Neural Turing Machines and Related

Arne Nix
arne.nix@rwth-aachen.de

Feburary 1, 2017, Aachen

Human Language Technology and Pattern Recognition
Computer Science Department, RWTH Aachen University

Outline

Introduction
Related Work
Computational Power of Neural Networks

Augmenting RNNs with Memory
Attention
Stack Augmented RNN
Giving RNNs controllable RAM

Results
Arithmetic Tasks
Question Answering Tasks
Machine Translation

Conclusion and Discussion

Related Work

H. Siegelmann, E. Sontag [Siegelmann & Sontag 92]

On the computational power of neural nets.

- **Theoretical proof that RNNs are Turing complete.**

A. Graves [Graves 13]:

Generating sequences with recurrent neural networks
***On arXiv: August 2013*.**

- **Introduces the attention mechanism for hand-writing synthesis**
- **Most popular application: neural machine translation (NMT) [Bahdanau & Cho^{+} 15]**
- **Used as addressing for many augmented memory approaches**

Related Work

J. Weston, S. Chopra, A. Bordes [Weston & Chopra$^+$ 14]:

Memory Networks. ***ICLR: May 2015; On arXiv: October 2014***

- Introducing memory networks
- Application to Question Answering

A. Graves, G. Wayne, I. Danihelka [Graves & Wayne$^+$ 14]

Neural Turing Machines. ***On arXiv: October 2014***

- Introducing neural Turing machines with read and write heads
- Promising results on algorithmic toy tasks
- Many extensions:
  - Dynamic NTM (D-NTM) [Gulcehre & Chandar$^+$ 16]
  - Differentiable Neural Computer (DNC) [Graves & Wayne$^+$ 16]

Computational Power of Neural Networks

Theory:

- Sigelmann and Sontag [Siegelmann & Sontag 92] proved:

  RNNs are Turing complete.

- Proof by simulating two-stack machine which is also Turing complete.
  - Representing stack as a rational number: $s = \sum_{i=1}^{n} \frac{a_i}{4^i}$

Practice:

- Rational numbers have limited precision
  $\Rightarrow$ proof does not hold in practice.
- Standard RNNs are limited to simulate finite state machines [Tino & Horne^{+} 98, Kolen 94].
- Reason:
  - Memory fixed and limited
    $\Rightarrow$ no generalization on problems with $\mathcal{O}(N)$ memory requirement.
- Solution:
  - Augment RNN with memory that can be increased without retraining.

Outline

Introduction

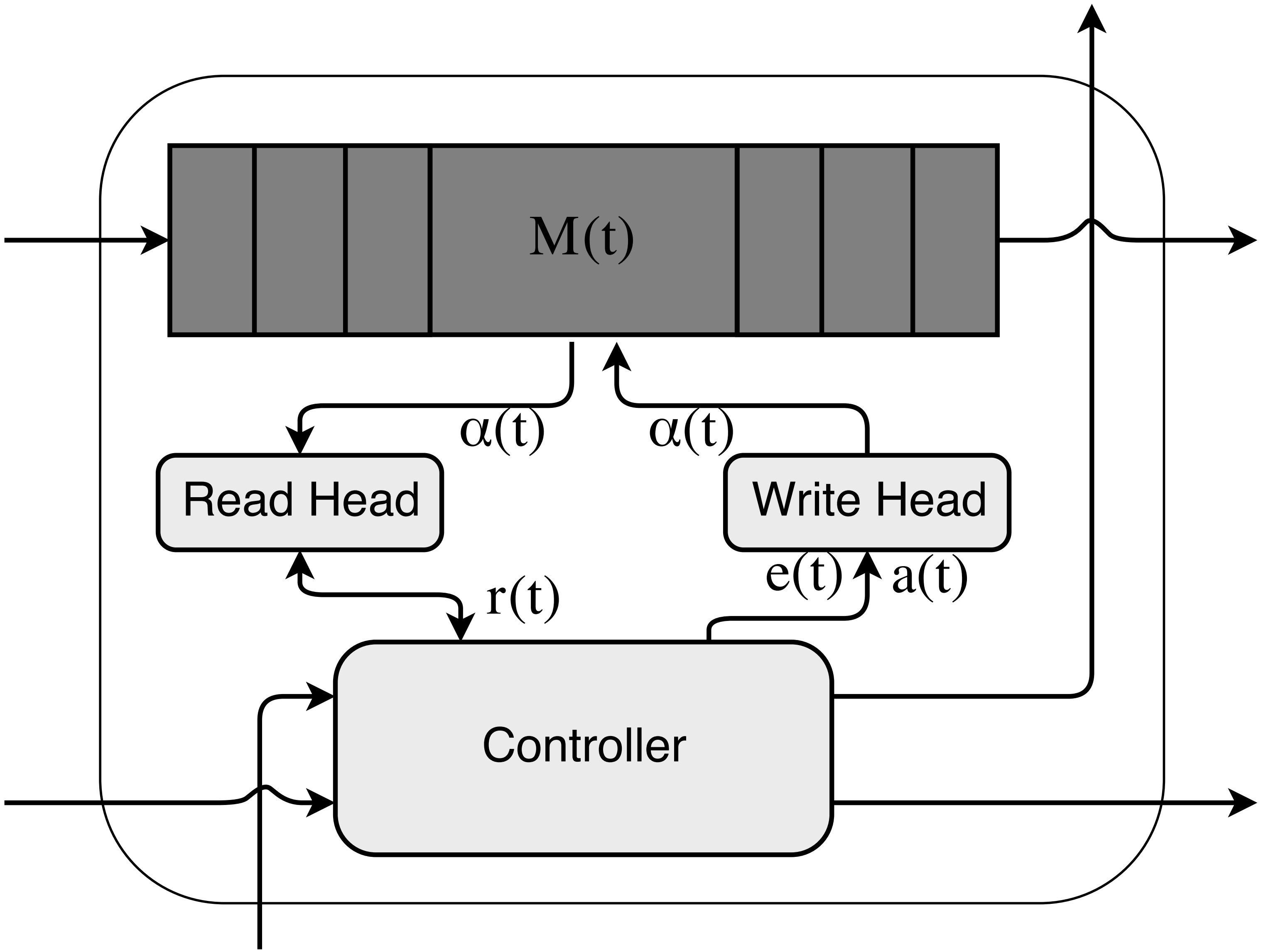
Augmenting RNNs with Memory

- Attention
- Stack Augmented RNN
- Giving RNNs controllable RAM

Results

Conclusion and Discussion

Augmenting RNNs with Memory



General concept of memory-augmented RNNs

Computational Hierarchy

Turing Machines (2 Stacks/ Tape)
$\rightarrow$ computable functions

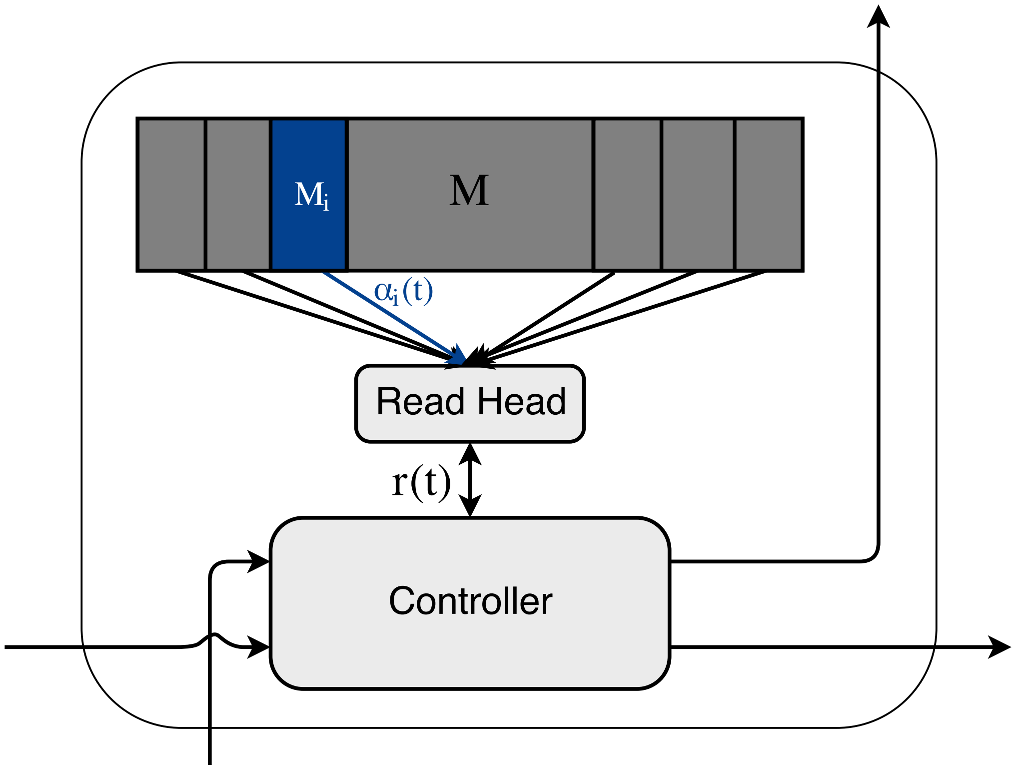
$\Uparrow \Uparrow \Uparrow$

Pushdown Automata (1 Stack)
$\rightarrow$ context free languages

$\Uparrow \Uparrow \Uparrow$

Finite State Machines (0 Stacks)
$\rightarrow$ regular languages

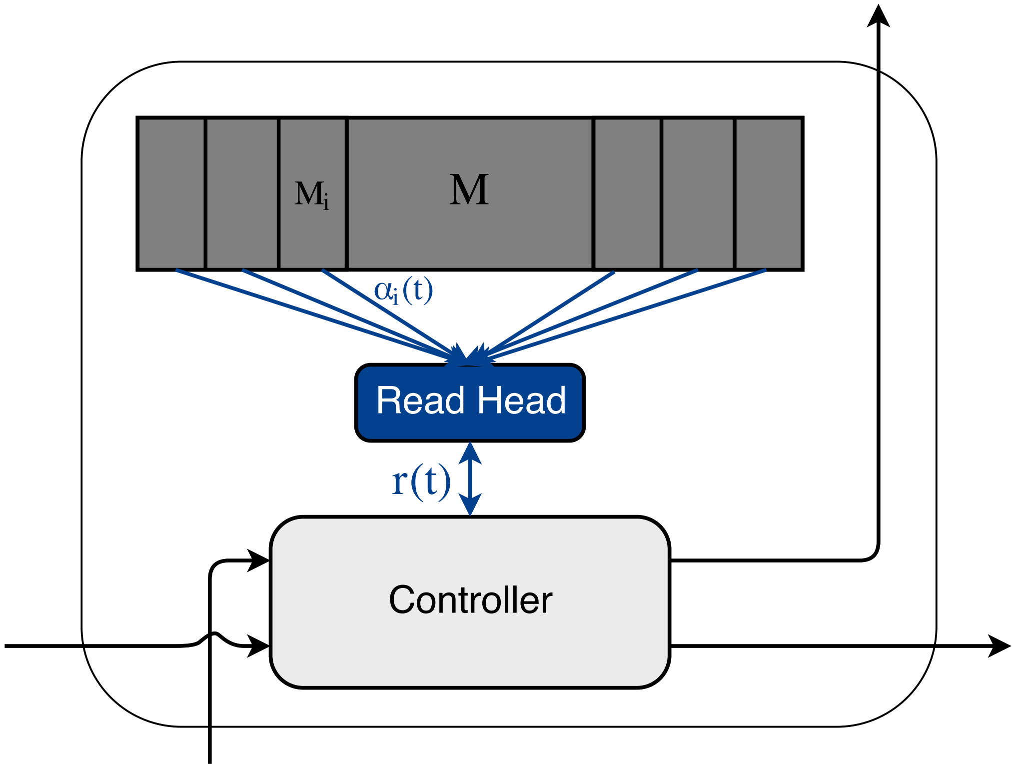
Attention [Bahdanau & Cho$^+$ 15]



- **Addressing with key $k_i(t)$, input $x(t)$ and some function f_{att}:**

$$\alpha_i(t) = \frac{f_{\text{att}}(k_i(t), x(t))}{\sum_j f_{\text{att}}(k_j(t), x(t))}$$

Attention [Bahdanau & Cho$^+$ 15]



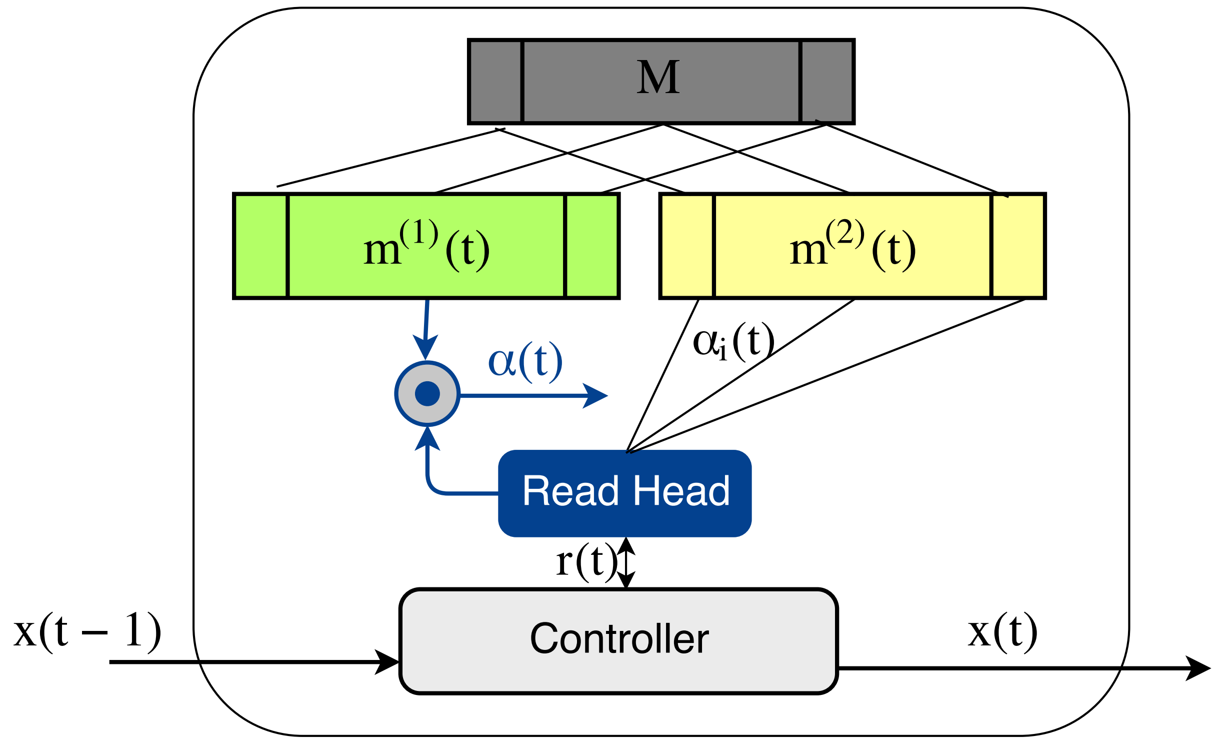
- **Lookup:**

$$r(t) = \sum_i \alpha_i(t) M_i$$

End To End Memory Networks (MemN2N) [Sukhbaatar & Weston^{+} 15]

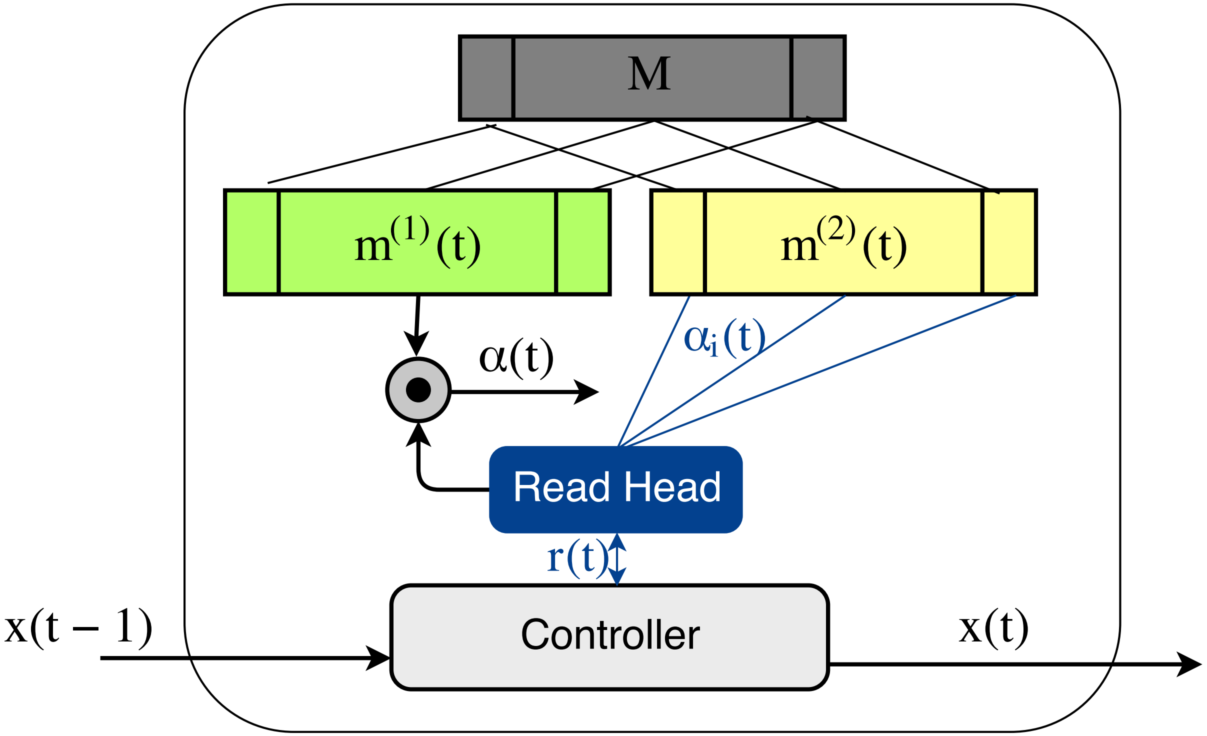
Addressing:

$$\alpha_i(t) = \frac{\exp([x(t-1)]^T m_i^{(1)}(t))}{\sum_j \exp([x(t-1)]^T m_j^{(1)}(t))}$$



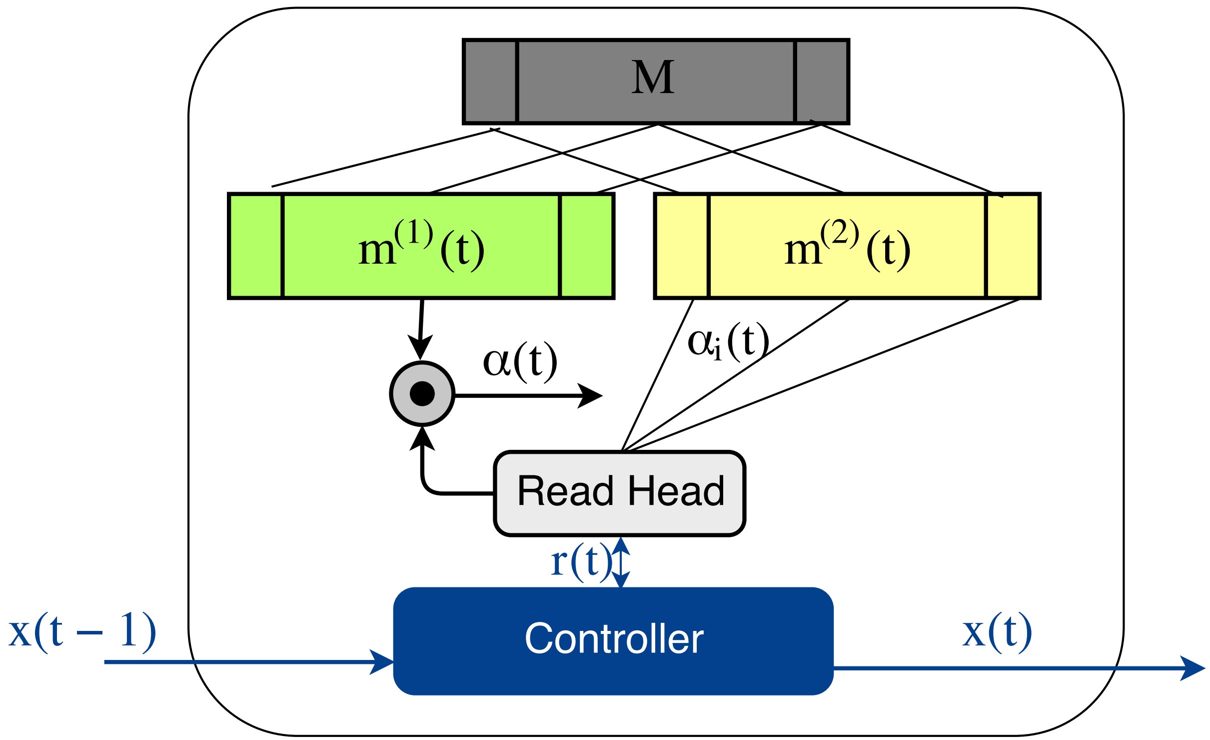
End To End Memory Networks (MemN2N) [Sukhbaatar & Weston$^+$ 15]

Lookup: $$r(t) = \sum_i \alpha_i(t) m_i^{(2)}(t)$$



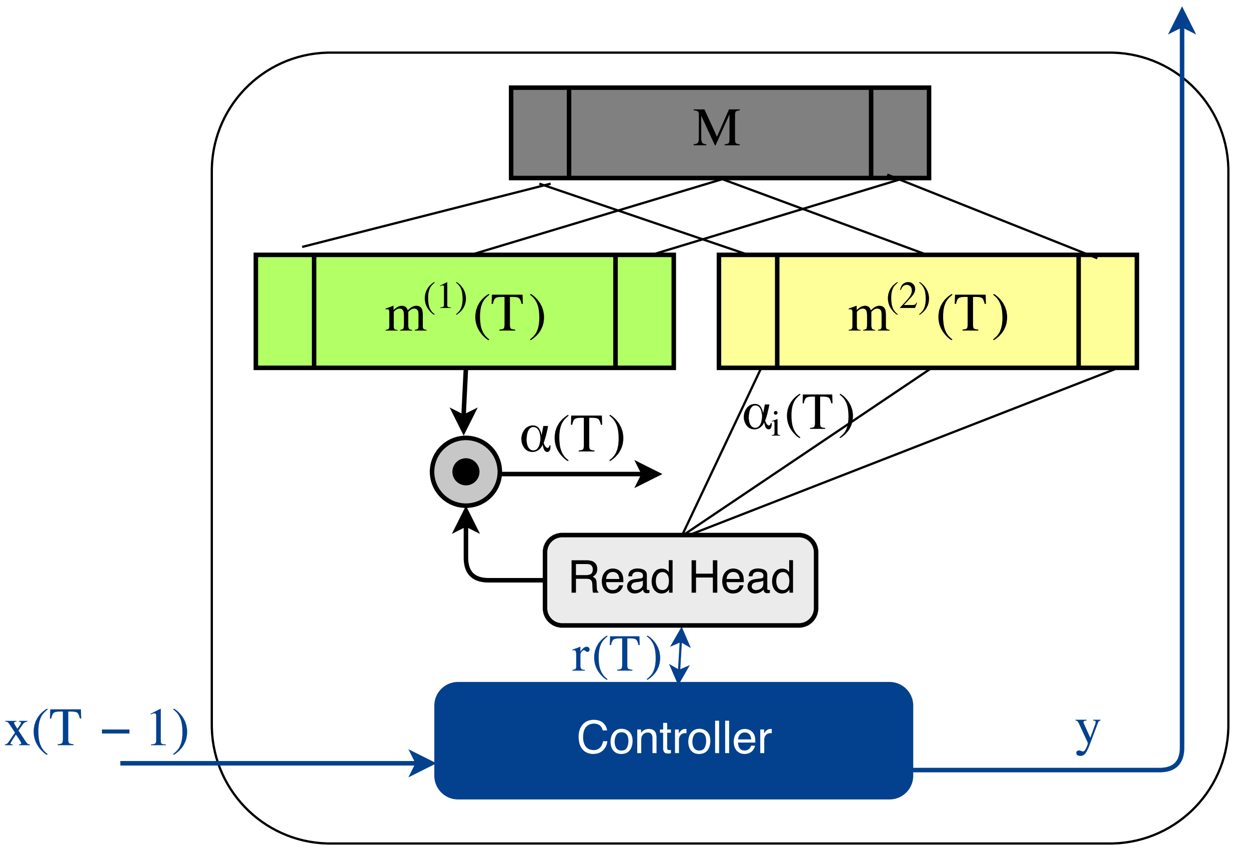
End To End Memory Networks (MemN2N) [Sukhbaatar & Weston+ 15]

State update: $$\boldsymbol{x}(\boldsymbol{t}) = \boldsymbol{x}(\boldsymbol{t}-1) + \boldsymbol{r}(\boldsymbol{t})$$



End To End Memory Networks (MemN2N) [Sukhbaatar & Weston+ 15]

Output: $$y = \textbf{softmax}(W[r(T) + x(T-1)])$$



Computational Hierarchy

Turing Machines (2 Stacks/ Tape)
$\rightarrow$ computable functions

$\Uparrow \Uparrow \Uparrow$

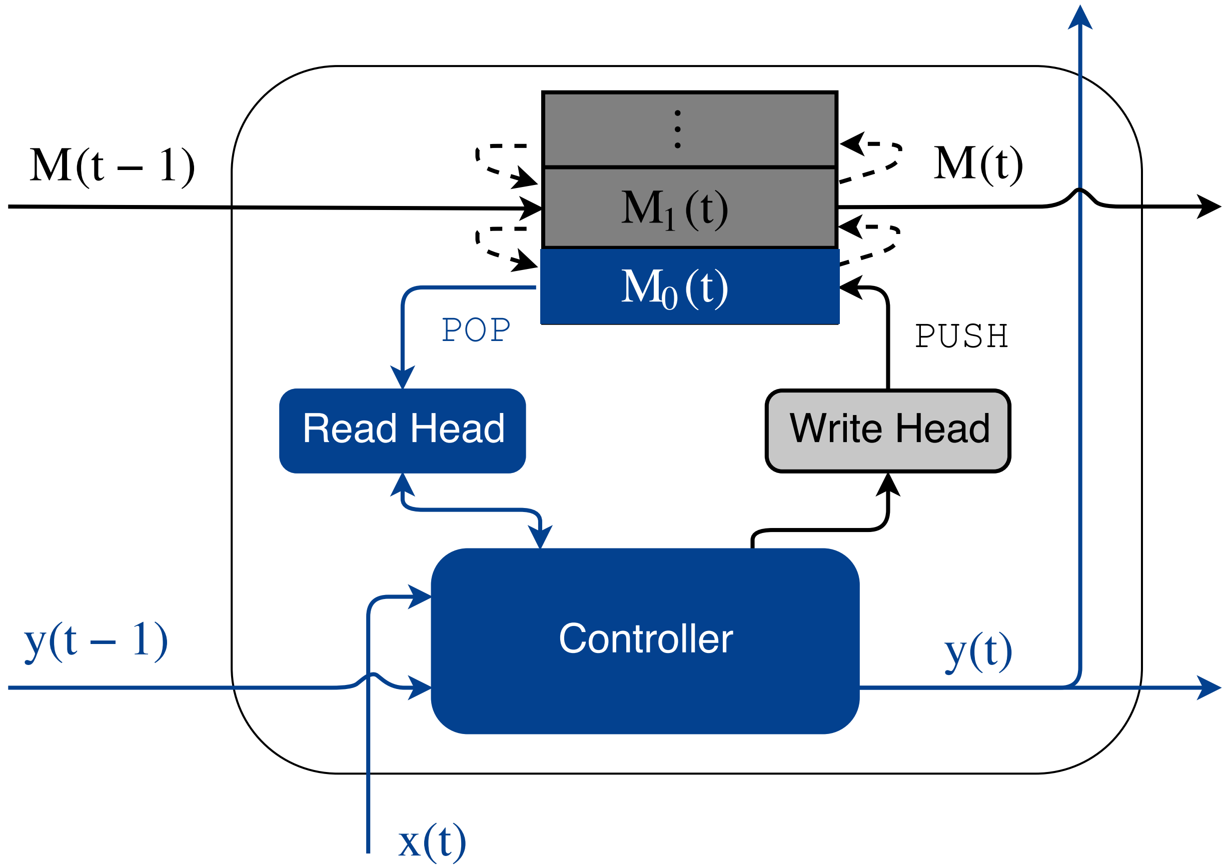
Pushdown Automata (1 Stack)
$\rightarrow$ context free languages

$\Uparrow \Uparrow \Uparrow$

Finite State Machines (0 Stacks)
$\rightarrow$ regular languages

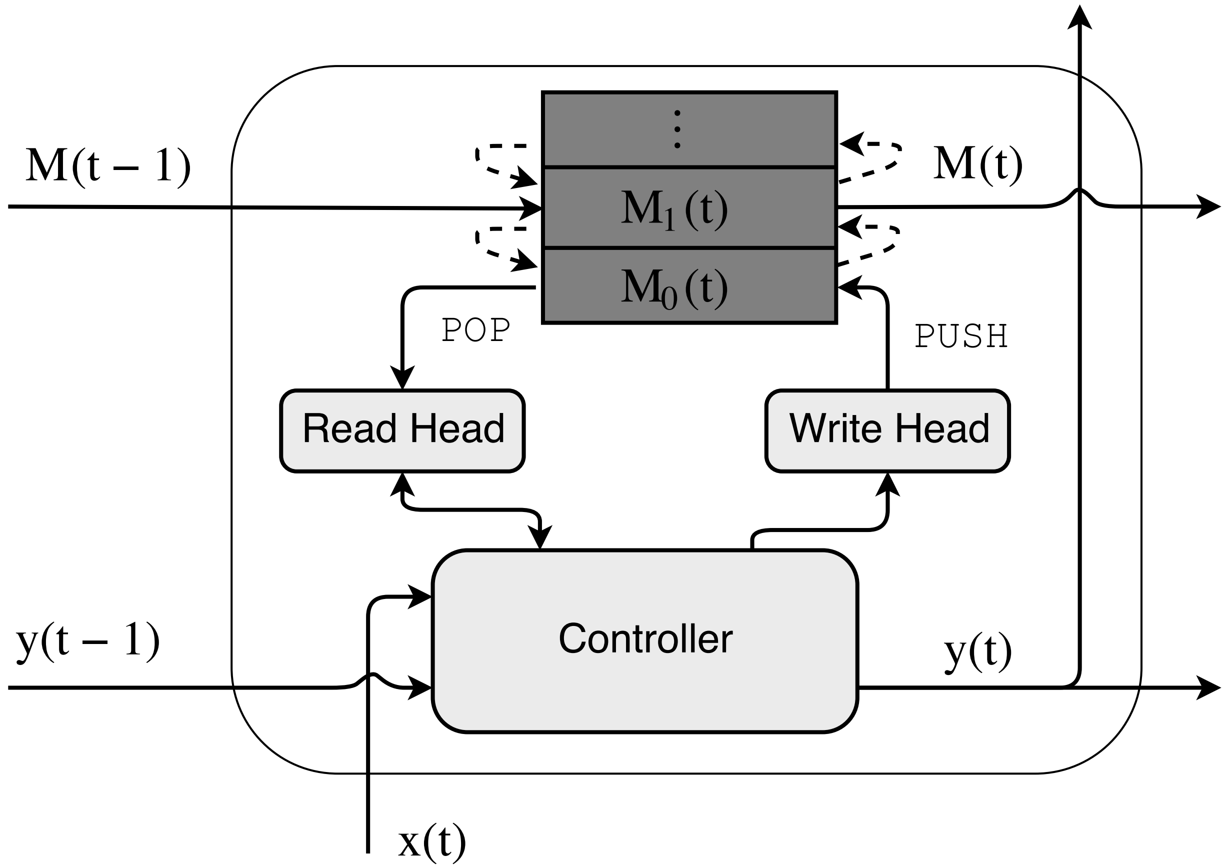
Stack Augmented RNN [Joulin & Mikolov 15]

RNN step: $$y(t) = \sigma(W_x x(t) + W_y y(t-1) + W_M M_{0:k}(t-1))$$



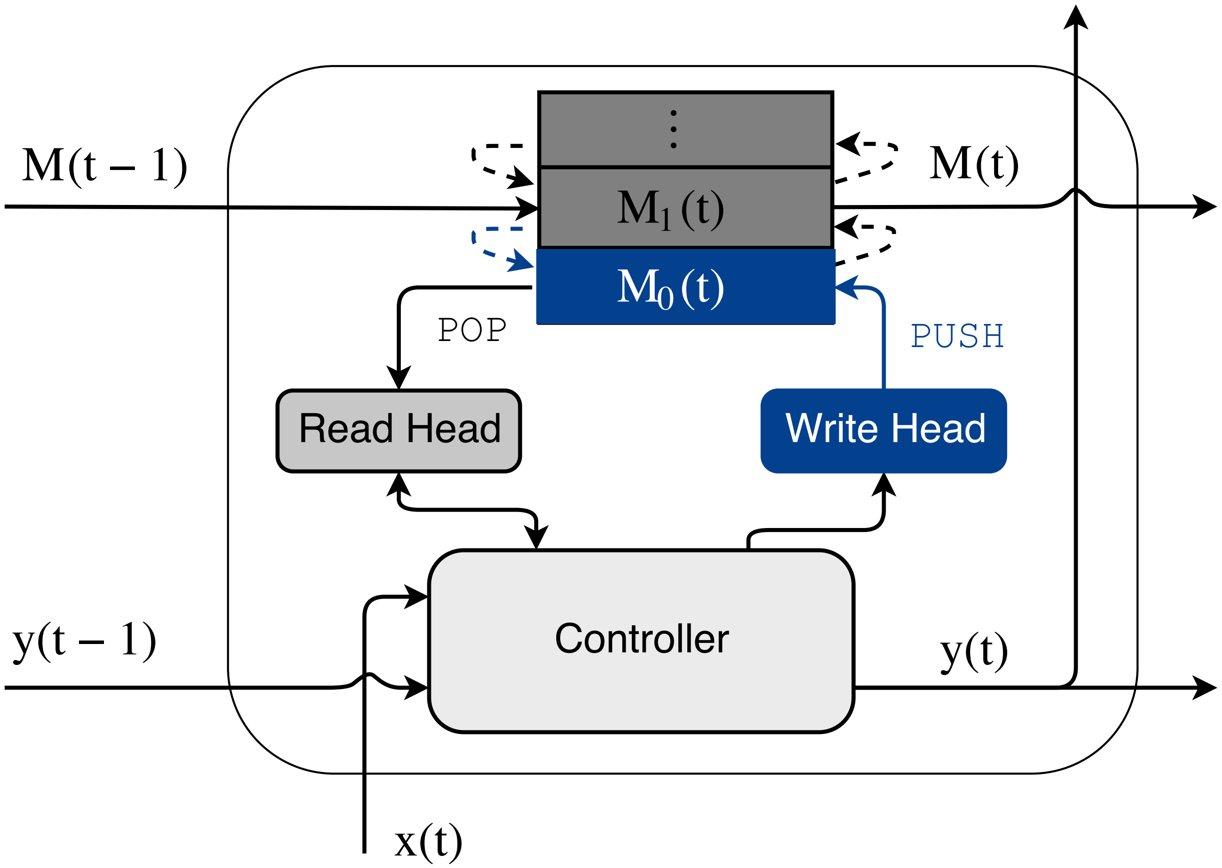
Stack Augmented RNN [Joulin & Mikolov 15]

Addressing: $\alpha(t) = \textbf{softmax}(W_\alpha y(t))$



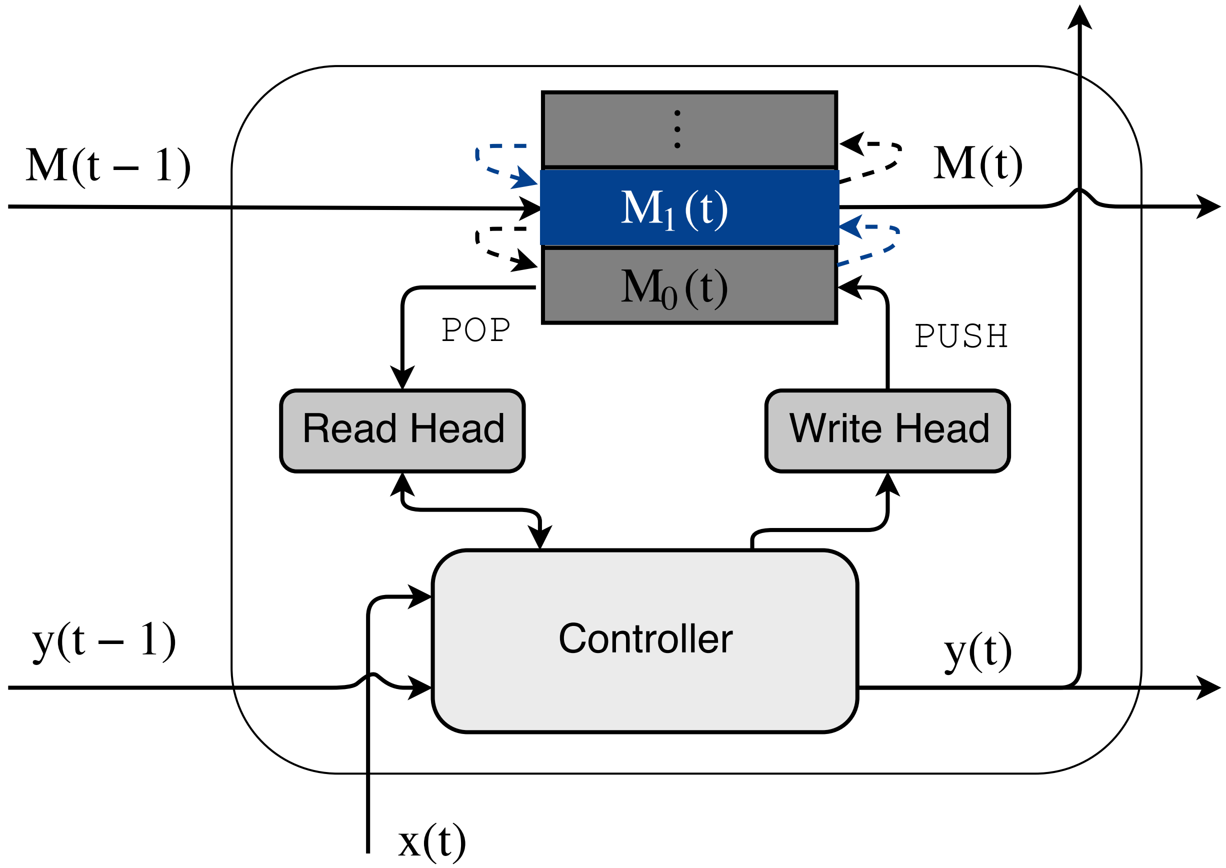
Stack Augmented RNN [Joulin & Mikolov 15]

Stack update: $M_0(t) = \alpha_{\texttt{PUSH}}(t)\sigma[W_{\texttt{PUSH}}y(t)] + \alpha_{\texttt{POP}}(t)M_1(t-1)$



Stack Augmented RNN [Joulin & Mikolov 15]

Stack update: $$M_i(t) = \alpha_{\texttt{PUSH}}(t) M_{i-1}(t-1) + \alpha_{\texttt{POP}}(t) M_{i+1}(t-1)$$



Computational Hierarchy

Turing Machines (2 Stacks/ Tape)
$\rightarrow$ computable functions

$\Uparrow \Uparrow \Uparrow$

Pushdown Automata (1 Stack)
$\rightarrow$ context free languages

$\Uparrow \Uparrow \Uparrow$

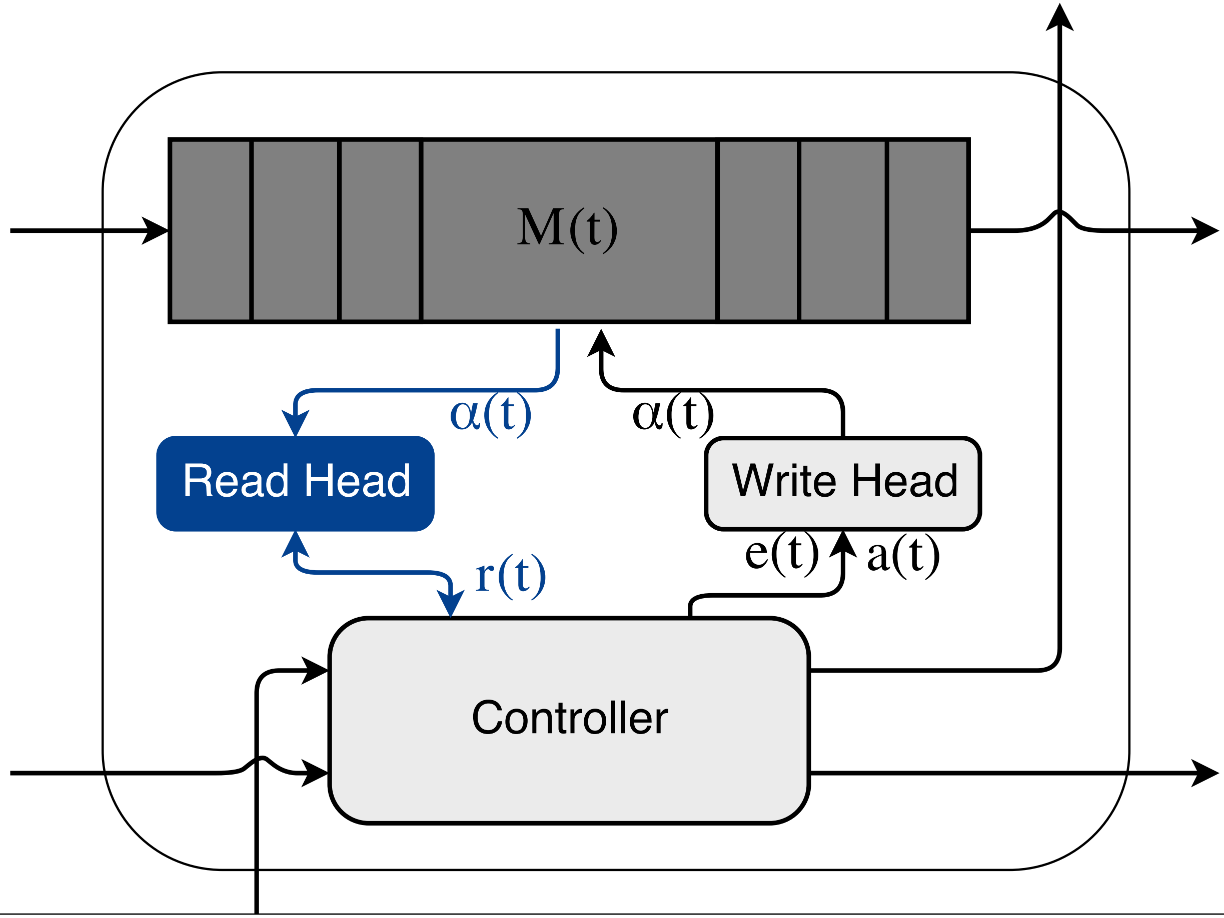
Finite State Machines (0 Stacks)
$\rightarrow$ regular languages

Neural Turing Machine [Graves & Wayne+ 14]

Read-Head:

$$r(t) = \sum_{i=1}^{N} \alpha_i^{\text{read}}(t) M_i(t)$$

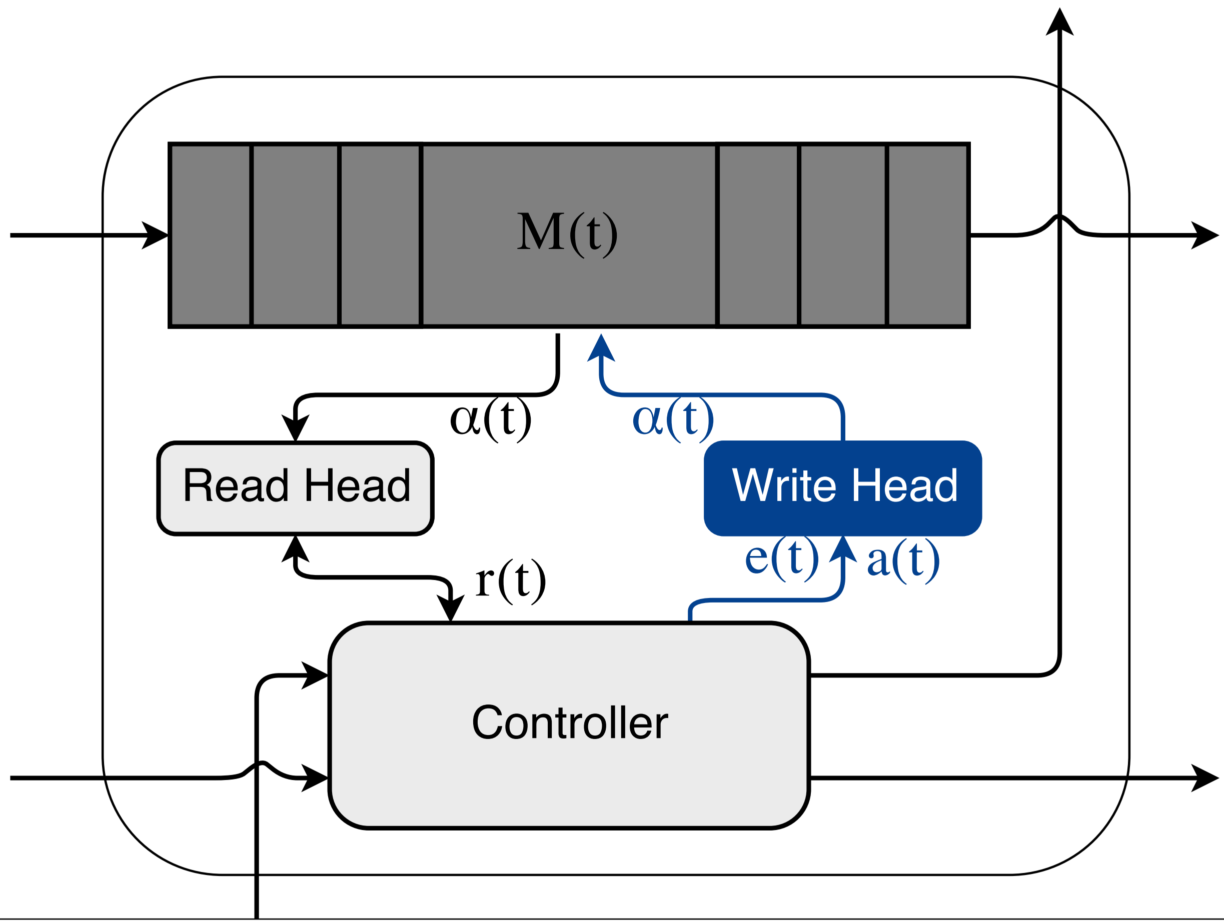
with addressing $\alpha_i^{\text{read}}(t)$



Neural Turing Machine [Graves & Wayne+ 14]

Write-Head: $M_i(t) = M_i(t-1)[1 - \alpha_i^{\text{erase}}(t)e(t)] + \alpha_i^{\text{add}}(t)a(t)$

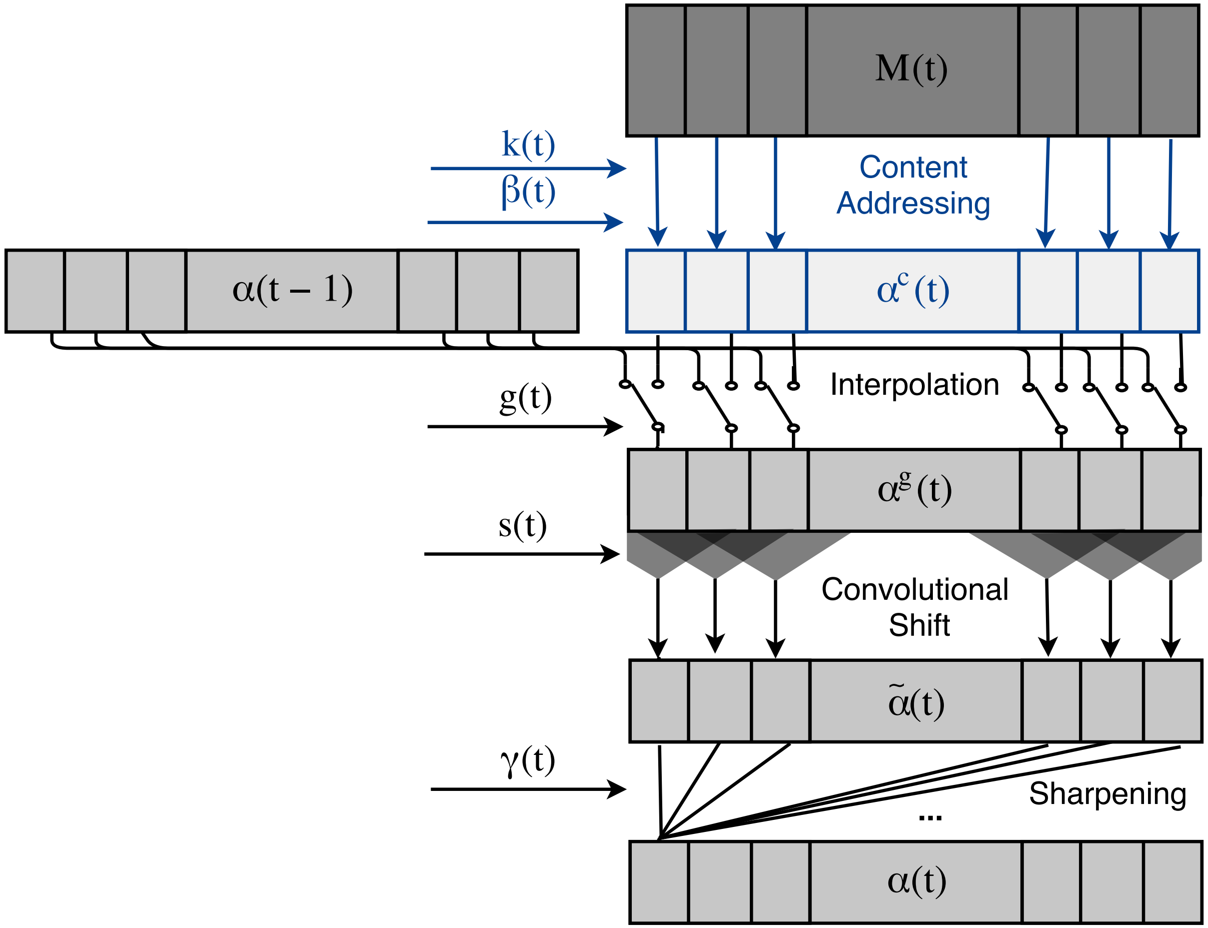
with erase vector $e(t)$, add vector $a(t)$ and addressings $\alpha_i^{\text{erase}}(t)$,$\alpha_i^{\text{add}}(t)$



NTM Addressing (Content-based)

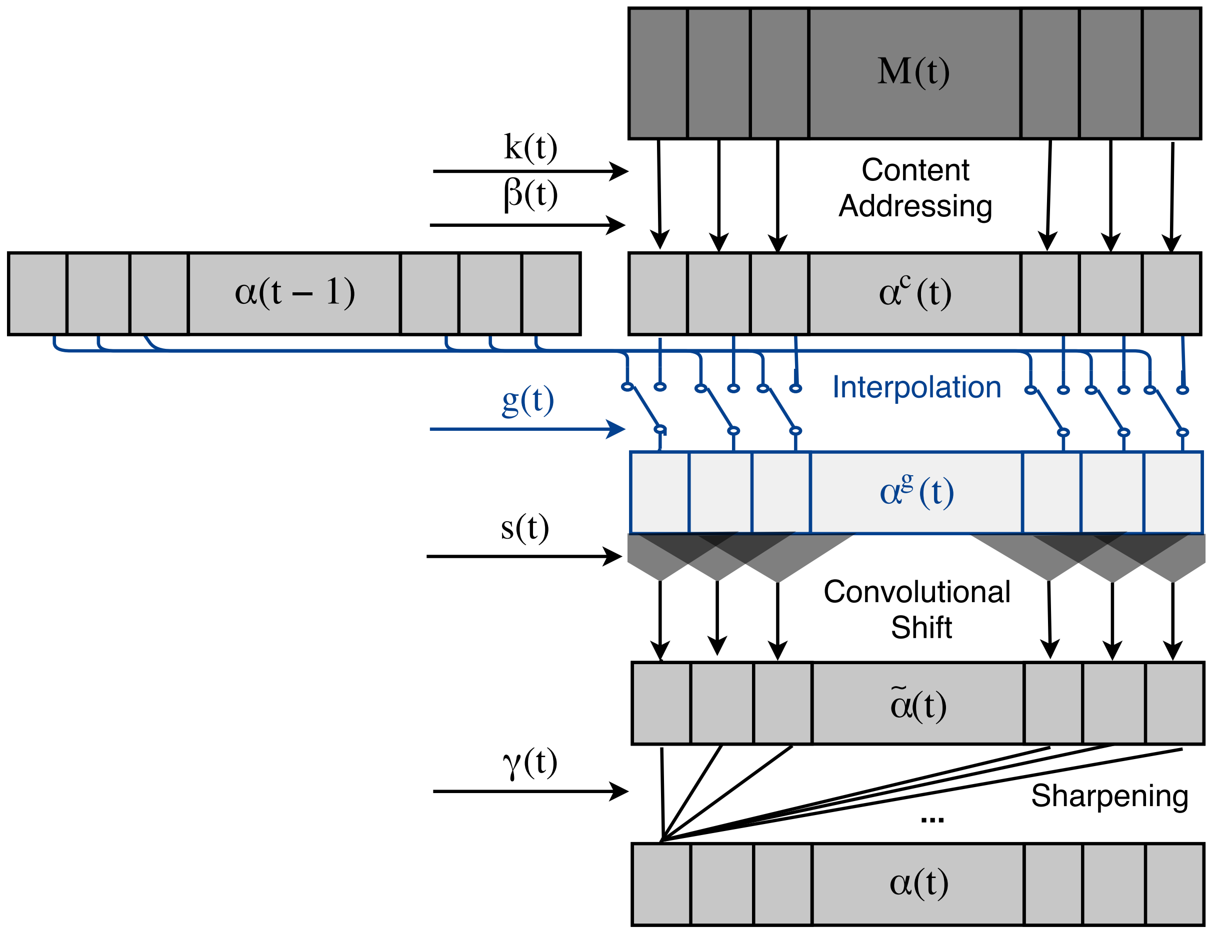
Content Addressing:

$$\alpha_i^c(t) = \frac{\beta(t) K[k(t), M_i(t)]}{\sum_j \beta(t) K[k(t), M_j(t)]}$$



NTM Addressing (Location-based)

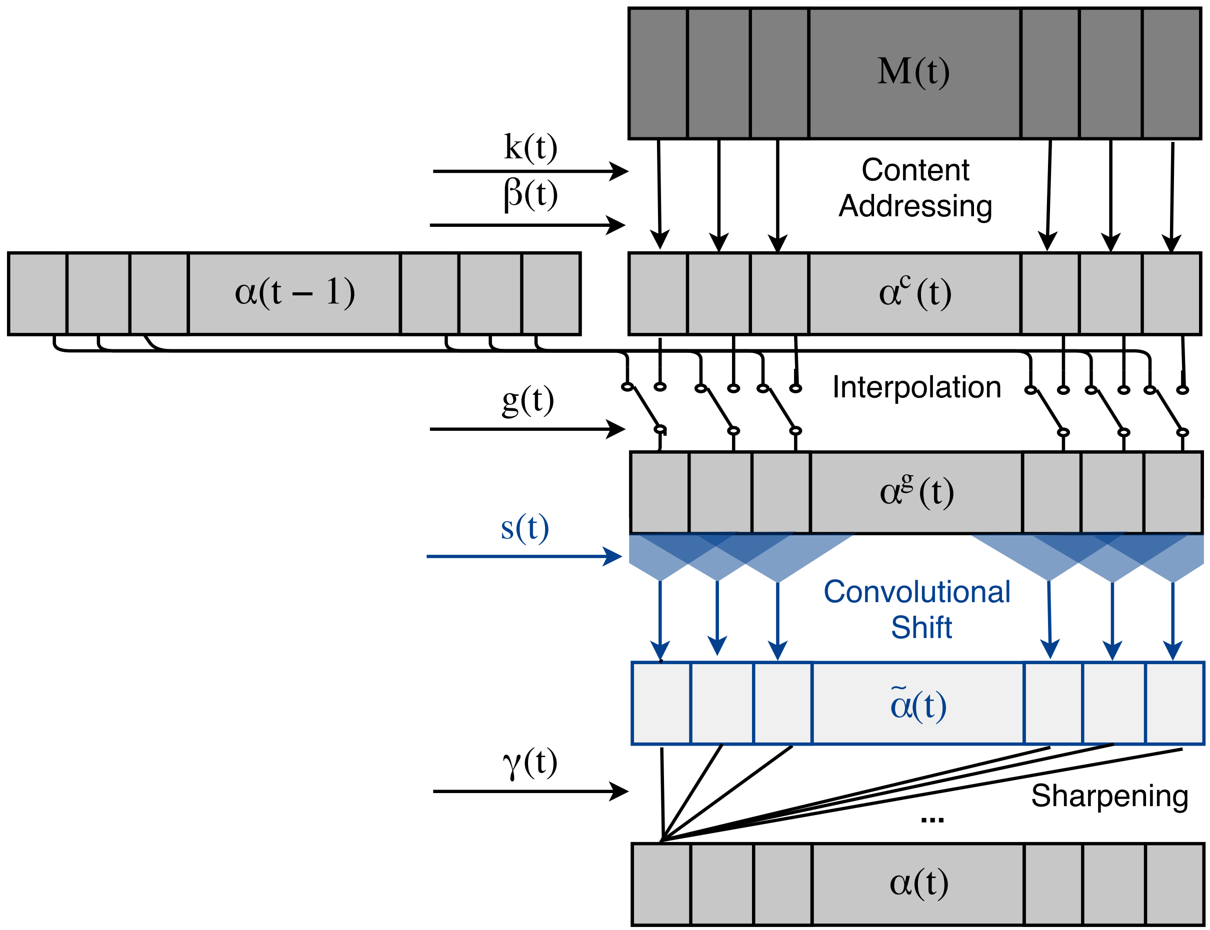
Interpolation: $$\alpha^g(t) = g(t)\alpha^c(t) + [1 - g(t)]\alpha(t-1)$$



NTM Addressing (Location-based)

Convolutional Shift:

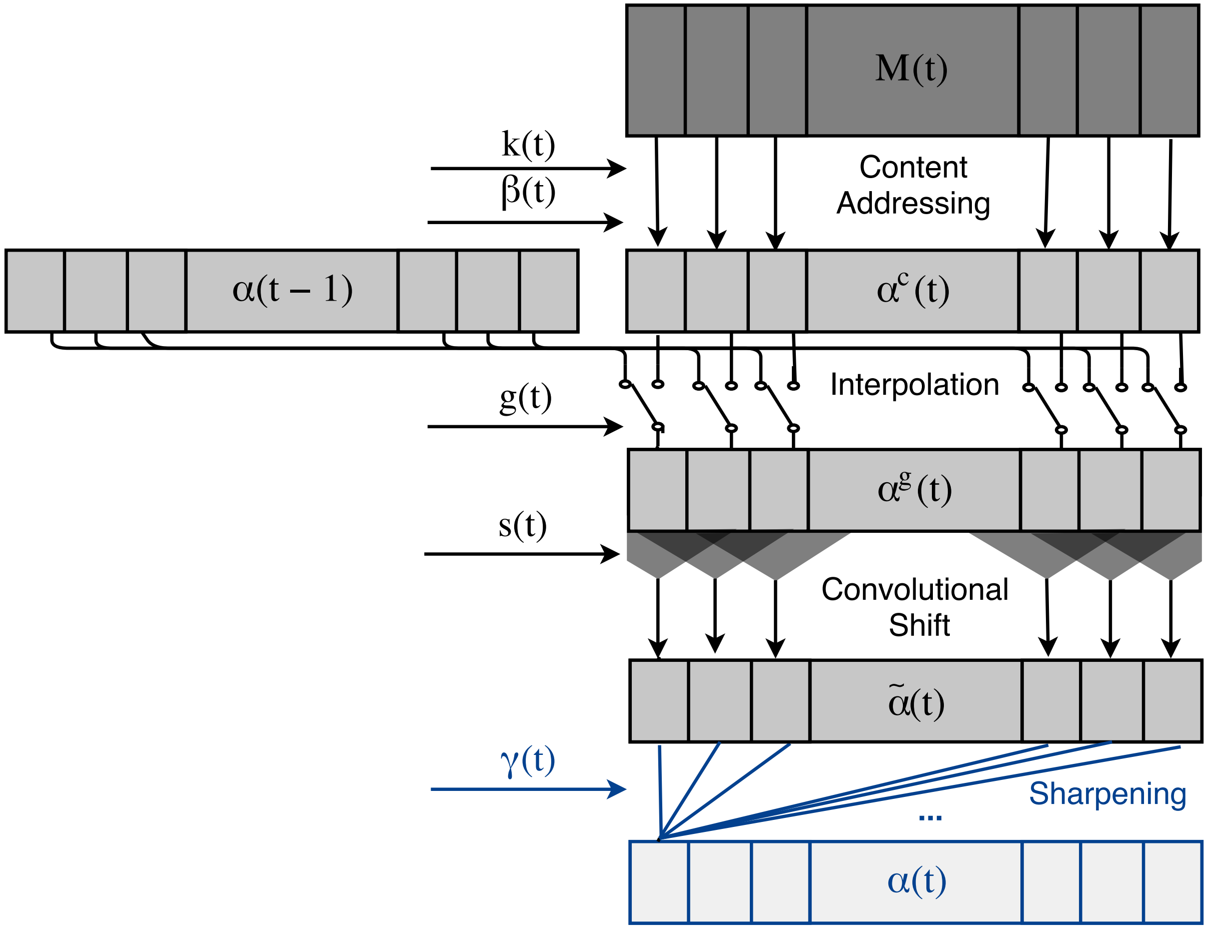
$$\tilde{\alpha}_i(t) = \sum_{j=0}^{N-1} \alpha_j^g(t) s_{i-j}(t)$$



NTM Addressing (Location-based)

Sharpening:

$$\alpha_i(t) = \frac{\tilde{\alpha}_i(t)^{\gamma(t)}}{\sum_{j=1}^{N} \tilde{\alpha}_j(t)^{\gamma(t)}}$$



Computational Hierarchy

Turing Machines (2 Stacks/ Tape)
$\rightarrow$ computable functions

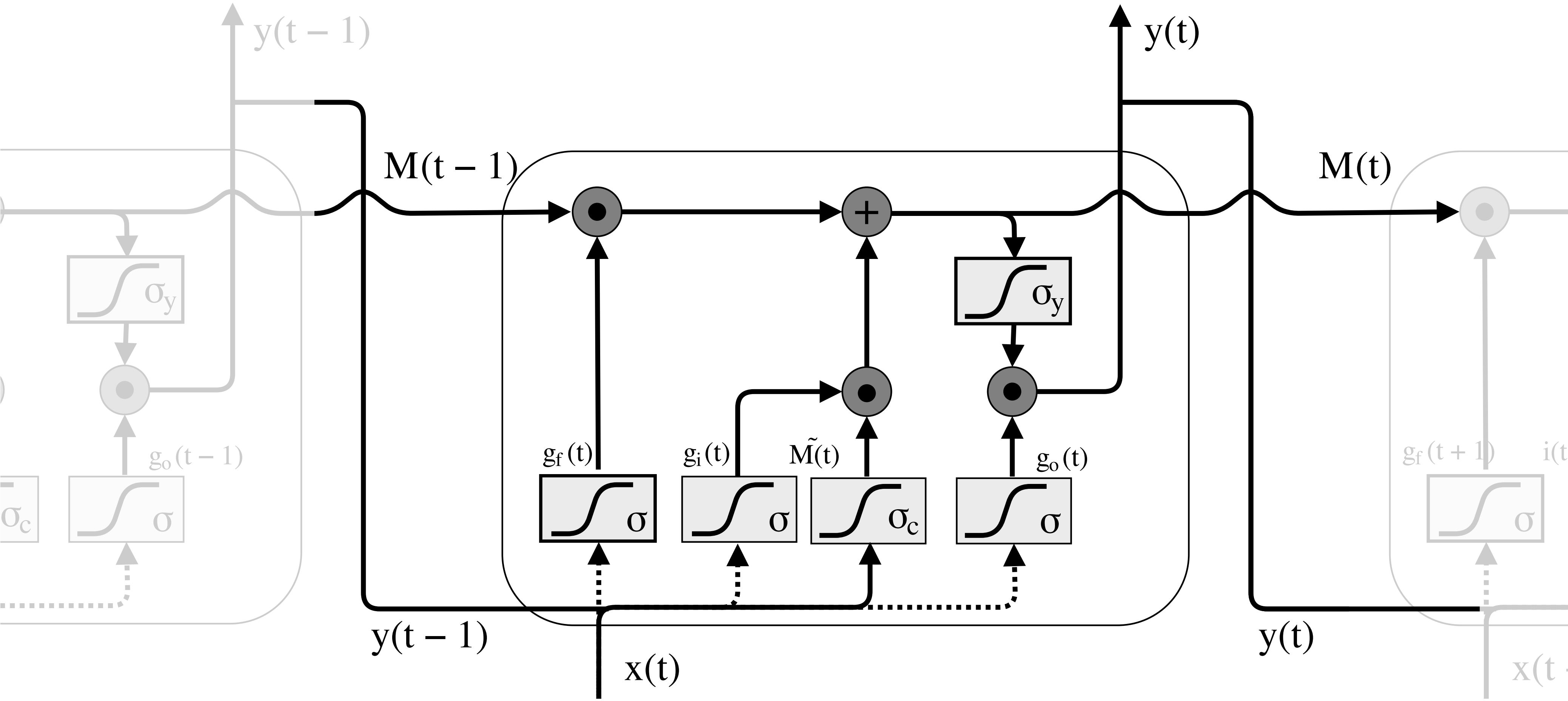
$\Uparrow \Uparrow \Uparrow$

Pushdown Automata (1 Stack)
$\rightarrow$ context free languages

$\Uparrow \Uparrow \Uparrow$

Finite State Machines (0 Stacks)
$\rightarrow$ regular languages

Long Short-term Memory (LSTM) [Hochreiter & Schmidhuber 97, Gers & Schmidhuber+ 00]



Long Short-term Memory (LSTM)

Associative LSTM [Danihelka & Wayne+ 16]

- Extend LSTM with key-value access
- Implemented using holographic reduced representations [Plate 95]
  - array of key-value pairs saved as the sum of the pairs
- All vectors interpreted as complex vectors:

$$h = \begin{bmatrix} h_{\text{real}} \\ h_{\text{imaginary}} \end{bmatrix}$$

- Activation function to restrict modulus to the range of zero to one:

$$\sigma_{\text{bound}}(h) = \begin{bmatrix} h_{\text{real}} \oslash d \\ h_{\text{imaginary}} \oslash d \end{bmatrix}$$

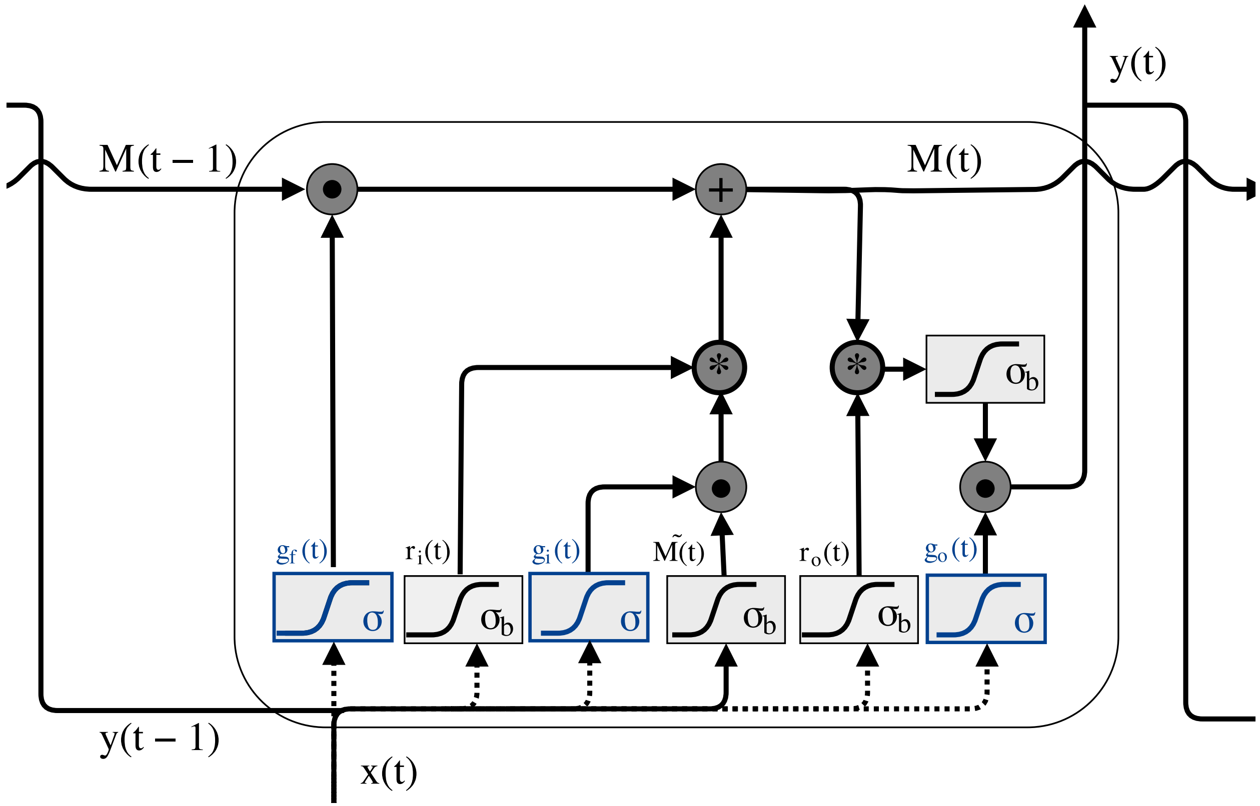
with $d = \max(1, \sqrt{h_{\text{real}} \odot h_{\text{real}} + h_{\text{imaginary}} \odot h_{\text{imaginary}}})$

Associative LSTM [Danihelka & Wayne+ 16]

Gating:

$$\hat{g}_\star(t) = W_{x\star}x(t) + W_{y\star}y(t-1) + b_\star \qquad \text{for } \star \in \{f,i,o\}$$

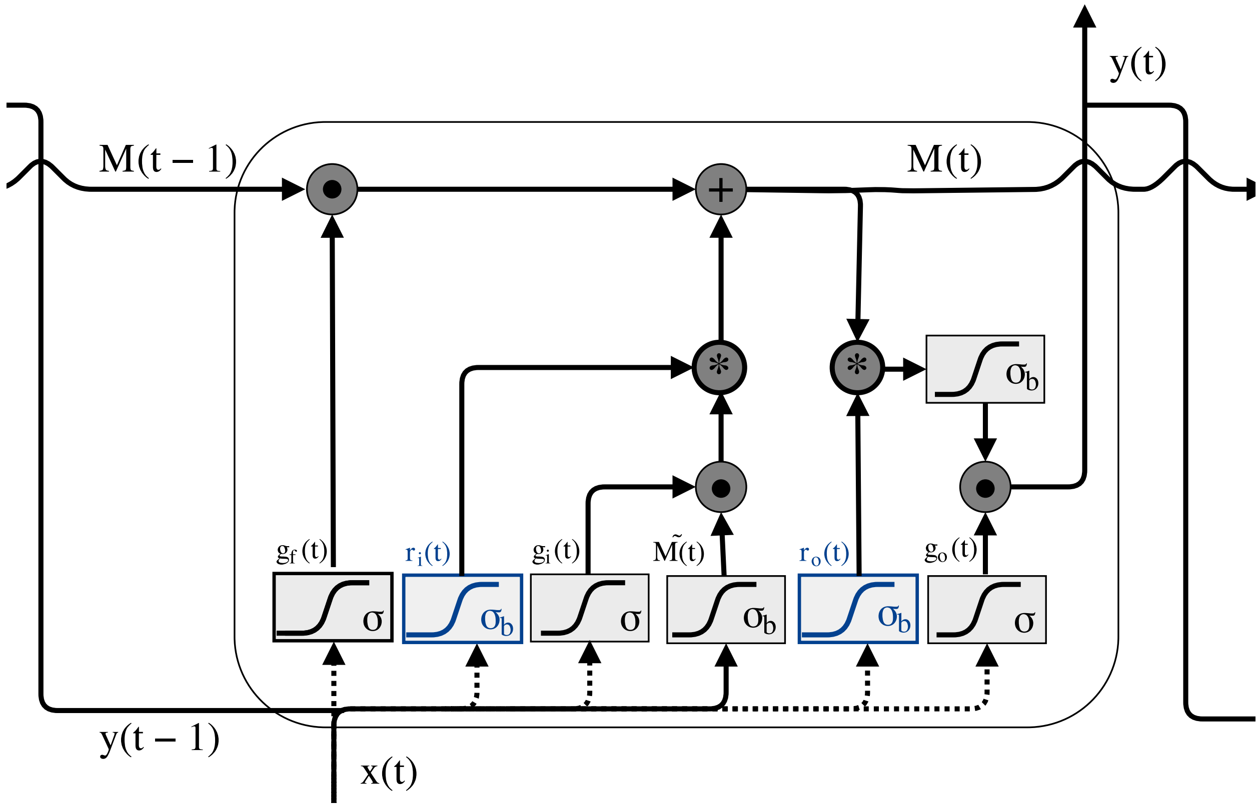
$$g_\star(t) = \begin{bmatrix} \sigma[\hat{g}_\star(t)] \\ \sigma[\hat{g}_\star(t)] \end{bmatrix} \qquad \text{for } \star \in \{f,i,o\}$$



Associative LSTM [Danihelka & Wayne^{+} 16]

Keys:

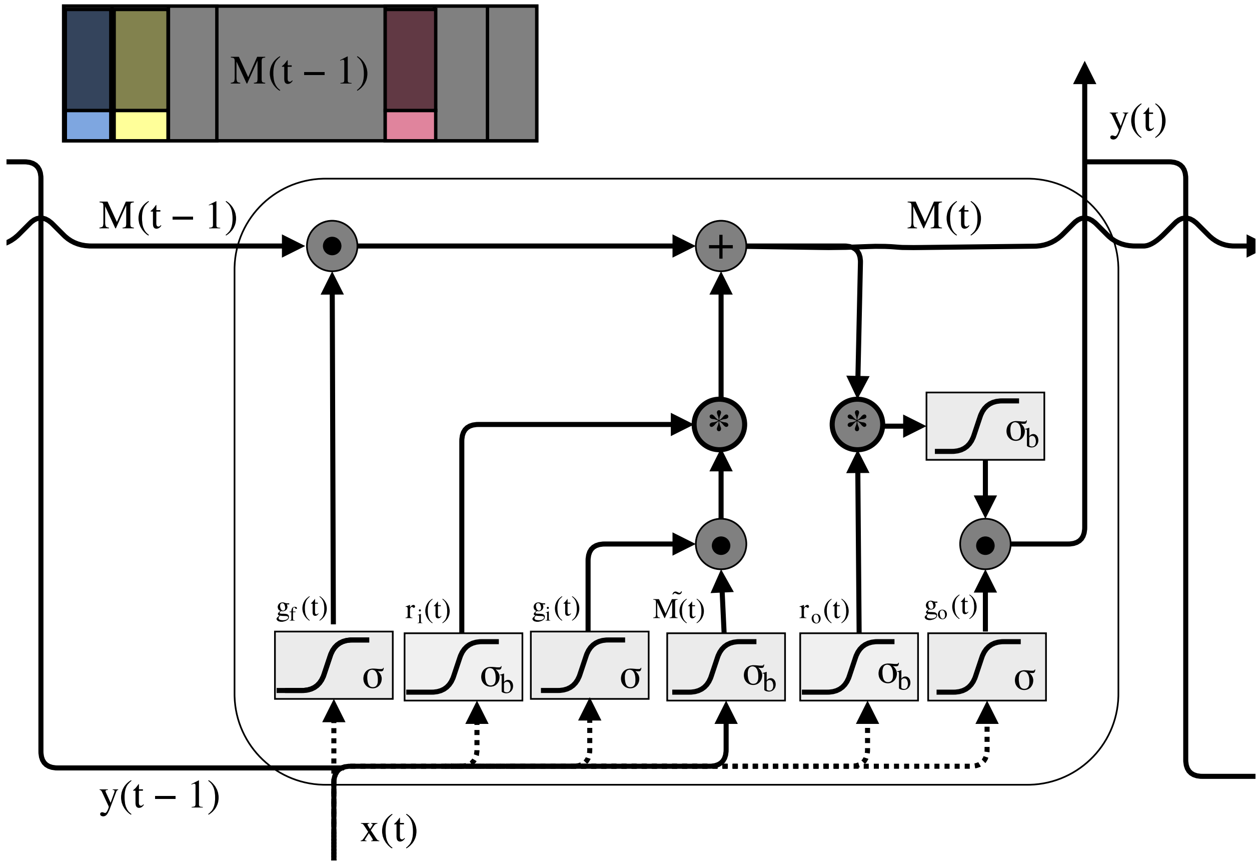
$$\hat{r}_\star(t) = W_{x\star}x(t) + W_{y\star}y(t-1) + b_\star \qquad \text{for } \star \in \{i,o\}$$
$$r_\star(t) = \sigma_{\text{bound}}[\hat{r}_\star(t)] \qquad \text{for } \star \in \{i,o\}$$



Associative LSTM [Danihelka & Wayne+ 16]

Memory update:

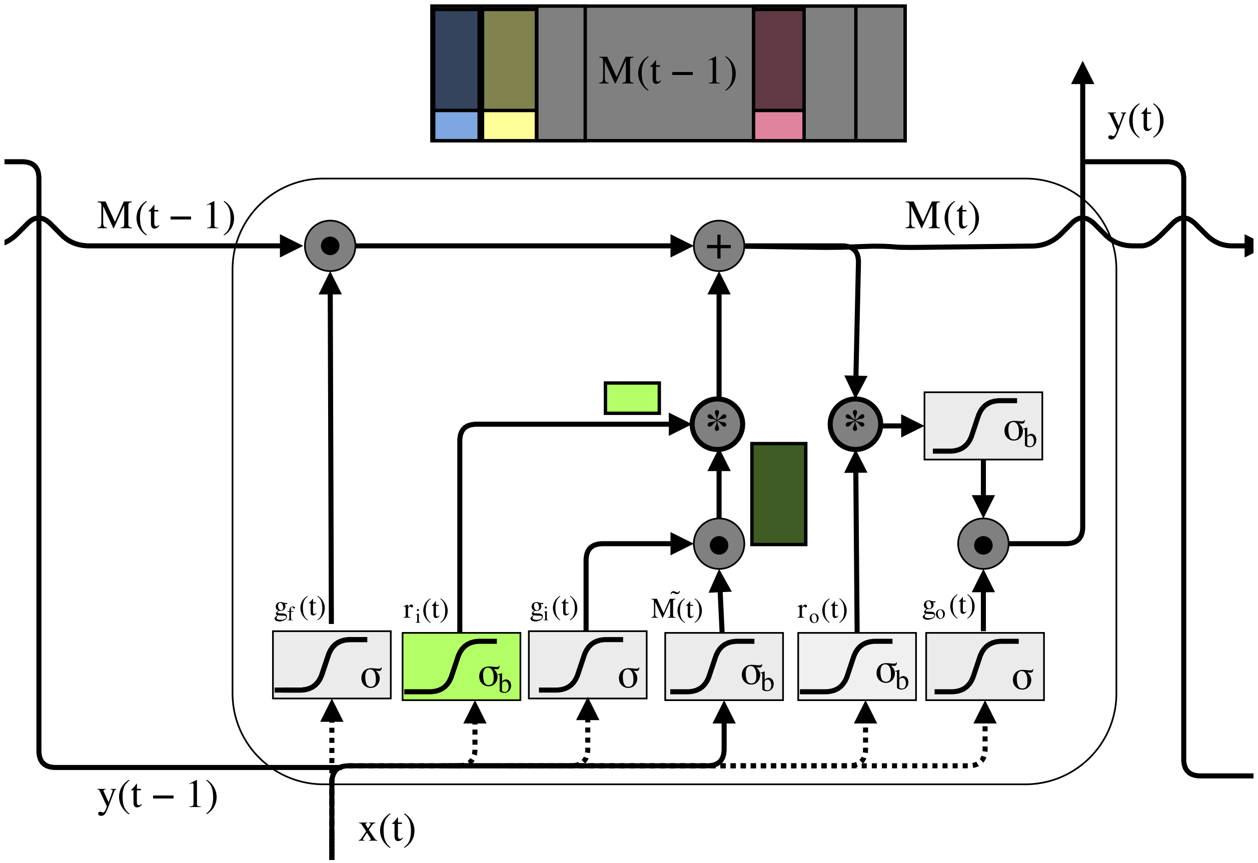
$$\tilde{M}(t) = \sigma_{\text{bound}}(W_{xM}x(t) + W_{yM}y(t-1) + b_M)$$
$$M_s(t) = g_f(t) \odot M_s(t-1) + r_{i,s}(t) \circledast [g_i(t) \odot \tilde{M}(t)]$$



Associative LSTM [Danihelka & Wayne^{+} 16]

Memory update:

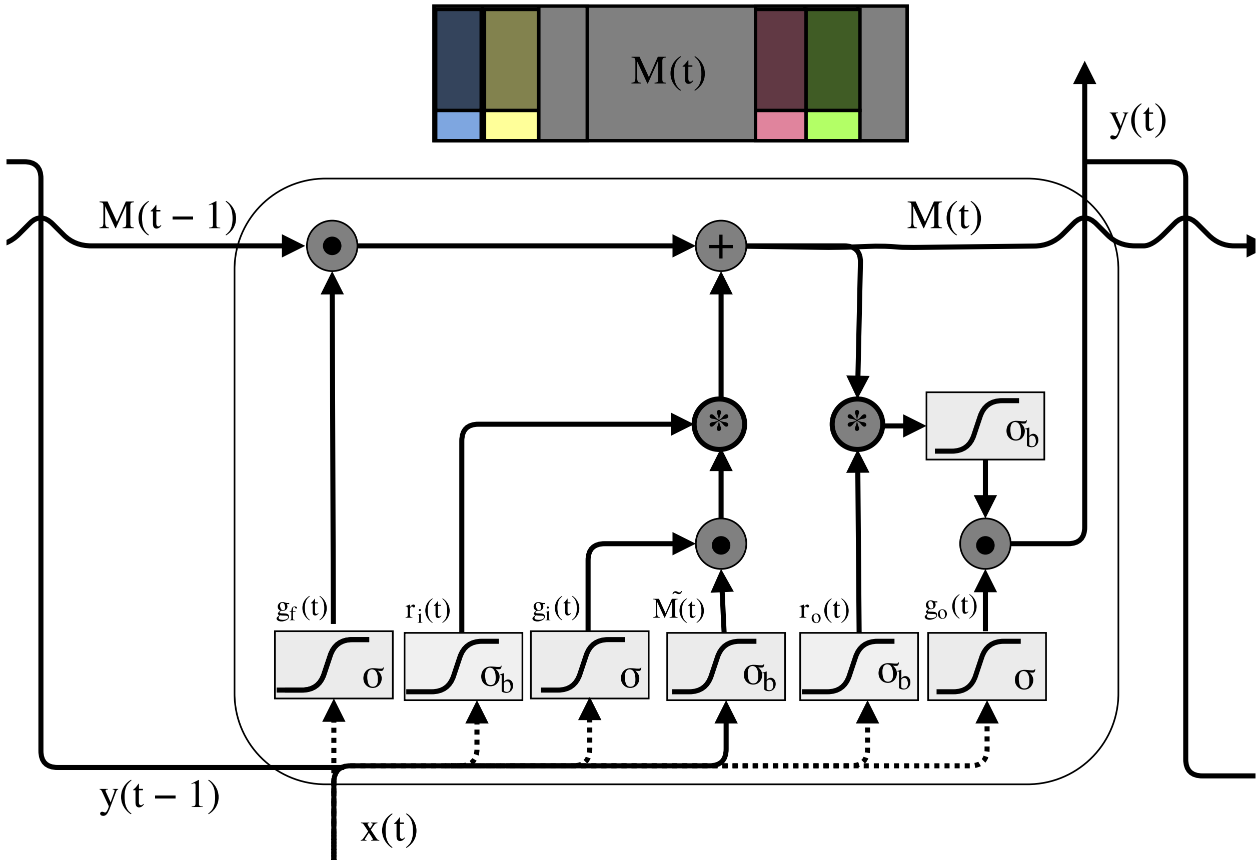
$$\tilde{M}(t) = \sigma_{\text{bound}}(W_{xM}x(t) + W_{yM}y(t-1) + b_M)$$
$$M_s(t) = g_f(t) \odot M_s(t-1) + r_{i,s}(t) \circledast [g_i(t) \odot \tilde{M}(t)]$$



Associative LSTM [Danihelka & Wayne$^+$ 16]

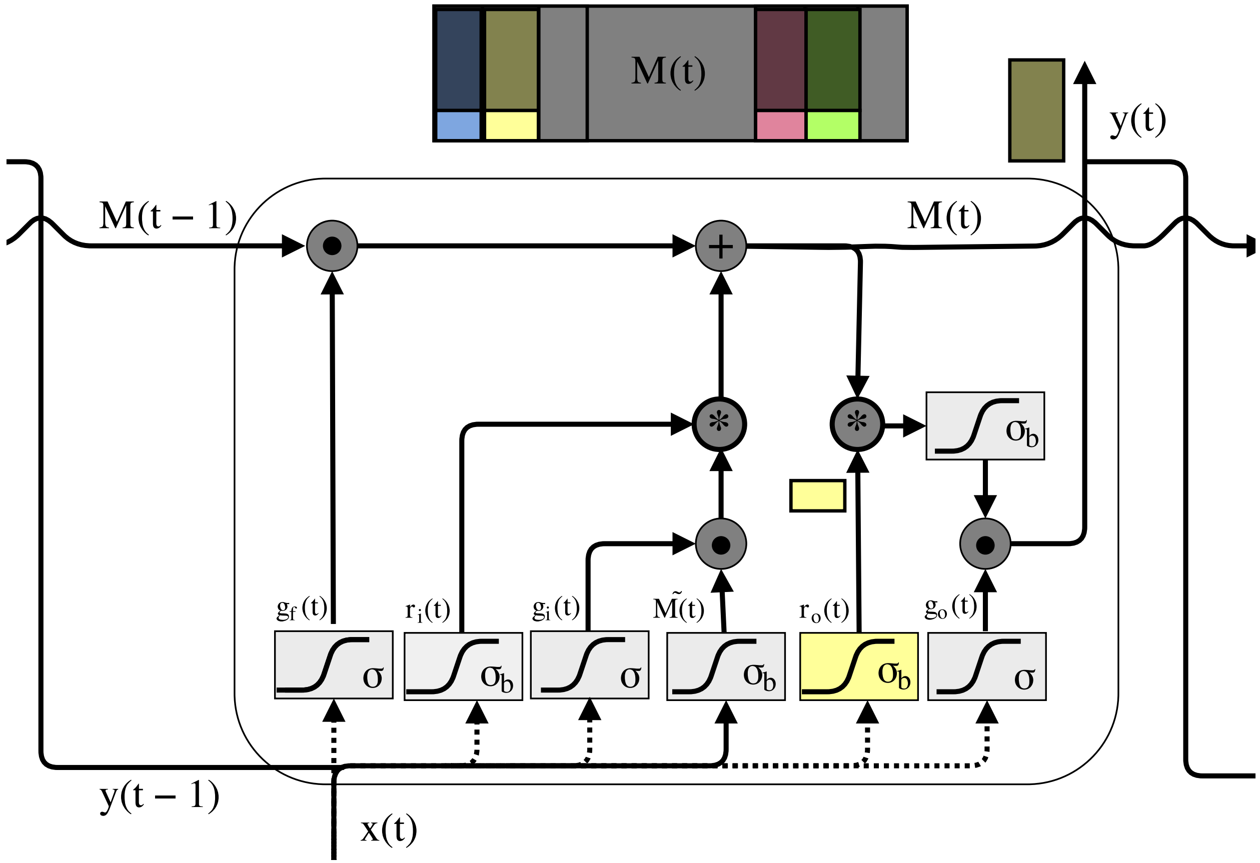
Memory update:

$$\tilde{M}(t) = \sigma_{\text{bound}}(W_{xM}x(t) + W_{yM}y(t-1) + b_M)$$
$$M_s(t) = g_f(t) \odot M_s(t-1) + r_{i,s}(t) \circledast [g_i(t) \odot \tilde{M}(t)]$$



Associative LSTM [Danihelka & Wayne+ 16]

Output: $$y(t) = g_o(t) \odot \sigma_{\text{bound}} \left(\frac{1}{N_{\text{copies}}} \sum_{s=1}^{N_{\text{copies}}} r_{o,s}(t) \circledast M_s(t) \right)$$



Computational Hierarchy

Turing Machines (2 Stacks/ Tape)
$\rightarrow$ computable functions

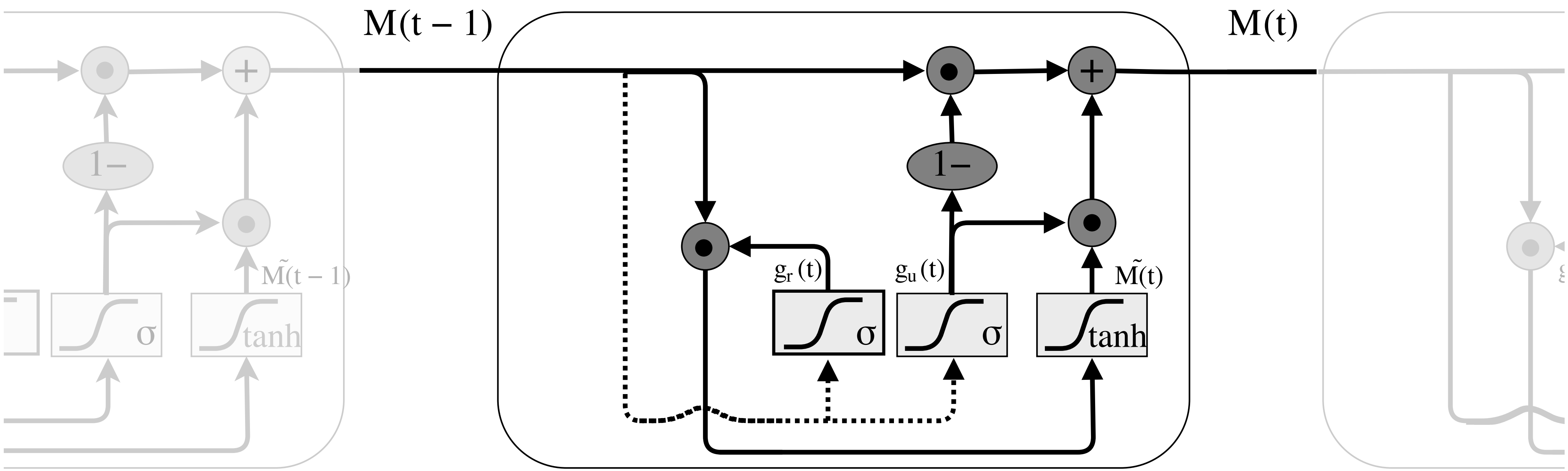
$\Uparrow \Uparrow \Uparrow$

Pushdown Automata (1 Stack)
$\rightarrow$ context free languages

$\Uparrow \Uparrow \Uparrow$

Finite State Machines (0 Stacks)
$\rightarrow$ regular languages

Neural GPU [Kaiser & Sutskever 15]

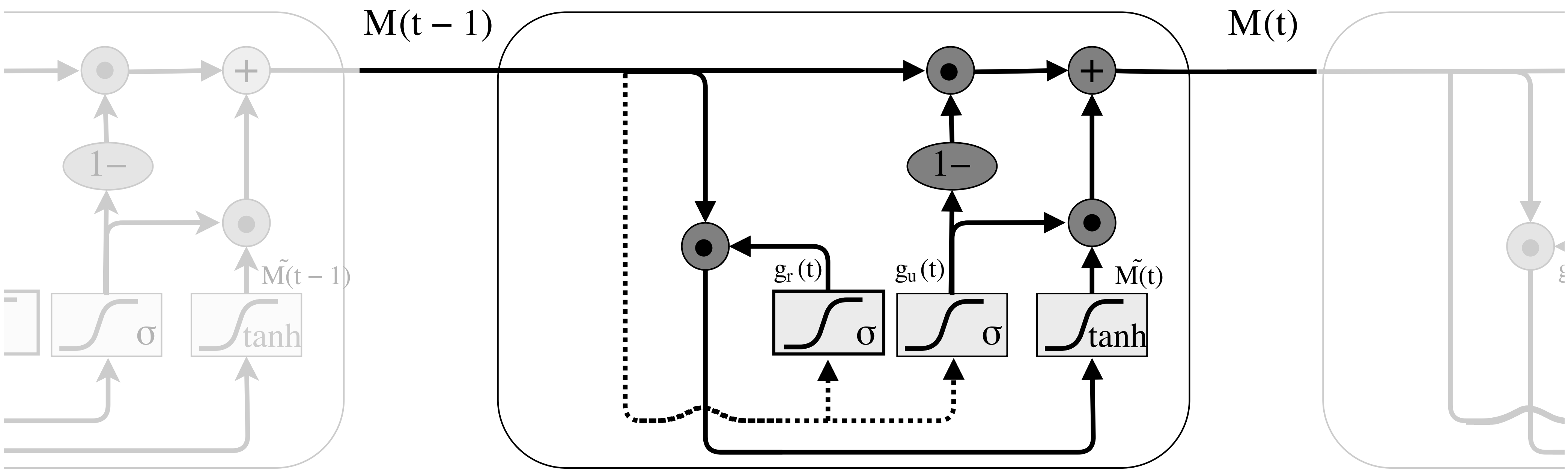


GRU unfolded over time

Neural GPU [Kaiser & Sutskever 15]

Memory Access:

$$\tilde{M}(t) = \sigma_{\tanh}(W_M * [g_r(t) \odot M(t)] + B_M)$$
$$M(t) = g_u(t) \odot \tilde{M}(t) + [1 - g_u(t)] \odot M(t-1)$$

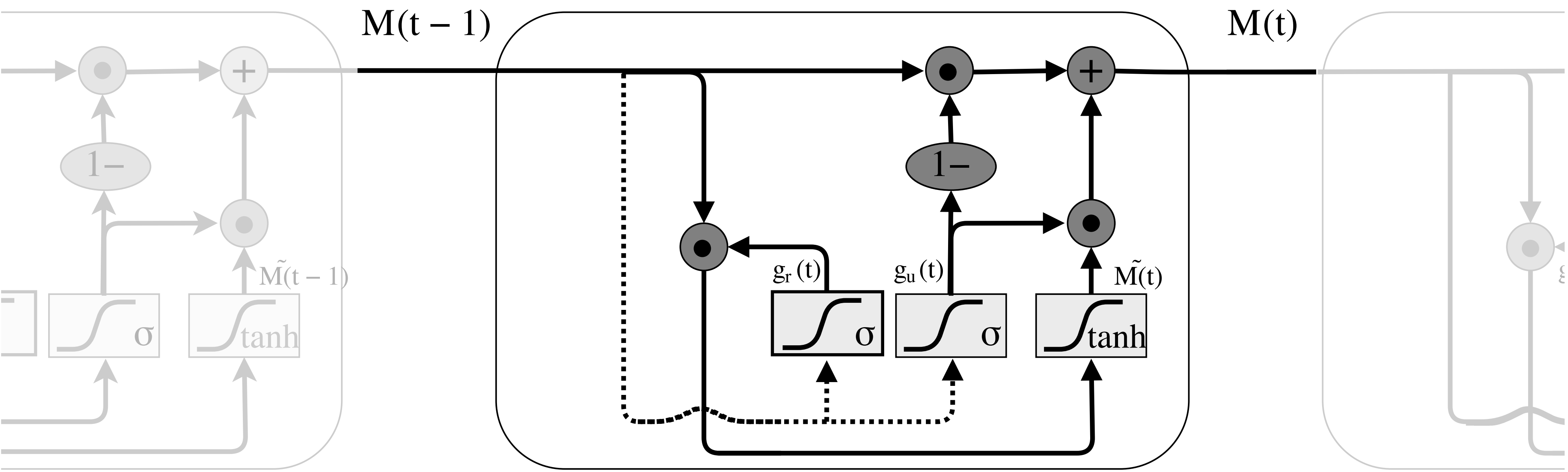


Convolutional GRU unfolded over time

Neural GPU [Kaiser & Sutskever 15]

Gating:

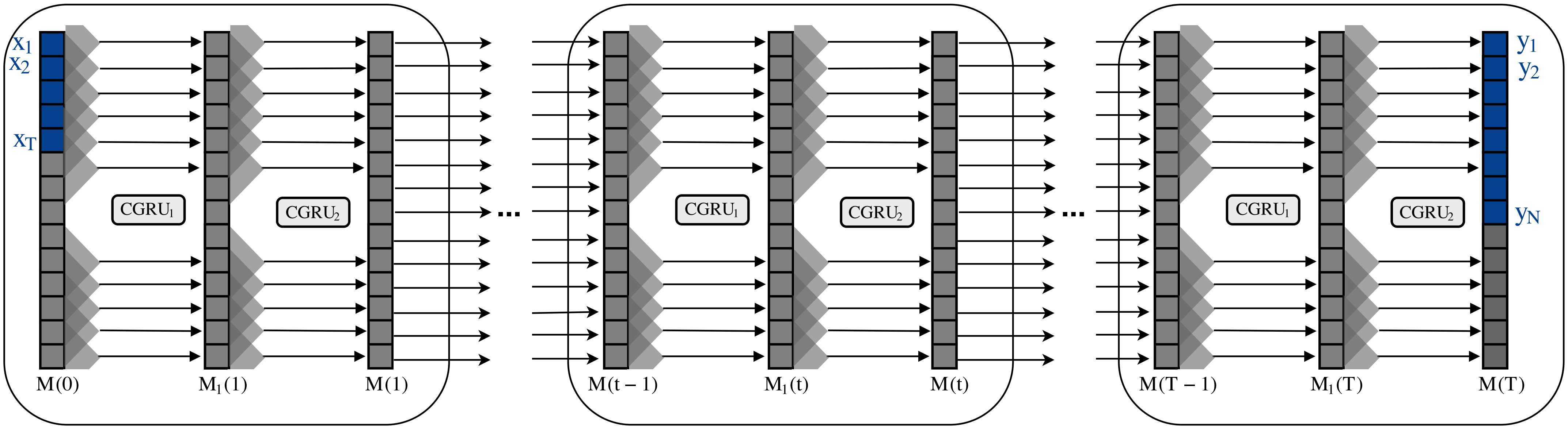
$$g_u(t) = \sigma(W_u * M(t) + B_u)$$
$$g_r(t) = \sigma(W_r * M(t) + B_r)$$



Convolutional GRU unfolded over time

Neural GPU [Kaiser & Sutskever 15]

- Apply multiple CGRUs in succession in every computation step
- Input written in the initial state $M(0)$
- Result can be extracted from $M(T)$



Outline

Introduction

Augmenting RNNs with Memory

Results

- Arithmetic Tasks
- Question Answering Tasks
- Machine Translation

Conclusion and Discussion

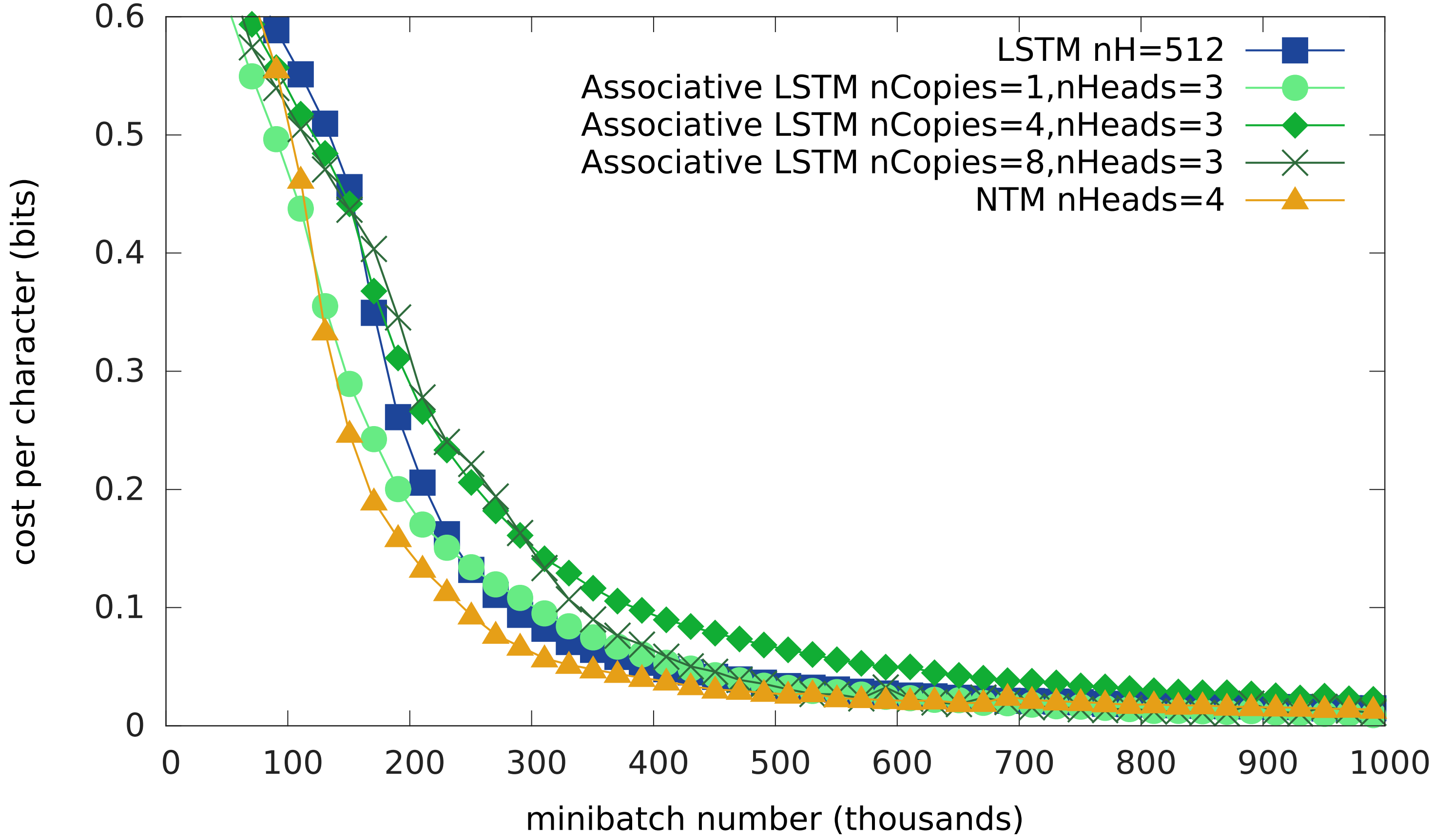
Binary Arithmetic [Kaiser & Sutskever 15]

Task	Bits	Neural GPU	Stack RNN	LSTM + Attention
addition	20	100%	100%	100%
	25	100%	100%	73%
	100	100%	88%	0%
	200	100%	0%	0%
	2000	100%	0%	0%
multiplication	20	100%	N/A	0%
	25	100%	N/A	0%
	100	100%	N/A	0%
	200	100%	N/A	0%
	2000	100%	N/A	0%

- All models trained on numbers of up to 20 bit length
- Percentage of test cases with perfect result (no bit error)

Arithmetic [Danihelka & Wayne+ 16]

- **Addition and Subtraction on decimal numbers**



Model	# Parameters
LSTM	1.26
Associative LSTM	0.78
NTM	1.10

Question Answering Tasks [Gulcehre & Chandar$^+$ 16]

Facebook bAbI QA task [Weston & Bordes$^+$ 15]

- **20 different sub-tasks**
- **Demands: chaining facts, simple induction, deduction, ...**

Example:

```
1 Mary moved to the bathroom.
2 John went to the hallway.
3 Where is Mary?
```

$\Rightarrow$ `Answer: bathroom`

Question Answering Tasks [Gulcehre & Chandar+ 16]

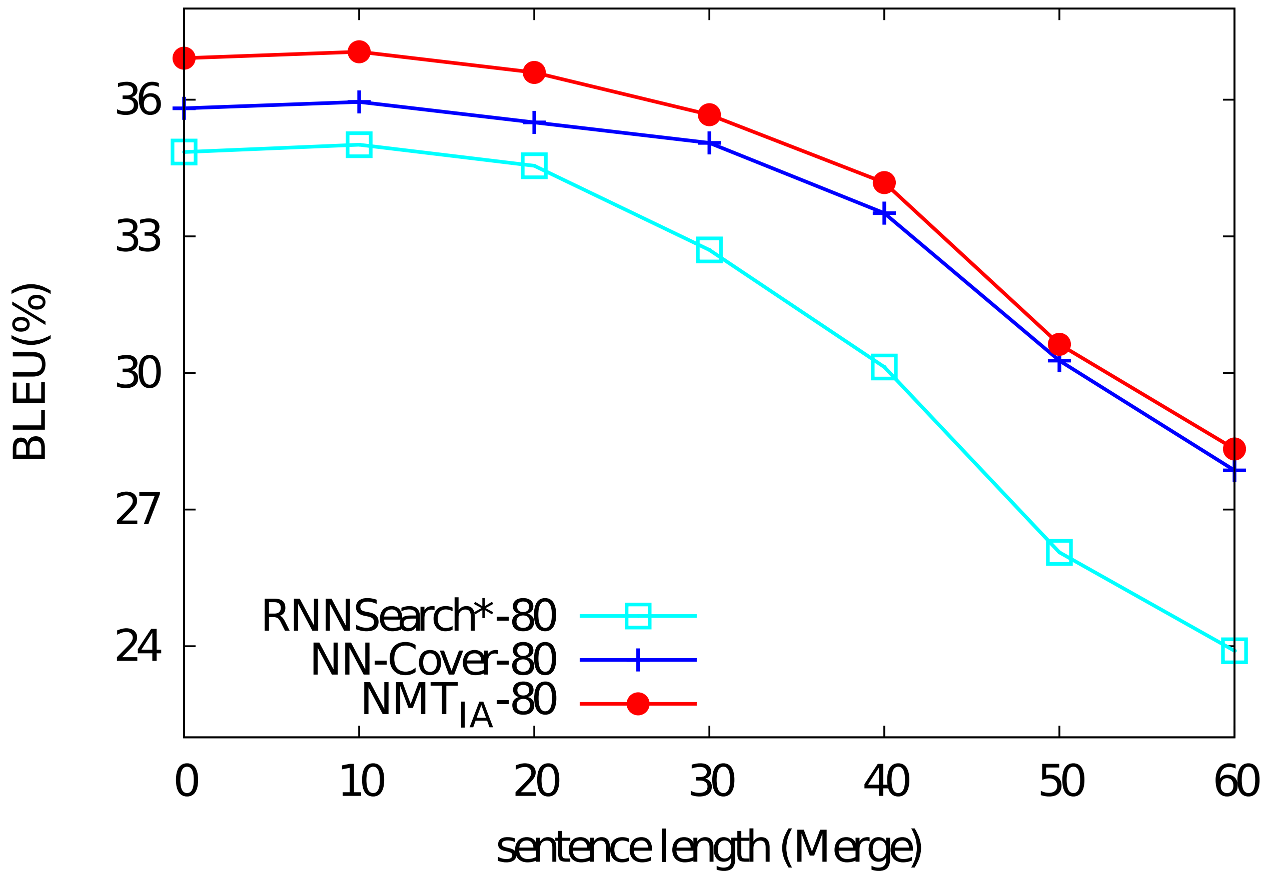
Task	Description	LSTM	MemN2N	NTM	D-NTM
1	1 Supporting Fact	0.00	0.00	16.30	6.66
2	2 Supporting Facts	81.90	0.30	57.08	56.04
3	3 Supporting Facts	83.10	2.10	74.16	72.08
4	2 Argument Relations	0.20	0.00	0.00	0.00
5	3 Argument Relations	1.20	0.80	1.46	1.04
6	Yes/No Questions	51.80	0.10	23.33	44.79
7	Counting	24.90	2.00	21.67	19.58
8	Lists/Sets	34.10	0.90	25.76	18.46
9	Simple Negation	20.20	0.30	24.79	34.37
10	Indefinite Knowledge	30.10	0.00	41.46	50.83
11	Basic Coreference	10.30	0.10	18.96	4.16
12	Conjunction	23.40	0.00	25.83	6.66
13	Compound Coreference	6.10	0.00	6.67	2.29
14	Time Manipulation	81.00	0.10	58.54	63.75
15	Basic Deduction	78.70	0.00	36.46	39.27
16	Basic Induction	51.90	51.80	71.15	51.35
17	Positional Reasoning	50.10	18.60	43.75	16.04
18	Reasoning About Size	6.80	5.30	3.96	3.54
19	Path Finding	90.30	2.30	75.89	64.63
20	Reasoning About Motivation	2.10	0.00	1.25	3.12
Avg.Err.		36.41	4.24	31.42	27.93

Machine Translation

Different approaches to neural machine translation (NMT)

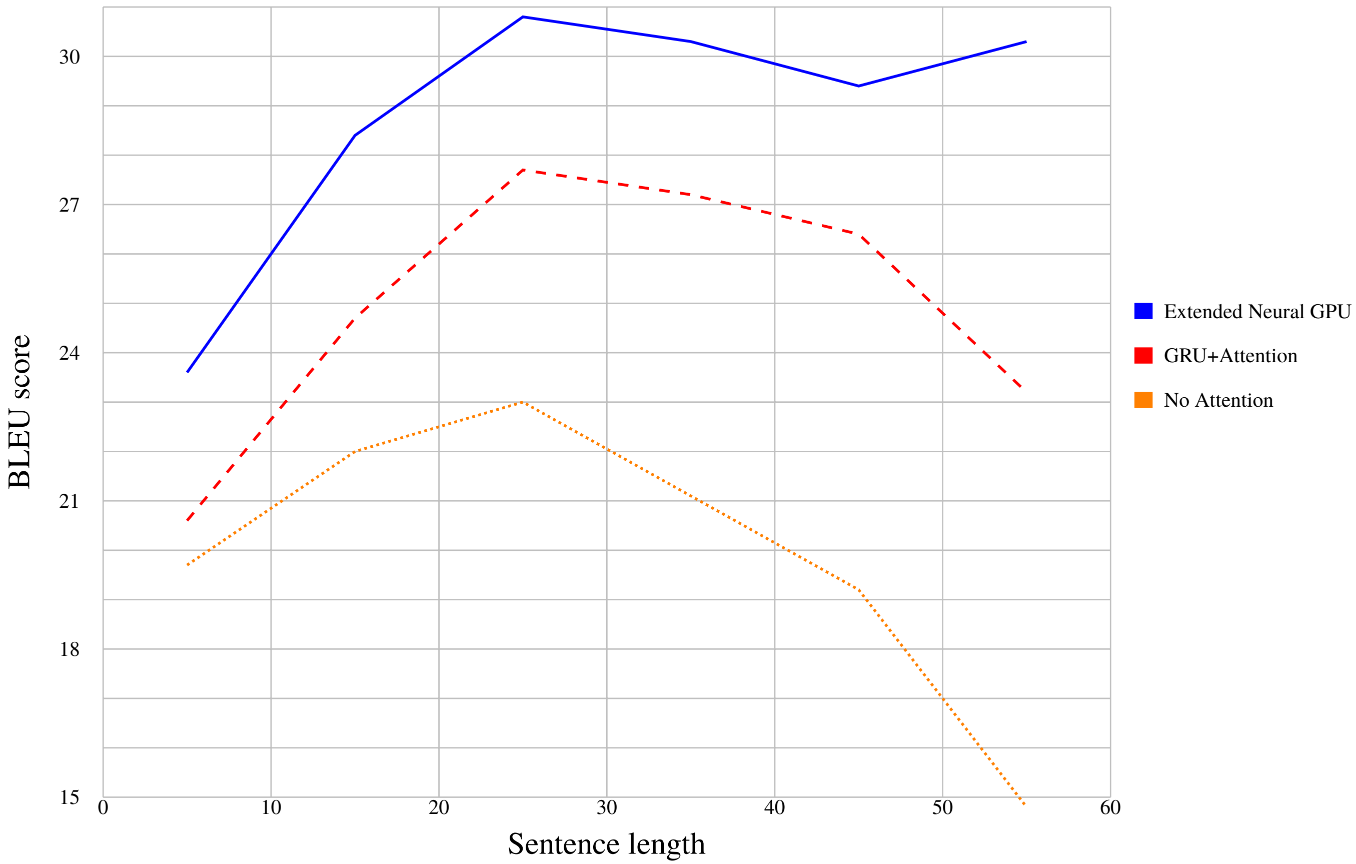
- Attention used in state-of-the-art NMT [Bahdanau & Cho^{+} 15].
- Replace standard content-based read operation by read and write operations of NTMs [Wang & Lu^{+} 16, Meng & Lu^{+} 16, Meng & Lu^{+} 15].
- Use (extended) neural GPU to compute translation [Kaiser & Bengio 16]

Machine Translation Results: NTM [Meng & Lu^{+} 16]



LDC Zh → En Model	BLEU				
	MT03	MT04	MT05	MT06	Average
NMT + NTM	35.1	37.7	35.5	34.3	35.7
Attention-based NMT	33.4	36.0	33.6	32.2	33.8

Machine Translation Results: Neural GPU [Kaiser & Bengio 16]



Model	WMT En → Fr Perplexity (log)	WMT En → Fr BLEU
Neural GPU	$30.1(3.5)$	< 5
Extended Neural GPU	$3.3(1.19)$	29.6
Attention-based NMT	$3.4(1.22)$	26.4

Outline

Introduction

Augmenting RNNs with Memory

Results

Conclusion and Discussion

Conclusion and Discussion

- RNNs are theoretically Turing complete
- Without extensions not successful on many algorithmic tasks in practice

Read Memory Extensions *($\Leftrightarrow$ Finite-State Machine)*:

- MemN2N and attention show great results in specific applications:
  - for problems with long inputs and non-monotonic access patterns
  - e.g. NMT and question answering
  - although English is no finite-state language [Chomsky & Halle^{+} 56]

Conclusion and Discussion

Read-Write Memory Extensions (⇔ *Turing Machine*):

- **NTM and stack RNN:**
  - **flexible addressing through attention (focused on one position)**
- **neural GPU:**
  - **active memory through convolution (modifies all positions equally)**
- **associative LSTM:**
  - **key-value access (modifies only entry associated with key)**

Remaining problems:

- **Number of computation steps needs to be set in advance**
  - **Solution: adaptive computation time [Graves 16]**
- **Memory size also hyperparameter that needs to be set for each task**

Thank you for your attention!

Arne Nix

`arne.nix@rwth-aachen.de`

Backup: Examples for bAbI Tasks

Task 1: Single Supporting Fact
Mary went to the bathroom.
John moved to the hallway.
Mary travelled to the office.
Where is Mary? A:office

Task 2: Two Supporting Facts
John is in the playground.
John picked up the football.
Bob went to the kitchen.
Where is the football? A:playground

Task 3: Three Supporting Facts
John picked up the apple.
John went to the office.
John went to the kitchen.
John dropped the apple.
Where was the apple before the kitchen? A:office

Task 4: Two Argument Relations
The office is north of the bedroom.
The bedroom is north of the bathroom.
The kitchen is west of the garden.
What is north of the bedroom? A: office
What is the bedroom north of? A: bathroom

Task 5: Three Argument Relations
Mary gave the cake to Fred.
Fred gave the cake to Bill.
Jeff was given the milk by Bill.
Who gave the cake to Fred? A: Mary
Who did Fred give the cake to? A: Bill

Task 6: Yes/No Questions
John moved to the playground.
Daniel went to the bathroom.
John went back to the hallway.
Is John in the playground? A:no
Is Daniel in the bathroom? A:yes

Task 7: Counting
Daniel picked up the football.
Daniel dropped the football.
Daniel got the milk.
Daniel took the apple.
How many objects is Daniel holding? A: two

Task 8: Lists/Sets
Daniel picks up the football.
Daniel drops the newspaper.
Daniel picks up the milk.
John took the apple.
What is Daniel holding? milk, football

Task 9: Simple Negation
Sandra travelled to the office.
Fred is no longer in the office.
Is Fred in the office? A:no
Is Sandra in the office? A:yes

Task 10: Indefinite Knowledge
John is either in the classroom or the playground.
Sandra is in the garden.
Is John in the classroom? A:maybe
Is John in the office? A:no

Backup: Examples for bAbI Tasks

Task 11: Basic Coreference Daniel was in the kitchen. Then he went to the studio. Sandra was in the office. Where is Daniel? A:studio	**Task 12: Conjunction** Mary and Jeff went to the kitchen. Then Jeff went to the park. Where is Mary? A: kitchen Where is Jeff? A: park
Task 13: Compound Coreference Daniel and Sandra journeyed to the office. Then they went to the garden. Sandra and John travelled to the kitchen. After that they moved to the hallway. Where is Daniel? A: garden	**Task 14: Time Reasoning** In the afternoon Julie went to the park. Yesterday Julie was at school. Julie went to the cinema this evening. Where did Julie go after the park? A:cinema Where was Julie before the park? A:school
Task 15: Basic Deduction Sheep are afraid of wolves. Cats are afraid of dogs. Mice are afraid of cats. Gertrude is a sheep. What is Gertrude afraid of? A:wolves	**Task 16: Basic Induction** Lily is a swan. Lily is white. Bernhard is green. Greg is a swan. What color is Greg? A:white
Task 17: Positional Reasoning The triangle is to the right of the blue square. The red square is on top of the blue square. The red sphere is to the right of the blue square. Is the red sphere to the right of the blue square? A:yes Is the red square to the left of the triangle? A:yes	**Task 18: Size Reasoning** The football fits in the suitcase. The suitcase fits in the cupboard. The box is smaller than the football. Will the box fit in the suitcase? A:yes Will the cupboard fit in the box? A:no
Task 19: Path Finding The kitchen is north of the hallway. The bathroom is west of the bedroom. The den is east of the hallway. The office is south of the bedroom. How do you go from den to kitchen? A: west, north How do you go from office to bathroom? A: north, west	**Task 20: Agent's Motivations** John is hungry. John goes to the kitchen. John grabbed the apple there. Daniel is hungry. Where does Daniel go? A:kitchen Why did John go to the kitchen? A:hungry

Backup: Holographic Reduced Representations

- **Use circular convolution to associate vectors $\rightarrow$ implemented through complex representation**
  - **Associative Array:**

$$\boldsymbol{c} = \boldsymbol{r}_1 \circledast \boldsymbol{x}_1 + \boldsymbol{r}_2 \circledast \boldsymbol{x}_2 + \boldsymbol{r}_3 \circledast \boldsymbol{x}_3$$

  - **Lookup:**

$$\begin{aligned} \boldsymbol{r}_2^{-1} \circledast \boldsymbol{c} &= \boldsymbol{r}_2^{-1} \circledast (\boldsymbol{r}_1 \circledast \boldsymbol{x}_1 + \boldsymbol{r}_2 \circledast \boldsymbol{x}_2 + \boldsymbol{r}_3 \circledast \boldsymbol{x}_3) \\ &= \boldsymbol{x}_2 + \boldsymbol{r}_2^{-1} \circledast (\boldsymbol{r}_1 \circledast \boldsymbol{x}_1 + \boldsymbol{r}_3 \circledast \boldsymbol{x}_3) \\ &= x_2 + \textbf{\textit{noise}} \end{aligned}$$

- D. Bahdanau, K. Cho, Y. Bengio:
  Neural machine translation by jointly learning to align and translate.
  Proc. *ICLR*, 2015.
- N. Chomsky, M. Halle, F. Lukoff:
  On accent and juncture in English.
  For Roman Jakobson, Vol., pp. 65–80, 1956.
- I. Danihelka, G. Wayne, B. Uria, N. Kalchbrenner, A. Graves:
  Associative Long Short-Term Memory.
  Proc. *The 33rd International Conference on Machine Learning*, pp. 1986–1994, 2016.
- F.A. Gers, J. Schmidhuber, F. Cummins:
  Learning to forget: Continual prediction with LSTM.
  Neural computation, Vol. 12, No. 10, pp. 2451–2471, 2000.
- A. Graves:
  Generating sequences with recurrent neural networks.
  arXiv preprint arXiv:1308.0850, Vol., August 2013.

- **A. Graves:**
  Adaptive computation time for recurrent neural networks.
  arXiv preprint arXiv:1603.08983, Vol., 2016.

- **A. Graves, G. Wayne, I. Danihelka:**
  Neural turing machines.
  arXiv preprint arXiv:1410.5401, Vol., 2014.

- **A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S.G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou et al.:**
  Hybrid computing using a neural network with dynamic external memory.
  Nature, Vol. 538, No. 7626, pp. 471–476, 2016.

- **C. Gulcehre, S. Chandar, K. Cho, Y. Bengio:**
  Dynamic Neural Turing Machine with Soft and Hard Addressing Schemes.
  arXiv preprint arXiv:1607.00036, Vol., 2016.

- **S. Hochreiter, J. Schmidhuber:**
  Long short-term memory.
  Neural computation, Vol. 9, No. 8, pp. 1735–1780, November 1997.
- **A. Joulin, T. Mikolov:**
  Inferring algorithmic patterns with stack-augmented recurrent nets.
  Proc. *Advances in Neural Information Processing Systems*, pp. 190–198, 2015.
- **Ł. Kaiser, S. Bengio:**
  Can Active Memory Replace Attention?
  Proc. *Advances In Neural Information Processing Systems*, pp. 3774–3782, 2016.
- **Ł. Kaiser, I. Sutskever:**
  Neural gpus learn algorithms.
  arXiv preprint arXiv:1511.08228, Vol., 2015.

- **J.F. Kolen:**
  Fool's gold: Extracting finite state machines from recurrent network dynamics.
  Advances in neural information processing systems, Vol., pp. 501–501, 1994.
- **F. Meng, Z. Lu, Z. Tu, H. Li, Q. Liu:**
  A Deep Memory-based Architecture for Sequence-to-Sequence Learning.
  arXiv preprint arXiv:1506.06442, Vol., 2015.
- **F. Meng, Z. Lu, H. Li, Q. Liu:**
  Interactive Attention for Neural Machine Translation.
  arXiv preprint arXiv:1610.05011, Vol., 2016.
- **T.A. Plate:**
  Holographic reduced representations.
  IEEE Transactions on Neural Networks, Vol. 6, No. 3, pp. 623–641, 1995.

- **H.T. Siegelmann, E.D. Sontag:**
  On the computational power of neural nets.
  Proc. *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 440–449. ACM, 1992.

- **S. Sukhbaatar, J. Weston, R. Fergus et al.:**
  End-to-end memory networks.
  Proc. *Advances in neural information processing systems*, pp. 2440–2448, 2015.

- **P. Tino, B.G. Horne, C.L. Giles:**
  Finite state machines and recurrent neural networks–automata and dynamical systems approaches.
  Vol., 1998.

- **M. Wang, Z. Lu, H. Li, Q. Liu:**
  Memory-enhanced Decoder for Neural Machine Translation.
  arXiv preprint arXiv:1606.02003, Vol., 2016.

- **J. Weston, A. Bordes, S. Chopra, A.M. Rush, B. van Merriënboer, A. Joulin, T. Mikolov:**
  Towards ai-complete question answering: A set of prerequisite toy tasks.
  arXiv preprint arXiv:1502.05698, Vol., 2015.
- **J. Weston, S. Chopra, A. Bordes:**
  Memory networks.
  arXiv preprint arXiv:1410.3916, Vol., 2014.