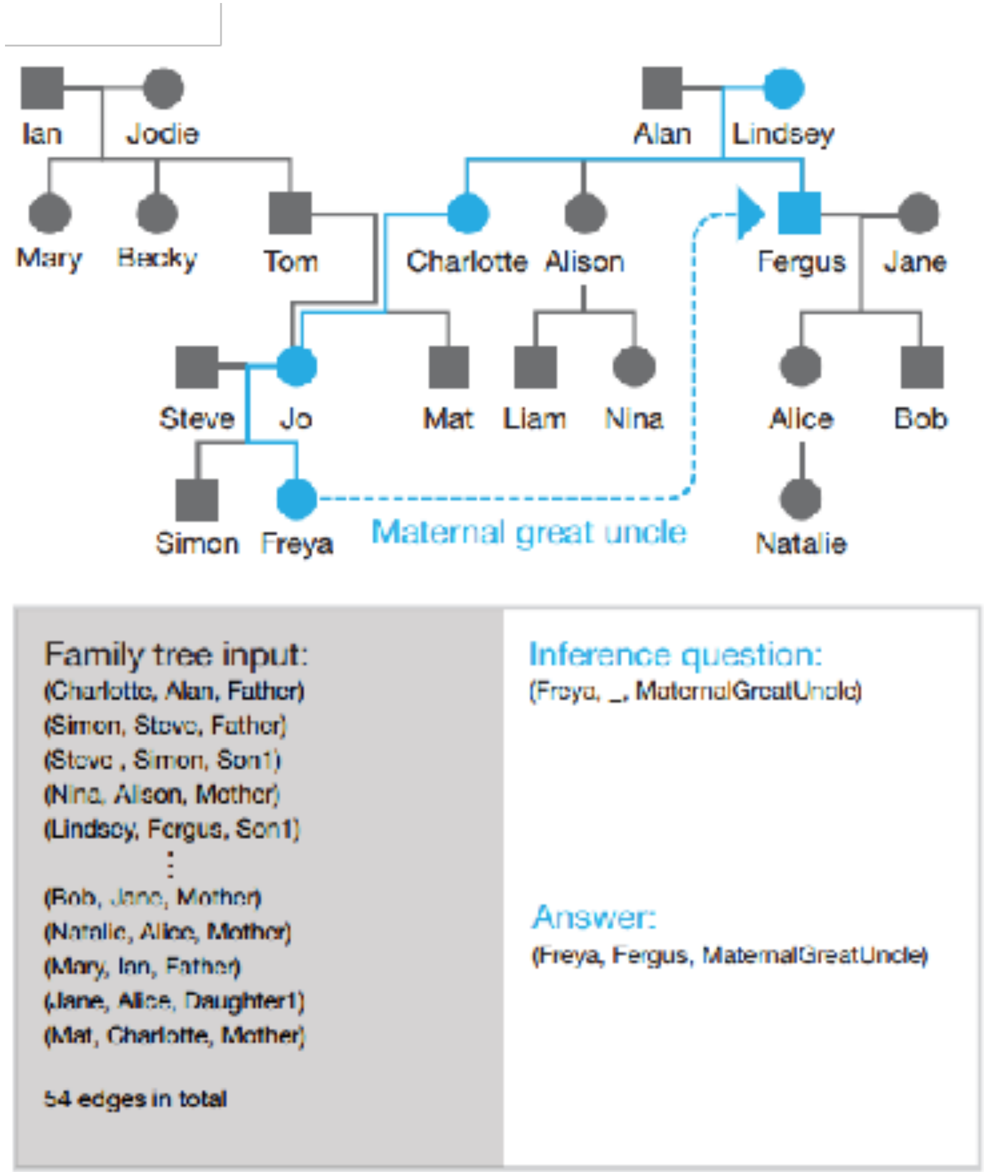
Differentiable Neural Computers (LSTM 2.0)

Itamar Ben-Ari
Intel, Advanced Analytics

Family tree example



DeepMind

Differentiable Neural Computer
Family tree inference task
(artistic rendering)

bAbI 20 tasks

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

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

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?

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?
What is the bedroom north of?

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?
Who did Fred give the cake to?

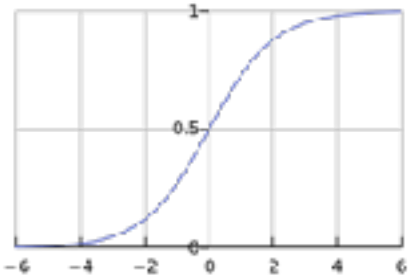
Task 6: Yes/No Questions
John moved to the playground.
Daniel went to the bathroom.
John went back to the hallway.
Is John in the playground?
Is Daniel in the bathroom?

Task	LSTM (Joint)	DNC (Joint)
1: 1 supporting fact	24.5	**0.0**
2: 2 supporting facts	53.2	0.4
3: 3 supporting facts	48.3	**1.8**
4: 2 argument rels.	0.4	**0.0**
5: 3 argument rels.	3.5	0.8
6: yes/no questions	11.5	**0.0**
7: counting	15.0	0.6
8: lists/sets	16.5	0.3
9: simple negation	10.5	0.2
10: indefinite knowl.	22.9	0.2
11: basic coreference	6.1	**0.0**
12: conjunction	3.8	**0.0**
13: compound coref.	0.5	0.1
14: time reasoning	55.3	0.4
15: basic deduction	44.7	**0.0**
16: basic induction	52.6	55.1
17: positional reas.	39.2	**12.0**
18: size reasoning	4.8	**0.8**
19: path finding	89.5	3.9
20: agent motiv.	1.3	**0.0**
Mean Err. (%)	25.2	**3.8**
Failed (err. > 5%)	15	**2**

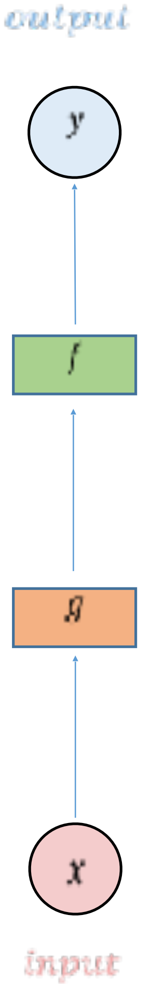
Overview of

DNN, RNN and LSTM

A ddd



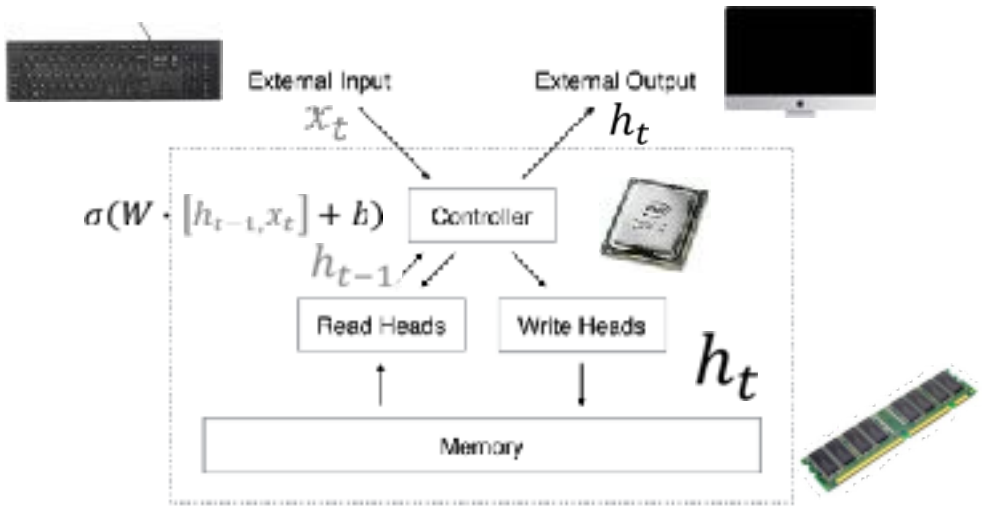
$\sigma: sigmoid\ function$

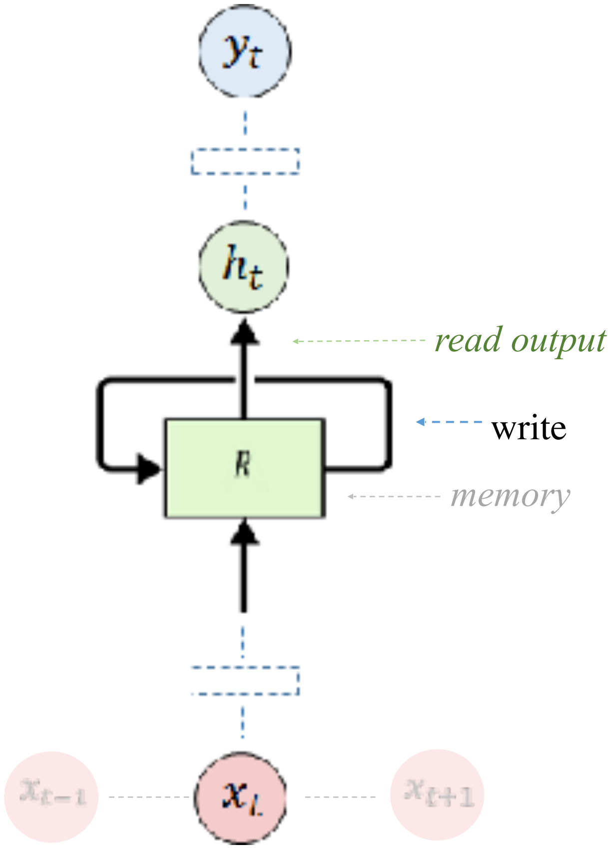


Fully connected function:

$$y_t = f(g(x)) = \sigma(W \cdot x + b)$$

Basic RNN

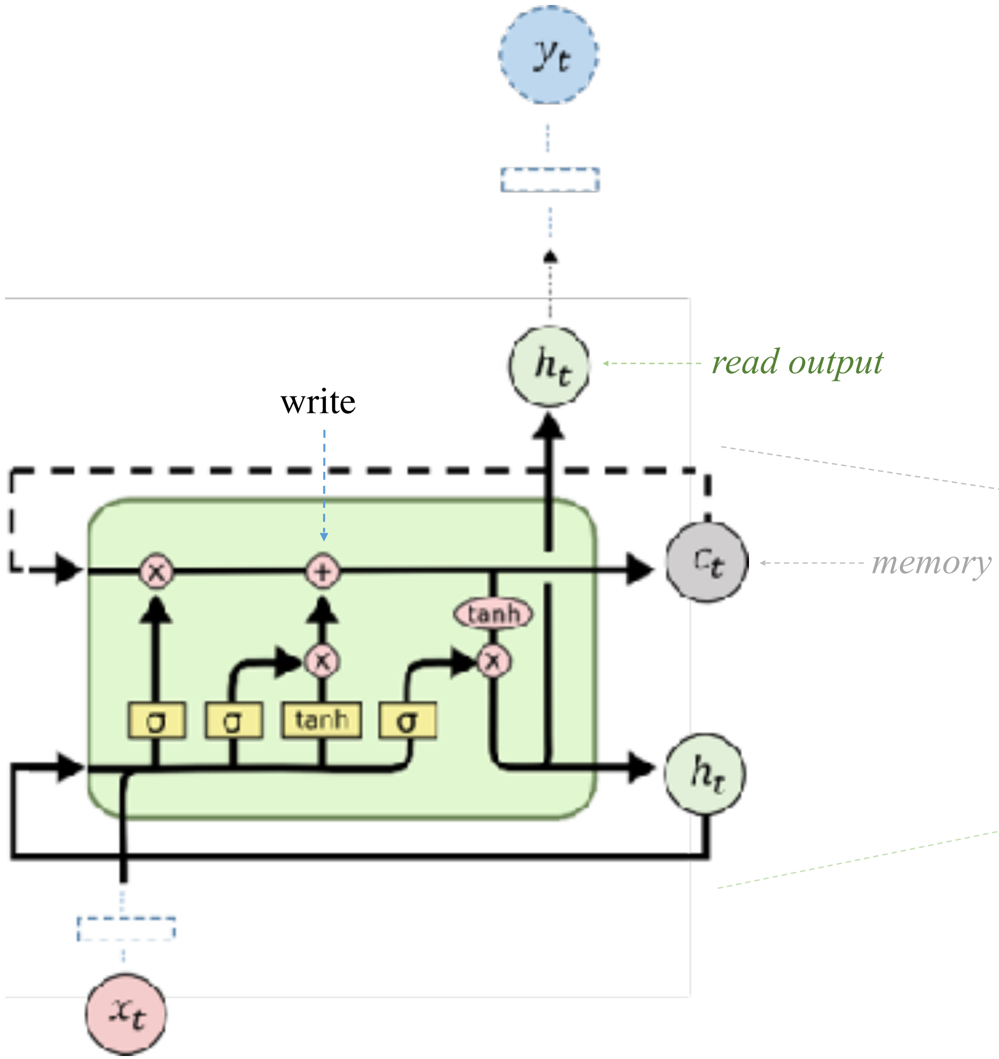


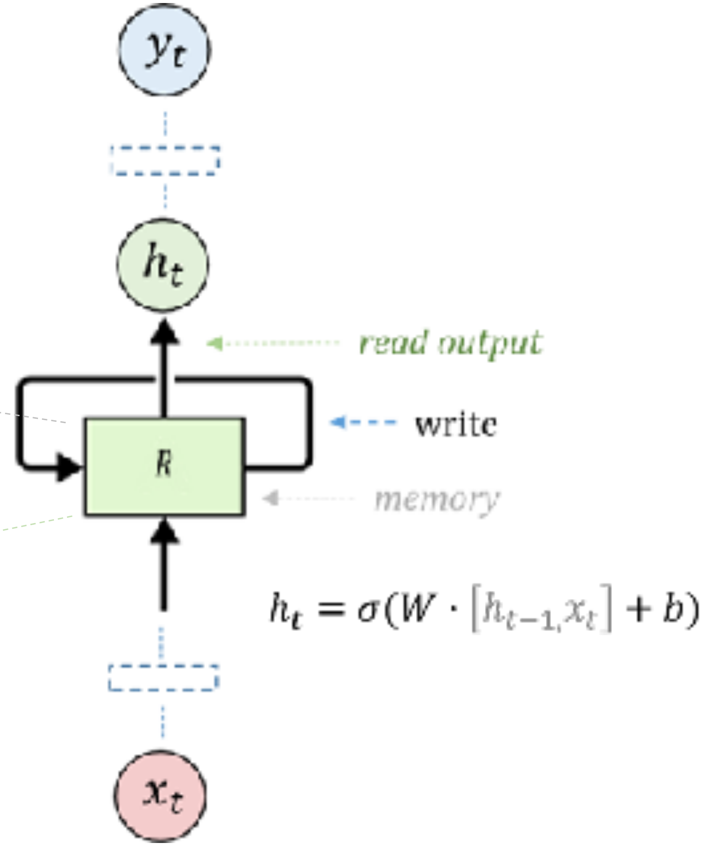


$$h_t = \sigma(W \cdot \boxed{[h_{t-1,}x_t]} + b)$$

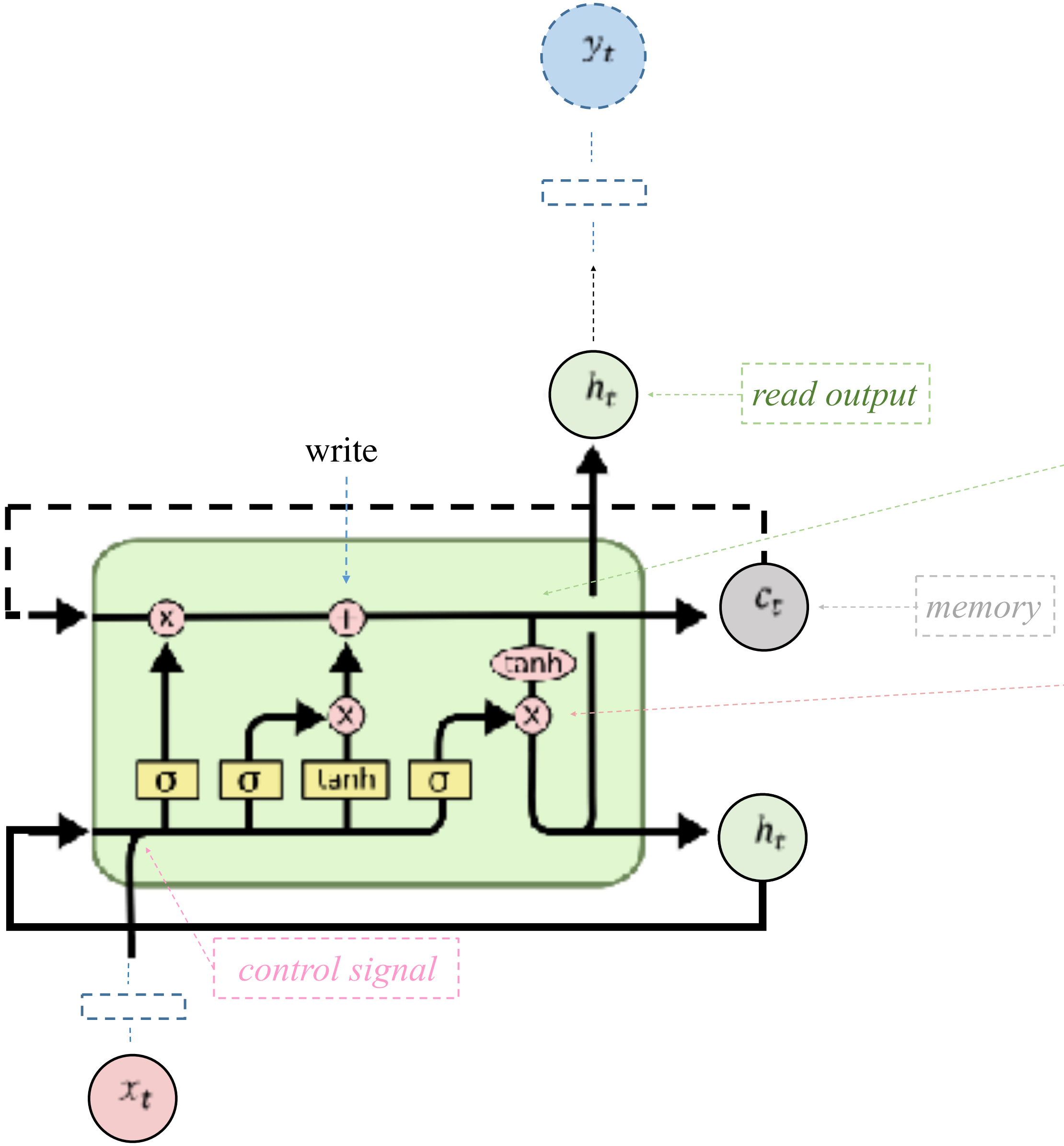
~~Memory addressing~~ – reads and writes the entire memory at once .

LSTM – Full-fledged memory controller





LSTM



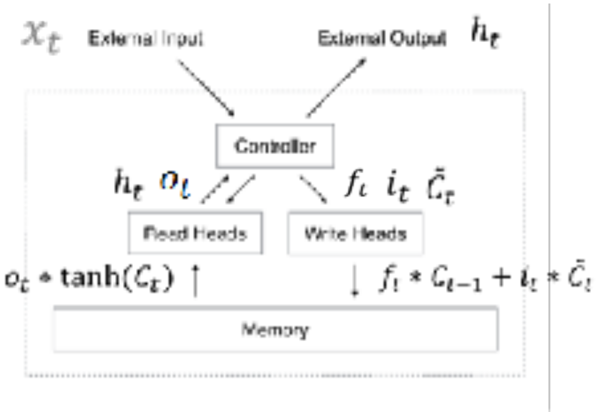
- Memory is separated from the output

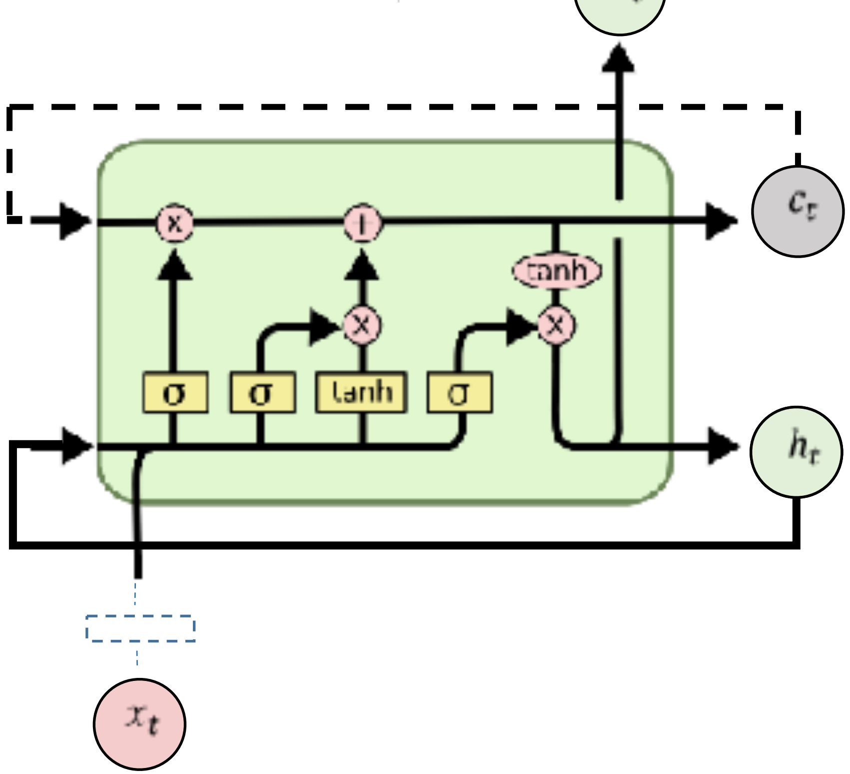
- Memory addressing mechanism

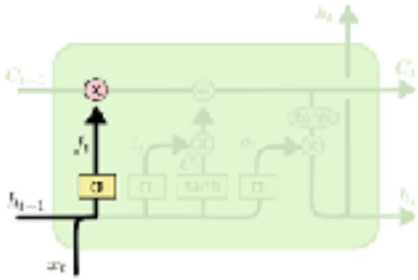
- Uses same building blocks as basic RNN:

$$\sigma(W \cdot [h_{t-1}, x_t] + b)$$

LSTM







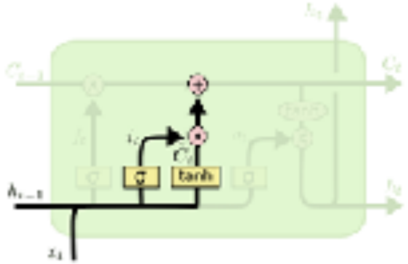
1. Erase memory

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$C'_t = f_t * C_{t-1}$$

f_t	0.1	1	1	1	0.5	1	0

erase most of block 1 and part of block 5



2. Write new data

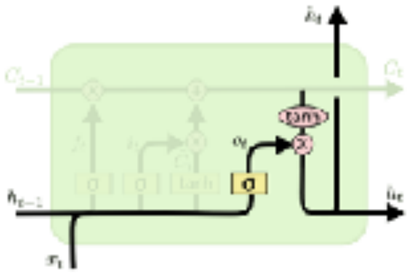
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = C'_t + i_t * \tilde{C}_t$$

i_t	1	0	0	0	0	0	0

write block 1 of $\tilde{C}_t$ to C_t



3. Read memory

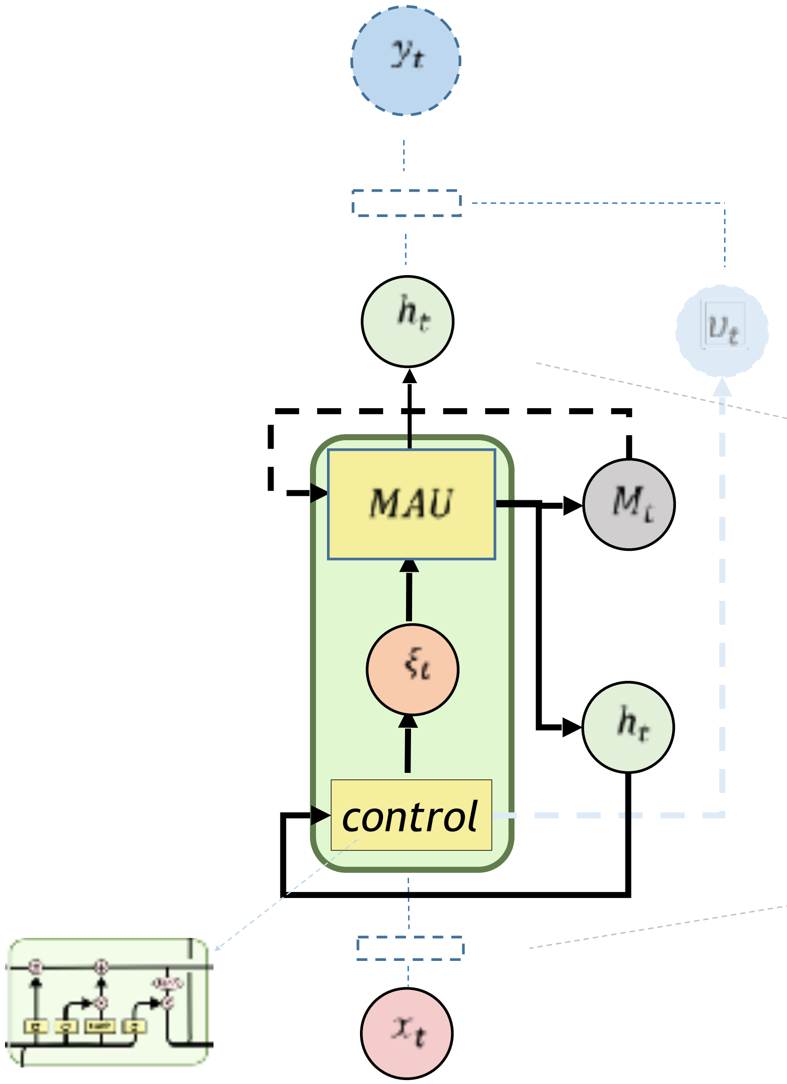
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

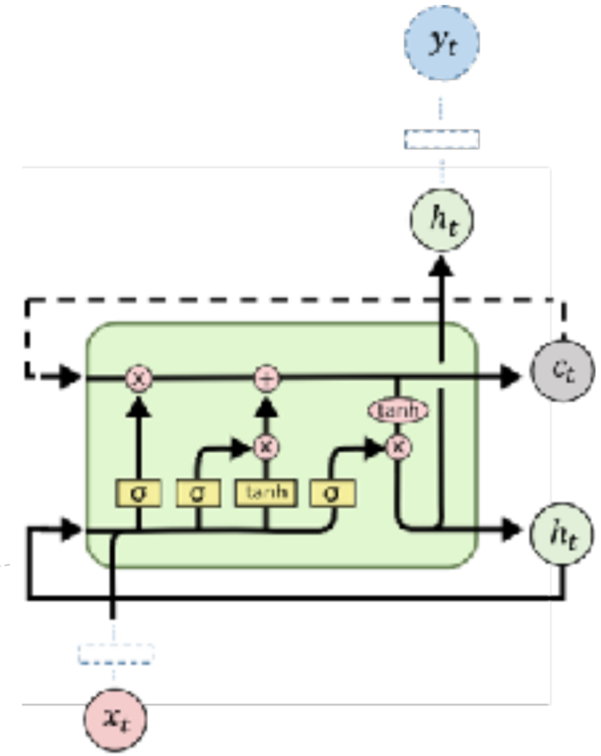
o_t	0	0.9	0	0	0	0	0.1

read most of block 2

Differentiable Neural Computer

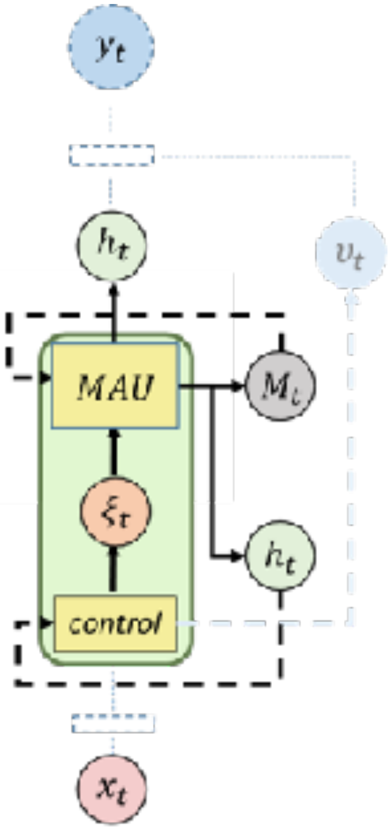


DNC



LSTM

DNC vs LSTM



DNC

$[h_{t-1}, x_t]$

$\xi_t = [e_t, v_t, k_t^w, \beta_t^w, f_t, g_t^a, g_t^w, k_t^r, \beta_t^r, \pi_t]$

$Control([h_{t-1}, x_t])$

ξ_t

read

write

erase

e_t

new content

v_t

$w_t^r = MAU(k_t^r, \beta_t^r, \pi_t)$

$w_t^w = MAU(k_t^w, \beta_t^w, f_t, g_t^a, g_t^w)$

$h_t = M_t^\intercal w_t^r$

Read operation

$M_t = M_t' + w_t^w v_t^\intercal$

Write new content

$M_t' = M_{t-1}(1 - w_t^w e_t^\intercal)$

Erase old content

softmax address

M_t

0.7	0.9	0.5	0.3	0.5	0.9	0.7
0.7	0.3	0.9	0.3	0.7	0.9	0.7
0.5	0.9	0.5	0.9	0.5	0.9	0.7
0.7	0	0.5	0.5	0.9	0.3	0

w_t^w

0
0.9
0.1
0

$e_t^\intercal$

0	0	0	0.9	0	0.1	0

$v_t^\intercal$

1	1	0	0	0	1	0

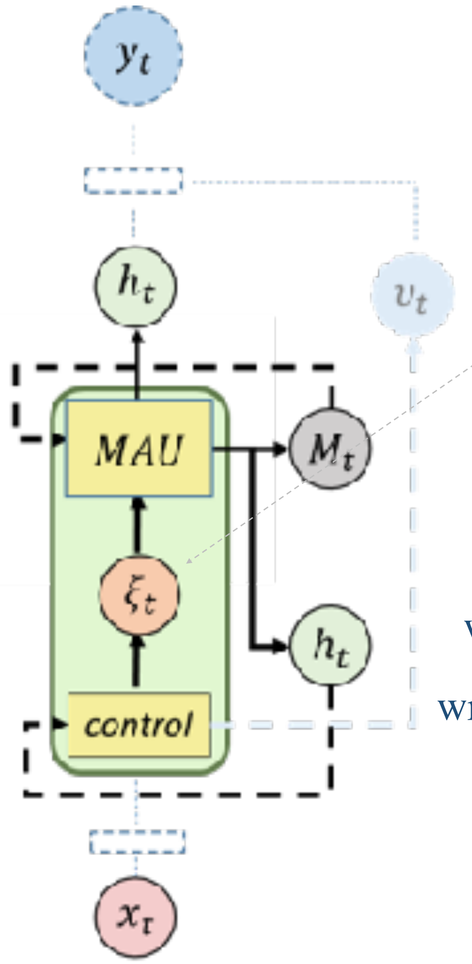
$w_t^w e_t^\intercal$

0	0	0	0	0	0	0
0	0	0	0.8	0	0.1	0
0	0	0	0.1	0	0	0
0	0	0	0	0	0	0

$w_t^w v_t^\intercal$

0	0	0	0	0	0	0
0.9	0.9	0	0	0	0.9	0
0.1	0.1	0	0	0	0.1	0
0	0	0	0	0	0	0

DNC vs LSTM

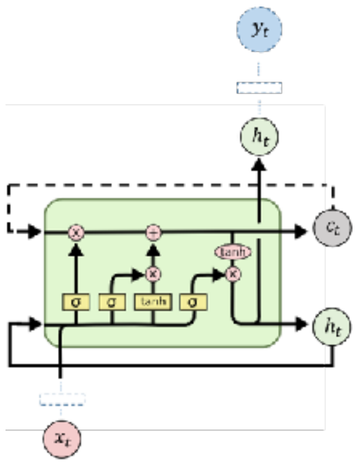


DNC

$$Control([h_{t-1}, x_t]) = \xi_t, v_t$$

$$\xi_t = [e_t, v_t, k_t^w, \beta_t^w, f_t, g_t^a, g_t^w, k_t^r, \beta_t^r, \pi_t]$$

memory:	M_t	C_t
erase address:	$e_t = Control([h_{t-1}, x_t])$	$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
write content:	$v_t = Control([h_{t-1}, x_t])$	$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$
write address:	$w_t^w = MAU(k_t^w, \beta_t^w, f_t, g_t^a, g_t^w)$	$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$
write operation:	$M_t' = M_{t-1}(1 - w_t^w e_t^\intercal)$	$C_t' = f_t * C_{t-1}$
	$M_t = M_t' + w_t^w v_t^\intercal$	$C_t = C_t' + i_t * \tilde{C}_t$
read address:	$w_t^r = MAU(k_t^r, \beta_t^r, \pi_t)$	$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$
read output:	$h_t = M_t^\intercal w_t^r$	$h_t = o_t * \tanh(C_t)$



LSTM

DNC - Drill Down

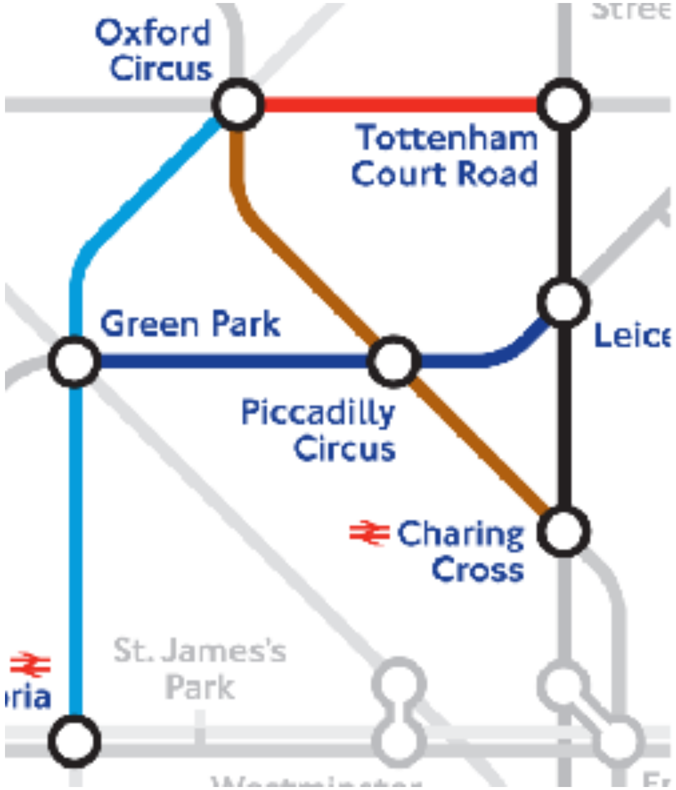
Associative memory

I cnduo't bvleiee taht I culod aulaclty uesdtannrd waht I was rdnaieg.

Oxford Circus > ________?

Line:

Central
Victoria
Piccadilly
Bakerloo



Memory

Oxford Circus>Tottenham Court Rd
Tottenham Court Rd>Oxford Circus
Green Park>Oxford Circus
Victoria>Green Park
Oxford Circus>Green Park
Green Park>Victoria
Green Park>Piccadilly Circus
Piccadilly Circus>Leicester Sq
Piccadilly Circus>Green Park
Leicester Sq>Piccadilly Circus
Piccadilly Circus>Oxford Circus
Charing Cross>Piccadilly Circus
Piccadilly Circus>Charing Cross
Oxford Circus>Piccadilly Circus
Leicester Sq>Tottenham Court Rd
Charing Cross>Leicester Sq
Leicester Sq>Charing Cross
Tottenham Court Rd>Leicester Sq

DNC - Drill Down

Associative memory

write strength

write key

read strength

read key

$$\xi_t = [e_t, \underbrace{k_t^w}_{\text{write key}}, \underbrace{\beta_t^w}_{\text{write strength}}, f_t, g_t^a, g_t^w, \underbrace{k_t^r}_{\text{read key}}, \underbrace{\beta_t^r}_{\text{read strength}}, \pi_t, v_t]$$

k_t

0.7	0	0.5	0.5	0.9	0.9	0

cosine similarity

$$w'_{t,i} = \frac{k_t \cdot M_{t,i\rightarrow}}{\|k_t\|\|M_{t,i\rightarrow}\|}$$

M_t

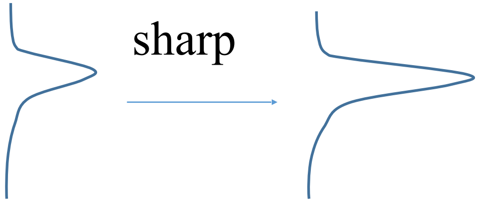
0.7	0.9	0.5	0.3	0.5	0.9	0.7
0.7	0.3	0.9	0.3	0.7	0.9	0.7
0.5	0.9	0.5	0.9	0.5	0.9	0.7
0.7	0	0.5	0.5	0.9	0.3	0

Dove body lotion

~~moisturizer~~

Weighted soft max - Sharp and normalize

$$w_{t,i}^{assoc} = \frac{\exp(w'_{t,i}\beta_t)}{\sum_j \exp(w'_{t,j}\beta_t)} \quad s.t\ \beta_t \in [1, \infty]$$

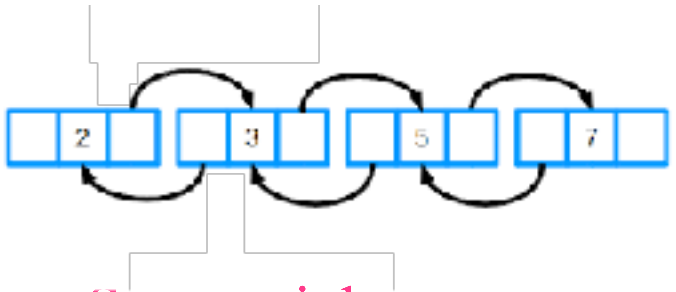


large beta = using a single block

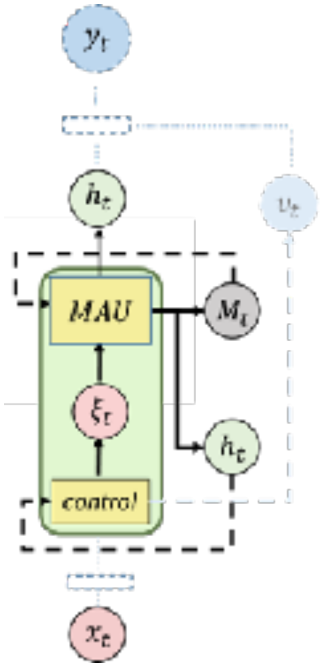
DNC - Drill Down

$\xi_t = [e_t, k_t^w, \beta_t^w, f_t, g_t^a, g_t^w, k_t^r, \beta_t^r, \pi_t, v_t]$

read mode



Sequential memory



L_t – link matrix

0	0.1	0.1	0.8
0.2	0	0.2	0.1
0.3	0.4	0	0.1
0.5	0.5	0.7	0

block 1 was written to after block 4 with degree 0.8

$L_t[i,j]$ *represents the degree to which memory block i was the location written to after block j*

Computing final read address: $w_t^r = MAU(k_t^r, \beta_t^r, \pi_t)$

$f_t = L_t w_{t-1}^r$ next memory block

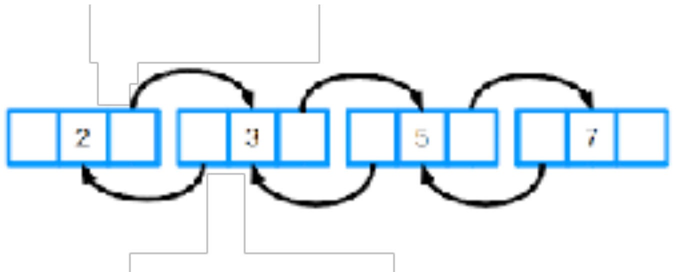
$b_t = L_t^T w_{t-1}^r$ previous memory block

$w_t^{assoc\_k_t^r}$ memory blocks similarity weight for read key k_t^r

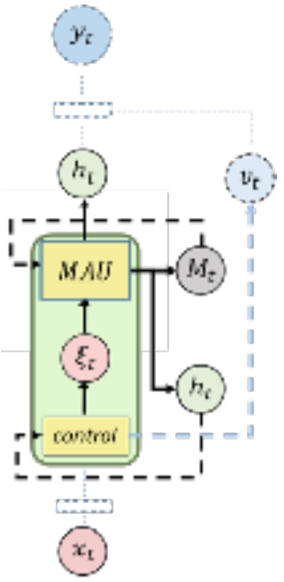
$\pi_t \epsilon S_3$ switch between *Associative* and *Sequentall* addressing mechanism

read weights: $\boxed{w_t^r = \pi_t[1] b_t + \pi_t[2] w_t^{assoc\_k_t^r} + \pi_t[3] \hat{f}_t}$

DNC – Reading from memory



Computation of L_t



location j was written to in the current step

*degree to which **block j** was the last one written to:*

$$P_{t-1}[j] = w^W_{t-1}[j] + (1 - \sum_k w^W_{t-1}[k])P_{t-2}[j]$$

location j was the last one written to and no block was written to in the current step

degree to which block i was written to after block j:

$$L_t[i,j] = w^W_t[i]P_{t-1}[j] + (1 - w^W_t[i] - w^W_t[j])L_{t-1}[i,j]$$

block j was the last block written to and block i is the current block written to

block i was written to after block j until the current step
and **both blocks were not writtent to in the current step**

DNC – Writing to memory

write weights: $w_t^w \rightarrow MAU(k_t^w, \beta_t^w, f_t, g_t^a, g_t^w)$

write key

erase vector

write strength

$\xi_t = [e_t, k_t^w, \beta_t^w, f_t, g_t^a, g_t^w, k_t^r, \beta_t^r, \pi_t, v_t]$

free gate

write gate

allocation gate

Address allocation

u_t memory blocks usage weights $\quad u_t = (1 - f_t w_{t-1}^r) * (u_{t-1} + w_{t-1}^w - u_{t-1} w_{t-1}^w)$

$\acute{a}_t = 1 - u_t$ memory blocks availability for allocation

$a_t = a$ *sharper version of* $\acute{a}_t$ *which requiers sorting (we used weigthed softmax instead)*

Address Association

$w_t^{assoc\_k_t^w}$ memory blocks similarity weights for write key (associative addressing)

g_t^a *switch between allocatio vector and write key similarity vector*

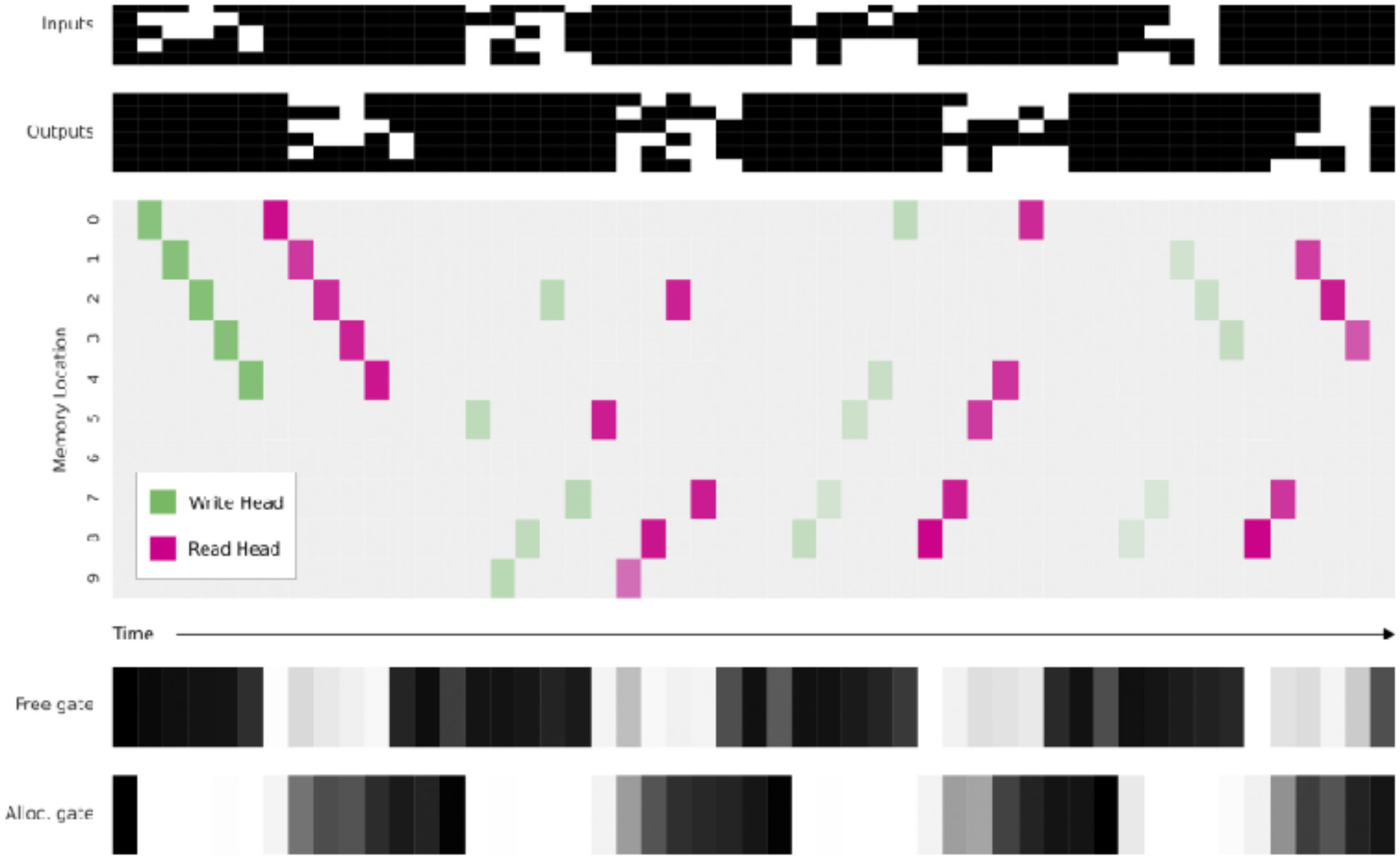
Final write address

write weights: $$\acute{w}_t^w = g_t^a a_t + (1 - g_t^a) w_t^{assoc\_k_t^w}$$
$$w_t^w = g_t^w \acute{w}_t^w$$

erase: $\boxed{M_t' = M_{t-1}(1 - w_t^w e_t^\intercal)}$

write: $\boxed{M_t = M_t' + w_t^w v_t^\intercal}$

Copy task


Inputs
Outputs
Memory Location
0
1
2
3
4
5
6
7
8
9
Write Head
Read Head
Time
Free gate
Alloc. gate

London underground

Traversal | **Shortest-path**

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

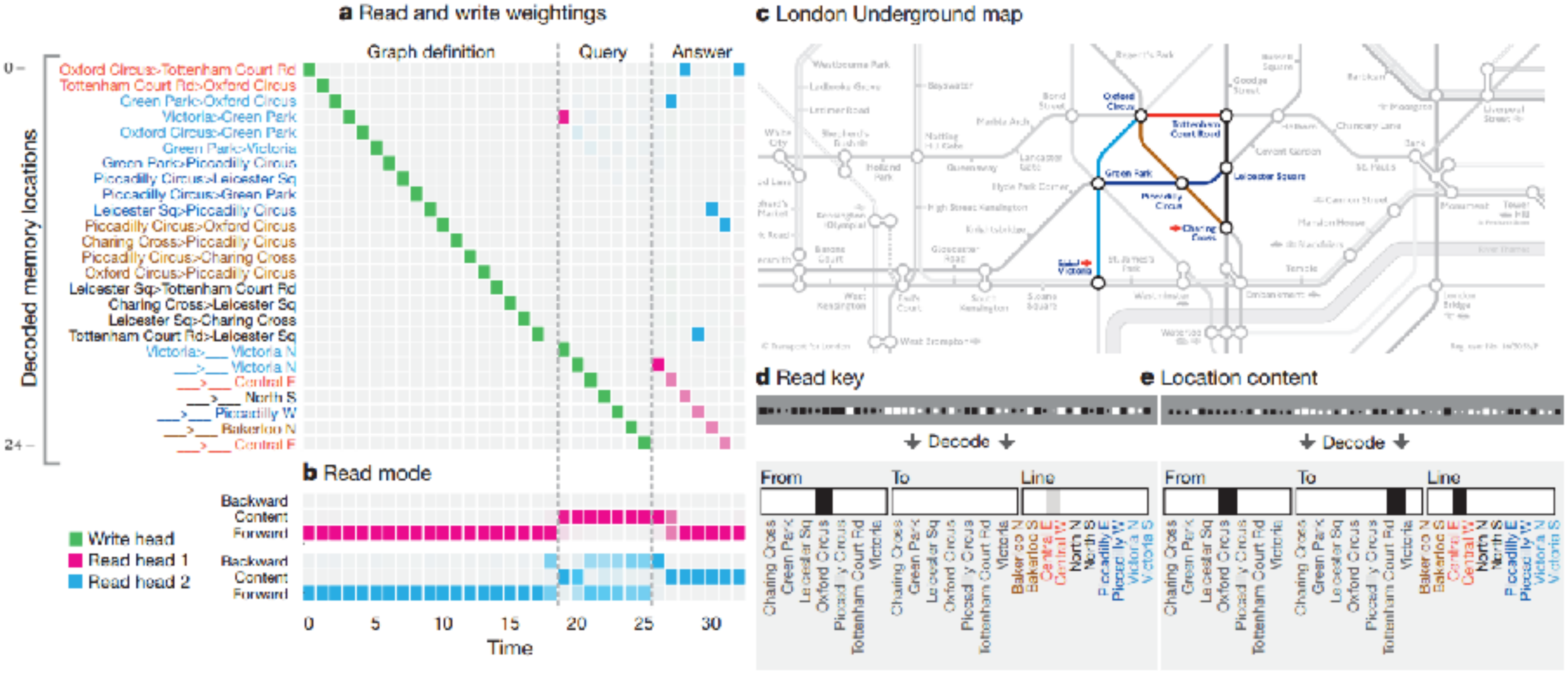
Traversal question:
(BondSt, _, Central),
(_, _, Circle), (_, _, Circle),
(_, _, Circle), (_, _, Circle),
(_, _, Jubilee), (_, _, Jubilee),

Answer:
(BondSt, NottingHillGate, Central)
(NottingHillGate, GloucesterRd, Circle)
⋮
(Westminster, GreenPark, Jubilee)
(GreenPark, BondSt, Jubilee)

Shortest-path question:
(Moorgate, PiccadillyCircus, _)

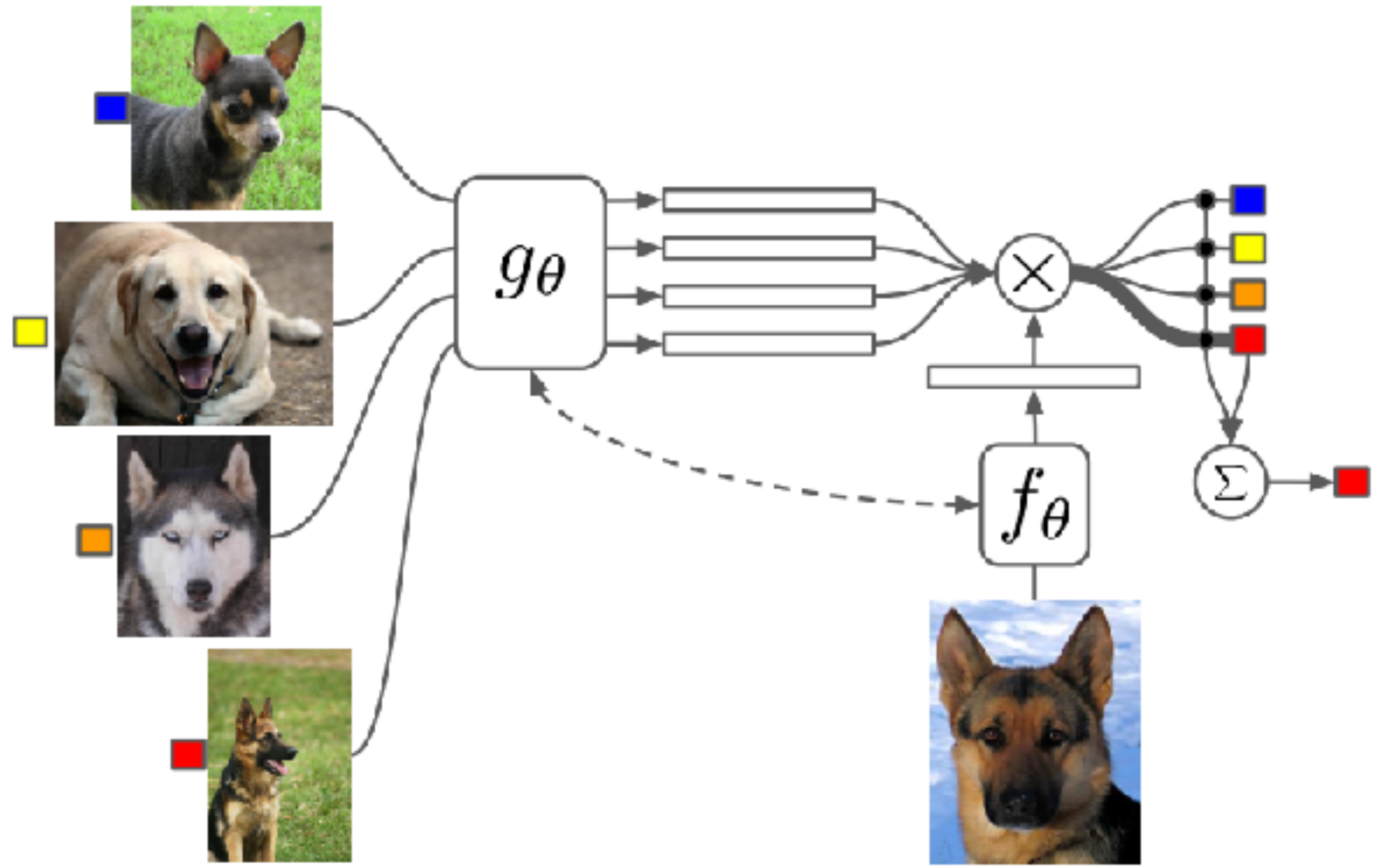
Answer:
(Moorgate, Bank, Northern)
(Bank, Holborn, Central)
(Holborn, LeicesterSq, Piccadilly)
(LeicesterSq, PiccadillyCircus, Piccadilly)

London underground

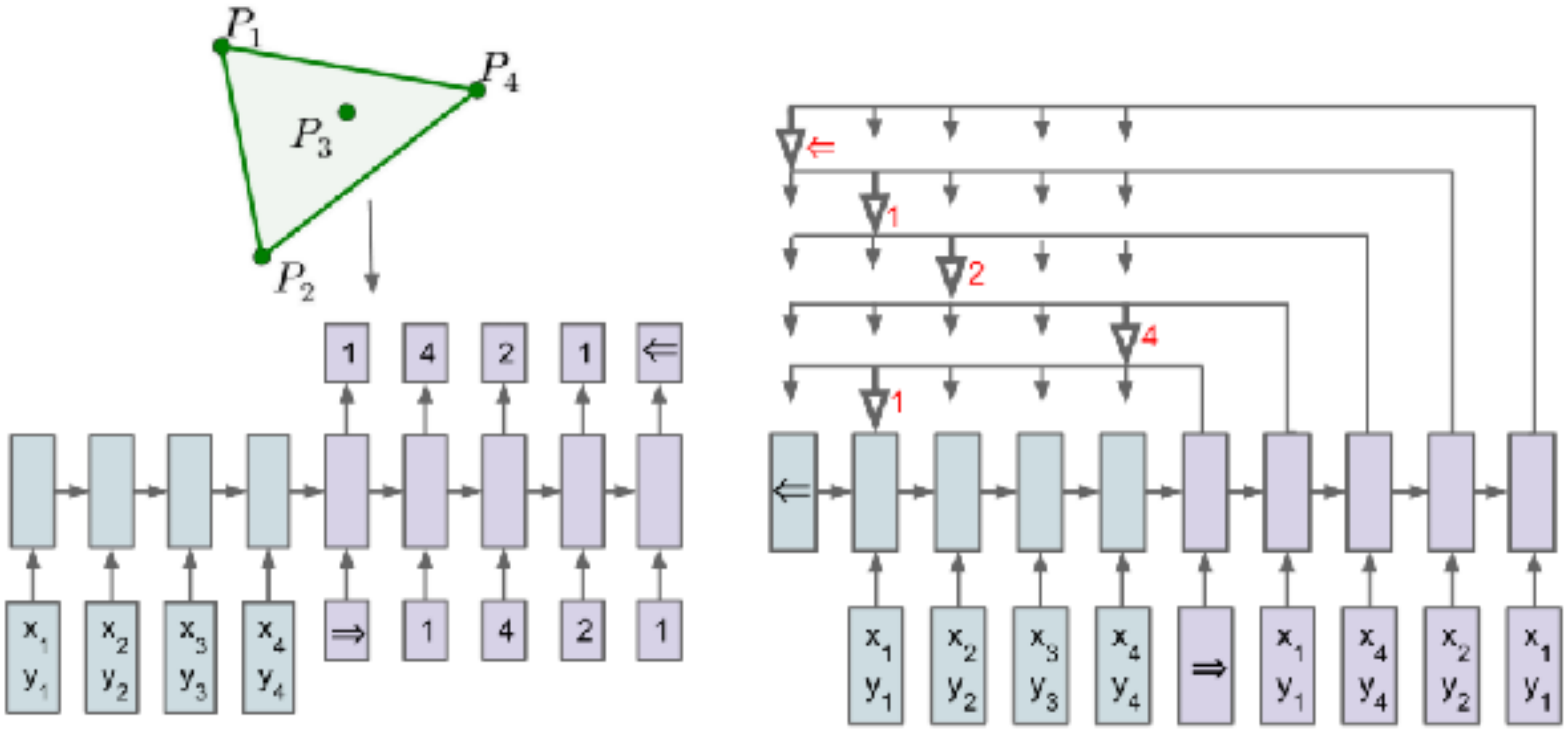


Associative memory in action: Oxford Circus>Tottenham Court Rd

Matching Networks



Pointer networks



(a) Sequence-to-Sequence

(b) Ptr-Net