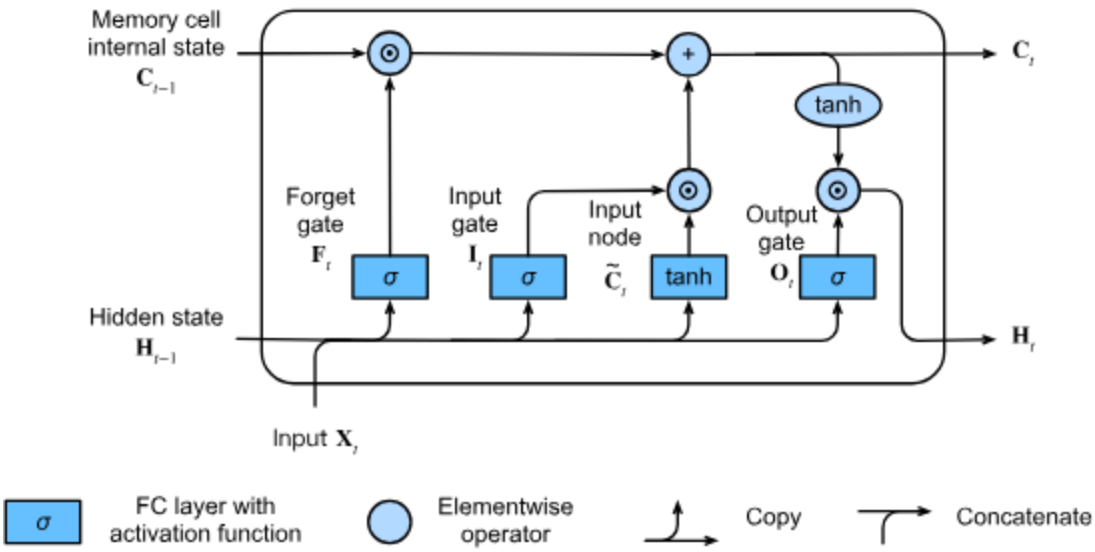

🔁 What is LSTM?

LSTM is a type of **Recurrent Neural Network (RNN)** designed to **remember long-term dependencies** and avoid problems like **vanishing/exploding gradients**.

📐 LSTM Cell – Mathematical Explanation

Each LSTM cell has three main gates:

1. **Forget Gate**
2. **Input Gate**
3. **Output Gate**

✅ Variables:

- Input at time t: xtx_t
- Hidden state at time t-1: ht−1h_{t-1}
- Cell state at time t-1: Ct−1C_{t-1}
- Output hidden state at time t: hth_t
- Cell state at time t: CtC_t

Let’s define the weight matrices and biases:

- W_f, W_i, W_C, W_o: weight matrices for forget, input, cell update, and output gates.
- b_f, b_i, b_C, b_o: biases

🧮 Equations

1. **Forget Gate**: What to forget from the previous cell state

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

2. **Input Gate**: What new information to add

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

3. **Cell State Update**:

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

4. **Output Gate**: What to output

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

σ = sigmoid activation
tanh = hyperbolic tangent activation

📊 Intuition Example

Let's say we're predicting the next word in the sentence:

"**The stock market is ___**"

The model should remember **"stock market"** to predict something like "volatile" or "bullish". A basic RNN might forget "stock", but LSTM can **retain** this context over long distances.

🧪 Code Example – Keras

Let's use LSTM to predict a sequence using a simple time-series dataset.

▶️ Predicting Sine Wave Using LSTM

```
import numpy as np

import matplotlib.pyplot as plt

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import LSTM, Dense
```

```
# Generate sine wave data
x = np.linspace(0, 50, 500)
y = np.sin(x)

# Prepare input/output sequences
def create_dataset(data, step=10):
    X, Y = [], []
    for i in range(len(data) - step):
        X.append(data[i:i+step])
        Y.append(data[i+step])
    return np.array(X), np.array(Y)

step = 10
X, Y = create_dataset(y, step)

# Reshape to [samples, time steps, features]
X = X.reshape((X.shape[0], X.shape[1], 1))

# Build LSTM model
model = Sequential([
    LSTM(50, activation='tanh', input_shape=(step, 1)),
    Dense(1)
])
model.compile(optimizer='adam', loss='mse')
model.fit(X, Y, epochs=20, verbose=1)

# Predict
pred = model.predict(X)
```

```
# Plot original and predicted
plt.plot(Y, label="True")
plt.plot(pred, label="Predicted")
plt.legend()
plt.title("LSTM Sine Wave Prediction")
plt.show()
```

✅ **Summary**

Component	Purpose
Forget Gate	Decides what to throw away from the cell state
Input Gate	Adds new information to the cell state
Output Gate	Decides what to output from the cell state
Cell State	Memory of the network
Hidden State	Output at the current timestep
