
Deep Learning Assignment 3

Anonymous Author(s)

Affiliation

Address

email

1 General Questions

2 Softmax regression gradient calculation

Given

$$\hat{y} = \sigma(Wx + b), \text{ where } x \in \mathbb{R}^d, W \in \mathbb{R}^{k \times d}, b \in \mathbb{R}^k \quad (1)$$

where d is the input dimension, k is the number of classes, σ is the softmax function:

$$\sigma(a)_i = \frac{\exp(a_i)}{\sum_j \exp(a_j)} \quad (2)$$

Which means a given input x will output y with probability of each class

2.1 Derive $\frac{\partial l}{\partial W_{ij}}$

If the given cross-entropy loss defined as followed:

$$l(y, \hat{y}) = - \sum_i y_i \log \hat{y}_i \quad (3)$$

As W_{ij} will affect the prediction of class i by multiplying index j in x , therefore we can derive:

$$\frac{\partial l}{\partial W_{ij}} = \frac{\partial l}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial W_{ij}} \quad (4)$$

where:

$$l(y, \hat{y}) = - \sum_i y_i \log \hat{y}_i = -(y_1 \log \hat{y}_1 + y_2 \log \hat{y}_2 + \dots + y_i \log \hat{y}_i + \dots) \quad (5)$$

and therefore

$$\frac{\partial l}{\partial \hat{y}_i} = \frac{-y_i}{\hat{y}_i \ln 10} \quad (6)$$

And we can rewrite (1) and (2) and care the value only for \hat{y}_i :

$$\hat{y}_i = \frac{e^{(W_{ij}X_j + b_i)}}{C + e^{(W_{ij}X_j + b_i)}}, \text{ where } C = \sum_k e^{W_{ik}X_k + b_i} - e^{(W_{ij}X_j + b_i)} \quad (7)$$

$$\frac{\partial \hat{y}_i}{\partial W_{ij}} = \frac{X_j e^{(W_{ij}X_j + b_j)}}{C + e^{(W_{ij}X_j + b_j)}} - \frac{X_j e^{2(W_{ij}X_j + b_j)}}{(C + e^{(W_{ij}X_j + b_j)})^2} \quad (8)$$

Combining (6) and (8) to (4), and we will get the result

14 **3 Chain rule**

15 **4 Variants of pooling**

16 **5 Convolution**

17 (a) As it is using 3x3 kernal along x and y axis of input, which is 5 and 5 respectively. The output of
18 this layer will be $(5 - 3 + 1) \times (5 - 3 + 1)$ which is 3x3.

19 (b) Assuming the kernel operation is point-point multiplication and summation, then the output of
20 this layer is:

21
$$\begin{pmatrix} 109 & 92 & 72 \\ 108 & 85 & 74 \\ 110 & 74 & 79 \end{pmatrix}$$

22 (c)
$$\begin{pmatrix} 4 & 7 & 10 & 6 & 3 \\ 9 & 17 & 25 & 16 & 8 \\ 11 & 23 & 34 & 23 & 11 \\ 7 & 16 & 24 & 17 & 8 \\ 2 & 6 & 9 & 7 & 3 \end{pmatrix}$$

23

24 **6 Optimization**

25 **7 Top-k error**

26 **8 t-SNE**

27 **9 Proximal gradient descent**