

---

# Deep Learning Assignment 3

---

Anonymous Author(s)

Affiliation

Address

email

## 1 General Questions

(a) Say if the first module is:

$$\max(W_1 X) \quad (1)$$

where the  $W$  input layer maybe doing summation and summation just like matrix multiplication does  $WX$ , and the  $\max$  function is a non-linear active function modifying the value like a neuron does before entering the next module:

$$W_2(\max(W_1 X)) \quad (2)$$

If now we don't have the active function then the formula will look like:

$$W_2(W_1 X) \rightarrow \bar{W} X \quad (3)$$

which eventually all  $W_i$  can become a single module  $\bar{W}$

## 2 Softmax regression gradient calculation

Given

$$\hat{y} = \sigma(Wx + b), \text{ where } x \in \mathbb{R}^d, W \in \mathbb{R}^{k \times d}, b \in \mathbb{R}^k \quad (4)$$

where  $d$  is the input dimension,  $k$  is the number of classes,  $\sigma$  is the softmax function:

$$\sigma(a)_i = \frac{\exp(a_i)}{\sum_j \exp(a_j)} \quad (5)$$

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:

$$l(y, \hat{y}) = - \sum_i y_i \log \hat{y}_i \quad (6)$$

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

$$\frac{\partial l}{\partial W_{ij}} = \frac{\partial l}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial W_{ij}} \quad (7)$$

where:

$$l(y, \hat{y}) = - \sum_i y_i \log \hat{y}_i = -(y_1 \log \hat{y}_1 + y_2 \log \hat{y}_2 + \dots + y_i \log \hat{y}_i + \dots) \quad (8)$$

and therefore

$$\frac{\partial l}{\partial \hat{y}_i} = \frac{-y_i}{\hat{y}_i} \quad (9)$$

17 And we can rewrite for only for  $\hat{y}_i$ :

$$\hat{y}_i = \frac{\exp(a_i)}{\sum_j \exp(a_j)} = \frac{\exp(a_i)}{C + \exp(a_i)}, \text{ where } C = \sum_{k \neq i} \exp(a_k) \quad (10)$$

18 Since

$$\frac{\partial \exp(a_i)}{\partial W_{ij}} = W_{ij} \exp(a_i) \quad (11)$$

19 Therefore

$$\frac{\partial \hat{y}_i}{\partial W_{ij}} = W_{ij} \hat{y}_i (1 - \hat{y}_i) \quad (12)$$

20 Finally, we will get the result of  $\frac{\partial l}{\partial W_{ij}}$ :

$$\frac{\partial l}{\partial W_{ij}} = \frac{\partial l}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial W_{ij}} = -X_j y_i (1 - \hat{y}_i) \quad (13)$$

21 **2.2 What happen when  $y_{c_1} = 1, \hat{y}_{c_2} = 1, c_1 \neq c_2$**

22 This means something like  $y = [1, 0, 0]^T$  and  $\hat{y} = [0, 0, 1]^T$ , and the predict is far different from true  
 23 lable. This will cause the log part in loss (3) become negative infinity. We may not need to worry this  
 24 because before one of the class predicted close to 1 and everything else close to 0, it will generate a  
 25 great positive loss the the class that is miss-predicted trying to make the predict right to true label.

## 26 **3 Chain rule**

## 27 **4 Variants of pooling**

## 28 **5 Convolution**

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

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

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

33

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

34 (c)

35

## 36 **6 Optimization**

## 37 **7 Top-k error**

## 38 **8 t-SNE**

## 39 **9 Proximal gradient descent**

40 (a) Since Proximal operator is defined as:

$$\text{prox}_{h,t}(x) = \underset{z}{\operatorname{argmin}} \frac{1}{2} \|z - x\|_2^2 + th(z) \quad (14)$$

41 which the optimal condition is to have the gradient w.r.t  $z$  equal to 0:

$$0 \in z - x + t\partial h(z) \quad (15)$$

42 if function  $h(z) = \|z\|_1$  and  $z_i \neq 0$ , then:

$$\partial h(z) = \text{sign}(z) \quad (16)$$

43 And therefore the optimal solution  $z^*$  will be:

$$z^* = x - t \cdot \text{sign}(z^*) \quad (17)$$

44 Noted that if  $z_i^* < 0$ , then  $x_i < -\lambda$ , and if  $z_i^* > 0$ , then  $x_i > \lambda$ . This implies  $|x_i| > \lambda$  and  
 45  $\text{sign}(z_i^*) = \text{sign}(x_i)$ , and we can rewrite formula to:

$$z_i^* = x_i - t \cdot \text{sign}(x_i) \quad (18)$$

46 Then if the solution  $z_i^* = 0$ , the subgradient of l1-norm is in the interval of  $[-1, 1]$ , and we can write:

$$0 \in -x_i + t \cdot [-1, 1] \implies x_i \in [-t, t] \implies |x_i| \leq t \quad (19)$$

47 Therefore the solution of Proximal operator will be:

$$z_i^* = \begin{cases} 0 & \text{if } |x_i| \leq t \\ x_i - t \cdot \text{sign}(x_i) & \text{if } |x_i| > t \end{cases} \quad (20)$$

48 which is

$$\text{prox}_{h,t}(x) = S_t(x) = (|x| - t)_+ \odot \text{sign}(x) \quad (\text{element-wise}) \quad (21)$$

49 which is a soft-threshold function with  $t$  as threshold value

50 (b) In the field of signal processing, the true signal usually will be blurred as followed:

$$Ax = b \quad (22)$$

51 where  $A$  is the blur operation,  $b$  is the known observed blurred-signal. The way to solve true signal  $x$   
 52 is called deblurring problem:

$$\min_x \{F(x) \equiv \frac{1}{2} \|b - Ax\|_2^2 + \lambda \|x\|_1\} \quad (23)$$

53 This is ISTA problem, and as we can see the first term is convex and differentiable, and the second  
 54 term is convex and simple l1-norm function. Then the ISTA is become one example of proximal  
 55 gradient descent