What if Neural Networks had explicit memory?

Batuhan Tömekçe

24.05.2021

Seminar in Deep Neural Networks

Current NNs fail at long sequences

- MLP can’t generalize to arbitrary sequence length
- RNNs are conceptually powerful but have problems in training.
    - Gradient vanishing and explosion
- LSTMs solve gradient problems but hard to train and still don’t scale well with long sequences

What about learning algorithms?

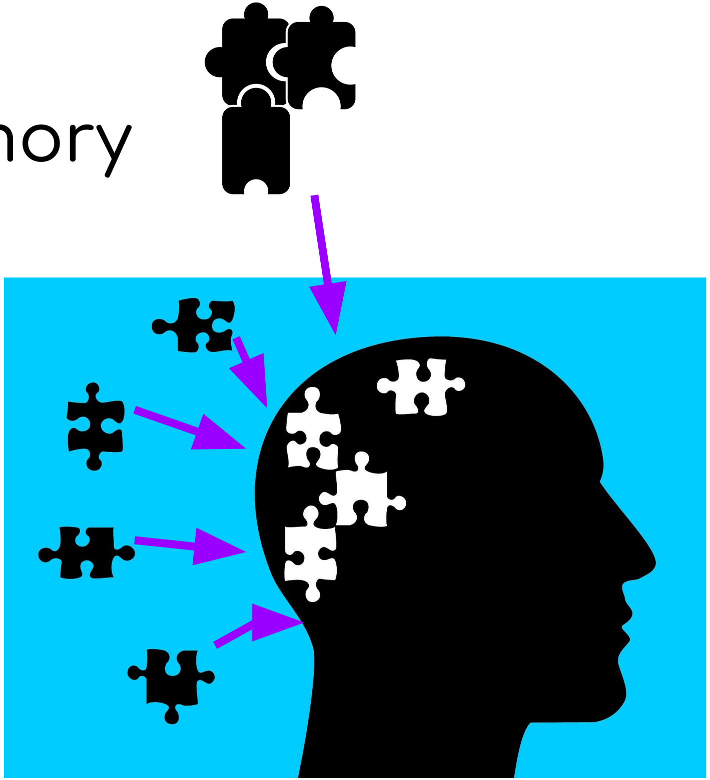
- Most of the use-cases of neural networks are regression or classification
- No functionality for rule based systems (branches, loops)
- The learnt functions aren’t interpretable
- Memory component is only implicit in weights or a single state vector

Outline

1. Inspirations from brains and computers
2. Where do neural networks come from?
3. Memory Networks - extending neural networks with its own knowledge base
4. Memory-Augmented Neural Network - how to build one?
5. Building a neural computer through MANNs
6. Discussion

Brain uses both short-term and long-term memory

- Humans use their short term memory with chunks
- Global workspace theory - retrieve necessary knowledge per decision

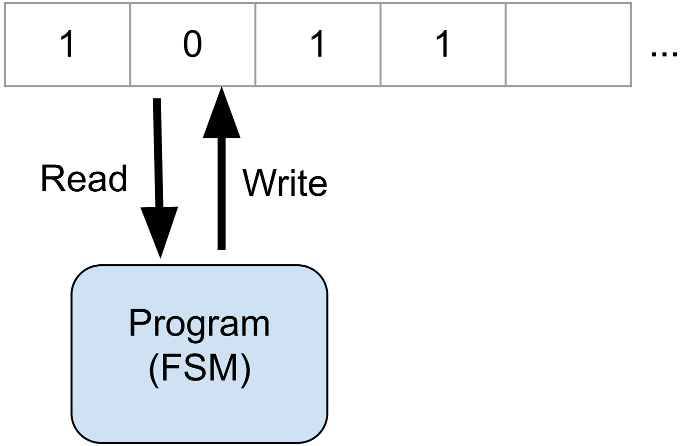


Source: www.pixabay.com

Miller, G. A. (1956) & Bernhard J. Baars, (1997)

FSM with memory makes a Turing Machine

A Turing Machine can be represented as a 4-tuple of states, symbols, program and initial state



$$(Q, \Gamma, \delta, q_0)$$

$$\delta : Q \times \Gamma \rightarrow \Gamma \times \{-1, 1\} \times Q$$

Computers can implement algorithms

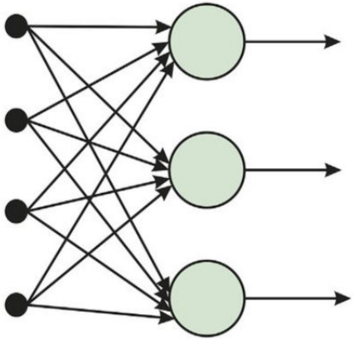
- Road to computers
  - Finite state machine to Turing Machine goes through Memory
- RNNs are turing complete
  - What does it mean?
- Universal Turing Machine
- Contrasting Turing Machines with statistical approach
  - MLPs are static function approximators

Siegelman and Sontag, (1995)

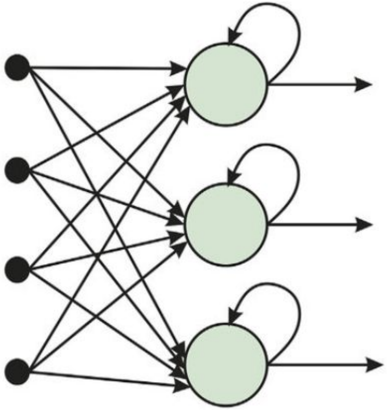
Everything is connected and distributed

- Connectionism
  - a cognitive theory
- Parallel distributed processing
  - implementation of the theory

Feed-forward



Recurrent



Using Memory to store input knowledge

Memory Networks equip neural networks with custom knowledge base

4 key ingredients

- I $\longrightarrow$ input feature map
  - e.g. word to vector
- G $\longrightarrow$ generalization
  - update memory
- O $\longrightarrow$ output feature map
  - produce output
- R $\longrightarrow$ response
  - e.g. output to word

Weston et al., (2014)

Memory Networks equip neural networks with custom knowledge base

- Main goal is to reason about multiple dependent connections
  - e.g. question answering, reasoning about connections between sentences

- Stores incoming knowledge

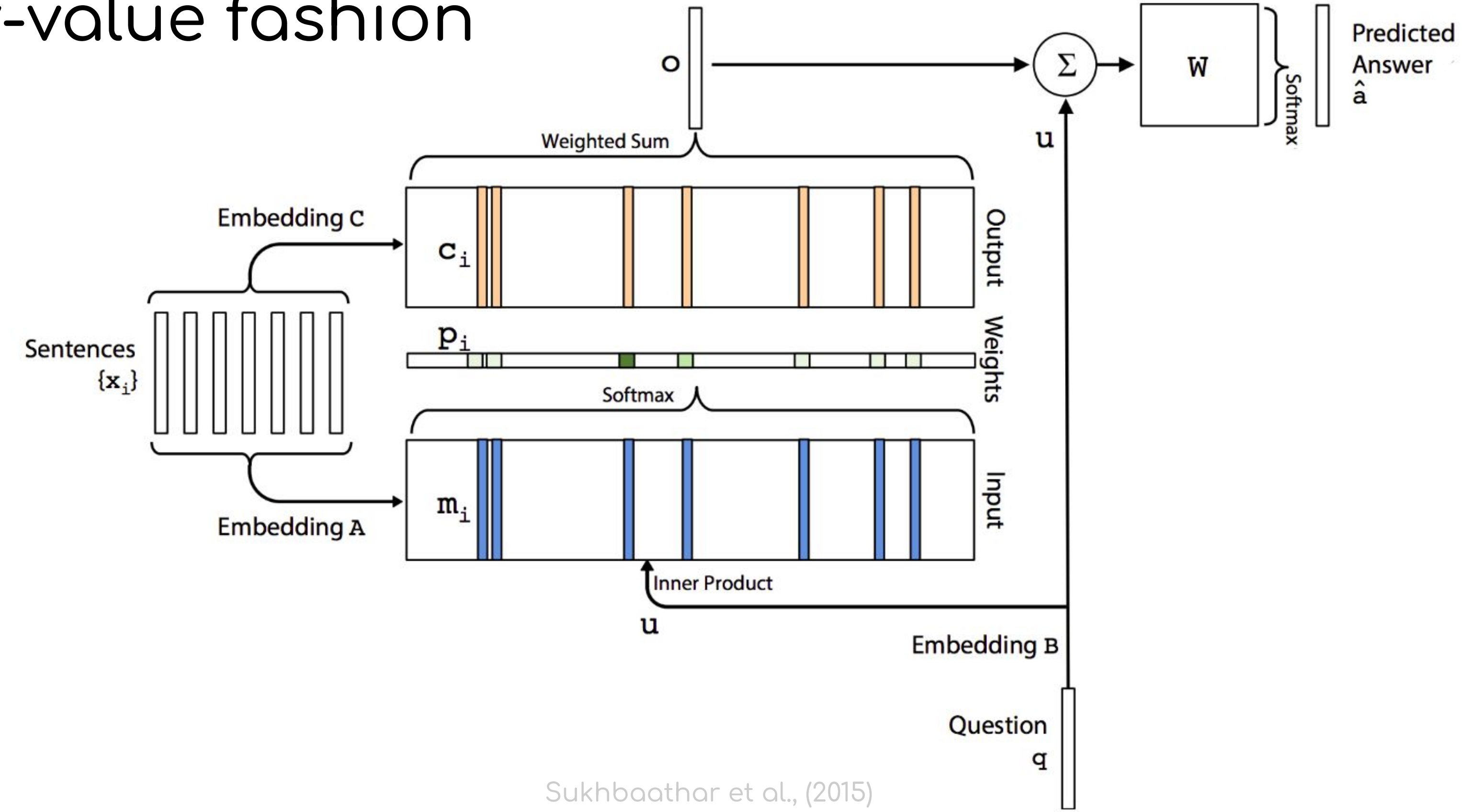
- Retrieves sequentially

Weston et al., (2014)

Neurons fire together wire together

- End-to-end training improves memory networks
- Used in language modeling and questions answering
- Multiple hops is important for reasoning tasks

The knowledge is encoded into memory in key-value fashion



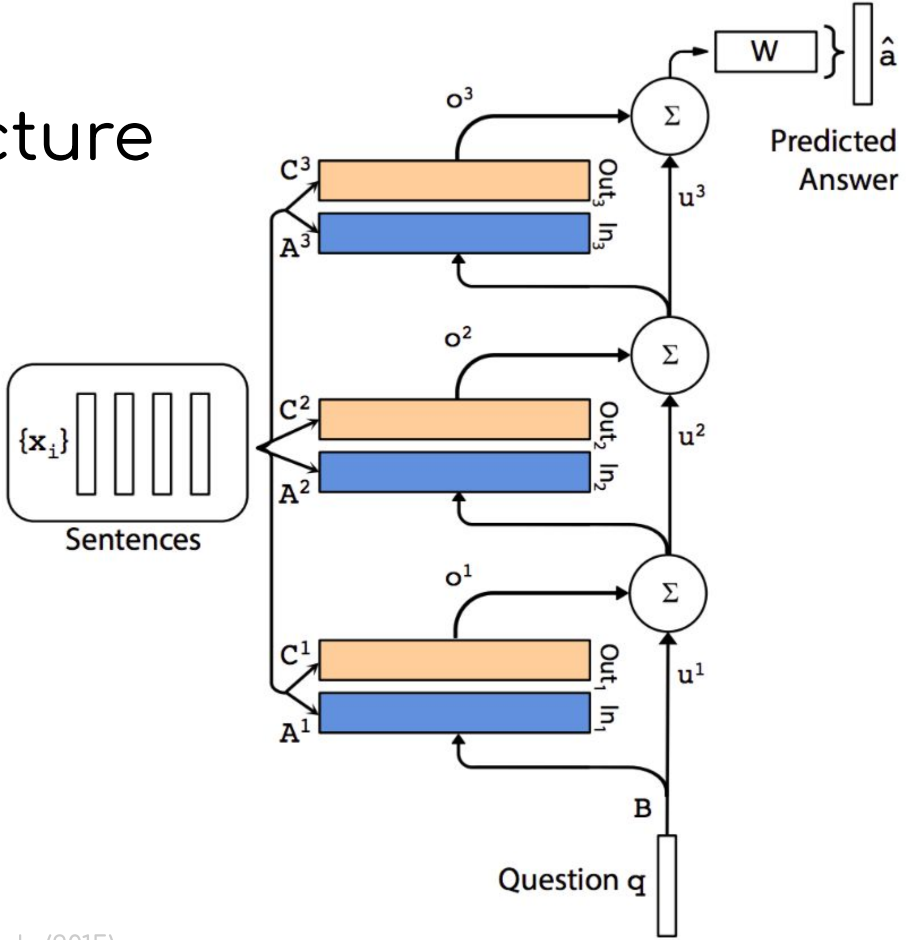
Sukhbaathar et al., (2015)

Multiple hops enable reasoning in a chain structure

Sentences are e.g. Wikipedia

Who is presenting neural network architectures for algorithms in Deep Neural Networks seminar at ETH?

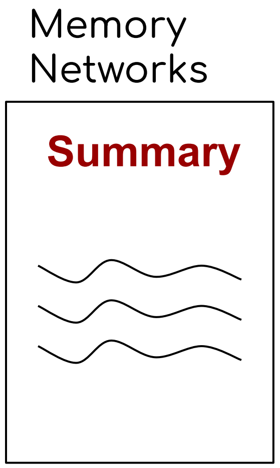
The question will be answered in hops



Sukhbaatar et al., (2015)

Memory Networks have static memory

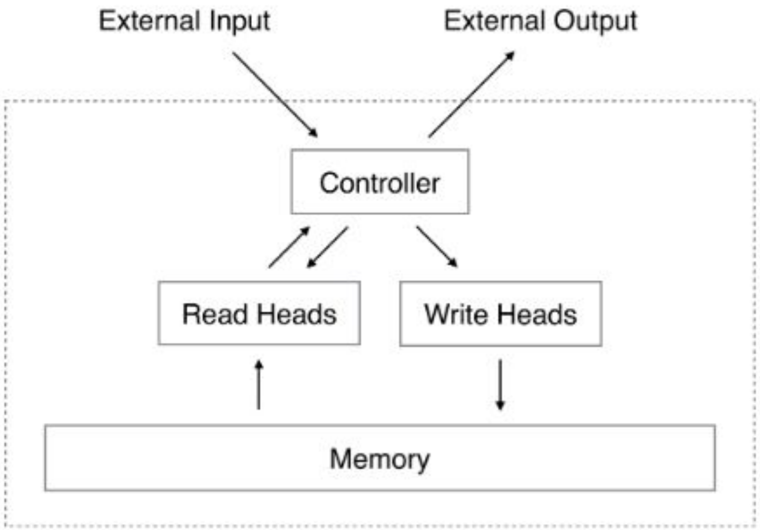
- Mostly used in real datasets (language, vision)
- Mostly fixes the memory at test time
- Mostly big memory size and works on discrete sets, multiple hops (multiple memory)



Enhancing a neural network with memory to solve algorithms

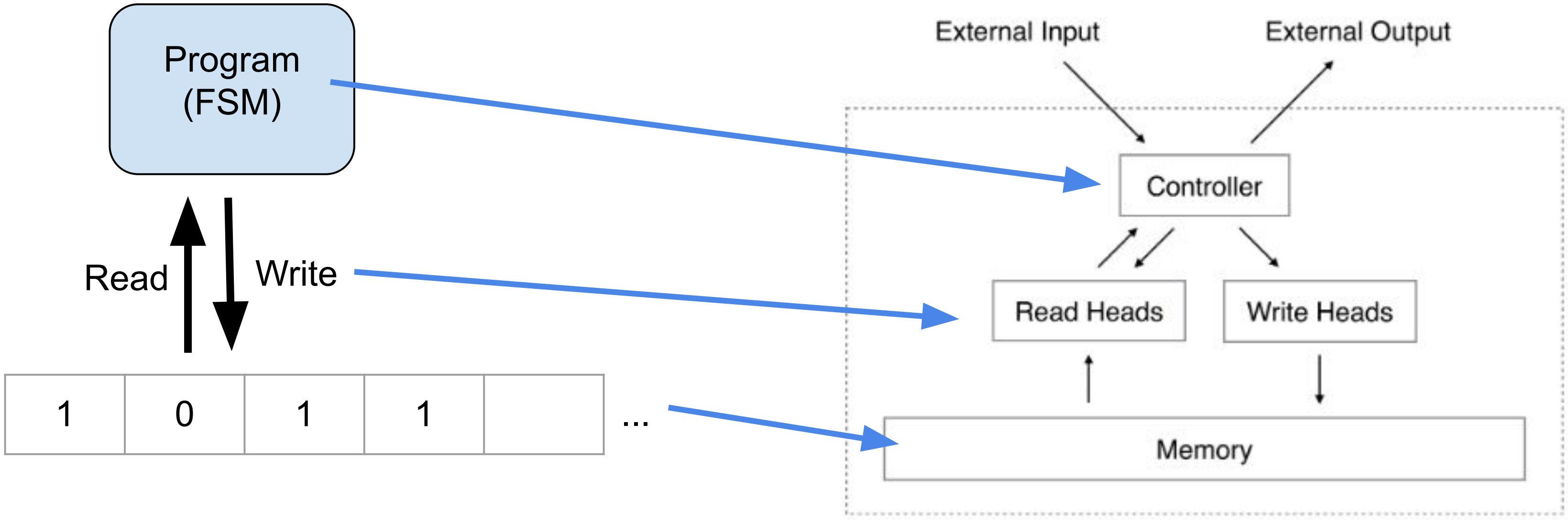
Neural Turing Machine augments a neural network with memory

- Controller consists of neural network(s)
- First read then write
- Main method of communication is attention



Graves et al., (2014)

Why it is called Neural Turing Machine?



Graves et al., (2014)

Read by convex combination of the memory cells

One read head at each time step computes

$$\boldsymbol{r}_t = \boldsymbol{M}_t^T \omega_t$$

$$\boldsymbol{r} = \omega_1 \boldsymbol{m}_1 + \omega_2 \boldsymbol{m}_2 + \dots + \omega_N \boldsymbol{m}_N$$

Read head generates the

normalized weight vector

$$\sum \omega_i = 1$$

Memory bank

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{m}_1^T \\ \boldsymbol{m}_2^T \\ \dots \\ \boldsymbol{m}_N^T \end{bmatrix}$$

Graves et al., (2014)

Write to memory by erasing and adding

Write head generates erase, add and weight vectors

$\boldsymbol{e}_t \quad \boldsymbol{a}_t \quad \boldsymbol{\omega}_t$

The i'th memory cell is erased by

$$\tilde{\boldsymbol{m}}_{t,i} = \boldsymbol{m}_{t-1,i} - \omega_i \boldsymbol{m}_{t-1,i} \odot \boldsymbol{e}_t$$

The i'th memory is added

$$\boldsymbol{m}_{t,i} = \tilde{\boldsymbol{m}}_{t,i} + \omega_i \boldsymbol{a}_t$$

Memory bank

$$\boldsymbol{M} = \begin{array}{|c|} \hline \boldsymbol{m}_1^T \\ \hline \boldsymbol{m}_2^T \\ \hline \ldots \\ \hline \boldsymbol{m}_N^T \\ \hline \end{array}$$

Graves et al., (2014)

Content-based addressing is used to communicate with memory

A weighted softmax distribution is used for content-based addressing using a key vector

$$\boldsymbol{\omega}_t^c = \frac{\exp \beta_t \mathcal{S}(\boldsymbol{k}_t, \boldsymbol{m}_{t,i})}{\sum_i^N \exp \beta_t \mathcal{S}(\boldsymbol{k}_t, \boldsymbol{m}_{t,i})}$$

Graves et al., (2014)

$$\boldsymbol{\omega}_t^c = \frac{\exp \beta_t \mathcal{S}(\boldsymbol{k}_t, \boldsymbol{m}_{t,i})}{\sum_i^N \exp \beta_t \mathcal{S}(\boldsymbol{k}_t, \boldsymbol{m}_{t,i})}$$

Location-based addressing is used for variable binding

First a convex combination of previous and content-based weight is taken

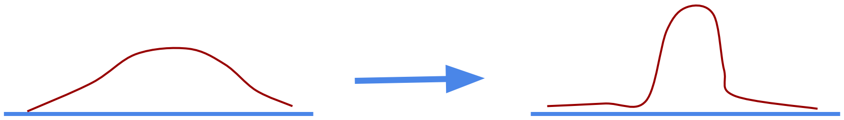
$$\boldsymbol{\omega}_t^g \longleftarrow (1 - g_t)\boldsymbol{\omega}_t^c + g_t \boldsymbol{\omega}_{\boldsymbol{t-1}} \qquad g_t \in (0, 1)$$

Then a circular convolution is taken

$$\boldsymbol{\omega}_t^g \quad * \quad \boldsymbol{s}_t \quad = \quad \tilde{\boldsymbol{\omega}}_t$$

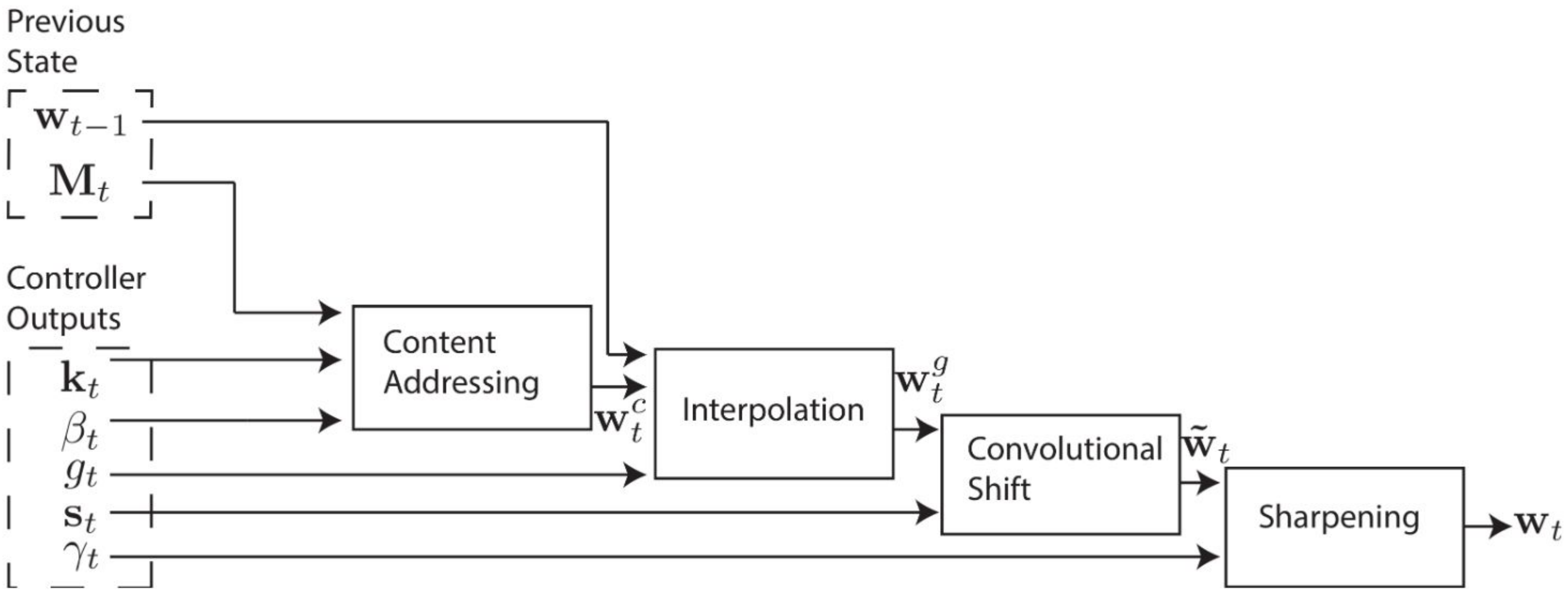
Then sharpen the distribution against accumulation errors

$$\boldsymbol{\omega}_t = softmax(\gamma_t \tilde{\boldsymbol{\omega}}_t)$$



Graves et al., (2014)

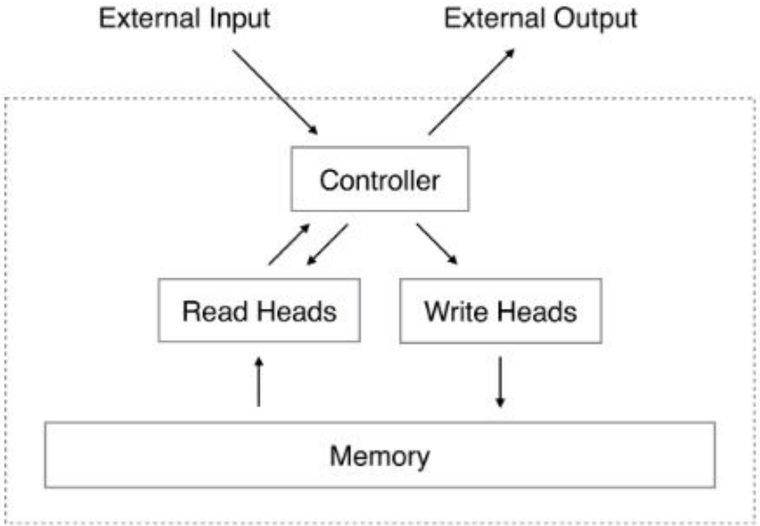
Content based and location based addressing are employed together



Graves et al., (2014)

Controller consists of interface and state networks

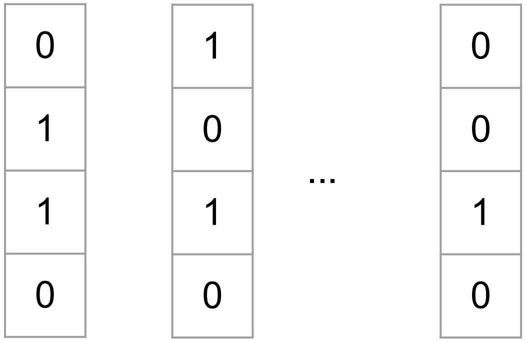
- State network is LSTM
- Interface network is MLP



Graves et al., (2014)

Evaluation is done through algorithmic rule based tasks that measure generalization

- Copy
- Repeat Copy



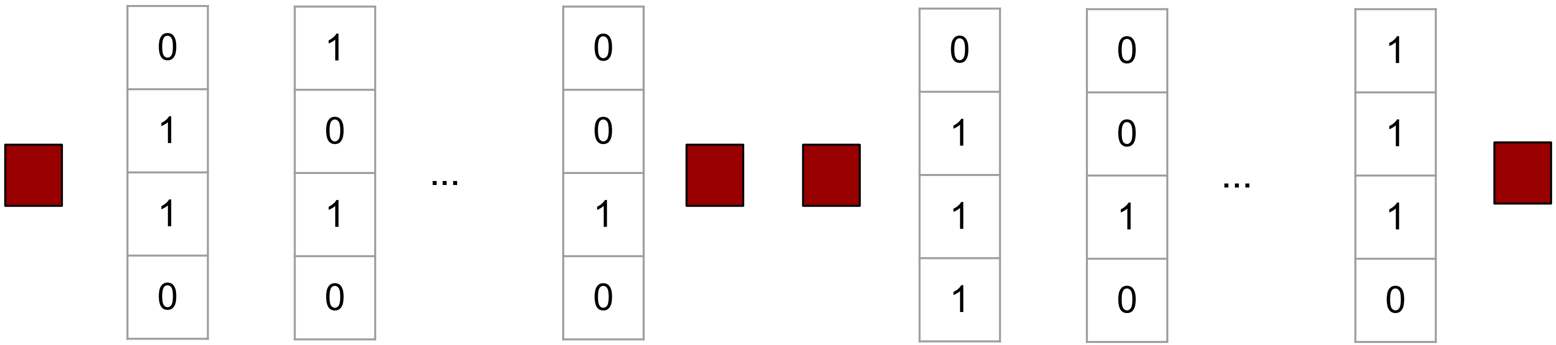
Sequence of binary vectors

Graves et al., (2014)

Evaluation is done through algorithmic rule based tasks that measure generalization

Associative Recall

- Given list of items and a query item from the list, the model predicts the item next to the query



Graves et al., (2014)

Evaluation is done through algorithmic rule based tasks that measure generalization

Dynamic N-gram

- Learn a distribution using the memory

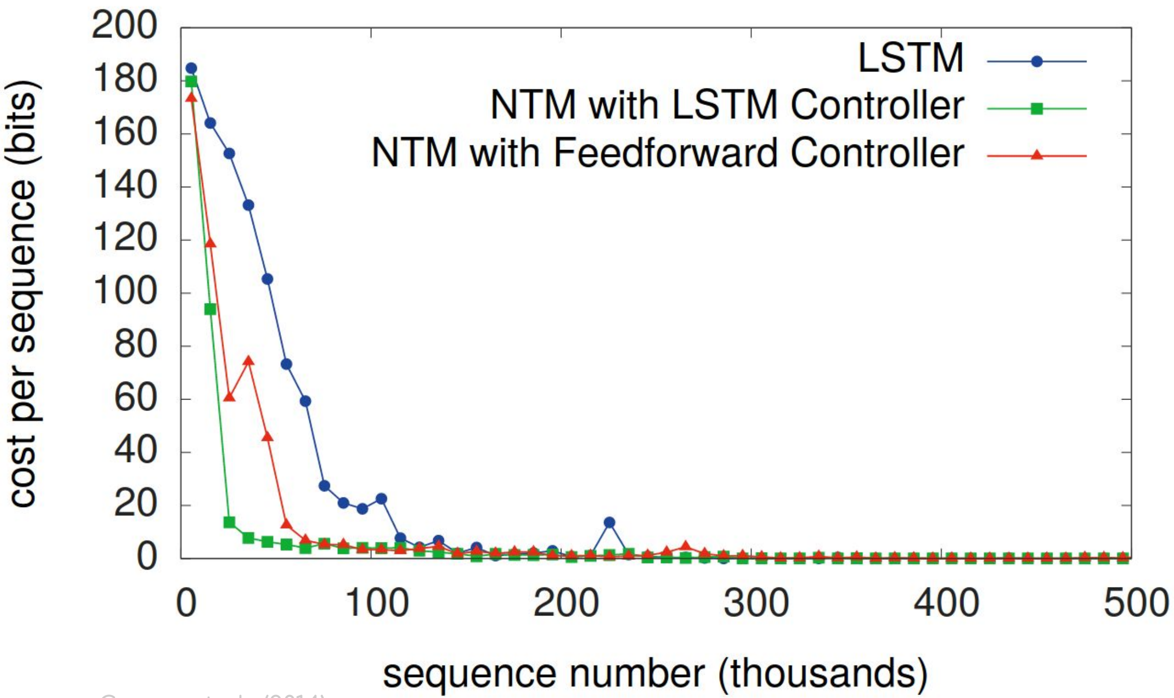
Priority Sorting

- Given vectors with preferences, sort them according to their preference

Graves et al., (2014)

NTM has more representational power and performs better

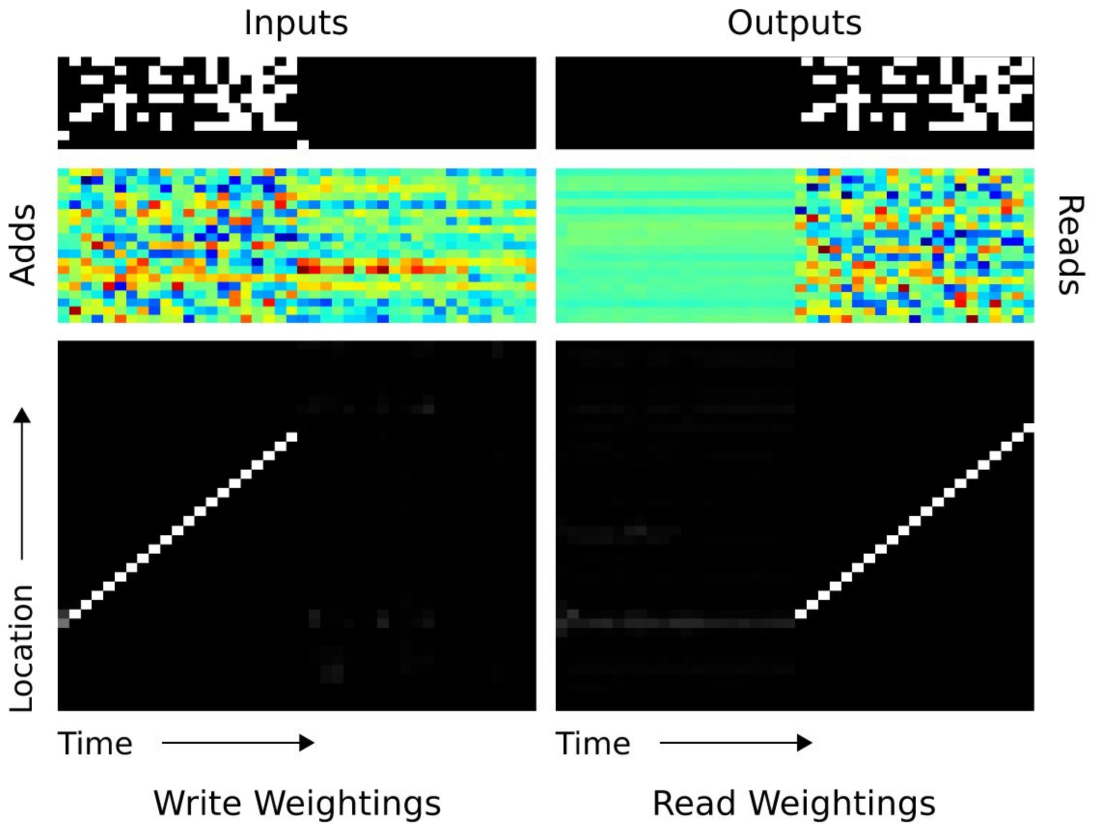
- First the input sequence is fed
- Then the the models produce output
- Multi-label binary classification
- The weights are resetted after each sequence



Graves et al., (2014)

NTM learns an algorithm

The sequence is stored in memory and then read from memory



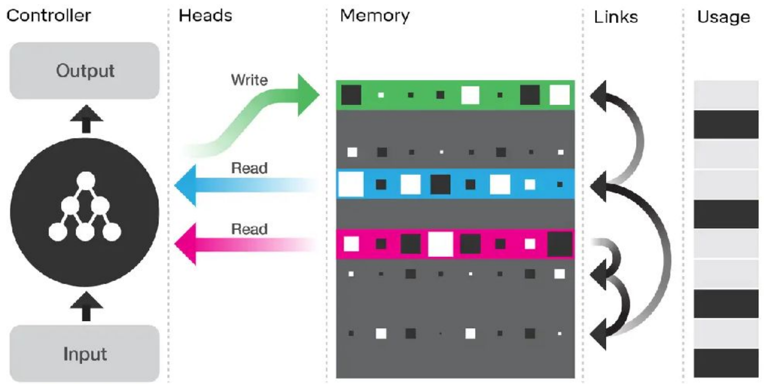
Graves et al., (2014)

Other examples of Memory-Augmented Neural Networks specialize for different task domains

Differentiable Neural Computer also stores the order of memory writes as a linked list

- used for complex data structures

DNC

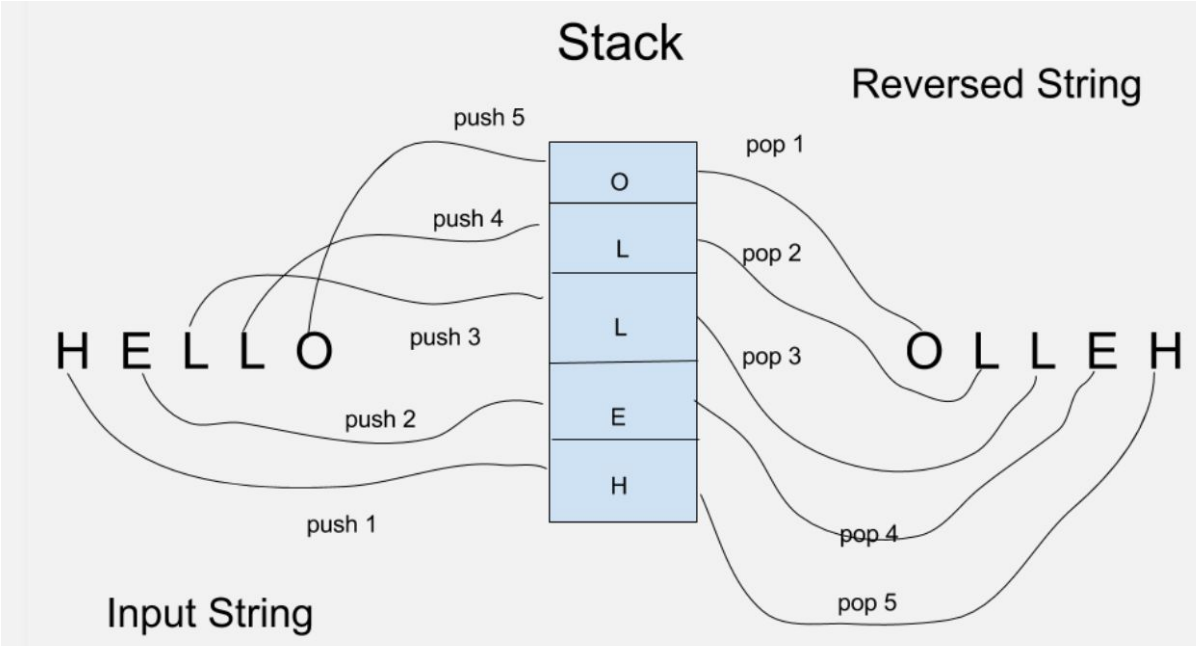


Least Recently Used Access employs a content based addressing as the name suggests

- used in few-shot meta learning

Graves and Wayne et al., (2016) & Santoro et al., (2016)

What about modifications to the memory bank?



Source: https://jasdeep06.github.io/posts/Neural-Stacks/

Grefenstette et al., (2015)

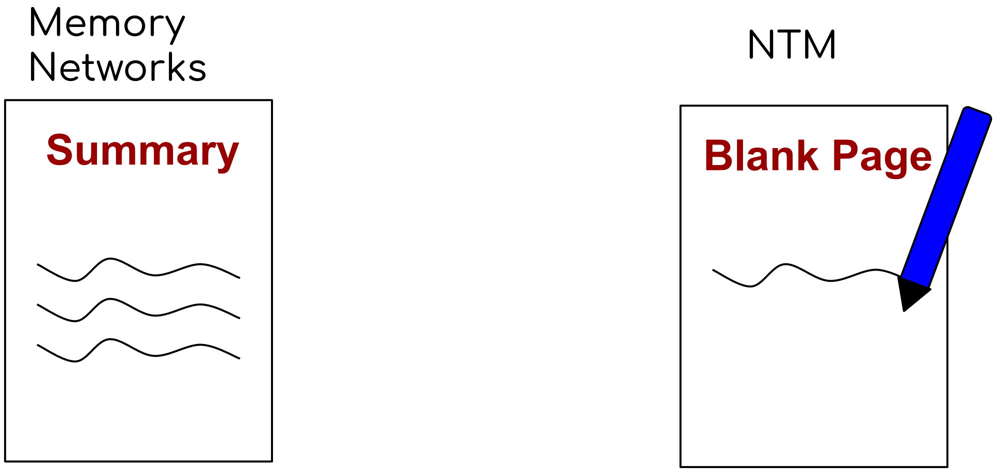
Memory Networks have static memory whereas MANNs have dynamic memory

Memory Networks

- Mostly used in real datasets (language, vision)
- Mostly fixes the memory at test time
- Mostly bigger memory size and works on discrete sets, multiple hops (multiple memory)

Memory-Augmented Neural Networks

- Mostly used in simulated algorithmic tasks
- Use its memory to store objects at test time for algorithmic purposes
- Smaller external memory (different version are possible) provides variable binding


Memory
Networks
Summary
NTM
Blank Page

Using memory to store neural networks

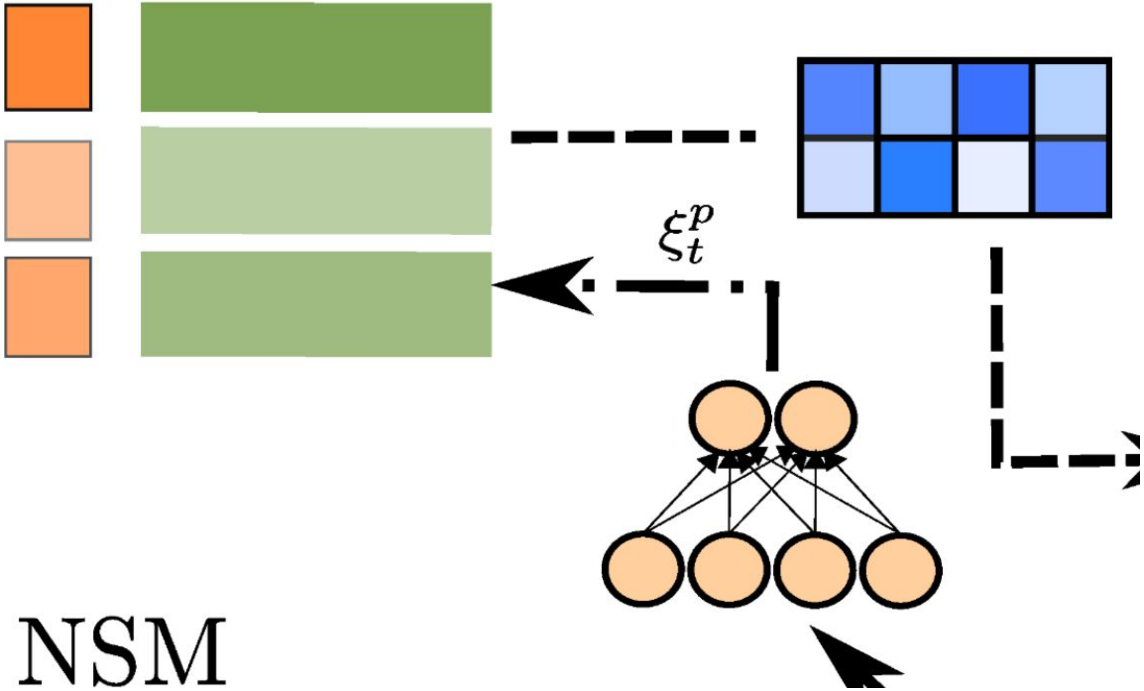
MANNs can be improved with program memory

- Computers store different programs and also the data in RAM
- Selection of different program/models
- Different programs can be used for meta learning and multi-task learning
- Going in the direction of a neural computer

Neural Stored Program Memory uses key-value attention to retrieve weights

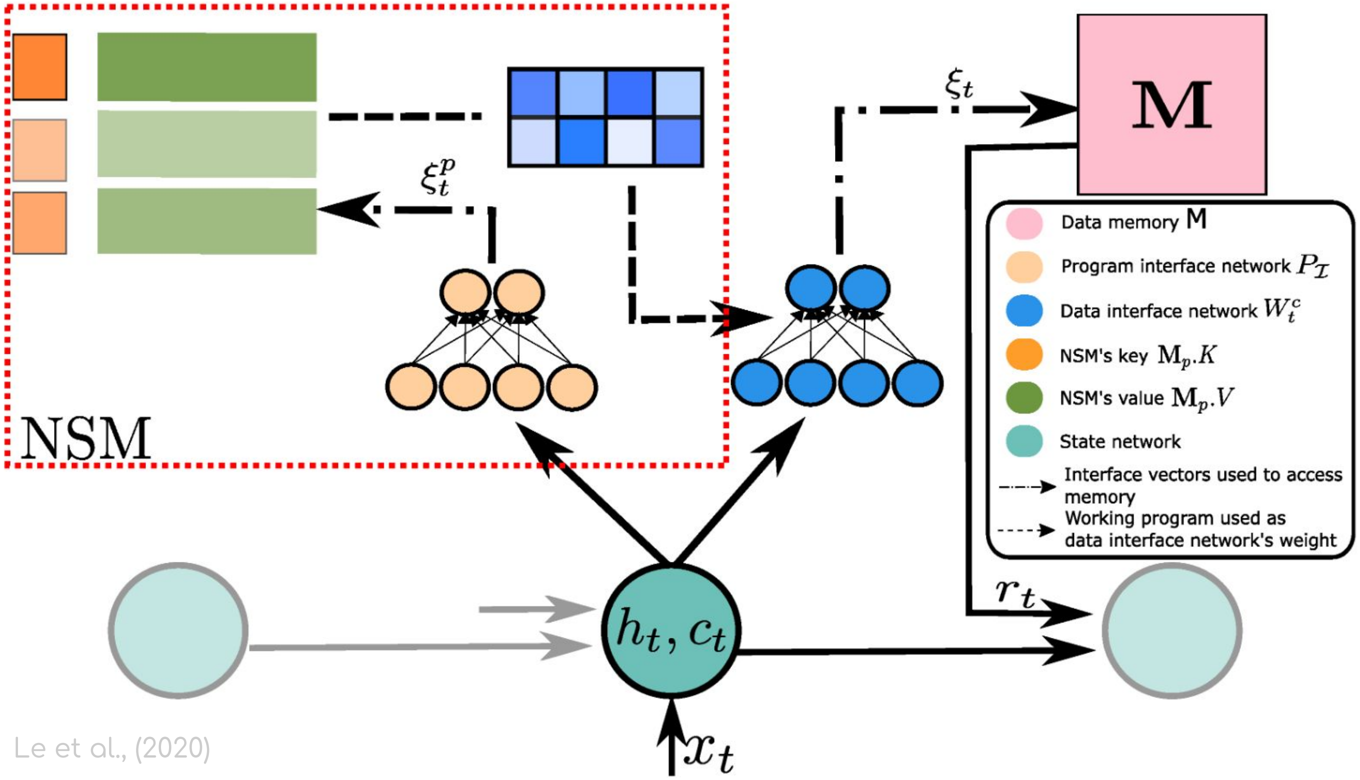
A memory bank to store the weights of the controller

A meta network emits the keys for weight retrieval



Le et al., (2020)

Neural Universal Turing Machine is built by equipping a MANN with NSM



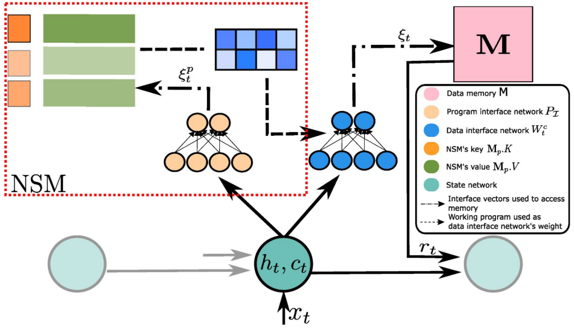
Le et al., (2020)

How does NSM combined with NTM makes a Neural Universal Turing Machine?

- We can build Universal Turing Machines by putting the Turing Machines into the tape

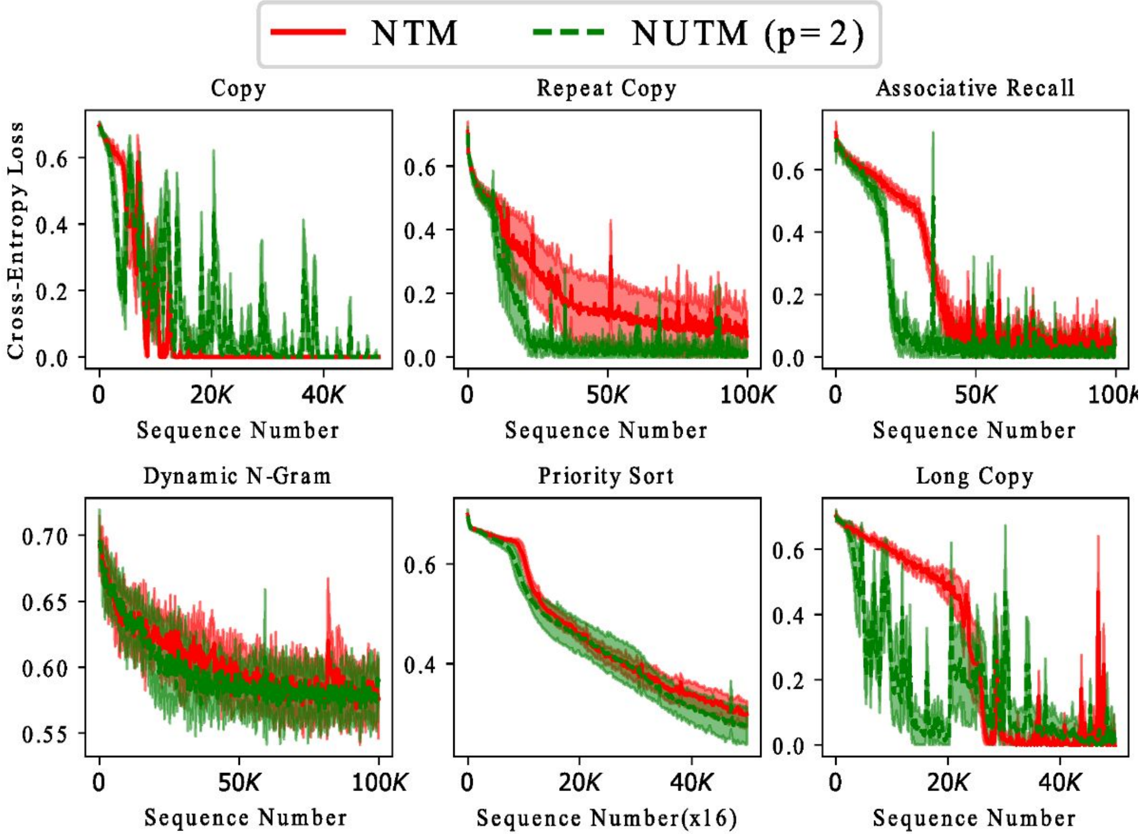
- NSM only stores the weights of MLP interface

- one state Universal Turing Machine



Le et al., (2020)

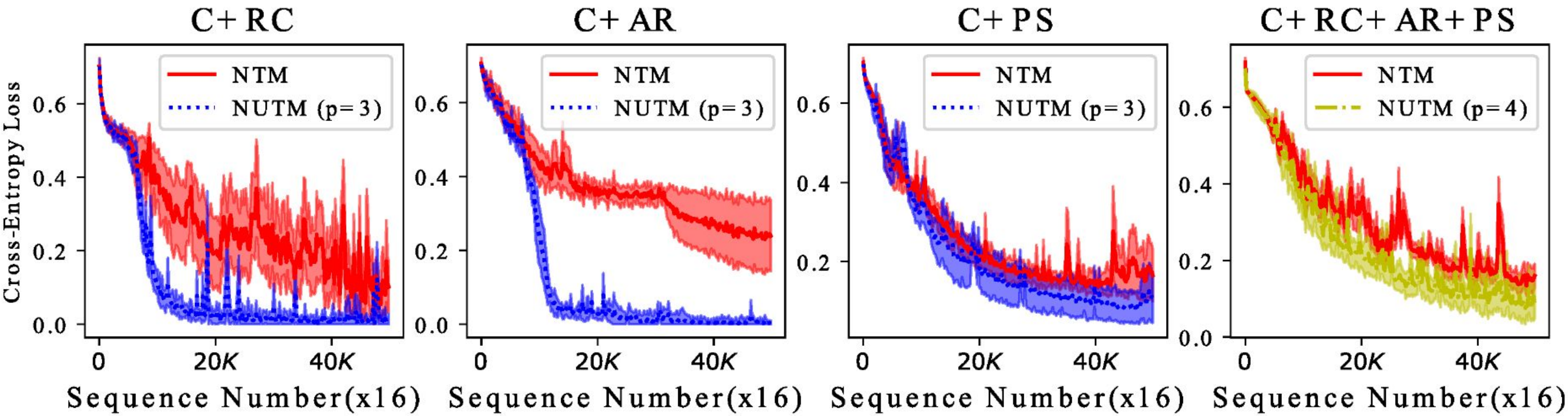
NUTM converges faster than NTM?



Le et al., (2020)

NUTM can learn different programs at once

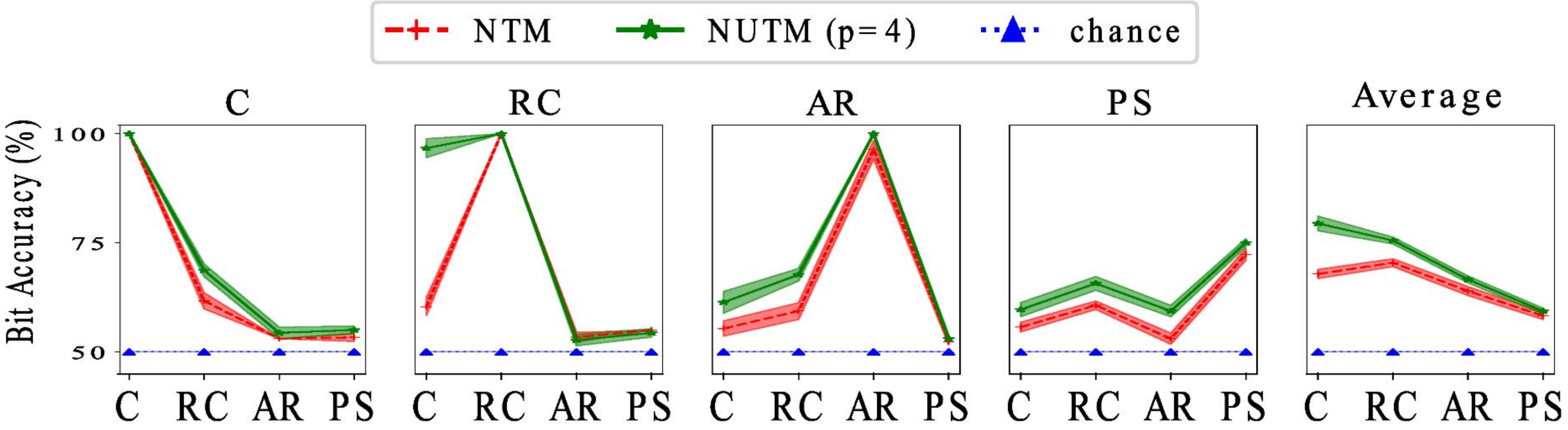
Combination of the atomic tasks after the other



Le et al., (2020)

NUTM forgets less

How much of the task does the model remember?



Le et al., (2020)

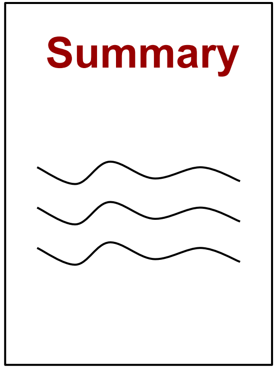
NSM with MANN implements a form of fast and slow weights

- A form of meta-learning
- slow weights are through backpropagation
- fast weights are through interpolation of programs

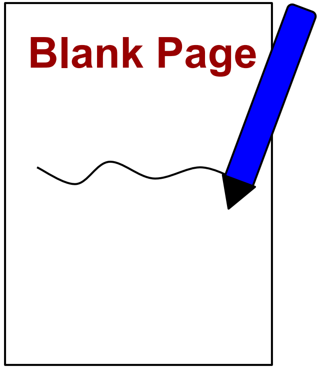
Le et al., (2020) & von der Malsburg, (1981)

Looking into the models as different exam types

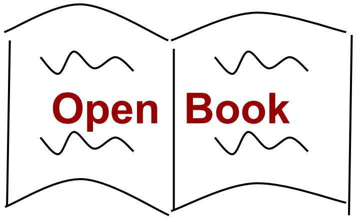
Memory Networks



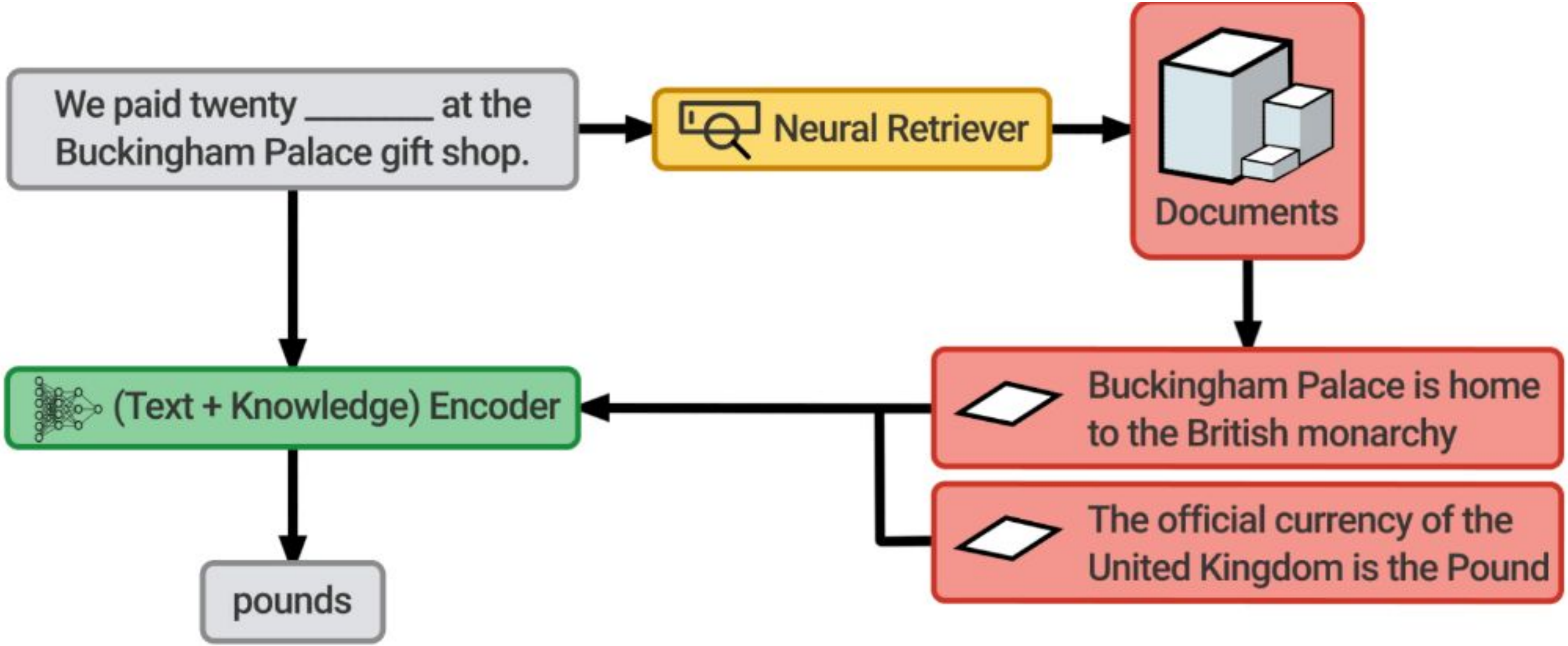
NTM



?



REALM can cite while answering questions



Source: https://ai.googleblog.com/2020/08/realm-integrating-retrieval-into.html

Guu et al., (2020)

Recap

- Importance of memory and differentiability for current intelligence systems
- Memory networks
- Memory-augmented neural networks
- Meta learning perspective, storing neural networks in memory

Discussion

- Memory in RL
- Neural stored-program memory for different application domains
- Learning algorithms with seq2seq + attention
- Trainability and reproducibility of MANNs
- What is the goal?

MLP is all you need

References

- Bernhard J. Baars, “In the theatre of consciousness: Global workspace theory, a rigorous scientific theory of consciousness” (1997)
- Miller, G. A. “The magical number seven, plus or minus two: some limits on our capacity for processing information” (1956)
- Siegelman and Sontag, “On the computational power of neural nets” (1995)
- Weston et al., “Memory networks” (2014)
- Sukhbaatar et al., “End-to-end memory networks” (2015)
- Graves et al., “Neural turing machine” (2014)
- Graves and Wayne et al., “Hybrid computing using a neural network with dynamic external memory” (2016)
- Santoro et al., “Meta-learning with memory-augmented neural networks” (2016)
- Grefenstette et al., “Learning to Transduce with Unbounded Memory” (2015)
- Le et al., “Neural stored-program memory” (2020)
- von der Magsburg, “The correlation theory of brain function” (1981)
- Guu et al., “REALM: Retrieval-Augmented Language Model Pre-Training” (2020)