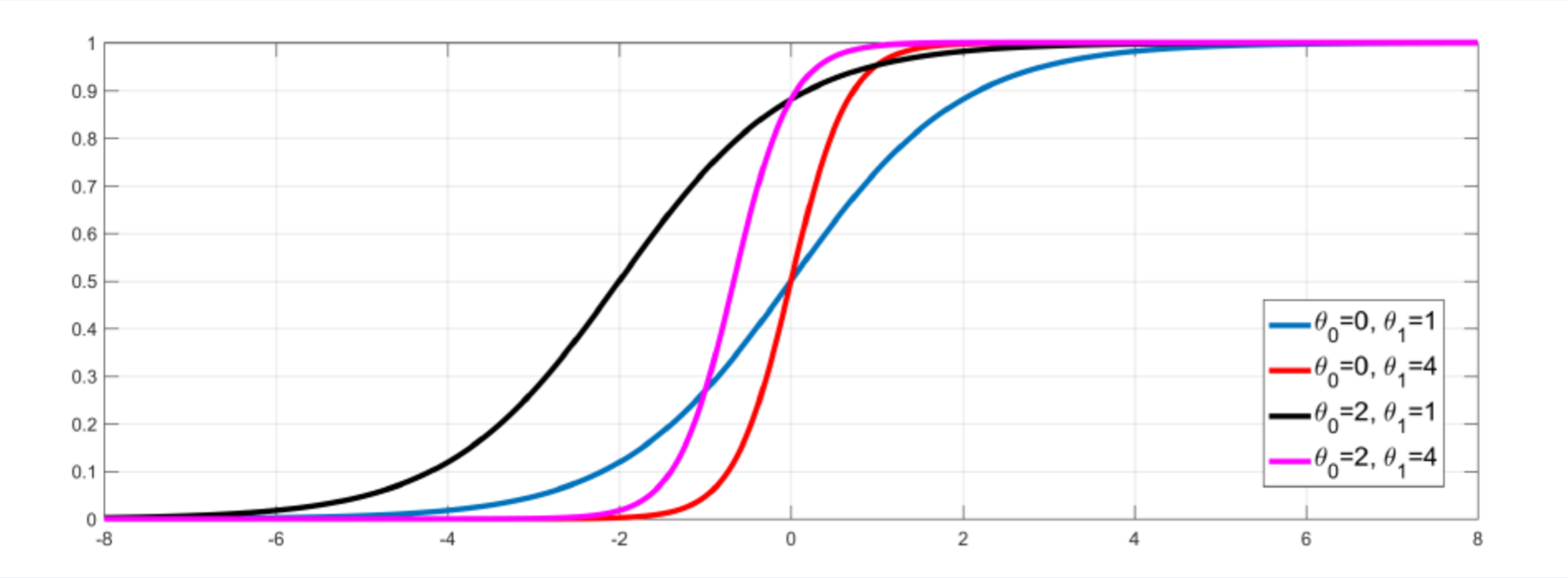


Making the logistic function more flexible

We generalize the logistic function to a location-scale family:

- Real θ_0 – location parameter
- Real θ_1 – scale parameter

$$g(\theta_0 + \theta_1 x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}}$$



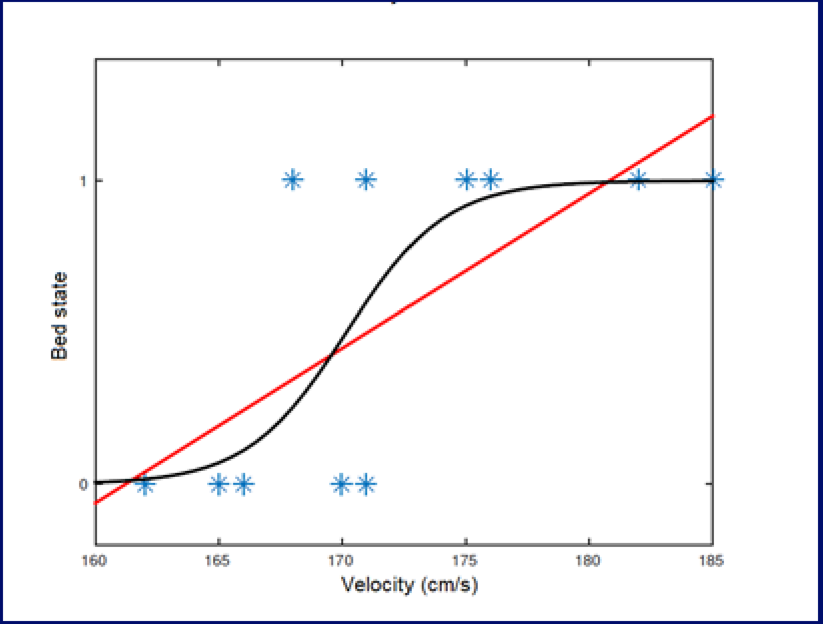
The logistic regression problem

Once we fix the template function $f(\theta_{0,}\theta_1,x)$, the logistic regression problem reduces to parameter estimation based on a dataset.

$$f_\theta(x) := f(\theta_0,\theta_1,x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x)}}$$

Problem:

Given training dataset of m examples $(x^{(i)}, y^{(i)})$, find $(\theta_{0,}\theta_1)$ such that the curve $f(\theta_{0,}\theta_1,x)$ fits the dataset in some optimal way.



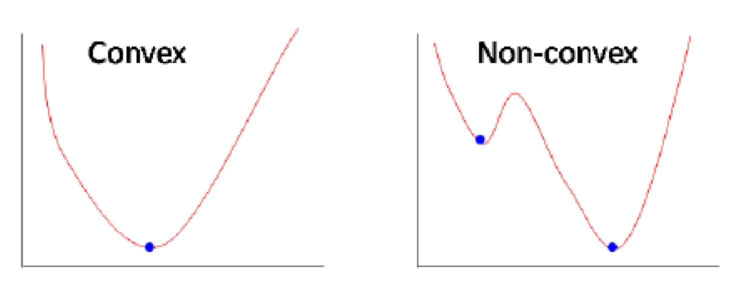
The optimization approaches

The cost function J(θ) based on the least square technique is applied to quantify the goodness of fit between the hypothesis function f($\theta_0, \theta_1, .., \theta_m, x_0, x_1, \ldots, x_m$) and the exact output.

$$error^{(i)} = f_\theta\left(X^{(i)}\right) - Y^{(i)}$$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(error^{(i)}\right)^2 = \frac{1}{2m}\sum_{i=1}^{m}\left(\frac{1}{e^{-\left(\theta_0+\theta_1 x_1^{(i)}\right)}} - Y^{(i)}\right)^2$$

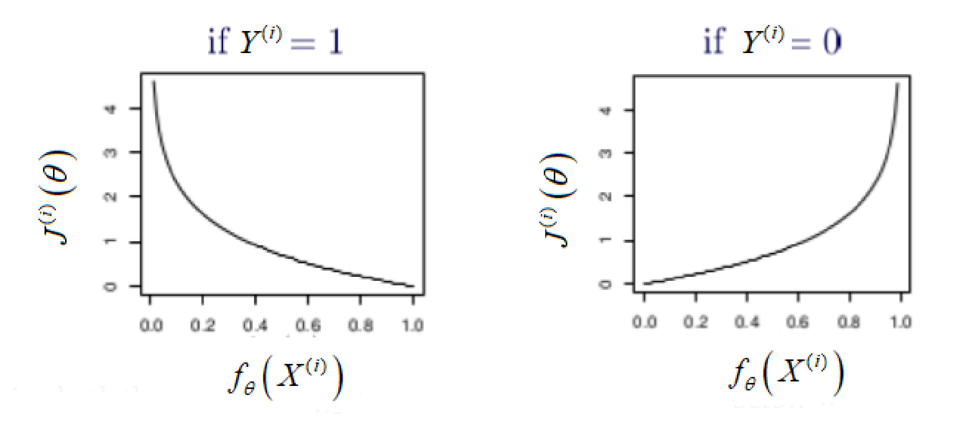
- Objective of Logistic regression is to minimize J(θ) by adjusting θ
- However for logistic regression, by calculating with the sigmoid function, the Least Squared Error will result in a non-convex loss function with local minimums.



least square cost function for linear regression (left)

least square cost function for logistic regression (right)

The logistic cost function J(θ)



- The lost/error function for each example:

$$J^{(i)}\left(f_\theta(X^{(i)}),Y^{(i)}\right)=\begin{cases} -\log\left(f_\theta(X^{(i)})\right) & if \quad Y^{(i)}=1 \\ -\log\left(1- f_\theta(X^{(i)})\right) & if \quad Y^{(i)}=0 \end{cases}$$

- The logistic cost function J(θ):

$$J(\theta)=\frac{1}{m}\sum_{i=1}^{m} J^{(i)}\left(f_\theta(X^{(i)}),Y^{(i)}\right)=\frac{1}{m}\sum_{i=1}^{m}\left[-Y^{(i)}\log\left(f_\theta(X^{(i)})\right)-\left(1-Y^{(i)}\right)\log\left(1- f_\theta(X^{(i)})\right)\right]$$

- Gradient of the cost function:

$$\frac{\partial J(\theta)}{\partial\theta_j}=\frac{1}{m}\sum_{i=1}^{m}\left[\left(f_\theta(X^{(i)})-Y^{(i)}\right)X_J^{(i)}\right]$$

- We store each example as a row in a MATLAB matrix:
  - *X* is (m×n) matrix.
  - *Y* is (m×1) column vector.
  - *θ* is (n×1) column vector.
- Adding an additional first column to *X* and set it to all ones, we can rewrite the hypothesis function, cost function and its gradients, and the gradient descent expression in matrix forms.

$$f_\theta(X) = g\left(X\theta\right)$$

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

$$J\left(\theta\right) = mean\left[\left(-Y\right).*\log\left(g(X*\theta\right) - \left(1-Y\right).*\log\left(1-g(X*\theta)\right)\right]$$

$$\nabla J\left(\theta\right) = \frac{1}{m}*X'*\left(g\left(X*\theta\right)-Y\right)$$

- Gradient descent

$$\theta_j := \theta_j - \alpha\frac{\partial J}{\partial\theta_j} \quad \Rightarrow \quad \theta := \theta - \alpha\nabla J\left(\theta\right)$$

Learning parameters using MATLAB built-in function called *fminunc*

- For logistic regression, you want to optimize the cost function J(θ) with parameters θ. Concretely, you are going to use *fminunc* to find the best parameters θ for the logistic regression cost function, given a fixed labeled dataset (of X and Y values).
- You will pass to *fminunc* the following inputs:
  - The initial values of the parameters we are trying to optimize.
  - A function that, when given the training set and a particular θ, computes the logistic regression cost and gradient with respect to θ for the dataset (X, Y)

Exercise 3.2:

Build a logistic regression model to predict whether a student gets admitted into a university.

- We have historical data from previous applicants that can be used as a training set for logistic regression. Data set: *example_3_2.txt* (with 100 examples):
  - $x_{1,}\ x_2$ - applicant's scores on two exams
  - *Y* - admissions decision.
- Uncompleted MATLAB-scripts
  - *Exercise3_2.m* - Script that steps you through this exercise
  - *plotData.m* - Function to display 2D classification data
  - *sigmoid.m* - Sigmoid function
  - *costFunction.m* - Function to compute the cost of logistic regression
  - *predict.m* – Logistic regression function
- Your task: to complete these MATLAB-scripts and run the program to predict whether a particular student with an Exam 1 score of 45 and an Exam 2 score of 85 will be admitted.

In order to optimize this convex function, we can either go with gradient-descent or newtons method. For both cases, we need to derive the gradient of this complex loss function.

Now we calculate the partial derivatives of the loss function for each example.

<u>Step 1</u>: Applying Chain-rule and writing in terms of partial derivative:

$$\frac{\partial J^{(i)}(\theta)}{\partial\theta_j} = -\frac{\partial}{\partial\theta_j}\left[Y^{(i)}\log\left(f_\theta(X^{(i)})\right)+\left(1-Y^{(i)}\right)\log\left(1-f_\theta(X^{(i)})\right)\right]$$

$$= -\left\{\left[Y^{(i)}\frac{1}{f_\theta(X^{(i)})}\frac{\partial}{\partial\theta_j}\left(f_\theta(X^{(i)})\right)\right]+\left[\left(1-Y^{(i)}\right)\frac{1}{\left(1-f_\theta(X^{(i)})\right)}\frac{\partial}{\partial\theta_j}\left(1-f_\theta(X^{(i)})\right)\right]\right\}$$

$$= -\left\{\begin{array}{l}\left[Y^{(i)}\frac{1}{f_\theta(X^{(i)})}g\left(X^{(i)}\theta\right)\left(1-g\left(X^{(i)}\theta\right)\right)\frac{\partial}{\partial\theta_j}\left(X^{(i)}\theta\right)\right]+ \\ +\left[\left(1-Y^{(i)}\right)\frac{1}{\left(1-f_\theta(X^{(i)})\right)}\left(-g\left(X^{(i)}\theta\right)\right)\left(1-g\left(X^{(i)}\theta\right)\right)\frac{\partial}{\partial\theta_j}\left(X^{(i)}\theta\right)\right]\end{array}\right\}$$

<u>Step 2</u>: Evaluating the partial derivative using the pattern of the derivative of the sigmoid function:

$$\frac{\partial J^{(i)}(\theta)}{\partial\theta_j} = -\left\{\begin{array}{l}\left[Y^{(i)}\frac{1}{f_\theta(X^{(i)})}g\left(X^{(i)}\theta\right)\left(1-g\left(X^{(i)}\theta\right)\right)\frac{\partial}{\partial\theta_j}\left(X^{(i)}\theta\right)\right]+ \\ +\left[\left(1-Y^{(i)}\right)\frac{1}{\left(1-f_\theta(X^{(i)})\right)}\left(-g\left(X^{(i)}\theta\right)\right)\left(1-g\left(X^{(i)}\theta\right)\right)\frac{\partial}{\partial\theta_j}\left(X^{(i)}\theta\right)\right]\end{array}\right\}$$

$$= -\left\{\begin{array}{l}\left[Y^{(i)}\frac{1}{f_\theta(X^{(i)})}f_\theta(X^{(i)})\left(1-f_\theta(X^{(i)})\right)x_j^{(i)}\right]+ \\ +\left[\left(1-Y^{(i)}\right)\frac{1}{\left(1-f_\theta(X^{(i)})\right)}\left(-f_\theta(X^{(i)})\right)\left(1-f_\theta(X^{(i)})\right)x_j^{(i)}\right]\end{array}\right\}$$

<u>Step 3</u>: Simplifying the terms by multiplication:

$$\frac{\partial J^{(i)}(\theta)}{\partial\theta_j} = -\left\{\begin{array}{l}\left[Y^{(i)}\frac{1}{f_\theta(X^{(i)})}f_\theta(X^{(i)})\left(1-f_\theta(X^{(i)})\right)x_j^{(i)}\right]+ \\ +\left[\left(1-Y^{(i)}\right)\frac{1}{\left(1-f_\theta(X^{(i)})\right)}\left(-f_\theta(X^{(i)})\right)\left(1-f_\theta(X^{(i)})\right)x_j^{(i)}\right]\end{array}\right\}$$

$$= -\left\{\left[Y^{(i)}\left(1-f_\theta(X^{(i)})\right)x_j^{(i)}\right]-\left[\left(1-Y^{(i)}\right)f_\theta(X^{(i)})x_j^{(i)}\right]\right\}$$

$$= -\left\{\left[Y^{(i)}\left(1-f_\theta(X^{(i)})\right)\right]-\left[\left(1-Y^{(i)}\right)f_\theta(X^{(i)})\right]x_j^{(i)}\right\}$$

$$= -\left[Y^{(i)}-f_\theta(X^{(i)})\right]x_j^{(i)}$$

Further, we get the partial derivatives of the cost function:

$$\frac{\partial J\left(\theta\right)}{\partial\theta_j}=\frac{1}{m}\sum_{i=1}^{m}\frac{\partial J^{(i)}\left(\theta\right)}{\partial\theta_j}=-\frac{1}{m}\sum_{i=1}^{m}\left\{\left[Y^{(i)}-f_\theta(X^{(i)})\right]x_j^{(i)}\right\}$$

Finally, we rewrite the partial derivatives of the cost function in a matrix-form:

$$\nabla J\left(\theta\right)=\frac{1}{m}*X^{'}*\left(g\left(X*\theta\right)-Y\right)$$

$$\nabla J\left(\theta\right)=\frac{1}{m}*X^{'}*\left(f_\theta\left(X\right)-Y\right)$$