Artificial intelligence in data science

Long Short-Term Memory Networks

Janos Török

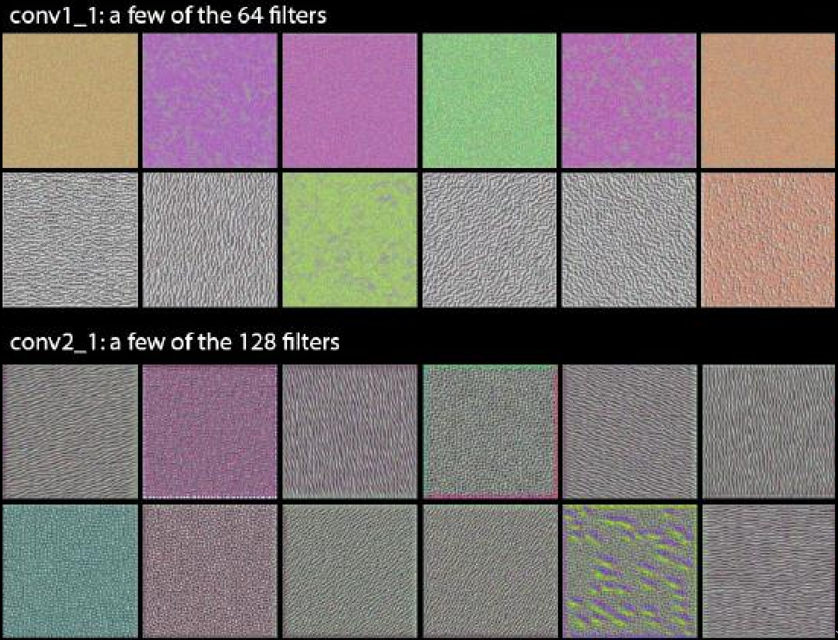
Department of Theoretical Physics

October 20, 2022

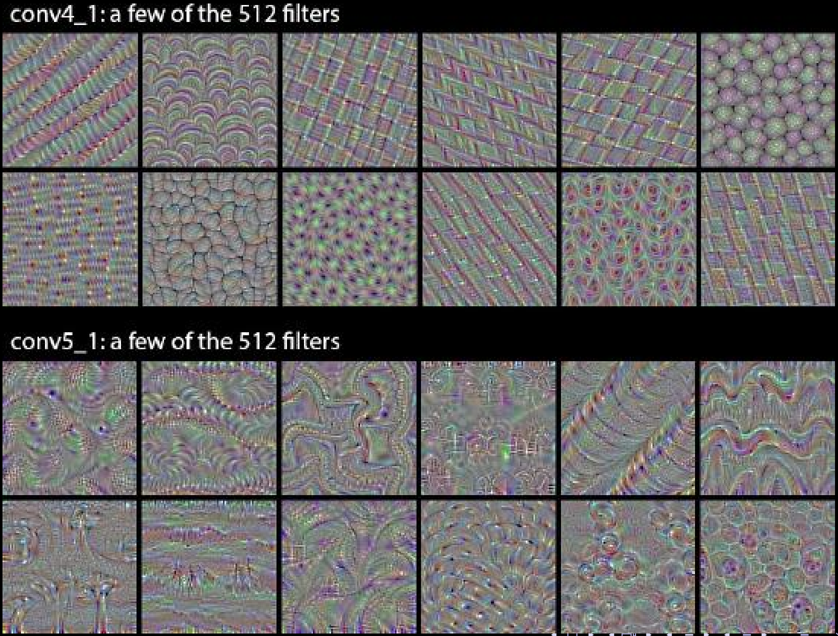
Visualization of different layers

- Algorithm:
- Use the optimization algorithm of tensorflow and maximize the outcome of one class
- We end up with images which will be 100% in one category
- How they look like?

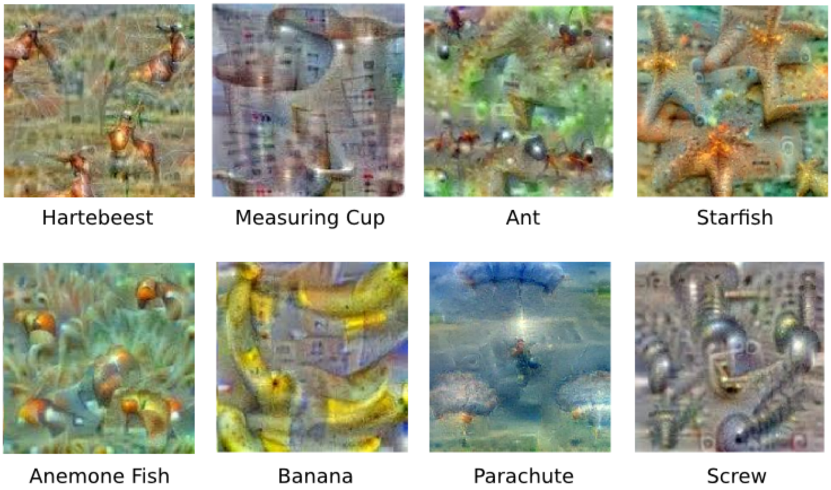
Visualization of different layers



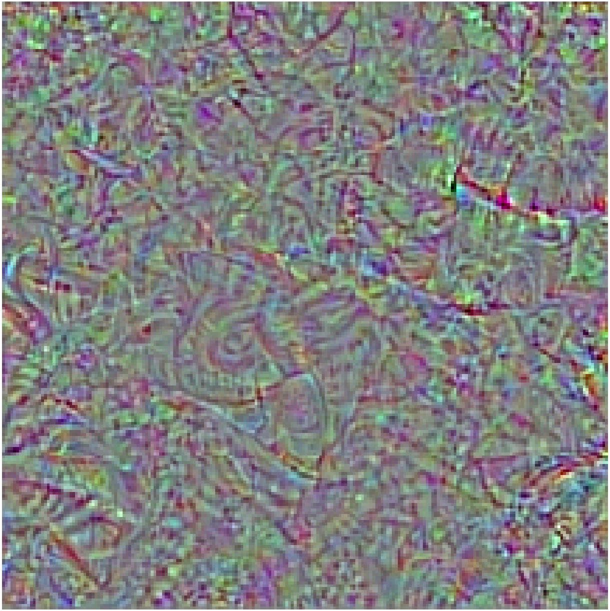
Visualization of different layers



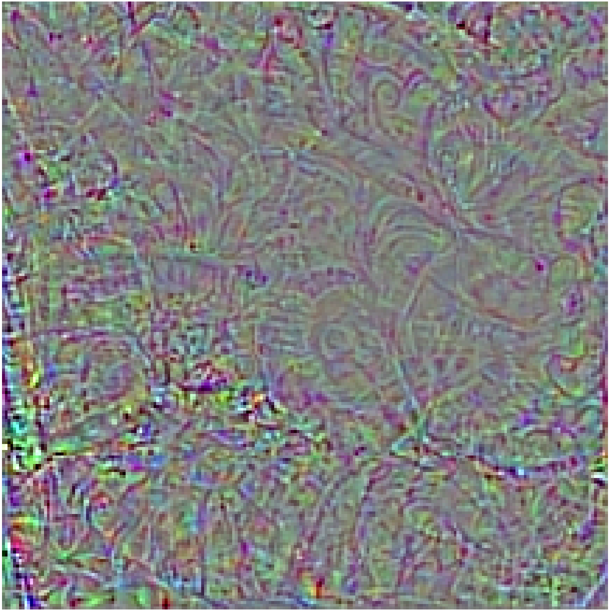
Visualization of different layers



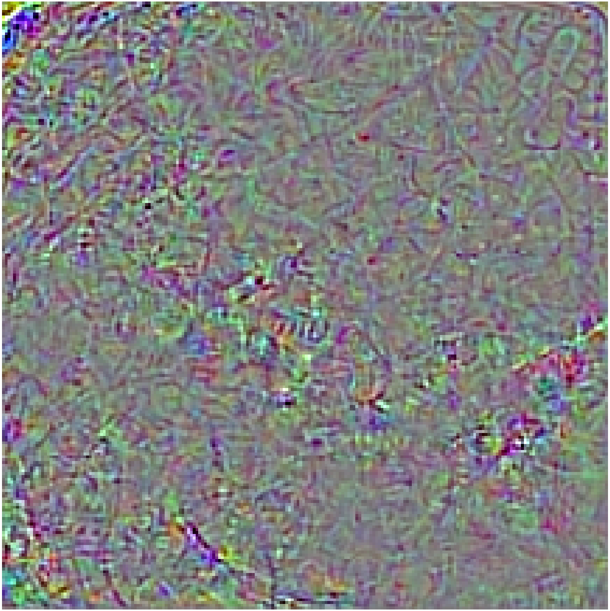
Sea snake



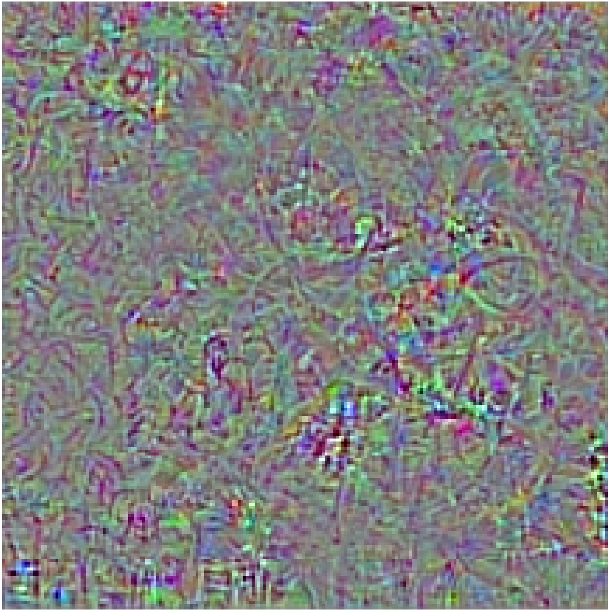
Rocking Chair



Rugby ball

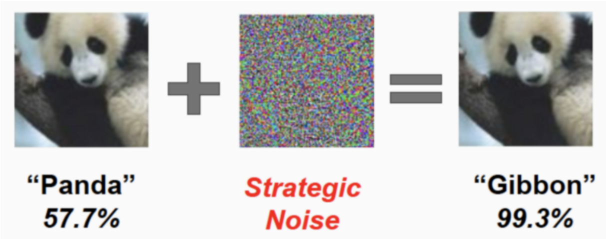


Lake side



Cheat neural networks

- If neural network is exactly known
- Images can be made which are cathegorized falsely
- You can hide your activity

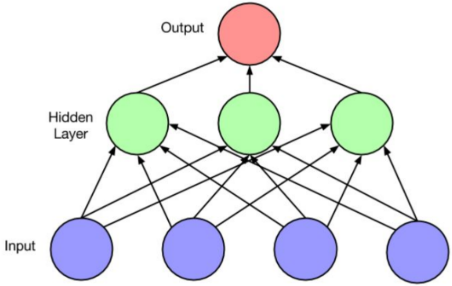


Temporal data

- Most of the data is sequential, can be ordered
- Very often time orders the data
- Prediction is very important
- For this we need history

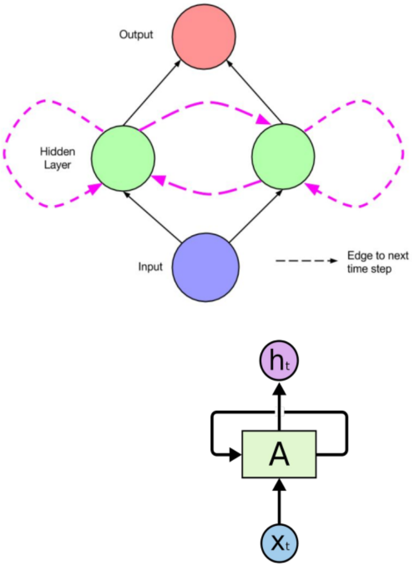
Source: Akshay Sood

Feedforward neural network



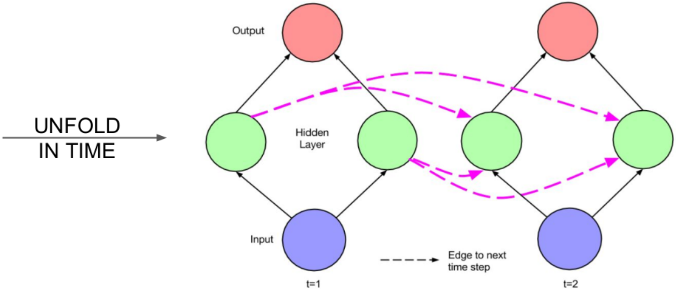
Recurrent Neural Networks (RNN)

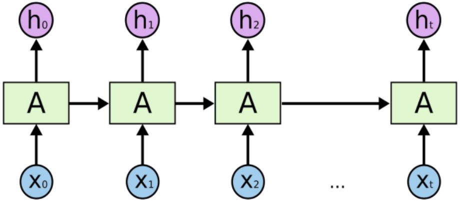
- Output depends on previous state and current output



Recurrent Neural Networks (RNN)

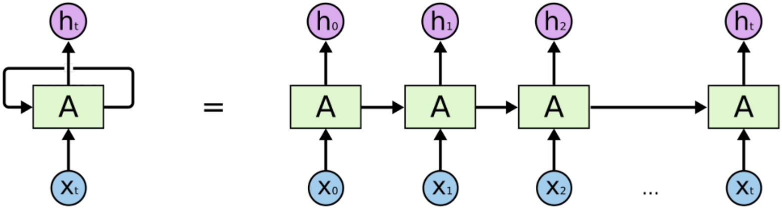
- Output depends on previous state and current output
- Feedback loops





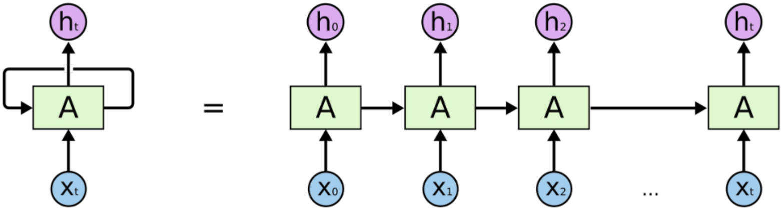
Training RNNs

- Backpropagation through time
- Regular (feedforward) backprop applied to RNN unfolded in time



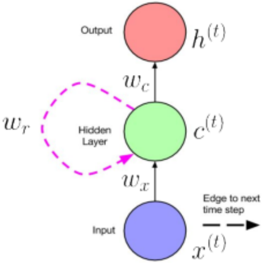
Training RNNs

- Backpropagation through time
- Regular (feedforward) backprop applied to RNN unfolded in time
- Problem: can't capture long-term dependencies due to vanishing/exploding gradients during backpropagation



Training RNNs

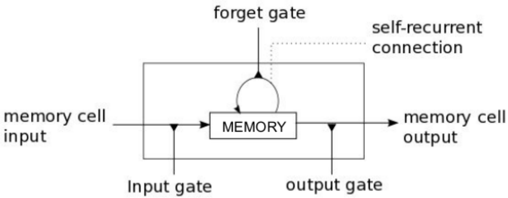
- Problem: can't capture long-term dependencies due to vanishing/exploding gradients during backpropagation



$$h^{(t)} = \sigma(w_c \cdot c^{(t)})$$
$$c^{(t)} = \sigma(w_r \cdot c^{(t-1)} + w_x \cdot x^{(t)})$$

Long Short-Term Memory networks (LSTM)

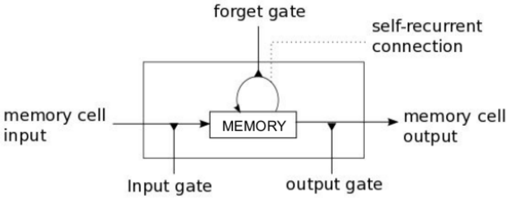
- A type of RNN architecture that addresses the vanishing/exploding gradient problem and allows learning of long-term dependencies
- Recently risen to prominence with state-of-the-art performance in speech recognition, language modeling, translation, image captioning



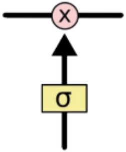
LSTM Memory Cell

Long Short-Term Memory networks (LSTM)

- Memory cell (block): maintains its state over time
- Gating units: regulate the information flow into and out of the memory

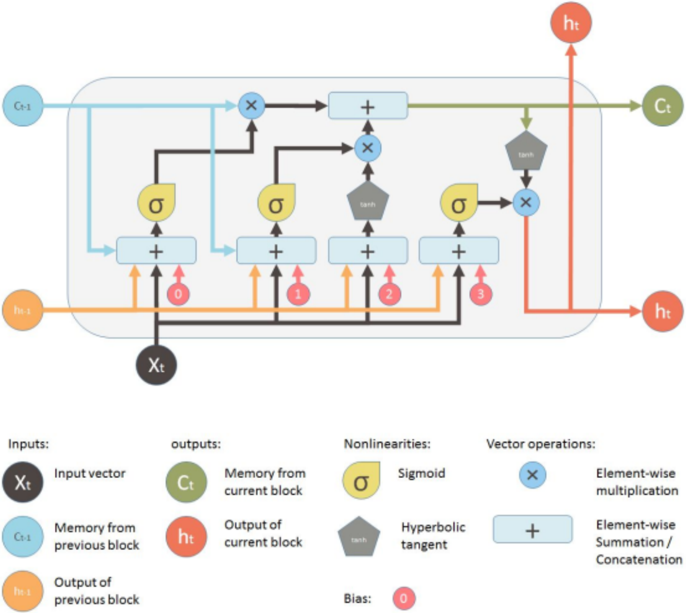


LSTM Memory Cell



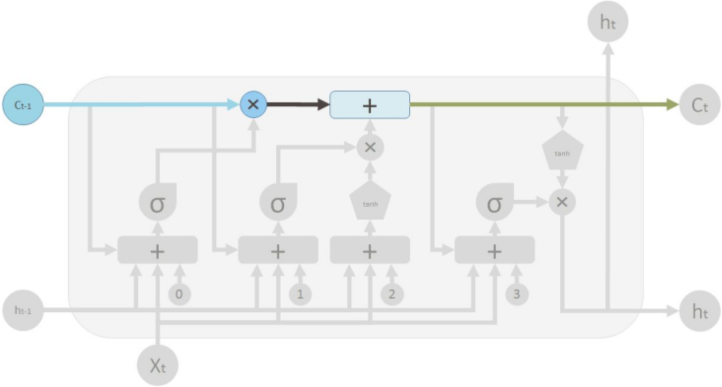
Gate (sigmoid layer followed by pointwise multiplication)

LSTM Memory Cell



LSTM Cell state vector (C)

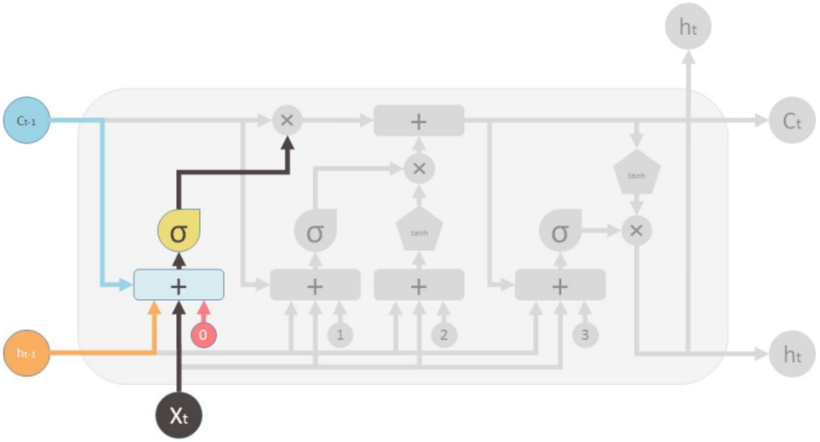
- Memory of the LSTM
- State can be changed by forgetting ($\times$) and addition of new data ($+$)
- Linear changes



LSTM Forget Gate

- Controls what remains of the previous memory

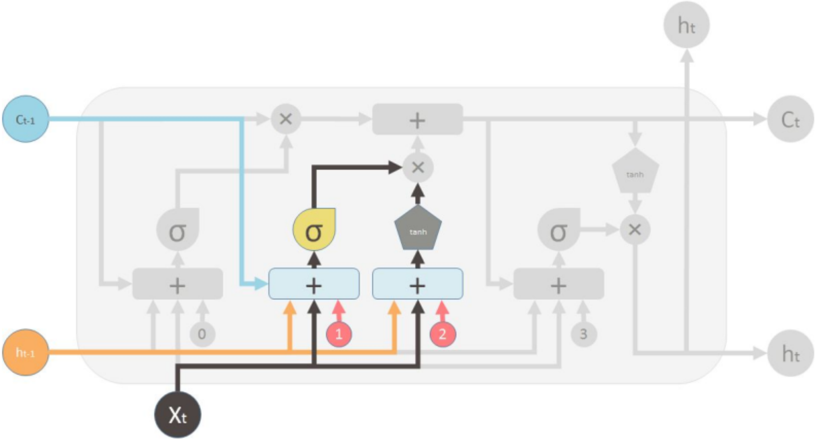
$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$



LSTM Input Gate

- Controls what what new information is added to the memory

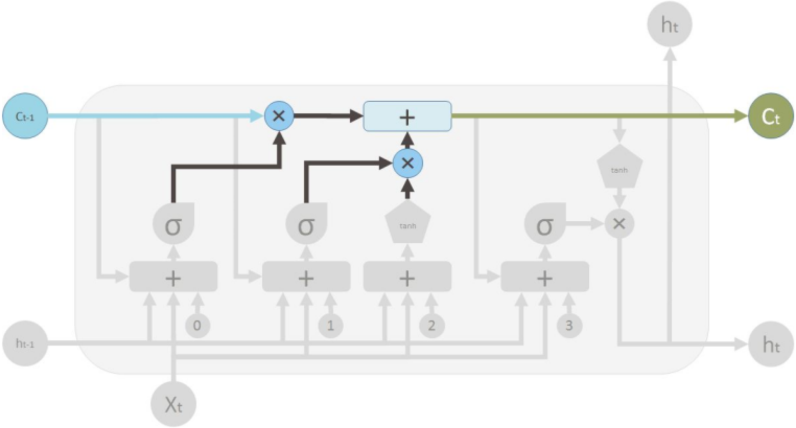
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$
$$\tilde{C}_t = \tanh(W_C x_t + U_C h_{t-1} + b_C)$$



LSTM Memory update

- Aggregation

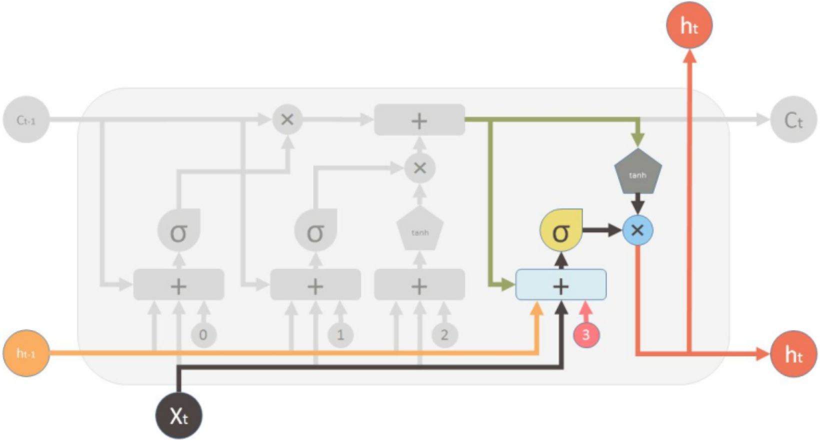
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



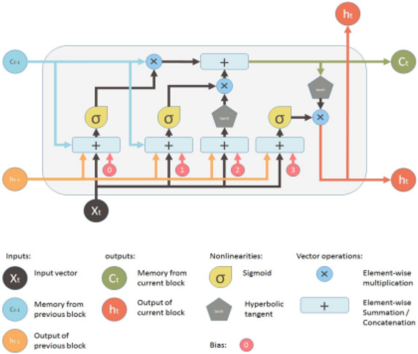
LSTM Output gate

- Conditionally decides what to output from the memory

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$
$$\tilde{h}_t = o_t * \tanh(C_t)$$



LSTM Memory Cell Summary



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

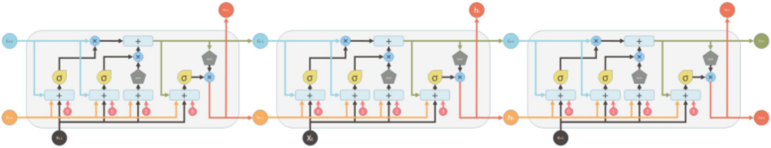
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

$$\tilde{C}_t = \tanh(W_C x_t + U_C h_{t-1} + b_C)$$

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

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

$$\tilde{h}_t = o_t * \tanh(C_t)$$



LSTM Training

- Number of parameters:
  - n number of LSTM units
  - m parameters in the input data
  - Dimension of U is $n \times m$
  - Dimension of W is $n \times n$
  - Dimension of b is n
  - There are four gates in an LSTM cell

$$\text{number of parameters} = 4(nm + n^2 + n)$$

$$
\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
\tilde{C}_t &= \tanh(W_C x_t + U_C h_{t-1} + b_C) \\
C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{h}_t &= o_t * \tanh(C_t)
\end{aligned}
$$

LSTM Training

- Backpropagation Through Time (BPTT) most common
- Weights: Gates, input tanh layer
- Output:
  - One output at each timestep
  - Single output for the whole task