End-to-End Memory Networks

Sainbayar Sukhbaatar (NYU),
Arthur Szlam, Jason Weston, Rob Fergus

New York University
Facebook AI Research

Overview

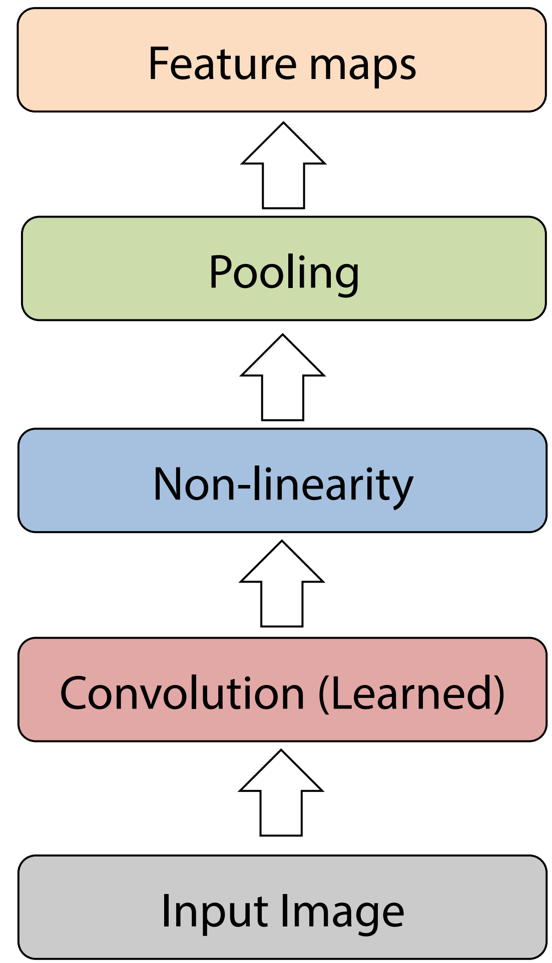
- Impressive performance of Deep Networks for range of perceptual tasks
  - Object recognition, speech, NLP

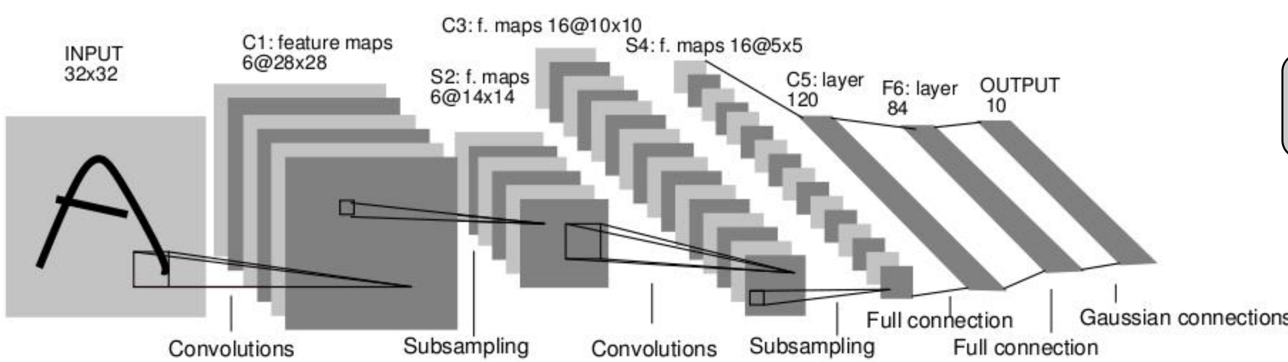
- But models lack explicit memory
  - Essential for some tasks, e.g. reasoning

This talk: Neural net models with explicit memory

Convolutional Network (ConvNet)

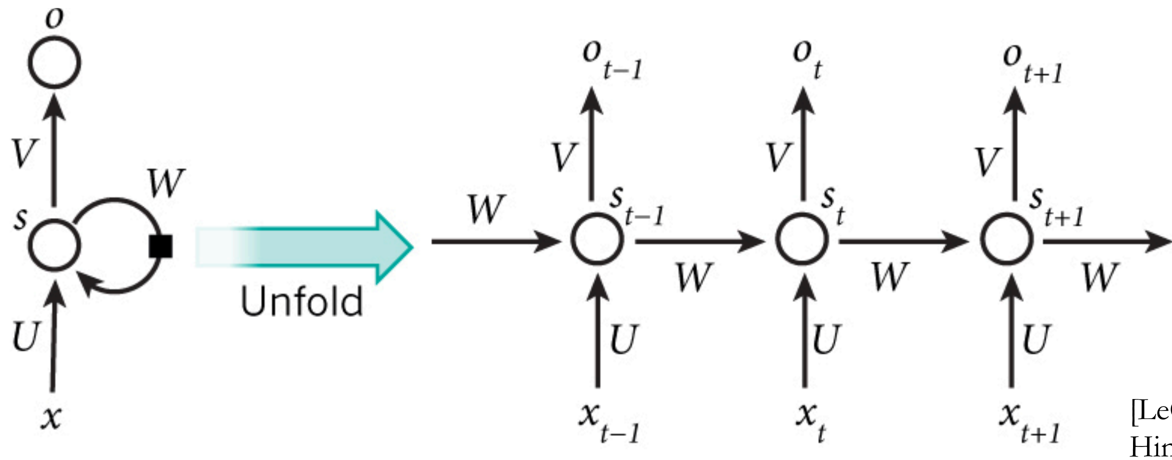
- Feed-forward operation:
  - Convolve input
  - Non-linearity (rectified linear)
  - Pooling (local max)
- Features computed independently per-image
- Only "memory" is in network weights
  - Learnt from training set





LeCun et al. 1998

Recurrent Neural Networks (RNNs)

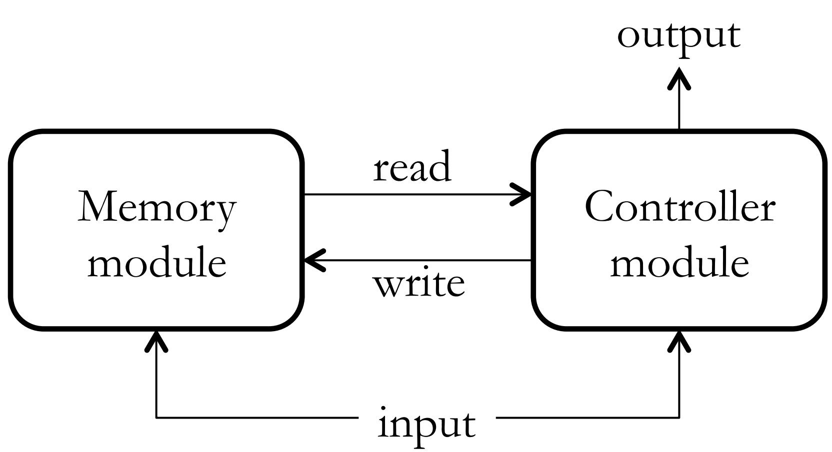


[LeCun, Bengio & Hinton, Nature 2015]

- Implicit memory within internal state s
- Mixing of computation & memory
  - Complex computation requires many layers of non-linearity
  - But some information is lost with each non-linearity
  - Gradient vanishing, catastrophic forgetting problems
  - Workarounds: gate units (e.g. LSTMs); impose slow/fast state

External Global Memory

- Separating memory from computation
  - Dedicated separate memory module
  - Memory can be stack or list/set of vectors



- Control module accesses memory (read, write)
- Advantage: stable, scalable

Memory Networks

Jason Weston, Antoine Bordes
& Sumit Chopra

arXiv: http://arxiv.org/abs/1410.3916 [ICLR 2015]

Memory Networks
(Weston et al., ICLR 2015)

- Neural network with large external memory
- Writes everything to the memory, but reads only relative information
- Hard addressing: max of the inner product between the internal state and memory contents

Example Task

- From bAbI dataset (Weston et al. arXiv 1502.05698, 2015)

Input sentences:

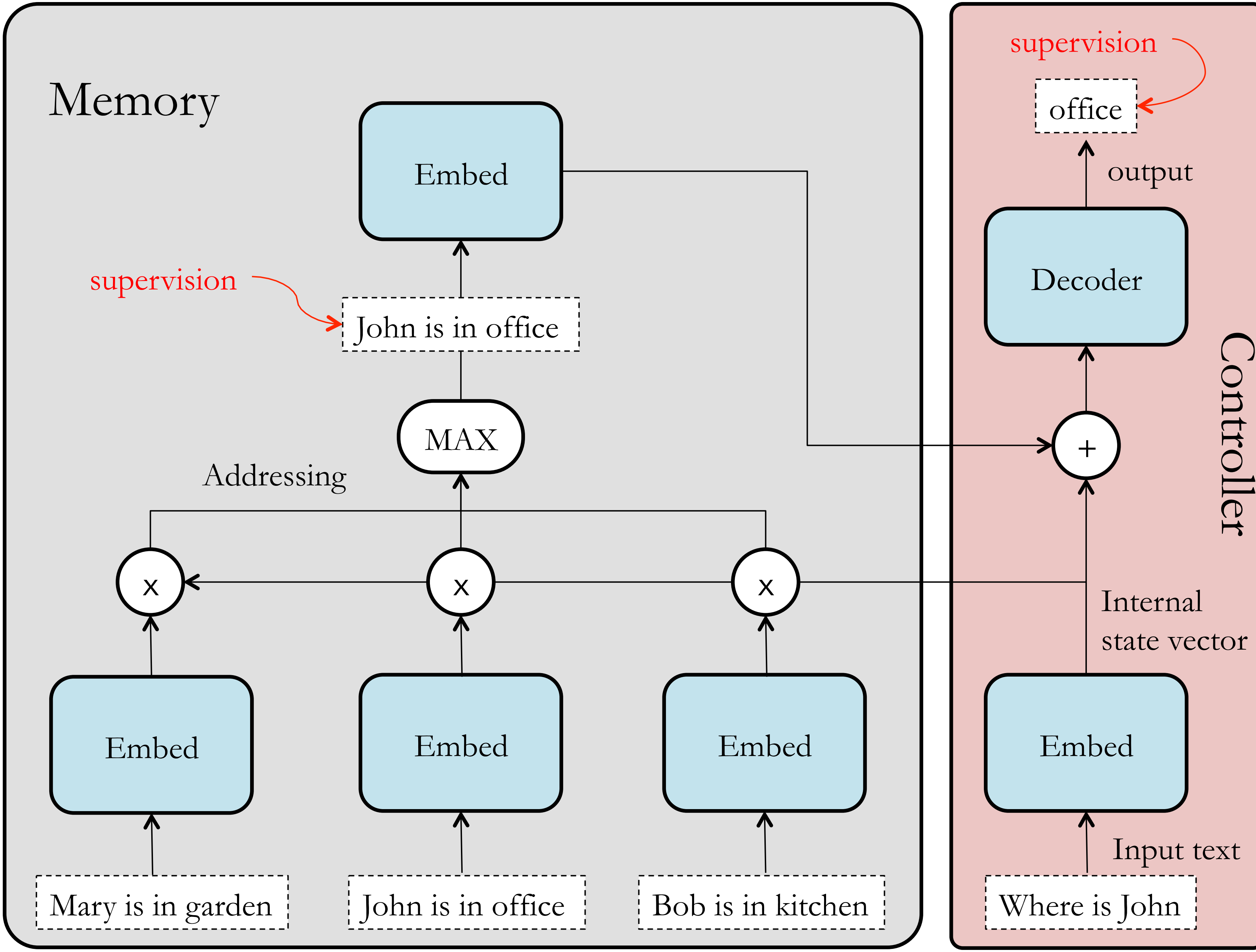
Mary is in garden.

John is in office.

Bob is in kitchen.

Q: Where is John?

A: office


Memory
Embed
supervision
John is in office
MAX
Addressing
x
x
x
Embed
Embed
Embed
Mary is in garden
John is in office
Bob is in kitchen
supervision
office
output
Decoder
Controller
+
Internal
state vector
Embed
Input text
Where is John

Issues with Memory Network

- Requires explicit supervision of attention during training
  - Need to say which memory the model should use
- Only feasible for simple tasks
  - Severely limits application of model
- Want model that just requires supervision at output
  - No supervision of attention required

End-to-End Memory Networks (MemN2N)

Sainbayar Sukhbaatar, Arthur Szlam,
Jason Weston, Rob Fergus

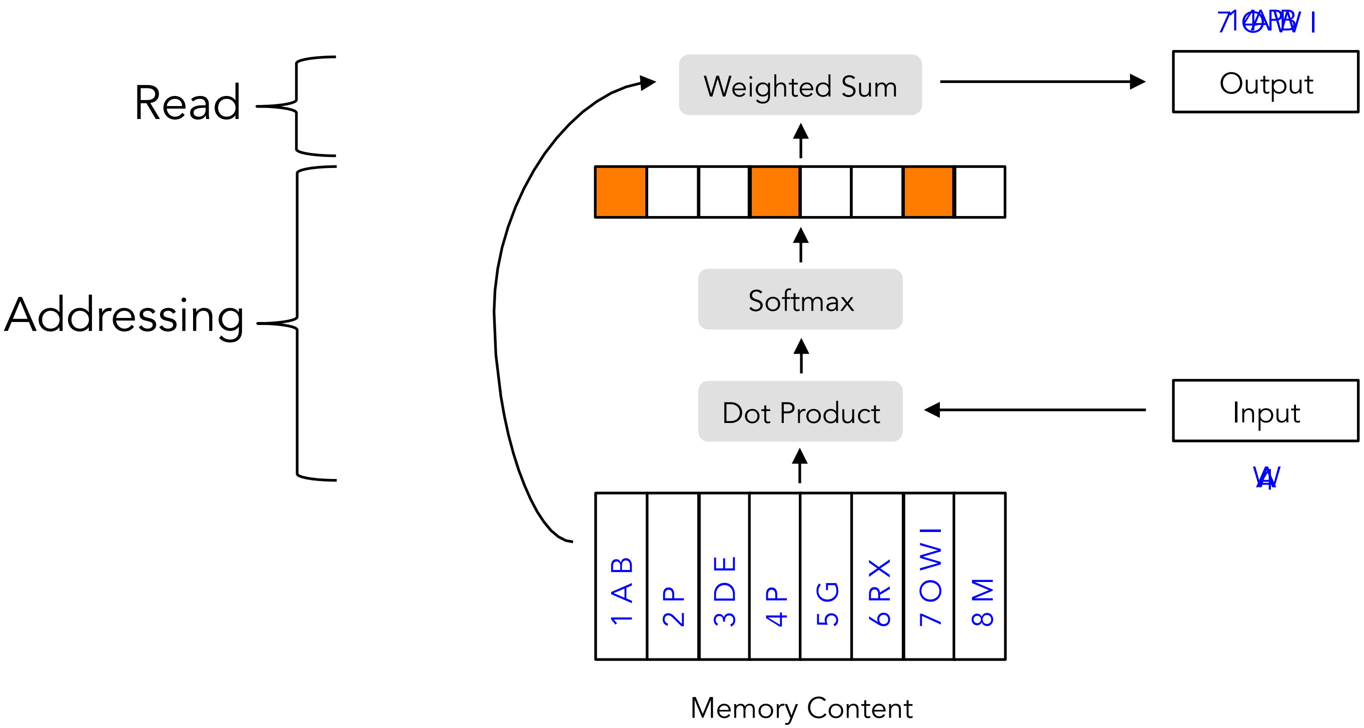
arXiv: http://arxiv.org/abs/1503.08895

[NIPS 2015]

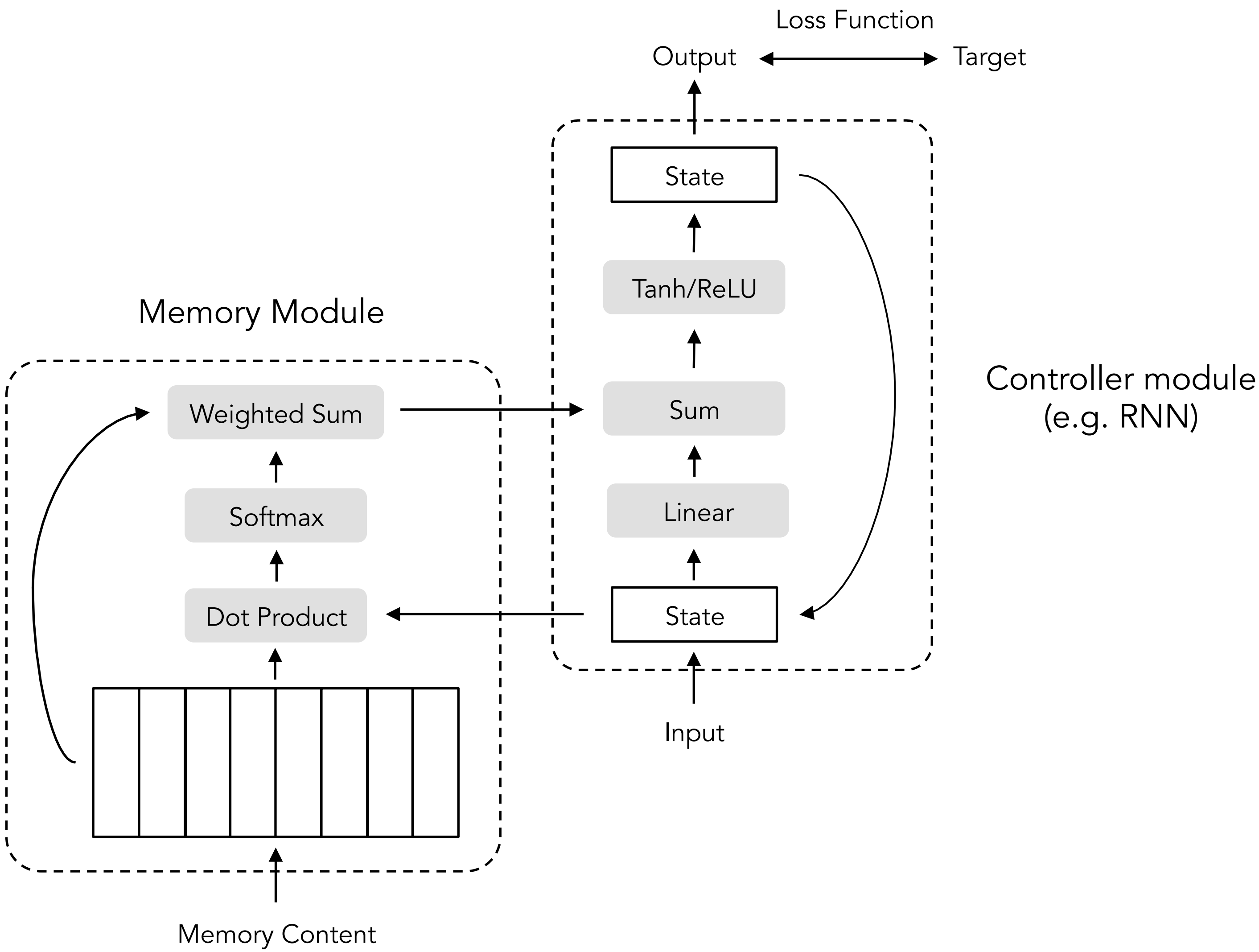
End-To-End Memory Networks (MemN2N)

- Soft attention version of MemNN
  - Flexible read-only memory
- End-to-end training
  - Only needs final output for training
  - Simple back-propagation
- Multiple memory lookups (hops)
  - Can consider multiple memory before deciding output
  - More reasoning power

Memory Module

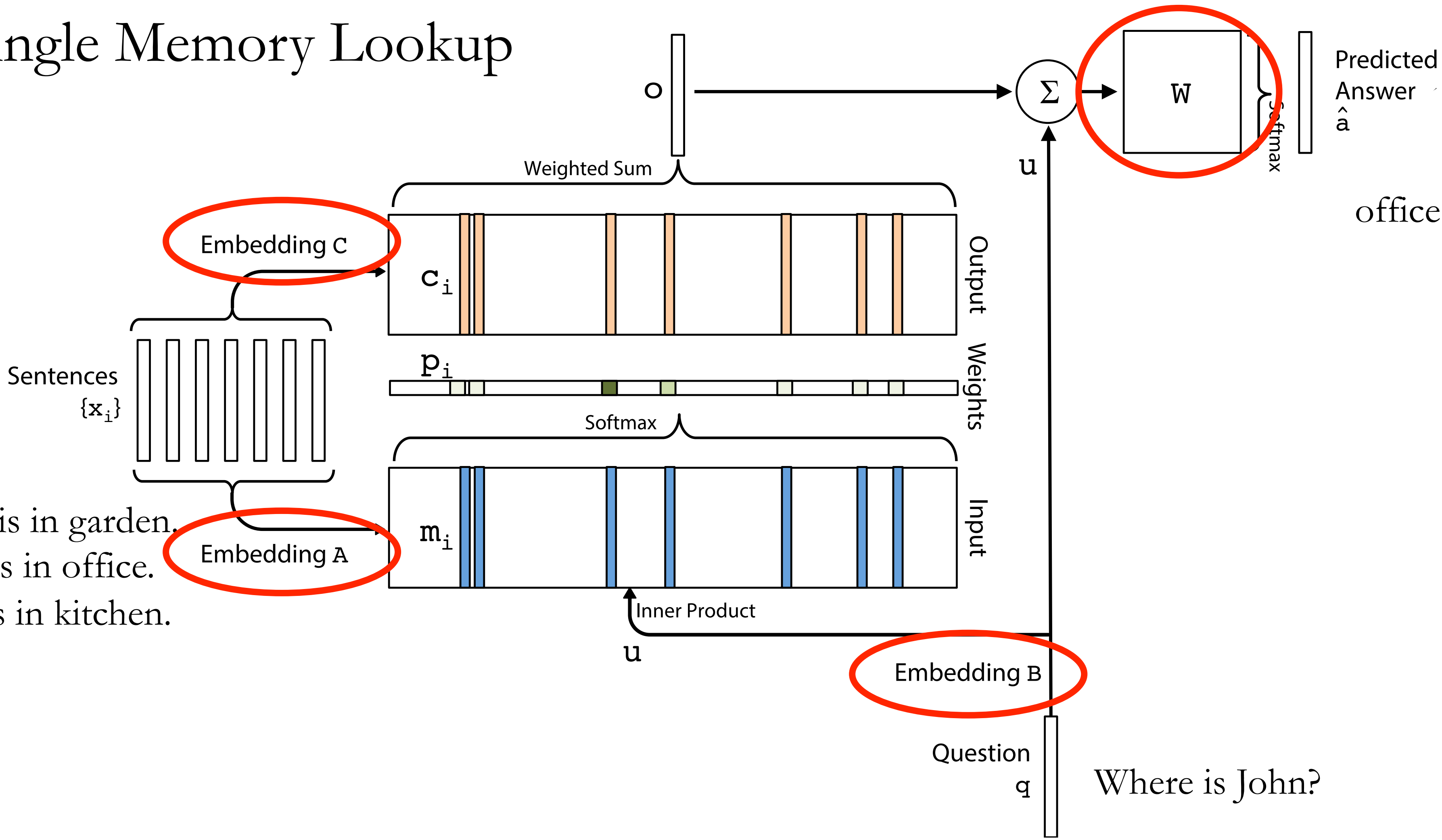


MemN2N architecture



MemN2N applied to bAbI task

Single Memory Lookup



office

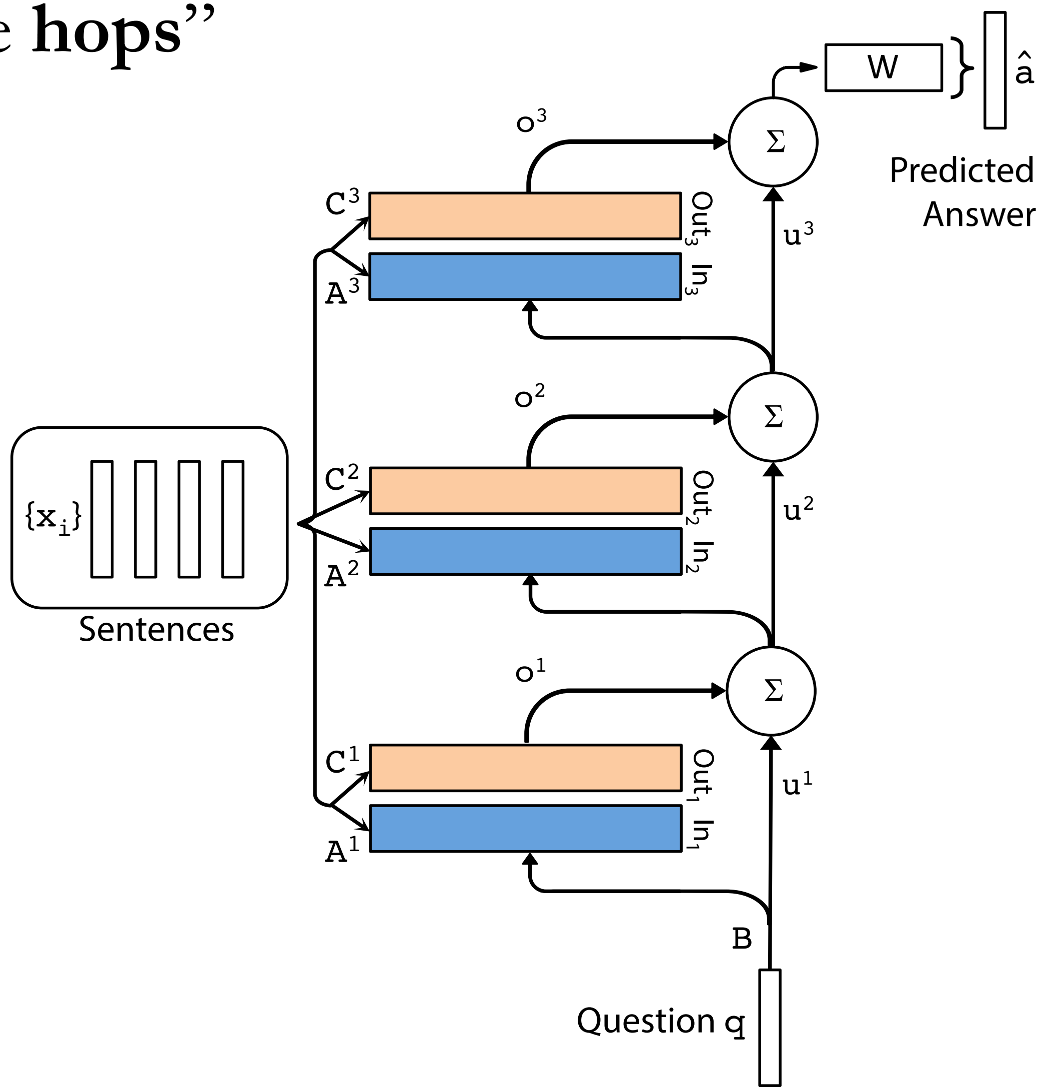
Mary is in garden.
John is in office.
Bob is in kitchen.
…
…

Where is John?

Training: estimate embedding matrices A, B & C and output matrix W

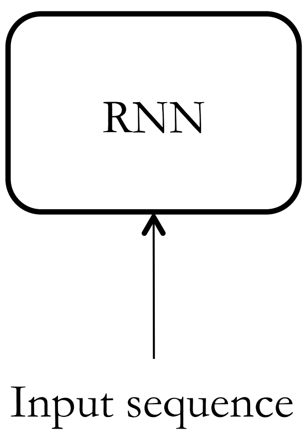
Multiple Memory Lookups

"Multiple **hops**"



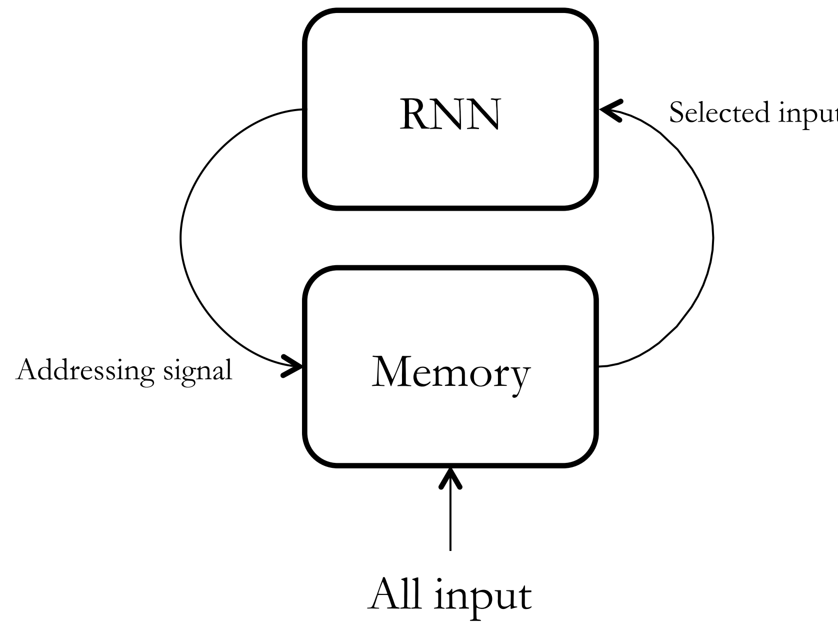
RNN viewpoint of MemN2N

Plain RNN



Inputs are fed to RNN one-by-one in order. RNN has only one chance to look at a certain input symbol.

Memory Network



Place all inputs in the memory. Let the model decide which part it reads next.

Advantages of MemN2N over RNN

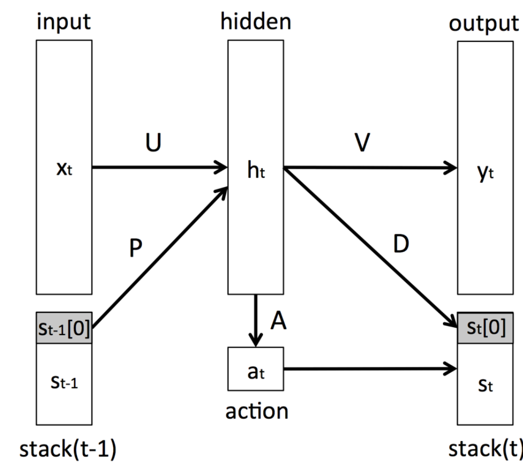
- More generic input format
  - Any **set** of vectors can be input
  - Each vector can be
    - BOW of symbols (including location)
    - Image feature + feature position
  - Location can be 1D, 2D, …
  - Variable size
- Out-of-order access to input data
- Less distracted by unimportant inputs
- Longer term memorization
- No vanishing or exploding gradient problems

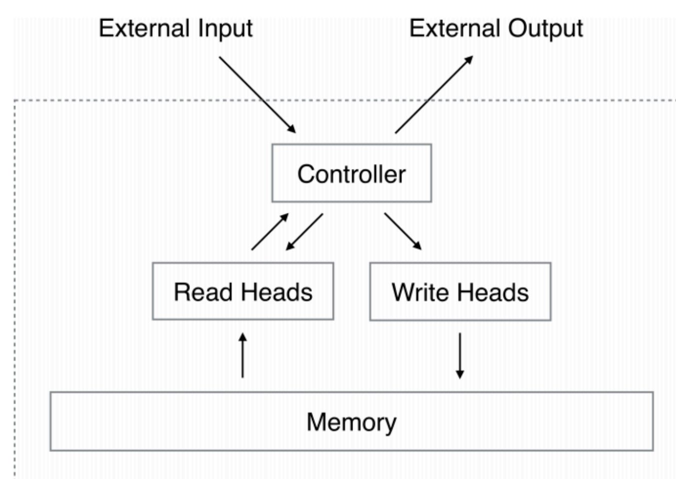
Related Work: Explicit Memory

- Stack memory for RNNs
  (Joulin et al. NIPS'15)
  - Continuous actions: PUSH, POP, NO-OP
  - Multiple stacks

- Neural Turing Machine
  (Graves et al. arXiv '14)
  - Learns how to read and write (erase + add) to the memory
  - Soft addressing
  - LSTM or feed-forward net controller
  - Can learn algorithms such as sort, associative recall and copy.

- Related to MemNN:
  [Kumar et al., arXiv:1506.07285]
  [Hermann et al., arXiv:1506.03340]





Attention-based Models

- RNNsearch: Attention in Machine Translation (Bahdanau et al., 2015)
  - Decoder can look at past encoder states using soft attention
- Image caption generation with attention (Xu et al., 2015)
  - Convnet + LSTM
  - Also Yao et al. 2015 for video

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

- Pointer Network: attention as an output (Vinyals et al., 2015)

Experiment on bAbI Q&A data

- Data: 20 bAbI tasks (Weston et al. arXiv 1502.05698, 2015)
- Answer questions after reading short story
- Small vocabulary, simple language
- Different tasks require different reasoning
- Training data size 10K for each task

```
Sam walks into the kitchen.
Sam picks up an apple.
Sam walks into the bedroom.
Sam drops the apple.
Q: Where is the apple?
A. Bedroom
```

```
Brian is a lion.
Julius is a lion.
Julius is white.
Bernhard is green.
Q: What color is Brian?
A. White
```

```
Mary journeyed to the den.
Mary went back to the kitchen.
John journeyed to the bedroom.
Mary discarded the milk.
Q: Where was the milk before the den?
A. Hallway
```

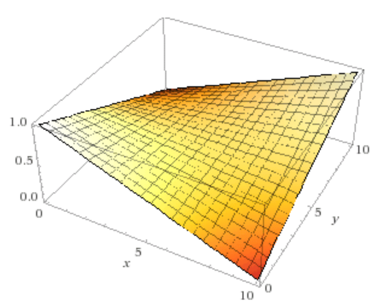
Model Details for bAbI dataset

- Sentence as memory unit
  - Need to encode sentences into vectors
- Initialize the internal state with the question
- Tried two weight tying schemes
  - Adjacent vs layer-wise
- Temporal encoding
  - Add special time words ("t1", "t2", …) into each sentences
  - Random noise injection into time/location

Sentence Representation

- Bag-of-Words
  - Embed each word into vectors and add them
- Position Encoding
  - Apply simple order dependent transformation before adding

$$l_{kj} = (1 - j/J) - (k/d)(1 - 2j/J)$$



Examples of Attention Weights

- 4 test cases:

Story (1: 1 supporting fact)	Support	Hop 1	Hop 2	Hop 3
Daniel went to the bathroom.		0.00	0.00	0.03
Mary travelled to the hallway.		0.00	0.00	0.00
John went to the bedroom.		0.37	0.02	0.00
John travelled to the bathroom.	yes	0.60	0.98	0.96
Mary went to the office.		0.01	0.00	0.00
Where is John? Answer: bathroom Prediction: bathroom				

Story (2: 2 supporting facts)	Support	Hop 1	Hop 2	Hop 3
John dropped the milk.		0.06	0.00	0.00
John took the milk there.	yes	0.88	1.00	0.00
Sandra went back to the bathroom.		0.00	0.00	0.00
John moved to the hallway.	yes	0.00	0.00	1.00
Mary went back to the bedroom.		0.00	0.00	0.00
Where is the milk? Answer: hallway Prediction: hallway				

Story (16: basic induction)	Support	Hop 1	Hop 2	Hop 3
Brian is a frog.	yes	0.00	0.98	0.00
Lily is gray.		0.07	0.00	0.00
Brian is yellow.	yes	0.07	0.00	1.00
Julius is green.		0.06	0.00	0.00
Greg is a frog.	yes	0.76	0.02	0.00
What color is Greg? Answer: yellow Prediction: yellow				

Story (18: size reasoning)	Support	Hop 1	Hop 2	Hop 3
The suitcase is bigger than the chest.	yes	0.00	0.88	0.00
The box is bigger than the chocolate.		0.04	0.05	0.10
The chest is bigger than the chocolate.	yes	0.17	0.07	0.90
The chest fits inside the container.		0.00	0.00	0.00
The chest fits inside the box.		0.00	0.00	0.00
Does the suitcase fit in the chocolate? Answer: no Prediction: no				

Results on 10k training data

	Baseline			MemN2N									
Task	Strongly Supervised MemNN	LSTM	MemNN WSH	BoW	PE	PE LS	PE LS RN	1 hop PE LS joint	2 hops PE LS joint	3 hops PE LS joint	PE LS RN joint	PE LS LW joint
1: 1 supporting fact	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2: 2 supporting facts	0.0	81.9	39.6	0.6	0.4	0.5	0.3	62.0	1.3	2.3	1.0	0.8
3: 3 supporting facts	0.0	83.1	79.5	17.8	12.6	15.0	9.3	80.0	15.8	14.0	6.8	18.3
4: 2 argument relations	0.0	0.2	36.6	31.8	0.0	0.0	0.0	21.4	0.0	0.0	0.0	0.0
5: 3 argument relations	0.3	1.2	21.1	14.2	0.8	0.6	0.8	8.7	7.2	7.5	6.1	0.8
6: yes/no questions	0.0	51.8	49.9	0.1	0.2	0.1	0.0	6.1	0.7	0.2	0.1	0.1
7: counting	3.3	24.9	35.1	10.7	5.7	3.2	3.7	14.8	10.5	6.1	6.6	8.4
8: lists/sets	1.0	34.1	42.7	1.4	2.4	2.2	0.8	8.9	4.7	4.0	2.7	1.4
9: simple negation	0.0	20.2	36.4	1.8	1.3	2.0	0.8	3.7	0.4	0.0	0.0	0.2
10: indefinite knowledge	0.0	30.1	76.0	1.9	1.7	3.3	2.4	10.3	0.6	0.4	0.5	0.0
11: basic coreference	0.0	10.3	25.3	0.0	0.0	0.0	0.0	8.3	0.0	0.0	0.0	0.4
12: conjunction	0.0	23.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0
13: compound coreference	0.0	6.1	12.3	0.0	0.1	0.0	0.0	5.6	0.0	0.0	0.0	0.0
14: time reasoning	0.0	81.0	8.7	0.0	0.2	0.0	0.0	30.9	0.2	0.2	0.0	1.7
15: basic deduction	0.0	78.7	68.8	12.5	0.0	0.0	0.0	42.6	0.0	0.0	0.2	0.0
16: basic induction	0.0	51.9	50.9	50.9	48.6	0.1	0.4	47.3	46.4	0.4	0.2	49.2
17: positional reasoning	24.6	50.1	51.1	47.4	40.3	41.1	40.7	40.0	39.7	41.7	41.8	40.0
18: size reasoning	2.1	6.8	45.8	41.3	7.4	8.6	6.7	9.2	10.1	8.6	8.0	8.4
19: path finding	31.9	90.3	100.0	75.4	66.6	66.7	66.5	91.0	80.8	73.3	75.7	89.5
20: agent's motivation	0.0	2.1	4.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Mean error (%)	3.2	36.4	39.2	15.4	9.4	7.2	6.6	24.5	10.9	7.9	7.5	11.0
Failed tasks (err. $> 5\%$)	2	16	17	9	6	4	4	16	7	6	6	6

Table 3: Test error rates (%) on the 20 bAbI QA tasks for models using 10k training examples. Key: BoW = bag-of-words representation; PE = position encoding representation; LS = linear start training; RN = random injection of time index noise; LW = RNN-style layer-wise weight tying (if not stated, adjacent weight tying is used); joint = joint training on all tasks (as opposed to per-task training).

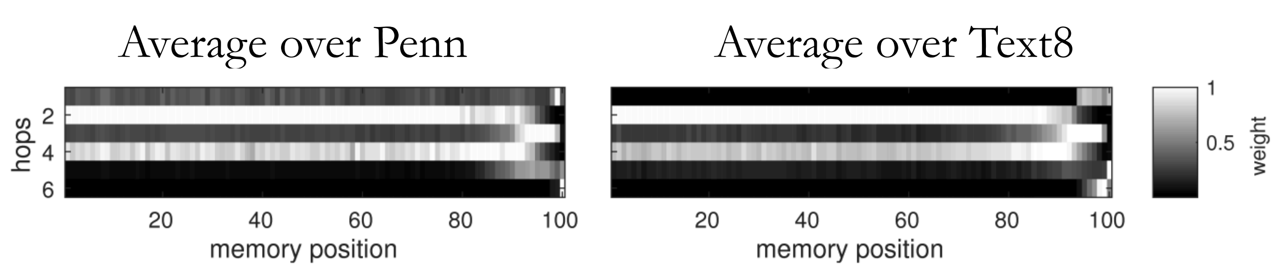
Experiment on Language Modeling

- Data
  - Penn Tree Bank (PTB): 1M words, 10K vocab
  - Text8: wikipedia 100M chars, 40K vocab
- Model
  - Main module: linear + non-linearity (half)
    - Layer-wise tying
    - Linear projection and non-linearity
  - Words as memory unit

Results on Language Modeling

Model	Penn Treebank # of hidden	# of hops	memory size	Valid. perp.	Test perp.	Text8 # of hidden	# of hops	memory size	Valid. perp.	Test perp.
RNN [15]	300	-	-	133	129	500	-	-	-	184
LSTM [15]	100	-	-	120	115	500	-	-	122	154
SCRN [15]	100	-	-	120	115	500	-	-	-	161
MemN2N	150	2	100	128	121	500	2	100	152	187
	150	3	100	129	122	500	3	100	142	178
	150	4	100	127	120	500	4	100	129	162
	150	5	100	127	118	500	5	100	123	154
	150	6	100	122	115	500	6	100	124	155
	150	7	100	120	114	500	7	100	118	**147**
	150	6	25	125	118	500	6	25	131	163
	150	6	50	121	114	500	6	50	132	166
	150	6	75	122	114	500	6	75	126	158
	150	6	100	122	115	500	6	100	124	155
	150	6	125	120	112	500	6	125	125	157
	150	6	150	121	114	500	6	150	123	154
	150	7	200	118	**111**	-	-	-	-	-

Table 2: The perplexity on the test sets of Penn Treebank and Text8 corpora. Note that increasing the number of memory hops improves performance.



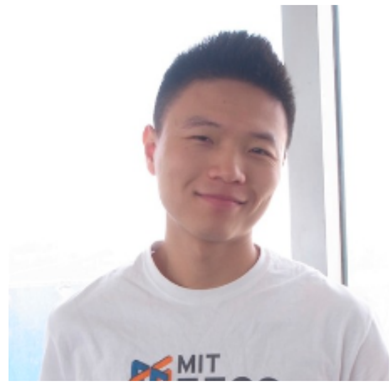
Conclusions

- Simple model that combines external memory with an RNN
- Versatile: can be applied to range of tasks
  - Language modeling, bAbI dataset
- Code available at: https://github.com/facebook/MemNN
- Interesting to explore biological parallels
  - E.g. hippocampus & PFC

Thanks!

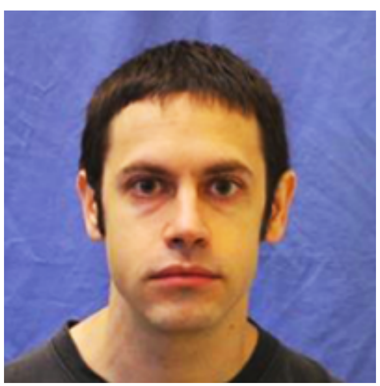
PhD students & Facebook AI Research colleagues

Sainbayar Sukhbaatar (NYU)



Bolei Zhu (MIT)

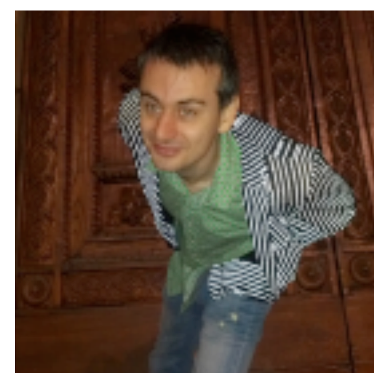
Yuandong Tian



Arthur Szlam

Sumit Chopra

Antoine Bordes



Jason Weston