======================================================================

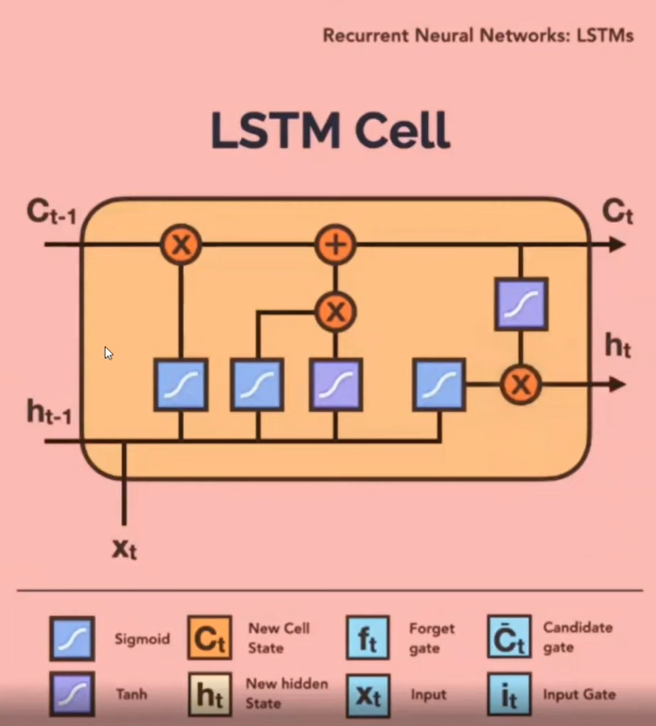
**LSTM**

======================================================================

* Long Short-Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies.
* LSTM — a fundamental concept in deep learning, especially for sequential data.
* LSTM (Long Short-Term Memory) is a special type of Recurrent Neural Network (RNN) that can learn long-term dependencies. It was designed to solve the vanishing gradient problem of traditional RNNs and remember patterns over long sequences.
* Main difference between RNN and LSTM is, RNN have only short-term context, where in LSTM there are two contexts one for short term context (**h**t) and another for long term context (**C**t)

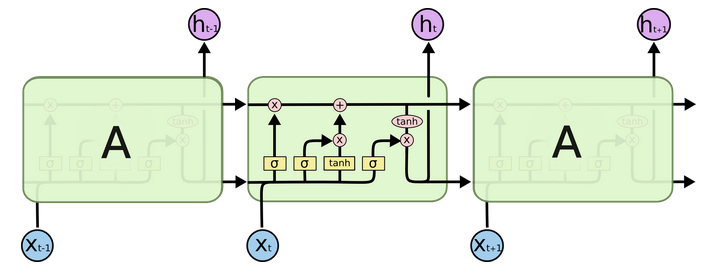
**Why LSTM?**

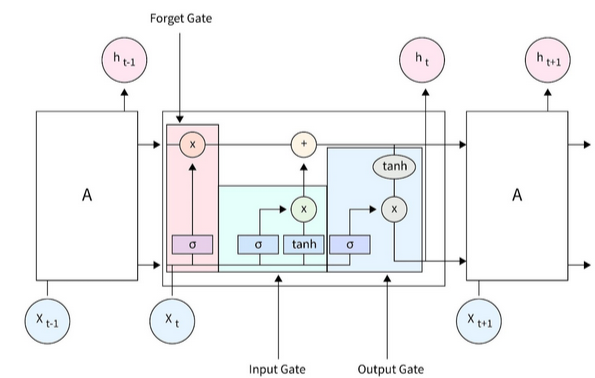
* Traditional RNNs struggle to retain information from earlier time steps as the sequence gets longer. LSTM solves this with **gates** that control the flow of information.



--------------------------------------------------------------------------------------------------------------

**LSTM Architecture:**

****

****

Each LSTM cell has **three gates** and a **cell state**:

**1. Forget Gate**

The forget gate determines what information from the previous cell state should be discarded. It uses a sigmoid function to output values between 0 and 1, where 0 indicates “completely forget” and 1 indicates “completely retain.” This allows the LSTM to reset irrelevant details while retaining essential context, ensuring efficient memory management.

* Input: Current input xₜ and previous hidden state hₜ₋₁
* Output: A number between 0 and 1 (0 = forget, 1 = keep)
* Formula: fₜ = σ(Wf · [hₜ₋₁, xₜ] + bf)

--------------------------------------------------------------------------------------------------------------

**2. Input Gate**

The input gate decides what new information should be added to the cell state. It combines a sigmoid activation (to determine which values to update) with a tanh activation (to create a candidate vector of new information). This mechanism allows the LSTM to integrate fresh, relevant data into its memory.

* + Input gate layer iₜ (controls what to update)
  + Candidate values ĉₜ (possible new content)

Formula:

* + iₜ = σ(Wi · [hₜ₋₁, xₜ] + bi)
  + ĉₜ = tanh(Wc · [hₜ₋₁, xₜ] + bc)

--------------------------------------------------------------------------------------------------------------

**3. Cell State Update**

* **Purpose:** Update the memory cell (Cₜ) with what we decided to forget and add.
* **Formula:**

Cₜ = fₜ \* Cₜ₋₁ + iₜ \* ĉₜ

--------------------------------------------------------------------------------------------------------------

**4. Output Gate**

The output gate determines the information to pass to the next layer or time step. It uses the updated cell state in conjunction with a sigmoid activation to filter relevant outputs. The processed data is scaled using a tan-h function, ensuring the LSTM focuses on meaningful features while suppressing noise.

* **Purpose:** Decide what to **output as the hidden state hₜ**
* **Formula:**

oₜ = σ(Wo · [hₜ₋₁, xₜ] + bo)

hₜ = oₜ \* tanh(Cₜ)

These gates collectively empower LSTMs to address long-term dependencies by dynamically retaining or discarding information, making them highly effective in sequence-based tasks.

--------------------------------------------------------------------------------------------------------------

**What flows through the LSTM:**

* **Cell State Cₜ**: Long-term memory
* **Hidden State hₜ**: Short-term output passed to next time step

--------------------------------------------------------------------------------------------------------------

**Summary of LSTM Gates:**

| Gate | Symbol | Function |
| --- | --- | --- |
| Forget Gate | fₜ | Decides what to forget |
| Input Gate | iₜ, ĉₜ | Adds new memory |
| Output Gate | oₜ | Controls output/hidden state |

Good reference if needed:

<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

<https://www.appliedaicourse.com/blog/lstm-in-machine-learning/>

