SOFTMAX REGRESSION

TS. Nguyễn Thị Kim Ngân

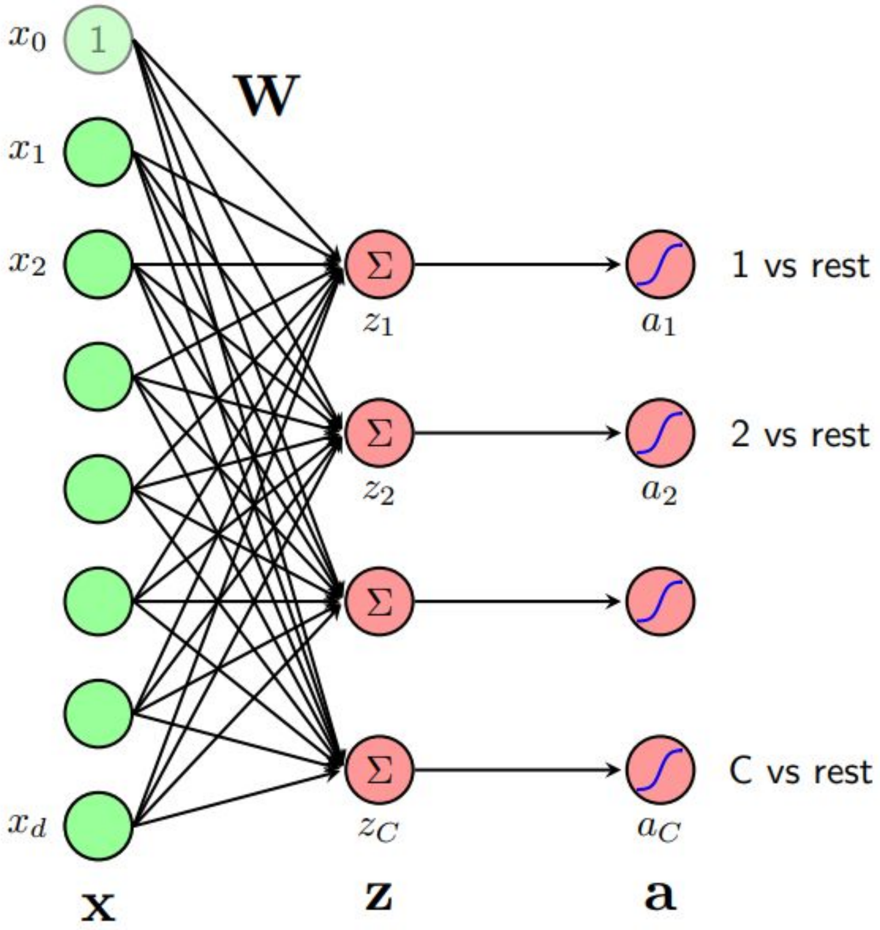
Bài toán phân lớp đa lớp

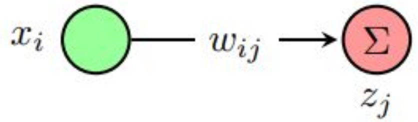
- Input: Tập dữ liệu đã được gán nhãn (X_train, y_train), nhãn của mâu xi là yi, yi nhận các giá trị nguyên thuộc đoạn [1, C]
- Output: Mô hình phân lớp

Kỹ thuật one-vs-rest

- Thực hiện C mô hình phân lớp.
- Mô hình phân lớp thứ i (i=1, …, C), xem xét bài toán gồm 2 phân lớp: lớp được gán nhãn i và lớp không được gán nhãn i. Nghĩa là, tập nhãn ytrain gồm 2 nhãn: nhãn i và nhãn khác i.

Phân đa lớp với logistic regression và one-vs-rest





w_{0j}: biases, don't forget!

d: data dimension

C: number of classes

$\mathbf{x} \in \mathbb{R}^{d+1}$

$\mathbf{W} \in \mathbb{R}^{(d+1)\times C}$

$z_i = \mathbf{w}_i^T \mathbf{x}$

$\mathbf{z} = \mathbf{W}^T \mathbf{x} \in \mathbb{R}^C$

$a_i = \text{sigmoid}(z_i) \in \mathbb{R}$

$0 < a_i < 1$

Công thức của Softmax function

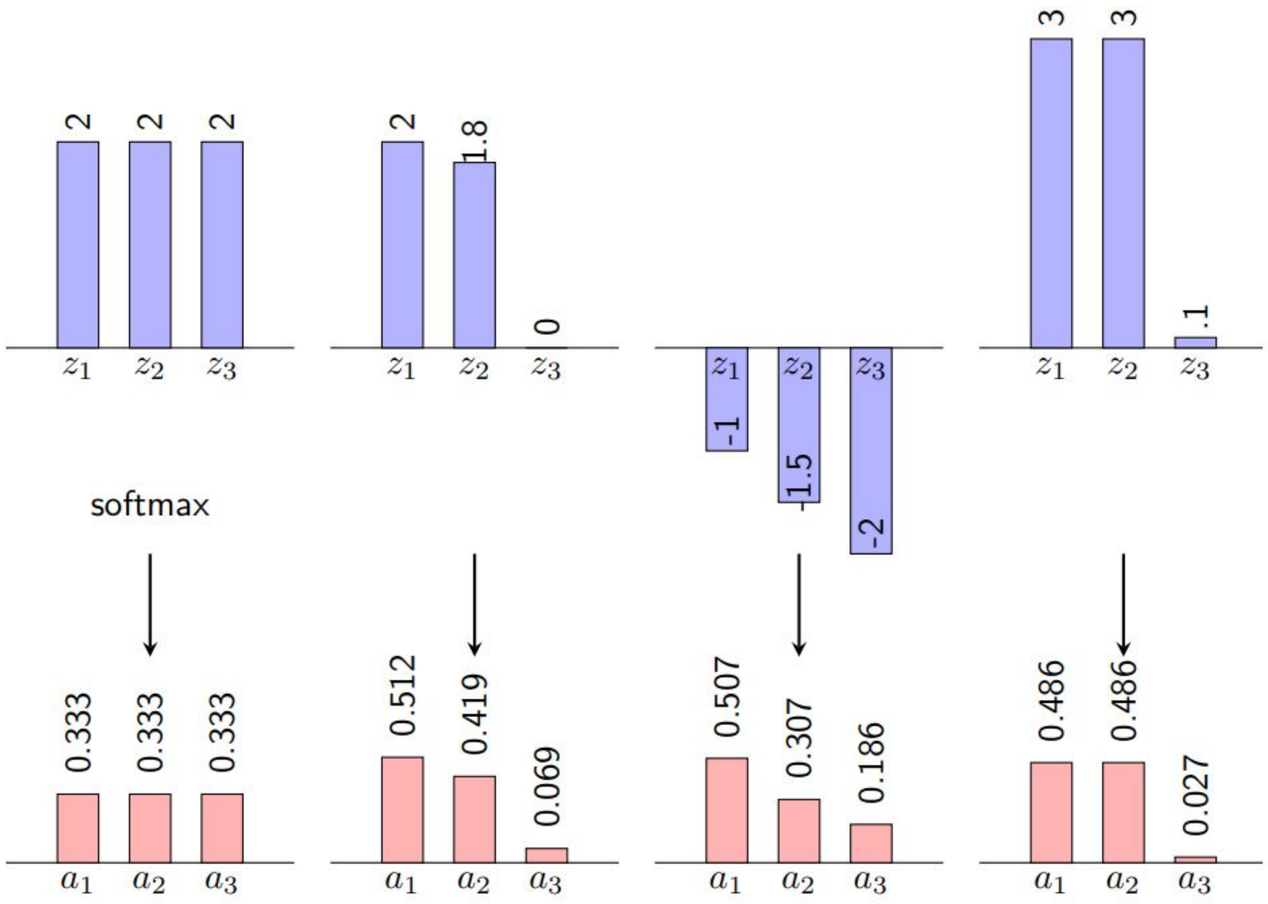
Cần tìm một hàm số khả vi, có giá trị dương, đồng biến, và tổng xác suất dự đoán là 1

$$z_i = \mathbf{w}_i^T \mathbf{x}$$

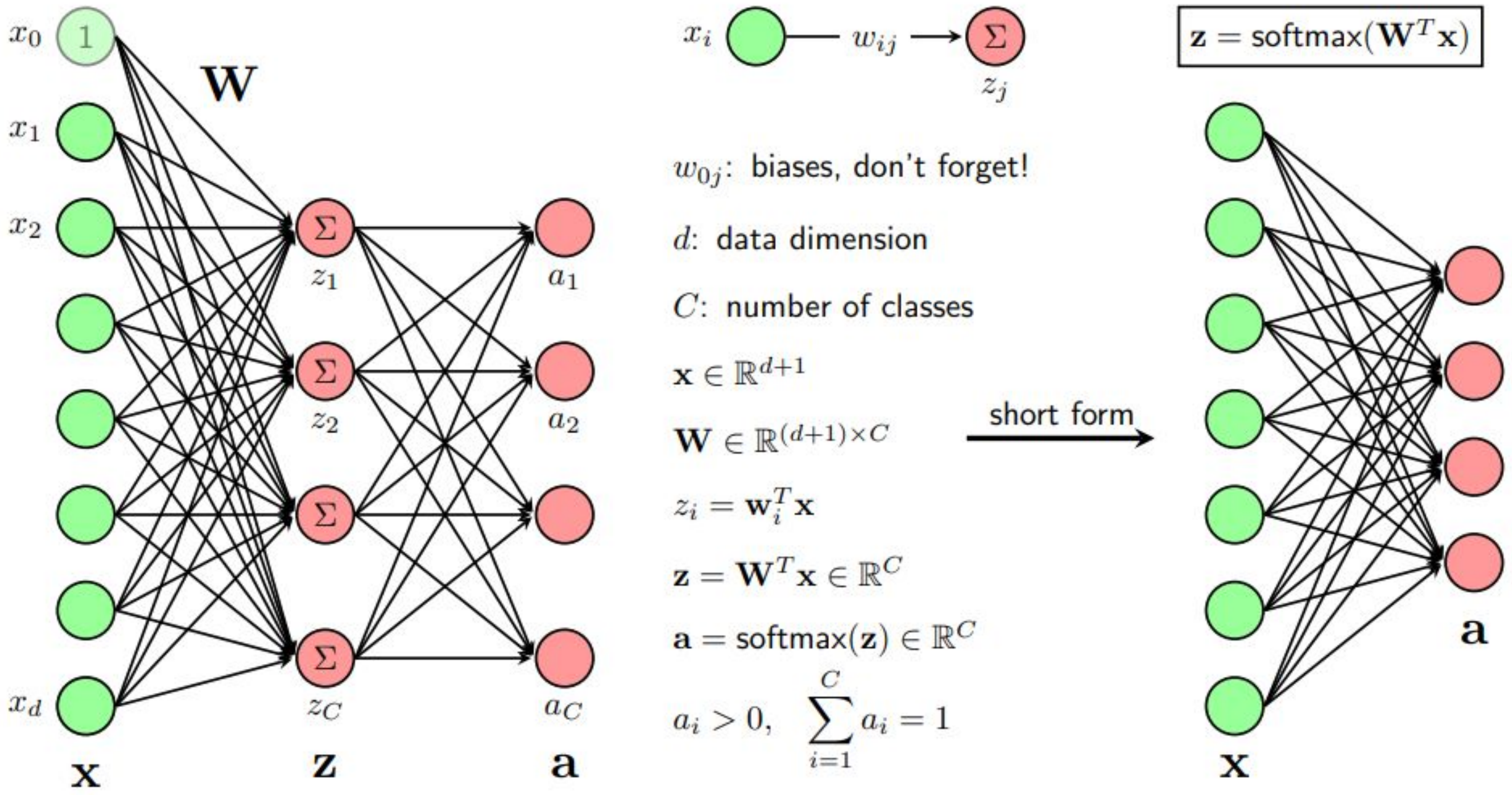
$$a_i = \frac{\exp(z_i)}{\sum_{j=1}^{C} \exp(z_j)}, \quad \forall i = 1, 2, \dots, C$$

$$p(y_k = i | \mathbf{x}_k; \mathbf{W}) = a_i$$

Một số ví dụ về input và output của Softmax



Mô hình softmax regression



Cross entropy

Cross entropy giữa hai vector phân phối p và q rời rạc được định nghĩa bởi

$$H(\mathbf{p}, \mathbf{q}) = -\sum_{i=1}^{C} p_i \log q_i$$

Xây dựng hàm mất mát

Trong trường hợp có C lớp dữ liệu, mất mát giữa đầu ra dự đoán và đầu ra thực sự của một điểm dữ liệu xi với label (one-hot) yi được tính bởi

$$J_i(\mathbf{W}) \triangleq J(\mathbf{W}; \mathbf{x}_i, \mathbf{y}_i) = -\sum_{j=1}^{C} y_{ji} \log(a_{ji})$$

$$J_i(\mathbf{W}) = -\log(a_{y_i,i})$$

Kết hợp tất cả các cặp dữ liệu xi , yi , i = 1, 2, . . . , N, hàm mất mát cho softmax regression được xác đinh bởi

$$J(\mathbf{W}; \mathbf{X}, \mathbf{Y}) = -\frac{1}{N}\sum_{i=1}^{N} \log(a_{y_i,i})$$

Tránh overfitting

$$\bar{J}(\mathbf{W}; \mathbf{X}, \mathbf{Y}) = -\frac{1}{N}\left(\sum_{i=1}^{N} \log(a_{y_i,i}) + \frac{\lambda}{2}||\mathbf{W}||_F^2\right)$$

Hàm mất mát với một điểm dữ liệu

$$
\begin{aligned}
J_i(\mathbf{W}) = -\sum_{j=1}^{C} y_{ji} \log(a_{ji}) &= -\sum_{j=1}^{C} y_{ji} \log\left(\frac{\exp(\mathbf{w}_j^T\mathbf{x}_i)}{\sum_{k=1}^{C}\exp(\mathbf{w}_k^T\mathbf{x}_i)}\right) \\
&= -\sum_{j=1}^{C}\left(y_{ji}\mathbf{w}_j^T\mathbf{x}_i - y_{ji}\log\left(\sum_{k=1}^{C}\exp(\mathbf{w}_k^T\mathbf{x}_i)\right)\right) \\
&= -\sum_{j=1}^{C} y_{ji}\mathbf{w}_j^T\mathbf{x}_i + \log\left(\sum_{k=1}^{C}\exp(\mathbf{w}_k^T\mathbf{x}_i)\right)
\end{aligned}
$$

Đạo hàm của hàm mất mát

- Công thức tính đạo hàm

$$\nabla_{\mathbf{W}} J_i(\mathbf{W}) = \left[\nabla_{\mathbf{w}_1} J_i(\mathbf{W}), \nabla_{\mathbf{w}_2} J_i(\mathbf{W}), \dots, \nabla_{\mathbf{w}_C} J_i(\mathbf{W})\right]$$

Đạo hàm của hàm mất mát

Gradient theo từng cột của wj

$$
\begin{aligned}
\nabla_{\mathbf{w}_j} J_i(\mathbf{W}) &= -y_{ji}\mathbf{x}_i + \frac{\exp(\mathbf{w}_j^T \mathbf{x}_i)}{\sum_{k=1}^{C} \exp(\mathbf{w}_k^T \mathbf{x}_i)} \mathbf{x}_i \\
&= -y_{ji}\mathbf{x}_i + a_{ji}\mathbf{x}_i = \mathbf{x}_i(a_{ji} - y_{ji}) \\
&= e_{ji}\mathbf{x}_i \ (\text{với } e_{ji} = a_{ji} - y_{ji})
\end{aligned}
$$

Đạo hàm của hàm mất mát

- Với ei=ai-yi, E=A-Y là sai khác giữa đầu ra dự đoán và đầu ra thực sự, ta có:

$$\nabla_{\mathbf{W}} J_i(\mathbf{W}) = \mathbf{x}_i [e_{1i}, e_{2i}, \dots, e_{Ci}] = \mathbf{x}_i \mathbf{e}_i^T$$

$$\Rightarrow \nabla_{\mathbf{W}} J(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \mathbf{e}_i^T = \frac{1}{N} \mathbf{X}\mathbf{E}^T$$

Mini-batch gradient descent

- Giả sử kích thước batch là k
- Kí hiệu , Nb là kích thước của mỗi batch

$$\mathbf{X}_b \in \mathbb{R}^{d\times k}, \mathbf{Y}_b \in \{0,1\}^{C\times k}, \mathbf{A}_b \in \mathbb{R}^{C\times k}$$

Công thức cập nhật W:

$$\mathbf{W} \leftarrow \mathbf{W} - \frac{\eta}{N_b}\mathbf{X}_b\mathbf{E}_b^T$$