Memory Augmented Neural Networks

Layton Hayes

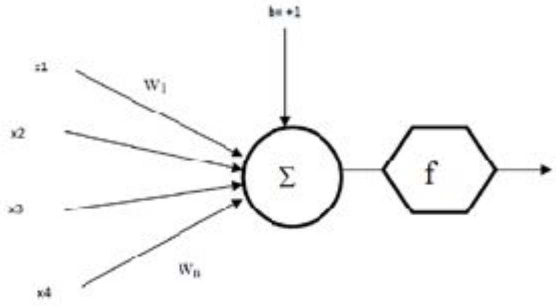
Memory in Neural Networks: RNNs

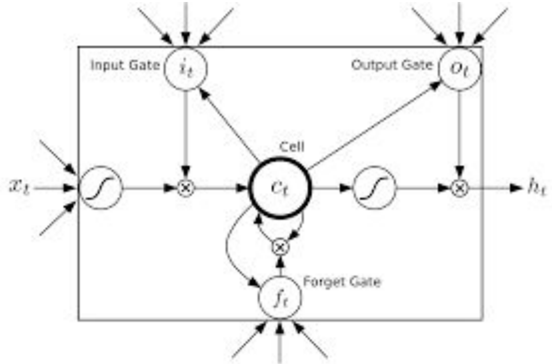
(Recurrent Neural Networks)

- Enables linking dependencies across examples / timesteps
- Many different models, LSTM being very powerful / popular

LSTM (Long Short-Term Memory):

- Each neuron stores some information about previous activations
- Memory is updated to some degree on every example, but degree varies





Problems with RNNs

- Short term memory -- memory isn't designed to last very long
  - Why?
    - Memory is overwritten and forgotten to some degree at every step
    - Even if the degree of overwrite is small, it compounds quickly
    - Always some amount of distortion
- Cell by cell updates means fragmentation
- Difficult to compartmentalize distinct memories

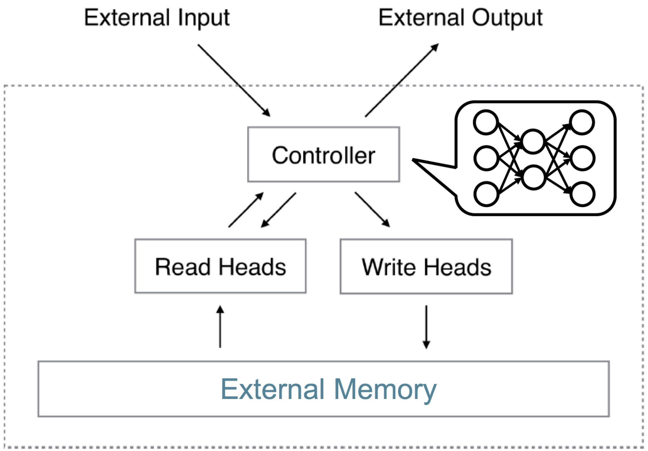
All because the memory is built into the network.

Memory outside Neural Networks: MANNs
(Memory Augmented Neural Network)

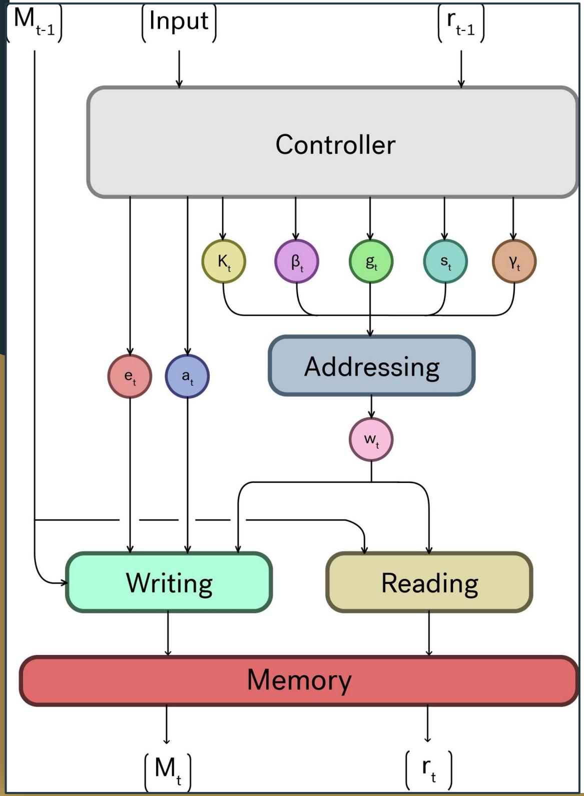
- Uses a neural network to interface an external memory
- Designed to solve problems of RNNs from last slide

NTM (Neural Turing Machine):

- Simple, early implementation of a MANN
- Controller outputs vectors to control read and write heads
- r / w heads interact with external memory
- Whole model is **end-to-end differentiable**



NTMs continued: how they work



Addressing Mechanism

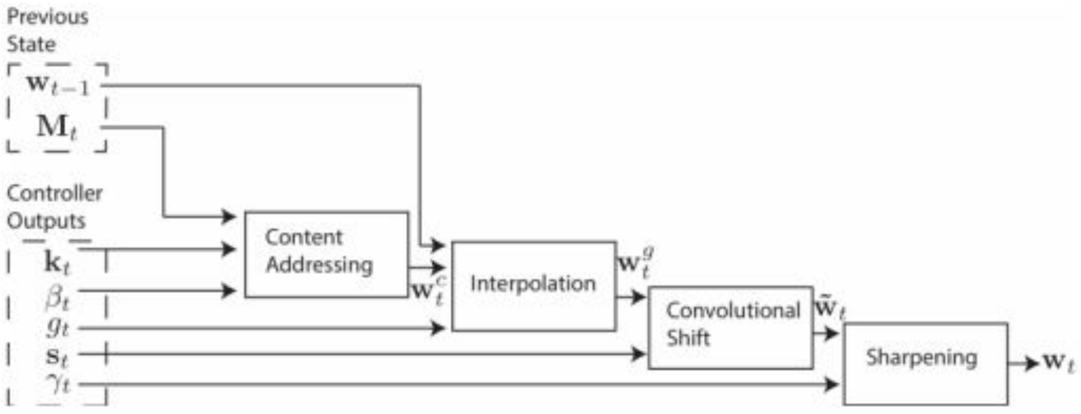


Figure 2: Flow Diagram of the Addressing Mechanism. The *key vector*, $\mathbf{k}_t$, and *key strength*, β_t, are used to perform content-based addressing of the memory matrix, $\mathbf{M}_t$. The resulting content-based weighting is interpolated with the weighting from the previous time step based on the value of the *interpolation gate*, g_t. The *shift weighting*, $\mathbf{s}_t$, determines whether and by how much the weighting is rotated. Finally, depending on γ_t, the weighting is sharpened and used for memory access.

NTM: reading and writing basic equations

$\mathbf{M}_t$ -> the N x M memory matrix,

w_t -> vector of weights, length N,

$$\sum_i w_t(i) = 1, \qquad 0 \leq w_t(i) \leq 1, \forall i.$$

r_t -> read vector

e_t -> erase vector, length M, range (0,1)

a_t -> add vector, length M

Reading:

$$\mathbf{r}_t \longleftarrow \sum_i w_t(i)\mathbf{M}_t(i)$$

Writing:

erase:

$$\tilde{\mathbf{M}}_t(i) \longleftarrow \mathbf{M}_{t-1}(i)\left[\mathbf{1} - w_t(i)\mathbf{e}_t\right]$$

add:

$$\mathbf{M}_t(i) \longleftarrow \tilde{\mathbf{M}}_t(i) + w_t(i)\,\mathbf{a}_t$$

NTM vs LSTM: Copy problem

Input:

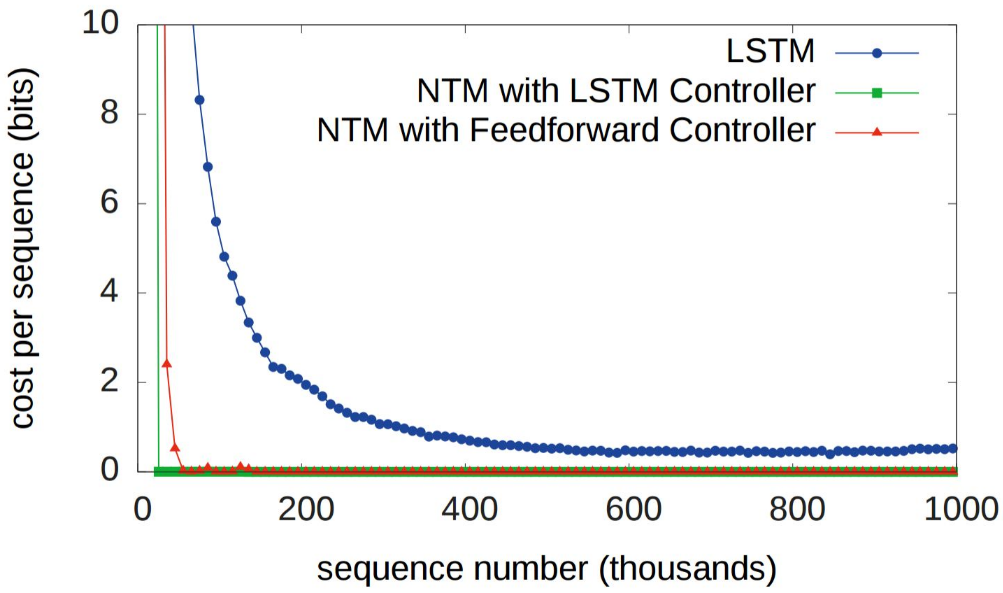
Sequence of length L,

then nothing for L steps.

Output:

Nothing for L steps,

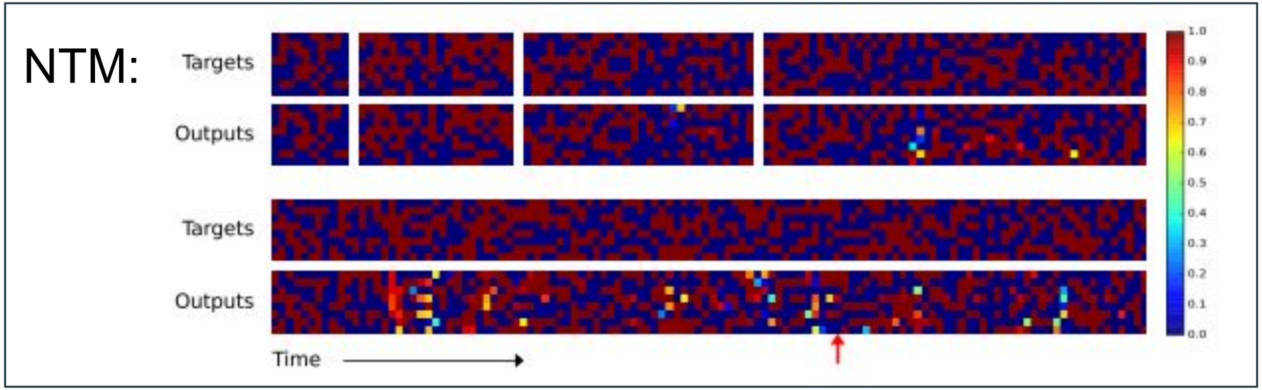
then repeat input sequence.

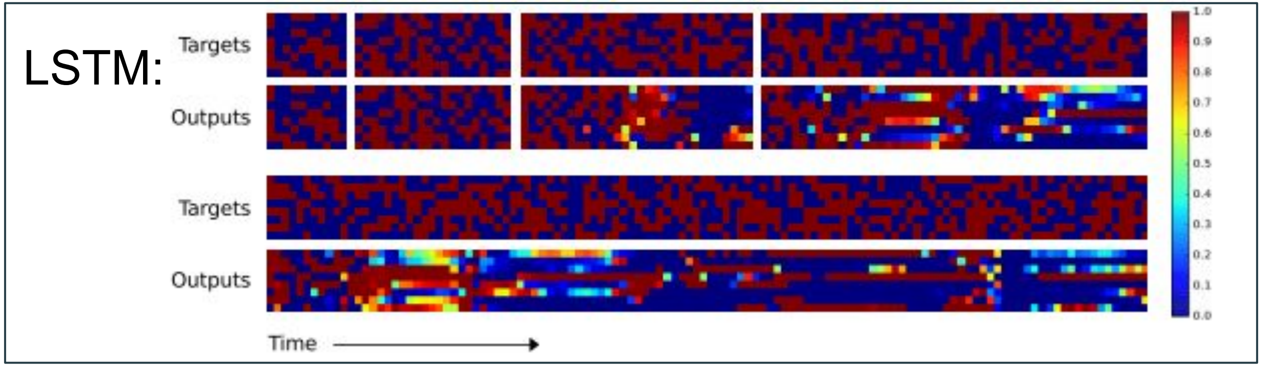


NTM vs LSTM: Copy problem generalization

Trained on copy problem for sequences of length L

results for sequences of length L, L*2, L*4, etc.

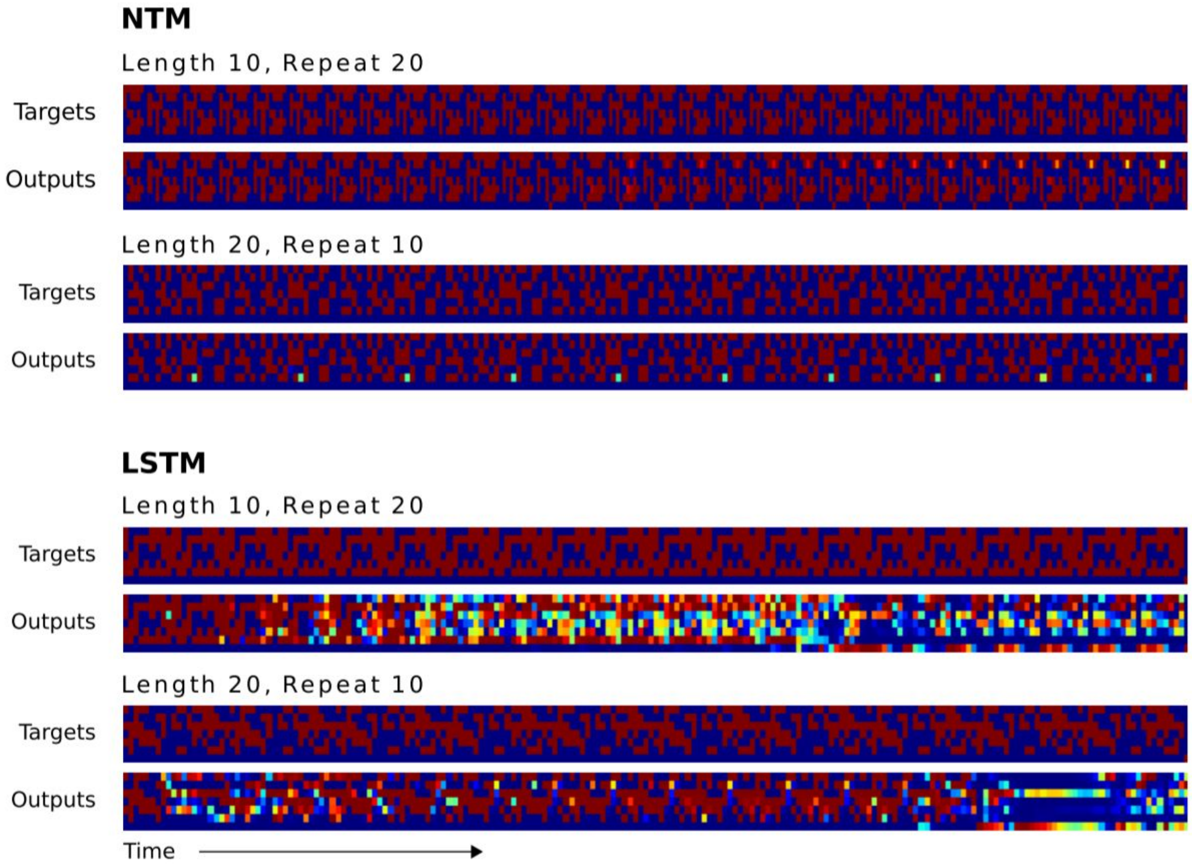




NTM vs LSTM: repeat copy problem

Input:
Sequence of length L,
number of repeats (X),
then nothing for L * X steps.

Output:
Nothing for L+1 steps,
then repeat input sequence
X times.



Exciting applications: one-shot learning

- Neural networks are powerful, but require a ton of data
- limits real world applications, small datasets are insufficient
- training requires large amounts of computing power and time
  (expensive and slow)
- Can’t learn in real time very well, takes too long and too much data

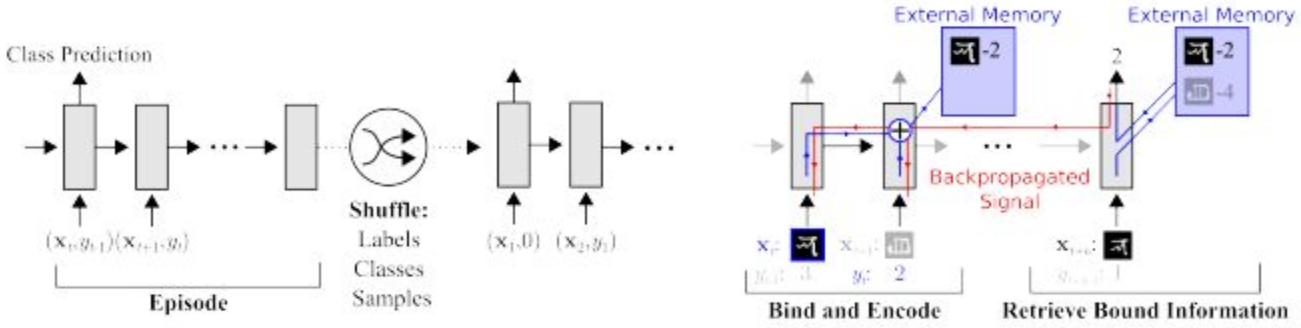
Solution: don’t just learn; learn how to learn first.

One-shot Learning with Memory-Augmented Neural Networks

Input: Image, class of previous image

Output: class of current image

classes used, labels for each class, and specific samples are all shuffled between episodes



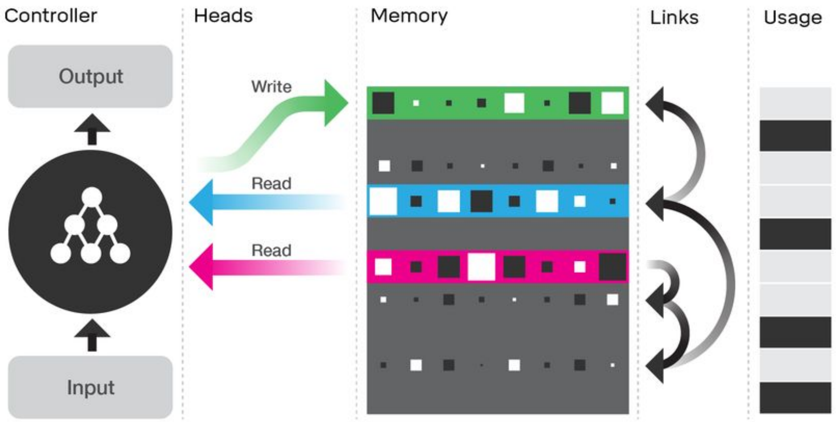
(a) Task setup (b) Network strategy

Model	Instance (% Correct) 1st	2nd	3rd	4th	5th	10th
Human	34.5	57.3	70.1	71.8	81.4	92.4
Feedforward	24.4	19.6	21.1	19.9	22.8	19.5
LSTM	24.4	49.5	55.3	61.0	63.6	62.5
MANN	**36.4**	**82.8**	**91.0**	**92.6**	**94.9**	**98.1**

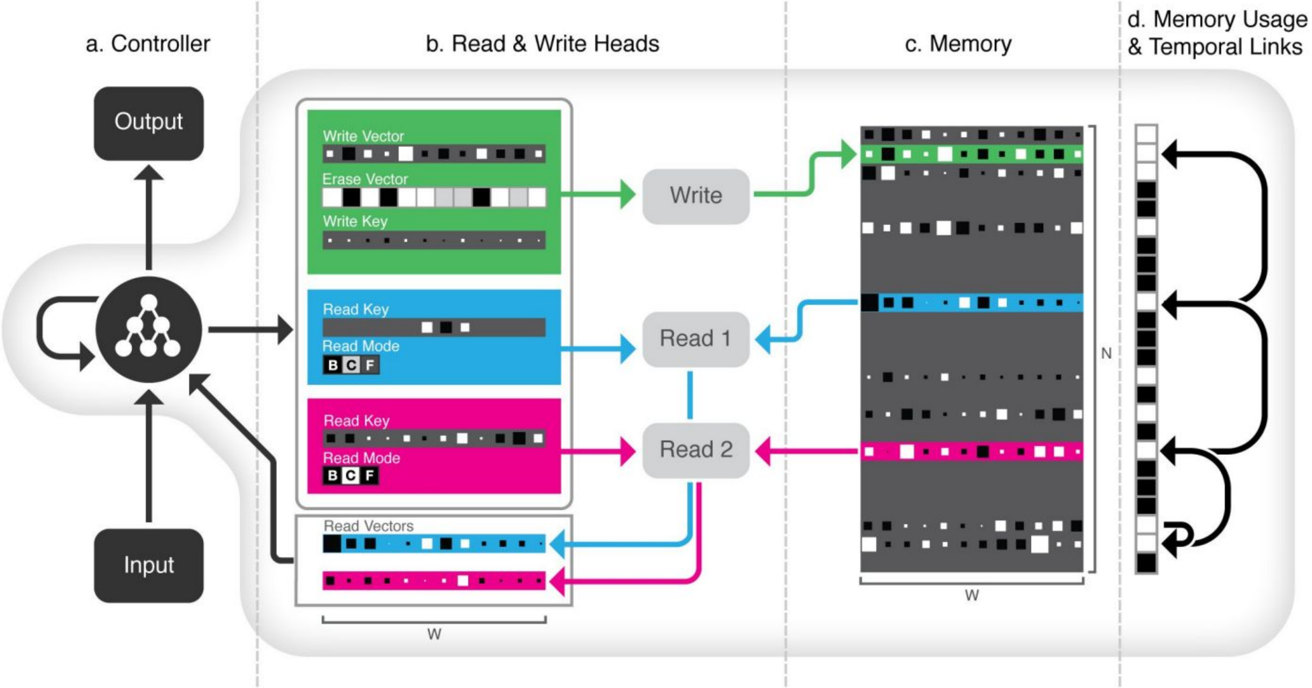
NTM++: Differentiable Neural Computer (DNC)

- Improves upon NTM by using discrete memory interaction and explicit temporal connections
- discrete memory updates reduce decay, enable longer memory retention
- maintains end-to-end differentiability

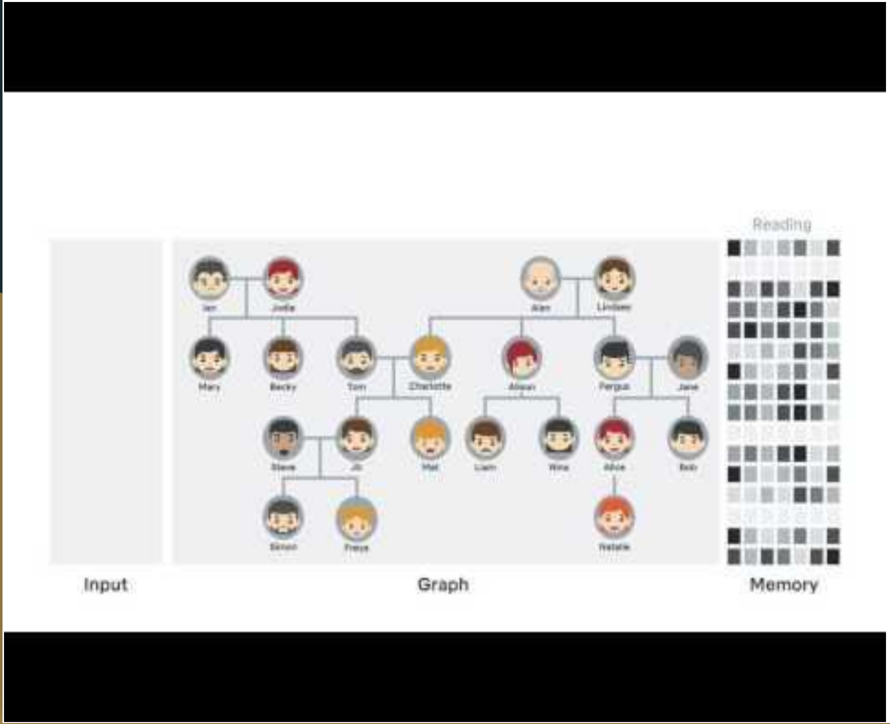
Illustration of the DNC architecture

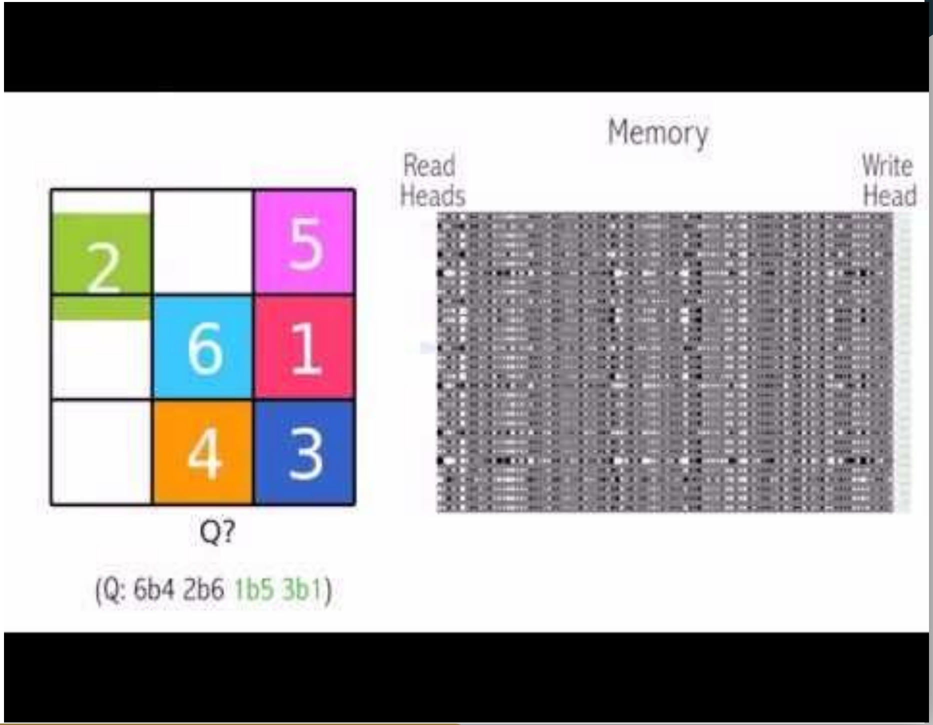


DNC: more detail



DNC: task demonstrations





bAbI

Task	bAbI Best Results						
	LSTM (Joint)	NTM (Joint)	DNC1 (Joint)	DNC2 (Joint)	MemN2N (Joint) [21]	MemN2N (Single) [21]	DMN (Single) [20]
1: 1 supporting fact	24.5	31.5	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
2: 2 supporting facts	53.2	54.5	1.3	0.4	1.0	**0.3**	1.8
3: 3 supporting facts	48.3	43.9	2.4	**1.8**	6.8	2.1	4.8
4: 2 argument rels.	0.4	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
5: 3 argument rels.	3.5	0.8	**0.5**	0.8	6.1	0.8	0.7
6: yes/no questions	11.5	17.1	**0.0**	**0.0**	0.1	0.1	**0.0**
7: counting	15.0	17.8	**0.2**	0.6	6.6	2.0	3.1
8: lists/sets	16.5	13.8	**0.1**	0.3	2.7	0.9	3.5
9: simple negation	10.5	16.4	**0.0**	0.2	**0.0**	0.3	**0.0**
10: indefinite knowl.	22.9	16.6	0.2	0.2	0.5	**0.0**	**0.0**
11: basic coreference	6.1	15.2	**0.0**	**0.0**	**0.0**	0.1	0.1
12: conjunction	3.8	8.9	0.1	**0.0**	0.1	**0.0**	**0.0**
13: compound coref.	0.5	7.4	**0.0**	0.1	**0.0**	**0.0**	0.2
14: time reasoning	55.3	24.2	0.3	0.4	**0.0**	0.1	**0.0**
15: basic deduction	44.7	47.0	**0.0**	**0.0**	0.2	**0.0**	**0.0**
16: basic induction	52.6	53.6	52.4	55.1	**0.2**	51.8	0.6
17: positional reas.	39.2	25.5	24.1	**12.0**	41.8	18.6	40.4
18: size reasoning	4.8	2.2	4.0	**0.8**	8.0	5.3	4.7
19: path finding	89.5	4.3	**0.1**	3.9	75.7	2.3	65.5
20: agent motiv.	1.3	1.5	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Mean Err. (%)	25.2	20.1	4.3	**3.8**	7.5	4.2	6.4
Failed (err. > 5%)	15	16	**2**	**2**	6	3	**2**

Issues with MANNs

- Memory size, addressing mechanism, and numbers of read and write heads are additional hyperparameters -- makes training more difficult
- Scales poorly with size of memory
- Additional complexity of implementation and experimental design

Source Papers

Neural Turing Machines (Dec 2014)

https://arxiv.org/abs/1410.5401

One-shot Learning with Memory-Augmented Neural Networks (May 2016)

https://arxiv.org/abs/1605.06065

Hybrid computing using a neural network with dynamic external memory (Oct 2016)

https://www.nature.com/articles/nature20101