RNN And LSTM

Adopted from
http://colah.github.io/posts/2015-08-Understanding-LSTMs/
https://medium.com/@shiyan/understanding-lstm-and-its-diagrams-37e2f46f1714

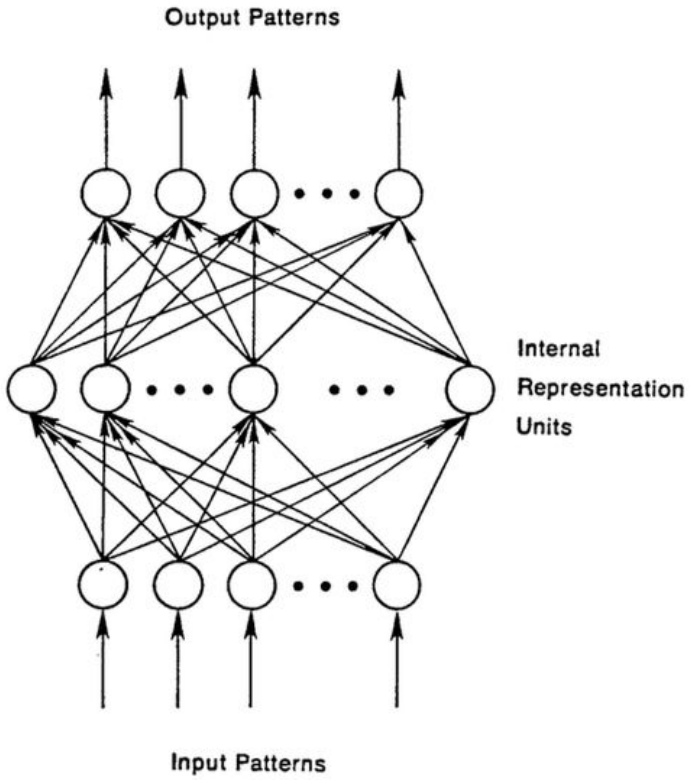
Agenda

1. Review of Feedforward Networks
2. RNN
3. Exploding And Disappearing Gradients
4. LSTM

Review of Feedforward Nets

FNN

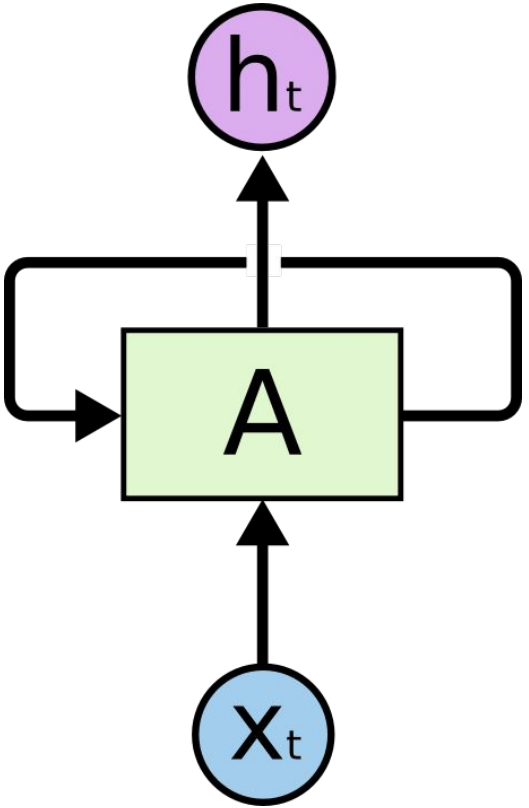
- feeds information straight through layers (never touching a given node twice)
- has **no notion of order in time**, and the only input it considers is the current example it has been exposed to



Recurrent Neural Net (RNN)

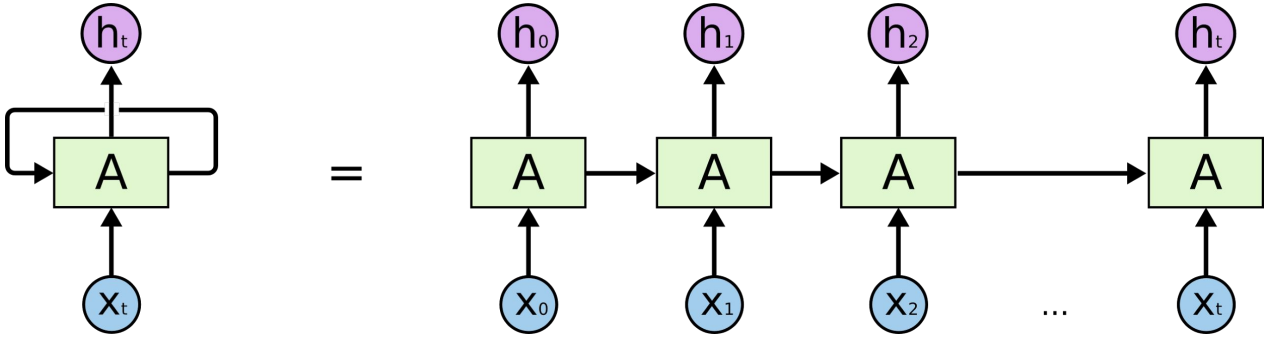
RNN

- We don’t start our thinking from scratch every second. E.g : understand each word based on understanding of previous words
- RNN are **networks with loops** in them, allowing information to persist.



RNN

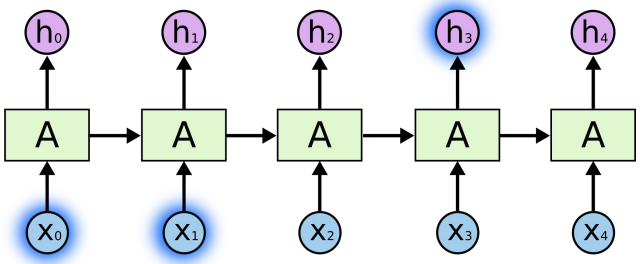
- can be thought of as multiple copies of the same network, each passing a message to a successor
- intimately related to sequences and lists. They’re the natural architecture of neural network to use for such data.
- another look at



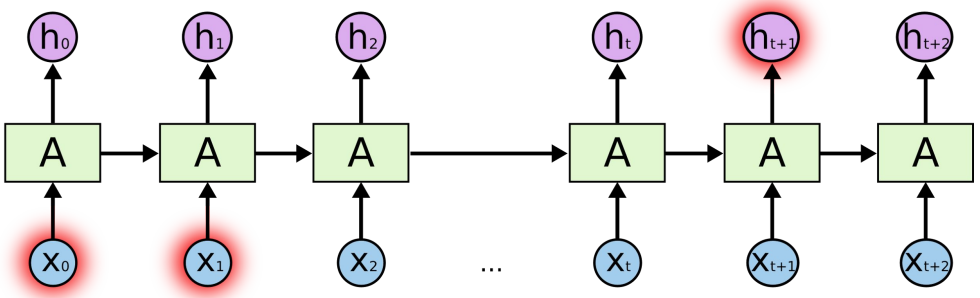
Exploding and Disappearing Gradients

RNN

- “the clouds are in the sky”

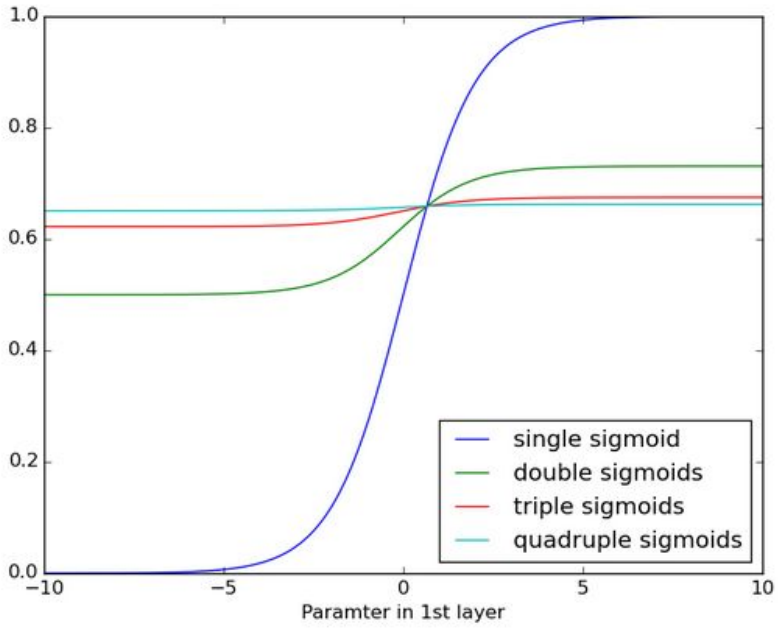


- “I grew up in France.....
  I speak fluent French.”
- **In theory**, RNNs are absolutely capable of handling such “long-term dependencies.”



Exploding and Disappearing Gradients

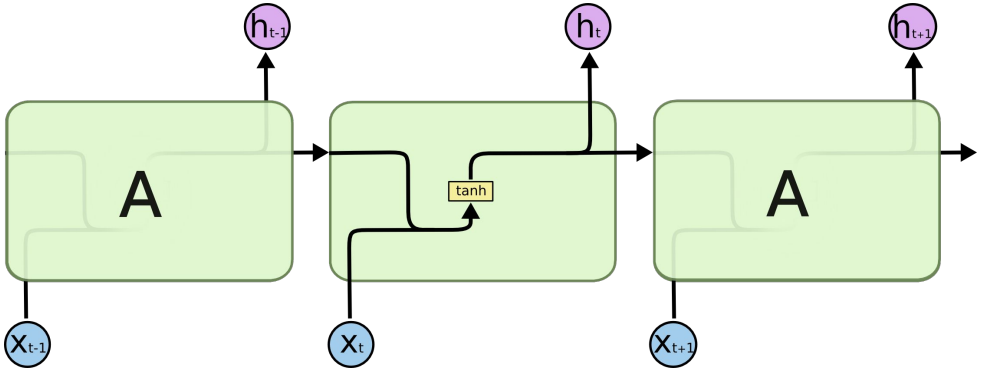
- information flowing through neural nets passes through many stages of multiplication.
- exploding gradients can be solved relatively easily -> truncated or squashed.
- vanishing gradients can become too small for computers to work with or for networks to learn – a **harder problem to solve.**



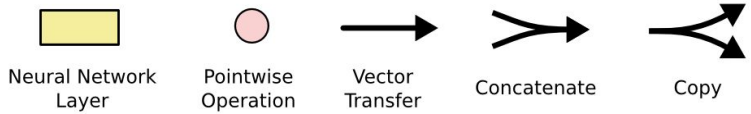
Long Short Term Memory Net (LSTM)

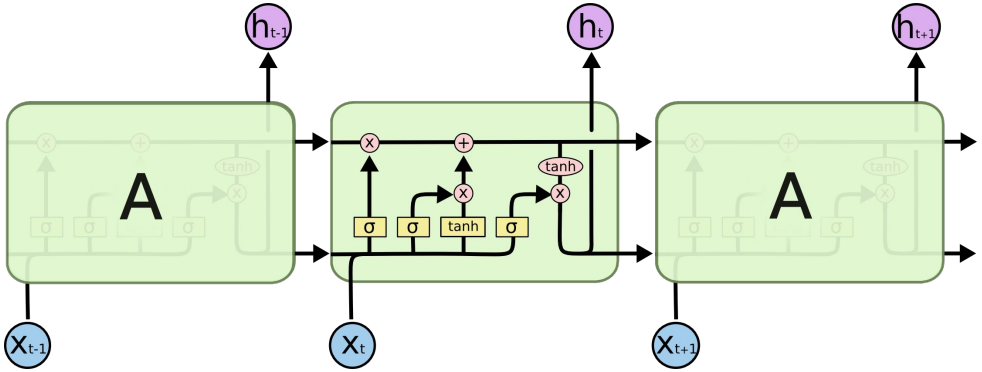
LSTM

In standard RNN



In LSTM





LSTM

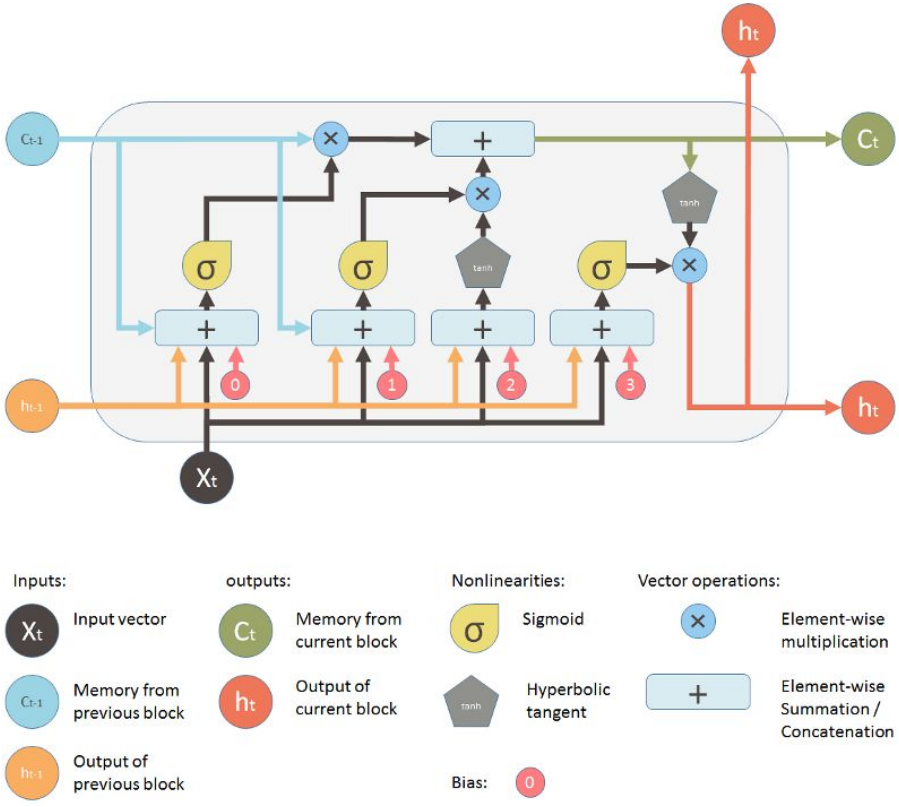
A better diagram

Input: three inputs.

- X_t is the input of the current time step.
- h_t-1 is the output from the previous LSTM unit
- C_t-1 is the “memory” of the previous unit,

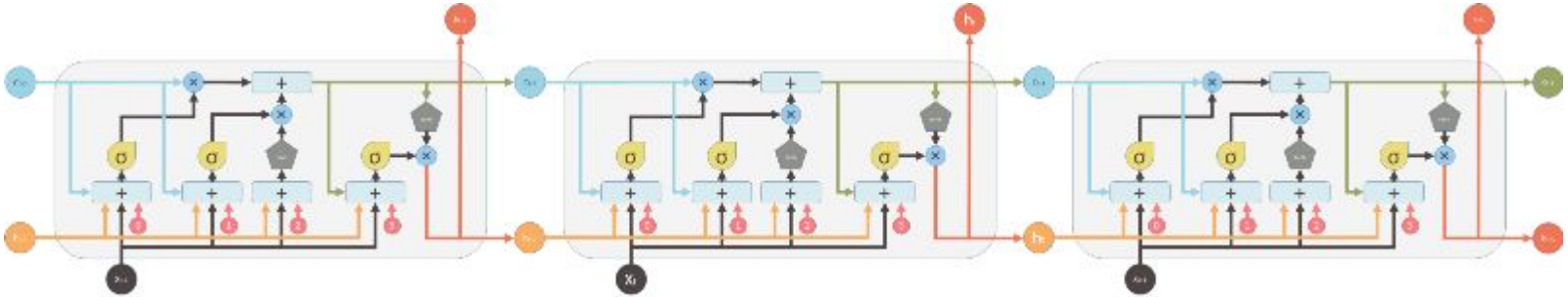
Outputs:

- h_t is the output of the current network.
- C_t is the memory of the current unit.



LSTM

A single unit makes decision by considering the current input, previous output and previous memory.
And it generates a new output and alters its memory.

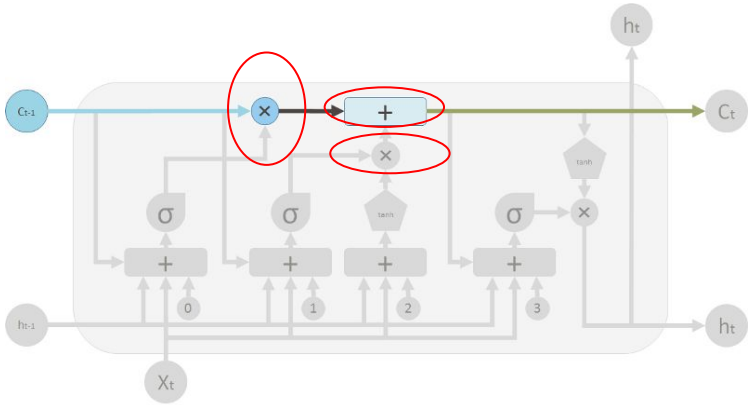


LSTM

The way its internal memory C_t changes is pretty similar to piping water through a pipe.

Forget pipe: If you shut it, no old memory will be kept. If you fully open this valve, all old memory will pass through.

Memory valve: New memory will come in through a T shaped joint like above and merge with the old memory.



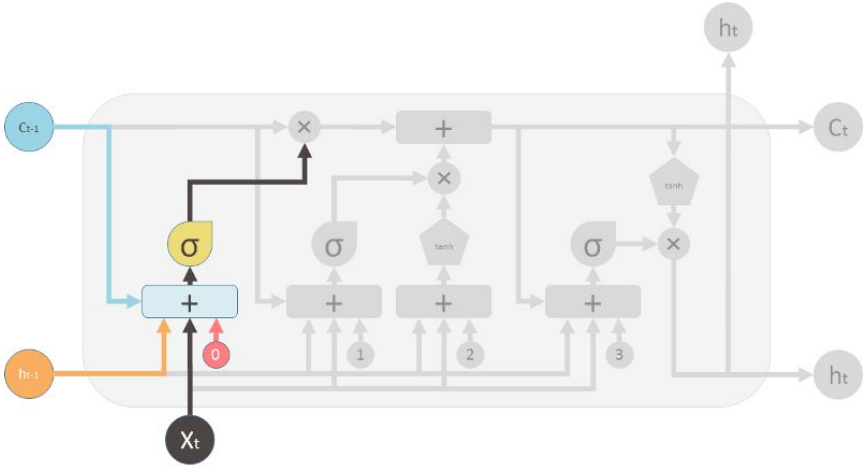
Forget Valve

The inputs:

- h_t-1: the output of the previous LSTM block
- X_t: the input for the current LSTM block
- C_t-1: the memory of the previous block and finally a bias vector b_0.

Sigmoid function as activation

Output vector is the forget valve, which will applied to the old memory C_t-1 by element-wise multiplication.



Memory Valve

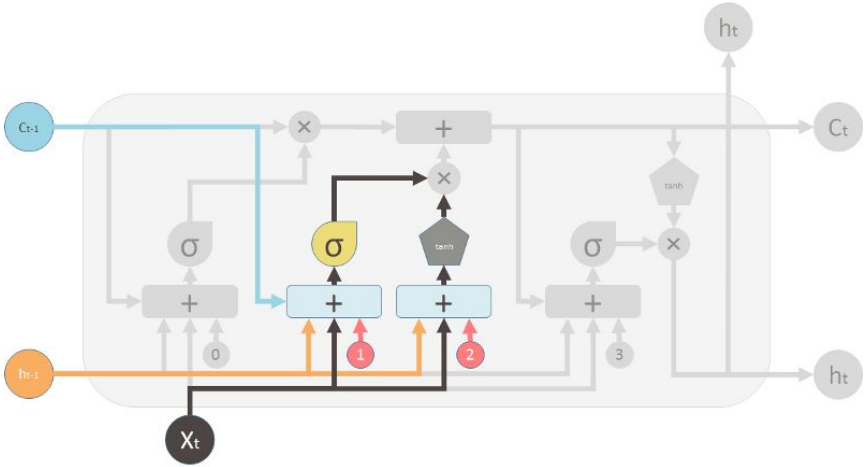
Controls how much the new memory should influence the old memory.

The inputs:

- h_t-1: the output of the previous LSTM block
- X_t: the input for the current LSTM block
- C_t-1: the memory of the previous block and finally a bias vector b_0.

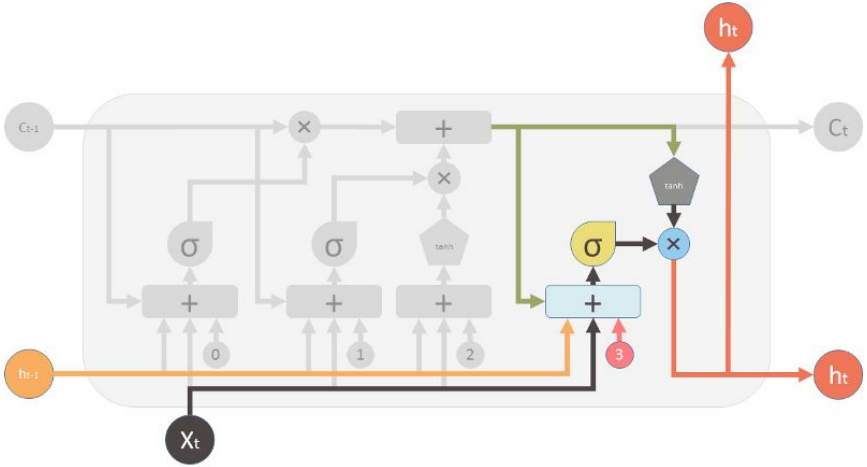
Tanh as activation for new memory

Output vector is element-wise multiple the new memory valve, and add to the old memory to form the new memory.



Memory Valve

Output valve that is controlled by the new memory, the previous output h_t-1, the input X_t and a bias vector. This valve controls how much new memory should output to the next LSTM unit.



Q??