Relational recurrent neural networks

Adam Santoro*, Ryan Faulkner*, David Raposo*, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, Timothy Lillicrap

contact: adamsantoro@; rfaulk@; draposo@

Abstract

Memory-based neural networks model temporal data by leveraging an ability to remember information for long periods. It is unclear, however, whether they also have an ability to perform complex relational reasoning with the information they remember. Here, we first confirm our intuitions that standard memory architectures may struggle at tasks that heavily involve an understanding of the ways in which entities are connected -- i.e., tasks involving relational reasoning. We then improve upon these deficits by using a new memory module -- a Relational Memory Core (RMC) -- which employs multi-head dot product attention to allow memories to interact. Finally, we test the RMC on a suite of tasks that may profit from more capable relational reasoning across sequential information, and show large gains in RL domains (e.g. Mini PacMan), program evaluation, and language modeling, achieving state-of-the-art results on the WikiText-103, Project Gutenberg, and GigaWord datasets.

Take-aways

It may be useful to consider memory-memory interactions.

Self-attention is a powerful mechanism for computing memory-memory interactions, and for writing new information to memory.

A 2D-LSTM augmented with a row-size self-attention operation on their cell state -- a **Relational Memory Core** -- can vastly outperform other models on tasks involving relational reasoning.

Core is checked in to Sonnet.

Code

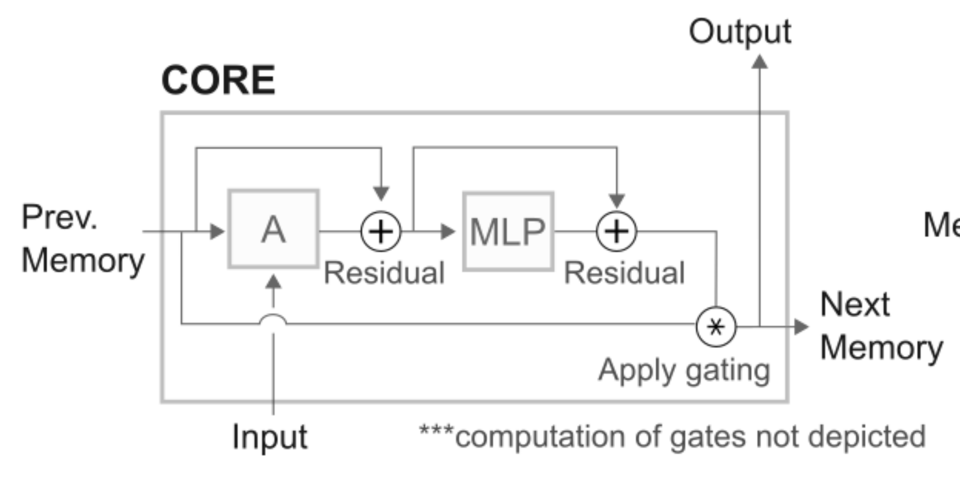
The core is available to use at:

https://cs.corp.google.com/piper///depot/google3/third_party/py/sonnet/python/modules/relational_memory.py

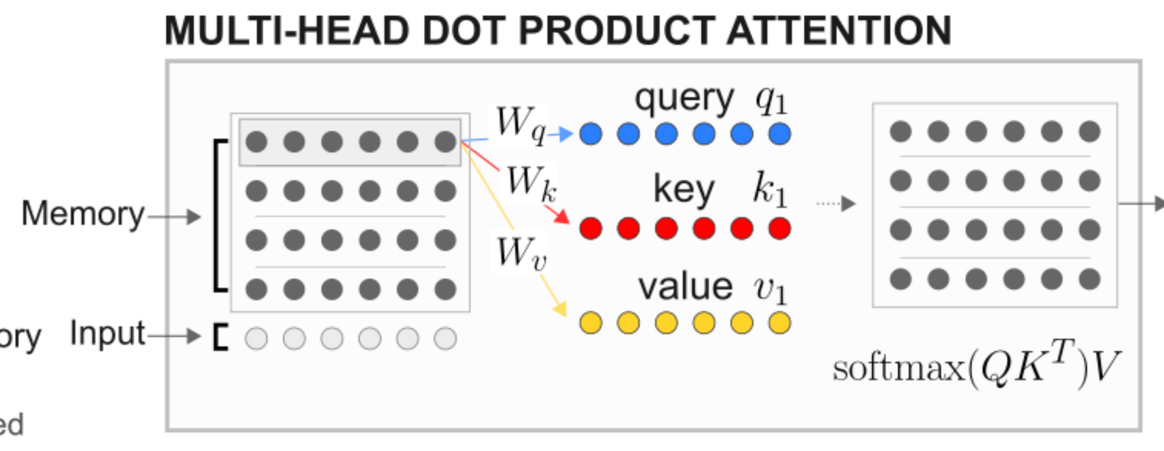
Acknowledgements

We thank Caglar Gulcehre, Matt Botvinick, Vinicius Zambaldi, Charles Blundell, Sebastien Racaniere, Chloe Hillier, Victoria Langston, for critical feedback, discussions, and support.

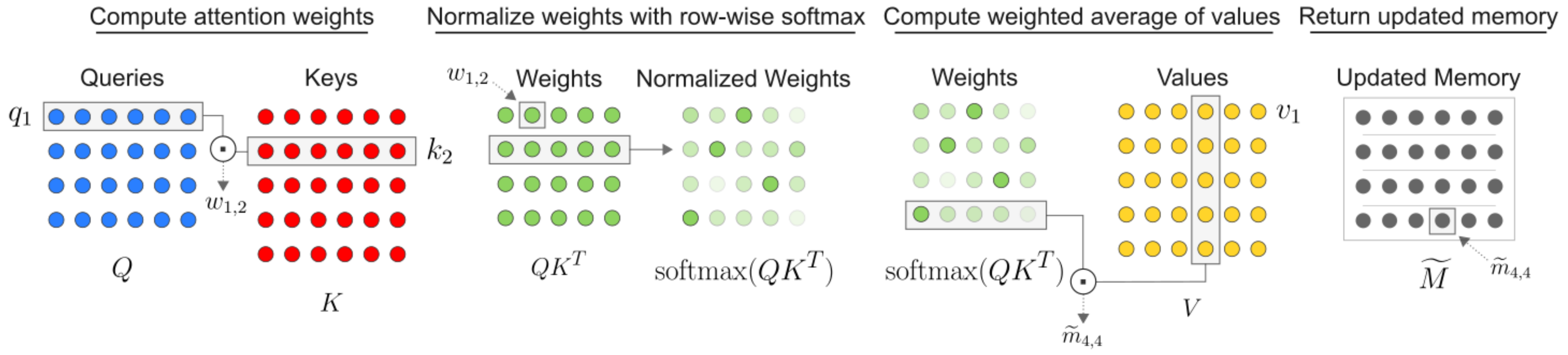
Model



(a)



(b)



Self-attention:

$$A_\theta(M) = \mathrm{softmax}\left(\frac{MW^q(MW^k)^T}{\sqrt{d_k}}\right)MW^v, \text{ where } \theta = (W^q, W^k, W^v)$$

Using self-attention to write to memory:

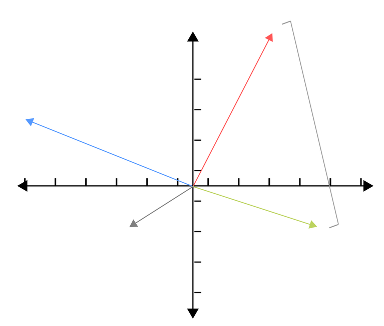
$$\widetilde{M} = \mathrm{softmax}\left(\frac{MW^q([M;x]W^k)^T}{\sqrt{d^k}}\right)[M;x]W^v$$

Embedding self-attention into an LSTM:

$$s_{i,t} = (h_{i,t-1}, m_{i,t-1})$$
$$f_{i,t} = W^f x_t + U^f h_{i,t-1} + b^f$$
$$i_{i,t} = W^i x_t + U^i h_{i,t-1} + b^i$$
$$o_{i,t} = W^o x_t + U^o h_{i,t-1} + b^o$$
$$m_{i,t} = \sigma(f_{i,t} + \tilde{b}^f) \circ m_{i,t-1} + \sigma(i_{i,t}) \circ \underbrace{g_\psi(\widetilde{m}_{i,t})}$$
$$h_{i,t} = \sigma(o_{i,t}) \circ \tanh(m_{i,t})$$
$$s_{i,t+1} = (m_{i,t}, h_{i,t})$$

Tasks

What is the N^{th} farthest from vector m?

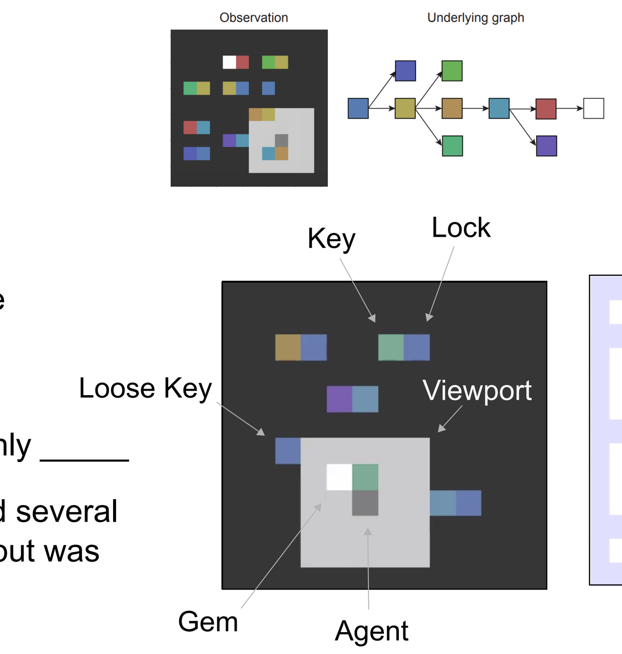


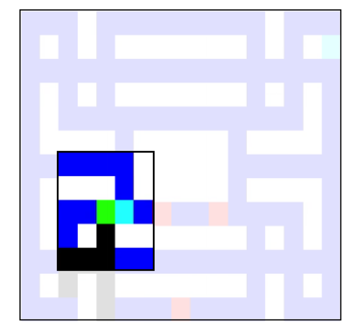
```
x = 339
for [19]:
    x += 597
        for[94]:
    x += 875
x if 428 < 778 else 652
print(x)
```

Super Mario Land is a 1989 side scrolling platform video _____

It had 24 step programming abilities, which meant it was highly _____

A gold dollar had been proposed several times in the 1830s and 1840s , but was not initially _____





Nth farthest | Program Evaluation | Language Modeling | BoxWorld | Mini-Pacman

Supervised Learning | Reinforcement Learning

Addition (nesting = 2, literal length = 7):

```
x=473278230+(1257657+32721978)
print(x % 10**length)
A: 7257865
```

Control (nesting = 3, literal length = 3):

```
x = 221 if ((411 if 918 > 314 else 680) + 321) < 778 else 652
print(x % 10**length)
A: 221
```

Full Program (nesting = 2, literal length = 5):

```
x=82930-31249
for[6]
  x+=98315
print(x % 10**length)
A: 641571
```

```
x = (28694 if 89425 > 31990 else 38662)
for[5]
  x += 54926
print(x % 10**length)
A: 303324
```

```
x = (
  x=76957
  for[7]
    x += 62117)
for[8]
  x+=90285
print(x % 10**length)
A: 234056
```

Results

Validation and test perplexities on WikiText-103, Project Gutenberg, and GigaWord v5.

	WikiText-103		Gutenberg		GigaWord
	Valid.	Test	Valid	Test	Test
LSTM [40]	-	48.7	-	-	-
Temporal CNN [41]	-	45.2	-	-	-
Gated CNN [42]	-	37.2	-	-	-
LSTM [32]	34.1	34.3	41.8	45.5	43.7
Quasi-RNN [43]	32	33	-	-	-
Relational Memory Core	**30.8**	**31.6**	**39.2**	**42.0**	**38.3**

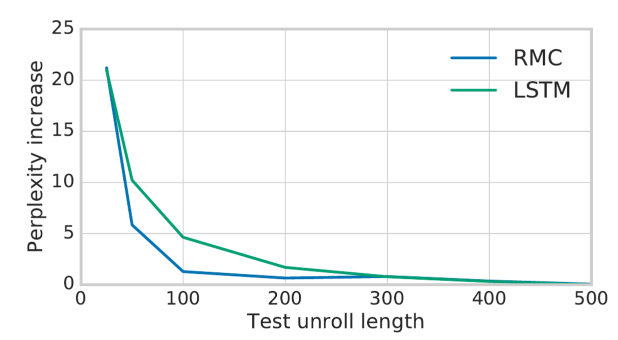
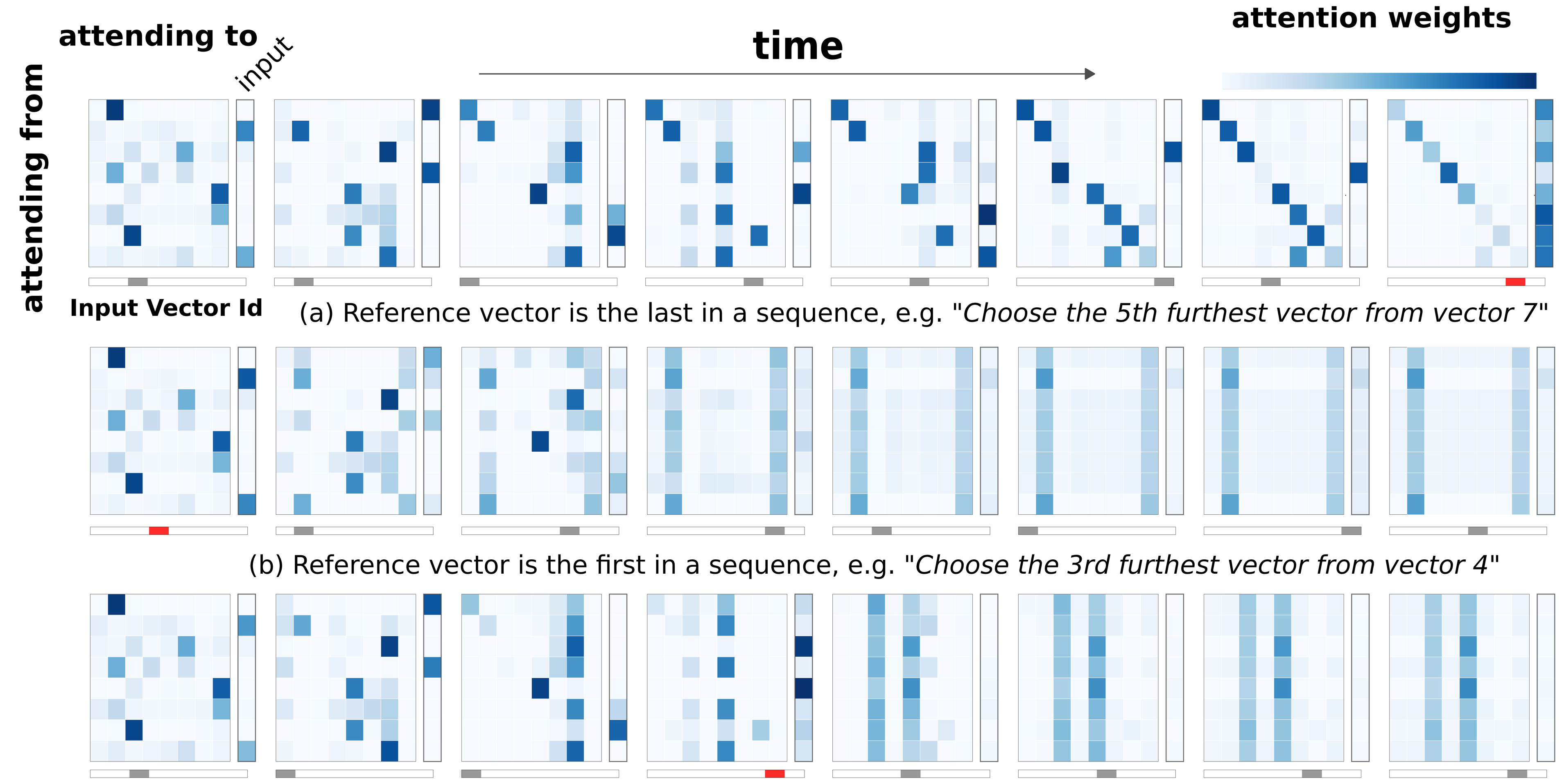


Table 1: Test per character Accuracy on Program Evaluation and Memorization tasks.

Model	Add	Control	Program	Copy	Reverse	Double
LSTM [3, 37]	99.8	97.4	66.1	99.8	99.7	99.7
EntNet [38]	98.4	98.0	73.4	91.8	**100.0**	62.3
DNC [5]	99.4	83.8	69.5	**100.0**	**100.0**	**100.0**
Relational Memory Core	**99.9**	**99.6**	**79.0**	**100.0**	**100.0**	99.8



(a) Reference vector is the last in a sequence, e.g. "*Choose the 5th furthest vector from vector 7*"

(b) Reference vector is the first in a sequence, e.g. "*Choose the 3rd furthest vector from vector 4*"

(c) Reference vector comes in the middle of a sequence, e.g. "*Choose the 6th furthest vector from vector 6*"