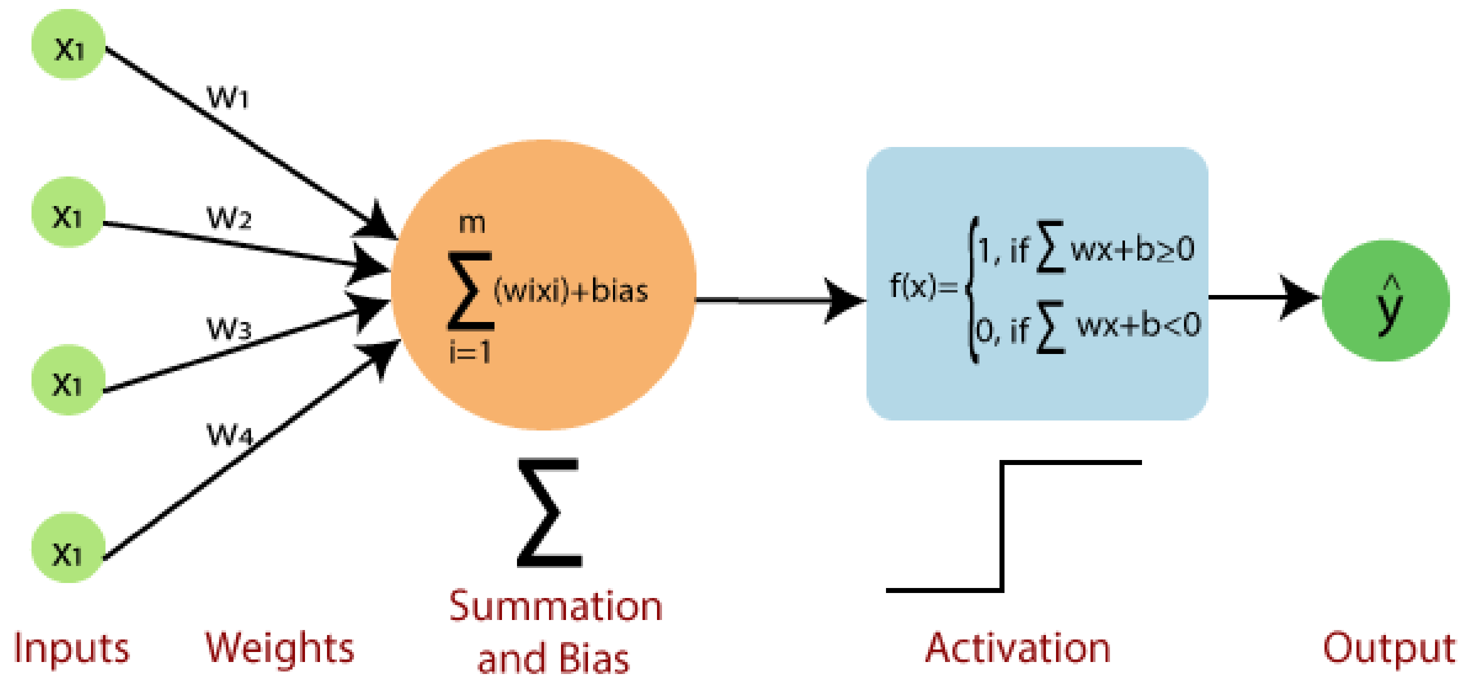
# Deep Learning : Perceptron

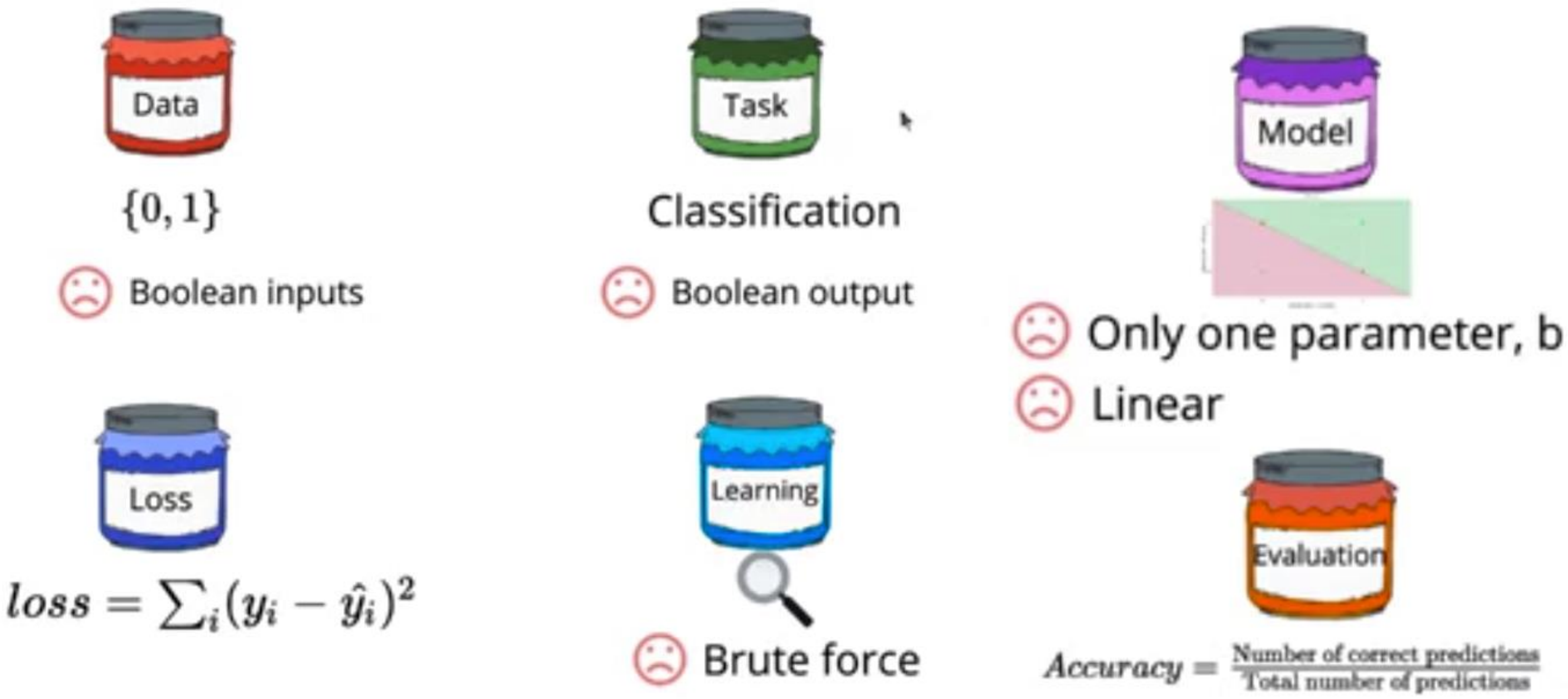
राष्ट्रीय प्रौद्योगिकी संस्थान सिक्किम
NATIONAL INSTITUTE OF TECHNOLOGY SIKKIM

**Course Instructor:**
Dr. Bam Bahadur Sinha
*Assistant Professor*
*Computer Science & Engineering*
*National Institute of Technology*
*Sikkim*

# Perceptron


x1
x1
x1
x1
w1
w2
w3
w4
m
∑ (wixi)+bias
i=1
f(x)= 1, if ∑ wx+b≥0
0, if ∑ wx+b<0
ŷ
∑
Inputs
Weights
Summation
and Bias
Activation
Output

# MP- Neuron (Recap)



# Perceptron

Data
$\{0, 1\}$
Real inputs
Task
Classification
Boolean output
Model
Weights for every input
Linear
Loss
$\cancel{loss = \sum_i (y_i - \hat{y}_i)^2}$
$loss = \sum_i max(0, 1 - y_i * \hat{y}_i)$
Learning
Our 1st learning algorithm
Evaluation
$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$

# Data
# (Real Inputs)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Launch (within 6 months)** | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| **Weight (g)** | 151 | 180 | 160 | 205 | 162 | 182 | 138 | 185 | 170 |
| **Screen size (inches)** | 5.8 | 6.18 | 5.84 | 6.2 | 5.9 | 6.26 | 4.7 | 6.41 | 5.5 |
| **dual sim** | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **Internal memory (>= 64 GB, 4GB RAM)** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **NFC** | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| **Radio** | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| **Battery(mAh)** | 3060 | 3500 | 3060 | 5000 | 3000 | 4000 | 1960 | 3700 | 3260 |
| **Price (INR)** | 15k | 32k | 25k | 18k | 14k | 12k | 35k | 42k | 44k |
| **Like ($y$)** | **1** | **0** | **1** | **0** | **1** | **1** | **0** | **1** | **0** |

Each Feature will have different ranges.

**Data Preparation:** Standardize the input

Feature Scaling

- Normalization
- Standardization

# Normalization & Standardization

- Normalization or Min-Max Scaling is used to transform features to be on a similar scale.
- It transforms data to fit within a specific range, typically [0, 1] or sometimes [-1, 1]. The formula for normalization is:

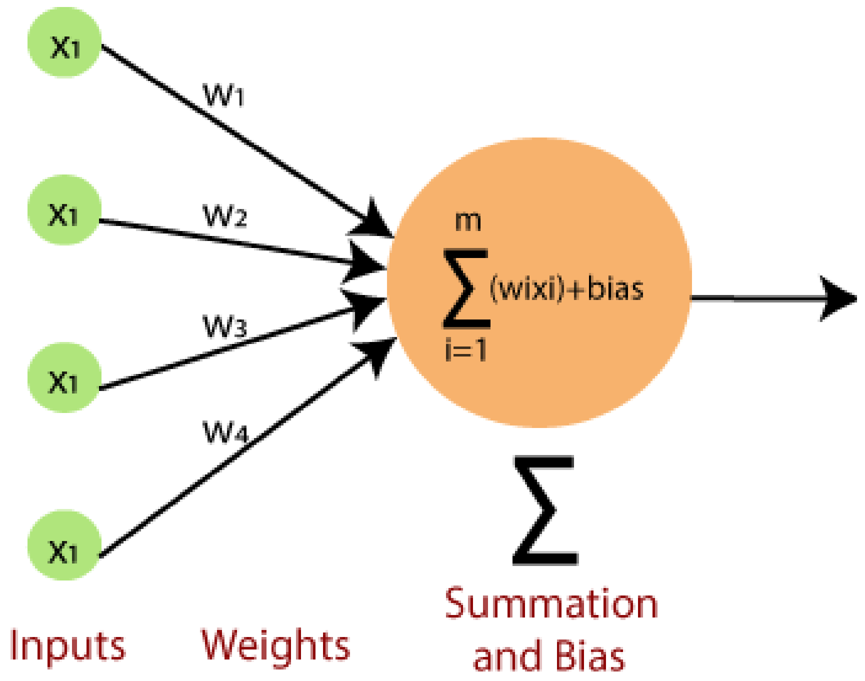
$$X\_new = (X - X\_min) / (X\_max - X\_min)$$

- Standardization can be helpful in cases where the data follows a Gaussian distribution. However, this does not have to be necessarily true. Geometrically speaking, it translates the data to the mean vector of original data to the origin and squishes or expands the points if std is 1 respectively.

$$\text{New value} = (x - \mu) / \sigma$$

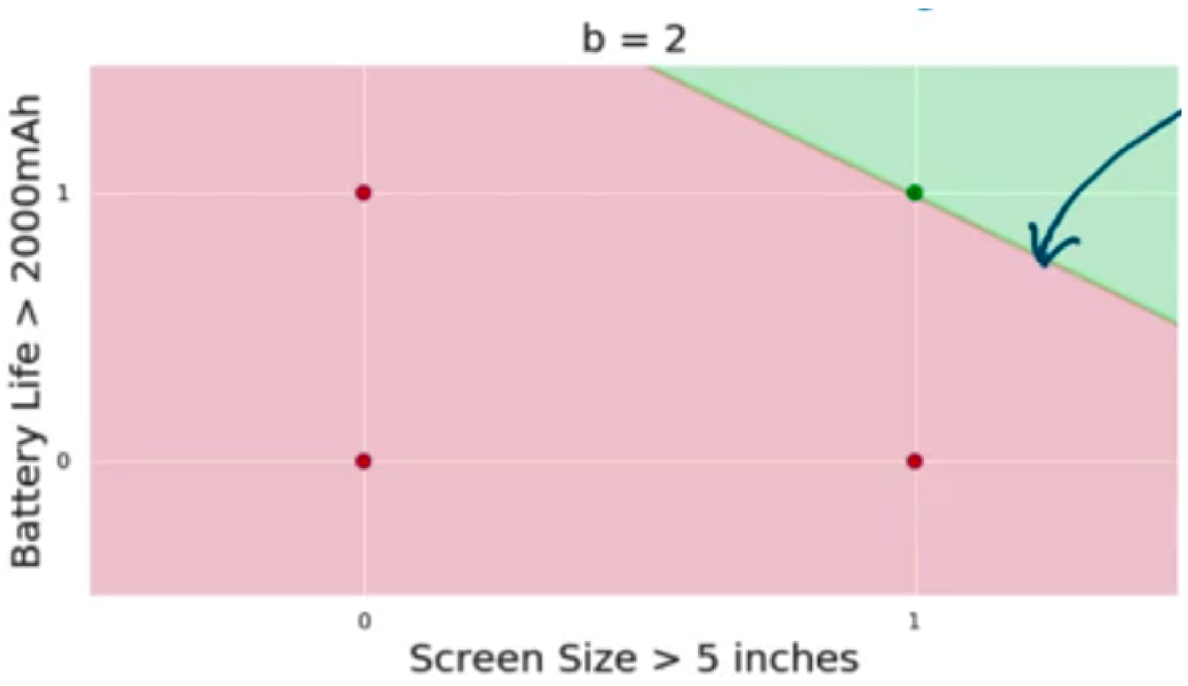
(This is also termed as Z-Score Normalization)

# Model (Binary Classification Task)

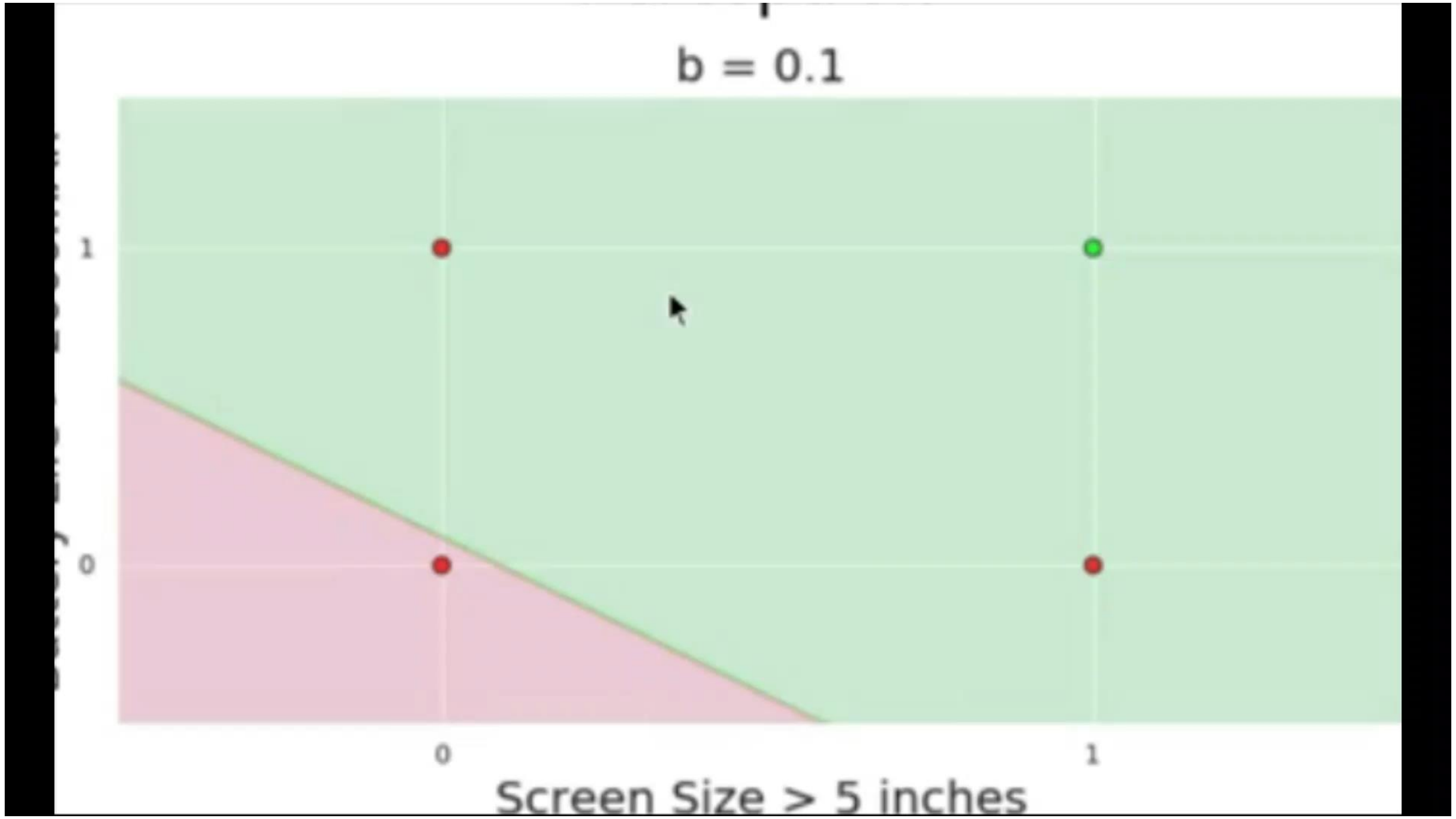



Why do we need weights?

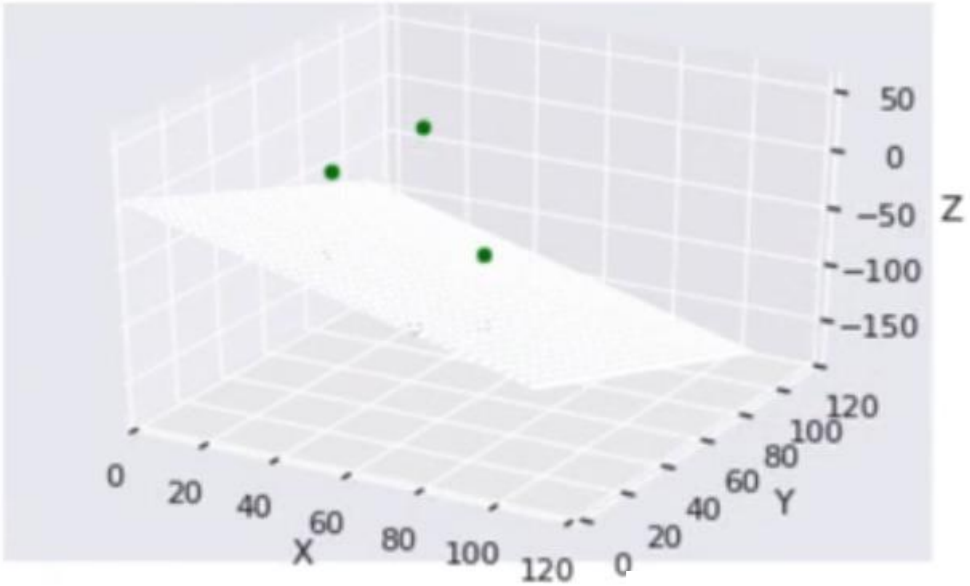
# Geometric Representation of MP-Neuron

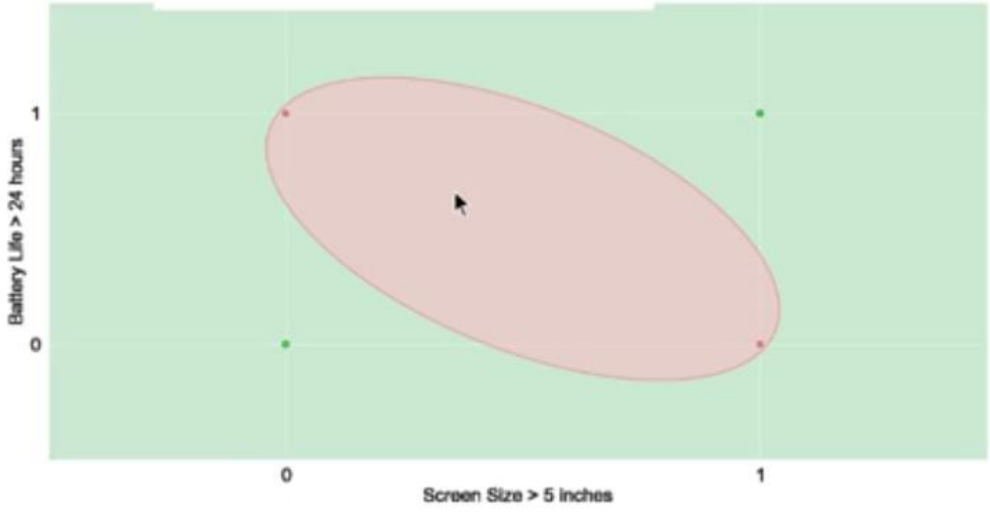


# Geometric Representation of Perceptron



# Do we need more freedom for modelling the input?




We want even more freedom

# Loss Function

*i.)* Perceptron Loss
*ii.)* Squared Error Loss

| Weight | Screen Size | Like (y) | ŷ | Perceptron Loss | Squared Error Loss |
|---|---|---|---|---|---|
| 0.19 | 0.64 | 1 | 1 | 0 | 0 |
| 0.63 | 0.87 | 1 | 0 | 1 | 1 |
| 0.33 | 0.67 | 0 | 1 | 1 | 1 |
| 1 | 0.88 | 0 | 0 | 0 | 0 |

$$L = 0, \text{ if } y = \hat{y}$$
$$\quad = 1, \textit{otherwise}$$

When Output is Boolean; squared error loss is same as perceptron loss.

$$L = \mathbf{1}_{(y \neq \hat{y})}$$

Squared Error loss = $(y - \hat{y})^2$