Analytics Vidhya

What is LSTM? Introduction to Long Short-Term Memory

ADVANCED DEEP LEARNING VIDEOS

Introduction

Long Short-Term Memory Networks is a deep learning, sequential neural network that allows information to persist. It is a special type of Recurrent Neural Network which is capable of handling the vanishing gradient problem faced by RNN. LSTM was designed by Hochreiter and Schmidhuber that resolves the problem caused by traditional rnns and machine learning algorithms. LSTM Model can be implemented in Python using the Keras library.

Let's say while watching a video, you remember the previous scene, or while reading a book, you know what happened in the earlier chapter. RNNs work similarly; they remember the previous information and use it for processing the current input. The shortcoming of RNN is they cannot remember long-term dependencies due to vanishing gradient. LSTMs are explicitly designed to avoid long-term dependency problems.

This article will cover all the basics about LSTM, including its meaning, architecture, applications, and gates.

Learning Objectives

- Understand what LSTM is.
- Understand the architecture and working of an LSTM network.
- Learn about the different parts/gates in an LSTM unit.

Note: If you are more interested in learning concepts in an Audio-Visual format, We have the tutorial of this entire article explained in the video below. If not, you may continue reading.

Table of contents

- What is LSTM?
- LSTM Architecture
- Forget Gate
- Input Gate
- New Information
- Output Gate
- LTSM vs RNN
- What are Bidirectional LSTMs?
- Conclusion

- Frequently Asked Questions

What is LSTM?

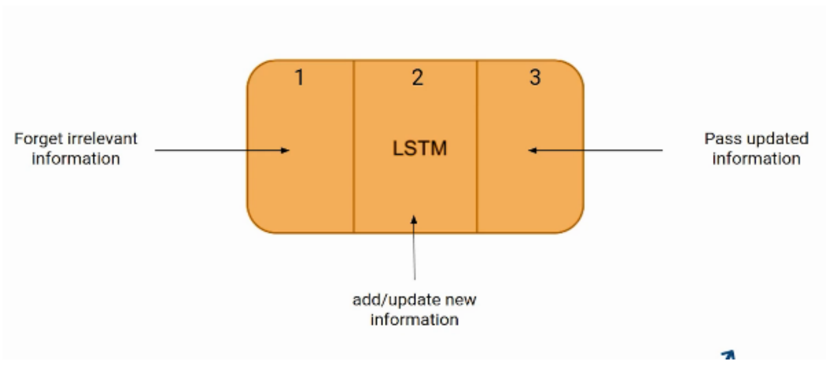
LSTM (Long Short-Term Memory) is a recurrent neural network (RNN) architecture widely used in Deep Learning. It excels at capturing long-term dependencies, making it ideal for sequence prediction tasks.

Unlike traditional neural networks, LSTM incorporates feedback connections, allowing it to process entire sequences of data, not just individual data points. This makes it highly effective in understanding and predicting patterns in sequential data like time series, text, and speech.

LSTM has become a powerful tool in artificial intelligence and deep learning, enabling breakthroughs in various fields by uncovering valuable insights from sequential data.

LSTM Architecture

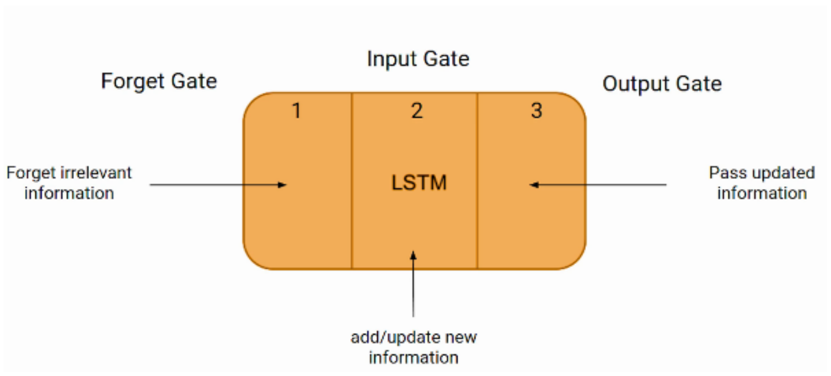
In the introduction to long short-term memory, we learned that it resolves the vanishing gradient problem faced by RNN, so now, in this section, we will see how it resolves this problem by learning the architecture of the LSTM. At a high level, LSTM works very much like an RNN cell. Here is the internal functioning of the LSTM network. The LSTM network architecture consists of three parts, as shown in the image below, and each part performs an individual function.



The Logic Behind LSTM

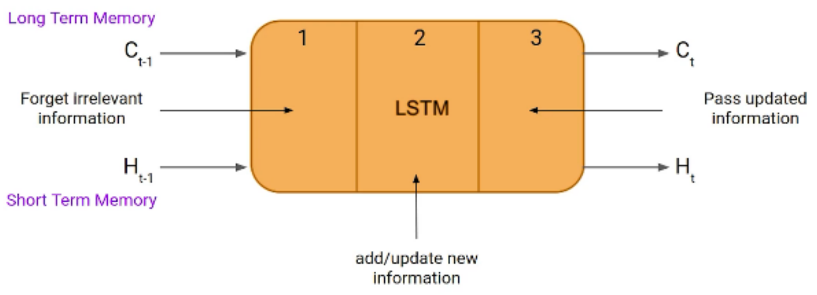
The first part chooses whether the information coming from the previous timestamp is to be remembered or is irrelevant and can be forgotten. In the second part, the cell tries to learn new information from the input to this cell. At last, in the third part, the cell passes the updated information from the current timestamp to the next timestamp. This one cycle of LSTM is considered a single-time step.

These three parts of an LSTM unit are known as gates. They control the flow of information in and out of the memory cell or lstm cell. The first gate is called **Forget gate, the** second gate is known as **the Input gate,** and the last one is **the Output gate**. An LSTM unit that consists of these three gates and a memory cell or lstm cell can be considered as a layer of neurons in traditional feedforward neural network, with each neuron having a hidden layer and a current state.

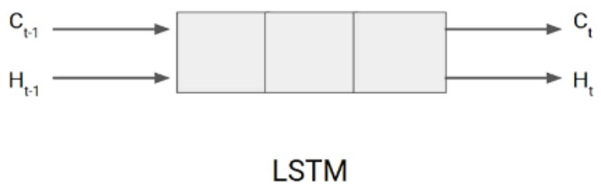


Just like a simple RNN, an LSTM also has a hidden state where H(t-1) represents the hidden state of the previous timestamp and Ht is the hidden state of the current timestamp. In addition to that, LSTM also has a cell state represented by C(t-1) and C(t) for the previous and current timestamps, respectively.

Here the hidden state is known as Short term memory, and the cell state is known as Long term memory. Refer to the following image.



It is interesting to note that the cell state carries the information along with all the timestamps.



Bob is a nice person. Dan on the other hand is evil.

Example of LTSM Working

Let's take an example to understand how LSTM works. Here we have two sentences separated by a full stop. The first sentence is "Bob is a nice person," and the second sentence is "Dan, on the Other hand, is evil”. It is very clear, in the first sentence, we are talking about Bob, and as soon as we encounter the full stop(.), we started talking about Dan.

As we move from the first sentence to the second sentence, our network should realize that we are no more talking about Bob. Now our subject is Dan. Here, the Forget gate of the network allows it to forget about it. Let’s understand the roles played by these gates in LSTM architecture.

Forget Gate

In a cell of the LSTM neural network, the first step is to decide whether we should keep the information from the previous time step or forget it. Here is the equation for forget gate.

Forget Gate:

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

Let’s try to understand the equation, here

- Xt: input to the current timestamp.
- Uf: weight associated with the input
- Ht-1: The hidden state of the previous timestamp
- Wf: It is the weight matrix associated with the hidden state

Later, a sigmoid function is applied to it. That will make ft a number between 0 and 1. This ft is later multiplied with the cell state of the previous timestamp, as shown below.

$C_{t-1} * f_t = 0$ ...if $f_t = 0$ (forget everything)

$C_{t-1} * f_t = C_{t-1}$ ...if $f_t = 1$ (forget nothing)

Input Gate

Let’s take another example.

“Bob knows swimming. He told me over the phone that he had served the navy for four long years.”

So, in both these sentences, we are talking about Bob. However, both give different kinds of information about Bob. In the first sentence, we get the information that he knows swimming. Whereas the second sentence tells, he uses the phone and served in the navy for four years.

Now just think about it, based on the context given in the first sentence, which information in the second sentence is critical? First, he used the phone to tell, or he served in the navy. In this context, it doesn’t matter whether he used the phone or any other medium of communication to pass on the information. The fact that he was in the navy is important information, and this is something we want our model to remember for future computation. This is the task of the Input gate.

The input gate is used to quantify the importance of the new information carried by the input. Here is the equation of the input gate

Input Gate:

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

Here,

- Xt: Input at the current timestamp t
- Ui: weight matrix of input
- Ht-1: A hidden state at the previous timestamp
- Wi: Weight matrix of input associated with hidden state

Again we have applied the sigmoid function over it. As a result, the value of I at timestamp t will be between 0 and 1.

New Information

- $N_t = \tanh(x_t * U_c + H_{t-1} * W_c)$ (new information)

Now the new information that needed to be passed to the cell state is a function of a hidden state at the previous timestamp t-1 and input x at timestamp t. The activation function here is tanh. Due to the tanh function, the value of new information will be between -1 and 1. If the value of Nt is negative, the information is subtracted from the cell state, and if the value is positive, the information is added to the cell state at the current timestamp.

However, the Nt won't be added directly to the cell state. Here comes the updated equation:

$$C_t = f_t * C_{t-1} + i_t * N_t \text{ (updating cell state)}$$

Here, Ct-1 is the cell state at the current timestamp, and the others are the values we have calculated previously.

Output Gate

Now consider this sentence.

"Bob single-handedly fought the enemy and died for his country. For his contributions, brave______."

During this task, we have to complete the second sentence. Now, the minute we see the word brave, we know that we are talking about a person. In the sentence, only Bob is brave, we can not say the enemy is brave, or the country is brave. So based on the current expectation, we have to give a relevant word to fill in the blank. That word is our output, and this is the function of our Output gate.

Here is the equation of the Output gate, which is pretty similar to the two previous gates.

Output Gate:

- $o_t = \sigma(x_t * U_o + H_{t-1} * W_o)$

Its value will also lie between 0 and 1 because of this sigmoid function. Now to calculate the current hidden state, we will use Ot and tanh of the updated cell state. As shown below.

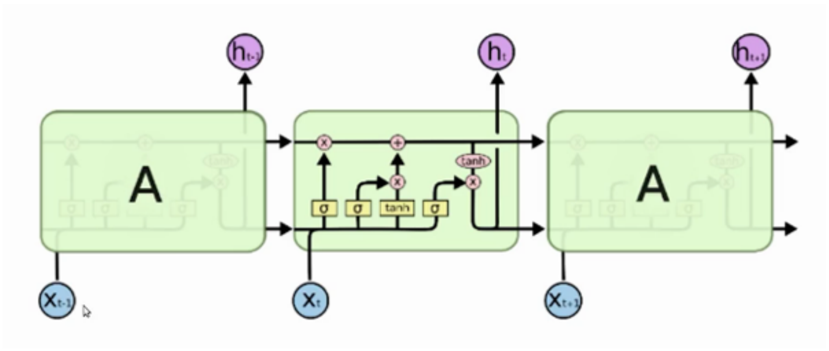
$$H_t = o_t * \tanh(C_t)$$

It turns out that the hidden state is a function of Long term memory (Ct) and the current output. If you need to take the output of the current timestamp, just apply the SoftMax activation on hidden state Ht.

$$\text{Output} = \text{Softmax}(H_t)$$

Here the token with the maximum score in the output is the prediction.

This is the More intuitive diagram of the LSTM network.



This diagram is taken from an interesting blog. I urge you all to go through it. Here is the link: [Understanding LSTM Networks]()

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

What are Bidirectional LSTMs?

Bidirectional LSTMs (Long Short-Term Memory) are a type of recurrent neural network (RNN) architecture that processes input data in both forward and backward directions. In a traditional LSTM, the information flows only from past to future, making predictions based on the preceding context. However, in bidirectional LSTMs, the network also considers future context, enabling it to capture dependencies in both directions.

The bidirectional LSTM comprises two LSTM layers, one processing the input sequence in the forward direction and the other in the backward direction. This allows the network to access information from past and future time steps simultaneously. As a result, bidirectional LSTMs are particularly useful for tasks that require a comprehensive understanding of the input sequence, such as natural language processing tasks like sentiment analysis, machine translation, and named entity recognition.

By incorporating information from both directions, bidirectional LSTMs enhance the model's ability to capture long-term dependencies and make more accurate predictions in complex sequential data.

Conclusion

In this article, we covered the basics and sequential architecture of a Long Short-Term Memory Network model. Knowing how it works helps you design an LSTM model with ease and better understanding. It is an important topic to cover as LSTM models are widely used in artificial intelligence for natural language processing tasks like language modeling and machine translation. Some other applications of lstm are speech recognition, image captioning, handwriting recognition, time series forecasting by learning time series data, etc.

If you are looking to kick-start your Data Science Journey and want every topic under one roof, your search stops here. Check out Analytics Vidhya's Certified AI & ML BlackBelt Plus Program.

Key Takeaways

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

Frequently Asked Questions

Q1. What is LSTM?

A. Long Short-Term Memory Networks is a deep learning, sequential neural net that allows information to persist. It is a special type of Recurrent Neural Network which is capable of handling the vanishing gradient problem faced by traditional RNN.

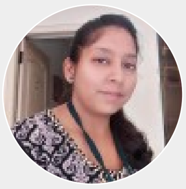
Q2. What is the difference between LSTM and Gated Recurrent Unit (GRU)?

A. LSTM and GRU are both variants of RNN that are used to resolve the vanishing gradient issue of the RNN, but they have some differences, which are:
1. LSTM uses three gates to compute the input of sequence data where, whereas GRU uses only two gates.
2. GRUs are generally simpler and faster than LSTM.
3. LSTMs are preferred for large datasets, whereas for small datasets GRUs are preferred.

Q3. What is the major difference between lstm and bidirectional lstm?

A. The major difference between the two is that LSTM can process the input sequence in a forward or backward direction at a time, whereas bidirectional lstm can process the input sequence in a forward or backward direction simultaneously.

Article Url - https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/



Shipra Saxena