Forward and Backward Propagation of Convolutional Layer

Jianfeng Wang [*]

Feb. 5, 2015

1 Forward

$$z_j^{l+1} = \sum_i \omega_{j,i}^l * a_i^l \tag{1}$$

Note a_i^l, $\omega_{j,i}^l$ are 2-D rather than 1-D. Denote L as the loss function.

From Eqn. 1, we have

$$z_j^{l+1}(p,q) = \sum_{i,u,v} \omega_{j,i}^l(u,v) a_i^l(p-u, q-v) \tag{2}$$

$$= \sum_{i,u,v} \omega_{j,i}^l(p-u, q-v) a_i^l(u,v) \tag{3}$$

2 Backward

$$\frac{\partial L}{\partial \omega_{j,i}^l(u,v)} = \sum_{p,q} \frac{\partial L}{\partial z_j^{l+1}(p,q)} \frac{\partial z_j^{l+1}(p,q)}{\partial \omega_{j,i}^l(u,v)} \tag{4}$$

$$= \sum_{p,q} \frac{\partial L}{\partial z_j^{l+1}(p,q)} a_i^l(p-u, q-v) \tag{5}$$

$$= \frac{\partial L}{\partial z_j^{l+1}(u,v)} * a_i^l(-u,-v) \tag{6}$$

$$\frac{\partial L}{\partial a_i^l(u,v)} = \sum_{j,p,q} \frac{\partial L}{\partial z_j^{l+1}(p,q)} \frac{\partial z_j^{l+1}(p,q)}{\partial a_i^l(u,v)} \tag{7}$$

$$= \sum_{j,p,q} \frac{\partial L}{\partial z_j^{l+1}(p,q)} \omega_{j,i}^l(p-u, q-v) \tag{8}$$

$$= \sum_j \frac{\partial L}{\partial z_j^{l+1}(u,v)} * \omega_{j,i}^l(-u,-v) \tag{9}$$

3 Implementation

In Caffe [1], the convolutional layer is implemented as a matrix multiplication. First, the inputs are re-organized so that the convolutional operation is converted to a multiplication. Then, call the BLAS interface.

[*]University of Science and Technology of China, wjf2006@mail.ustc.edu.cn

The implementation implicitly requires to solve the following problem. Given a function

$$\epsilon = f(\mathbf{Z}) \tag{10}$$

$$\mathbf{Z} = \mathbf{X}\mathbf{Y} \tag{11}$$

how to compute the partial derivative $\partial\epsilon/\partial\mathbf{X}$ and $\partial\epsilon/\partial\mathbf{Y}$.

Note both $\mathbf{X}$ and $\mathbf{Y}$ are the matrix, but ϵ is a scalar. Here we derive the general form and the result can be directly used.

First, we have

$$\begin{aligned}
\frac{\partial\epsilon}{\partial X_{p,q}} &= \sum_{i,j} \frac{\partial\epsilon}{Z_{i,j}} \frac{\partial Z_{i,j}}{\partial X_{p,q}} && (12)\\
&= \sum_{i,j} \frac{\partial\epsilon}{Z_{i,j}} \frac{\partial \sum_m X_{i,m} Y_{m,j}}{\partial X_{p,q}} && (13)\\
&= \sum_{i,j} \frac{\partial\epsilon}{Z_{i,j}} \delta_{i,p} Y_{q,j} && (14)\\
&= \sum_{j} \frac{\partial\epsilon}{\partial Z_{p,j}} Y_{q,j} && (15)
\end{aligned}$$

Thus, we can get

$$\frac{\partial\epsilon}{\partial\mathbf{X}} = \frac{\partial\epsilon}{\partial\mathbf{Z}} \mathbf{Y}^T \tag{16}$$

Next, let's see $\frac{\partial\epsilon}{\partial\mathbf{Y}}$,

$$\begin{aligned}
\frac{\partial\epsilon}{\partial Y_{p,q}} &= \sum_{i,j} \frac{\partial\epsilon}{\partial Z_{i,j}} \frac{\partial Z_{i,j}}{\partial Y_{p,q}} && (17)\\
&= \sum_{i,j} \frac{\partial\epsilon}{\partial Z_{i,j}} \frac{\partial \sum_m X_{i,m} Y_{m,j}}{\partial Y_{p,q}} && (18)\\
&= \sum_{i,j} \frac{\partial\epsilon}{\partial Z_{i,j}} X_{i,p} \delta_{j,q} && (19)\\
&= \sum_{i} \frac{\partial\epsilon}{\partial Z_{i,q}} X_{i,p} && (20)
\end{aligned}$$

Thus, we can get

$$\frac{\partial\epsilon}{\partial\mathbf{Y}} = \mathbf{X}^T \frac{\partial\epsilon}{\partial\mathbf{Z}} \tag{21}$$

In the derivative, the matrix $\mathbf{X}$ or $\mathbf{Y}$ is transposed. The $\frac{\partial\epsilon}{\partial\mathbf{Z}}$ exists and is not transposed. The position of $\mathbf{X}$ or $\mathbf{Y}$ is consistent with the position in Eqn. 15.

4 Transform

The convolutional operation is closely related with the Circular Discrete Fourier Transform. The summarization goes as follows.

The following equation may be useful to check the correctness of the inverse transform.

$$\delta(x) = \int_{-\infty}^{\infty} e^{2\pi i p x} \mathrm{d}p \tag{22}$$

4.1 Fourier Transform

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-ix\omega} \mathrm{d}x \tag{23}$$

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{ix\omega} \mathrm{d}\omega \tag{24}$$

If $z(x) = f(x) * g(x)$, we have $Z(\omega) = F(\omega)G(\omega)$.

4.2 Discrete-Time Fourier Transform

$$F(\omega) = \sum_{n=-\infty}^{\infty} f(n) e^{-j\omega n} \tag{25}$$

$$f(n) = \frac{1}{2\pi} \int_{\langle 2\pi \rangle} F(\omega) e^{j\omega n} \mathrm{d}\omega \tag{26}$$

4.3 Discrete Fourier Transform

$$F(k) = \sum_{n=0}^{N-1} f(n) e^{-j2\pi kn/N} \tag{27}$$

$$f(n) = \frac{1}{N} \sum_{k=0}^{N-1} F(k) e^{j2\pi kn/N} \tag{28}$$

Here, we assume that two signals $f(n)$ and $g(n)$ are 0 if $n < 0$ or $n \geq N$. and

$$f_N(n) = \sum_{k=-\infty}^{+\infty} f(n - kN) \tag{29}$$

$$g_N(n) = \sum_{k=-\infty}^{+\infty} g(n - kN) \tag{30}$$

Thus, $f_N(n)$ and $g_N(n)$ are two periodic signals. The circular convolution is

$$(f_N \circledast g_N)(n) = \sum_{m=0}^{N-1} f_N(m) g_N(n - m) = z(n) \tag{31}$$

Then, we have $F(k)G(k) = Z(k)$.

References

[1] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.