LSTM Architecture: Memory Management

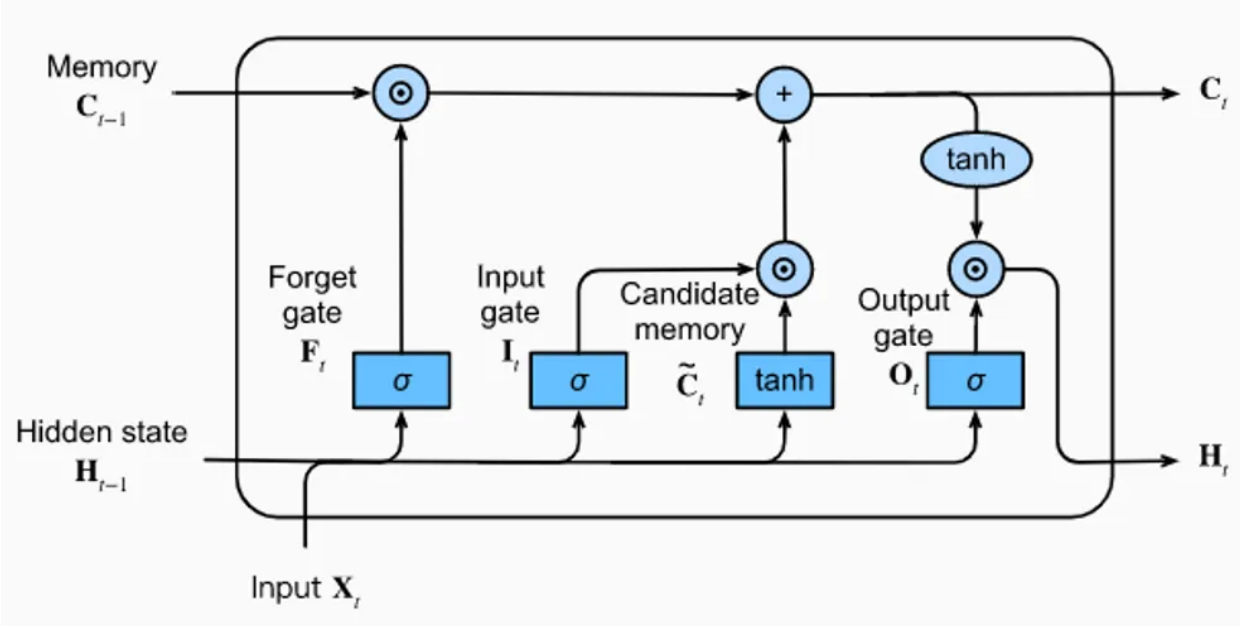
Overview

Long Short-Term Memory (LSTM) networks represent a breakthrough in handling sequential data, particularly in the domain of Recurrent Neural Networks (RNNs). This section delves into the intricate architecture of LSTMs, with a keen focus on memory management mechanisms and their role in mitigating the vanishing gradient problem.

Contents

Introduction

LSTM networks, pioneered by Hochreiter and Schmidhuber in 1997, have revolutionized the processing of sequential data by addressing the notorious vanishing gradient problem prevalent in traditional RNNs. Their remarkable capability to capture long-term dependencies has rendered them indispensable across various applications, including natural language processing, time series analysis, and speech recognition.



LLM as Aligner

Memory Management in LSTMs

At the heart of LSTM networks lie intricate memory management mechanisms orchestrated through specialized components known as gates. These gates meticulously regulate the flow of information in and out of the memory cell, empowering LSTMs to selectively retain or discard information over time.

Input Gate

The input gate, denoted as i_t, governs the influx of new information into the cell state. It

generates a candidate state $\tilde{C}_t$, comprising potential new values for the cell state. Analogous to the forget gate, it operates on the concatenation of the current input x_t and the previous hidden state h_{t-1}:

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

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

Where:

- W_i denotes the weight matrix for the input gate.
- W_c represents the weight matrix for generating the candidate state.
- b_i and b_c stand for the bias vectors for the input gate and candidate state, respectively.
- σ denotes the sigmoid activation function.
- $\tanh$ signifies the hyperbolic tangent activation function.

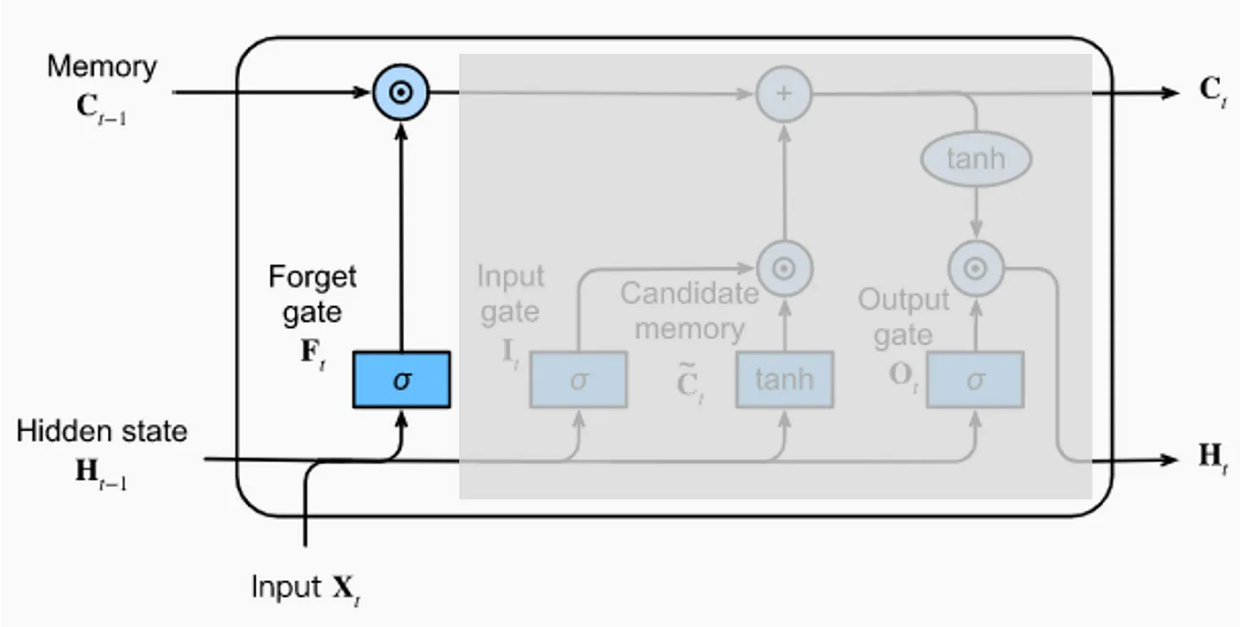
Forget Gate

The forget gate, denoted as f_t, assumes the critical role of deciding which information to discard from the previous cell state C_{t-1}. Operating on the concatenation of the current input x_t and the previous hidden state h_{t-1}, the forget gate employs a sigmoid activation function to produce a forget vector f_t:

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

Where:

- W_f represents the weight matrix for the forget gate.
- b_f signifies the bias vector for the forget gate.
- σ denotes the sigmoid activation function.



Forget Gate

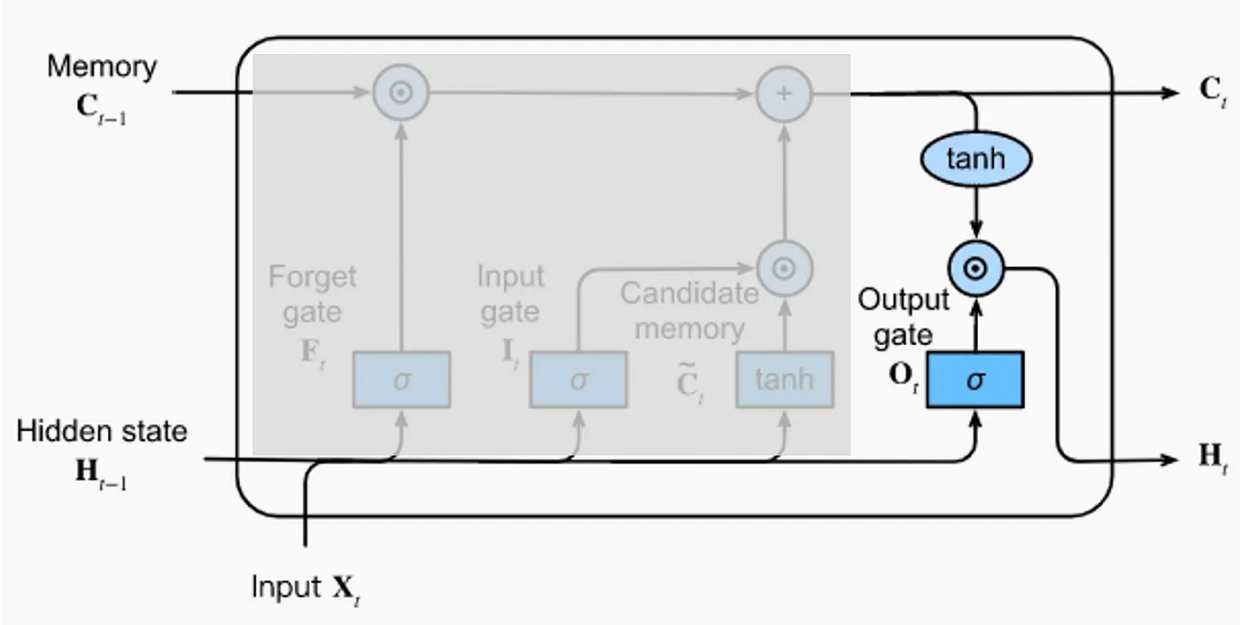
Output Gate

The output gate, denoted as o_t, determines the relevance of processed information from the current cell state C_t for the network's output. Similar to the forget and input gates, it operates on the concatenation of the current input x_t and the previous hidden state h_{t-1}:

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

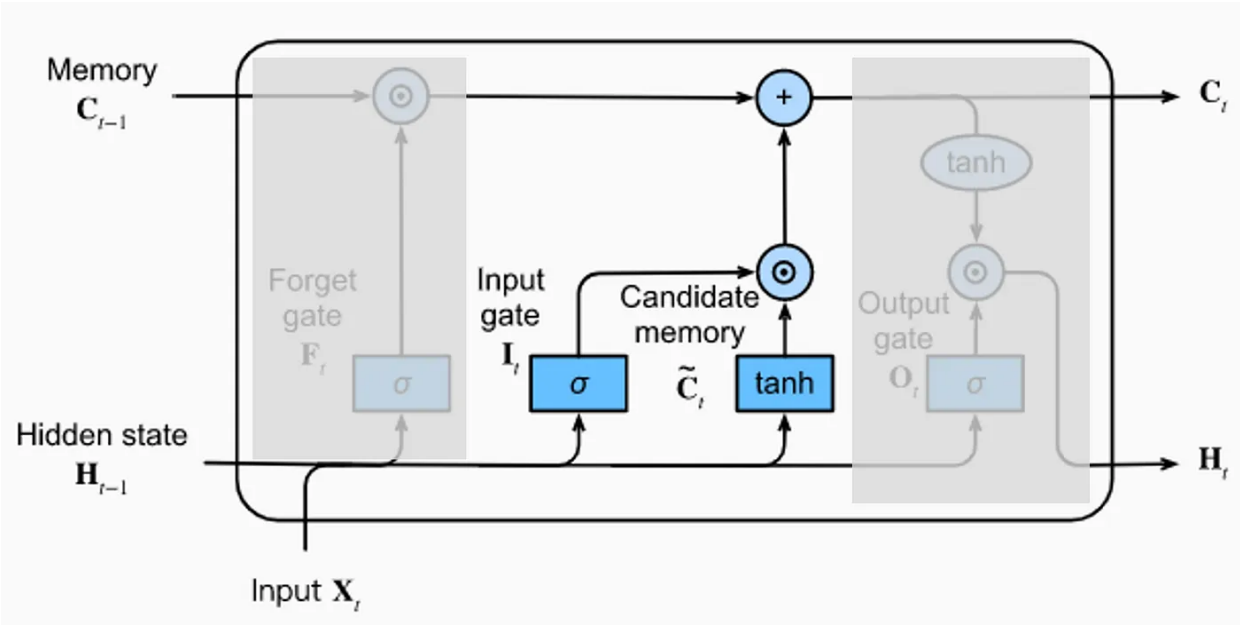
Where:

- W_o stands for the weight matrix for the output gate.
- b_o denotes the bias vector for the output gate.
- σ signifies the sigmoid activation function.



Output Gater

Candidate Memory The candidate memory generates new information to be potentially added to the cell state. It operates on a hyperbolic tangent function to ensure the candidate vector's values fall within a normalized range, facilitating its integration into the cell state.



Memory LTSM

Information Flow in LSTMs

LSTMs carefully regulate the flow of information through the memory cell, balancing

past, present, and output. This orchestration ensures that relevant information is retained while irrelevant information is discarded, allowing the network to learn and adapt over time.

LSTM Gates: Regulating Information Flow

The gates in LSTMs, including the forget gate, input gate, and output gate, play a crucial role in regulating information flow within the memory cell. These gates employ activation functions to control the passage of information, allowing LSTMs to selectively update their memory based on current inputs and past states.

Backpropagation Through Time (BPTT)

Backpropagation Through Time (BPTT) is a pivotal algorithm tailored for Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTM) networks. Its primary objective is to combat the vanishing gradient problem, a common obstacle in training RNNs on long sequential data.

The Vanishing Gradient Problem

When traditional RNNs attempt to learn from sequences extending over many time steps, the gradients used to update the network's parameters tend to diminish exponentially as they propagate backward through time. This phenomenon is known as the vanishing gradient problem, which severely impedes the network's ability to capture long-term dependencies.

BPTT Algorithm

BPTT addresses the vanishing gradient problem by unfolding the RNN across time steps, effectively converting it into a deep feedforward neural network. This unfolding allows for the application of standard backpropagation, enabling the network to learn from long sequences and capture long-term dependencies more effectively.

Notation:

- t: Time step (integer)
- x_t: Input vector at time step t
- h_t: Hidden state vector at time step t
- c_t: Cell state vector at time step t
- f_t: Forget gate vector at time step t (output of forget gate)
- i_t: Input gate vector at time step t (output of input gate)
- o_t: Output gate vector at time step t (output of output gate)
- W_f, W_i, W_o, W_c: Weight matrices for forget gate, input gate, output gate, and candidate memory network, respectively
- b_f, b_i, b_o, b_c: Bias vectors for forget gate, input gate, output gate, and candidate memory network, respectively
- σ: Sigmoid activation function
- $\tanh$: Hyperbolic tangent activation function
- d_t: Target value at time step t
- E_t: Error at time step t

1. Forward Pass:

In the forward pass, we calculate the hidden state (h_t) and cell state (c_t) for each time step based on the input (x_t), previous hidden state (h_{t-1}), and previous cell state c_{t-1}.

Forget Gate : Selective Memory in LSTMs

The forget gate (f_t) plays a critical role in LSTM networks by acting as a memory filter, deciding which information from the previous cell state $C_(t-1)$ should be retained and which can be discarded. It ensures the LSTM doesn't become overloaded with irrelevant or outdated information, allowing it to focus on what's most relevant for the current task.

1. Input and Processing:

- The forget gate takes two inputs:

- **Current Input (x_t):** Represents the new information at the current time step (t).
- **Previous Hidden State $(h_(t-1))$:** Captures the network's understanding of the sequence up to the previous time step.
- These inputs are combined using a matrix multiplication with separate weight matrices for the forget gate (W_f) and then added to a bias vector (b_f). This weighted summation allows the network to learn which aspects of the current input and previous hidden state are most relevant for determining what to forget.
- The combined result is passed through the sigmoid activation function (σ).
  2. Sigmoid Activation Function (σ):
- The sigmoid function squashes the weighted sum between 0 and 1. This transformation ensures the output of the forget gate (f_t) lies within this range.
  3. Output Interpretation:
- Each element in the forget vector f_t) corresponds to a specific element in the previous cell state $C_(t-1)$.
- A value close to **1** in f_t indicates that the corresponding information in $t-1$ is deemed important and will be **retained**.
- A value close to **0** in f_t signifies that the corresponding information is considered less relevant and will be **forgotten**.

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

b. Input Gate: Gating the Flow of New Information

The input gate (i_t) in an LSTM network acts as a control mechanism, regulating the flow of new information (candidate state) into the cell state (C_t). It ensures that only the most relevant parts of the current input (x_t) and the previous hidden state $h_(t-1)$ are incorporated into the cell state, preventing the network from becoming overwhelmed with irrelevant data.
Here's an input gate's function:

1. Input and Processing:

- Similar to the forget gate, the input gate takes two inputs:
- **Current Input** (x_t)**:** Represents the new information at the current time step (t).
- **Previous Hidden State** $(h_(t-1))$**:** Captures the network's understanding of the sequence up to the previous time step.
- These inputs are combined using matrix multiplication with separate weight matrices for the input gate (W_i) and then added to a bias vector (b_i). This weighted summation allows the network to learn how to combine the current input and past knowledge to determine what new information is most valuable.

2. Activation Functions:

- The combined result is passed through two separate activation functions:
- **Sigmoid Activation Function** (σ)**:** This function transforms the weighted sum for the input gate itself, generating an output (i_t) between 0 and 1. This output acts as a gatekeeper, controlling the extent to which new information can influence the cell state update.
- **Hyperbolic Tangent Function** $(tanh)$**:** This function is applied to a separate weighted sum involving the current input and previous hidden state (similar to the forget gate). The tanh function ensures the candidate state $(C_t\ )$ has values ranging from -1 to 1, facilitating its integration into the cell state.

3. Output Interpretation:

- The sigmoid activation function's output (i_t) determines the degree of influence the candidate state $(C_t\ )$ has on the cell state update.
- A value close to **1** in i_t signifies that the candidate state $(C_t\ )$ will have a **strong impact**, allowing more new information to be incorporated into the cell state.
- A value close to **0** in i_t indicates that the candidate state $(C_t\ )$ will have a **weak influence**, resulting in less new information being integrated.

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

$$C_t = \tanh(W_c \cdot [x_t, h_{t-1}] + b_c)$$

c. Cell State: The Core Memory of an LSTM

The cell state (C_t) in an LSTM serves as the network's internal memory, carrying vital information across time steps. It acts as a central hub where the forget gate and input gate collaborate to determine what information to retain from the past (previous cell state) and what new information to integrate from the current input.

1. Leveraging the Forget Gate and Input Gate:

- The forget gate (f_t) and input gate (i_t) play crucial roles in influencing the cell state update.
- The forget gate's output (f_t) acts as a scaling factor for the previous cell state $(C_(t-1))$, determining how much information from the past is retained.
- The input gate's output (i_t) controls the influence of the candidate state (C_t~) on the cell state update. The candidate state, generated using the hyperbolic tangent function $(tanh)$, represents potential new information to be integrated.

The cell state update is governed by the following formula:

$$lC_t = f_t \cdot C_{t-1} + i_t \cdot C_t$$

- C_t**:** Current cell state at time step t.
- f_t**:** Forget gate vector at time step t (output of forget gate).
- $C_(t-1)$**:** Previous cell state at time step t-1.
- i_t**:** Input gate vector at time step t (output of input gate).
- C_t**:** Candidate state at time step t.

3. Interpretation:

- The cell state update effectively combines the following elements:
- **Retained Information** $(f_t * C_(t-1))$: The forget gate selectively scales the previous cell state, preserving relevant past information.

- **New Information** $(i_t * C_t)$: The input gate controls the influence of the candidate state, allowing the cell state to incorporate valuable new information from the current input.

$$c_t = f_t \cdot c_{t-1} + i_t \cdot C_t$$

e. Hidden State: The Bridge Between Past and Present

In LSTMs, the hidden state (h_t) serves as a crucial intermediary, bridging the gap between the network's understanding of the past and its ability to process the present. It acts as a compressed representation of the most relevant information extracted from the sequence up to the current time step (t).

Key Functions of the Hidden State:

1. **Output at the Current Time Step (h_t):** This function directly reflects the network's current understanding of the sequence. It represents the culmination of how the LSTM has processed information from the current input (x_t) and the previous hidden state (h_(t-1)), incorporating knowledge from the past (via the cell state) to create a meaningful representation of the present context. This output can be used for various purposes depending on the application:

- **Classification:** In tasks like sentiment analysis of a review, the hidden state at the final time step might capture the overall sentiment and be used to classify the review as positive, negative, or neutral.

- **Prediction:** For tasks like machine translation or text generation, the hidden state at each time step might influence the network's prediction of the next word or character, ensuring coherence with the preceding sequence.

- **Feature Extraction:** In tasks like video analysis or audio processing, the hidden state could be used as a compact representation of the features extracted from the current frame or audio segment, feeding into further processing stages.

2. **Input for the Next Time Step (h_(t+1)):** The hidden state not only reflects the current context but also acts as a crucial input for the next time step (h_(t+1)). This

allows the LSTM to maintain a continuous understanding as it progresses through the sequence. By passing on the most relevant information from the previous time step, the hidden state enables the network to analyze new information in the context of what has already been processed.

Relationship with the Cell State:

While the hidden state provides a compressed representation for the network's output and future processing, it's important to distinguish it from the cell state (C_t). The cell state holds the complete, uncompressed information from past time steps. The output gate (o_t) acts as a filter, deciding which aspects of the cell state are most relevant to be included in the hidden state, considering both the current input and past knowledge.

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

c. Hidden State Gradient:

$$\frac{\partial E_t}{\partial h_t} = o_t \cdot (1 - \tanh^2(c_t)) \cdot \frac{\partial E_t}{\partial y_t} + \sum \left(\frac{\partial E_{t+1}}{\partial h_{t+1}} \right) \cdot W_{o,(t+1)} \cdot o_{t+1}$$

d. Cell State Gradient:

$$\frac{\partial E_t}{\partial c_t} = \left(\frac{\partial E_t}{\partial h_t} \right) \cdot o_t \cdot (1 - \tanh^2(c_t)) + \frac{\partial E_{t+1}}{\partial c_{t+1}} \cdot f_{t+1}$$

e. Forget Gate Gradient:

$$\frac{\partial E_t}{\partial f_t} = \frac{\partial E_t}{\partial c_t} \cdot c_{t-1} + \sum \left(\frac{\partial E_{t+1}}{\partial c_{t+1}} \right) \cdot W_c \cdot i_{t+1} \cdot \tanh(W_c \cdot [x_t, h_{t-1}] + b_c)$$