



Understanding of LSTM Networks

[Read](#)[Courses](#)[Jobs](#)

This article talks about the problems of conventional RNNs, namely, the vanishing and exploding gradients, and provides a convenient solution to these problems in the form of Long Short Term Memory (LSTM). Long Short-Term Memory is an advanced version of recurrent neural network (RNN) architecture that was designed to model chronological sequences and their long-range dependencies more precisely than conventional RNNs.

Introduction to LSTM

LSTM networks are an extension of recurrent neural networks ([RNNs](#)) mainly introduced to handle situations where RNNs fail.

- It fails to store information for a longer period of time. At times, a reference to certain information stored quite a long time ago is required to predict the current output. But RNNs are absolutely incapable of handling such “long-term dependencies”.
- There is no finer control over which part of the context needs to be carried forward and how much of the past needs to be ‘forgotten’.
- Other issues with RNNs are exploding and vanishing gradients (explained later) which occur during the training process of a network through backtracking.

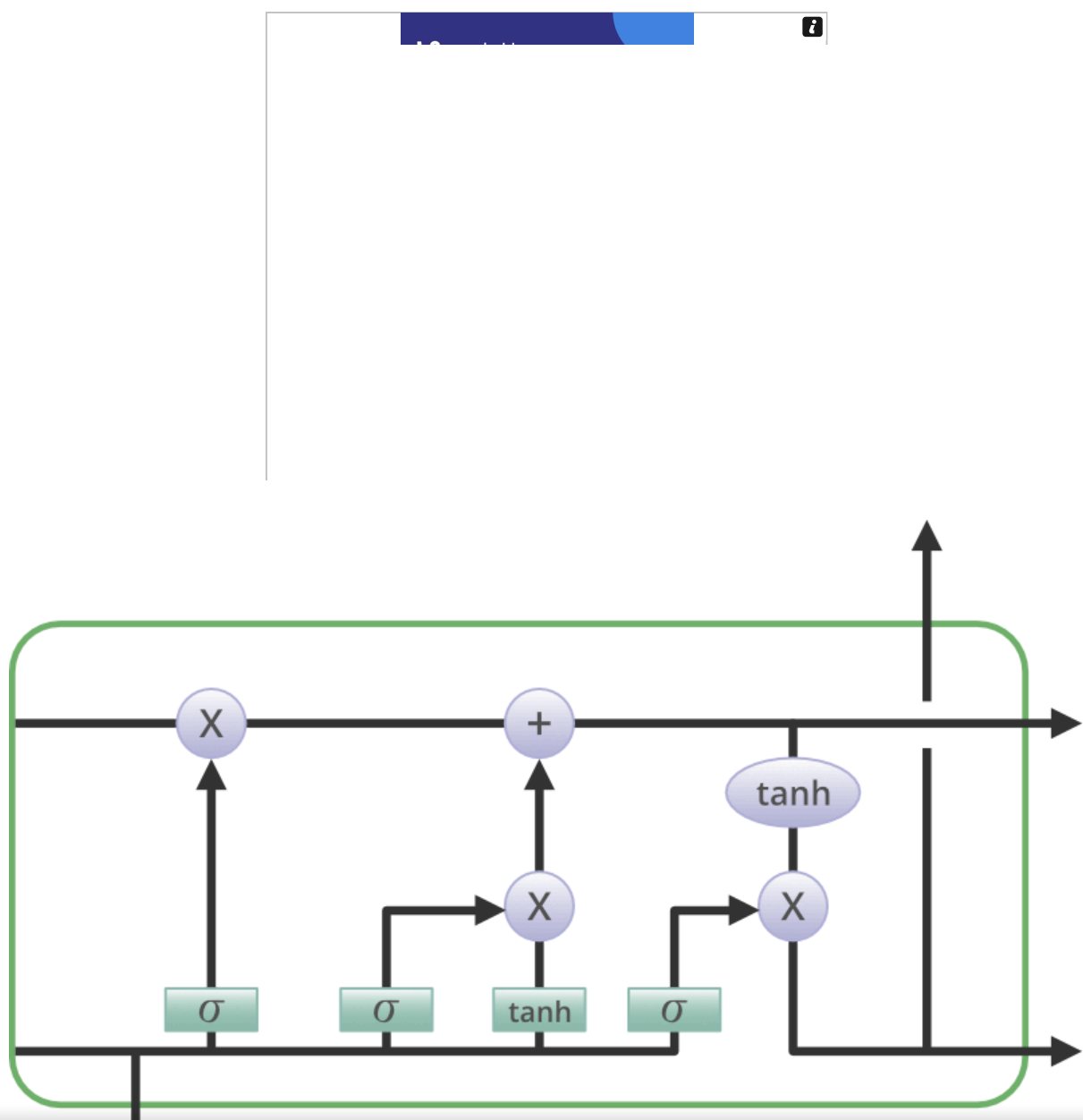
Thus, Long Short-Term Memory ([LSTM](#)) was brought into the picture. It has been so designed that the vanishing gradient problem is almost completely removed, while the training model is left unaltered. Long-time lags in certain problems are bridged using LSTMs which also handle noise, distributed representations, and continuous values. With LSTMs, there is no need to keep a finite number of states from beforehand as required in the hidden [Markov model](#) (HMM). LSTMs provide us with a large range of parameters



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

The basic difference between the architectures of RNNs and LSTMs is that the hidden layer of LSTM is a gated unit or gated cell. It consists of four layers that interact with one another in a way to produce the output of that cell along with the cell state. These two things are then passed onto the next hidden layer. Unlike RNNs which have got only a single neural net layer of tanh, LSTMs comprise three logistic sigmoid gates and one tanh layer. Gates have been introduced in order to limit the information that is passed through the cell. They determine which part of the information will be needed by the next cell and which part is to be discarded. The output is usually in the range of 0-1 where '0' means 'reject all' and '1' means 'include all'.

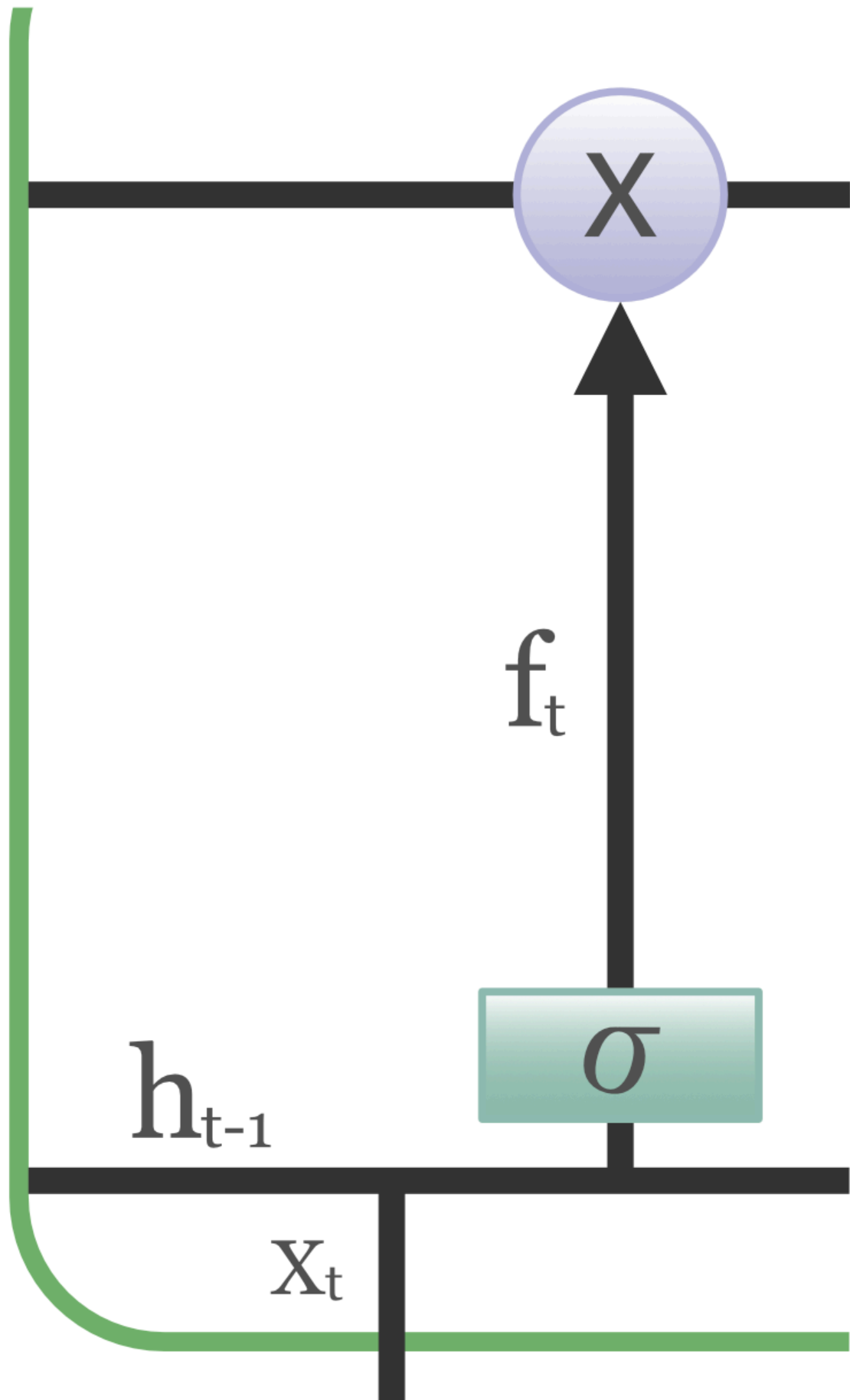


We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Information is retained by the cells and the memory manipulations are done by the gates. There are three gates which are explained below:

Forget Gate

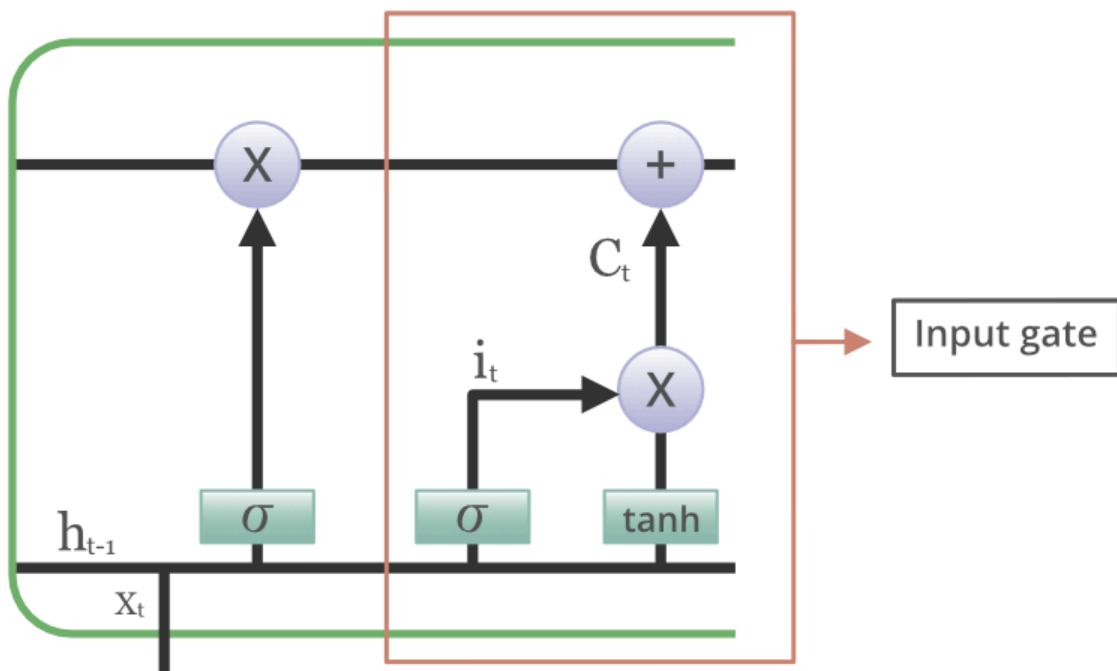
The information that is no longer useful in the cell state is removed with the forget gate. Two inputs x_t (input at the particular time) and h_{t-1} (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias. The resultant is passed through an activation 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.



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Input gate

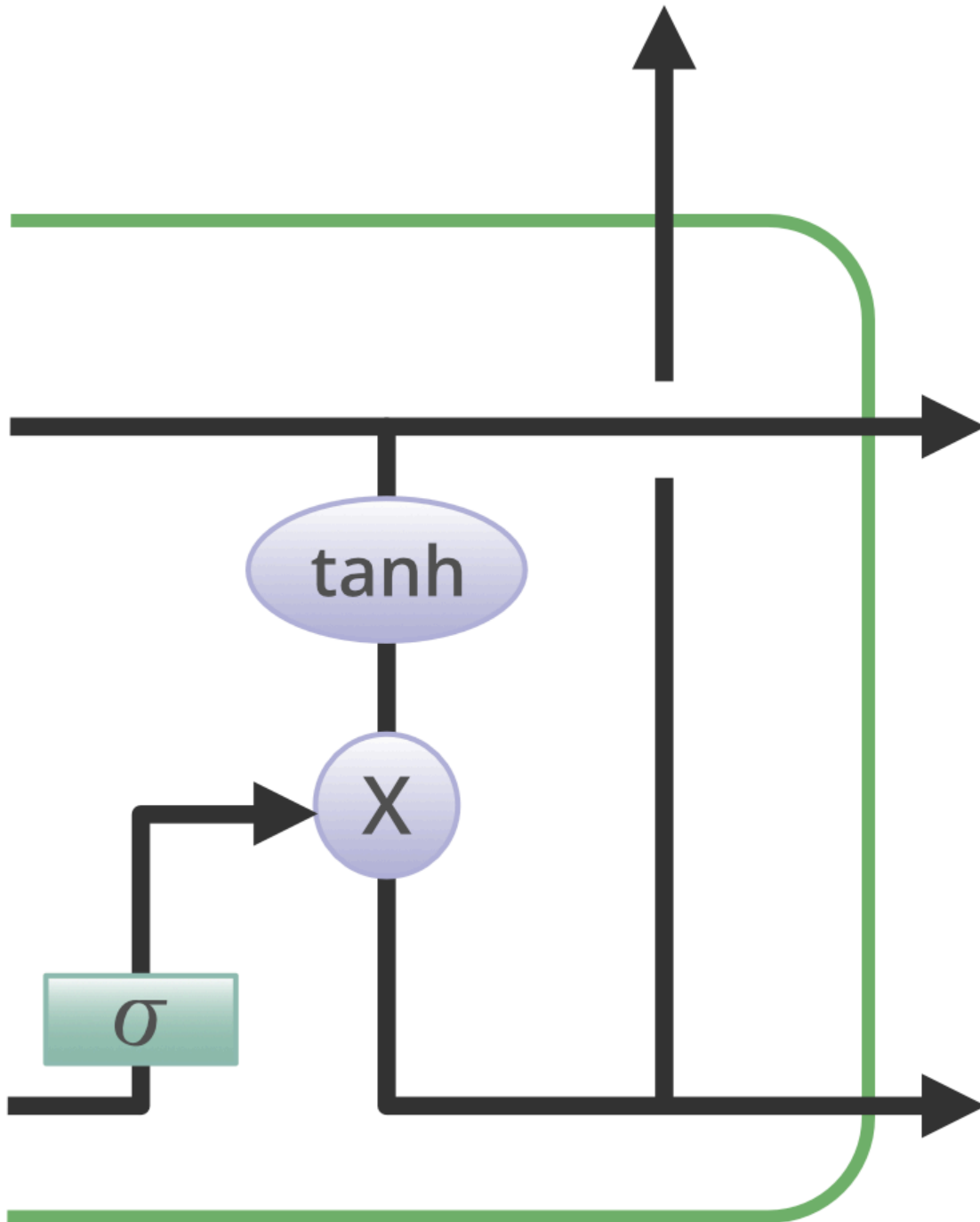
The addition of useful information to the cell state is done by the input gate. First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs h_{t-1} and x_t . Then, a vector is created using the tanh function that gives 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 useful information.



Input gate in the LSTM cell

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 the tanh function on the cell. Then, the information is regulated using the [sigmoid function](#) and filtered 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.



Output gate in the LSTM cell

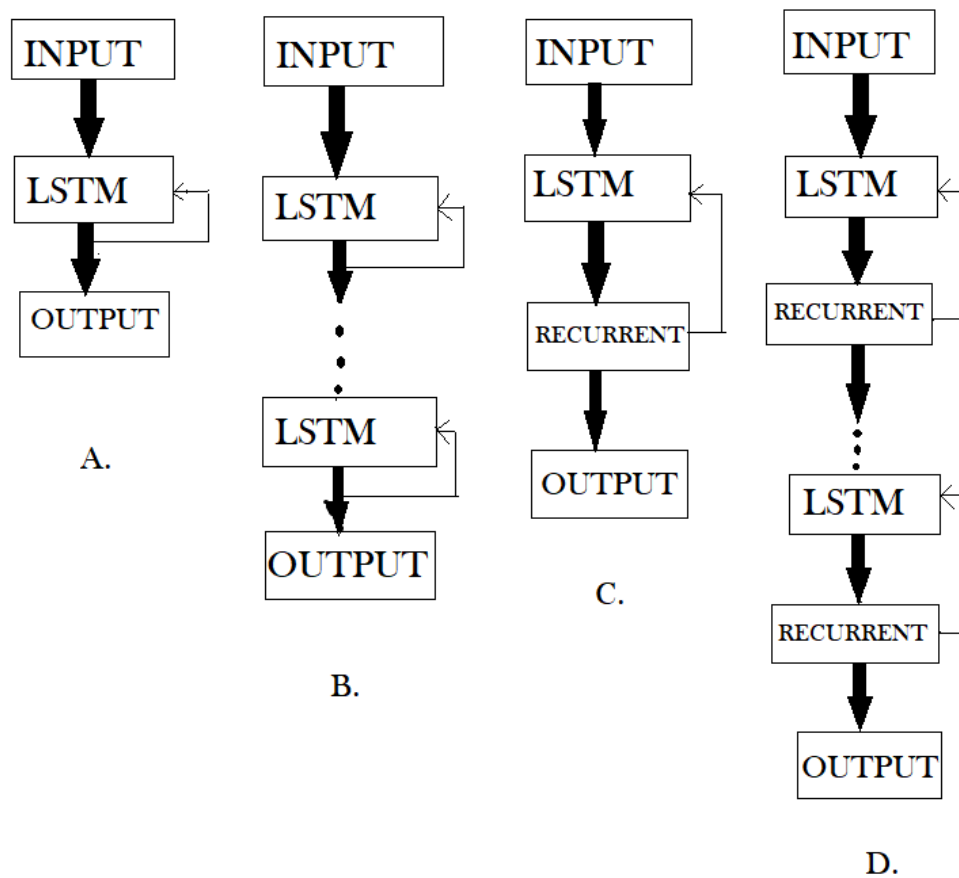
Variations in LSTM Networks

With the increasing popularity of LSTMs, various alterations have been tried on the conventional LSTM architecture to simplify the internal design of cells to make them work in a more efficient way and to reduce computational complexity. Gers and Schmidhuber introduced peephole connections which allowed gate layers to have knowledge about the cell state at every instant.

Some LSTMs also made use of a coupled input and forget gate instead of

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

improved the design complexity by reducing the number of gates. It uses a combination of the cell state and hidden state and also an update gate which has forgotten and input gates merged into it.



LSTM(Figure-A), DLSTM(Figure-B), LSTMP(Figure-C) and DLSTMP(Figure-D)

1. Figure-A represents what a basic LSTM network looks like. Only one layer of LSTM between an input and output layer has been shown here.
2. Figure B represents Deep LSTM which includes a number of LSTM layers in between the input and output. The advantage is that the input values fed to the network not only go through several LSTM layers but also propagate through time within one LSTM cell. Hence, parameters are well distributed within multiple layers. This results in a thorough process of inputs in each time step.
3. Figure C represents LSTM with the Recurrent Projection layer where the recurrent connections are taken from the projection layer to the LSTM

4. Figure D represents Deep LSTM with a Recurrent Projection Layer consisting of multiple LSTM layers where each layer has its own projection layer. The increased depth is quite useful in the case where the memory size is too large. Having increased depth prevents overfitting in models as the inputs to the network need to go through many nonlinear functions.

GRU v/s LSTM

In spite of being quite similar to LSTMs, [GRUs](#) have never been so popular. But what are GRUs? GRU stands for Gated Recurrent Units. As the name suggests, these recurrent units, proposed by Cho, are also provided with a gated mechanism to effectively and adaptively capture dependencies of different time scales. They have an update gate and a reset gate. The former is responsible for selecting what piece of knowledge is to be carried forward, whereas the latter lies in between two successive recurrent units and decides how much information needs to be forgotten.

Activation at time t :

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j * \tilde{h}_t^j$$

Update gate:

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1})^j$$

Candidate activation:

$$\tilde{h}_t^j = \tanh(W x_t + U(r_t \otimes h_{t-1}))^j$$

Reset gate:

$$r_t^j = \sigma(W_r x_t + U_r h_{t-1})^j$$

Another striking aspect of GRUs is that they do not store cell state in any way, hence, they are unable to regulate the amount of memory content to which the next unit is exposed. Instead, LSTMs regulate the amount of new information being included in the cell. On the other hand, the GRU controls the information flow from the previous activation when computing the new,

Applications of LSTM Networks

LSTM models need to be trained with a training dataset prior to their employment in real-world applications. Some of the most demanding applications are discussed below:

1. [Language modeling](#) or text generation, involves the computation of words when a sequence of words is fed as input. Language models can be operated at the character level, n-gram level, sentence level, or even paragraph level.
2. Image processing involves performing an analysis of a picture and concluding its result into a sentence. For this, it's required to have a dataset comprising a good amount of pictures with their corresponding descriptive captions. A model that has already been trained is used to predict features of images present in the dataset. This is photo data. The dataset is then processed in such a way that only the words that are most suggestive are present in it. This is text data. Using these two types of data, we try to fit the model. The work of the model is to generate a descriptive sentence for the picture one word at a time by taking input words that were predicted previously by the model and also the image.
3. Speech and Handwriting Recognition.
4. Music generation is quite similar to that of text generation where LSTMs predict musical notes instead of text by analyzing a combination of given notes fed as input.
5. [Language Translation](#) involves mapping a sequence in one language to a sequence in another language. Similar to image processing, a dataset, containing phrases and their translations, is first cleaned and only a part of it is used to train the model. An encoder-decoder LSTM model is used which first converts the input sequence to its vector representation (encoding) and then outputs it to its translated version.

Drawbacks of Using LSTM Networks

As it is said, everything in this world comes with its own advantages and disadvantages, LSTMs too, have a few drawbacks which are discussed below:

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

- its evaluation. Moreover, the cell has become quite complex now with additional features (such as forget gates) being brought into the picture.
2. They require a lot of resources and time to get trained and become ready for real-world applications. In technical terms, they need high memory bandwidth because of the linear layers present in each cell which the system usually fails to provide. Thus, hardware-wise, LSTMs become quite inefficient.
 3. With the rise of data mining, developers are looking for a model that can remember past information for a longer time than LSTMs. The source of inspiration for such kind of model is the human habit of dividing a given piece of information into small parts for easy remembrance.
 4. LSTMs get affected by different random weight initialization and hence behave quite similarly to that of a feed-forward neural net. They prefer small-weight initialization instead.
 5. LSTMs are prone to overfitting and it is difficult to apply the dropout algorithm to curb this issue. Dropout is a regularization method where input and recurrent connections to LSTM units are probabilistically excluded from activation and weight updates while training a network.

Whether you're preparing for your first job interview or aiming to upskill in this ever-evolving tech landscape, [GeeksforGeeks Courses](#) are your key to success. We provide top-quality content at affordable prices, all geared towards accelerating your growth in a time-bound manner. Join the millions we've already empowered, and we're here to do the same for you. Don't miss out - [check it out now!](#)

Get paid for your published articles and stand a chance to win tablet, smartwatch and exclusive GfG goodies! Submit your entries in Dev Scriptor 2024 today.

Participate in Three 90 Challenge! Enroll in any GeeksforGeeks course and get 90% refund by completing 90% course. [Explore](#) offer now.

Last Updated : 05 Jun, 2023

11

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

ANN - Bidirectional Associative Memory (BAM)

Share your thoughts in the comments

Add Your Comment

Similar Reads

Bidirectional LSTM in NLP

LSTM - Derivation of Back propagation through time

Emotion Detection using Bidirectional LSTM

LSTM Based Poetry Generation Using NLP in Python

Basic Understanding of Bayesian Belief Networks

Understanding Multi-Layer Feed Forward Networks

Difference Between Feed-Forward Neural Networks and Recurrent Neural Networks

ML | Understanding Data Processing

Understanding Types of Means | Set 1

Understanding Types of Mean | Set 2

A **aditianu1...**

Follow

Article Tags : [Deep-Learning](#) , [Neural Network](#) , [Machine Learning](#)

Practice Tags : [Machine Learning](#)

Additional Information

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).



A-143, 9th Floor, Sovereign Corporate
Tower, Sector-136, Noida, Uttar Pradesh -
201305



Company

About Us
Legal
Careers
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program

Explore

Job-A-Thon Hiring Challenge
Hack-A-Thon
GfG Weekly Contest
Offline Classes (Delhi/NCR)
DSA in JAVA/C++
Master System Design
Master CP
GeeksforGeeks Videos
Geeks Community

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain

Android Tutorial

All Cheat Sheets

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning Tutorial
ML Maths
Data Visualisation Tutorial
Pandas Tutorial
NumPy Tutorial
NLP Tutorial
Deep Learning Tutorial

Python

Python Programming Examples
Django Tutorial
Python Projects
Python Tkinter
Web Scraping
OpenCV Python Tutorial
Python Interview Question

DevOps

Git
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide

HTML & CSS

HTML
CSS
Web Templates
CSS Frameworks
Bootstrap
Tailwind CSS
SASS
LESS
Web Design

Computer Science

GATE CS Notes
Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths

Competitive Programming

Top DS or Algo for CP
Top 50 Tree
Top 50 Graph
Top 50 Array
Top 50 String
Top 50 DP
Top 15 Websites for CP

JavaScript

JavaScript Examples
TypeScript
ReactJS
NextJS

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

[Interview Questions](#)[Web Browser](#)

NCERT Solutions

[Class 12](#)[Class 11](#)[Class 10](#)[Class 9](#)[Class 8](#)[Complete Study Material](#)

Commerce

[Accountancy](#)[Business Studies](#)[Economics](#)[Management](#)[HR Management](#)[Finance](#)[Income Tax](#)

SSC/ BANKING

[SSC CGL Syllabus](#)[SBI PO Syllabus](#)[SBI Clerk Syllabus](#)[IBPS PO Syllabus](#)[IBPS Clerk Syllabus](#)[SSC CGL Practice Papers](#)

Companies

[META Owned Companies](#)[Alphabet Owned Companies](#)[TATA Group Owned Companies](#)[Reliance Owned Companies](#)[Fintech Companies](#)[EdTech Companies](#)

School Subjects

[Mathematics](#)[Physics](#)[Chemistry](#)[Biology](#)[Social Science](#)[English Grammar](#)

UPSC Study Material

[Polity Notes](#)[Geography Notes](#)[History Notes](#)[Science and Technology Notes](#)[Economy Notes](#)[Ethics Notes](#)[Previous Year Papers](#)

Colleges

[Indian Colleges Admission & Campus Experiences](#)[List of Central Universities - In India](#)[Colleges in Delhi University](#)[IIT Colleges](#)[NIT Colleges](#)[IIIT Colleges](#)

Preparation Corner

[Company-Wise Recruitment Process](#)[Resume Templates](#)[Aptitude Preparation](#)[Puzzles](#)[Company-Wise Preparation](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

[JEE Advanced](#)[Software Testing](#)[GATE CS](#)[Product Management](#)[NEET](#)[SAP](#)[UGC NET](#)[SEO - Search Engine Optimization](#)[Linux](#)[Excel](#)

Free Online Tools

[Typing Test](#)[Image Editor](#)[Code Formatters](#)[Code Converters](#)[Currency Converter](#)[Random Number Generator](#)[Random Password Generator](#)

Write & Earn

[Write an Article](#)[Improve an Article](#)[Pick Topics to Write](#)[Share your Experiences](#)[Internships](#)

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).