edureka!
a Veranda Enterprise | ILLINOIS TECH

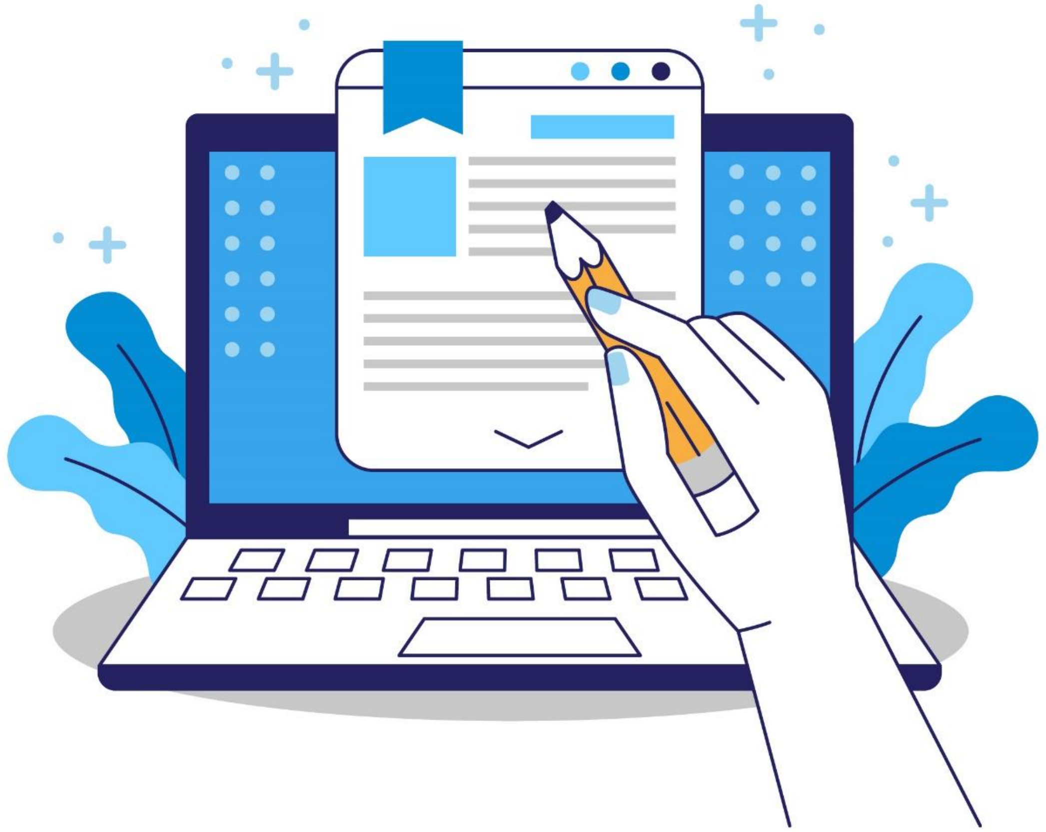
POST GRADUATE PROGRAM IN GENERATIVE AI AND ML

Deep Learning and Neural Network Architectures

Long Short-Term Memory (LSTM) Networks

Topics

- Need for LSTM and Problems with RNNs
- Gradient Issues in RNNs and the Need for Long Memory
- LSTM Cell and Full LSTM Cell
- Forget Gate, Input Gate, Output Gate, and Cell State Update
- Bidirectional LSTM and Why Bidirectional Helps
- Stacked LSTM and Seq2Seq
- Choosing the Right LSTM
- LSTM for Text Classification and Sentiment Analysis Pipeline
- Intent Detection with LSTM and LSTM in Speech Tasks
- Tools Using LSTM in Real Life
- Why LSTM Needs Attention and How to Add Attention
- What is GRU, and the working of GRU
- When to Use LSTM or GRU



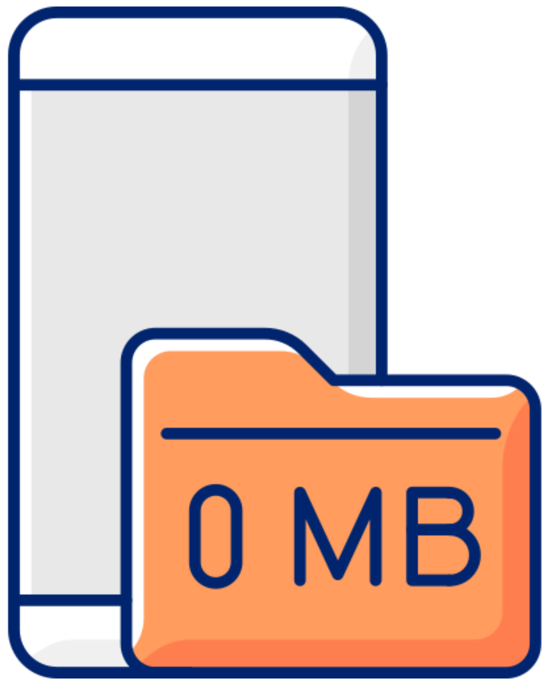
Learning Objectives

By the end of this lesson, you will be able to:

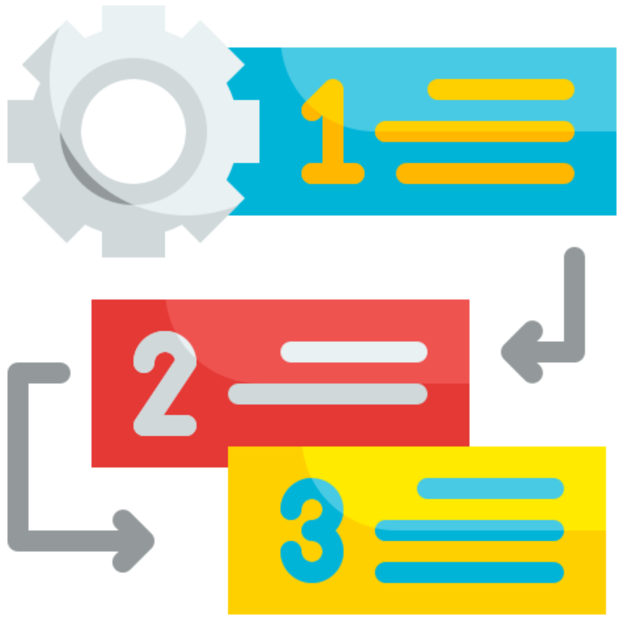
- Understand the limitations of RNNs and the motivation behind LSTM architecture, including its gating mechanisms.
- Explore different LSTM variants such as Bidirectional and Stacked LSTMs and learn when to apply them effectively.
- Apply LSTMs to real-world NLP tasks like text classification, sentiment analysis, and intent detection.
- Analyze the role of attention mechanisms and compare LSTM with GRU to choose the right model for specific tasks.

Need for LSTM

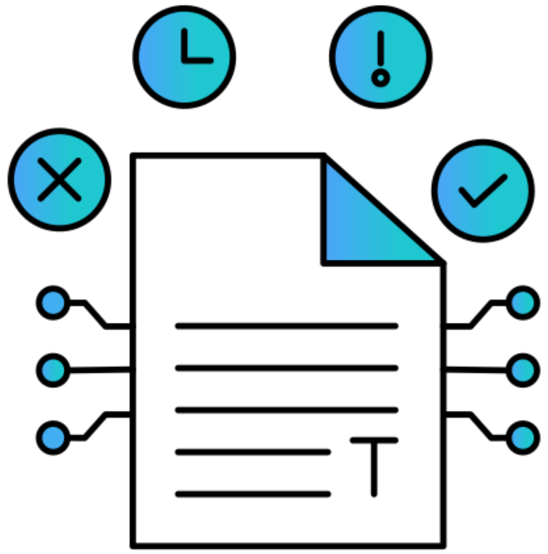
Problem with RNNs



Short Memory

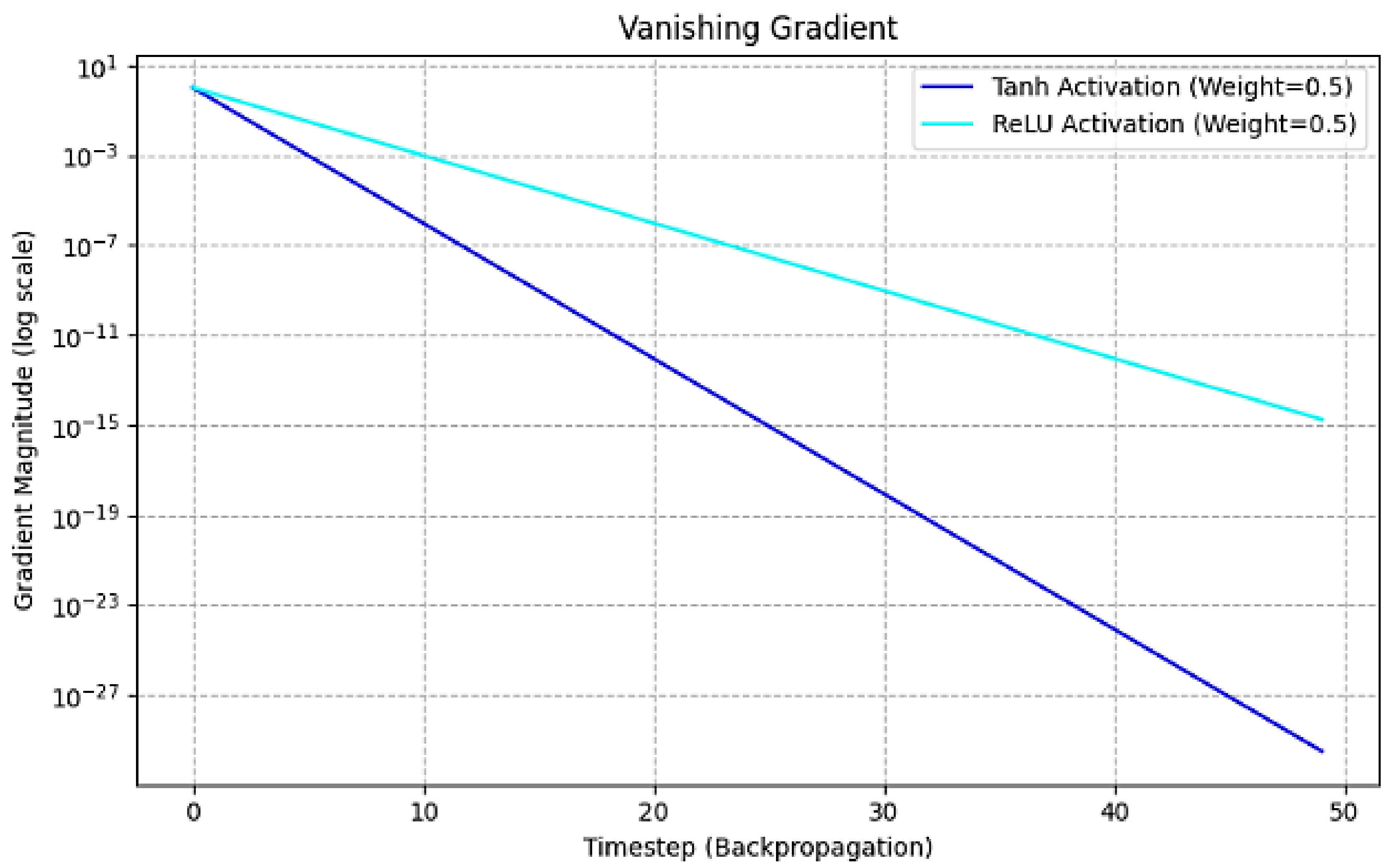


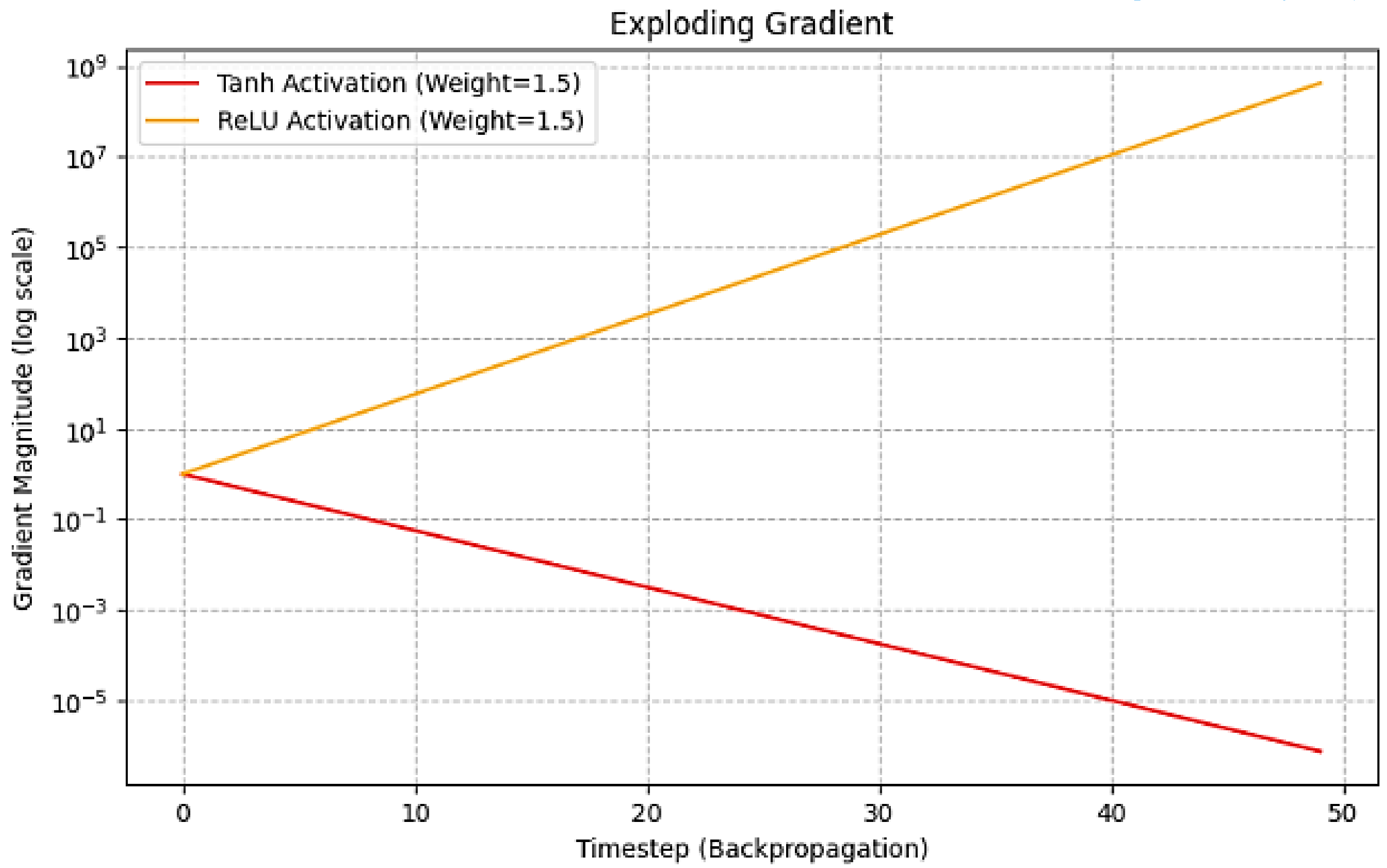
Long Sequences



Context gets lost

Gradient issues in RNNs





Vanishing Gradient

Exploding Gradient

Need for Long Memory

01 Many tasks depend on earlier parts of a sequence for correct interpretation.

02 Forget: They struggle to retain information from distant past inputs.

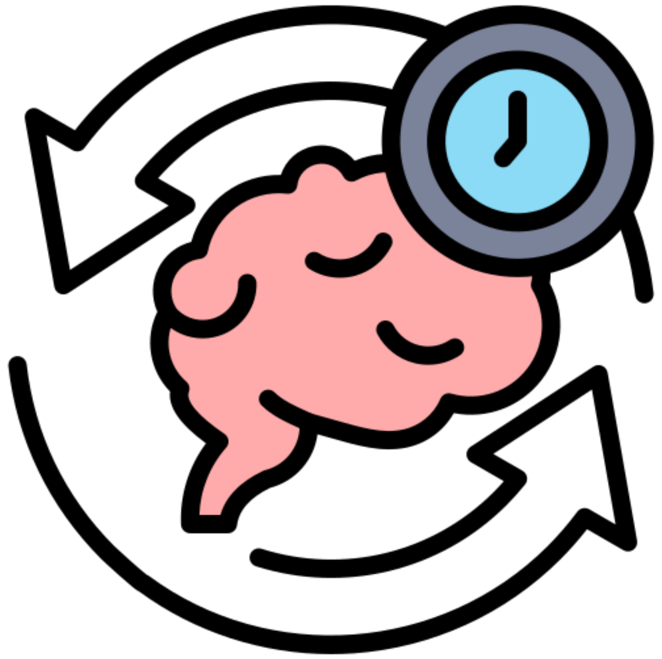
03 Some outputs rely on inputs far back in time (e.g., long sentences).

04 Learning long-term patterns is hard due to gradient decay.

05 These challenges led to memory-augmented models like LSTM.

Why was LSTM introduced?

Memory Cell

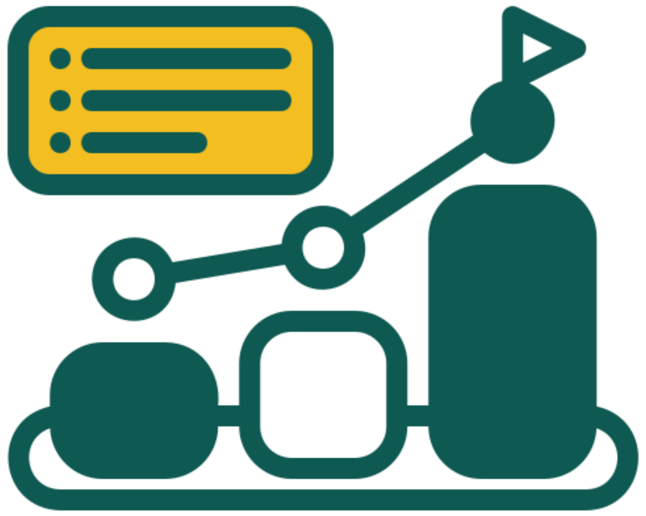


Stores long-term information

Gates Mechanism

Controls what to keep and discard

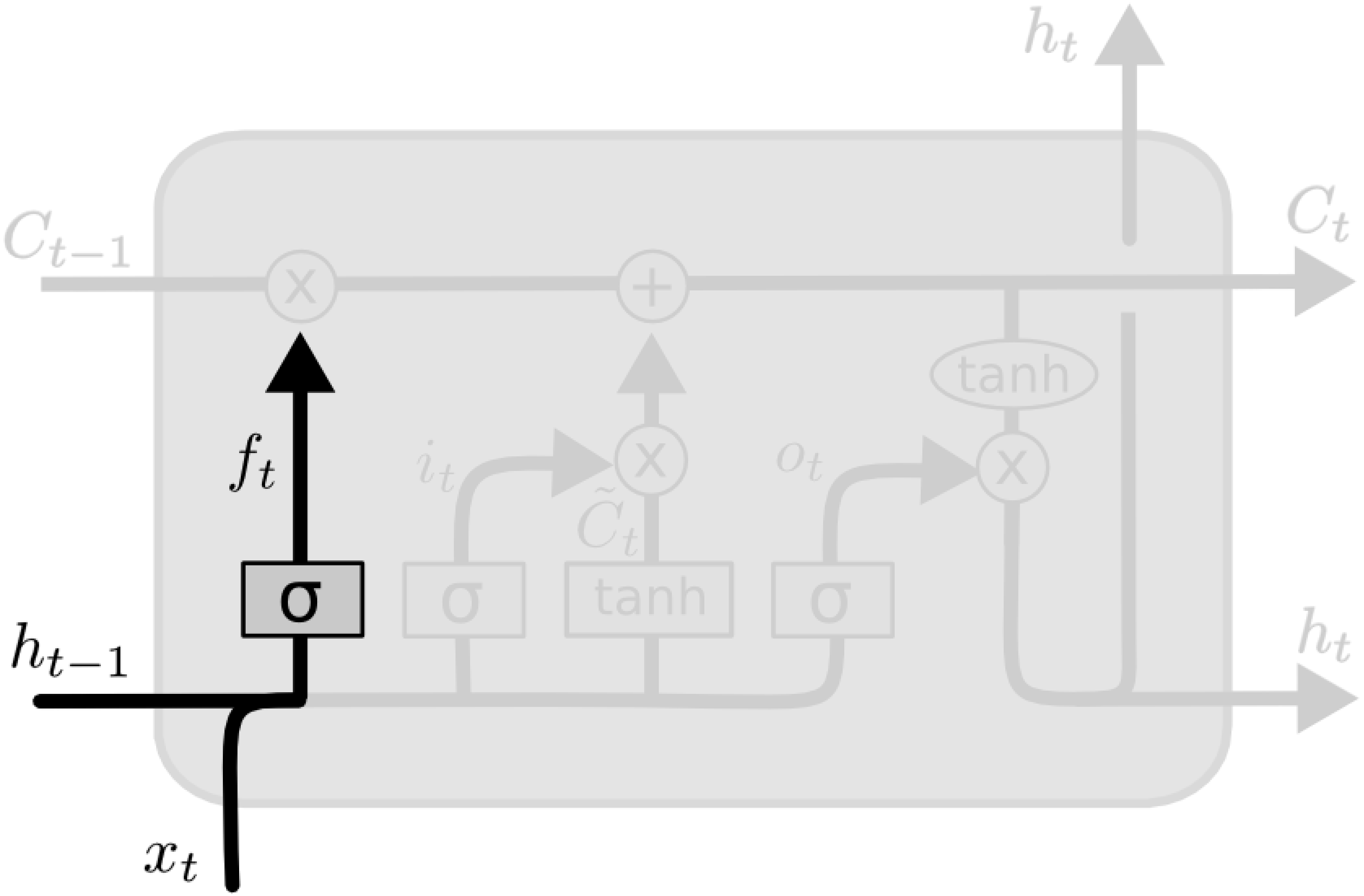
Solves Gradient Issues



Stable training across long steps

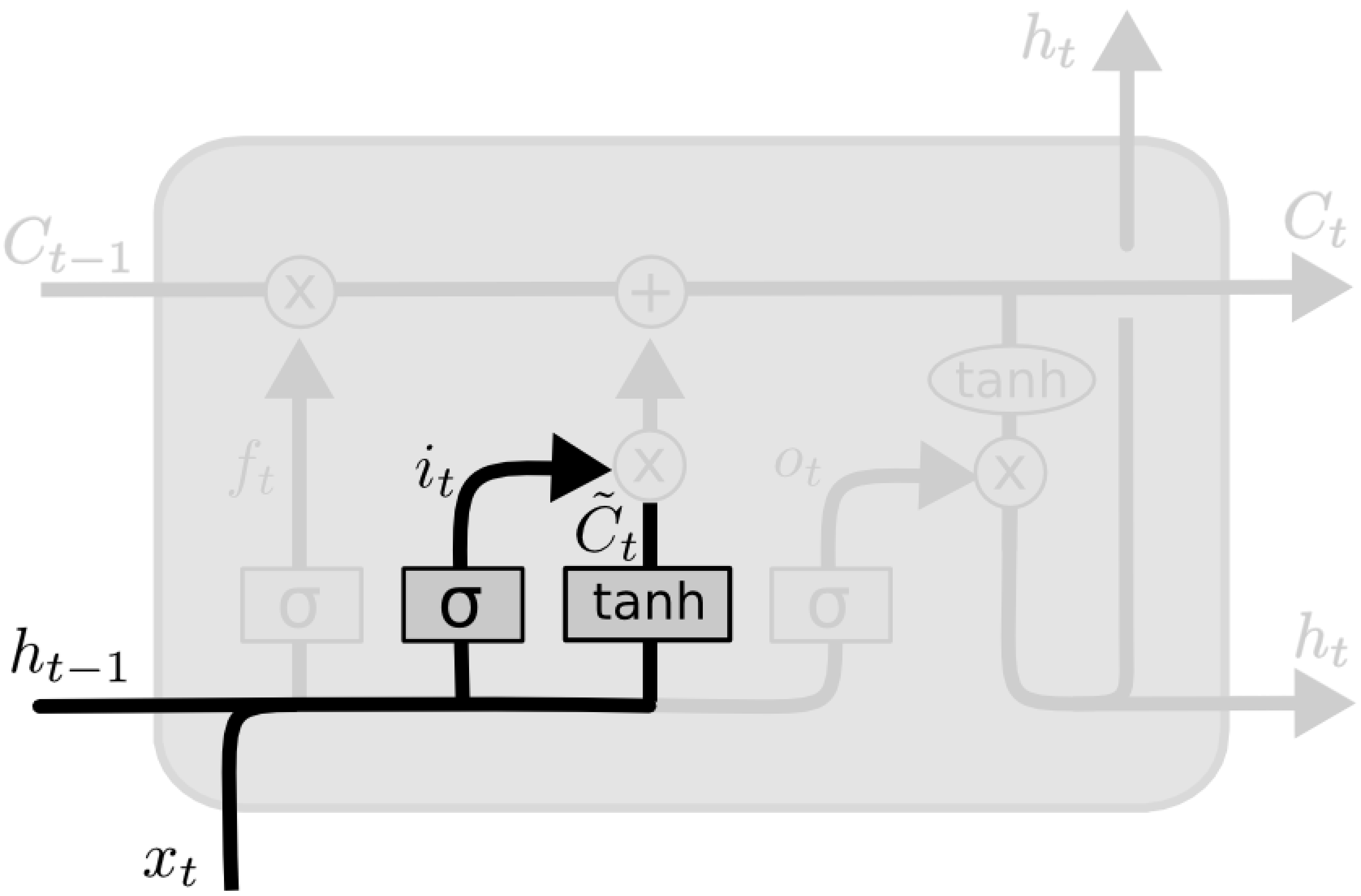
LSTM Cell

Forget Gate – Controlling Memory Retention



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

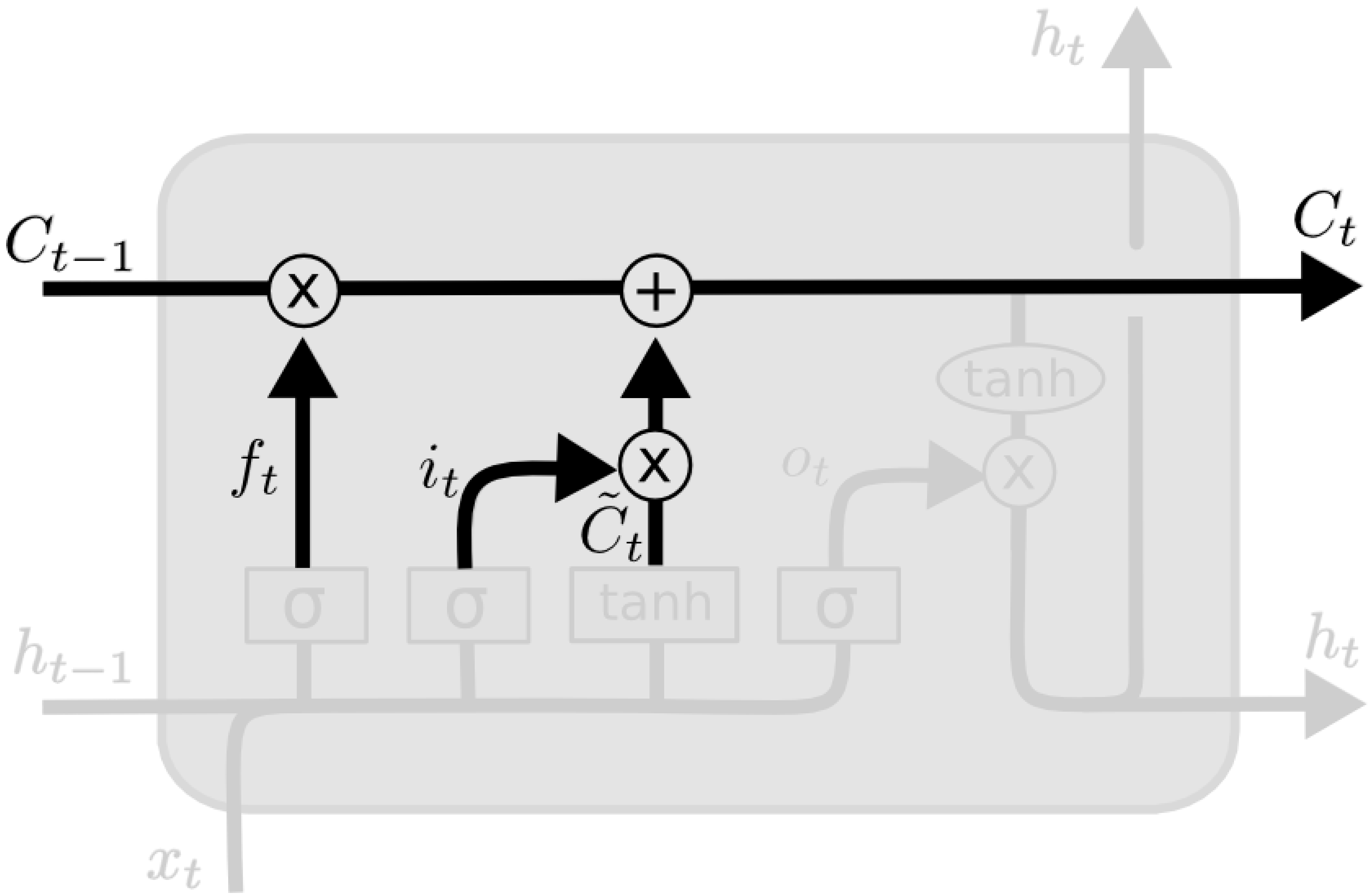
Input Gate – Deciding What to Store



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

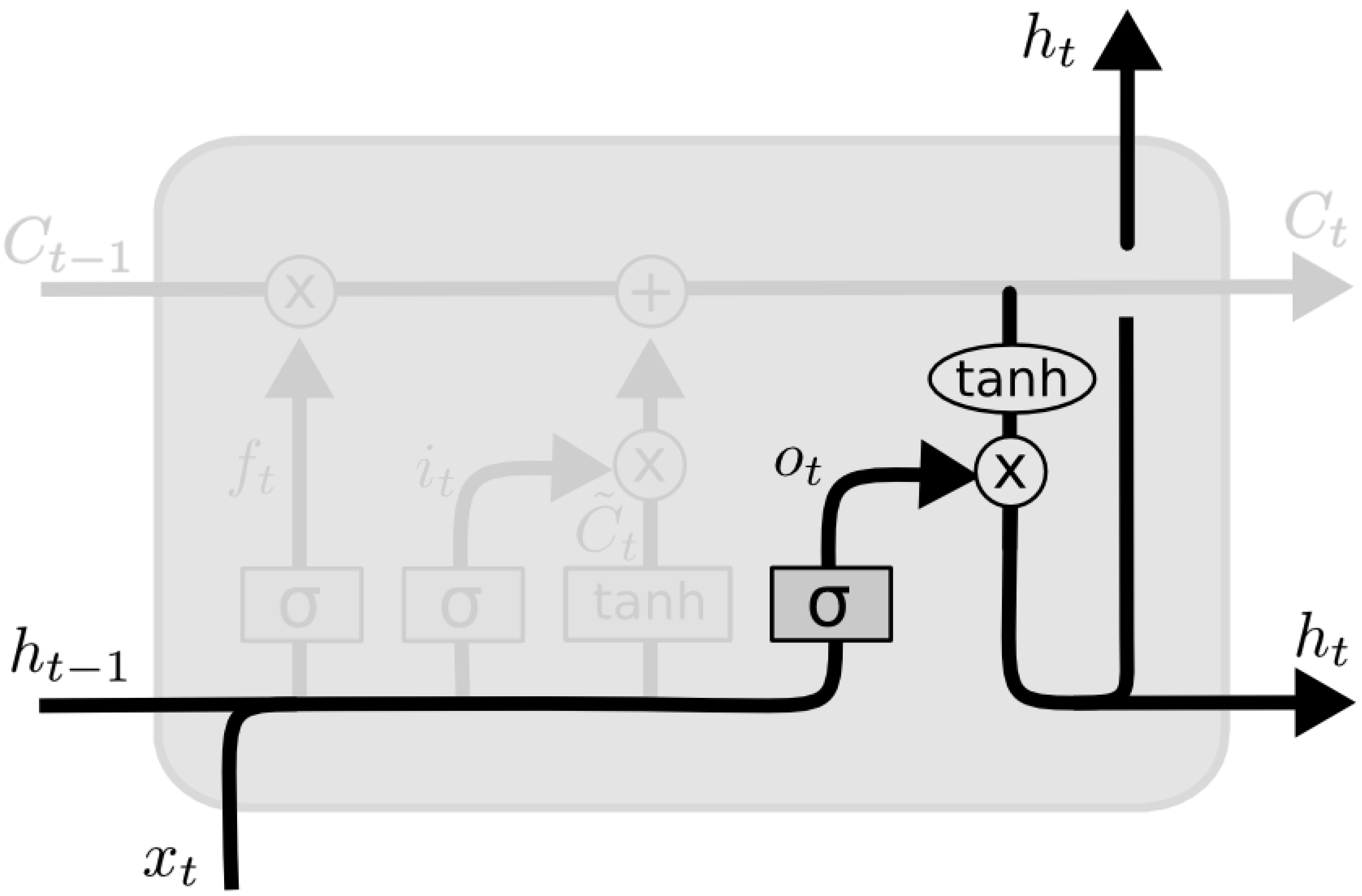
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Cell State Update – Memory Modification



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

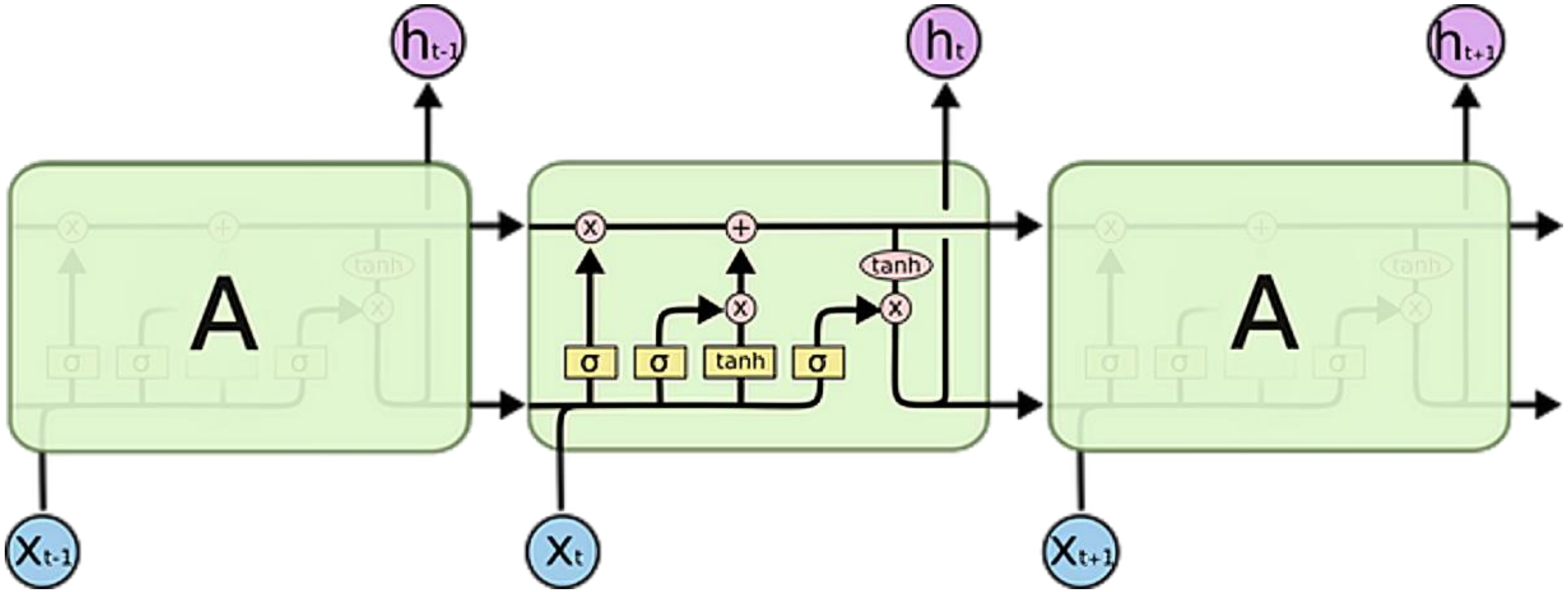
Output Gate – Producing the Hidden State



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

$$h_t = o_t * \tanh\left(C_t\right)$$

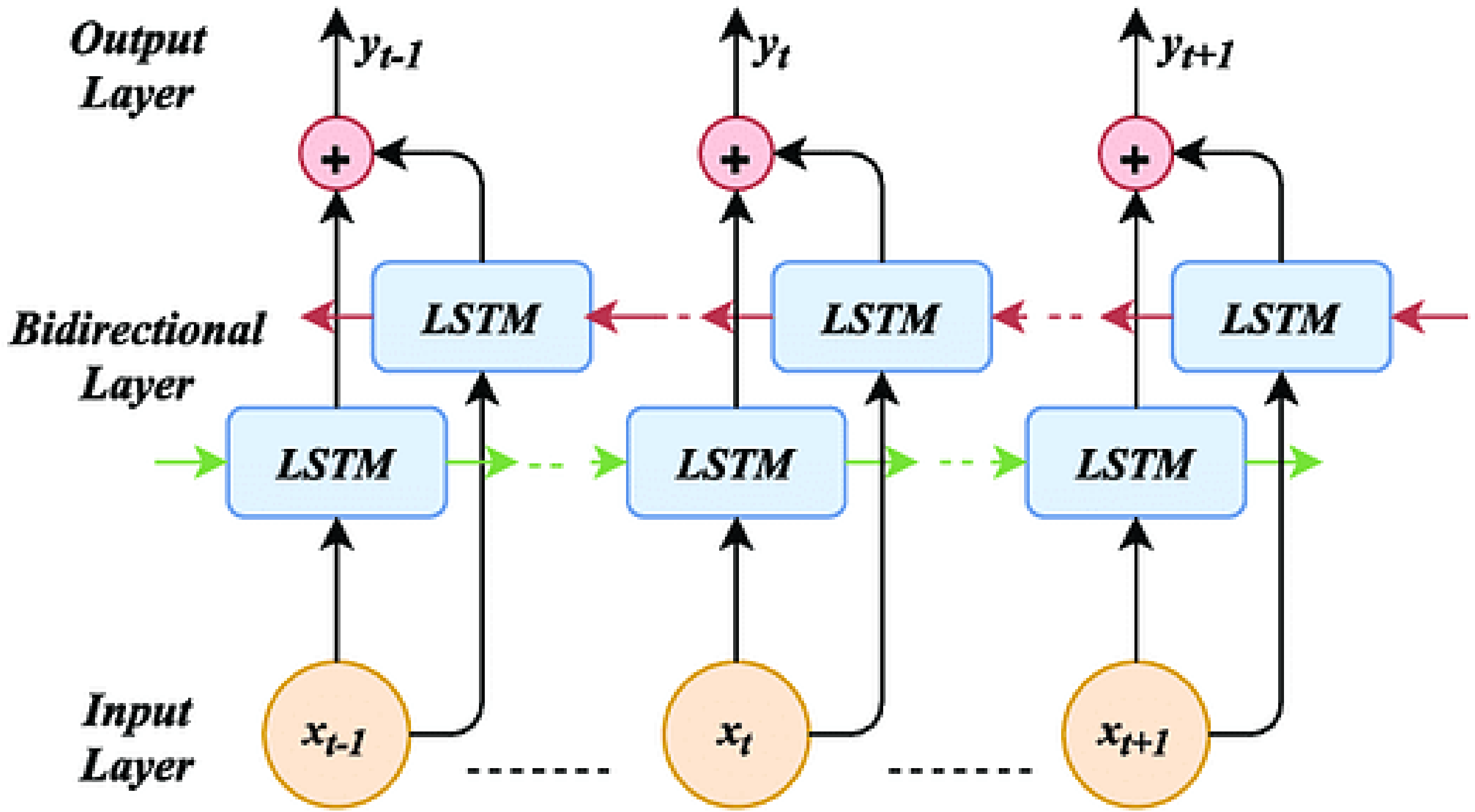
Full LSTM cell



LSTM Architectures

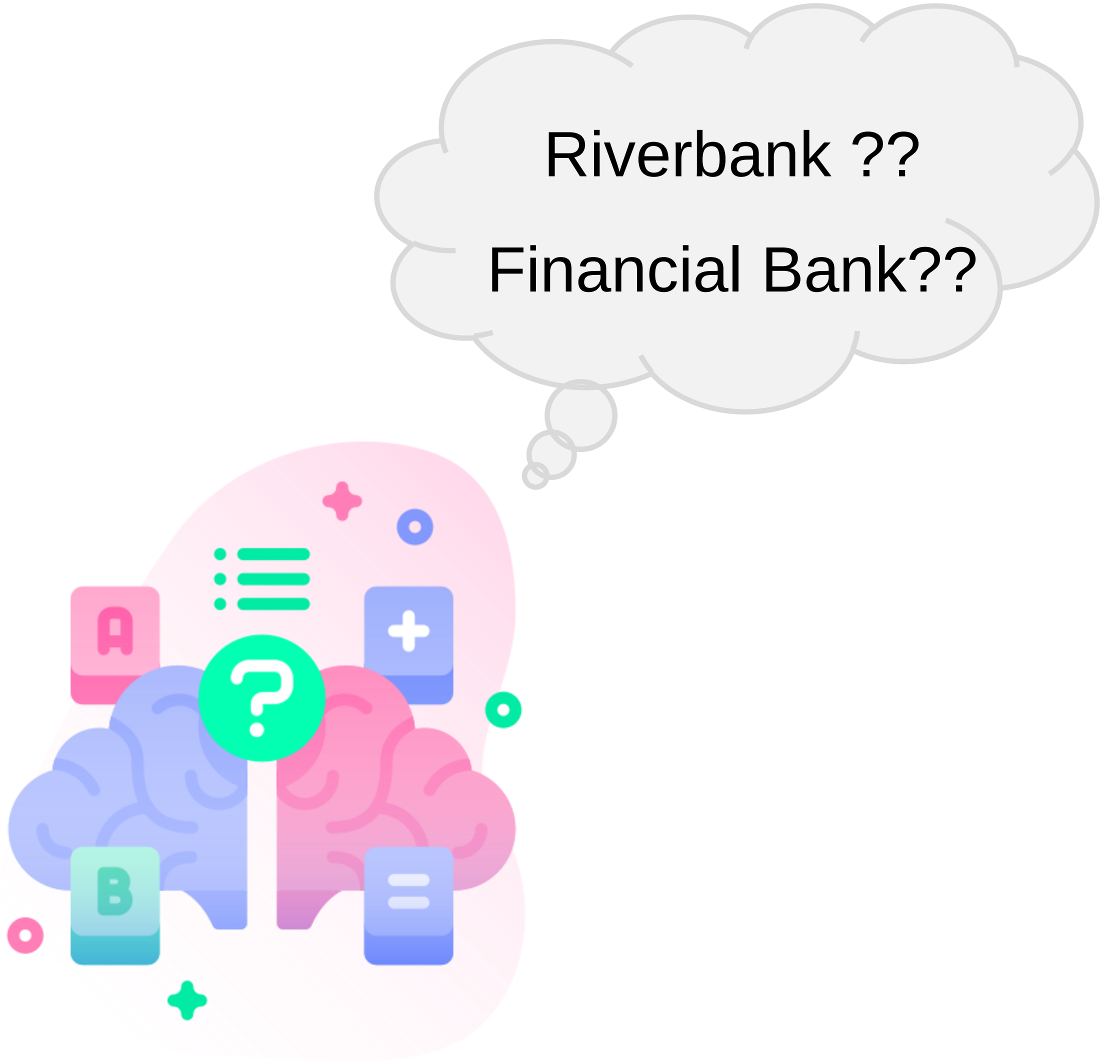
What is bidirectional LSTM?

Bidirectional LSTM (Bi LSTM/ BLSTM) is a variation of normal LSTM that processes sequential data in both forward and backward directions.



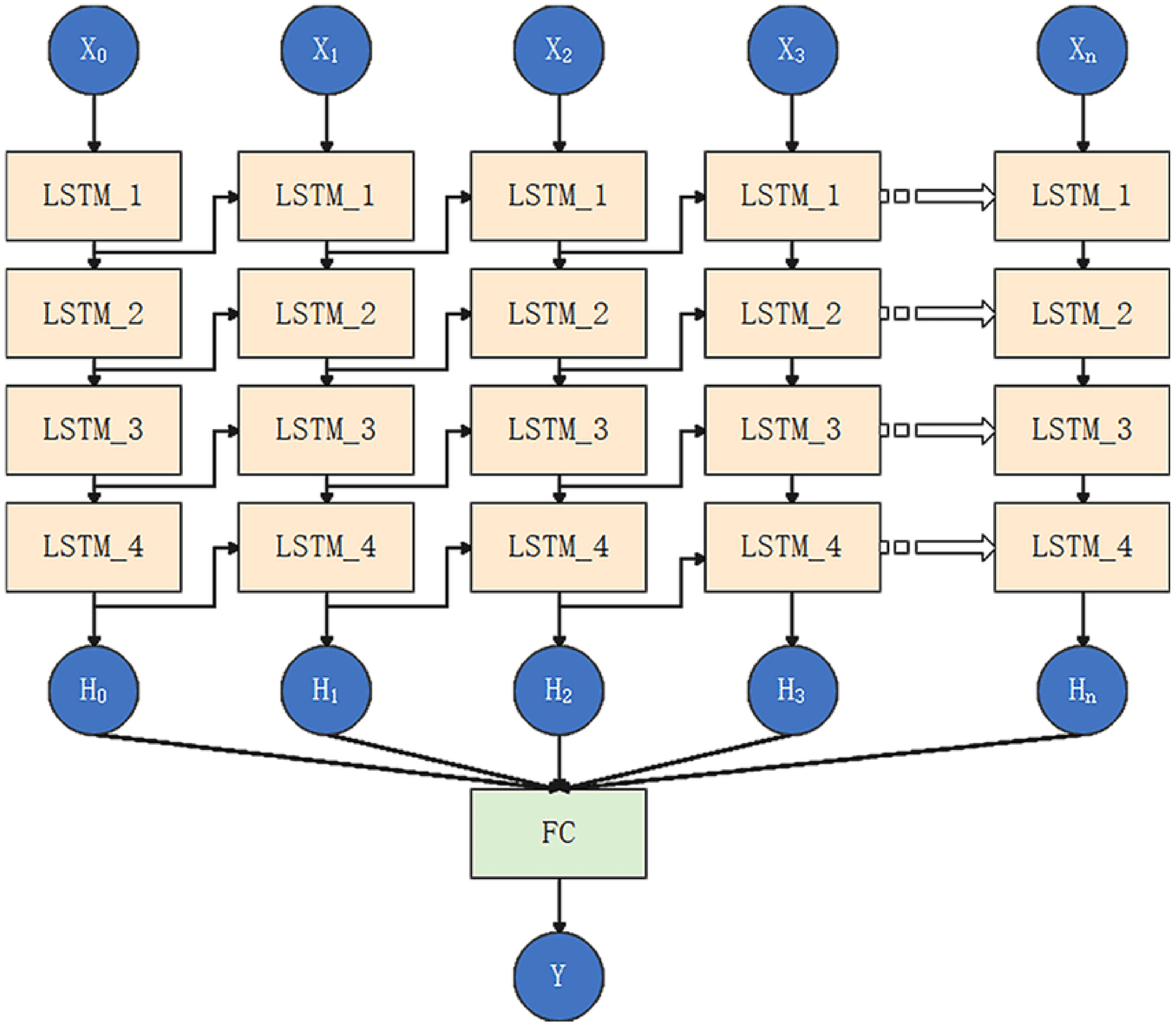
Why bidirectional helps?

“He went to the bank.”

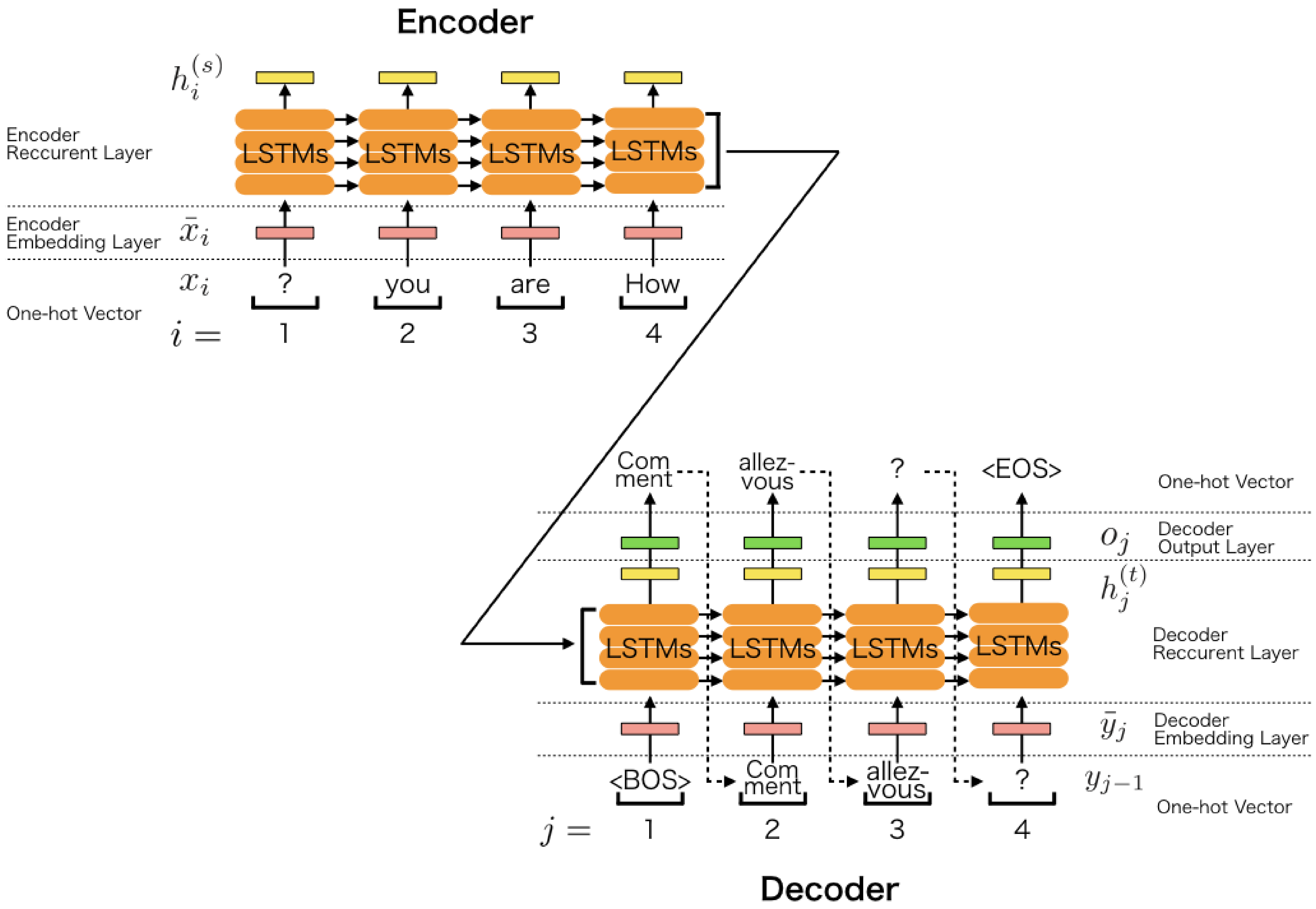


“to withdraw cash”

What is stacked LSTM?



What is sequence to sequence?



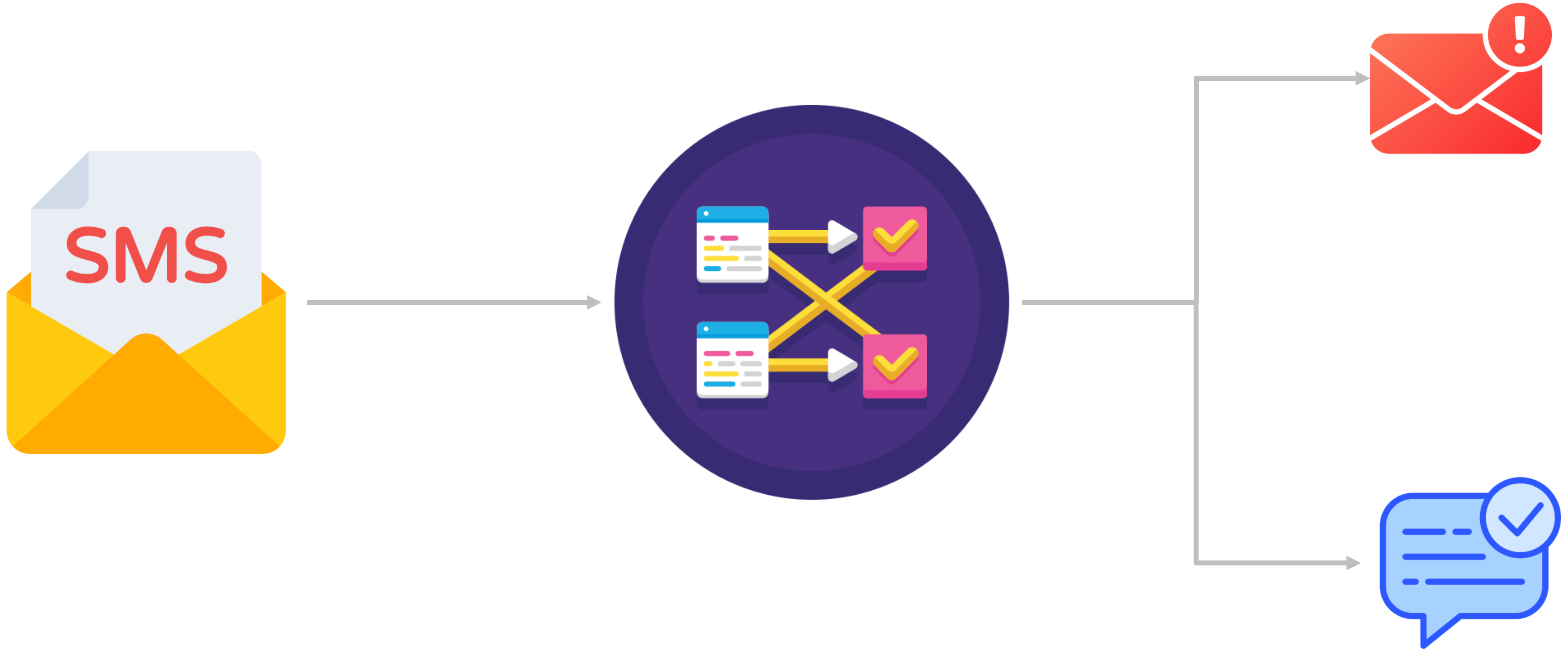
Choosing the Right LSTM

Task	Recommended LSTM
Text Classification	Basic / Stacked LSTM
Sequence Labeling	Bidirectional LSTM
Machine Translation	Seq2Seq with LSTM
Long-Term Dependencies	Stacked or Attention-based LSTM
Speech/Text Generation	Stacked Bidirectional LSTM

edureka! a Veranda ENTERPRISE

Real World Applications

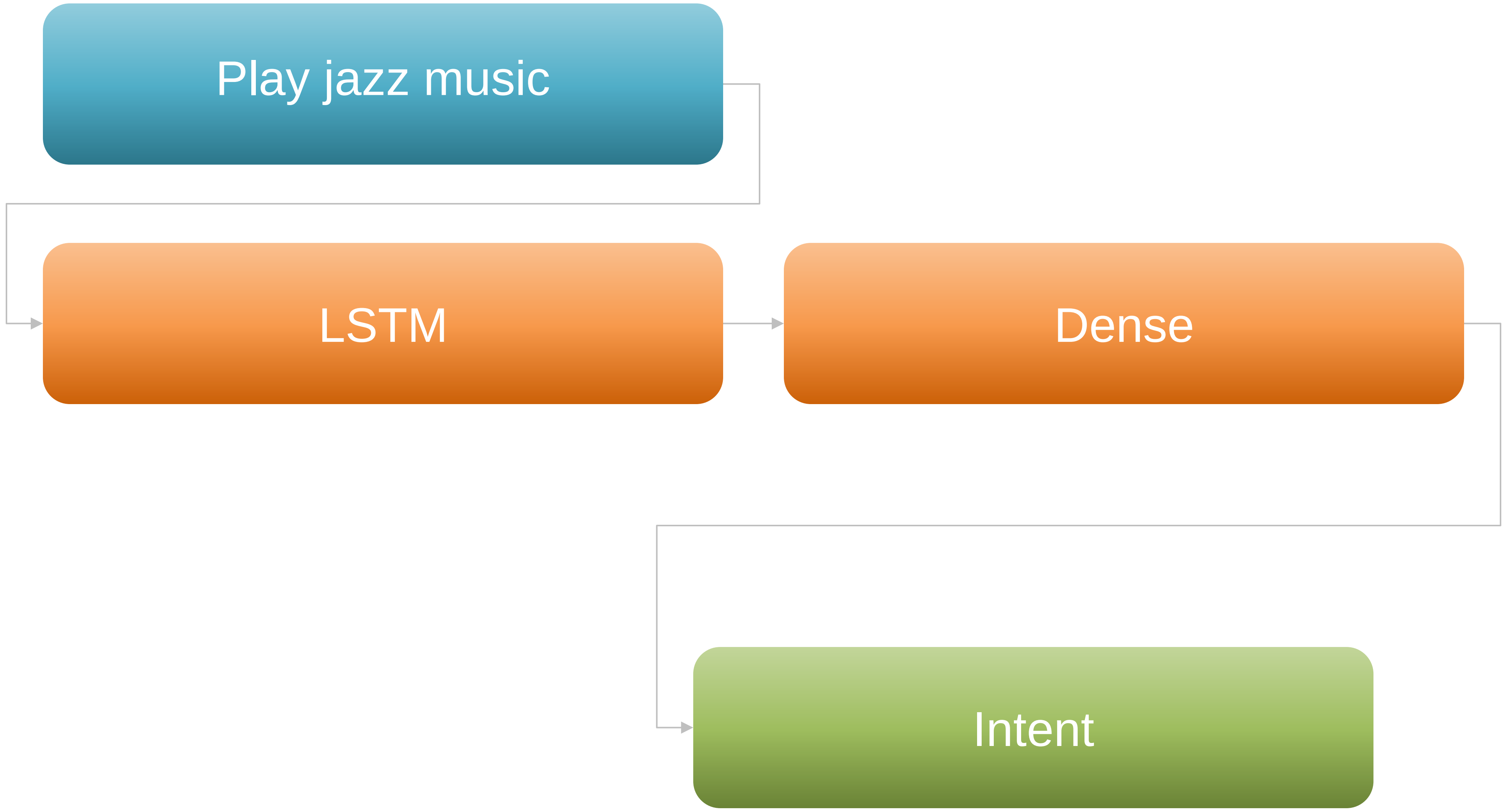
LSTM for Text Classification



Sentiment analysis pipeline

Text Preprocessing	Tokenization, stop word removal	1
Embedding Layer	Word2Vec, GloVe, or trainable embeddings	2
LSTM Layer	Learns long-range patterns	3
Dense + Softmax	Outputs sentiment class (Pos/Neg/Neutral)	4

Intent detection with LSTM



Play music

LSTM in speech tasks

Voice assistants

Emotion detection from tone

Automatic Speech Recognition

Tools using LSTM in real life

Google Assistant
Speech recognition

Apple Siri
Voice understanding

Amazon Alexa
Intent detection

Spotify
Mood-based playlist curation

Grammarly
Contextual spelling/ syntax suggestions

Chatbots
LSTM-based intent routing (e.g., Rasa NLU)

Adding Attention

Why does LSTM need attention?

1. LSTMs encode long sequences into a single fixed-size vector, causing information loss.
2. Attention lets the model focus on relevant parts of the input dynamically at each step.
3. It improves handling of long-range dependencies beyond LSTM’s memory capacity.
4. Attention provides interpretability by showing which inputs influence each output.

How to add attention?

Input

LSTM

(Return Sequence)

Attention

Dense Output

Model: "functional_2"

Layer (type)	Output Shape	Param #
input_layer_6 (InputLayer)	(None, 10, 64)	0
lstm_10 (LSTM)	(None, 10, 32)	12,416
attention (Attention)	(None, 32)	42
dense_6 (Dense)	(None, 1)	33

Total params: 12,491 (48.79 KB)
Trainable params: 12,491 (48.79 KB)
Non-trainable params: 0 (0.00 B)

Visual example of attention

"The cat sat on the mat"

When predicting the word "mat", the attention mechanism assigns weights to all words in the input to focus on "mat" related parts — maybe giving more weight to "sat" and "mat" itself.

Input Words	The	cat	sat	on	the	mat
Attention	0.05	0.10	0.30	0.10	0.05	0.40

Benefits of attention LSTM

01 Improves handling of long input sequences

02 Enhances focus on relevant input tokens

03 Boosts performance in NLP tasks

04 Reduces reliance on a fixed context vector

05 Enables dynamic context generation per step

06 Makes model predictions more interpretable

07 Supports visual analysis through attention maps

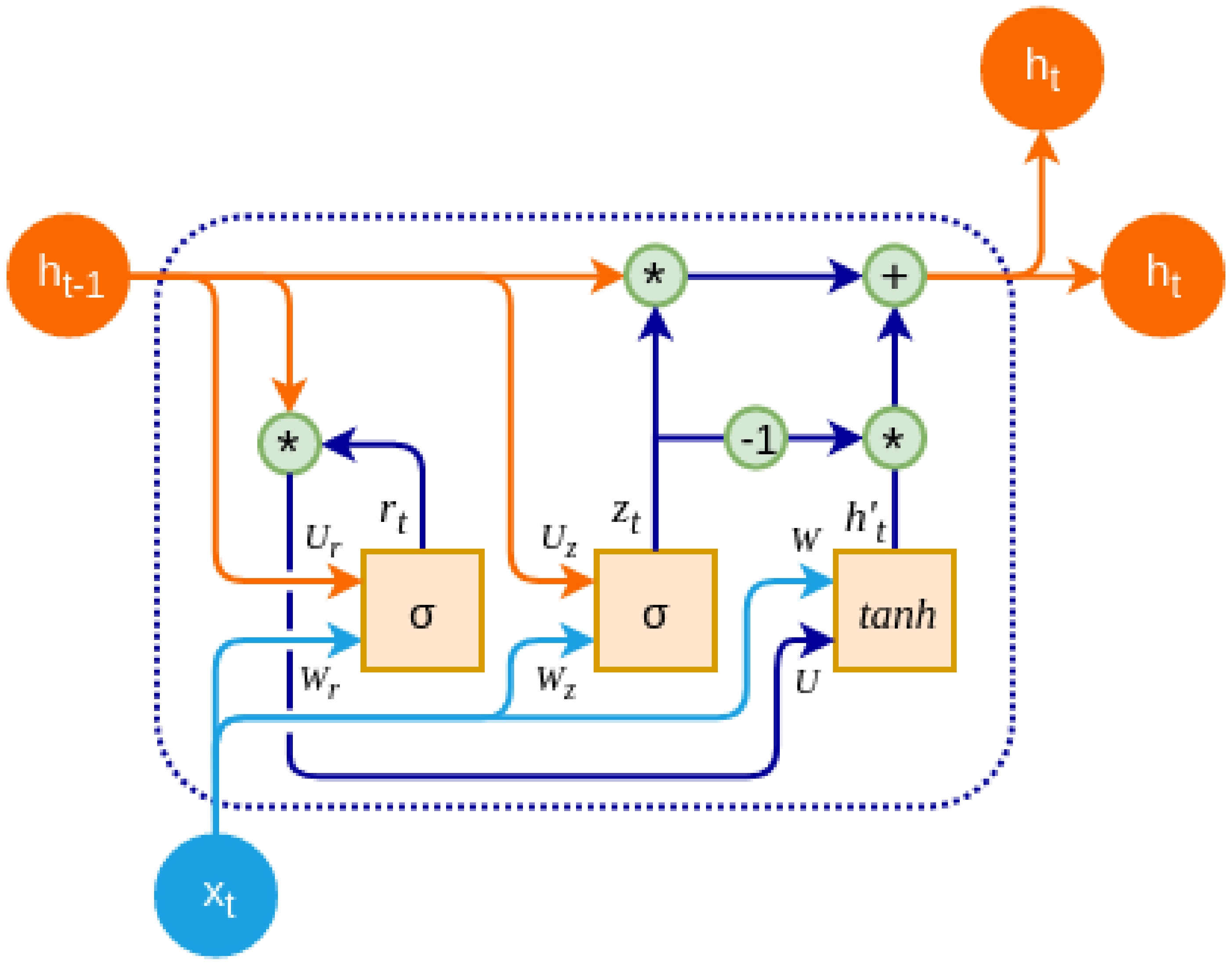
08 Forms the basis for advanced models like Transformers

Comparing LSTM and GRU

What is GRU?

GRU is a type of recurrent neural network like LSTM, but with a simpler architecture.

- Fewer gates: Uses only 2 gates (Update & Reset)
- No separate cell state: Hidden state alone carries memory
- Faster training: Fewer parameters than LSTM

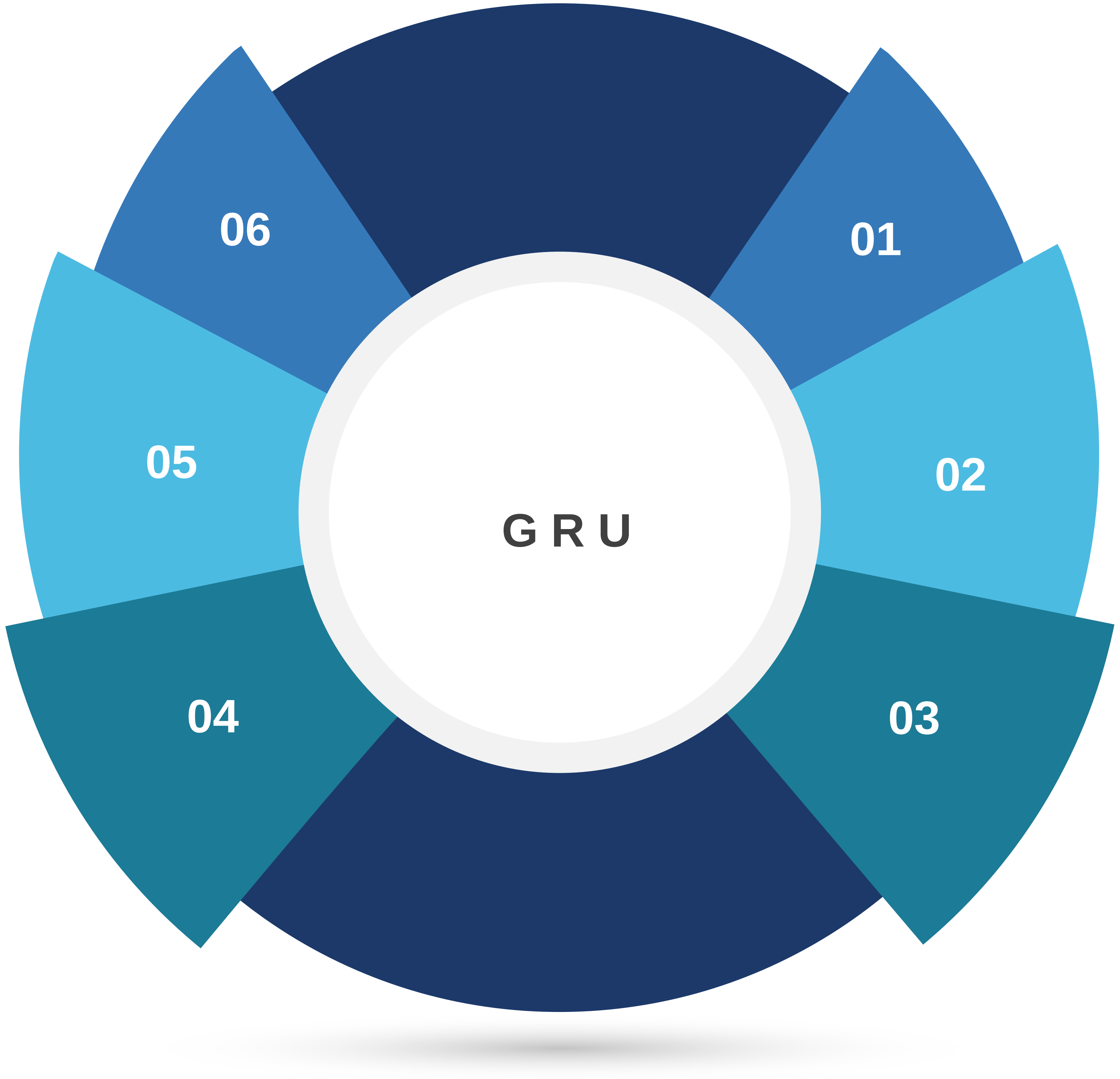


Working of GRU

Decides how much new info to keep.

Modified past state, processed with tanh.

Converts final hidden state to output (label, sequence, etc.).



Feeds sequence data (e.g., words, time steps) into the GRU.

Updates memory at each step using input + past state.

Controls how much past info to forget.

When to use LSTM or GRU?

LSTM	GRU
Better at modeling long-term dependencies	Faster to train with fewer parameters
Has more parameters (3 gates + separate cell state)	Simplified structure with 2 gates and no cell state
Works well on complex tasks like translation or speech	Performs well on smaller datasets or less complex sequences

LSTM Power Demand Forecasting (Demonstration)

Note: Refer to the Module 6: Demo 1 on LMS for detailed steps.

Training Bidirectional LSTM for Question Classification (Demonstration)

Note: Refer to the Module 6: Demo 2 on LMS for detailed steps.

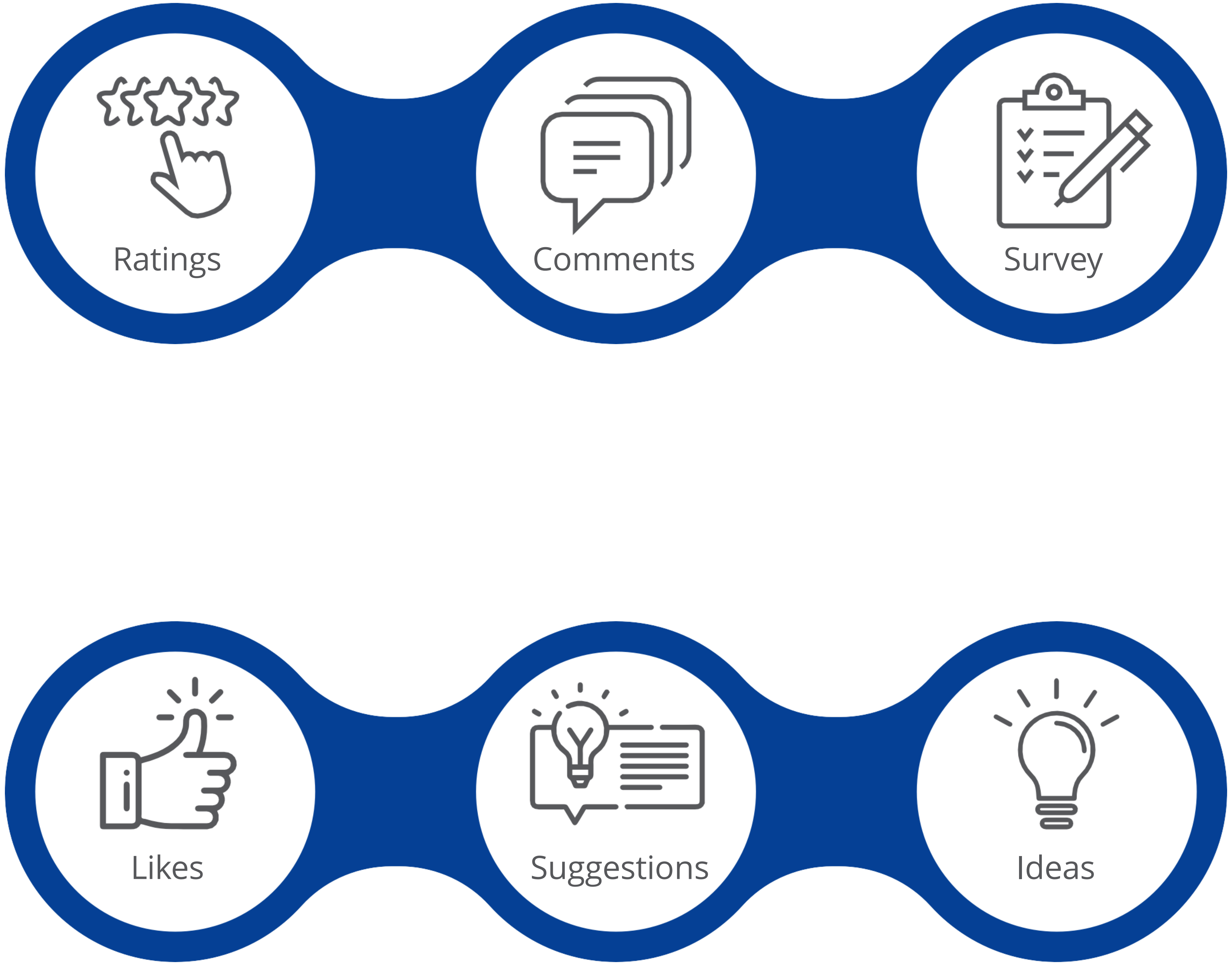
Summary

In this lesson, you have learned to:

- Identify the challenges of traditional RNNs and explain how LSTMs address them through specialized gate structures.
- Examine key LSTM variants, including Bidirectional and Stacked models, and determine their practical use cases.
- Implement LSTM-based solutions for real-world tasks such as sentiment analysis, text classification, and intent detection.
- Evaluate the benefits of adding attention to LSTMs and distinguish between LSTM and GRU architectures for informed model selection.

Questions

Feedback



Thank You

For information, Please Visit our Website
www.edureka.co