卷积神经网络(CNN)反向传播算法推导

> 主要参考了Convolutional Neural Networks backpropagation: from intuition to derivation – Grzegorz Gwardys (wordpress.com)

卷积神经网络(CNN)反向传播算法主要分为两部分：**池化层反向传播**和**卷积层反向传播。**

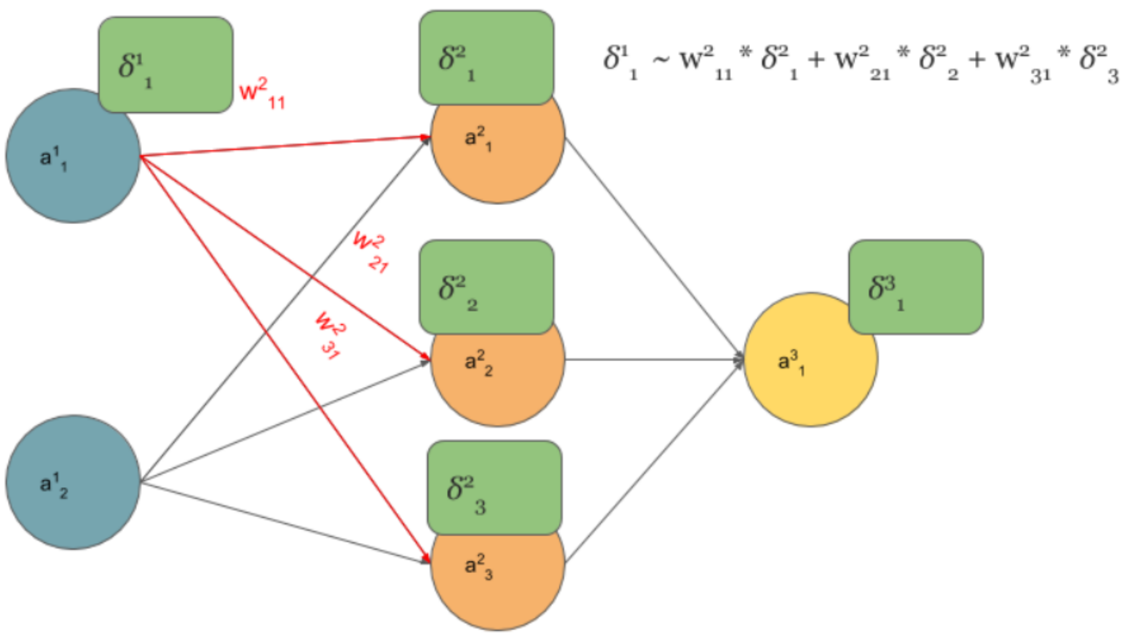
大致的思路：

池化层反向传播

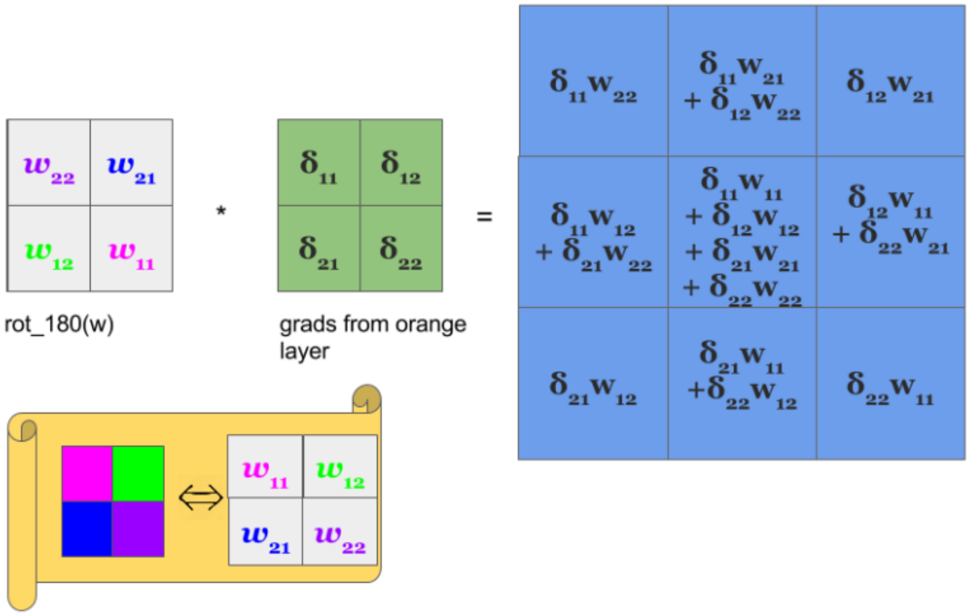
池化层的反向传播比较容易理解，我们以**最大池化**举例，上图中，池化后的数字6对应于池化前的红色区域，实际上只有红色区域中最大值数字6对池化后的结果有影响，权重为1，而其它的数字对池化后的结果影响都为0。假设池化后数字6的位置误差为δ，误差反向传播回去时，红色区域中最大值对应的位置误差即等于δ，而其它3个位置对应的δ误差为0。因此，在卷积神经网络最大池化前向传播时，不仅要记录区域的最大值，同时也要记录下来区域最大值的位置，方便δ误差的反向传播。而**平均池化**就更简单了，由于平均池化时，区域中每个值对池化后结果贡献的权重都为区域大小的倒数，所以δ误差反向传播回来时，在区域每个位置的误差都为池化后δ误差除以区域的大小。

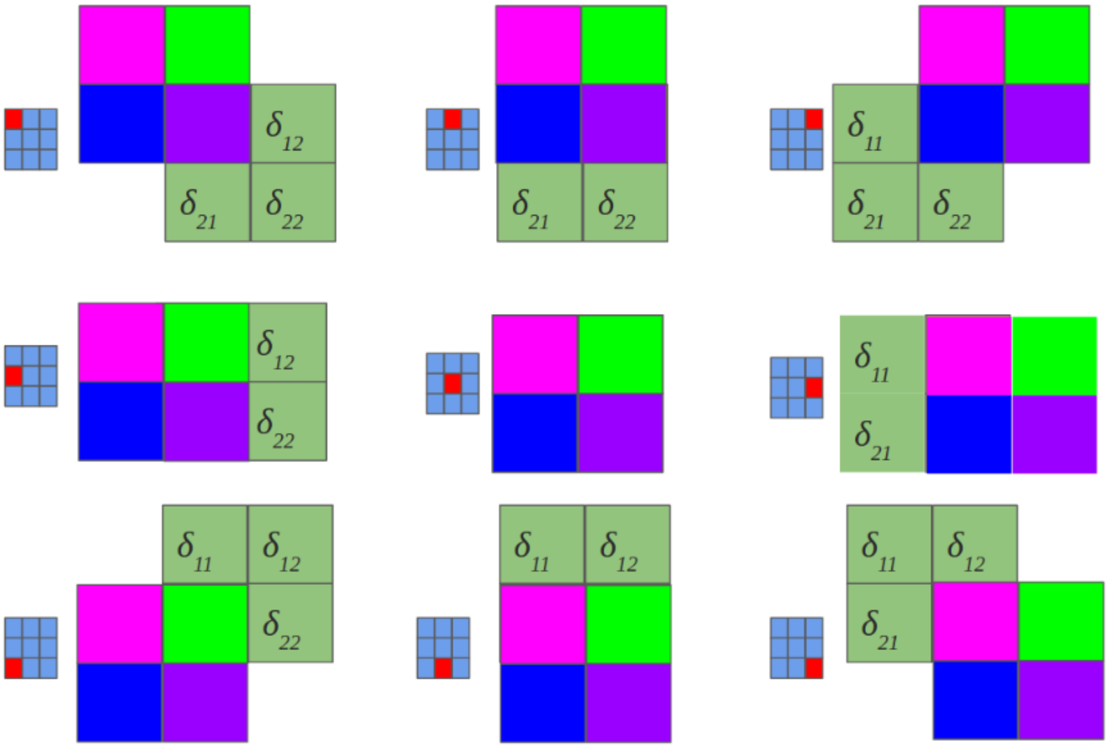
卷积层反向传播

核心的计算只涉及二维卷积，因此我们先从二维的卷积运算来进行分析：求原图某处的δ,需分析它在前向传播中影响了下一层的哪些结点，大致为以下过程



我们可以发现这样一个规律，原图的δ误差，等于卷积结果的δ误差**经过零填充**后，与**卷积核旋转180度**后的卷积。如下图所示：





具体的数学表达式如下

$$\delta_j^l = f^{'}(u_j^l) \odot \mathrm{conv2}(\delta_j^{l+1}, \mathrm{rot180}(k_j^{l+1}), '\mathrm{full}')$$

接下来进行数学公式的推导:

考虑δ误差是损失函数对于当前层未激活输出z^l的导数，我们现在考虑的是二维卷积，因此，每一层的δ误差是一个二维的矩阵。$\delta^l(x,y)$表示的是第l层坐标为(x,y)处的δ误差。假设我们已经知道第$l+1$层的δ误差，利用求导的链式法则，可以很容易写出下式:

$$\delta^l(x,y) = \frac{\partial C}{\partial z^l(x,y)} = \sum_{x^{'}}\sum_{y^{'}} \frac{\partial C}{\partial z^{l+1}(x^{'},y^{'})} \frac{\partial z^{l+1}(x^{'},y^{'})}{\partial z^l(x,y)} = \sum_{x^{'}}\sum_{y^{'}} \delta^{l+1}(x^{'},y^{'}) \frac{\partial z^{l+1}(x^{'},y^{'})}{\partial z^l(x,y)}$$

其中，坐标$(x^{'},y^{'})$是第$l+1$层在前向传播中受第l层坐标(x,y)影响的点，我们需要将所有受影响的点都加起来，再用到前向传播的关系式:

$$z^{l+1}(x^{'},y^{'})=\sum_a\sum_b\sigma(z^l(x^{'}+a,y^{'}+b))w^{l+1}(a,b)+b^{l+1}$$

那么我们可以将表达式展开：

$$\delta^l(x,y)=\sum_{x^{'}}\sum_{y^{'}}\delta^{l+1}(x^{'},y^{'})\frac{\partial\sum_a\sum_b\sigma(z^l(x^{'}+a,y^{'}+b))w^{l+1}(a,b)+b^{l+1})}{\partial z^l(x,y)}$$

同时我们可以对他进行化简：

$$\delta^l(x,y)=\sum_{x^{'}}\sum_{y^{'}}\delta^{l+1}(x^{'},y^{'})w^{l+1}(a,b)\sigma^{'}(z^l(x,y))$$

同时因为限制条件$x^{'}+a=x$和$y^{'}+b=y$

那么上式可变为

$$\delta^l(x,y)=\sum_a\sum_b\delta^{l+1}(x-a,y-b)w^{l+1}(a,b)\sigma^{'}(z^l(x,y))$$

令$a^{'}=-a$以及$b^{'}=-b$那么我们有

$$\delta^l(x,y)=\sum_{a^{'}}\sum_{b^{'}}\delta^{l+1}(x+a^{'},y+b^{'})w^{l+1}(-a^{'},-b^{'})\sigma^{'}(z^l(x,y))$$

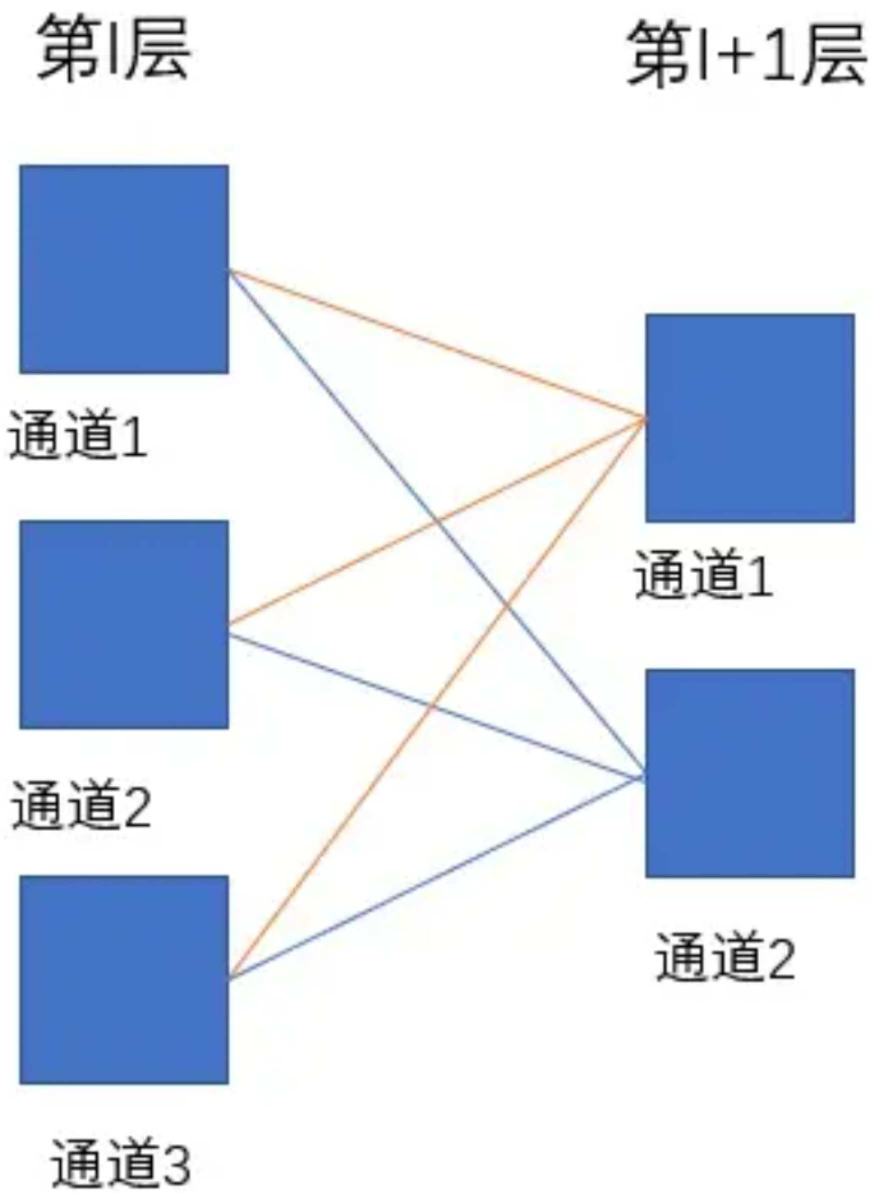
即

$$\delta^l=\mathrm{conv2}(\delta^{l+1},\mathrm{rot180}(w^{l+1}))\odot\sigma^{'}(z^l)$$

这个表达式只是基于二维卷积，我们需要把它推广到卷积神经网络中张量的卷积里去。

张量的卷积：后一层的每个通道都是由前一层的各个通道经过卷积再求和得到的

类比全连接神经网络：把通道变成结点，把卷积变成乘上权重。



上图中每根连线都代表与一个二维卷积核的卷积操作，假设第l层深度为3，第l+1层深度为2，卷积核的维度就应该为2×filter_size×filter_size×3。第l层的通道1通过卷积影响了第l+1层的通道1和通道2，那么求第l层通道1的误差时，就应该根据求得的二维卷积的误差传播方式，将第l+1层通道1和通道2的δ误差传播到第l层的δ误差进行简单**求和**即可。

接下来我们分析一些情况

已知第l层delta误差，求该层的参数的导数 $\frac{\partial C}{\partial w^l}$

$$\delta^l = \frac{\partial C}{\partial z^l}, z^l = a^{l+1} * w^l + b^l$$

第l层卷积核w^l是一个4维张量,它的维度表示为卷积核个数×行数×列数×通道数。实际上，可以把它视为有卷积核个数×通道数个二维卷积核，每个都对应输入图像的对应通道和输出图像的对应通道，每一个二维卷积核只涉及到一次二维卷积运算。那求得整个卷积核的导数，只需分析卷积核数×通道数次二维卷积中每个二维卷积核的导数，再将其组合成4维张量即可。

而二维卷积核的导数等于原图对应通道与卷积结果对应通道的δ误差直接进行卷积。

那么

$$\frac{\partial C}{\partial w^l} = \frac{\partial C}{\partial z^l}\frac{\partial z^l}{\partial w^l} = \delta^l * a^{l-1}$$

然后我们将**原图通道数×卷积结果通道数**个二维卷积核的导数重新进行组合成4为张量，即可得到整个卷积核的导数。

$$\frac{\partial C}{\partial w^l(a,b)} = \sum_x \sum_y \delta^l(x,y) \frac{\partial z^l(x,y)}{\partial w^l(a,b)}$$

$$\frac{\partial C}{\partial w^l(a,b)} = \sum_x \sum_y \delta^l(x,y) \frac{\partial \sum_{a'} \sum_{b'} \sigma(z^{l-1}(x+a^{'},y+b^{'}))w^l(a^{'},b^{'})+b^l)}{\partial w^l(a,b)}$$

化简得到限制条件：$a^{'}=a, b^{'}=b$:

$$\frac{\partial C}{\partial w^l(a,b)} = \sum_x \sum_y \delta^l(x,y)\sigma(z^{l-1}(x+a,y+b))\sigma^{'}(z^{l-1}(x+b,y+b))$$

$$\frac{\partial C}{\partial w^l} = \delta^l * \sigma(z^{l-1})$$

这边我们发现并不需要进行旋转180度

已知第l层delta误差，求该层的参数的导数 $\frac{\partial C}{\partial b^l}$

b^l是一个列向量，它给卷积结果的每一个通道都加上了同一个标量。在反向传播时，它的导数等于卷积结果的delta误差在每一个通道上将所有delta误差进行求和的结果。

即

$$\frac{\partial C}{\partial b^l} = \frac{\partial C}{\partial z^l}\frac{\partial z^l}{\partial b^l} = \sum_x \sum_y \delta^l$$

证明也非常方便由于$\frac{\partial z^l(x,y)}{\partial b^l} = 1$