AI VIET NAM – AIO2024

Insight into Logistic Regression

Ngày 21 tháng 3 năm 2024

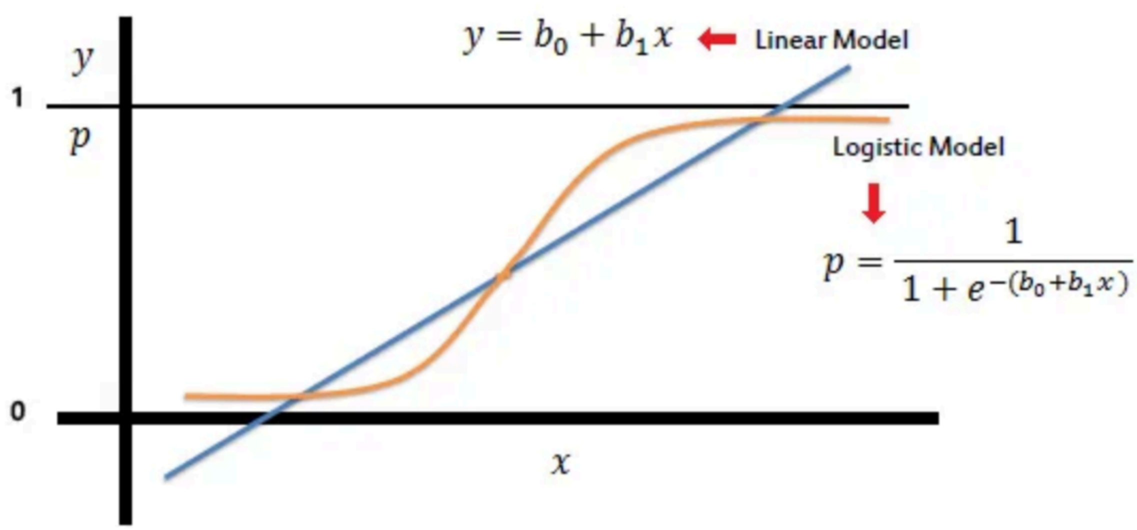
Người tóm tắt	Ngọc Trúc
Nguồn dữ liệu:	Module 5
Từ khóa:	Logistic Regression, Vectorization, Loss Function

Bài học này sẽ tìm hiểu về mô hình Hồi quy Logistic (Logistic Regression)

1 Logistic Regression

Hồi quy Logistic là một mô hình thống kê được sử dụng để phân loại nhị phân, tức dự đoán một đối tượng thuộc vào một trong hai nhóm. Hồi quy Logistic làm việc dựa trên nguyên tắc của hàm sigmoid – một hàm phi tuyến tự chuyển đầu vào của nó thành xác suất thuộc về một trong hai lớp nhị phân.

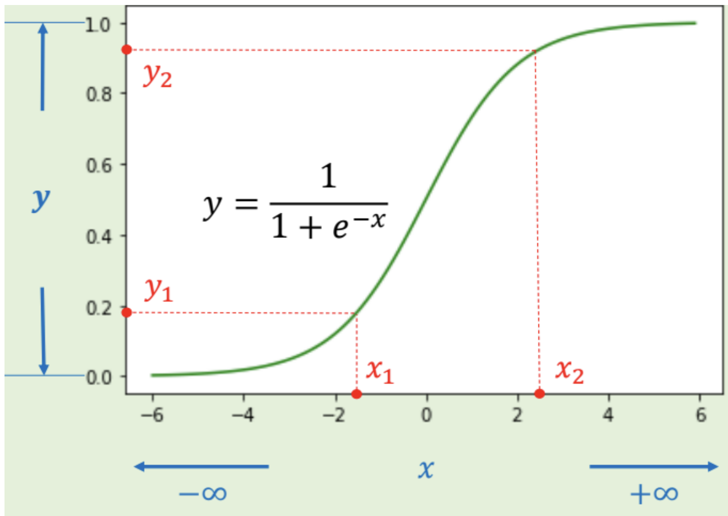
Khác với bài toán Linear Regression sẽ kết quả trả về đại lượng liên tục ví dụ như là giá tiền của một ngôi nhà sau khi biết được diện tích của nó còn với Logistic Regression kết quả trả về sẽ là đại lượng rời rạc như 0,1 biểu thị cho một sự kiện được dự đoán có xảy ra hay không.



Hình 1: Logistic Regression sử dụng hàm phi tuyến để xác định xác suất của hai lớp 0 và 1.

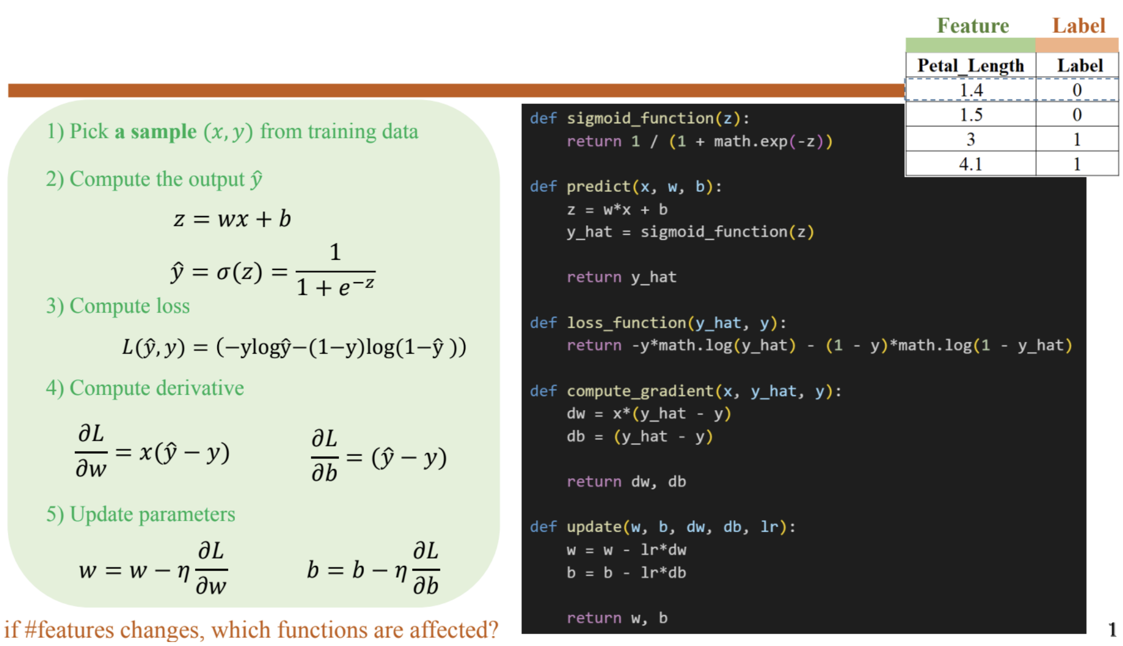
Hồi quy Logistic hoạt động dựa trên hàm Sigmoid, được biểu diễn như sau:

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



Hình 2: Sigmoid function

Hãy bắt đầu với bài toán phân loại khi có 1 feature, ta có các bước thực hiện và code như sau:

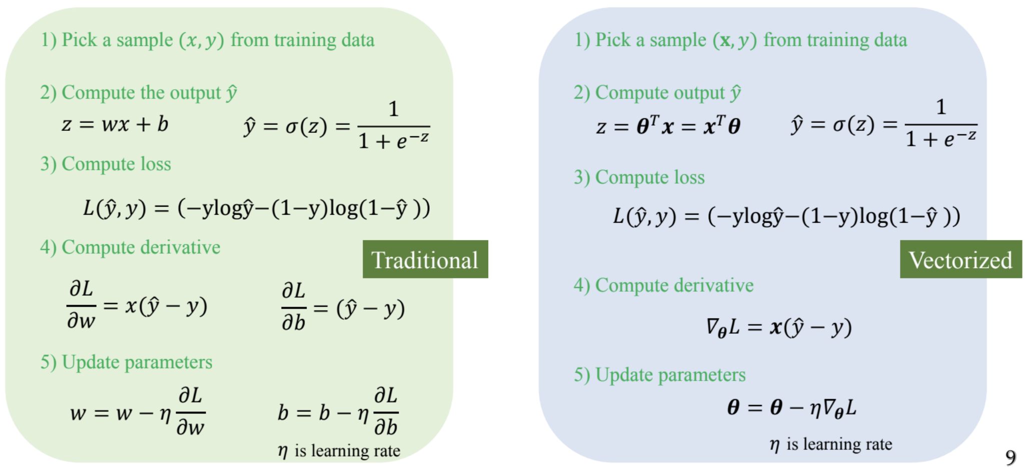


Hình 3: Bài toán logistic regression 1 sample và 1 feature

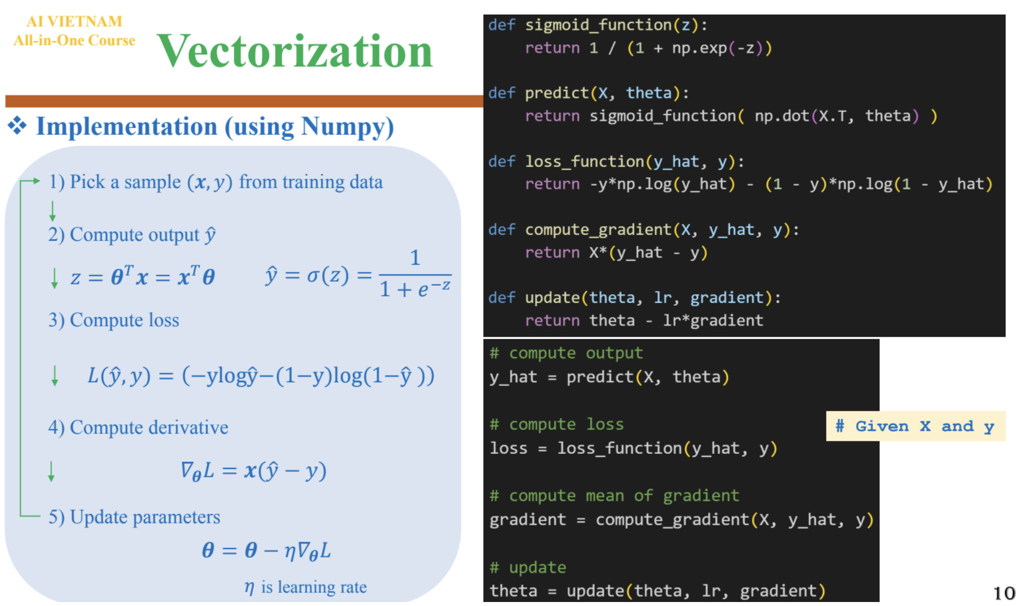
Tuy nhiên,khi tăng số lượng feature của biến, công thức sẽ có sự thay đổi từ biến x, w, b dẫn đến công thức và code ở các bước 2,4,5 sẽ cần điều chỉnh. Vậy để khái quát hóa bài toán phân loại với số lượng N feature bất kỳ, các biến sẽ cần chuyển đổi qua dạng vector.

2 Vectorization

Để xử lý bài toán khi số lượng features của biến thay đổi, bài toán sẽ cần chuyển đổi thành vector.



Hình 4: Các bước chuyển đổi qua vector



Hình 5: Sử dụng numpy cho bài toán logistic regression 1 sample với n features

Với việc xây dựng công thức trên cho bài toán 1 sample với nhiều feature, hãy tiếp tục triển khai với bài toán có nhiều sample hơn.

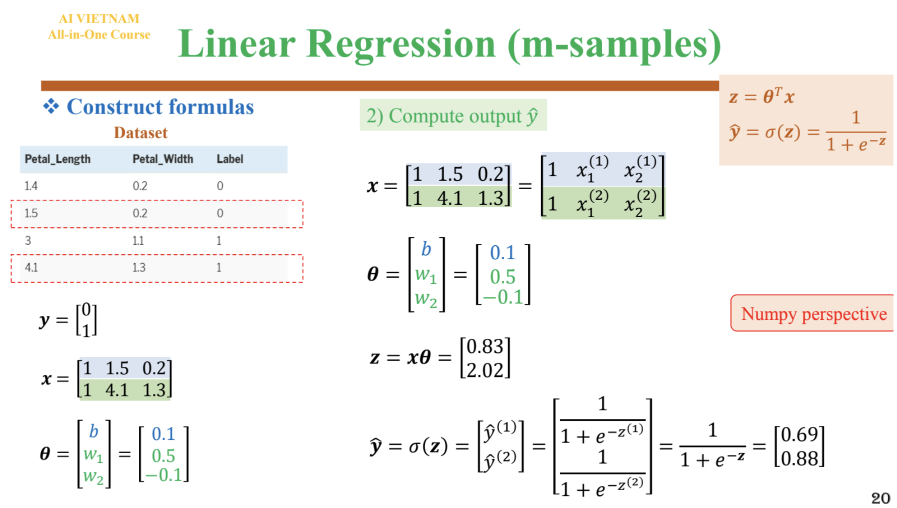
3 Optimiztion for 1+ samples

Trong hồi quy logistic, khi có nhiều hơn 1 sample thì việc tối ưu hóa trở nên quan trọng hơn và khi số lượng sample tăng lên, mô hình thường trở nên ổn định hơn và chính xác cao hơn. Để làm điều này, có 2 cách:

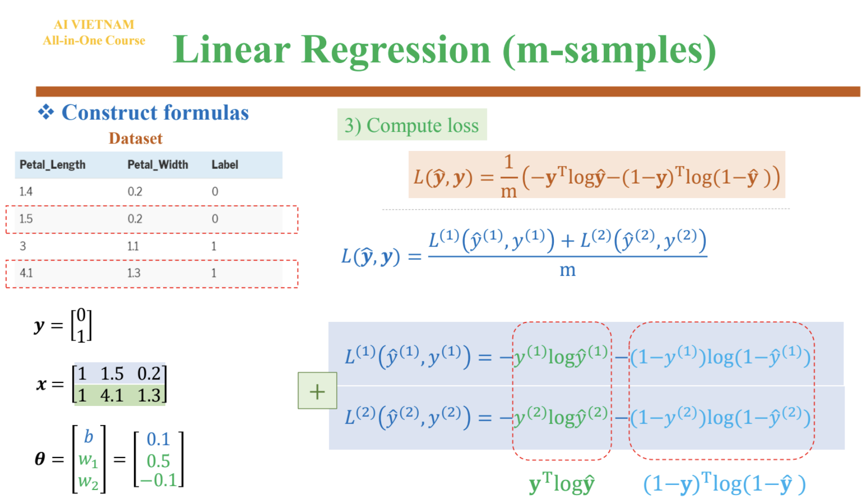
- Cách 1: Cập nhật mô hình lần lượt từng sample theo thứ tự, cách này có nhược điểm là kết quả sẽ rời rạc và sẽ bị ảnh hưởng bởi sample cuối cùng.
- Cách 2: Cập nhật cùng lúc các sample bằng phương pháp tính tổng hoặc giá trị trung bình - mean của các đạo hàm từ sample (thường phổ biến hơn) và sau đó cập nhật 1 lần.

==> Cách 2 tối ưu hơn, hãy cùng xây dựng tiếp công thức tổng quát như sau

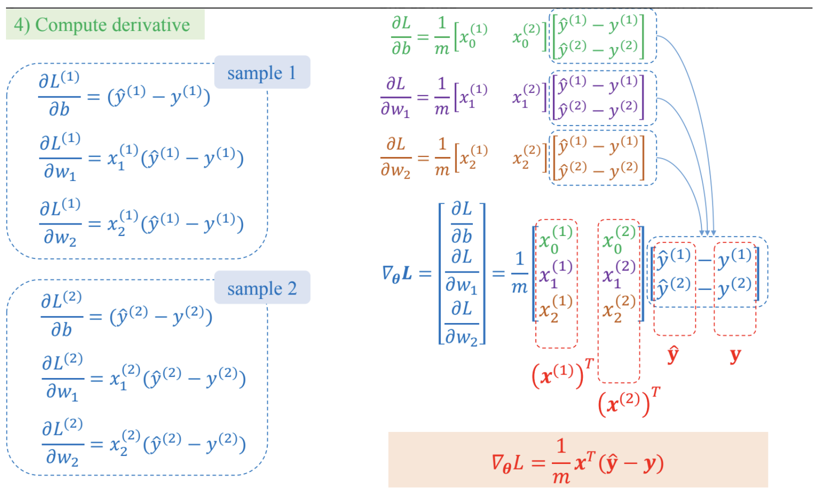
- **Bước 1: Xác định các giá trị x, y, θ**
  Trong đó x là 1 ma trận, mỗi sample nằm trên 1 dòng và θ là một vector cột.
- **Bước 2: Tính giá trị của $\hat{y}$**
  Theo công thức $\hat{y} = \theta(\text{z})$, hãy thực hiện phép tính như hình dưới đây:



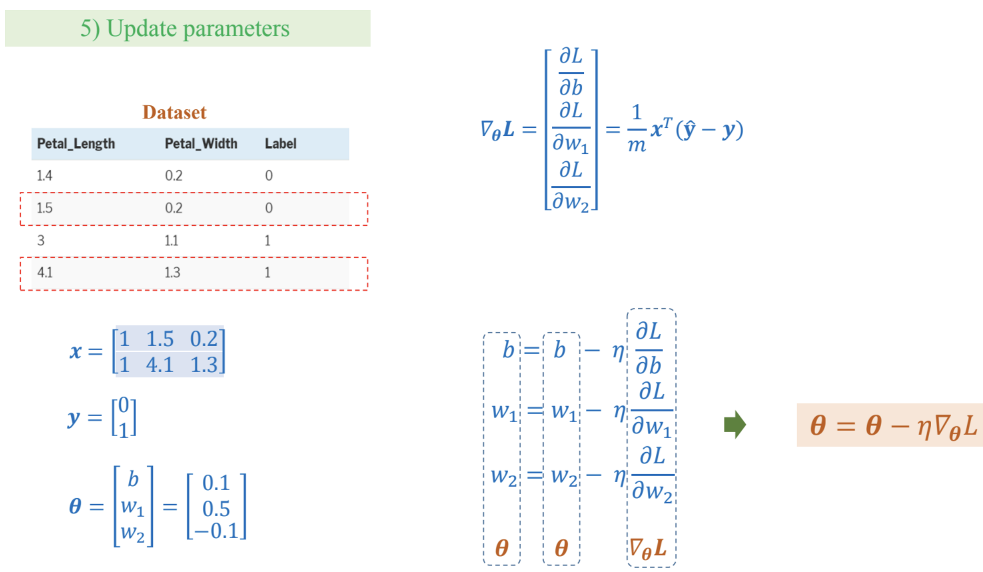
- **Bước 3: Tính loss**
  Loss của mô hình:L($\hat{y}$,y) là trung bình các Loss của tất cả sample, chi tiết như hình minh họa như sau:



- **Bước 4: Tính đạo hàm Loss**
  Xây dựng được công thức như sau:



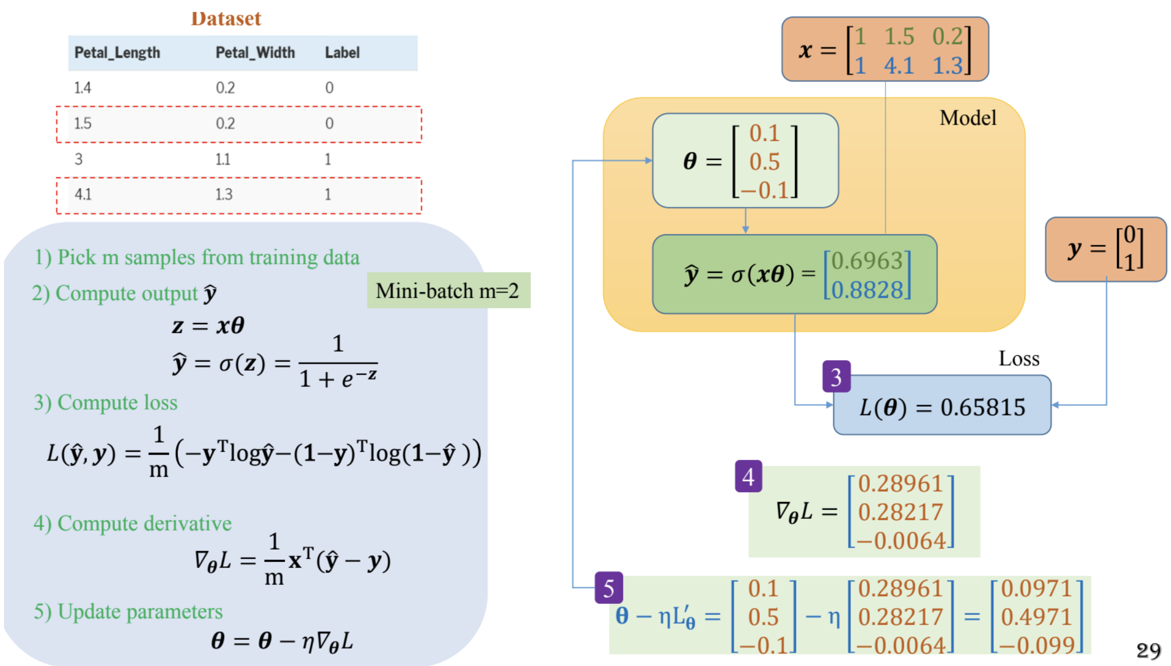
- **Bước 5: Cập nhật các giá trị của b, w, và θ**



Trên đây là các bước để xây dựng công thức cho Logitic Regression cho m samples.

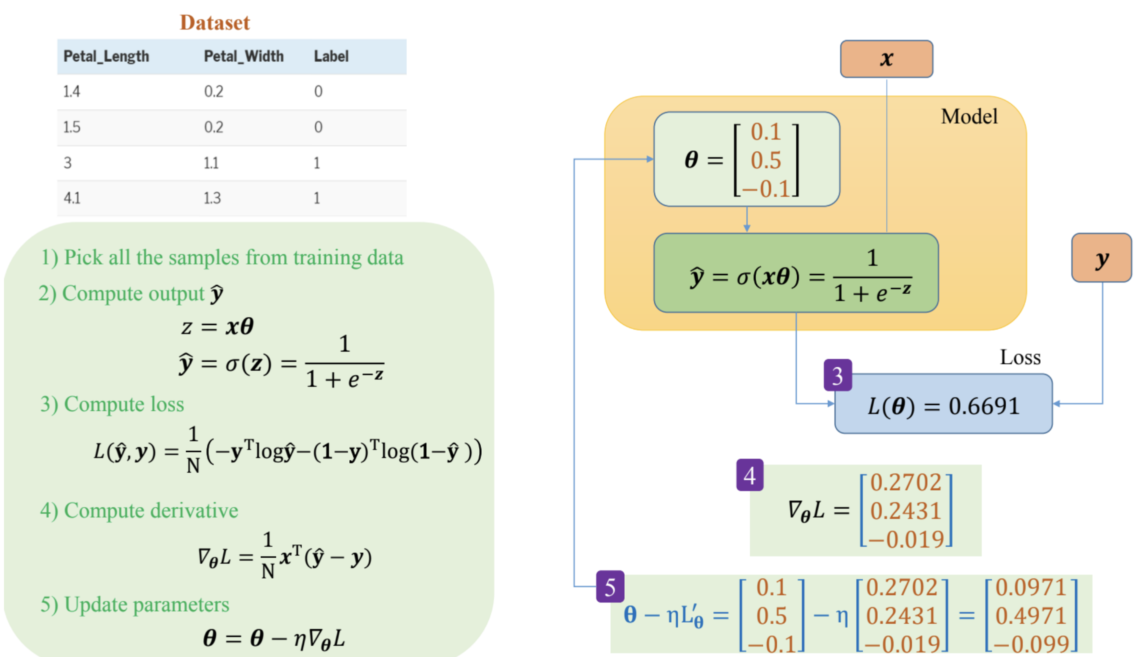
4 Logistic Regression - Mini Batch

Để thực hành, hãy tính bài toán đơn giản khi chọn 2 samples - minibatch với đề bài và kết quả như hình sau:



5 Logistic Regression - Batch

Hãy tiếp tục xây dựng mô hình với 4 sample, từ đó mở rộng với m sample.



6 Binary Cross-entropy (BCE) and Mean Squared Error (MSE) Loss Functions

Ma trận Hessian

- Ma trận Hessian là một công cụ quan trọng được sử dụng để kiểm tra độ cong của Loss function trong quá trình tối ưu hóa. Cụ thể, ma trận Hessian của một hàm Loss $L(\theta)$ là ma trận bậc hai của các đạo hàm riêng của hàm Loss theo các tham số của mô hình. Nó thường được ký hiệu là **H** và được xác định như sau:

$$H_{ij} = \frac{\partial^2 L(\theta)}{\partial \theta_i \partial \theta_j}$$

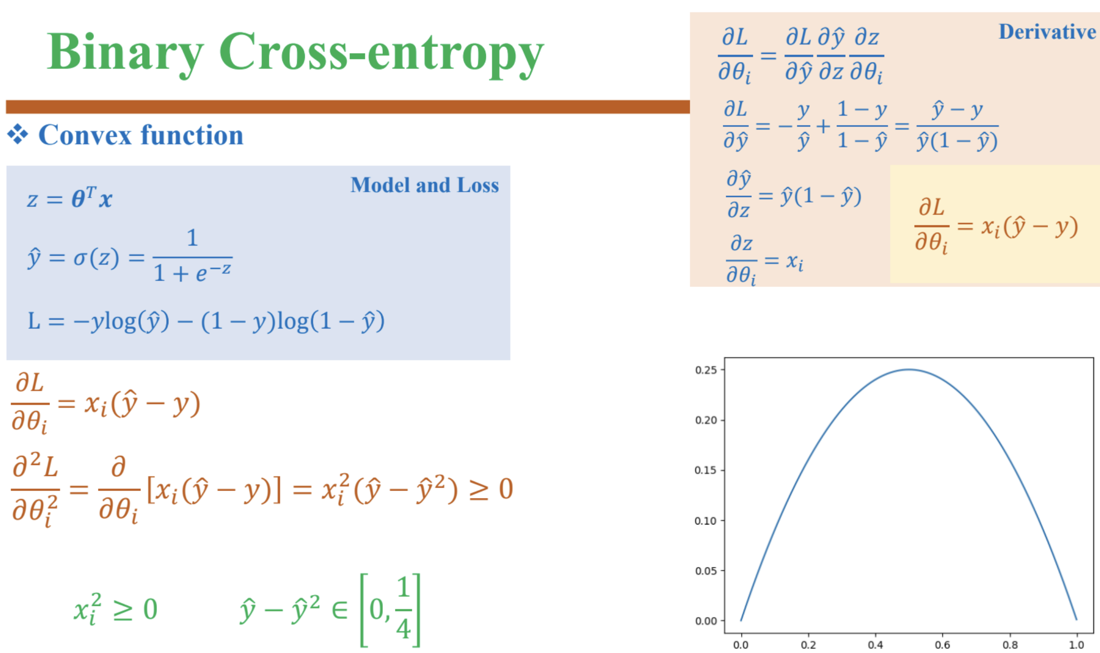
  Trong đó:

  - $\boldsymbol{\theta}$ là vector tham số của mô hình.

  - $\frac{\partial^2 L(\theta)}{\partial \theta_i \partial \theta_j}$ là đạo hàm bậc hai của hàm mất mát $L(\theta)$ theo các tham số θ_i và θ_j.

- Ma trận Hessian được sử dụng trong logistic regression để tối ưu hóa tham số của mô hình, đặc biệt là trong việc xác định hướng và độ lớn của điểm cực tiểu hoặc điểm cực đại của hàm mất mát.

- Trong logistic regression, Loss function phổ biến được sử dụng là binary cross-entropy (hay còn gọi là log loss) vì nó phản ánh tốt hơn sự khác biệt giữa các xác suất dự đoán và nhãn thực tế.



- Bên cạnh hàm Binary cross-entropy, Mean Squared Error (MSE) có thể được sử dụng làm Loss function nhưng sẽ không phù hợp trong logistic regression vì nó không đảm bảo Loss function convex (lồi) và giới hạn các dự đoán trong khoảng [0, 1], dẫn đến các vấn đề trong tối ưu hóa và hiệu suất của mô hình.

Mean Squared Error

Model and Loss

$$z = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

$$\hat{y} = \sigma(z) = \frac{1}{1+e^{-z}}$$

$$\mathrm{L} = (\hat{y} - y)^2$$

Derivative

$$\frac{\partial L}{\partial \theta_i} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial \theta_i} \qquad \frac{\partial \hat{y}}{\partial z} = \hat{y}(1-\hat{y})$$

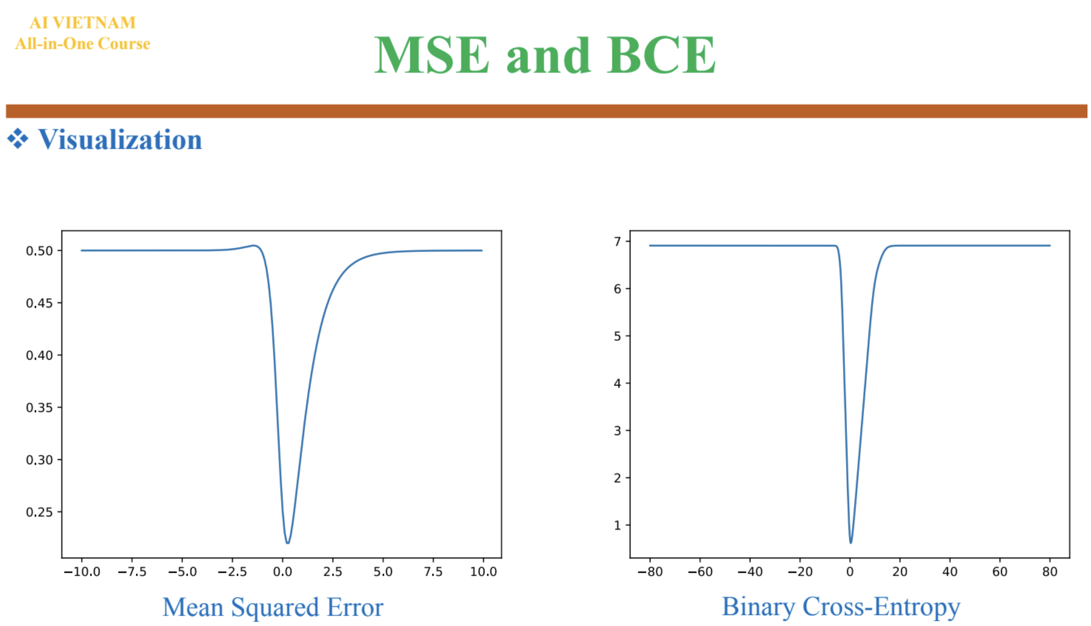
$$\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y) \qquad \frac{\partial z}{\partial \theta_i} = x_i$$

$$\frac{\partial L}{\partial \theta_i} = 2x_i(\hat{y} - y)\hat{y}(1-\hat{y})$$

$$\frac{\partial L}{\partial \theta_i} = 2x_i(\hat{y} - y)\hat{y}(1-\hat{y}) = 2x_i\left(-\hat{y}^3 + \hat{y}^2 - y\hat{y} + y\hat{y}^2\right)$$

$$\begin{aligned}\frac{\partial^2 L}{\partial \theta_i^2} &= \frac{\partial}{\partial \theta_i}\left[2x_i(-\hat{y}^3 + \hat{y}^2 - y\hat{y} + y\hat{y}^2)\right] \\ &= 2x_i\left[-3\hat{y}^2 x_i \hat{y}(1-\hat{y}) + 2x_i\hat{y}\hat{y}(1-\hat{y}) - yx_i\hat{y}(1-\hat{y}) + 2x_i y\hat{y}\hat{y}(1-\hat{y})\right] \\ &= 2x_i^2\hat{y}(1-\hat{y})\left[-3\hat{y}^2 + 2\hat{y} - y + 2y\hat{y}\right]\end{aligned}$$

- Hãy so sánh hình ảnh minh họa như sau:
  BCE - Binary cross-entropy thường được sử dụng nhiều hơn khi phân loại (classification), trong khi MSE - Mean Squared Error thường sử dụng cho hồi quy (regression)



7 Logistic Regression Tanh

Ngoài Sigmoid được sử dụng phổ biến thì Hồi quy Logistic cũng có thể hoạt động dựa trên hàm Tanh, được biểu diễn như sau:

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

Model and Loss

$$z = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

$$\hat{y} = tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$\hat{y}_s = \frac{\hat{y} + 1}{2}$$

$$\mathrm{L} = -y\log(\hat{y}_s) - (1 - y)\log(1 - \hat{y}_s)$$

-Hết-