

★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



What is LSTM , peephole LSTM and GRU?



Jaimin Mungalpara · [Follow](#)

Published in Nerd For Tech

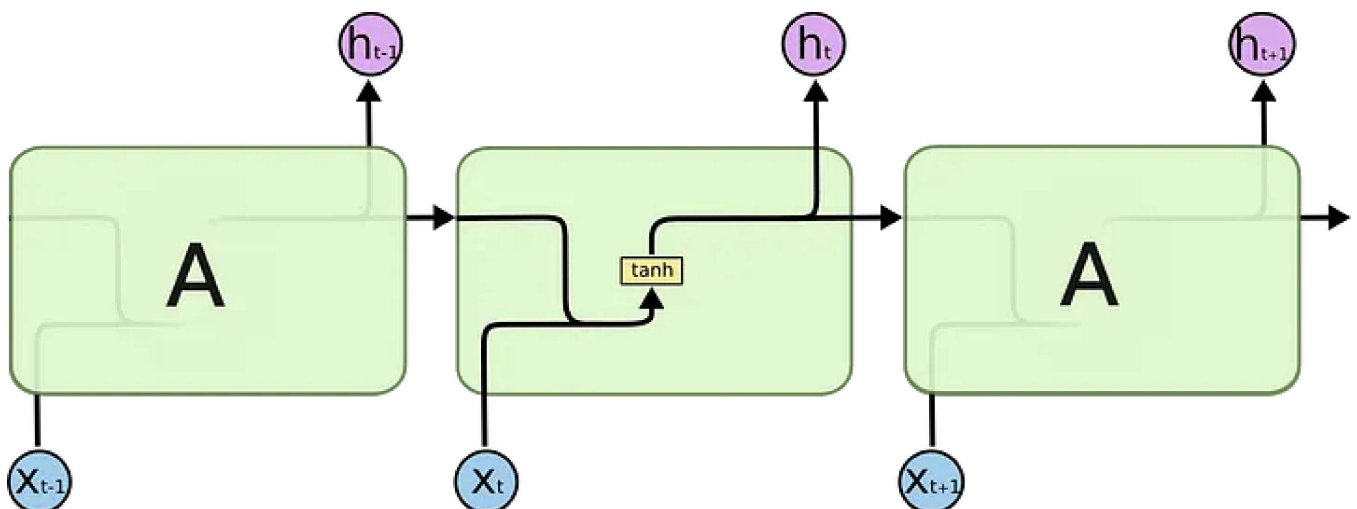
5 min read · Feb 5, 2021

Listen

Share

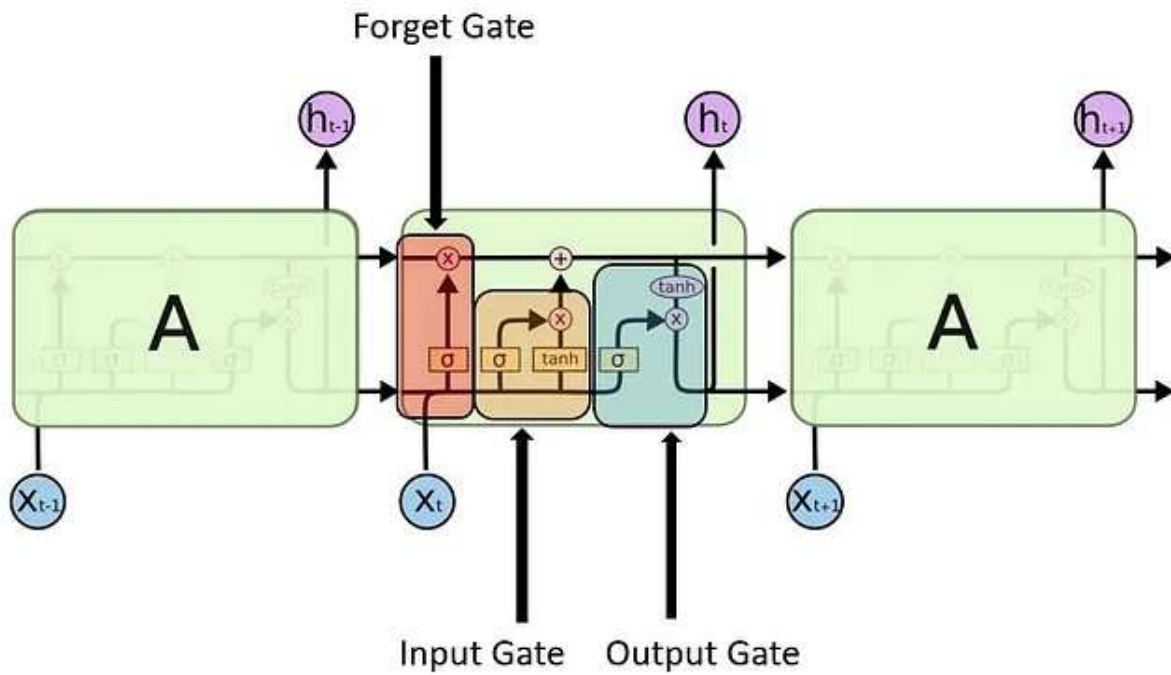
More

Long Short Term Memory (LSTM) was introduced by [Hochreiter & Schmidhuber \(1997\)](#) and it was refined by many researchers. LSTM is special kind of RNN which can remember long term dependencies. LSTM are specially designed to avoid the problems which are faced in RNN. You can learn about RNN in my previous article [Understanding RNN](#). The architectural behavior made it strong t remember long term dependencies. In simple RNN network, it has simple repeating neural network such a simple tanh or relu network which is represented in below figure.



The repeating module in a standard RNN contains a single layer. Image taken from <https://colah.github.io/>

Well, LSTM is also having a same kind of chain structure but the repeating module does not have simple neural network architecture. These repeating module contains 4 neural networks.



LSTM Gates

The notations in above figure can be explained like this. In this notation yellow boxes represents neural networks. Pink dots are point wise operation weather it would be vector multiplication or addition. Merger of line is concatenation and splitting of line means same content is going at different location.

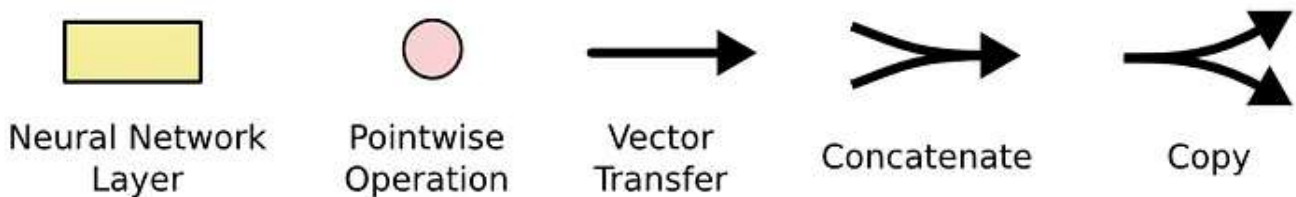


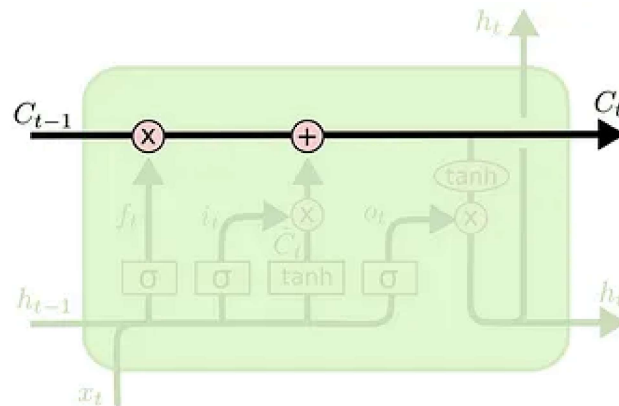
Image taken from <https://colah.github.io/>

There are 3 Gates in LSTM architecture which are

1. Forget Gate
2. Input Gate
3. Output Gate

The Core Idea Behind LSTM

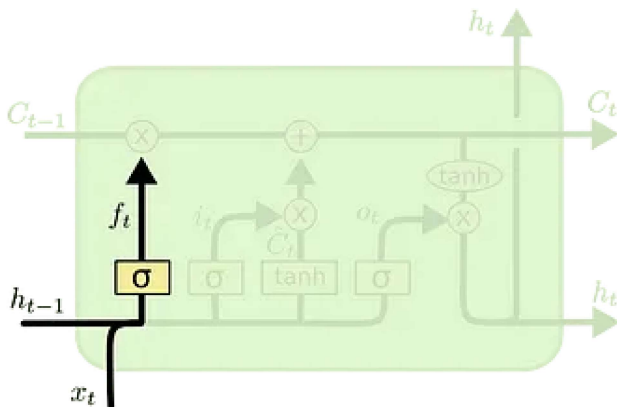
The top of the diagram contains cell state, it is a path in which information can pass easily with some minor linear operations. Gates can add or remove information in the cell state.



The gates are composed of sigmoid neural network and pointwise multiplication operation. As sigmoid given an output 0 & 1 this means if it is 0 then nothing would be passed through and if 1 then everything would be passed.

Forget Gate

This is the first step in LSTM network which decides which information would be passed through the cell state and this decision is taken by forget gate.

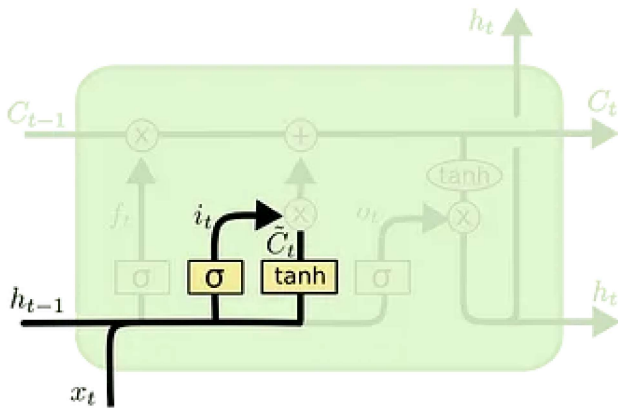


$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$$

It takes h_{t-1} and x_t as input and output is given as 0 and 1 which is then pointwise multiplied with C_{t-1} and finally it will decide which information would be passed through. If we are working with some context-based data and the context is changed, this forget cell will discard the information which is not relevant to the context.

Input Gate

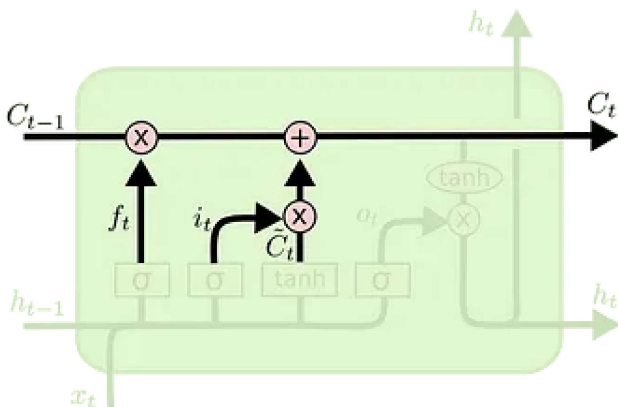
In this step will decide which information is going to be stored in cell state. This operation is one in two step. First, a sigmoid neural network decides which values we will update and a tanh layer that creates a vector of new candidate values, C_t , that could be added to the state.



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

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

Now we have to update cell state which we got at C_{t-1} to C_t .

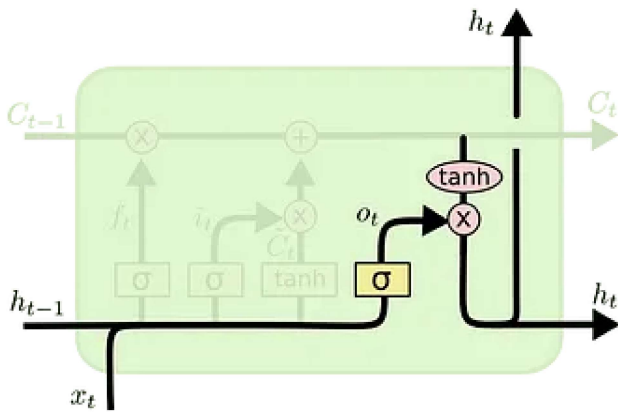


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

We multiply old state C_{t-1} with f_t and forget the things which are not required. Then we add $i_t * C_t$ to update the context which we need to remember. Here we add new information in context to be remembered and passed to next stage.

Output Gate

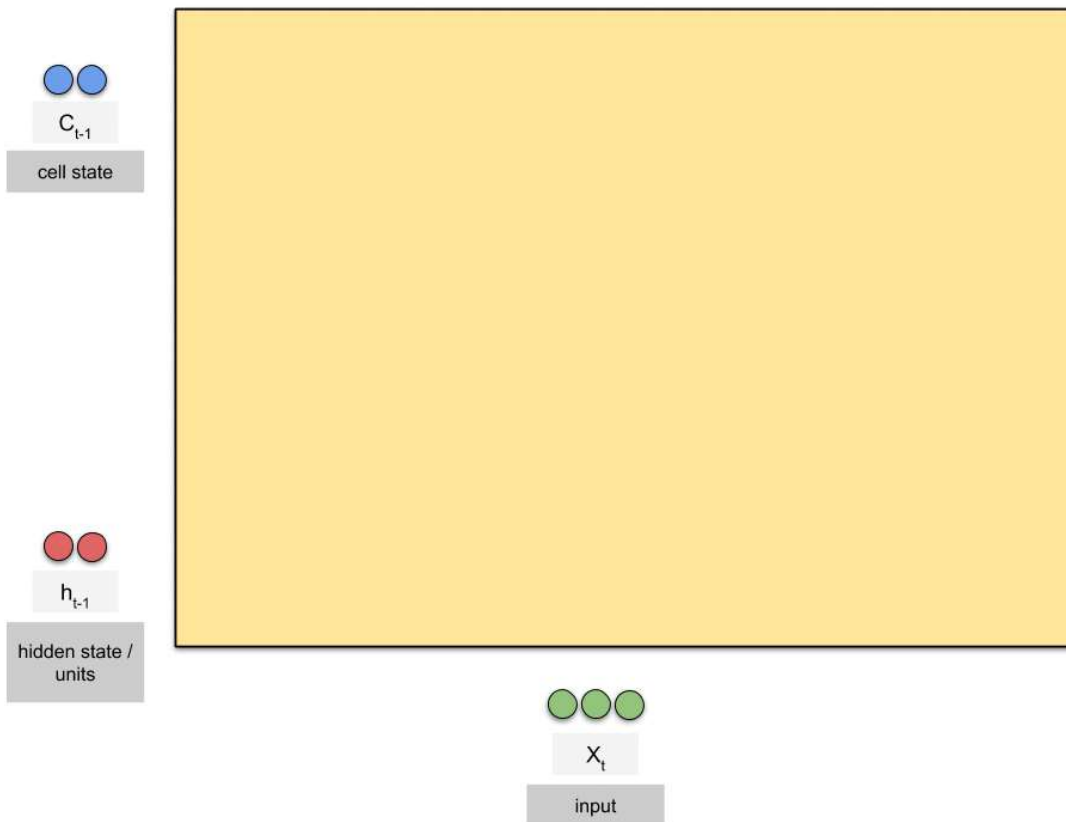
At this stage, we have to decide about what we are going to send in output. This would be based on our cell state, but we run a sigmoid neural network which decides what parts of the cell state we're going to output. Then, we put the cell state through tanh (to push the values to be between -1 and 1) and multiply it by the output of the sigmoid gate, so that we can decide about output data.



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

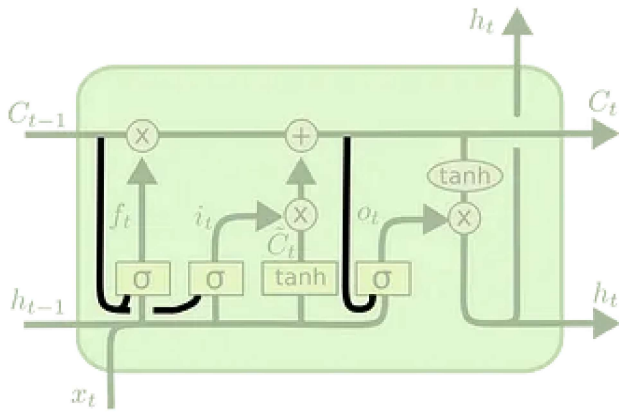
$$h_t = o_t * \tanh(C_t)$$

Here we can see whole idea in one gif about LSTM.



Peephole Architecture

Until now we have seen simple LSTM network but this architecture is modified along with time in each and every research paper. One popular LSTM variant, introduced by Gers & Schmidhuber (2000), is adding “peephole connections.” This means that we let the gate layers look at the cell state.



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

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

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

In this peephole connection we can see that all the gates are having an input along

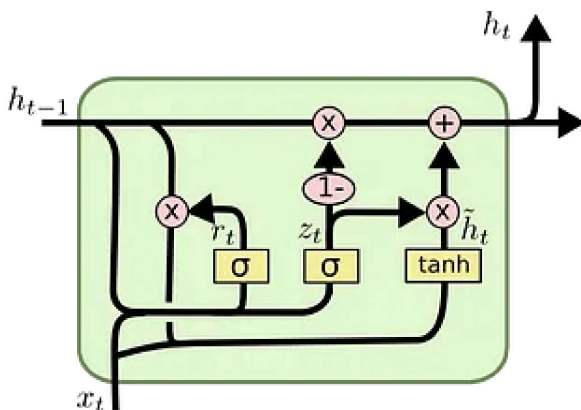
Open in app ↗



Search Medium



Another variation on the LSTM is the Gated Recurrent Unit, or GRU, introduced by [Cho, et al. \(2014\)](#). it combines the forget and input gate into update gate which is newly added in this architecture. It also merges the cell state and hidden state. The resulting model is simpler than traditional LSMT and it is growing more popularity.



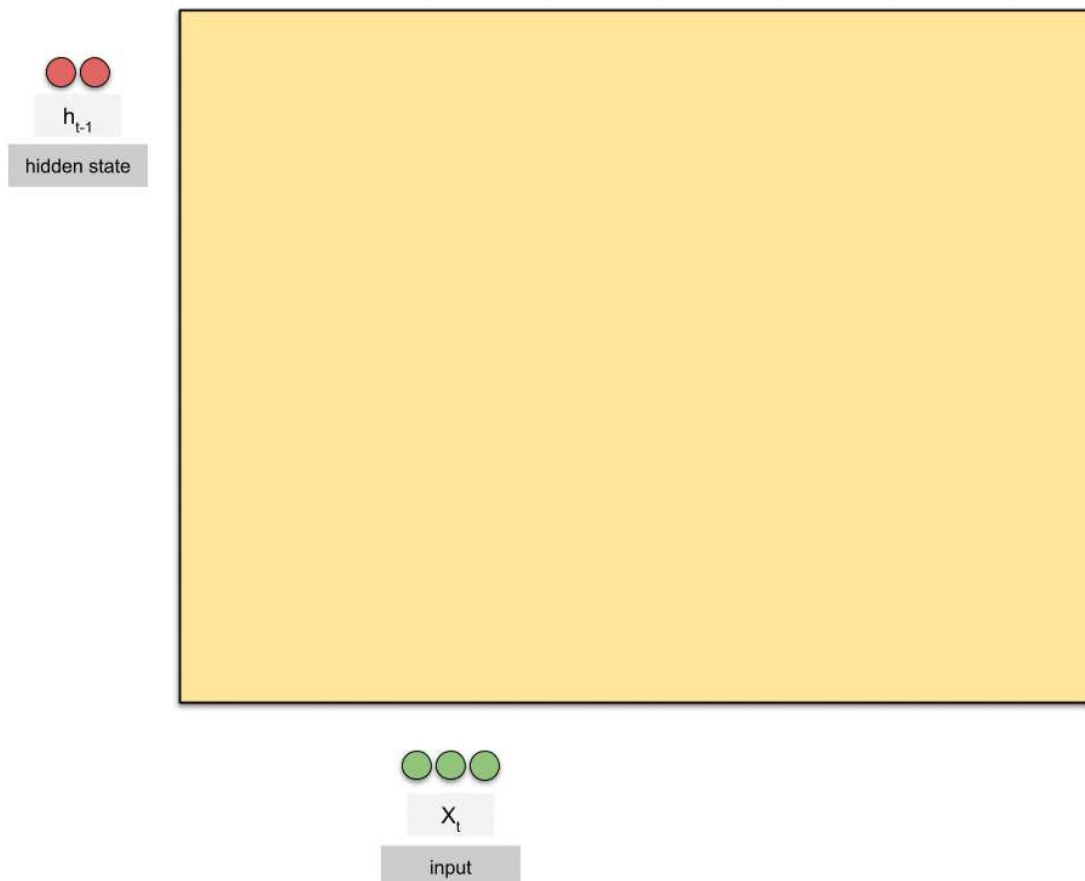
$$z_t = \sigma (W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma (W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh (W \cdot [r_t * h_{t-1}, x_t])$$

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

Entire GRU operation can be seen as .



Here , we checked some LSTM variants but there are another variants as well. All the drawbacks of RNN are already achieved in LSTM, still researchers are asking for another step which is called attention.

References

[Understanding LSTM Networks — colah's blog](#) **These loops make recurrent neural networks seem kind of mysterious. However, if you think a bit more, it turns out that...**

colah.github.io

<https://towardsdatascience.com/animated-rnn-lstm-and-gru-ef124d06cf45>

Lstm

Gru

Peephole



Written by Jaimin Mungalpara

88 Followers · Writer for Nerd For Tech

AI | ML Enthusiastic

More from Jaimin Mungalpara and Nerd For Tech

Step 1 Prepare input ($W_a s_{i-1} + U_a h_j$)

Encoder Hidden state

state at time stamp 4 (s_{i-1} as per formula)

h_1 h_2 h_3 h_4

Step 2 addition of s_{i-1} and h_j and pass it in tanh function

$\tanh \left(\begin{bmatrix} \text{blue} & \text{orange} & \text{green} & \text{red} \\ \text{blue} & \text{orange} & \text{green} & \text{red} \\ \text{blue} & \text{orange} & \text{green} & \text{red} \\ \text{blue} & \text{orange} & \text{green} & \text{red} \end{bmatrix} + \begin{bmatrix} \text{green} \\ \text{green} \\ \text{green} \\ \text{green} \end{bmatrix} \right) = \begin{bmatrix} \text{yellow} & \text{yellow} & \text{yellow} & \text{yellow} \\ \text{yellow} & \text{yellow} & \text{yellow} & \text{yellow} \\ \text{yellow} & \text{yellow} & \text{yellow} & \text{yellow} \\ \text{yellow} & \text{yellow} & \text{yellow} & \text{yellow} \end{bmatrix}$

Step 3 Multiply step2 with alignment transpose weight V_a

V_a \times $\begin{bmatrix} \text{yellow} & \text{yellow} & \text{yellow} & \text{yellow} \\ \text{yellow} & \text{yellow} & \text{yellow} & \text{yellow} \\ \text{yellow} & \text{yellow} & \text{yellow} & \text{yellow} \\ \text{yellow} & \text{yellow} & \text{yellow} & \text{yellow} \end{bmatrix} = \begin{bmatrix} 10 & 8 & 8 & 5 \end{bmatrix}$ Alignment Score

$\alpha(s_{i-1}, h_j) = v_a^T \tanh(W_a s_{i-1} + U_a h_j)$

Step 4 Softmax the alignment score

$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$

$\text{softmax} \left(\begin{bmatrix} 10 & 8 & 8 & 5 \end{bmatrix} \right) = \begin{bmatrix} 0.78 & 0.11 & 0.11 & 0.005 \end{bmatrix}$

α_{11} α_{12} α_{13} α_{14}

Step 5 Multiply each vector with softmax score and sum according to formula

$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$

$\alpha_{11} \times \begin{bmatrix} \text{blue} \\ \text{blue} \\ \text{blue} \\ \text{blue} \end{bmatrix} + \alpha_{12} \times \begin{bmatrix} \text{orange} \\ \text{orange} \\ \text{orange} \\ \text{orange} \end{bmatrix} + \alpha_{13} \times \begin{bmatrix} \text{green} \\ \text{green} \\ \text{green} \\ \text{green} \end{bmatrix} + \alpha_{14} \times \begin{bmatrix} \text{red} \\ \text{red} \\ \text{red} \\ \text{red} \end{bmatrix} = \begin{bmatrix} \text{purple} \\ \text{purple} \\ \text{purple} \\ \text{purple} \end{bmatrix}$ Context vector C_i for decoder

J Jaimin Mungalpara in Nerd For Tech

What is Attention in NLP?

In this blog we will look on the pivotal research in the area of NLP which has changed the view of NLP for entire world that is called...

6 min read · Feb 26, 2021





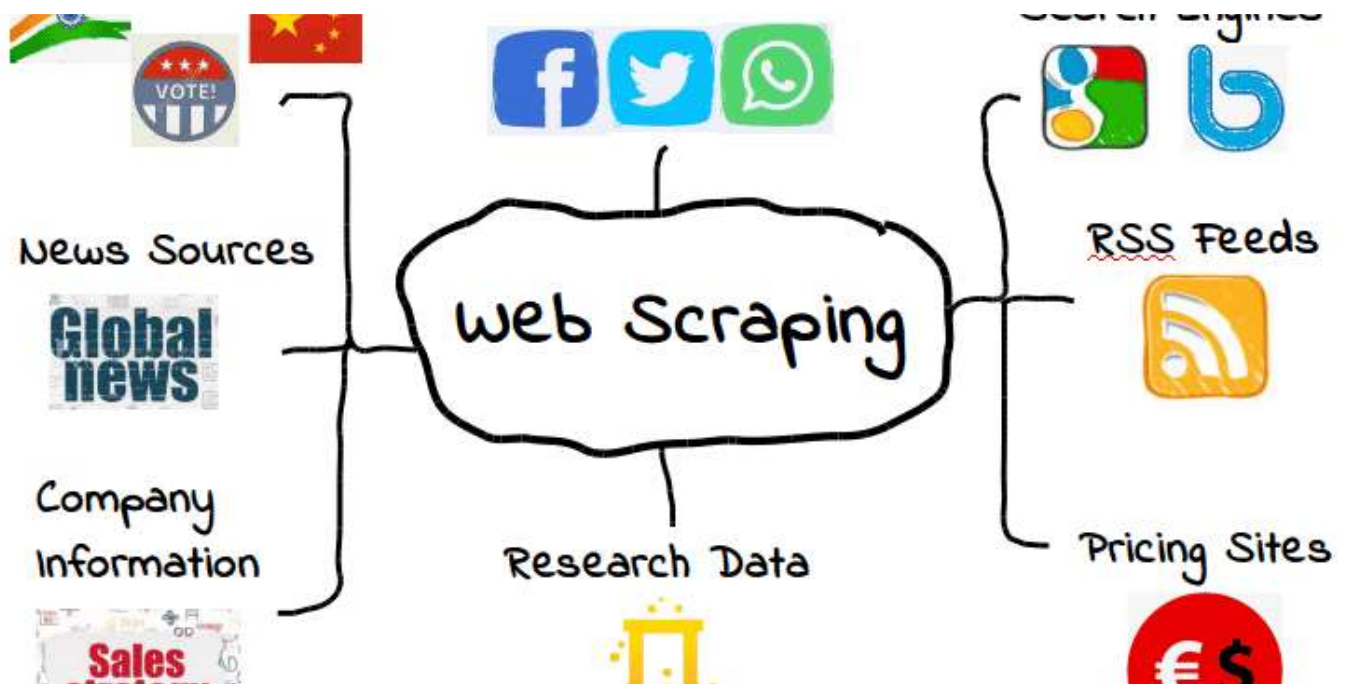
Gunjan Sahu in Nerd For Tech

Flipkart Data Engineer Interview

Flipkart Data Engineer Interview

6 min read · Jul 12

187 1



Yenni95zz in Nerd For Tech

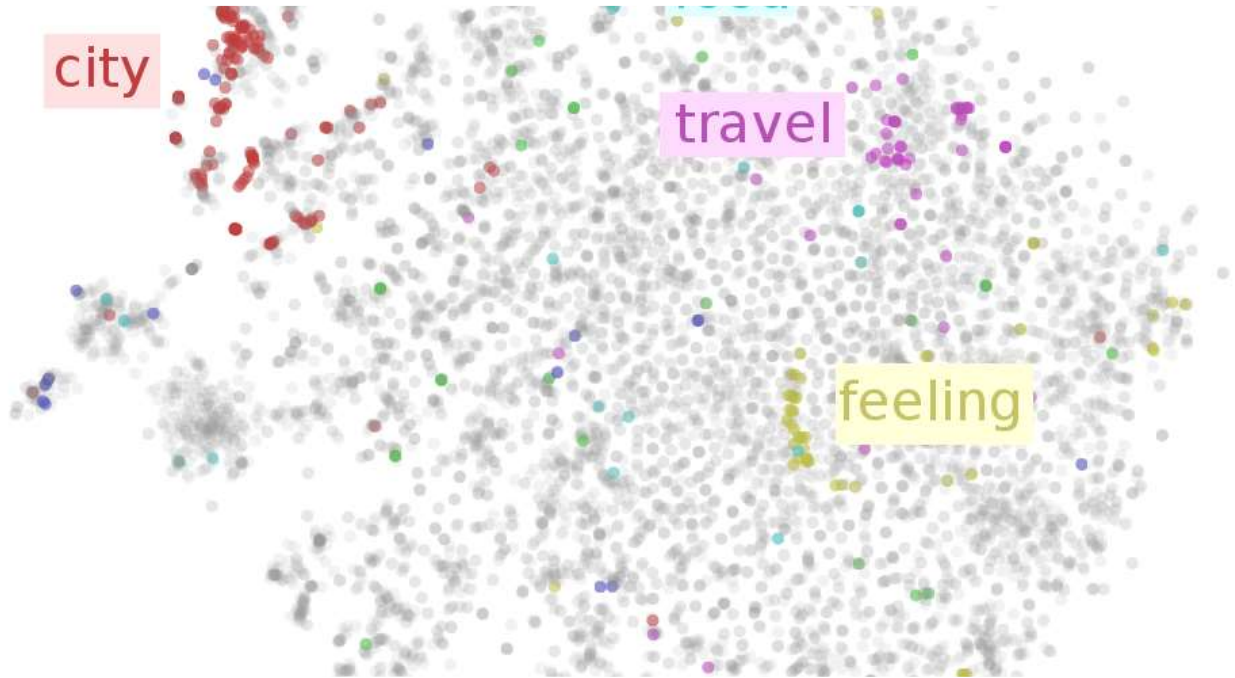
\$4000 Freelance Income: Exploring More Than Just Web Scraping

I've just been using Upwork for 3 months, but in that time I've completed various projects in addition to web scraping, earning a total of...

★ · 7 min read · Jul 17

👏 374 💬 2

🔖+ ⋮



J Jaimin Mungalpara in Nerd For Tech

Word Embedding : Text Analysis : NLP : Part-3 : GloVe and FastText

5 min read · May 9, 2021

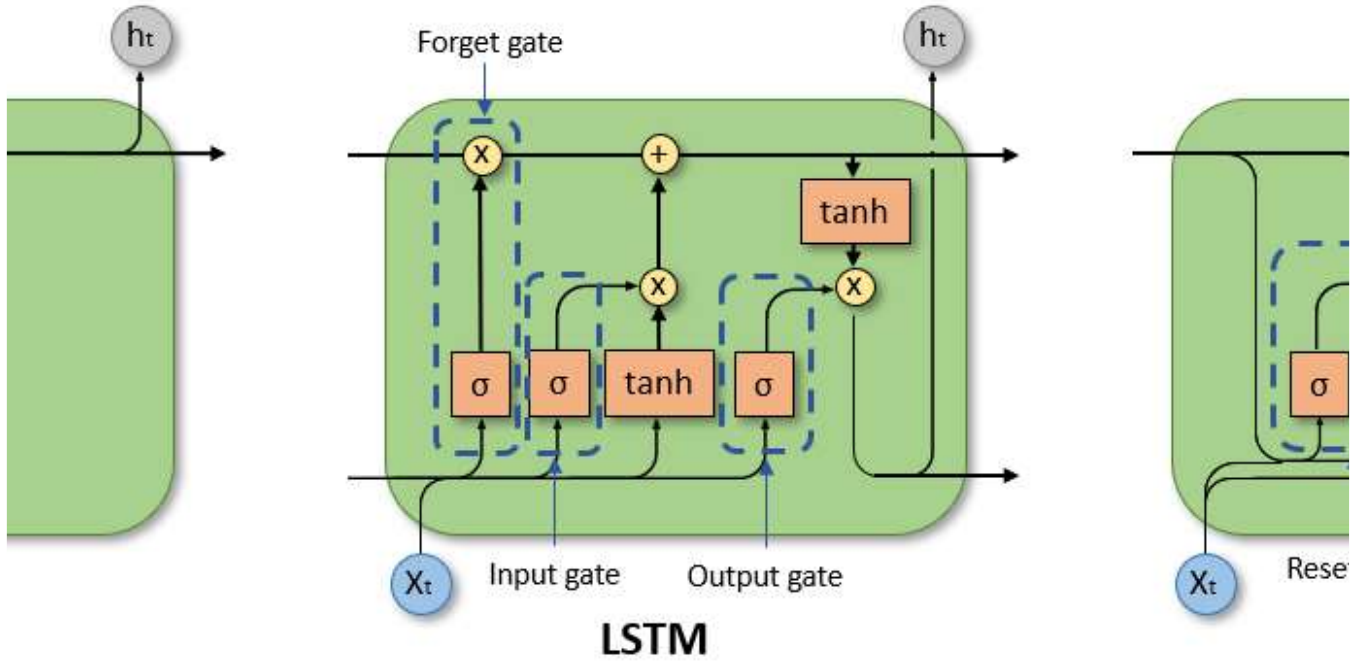
👏 33 💬

🔖+ ⋮

See all from Jaimin Mungalpara

See all from Nerd For Tech

Recommended from Medium



 Jonte Dancker in Towards Data Science

A Brief Introduction to Recurrent Neural Networks

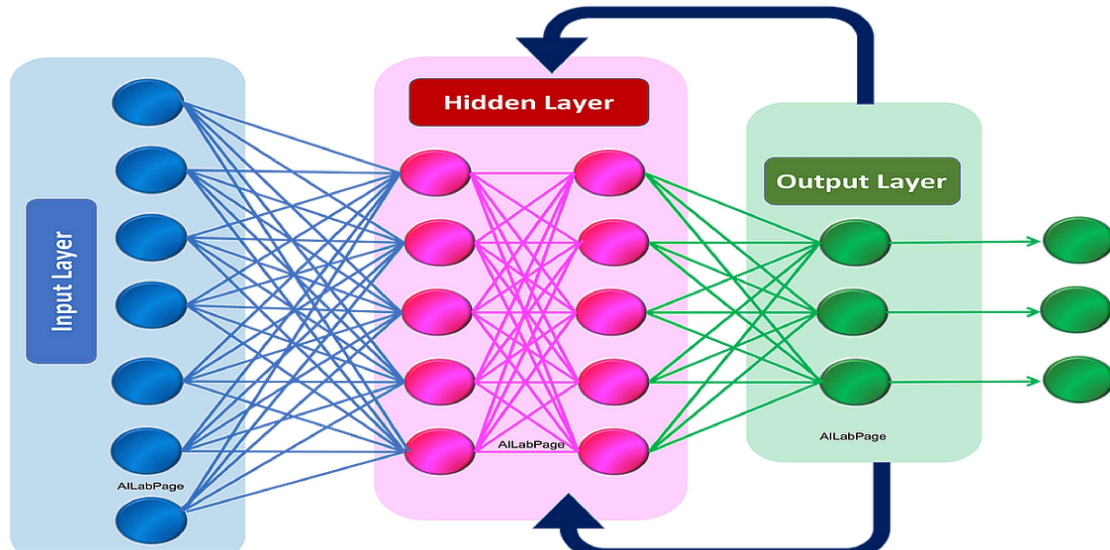
An introduction to RNN, LSTM, and GRU and their implementation

12 min read · Dec 26, 2022

 359  5

Recurrent Neural Networks



 Farheenshaukat

Recurrent Neural Network (RNN)

A recurrent neural network (RNN) is a type of neural network that is commonly used for processing sequential data such as speech, natural...

8 min read · May 11

 62 

Lists



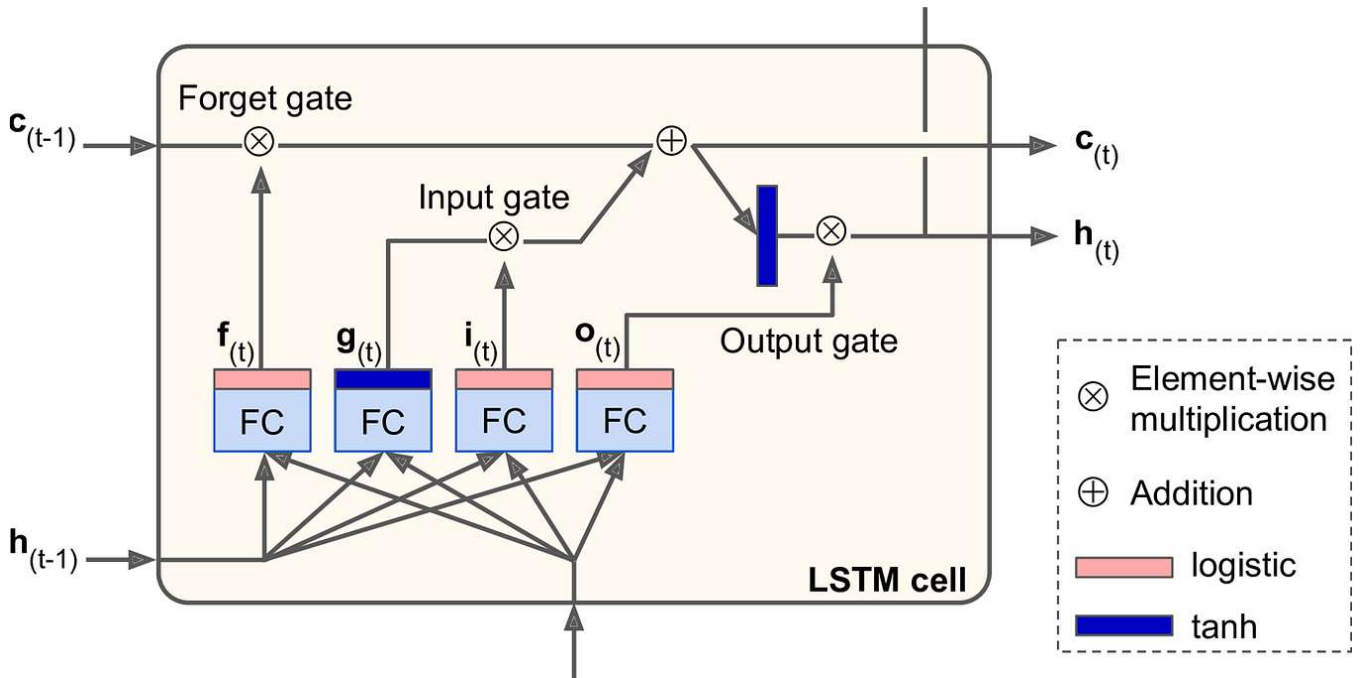
Natural Language Processing

669 stories · 283 saves



Staff Picks

465 stories · 320 saves



Anishnama

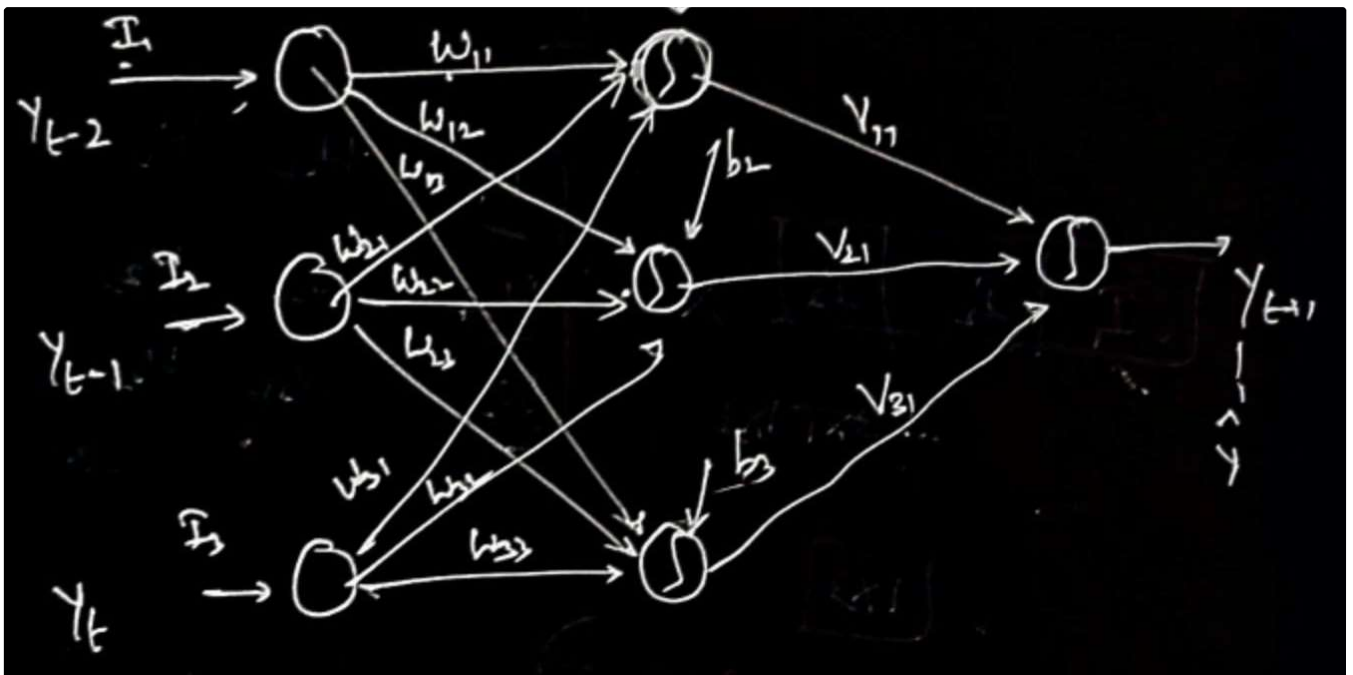
Understanding LSTM: Architecture, Pros and Cons, and Implementation

What is LSTM and How it works?

7 min read · Apr 28

153 2

🔖 ...



Utsav Poudel in Level Up Coding

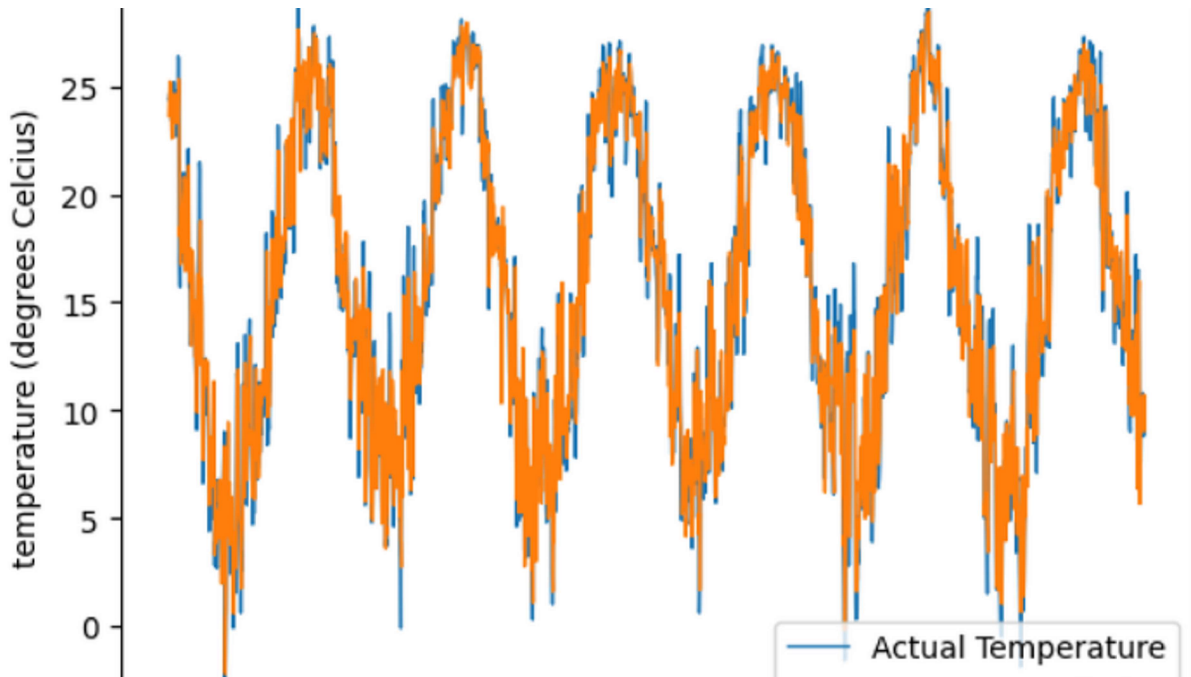
Time and Series Forecasting with LSTM- Recurrent Neural Networks


Every day, humans make passive predictions when performing tasks such as crossing a road, where they estimate the speed of cars and their...

10 min read · May 10

 306  2



 Can Ozdogar

Time-series Forecasting using LSTM (PyTorch implementation)

Exploring implementation of long short-term memory network using PyTorch and weather dataset

8 min read · Sep 9

 17 



 Manohar Varma in AI Skunks

Time Series Forecasting using Deep Learning

A detailed intuition on time series data using neural model RNN(LSTM)

10 min read · Apr 11

 11 

See more recommendations