AI and Deep Learning

2-7 Softmax Regression

Zhonglei Wang

WISE, SOE, CHOW

XMU

(Version v1)

Contents

Introduction

1. Binary classification problems
   - $y \in \{0, 1\}$
   - Model is built for $P(y = 1 \mid \boldsymbol{x})$
   - Loss function: Cross entropy
2. In practice, we may have $K(> 2)$ classes: $y \in \{0, 1, \ldots, K-1\}$
   - Consider one-hot representation
   - $\boldsymbol{y} = (0, \ldots, 1, \ldots, 0)^{\mathrm{T}} \in \mathbb{R}^{K \times 1}$
   - If $y = k$, only the $(k+1)$th element of $\boldsymbol{y}$ is 1

Model

1. For a feature $\boldsymbol{x}$,
   - We consider a neural network with K outputs
   - Each output is a score (conditional probability) for the corresponding class
2. That is, the only difference is the number of neurons in the last layer
   - For binary classification problems, we only have one neuron for the output layer
   - For general classification problems, we have $d^{[L]} = K$ neurons for the output layer

Softmax function

1. For a d-dimensional vector $\boldsymbol{z} = (z_1, \ldots, z_d)^{\mathrm{T}} \in \mathbb{R}^{d \times 1}$, the softmax function is

$$\mathrm{Softmax}(\boldsymbol{z}) = \begin{pmatrix} \exp(z_1) / \sum_{i=1}^{d} \exp(z_i) \\ \exp(z_2) / \sum_{i=1}^{d} \exp(z_i) \\ \vdots \\ \exp(z_d) / \sum_{i=1}^{d} \exp(z_i) \end{pmatrix}$$

2. Remarks

- Amplify gaps among $\{z_1, \ldots, z_d\}$ due to the exponential function
- Location invariant: $\mathrm{Softmax}(\boldsymbol{z} + c) = \mathrm{Softmax}(\boldsymbol{z})$ for all $c \in \mathbb{R}$
- Only consider relative order

Softmax function

1. Denote $\boldsymbol{a} = \text{Softmax}(\boldsymbol{z}) \in \mathbb{R}^{d\times 1}$
   - $\boldsymbol{a} = (a_1, \ldots, a_d)^{\mathrm{T}}$
2. Then, we have (check by yourself)

$$\frac{\partial \boldsymbol{a}^{\mathrm{T}}}{\partial \boldsymbol{z}} = \left(\frac{\partial a_m}{\partial z_k}\right) = (a_m \cdot (\delta_{mk} - a_k))$$

   - $k = 1, \ldots, d$: Row index
   - $m = 1, \ldots, d$: Column index
   - δ_{mk} : Indicator function

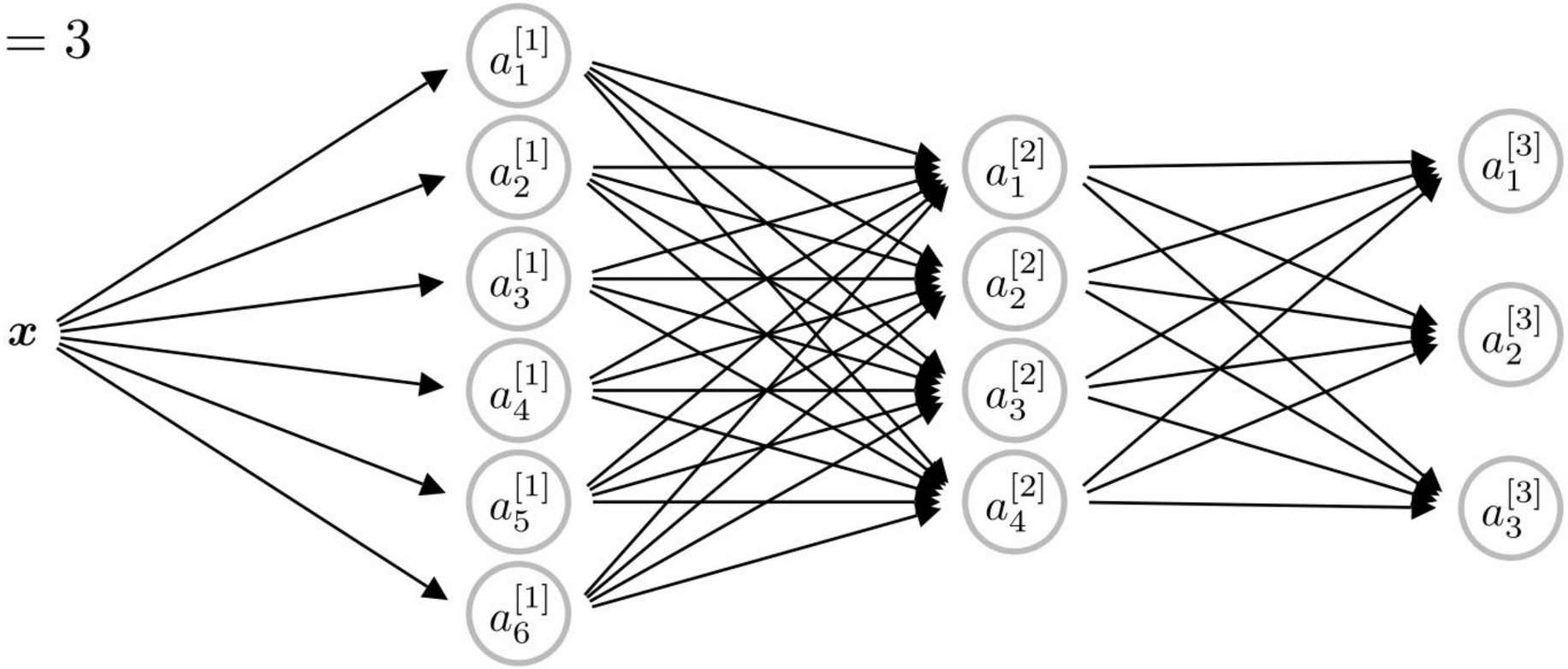
$$\delta_{mk} = \begin{cases} 1, & m = k \\ 0, & m \neq k \end{cases}$$

Notations

1. Recall that
   - L : Number of layers in the neural network
   - $d^{[l]}$: Number of neurons in the lth layer $\quad (l = 0, \ldots, L)$
   - $\boldsymbol{a}^{[l]} = (a_1^{[l]}, \ldots, a_{d^{[l]}}^{[l]})^{\mathrm{T}} \in \mathbb{R}^{d^{[l]} \times 1}$
   - $\boldsymbol{W}^{[l]} = (\boldsymbol{w}_1^{[l]}, \ldots, \boldsymbol{w}_{d^{[l]}}^{[l]})^{\mathrm{T}} \in \mathbb{R}^{d^{[l]} \times d^{[l-1]}}$
   - $\boldsymbol{b}^{[l]} = (b_1^{[l]}, \ldots, b_{d^{[l]}}^{[l]})^{\mathrm{T}} \in \mathbb{R}^{d^{[l]} \times 1}$
2. For binary classification problems, we have $d^{[L]} = 1$
3. For general softmax regression problems, we have $d^{[L]} = K$

Notations

1. For $K = 3$



Input layer $(d^{[0]} = d)$

1st hidden layer $(d^{[1]} = 6)$

2nd hidden layer $(d^{[2]} = 4)$

Output layer $(d^{[3]} = 3)$

Forward propagation

1. Let $\boldsymbol{A}^{[0]} = \boldsymbol{X} \in \mathbb{R}^{n \times d}$
2. For $l = 1, \ldots, L-1$,

$$\boldsymbol{Z}^{[l]} = \left(\boldsymbol{b}^{[l]}\right)^{\mathrm{T}} + \boldsymbol{A}^{[l-1]} \cdot \left(\boldsymbol{W}^{[l]}\right)^{\mathrm{T}},$$
$$\boldsymbol{A}^{[l]} = \sigma^{[l]}\left(\boldsymbol{Z}^{[l]}\right)$$

- $\sigma^{[l]}(z)$: Activation function for the lth layer
- Broadcasting is used for activation functions

Forward propagation

1. For the last layer,

$$\boldsymbol{W}^{[L]} \in \mathbb{R}^{K \times d_{L-1}}, \quad \boldsymbol{b} \in \mathbb{R}^{K \times 1}$$

2. Linear transformation is the same:

$$\boldsymbol{Z}^{[L]} = \left(\boldsymbol{b}^{[L]}\right)^{\mathrm{T}} + \boldsymbol{A}^{[L-1]} \cdot \left(\boldsymbol{W}^{[L]}\right)^{\mathrm{T}}$$

   - $\boldsymbol{Z}^{[L]} \in \mathbb{R}^{n \times K}$

3. Softmax function is applied for activation:

$$\boldsymbol{A}^{[L]} = \mathrm{Softmax}(\boldsymbol{Z}^{[L]})$$

   - $\boldsymbol{A}^{[L]} \in \mathbb{R}^{n \times K}$
   - Softmax function is applied to each row of $\boldsymbol{Z}^{[L]}$
   - The sum of each row of $\boldsymbol{A}^{[L]}$ is 1

Forward propagation

1. The dimension of the $\boldsymbol{A}^{[L]}$, containing estimated probabilities in the last layer
   - $\boldsymbol{A}^{[L]} \in \mathbb{R}^{n \times 1}$ for binary classification problems
   - $\boldsymbol{A}^{[L]} \in \mathbb{R}^{n \times K}$ for softmax regression problems
2. The cost function for softmax regression is

$$\mathcal{J}(\boldsymbol{\theta}) = -n^{-1} \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log a_{ik}^{[L]}$$

   - $\boldsymbol{\theta}$: Model parameters
   - $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{iK})^{\mathrm{T}}$: One-hot representation for the ith example
   - $a_{ik}^{[L]}$: Estimated probablity for the $(k-1)$th class of the ith example

Backpropagation

1. We only need to focus on the backpropagation for the output layer
2. $\mathrm{d}\boldsymbol{A}^{[L]}$ can be obtained from the cost function
3. The cost function for softmax regression is

$$\mathcal{J}(\boldsymbol{\theta}) = -n^{-1} \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log a_{ik}^{[L]}$$

4. Thus, the (i,k)th component of $\mathrm{d}\boldsymbol{A}^{[L]}$is

$$\mathrm{d}a_{ik}^{[L]} = \frac{\partial \mathcal{J}}{\partial a_{ik}^{[L]}} = -n^{-1} \frac{y_{ik}}{a_{ik}^{[L]}}$$

5. $\mathrm{d}\boldsymbol{A}^{[L]} = -n^{-1}\boldsymbol{Y}/\boldsymbol{A}^{[L]}$ (element-wise division by Python broadcasting)

Backpropagation

1. Next, we focus on obtaining $\mathrm{d}\boldsymbol{Z}^{[L]}$
2. For $i = 1, \ldots, n$ and $k = 1, \ldots, K$, $z_{ik}^{[L]}$ is only used to get the ith row of $A^{[L]}$
3. For $h \neq i$, we have

$$\frac{\partial a_{hm}^{[L]}}{\partial z_{ik}^{[L]}} = 0 \quad (m = 1, \ldots, K)$$

4. For $i = 1, \ldots, n; k = 1, \ldots, K$, we have

$$\mathrm{d}z_{ik}^{[L]} = \frac{\partial \mathcal{J}}{\partial z_{ik}^{[L]}} = \sum_{h=1}^{n} \sum_{m=1}^{K} \frac{\partial \mathcal{J}}{\partial a_{hm}^{[L]}} \frac{\partial a_{hm}^{[L]}}{\partial z_{ik}^{[L]}} = \sum_{m=1}^{K} \frac{\partial \mathcal{J}}{\partial a_{im}^{[L]}} \frac{\partial a_{im}^{[L]}}{\partial z_{ik}^{[L]}}$$

5. To obtain $\mathrm{d}\boldsymbol{Z}^{[L]}$, it is enough to get its elements $\{\mathrm{d}z_{ik}^{[L]} : i = 1, \ldots, n; k = 1, \ldots, K\}$

Backpropagation

1. We have obtained

$$\frac{\partial \mathcal{J}}{\partial a_{im}^{[L]}} = -n^{-1}\frac{y_{im}}{a_{im}^{[L]}}, \quad \frac{\partial a_{im}^{[L]}}{\partial z_{ik}^{[L]}} = a_{im}^{[L]} \cdot (\delta_{mk} - a_{ik}^{[L]})$$

2. Thus, we have

$$\begin{aligned} \mathrm{d}z_{ik}^{[L]} &= \sum_{m=1}^{K} \frac{\partial \mathcal{J}}{\partial a_{im}^{[L]}} \frac{\partial a_{im}^{[L]}}{\partial z_{ik}^{[L]}} = -n^{-1} \sum_{m=1}^{K} \frac{y_{im}}{a_{im}^{[L]}} \cdot a_{im}^{[L]} \cdot (\delta_{mk} - a_{ik}^{[L]}) \\ &= -n^{-1} \sum_{m=1}^{K} y_{im} \cdot \delta_{mk} + n^{-1} a_{ik}^{[L]} \cdot \sum_{m=1}^{K} y_{im} = -n^{-1}(y_{ik} - a_{ik}^{[L]}) \end{aligned}$$

   - The last equality holds since $\sum_{m=1}^{K} y_{im} = 1$

3. Thus, we have $\mathrm{d}\boldsymbol{Z}^{[L]} = -n^{-1}(\boldsymbol{Y} - \boldsymbol{A}^{[L]})$ (error matrix from the output layer)

Backpropagation

1. To sum up, for the output layer, we have

$$
\begin{aligned}
\mathrm{d}\boldsymbol{A}^{[L]} &= -n^{-1}\boldsymbol{Y}/\boldsymbol{A}^{[L]} \quad \text{(element-wise division)}, \\
\mathrm{d}\boldsymbol{Z}^{[L]} &= -n^{-1}(\boldsymbol{Y}-\boldsymbol{A}^{[L]}) \quad \text{(exactly the same as binary models)}, \\
\mathrm{d}\boldsymbol{W}^{[L]} &= \left(\mathrm{d}\boldsymbol{Z}^{[L]}\right)^{\mathrm{T}} \cdot \boldsymbol{A}^{[L-1]}, \\
\mathrm{d}\boldsymbol{b}^{[l]} &= \left(\mathrm{d}\boldsymbol{Z}^{[l]}\right)^{\mathrm{T}} \cdot \mathbf{1}, \\
\mathrm{d}\boldsymbol{A}^{[L-1]} &= \mathrm{d}\boldsymbol{Z}^{[L]} \cdot \boldsymbol{W}^{[L]}
\end{aligned}
$$

- $\mathbf{1} \in \mathbb{R}^{n\times 1}$: Column vector of 1's

Backpropagation

1. The hidden layers for softmax regression is exactly the same as binary regression
2. We can directly use the backpropagation result for binary regression models
3. That is, by assuming $\mathrm{d}\boldsymbol{A}^{[l]}$ is available for $l = L-1, \ldots, 1$, we have

$$\mathrm{d}\boldsymbol{Z}^{[l]} = \mathrm{d}\boldsymbol{A}^{[l]} \circ \sigma^{[l]'}\left(\boldsymbol{Z}^{[l]}\right),$$

$$\mathrm{d}\boldsymbol{W}^{[l]} = \left(\mathrm{d}\boldsymbol{Z}^{[l]}\right)^{\mathrm{T}} \cdot \boldsymbol{A}^{[l-1]},$$

$$\mathrm{d}\boldsymbol{b}^{[l]} = \left(\mathrm{d}\boldsymbol{Z}^{[l]}\right)^{\mathrm{T}} \cdot \mathbf{1},$$

$$\mathrm{d}\boldsymbol{A}^{[l-1]} = \mathrm{d}\boldsymbol{Z}^{[l]} \cdot \boldsymbol{W}^{[l]}$$

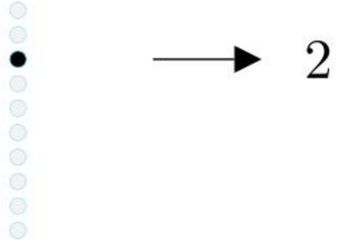
- $\mathbf{1} \in \mathbb{R}^{n \times 1}$: Column vector of 1's

Example

Test image

Model (FNN with 2 hidden layers)

Estimated result



Input layer	1st hidden layer	2nd hidden layer	Output layer
$d^{[0]} = 784$	$d^{[1]} = 128$	$d^{[2]} = 64$	$d^{[3]} = 10$

Summary

1. Four aspects for softmax regression

- Model structure: K outputs, estimating a conditional probability for each class
- Forward propagation: Hidden layers are the same, but apply softmax to the output layer
- Cost function: Cross entropy
- Backpropagation: Hidden layers are the same, and scaled estimated error for the output layer, $\mathrm{d}\boldsymbol{Z}^{[L]} = n^{-1}(\boldsymbol{A}^{[L]} - \boldsymbol{Y})$