Non-local Neural Networks

Liangjie Cao

Aug. 15, 2018

1. Non-local network advantage

In deep neural networks, capturing long-term dependencies is critical. For continuous data (such as speech-speaking languages), loop operations are the primary solution to long-term dependency problems in the time domain. For image data, long-distance dependencies are modeled by large receptive fields formed by a large number of convolution operations.

Convolution operations or loop operations are all processing local neighborhoods in space or time. In this way, long-distance dependencies can only be captured when these operations are applied repeatedly, and the signal can be continuously transmitted through the data. Repeated local operations have some limitations: first, the computational efficiency is low; second, the optimization difficulty is increased; finally, these challenges lead to multi-hop dependency modeling, for example, when messages need to be passed back and forth between long distances, it is very difficult .

In this paper, they propose non-local operations as an efficient, simple, and versatile component, and use deep neural networks to capture long-range dependencies. The non-local operations we propose are inspired by the general meaning of classical non-local operations in computer vision. Intuitively, the calculated response of a non-local operation at a location is a weighted sum of features at all locations in the input profile (Figure 1). A set of locations can be in space, time, or time and space, suggesting that our operations can be applied to image, sequence, and video problems.

2. Non-local Neural Networks

A non-local operation 1 is a flexible building block and can be easily used together with convolutional/recurrent layers. It can be added into the earlier part of deep neural networks, unlike fc layers that are often used in the end. This allows us to build a richer hierarchy that combines both non-local and local information.

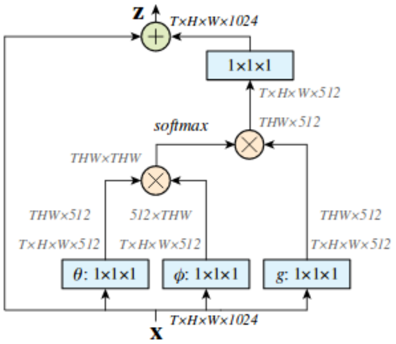

Figure 1. Non-local network

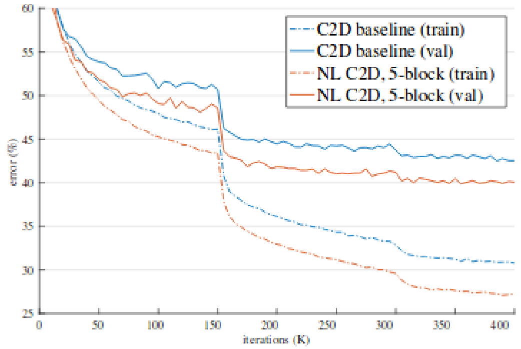

Figure 2. Iterations

3. Details

Our models are pre-trained on ImageNet [2].Unless specified, we fine-tune our models using 32-frame input clips. These clips are formed by randomly cropping out 64 consecutive frames from the original full-length video and then dropping every other frame. The spatial size is 224224 pixels, randomly cropped from a scaled video whose shorter side is randomly sampled in [256, 320] pixels, following [1]. We train on an 8-GPU machine and each GPU has 8 clips in a mini-batch (so in total with a mini-batch size of 64 clips). We train our models for 400k iterations in total, starting with a learning rate of 0.01 and reducing it by a factor of 10 at every 150k iterations (see also Figure 2). We use a momentum of 0.9 and a weight decay of 0.0001. They adopt dropout after the global pooling layer, with a dropout ratio of 0.5. We fine-tune our models with BatchNorm (BN) enabled when it is applied. This is in contrast to common practice [20] of fine-tuning ResNets, where BN was frozen. We have found that enabling BN in our application reduces overfitting.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[2] L. Wang, Y. Qiao, X. Tang, and L. Van Gool. Actionness estimation using hybrid fully convolutional networks. In *CVPR*, 2016. 1