Deep Learning : Dealing with Longer Sequences – LSTMs and GRUs

राष्ट्रीय प्रौद्योगिकी संस्थान सिक्किम

NATIONAL INSTITUTE OF TECHNOLOGY SIKKIM

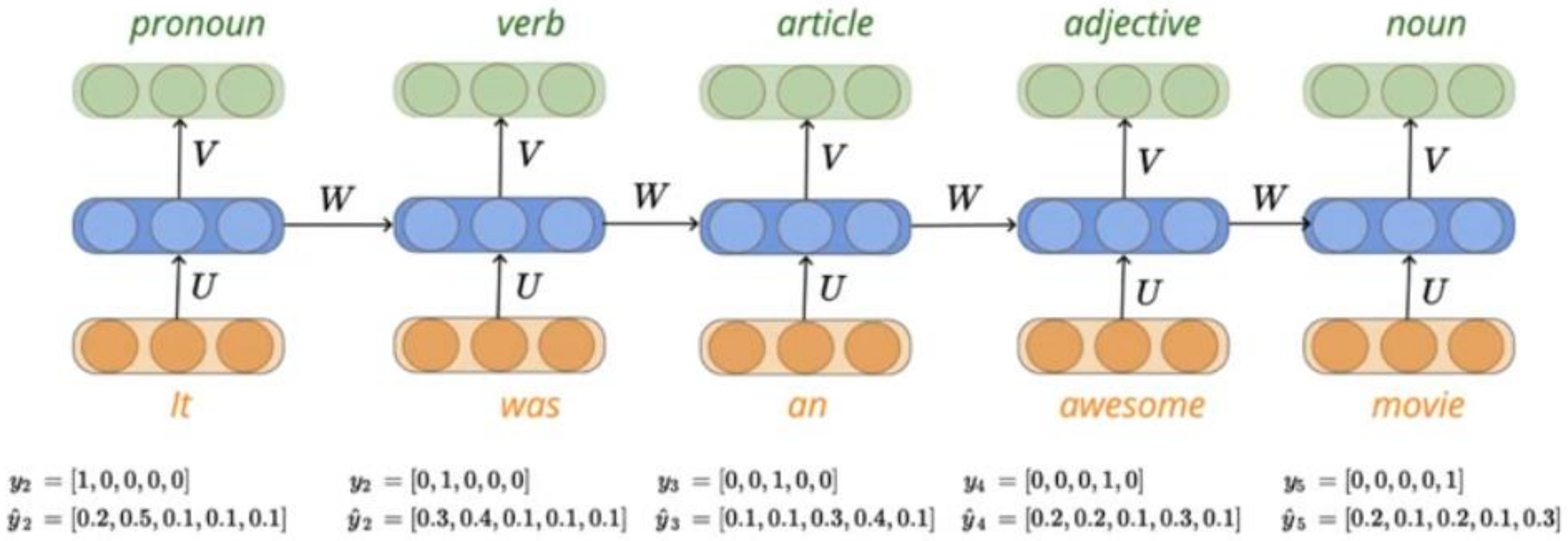
Course Instructor:

Dr. Bam Bahadur Sinha

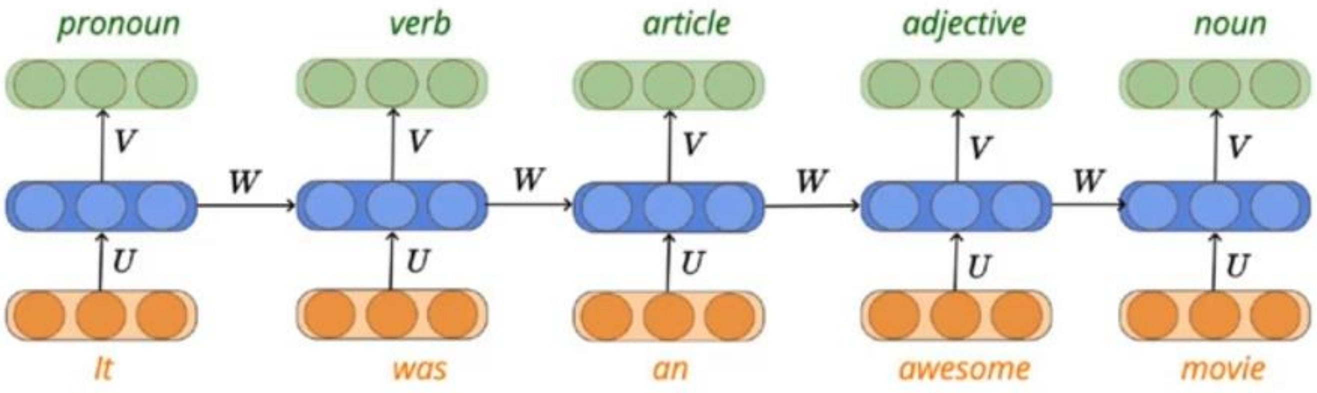
Assistant Professor

Computer Science & Engineering

National Institute of Technology Sikkim

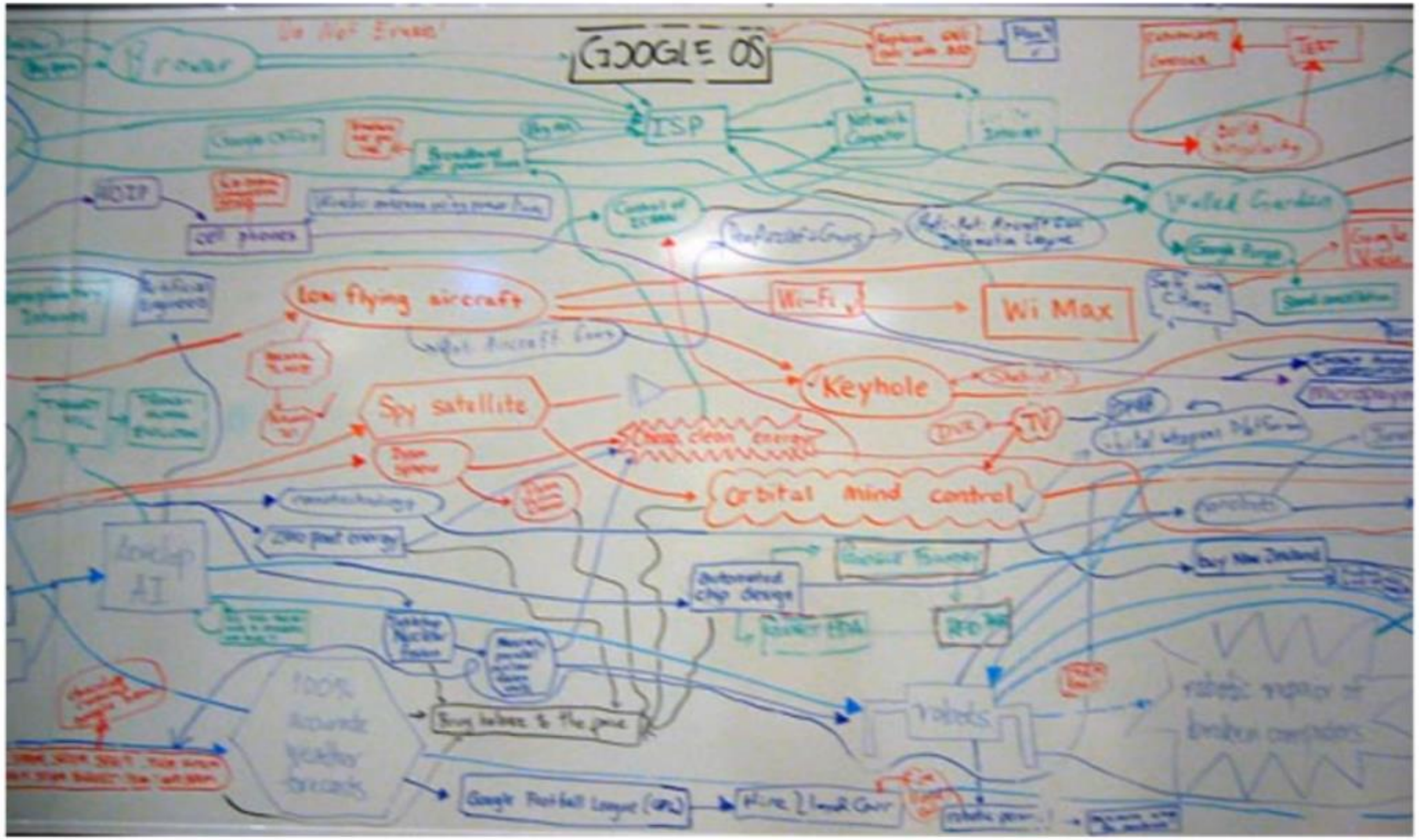


Recurrent Neural Network (RNNs)



- At each new timestep the old information gets morphed by the current input
- One could imagine that after t steps the information stored at time step t − k (for some k < t) gets completely morphed
- Even during backpropagation the information does not flow well

Dealing with Longer Sequences



Whiteboard Analogy

$a = 1 \quad b = 3 \quad c = 5 \quad d = 11$

Compute $ac(bd + a) + ad$

1. ac
2. bd
3. $bd + a$
4. $ac(bd + a)$
5. ad
6. $ac(bd + a) + ad$

$$ac = 5$$

$$bd = 33$$

$$bd + a = 34$$

Strategy

- ✓ Selectively write on the board
- ✓ Selectively read the already written content
- ✓ Selectively forget (erase) some content

Whiteboard Analogy

$a = 1 \quad b = 3 \quad c = 5 \quad d = 11$

Compute $ac(bd + a) + ad$

1. ac
2. bd
3. $bd + a$
4. $ac(bd + a)$
5. ad
6. $ac(bd + a) + ad$

$$ac = 5$$

$$ac(bd + a) = 170$$

$$bd + a = 34$$

Strategy

- Selectively write on the board
- Selectively read the already written content
- Selectively forget (erase) some content

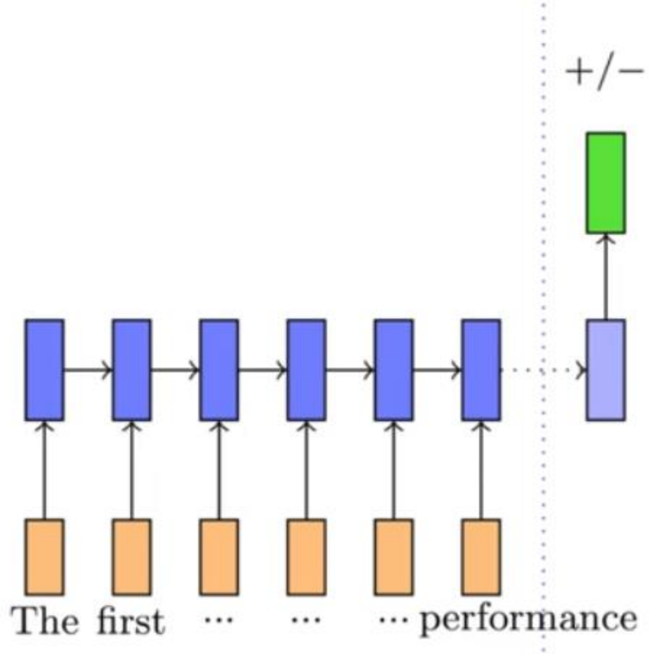
Whiteboard Analogy

Since the RNN also has a finite state size, we need to figure out a way to allow it to selectively read, write and forget

Strategy

- Selectively write to the state
- Selectively read the already written content
- Selectively forget (erase) some content

Can we use similar strategy in RNNs?

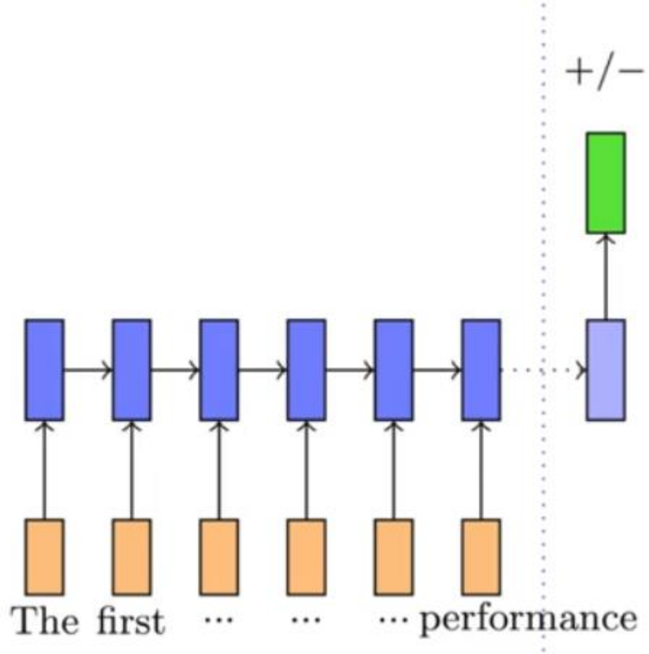


Review: The first half of the movie was dry but the second half really picked up pace. The lead actor delivered an amazing performance

Ideally, we want to

- forget the information added by stop words (a, the, etc.)
- selectively read the information added by previous sentiment bearing words (awesome, amazing, etc.)
- selectively write new information from the current word to the state

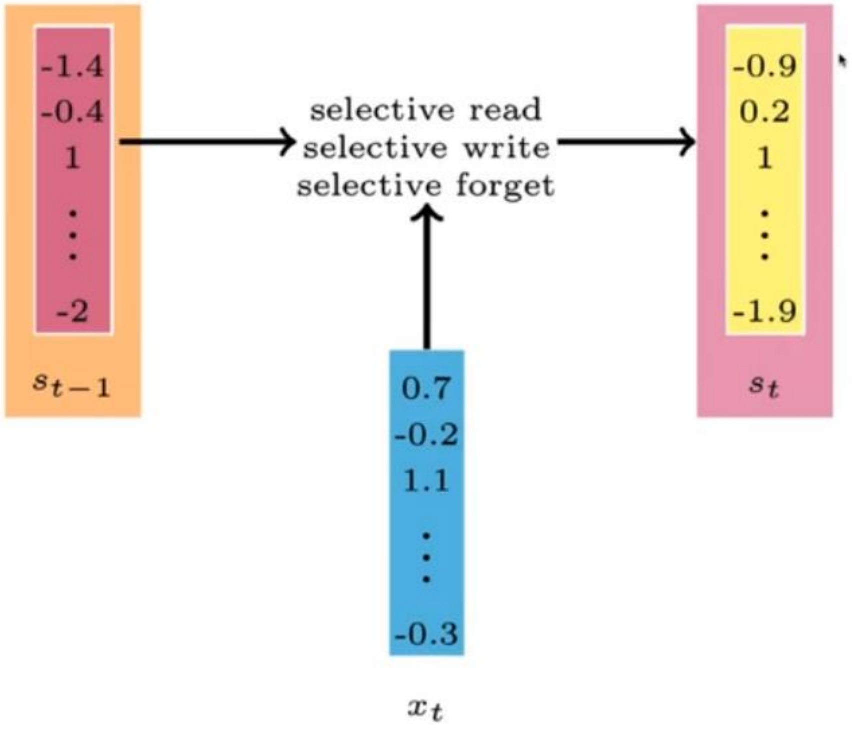
Can we use similar strategy in RNNs?



Review: The first half of the movie was dry but the second half really picked up pace. The lead actor delivered an amazing performance

Wishlist: selective write, selective read and selective forget to ensure that this finite sized state vector is used effectively

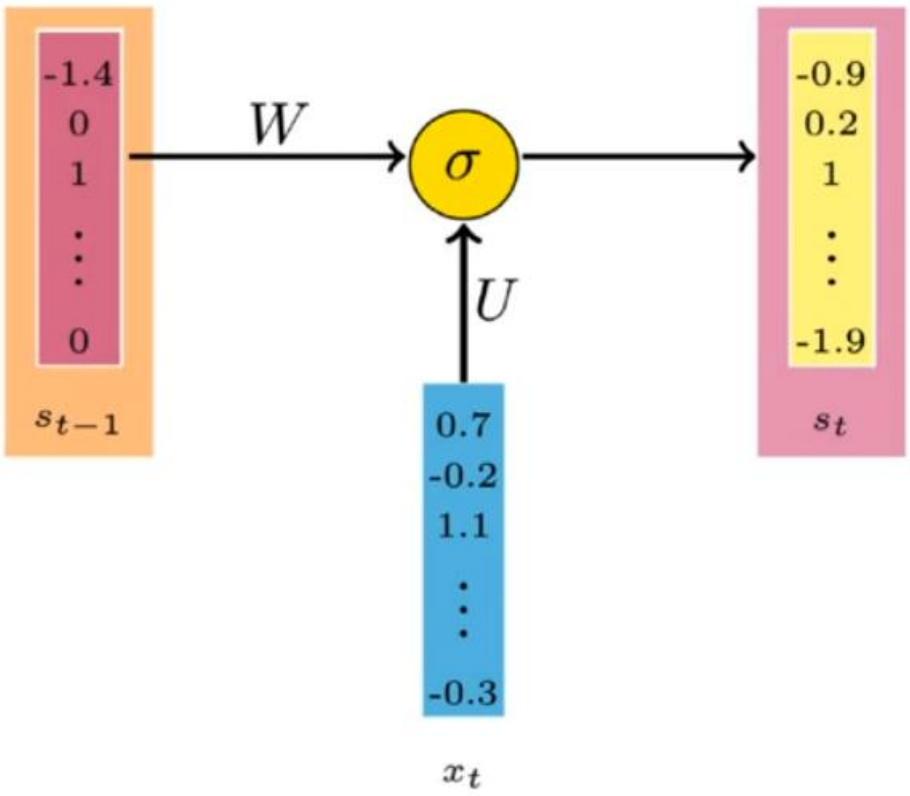
Wishlist – Dealing with longer sequences



While computing s_t from s_{t-1} we want to make sure that we use selective write, selective read and selective forget so that only important information is retained in s_t

Long Short Term Memory Cells

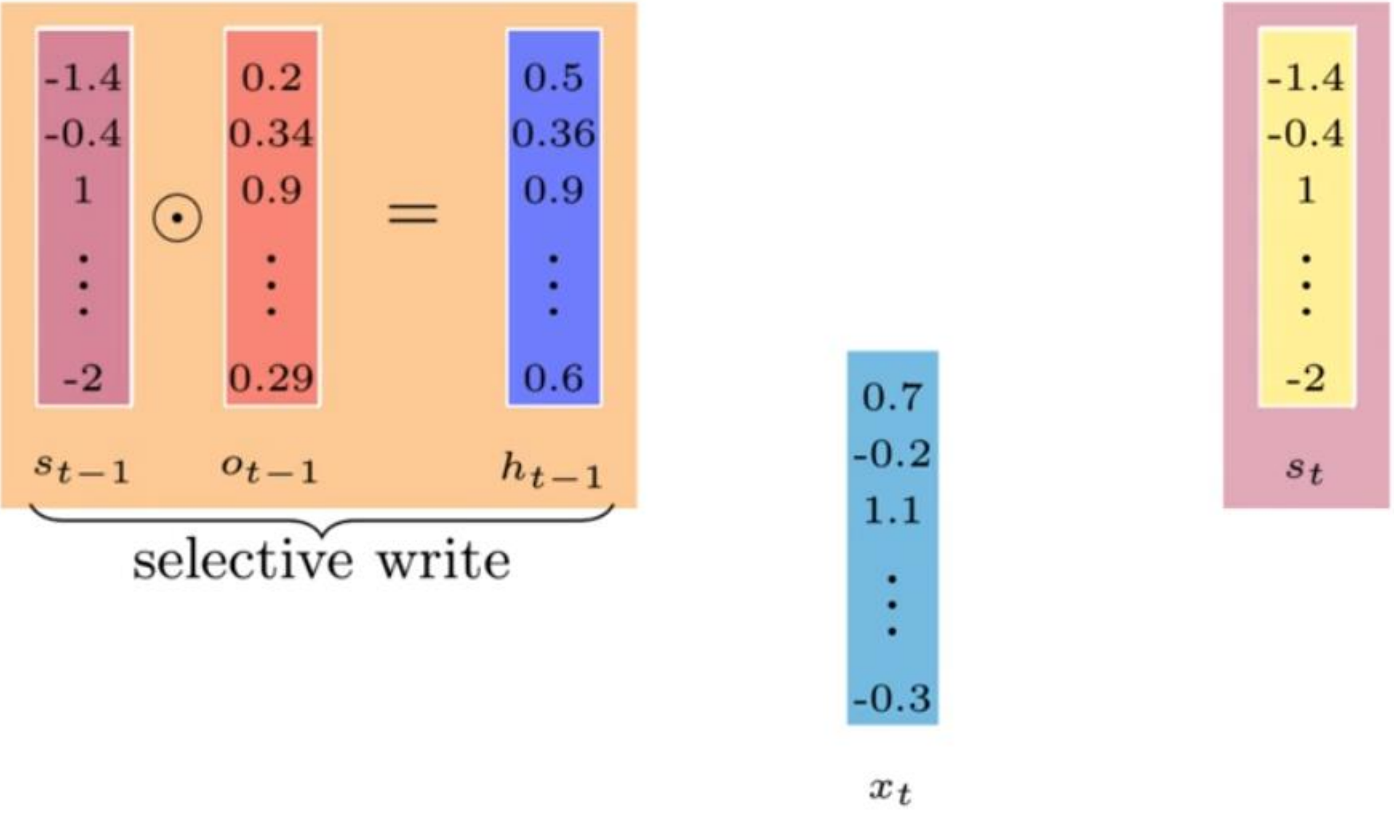
How do you implement selective read, write and forget?



$$s_t = \sigma(Ux_t + W\mathbf{s_{t-1}} + b)$$

- instead of passing s_{t-1} as it is to s_t we want to pass (write) only some portions of it to the next state
- A reasonable way of doing this would be to assign a value between 0 and 1 which determines what fraction of the current state to pass on to the next state

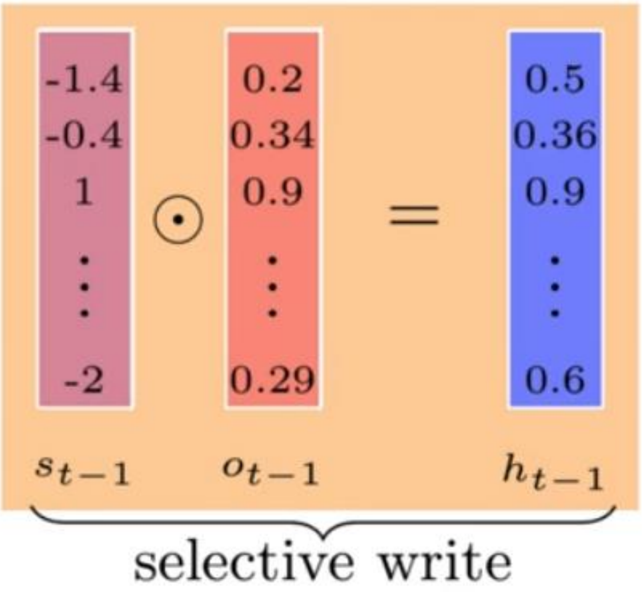
Selective Write

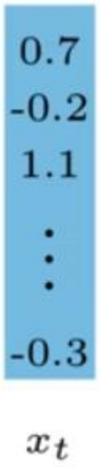


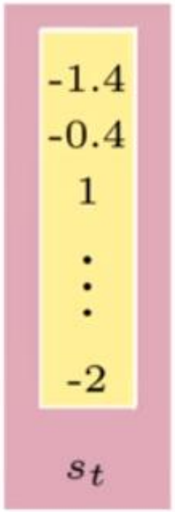
But how do we compute o_{t-1} ? How does the RNN know what fraction of the state to pass on?

- learn o_{t-1} from data
- the only thing that we learn from data is parameters
- **Solution:** express o_{t-1} using parameters

Selective Write





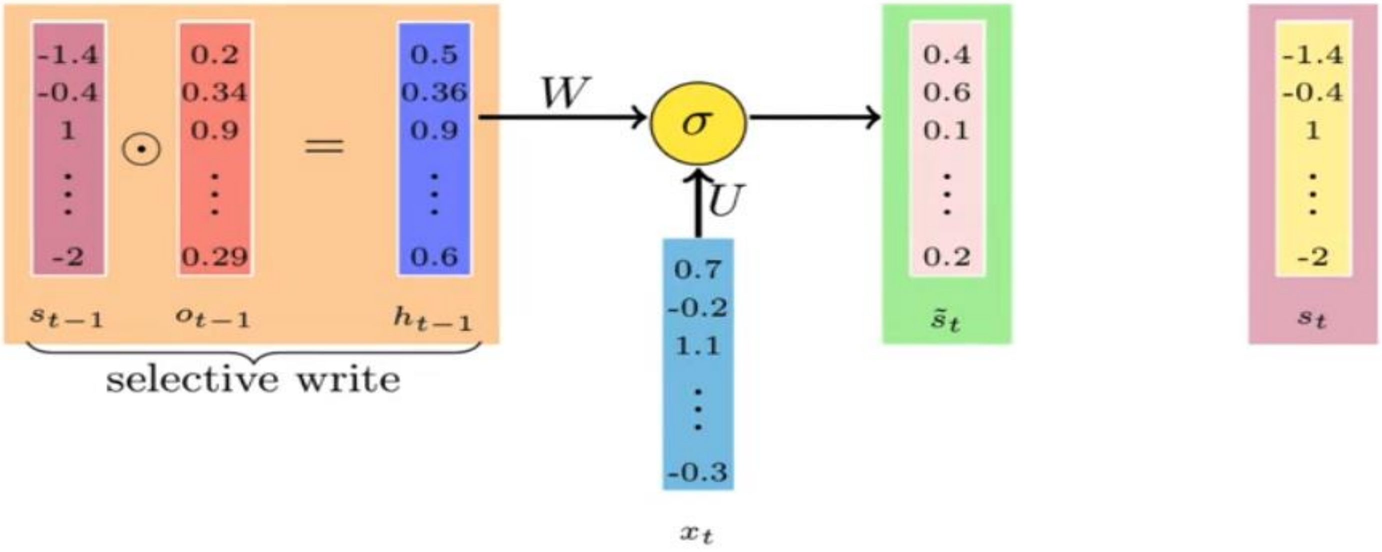


$$o_{t-1} = \sigma(U_o x_{t-1} + W_o h_{t-2} + b_o)$$

$$h_{t-1} = s_{t-1} \odot o_{t-1}$$

o_t is called the **output** gate

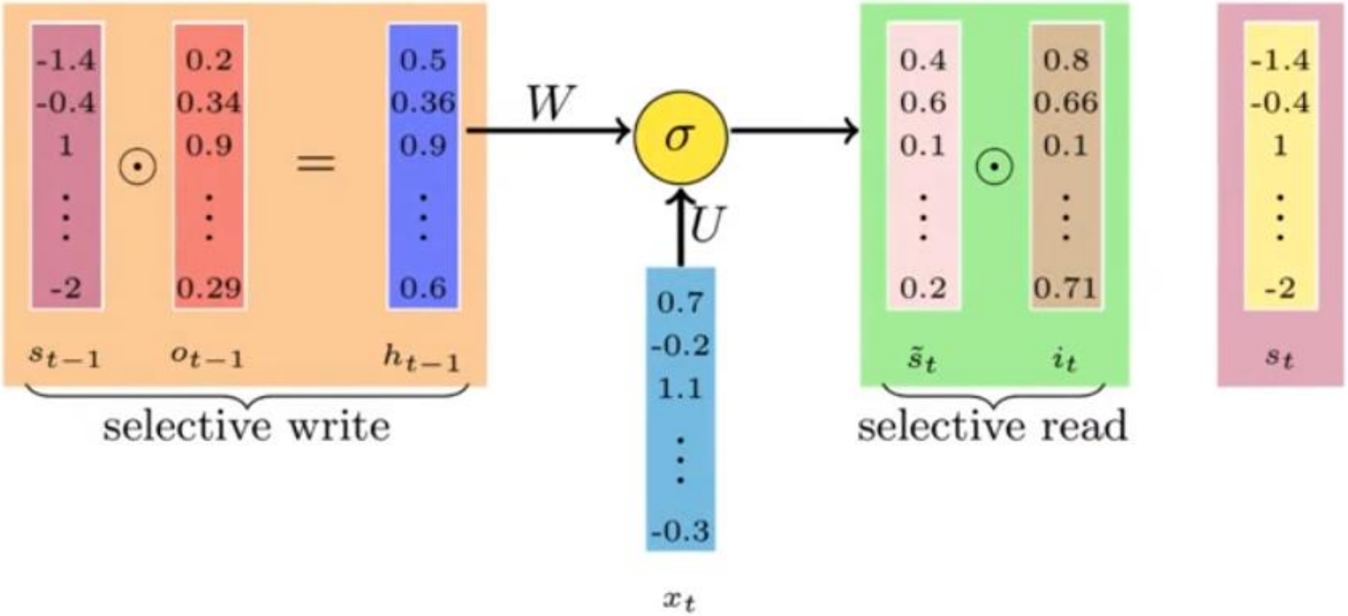
Selective Write



$$\tilde{s}_t = \sigma(Ux_t + Wh_{t-1} + b)$$

- $\tilde{s}_t$ thus captures all the information from the previous state h_{t-1} and the current input x_t
- However, we may not want to use all this new information and only selectively read from it before constructing the new cell st

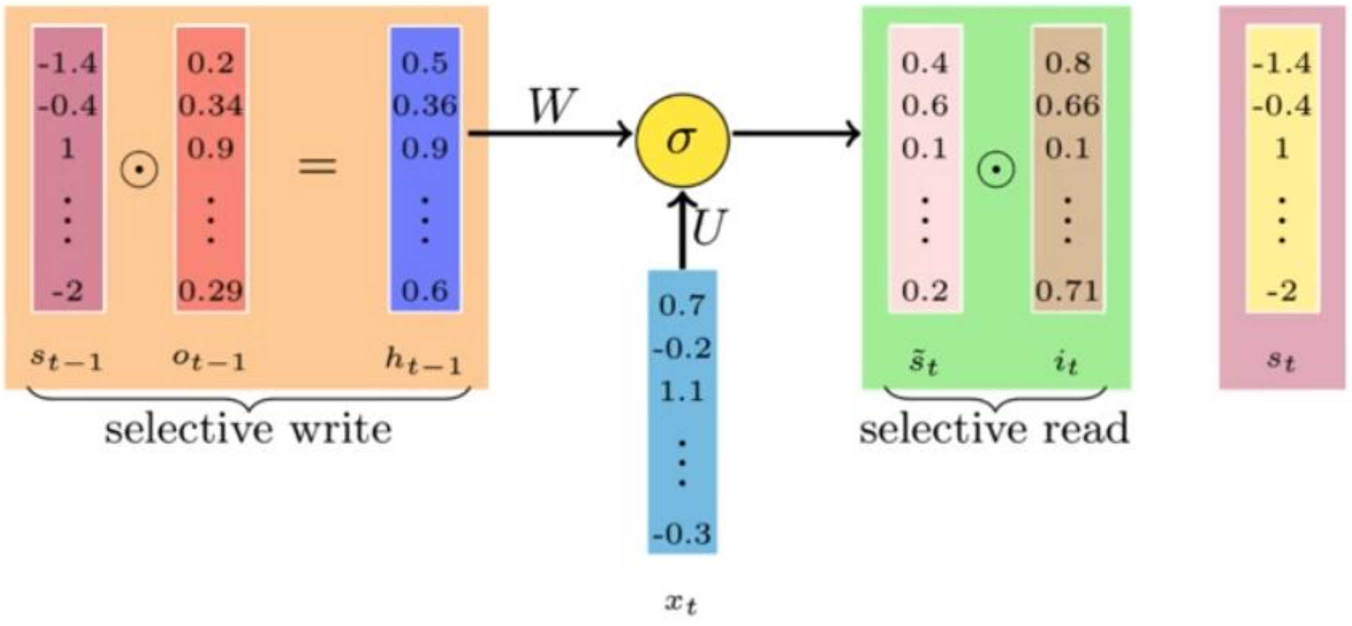
Selective Read



$$\begin{aligned} i_t &= \sigma(U_i x_t + W_i h_{t-1} + b_i) \\ &= \tilde{s}_t \odot i_t \end{aligned}$$

i_t is called the **input** gate

Selective Read



Previous state:

s_{t-1}

Output gate:

$o_{t-1} = \sigma(W_o h_{t-2} + U_o x_{t-1} + b_o)$

Selectively Write:

$h_{t-1} = o_{t-1} \odot \sigma(s_{t-1})$

Current (temporary) state:

$\tilde{s}_t = \sigma(W h_{t-1} + U x_t + b)$

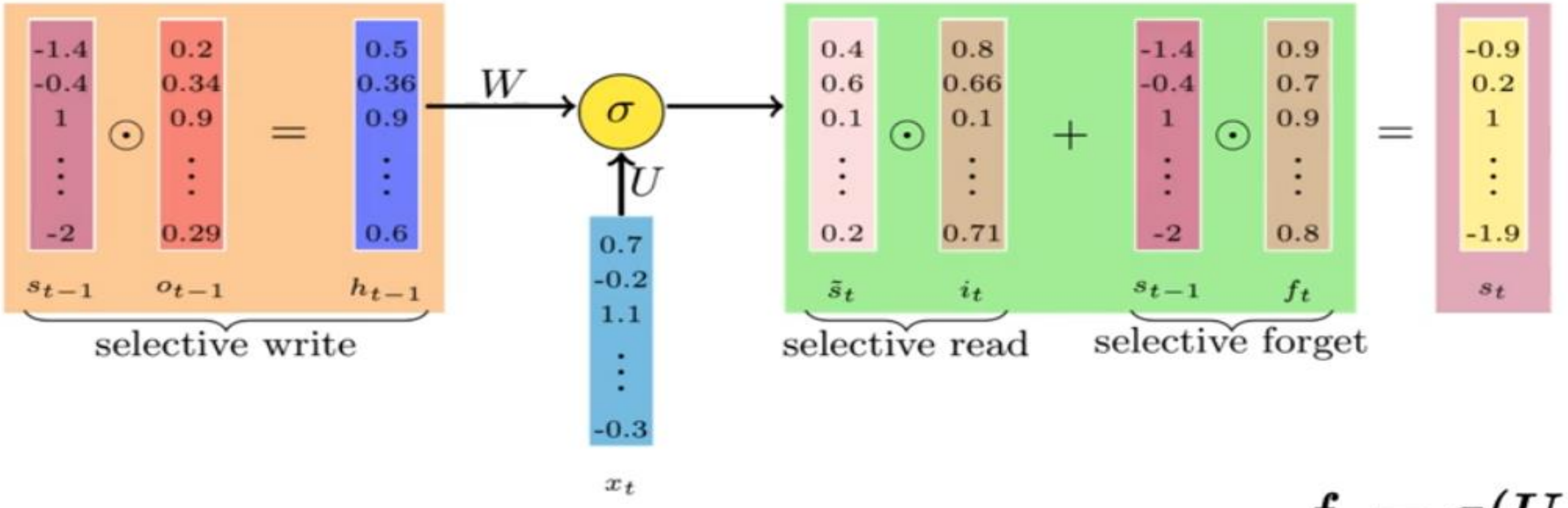
Input gate:

$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i)$

Selectively Read:

$i_t \odot \tilde{s}_t$

Summary – till selective read and write

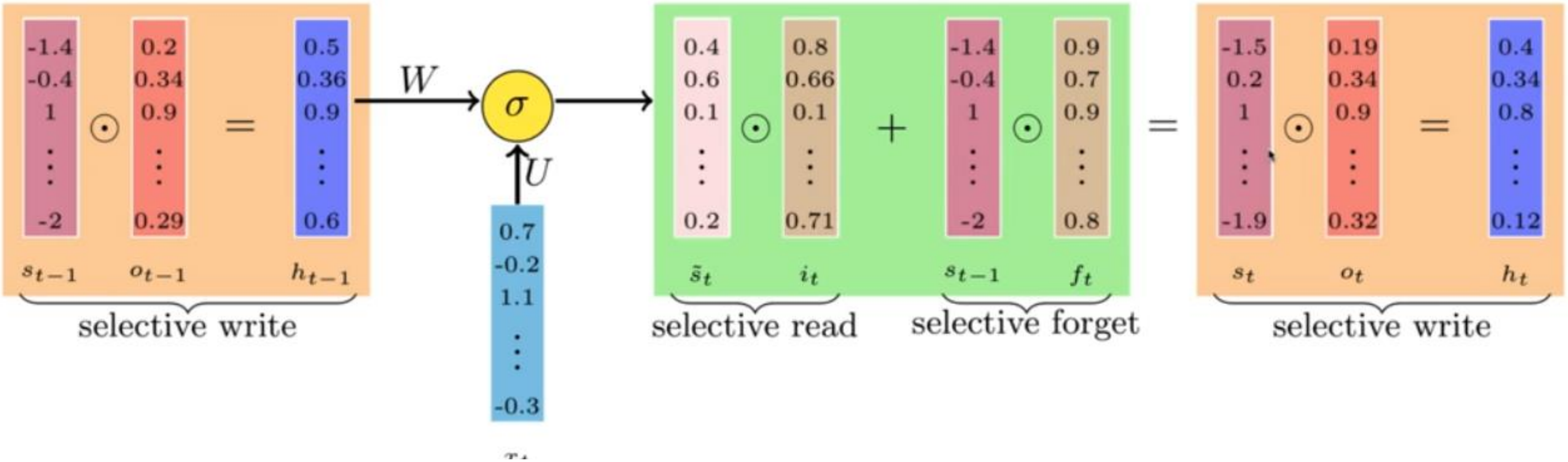


$$f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f)$$

$$s_t = \tilde{s}_t \odot i_t + s_{t-1} \odot f_t$$

How do we combine $\tilde{s}_t$ and s_{t-1} to get the new state s_t

Selective Forget



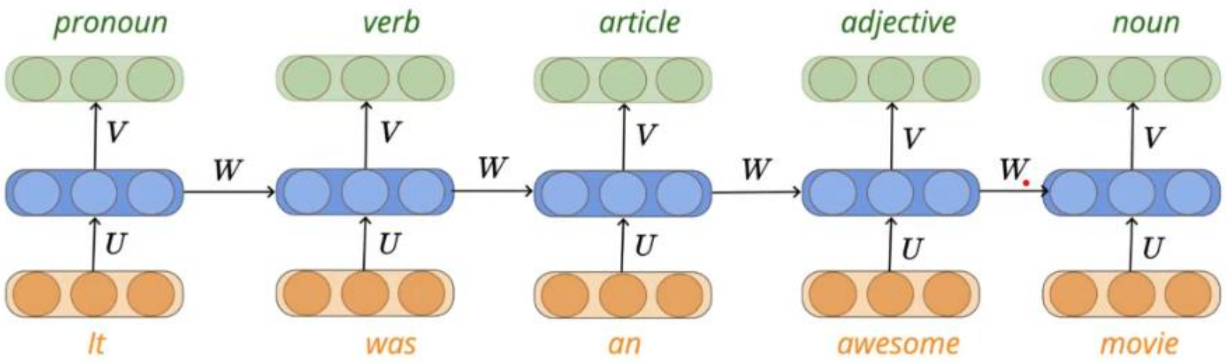
Gates:

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o)$$
$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i)$$
$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f)$$

States:

$$\tilde{s}_t = \sigma(W h_{t-1} + U x_t + b)$$
$$s_t = f_t \odot s_{t-1} + i_t \odot \tilde{s}_t$$
$$h_t = o_t \odot \sigma(s_t)$$

Full set of equations
(Selective Read, Selective Write and Selective Forget)



Gates:

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o)$$
$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i)$$
$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f)$$

States:

$$\tilde{s}_t = \sigma(W h_{t-1} + U x_t + b)$$
$$s_t = f_t \odot s_{t-1} + i_t \odot \tilde{s}_t$$
$$h_t = o_t \odot \sigma(s_t)$$

Long Short Term Memory Cells

- LSTM has many variants which include different number of gates and also different arrangement of gates
- The one which we just saw is one of the most popular variants of LSTM
- Another equally popular variant of LSTM is Gated Recurrent Unit which we will see next

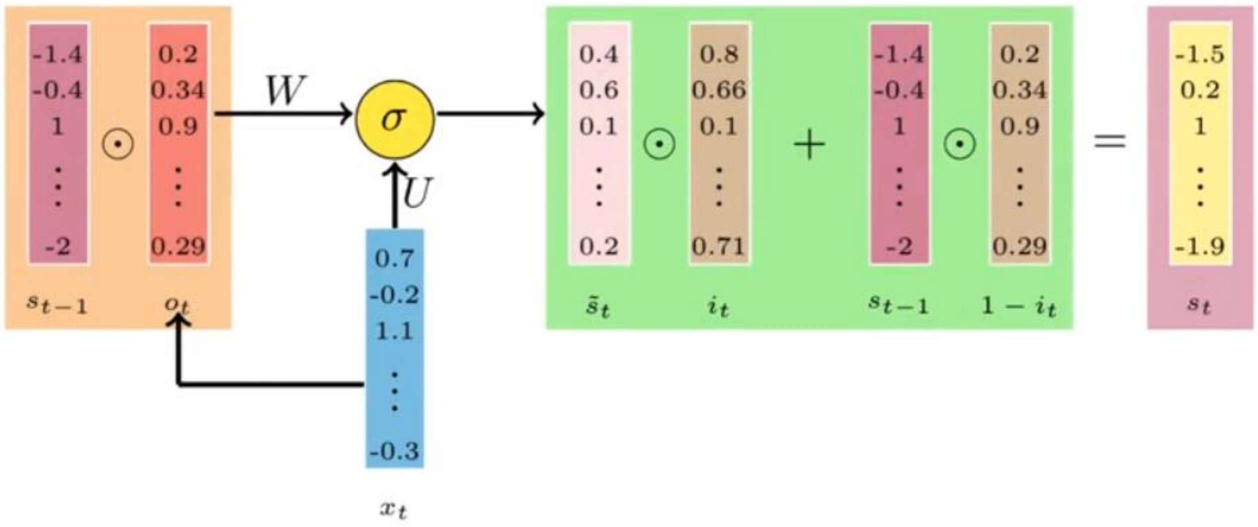
Gates:

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o)$$
$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i)$$
$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f)$$

States:

$$\tilde{s}_t = \sigma(W h_{t-1} + U x_t + b)$$
$$s_t = f_t \odot s_{t-1} + i_t \odot \tilde{s}_t$$
$$h_t = o_t \odot \sigma(s_t)$$

Long Short Term Memory Cells



Gates:

$$o_t = \sigma(W_o s_{t-1} + U_o x_t + b_o)$$

$$i_t = \sigma(W_i s_{t-1} + U_i x_t + b_i)$$

States:

$$\tilde{s}_t = \sigma(W(o_t \odot s_{t-1}) + U x_t + b)$$

$$s_t = (1 - i_t) \odot s_{t-1} + i_t \odot \tilde{s}_t$$

Gated Recurrent Units (Fewer Gates)