
CO4

# DEEP LEARNING

23AD2205A

Topic:

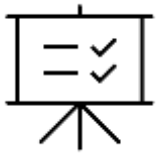
# Solving the Vanishing Gradient Problem with LSTMs

Session - 25

# AIM OF THE SESSION

To familiarize students with the  Solving the Vanishing Gradient Problem with LSTMs

# INSTRUCTIONAL OBJECTIVES



This Session is designed to:

1 Discuss Solving the Vanishing Gradient Problem with LSTMs
2 Demonstrate the  Solving the Vanishing Gradient Problem with LSTMs
3 Discussion on Solving the Vanishing Gradient Problem with LSTMs

# LEARNING OUTCOMES

At the end of this session, you should be able to: concepts for real time applications

1. To demonstrate Solving the Vanishing Gradient Problem with LSTMs
2. To apply **SOLVING THE VANISHING GRADIENT PROBLEM WITH LSTMS**

# Solving the Vanishing Gradient Problem with LSTMs

Introduction

Vanishing Gradient Problem: A significant challenge in training traditional RNNs, where gradients used to update the network weights diminish exponentially as they propagate back through time, making it difficult for the network to learn long-term dependencies.

Key Aspects of the Vanishing Gradient Problem

Definition: When training neural networks, the gradient of the loss function with respect to the weights becomes very small, causing the weights to update very slowly or not at all.
Impact: This problem severely affects the learning capacity of RNNs for long sequences, as the network fails to capture long-term dependencies.

LSTM Networks
Long Short-Term Memory (LSTM) networks are designed to overcome the vanishing gradient problem.
Architecture: LSTMs introduce a more complex architecture with additional gates to control the flow of information.

LSTM Components
Memory Cell: Stores information across time steps.
Gates: Regulate the flow of information into and out of the cell.
Input Gate: Controls how much new information from the current input flows into the memory cell.
Forget Gate: Decides how much of the past information to forget.
Output Gate: Determines how much information from the memory cell to output.

How LSTMs Solve the Vanishing Gradient Problem

Constant Error Flow: The architecture of LSTMs allows for a constant error flow through the network, which helps in preserving gradients over long sequences.

2. Gate Mechanism:

- Input Gate: $i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$
- Forget Gate: $f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$
- Output Gate: $o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$
- Cell State Update: $C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$
- Hidden State Update: $h_t = o_t \odot \tanh(C_t)$

Advantages of LSTMs

Capture Long-Term Dependencies: LSTMs can effectively learn long-range dependencies in sequential data.

Better Gradient Flow: The gating mechanisms allow gradients to flow more effectively through the network during backpropagation.

Versatility: LSTMs are used in a wide range of applications, including language modeling, machine translation, and time series forecasting.

Practical Applications

Natural Language Processing (NLP): LSTMs are used in tasks such as text generation, language translation, and sentiment analysis.

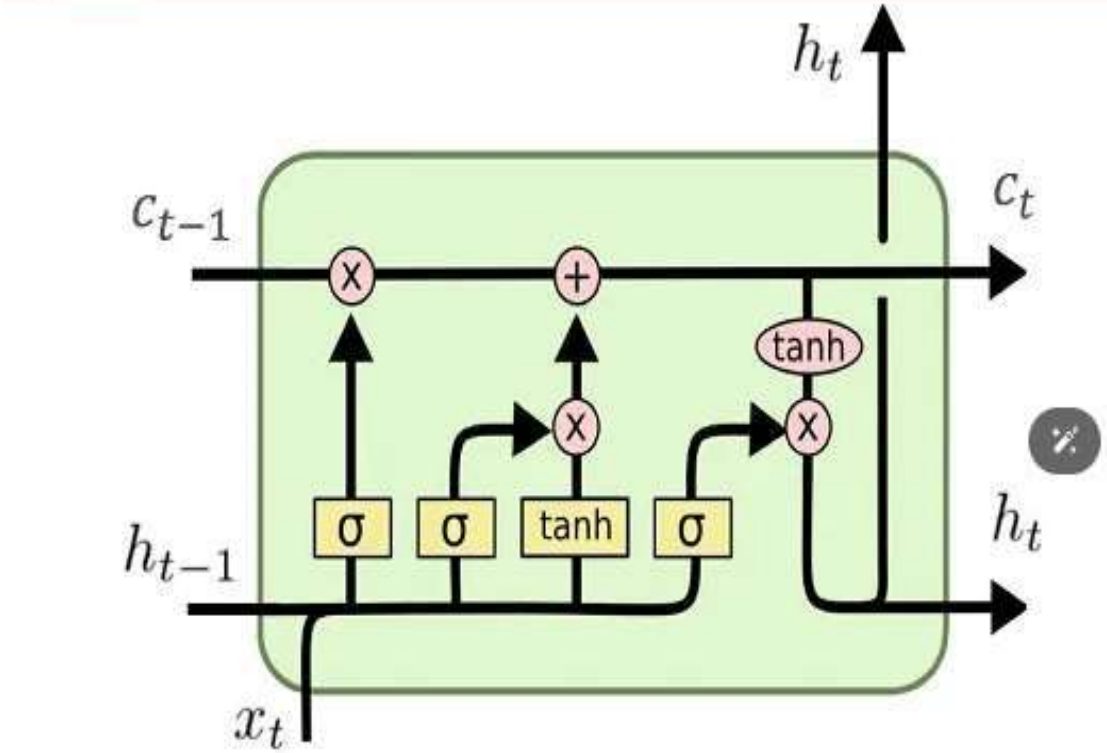
Time Series Prediction: LSTMs can predict future values in time series data, such as stock prices and weather conditions.

Speech Recognition: LSTMs improve the performance of speech-to-text systems by capturing temporal dependencies in audio signals.

How Does LSTM Improve the Vanishing Gradient Problem?

LSTMs, introduced by Sepp Hochreiter and Jürgen Schmidhuber in 1997, were designed to focus on overcoming the vanishing gradient problem. To do so, LSTM leverages gating mechanisms to control the flow of information and gradients. This helps prevent the vanishing gradient problem and allows the network to learn and retain information over longer sequences.

The figure below shows the structure of an LSTM cell:




LSTM
(Long-Short Term Memory)

There are three gates included in LSTMs: the input gate, the forget gate, and the output gate. These gates control the flow of information through the LSTM cell, allowing it to decide what to remember, what to forget, and what to output. Furthermore, these gates allow LSTMs to control the flow of gradients through time, effectively addressing the vanishing gradient problem.

4.1. Forget Gate

The forget gate decides which information to forget from the previous hidden state and the current input. Using a sigmoid activation function, it produces values between 0 and 1, ideally indicating the fraction of data that passes through the cell state.

In addition, the forget gate lets LSTM hold onto crucial data and let go of the unnecessary, ensuring the gradients remain relevant.

Input Gate and Output Gate

In contrast to the forget gate, the input gate establishes which new details should be added to the cell state. It has two parts: a sigmoid activation function that selects values to update the cell and a tanh activation function that produces new candidate values for the cell state.

Meanwhile, the output gate defines what data should be presented as the LSTM cell's current output. It processes the updated cell state and the present hidden state to produce the output. The output gate delivers meaningful results by refining the data inside the cell state. These gates are crucial in pinpointing the essential data for the given task, preserving significant information across extended sequences, and thus combating the vanishing gradient issue.

Use below link for detailed explanation

https://prvnk10.medium.com/how-lstms-solve-the-problem-of-vanishing-gradients-ea88f08c78ca

Both the input and output gates multiply their results with other factors. Their nature ensures that if the output is near 1, gradients pass unhindered, but if it's around 0, the flow is stopped. These gates help select which data is essential for the task at hand. Therefore, they help maintain only crucial information over longer sequences, which relieves the vanishing gradient problem.

Memory Cell State

Combining results from the input gate and the previous cell state refines the cell's state at the current time step. This memory cell state holds data across various time steps, allowing the network to capture longer-range dependencies. Due to the additive update mechanism, the LSTM's memory cell ensures gradients remain consistent over lengthy sequences.

With these gate mechanisms in place, LSTMs better handle the vanishing gradient issue. They excel in tasks with an understanding of long-term sequences and dependencies.

Limitation of LSTM

While Long Short-Term Memory (LSTM) networks effectively address the vanishing gradient problem and recognise long-term patterns in sequences, they come with their own set of challenges.

Gradient Explosion

Even though LSTMs are better at addressing the vanishing gradient problem than traditional RNNs, they can still suffer from gradient explosion in certain cases. Gradient explosion happens when the gradients become extremely large during backpropagation, especially with the very long input sequence or initialising the network with large weights.

Limited Contextual Information

LSTMs can capture dependencies over longer sequences than simple RNNs, but they still have a limited context window. When processing extremely long sequences, LSTMs might struggle to remember earlier information and replace the older data with newer inputs. This happens when the forget gate becomes overly dominant.

Complexity and Training Time

LSTMs, with their gating mechanisms and multiple components, are more complex than the vanilla RNNs. As a result, they possess about four times the parameters of a basic RNN. This complexity can lead to longer training times and require more careful hyperparameter tuning. Additionally, training deep LSTM networks remains challenging due to vanishing gradients in very deep architectures.

Gating Mechanism Sensitivity

LSTM's performance relies on the precise configuration of its gating mechanism. Inappropriate gate weight setups or poorly chosen hyperparameters might degrade the model's learning efficiency, giving rise to gradient-related problems.

# SELF-ASSESSMENT QUESTIONS

LSTM networks use _____ to control the flow of information and solve the vanishing gradient problem.

gates

The _____ gate in an LSTM decides how much of the past information should be forgotten

forget

The vanishing gradient problem makes it difficult for traditional RNNs to learn _____ dependencies in sequential data.

long-term

# Conclusion

LSTMs address the vanishing gradient problem by using memory cells and gates to regulate the flow of information.

The architecture allows for better gradient flow, enabling the learning of long-term dependencies

.Applications: LSTMs are widely used in various domains due to their ability to handle sequential data effectively.

# TERMINAL QUESTIONS

•**Apply** the concept of LSTM to solve the vanishing gradient problem in sequence prediction. What are the key components of an LSTM?

•**Demonstrate** how the forget gate in anLSTM works.

•**Use** an LSTM model to predict stock prices. How would you prepare the data?

•**Illustrate** the training process of an LSTM network for text generation. How do the gates help?

# REFERENCES FOR FURTHER LEARNING OF THE SESSION

**Books:**

1 Ian Goodfellow and Yoshua Bengio and Aaron Courville (2016) Deep Learning Book
2.Deep Learning Book.  eep Learning with Python, Francois Chollet , Manning publications, 2018

# Resources

- https://www.tensorflow.org/tutorials/generative/autoencoder
- https://www.linkedin.com/company/autoencoder?originalSubdomain=in
- https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-are-autoencoders-in-deep-learning
- https://blog.keras.io/building-autoencoders-in-keras.