Logistic Regression
(Hồi quy Logistic)

Nhóm 8
Đỗ Tất Thành
Cao Thọ Hiếu

Mở đầu

- Là **thuật toán học máy có giám sát**
- Sử dụng cho **phân lớp đối tượng**
- Mẫu có thể được phân theo hai (**two-class case**) hoặc nhiều lớp (**multinomial logistic regression**)
- Là **discriminative classifier** (thuật toán tìm hiểu sự khác nhau giữa các lớp) thay vì **generative classifier** (tìm hiểu đặc điểm của mỗi lớp và phân loại dựa trên)
- Là thuật toán phân loại xác suất

Mở đầu

- Một thuật toán học máy phân loại thường có 4 thành phần:
  - Các đặt trưng của đầu vào (**feature representation**)
    - mẫu $x^{(i)}$ có các đặc trưng **$[x^{(i)}_1, x^{(i)}_2, ..., x^{(i)}_n]$**
  - Hàm phân loại (**classification function**)
    - phân loại mẫu vào **lớp dự đoán ŷ**
    - **sigmoid** (two-class) hoặc **softmax** (multinomial classifier)
  - Hàm mục tiêu cho quá trình học (**objective function for learning**)
    - Đo lường lỗi (sai khác giữa nhãn thực và nhãn dự đoán của mô hình) trong quá trình huấn luyện
    - **cross-entropy loss function**

Mở đầu

- Một thuật toán học máy phân loại thường có 4 thành phần (tiếp):
  - Thuật toán tối ưu hàm mục tiêu:
    - **stochastic gradient descent**
    - **batch gradient descent**
    - **mini-batch gradient descent**

- Hai pha chính:
  - Huấn luyện - **training:**
    - huấn luyện hệ thống (tìm trọng số – **weight** - **w** và **intercept** - **b** phù hợp cho mô hình dự đoán) dựa trên **mẫu** $\mathbf{x}^{(i)}$ và **nhãn** $\mathbf{y}^{(i)}$ cho trước trong tập huấn luyện
    - sử dụng
      - hàm mục tiêu **cross-entropy loss**

Mở đầu

- Hai pha chính:
  - Huấn luyện - **training**:
    - sử dụng (tiếp):
      - thuật toán tối ưu **stochastic** / **batch** / **mini-batch** gradient descent
  - Kiểm tra – **test**:
    - cho dữ liệu kiểm tra x, tính p(y|x) và trả về nhãn y của x cho p cao hơn
  - Pha hiệu chỉnh - **Validation** (có thể có hoặc không):
    - Hiệu chỉnh mô hình (**Regularization**) để tránh **overfitting** hoặc **underfitting**
    - Hiệu chỉnh L1 (**L1 regularization**)
    - Hiệu chỉnh L2 (**L2 regularization**)

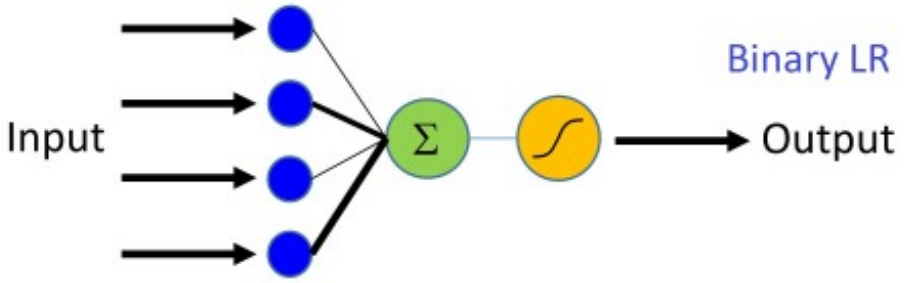
Cơ chế hoạt động

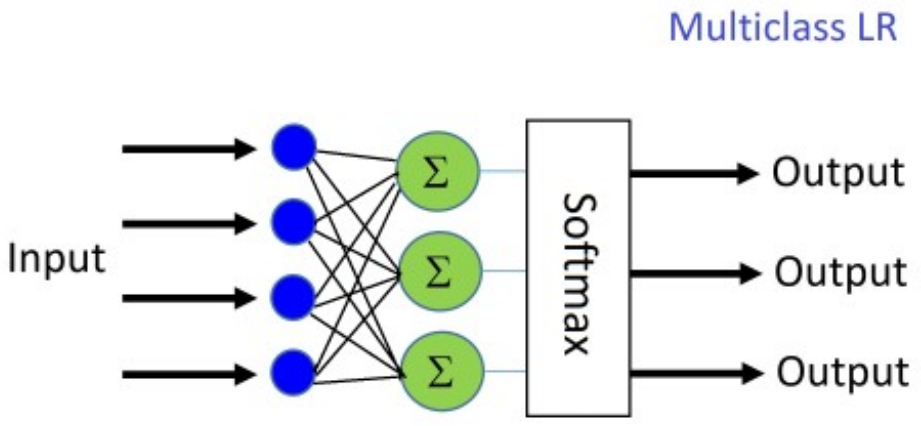
- Cho data set:
  - [[x^1_1 , x^1_2 , ..., x^1_n , y^1],

    …

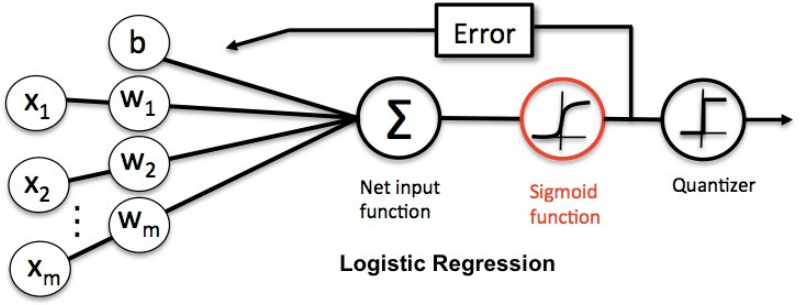
    [x^m_1 , x^m_2 , ..., x^m_n , y^m]]
  - $\mathbf{x}^i_j$ là đặc trưng j của quan sát thứ i, y^i là nhãn của quan sát thứ I
- Tìm trọng số w và **intercept** / **bias term** b
  - w := [w_1 , w_2 , ..., w_n]’
  - b

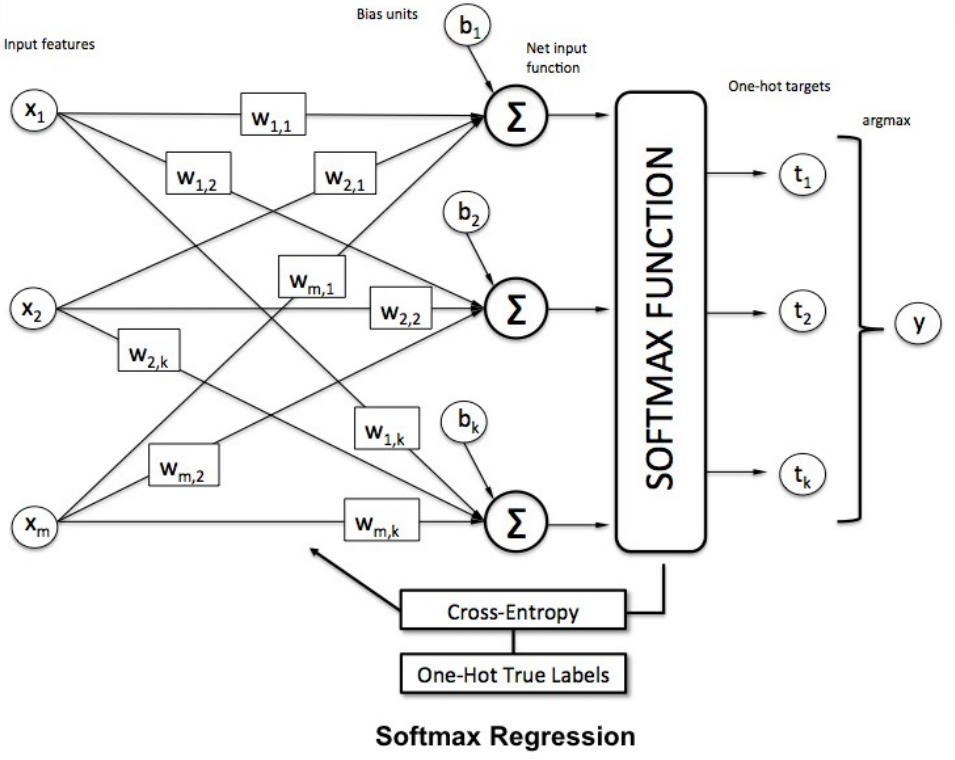
  sao cho $L_{CE}(\hat{y}, y) \rightarrow$ min
  - $L_{CE}(\hat{y}, y)$:= cross-entropy loss function
  - $\hat{Y} \rightarrow$ nhãn dự đoán, y $\rightarrow$ nhãn thực



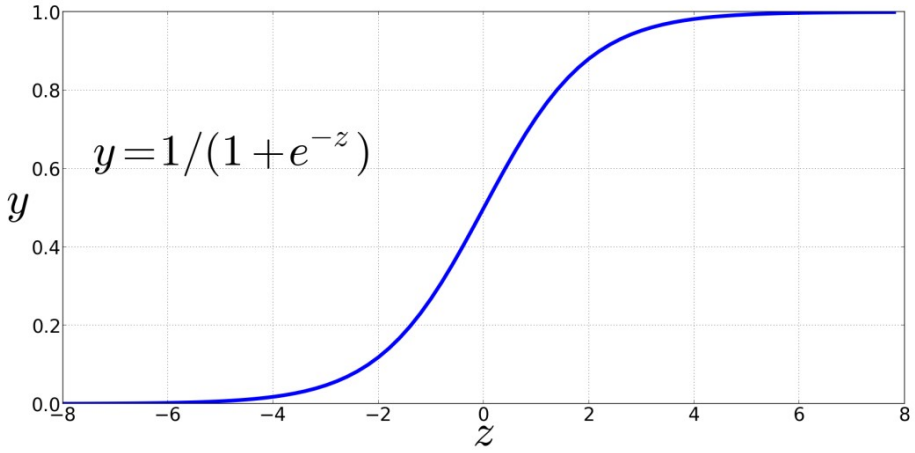


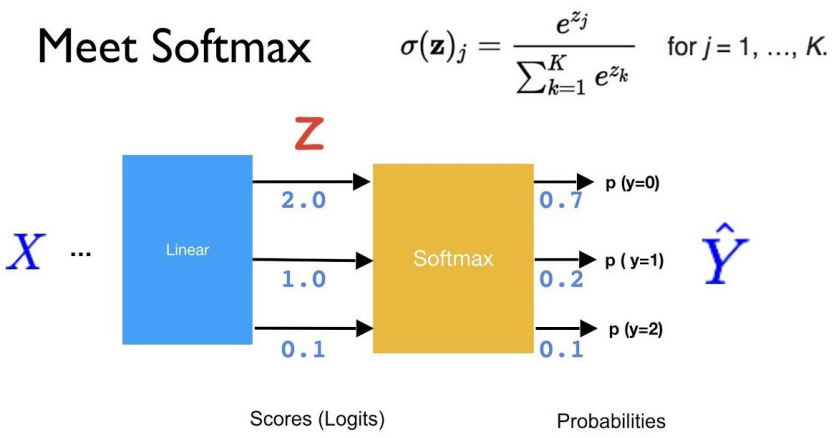
Cơ chế hoạt động





Hàm phân lớp - Classification Function





- Hàm sigmoid σ:
  - Phân loại 2 lớp
    - y = 0 và y = 1
    - Y = + (positive) và y = - (negative)
    - ...

- Hàm softmax
  - Phân loại k lớp (k >= 2)

Hàm phân lớp – Classification Function

- $x = [x_1 , x_2 , ..., x_n]$
- nhãn y
- $w = [w_1 , w_2 , ..., w_n]$‘
- bias/intercept b
- z = x * w + b

- Phân loại nhị phân:
  - $P(y = 1|x) = \sigma (w \cdot x + b)$
  - $P(y = 0|x) = 1 - \sigma (w \cdot x + b)$
  - $\hat{y} = 1$ if $P(y = 1|x) > 0.5$ else $\hat{y} = 0$
- Phân loại k lớp:

$$p(y = c|x) = \frac{e^{w_c \cdot x + b_c}}{\sum_{j=1}^{k} e^{w_j \cdot x + b_j}}$$

Hàm mục tiêu - Cross-Entropy Loss Function

- Loss Function:
  - L(ŷ, y) := Định lượng mức độ sai khác giữa ŷ và y
  - L ưu tiên cho các nhãn được gán đúng.
  - mục đích của thuật toán LR là chọn ra được w, b có thể cực đại được log-xác suất của các nhãn gán đúng y trong bộ dữ liệu huấn luyện
  - loss function thường dùng là cross-entropy loss.

Hàm mục tiêu - Cross-Entropy Loss Function

- Với phân lớp nhị phân (y = 1 || y = 0):

$$
\begin{aligned}
p(y|x) &= \hat{y}^{y}(1-\hat{y})^{1-y} \\
\log p(y|x) &= \log\left[\hat{y}^{y}(1-\hat{y})^{1-y}\right] \\
&= y\log\hat{y} + (1-y)\log(1-\hat{y}) \\
L_{CE}(\hat{y},y) = -\log p(y|x) &= -[y\log\hat{y} + (1-y)\log(1-\hat{y})] \\
L_{CE}(w,b) &= -[y\log\sigma(w\cdot x+b) + (1-y)\log(1-\sigma(w\cdot x+b))]
\end{aligned}
$$

Hàm mục tiêu - Cross-Entropy Loss Function

- Với phân lớp k lớp (k >= 2):

$$L_{CE}(\hat{y}, y) = -\sum_{k=1}^{K} 1\{y = k\} \log p(y = k|x)$$

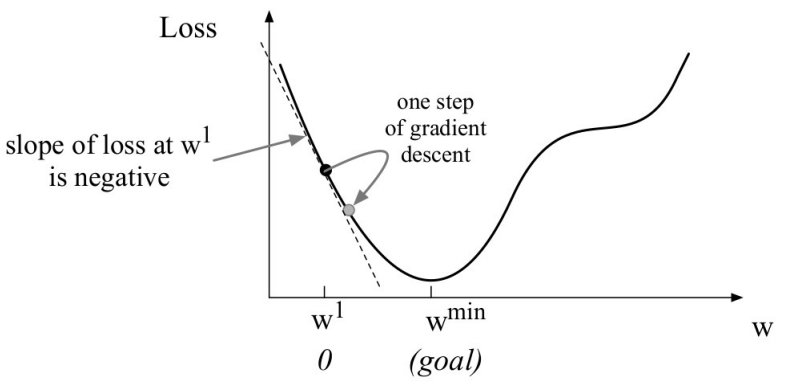
$$= -\sum_{k=1}^{K} 1\{y = k\} \log \frac{e^{w_k \cdot x + b_k}}{\sum_{j=1}^{K} e^{w_j \cdot x + b_j}}$$

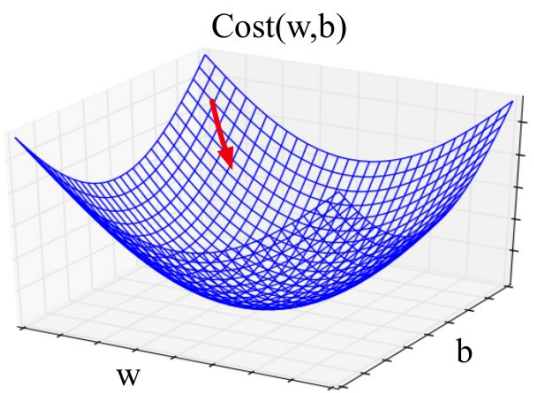
Tối ưu hàm mục tiêu - Gradient Descent

- Mục tiêu của LR:

$$\theta = w, b \qquad \hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} L_{CE}(y^{(i)}, x^{(i)}; \theta)$$

- Gradient của L_{CE} ứng với theta cho biết chiều tăng của L_{CE} tại theta. Đi theo chiều giảm gradient (Gradient Descent), tìm được vị trí $(w, b)^{argmin}$ ứng với cực tiểu hàm LCE. $(w, b)^{argmin}$ là tham số cần tìm của mô hình.





Gradient Descent

- Phương pháp chung tính gradient:

$$\nabla_\theta L(f(x;\theta),y)) = \begin{bmatrix} \frac{\partial}{\partial w_1} L(f(x;\theta),y) \\ \frac{\partial}{\partial w_2} L(f(x;\theta),y) \\ \vdots \\ \frac{\partial}{\partial w_n} L(f(x;\theta),y) \end{bmatrix}$$

- Gradient của phân lớp nhị phân:

$$L_{CE}(w,b) = -[y \log \sigma(w \cdot x + b) + (1-y) \log (1 - \sigma(w \cdot x + b))]$$

$$\frac{\partial L_{CE}(w,b)}{\partial w_j} = [\sigma(w \cdot x + b) - y] x_j$$

- Gradient của phân lớp k lớp:

$$L_{CE}(\hat{y},y) = -\sum_{k=1}^{K} 1\{y=k\} \log p(y=k|x)$$

$$= -\sum_{k=1}^{K} 1\{y=k\} \log \frac{e^{w_k \cdot x + b_k}}{\sum_{j=1}^{K} e^{w_j \cdot x + b_j}}$$

$$\frac{\partial L_{CE}}{\partial w_k} = -(1\{y=k\} - p(y=k|x)) x_k$$

$$= -\left(1\{y=k\} - \frac{e^{w_k \cdot x + b_k}}{\sum_{j=1}^{K} e^{w_j \cdot x + b_j}}\right) x_k$$

- Cập nhật trọng số:

$$\theta_{t+1} = \theta_t - \eta \nabla L(f(x;\theta),y)$$

Một số kỹ thuật tính Gradient

- Stochastic GD:
  - thuật toán online nhằm tối thiểu hóa loss function
  - tính gradient và cập nhật trọng số trên mỗi mẫu huấn luyện
- Mini-Batch GD:
  - huấn luyện nhóm m mẫu. Nếu m = 1 thì là Stochastic GD. Hoặc m <= kích thước tập mẫu.
  - Cho hiệu suất tính toán tốt do có thể chọn kích thước nhóm phù hợp với năng lực tính toán của máy tính hiện có. Vector hóa dữ liệu dễ dàng và có thể tận dụng năng lực tính toán song song.
- Batch GD:
  - huấn luyện với toàn bộ tập mẫu đồng thời (mini-batch với m = kích thước tập train).

function STOCHASTIC GRADIENT DESCENT($L()$, $f()$, x, y) **returns** θ
    # where: L is the loss function
    # f is a function parameterized by θ
    # x is the set of training inputs $x^{(1)}$, $x^{(2)}$, ..., $x^{(n)}$
    # y is the set of training outputs (labels) $y^{(1)}$, $y^{(2)}$, ..., $y^{(n)}$

$\theta \leftarrow 0$
repeat til done # see caption
  For each training tuple $(x^{(i)}, y^{(i)})$ (in random order)
    1. Optional (for reporting): # How are we doing on this tuple?
       Compute $\hat{y}^{(i)} = f(x^{(i)}; \theta)$ # What is our estimated output $\hat{y}$?
       Compute the loss $L(\hat{y}^{(i)}, y^{(i)})$ # How far off is $\hat{y}^{(i)}$) from the true output $y^{(i)}$?
    2. $g \leftarrow \nabla_\theta L(f(x^{(i)}; \theta), y^{(i)})$ # How should we move θ to maximize loss?
    3. $\theta \leftarrow \theta - \eta\, g$ # Go the other way instead
return θ