SUPERVISED MACHINE LEARNING (REGRESSION)

Machine Learning – Yousef Elbaroudy

GUIDELINES

- Try to focus on the important information mentioned through the session
- Apply what you take on the practical section
- Do not try to memorize everything you got, just learn
- Don’t mind to ask about anything you want to know

Enjoy the Session ☺

EVALUATION METRICS

Each Machine Learning Problem has it is own Evaluation Metrics

CLASSIFICATION EVALUATION METRICS

- **Evaluation metrics** for classification are used to assess the performance of a classification model by comparing its predictions to the actual class labels in a dataset.

- These metrics help quantify how well the model is performing and provide insights into its strengths and weaknesses.

Accuracy	Precision	Recall	F1-Score	ROC	AUC

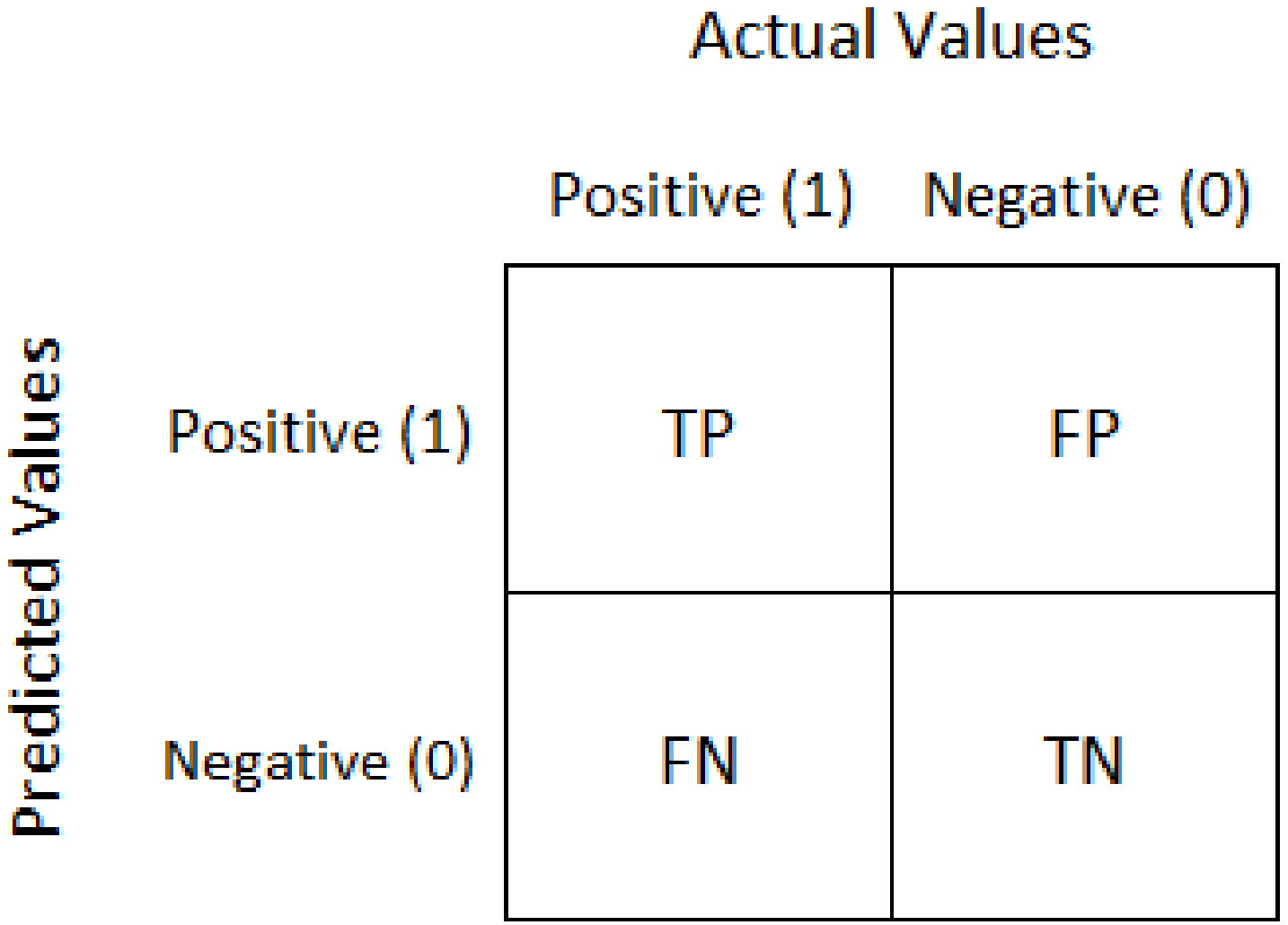
ACCURACY

Accuracy: Accuracy measures the proportion of correctly predicted instances out of the total instances in the dataset. It's a simple and intuitive metric but may not be suitable for imbalanced datasets.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

Confusion Matrix: A confusion matrix provides a detailed breakdown of true positive, true negative, false positive, and false negative predictions. It's a useful tool for understanding the distribution of prediction errors.

CONFUSION MATRIX


Actual Values
Positive (1)
Negative (0)
Predicted Values
Positive (1)
Negative (0)
TP
FP
FN
TN

Precision: Precision measures the proportion of true positive predictions (correctly predicted positives) out of all predicted positives. It focuses on the correctness of positive predictions.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Used for Medical Diagnosis

PRECISION

Recall (Sensitivity or True Positive Rate): Recall measures the proportion of true positive predictions out of all actual positives. It focuses on the completeness of positive predictions.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Used for information Retrieving

RECALL

F1-SCORE

F1-Score: The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is especially useful when class imbalance is present.

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Specificity (True Negative Rate): Specificity measures the proportion of true negative predictions out of all actual negatives. It's the complement of the false positive rate.

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

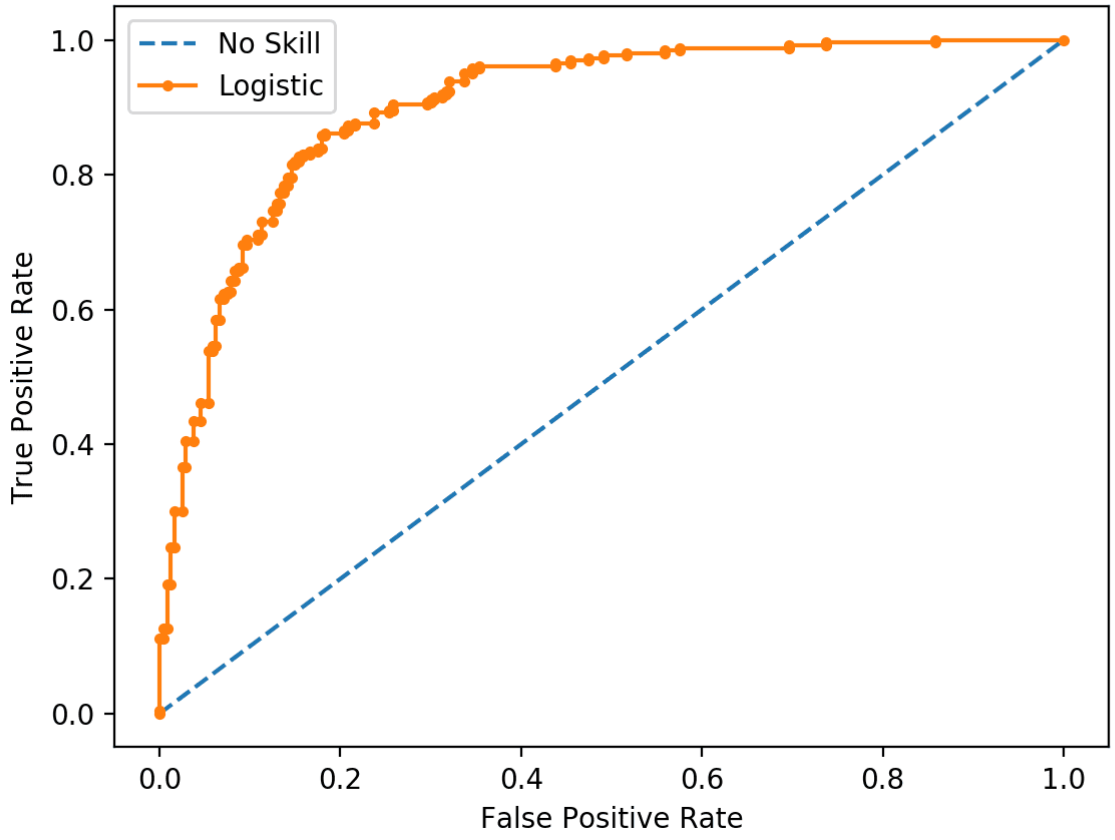
SPECIFICITY

RECEIVER OPERATION CHARACTERISTIC (ROC)

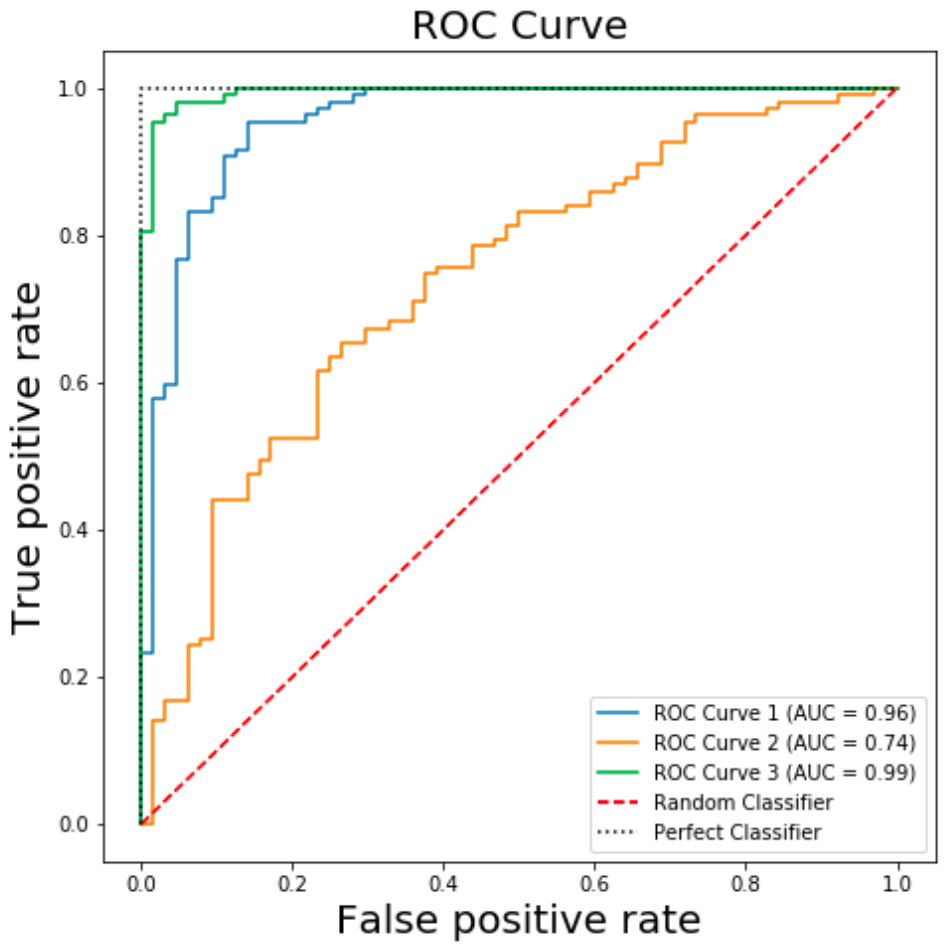
- **ROC curve** is a graphical representation of the true positive rate (recall) versus the false positive rate (1 – Specificity) for different threshold values.

- **The area under the ROC curve (AUC-ROC)** is a common metric that quantifies the overall performance of a classifier.

AUC-ROC CURVE



AUC-ROC CURVE



FOR MULTICLASS

It require one-hot encoding to Output label (preferred)

TIME FOR PRACTICALITY (5 MINUTES)

BREAK (10 MINUTES)

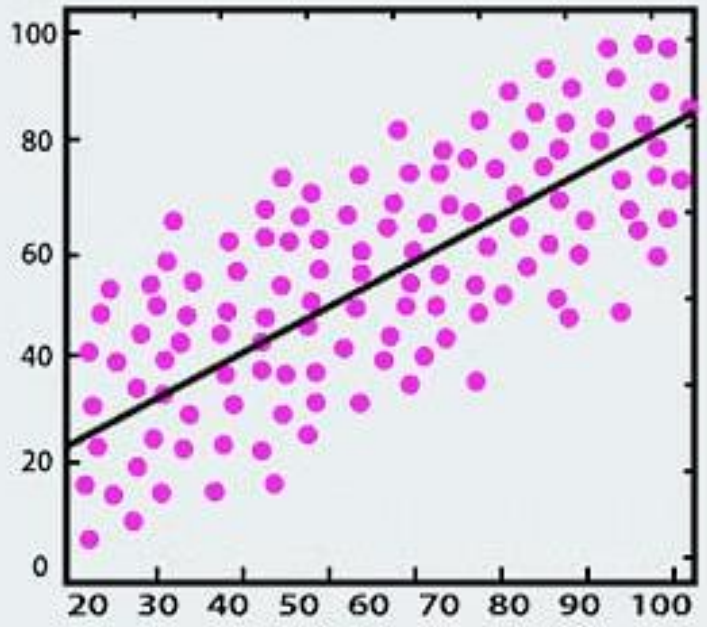
SUPERVISED MACHINE LEARNING (REGRESSION)

WHAT IS REGRESSION ?

- **Regression** is a statistical technique used in data analysis to model the relationship between a **dependent variable** (also called the outcome or target variable) and **one or more independent variables** (also known as predictor variables or features).

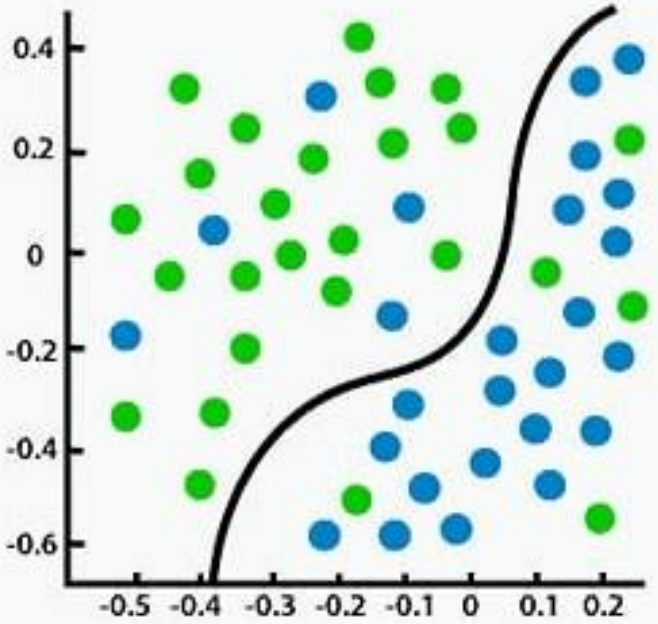
- **The main goal** of regression analysis is to understand and quantify how changes in the independent variables are associated with changes in the dependent variable.

In simpler terms, regression helps us predict or estimate a continuous numerical outcome based on one or more input variables.



Regression

versus



Classification

WHAT IS REGRESSION USED FOR ?

Prediction: Given new input values, regression models can be used to predict the likely value of the dependent variable.

Inference: Regression analysis can help us understand the relationship between variables and provide insights into how changes in the predictors affect the outcome.

Control: In experimental settings, regression can be used to determine how changes in the independent variables impact the dependent variable, allowing for better control over the experimental conditions.

TYPES OF REGRESSION

1. **Linear Regression:** The most common type, where the relationship between the variables is modeled as a straight line. It assumes a linear relationship between the predictors and the outcome.
2. **Multiple Regression:** Extends linear regression to multiple independent variables, allowing for more complex modeling.
3. **Polynomial Regression:** Accounts for curved relationships between variables by using polynomial equations.
4. **Logistic Regression:** Despite its name, it's used for binary classification problems. It estimates the probability of a binary outcome (e.g., yes/no) based on input variables.
5. **Ridge and Lasso Regression:** Variations of linear regression that include regularization techniques to prevent overfitting and improve model generalization.

TYPES OF REGRESSION (CONT.)

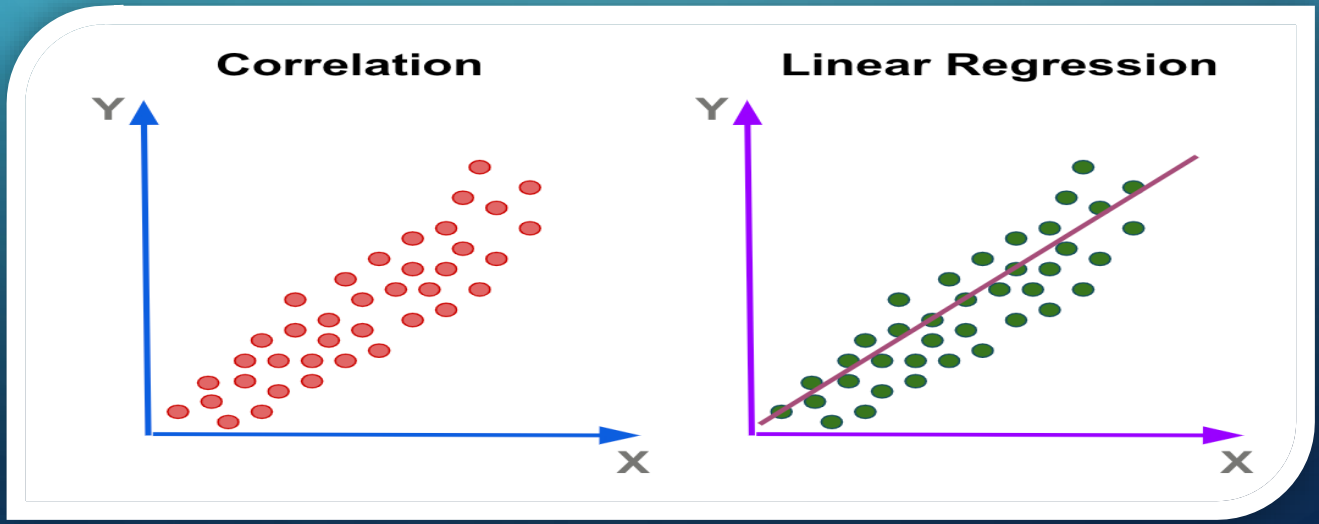
6. **Support Vector Regression:** Uses support vector machines to perform regression tasks, particularly useful for complex datasets.
7. **Decision Tree Regression:** Utilizes decision trees to predict continuous values based on feature inputs.
8. **Random Forest Regression:** Averages the predictions of multiple decision tree regressors to improve accuracy and control overfitting.
9. **Gradient Boosting Regression:** Builds an ensemble of weak prediction models sequentially to create a strong predictive model.

LINEAR REGRESSION

LINEAR REGRESSION

- **Linear regression** is a statistical method used to model the relationship between a dependent variable (target) and one or more independent variables (predictors) by ***assuming a linear relationship between them***.

- It's one of the simplest and most commonly used regression techniques.



EXAMPLE

X	1	2	3	4	5	6
y	10	15	20	25	??	??

SUGGEST A WAY TO SOLVE FOR X = 5, X = 6

ONE OF THE IDEAS

$$y = mx + b$$

1

Try to fit the previous points on a line

2

Use the slope equation and find the slope

$m = y2 - y1 / x2 - x1$

3

Find the y-intercept e.g. b of the equation

4

Substitute x values on the equation to find the y values

x_1	y_1
1	10
2	15
3	20
4	25

$$y_1 \sim mx_1 + b$$

STATISTICS

$r^2 = 1$

$r = 1$

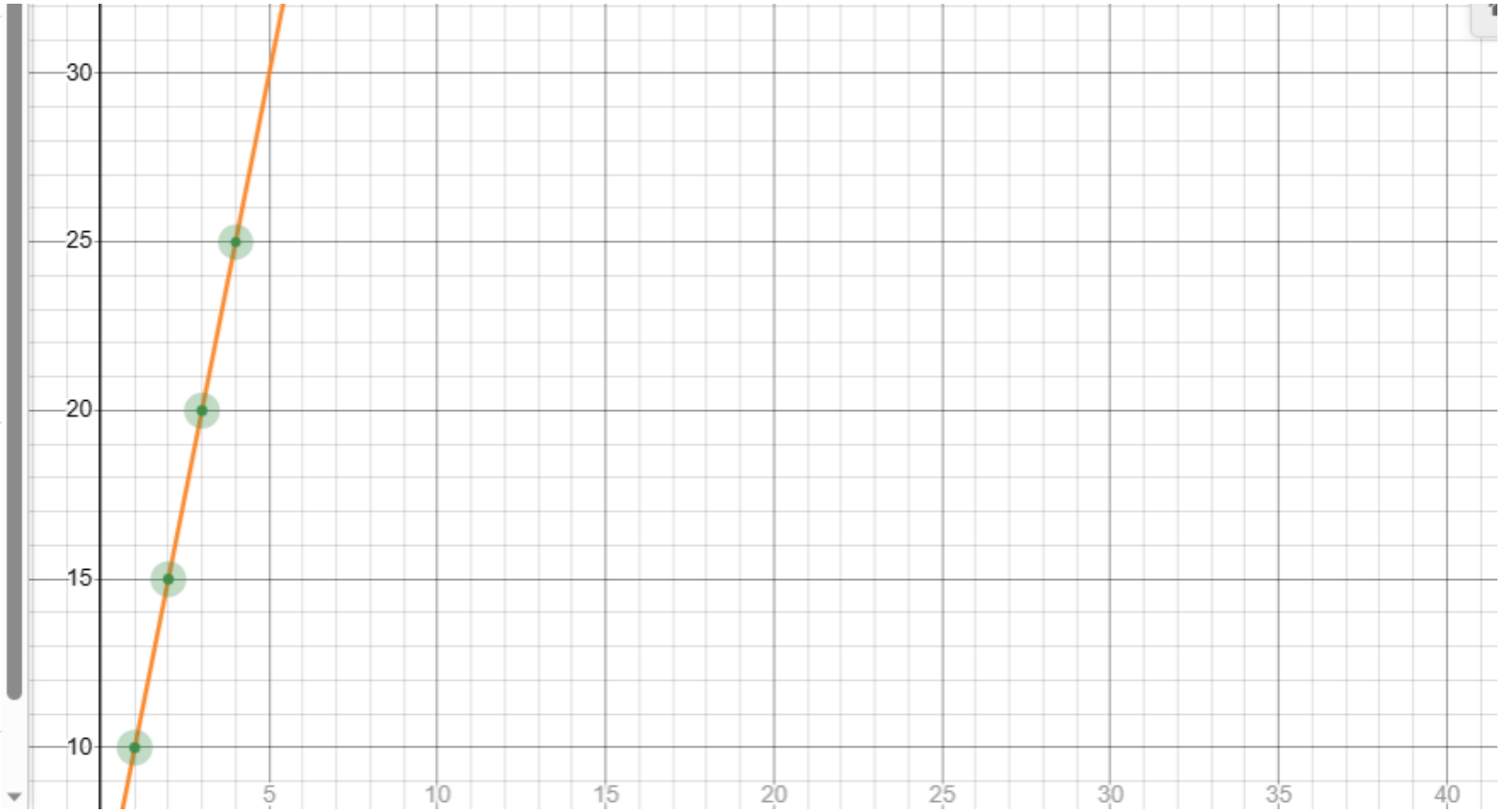
RESIDUALS

e_1 plot

PARAMETERS

$m = 5$

$b = 5$



$Y = 5X + 5$

X	1	2	3	4	5	6
y	10	15	20	25	30	??

$Y = 5X + 5$

X	1	2	3	4	5	6
y	10	15	20	25	30	35

WHAT IF

X	1	2	3	4	5	6
y	1	9	11	31	??	??

x_1	y_1
1	1
2	9
3	11
4	31

$y_1 \sim mx_1 + b$

STATISTICS

$r^2 = 0.8672$

$r = 0.9312$

RESIDUALS

e_1 plot

THE ENTIRE PROBLEM

THE IDEA

- In linear regression, the goal is to **FIND THE BEST-FITTING STRAIGHT LINE (LINEAR EQUATION)** that represents the relationship between the predictor(s) and the target variable.

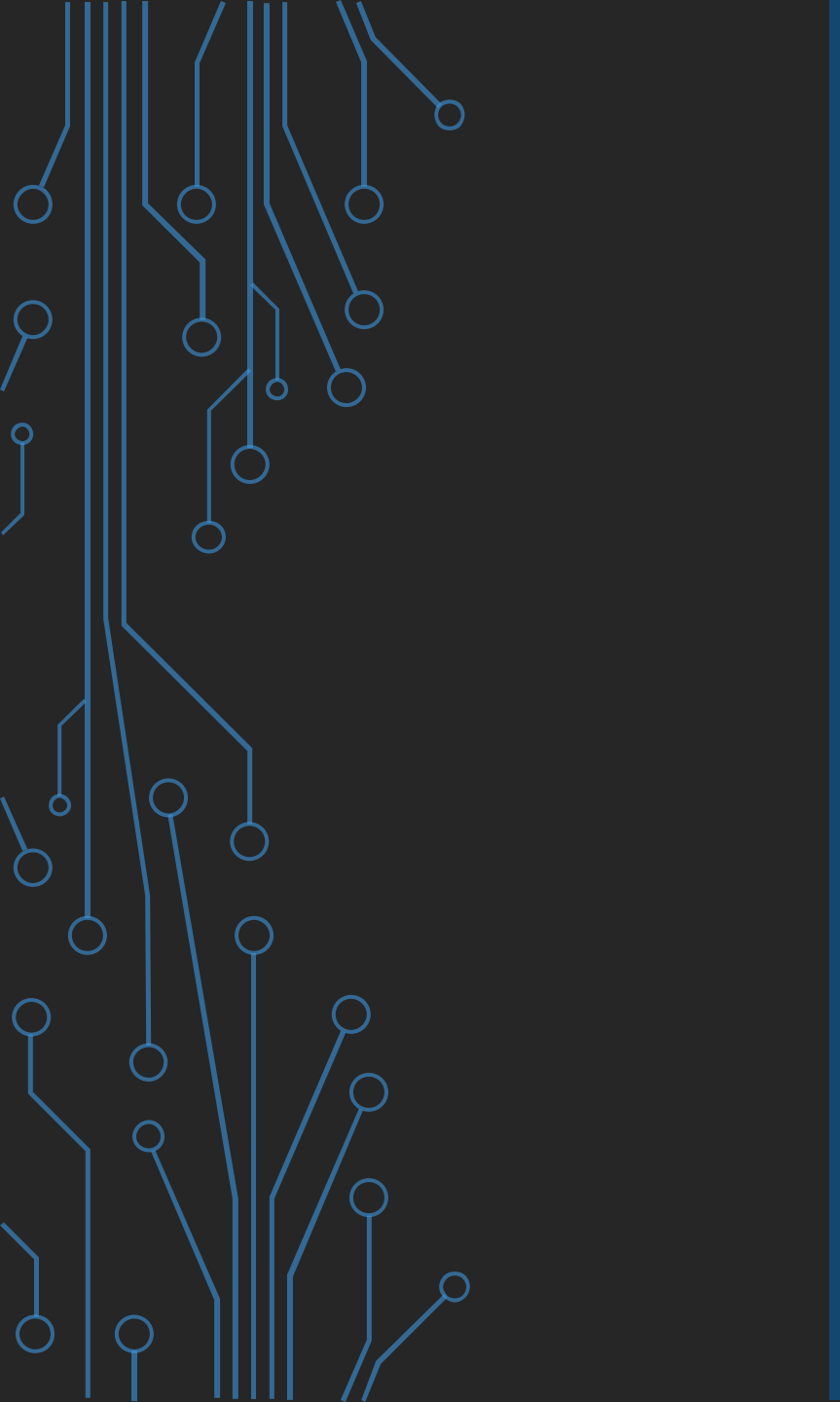
$$y = mx + b$$

Where:

- y is the dependent variable (target).
- x is the independent variable (predictor).
- m is the slope of the line, representing how much y changes for a unit change in x.
- b is the y-intercept, indicating the value of y when x is 0.

METHODS TO FIND THE BEST FITTING-LINE

- Ordinary Least Squares (OLS)
- Gradient Descent (GD)
- Ridge
- Lasso
- Elastic Net
- Stochastic Gradient Decent (SGD)



LEAST-SQUARES

LEAST-SQUARES METHOD

- The method of **least squares** is a mathematical approach used in various fields, to estimate the parameters of a model that minimizes the sum of the squared differences between the observed data points and the values predicted by the model.

- In the context of linear regression, the goal is to find the best-fitting line (linear equation) that represents the relationship between the independent variable(s) and the dependent variable.

The cost function of the Linear Regression

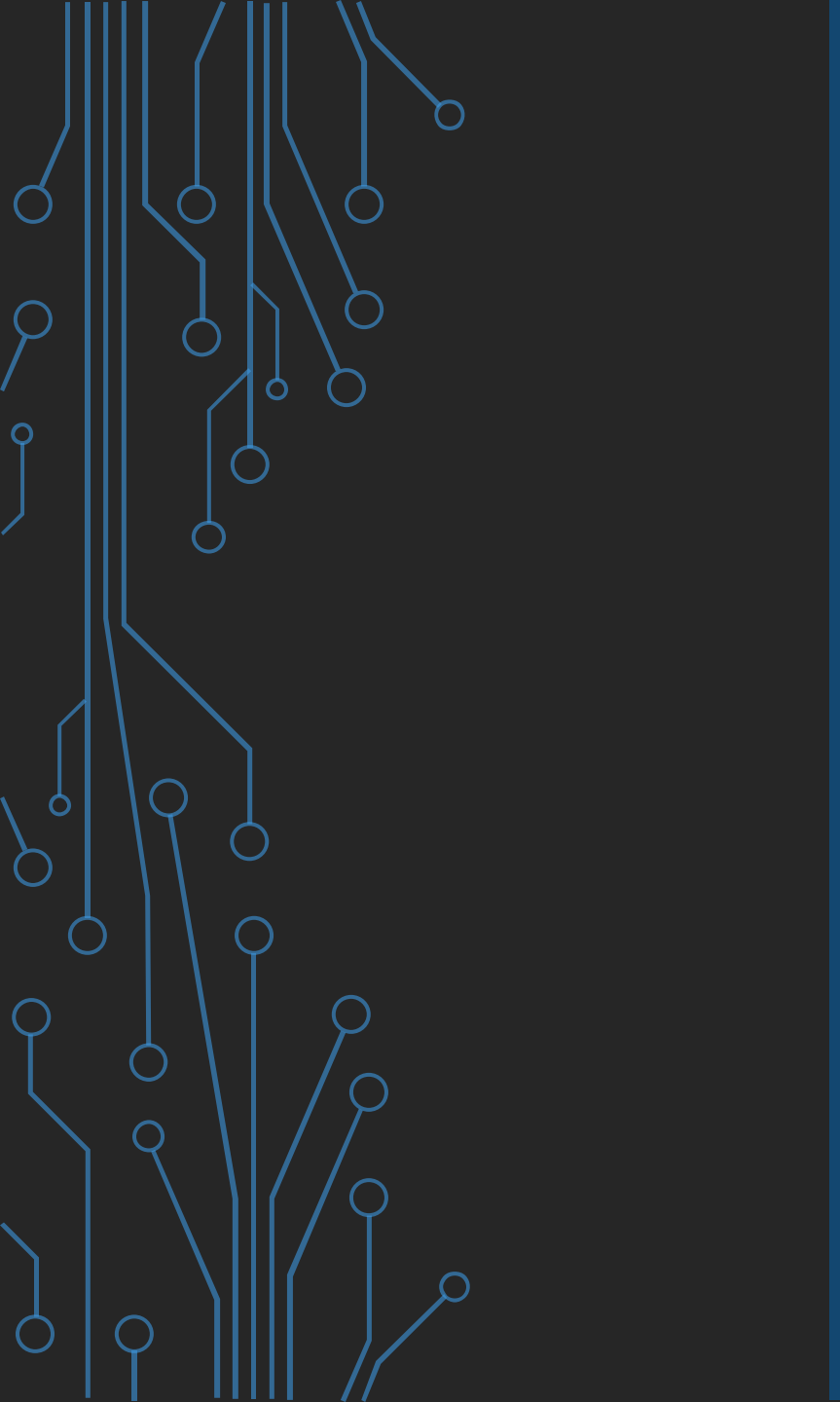
LEAST SQUARES METHOD (CONT.)

1) **Formulate the model:** Start with a linear model that represents the relationship between the independent variable(s) and the dependent variable, using the equation $y = mx + b$
2) **Calculate Residuals (Error Rate):** Calculate the residuals for each data point. Residuals are the differences between the observed values (y) and the values predicted by the model (mx+b) for each corresponding x, $[yi – (mx + b)]$.
3) **Square and Sum the Residuals:** Square each of the calculated residuals and then sum up these squared values. This step emphasizes larger errors while keeping the values positive.
4) **Minimize the Sum of Squared Residuals:** The primary objective of the least squares method is to find the values of mm and bb that minimize the sum of squared residuals.
5) **Calculate Optimal Coefficients:** Solving the equations from the previous step provides the optimal values for mm and bb that result in the best-fitting line according to the least squares criterion.
6) **Interpret the Results:** Once the optimal coefficients (mm and bb) are calculated, you have the equation of the best-fitting line.

THERE ARE STILL A QUESTIONS

- How to choose initial value to M (slope) and b (y-intercept) for the Least Squares ?
- How the change the values of M and b then and after ?

Here comes the OPTIMIZATION



NUMERICAL METHODS (OPTIMIZATION)

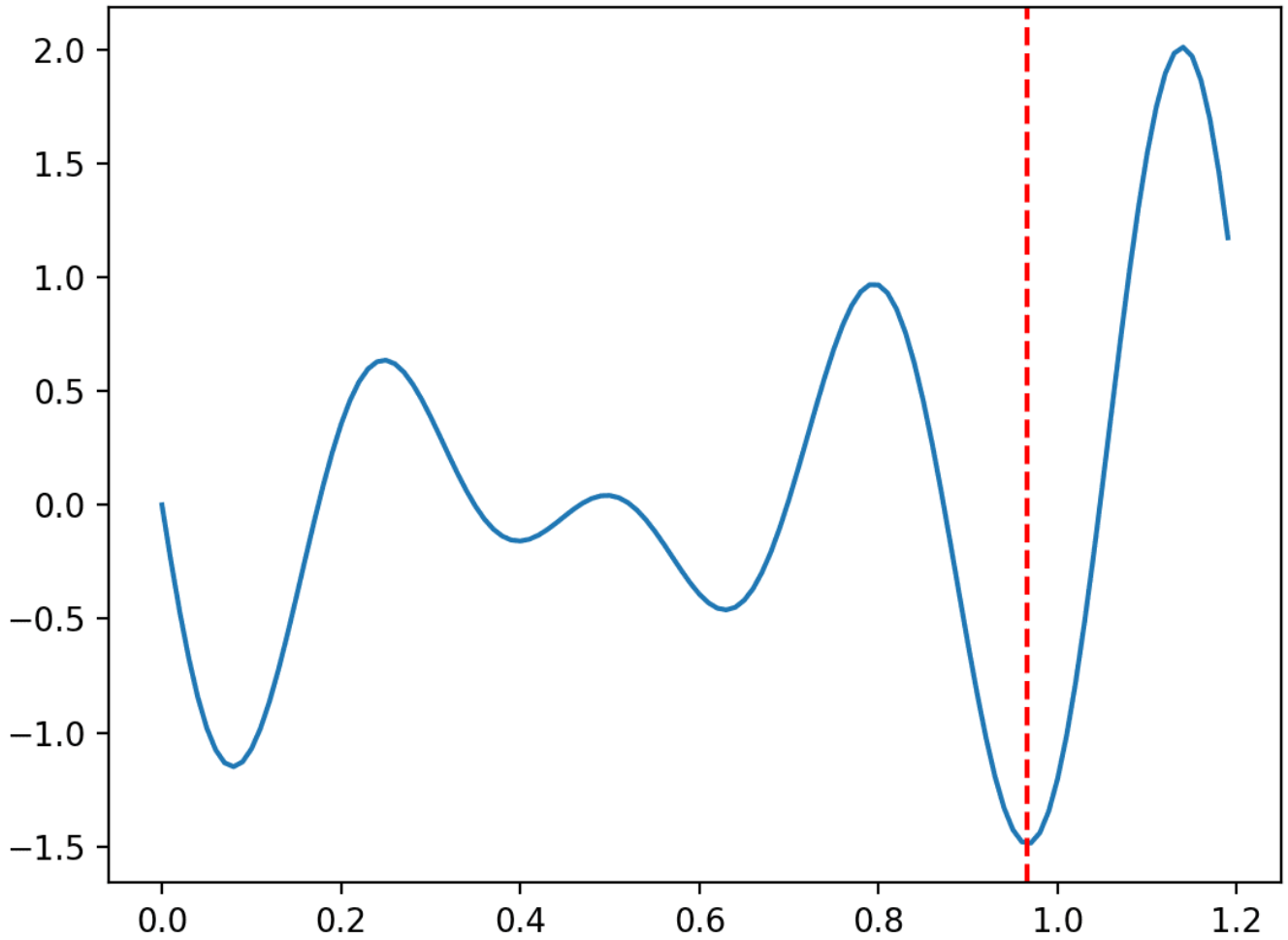
WHAT IS OPTIMIZATION ?

- **Optimization** refers to the process of finding the best possible solution among a set of possible alternatives, with the goal of maximizing or minimizing a specific objective or criteria.
- In various fields, optimization is used to make decisions, allocate resources, design systems, and solve problems in the most efficient or effective way.
- In mathematical terms, optimization involves finding the values of variables that either maximize or minimize a particular function, known as the objective function, while satisfying a set of constraints.

CATEGORIES OF OPTIMIZATION

Maximization Problems: In these problems, the goal is to find the values of variables that maximize the value of the objective function while adhering to any given constraints.

Minimization Problems: In these problems, the goal is to find the values of variables that minimize the value of the objective function while adhering to any given constraints.


2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0
1.2

OPTIMIZATION IS USED IN MANY DISCIPLINES

- **Mathematics:** Optimization theory is a branch of mathematics that deals with formalizing and solving optimization problems. It includes various algorithms and techniques for finding optimal solutions.
- **Engineering:** Engineers use optimization to design systems, products, and processes that are efficient, cost-effective, and meet specific performance criteria.
- **Economics:** Economists use optimization to model and analyze decision-making processes, resource allocation, and economic systems.
- **Operations Research:** This field focuses on applying mathematical and analytical methods to solve complex decision-making problems related to resource allocation, logistics, scheduling, and more.
- **Machine Learning:** Optimization plays a crucial role in training machine learning models by adjusting model parameters to minimize prediction errors.

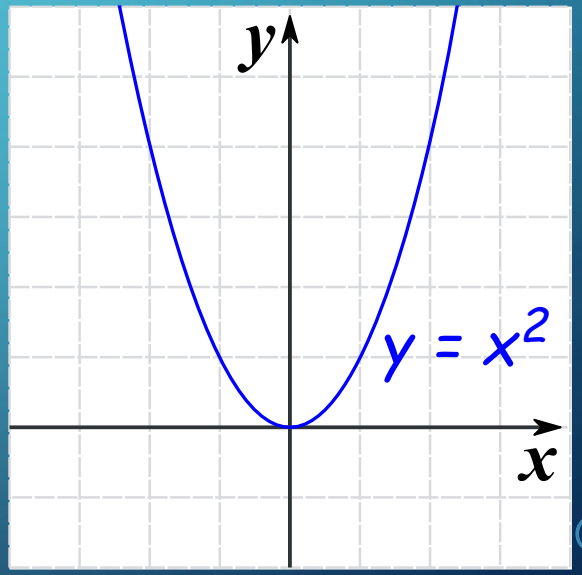
SOME METHODS TO OPTIMIZE THE LEAST-SQUARES

- Newton-Raphson Method
- Gradient Descent
- Conjugate Gradient
- Levenberg-Marquardt Algorithm

NEWTON-RAPHSON METHOD

In discuss of the square function

- You know that the solution (roots) of any **SECOND-DEGREE** equations means the x values that satisfy y = 0
- Assume the following function (square function) has only one **GLOBAL MINIMUM,** how to find it ?

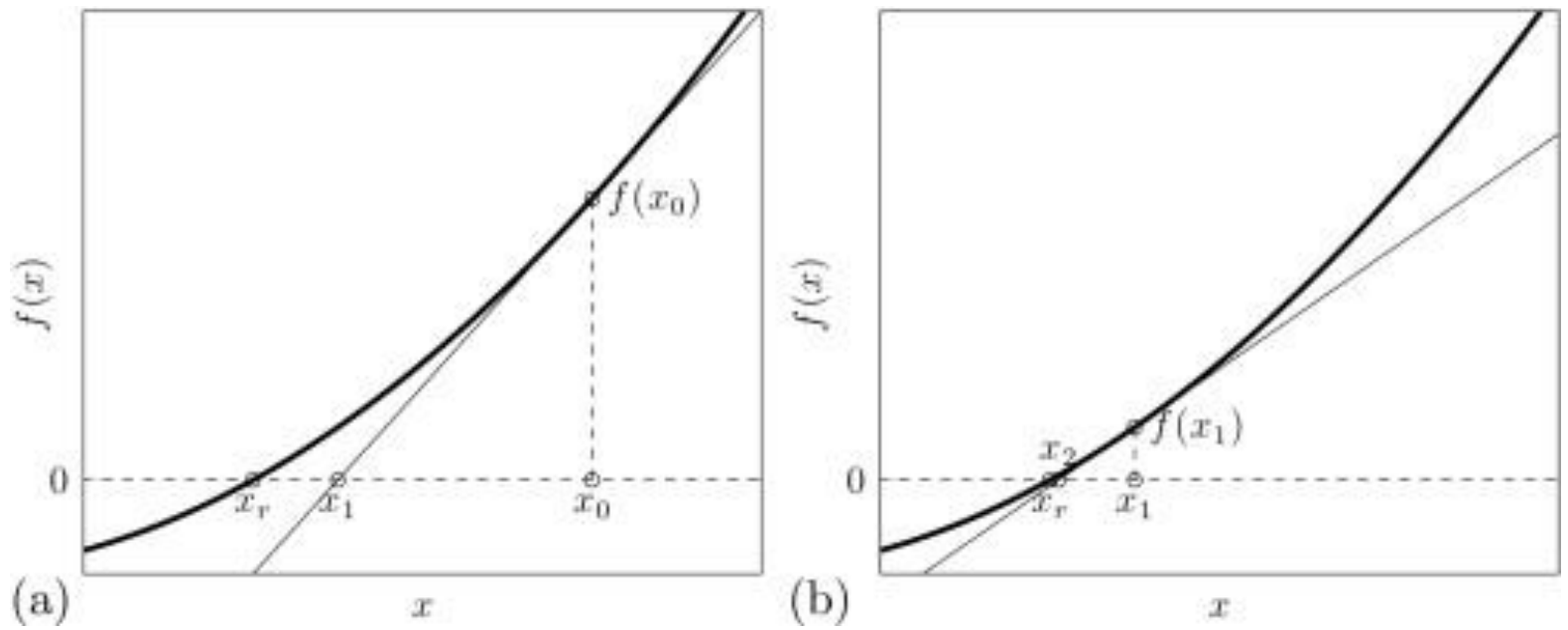


NEWTON-RAPHSON METHOD (CONT.)

- **The Newton-Raphson method**, often simply referred to as Newton's method, is an **ITERATIVE** optimization algorithm used to find approximate solutions to equations by successively refining an initial guess.

- **NOTE !** You know in **Differential Calculus** that at any critical point (whether minimum or maximum points) the slope or the change rate equal to zero

- **The Newton-Raphson method** begins with an initial estimate of the root, denoted $x_0 \neq x_r$, and uses the tangent of $f(x)$ at x_0 to improve on the estimate of the root.



NEWTON-RAPHSON METHOD (CONT.)

NEWTON-RAPHSON METHOD (CONT.)

$$x_{n+1} = x_n - \frac{mf(x_n)}{f'(x_n)}, \quad \text{for} \quad n = 0, 1, \ldots \qquad (11.264)$$

Note for the sign of the equation (-)

m is a parameter will be discussed later

Assume m = 1

STOPPING CRITERIA

- **The stopping criteria**, also known as the termination criteria, refer to the conditions that determine when an iterative optimization algorithm should stop or terminate its iterations.

1. **Maximum Number of Iterations:** A fixed number of iterations is predefined, and the algorithm stops when this maximum number is reached.

2. **Small Gradient or Gradient Norm:** For optimization algorithms that use gradients (e.g., gradient descent), the algorithm can stop when the gradient becomes very small or when its norm falls below a certain threshold.

3. **Early Stopping:** In iterative algorithms like neural network training, early stopping monitors the validation performance and stops training if the validation error starts increasing, preventing overfitting.

EXAMPLE

Solve $g(x) = x^3 - 2x - 4$ **from initial estimate** $x_0 = 2.5$

Stopping Criteria: **Maximum iteration = 1**

Answer = 2.1045

BUT THERE IS A PROBLEM !

Newton-Raphson Method is used to solve the roots of any function only !

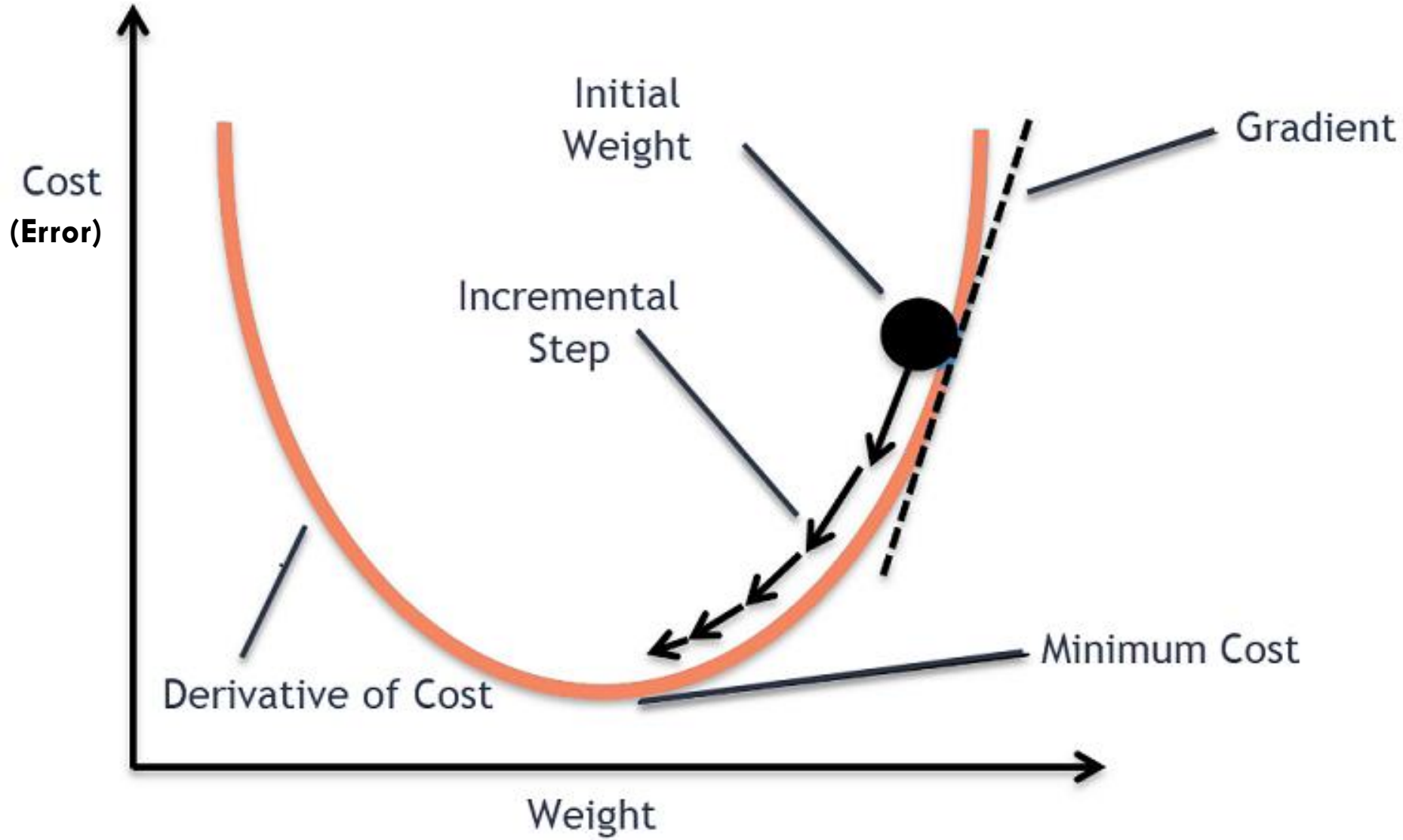
GRADIENT DESCENT

- **Gradient descent** is an iterative optimization algorithm used to find the minimum of a function, particularly in machine learning and numerical optimization.
- It's widely used for updating the parameters of a model in order to minimize a ***COST FUNCTION*** or objective function.
- The "gradient" in gradient descent refers to the gradient vector of **PARTIAL DERIVATIVES** of the function with respect to its parameters.

THE BASIC IDEA OF GRADIENT DESCENT

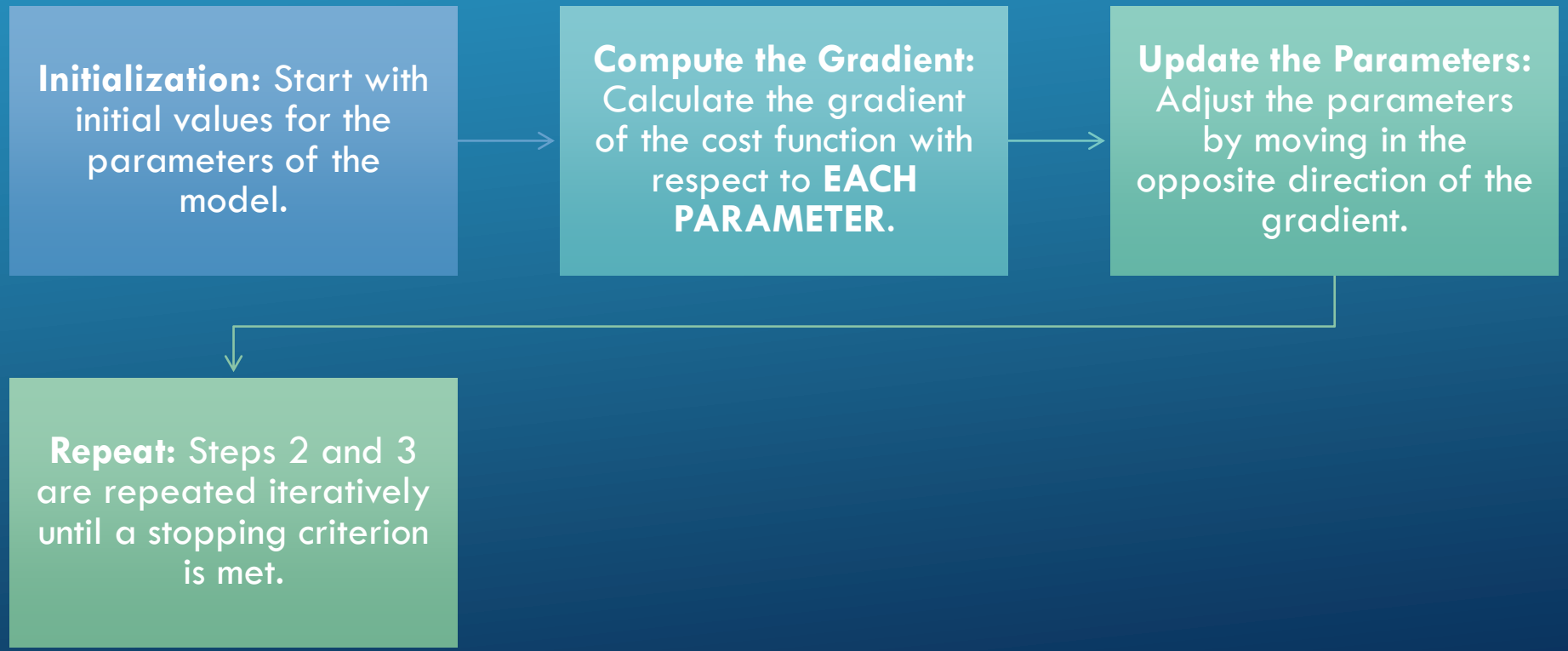
- The fundamental idea behind gradient descent is to **ITERATIVELY** adjust the parameters of a model in the direction of the steepest descent (negative gradient) of the cost function.

- By repeatedly taking steps in the direction that reduces the value of the cost function, the algorithm eventually converges to a local or global minimum.


Cost
(Error)
Initial
Weight
Gradient
Incremental
Step
Derivative of Cost
Minimum Cost
Weight

STEPS OF THE GRADIENT DESCENT

New Parameter=Old Parameter−Learning Rate × Gradient



TYPES OF GRADIENT DESCENT

1. **Batch Gradient Descent (BGD):** In batch gradient descent, the entire dataset is used to compute the gradient of the cost function in each iteration. It provides a precise estimate of the gradient but can be computationally expensive for large datasets.
2. **Stochastic Gradient Descent (SGD):** In stochastic gradient descent, a single randomly chosen data point is used to compute the gradient in each iteration. This approach is faster but can result in noisy updates and slower convergence due to the randomness.
3. **Mini-Batch Gradient Descent:** Mini-batch gradient descent strikes a balance between BGD and SGD by using a small batch of data points for each iteration. It provides a good trade-off between computational efficiency and noise in the gradient estimate.

BREAK (5 MINUTES)

BACK TO LINEAR REGRESSION

PUTTING ALL TOGETHER

- We are using (Least Square) and must be added to **COST FUNCTION**
- Our required parameters is M slope and b y-intercept
- We will use **Batch Gradient Descent** in order to modify the M and b parameters as to minimize the cost function (Error rate)
- This process will be iterative until the stopping criteria is met
- The line generated by using best m and b values are **BEST FITTING-LINE**

SOME NOTES

- The line equation $y = mx + b$ will be called the Hypothesis h → $h = mx + b$
- From now and on, we will call M and b as b1 and b0 respectively
- The final form of the equation → $h = b0 + b1x$

COST FUNCTIONS

- In linear regression, **the cost function** (also known as the loss function or objective function) quantifies how well the model's predictions match the actual observed values.
- The goal of linear regression is to find the parameters (coefficients) of the linear equation **THAT MINIMIZE THIS COST FUNCTION**.

1. **Mean Squared Error (MSE):** MSE is one of the most widely used cost functions in linear regression. It calculates the average squared difference between the predicted values and the actual values for all data points.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Least Squares

Where:

- n is the number of data points.
- y_i is the actual value of the dependent variable for the ith data point.
- $\hat{y}_i$ is the predicted value of the dependent variable for the ith data point.

Minimizing MSE results in the coefficients that provide the best-fitting line in terms of minimizing the sum of squared residuals.

This one will be used as cost function

MEAN SQUARED ERROR (MSE)

2. **Root Mean Squared Error (RMSE):** RMSE is the square root of the MSE and provides a measure of the average error between the predicted and actual values. It is commonly used when you want the error metric to be in the same units as the dependent variable.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

ROOT MEAN SQUARED ERROR (RMSE)

3. **Mean Absolute Error (MAE):** MAE computes the average absolute difference between the predicted and actual values. It is less sensitive to outliers compared to MSE.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

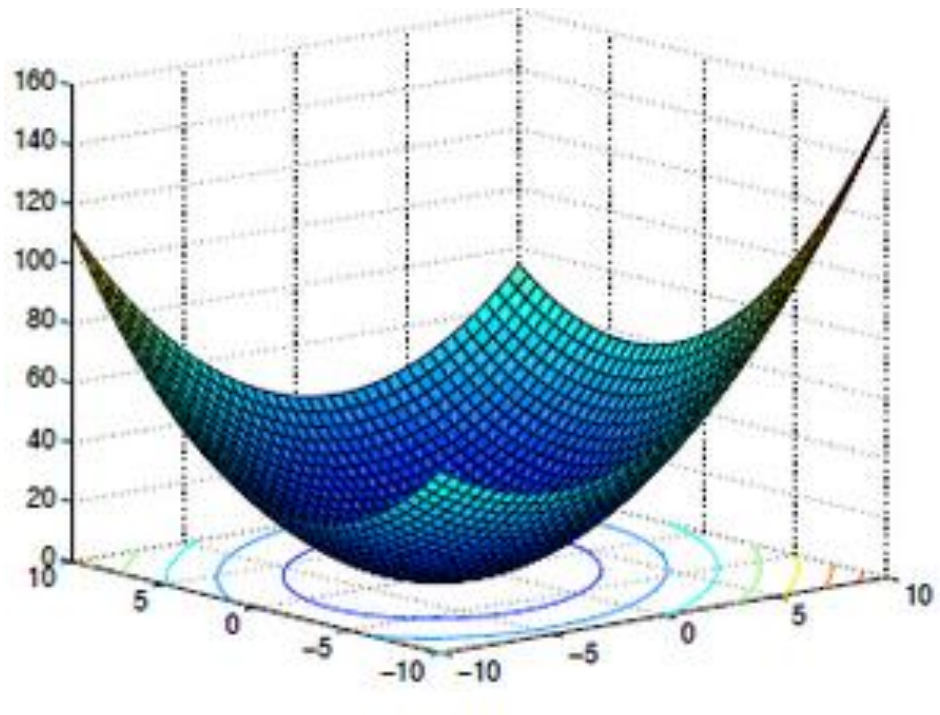
MEAN ABSOLUTE ERROR (MAE)

THE FULL EQUATION

$$J(b0, b1) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - h_i)^2$$

- Where $h = b_0 + b_1 x$
- And J is cost function of MSE

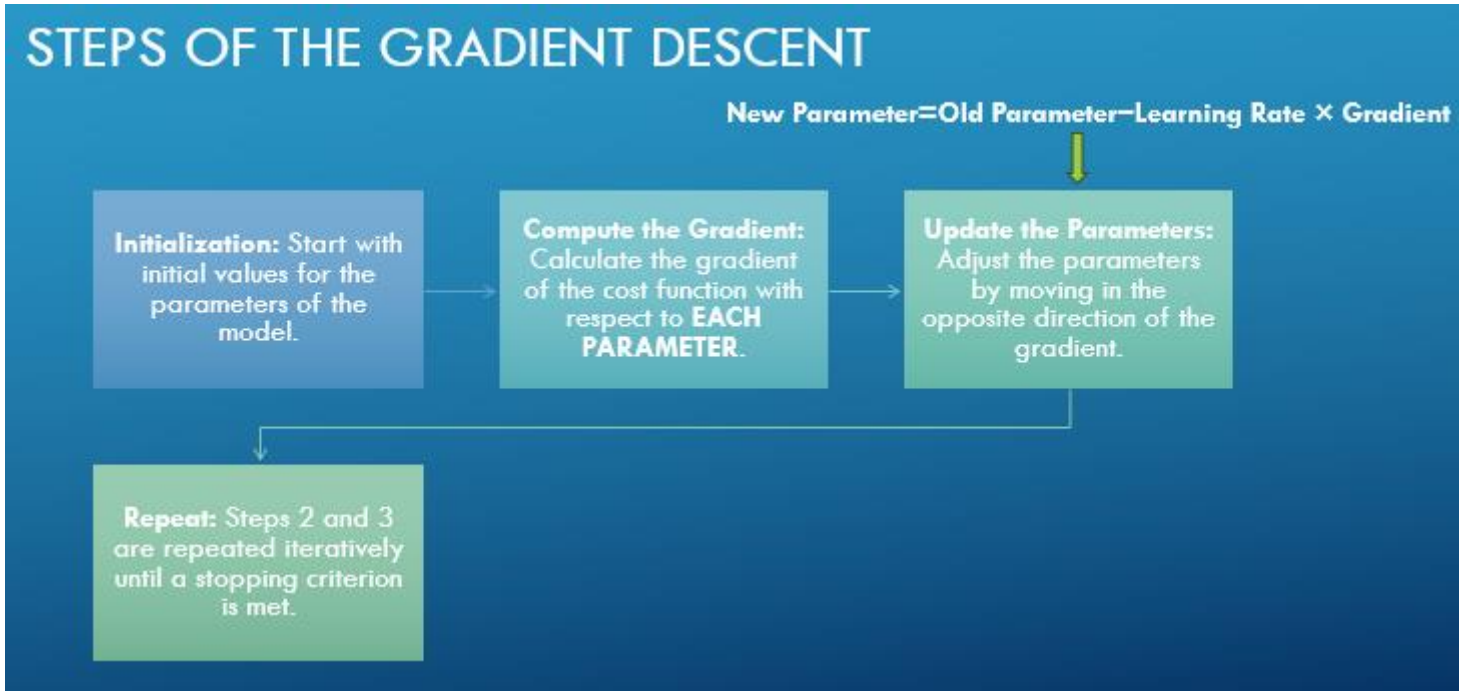
THE RELATION BETWEEN J, B0 AND B1



3D

THE TASK (LINEAR REGRESSION)

Using the gradient descent to minimize the cost function (error) that satisfy the best values for m [b1] and b [b0] (best fitting-line)


STEPS OF THE GRADIENT DESCENT
New Parameter=Old Parameter−Learning Rate × Gradient
Initialization: Start with initial values for the parameters of the model.
Compute the Gradient: Calculate the gradient of the cost function with respect to EACH PARAMETER.
Update the Parameters: Adjust the parameters by moving in the opposite direction of the gradient.
Repeat: Steps 2 and 3 are repeated iteratively until a stopping criterion is met.

TO REMEMBER

GETTING STARTED

Assume the following dataset:

X	1	2	3	4
y	1	9	11	31

Start with initial m = 7.6 and b = -8 for 1 iteration

Find thevalues of m [b_1] and b [b_0]

Then predict x = 5 and x = 6

FIRST (GRADIENT DESCENT)

New Parameter = Old Parameter − Learning Rate × Gradient

$$b_{0new} = b_{0old} - \text{Learning Rate} \times \frac{\partial J}{\partial b_0}$$

$$b_{1new} = b_{1old} - \text{Learning Rate} \times \frac{\partial J}{\partial b_1}$$

SECOND (FIND THE PARTIAL DERIVATIVE FOR B_0)

$$\frac{\partial J}{\partial b_0} = \frac{1}{2n}\sum_{i=1}^{n}(y_i - h_i)^2$$

$$\frac{\partial J}{\partial b_0} = \frac{1}{2n}\sum_{i=1}^{n}[y_i - (b_0 + b_1 x)]^2$$

$$\frac{\partial J}{\partial b_0} = \frac{1}{n}\sum_{i=1}^{n} -1 * [y_i - (b_0 + b_1 x)]$$

THIRD (FIND THE PARTIAL DERIVATIVE FOR B_1)

$$\frac{\partial J}{\partial b_1} = \frac{1}{2n}\sum_{i=1}^{n}(y_i - h_i)^2$$

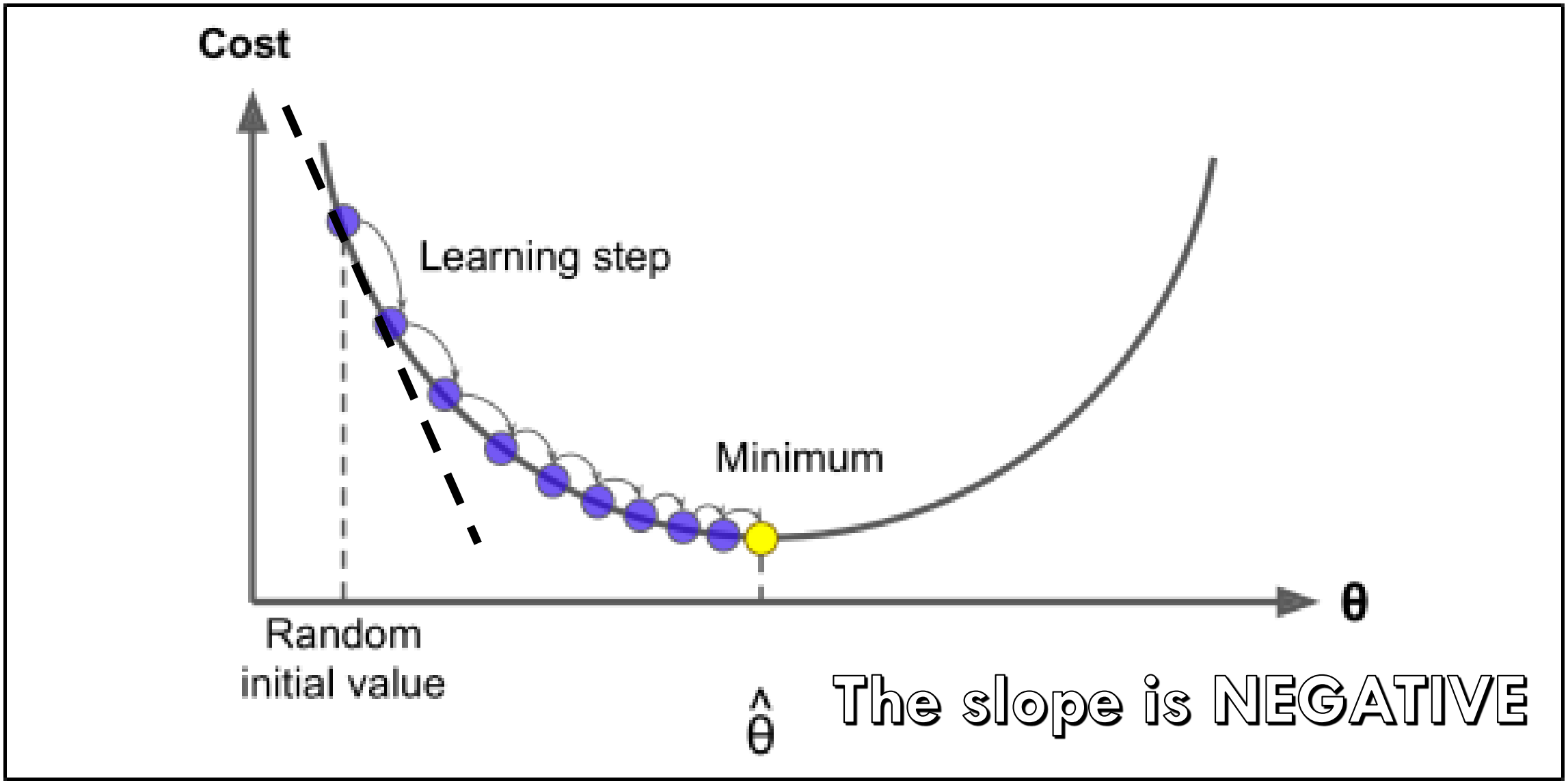
$$\frac{\partial J}{\partial b_1} = \frac{1}{2n}\sum_{i=1}^{n}[y_i - (b_0 + b_1 x)]^2$$

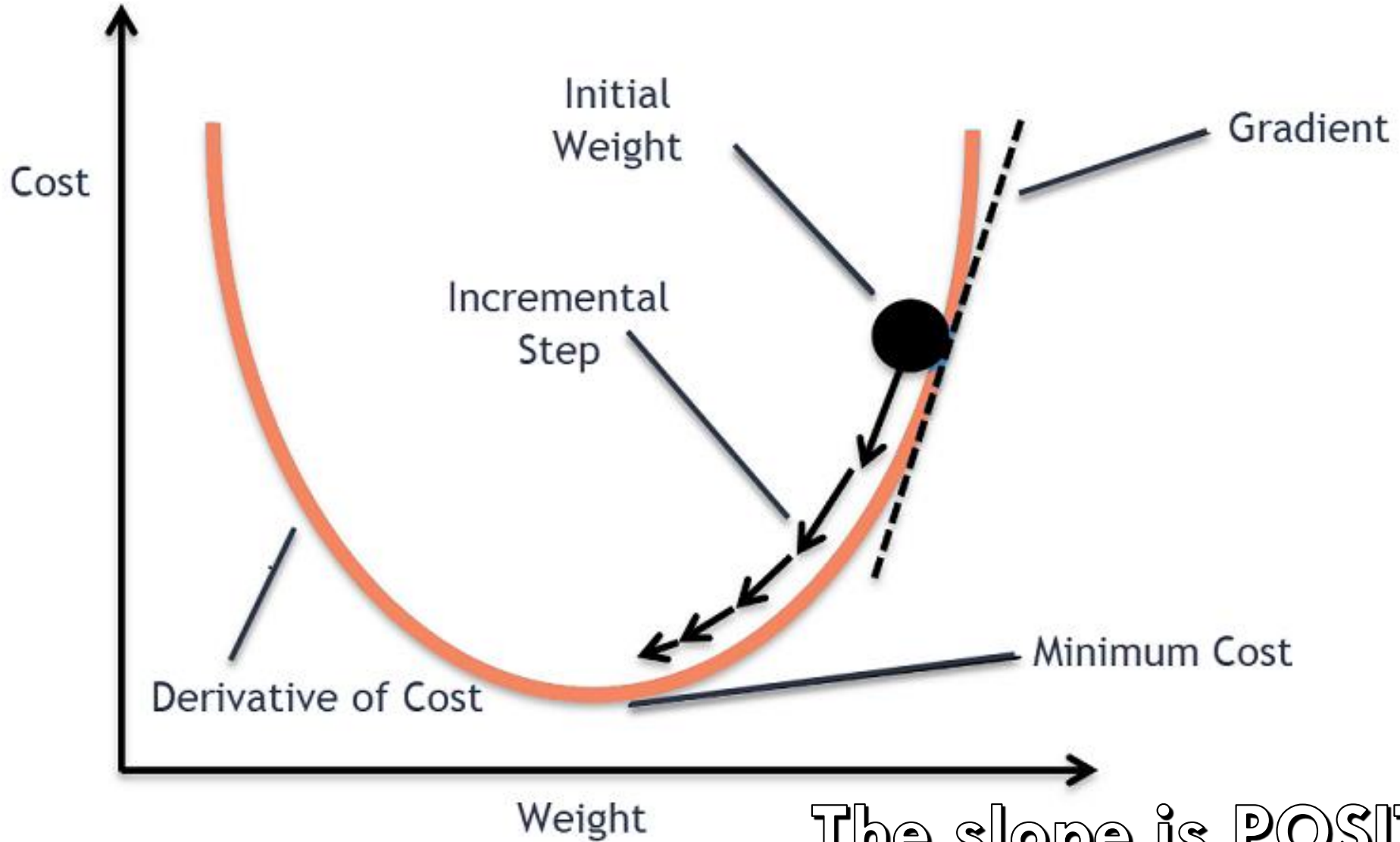
$$\frac{\partial J}{\partial b_1} = \frac{1}{n}\sum_{i=1}^{n} -x * [y_i - (b_0 + b_1 x)]$$

FOURTH (PUTTING THE GRADIENTS)

$$b_{0new} = b_{0old} - \text{Learning Rate} \times \frac{1}{n}\sum_{i=1}^{n} -1 * [y_i - (b_0 + b_1 x)]$$

$$b_{1new} = b_{1old} - \text{Learning Rate} \times \frac{1}{n}\sum_{i=1}^{n} -x * [y_i - (b_0 + b_1 x)]$$


Cost
Learning step
Minimum
θ
Random
initial value
θ̂
The slope is NEGATIVE



The slope is POSITIVE

FIFTH (COMPUTE THE GRADIENTS)

X	1	2	3	4
y	1	9	11	31

$$\frac{\partial J}{\partial b_0} = \frac{1}{n}\sum_{i=1}^{n} -1 * [y_i - (b_0 + b_1 x)]$$

$$\frac{\partial J}{\partial b_1} = \frac{1}{n}\sum_{i=1}^{n} -x_i * [y_i - (b_0 + b_1 x_i)]$$

$$\frac{\partial J}{\partial b_0} = \frac{1}{4}\sum_{i=1}^{4} -1 * [y_i - (-8 + 7.6 x_i)]$$

$$\frac{\partial J}{\partial b_1} = \frac{1}{4}\sum_{i=1}^{4} -x_i * [y_i - (-8 + 7.6 x_i)]$$

$$\frac{\partial J}{\partial b_0} = -2$$

$$\frac{\partial J}{\partial b_1} = -7$$

SIXTH (PUTTING THE GRADIENTS)

b_{0new} = -8 − Learning Rate × -2

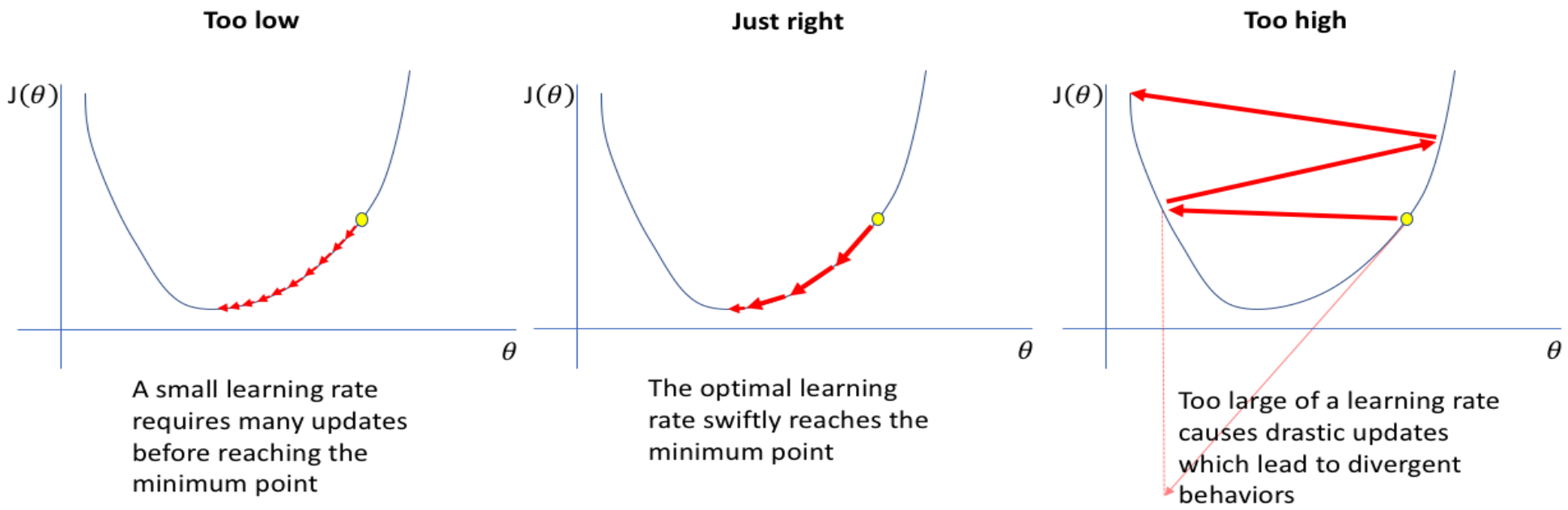
b_{1new} = 7.6 − Learning Rate × -7

LEARNING RATE (LR)

- The learning rate is a hyperparameter in optimization algorithms, such as gradient descent, **that determines the step size taken** in the direction of the negative gradient during each iteration.

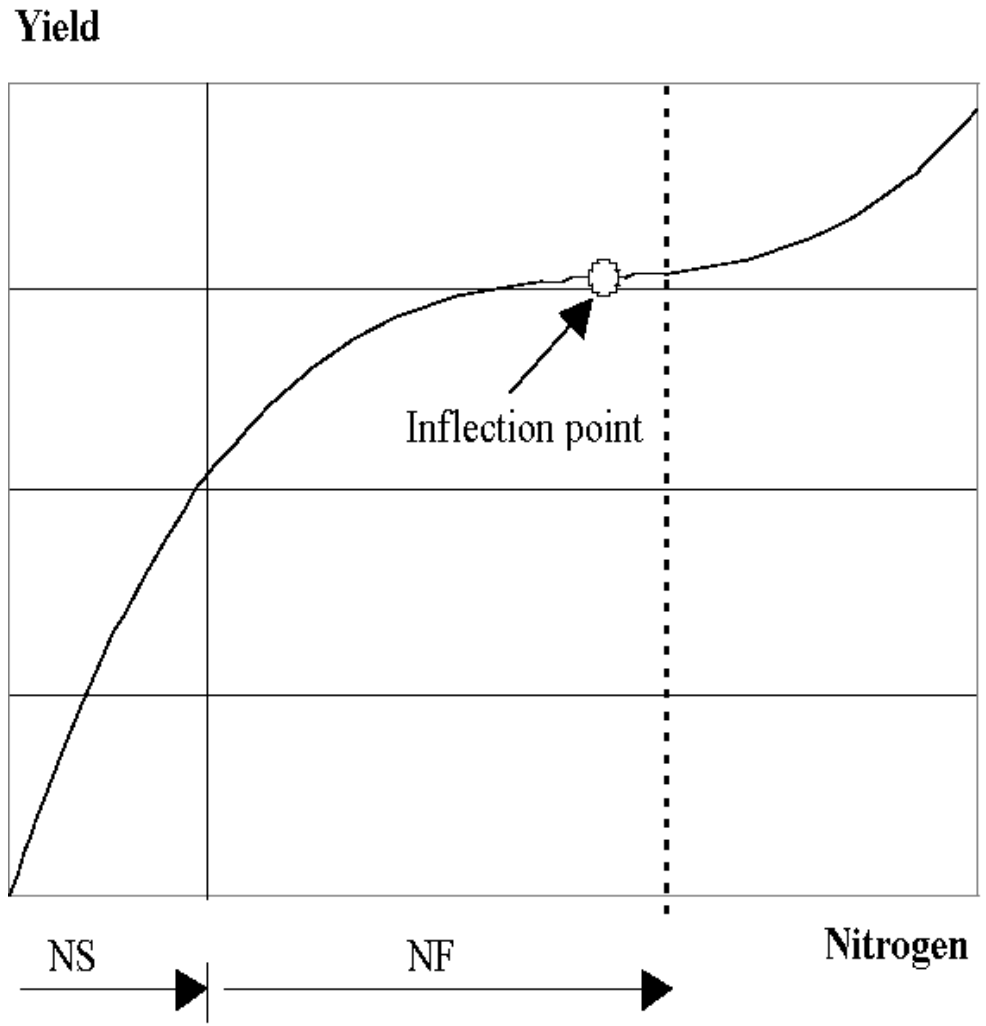
- It plays a crucial role in **CONTROLLING THE CONVERGENCE** and behavior of optimization algorithms, influencing how quickly or slowly the algorithm converges to the optimal solution.

LEARNING RATE (CONT.)

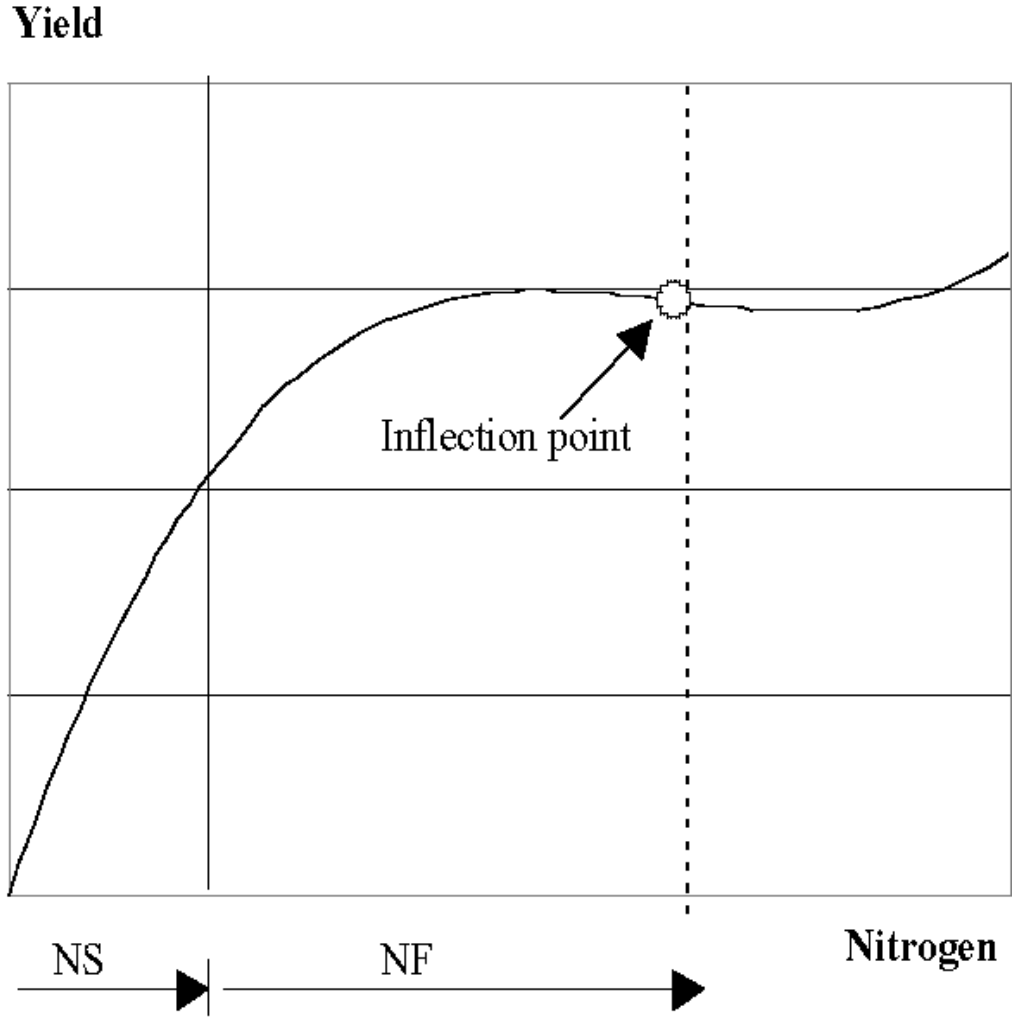


1. **Manual Tuning:** You can manually experiment with different learning rate values to find the one that works best for your specific problem. This approach might require trial and error.
2. **Learning Rate Scheduling:** Gradually reducing the learning rate during training can be helpful. For example, starting with a higher learning rate and gradually decreasing it as the optimization progresses can help achieve both fast initial convergence and stable fine-tuning. **Reduce on Plateau**

HOW TO FIND OPTIMAL VALUE FOR THE LEARNING RATE ? (DEEP LEARNING)

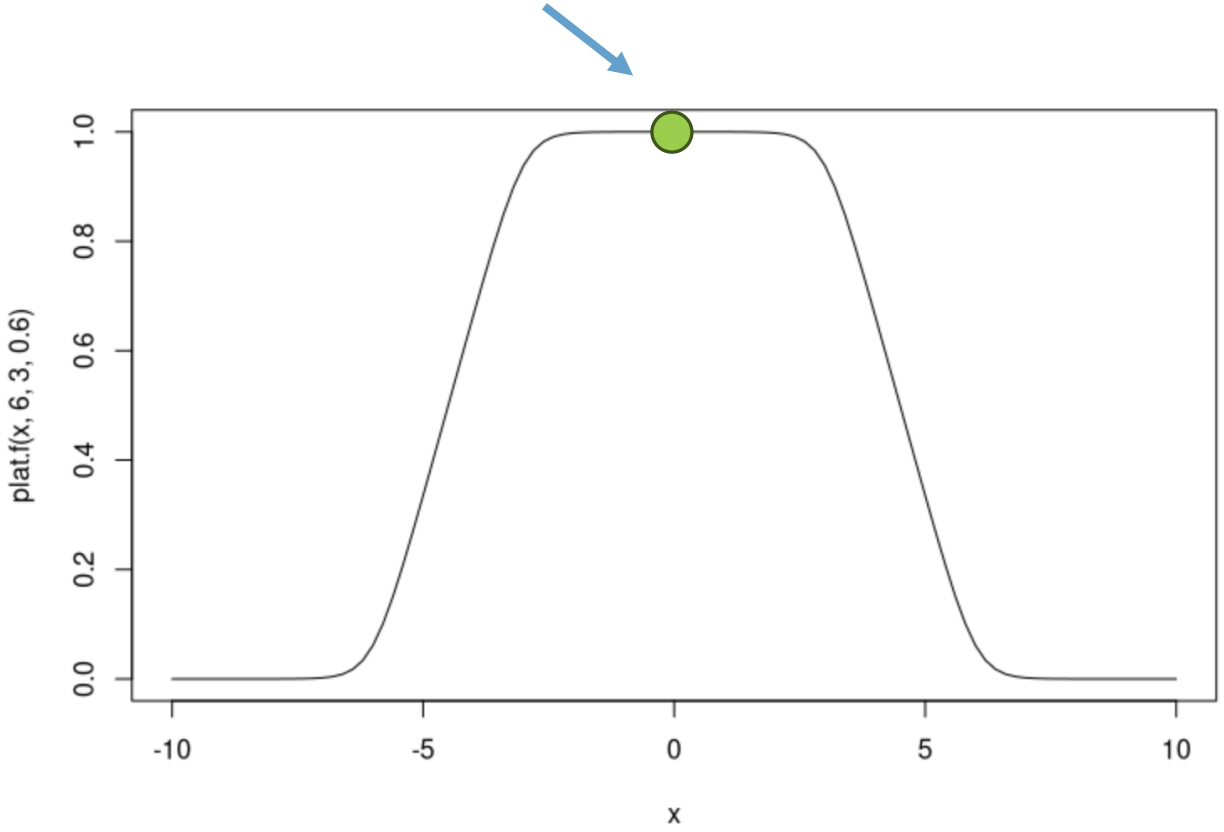


Panel A
Positive Slope at Inflection Point



Panel B
Negative Slope at Inflection Point

PLATEAU PROBLEM



3. **Adaptive Learning Rate Methods:** Algorithms like Adagrad, RMSProp, and Adam adjust the learning rate dynamically based on the history of gradients and updates. These methods adapt the learning rate for each parameter individually, which can lead to more balanced convergence.
4. **Learning Rate Range Test:** You can perform a learning rate range test by starting with a small learning rate, gradually increasing it, and monitoring the behavior of the cost function. This can help identify a suitable learning rate range.

HOW TO FIND OPTIMAL VALUE FOR THE LEARNING RATE ? (DEEP LEARNING)

SEVENTH (LEARNING RATE)

$b_{0new} = -8 - 0.01 \times -2$

$b_{1new} = 7.6 - 0.01 \times -7$

$b_{0new} = -7.98$

$b_{1new} = 7.68$

Usually the value in range [0.001,0.1]

EIGHTH (REPEAT UNTIL STOPPING CRITERIA IS MET)

$b_{0new} = -7.98$

$b_{1new} = 7.68$

You can confirm that by using MSE

$$\frac{1}{2n}\sum_{i=1}^{n}(y_i - h_i)^2$$

If it is still high, then we should repeat

Otherwise, the function is converged

THIS IS CALLED UNIVARIATE LINEAR REGRESSION (ONE FEATURE)

WHAT ABOUT TWO OR MORE FEATURES ?

Multivariate Linear Regression

LOOK AT THE DIFFERENT

- In Univariate Linear Regression, The relationship between the independent and dependent variables is modeled as a linear equation of the form $y = b_0 + b_1 x$
- In Multivariate Linear Regression, The general form of the equation is $y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$

For one feature, we had 2 B (that is m and x), but for n features we had n + 1 B

THE MAIN GRADIENT

$$J(b) = \frac{1}{2n}\sum_{i=0}^{n}[y_i - (bx)]^2$$ → b and x are now vectors

$$\frac{\partial J}{\partial b_k} = \frac{1}{n}\sum_{i=0}^{n} -x_k^i * [y_i - (b_0x_0 + b_1x_1 + b_2x_2 + \ldots + b_kx_k)]$$

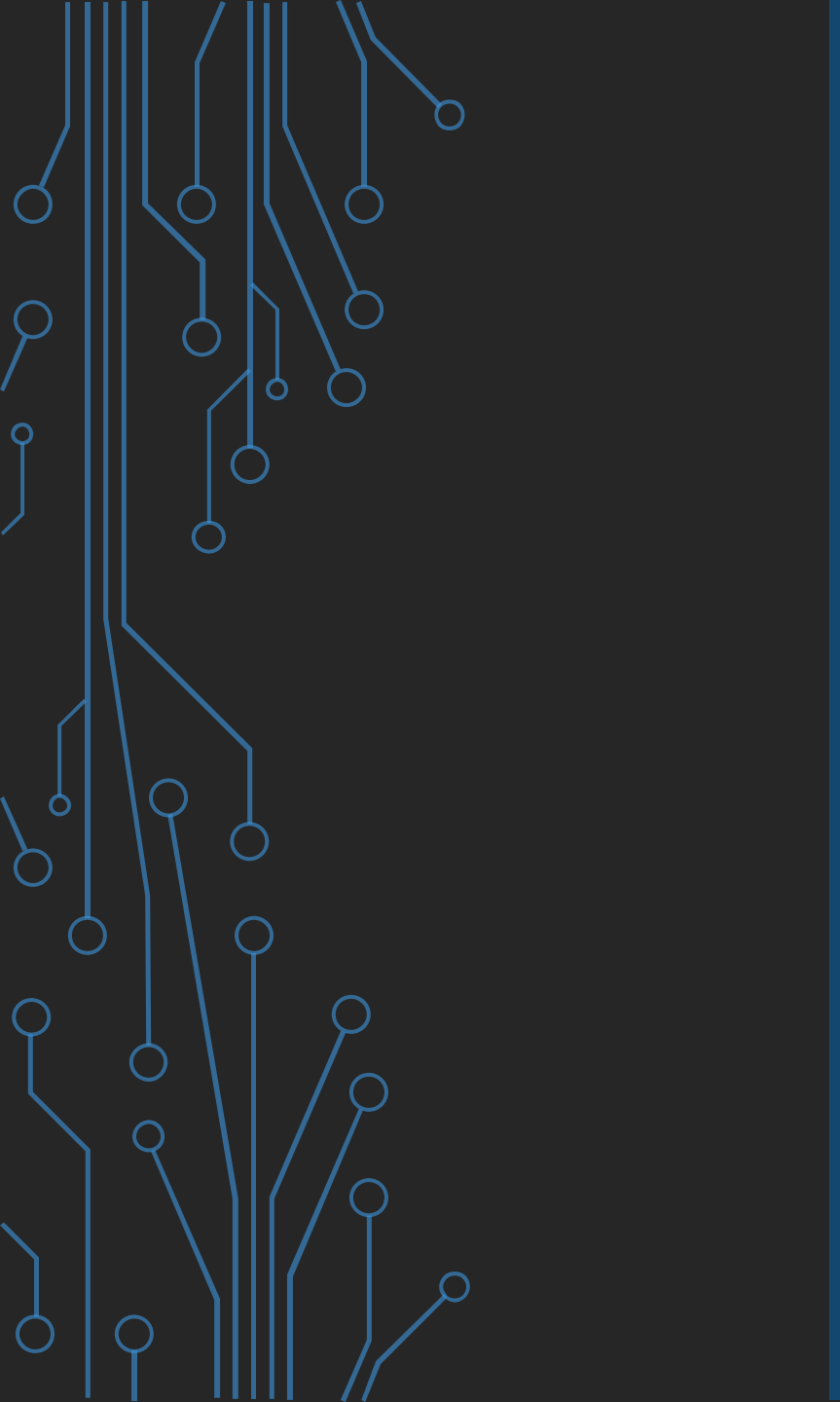
$b_{k\ new} = b_{k\ old} - \text{Learning Rate} \times \nabla j$ → Gradient Vector

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \mathbf{f}}{\partial x_1} & \cdots & \dfrac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla^{\mathsf{T}} f_1 \\ \vdots \\ \nabla^{\mathsf{T}} f_m \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{bmatrix}.$$

GRADIENT VECTOR

TIME FOR PRACTICALITY (15 MINUTES)

BREAK (10 MINUTES)



RIDGE REGRESSION

WHAT IS RIDGE REGRESSION ? (L2 REGULARIZATION)

- **Ridge regression**, is a linear regression technique that introduces a **REGULARIZATION TERM** to the traditional linear regression cost function.
- The purpose of ridge regression is to mitigate issues like **multicollinearity** (high correlation between predictor variables) and overfitting by adding a penalty term to the coefficients of the linear regression model.

Regularization is used to prevent overfitting

RIDGE REGRESSION

In ridge regression, a regularization term is added to the cost function, which is proportional to the sum of the squared values of the coefficients θ. The purpose of this regularization term is to impose a penalty on large coefficient values, encouraging the model to choose smaller coefficient values. The ridge regression cost function becomes:

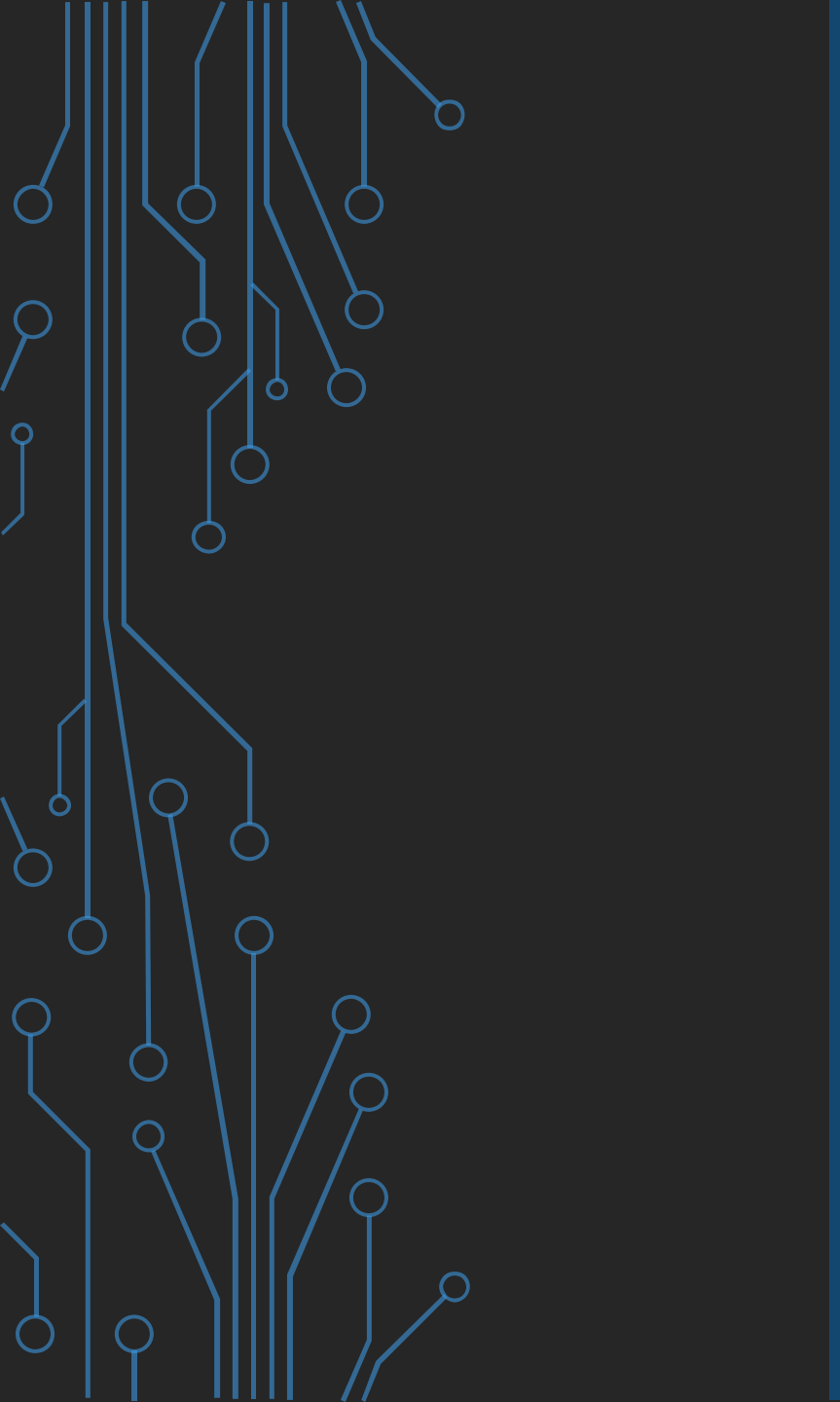
$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2$$

Where:

- λ is the regularization parameter that controls the strength of regularization. A higher λ leads to stronger regularization.
- n is the number of features.

SOLVING THE MULTICOLLINEARITY

- The regularization term penalizes large coefficients. Ridge regression effectively shrinks the coefficients toward zero while still allowing them to contribute to the model.
- This can help in reducing overfitting, improving model generalization, and handling multicollinearity by distributing the impact of correlated features across multiple features.



LASSO REGRESSION

WHAT IS LASSO REGRESSION ? (L1 REGULARIZATION)

- **Lasso regression**, short for **"Least Absolute Shrinkage and Selection Operator"** regression, is another form of regularized linear regression that introduces a regularization term to the traditional linear regression cost function.

- Like ridge regression, lasso regression is designed to address issues such as multicollinearity and overfitting by adding a penalty term to the coefficients of the linear regression model.

LASSO REGRESSION

In lasso regression, the cost function is modified to include a regularization term that is proportional to the absolute sum of the coefficients θ. The lasso regression cost function is given by:

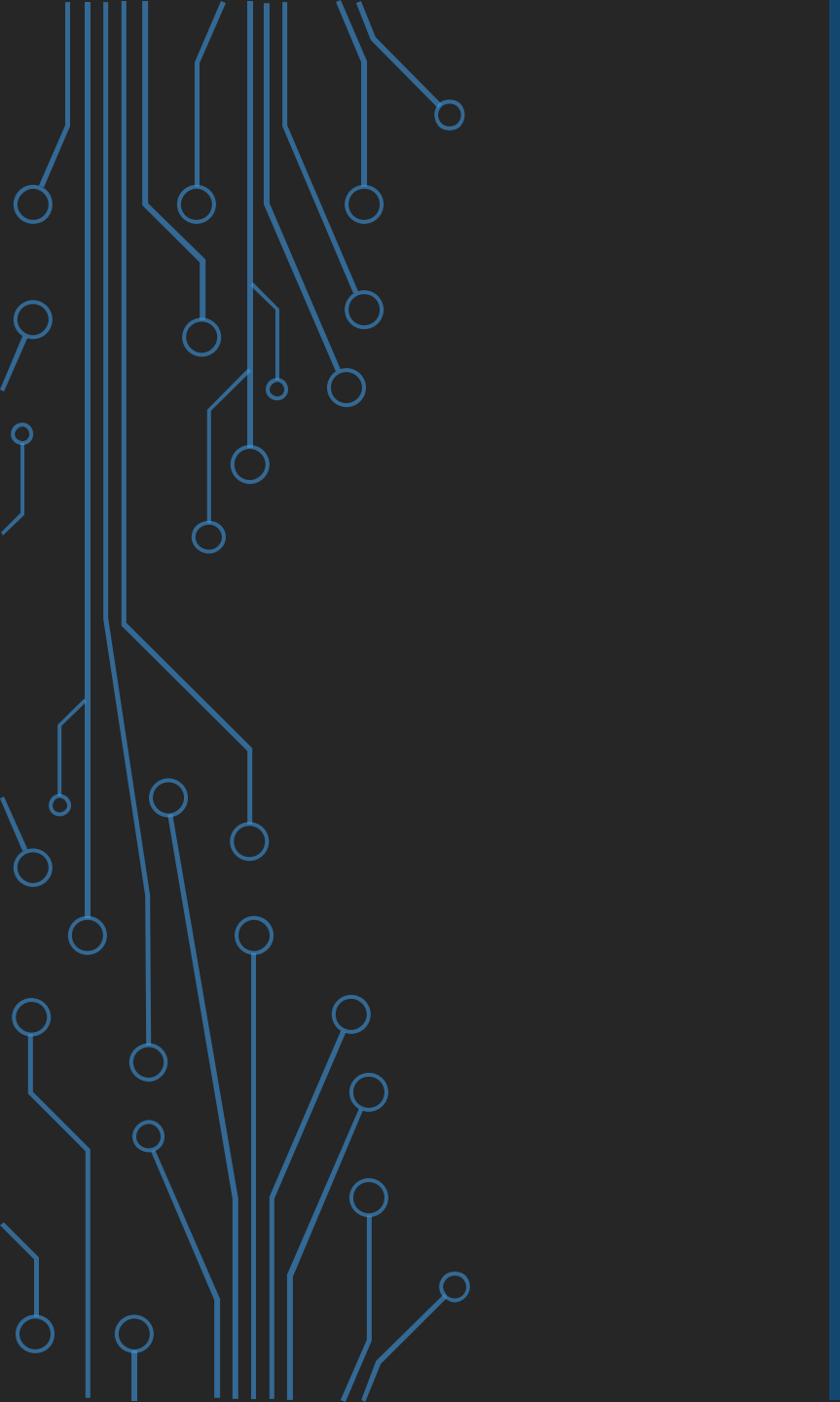
$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} |\theta_j|$$

Where:

- $J(\theta)$ is the cost function.
- m is the number of training examples.
- $h_\theta(x^{(i)})$ is the predicted value for the ith example using the linear regression model with coefficients θ.
- $y^{(i)}$ is the actual target value for the ith example.
- λ is the regularization parameter that controls the strength of regularization. A higher λ leads to stronger regularization.
- n is the number of features.

WHAT DOES LASSO REGRESSION SOLVE ?

- The lasso regression encourages sparsity in the coefficient values, meaning that some coefficients are driven to exactly zero.
- This has the effect of performing feature selection, as features with non-zero coefficients are considered more important by the model.
- Lasso regression is particularly useful when dealing with high-dimensional datasets *where many features may not be relevant to the target variable*, By shrinking some coefficients to zero and preventing overfitting.



LOGISTIC REGRESSION

WHAT IS LOGISTIC REGRESSION ?

- Logistic regression is a statistical model used for **BINARY *CLASSIFICATION* TASKS**, where the goal is to predict the probability that an input example belongs to a particular class (usually represented as 0 or 1).

- Despite its name, logistic regression is a classification algorithm, not a regression algorithm.

It depends on the probability

STORYTELLING

What is the probability of a record to being belong to class 1 ? $P(y \mid x)$

- The logistic regression using probability with regression to make a classification.
- It is called **ODD Function,** such that whether a class being belong to the positive class or negative class.

Linear Combination: Similar to linear regression, logistic regression also involves a linear combination of input features and model coefficients, which is often denoted as z:

$$z = b_0 + b_1x_1 + b_2x_2 + \ldots + b_nx_n$$

where $b_0, b_1, b_2, \ldots, b_n$ are the model coefficients, and $x_1, x_2, \ldots, x_n$ are the input features.

USING THE REGRESSION AND PROBABILITY

USING THE REGRESSION AND PROBABILITY (CONT.)

Linear Regression: $z = b_0+b_1x_1+b_2x_2+\ldots+b_nx_n$

ODD Function (logit): $\log_e\left(\frac{p_i}{1-p_i}\right) = z$ Map linearity to [0,1] for probability

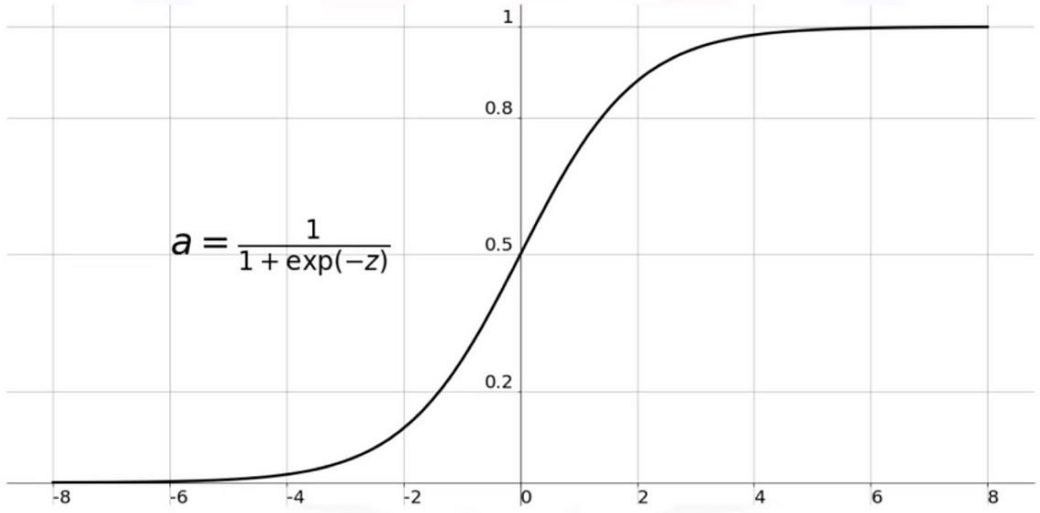
$$\log_e\left(\frac{p_i}{1-p_i}\right) = z \rightarrow \frac{p_i}{1-p_i} = e^z$$

$$\frac{p_i}{1-p_i} = e^z \rightarrow p_i = (1-p_i)e^z \rightarrow p_i + p_ie^z = e^z \rightarrow p_i(1+e^z) = e^z$$

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

SIGMOID (LOGISTIC) FUNCTION

Sigmoid Function



3. **Prediction:** The output of the sigmoid function $\sigma(z)$ represents the probability that the input example belongs to class 1. The probability of class 0 is $1 - \sigma(z)$. Mathematically, this can be written as:

   $P(y = 1 \mid x) = \sigma(z)$

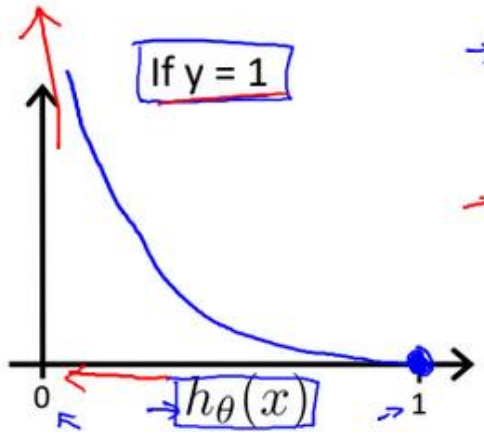
   $P(y = 0 \mid x) = 1 - \sigma(z)$

4. **Decision Boundary:** To make a classification decision, a threshold (usually 0.5) is applied to the predicted probability. If $P(y = 1 \mid x)$ is greater than or equal to 0.5, the example is classified as class 1; otherwise, it's classified as class 0.

USING THE REGRESSION AND PROBABILITY (CONT.)

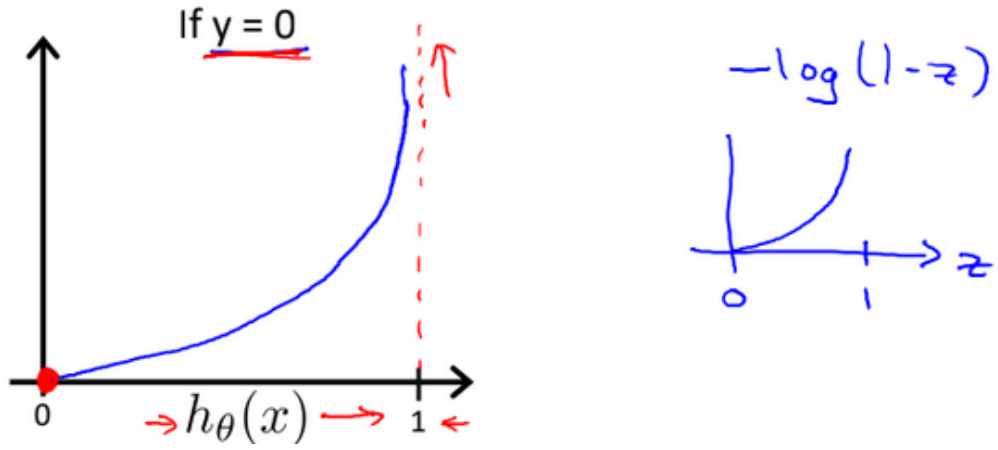
COST FUNCTION (LOG LOSS)

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



- Cost = 0 if $y = 1, h_\theta(x) = 1$
  - But as $h_\theta(x) \to 0$
    $Cost \to \infty$
- Captures intuition that if $h_\theta(x) = 0$, (predict $P(y = 1|x; \theta) = 0$), but $y = 1$, we'll penalize learning algorithm by a very large cost.

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



COST FUNCTION (LOG LOSS) – CONT.

GRADIENT DESCENT WITH LOG LOSS

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

TYPES OF LOGISTIC REGRESSION

Binomial: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

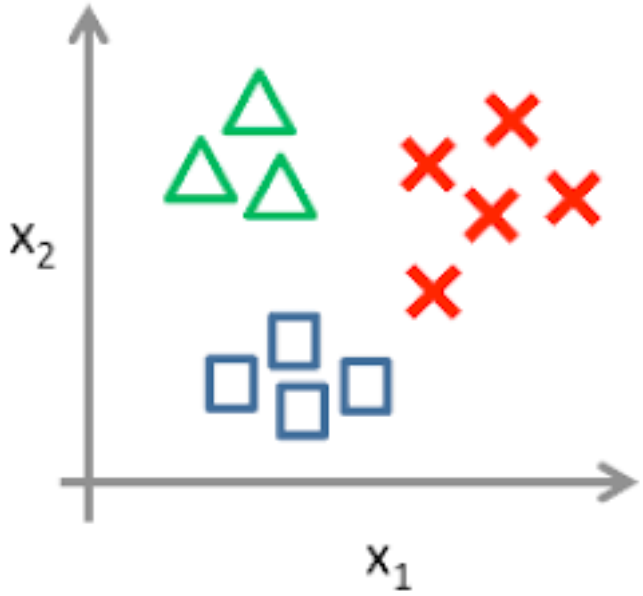
Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep".

Ordinal: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

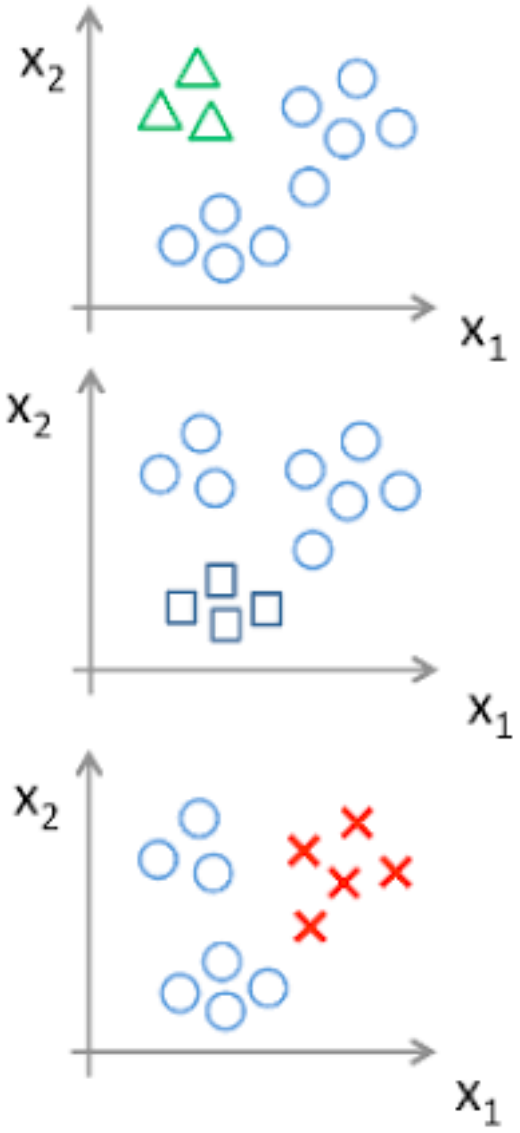
WHAT ABOUT MULTICLASSIFICATION ?

- One-Versus-All Technique
- One-Versus-Rest Technique
- They are all the same

One-vs-all (one-vs-rest):



Class 1: Green
Class 2: Blue
Class 3: Red



ONE-VERSUS-REST (OVR)

- At **OvR**, each class is taken against the rest
- The probability for that class will be computed using sigmoid
- The same thing will be applied for the remaining classes
- The highest probability for a specific class means that the record belongs to that class

This technique require one-hot encoding to the output !

TIME FOR PRACTICALITY (5 MINUTES)

QUESTIONS


THANK YOU