1. 如何理解后向传播

参考CNN卷积神经网络学习笔记3：权值更新公式推导 后向传播的过程就是梯度向回传递，在CNN中，梯度的计算主要涉及三种情形

1. 卷积层
2. 池化层
3. 全连接层

其中，卷积层涉及3种操作下的梯度计算

1. 卷积操作
2. 偏置
3. 激活操作

池化层则有两种情形：

1. 平均池化
2. 最大池化

而全连接层的后向传播与全连接神经网络的后向传播原理一致。涉及：

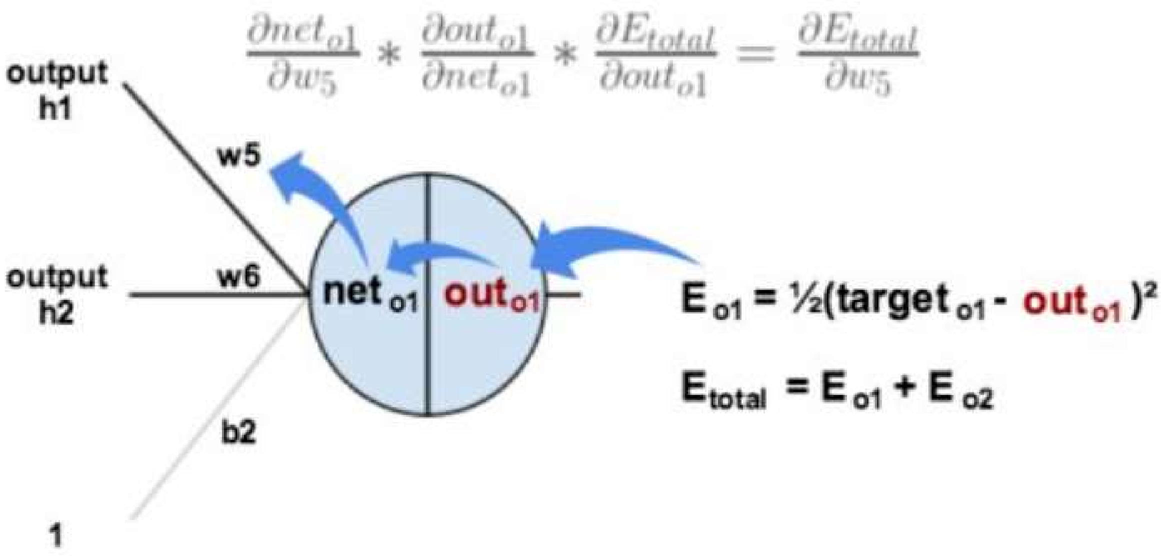
1. 权重的相乘与偏置
2. 激活操作

本文先讨论全连接层的后向传播，再讨论卷积层、池化层的梯度传递。

2. 全连接层的梯度计算

知乎的如何理解神经网络里面的反向传播算法讲的很好。 主要是输出层与隐藏层的梯度传递

2.1 输出层的梯度传递



通过梯度下降调整 w_5 ，需要求 $\frac{\partial E_{total}}{\partial w_5}$ ，由链式法则：

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o_1}} \frac{\partial out_{o_1}}{\partial net_{o_1}} \frac{\partial net_{o_1}}{\partial w_5},$$

$$\frac{\partial E_{total}}{\partial out_{o_1}} = \frac{\partial}{\partial out_{o_1}} (\frac{1}{2}(target_{o_1} - out_{o_1})^2 + \frac{1}{2}(target_{o_2} - out_{o_2})^2) = -(target_{o_1} - out_{o_1})$$

$$\frac{\partial out_{o_1}}{\partial net_{o_1}} = \frac{\partial}{\partial net_{o_1}} \frac{1}{1 + e^{-net_{o_1}}} = out_{o_1}(1 - out_{o_1})$$

$$\frac{\partial net_{o_1}}{\partial w_5} = \frac{\partial}{\partial w_5}(w_5 \times out_{h_1} + w_6 \times out_{h_2} + b_2 \times 1) = out_{h_1}$$

以上3个相乘得到梯度 $\frac{\partial E_{total}}{\partial w_5}$ ，之后就可以用这个梯度训练了：

$$w_5^+ = w_5 - \eta \frac{\partial E_{total}}{\partial w_5}$$

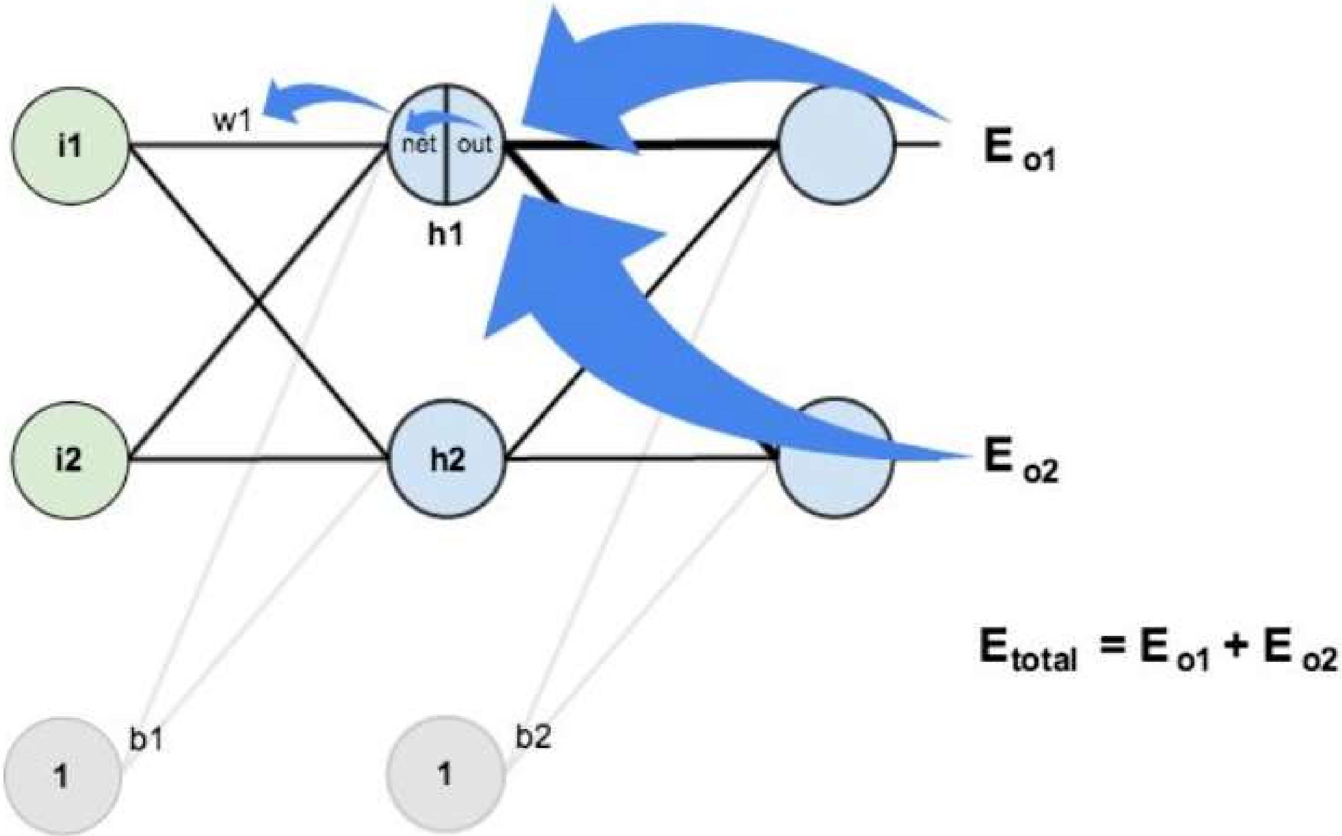
很多教材比如Stanford的课程，会把中间结果 $\frac{\partial E_{total}}{\partial net_{o_1}} = \frac{\partial E_{total}}{\partial out_{o_1}} \frac{\partial out_{o_1}}{\partial net_{o_1}}$ 记做 δ_{o_1} ，表示这个节点对最终的误差需要负多少责任。。所以有 $\frac{\partial E_{total}}{\partial w_5} = \delta_{o_1} out_{h_1}$ 。

这个就是关于节点的梯度的计算(相对于权重的梯度的计算。因为我们是要用梯度下降改变权值，所以要求**权重的梯度**，但在过程中总是要得到关于**每一层的节点的梯度**)，又称**灵敏度**，表示了对最终误差造成的影响。正因

为它的这个意义，关于一个权重的梯度可以由**该权重的上的输出**乘以**节点的灵敏度**得到，也就是

$\delta_{o_1} out_{h_1}$ 这个公式同样适用于隐藏层。

2.2 隐藏层的梯度传递



通过梯度下降调整 w_1，需要求 $\frac{\partial E_{total}}{\partial w_1}$，由链式法则：

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h_1}} \frac{\partial out_{h_1}}{\partial net_{h_1}} \frac{\partial net_{h_1}}{\partial w_1},$$

如下图所示：

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\downarrow$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$

其中

$$\frac{\partial E_{o_1}}{\partial out_{h_1}} = \frac{\partial E_{o_1}}{\partial net_{o_1}} \times \frac{\partial net_{o_1}}{\partial out_{h_1}} = \delta_{o_1} \times \frac{\partial net_{o_1}}{\partial out_{h_1}} = \delta_{o_1} \times \frac{\partial}{\partial out_{h_1}}(w_5 \times out_{h_1} + w_6 \times out_{h_2} + b_2 \times 1) = \delta_{o_1} w_5$$

，这里 δ_{o_1} 之前计算过。

$\frac{\partial E_{o_2}}{\partial out_{h_1}}$ 的计算也类似，所以得到

$\frac{\partial E_{total}}{\partial out_{h_1}} = \delta_{o_1} w_5 + \delta_{o_2} w_7$。

$\frac{\partial E_{total}}{\partial w_1}$ 的链式中其他两项如下：

$\frac{\partial out_{h_1}}{\partial net_{h_1}} = out_{h_1}(1 - out_{h_1})$，

$$\frac{\partial net_{h_1}}{\partial w_1} = \frac{\partial}{\partial w_1}(w_1 \times i_1 + w_2 \times i_2 + b_1 \times 1) = i_1$$

相乘得到

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h_1}} \frac{\partial out_{h_1}}{\partial net_{h_1}} \frac{\partial net_{h_1}}{\partial w_1} = (\delta_{o_1} w_5 + \delta_{o_2} w_7) \times out_{h_1}(1 - out_{h_1}) \times i_1$$

得到梯度后，就可以对 w_1 迭代了：

$w_1^+ = w_1 - \eta \frac{\partial E_{total}}{\partial w_1}$。

在前一个式子里同样可以对 δ_{h_1} 进行定义，

$$\delta_{h_1} = \frac{\partial E_{total}}{\partial out_{h_1}} \frac{\partial out_{h_1}}{\partial net_{h_1}} = (\delta_{o_1} w_5 + \delta_{o_2} w_7) \times out_{h_1}(1 - out_{h_1}) = (\sum_o \delta_o w_{ho}) \times out_{h_1}(1 - out_{h_1})$$

，所以整个梯度可以写成 $\frac{\partial E_{total}}{\partial w_1} = \delta_{h_1} \times i_1$

这里同样印证了上文的公式： 权重的梯度=输出节点的灵敏度 * 权重上的值

3. 卷积层

3.1 卷积操作

3.1.1 卷积操作的各个梯度

参考 Forward And Backpropagation in Convolutional Neural Network. 假如有特征图与卷积核如下：

$$\underbrace{\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}}_{\textbf{Input}} \qquad \underbrace{\begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}}_{\textbf{Filter}}$$

且输出与这两个矩阵的关系如下：

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \textbf{Correlation}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$

$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$

$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$

$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$

那么，关于卷积核**F**的每一项**F_ij**的梯度计算公式如下：

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{11}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{11}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{11}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{11}}$$

$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{12}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{12}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{12}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{12}}$$

$$\frac{\partial E}{\partial F_{21}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{21}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{21}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{21}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{21}}$$

$$\frac{\partial E}{\partial F_{22}} = \frac{\partial E}{\partial O_{11}}\frac{\partial O_{11}}{\partial F_{22}} + \frac{\partial E}{\partial O_{12}}\frac{\partial O_{12}}{\partial F_{22}} + \frac{\partial E}{\partial O_{21}}\frac{\partial O_{21}}{\partial F_{22}} + \frac{\partial E}{\partial O_{22}}\frac{\partial O_{22}}{\partial F_{22}}$$

也就等于：

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}}X_{11} + \frac{\partial E}{\partial O_{12}}X_{12} + \frac{\partial E}{\partial O_{21}}X_{21} + \frac{\partial E}{\partial O_{22}}X_{22}$$

$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}}X_{12} + \frac{\partial E}{\partial O_{12}}X_{13} + \frac{\partial E}{\partial O_{21}}X_{22} + \frac{\partial E}{\partial O_{22}}X_{23}$$

$$\frac{\partial E}{\partial F_{21}} = \frac{\partial E}{\partial O_{11}}X_{21} + \frac{\partial E}{\partial O_{12}}X_{22} + \frac{\partial E}{\partial O_{21}}X_{31} + \frac{\partial E}{\partial O_{22}}X_{32}$$

$$\frac{\partial E}{\partial F_{22}} = \frac{\partial E}{\partial O_{11}}X_{22} + \frac{\partial E}{\partial O_{12}}X_{23} + \frac{\partial E}{\partial O_{21}}X_{32} + \frac{\partial E}{\partial O_{22}}X_{33}$$

当我们仔细观察上图这几个式子的规律，可以发现，卷积核的梯度可以这样得来：

$$\begin{bmatrix} \partial E/\partial F_{11} & \partial E/\partial F_{12} \\ \partial E/\partial F_{21} & \partial E/\partial F_{22} \end{bmatrix} = \textbf{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} \partial E/\partial O_{11} & \partial E/\partial O_{12} \\ \partial E/\partial O_{21} & \partial E/\partial O_{22} \end{bmatrix} \right)$$

然后卷积核各项都可以根据此梯度进行调整。但是，我们还要把梯度传递给上一层，就需要计算**关于输入的梯度**。通过与计算卷积核的梯度同样的方法，我们可以得到关于各个**X_ij**的梯度：

$$\frac{\partial E}{\partial X_{11}} = \frac{\partial E}{\partial O_{11}}F_{11} + \frac{\partial E}{\partial O_{12}}0 + \frac{\partial E}{\partial O_{21}}0 + \frac{\partial E}{\partial O_{22}}0$$

$$\frac{\partial E}{\partial X_{12}} = \frac{\partial E}{\partial O_{11}}F_{12} + \frac{\partial E}{\partial O_{12}}F_{11} + \frac{\partial E}{\partial O_{21}}0 + \frac{\partial E}{\partial O_{22}}0$$

$$\frac{\partial E}{\partial X_{13}} = \frac{\partial E}{\partial O_{11}}0 + \frac{\partial E}{\partial O_{12}}F_{12} + \frac{\partial E}{\partial O_{21}}0 + \frac{\partial E}{\partial O_{22}}0$$

$$\frac{\partial E}{\partial X_{21}} = \frac{\partial E}{\partial O_{11}}F_{21} + \frac{\partial E}{\partial O_{12}}0 + \frac{\partial E}{\partial O_{21}}F_{11} + \frac{\partial E}{\partial O_{22}}0$$

$$\frac{\partial E}{\partial X_{22}} = \frac{\partial E}{\partial O_{11}}F_{22} + \frac{\partial E}{\partial O_{12}}F_{21} + \frac{\partial E}{\partial O_{21}}f_{12} + \frac{\partial E}{\partial O_{22}}F_{11}$$

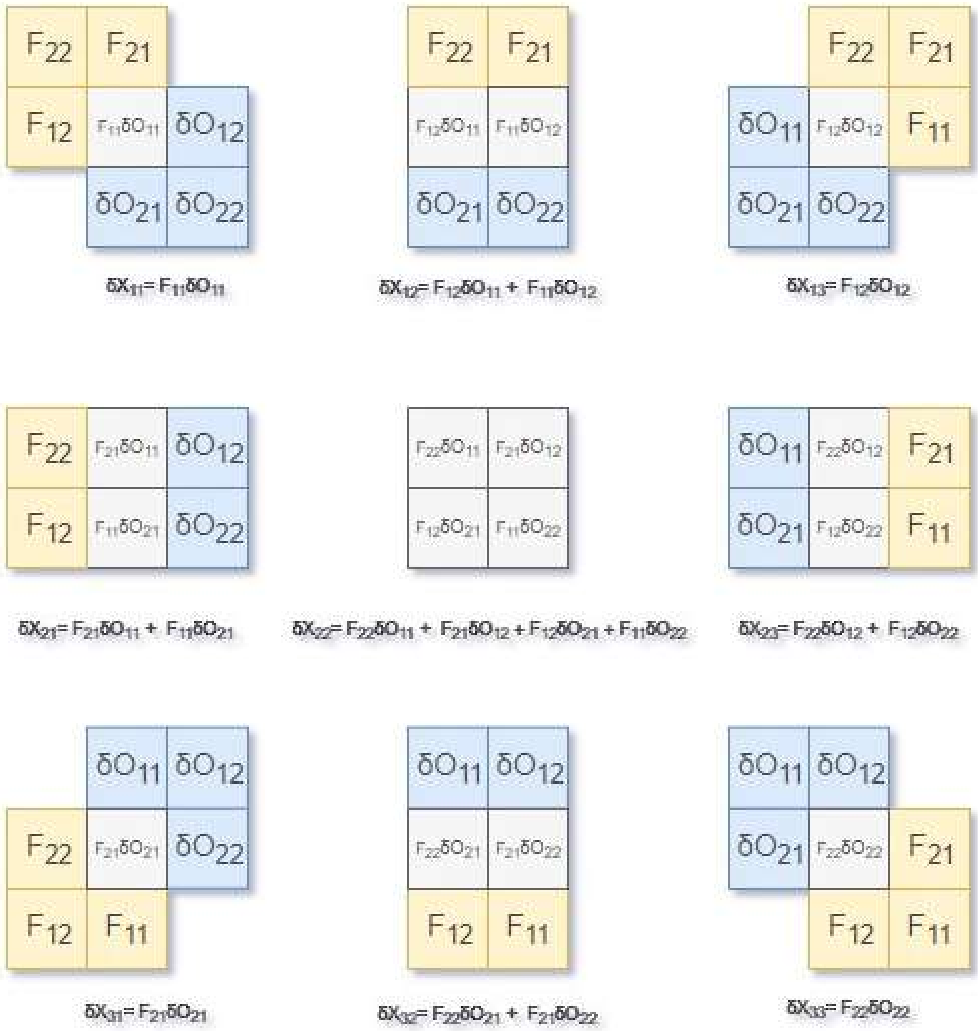
$$\frac{\partial E}{\partial X_{23}} = \frac{\partial E}{\partial O_{11}}0 + \frac{\partial E}{\partial O_{12}}F_{22} + \frac{\partial E}{\partial O_{21}}0 + \frac{\partial E}{\partial O_{22}}F_{11}$$

$$\frac{\partial E}{\partial X_{31}} = \frac{\partial E}{\partial O_{11}}0 + \frac{\partial E}{\partial O_{12}}0 + \frac{\partial E}{\partial O_{21}}F_{21} + \frac{\partial E}{\partial O_{22}}0$$

$$\frac{\partial E}{\partial X_{32}} = \frac{\partial E}{\partial O_{11}}0 + \frac{\partial E}{\partial O_{12}}0 + \frac{\partial E}{\partial O_{21}}F_{22} + \frac{\partial E}{\partial O_{22}}F_{21}$$

$$\frac{\partial E}{\partial X_{33}} = \frac{\partial E}{\partial O_{11}}0 + \frac{\partial E}{\partial O_{12}}0 + \frac{\partial E}{\partial O_{21}}0 + \frac{\partial E}{\partial O_{22}}F_{22}$$

仔细观察上图这几个式子的规律，可以发现，输入的梯度可以化为**全卷积操作**：

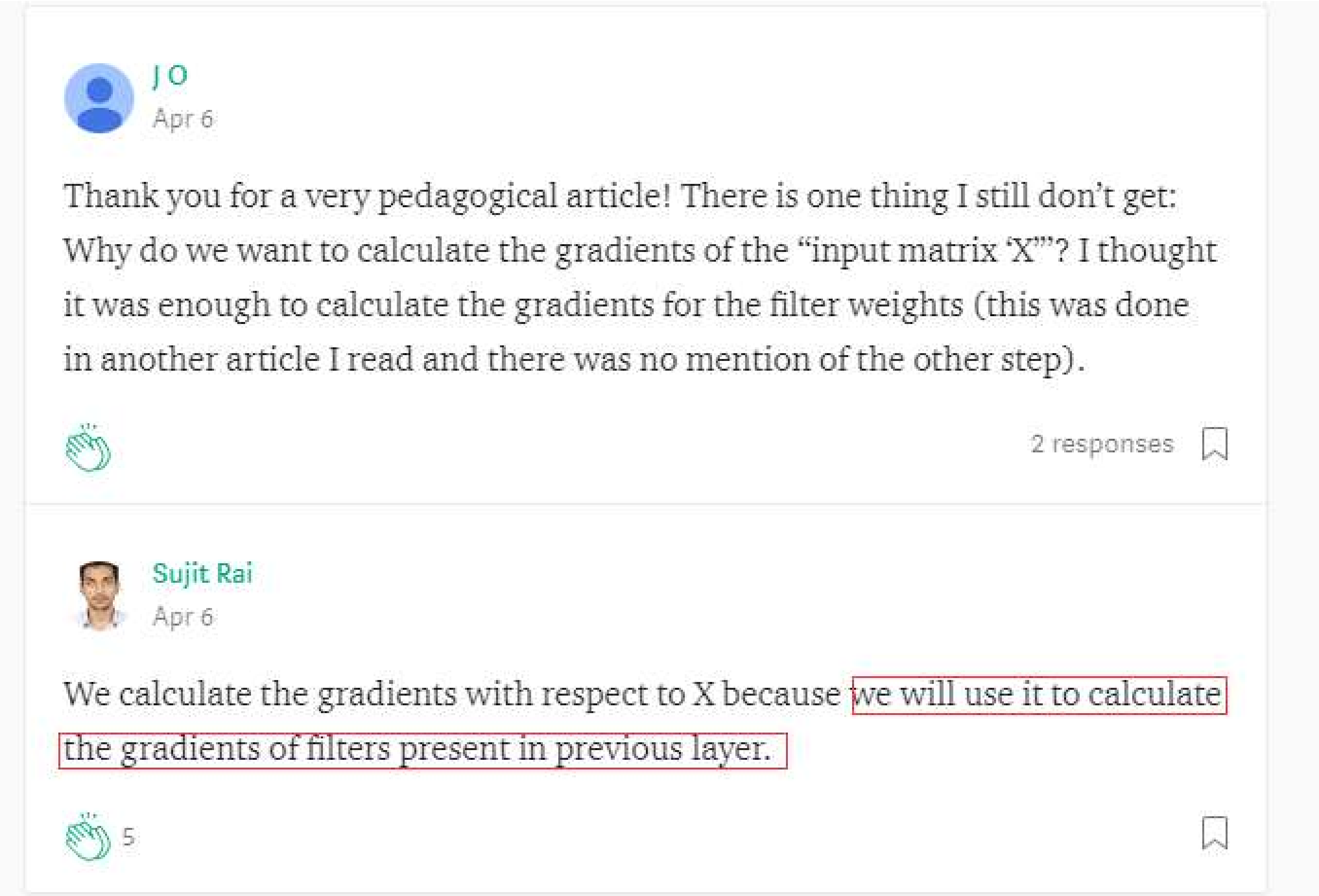


Here 'δX' represents the gradients of error with respect to X

全卷积的具体操作如下：

3.1.2 关于输入的梯度的用途

本来我感觉奇怪，如果关于**卷积核的梯度**是用于**调整卷积核各项的值**的话，那**关于输入的梯度**是用来做什么的呢？我看到了文章评论区有人刚好问了这个问题：


J O
Apr 6

Thank you for a very pedagogical article! There is one thing I still don't get: Why do we want to calculate the gradients of the "input matrix 'X'"? I thought it was enough to calculate the gradients for the filter weights (this was done in another article I read and there was no mention of the other step).

2 responses

Sujit Rai
Apr 6

We calculate the gradients with respect to X because we will use it to calculate the gradients of filters present in previous layer.

5

原来，它是**用于计算上一层的梯度用的。**其实，这一层对输入的梯度$\partial E/\partial F$就等于上一层对输出的梯度$\frac{\partial E}{\partial O}$

这篇文章Back Propagation in Convolutional Neural Networks—Intuition and Code也提到了它的用处：

> It is important to understand that ∂x (or ∂h for previous layer) would be the input for the backward pass of the previous layer. This is the core principle behind the success of back propagation.

3.1.3 概括

也就是说，卷积操作主要是求出两个：**关于卷积核的梯度**以及**关于输入的梯度。**其中。关于卷积核的梯度是用于调整卷积核各项的值的，关于输入的梯度则是用于给更上一层作为输出梯度的。

3.2 偏置与激活

梯度的传递在经过偏置操作与激活操作时的变化都在**2. 全连接层的梯度计算**里讲解了，卷积层的处理与全连接层在此方向的处理是一致的。

4. 池化层

4.1 average-pooling

平均池化的操作可以转化为卷积操作。比如，2 * 2的平均池化可以转化为卷积核为2 * 2，每项为1/4 的卷积操作。

4.2 max-pooling

知乎的 卷积神经网络(CNN)中卷积层与池化层如何进行BP残差传递与参数更新？中提到的 Backpropagation in Convolutional Neural Network 解释了平均池化与最大池化的梯度传递

$$
g(x)=\begin{cases}
\dfrac{\sum_{k=1}^{m} x_k}{m},\ \dfrac{\partial g}{\partial x}=\dfrac{1}{m} & \text{mean pooling}\\[2ex]
\max(x),\ \dfrac{\partial g}{\partial x_i}=\begin{cases}1 \text{ if } x_i=\max(x)\\ 0 \text{ otherwise}\end{cases} & \text{max pooling}\\[2ex]
\|x\|_p=\left(\sum_{k=1}^{m}|x_k|^p\right)^{1/p},\ \dfrac{\partial g}{\partial x_i}=\left(\sum_{k=1}^{m}|x_k|^p\right)^{1/p-1}|x_i|^{p-1} & L^p \text{ pooling}\\[2ex]
\text{or any other differentiable } \mathbf{R}^m \to \mathbf{R} \text{ functions}
\end{cases}
$$

假如某个矩阵被圈中的部分是最大项：

L1

$$
\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_5 & x_6 & x_7 & x_8 \\ x_9 & x_{10} & x_{11} & x_{12} \\ x_{13} & x_{14} & x_{15} & x_{16} \end{bmatrix}
$$

它们对应的梯度就是：

$$
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}
$$

当该项被选取为最大项时，它的对应梯度为1，否则为0.

此文同样表达了这一点 Backpropagation in Pooling Layer (Subsamplig layer) in CNN

> 加入矩阵M有4个元素 a b c d 而且maxpool(M)返回d. 那么，maxpool函数就只依赖于d了. 那么，关于d的导数为1，关于a,b,c的导数为0. 所以，在计算关于d的梯度时，就是乘上1, 对其它的梯度乘上0.

参考

1. CNN卷积神经网络学习笔记3：权值更新公式推导
2. BP神经网络后向传播算法
3. Only Numpy: Understanding Back Propagation for Max Pooling Layer in Multi Layer CNN with Example and Interactive Code. (With and Without Activation Layer)
4. 卷积神经网络(CNN)中卷积层与池化层如何进行BP残差传递与参数更新？

5. Backpropagation in Convolutional Neural Network