Long Short-Term Memory (LSTM)

LSTM → Long short-Term Memory

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) designed to handle the vanishing gradient problem that can be encountered when training traditional RNNs. LSTMs are well-suited for tasks that involve sequential data, such as time series prediction, natural language processing, and speech recognition.

gradient - small value
≃ 0

Intution

RNN and the Vanishing Gradient Problem

Traditional RNNs maintain a hidden state that is passed from one time step to the next. However, when training these networks with backpropagation through time, the gradients can either vanish or explode, making it difficult for the network to learn long-term dependencies.

LSTM Structure

LSTM networks mitigate this issue by using a more complex architecture that includes a cell state and three types of gates (input, forget, and output gates) to control the flow of information.

- **Cell state (C):**

This is the memory of the network that carries information across different time steps.

- **Hidden state (h):**

This is the output at each time step.

- **Input Gate (i):**

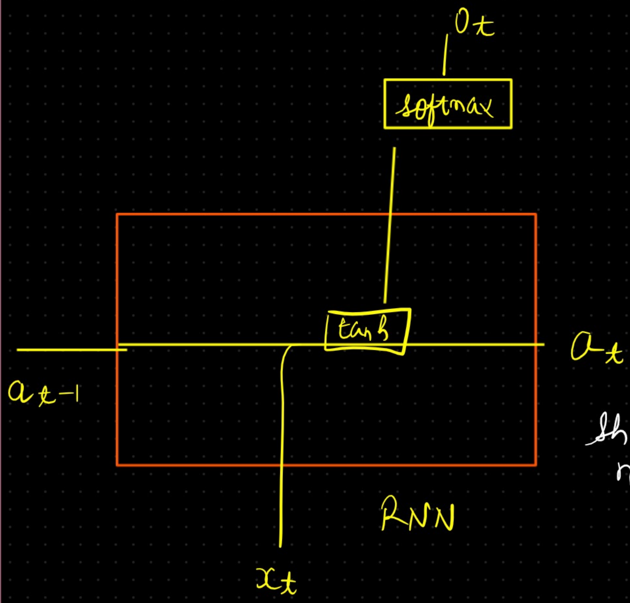
Controls how much of the new information to add to the cell state.

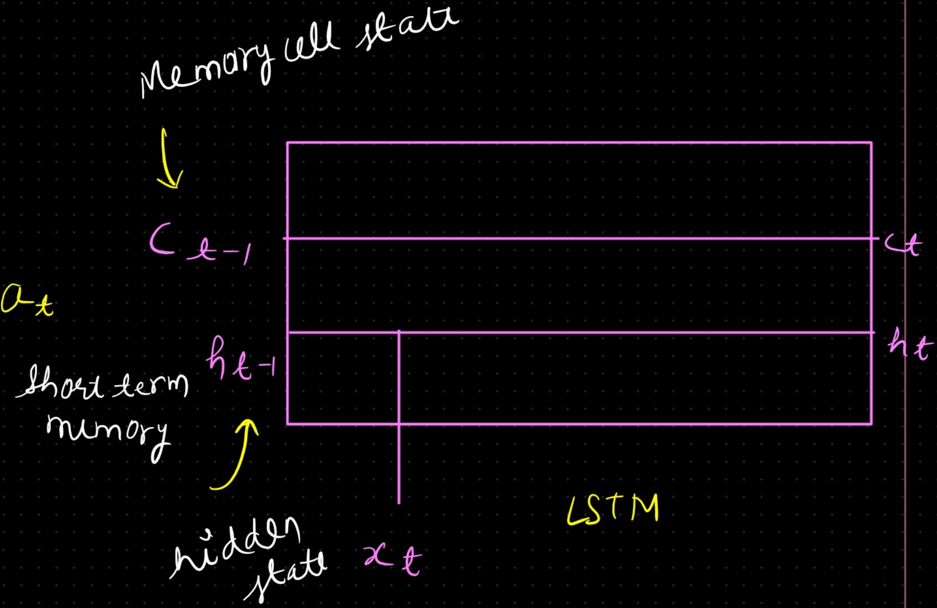
- **Forget Gate (f)**

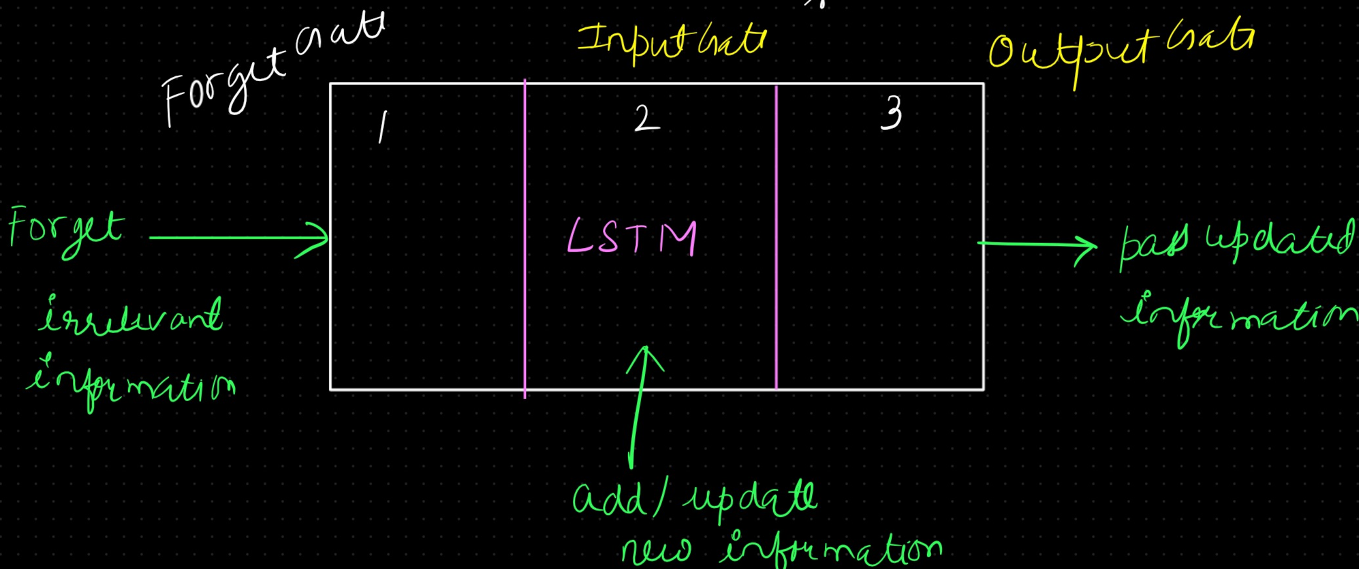
Controls what information to throw away from the cell state.

- **Output Gate (o):**

Controls what part of the cell state to output as the hidden state.

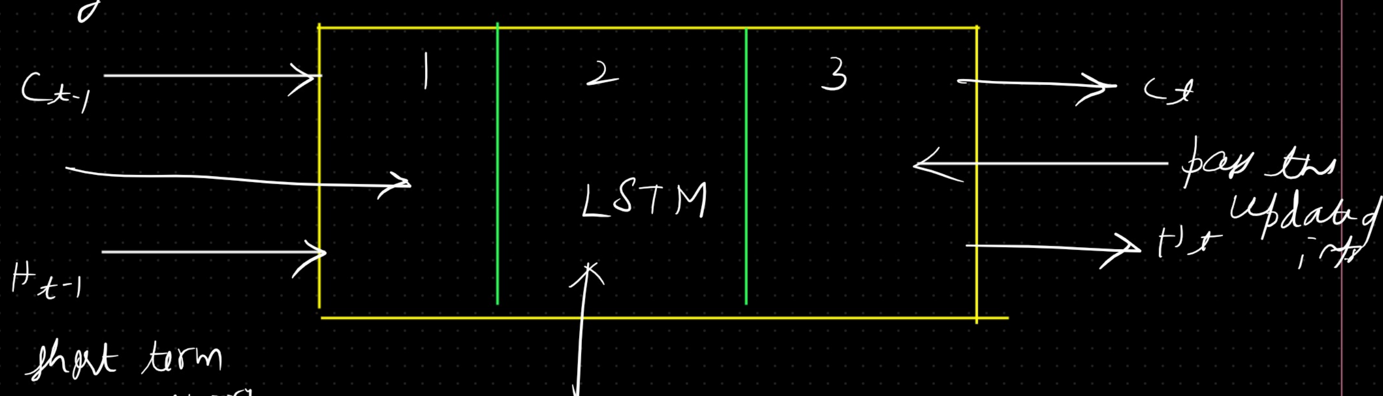


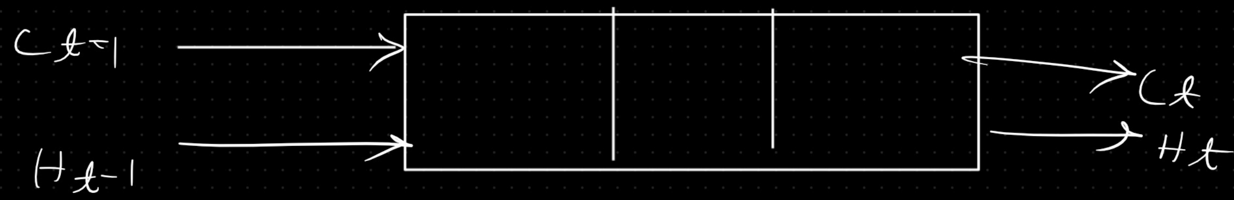




∴ They control the flow of Information

Long Term Memory





LSTM

LSTM Equation

1. Forget Gate

Decides what information to discard from the cell.

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

2. Input Gate

Decides which value to update

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

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

③ Update cell state

Update the cell state based on the Input gate and forget gate.

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

④ Output Gate

Decides what the next hidden state should be.

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

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

Example: Sentiment Analysis

"I really enjoyed the movie"

① Words ⟶ vectors

"I", "really", "enjoyed", "the", "movie"

② LSTM

a) I → passed to the LSTM

The network updates its cell state and hidden state

b) The next word → really

The LSTM update its states based on the current words and the previous state

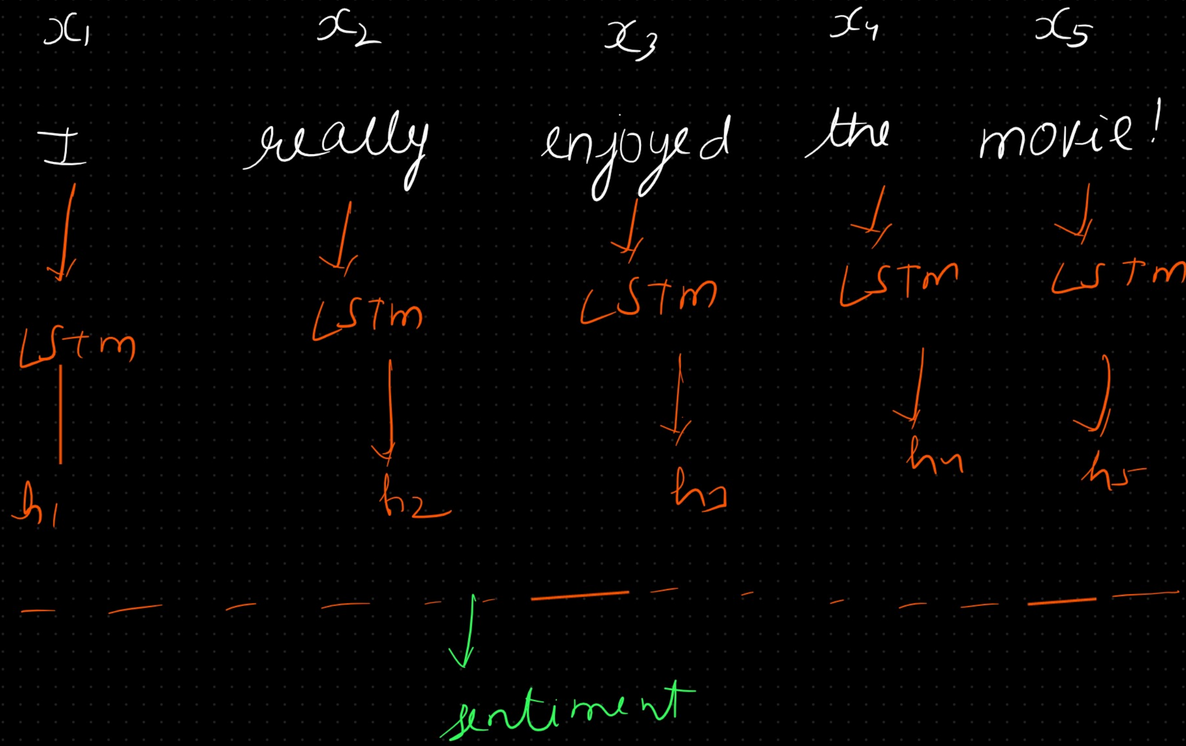
c) This continuous " _ _ _

Output

After processing the entire sentence, the final hidden state is used to predict the sentiment.

⇓

positive sentiment



Key Points

1. Long term Dependencies
2. Gated Mechanism
3. sequential Data

LSTM resolves the vanishing gradient problem of the RNN.
LSTM uses three gates: input gate, forget gate, and output gate for processing.

LTSM vs RNN

Aspect	LSTM (Long Short-Term Memory)	RNN (Recurrent Neural Network)
Architecture	A type of RNN with additional memory cells	A basic type of RNN
Memory Retention	Handles long-term dependencies and prevents vanishing gradient problem	Struggles with long-term dependencies and vanishing gradient problem
Cell Structure	Complex cell structure with input, output, and forget gates	Simple cell structure with only hidden state
Handling Sequences	Suitable for processing sequential data	Also designed for sequential data, but limited memory
Training Efficiency	Slower training process due to increased complexity	Faster training process due to simpler architecture
Performance on Long Sequences	Performs better on long sequences	Struggles to retain information on long sequences
Usage	Best suited for tasks requiring long-term memory, such as language translation and sentiment analysis	Appropriate for simple sequential tasks, such as time series forecasting
Vanishing Gradient Problem	Addresses the vanishing gradient problem	Prone to the vanishing gradient problem

https://colah.github.io/posts/2015-08-Understanding-LSTMs/