AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

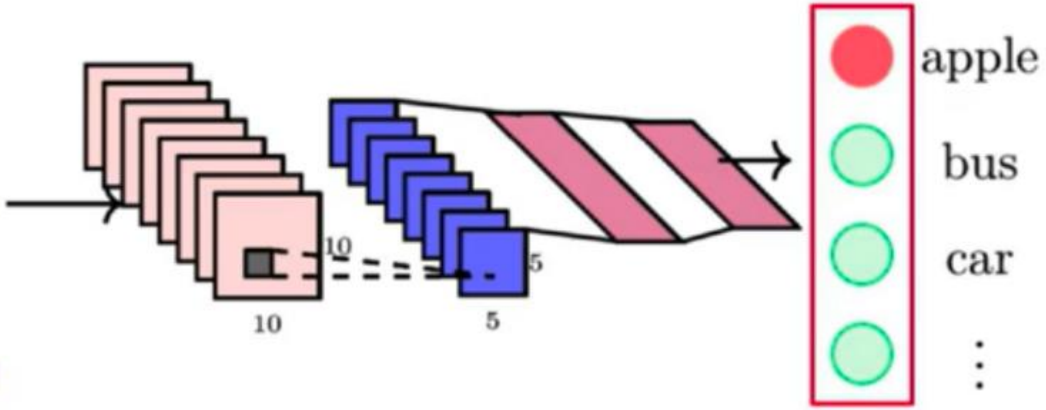
Introduction to Deep Learning

Sequential Models

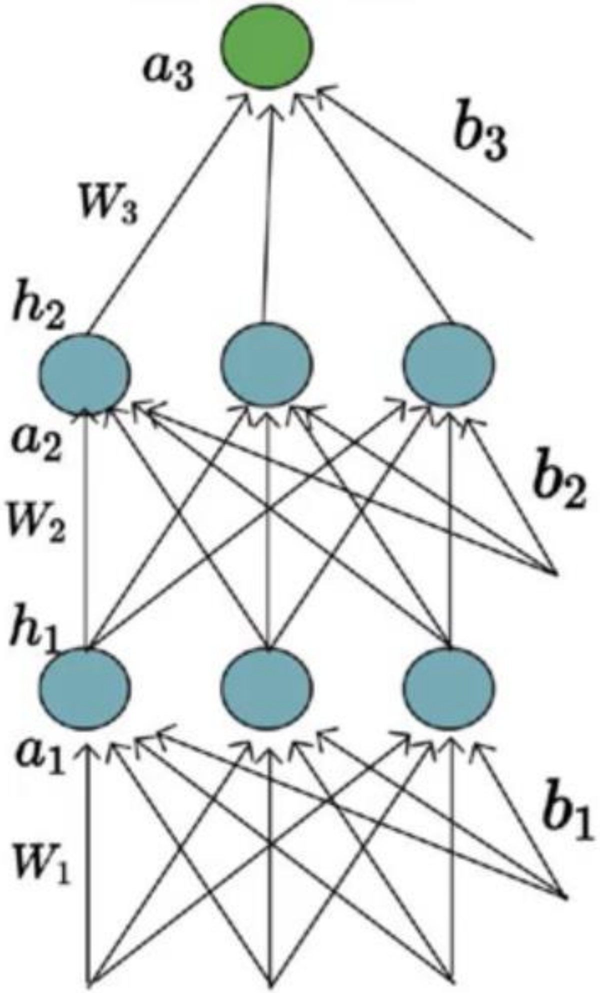
- Recurrent Neural Network(RNN)
- Long Short-Term Memory Cells(LSTM)
- Gated Recurrent Unit (GRU)

- **Recurrent Neural Network(RNN)**

Introduction

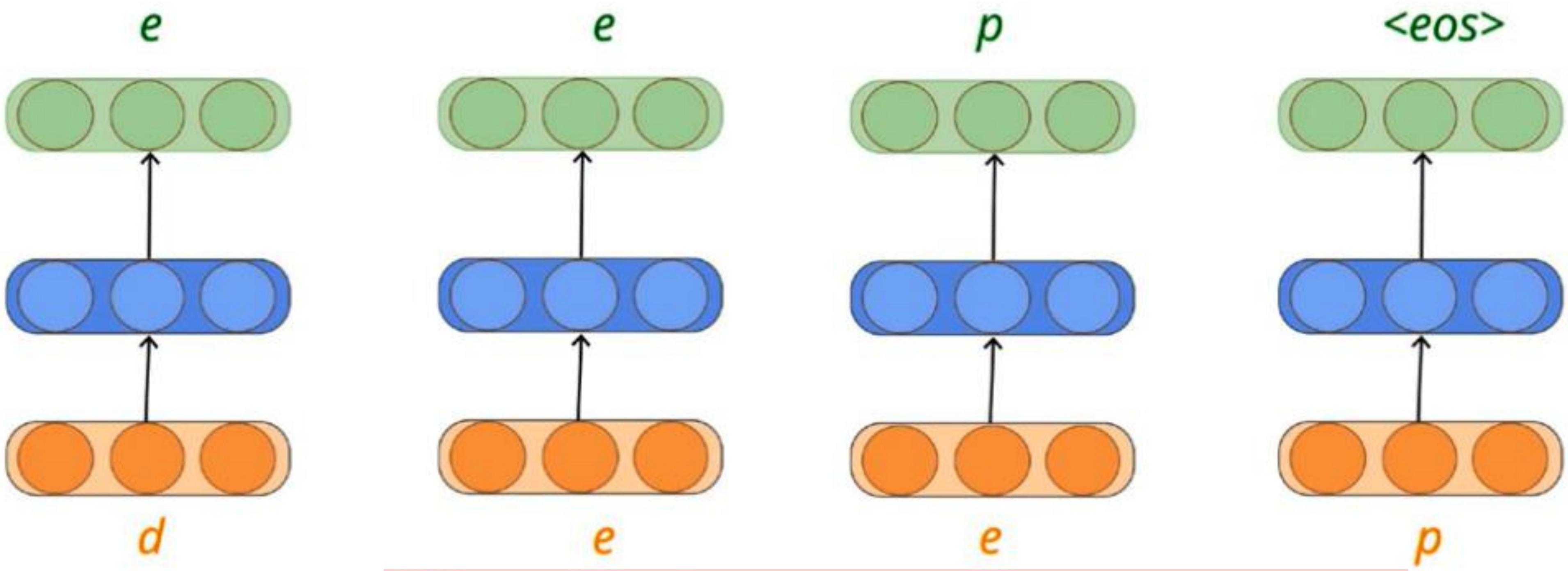


- Outputs are independent of previous inputs
- Input is of a fixed length



x1 height weight ... . sugar bp ECG
x2 height weight ... . sugar bp ECG

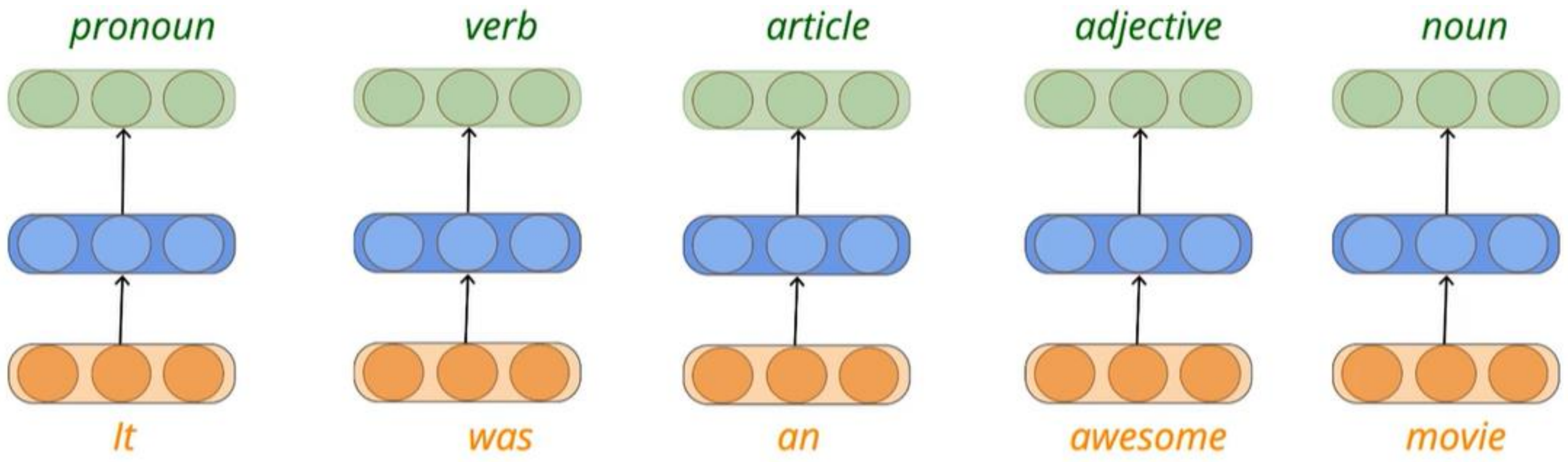
Sequence Learning Problems



- Outputs depend on previous inputs also
- The length of the input is not fixed

Sequence Learning Problems

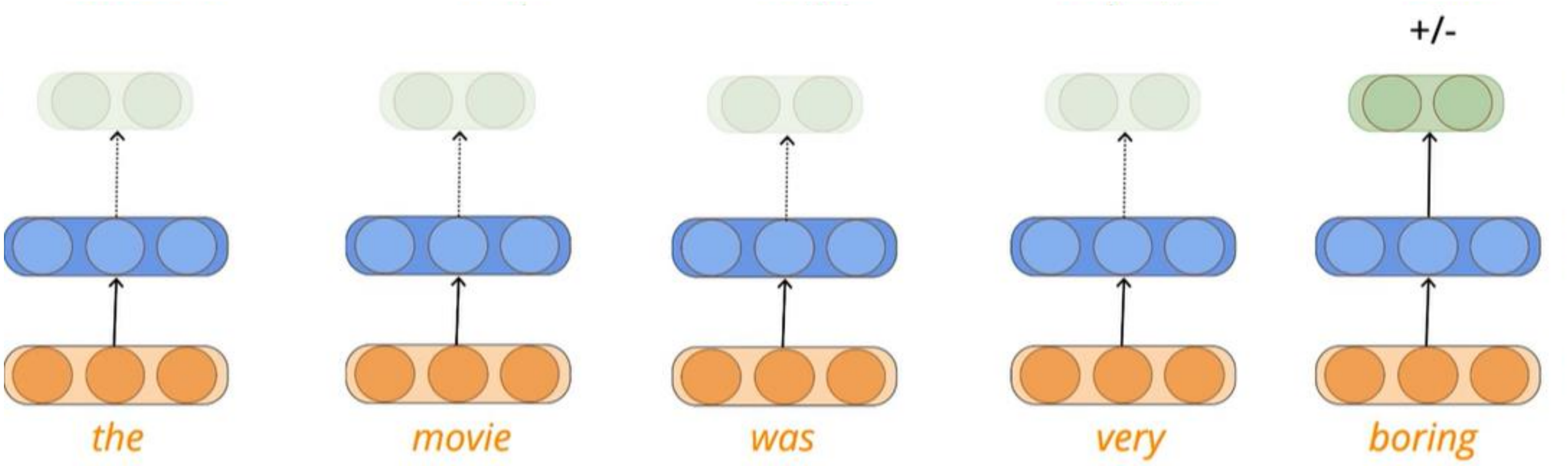
Sequence of words



1 hot encoding used to convert input to number

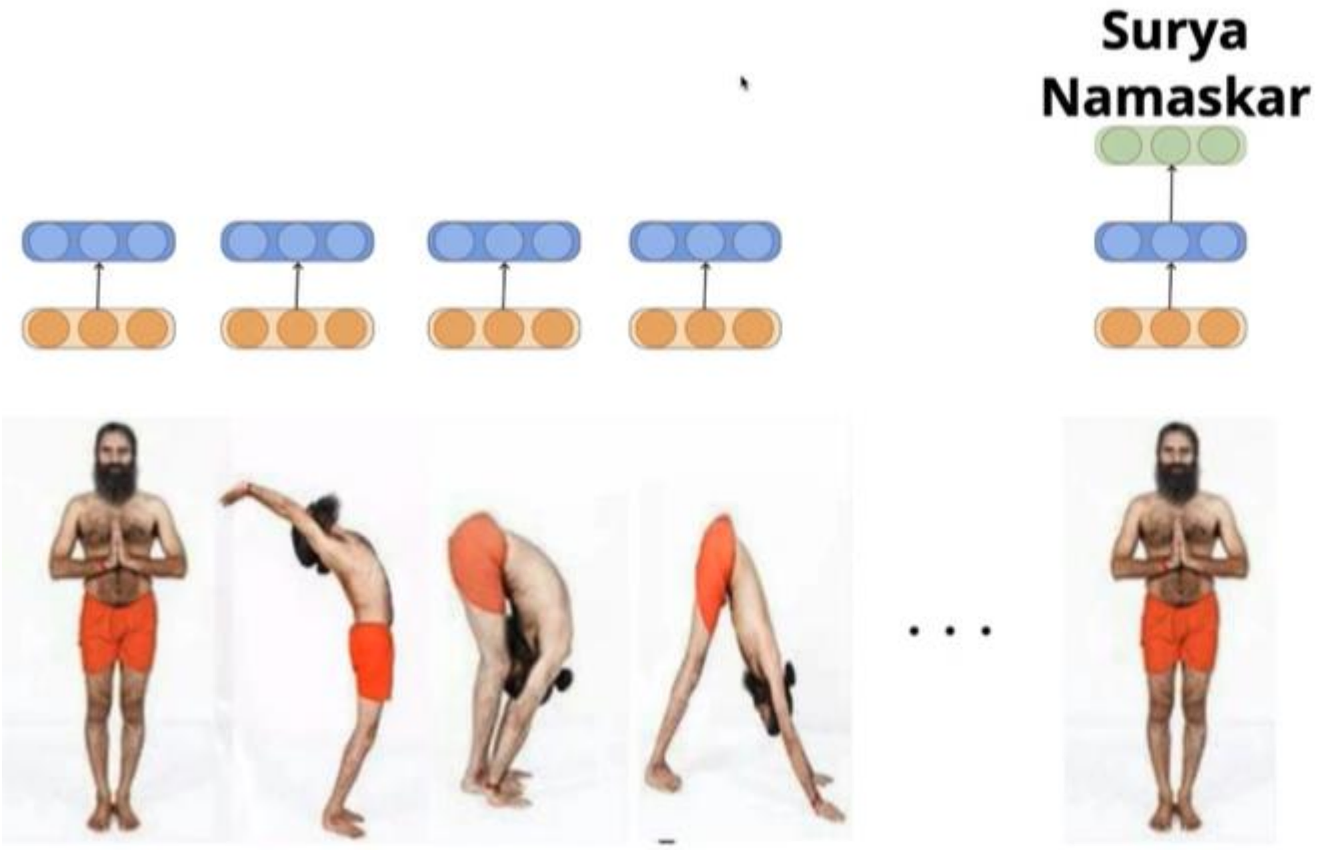
Awesome being an adjective has helped in identifying **movie** as noun
Some words like **bank**- may be used as noun or verb. So the context matters

Sequence Learning Problems- Predict the polarity of the sentence



The no: of input and output may not be same. Here the input is a sequence of words and output is single which classifies the sentence polarity- positive or negative.

Other sequence learning problems



Classify the sequence of yoga posters as one Yoga exercise name - Input is a sequence of frames, and output is a single Yoga exercise name. challenges- Variable no of frames based on speed of action

Speech

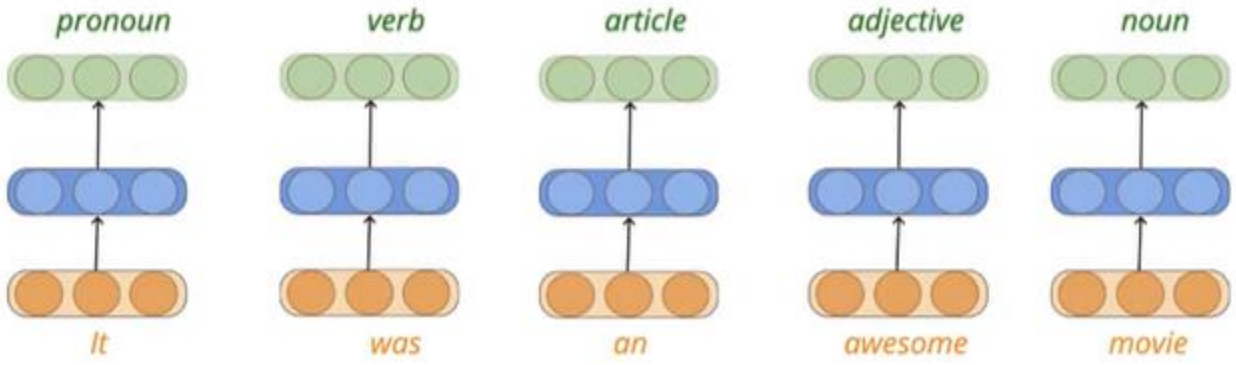
Speech Processing- another sequence learning problem. Take audio signals as input and classify each of them as phonemes

Video

Yoga Video classification :. Each frame in the video correspond to a pose and we want to classify each of the frames into one pose resulting in a sequence of poses

What can be a solution?

- Ensure that y_t is dependent on previous inputs also
- Ensure that the function can deal with variable number of inputs
- **Ensure that the function executed at each time step is the same**



$$h_i = \sigma(W_1 x_i + b_1)$$

$$y_i = O(W_2 h_i + b_2)$$

$$i = timestep$$

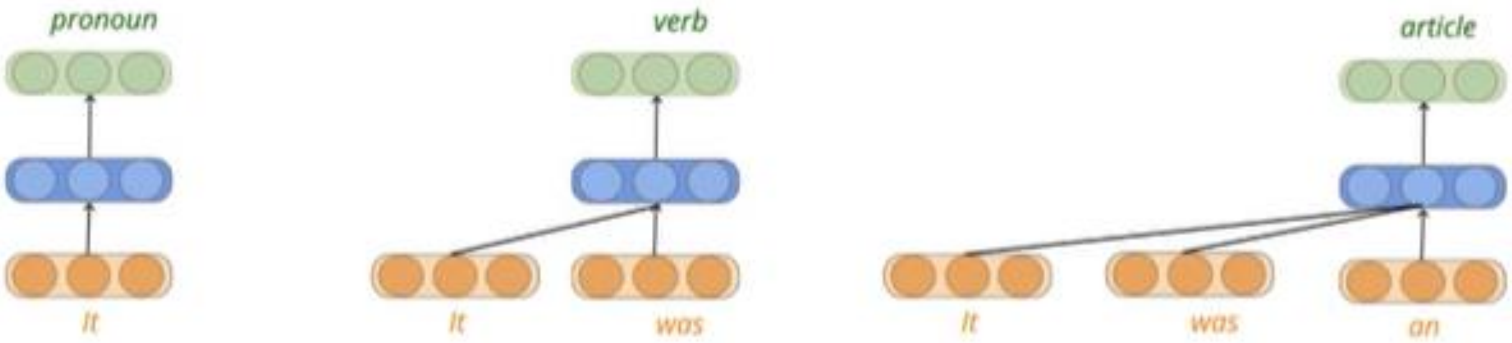
$$s_i = \sigma(U x_i + b)$$

$$y_i = O(V s_i + c)$$

Parameter Sharing

$$y_t = \hat{f}(x_1, x_2, ..., x_t)$$

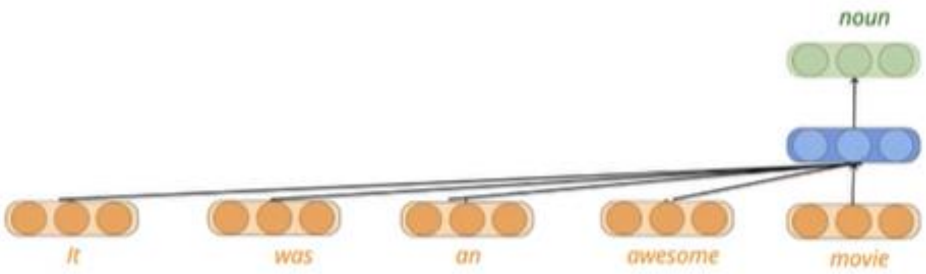
This solution also does not satisfy all 3 criterias



$$s_i = \sigma(Ux_i + b)$$

$$y_i = O(Vs_i + c)$$

✓ **Parameter Sharing**



- ✓ Ensure that y_t is dependent on previous inputs also
- ✗ Ensure that the function can deal with variable number of inputs
- ✗ Ensure that the function executed at each time step is the same

$$y_1 = f(x_1)$$

$$y_2 = f(x_1, x_2)$$

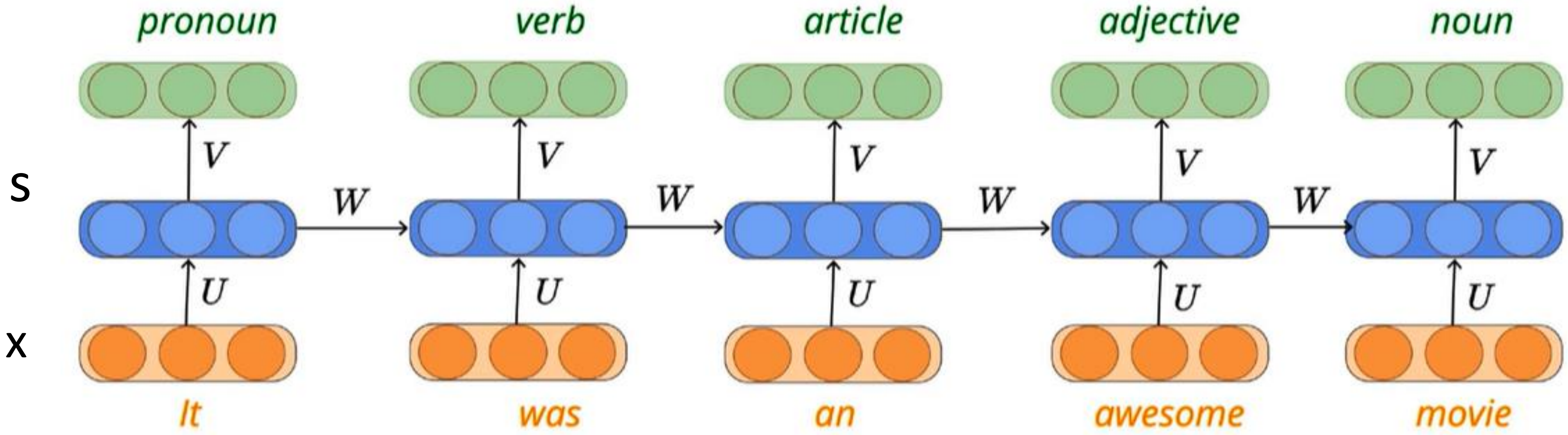
$$y_3 = f(x_1, x_2, x_3)$$

$$y_n = f(x_1, x_2, x_3, \ldots, x_n)$$

Still this solution does not satisfy all the 3 criterias

A solution – Recurrent Neural Networks (RNN)

RNN satisfies all the 3 criterias



$$s_i = \sigma(Ux_i + Ws_{i-1} + b)$$

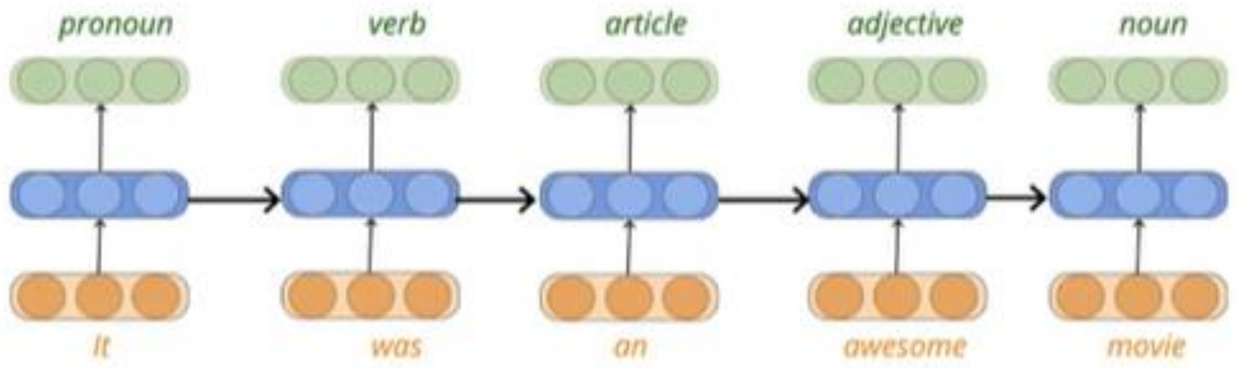
$$y_i = O(Vs_i + c)$$

$$y_i = \hat{f}(x_i, s_{i-1}, W, U, V, b, c)$$

RNN – Types of problems

$$y_i = \hat{f}(x_i, s_{i-1}, W, U, V, b, c)$$

- How do you represent words and characters as numbers ? **(data and tasks)**
- What is an appropriate loss function ? **(loss)**
- How do you train the model ? **(learning algorithm)**



No of inputs	No of Outputs	Appln
n	**1**	**Classification**
n	**n**	**Parts of Speech tagging**
n	**m**	**Machine translation**

- **Sequence Classification** (sentiment classification, video classification)
- **Sequence Labelling** (part of speech tagging, named entity recognition)
- **Sequence Generation** (machine translation, transliteration)

Data and Task

<sos> start of sequence- to indicate that sentences is starting

<eos> end of sequence- to indicate that sentences is ending. Sometimes sentence end with.,?! Or nothing. Hence we give this

<pad> artificial word to make sure all sentences of equal length

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
<sos>	The	first	half	was	very	boring	.	<eos>	<pad>
<sos>	Great	performance	by	all	the	lead	actors	.	<eos>
<sos>	The	bacground	music	was	awesome	.	<eos>	<pad>	<pad>
<sos>	The	movie	was	a	waste	of	time	.	<eos>

y
0
1
1
0

- lower case all words
- compute the total number of unique words across all sentences (say, L --> 24 in the above case)
- Assign a unique id to each word (between 1 to L)
- Represent each word using a L dimensional binary vector with only the bit corresponding to the word id set to 1

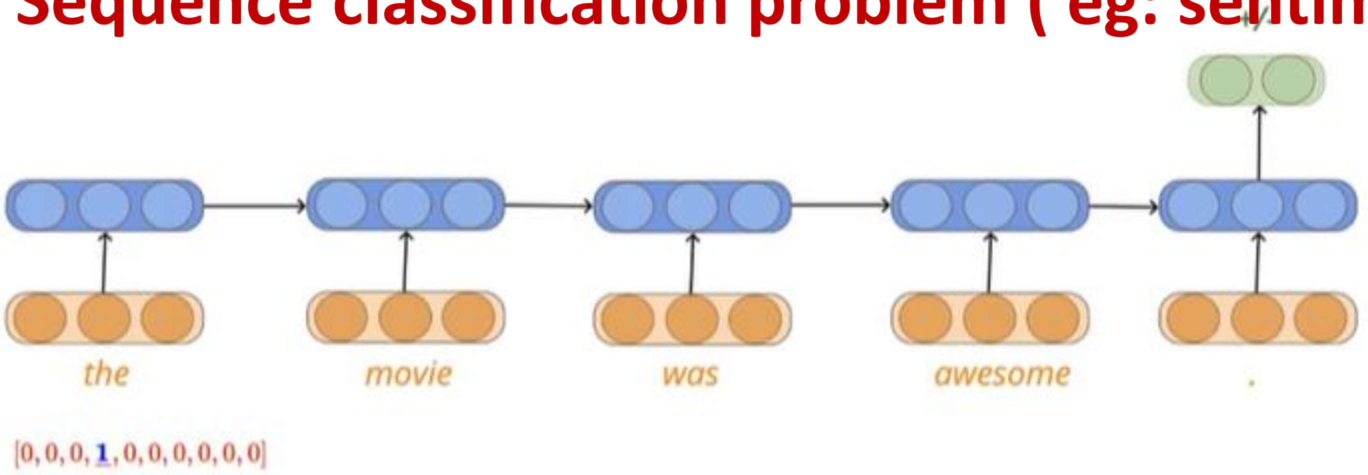
word	id
<sos>	1
<eos>	2
<pad>	3
the	4
first	5
half	6
...	
...	
time	24

1-hot vector representation

[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

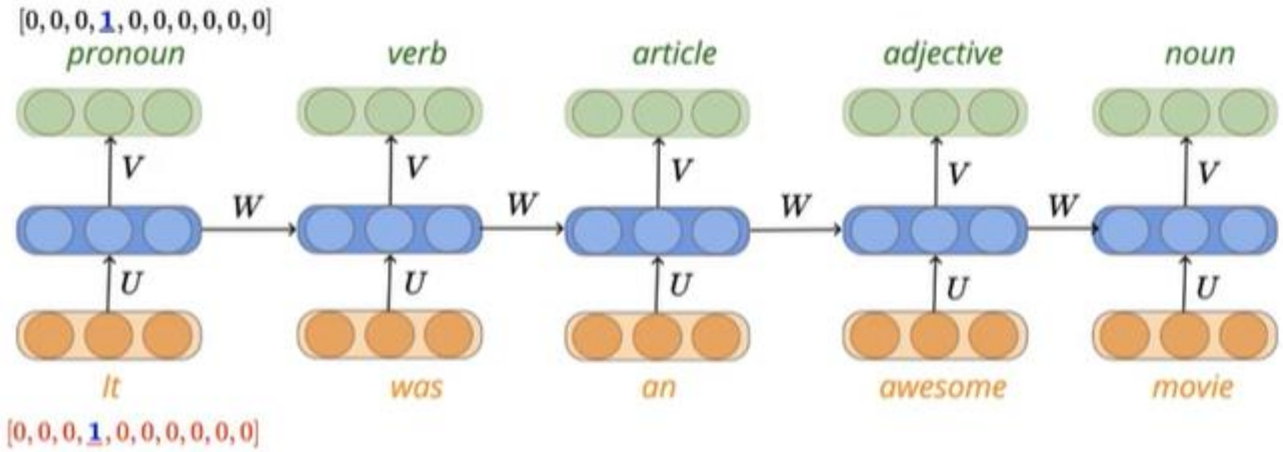
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Sequence classification problem (eg: sentiment analysis-Polarity)



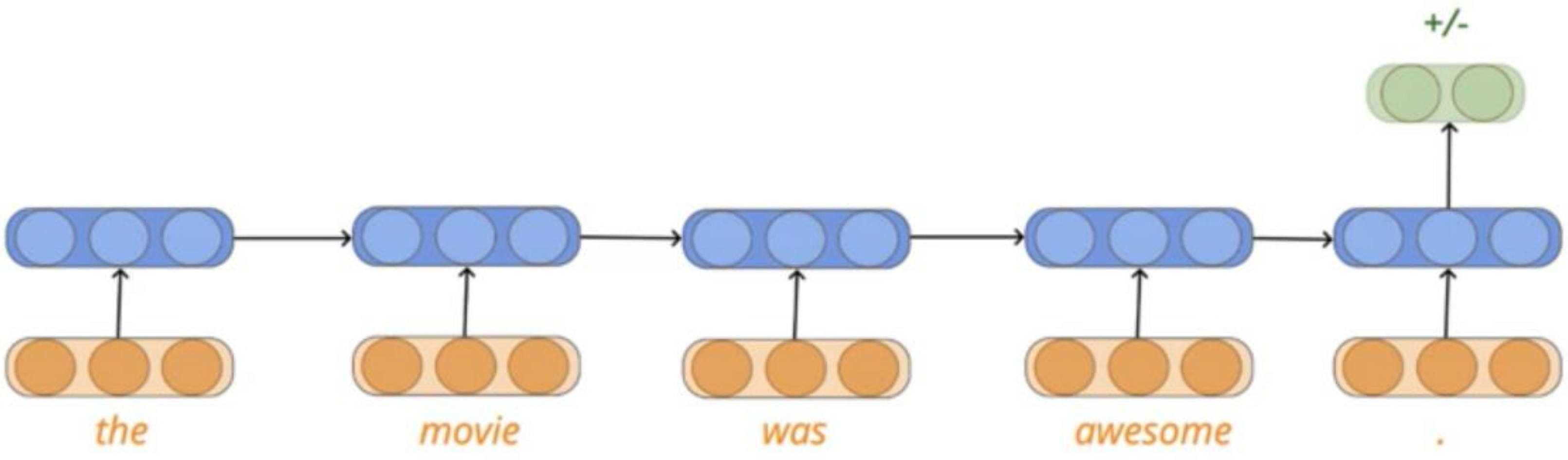
$$s_i = \sigma(Ux_i + Ws_{i-1} + b)$$
$$\hat{y} = O(Vs_T + c)$$
$$T = sequence\ length$$

Sequence Labelling problem (eg: parts of speech tagging)



$$s_i = \sigma(Ux_i + Ws_{i-1} + b)$$
$$\hat{y}_i = O(Vs_i + c)$$

Loss function for sequence classification problem

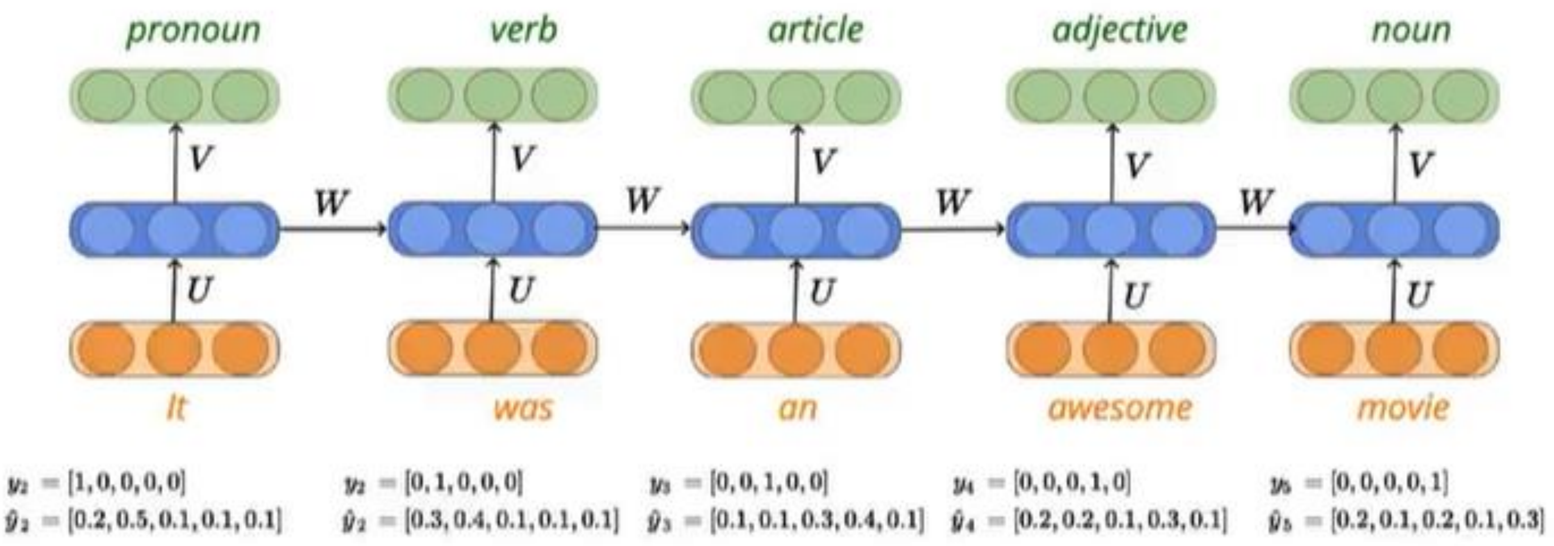


$$\mathscr{L}(\theta) = -\sum_{i=0}^{1} y[i] \log \hat{y}[i]$$
$$= -\log y_c$$
$$= -\log 0.7$$

$$y = [1, 0]$$
$$\hat{y} = [0.7, 0.3]$$

Only one output at the end
Possible output classes [1,0]
Y[i]=0 for one hence -log yc

Loss function for sequence labelling problem



$$\mathscr{L}(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{T}\log y_{ijc}$$

Every time step has an output

Sum up for each time step (T)for each data samples (m) and average

Training Algorithm – Back propagation

Initialise w, b

Iterate over data:

$compute\ \hat{y}$

$compute\ \mathscr{L}(w, b)$

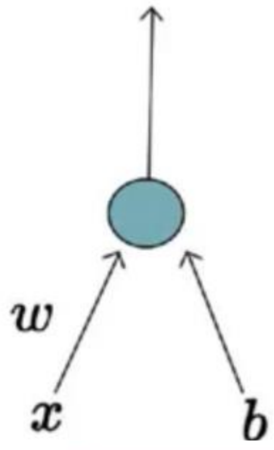
$$w_{11} = w_{11} - \eta \Delta w_{11}$$

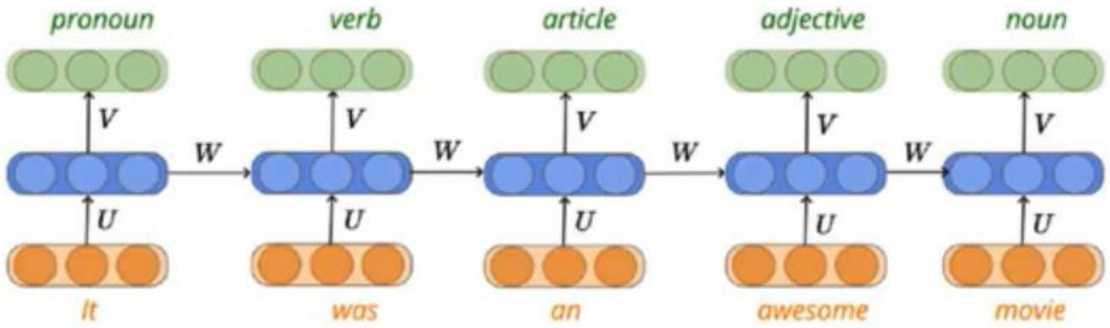
$$u_{12} = u_{12} - \eta \Delta u_{12}$$

....

$$v_{13} = v_{13} - \eta \Delta v_{13}$$

till satisfied





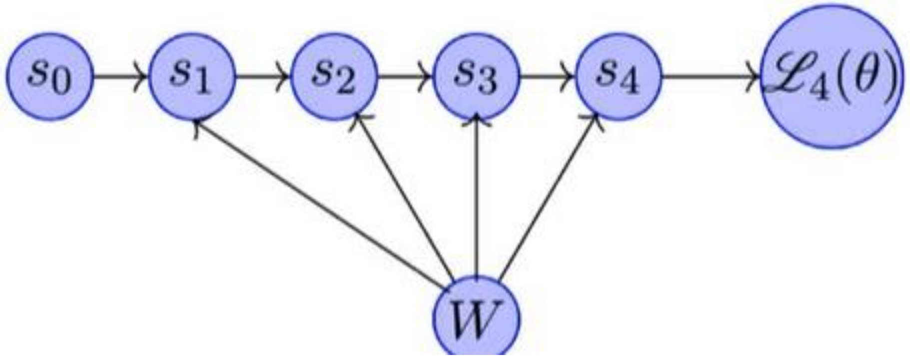
$$Earlier : w, b$$

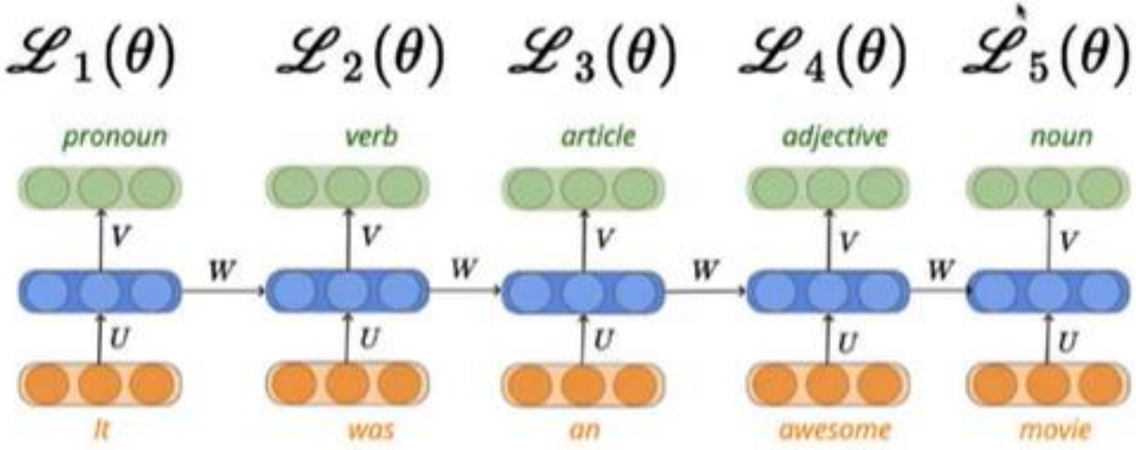
$$Now : w_{11}, w_{12}, ..., u_{11}, u_{12}, ..., v_{11}, v_{12}$$

$$Earlier : L(w, b)$$

$$Now : L(W, U, V)$$

Learning Algorithm (Derivative of loss function w.r.t w)





$$\frac{\partial \mathscr{L}_4(\theta)}{\partial W} = \frac{\partial \mathscr{L}_4(\theta)}{\partial s_4}\frac{\partial s_4}{\partial W}$$

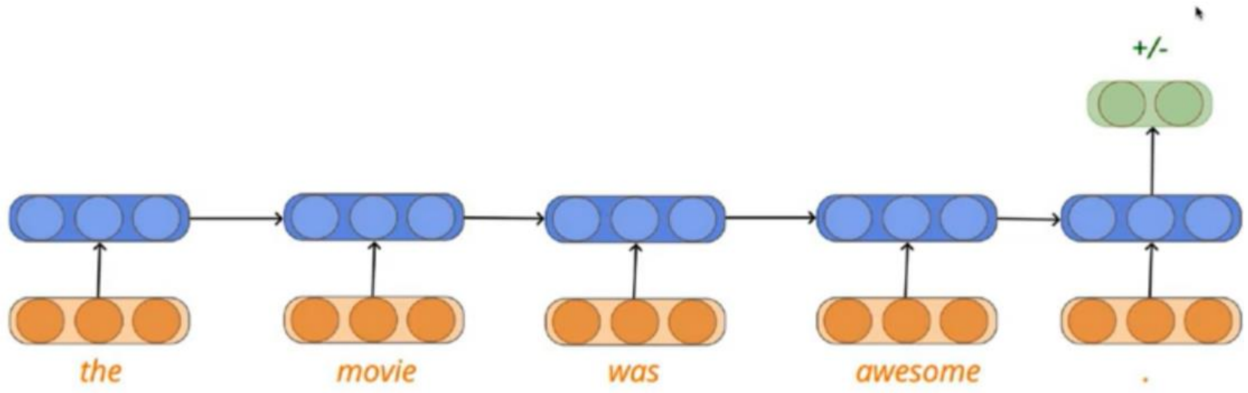
$$\frac{\partial s_4}{\partial W} = \frac{\partial s_4}{\partial W} + \frac{\partial s_4}{\partial s_3}\frac{\partial s_3}{\partial W} + \frac{\partial s_4}{\partial s_3}\frac{\partial s_3}{\partial s_2}\frac{\partial s_2}{\partial W} + \frac{\partial s_4}{\partial s_3}\frac{\partial s_3}{\partial s_2}\frac{\partial s_2}{\partial s_1}\frac{\partial s_1}{\partial W}$$

$$\frac{\partial s_4}{\partial W} = \frac{\partial s_4}{\partial s_4}\frac{\partial s_4}{\partial W} + \frac{\partial s_4}{\partial s_3}\frac{\partial s_3}{\partial W} + \frac{\partial s_4}{\partial s_2}\frac{\partial s_2}{\partial W} + \frac{\partial s_4}{\partial s_1}\frac{\partial s_1}{\partial W}$$

$$\frac{\partial s_4}{\partial W} = \sum_{k=1}^{4}\frac{\partial s_4}{\partial s_k}\frac{\partial s_k}{\partial W}$$

Similarly derivative w.r.t V and U

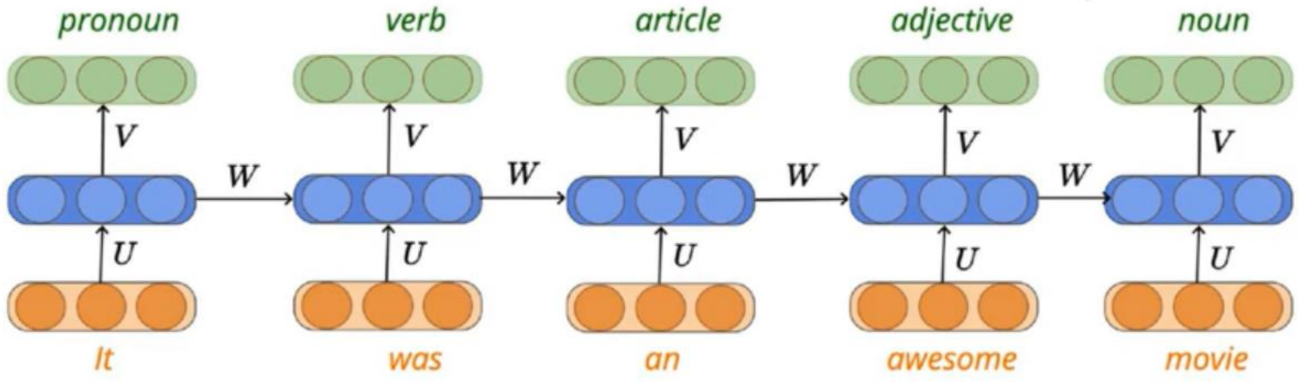
Evaluation- Sequence classification



$$\textbf{Accuracy} = \frac{No:of\ correctly\ classified}{Total\ samples}$$

Predicted y cap	Ground Truth y	Correct/incorrect
1(P)	1(P)	correct
0(N)	0(N)	incorrect
1(P)	0(N)	incorrect
0(N)	0(N)	correct
1(P)	1(P)	correct
0(N)	1(P)	incorrect

Evaluation Sequence labelling



Overall accuracy /Accuracy per class

Data sample	Pronoun	verb	article	adjective	noun
D1	P	V	Ar	Ad	N
D2					
D3					
D4					
D5					

- **Overall Accuracy**
- **Accuracy Per Class**
- **Confusion Matrix**

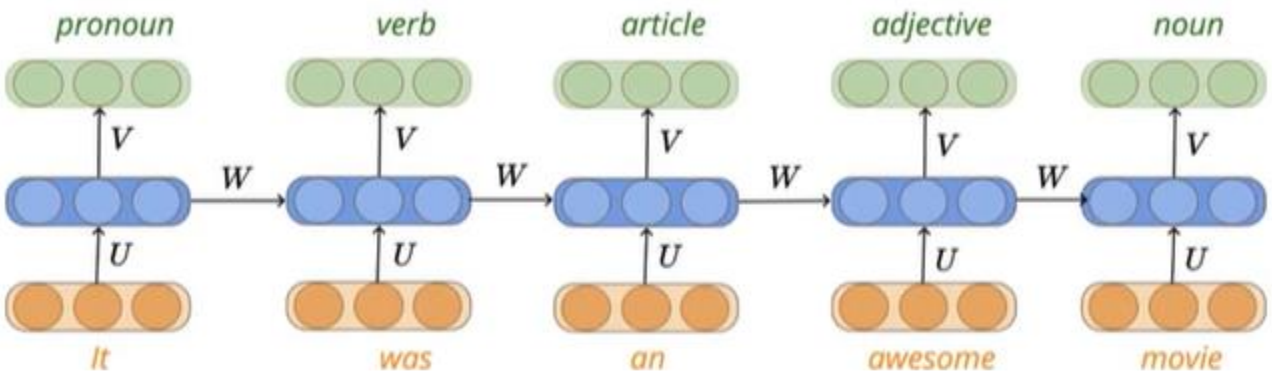
Confusion Matrix

	Pronoun	verb	article	adjective	noun
Pronoun	3	2	3	5	6
verb	2	7			
article			3		
adjective				2	
noun					1

- **Long Short-Term Memory Cells(LSTM)**

RNN not dealing with longer sequence
(How the state record information when the sequence is very long)

RNN- Recap

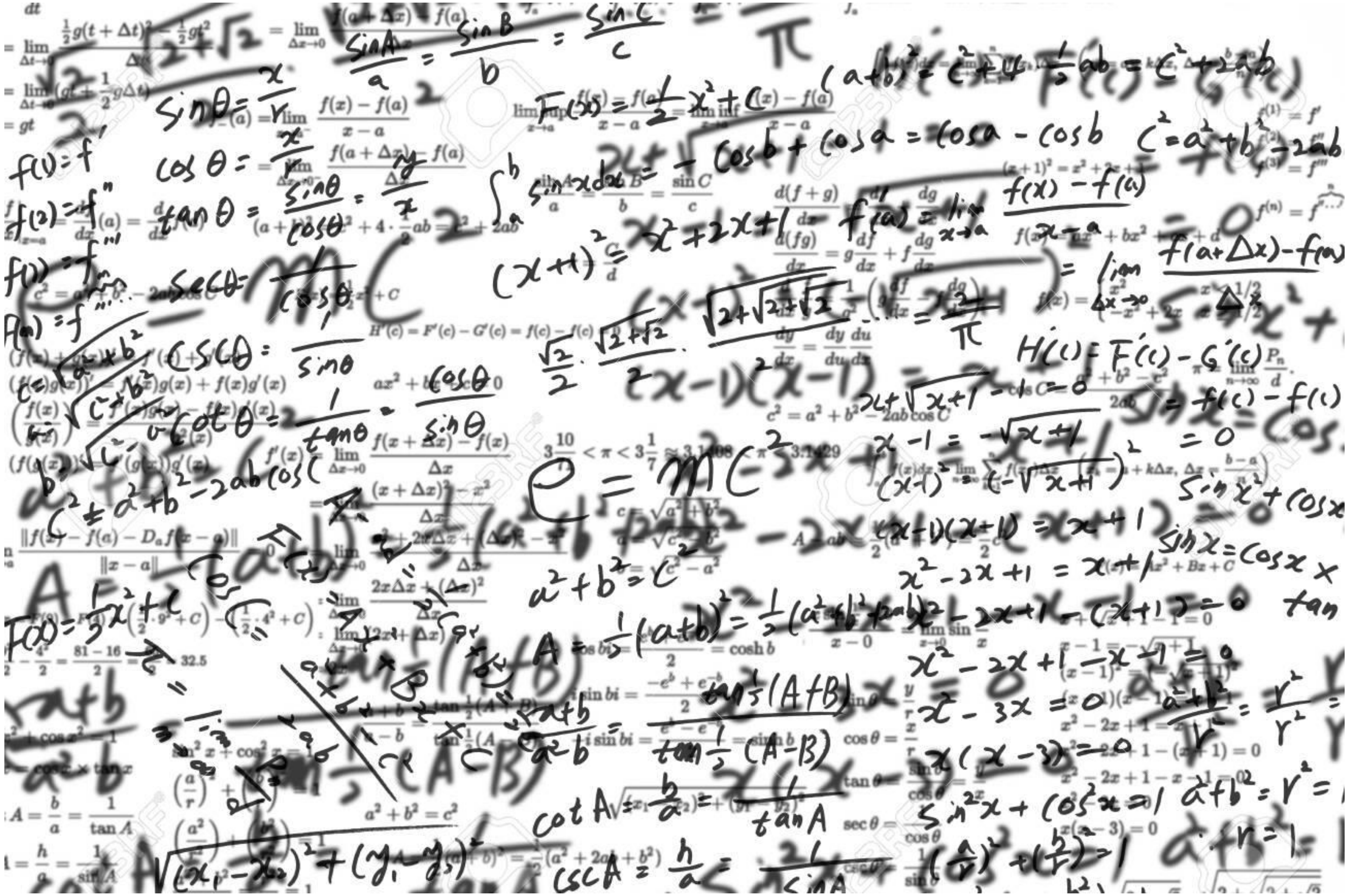


RNN: Exploding and vanishing gradient problem occurs!!
Not suitable for longer sequence

- 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

$$s_i = \sigma(Ux_i + Ws_{i-1} + b)$$
$$y_i = O(Vs_i + c)$$

White board Analogy (over time white board become so messy and u cant make out anything)



White board 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

White board 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. $ac(bd + a) + ad$

$$ac = 5$$

$$bd + a = 34$$

Strategy

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

White board 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

White board 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. $ac(bd + a) + ad$

$$ac = 5$$

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

$$ad = 11$$

Strategy

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

White board 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. $ac(bd + a) + ad$

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

$$ad = 11$$

Strategy

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

White board 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. $ac(bd + a) + ad$

$$ad + ac(bd + a) = 181$$

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

$$ad = 11$$

Strategy

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

White board Analogy

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

Compute $ac(bd + a) + ad$

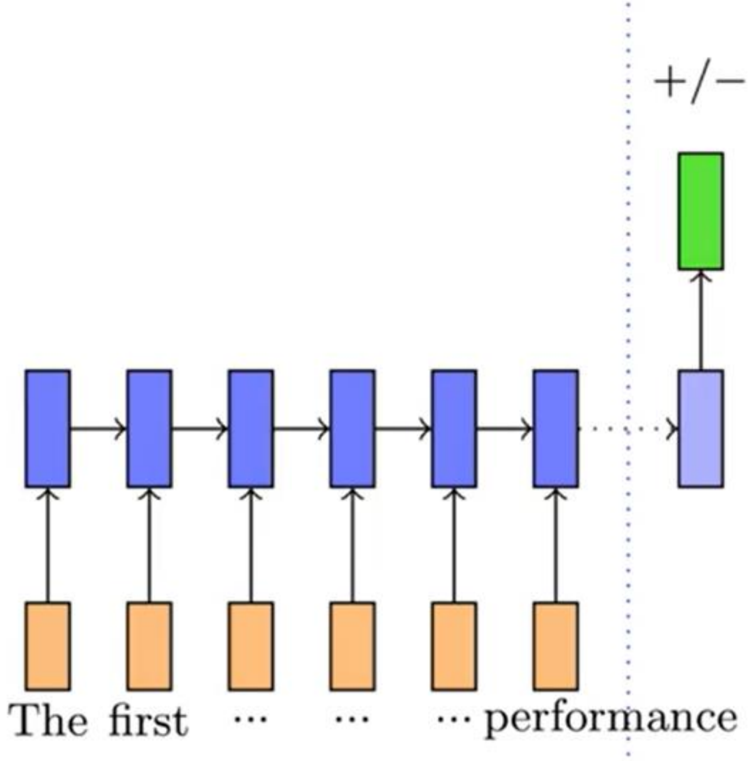
1. ac
2. bd
3. $bd + a$
4. $ac(bd + a)$
5. ad
6. $ac(bd + a) + ad$

$$ad + ac(bd + a) = 181$$

Strategy

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

Dealing with longer sequence
(An example where RNN need to selectively read write and forget)

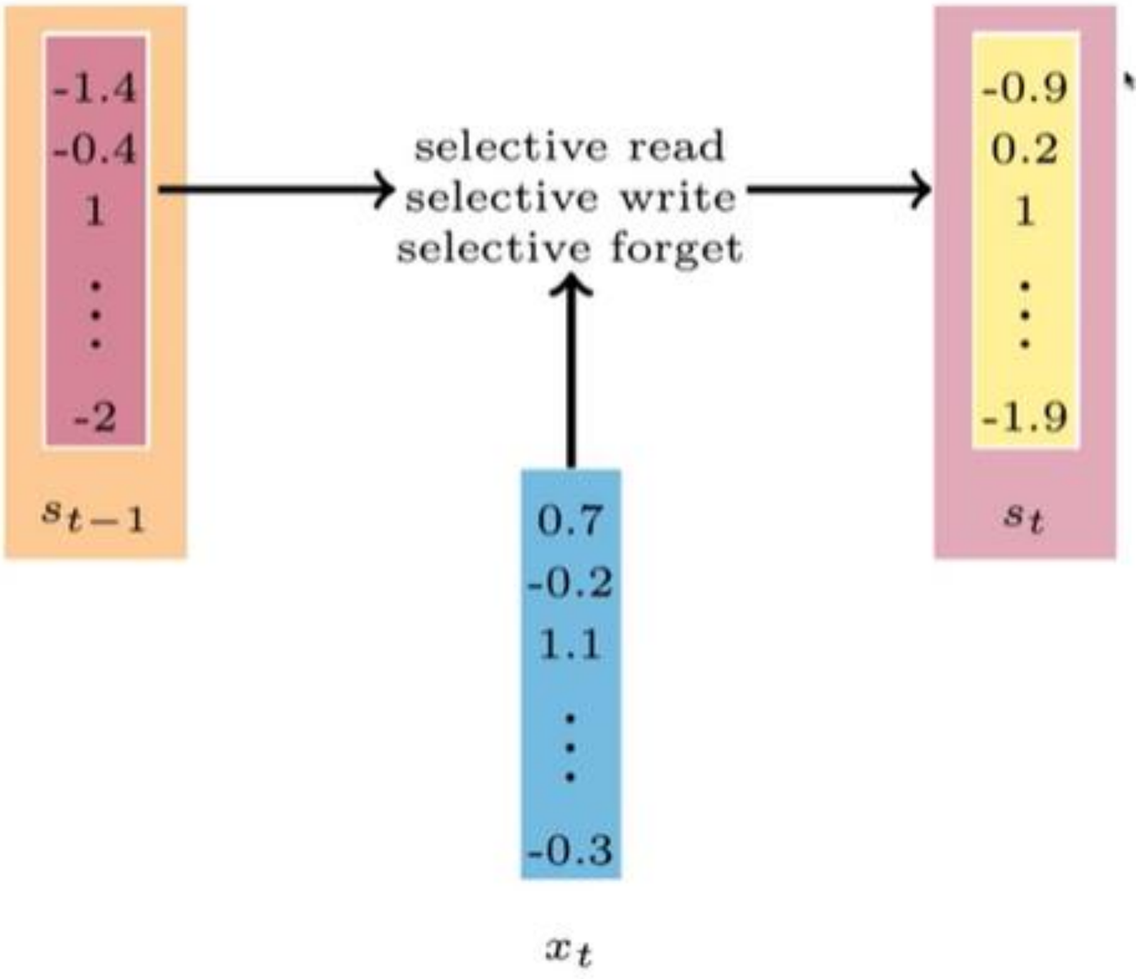


Review: The first half of the movie was dry but the second half really picked up pace. The lead actor delivered an amazing performance

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

Long Short-Term Memory Cells – deals with longer sequences
(How to implement selective read write and Forget)



LSTM

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

Recap- RNN

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

Selective write

o_t is called the **output** gate

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

$$\underbrace{\underset{s_{t-1}}{\begin{bmatrix} -1.4 \\ -0.4 \\ 1 \\ \vdots \\ -2 \end{bmatrix}} \odot \underset{o_{t-1}}{\begin{bmatrix} 0.2 \\ 0.34 \\ 0.9 \\ \vdots \\ 0.29 \end{bmatrix}} = \underset{h_{t-1}}{\begin{bmatrix} \\ \\ \\ \\ \end{bmatrix}}}_{\text{selective write}} \qquad \underset{x_t}{\begin{bmatrix} 0.7 \\ -0.2 \\ 1.1 \\ \vdots \\ -0.3 \end{bmatrix}} \qquad \underset{s_t}{\begin{bmatrix} -1.4 \\ -0.4 \\ 1 \\ \vdots \\ -2 \end{bmatrix}}$$

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

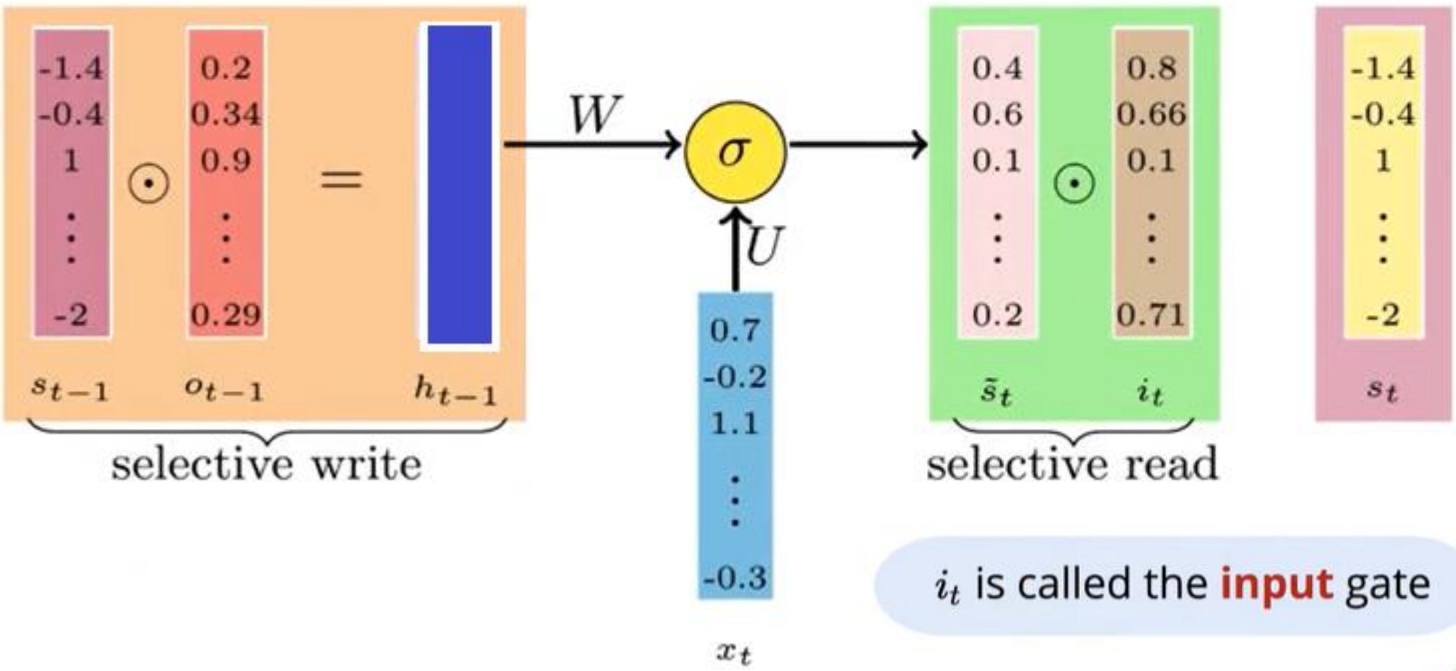
- instead of passing s_{t-1} as it is to s_t we want to pass (write) only some portions of it to the next state
- A reasonable way of doing this would be to assign a value between 0 and 1 which determines what fraction of the current state to pass on to the next state

Selective Read

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

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

$$= \tilde{s}_t \odot i_t$$



i_t is called the **input** gate

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:

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

Input gate:

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

Selectively Read:

$i_t \odot \tilde{s}_t$

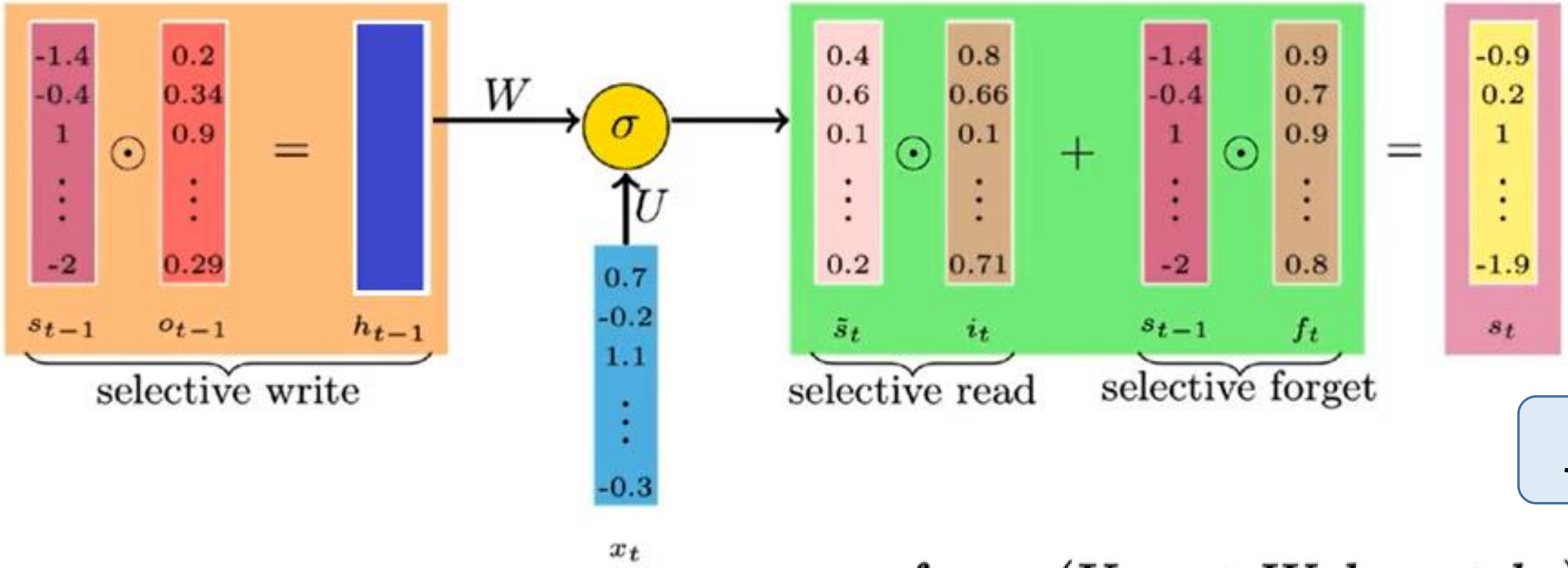
- $\tilde{s}_t$ thus captures all the information from the previous state h_{t-1} and the current input x_t

- However, we may not want to use all this new information and only selectively read from it before constructing the new cell

Selective Forget

Some LSTMs stop after selective read but the actual version LSTM has forget gate also

Want to make s_t depended on s_{t-1} but only the relevant portion of s_{t-1} (so forgetting some info of s_{t-1} and adding that to $\tilde{s}_t \odot i_t$)



ft **is called forget gate**

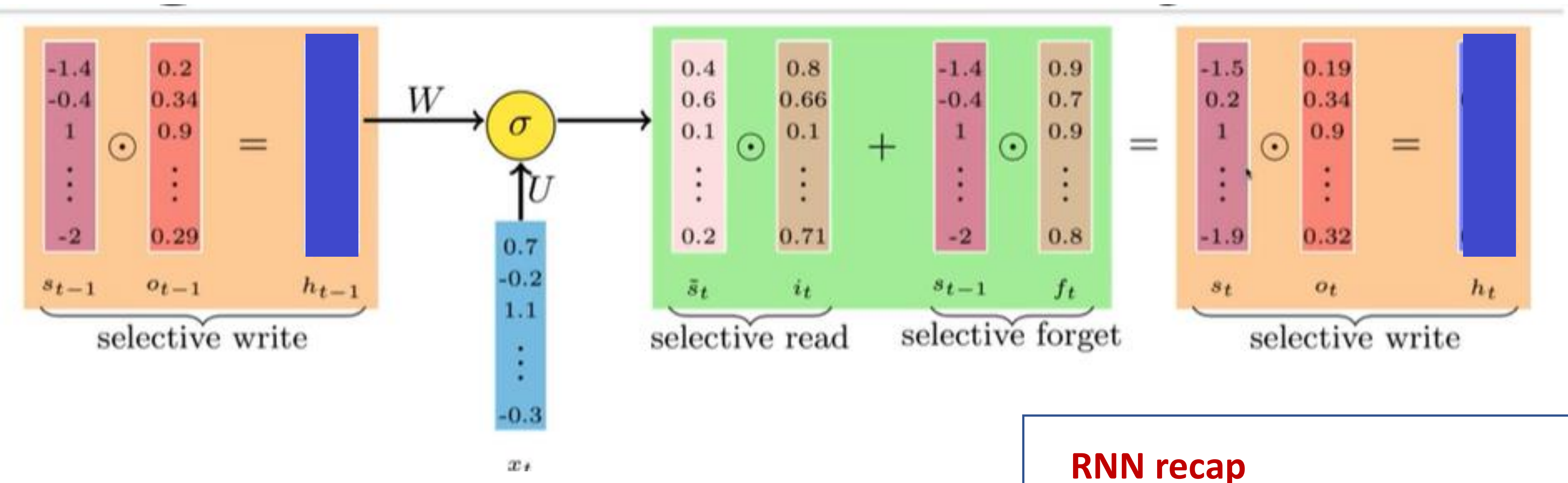
How do we combine $\tilde{s}_t$ and s_{t-1} to get the new state s_t

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

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

Summary-LSTM

LSTM



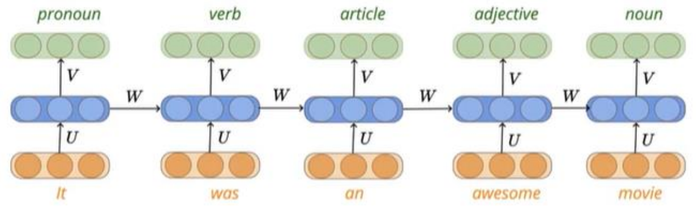
Gates:

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

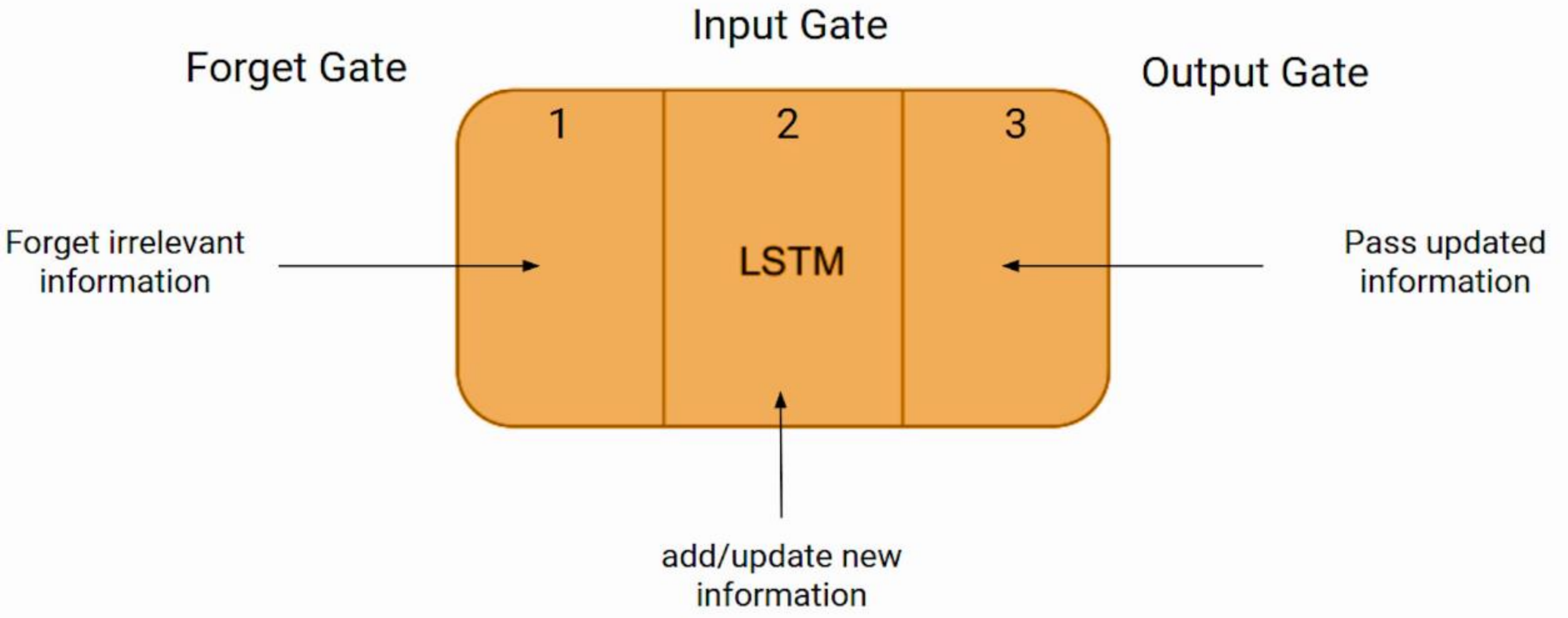
States:

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

RNN recap



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


Input Gate
Forget Gate
Output Gate
1
2
3
Forget irrelevant information
LSTM
Pass updated information
add/update new information

- **Gated Recurrent Unit (GRU)**

LSTM variants

- 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:

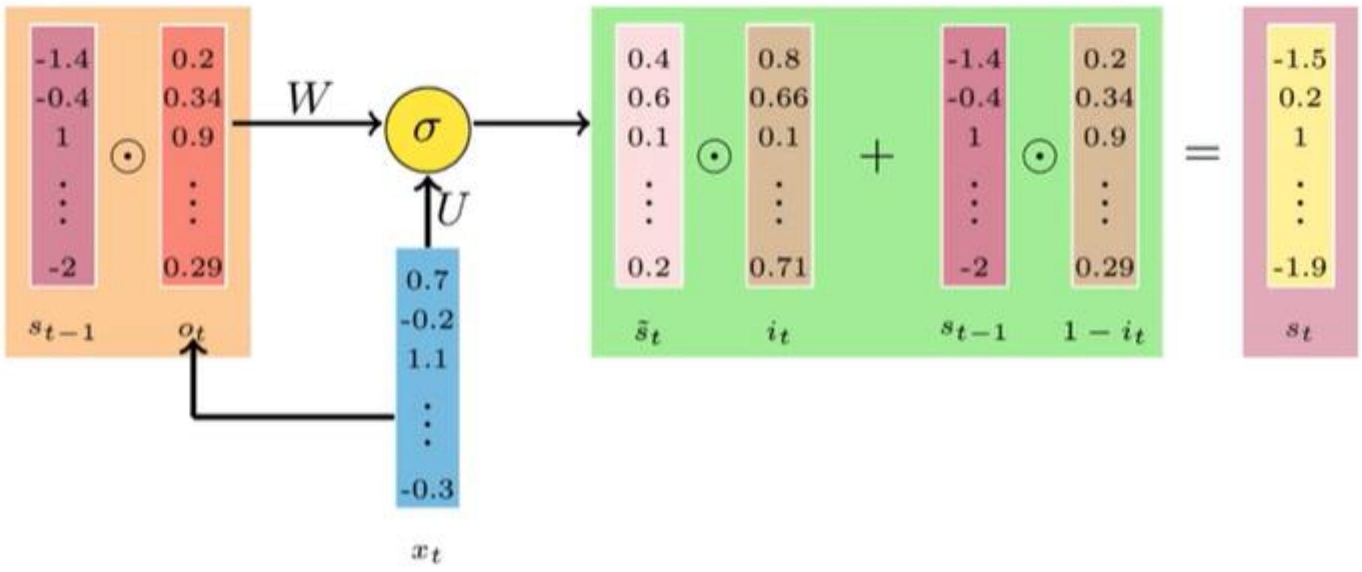
$$\tilde{s}_t = \sigma(W h_{t-1} + U x_t + b)$$
$$s_t = f_t \odot s_{t-1} + i_t \odot \tilde{s}_t$$
$$h_t = o_t \odot \sigma(s_t)$$

Gated Recurrent Units (GRU)

Recap LSTM

States:

$$\tilde{s}_t = \sigma(Wh_{t-1} + Ux_t + b)$$
$$s_t = f_t \odot s_{t-1} + i_t \odot \tilde{s}_t$$
$$h_t = o_t \odot \sigma(s_t)$$



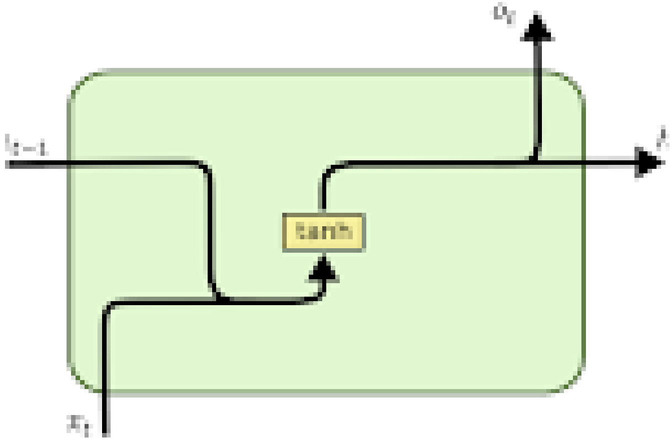
Gates:

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

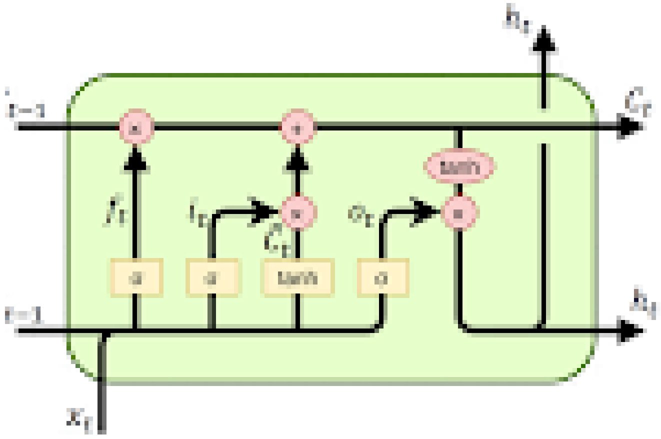
States:

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

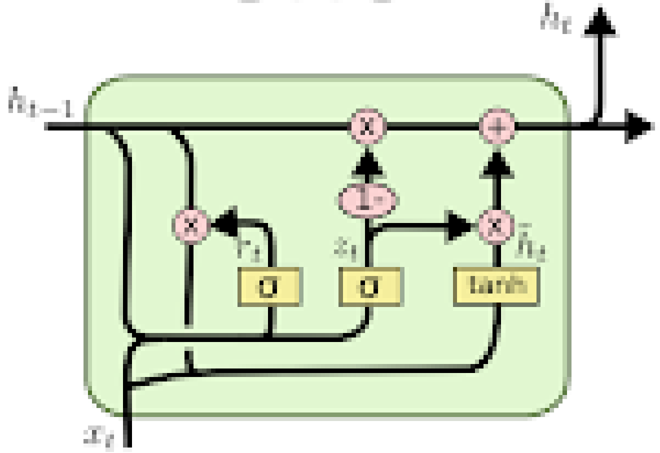
RNN



LSTM



GRU



Namah Shivaya

Courtesy : Video lectures of Dr.Mitesh Kapra