Explainable inference on sequential data with Memory Augmented Neural Network

BENAZHA Hamed

November 16, 2021

Introduction

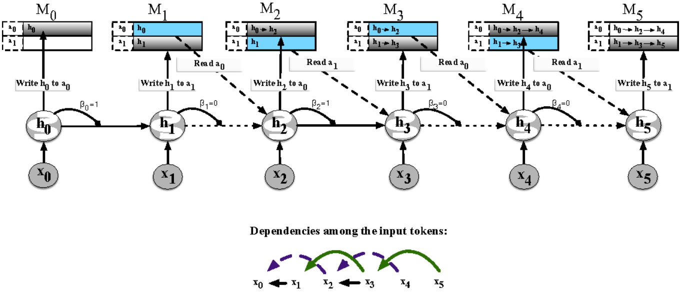
Introduction

- Memory Augmented Neural Network were originally used to solves problem that classical RNN cannot solves (reversing, sorting...)
- Like a computer, it uses external memory and is Turing Complete
- It's also capable of basic reasoning (e.g. with the babi dataset)
- The external memory provide some valuable insight on the decision process of the network

Different type of MANN

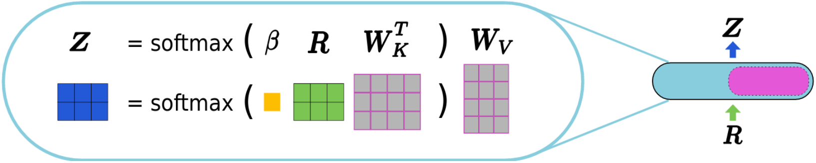
TARDIS

- Bengio et al [3] proposed an architecture to help with long-term dependencies in LSTM
- The memory here serves as a buffer (and also as a shortcut) for hidden states
- The idea is comparable to Residual Neural Network, but with residual connections through time



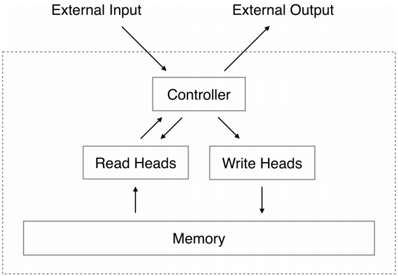
Hopfield Networks

- Ramsauer et al [5] proposed a generalization of the attention
- This model allow standard (and recurrent) neural networks to be augmented with an associative memory
- The associative recall is based on Modern Hopfield Network
- The memory is static (i.e. not interactive), learned during training and doesn't change during inference



Neural Turing Machine and derivative

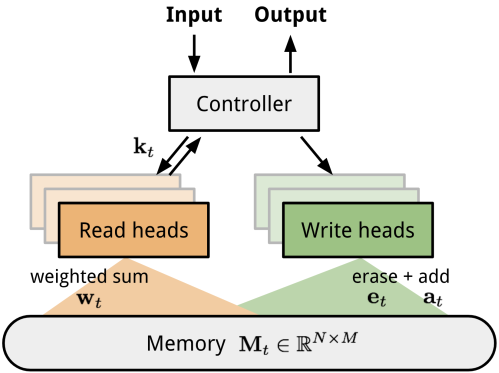
- This model was proposed Graves et al [1] and is based on Von Neumann model
- An extension was also proposed by Graves et al [2]
- The memory is dynamic
- We interact (reading, writing) in a differentiable manner with the memory at each time-step



Differentiable Neural Computer

Neural Turing Machine

- Graves et al [1] proposed a MANN architecture
- The controller can now read and write in a differentiable manner by using attention mechanism
- To do that, the controller emits a read and a write attention vector



Content Based Addressing

- At each time-step, the controller emit a key
- The key is compared to each location in the memory according to a similarity measure
- A softmax is applied to the similarity score to obtain the attention vector

Reading and writing

- We can write the read vector as :

$$r_t = \sum_{i=1}^{N} w_t(i) M_t(i)$$

- where $\sum_i^N w_t(i) = 1, \; \forall i : 0 \leq w_t(i) \leq 1$

$$\begin{bmatrix} -0.5 & 0.01 & 3.1 \\ 0.2 & 0.6 & 1.2 \\ 0 & 0 & 0 \\ -0.1 & -0.05 & 0 \end{bmatrix}^\top \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 & 0.2 & 0 & -0.1 \\ 0.01 & 0.6 & 0 & -0.05 \\ 3.1 & 1.2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.6 \\ 1.2 \end{bmatrix}$$

Reading and writing

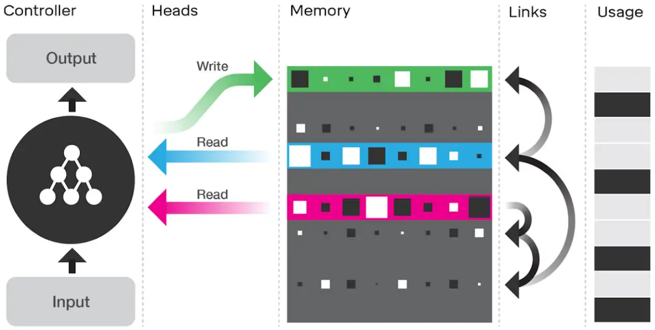
- The writing operation is inspired by the input and forget gates in LSTM

$$\tilde{\mathbf{M}}_t(i) = \mathbf{M}_{t-1}(i)[\mathbf{1} - w_t(i)\mathbf{e}_t] \quad \text{; erase}$$
$$\mathbf{M}_t(i) = \tilde{\mathbf{M}}_t(i) + w_t(i)\mathbf{a}_t \quad \text{; add}$$

Differentiable Neural Computer

- An extension to the NTM was proposed by Graves et al [2]
- The controller have now new ways to interact with the memory
- It can now also handle full memory issues

Illustration of the DNC architecture



Temporal memory linkage

- The controller can now read the memory cells sequentially in the order they were written
- This matrix $L \in R^{N \times N}$ tracks the order in which location have been written
- Example : If the memory location 4 was written after the location 2. Then the location 1 was written after location 4

$$L_t = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Dynamic Memory Allocation

- The DMA can free unused cell
- The objective of the dynamic memory allocation is to rewrite the memory content
- The allocation vector a_t indicate to what degree, each memory location is allocable
- For example if $a_t = [0.8, 0.4, 0.1, 0]$ then the first location is more allocable
- If $a_t = 0$ then the DNC is out of allocable memory location

Reading vector

- To compute the reading vector, the controller emits a reading mode vector $\pi_t \in \mathbb{R}^3$ where $\sum \pi_t(i) = 1$ and $0 \leq \pi_t(i) \leq 1$
- the read vector is the sum of the temporal interaction mode and the content retrieval modes

$$r = \pi_t(1)b_t + \pi_t(2)c_t + \pi_t(3)f_t$$

- where b, c, f are respectively the backward (temporal), content and forward(temporal)

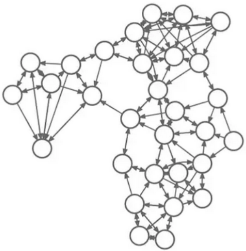
Write vector

- As seen earlier, the allocation vector a_t indicate to what degree, each memory location is allocable
- The controller also emits two scalars
- A scalar $g_t^w \in [0, 1]$ that governs writing intensity ($g_t^w = 0$ imply no writing)
- A scalar $g_t^a \in [0, 1]$ that governs the interpolation between a_t and c_t^w

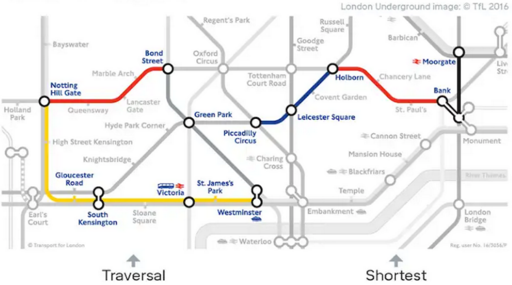
$$w_t = g_t^w \left[g_t^a a_t + (1 - g_t^a) c_t^w \right]$$

Experiments

Random Training Graph



London Underground



Underground Input:

(OxfordCircus, TottenhamCtRd, Central)
(TottenhamCtRd, OxfordCircus, Central)
(BakerSt, Marylebone, Circle)
(BakerSt, Marylebone, Bakerloo)
(BakerSt, OxfordCircus, Bakerloo)
...
(LeicesterSq, CharingCross, Northern)
(TottenhamCtRd, LeicesterSq, Northern)
(OxfordCircus, PiccadillyCircus, Bakerloo)
(OxfordCircus, NottingHillGate, Central)
(OxfordCircus, Euston, Victoria)

- 84 edges in total

Traversal Question:

(BondSt, _, Central),
(_, _, Circle), (_, _, Circle),
(_, _, Circle), (_, _, Circle),
(_, _, Jubilee), (_, _, Jubilee),

Answer:

(BondSt, NottingHillGate, Central)
(NottingHillGate, GloucesterRd, Circle)
...
(Westminster, GreenPark, Jubilee)
(GreenPark, BondSt, Jubilee)

Shortest Path Question:

(Moorgate, PiccadillyCircus, _)

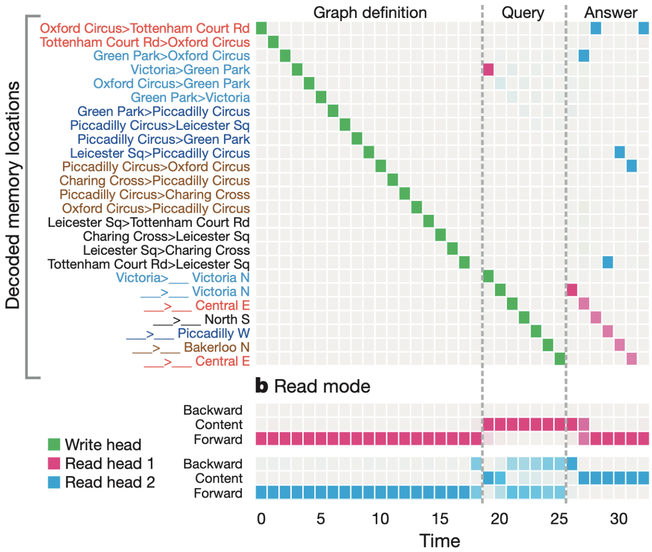
Answer:

(Moorgate, Bank, Northern)
(Bank, Holborn, Central)
(Holborn, LeicesterSq, Piccadilly)
(LeicesterSq, PiccadillyCircus, Piccadilly)

Experiments

https://www.youtube.com/watch?v=B9U8sI7TcMY

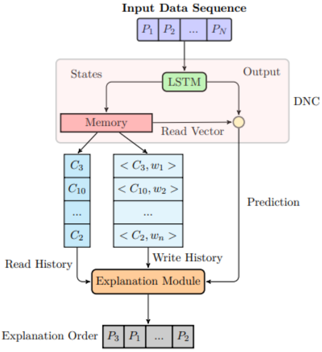
Using memory to generate explanation



Explainable MANN

Explainable inference via Memory Tracking

- La Rosa et al [4] proposed a new MANN architecture based on DNC
- They augmented the DNC with a memory tracking module (Also called explanation Module)



Explainable inference via Memory Tracking

- The memory module keeps track of every reading and writing operation
- At each time-step, it stores where the information is read/write and associate it with the input
- With all these information, the explanation module can extract insights from memory access during the inference

Example

- We take for example the babi stories datasets
- Consider inputs : $P_1 = X_1 X_2 X_3$ and $P_2 = X_4 X_5$
- Where P_i is the ith sentence and X_i a word
- Suppose each X_i is stored in a cell called C_i
- If during the inference, C_2 was read 5 times, C_1 was read 2 times and C_4 1 times, then the explanation module infer that P_1 decision weight is :

$$12.5 \times (5 + 2) = 87.5\%$$

Experiments

Earl woke up early to make some coffee. (48.3%) He wanted to be alert for work that day. (47.4%) The aroma woke up all his roommates. (0%) They wanted to make coffee too. (4.2%)

E1. All of his roommates made coffee (**CORRECT**) – E2. All of his roommates were sick of coffee.

Samantha had recently purchased a used car. (15.6%) She loved everything about the car except for the color. (30.3%) She took her car to her local paint shop. (31%) She got it painted a bright pink color. (23%)

E1. Samantha likes the color of her car now. (**CORRECT**) – E2. Samantha thinks her bus looks pretty now.

Tim didn't like school very much. (23.6%) His teacher told him he had a test on Friday. (15%) If he didn't pass this test, he could not go on the class trip. (4.5%) Tim decided to play with his kites instead of study for the test. (56.8%)

E1. Tim was unprepared and failed the test. – E2. Tim aced the test and passed with flying colors. (**WRONG**)

Neil took a ferry to the island of Sicily. (87.2%) The wind blew his hair as he watched the waves. (0%) Soon it docked, and he stepped onto the island. (0%) It was so breathtakingly beautiful. (12.7%)

E1. Neil enjoyed Sicily (**CORRECT**) – E2. Sicily was the worst place neil had ever been.

Thank you

Thank you

Alex Graves, Greg Wayne, and Ivo Danihelka.
Neural Turing Machines.
arXiv:1410.5401 [cs], December 2014.
arXiv: 1410.5401.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis.
Hybrid computing using a neural network with dynamic external memory.
Nature, 538(7626):471–476, October 2016.
Bandiera_abtest: a Cg_type: Nature Research Journals Number: 7626 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Learning algorithms;Network models Subject_term_id: learning-algorithms;network-models.

Caglar Gulcehre, Sarath Chandar, and Yoshua Bengio.
Memory Augmented Neural Networks with Wormhole Connections.
arXiv:1701.08718 [cs, stat], January 2017.
arXiv: 1701.08718.

Biagio La Rosa, Roberto Capobianco, and Daniele Nardi.
Explainable Inference on Sequential Data via Memory-Tracking.
In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 2006–2013, Yokohama, Japan, July 2020. International Joint Conferences on Artificial Intelligence Organization.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter.
Hopfield Networks is All You Need.

arXiv:2008.02217 [cs, stat], April 2021.
arXiv: 2008.02217.