Introduction to Deep Learning

Course Instructor: Dr. M. Umair
Topic: Long Short-Term Memory Networks

Agenda

1. Long Short-Term Memory Networks
2. Forget Gate
3. Input Gate
4. Cell State
5. Output Gate
6. Code Example
7. References

Long Short-Term Memory Networks

Long Short-Term Memory Networks

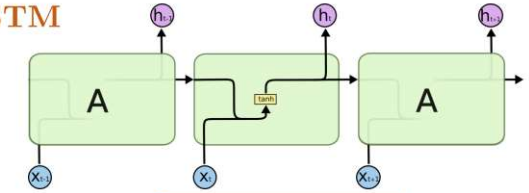
- **Introduction**
  - **Long Short-Term Memory Networks** (LSTMs) are a *special kind of RNN*, capable of learning long-term dependencies.
  - Introduced by Hochreiter & Schmidhuber (1997).
  - LSTMs are explicitly designed to avoid the *long-term dependency problem.*

The *long-term dependency problem* in (RNNs) refers to the *difficulty of capturing and learning relationships or dependencies between distant time steps in sequential data.*

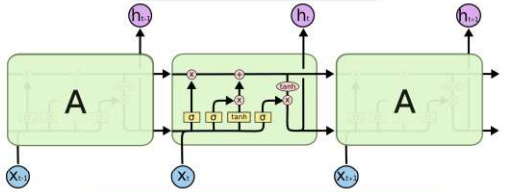
Traditional *RNNs struggle* with retaining information over long sequences due to issues like *vanishing or exploding gradients* during training.

Long Short-Term Memory Networks

- **RNN vs LSTM**



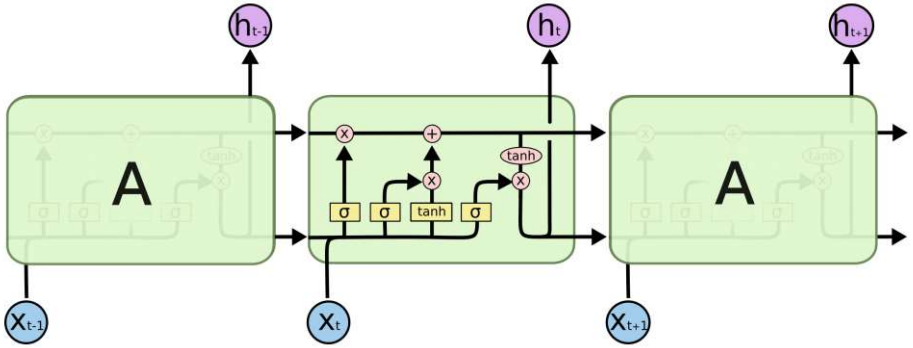
Recurrent Neural Network



Long Short-Term Memory Network

Long Short-Term Memory Networks

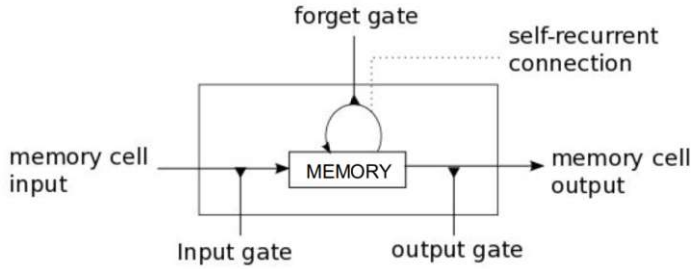
- **LSTM**



Long Short-Term Memory Network

Long Short-Term Memory Networks

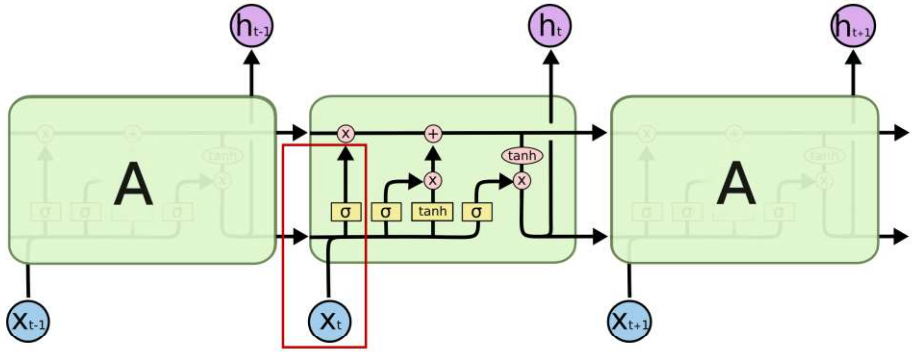
- **Introduction**
  - Central Idea: A memory cell which can *maintain its state over time*, consisting of an *explicit memory (cell state)* and gating units which *regulate the information* flow into and out of the memory.



Forget gate

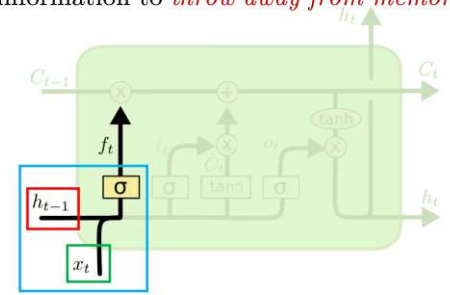
Forget Gate

- Illustration



Forget Gate

- What does it do?
  - Controls what information to *throw away from memory*.



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

Long Short-Term Memory Networks

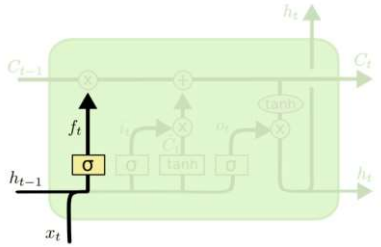
- Forget Gate Calculations

$h_{(t-1)}$	0.100	1.000	0.500			
$x_{(t)}$	1.256	0.200	0.800			
W_f	0.000	0.000	0.000	-0.025	0.000	0.000
	-0.030	0.900	-0.800	-0.500	0.000	0.000
	0.070	0.600	0.200	1.000	1.000	1.000
b_f	0.010	0.050	0.010			
$h_{(t-1)} + x_{(t)}$	0.100	1.000	0.500	1.256	0.200	0.800

Transpose $(h_{(t-1)} + x_{(t)})$
0.100
1.000
0.500
1.256
0.200
0.800

	Bias Added	Sigmoid
Cell 1	-0.031	-0.021	0.495
Cell 2	-0.131	-0.081	0.480
Cell 3	2.963	2.973	0.951
			f_t

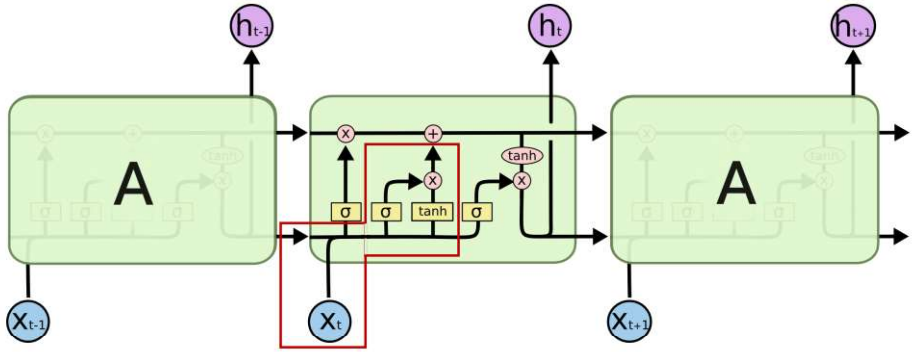
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$



Input Gate

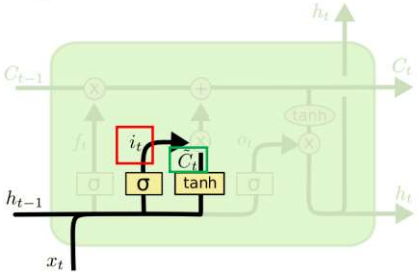
Input Gate

- **Illustration**



Input Gate

- **What does it do?**
  - Controls *what new information is added* to cell state from current input.

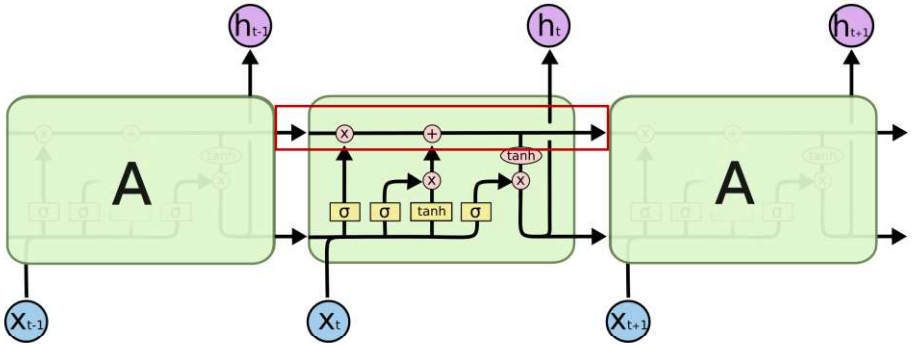


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \qquad \tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Cell State

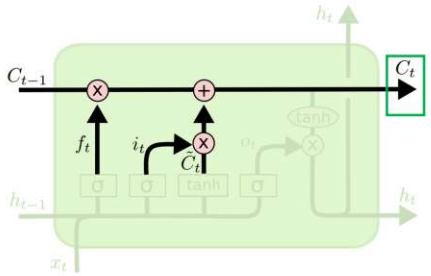
Cell State

- **Illustration**



Cell State

- **What does it do?**
  - Controls *what new information is added* to cell state *from current input.*



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

Long Short-Term Memory Networks

- **Input Gate Calculation**

f_t
0.495
0.480
0.951

$C_{[t-1]}$
0.033
0.589
0.924

$f_t * C_{[t-1]}$
0.016
0.283
0.879

	Bias Added	Sigmoid
Cell 1	1.041	0.768
Cell 2	-1.088	0.383
Cell 3	0.000	0.502
		i_t

	Bias Added	Tanh
Cell 1	0.376	0.520
Cell 2	0.471	0.447
Cell 3	-0.133	-0.103
		$\tilde{C}_t$

$i_t * \tilde{C}_t$	C_t
0.399	0.415
0.127	0.409
-0.052	0.827

$h_{[t-1]}$		
0.100	1.000	0.500

$x_{[t]}$		
1.256	0.200	0.800

W						
0.659	0.111	0.287	0.345	0.000	0.359
-0.236	-1.000	0.580	-0.123	-1.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000

b_i		
0.154	0.160	0.010

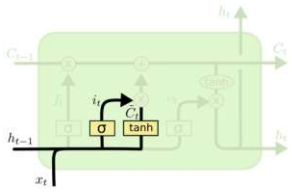
W_i						
0.356	0.124	0.222	0.025	0.365	0.001
0.485	0.548	0.324	0.001	-0.145	-0.325
0.125	0.257	-0.365	-0.256	-0.122	0.157

b_i		
0.200	0.010	0.030

$h_{[t-1]} + x_{[t]}$					
0.100	1.000	0.500	1.256	0.200	0.800

$Transpose\ (h_{[t-1]} + x_{[t]})$
0.100
1.000
0.500
1.256
0.200
0.800

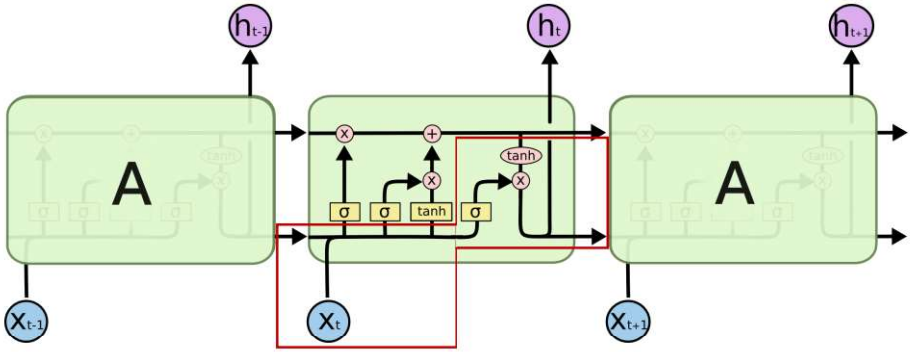
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \quad \tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



Output Gate

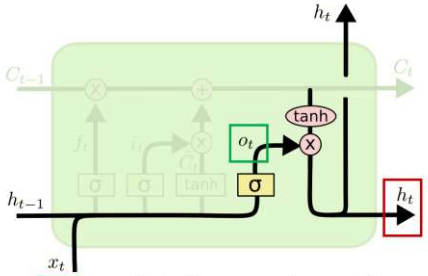
Output Gate

- **Illustration**



Long Short-Term Memory Networks

- **What does it do?**
  - Conditionally decides what to output from the memory.



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

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

Long Short-Term Memory Networks

- **LSTM calculation - Hands-on (Output Gate)**

C_t
0.415
0.409
0.827

	Bias Added	Sigmoid
Cell 1	0.241	0.560
Cell 2	0.408	0.601
Cell 3	0.533	0.630

o_t

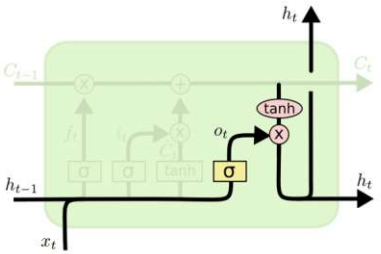
$tanh\,(C_t)$
0.393
0.388
0.679

h_t
0.220
0.233
0.428

$h_{(t-1)}$	0.100	1.000	0.500			
$x_{(t)}$	1.256	0.200	0.800			
W_o	0.033	0.213	0.000	0.000	0.006	0.023
	0.000	0.068	0.048	0.215	0.002	0.056
	0.012	0.000	0.165	0.333	0.000	0.000
b_o	0.005	0.001	0.031			
$h_{(t-1)} + x_{(t)}$	0.100	1.000	0.500	1.256	0.200	0.800
$Transpose\,(h_{(t-1)} + x_{(t)})$	0.100					
	1.000					
	0.500					
	1.256					
	0.200					
	0.800					

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

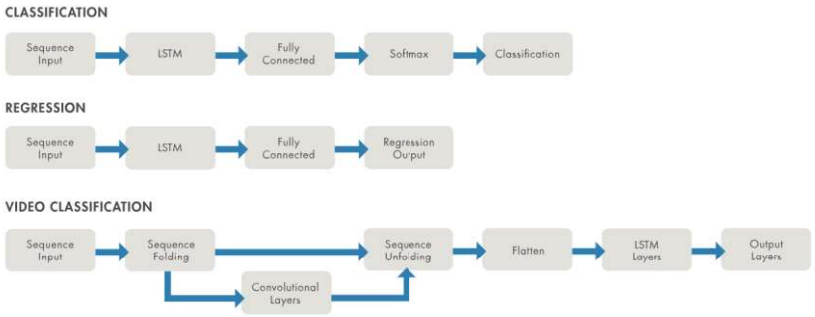
$$h_t = o_t * \tanh\left(C_t\right)$$



LSTM Network Architecture

LSTM Network Architecture

- **How to Design an LSTM Network for Different Tasks?**
  - LSTMs work well with sequence and time-series data for classification and regression tasks.



Code Example

Code Example

- **LSTM**
  - https://pieriantraining.com/tensorflow-lstm-example-a-beginners-guide/

References

References

1. https://www.mathworks.com/discovery/lstm.html
2. https://colah.github.io/posts/2015-08-Understanding-LSTMs/