Long Short Term Memory

Long Short Term Memory

- Long Short-Term Memory (LSTM) is an enhanced version of the Recurrent Neural

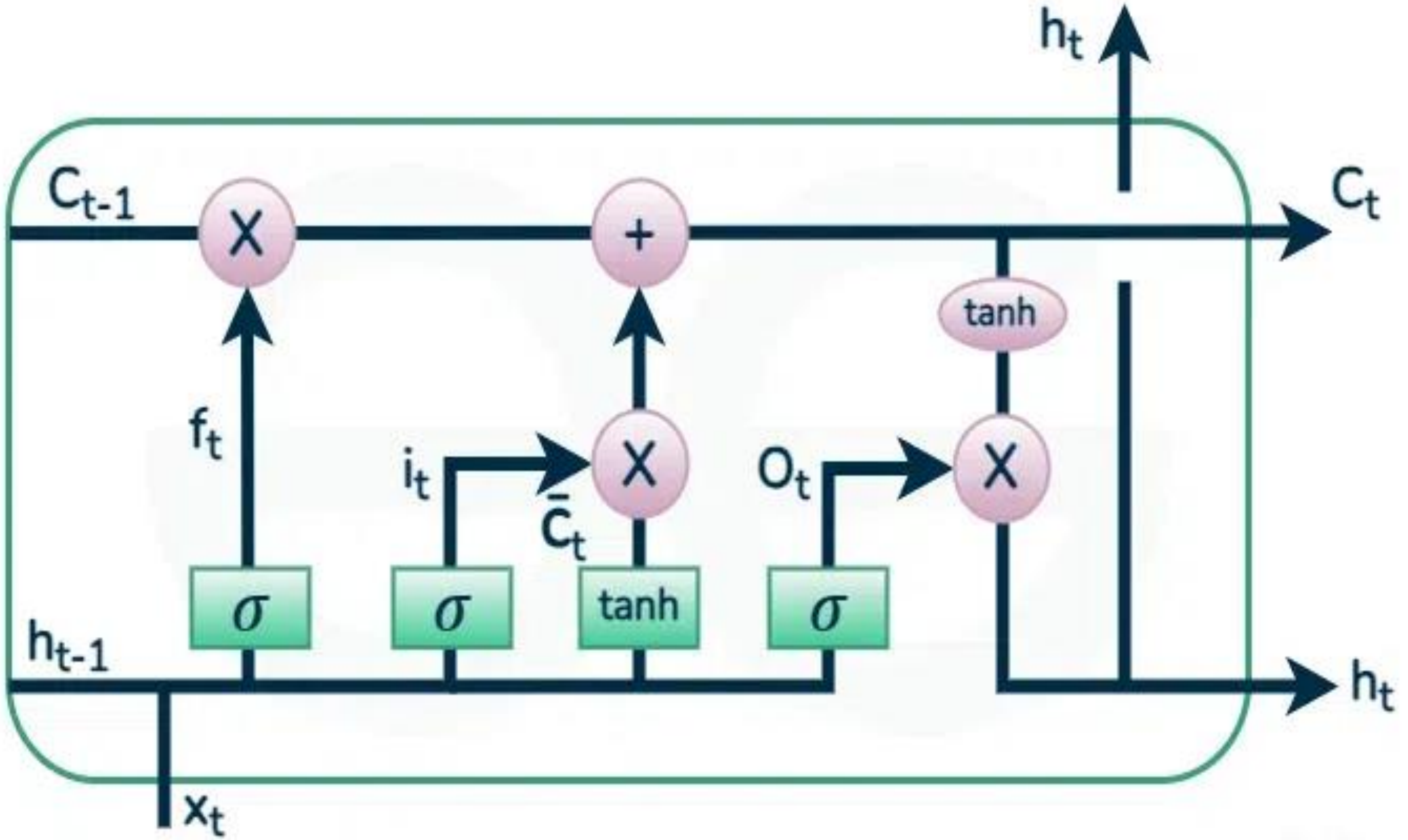
LSTM Architecture

LSTM architectures involves the memory cell which is controlled by three gates

- **Input gate**: Controls what information is added to the memory cell.
- **Forget gate**: Determines what information is removed from the memory cell.
- **Output gate**: Controls what information is output from the memory cell.

These gates decide what information to add to, remove from and output from the memory cell.

Long Short Term Memory



- LSTM architecture has a chain structure that contains four neural networks and different memory blocks called **cells**.
- Information is retained by the cells and the memory manipulations are done by the **gates.** There are three gates

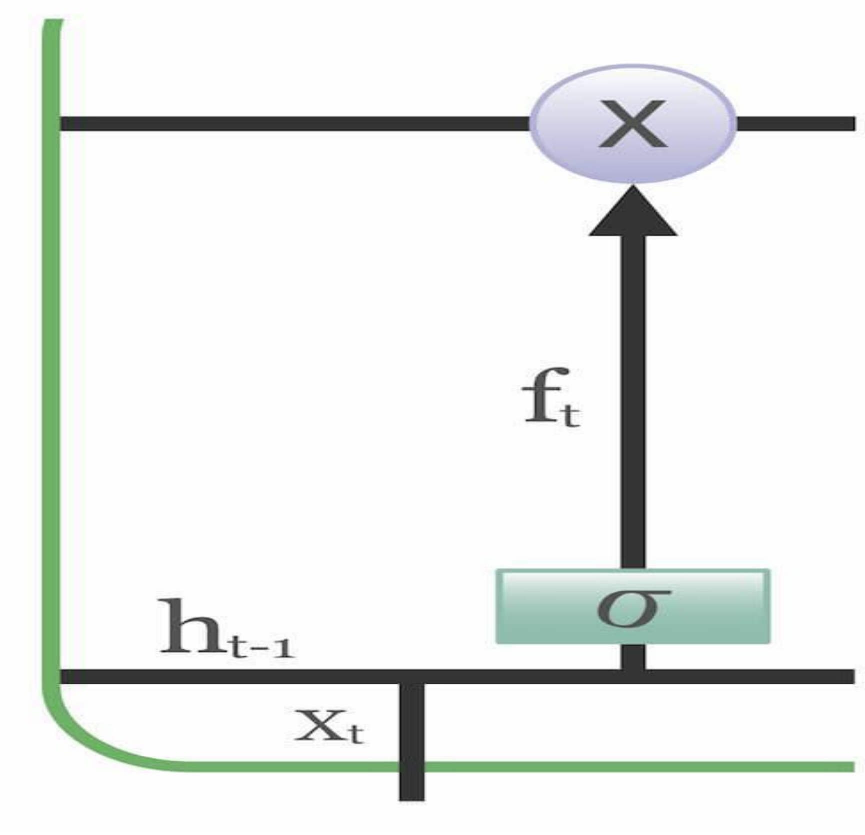
Forget Gate

The equation for the forget gate is:

$$f_t = \sigma(W_f \cdot [h_t - 1, x_t] + b_f)$$

where:

- W_f represents the weight matrix associated with the forget gate.
- $[h_t - 1, x_t]$ denotes the concatenation of the current input and the previous hidden state.
- b_f is the bias with the forget gate.
- σ is the sigmoid activation function.



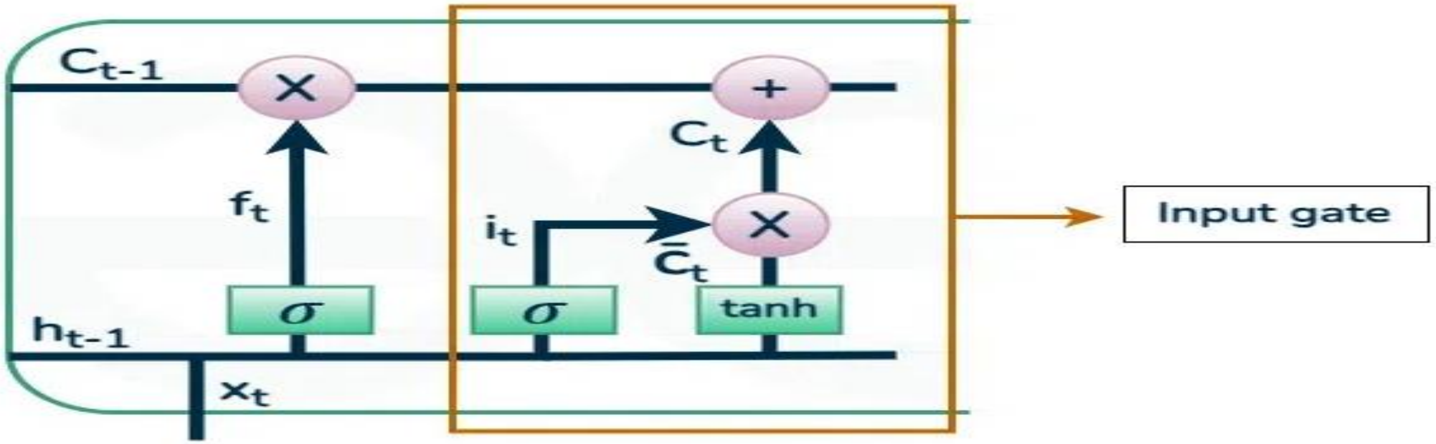
- The information that is no longer useful in the cell state is removed with the forget gate.
- Two inputs x_t (input at the particular time) and h_{t-1} (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias.
- The resultant is passed through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

Input gate

- The 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 *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 the useful information. The equation for the input gate is

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

$$C^t=\tanh(Wc\cdot[h_{t-1},x_t]+b_c$$



- We multiply the previous state by f_t, disregarding the information we had previously chosen to ignore. Next, we include i_t*C_t.
- This represents the updated candidate values, adjusted for the amount that we chose to update each state value.

- $C_t=f_t\odot C_{t-1}+i_t\odot C^t$

Where ⊙ denotes element-wise multiplication

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 tanh function on the cell.
- Then, the information is regulated using the sigmoid function and filter by the values to be remembered using inputs $h_{t-1}$and $_{xt}$.
- 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. The equation for the output gate is:

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

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

> "She loves coffee."

We calculate the forget gate activation for the word **"coffee"** using:

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

Given Values

- **Previous hidden state h_{t-1}:**

$$h_{t-1} = [0.5, 0.3]$$

- **Word embedding of "coffee" x_t:**

$$x_t = [0.6, 0.8]$$

- **Forget gate weight matrix W_f:**

$$W_f = \begin{bmatrix} 0.2 & -0.4 & 0.1 & 0.3 \\ -0.5 & 0.6 & 0.2 & -0.1 \end{bmatrix}$$

- **Forget gate bias b_f:**

$$b_f = [0.1, -0.2]$$

Step 1: Concatenate h_{t-1} and x_t

$$[h_{t-1}, x_t] = [0.5, 0.3, 0.6, 0.8]$$

Step 2: Compute Weighted Sum

$$z_f = W_f \cdot [h_{t-1}, x_t] + b_f$$

Matrix Multiplication

$$\begin{bmatrix} 0.2 & -0.4 & 0.1 & 0.3 \\ -0.5 & 0.6 & 0.2 & -0.1 \end{bmatrix} \cdot \begin{bmatrix} 0.5 \\ 0.3 \\ 0.6 \\ 0.8 \end{bmatrix} + \begin{bmatrix} 0.1 \\ -0.2 \end{bmatrix}$$

$$= \begin{bmatrix} (0.2 \times 0.5) + (-0.4 \times 0.3) + (0.1 \times 0.6) + (0.3 \times 0.8) \\ (-0.5 \times 0.5) + (0.6 \times 0.3) + (0.2 \times 0.6) + (-0.1 \times 0.8) \end{bmatrix} + \begin{bmatrix} 0.1 \\ -0.2 \end{bmatrix}$$

$$= \begin{bmatrix} 0.1 - 0.12 + 0.06 + 0.24 \\ -0.25 + 0.18 + 0.12 - 0.08 \end{bmatrix} + \begin{bmatrix} 0.1 \\ -0.2 \end{bmatrix}$$

$$= \begin{bmatrix} 0.28 \\ -0.23 \end{bmatrix}$$

Step 3: Apply Sigmoid Function

$$f_t = \sigma(z_f) = \begin{bmatrix} \sigma(0.28) \\ \sigma(-0.23) \end{bmatrix}$$

Using:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Approximating:

$$f_t = \begin{bmatrix} 0.57 \\ 0.44 \end{bmatrix}$$

Step 4: Interpretation

- **0.57 (1st unit)** → Keeps **57%** of past memory.
- **0.44 (2nd unit)** → Forgets **56%** of past memory.

Since the values are **not close to 1**, **some past information is forgotten**, allowing the LSTM to process **"coffee"** effectively.