Long Short-Term Memory (LSTM) – Easy Explanation and Breakdown

1. What is LSTM?

A **Long Short-Term Memory (LSTM)** is a special kind of Recurrent Neural Network (RNN) designed to **remember important information for a long time** and **forget unimportant details automatically**.

In simple words:

> A normal RNN works like short-term memory, while an LSTM acts like long-term memory — it keeps important facts and forgets what's not needed.

2. Why We Need LSTM

Traditional RNNs can handle short sequences (like 3–4 words), but when the input becomes long (like a full sentence or paragraph), they forget earlier information. This problem is called the **vanishing gradient problem** — the older information's influence becomes smaller and smaller as time passes.

Example: In the sentence:

> *"The movie that I watched last night was really interesting."*

To understand "was really interesting," the network must remember the word "movie" from the start. A simple RNN forgets it, but an LSTM remembers it by maintaining a controlled memory called the **cell state**.

3. How Does LSTM Work?

Each LSTM cell has:

- A **Forget Gate** (f_t) – decides what to forget.
- An **Input Gate** (i_t) – decides what new information to add.
- A **Cell State** (C_t) – the main memory line.
- An **Output Gate** (o_t) – decides what part of memory to show as output.

4. Mathematical Formulas

At each time step t:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C)$$
$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \cdot \tanh(C_t)$$

Where:

- x_t = input at time t
- h_{t-1} = previous hidden output
- C_t = cell memory (long-term memory)
- σ = sigmoid activation (outputs between 0 and 1)
- tanh = hyperbolic tangent function (outputs between -1 and +1)

5. Step-by-Step Example (Including h_t Calculation)

Let's take one time step with small numbers:

$$h_{t-1} = 0.2, \quad C_{t-1} = 0.4, \quad x_t = 0.5$$

Weights (same for all gates):

$$W_f = 0.7, \quad W_i = 0.6, \quad W_C = 0.5, \quad W_o = 0.9, \quad b = 0$$

Step 1 – Forget Gate

$$f_t = \sigma(W_f h_{t-1} + W_f x_t) = \sigma(0.7 \times 0.2 + 0.7 \times 0.5) = \sigma(0.49)$$

$$f_t = \frac{1}{1 + e^{-0.49}} = 0.62$$

Step 2 – Input Gate

$$i_t = \sigma(W_i h_{t-1} + W_i x_t) = \sigma(0.6 \times 0.2 + 0.6 \times 0.5) = \sigma(0.42)$$

$$i_t = \frac{1}{1 + e^{-0.42}} = 0.603$$

Step 3 – Candidate Cell Value

$$\tilde{C}_t = \tanh(W_C h_{t-1} + W_C x_t) = \tanh(0.5 \times 0.2 + 0.5 \times 0.5) = \tanh(0.35)$$

$$\tanh(0.35) = 0.336$$

Step 4 – Update Cell State

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t = (0.62)(0.4) + (0.603)(0.336)$$

$$C_t = 0.248 + 0.203 = 0.451$$

Step 5 – Output Gate

$$o_t = \sigma(W_o h_{t-1} + W_o x_t) = \sigma(0.9 \times 0.2 + 0.9 \times 0.5) = \sigma(0.63)$$

$$\sigma(0.63) = \frac{1}{1 + e^{-0.63}} = 0.652$$

Step 6 – Calculate Hidden State (h_t) The hidden state combines the current memory with the output gate:

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

First, calculate $\tanh(C_t)$:

$$\tanh(0.451) = 0.423$$

Then multiply with the output gate:

$$h_t = 0.652 \times 0.423 = 0.276$$

Therefore, $h_t = 0.276$

Meaning: h_t tells how much of the memory (C_t) the LSTM decides to output. In this case, the model keeps 45% of its total memory internally ($C_t = 0.451$) but shows about 27.6% ($h_t = 0.276$) as visible output.

Final Values:

$$f_t = 0.62, \quad i_t = 0.603, \quad \tilde{C}_t = 0.336, \quad C_t = 0.451, \quad o_t = 0.652, \quad h_t = 0.276$$

6. Real-Life Example: Understanding a Sentence

Consider the sentence:

"The weather today is cold, but tomorrow it will be..."

Each word acts as an input to the LSTM:

$$x_1 = 0.2 \text{ (today)}, \quad x_2 = 0.4 \text{ (cold)}, \quad x_3 = 0.6 \text{ (tomorrow)}$$

At $t = 1$ (today):

$$C_1 = 0.05, \quad h_1 = 0.03$$

A small part of "today" is remembered.

At $t = 2$ (cold):

$$C_2 = 0.15, \quad h_2 = 0.09$$

More focus is given to "cold" (the main weather condition).

At $t = 3$ (tomorrow):

$$C_3 = 0.29, \quad h_3 = 0.18$$

Now the LSTM shifts attention toward "tomorrow," predicting that the next word might be "warm."

7. Understanding h_t in Real Life

- The **forget gate** drops less useful words like "today."
- The **input gate** adds new information like "cold" or "tomorrow."
- The **cell state** keeps long-term meaning ("weather").
- The **output gate** filters this memory to produce visible output h_t.

So, h_t is like your "current thought" — it combines what you just read with what you still remember.

8. Summary Table

Word (x_t)	Forget Gate f_t	Cell Memory C_t	Hidden Output h_t
today (0.2)	0.53	0.05	0.03
cold (0.4)	0.57	0.15	0.09
tomorrow (0.6)	0.61	0.29	0.18

9. In Simple Words

The LSTM reads the sentence word by word. It forgets old details ("today"), remembers key facts ("cold"), and predicts what's next ("tomorrow → warm"). Mathematically, $h_t = o_t \times \tanh(C_t)$ means the output is a filtered version of the memory — just like how your brain only expresses what's most relevant right now.