Convolution and its gradients

Conv2D operation and/or layer in machine learning can be expressed as many small convolution kernels, followed by adding biases.

Let's first define a convolution kernel in its simplied form. Given X, y, and W, where X and y are the input matrix with shape $< m, n >$ and scalar output respectively and W is the filter with shape $< a, b >$, we define the convolution kernel

$$y = X \odot W = \sum_{x=0}^{a-1} \sum_{y=0}^{b-1} X_{x,y} W_{x,y}$$

as a two-dimension dot product over all non-zero elements where the top-left corners of X and W are aligned. In addition, if $a > m$ or $b > n$, the X is padded with zeros.

With this, gradients are straightforward:

$$\frac{\mathrm{d}y}{\mathrm{d}W_{x,y}} = X_{x,y}$$
$$\frac{\mathrm{d}y}{\mathrm{d}X_{x,y}} = W_{x,y}$$

Secondly, let's define a shifted version for convolution kernel:

$$y_{\triangleright\, \alpha,\beta} = \big(X \odot W\big)_{\triangleright\, \alpha,\beta} = X \odot W_{\triangleright\, \alpha,\beta} = \sum_{x=0}^{a-1} \sum_{y=0}^{b-1} X_{x+\alpha,y+\beta} W_{x,y}$$

where $\triangleright\, \alpha, \beta$ means shifting to right and down with (α, β) offsets. Any missing values due to out of boundaries are padded with zeros. Accordingly, gradients are as follows:

$$\frac{\mathrm{d}y_{\triangleright\, \alpha,\beta}}{\mathrm{d}W_{x,y}} = X_{x+\alpha,y+\beta}$$
$$\frac{\mathrm{d}y_{\triangleright\, \alpha,\beta}}{\mathrm{d}X_{x,y}} = W_{x-\alpha,y-\beta}$$

Conv2D and its Gradients

Forward Pass Conv2D is quite simple. Given X, W, and Y, where X and Y are the input and output respectively and W is a typically quite small filter, we denote the formula as

$$Y = X \otimes W.$$

There is a hidden controlling argument for the Conv2D, which is the padding. Assuming the padding is `same`, which means the shape of the output Y is same as the shape of input X, denocated as $< m, n >$ and, W has odd number of rows and columns, denoted as $< a, b >$, then we have

$$\begin{aligned} Y_{i,j} &= y_{\triangleright\, i,j} \\ &= (X \odot W)_{\triangleright\, i-\frac{a-1}{2}, j-\frac{b-1}{2}} \\ &= \sum_{x=0}^{a-1} \sum_{y=0}^{b-1} X_{i+x-\frac{a-1}{2}, j+y-\frac{b-1}{2}} W_{x,y}. \end{aligned}$$

where $0 \le i \le m-1$ and $0 \le j \le n-1$. Here, there is a hidden padding in the input X, i.e., for any coordinates out of the boundaries, the input value is zero.

Grad of X If we examine the Jacobian matrix, each column, i.e., for a fixed input x, has ab non-zero elements, due to the filter shape $< a, b >$. And due to the symmetricity, it is trivial to prove:

$$dX = dY \otimes W'.$$

where the padding is still `same` and W' is a matrix with same shape $< a, b >$ as W, but all elements reversed, i.e.,

$$W' = \begin{bmatrix} W_{a-1,b-1} & W_{a-1,b-2} & \cdots & W_{a-1,0} \\ W_{a-2,b-1} & W_{a-2,b-2} & \cdots & W_{a-2,0} \\ \vdots & \vdots & \ddots & \vdots \\ W_{0,b-1} & W_{0,b-2} & \cdots & W_{0,0} \end{bmatrix}$$

Grad of W For the central point of W, the gradient is

$$dW_{\frac{a-1}{2},\frac{b-1}{2}} = dY \odot X$$

where $\odot$ is a 2-D point-wise dot product. With that, we can summerize the gradients for each point of W as

$$dW_{x+\frac{a-1}{2},y+\frac{b-1}{2}} = dY_{\triangleright(x,y)} \odot X$$

where $-\frac{a-1}{2} \leq x \leq \frac{a-1}{2}, -\frac{b-1}{2} \leq y \leq \frac{b-1}{2}$ and $\triangleright(x, y)$ means shifting the matrix to right with (x, y) offsets.