BITS Pilani
Pilani Campus

Applied Machine Learning

Dr. Harikrishnan N B
Computer Science and Information Systems

innovate achieve lead

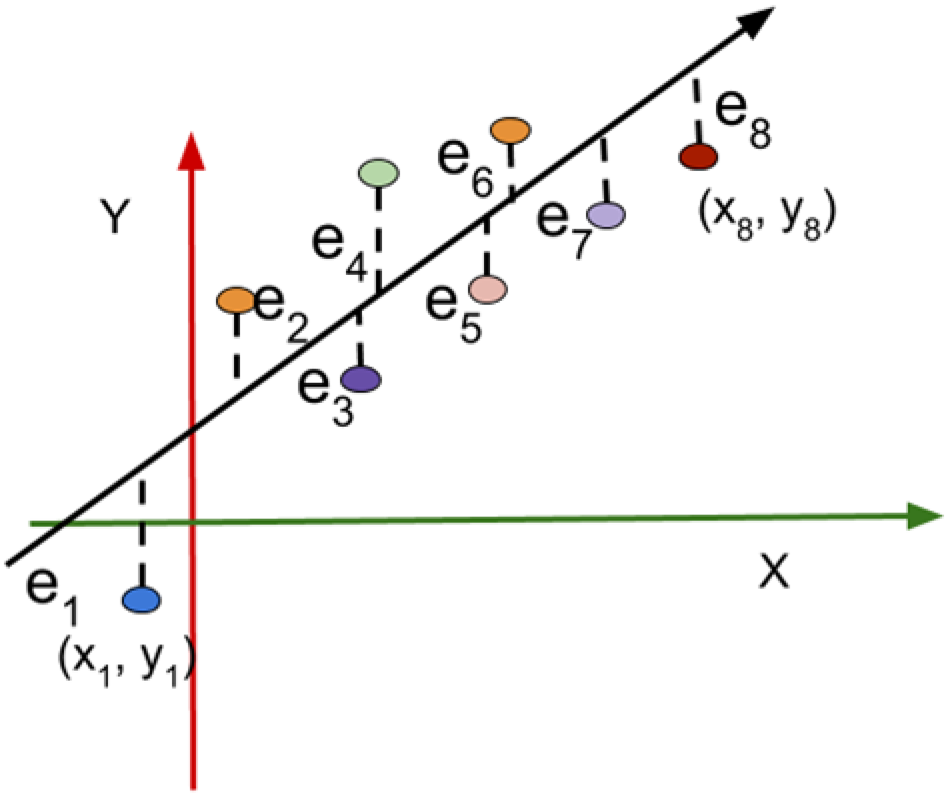
BITS Pilani
Pilani Campus

SE ZG568 / SS ZG568,
Applied Machine Learning
Lecture No. 11 [13 April 2025]

Linear Regression using Optimization

Logistic Regression

Linear Least Square Regression



$$y_1 = mx_1 + c + e_1$$
$$y_2 = mx_2 + c + e_2$$
$$y_3 = mx_3 + c + e_3$$
$$y_4 = mx_4 + c + e_4$$
$$y_5 = mx_5 + c + e_5$$
$$y_6 = mx_6 + c + e_6$$
$$y_7 = mx_7 + c + e_7$$
$$y_8 = mx_8 + c + e_8$$

Quick Recap

Linear regression models the relationship between the input x and the output y as a linear combination of the features:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

feature

where:

- $x = [1, x_1, x_2, ..., x_n]^T$ (including $x_0 = 1$ for bias)
- $\theta = [\theta_0, \theta_1, ..., \theta_n]^T$ are the parameters (weights)
- $h_\theta(x)$ is the predicted output.

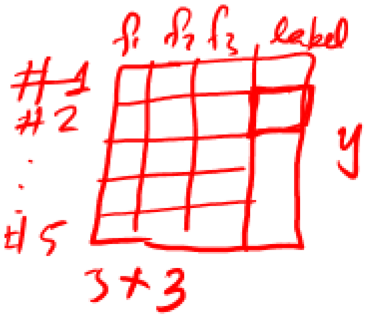
In vector form:

$$\boxed{h_\theta(x) = \theta^T x}$$

To measure how well our model predicts, we use the **Mean Squared Error (MSE)**:

$y^{(i)}$

$i:2$

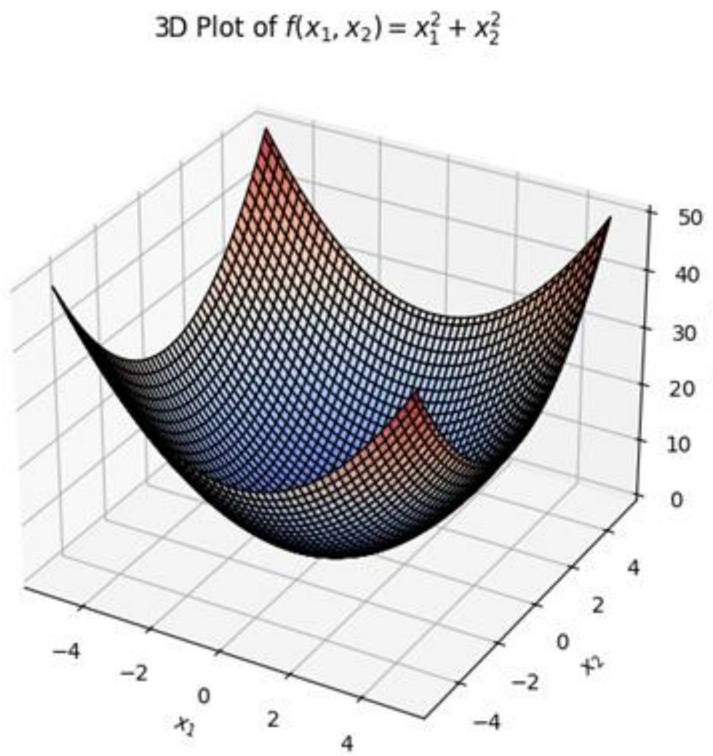


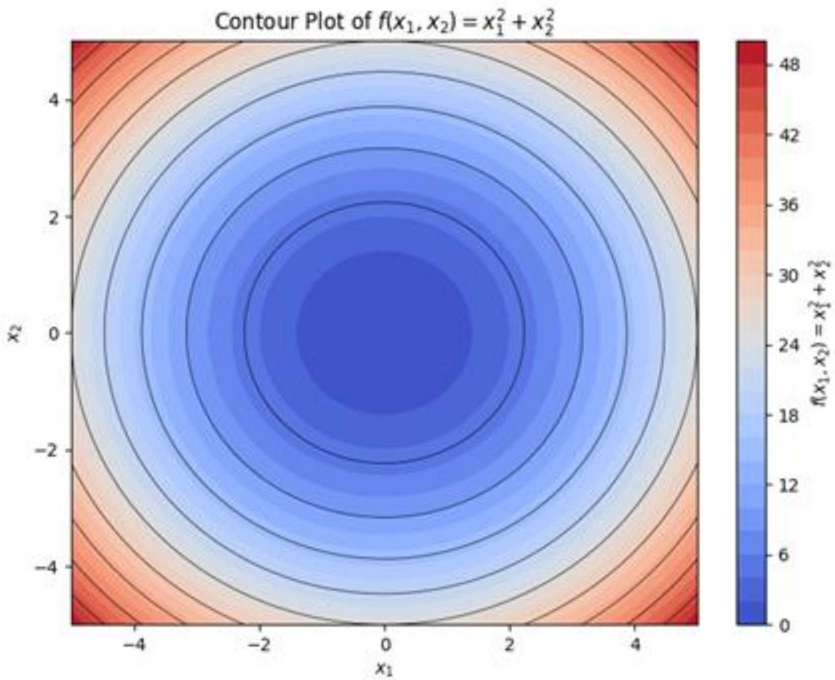
$y^{(2)}$

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

where:

- m is the number of training examples
- $y^{(i)}$ is the actual output for the i-th training example
- $h_\theta(x^{(i)})$ is the predicted output.
- The factor $\frac{1}{2}$ is used for mathematical convenience when differentiating.


3D Plot of $f(x_1, x_2) = x_1^2 + x_2^2$
50
40
30
20
10
0
−4
−2
0
2
4
x_1
4
2
0
−2
−4
x_2


Contour Plot of $f(x_1, x_2) = x_1^2 + x_2^2$
4
2
0
−2
−4
x_2
−4
−2
0
2
4
x_1
48
42
36
30
24
18
12
6
0
$f(x_1, x_2) = x_1^2 + x_2^2$

Gradient descent is used to minimize $J(\theta)$. The update rule is:

$$\boxed{\theta_j := \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}}$$

where:

- α is the **learning rate**, controlling step size
- $\frac{\partial J(\theta)}{\partial \theta_j}$ is the gradient (partial derivative)

Computing the gradient:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Thus, the update step becomes:

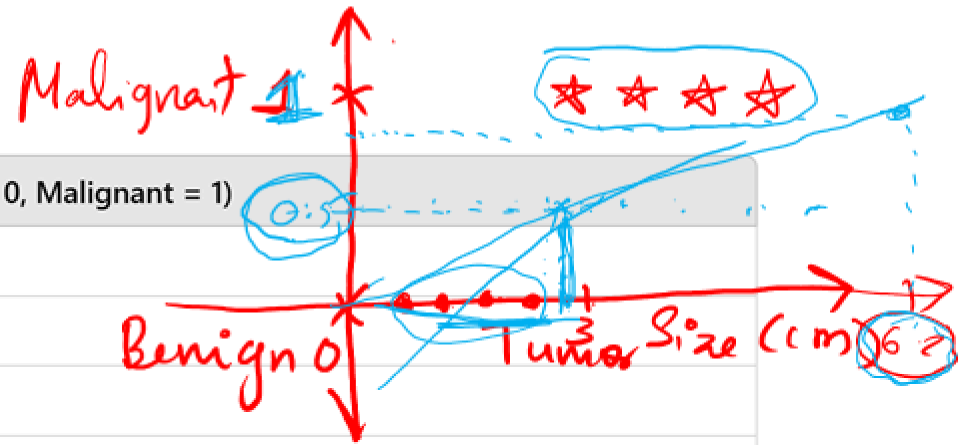
$$\boxed{\theta_j := \theta_j - \alpha \cdot \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}}$$

This process repeats until convergence (i.e., when $J(\theta)$ stops changing significantly).

Logistic Regression

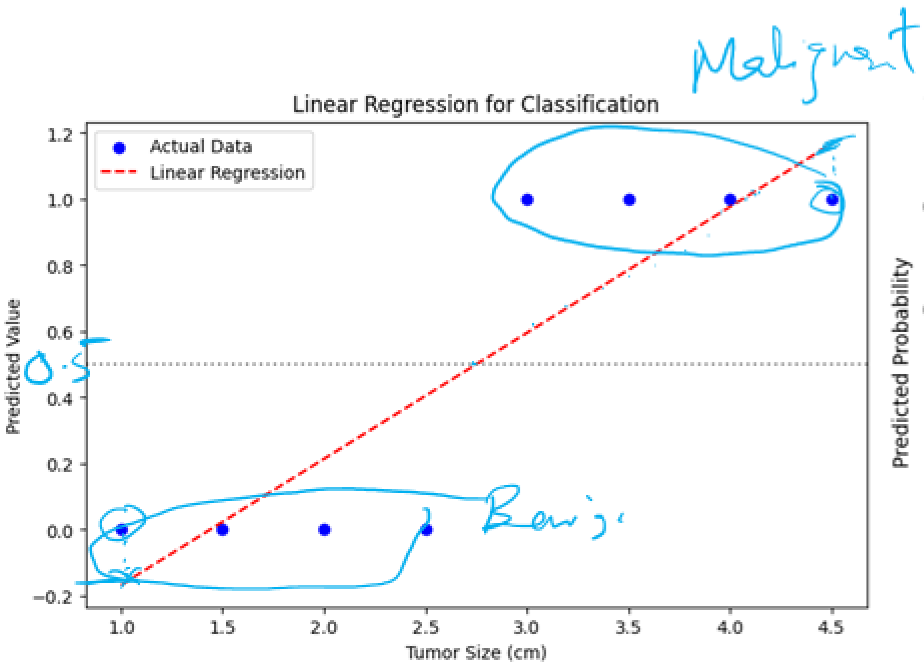
Scenario

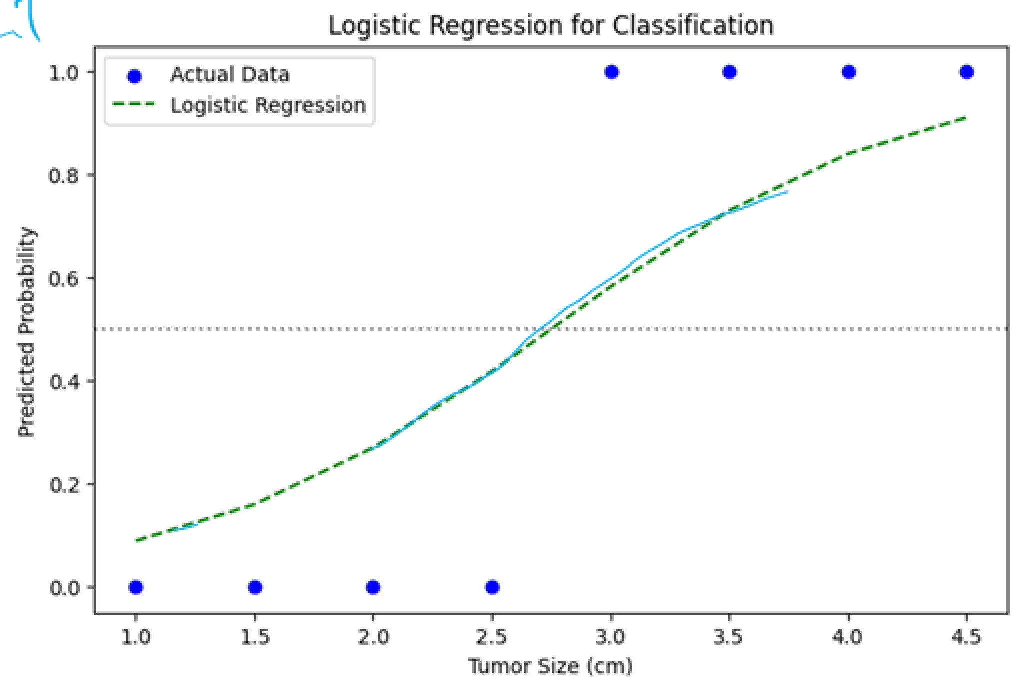
Doctors want to classify tumors as **benign (0) or malignant (1)** based on tumor size. Larger tumors are more likely to be malignant, but there's no hard cutoff—so we use logistic regression to model this probability.



Hypothetical Data

Tumor Size (cm)	Diagnosis (Benign = 0, Malignant = 1)
1.0	0 (Benign)
1.5	0
2.0	0
2.5	0
3.0	1
3.5	1
4.0	1
4.5	1


Malignant
Linear Regression for Classification
Actual Data
Linear Regression
Predicted Value
0.5
Benign
Tumor Size (cm)
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5


Logistic Regression for Classification
Actual Data
Logistic Regression
Predicted Probability
Tumor Size (cm)
1.0
0.8
0.6
0.4
0.2
0.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5

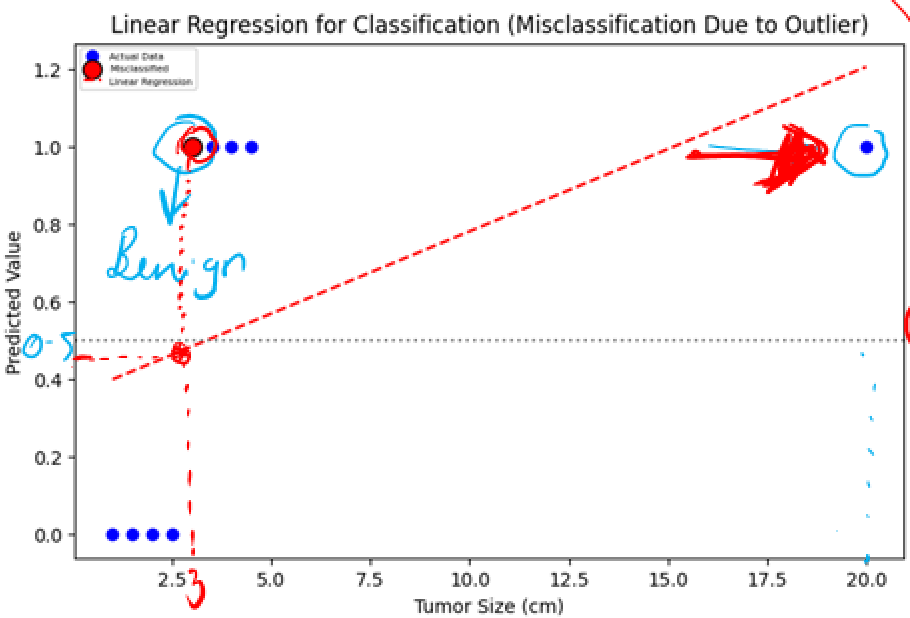
Effect of Outlier

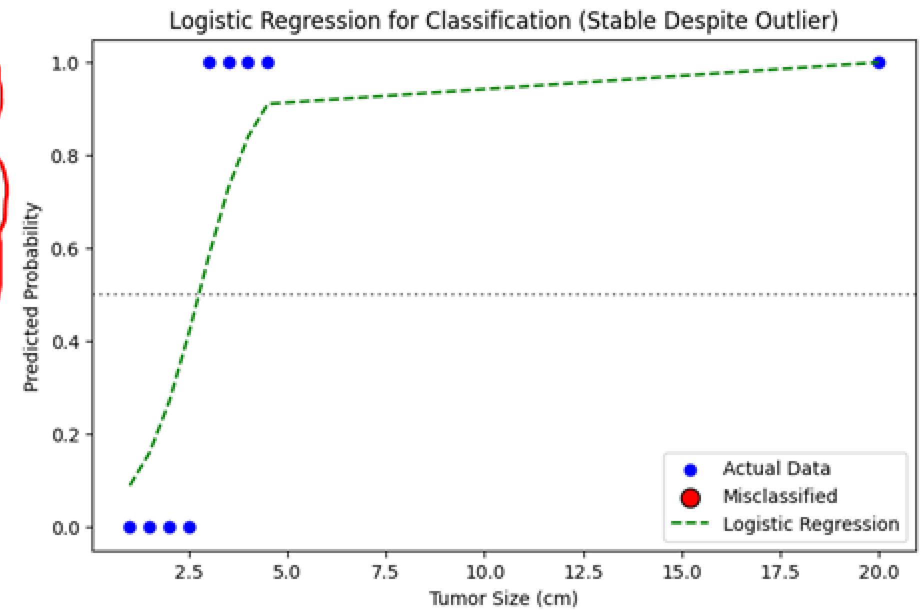
Tumor Size (cm)	Diagnosis (0 = Benign, 1 = Malignant)
1.0	0
1.5	0
2.0	0
2.5	0
3.0	1
3.5	1
4.0	1
4.5	1
20.0	1 (Outlier)

Effect of Outlier

Regularizer

$$\min_x \left(\|Ax - b\|_2^2 + \lambda \|Dx\|_2^2 \right)$$





Threshold = 0.5,

Accuracy = 0.89

1. Effect on Linear Regression

- Linear regression tries to fit a straight line, minimizing squared errors.
- A **large outlier (e.g., tumor size = 10 cm, malignant = 1)** will **pull the line upwards**, making it **steeper**.
- This **shifts the decision threshold** (where the prediction crosses 0.5), possibly classifying smaller tumors incorrectly.

2. Effect on Logistic Regression

- Logistic regression models **probabilities** using the **sigmoid function**, which is bounded between **0 and 1**.
- A large outlier **won't distort the S-curve** as much because it saturates towards 1.
- The decision boundary (where probability = 0.5) **remains relatively stable**.

Logistic Regression

$x^{(1)} = [x_1^{(1)} \; x_2^{(1)}]$

$\theta^T x^{(1)} = \theta_0 + \theta_1 x_1^{(1)} + \theta_2 x_2^{(1)}$

Goal: learn θ_0, θ_1 and θ_2

The hypothesis function in **logistic regression** is given by:

$$\boxed{h_\theta(x) = g(\theta^T x)}$$

function of $\theta^T x$

$h_\theta(x) = g(\theta^T x)$

and hypothesis function of linear regression

$h_\theta(x) = wx + b$

$[w \; b] \begin{bmatrix} x \\ 1 \end{bmatrix}$

$\theta^T \vec{x}$

where:

- θ is the parameter vector (weights).
- x is the feature vector (input data).
- $\theta^T x$ represents the **linear combination** of features and parameters.
- $g(z)$ is the **sigmoid function**, defined as:

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

Thus, expanding $h_\theta(x)$:

$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Dataset = f_1 f_2 y

$x^{(1)}$ 1 → row 1

$x^{(2)}$ 2 → row 2

$x^{(3)}$ ⋮

$x^{(m)}$ m → row m

$y = [0, 1, 0, 1, \ldots, 0]^T$

$m \times 2$

This function maps any real number to the range **(0,1)**, making it suitable for **probability estimation** in classification problems.

Notation

Superscript

$$\text{Data}\quad \begin{array}{c|cc} & f_1 & f_2 \\ \hline x^{(1)} & x_1^{(1)} & x_2^{(1)} \\ x^{(2)} & & \\ x^{(3)} & \vdots & \\ \vdots & & \\ x^{(m)} & x_1^{(m)} & x_2^{(m)} \end{array}$$

$$\theta^T x^{(i)} \Rightarrow \theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)}$$

where $i \in \{1, 2, \dots m\}$

Goal: learn $\theta_0, \theta_1, \theta_2$

$$h_\theta(x^{(i)}) = g(\underline{\theta^T x^{(i)}}) = \frac{1}{1 + e^{-\theta^T x^{(i)}}}$$

Is $\theta^T x^{(i)}$ is scalar or vector?

~~$x = 2$~~ $\quad z = 2$

$$x = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$x^{(i)} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}$$

$$\theta^T x^{(i)} = \begin{bmatrix} \theta_0 & \theta_1 & \theta_2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \underbrace{\theta_0 + \theta_1 2 + \theta_2 3}_{\text{scalar}}$$

$$\underbrace{\frac{1}{1+e^{-\theta^T x}}}_{y\text{-axis}} \quad \text{vs} \quad \underbrace{\theta^T x}_{x\text{-axis}}$$

Case I

What happens when

$$\boxed{\theta^T x \to \infty}$$

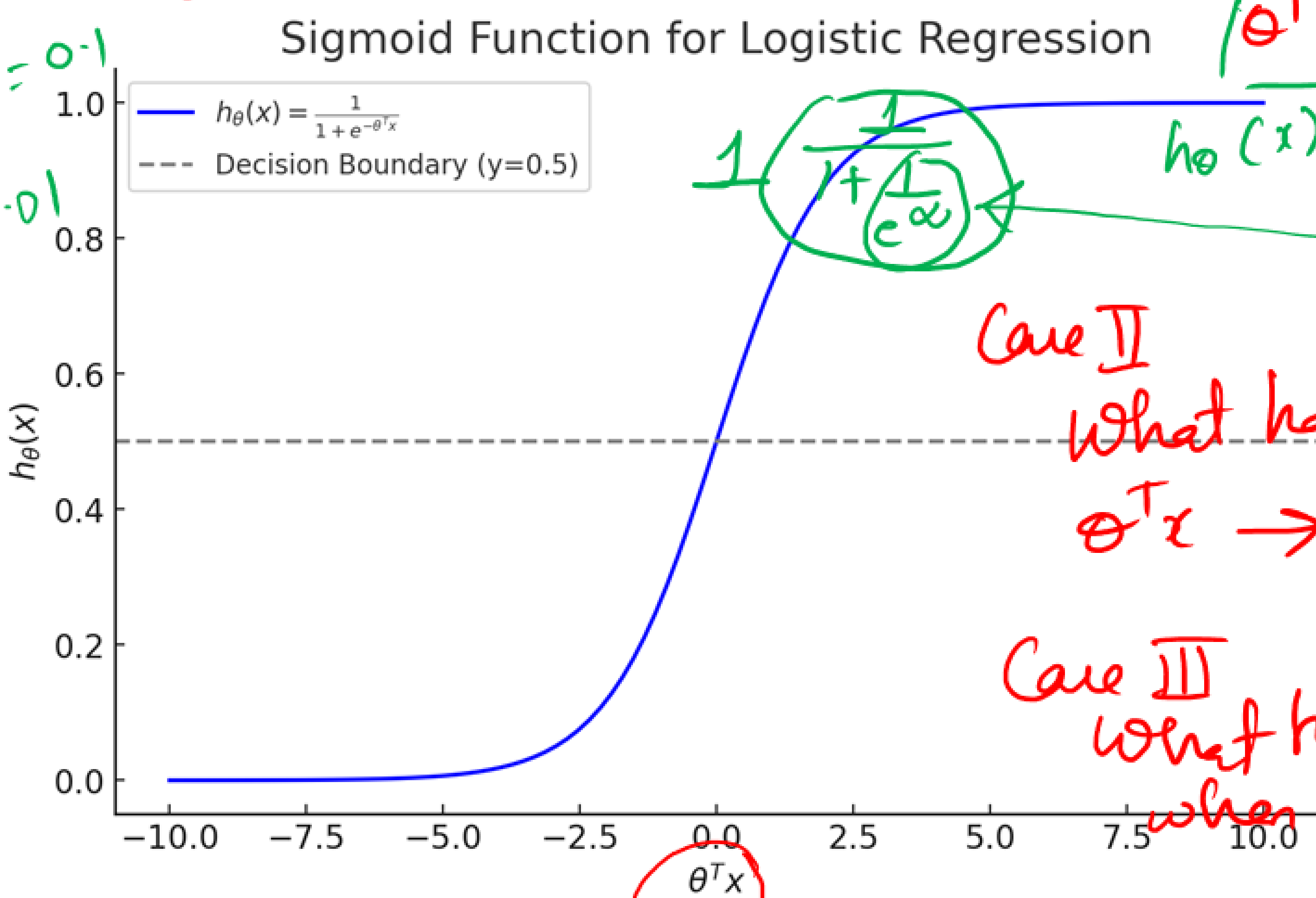
$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} = \frac{1}{1+e^{-\infty}}$$

$$1 \quad \frac{1}{1+\frac{1}{e^\infty}}$$

$\frac{1}{10} = 0.1$

$\frac{1}{100} = 0.01$

$\frac{1}{1000}$



Case II

What happens

$$\theta^T x \to -\infty$$

Case III

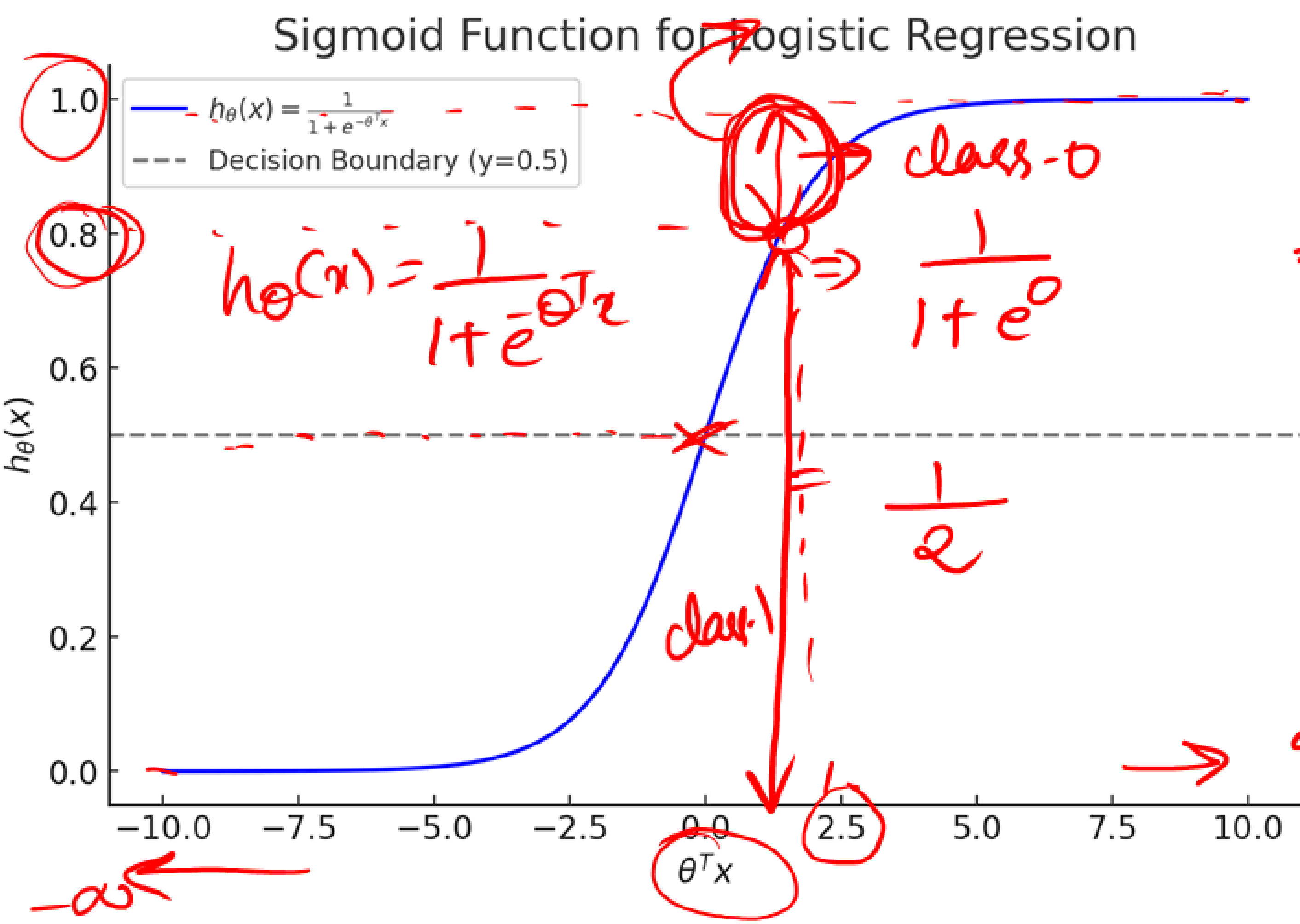
What happens when $\theta^T x = 0$

Case III

$\theta^T x = 0$

Case II $\theta^T x \to -\infty$

$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$



$\frac{1}{1+e^{-(-\infty)}}$

$= \frac{1}{1+e^{\infty}}$

$= \frac{1}{1+\infty}$

$= 0$

Mathematical Formulation of Logistic Regression

We have already defined the **hypothesis function** as:

$$\boxed{h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}}$$

to Probabilites

Now, let's go through the remaining key mathematical components of logistic regression.

Binary Classification Problem   Class labels are 0 or 1

$$P(y=y|\theta;x) = h_\theta(x)^{y}\,(1-h_\theta(x))^{1-y}$$

1. Probability Interpretation

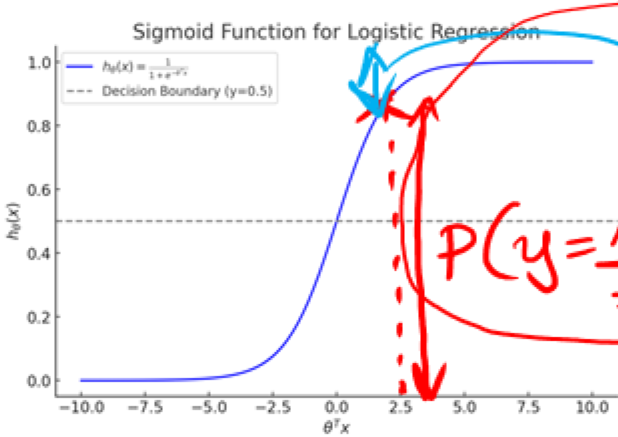
$$P(y=1|\theta;x) = h_\theta(x)^{1}\,(1-h_\theta(x))^{0}$$

$$\Rightarrow h_\theta(x)$$

Since $h_\theta(x)$ represents the probability of $y = 1$, we can write:

$$P(y = 1|x;\theta) = h_\theta(x)$$

$$P(y = 0|x;\theta) = 1 - h_\theta(x)$$



$$P(y=0|\theta^T x) = 1-h_\theta(x)$$

$$P(y=1|\theta^T x) = h_\theta(x)$$

$$P(y=0|\theta;x) = h_\theta(x)^{0}\,(1-h_\theta(x))^{1-0}$$

$$\Rightarrow 1-h_\theta(x)$$

Likelihood Function

Prob of getting Head (P), P(T) = 1-p

H T T H H T H T T H

$p \cdot (1-p)(1-p)\, p\, p$

$= p^5 \times (1-p)^5$

independent

$p(A \cap B \cap C) = p(A) \cdot p(B) \cdot p(C)$

1. Likelihood Function

Given a dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$, we assume that each training example follows a **Bernoulli** distribution:

$$P(y|x;\theta) = h_\theta(x)^y (1 - h_\theta(x))^{(1-y)}$$

For all m training samples, assuming independence, the **likelihood function** is the product of the probabilities for all data points:

$$L(\theta) = \prod_{i=1}^{m} h_\theta(x^{(i)})^{y^{(i)}} (1 - h_\theta(x^{(i)}))^{(1-y^{(i)})}$$

product

$\theta_0, \theta_1, \theta_2$

Data f_1 f_2

$x^{(1)}$
$x^{(2)}$
$x^{(3)}$
⋮

Train Data

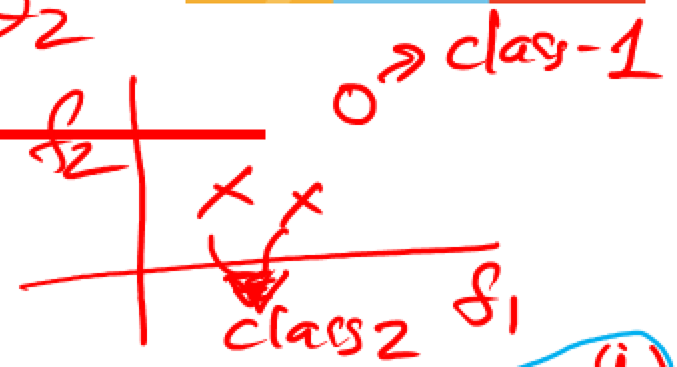
where:

- $h_\theta(x^{(i)}) = \frac{1}{1+e^{-\theta^T x^{(i)}}}$ is the predicted probability of $y = 1$.
- If $y^{(i)} = 1$, the term $(1 - h_\theta(x^{(i)}))^{(1-y^{(i)})}$ vanishes.
- If $y^{(i)} = 0$, the term $h_\theta(x^{(i)})^{y^{(i)}}$ vanishes.

$y^{(i)} = 1$

$h_\theta(x^{(i)}) = 0.01$

Take some random $\theta_0, \theta_1, \theta_2$



$$\begin{array}{c|cc|l}
 & f_1 & f_2 & \\
x^{(1)} & & & \text{class } 1 \quad y^{(1)} = 1 \\
x^{(2)} & & & \text{class } 0 \quad y^{(2)} = 0 \\
x^{(3)} & & & \text{class } 0 \quad y^{(3)} = 0
\end{array}$$

$$L(\theta) = \prod_{i=1}^{m} h_\theta(x^{(i)})^{y^{(i)}} \times \left(1 - h_\theta(x^{(i)})\right)^{1-y^{(i)}}$$

$h_\theta(x^{(1)}) = 0.001$

$h_\theta(x^{(2)}) = 0.8$

$h_\theta(x^{(3)}) = 0.9$

$L(\theta)$ = 0.001 × 0.2 × 0.1

$i = 1$ | $i = 2$ | $i = 2$

Class 1 | class-0 | class0

$P(y=1 \mid x;\theta)$ | $P(y=0 \mid x,\theta)$ | $P(y=0 \mid x,\theta)$

Maximizing Likelihood

innovate achieve lead

Maximizing the Likelihood Function in Logistic Regression

Once we have the **likelihood function:**

$$L(\theta) = \prod_{i=1}^{m} h_\theta(x^{(i)})^{y^{(i)}} (1 - h_\theta(x^{(i)}))^{(1-y^{(i)})}$$

our goal is to find the parameters θ that **maximize** this function. In other words, we want to find the **best** θ such that the predicted probabilities align as closely as possible with the actual labels in the training data.

New θ $\rightarrow$

$h_\theta(x^{(1)}) = 0.9$

$h_\theta(x^{(2)}) = 0.2$

$h_\theta(x^{(3)}) = 0.1$

$L(\theta) = 0.9 \times 0.8 \times 0.9$

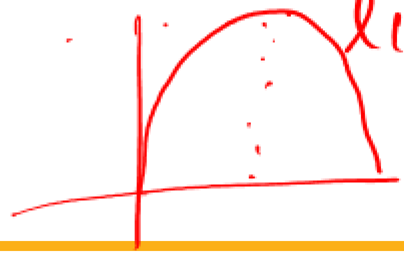
$0.9 \rightarrow P(y=1 \mid x^{(1)}; \theta)$

$0.8 \rightarrow P(y=0 \mid x^{(2)}; \theta)$

$0.9 \rightarrow P(y=0 \mid x^{(3)}; \theta)$

Why Maximize the Likelihood?

- The likelihood function represents the probability of observing the given dataset, assuming the logistic regression model is correct.
- A higher likelihood means that the model's predicted probabilities closely match the actual outcomes.
- By maximizing $L(\theta)$, we are **choosing parameters** θ **that make the observed data most probable** under our model.



$\max(l(\theta))$ $\quad$ $\min(-l(\theta))$

$0.1 \times 0.2 \times 0.01$

$\log(a \cdot b) = \log(a) + \log(b)$

$\log a^y = y \log a$

Step 1: Taking the Log of the Likelihood

Given the likelihood function:

$$\boxed{L(\theta) = \prod_{i=1}^{m} h_\theta(x^{(i)})^{y^{(i)}} (1 - h_\theta(x^{(i)}))^{(1-y^{(i)})}}$$

we take the **log-likelihood**:

$$\log\left(\underbrace{h_\theta(x^{(i)})^{y^{(i)}}}_{a} \times \underbrace{(1-h_\theta(x^{(i)}))^{1-y^{(i)}}}_{b}\right)$$

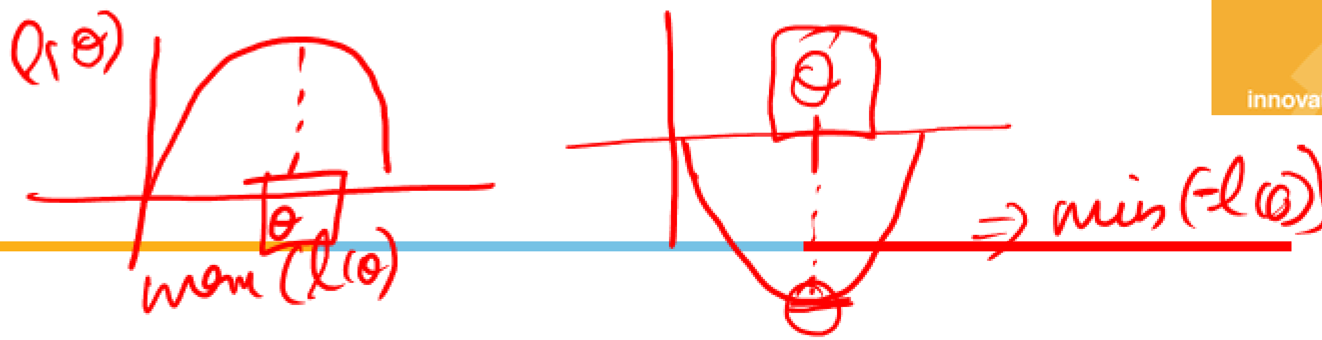
$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} \left[y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))\right]$$

To ensure consistency with loss functions commonly used in machine learning, we take the **average log-likelihood** (by including $\frac{1}{m}$):

$\log(a) + \log b$

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

max $l(\theta)$ is same as min $-l(\theta)$



Step 3: Converting to a Minimization Problem

Optimization algorithms typically **minimize** a function rather than maximize it. Instead of maximizing $\ell(\theta)$, we minimize its **negative**:

$$J(\theta) = -\ell(\theta)$$

Substituting the log-likelihood:

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

This is the **binary cross-entropy loss** (also called **log-loss**), which is the function we minimize using **gradient descent**.

iterations

$$\boxed{\theta^{t+1} := \theta^{t} - \alpha \frac{\partial J(\theta)}{\partial \theta}}$$

$$\frac{\partial J(\theta)}{\partial \theta} =$$

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

$x^{(i)} = [x_1^{(i)} \; x_2^{(i)}]$ $\quad \theta = \theta_0, \theta_1 \; \theta_2$

$[1 \; x_1^{(i)} \; x_2^{(i)}$

$x_0^{(i)}$

Gradient Descent for Parameter Estimation

The gradient of the **log-loss function** with respect to θ_j is:

$$\boxed{\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}}$$

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \times x_0^{(i)}$$

$$\frac{\partial J}{\partial \theta_1} = \quad x_1^{(i)}$$

$$\frac{\partial J}{\partial \theta_2} = \quad x_2^{(i)}$$

where:

- m is the number of training examples.
- $h_\theta(x^{(i)}) = \frac{1}{1+e^{-\theta^T x^{(i)}}}$ is the predicted probability.
- $y^{(i)}$ is the actual class label (0 or 1).
- $x_j^{(i)}$ is the j-th feature of the i-th training example.

We update θ_j using **gradient descent**:

$$\boxed{\theta_j := \theta_j - \alpha \cdot \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}}$$

where:

- α is the **learning rate**, controlling how big the update steps are.

Coding

$y = mx + b$

$\theta^T x$

$$\begin{bmatrix} \theta_0 & \theta_1 \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}$$

$b \quad m$

$m_1 x_1 + m_2 x_2 + b_2$

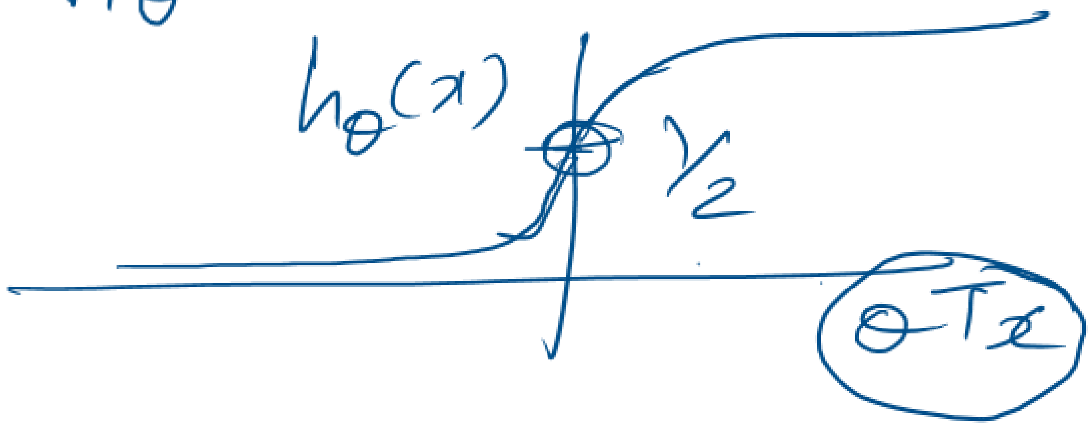
$$\begin{bmatrix} b_2 & m_1 & m_2 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$$

θ^T

Decision Rule

$$h_\theta(x_i) \geq 0.5 \quad \text{class-1}$$

$$h_\theta(x) < 0.5 \quad \text{class-0}$$



$\theta^T x$ $\quad y = mx + b$

$mx - y + b$

$$\boxed{\begin{pmatrix} b \\ m \\ -1 \end{pmatrix}} \begin{pmatrix} 1 \\ x \\ y \end{pmatrix}$$

θ

$$\theta^T x = b + mx - y$$

$$P(y=1 \mid x, \theta) = h_\theta(x)$$

$$P(y=0 \mid x, \theta) = 1 - h_\theta(x)$$

For a test data

if $P(y=1 \mid x;\theta) \geq P(y=0 \mid x;\theta)$

$$\frac{P(y=1 \mid x;\theta)}{P(y=0 \mid x;\theta)} \geq 1$$

$$\Rightarrow \boxed{\frac{h_\theta(x)}{1-h_\theta(x)} \geq 1}$$

$$\frac{\frac{1}{1+e^{-\theta^T x}}}{1-\frac{1}{1+e^{-\theta^T x}}} \geq 1$$

$$\frac{\frac{1}{1+e^{-\theta^T x}}}{\frac{\cancel{1}+e^{-\theta^T x}-\cancel{1}}{\underbrace{1+e^{-\theta^T x}}}}$$

$$\Rightarrow \frac{1}{\cancel{1+e^{-\theta^T x}}} \times \frac{\cancel{1+e^{-\theta^T x}}}{e^{-\theta^T x}} \geq 1$$

$$e^{\theta^T x} \geq 1$$

$$\ln e^{\theta^T x} \geq \ln(1)$$

$$\boxed{\theta^T x \geq 0}$$

innovate achieve lead

Perceptron
Logistic Regression
SVM

Perceptron

Vol. VI, No. 2, Summer 1958

research trends

CORNELL AERONAUTICAL LABORATORY, INC., BUFFALO 21, NEW YORK

The Design of an Intelligent AUTOMATON

by FRANK ROSENBLATT

Introducing the perceptron — A machine which senses, recognizes, remembers, and responds like the human mind.

STORIES about the creation of machines having human qualities have long been a fascinating province in the realm of science fiction. Yet we are now about to witness the birth of such a machine — a machine capable of perceiving, recognizing, and identifying its surroundings without any human training or control.

Development of that machine has stemmed from a search for an understanding of the physical mechanisms which underlie human experience and intelligence. The question of the nature of these processes is at least as ancient as any other question in western science and philosophy, and, indeed, ranks as one of the greatest scientific challenges of our time.

Our understanding of this problem has gone perhaps as far as had the development of physics before Newton. We have some excellent descriptions of the phenomena to be explained, a number of interesting hypotheses, and a little detailed knowledge about events in the nervous system. But we lack agreement on any integrated set of principles by which the functioning of the nervous system can be understood.

We believe now that this ancient problem is about to yield to our theoretical investigation for three reasons:

First, in recent years our knowledge of the functioning of individual cells in the central nervous system has vastly increased.

Second, large numbers of engineers and mathematicians are, for the first time, undertaking serious study of the mathematical basis for thinking, perception, and the handling of information by the central nervous system, thus providing the hope that these problems may be within our intellectual grasp.

Third, recent developments in probability theory and in the mathematics of random processes provide new tools for the study of events in the nervous system, where only the gross statistical organization is known and the precise cell-by-cell "wiring diagram" may never be obtained.

Receives Navy Support

In July, 1957, Project PARA (Perceiving and Recognizing Automaton), an internal research program which had been in progress for over a year at Cornell Aeronautical Laboratory, received the support of the Office of Naval Research. The program had been concerned primarily with the application of probability theory to

innovate achieve lead