Spring 2023

SWCON253: Machine Learning

Lecture 08
Multiclass Classification

Jinwoo Choi
Assistant Professor
CSE, Kyung Hee University

Disclaimer: Many slides have been borrowed from Hui Yong Kim, who may have borrowed some of them from others. Any time a slide does not show any explicit credit, I credit the slide to Hui Yong Kim.

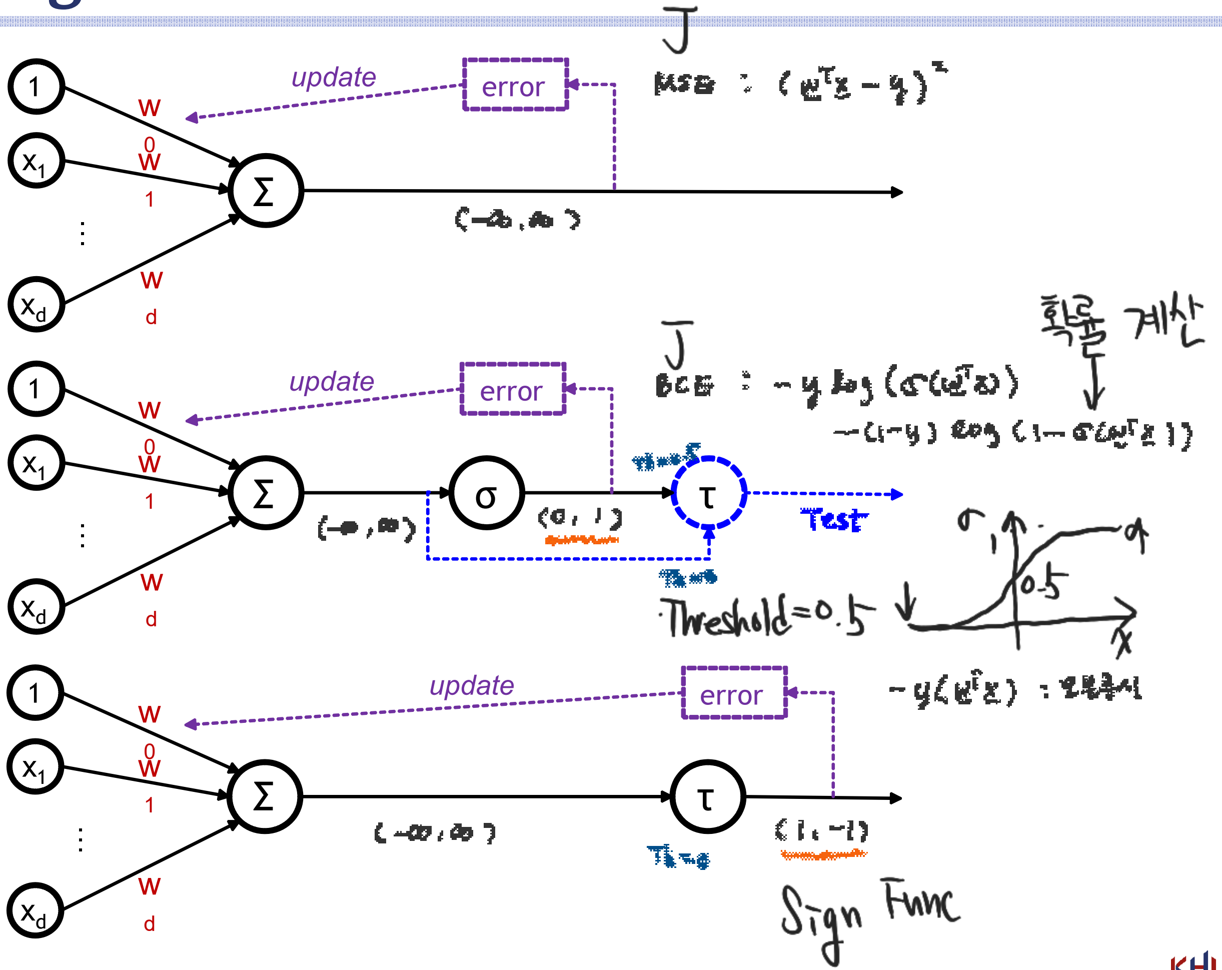
Contents

References

- *http://stat.wisc.edu/~sraschka/teaching*
- *https://en.wikipedia.org/wiki/Multiclass_classification*
- *https://en.wikipedia.org/wiki/One-hot*
- *기계 학습* by 오일석

(Recap) Training Single Neuron Models

- **Linear Regression :**
- **Logistic Regression :**
  - sigmoid
- **Perceptron :**
  - threshold



1. Multi-Class Classification

1. Categorical Data
2. One-Hot Encoding
3. Multiclass Classification
4. One-vs-Rest (One-vs-All) Method
5. One-vs-One Method

Categorical Data

- **Ordinal (서열) vs. Nominal (명목) Variables**
  - Ordinal: 값의 순서를 정할 수 있음 (크기/거리 가능)
    - ★ e.g. price, height, weight, image pixel values, …
  - Nominal: 값의 순서를 정할 수 없음 (크기/거리 불가)
    - ★ e.g. object category, blood type, roll of a die

정렬이 가능해야

- **Numerical vs. Categorical Variables**
  - Numerical (quantitative) variables:
    - ★ it has some *order* (thus can be continuous numbers) 순서 O
  - Categorical (qualitative) variables
    - ★ take on one of a limited number of possible values, assigning each individual observation to a particular *nominal category* on the basis of some qualitative property. 순서 X
    - ★ For ease in statistical processing, categorical variables may be assigned *numeric indices*.

- **Examples**
  - IRIS dataset
    - ★ in: length, width → numerical ≈ ordinal
    - ★ out: species categorical (nominal)
  - MNIST dataset
    - ★ in: pixel values → numerical
    - ★ out: digits → categorical (nominal)
  - ImageNet dataset
    - ★ in: pixel values → numerical
    - ★ out: object categories → categorical (nominal)

One-Hot Encoding

◆ One-Hot Code (One-Hot Vector)

- A group of bits among which the legal combinations of values are only those with ***a single high (1) bit*** and all the others low (0)
  - ★ A similar implementation in which all bits are '1' except one '0' is sometimes called "***one-cold code***". 하나만 0, 나머지 1
- One-hot Encoding is frequently used to deal with ***categorical data***
  - ★ because many ML models need variables to be numeric

Binary	Gray code	One-hot
000	000	00000001
001	001	00000010
010	011	00000100
011	010	00001000
100	110	00010000
101	111	00100000
110	101	01000000
111	100	10000000

◆ One-Hot Encoding in ML

- *k*번째 class의 target vector를 *k*번째 자리는 1, 나머지는 0이 되도록 설정

Index	Job
1	Police
2	Doctor
3	Student
4	Teacher
5	Driver

순서X

One hot encoded data
[1 0 0 0 0]
[0 1 0 0 0]
[0 0 1 0 0]
[0 0 0 1 0]
[0 0 0 0 1]

Multiclass Classification

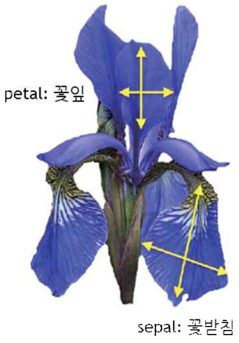
Multiclass (Multinomial) Classification

- The problem of classifying instances into one of three or more classes.
  - ★ The output is categorical (e.g., Iris, MNIST, ImageNet)

    ≈ categories

- $y \in \{1, 2, 3, \ldots, c\}$.

  numerical index



c = 3

c = 10

c = 21,841

(a) 'swing' 부류

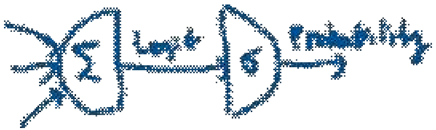
(b) 'Great white shark' 부류

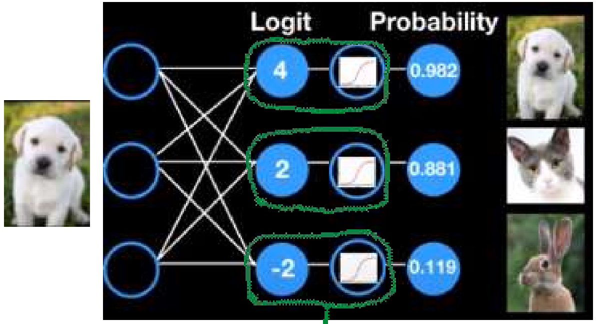
Multiclass Classification – *One-vs-Rest Method*

One-versus-Rest (One-versus-All) Method

- 이진 분류기 c개를 독립적으로 사용하여 class k와 나머지 c-1개 class를 분류 (1 : c-1)
- Class k에 대한 이진 분류기를 h_k라 하면, $h_k(\mathbf{x})$가 가장 큰 값을 갖는 k로 분류함

$$\boxed{\hat{k} = \arg\max_{k} h_k(\mathbf{x})}$$





dog vs. rest

cat vs. rest

rabbit vs. rest

a binary classifier (e.g. logistic regression)

Remarks

- 각 이진 분류기에 대해 훈련집합의 불균형을 일으킴 (class k 샘플수 ≪ 나머지 샘플수)

Multiclass Classification – *One-vs-One Method*

- One-versus-One Method C개 중 2개를 고르는 경우의 수
  - 이진 분류기 C(c, 2)개를 독립적으로 사용하여 class k와 class l을 분류 (1 : 1)
    - ★ $C(c,2) = \dfrac{c!}{(c-2)!\,2!} = c(c-1)/2$ Big O-Notation $O(c^2)$

① dog vs. cat
② cat vs. rabbit
③ rabbit vs. dog

  - 가장 많은 이진 분류기가 선택(투표)한 class를 최종 결과로 결정
    - ★ Class k와 l을 비교하는 이진 분류기를 $h_{(k,l)}(\mathbf{x})$라 하자.
    - ★ $h_{(k,l)}(\mathbf{x})$가 class k(또는 l)를 출력하면, class k(또는 l)에 한 표를 추가.
    - ★ C(c, 2)개 이진 분류기에 대해 가장 많은 표를 획득한 class를 최종 결과로 결정 (최대 표의 개수: c-1)
      - 비유) 야구나 축구 리그에서 가장 승리를 많이 한 팀이 우승

Testing 복잡도도 높음

- Remarks

One versus Rest의 문제 해결

  - 훈련집합의 불균형을 일으키지 않음: class k 샘플수 ≈ class l 샘플수
  - 사용되는 이진 분류기의 개수가 c^2에 비례: 높은 training/testing 복잡도

One Versus Rest는 $O(c)$

2. Softmax Classifier

Classifier 해나로 해결
확률을 전부 더하면 1

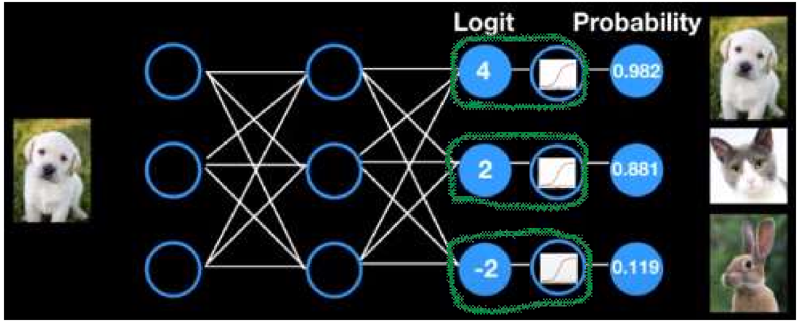
1. Softmax – Motivation & Definition
2. One-Hot Encoding
3. Cross-Entropy Loss
4. Training Softmax Classifier with CE

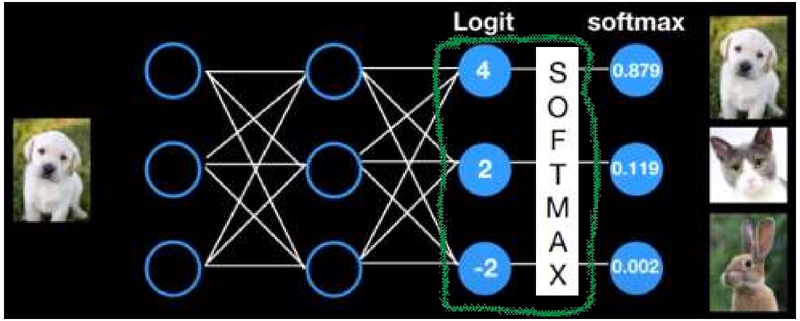
Softmax – *Motivation*

- What is the best way to convert (-∞, ∞) to probability for multiclass classification?
  - ***What we want*** in the output layer
    - ★ conditional probabilities $p(y|x)$
  - ***Sigmoid*** activations in the output layer
    - ★ do ***not*** sum up to 1

확률 합을 1로 만들어줌

  - ***Softmax*** activations in the output layer
    - ★ ***do*** sum up to 1
    - ★ suits well to *Cross-Entropy Loss*





Figures (modified): https://www.youtube.com/watch?v=K7HTd_Zgr3w

Softmax – *Definition*

https://en.wikipedia.org/wiki/Softmax_function

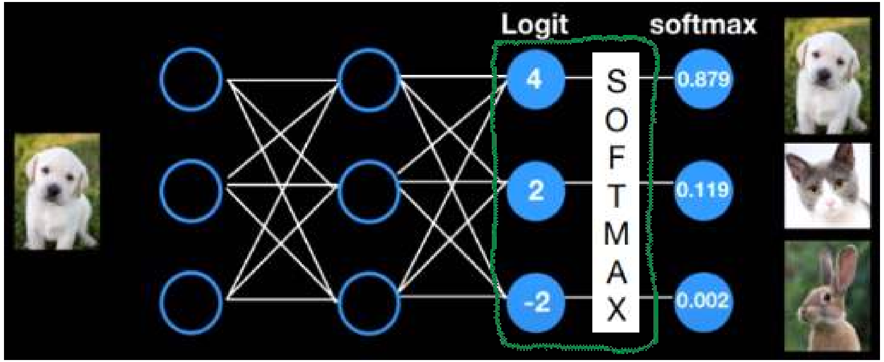
- **Softmax Function** → 확률 분포로 만들어줌
  - Takes as input a vector **z** of *K* real numbers,
  - and normalizes it into a probability distribution consisting of *K* probabilities

$$\sigma : \mathbb{R}^K \to (0,1)^K$$

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad \text{for } i = 1, \dots, K \text{ and } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K$$

$$\mathbf{z} = \begin{bmatrix} 4 \\ 2 \\ -2 \end{bmatrix}$$

$$\sigma(\mathbf{z}) = \begin{bmatrix} e^4 \\ e^2 \\ e^{-2} \end{bmatrix} / (e^4 + e^2 + e^{-2})$$



argmax

Cross-Entropy Loss

- **Assume One-Hot Encoding**
  - y's are either 0 or 1 정답만 1

- ***Binary CE*** **(for binary classification)**

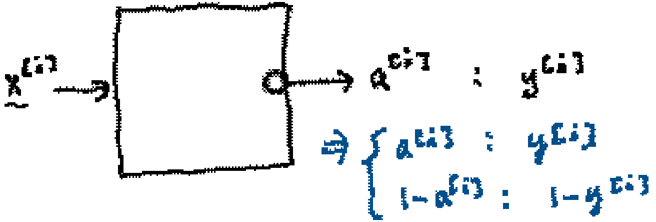
$$-\frac{1}{n}\sum_{i=1}^{n}\left(y^{[i]}\log(a^{[i]}) + (1-y^{[i]})\log(1-a^{[i]})\right)$$

↓ 샘플수

→ y가 0 아니면 1

↓ 일반화

$k=2$



- ***Multinomial CE*** **(for multiclass classification)** ex) $k=3$

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} -y_k^{[i]}\log\left(a_k^{[i]}\right)$$

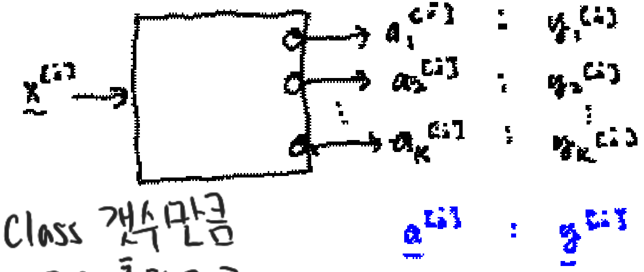
↓ 샘플수 ↓ 클래스수

n : # of training samples

K : # of classes

$$\begin{array}{ccc} y_3 & y_2 & y_1 \end{array}$$
$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

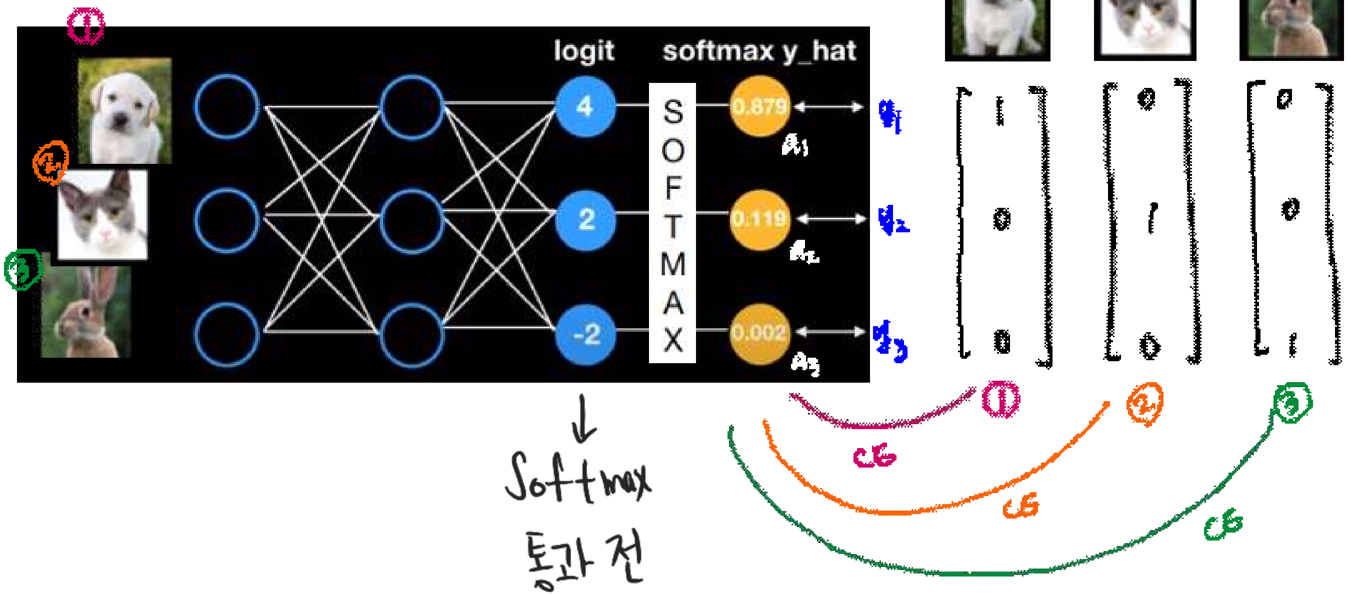
Class 갯수만큼 확률을 종합하고 평균을 구함



Training Softmax Classifier with CE

- Cross-Entropy Loss with One-Hot Encoded Targets

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} -y_k^{[i]} \log\left(a_k^{[i]}\right)$$

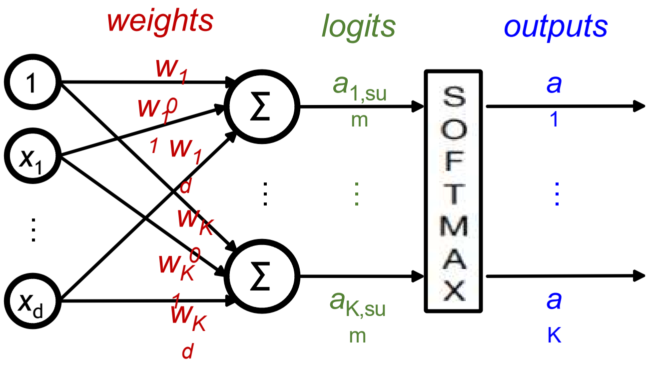


Training Softmax Classifier with CE (cont'd)

Gradient of the Cross-Entropy Loss at the Output Layer

$z_k = a_{k,sum} = \sum_{i=1}^{n} W_{ki} x_i = W_k^T x^{(n)}$

$\frac{\partial z_k}{\partial W_k} = x^{(n)}$



- Cross-Entropy Loss

$$J(\mathbf{W}) = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{K} -y_i^{(n)} \log a_i^{(n)}$$

$$= \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{K} -y_i^{(n)} \log \frac{\exp(\mathbf{w}_i^T\mathbf{x}^{(n)})}{\sum_{j=1}^{K}\exp(\mathbf{w}_j^T\mathbf{x}^{(n)})}$$

- It's Gradient

입력벡터와 오차의 곱

$$\frac{\partial J(\mathbf{W})}{\partial \mathbf{w}_k} = \frac{1}{N}\sum_{n=1}^{N}\left(a_k^{(n)} - y_k^{(n)}\right)\mathbf{x}^{(n)}$$

$$= \frac{1}{N}\sum_{n=1}^{N}\left(\log \frac{\exp(\mathbf{w}_k^T\mathbf{x}^{(n)})}{\sum_{j=1}^{K}\exp(\mathbf{w}_j^T\mathbf{x}^{(n)})} - y_k^{(n)}\right)\mathbf{x}^{(n)}$$

$$\frac{\partial \log a_i}{\partial z_k} = \frac{\partial}{\partial z_k}\log\frac{\exp(z_i)}{\sum_{j=1}^{K}\exp(z_j)}$$

$$= \frac{\partial}{\partial z_k}\left(\log\exp(z_i) - \log\textstyle\sum_{j=1}^{K}\exp(z_j)\right)$$

$$= 1\{i=k\} - \frac{\exp(z_k)}{\sum_{j=1}^{K}\exp(z_j)} = 1\{i=k\} - a_k$$

$$\therefore \sum_{i=1}^{K} -y_i\frac{\partial \log a_i}{\partial z_k} = \sum_{i=1}^{K} -y_i(1\{i=k\} - a_k)$$

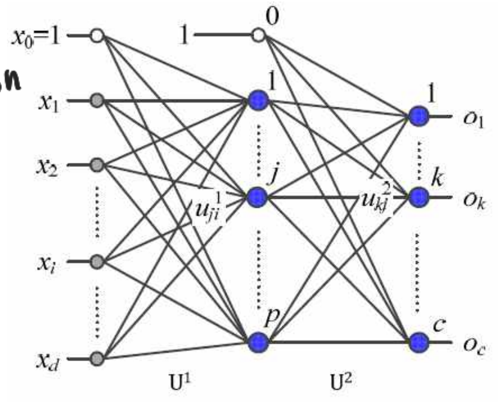
$$= -\left(y_k - \left(\textstyle\sum_{i=1}^{K} y_i\right)a_k\right) = a_k - y_k$$

$$\underbrace{\textstyle\sum_{i=1}^{K} y_i}_{1}$$

3. Gradient of MSE & CE Losses

Gradient of MSE & CE Losses

- For a *linear* neuron (i.e., neuron without activation = Linear Regressor):
  - Gradient of **MSE**: (output – label) ·(input_i) Linear Regression



- For a *non-linear* neuron (e.g., Logistic Regression or Perceptron)
  - ~~Gradient of **MSE**: (output – label) · $\sigma'()$ ·(input_i)~~
  - Gradient of **CE**: (output – label) · (input_i)
- For a neural network (of *non-linear* neurons)
  - Gradient of **MSE** $(y_k - o_k)\sigma_k'()\cdot(x_j)$
    - ★ For k^{th} output neuron: $(\text{output}_k - \text{label}_k)\cdot\sigma_k'()\cdot(\text{input}_j) = (\text{delta}_k)\cdot(\text{input}_j)$
    - ★ For j^{th} intermediate neuron: $\left(\sum_{\forall k \in \{\text{next layer neurons}\}}(\text{delta}_k)\cdot(\text{weight}_{kj})\right)\cdot\sigma_j'()\,(\text{input}_i) = (\text{eta}_j)\cdot(\text{input}_i)$
  - Gradient of **CE** (with Softmax output)
    - ★ For k^{th} Softmax output: $(\text{output}_k - \text{label}_k)\cdot(\text{input}_j)$