NON-LOCAL SELF-ATTENTION STRUCTURE FOR FUNCTION APPROXIMATION IN DEEP REINFORCEMENT LEARNING

Zhixiang Wang[1], Xi Xiao[1], Guangwu Hu[2], Yao Yao[1], Dianyan Zhang[1], Zhendong Peng[1], Qing Li[3], Shutao Xia[1,3]

[1] Tsinghua University, Beijing, China

[2] School of Computer Science, Shenzhen Institute of Information Technology, Shenzhenon, China

[3] Southern University of Science and Technology, Pengcheng Laboratory, Shenzhen, China

Abstract

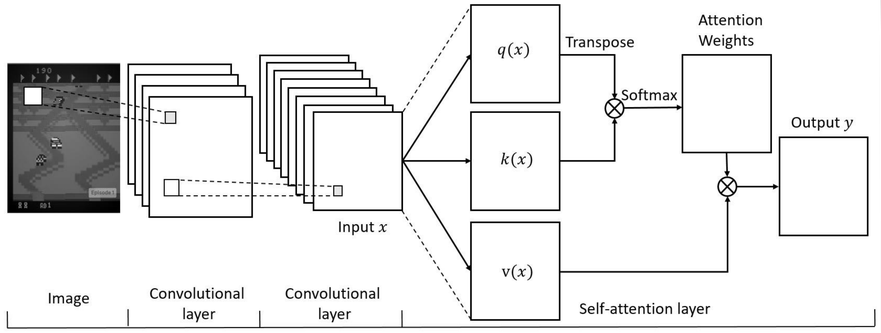
Reinforcement learning is a framework to make sequential decisions. The combination with deep neural networks further improves the ability of this framework. Convolutional neural networks make it possible to make sequential decisions based on raw pixels information directly and make reinforcement learning achieve satisfying performances in series of tasks. However, convolutional neural networks still have own limitations in representing geometric patterns and long-term dependencies that occur consistently in state inputs. To tackle with the limitation, we propose the self-attention architecture to augment the original network. It provides a better balance between ability to model long-range dependencies and computational efficiency. Experiments on Atari games illustrate that self-attention structure is significantly effective for function approximation in deep reinforcement learning.

Reinforcement Learning Formulation

In reinforcement learning, the policy $\pi(a|s)$ was represented by a machine learning module. In deep reinforcement learning, this machine learning module is a neural network. State value function and action value function are also represented by machine learning modules (e.g. convolutional neural network) to guide the policy to update itself towards a right direction. Functions $\pi(a|s)$, $V(s)$ and $Q(s,a)$ need to be approximated by proper machine learning modules.

The Proposed Model

The input pixel image is first processed by several convolution layers and transformed into image features $x \in R^{C\times H\times W}$ where C means 'channel', H means 'height' and W means 'width'. x_i represents the element in ith position among all the $C \times H \times W$ positions. The input x is duplicated into 3 copies which are query x_{query}, key x_{key} and value x_{value}.Then we do transformation and get $q(x_{query}) = W_{query}x_{query}$, $k(x_{key}) = W_{key}x_{key}$, $v(x_{value}) = W_{value}x_{value}$ separately. The metric W_{query}, W_{key} and W_{value} are the network parameters to be learned.



Since $x_{query}, x_{key}, x_{value}$ are the same, we just use $q(x)$, $k(x)$, $v(x)$ to represent $q(x_{query})$, $k(x_{key})$, $v(x_{value})$ and $s_{ij} = q(x_i)k(x_j)$. The output of the self-attention layer has the same size as the input. When rebuilding the element in the jth position, $a_{j,i}$ indicates how much attention should be paid to the ith location.

$$a_{j,i} = \frac{\exp(s_{ij})}{\sum_{j=1}^{C\times H\times W}\exp(s_{ij})}$$

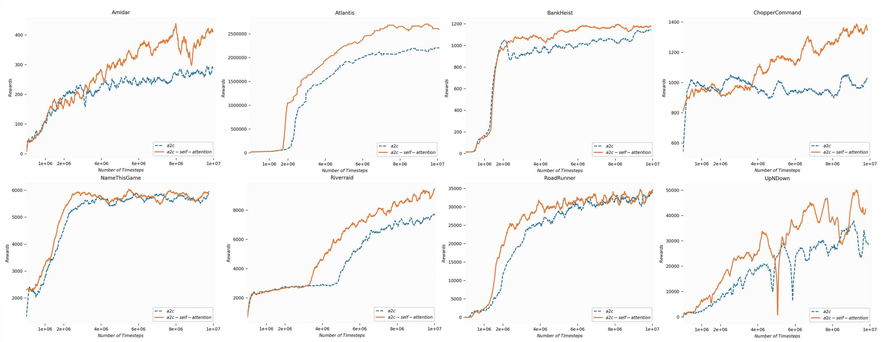
The output element in the jth position is calculated in the following way.

$$y_j = \sum_{i=1}^{C\times H\times W} a_{j,i}v(x_i)$$

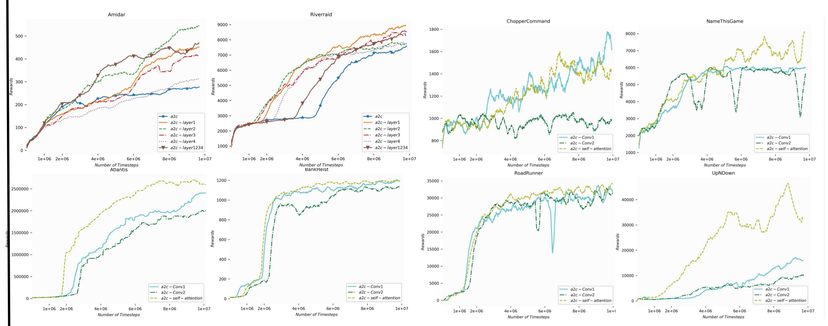
Finally, we multiply the output by a parameterized scalar β and add it back to the input feature map. Then we get the final output y_{out}.

$$y_{out} = \beta y + x$$

Experiment



Game	CNN	Self-attention	Game	CNN	Self-attention	Game	CNN	Self-attention
Amidar	275	**391**	Frostbite	250	**251**	PrivateEye	64.8	**144**
Asterix	5588	**6510**	Gopher	907	**924**	Qbert	12412	**14142**
Atlantis	1729268	**1958670**	Gravitar	335	**372**	Riverraid	7426	**9100**
BankHeist	1128	**1173**	Hero	19462	**19957**	RoadRunner	32470	**32984**
BeamRider	4184	**4252**	IceHockey	-6.8	**-6.5**	RoboTank	2.2	**2.8**
Berzerk	766	**785**	JamesBond	**2429**	2312	Seaquest	1701	**1727**
Bowling	23.7	**24.1**	Kangaroo	**559**	462	Skilling	-28680	**-16189**
Breakout	382	**387**	Krull	7886	**7903**	SpaceInvader	758	**803**
Centipede	**4178**	3812	KungFuMaster	**28564**	27334	StarGunner	44623	**45296**
Chopper Command	995	**1362**	Montezuma's Revenge	0.1187	**0.1209**	TimePilot	**3491**	3431
CrazyClimer	103959	**105340**	MsPacMan	1942	**2127**	Tutankhan	176	**185**
DemonAttack	18005	**22932**	NameThisGame	5658	**5750**	UpNDown	31341	**43661**
DoubleDunk	**-15.8**	-16.1	Phoenix	13188	**14938**	Venture	0	0
FishingDerby	**21.5**	16.6	Pitfall	-65.2	**-50.3**	YarsRevenge	**13793**	12365
Freeway	0.0017	0.0017	Pong	18.1	**19.5**	Zaxxon	15.3	**21.4**



Conclusion

In order to tackle with the weakness of convolutional networks in building up long-range dependencies, in this paper, we incorporate the self-attention mechanism into the deep reinforcement learning framework and propose the self-attention function approximation. This structure can build up relationships among different regions of the state input effectively with satisfying computational efficiency. Our experiments illustrate the effectiveness of the self-attention structure compared with a pure convolutional structure.