神经网络的梯度反向传播的数学推导

全连接层

简单全连接网络的反向传播过程，

设第 l 层，

$$z^l = W^l \cdot x^{l-1} + b^l \qquad x^{l-1} \in R^m$$

$$x^l = f(z^l) \qquad x^l \in R^n$$

$$W^l \in R^{n \times m}$$

其中 f 是非线性激励函数，

$J = J(W, b)$ 是损失函数，x^{l-1} 是第 $l-1$ 层的输出。

记 $\delta^l = \frac{\partial J}{\partial z^l}$，$\because z_i^l = \sum_j^m w_{ij}^l \cdot x_j^{l-1} + b_i^l \quad (i = 0, 1, 2, \cdots, n-1)$

$$\therefore \frac{\partial J}{\partial w_{ij}^l} = \frac{\partial J}{\partial z_i^l} \cdot \frac{\partial z_i^l}{\partial w_{ij}^l} = \delta_i^l \cdot x_j^{l-1}$$

$$\Rightarrow \boxed{\frac{\partial J}{\partial W^l} = \delta^l \cdot (x^{l-1})^T} \quad ①$$

$$\frac{\partial J}{\partial b_i^l} = \frac{\partial J}{\partial z_i^l} \cdot \frac{\partial z_i^l}{\partial b_i^l}$$

$$\Rightarrow \boxed{\frac{\partial J}{\partial b^l} = \delta^l} \quad ②$$

$\because z^l = W^l \cdot f(z^{l-1}) + b^l$

$\therefore z_i^l = \sum_j^m w_{ij}^l \cdot f(z_j^{l-1}) + b_i^l \quad (i = 0, 1, \cdots, n-1)$

$$\therefore \frac{\partial J}{\partial z_i^{l-1}} = \sum_k^n \frac{\partial J}{\partial z_k^l} \cdot \frac{\partial z_k^l}{\partial z_i^{l-1}}$$

$$= \sum_k^n \delta_k^l \cdot w_{ki}^l \cdot f'(z_i^{l-1})$$

$$\Rightarrow \delta_i^{l-1} = \left(\sum_k^n \delta_k^l \cdot w_{ki}^l\right) \cdot f'(z_i^{l-1}) \Rightarrow \boxed{\delta^{l-1} = (W^l)^T \cdot \delta^l \odot f'(z^{l-1})} \quad ③$$

对于输出层 L 而言，$\delta^L = \frac{\partial J}{\partial z^L} = \frac{\partial J}{\partial x^L} \odot \frac{\partial x^L}{\partial z^L}$

$$\Rightarrow \boxed{\delta^L = \frac{\partial J}{\partial x^L} \odot f'(z^L)} \quad ④$$

$\odot$ 是矩阵 Hadamard 积，也就是逐元素相乘。

最后总结为：

1) 由训练样本给网络输入 X 和输出 Y.

2) 从第 2 层到第 L 层进行前向网络计算

3) 由公式④，计算第 L 层的残差 δ^L

4) 从第 L 层到第 2 层，由公式③递推计算各层残差 δ^l，然后由公式①，②计算梯度

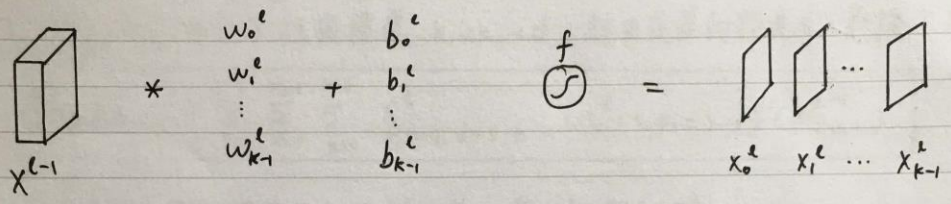
卷积层

卷积层的误差反向传播 [参考 3)4)5)6)]

设第 l 层为卷积层，输入 $X^{l-1} \in R^{S_1 \times S_2 \times d}$，有 K 个卷积核 $W_0^l, W_1^l, \cdots W_{K-1}^l$，且 $W_k^l \in R^{k_1 \times k_2 \times d}$，输出为 $X^l \in R^{q_1 \times q_2 \times K}$ 其中 $q_1 = S_1 - k_1 + 1$，$q_2 = S_2 - k_2 + 1$

（以 stride = 1 为例）

记：$W^l = [W_0^l, W_1^l, \cdots W_{K-1}^l]$，每一个卷积核 W_k^l 对应一个实数偏置 $b_k^l \in R$



X_k^l 是第 k 个卷积核对应的特征图（feature map），这 K 个特征图按通道叠起来就是输出 X^l 了。

$$Z^l = X^{l-1} * W^l + b^l$$

$$X^l = f(Z^l)$$

其中 $Z^l \in R^{q_1 \times q_2 \times K}$ 为中间值，f 是激活函数。

设 J 是网络损失函数，记 $\frac{\partial J}{\partial Z^l} = \delta^l$ 为误差项

由 $Z^l = f(Z^{l-1}) * W^l + b^l$，按卷积定义：

$$Z_{i,j,k}^l = \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} \sum_{c=0}^{K-1} f(Z_{i+m,j+n,c}^{l-1}) \cdot W_k^l(m,n,c) + b_k^l$$

$(0 \le i < q_1, \quad 0 \le j < q_2)$

$$\frac{\partial J}{\partial Z_{u,v,c}^{l-1}} = \sum_{i=0}^{q_1-1} \sum_{j=0}^{q_2-1} \sum_{k=0}^{K-1} \frac{\partial J}{\partial Z_{i,j,k}^l} \cdot \frac{\partial Z_{i,j,k}^l}{\partial Z_{u,v,c}^{l-1}}$$

$0 \le u < S_1$
$0 \le v < S_2$
$0 \le c < d$

即：$$\delta_{u,v,c}^{l-1} = \sum_{i=0}^{q_1-1} \sum_{j=0}^{q_2-1} \sum_{k=0}^{K-1} \delta_{i,j,k}^l \cdot \frac{\partial Z_{i,j,k}^l}{\partial Z_{u,v,c}^{l-1}}$$

由于 $\frac{\partial z^{l}_{i,j,k}}{\partial z^{l-1}_{u,v,c}} = f'(z^{l-1}_{u,v,c}) \cdot w^{l}_{k}(u-i, v-j, c)$

$\therefore \delta^{l-1}_{u,v,c} = \sum_{i=0}^{q_1-1}\sum_{j=0}^{q_2-1}\sum_{k=0}^{K-1} \delta^{l}_{i,j,k} \cdot w^{l}_{k}(u-i, v-j, c) \cdot f'(z^{l-1}_{u,v,c})$

对于 w^{l}_{k} 而言下标范围是 $K_1 \times K_2 \times d$，超出范围的都是0. 更换下求和下标

得：$$\boxed{\delta^{l-1}_{u,v,c} = \sum_{s=0}^{K_1-1}\sum_{t=0}^{K_2-1}\sum_{k=0}^{K-1} \delta^{l}_{u-s,v-t,k} \cdot w^{l}_{k}(s,t,c) \cdot f'(z^{l-1}_{u,v,c})} \quad ①$$

这就是卷积层的误差反向传播公式，其中的求和项

$\delta^{l}_{u-s,v-t,k} \cdot w^{l}_{k}(s,t,c)$，不考虑通道的话，其实就是把 w^{l}_{k} 进行上下翻转，再进行左右翻转后作为卷积核与 δ^{l} 进行卷积.

有了 δ 后再计算梯度 $\frac{\partial J}{\partial w}$，$\frac{\partial J}{\partial b}$

$\because z^{l}_{i,j,k} = \sum_{m=0}^{K_1-1}\sum_{n=0}^{K_2-1}\sum_{c} x^{l-1}_{i+m,j+n,c} \cdot w^{l}_{k}(m,n,c) + b^{l}_{k}$

由于第 k 个卷积核只影响到第 k 个特征图，所以：

$$\frac{\partial J}{\partial w^{l}_{k}(m,n,c)} = \sum_{i=0}^{q_1-1}\sum_{j=0}^{q_2-1} \frac{\partial J}{\partial z^{l}_{i,j,k}} \cdot \frac{\partial z^{l}_{i,j,k}}{\partial w^{l}_{k}(m,n,c)}$$

$$= \sum_{i=0}^{q_1-1}\sum_{j=0}^{q_2-1} \delta^{l}_{i,j,k} \cdot x^{l-1}_{i+m,j+n,c}$$

可以看出这也是一个卷积计算.

也即：$$\boxed{\frac{\partial J}{\partial w^{l}_{k}(c)} = x^{l-1}_{(c)} * \delta^{l}_{k}} \quad ②$$

$x^{l-1}_{(c)} \in R^{K_1 \times K_2}$，$\delta^{l}_{k} \in R^{q_1 \times q_2}$，$* \in R^{s_1 \times s_2}$

$$\frac{\partial J}{\partial b^{l}_{k}} = \sum_{i=0}^{q_1-1}\sum_{j=0}^{q_2-1} \frac{\partial J}{\partial z^{l}_{i,j,k}} \cdot \frac{\partial z^{l}_{i,j,k}}{\partial b^{l}_{k}}$$

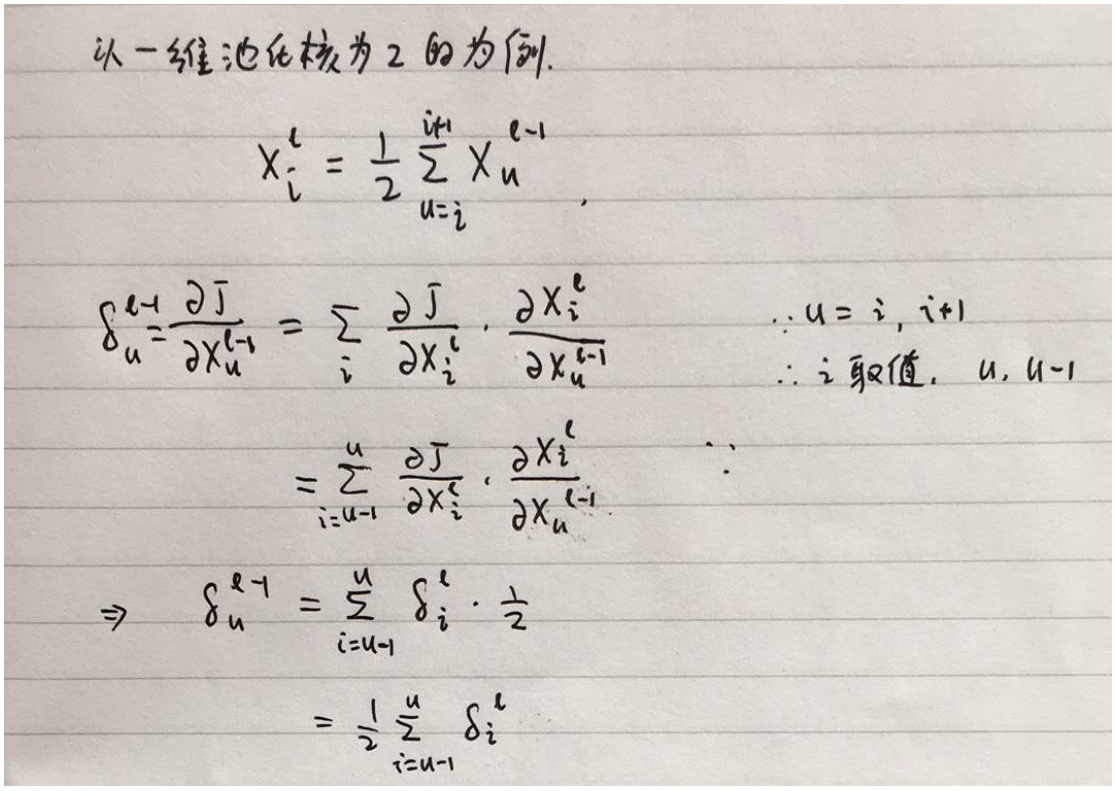
$$= \sum_{i=0}^{q_1-1}\sum_{j=0}^{q_2-1} \delta^{l}_{i,j,k}$$

$$\Rightarrow \boxed{\frac{\partial J}{\partial b^{l}_{k}} = \sum_{i=0}^{q_1-1}\sum_{j=0}^{q_2-1} \delta^{l}_{i,j,k}} \quad ③$$

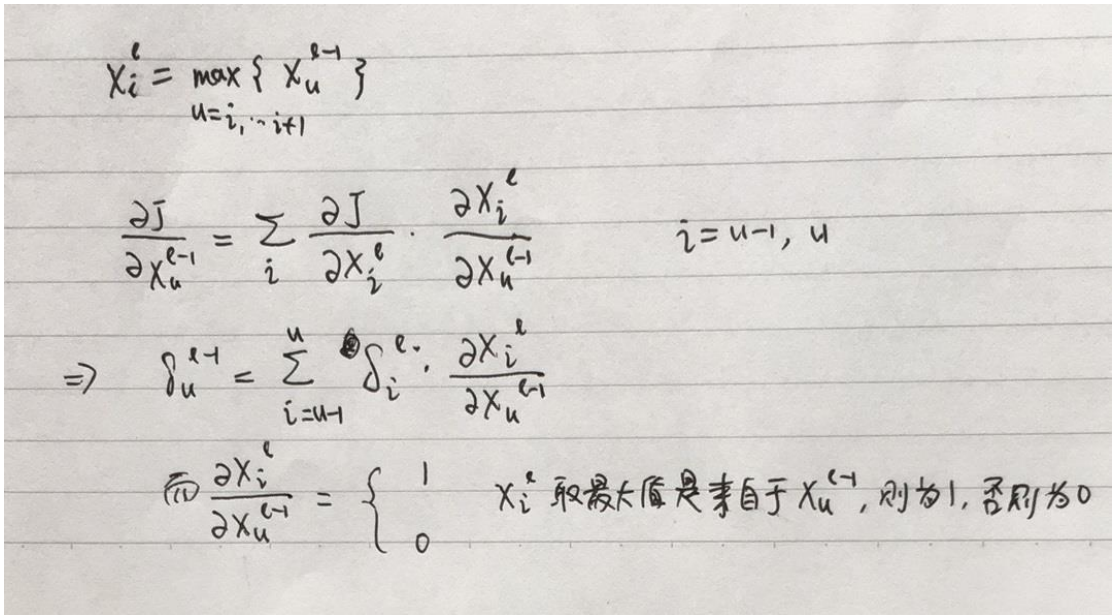
与全连接层类似：由式子（1）从输出层开始反向递推计算各层的误差项，然后由式子（2）（3）分别计算各层的梯度（损失函数对卷积核和偏置的导数）

池化层[参考 8)]

平均池化



最大池化



[参考文献]

1) [Neural Networks and Deep Learning](http://neuralnetworksanddeeplearning.com/) by By Michael Nielsen</br>
2) [Deep Learning](http://www.deeplearningbook.org/), book by Ian Goodfellow, Yoshua Bengio, and Aaron Courville</br>
3) http://cs231n.github.io/convolutional-networks </br>
4) http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture05.pdf
5) http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/
6) http://deeplearning.net/software/theano_versions/dev/tutorial/conv_arithmetic.html#transposed-convolution-arithmetic
7) http://ufldl.stanford.edu/wiki/index.php/Gradient_checking_and_advanced_optimization
8） https://software.intel.com/sites/products/documentation/doclib/daal/daal-user-and-reference-guides/daal_prog_guide/GUID-2C3AA967-AE6A-4162-84EB-93BE438E3A05.htm