

# Deep Learning | Introduction to Long Short Term Memory

---

**Long Short-Term Memory** is an improved version of recurrent neural network designed by Hochreiter & Schmidhuber. **LSTM** is well-suited for sequence prediction tasks and excels in capturing long-term dependencies. Its applications extend to tasks involving time series and sequences. LSTM's strength lies in its ability to grasp the order dependence crucial for solving intricate problems, such as machine translation and speech recognition. The article provides an in-depth introduction to LSTM, covering the LSTM model, architecture, working principles, and the critical role they play in various applications.

## What is LSTM?

---

A traditional RNN has a single hidden state that is passed through time, which can make it difficult for the network to learn long-term dependencies. LSTMs address this problem by introducing a memory cell, which is a container that can hold information for an extended period. LSTM networks are capable of learning long-term dependencies in sequential data, which makes them well-suited for tasks such as [language translation](#), speech recognition, and [time series forecasting](#). LSTMs can also be used in combination with other neural network architectures, such as [Convolutional Neural Networks](#) (CNNs) for image and video analysis.

The memory cell is controlled by three gates: the input gate, the forget gate, and the output gate. These gates decide what information to add to, remove from, and output from the memory cell. The input gate controls what information is added to the memory cell. The forget gate controls what information is removed from the memory cell. And the output gate controls what information is output from the memory cell. This allows LSTM networks to selectively retain or discard information as it flows through the network, which allows them to learn long-term dependencies.

## Bidirectional LSTM

[Bidirectional LSTM](#) (Bi LSTM/ BLSTM) is recurrent neural network (RNN) that is able to process sequential data in both forward and backward directions. This allows Bi LSTM to learn longer-range dependencies in sequential data than traditional LSTMs, which can only process sequential data in one direction.

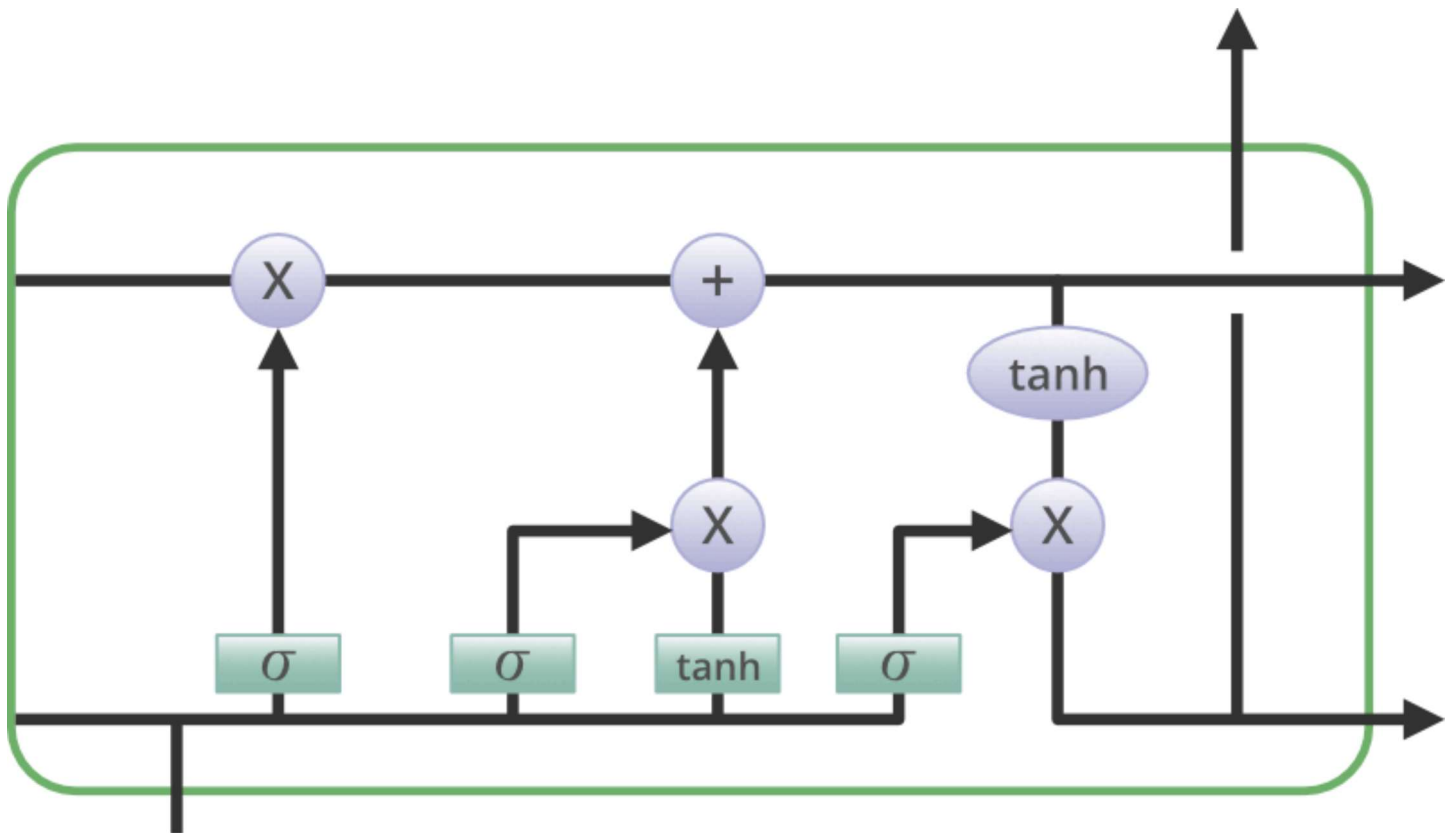
- Bi LSTMs are made up of two LSTM networks, one that processes the input sequence in the forward direction and one that processes the input sequence in the backward direction. The outputs of the two LSTM networks are then combined to produce the final output.

- Bi LSTM have been shown to achieve state-of-the-art results on a wide variety of tasks, including machine translation, speech recognition, and text summarization.

LSTMs can be stacked to create deep LSTM networks, which can learn even more complex patterns in sequential data. Each LSTM layer captures different levels of abstraction and temporal dependencies in the input data.

## Architecture and Working of LSTM

LSTM architecture has a chain structure that contains four neural networks and different memory blocks called **cells**.



Information is retained by the cells and the memory manipulations are done by the **gates**. There are three gates –

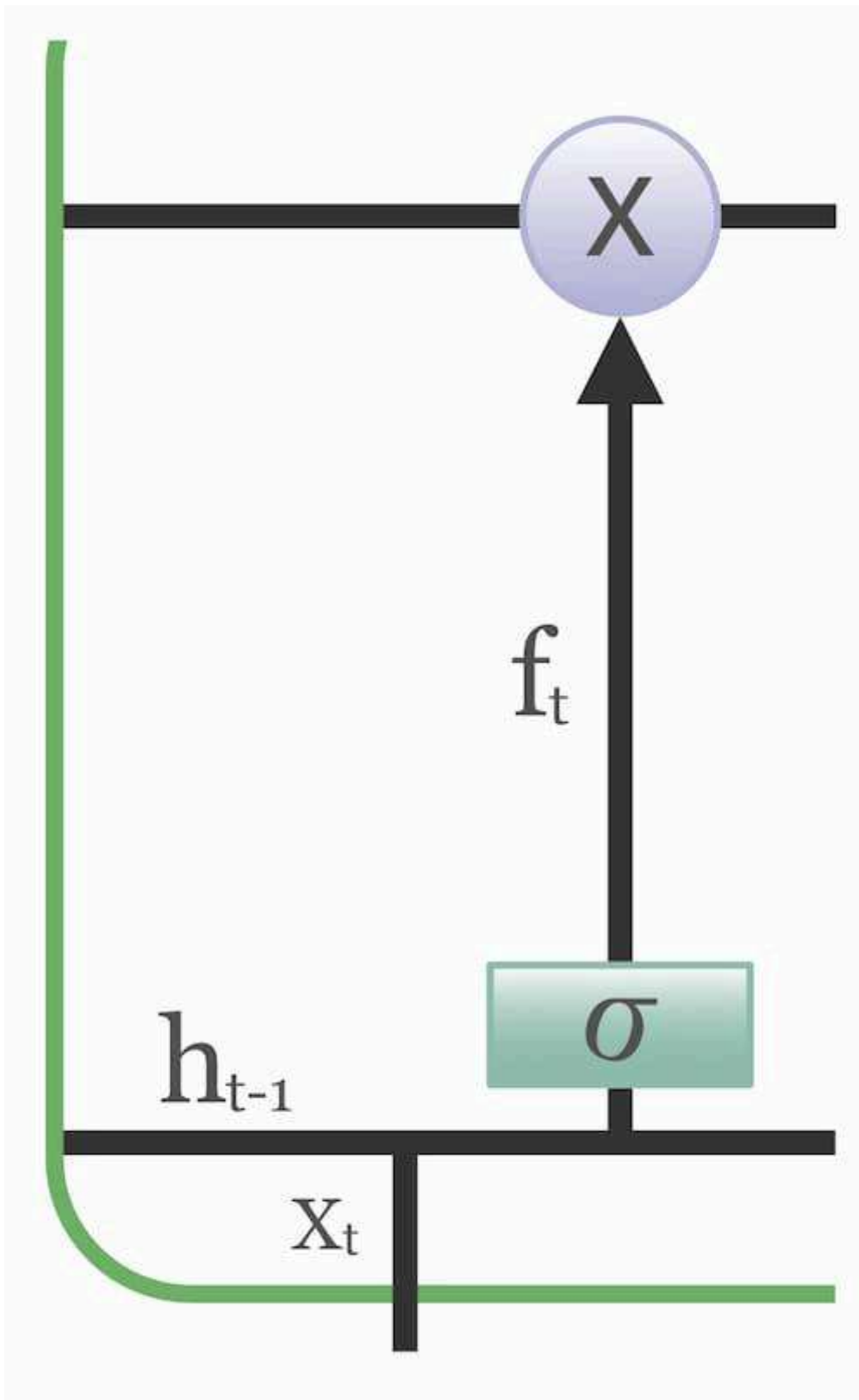
### Forget Gate

The information that is no longer useful in the cell state is removed with the forget gate. Two inputs  $x_t$  (input at the particular time) and  $h_{t-1}$  (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias. The resultant is passed through an activation function which gives a binary output. If for a particular cell state the output is 0, the piece of information is forgotten and for output 1, the information is retained for future use. The equation for the forget gate is:

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

where:

- $W_f$  represents the weight matrix associated with the forget gate.
- $[h_{t-1}, x_t]$  denotes the concatenation of the current input and the previous hidden state.
- $b_f$  is the bias with the forget gate.
- $\sigma$  is the sigmoid activation function.



## Input gate

The addition of useful information to the cell state is done by the input gate. First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs  $h_{t-1}$  and  $x_t$ . Then, a vector is created using  $\tanh$  function that gives an output from -1 to +1, which contains all the possible values from  $h_{t-1}$  and  $x_t$ . At last, the values of

the vector and the regulated values are multiplied to obtain the useful information. The equation for the input gate is:

$$i_t = (W_i[h_{t-1}, x_t] + b_i)$$

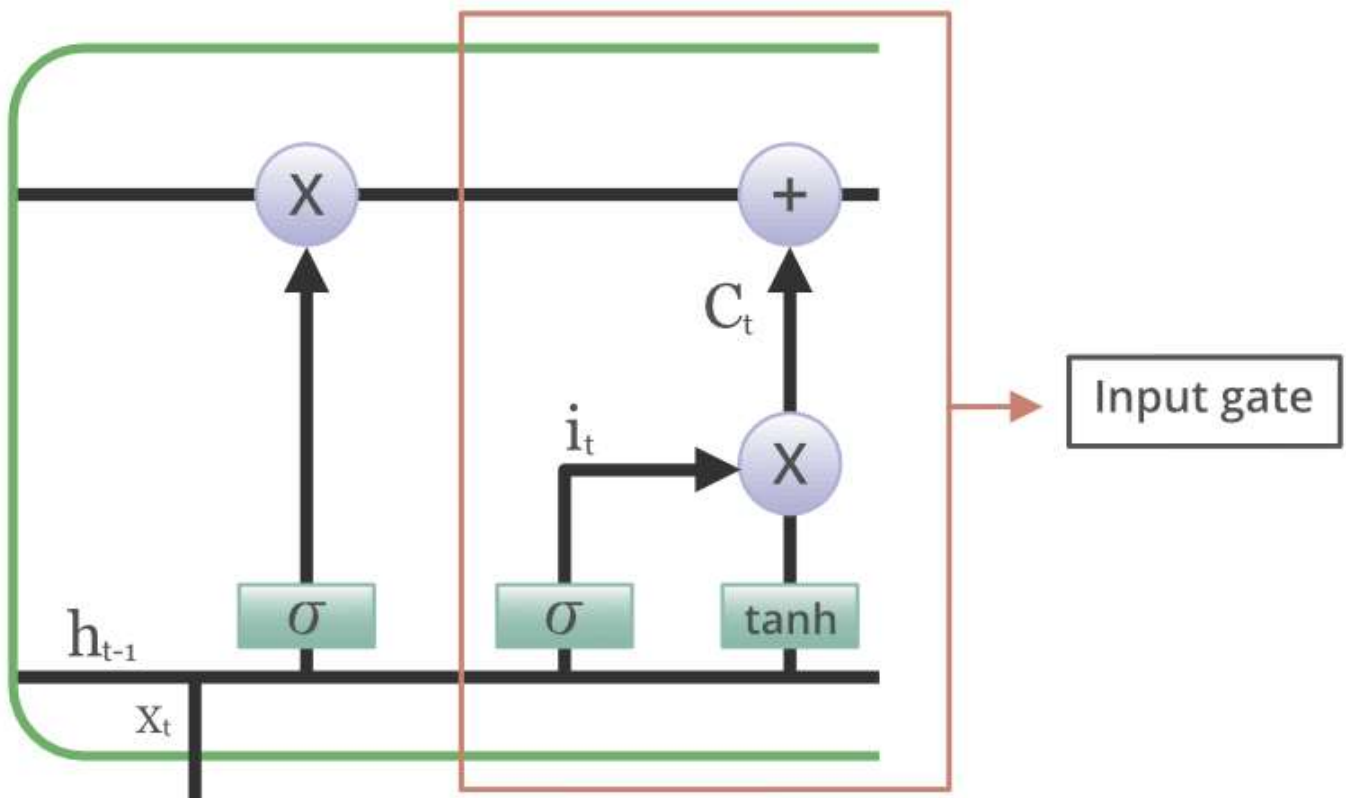
$$\hat{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$$

We multiply the previous state by  $f_t$ , disregarding the information we had previously chosen to ignore. Next, we include  $i_t \cdot \hat{C}_t$ . This represents the updated candidate values, adjusted for the amount that we chose to update each state value.

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

where

- $\odot$  denotes element-wise multiplication
- $\tanh$  is tanh activation function

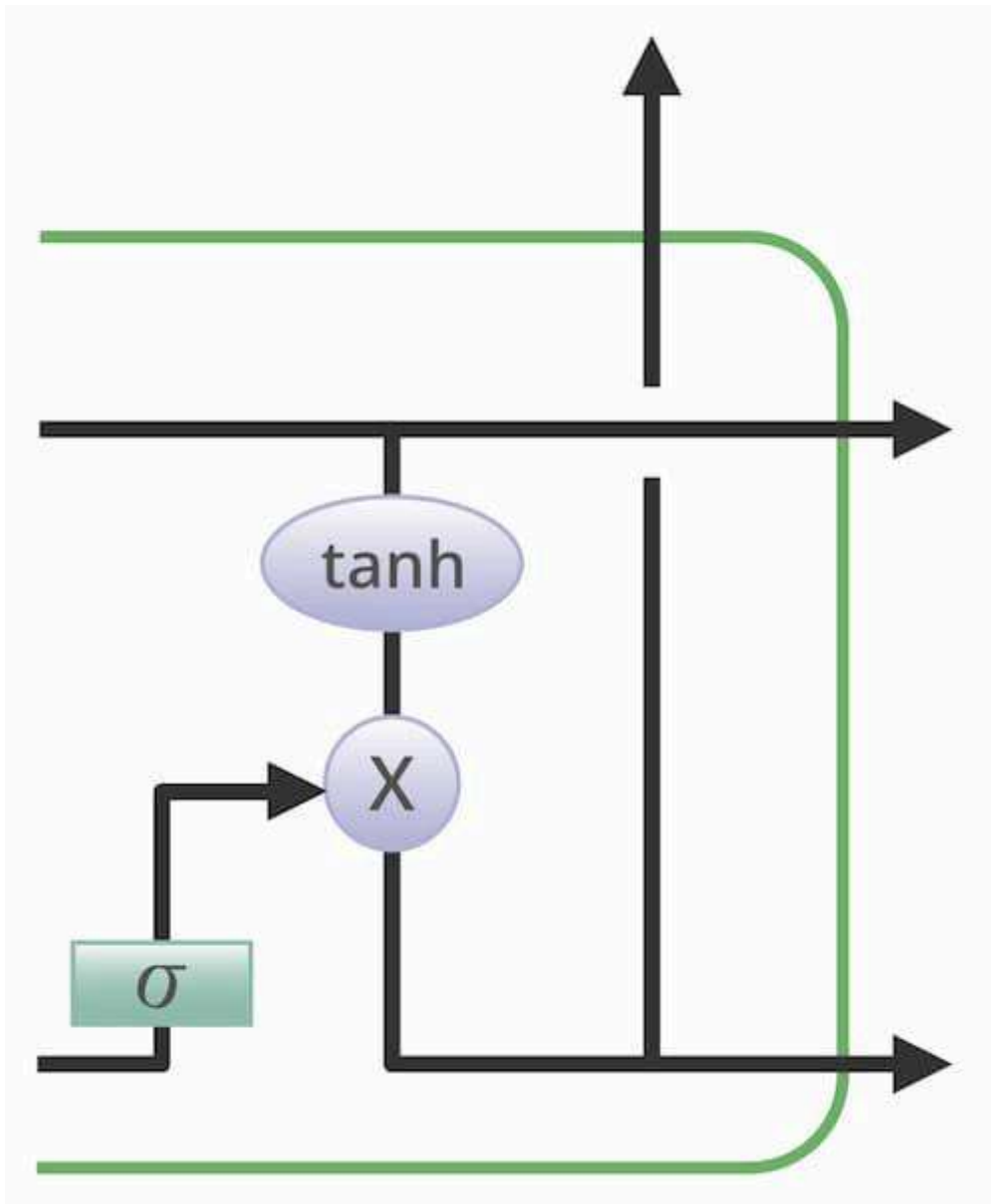


## Output gate

The task of extracting useful information from the current cell state to be presented as output is done by the output gate. First, a vector is generated by applying tanh function on the cell. Then, the

information is regulated using the sigmoid function and filter by the values to be remembered using inputs  $h_{t-1}$  and  $x_t$ . At last, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell. The equation for the output gate is:

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



## LTSM vs RNN

- Feature: Memory
  - LSTM (Long Short-term Memory): Has a special memory unit that allows it to learn long-term dependencies in sequential data
  - RNN (Recurrent Neural Network): Does not have a memory unit
- Feature: Directionality

- LSTM (Long Short-term Memory): Can be trained to process sequential data in both forward and backward directions
- RNN (Recurrent Neural Network): Can only be trained to process sequential data in one direction
- Feature: Training
  - LSTM (Long Short-term Memory): More difficult to train than RNN due to the complexity of the gates and memory unit
  - RNN (Recurrent Neural Network): Easier to train than LSTM
- Feature: Long-term dependency learning
  - LSTM (Long Short-term Memory): Yes
  - RNN (Recurrent Neural Network): Limited
- Feature: Ability to learn sequential data
  - LSTM (Long Short-term Memory): Yes
  - RNN (Recurrent Neural Network): Yes
- Feature: Applications
  - LSTM (Long Short-term Memory): Machine translation, speech recognition, text summarization, natural language processing, time series forecasting
  - RNN (Recurrent Neural Network): Natural language processing, machine translation, speech recognition, image processing, video processing

## Advantages and Disadvantages of LSTM

---

The advantages of LSTM (Long-Short Term Memory) are as follows:

- Long-term dependencies can be captured by LSTM networks. They have a memory cell that is capable of long-term information storage.
- In traditional RNNs, there is a problem of vanishing and exploding gradients when models are trained over long sequences. By using a gating mechanism that selectively recalls or forgets information, LSTM networks deal with this problem.
- LSTM enables the model to capture and remember the important context, even when there is a significant time gap between relevant events in the sequence. So where understanding context is important, LSTMs are used. eg. machine translation.

The disadvantages of LSTM (Long-Short Term Memory) are as follows:

- Compared to simpler architectures like feed-forward neural networks LSTM networks are computationally more expensive. This can limit their scalability for large-scale datasets or constrained environments.

- Training LSTM networks can be more time-consuming compared to simpler models due to their computational complexity. So training LSTMs often requires more data and longer training times to achieve high performance.
- Since it is processed word by word in a sequential manner, it is hard to parallelize the work of processing the sentences.

## Applications of LSTM

---

Some of the famous applications of LSTM includes:

- **Language Modeling:** LSTMs have been used for natural language processing tasks such as language modeling, machine translation, and text summarization. They can be trained to generate coherent and grammatically correct sentences by learning the dependencies between words in a sentence.
- **Speech Recognition:** LSTMs have been used for speech recognition tasks such as transcribing speech to text and recognizing spoken commands. They can be trained to recognize patterns in speech and match them to the corresponding text.
- **Time Series Forecasting:** LSTMs have been used for time series forecasting tasks such as predicting stock prices, weather, and energy consumption. They can learn patterns in time series data and use them to make predictions about future events.
- **Anomaly Detection:** LSTMs have been used for anomaly detection tasks such as detecting fraud and network intrusion. They can be trained to identify patterns in data that deviate from the norm and flag them as potential anomalies.
- **Recommender Systems:** LSTMs have been used for recommendation tasks such as recommending movies, music, and books. They can learn patterns in user behavior and use them to make personalized recommendations.
- **Video Analysis:** LSTMs have been used for video analysis tasks such as object detection, activity recognition, and action classification. They can be used in combination with other neural network architectures, such as Convolutional Neural Networks (CNNs), to analyze video data and extract useful information.

## Conclusion

---

Long Short-Term Memory (LSTM) is a powerful type of recurrent neural network (RNN) that is well-suited for handling sequential data with long-term dependencies. It addresses the vanishing gradient problem, a common limitation of RNNs, by introducing a gating mechanism that controls the flow of information through the network. This allows LSTMs to learn and retain information from the past, making them effective for tasks like machine translation, speech recognition, and natural language processing.



## Also Check:

- [Long short-term memory \(LSTM\) RNN in Tensorflow](#)
- [Text Generation using Recurrent Long Short Term Memory Network](#)
- [Long Short Term Memory Networks Explanation](#)

## Frequently Asked Questions (FAQs)

---

### 1. What is LSTM?

LSTM is a type of recurrent neural network (RNN) that is designed to address the vanishing gradient problem, which is a common issue with RNNs. LSTMs have a special architecture that allows them to learn long-term dependencies in sequences of data, which makes them well-suited for tasks such as machine translation, speech recognition, and text generation.

### 2. How does LSTM work?

LSTMs use a cell state to store information about past inputs. This cell state is updated at each step of the network, and the network uses it to make predictions about the current input. The cell state is updated using a series of gates that control how much information is allowed to flow into and out of the cell.

### 3. What is the major difference between lstm and bidirectional lstm?

The vanishing gradient problem of the RNN is addressed by both LSTM and GRU, which differ in a few ways. These distinctions are as follows:

- Bidirectional LSTM can utilize information from both past and future, whereas standard LSTM can only utilize past info.
- Whereas GRU only employs two gates, LSTM uses three gates to compute the input of sequence data.
- Compared to LSTM, GRUs are typically faster and simpler.
- GRUs are favored for small datasets, while LSTMs are preferable for large datasets.

### 4. What is the difference between LSTM and Gated Recurrent Unit (GRU)?

LSTM has a cell state and gating mechanism which controls information flow, whereas GRU has a simpler single gate update mechanism. LSTM is more powerful but slower to train, while GRU is simpler and faster.

### 5. What is difference between LSTM and RNN?

- RNNs have a simple recurrent structure with unidirectional information flow.
- LSTMs have a gating mechanism that controls information flow and a cell state for long-term memory.
- LSTMs generally outperform RNNs in tasks that require learning long-term dependencies.