CO4

DEEP LEARNING

23AD2205A

Topic:

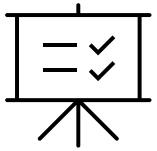
LSTM

Session - 23

AIM OF THE SESSION

To familiarize students with the sequence prediction problems

INSTRUCTIONAL OBJECTIVES



This Session is designed to:

1. Discuss the Contractive Autoencoders and Variational autoencoder
2. Demonstrate the concept of Contractive Autoencoders and Variational autoencoder Discussion on Contractive Autoencoders and Variational autoencoder

LEARNING OUTCOMES

At the end of this session, you should be able to: concepts for real time applications

1. To build Contractive Autoencoders and Variational autoencoder
2. To apply different types of Contractive Autoencoders and Variational autoencoder

Long Short Term Memory Networks (LSTMs)

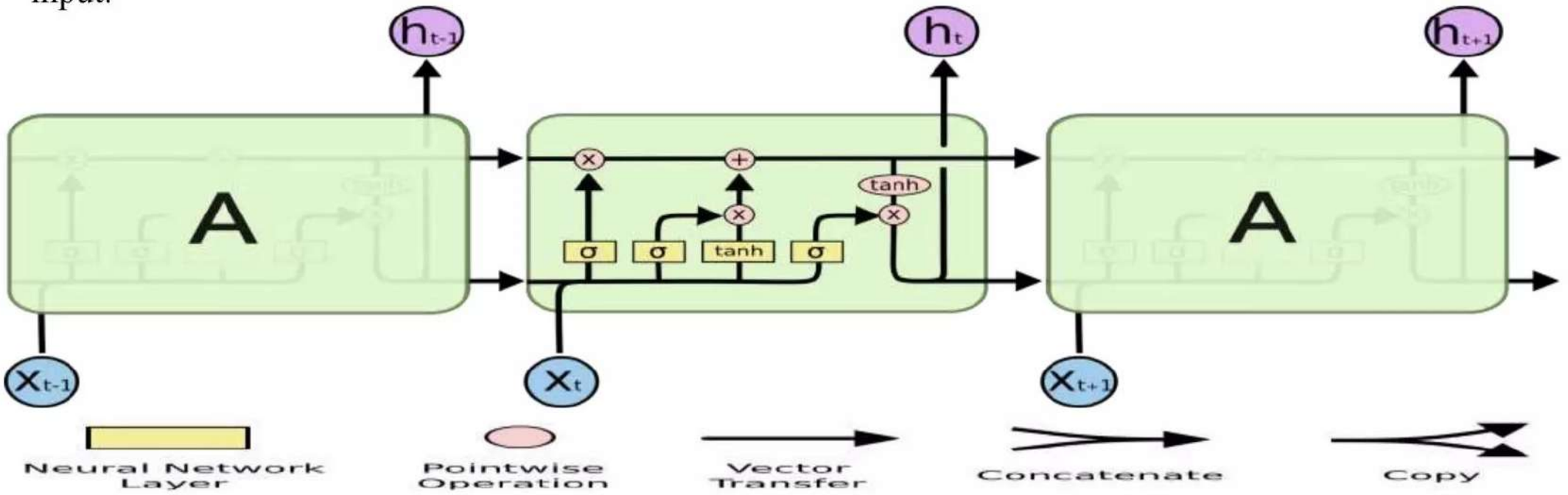
- LSTMs are a type of recurrent neural network (RNN) that can learn and memorize **long-term dependencies.**
- LSTMs retain past information for long period of time. Hence, It is very useful in time-series prediction.
- LSTMs have a chain-like structure where four (memory cell, forget, input, output) interacting layers communicate in a unique way.
- LSTM has three gates (forget, input, output) to protect and control the cell state.

LSTMs Working principle:

- First, they forget irrelevant information of the previous state or keep the relevant information of the previous state.
- Next, they selectively update the memory cell-state values.
- The memory cell state carry relevant information from the earlier time steps to later time steps throughout the processing of the sequence that reducing the effects of short-term memory.
- As the cell state goes on its journey, information get's **added** or **removed** to the cell state via gates.
- The gates are different neural networks that decide which **information is allowed** on the cell state. The gates can learn what information is relevant to keep or forget during training.
- Finally, provides the output of certain parts of the cell state.

Long Term Short Memory(LSTM):

Long short-term memory is a type of RNN model designed to prevent the output of a neural network from either exploding or decaying (long-term dependency) as it passes through the feedback loops for a given input.



Gates of LSTM

- An LSTM has three of these gates, to protect and control the cell state:
  - Forget gate layer → Keep gate
  - Input gate layer → Write gate
  - Output gate layer → Read gate

Cont…

Forget gate

- This gate decides what information should be thrown away or kept.
- Information from the previous hidden state and information from the current input is passed through the sigmoid function.
- Values come out between 0 and 1. The closer to 0 means to forget, and the closer to 1 means to keep.

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

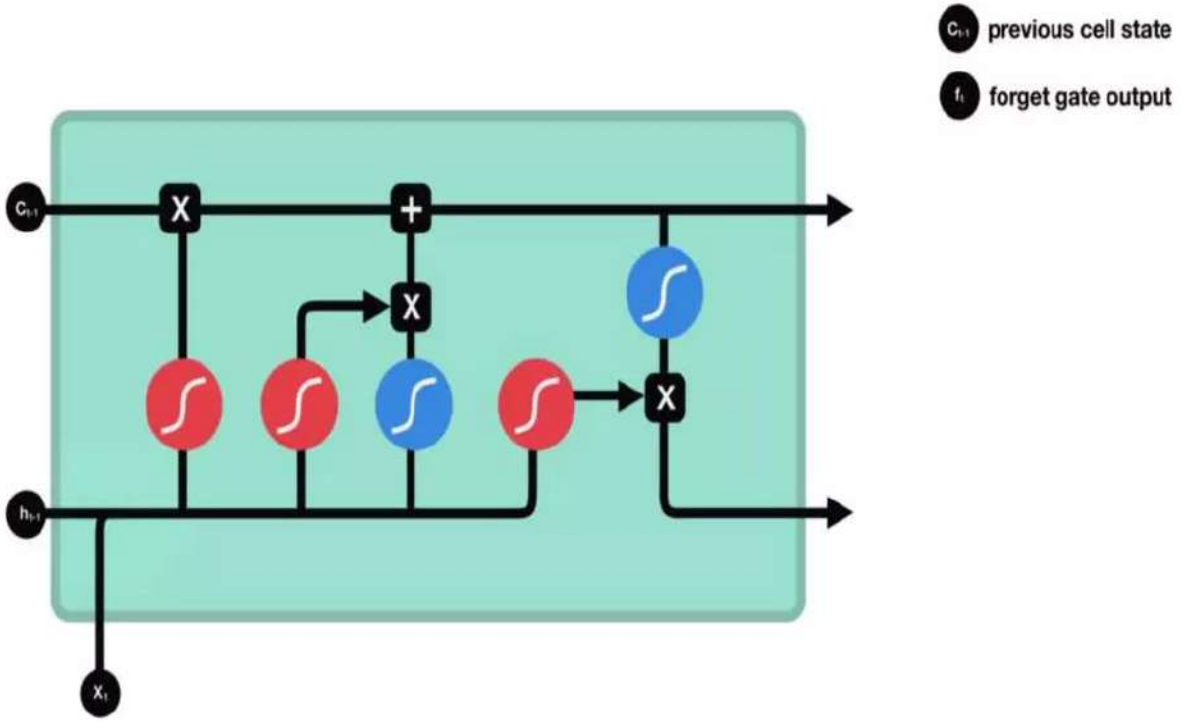


Figure: Forget Gate.

Cont…

Input Gate

- The goal of this gate is to determine what new information should be added to the networks long-term memory (cell state), given the previous hidden state and new input data.
- The input gate is a sigmoid activated network which acts as a filter, identifying which components of the 'new memory vector' are worth retaining. This network will output a vector of values in [0,1].
- It is also passed the hidden state and current input into the tanh function to squish values between -1 and 1 to help regulate the network.

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

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

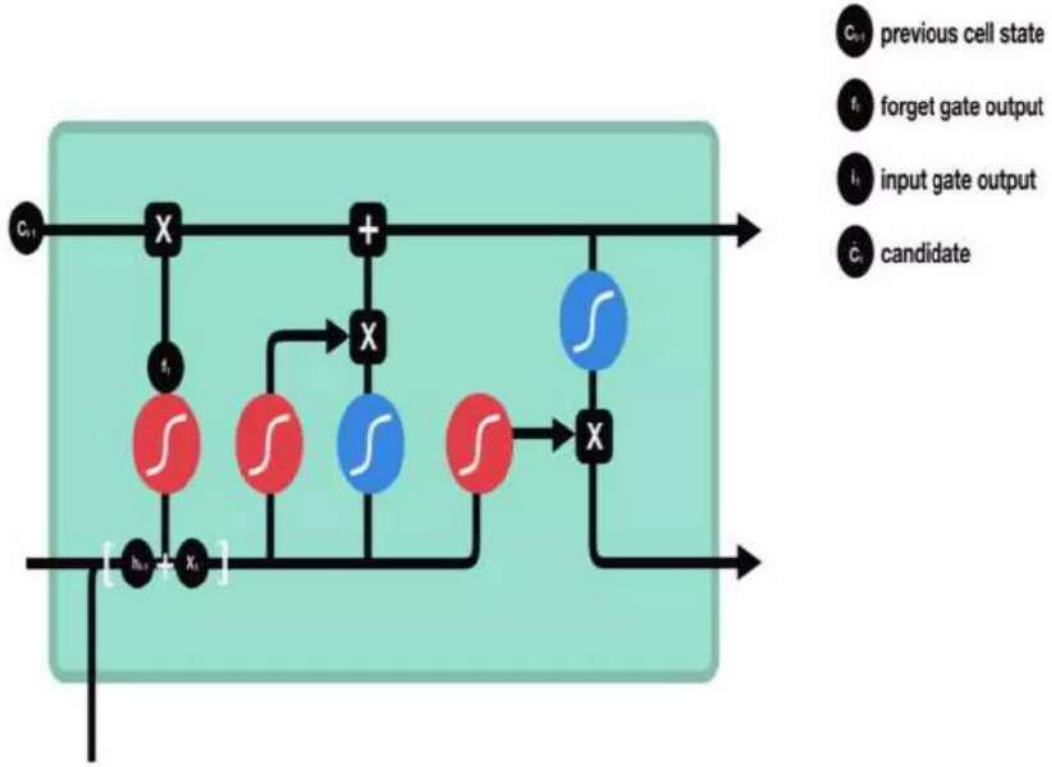


Figure: Input Gate.

Cont…

Cell State

- The next step is to decide and store the information from the new state in the cell state.
- The previous cell state C(t-1) gets multiplied with forget vector f(t). If the outcome is 0, then values will get dropped in the cell state.
- Next, the network takes the output value of the input vector i(t) and performs point-by-point addition, which updates the cell state giving the network a *new cell state C(t).*

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

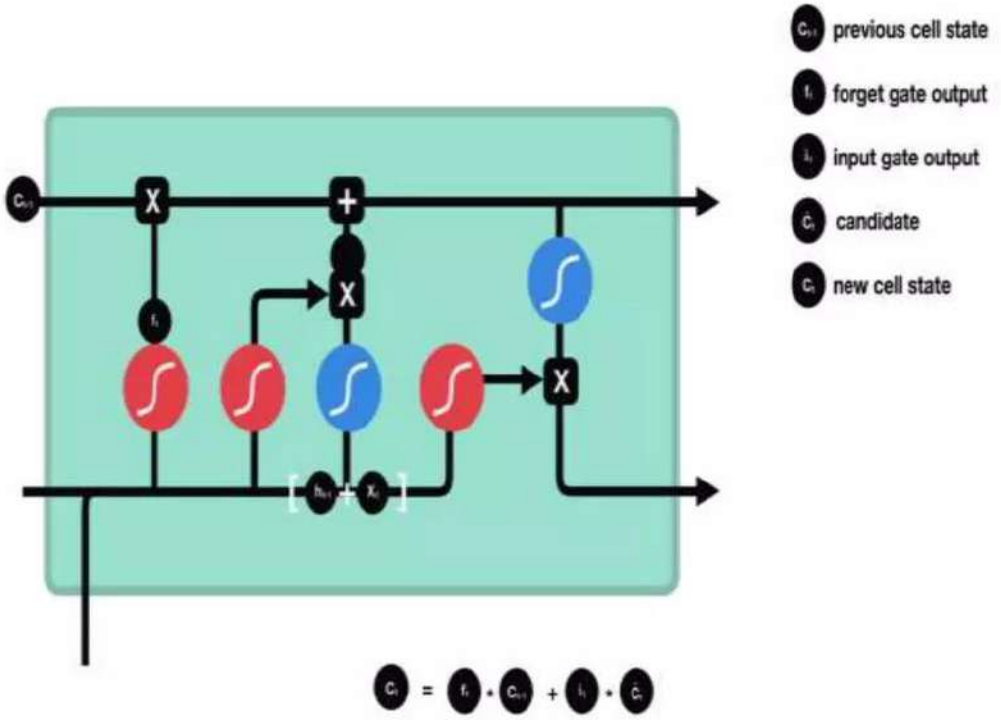


Figure: Cell State.

Cont…

Output Gate

- The output gate decides what the next hidden state should be. The hidden state contains information on previous inputs. The hidden state is also used for predictions.

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

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

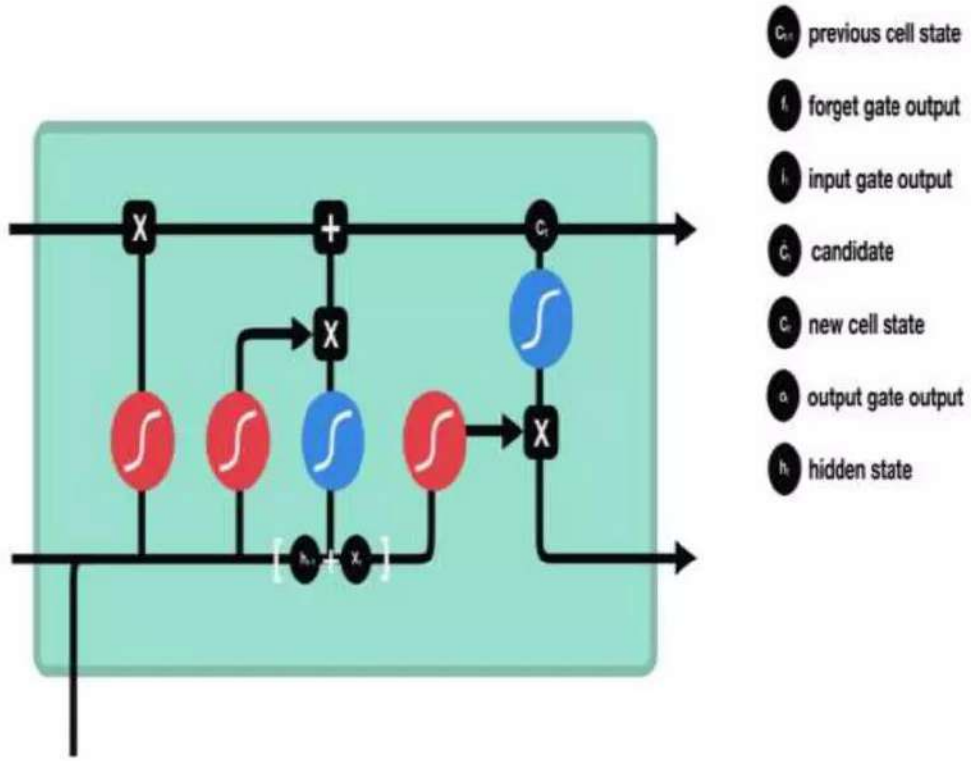
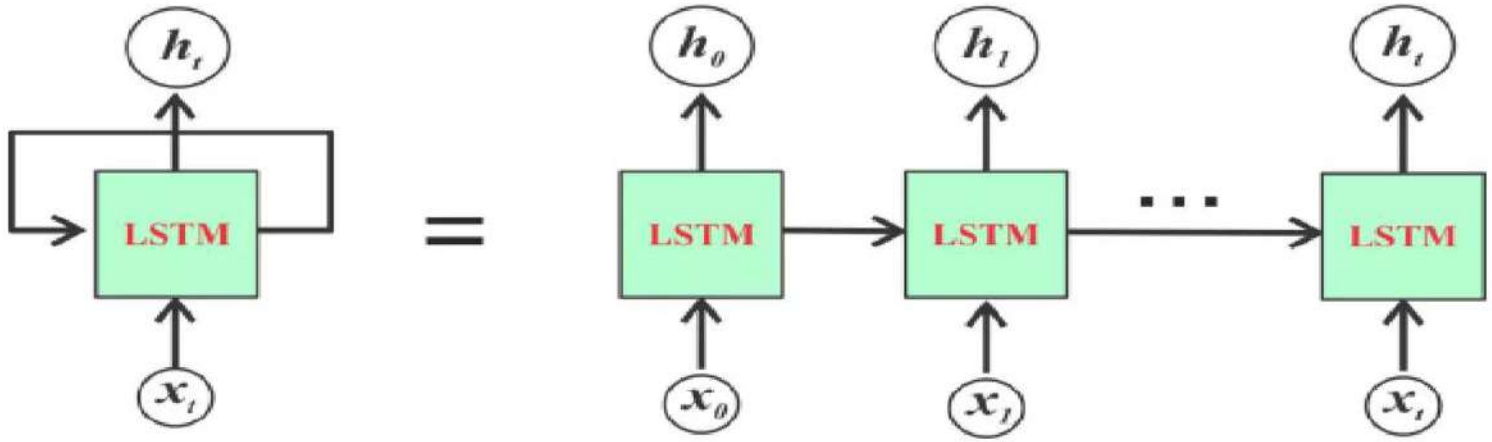


Figure: Output Gate.

LSTM Architecture



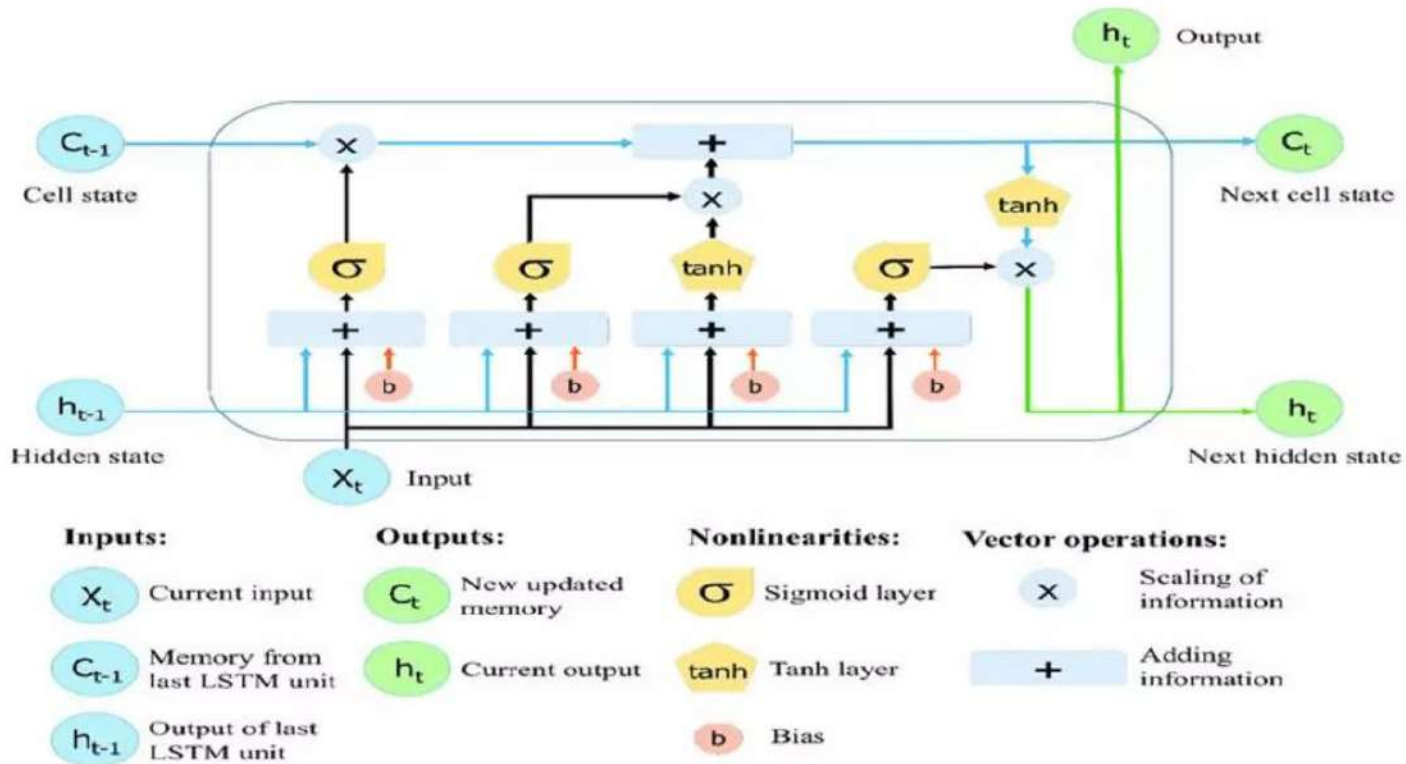
Components in LSTM:

- *Three gates:* forget gate, input gate and output gate.
- Memory cell state

- **Forward Propagation:** Processes the data passing on information. The differences are the operations within the LSTM's cells.
- These operations are used to allow the LSTM to **keep** or **forget** information.

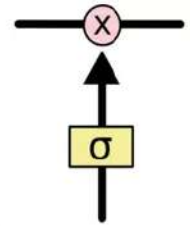
Backward Propagation

- Update the parameters to reduce the error.



LSTM Unit Structure

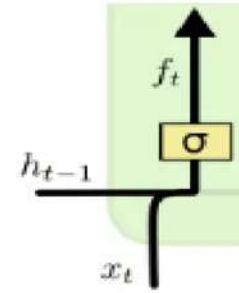
LSTM layers working principle



- Gates are composed with a sigmoid neural net layer and a pointwise multiplication operation.
- The sigmoid layer output range is between zero and one that describe how much of each component pass/remove information. A value of zero means "no information allows," while a value of one means "pass everything".

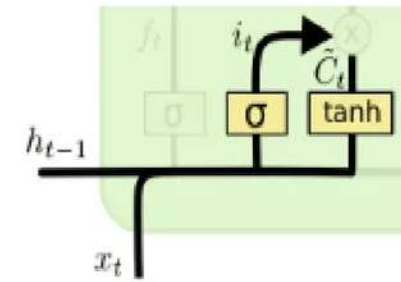
Forget gate layer:

- Decides what information going to throw away from the memory cell state.
- 1 represents "completely keep this" while a 0 represents "completely reject this."



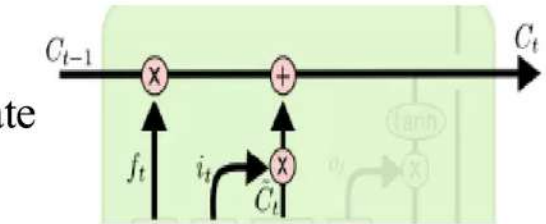
Input gate layer:

- The next step is to decide what new information we're going to store in the cell state.
- This has two parts. First, a sigmoid layer called the "input gate layer" decides which values we'll update.
- Next, a tanh layer creates a vector of new candidate values, C_t, that could be added to the state.
- In the next step, we'll combine these two to create an update to the state.



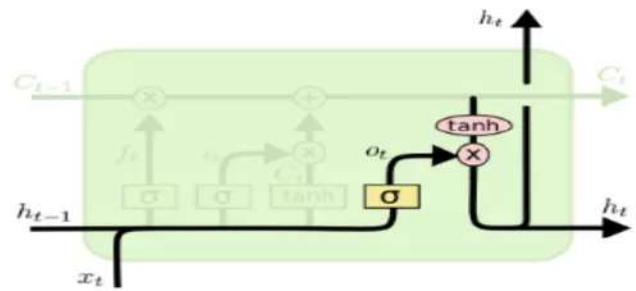
Memory Cell State:

- Update the old cell state, C_{t-1}, into the new cell state C_t.
- Multiply the old state by f_t, forgetting the things we decided to forget earlier.
- Then we add $i_t * C_t$. This is the new candidate values, scaled by how much we decided to update each state



Output gate layer:

- Decides output based on cell state, but will be a filtered version.
- First, run a sigmoid layer which decides what parts of the cell state to be output.
- Then, we put the cell state through *tanh* (to push the values to be between −1 and 1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.



Types of LSTM models based on input and output

- **One input to One output -** eg : Giving labels to image
- **One input to many outputs-** eg : Giving description/caption to image

  (description will have sequence of words - many output)
- **Many inputs to one output -** eg : Predicting the next word in given incomplete statement
- **Many inputs to Many outputs-** eg : Stock market prediction for following days based on past data

Applications of LSTM

1. Speech Recognition (Input is audio and output is text) - Google Assistant, Microsoft Cortana, Apple Siri
2. Machine Translation (Input is text and output is also text) - Google Translate
3. Image Captioning (Input is image and output is text)
4. Sentiment Analysis (Input is text and output is rating)
5. Music Generation/Synthesis (input music notes and output is music)
6. Video Activity Recognition (input is video and output is type of activity)