PLACEMENT REFRESHER PROGRAM

Session 14 - Deep Learning 2

RNN & LSTM

By
Ritesh Kumar Pandey

Agenda

- RNN
- LSTM

CNN - Question

upGrad

If the array output from convolution C looks like

1 2 3 4

5 6 7 8

1 2 3 4

5 7 6 8

and filter F is 2 x 2

and stride is 2.

How does the feature map looks like after the subsampling of array C with the filter F using average pooling?

CNN - Question

upGrad

If the array output from convolution C looks like

1 2 3 4
5 6 7 8
1 2 3 4
5 7 6 8

and filter F is 2 x 2
and stride is 2.

3.5	5.5
3.75	5.25

How does the feature map looks like after the subsampling of array C with the filter F using average pooling?

RNN Stands for ________.

A) Recursive Neural Network
B) Recurrent Neural Network
C) Recurring Neural Network
D) Removable Neural Network

RNN Stands for ________.

A) Recursive Neural Network
B) Recurrent Neural Network
C) Recurring Neural Network
D) Removable Neural Network

RNN remembers each and every information through_________.

A) Work
B) Time
C) Hours
D) Memory

RNN remembers each and every information through_________.

A) Work
B) Time
C) Hours
D) Memory

What is the basic concept of Recurrent Neural Network?

A) Use previous inputs to find the next output according to the training set.
B) Use a loop between inputs and outputs in order to achieve the better prediction.
C) Use recurrent features from dataset to find the best answers.
D) Use loops between the most important features to predict next output.

What is the basic concept of Recurrent Neural Network?

A) Use previous inputs to find the next output according to the training set.
B) Use a loop between inputs and outputs in order to achieve the better prediction.
C) Use recurrent features from dataset to find the best answers.
D) Use loops between the most important features to predict next output.

Example Data:

- Temperature Data
- Audio Data
- Stock Market Data

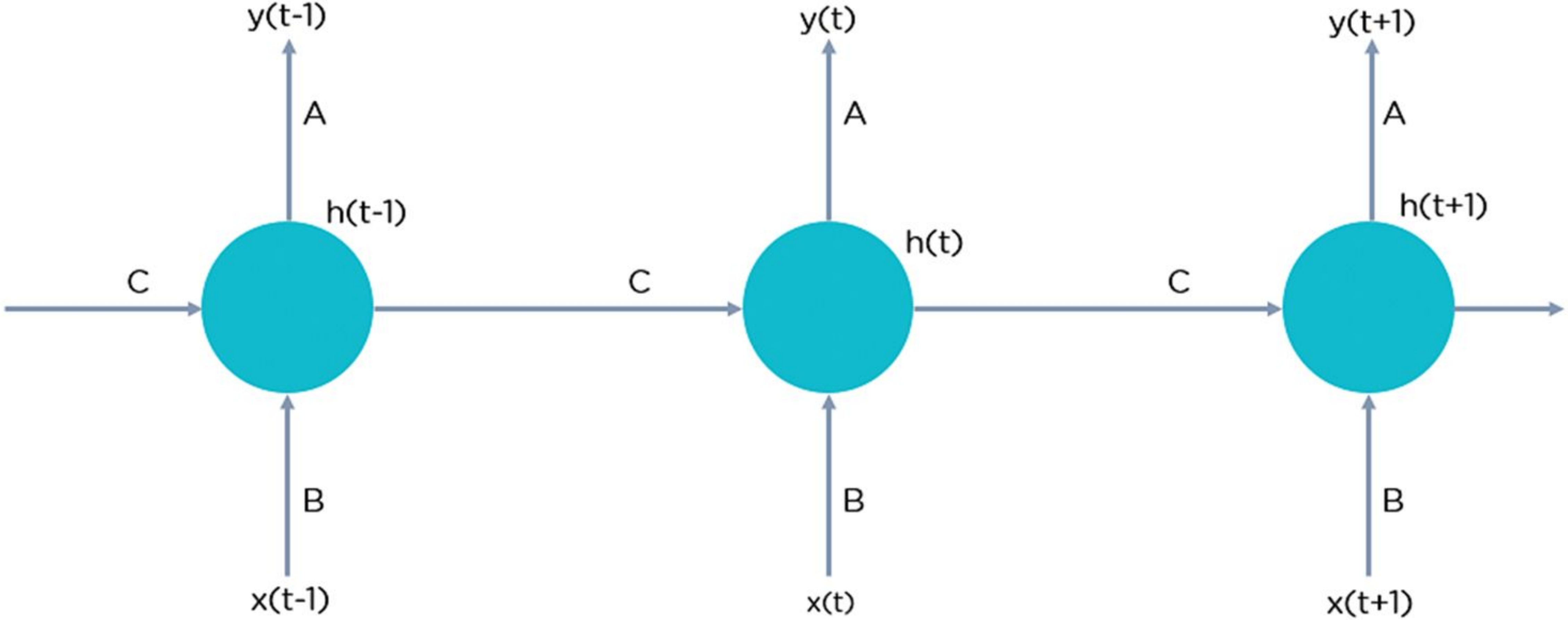
upGrad

- RNN works on the principle of saving the output of a particular layer and feeding this back to the input in order to predict the output of the layer.

- RNN were created because there were a few issues in the feed-forward neural network:
    - Cannot handle sequential data
    - Considers only the current input
    - Cannot memorize previous inputs

- An RNN can handle sequential data, accepting the current input data, and previously received inputs. RNNs can memorize previous inputs due to their internal memory.

RNN



$$h(t) = f_c\,(h(t-1),\ x(t))$$

$h(t)$ = new state
f_c = function with parameter c
$h(t-1)$ = old state
$x(t)$ = input vector at time step t

In RNN Each unit has an internal state which is called the________.

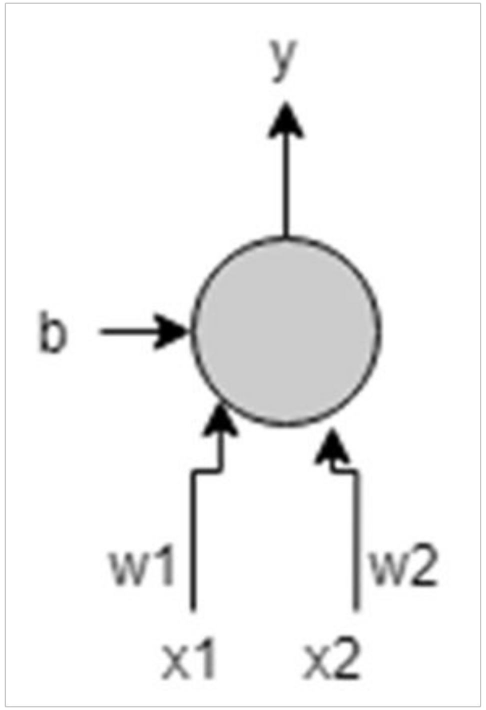
A) visible state of unit
B) hidden state of the unit.
C) Visible function
D) Hidden function

In RNN Each unit has an internal state which is called the________.

A) visible state of unit
B) hidden state of the unit
C) Visible function
D) Hidden function

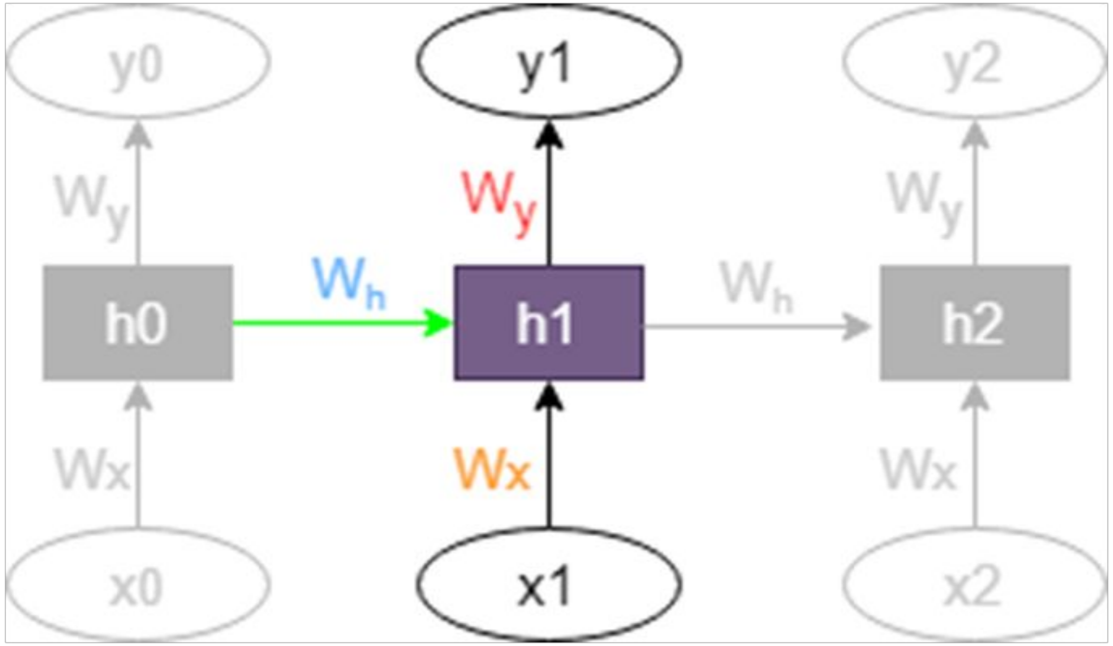
upGrad

Feed-forward Propagation



$$y = (w1.x1 + w2.x2) + b$$

Recurrent Neural Network

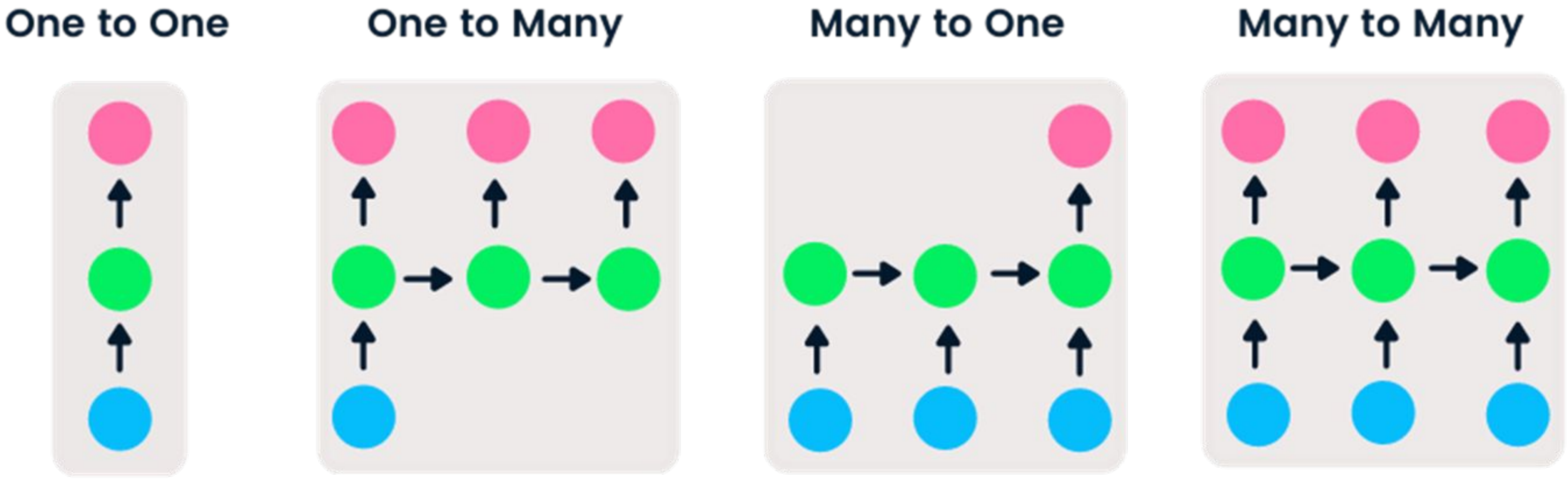


$$h1 = A(W_h * h0 + Wx * x1 + b_h) \longrightarrow A = tanh/ReLU$$

$$y1 = A(W_y * h1 + by) \longrightarrow A = Sigmoid/Softmax$$

RNN - Types

- The four types of RNN are:
  - One to One RNN
  - One to Many RNN
  - Many to One RNN
  - Many to Many RNN



What is 'gradient' when we are talking about RNN?

A) A gradient is a partial derivative with respect to its inputs
B) It is how RNN calls it´s features
C) The most important step of RNN algorithm
D) A parameter that can help you improve the algorithm’s accuracy

upGrad

What is 'gradient' when we are talking about RNN?

A) A gradient is a partial derivative with respect to its inputs
B) It is how RNN calls it´s features
C) The most important step of RNN algorhitm
D) A parameter that can help you improve the algorhitm´s accuracy

A gradient measures how much the output of a function changes, if you change the inputs a little bit. The higher the gradient, the steeper the slope and the faster a model can learn. But if the slope is zero, the model stops to learning. A gradient simply measures the change in all weights with regard to the change in error.

__________occurs when the gradients become too large due to back-propagation.

A) Exploding Gradients
B) Vanishing Gradients
C) Long Short Term Memory Networks
D) Gated Recurrent Unit Networks

__________occurs when the gradients become too large due to back-propagation.

A) Exploding Gradients
B) Vanishing Gradients
C) Long Short Term Memory Networks
D) Gated Recurrent Unit Networks

The other RNN´s issue is called 'Vanishing Gradients'. What is that?

A) When the values of a gradient are too small and the model stops learning or takes way too long because of that.
B) When the values of a gradient are too big and the model stops learning or takes way too long because of that.
C) When the values of a gradient are too small and the model joins in a loop because of that.
D) When the values of a gradient are too big and the model joins in a loop because of that.

upGrad

The other RNN´s issue is called 'Vanishing Gradients'. What is that?

A) When the values of a gradient are too small and the model stops learning or takes way too long because of that.
B) When the values of a gradient are too big and the model stops learning or takes way too long because of that.
C) When the values of a gradient are too small and the model joins in a loop because of that.
D) When the values of a gradient are too big and the model joins in a loop because of that.

LSTM

upGrad

- Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by many people in following work.

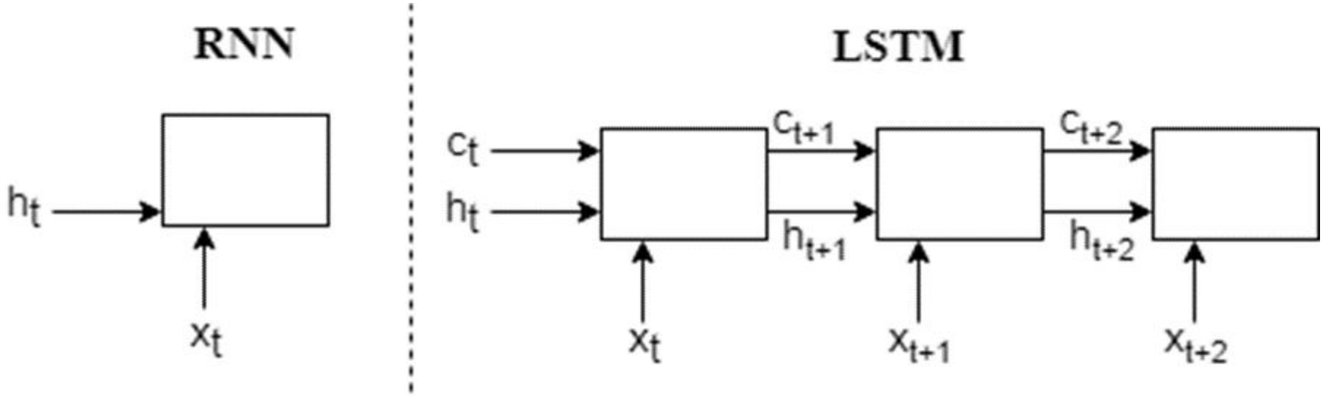
- LSTMs work tremendously well on a large variety of problems, and are widely used.

- LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior.

LSTM vs RNN

upGrad

- RNN is not able to distinguish between important or not so important information.
- LSTM contains gates that decide which data is important and can be useful in future and which data has to be erased.
- RNN cell receives the inputs of hidden state(h) and input(x) while LSTM cell has an additional input called cell state (c).
- Cell state is an internal memory where info is stored and hidden state is where computations are done.

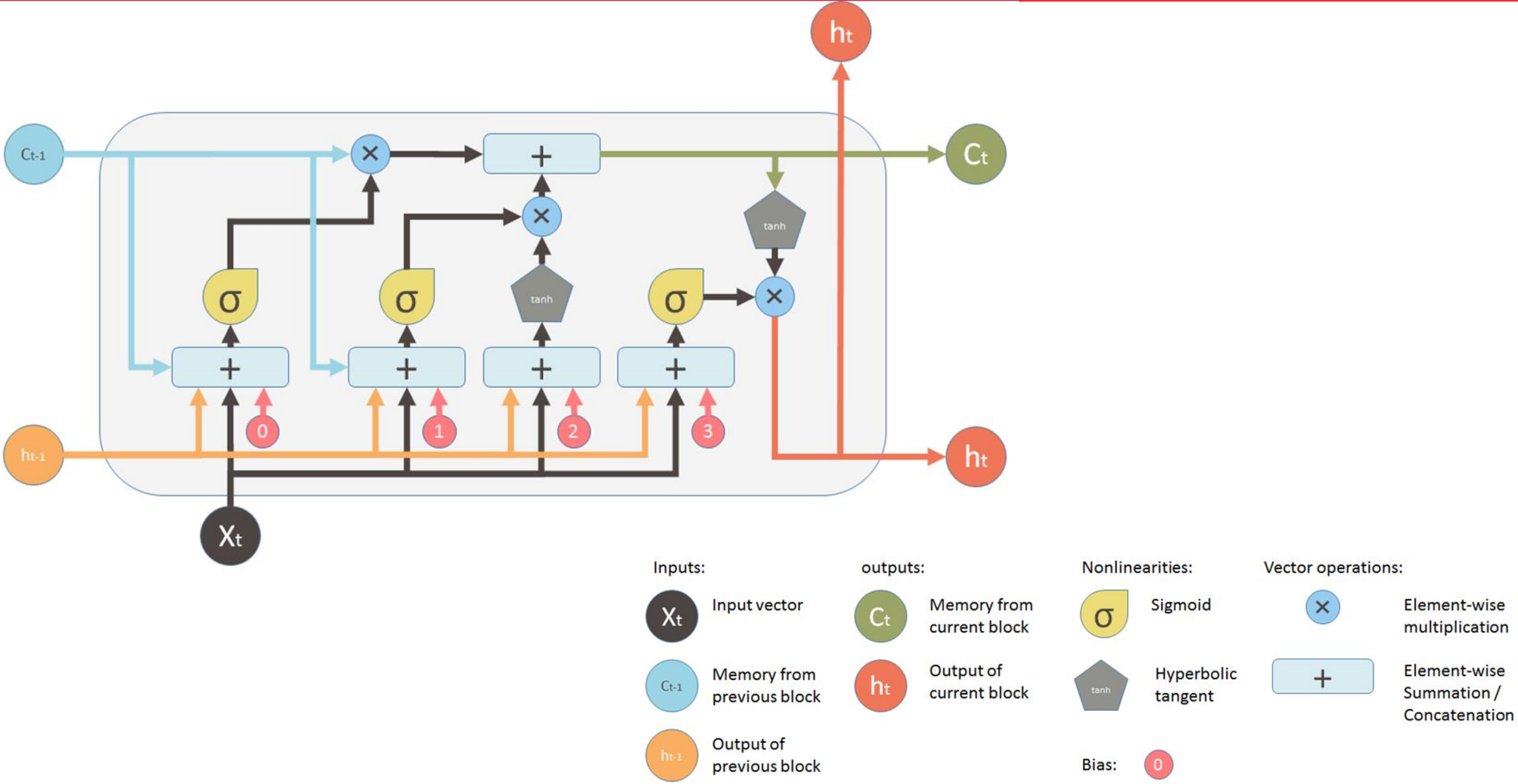


LSTM - Components

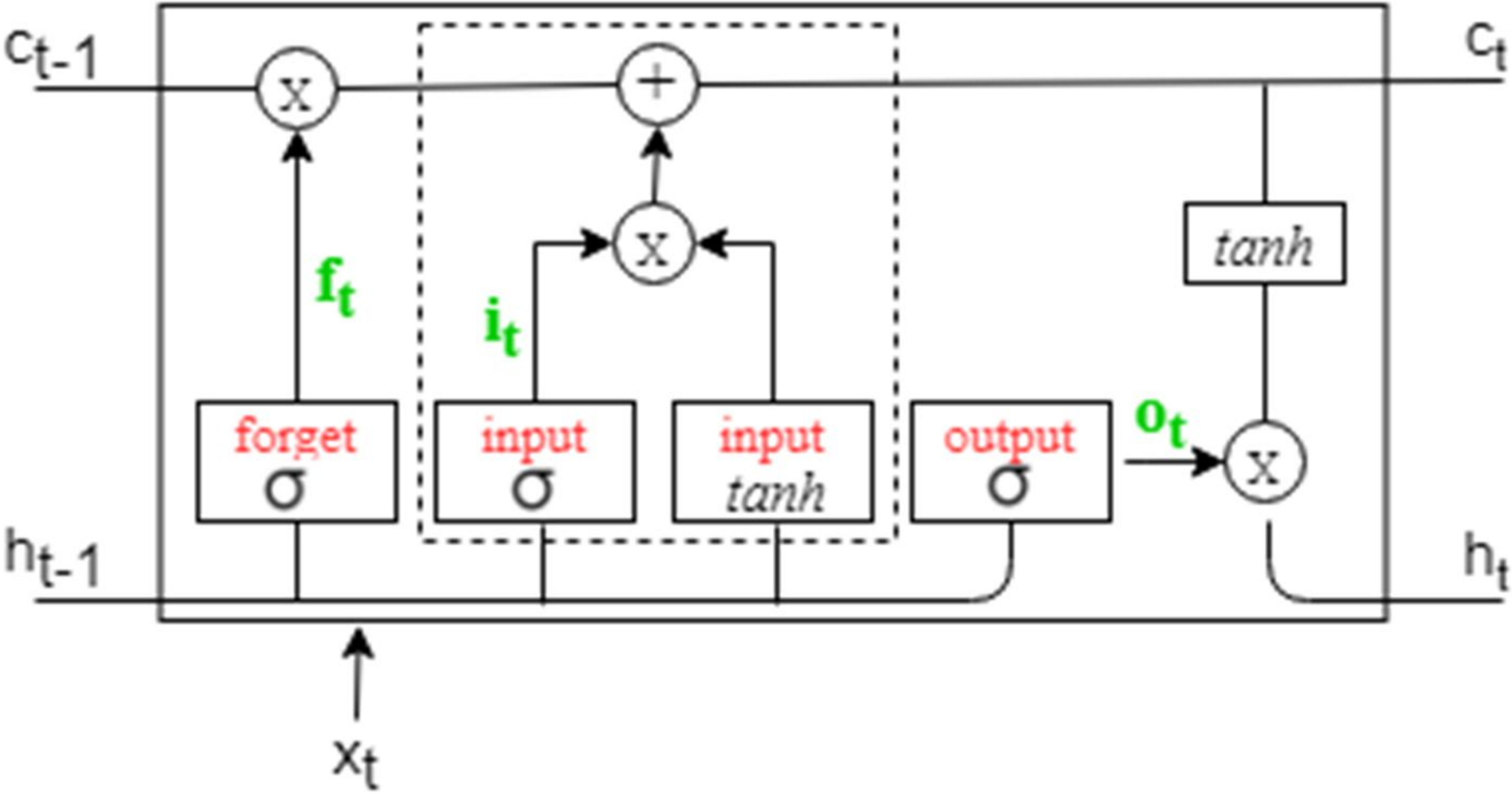
upGrad

- Forget gate (f): Neural Network with Sigmoid
- Input gate (i): Neural Network with Sigmoid
- Candidate gate (g): Neural Network with tanh
- Output gate (o): Neural Network with Sigmoid
- Cell state (c): Vector
- Hidden state (h): Vector

LSTM - Architecture

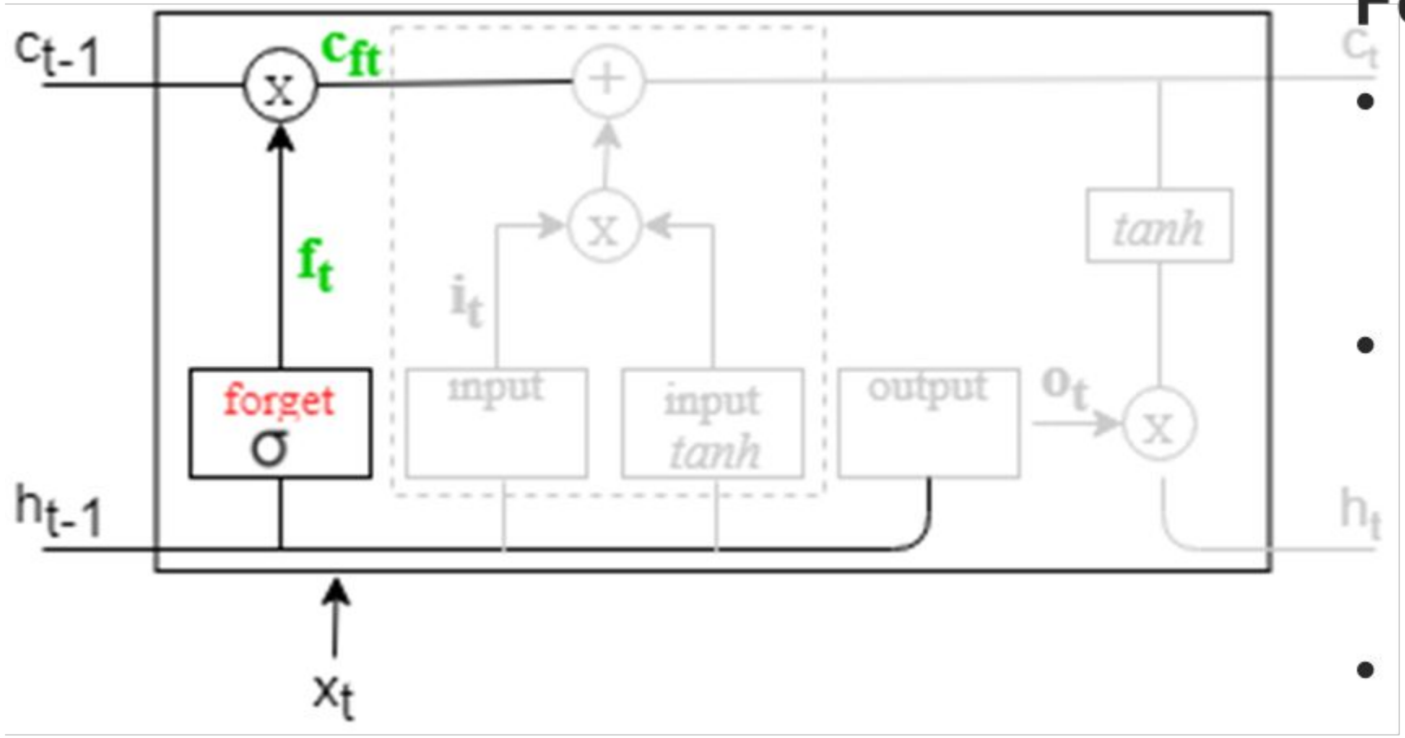


LSTM - Architecture



LSTM - Process - Step 1

upGrad

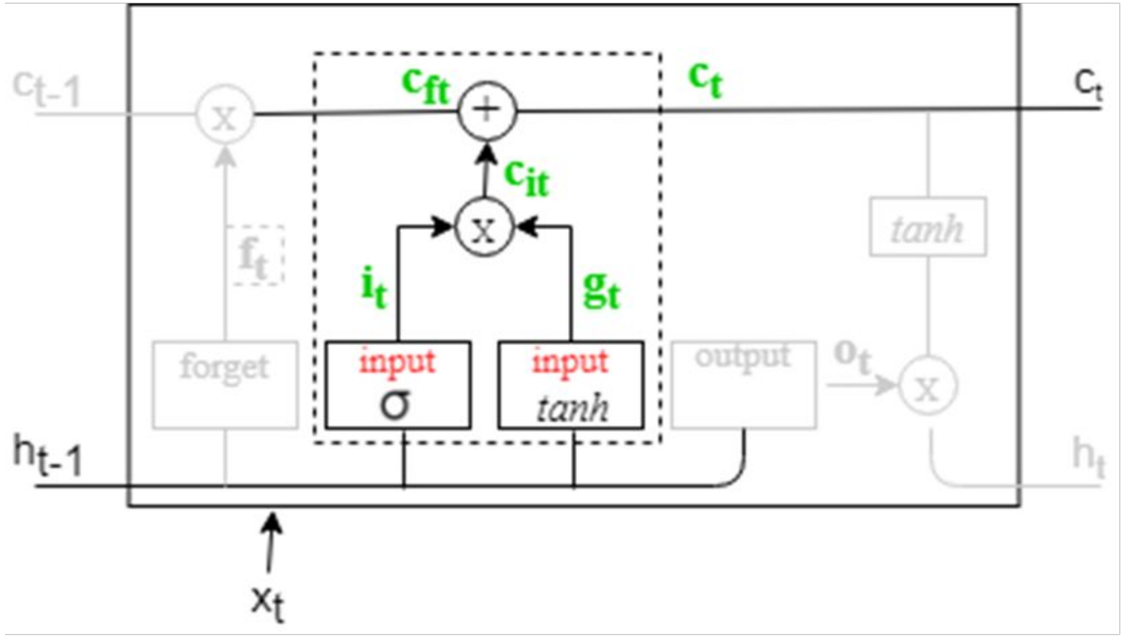


$$f_t = \sigma\,[\,(W_{fh} * h_{t-1}) + (W_{fx} * x_t) + b_f\,]$$
$$c_{ft} = c_{t-1} * f_t$$

Forget Gate:

- Irrelevant info coming from previous timestep (h_{t-1}) is removed.
- This decision is made by the **forget gate** having **Sigmoid** function of range(0,1).
- Then, multiplying its output with previous cell state gives the necessary info (c_{ft}).
- If the value of forget gate is 0 then info from cell state is supposed to be removed.

LSTM - Process - Step 2

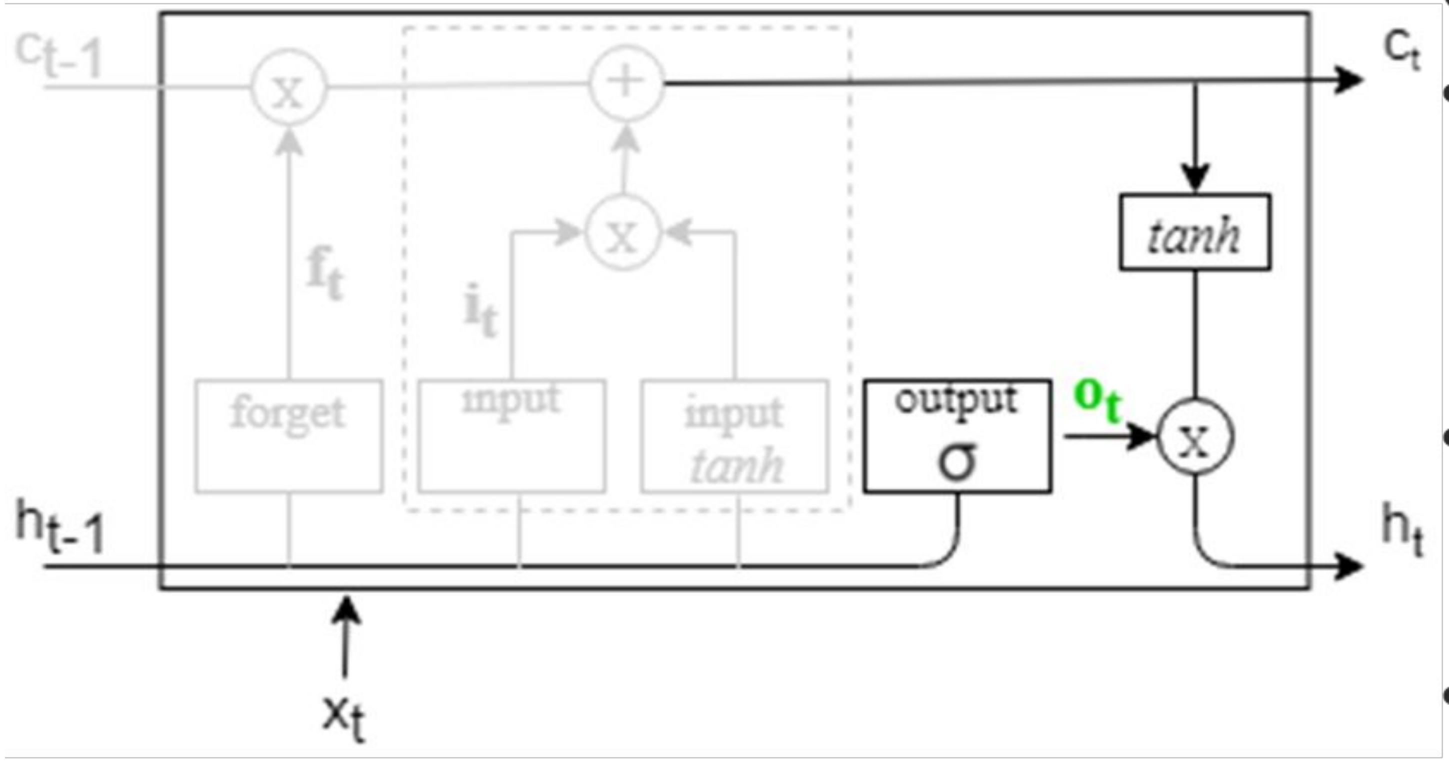


$$i_t = \sigma[(W_{ih} * h_{t-1}) + (W_{ix} * x_t) + b_i]$$
$$g_t = \tanh[(W_{ch} * h_{t-1}) + (W_{cx} * x_t) + b_c]$$
$$c_{it} = i_t * g_t$$
$$c_t = c_{it} + c_{ft}$$

Input Gate (with Candidate Gate):

- Decides what info should be stored in cell state.
- The previous cell info is passed to Sigmoid function of input gate(it). For value 0 indicates the info need not be stored. The previous timestep info passed to tanh layer to create candidate values to be added in current cell state.
- The multiplied result of both above when added to the cell state, the current cell state is updated with the new necessary info (ct).

LSTM - Process - Step 3



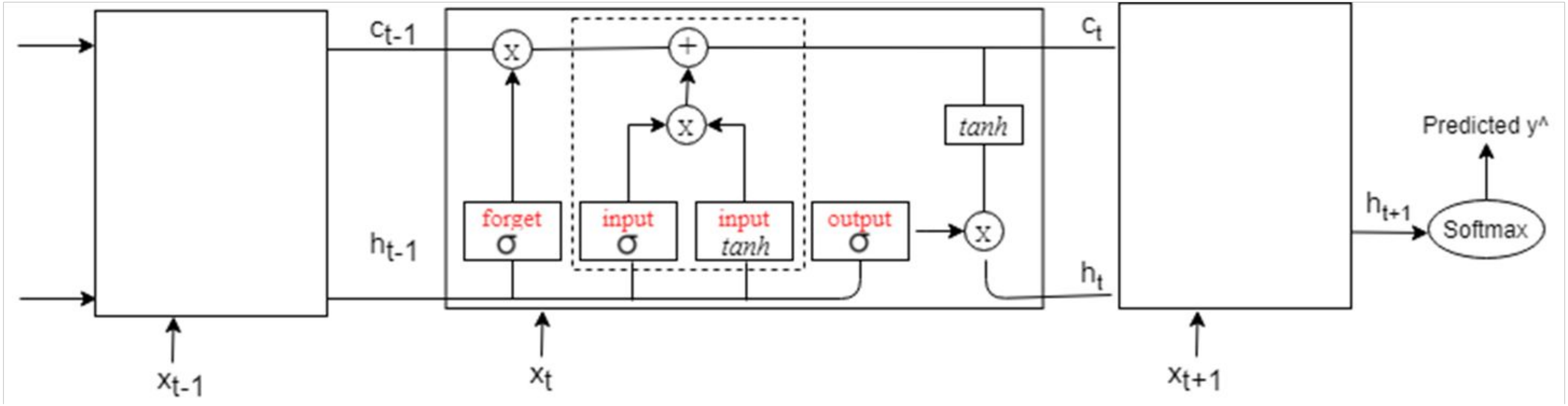
$$O_t = \sigma\left[(W_{oh} * h_{t-1}) + (W_{ox} * x_t) + b_o \right]$$
$$h_t = O_t * \tanh(c_t)$$

Output Gate:

- The output gate with Sigmoid function decides what goes from cell for the output.
- For value 0, hidden state (ht-1) is not passed to the output.
- Updated cell info(ct) is passed to tanh and multiplied to the output of output gate to give current timestep info (ht).

LTSM - Process - Step 4

If it's the last LSTM cell then the hidden state (h_t) is passed to the softmax function to give the output y.



LSTM - MCQ/MSQ

What is the purpose of the input gate in an LSTM network?

A) To control the flow of information from the current input
B) To adjust the learning rate during training
C) To introduce non-linearity to the network
D) None of the above

LSTM

upGrad

What is the purpose of the input gate in an LSTM network?

A) To control the flow of information from the current input
B) To adjust the learning rate during training
C) To introduce non-linearity to the network
D) None of the above

What is the purpose of the cell state in an LSTM network?

A) To store long-term dependencies in the input sequence
B) To adjust the learning rate during training
C) To compute the gradients for backpropagation
D) None of the above

LTSM

What is the purpose of the cell state in an LSTM network?

A) To store long-term dependencies in the input sequence
B) To adjust the learning rate during training
C) To compute the gradients for backpropagation
D) None of the above

Sample Questions

upGrad

- Explain RNN.
- Explain the need of RNN for sequential data.
- Explain the working of RNN in general.
- Differentiate CNN and RNN.
- Differentiate RNN and LSTM.
- List various Gates and States in LSTM.
- Explain the generic architecture of LSTM.
- Explain issues with LSTM.
- Explain the generic architecture of GRU.
- Explain the role of Update Gate in GRU.
- Differentiate LSTM and GRU.
- Apply LSTM / GRU for any application (E.g. time-series based stock market prediction and explain its architecture)

THANK YOU