Deep Learning Assignment 3

Anonymous Author(s)
Affiliation
Address
`email`

1 General Questions

2 Softmax regression gradient calculation

Given

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

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

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

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:

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

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

$$\frac{\partial l}{\partial W_{ij}} = \frac{\partial l}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial W_{ij}} \tag{4}$$

where:

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

and therefore

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

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

$$\hat{y_i} = \frac{e^{(W_{ij}X_j+b_i)}}{C+e^{(W_{ij}X_j+b_i)}}\text{, **where** } C = \textstyle\sum_k e^{W_{ik}X_k+b_i} - e^{(W_{ij}X_j+b_i)} \tag{7}$$

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

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

3 Chain rule

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

Submitted to 29th Conference on Neural Information Processing Systems (NIPS 2016). Do not distribute.

4 Variants of pooling

5 Convolution

6 Optimization

7 Top-k error

8 t-SNE

9 Proximal gradient descent