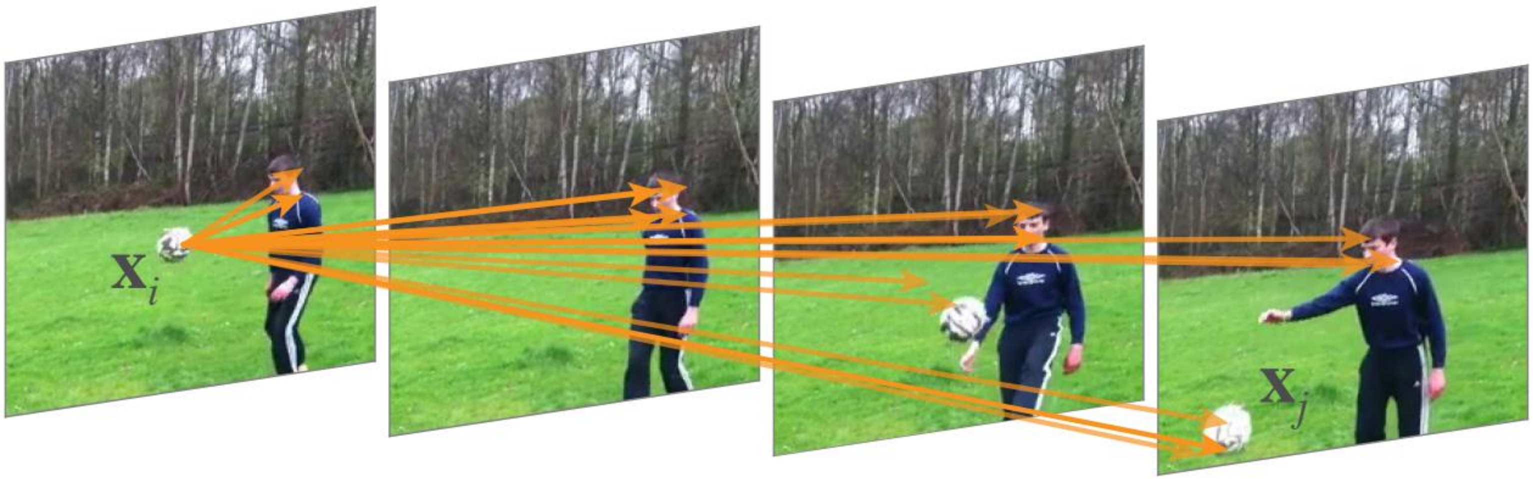
Non-local Neural Networks

Xiaolong Wang[1,2*] Ross Girshick[2] Abhinav Gupta[1] Kaiming He[2]
[1]Carnegie Mellon University [2]Facebook AI Research

Introduction

- Long-range dependencies can only be captured when recurrent and convolutional operations are applied repeatedly, propagating signals progressively through the data.
  - Computational inefficient
  - Optimization difficulties
  - Multi-hop dependency modeling is difficult.
- This paper presents ***non-local*** **operations** as an efficient, simple, and generic component for capturing long-range dependencies with deep neural networks.

Non-local Neural Network

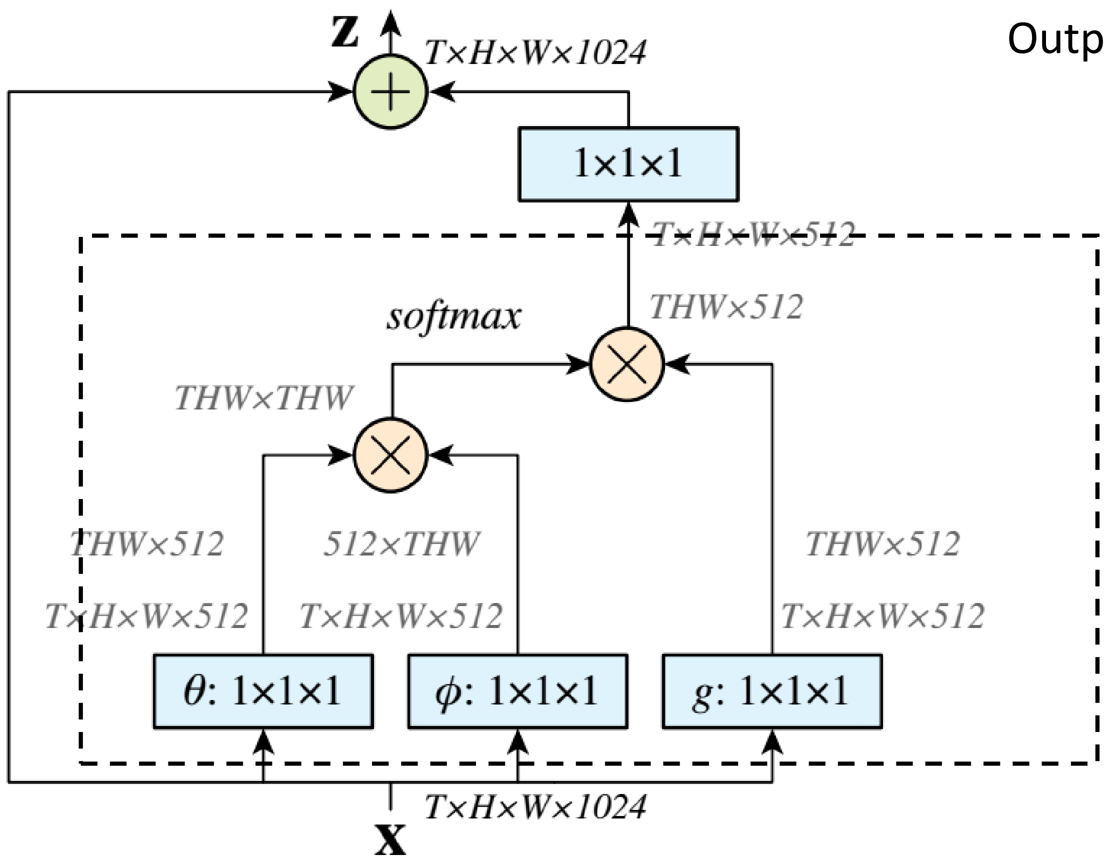


A non-local operation computes the response at a position as a weighted sum of the features at *all positions* in the input feature maps.

Non-local Block

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

Output feature map, Equal-sized as input



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

Input feature map

Formulation

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

- i: index of an output position
- x: input signal
- y: output signal with size equal to x
- C: normalization factor

Instantiation

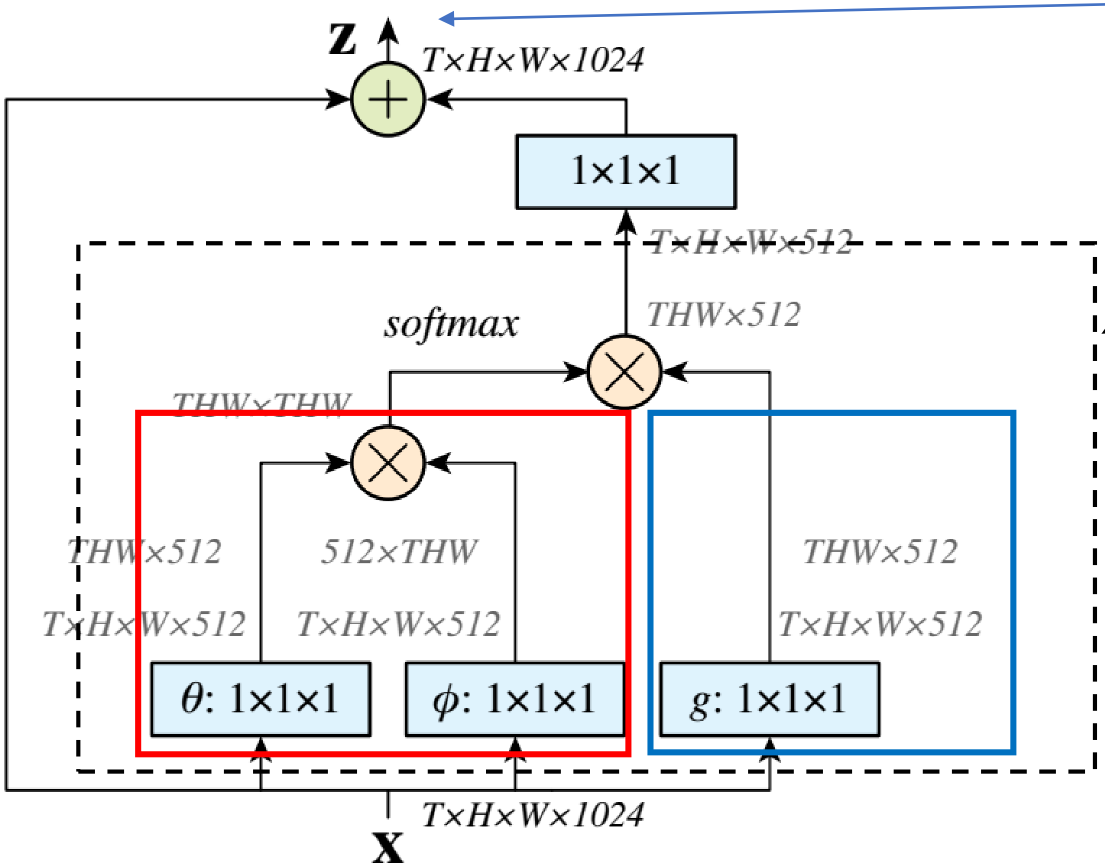
- $g(\mathbf{x}_j) = W_g\mathbf{x}_j$, linear embedding, W_g weight matrix
- *f :*
  - Gaussian: $f(\mathbf{x}_i, \mathbf{x}_j) = e^{\mathbf{x}_i^T \mathbf{x}_j}$
  - Embedded Gaussian: $f(\mathbf{x}_i, \mathbf{x}_j) = e^{\theta(\mathbf{x}_i)^T \phi(\mathbf{x}_j)}$.

  Self Attention module: $\mathbf{y} = softmax(\mathbf{x}^T W_\theta^T W_\phi \mathbf{x}) g(\mathbf{x})$

  - Dot product: $f(\mathbf{x}_i, \mathbf{x}_j) = \theta(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$.
  - Concatenation: $f(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{ReLU}(\mathbf{w}_f^T [\theta(\mathbf{x}_i), \ \phi(\mathbf{x}_j)])$.

$$\theta(\mathbf{x}_i) = W_\theta \mathbf{x}_i \qquad \phi(\mathbf{x}_j) = W_\phi \mathbf{x}_j$$

Non-local Block



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

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

$g(x_{i,}\ x_j)$

Build connection b/w any two points: $f(x_{i,}\ x_j)$

Input feature map

Non-local Neural Network

- Capture long-range dependencies directly by computing interactions between any two positions
- Achieve their best results even with only a few layers
- Maintain the variable input sizes and can be easily combined with other operations

Video Verification model

- 2D ConvNet baseline (C2D)
  - 2D kernel , 1×*k*×*k*
- Inflated 3D ConvNet (I3D)
  - 3D kernel, *t*×*k*×*k*
- Non-local network
  - Insert non-local blocks into C2D or I3D
  - 1,5, or 10 non-local blocks

layer		output size
conv_1	7×7, 64, stride 2, 2, 2	$16\times112\times112$
pool_1	$3\times3\times3$ max, stride 2, 2, 2	$8\times56\times56$
res_2	$\begin{bmatrix}1\times1, 64\\ 3\times3, 64\\ 1\times1, 256\end{bmatrix}\times3$	$8\times56\times56$
pool_2	$3\times1\times1$ max, stride 2, 1, 1	$4\times56\times56$
res_3	$\begin{bmatrix}1\times1, 128\\ 3\times3, 128\\ 1\times1, 512\end{bmatrix}\times4$	$4\times28\times28$
res_4	$\begin{bmatrix}1\times1, 256\\ 3\times3, 256\\ 1\times1, 1024\end{bmatrix}\times6$	$4\times14\times14$
res_5	$\begin{bmatrix}1\times1, 512\\ 3\times3, 512\\ 1\times1, 2048\end{bmatrix}\times3$	$4\times7\times7$
global average pool, fc		$1\times1\times1$

Baseline ResNet-50 C2D model for video

Non-local Block

Examples of the behavior of a non-local block in res_3 computed by a 5-block non-local model trained on Kinetics. The 20 highest weighted arrows for each $\mathbf{x}_i$ are visualized.

Results

model, R50	top-1	top-5
C2D baseline	71.8	89.7
Gaussian	72.5	90.2
Gaussian, embed	72.7	**90.5**
dot-product	**72.9**	90.3
concatenation	72.8	**90.5**

(a) **Instantiations**: 1 non-local block of different types is added into the C2D baseline. All entries are with ResNet-50.

model, R50	top-1	top-5
baseline	71.8	89.7
res_2	72.7	90.3
res_3	**72.9**	90.4
res_4	72.7	**90.5**
res_5	72.3	90.1

(b) **Stages**: 1 non-local block is added into different stages. All entries are with ResNet-50.

model		top-1	top-5
R50	baseline	71.8	89.7
	1-block	72.7	90.5
	5-block	$\boxed{73.8}$	91.0
	10-block	**74.3**	**91.2**
R101	baseline	$\boxed{73.1}$	91.0
	1-block	74.3	91.3
	5-block	**75.1**	**91.7**
	10-block	**75.1**	91.6

(c) **Deeper non-local models**: we compare 1, 5, and 10 non-local blocks added to the C2D baseline. We show ResNet-50 (top) and ResNet-101 (bottom) results.

5-block ResNet-50 has only ~**70% parameters** and ~**80% FLOPs** of the ResNet-101 baseline

Results

model, R101	params	FLOPs	top-1	top-5
C2D baseline	1×	1×	73.1	91.0
$\text{I3D}_{3\times3\times3}$	1.5×	1.8×	74.1	91.2
$\text{I3D}_{3\times1\times1}$	**1.2**×	1.5×	74.4	91.1
NL C2D, 5-block	**1.2**×	**1.2**×	**75.1**	**91.7**

(e) **Non-local *vs*. 3D Conv**: A 5-block non-local C2D *vs*. inflated 3D ConvNet (I3D) [7]. All entries are with ResNet-101. The numbers of parameters and FLOPs are relative to the C2D baseline (43.2M and 34.2B).

	model	top-1	top-5
R50	C2D baseline	71.8	89.7
	I3D	73.3	90.7
	NL I3D	**74.9**	**91.6**
R101	C2D baseline	73.1	91.0
	I3D	74.4	91.1
	NL I3D	**76.0**	**92.1**

(f) **Non-local 3D ConvNet**: 5 non-local blocks are added on top of our best I3D models. These results show that non-local operations are complementary to 3D convolutions.

Results

model	backbone	modality	top-1 val	top-5 val	top-1 test	top-5 test	avg test†
I3D in [7]	Inception	RGB	72.1	90.3	71.1	89.3	80.2
2-Stream I3D in [7]	Inception	RGB + flow	75.7	92.0	74.2	91.3	82.8
RGB baseline in [3]	Inception-ResNet-v2	RGB	73.0	90.9	-	-	-
3-stream late fusion [3]	Inception-ResNet-v2	RGB + flow + audio	74.9	91.6	-	-	-
3-stream LSTM [3]	Inception-ResNet-v2	RGB + flow + audio	77.1	93.2	-	-	-
3-stream SATT [3]	Inception-ResNet-v2	RGB + flow + audio	77.7	93.2	-	-	-
NL I3D [ours]	ResNet-50	RGB	76.5	92.6	-	-	-
	ResNet-101	RGB	**77.7**	**93.3**	-	-	**83.8**

Table 3. Comparisons with state-of-the-art results in **Kinetics**, reported on the val and test sets. We include the Kinetics 2017 competition winner's results [3], but their best results exploited audio signals (marked in gray) so were not vision-only solutions. †: "avg" is the average of top-1 and top-5 accuracy; individual top-1 or top-5 numbers are not available from the test server at the time of submitting this manuscript.

Results

method		AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
R50	baseline	38.0	59.6	41.0	34.6	56.4	36.5
	+1 NL	**39.0**	**61.1**	**41.9**	**35.5**	**58.0**	**37.4**
R101	baseline	39.5	61.4	42.9	36.0	58.1	38.3
	+1 NL	**40.8**	**63.1**	**44.5**	**37.1**	**59.9**	**39.2**
X152	baseline	44.1	66.4	48.4	39.7	63.2	42.2
	+1 NL	**45.0**	**67.8**	**48.9**	**40.3**	**64.4**	**42.8**

Table 5. Adding 1 non-local block to Mask R-CNN for COCO **object detection** and **instance segmentation**. The backbone is ResNet-50/101 or ResNeXt-152 [53], both with FPN [32].

model	AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}
R101 baseline	65.1	86.8	70.4
NL, +4 in head	66.0	87.1	71.7
NL, +4 in head, +1 in backbone	**66.5**	**87.3**	**72.8**

Table 6. Adding non-local blocks to Mask R-CNN for COCO **keypoint detection**. The backbone is ResNet-101 with FPN [32].

Summary

- The authors presented a new class of neural networks which capture long-range dependencies via non-local operations
- The non-local blocks can be combined with any existing architectures.
- A simple addition of non-local blocks provides solid improvement over baselines.