Non-local Neural Networks

Cheng Guan

August 17, 2018

1. Non-local Neural Networks

The authors first give a general definition of non-local operations and then They provide several specific instantiations of it.

1.1. Formulation

Following the non-local mean operation [1], they define a generic non-local operation in deep neural networks as Eq.1:

$$\mathbf{y}_i = \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j)\, g(\mathbf{x}_j) \tag{1}$$

Here i is the index of an output position (in space, time, or spacetime) whose response is to be computed and j is the index that enumerates all possible positions. $\mathbf{x}$ is the input signal (image, sequence, video; often their features) and y is the output signal of the same size as $\mathbf{x}$. A pairwise function f computes a scalar (representing relationship such as affinity) between i and all j. The unary function g computes a representation of the input signal at the position j. The response is normalized by a factor $\mathcal{C}(\mathbf{x})$.

1.2. Instantiations

For simplicity, The authors only consider g in the form of a linear embedding: $g(\mathbf{x}_j) = W_g \mathbf{x}_j$, where W_g is a weight matrix to be learned. This is implemented as, *e.g.*, 1×1 convolution in space or $1 \times 1 \times 1$ convolution in spacetime.

Next They discuss choices for the pairwise function f.

Gaussian. Following the non-local mean [1] and bilateral filters [3], a natural choice of f is the Gaussian function. In this paper They consider:

$$f(\mathbf{x}_i, \mathbf{x}_j) = e^{\mathbf{x}_i^\top \mathbf{x}_j} \tag{2}$$

Embedded Gaussian. A simple extension of the Gaussian function is to compute similarity in an embedding space. In this paper they consider:

$$f(\mathbf{x}_i, \mathbf{x}_j) = e^{\theta(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)} \tag{3}$$

Dot product. f can be defined as a dot-product similarity:

$$f(\mathbf{x}_i, \mathbf{x}_j) = \theta(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) \tag{4}$$

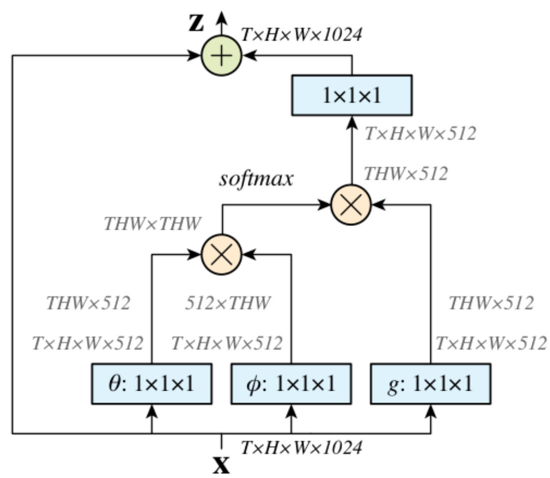

Figure 1. A spacetime non-local block.

Here The authors adopt the embedded version. In this case, They set the normalization factor as $\mathcal{C}(\mathbf{x}) = N$, where N is the number of positions in x, rather than the sum of f , because it simplifies gradient computation. A normalization like this is necessary because the input can have variable size.

Concatenation. Concatenation is used by the pairwise function in Relation Networks for visual reasoning. They also evaluate a concatenation form of f :

$$f(\mathbf{x}_i, \mathbf{x}_j) = \textbf{ReLU}\left(w_f^\top \left[\theta(\mathbf{x}), \phi(\mathbf{x})\right]\right) \tag{5}$$

The above several variants demonstrate the flexibility of their generic non-local operation. The authors believe alternative versions are possible and may improve results.

2. Non-local Block

The authors wrap the non-local operation in Eq.1 into a non-local block that can be incorporated into many existing architectures. They define a non-local block as:

$$\mathbf{z}_i = W_z \mathbf{y}_i + \mathbf{x}_i \tag{6}$$

where y_i is given in Eq.1 and $\mathbf{x}_i$ denotes a residual connection [2]. The residual connection allows us to insert a new non-local block into any pre-trained model, without breaking its initial behavior.

Implementation of Non-local Blocks. This follows the bottleneck design of [2] and reduces the computation of a block by about a half. The weight matrix W_z in Eq.6 computes a position-wise embedding on y_i , matching the number of channels to that of x,as shown in Fig. 1.

References

[1] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *CVPR*, 2005. 1

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 1, 2

[3] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV*, 1998. 1