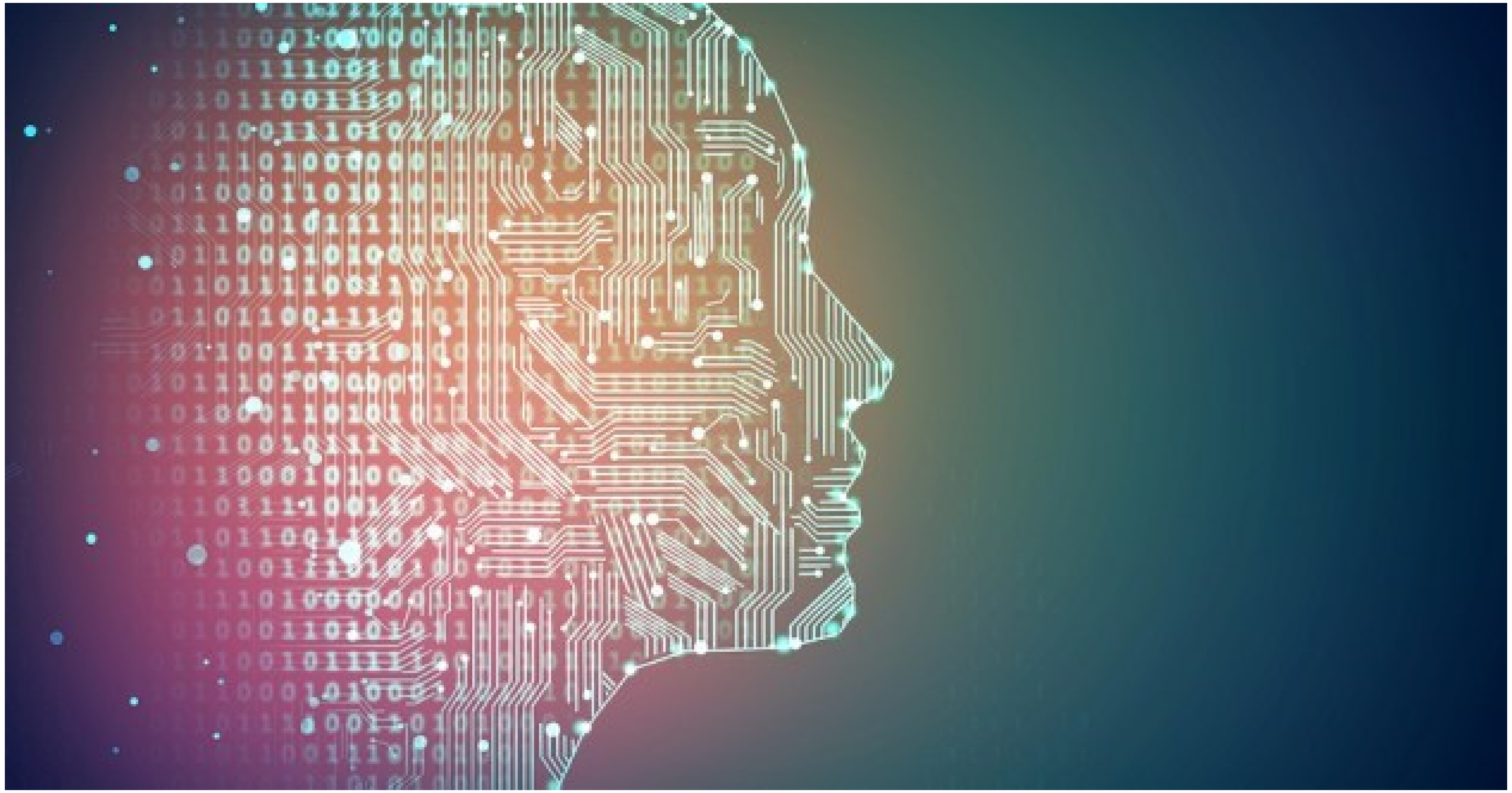
AI Deep Learning: Recurrent Neural Networks
Long Short-Term Memory (LSTM)

Thuan L Nguyen, PhD

Slide 2: Long Short-Term Memory Neural Networks (LSTM)



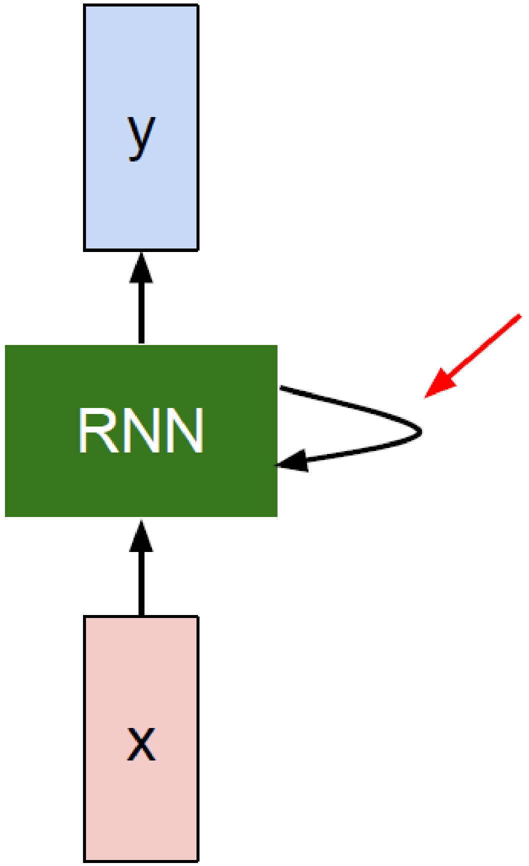
AI Deep learning (Source: mindovermachines.com)

Slide 3: Long Short-Term Memory Neural Networks (LSTM)

1. LSTM Neural Networks: Simple RNN with Short-Term Memory
2. LSTM Neural Networks : Introduction
3. LSTM Neural Networks : Core Concepts: Cell State & Gates: Overview
4. LSTM Neural Networks: Core Concepts: Cell State & Gates: Cell State
5. LSTM Neural Networks: Core Concepts: Cell State & Gates: Gates
6. LSTM Neural Networks: Flow of Information & Mathematical Models
7. LSTM Neural Networks: HOWTO Solve Vanishing Gradient Problem

Slide 4: AI Deep Learning: Recurrent Neural Networks (RNN)

Recurrent Neural Networks: Fundamentals



Key idea: RNNs have an "internal state" that is updated as a sequence is processed

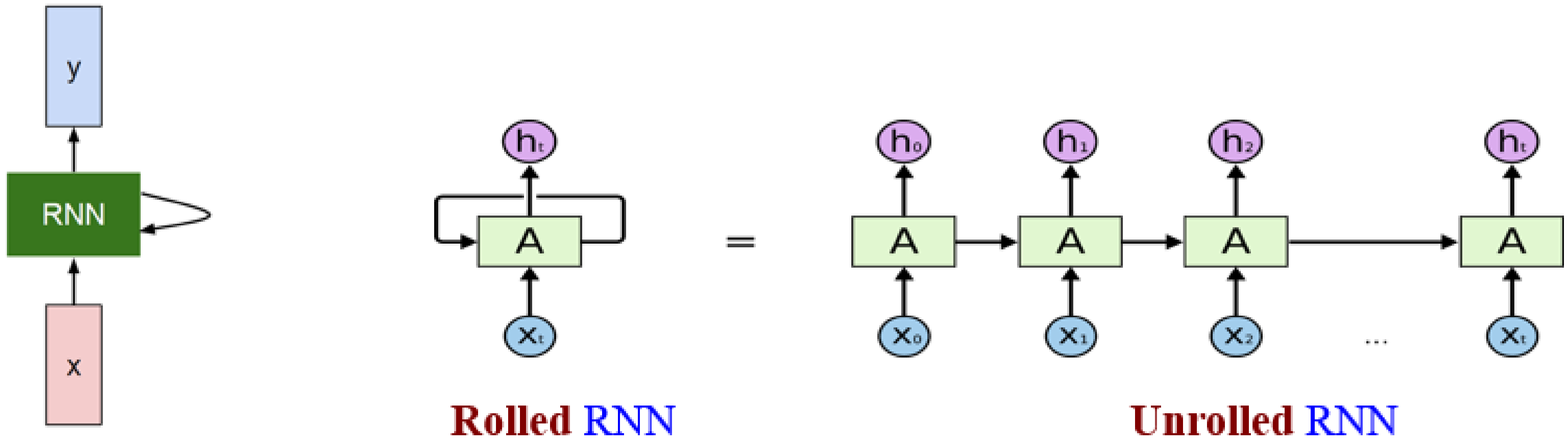
Key words:

- An **internal** (or "**hidden**") **state** is **updated**
- When a **sequence** is **processed**

Recurrent Neural Network (Source: Stanford.edu)

Slide 5: AI Deep Learning: Recurrent Neural Networks (RNN)

RNN: Recurrent Neural Network: Rolled & Unrolled



Rolled and Unrolled RNN (Source: Stanford.edu and Colah Blogs)

The fundamental feature of a Recurrent Neural Network (RNN):

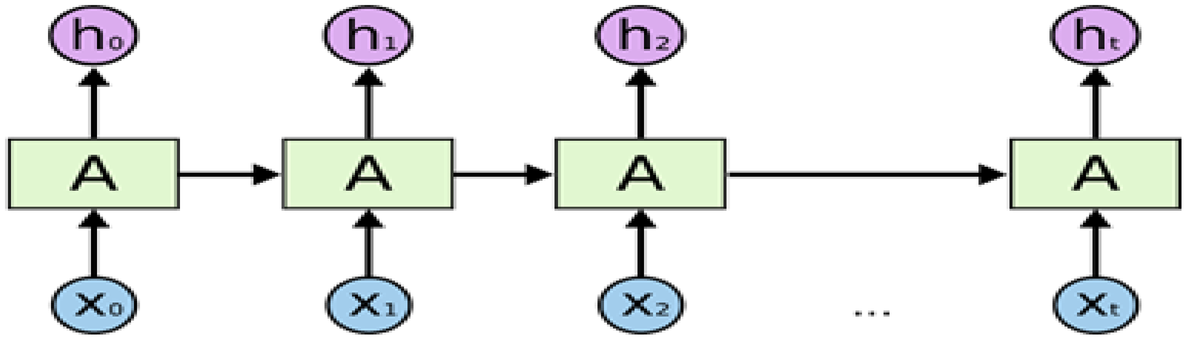
- The **input sequence x** can be processed by applying a **recurrent formula** at each step.
- In other words, the same function and the same set of parameters can be used at each step of processing the input sequence.

Slide 6: Long Short-Term Memory Neural Networks (LSTM)

RNN: Problem of Short-Term Memory

Recurrent Neural Networks suffer from short-term memory.

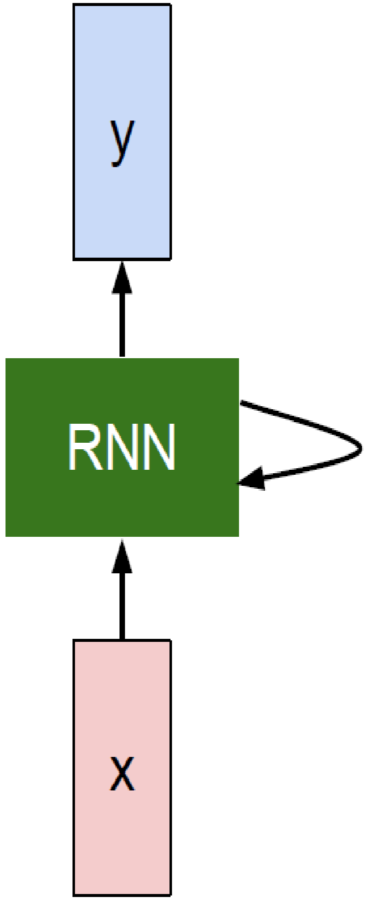
- If a sequence is long, it's hard for RNN's to "memorize" the information at earlier time steps.
- For example:
  - Process a long sentence of text to do predictions, **RNN's** may "**forget**" words from the beginning.



Unrolled RNN (Source: Colah Blogs)

Slide 7: Long Short-Term Memory Neural Networks (LSTM)

RNN: Problem of Short-Term Memory



Recurrent Neural Network
(Source: Stanford.edu)

During back propagation, recurrent neural networks suffer from the vanishing gradient problem.

- Gradients are values used to update **weights** that are applied to a neural network.
- The vanishing gradient problem:
  - When the gradient shrinks as it back propagates **through time.**
    - If a gradient value becomes extremely small, it stops contributing to the network's learning.
    - In other words, the neural network **stops** learning.

Slide 8: Long Short-Term Memory Neural Networks (LSTM)

RNN: Problem of Short-Term Memory or Vanishing Gradient

Updated Weight = Weight – Learning Rate * Gradient

1.0999999 = **1.1** -- 0.01 * 0.00001

Weight is barely updated
→ **Stop** Learning

Extremely **small**

During back propagation, recurrent neural networks suffer from the vanishing gradient problem.

- **Layers** that get a **very small** gradient update **stops** learning.
  - Those are usually the earlier layers → these layers **don't learn**.
  - RNN's can **forget** what it has seen in the earlier time steps of long sequences, thus having a **short-term memory** due to the problem of **vanishing gradient**.

Slide 9: Long Short-Term Memory Neural Networks (LSTM)

RNN: Problem of Short-Term Memory

For example:

- In a sentence analysis problem:
  - Use a simple RNN to process a sentence and classify the intention from the inputs.

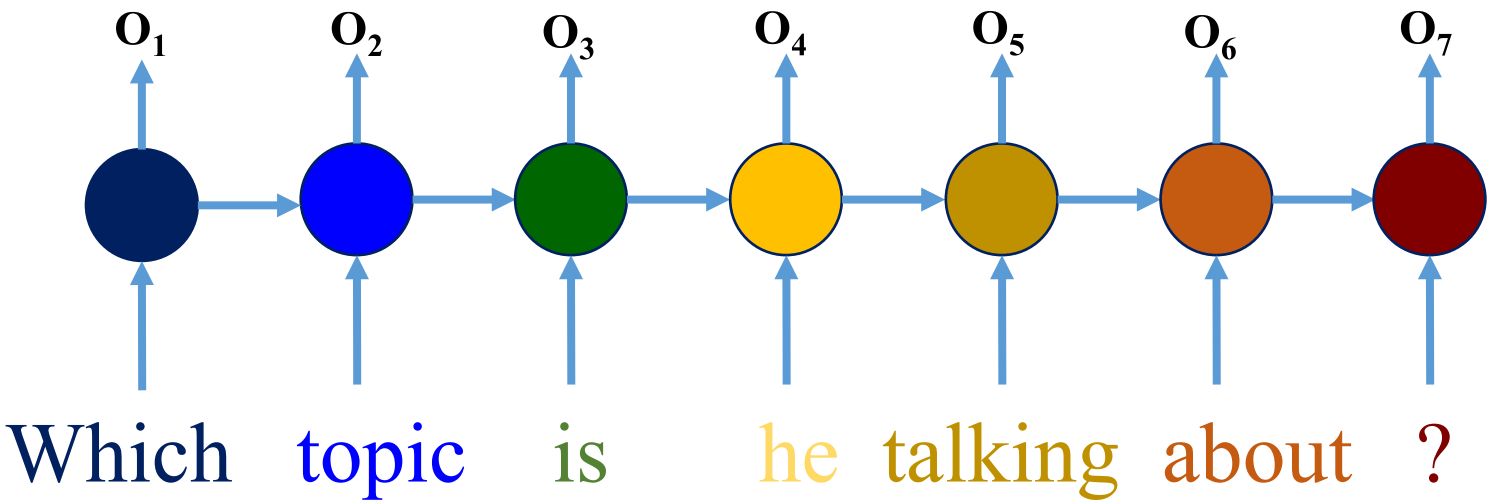
Which topic is he talking about?

Slide 10: Long Short-Term Memory Neural Networks (LSTM)

RNN: Problem of Short-Term Memory

For example (Cont.):

- **First**, to feed "**Which**" into the RNN.
  - The RNN encodes "Which" and produces an output.
- **Next**, to feed "**topic**" and the **hidden state** from the first step into the RNN.
  - The RNN has both the words "Which" and "topic."
- And **so on** …
- **Finally**, the **last output O_7** is fed into a feed-forward neural network to classify an intent.



Unrolled RNN and language as sequence data

Slide 11: Long Short-Term Memory Neural Networks (LSTM)

RNN: Problem of Short-Term Memory

For example (Cont.):

- What happens while training an RNN? → **Three major steps**:
  - **First**, it does a forward pass and makes a prediction.
  - **Second**, it compares the prediction to the real value using a loss function.
    - The loss function outputs an error value that estimates the network performance.
  - **Last**, it uses the generated error value to perform **back propagation**
    - → Calculate the gradients for each node in the network.

- **Gradient** and the **neural network's learning**
  - The **gradient** is the value used to adjust the network internal weights.
    - The **adjustment** of the gradient **enables** the network **to learn**.
      - Larger gradient → Bigger adjustments → Network **learns more**
      - And vice versa.

Slide 12: Long Short-Term Memory Neural Networks (LSTM)

RNN: Problem of Short-Term Memory

For example (Cont.):

- **The problem: Vanishing Gradient**
  - While back propagation is done:
    - Each node in a layer calculates its gradient with respect to the effects of the gradients in the layer before it (before: in the backward direction).
    - If the adjustments to the nodes of the previous layers is small
      - → Adjustments to the nodes of a layer are **even smaller**.

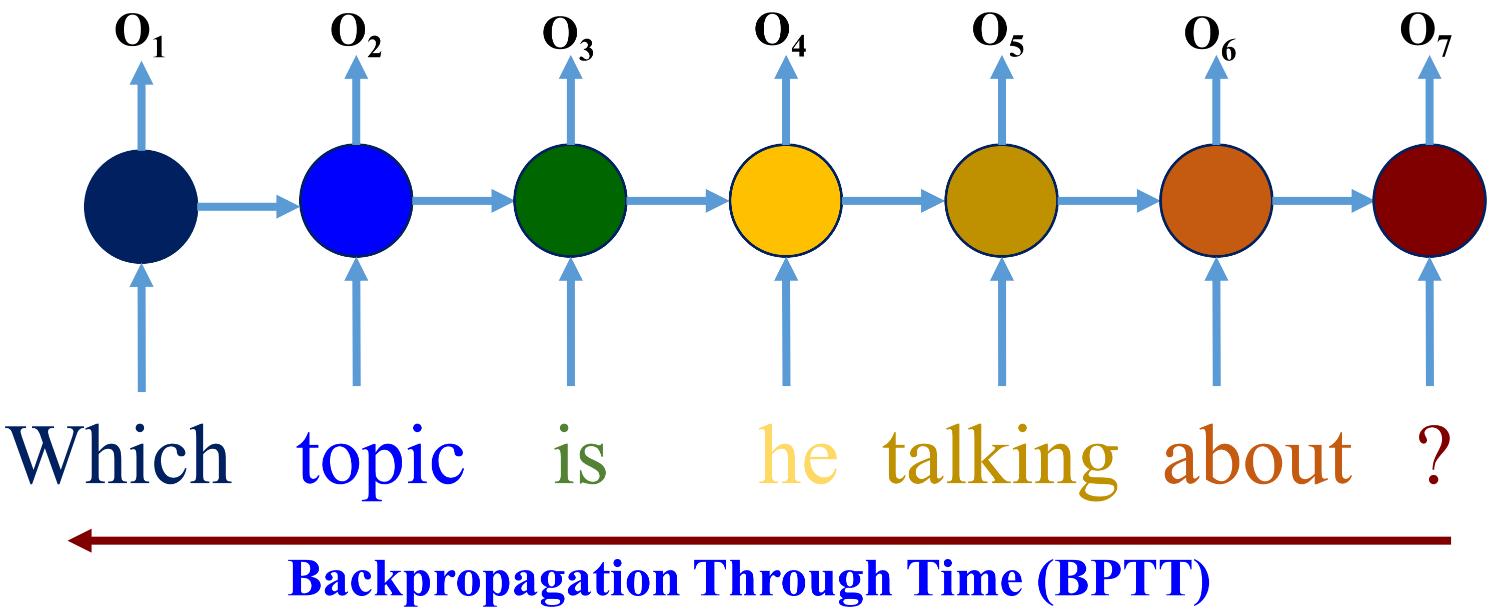
  - **As a result**:
    - Gradients are **exponentially shrunk** along the path of **backpropagation**.
    - The **earlier** layers (the **later** ones on the path of backpropagation)
      - Internal weights are barely adjusted because of extremely small gradients.
      - Hardly get any learning
    - The **learning stops**.

Slide 13: Long Short-Term Memory Neural Networks (LSTM)

RNN: Problem of Short-Term Memory

For example (Cont.):

- **Vanishing Gradient** and **RNN**
  - Each **time step** in an **unrolled** recurrent neural network is considered as a **layer**.
  - To train an RNN, back-propagation through time (a variance of back-propagation) is used.
    - The **gradients** are exponentially shrunk as it **back-propagates** through each time step.

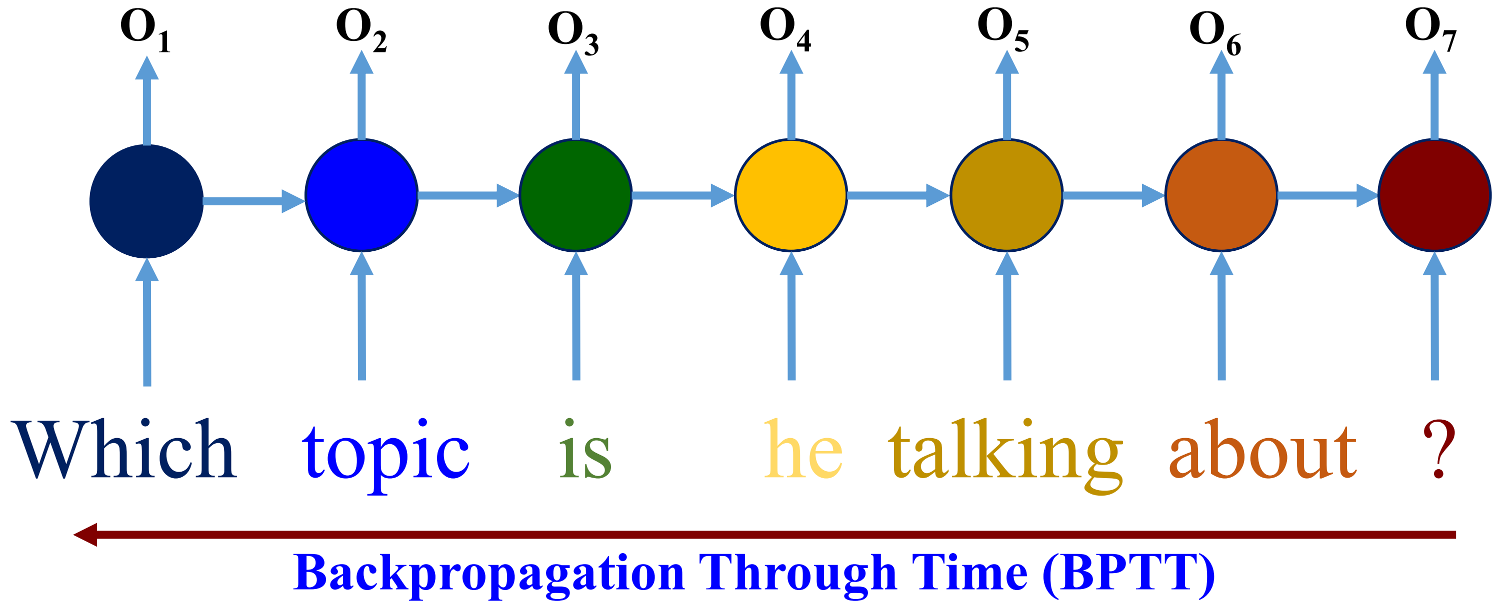


Slide 14: Long Short-Term Memory Neural Networks (LSTM)

RNN: Problem of Short-Term Memory

For example (Cont.):

- **Vanishing Gradient** and **RNN**
  - **RNN** likely "**forget**" the **first** words, like "Which" and "topic", and does **not learn** about them.

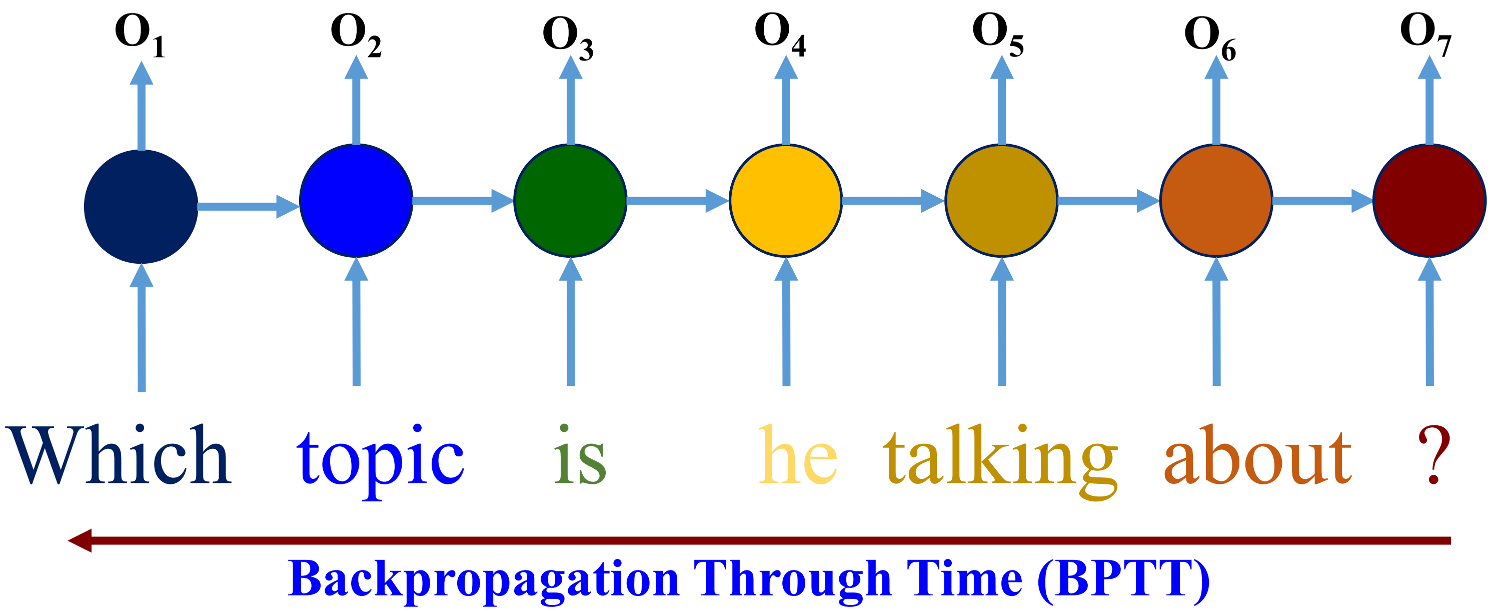


Slide 15: Long Short-Term Memory Neural Networks (LSTM)

RNN: Problem of Short-Term Memory

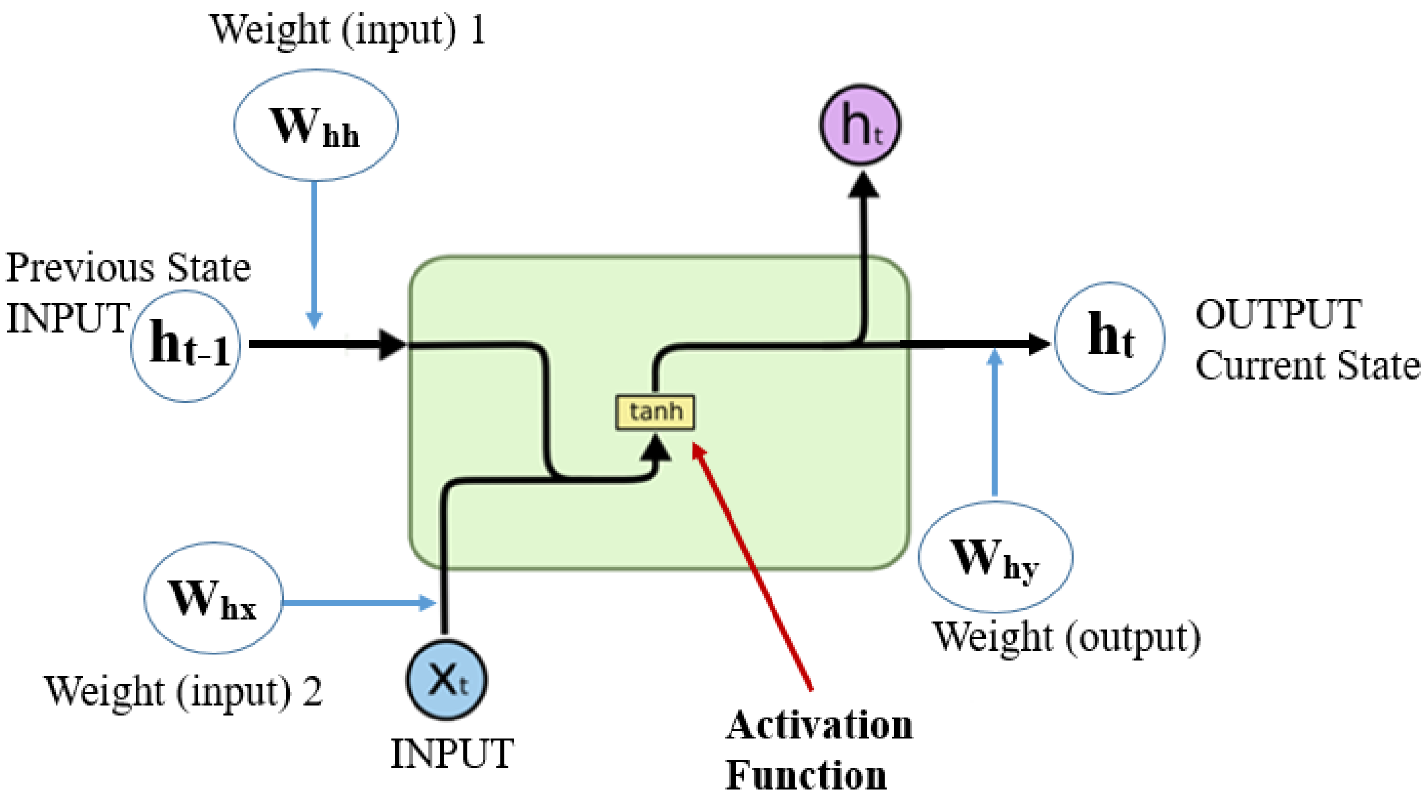
<u>**For example**</u> (Cont.):

- **Vanishing Gradient** and **RNN**
    - **RNN** suffers from the **problem** of **short-term memory**.



Slide 16: Long Short-Term Memory Neural Networks (LSTM)

RNN: Simple Recurrent Neural Network: Anatomy of Simple RNN Cell

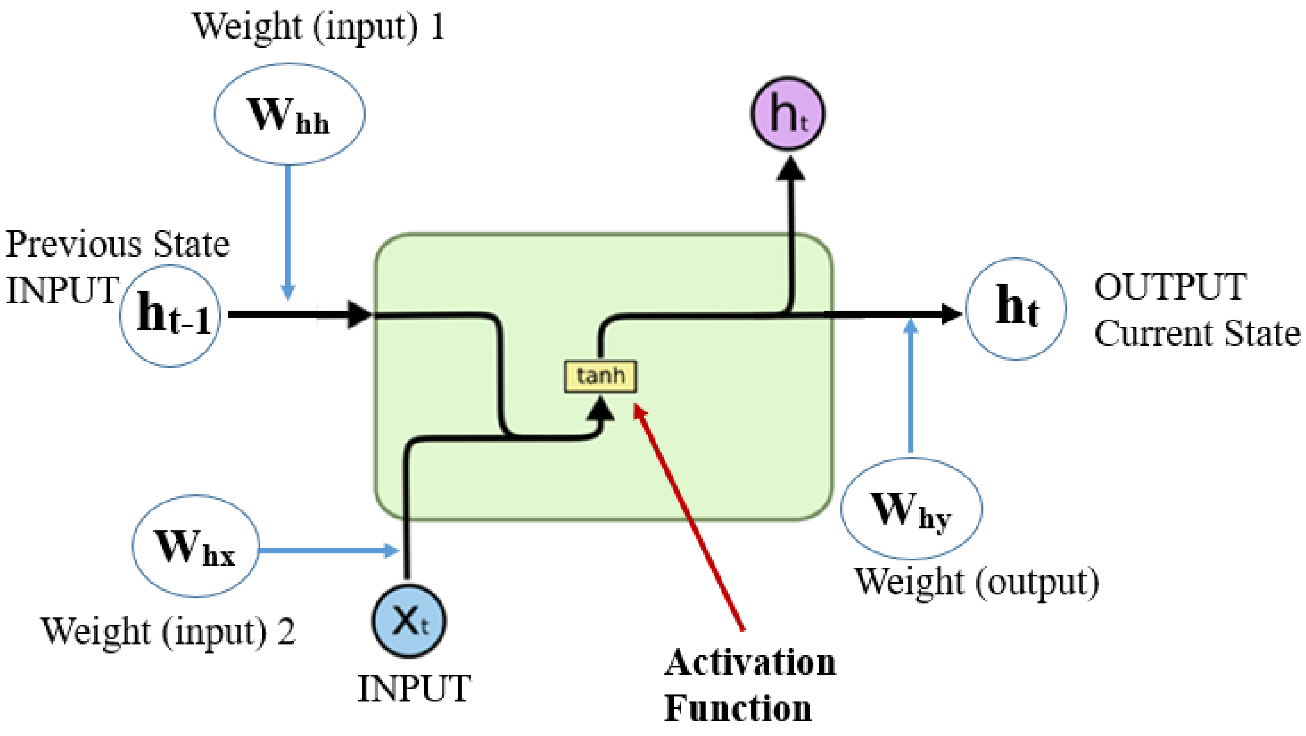


Simple RNN Cell)

Where is the **memory**?

Slide 17: Long Short-Term Memory Neural Networks (LSTM)

RNN: Simple Recurrent Neural Network: Anatomy of Simple RNN Cell



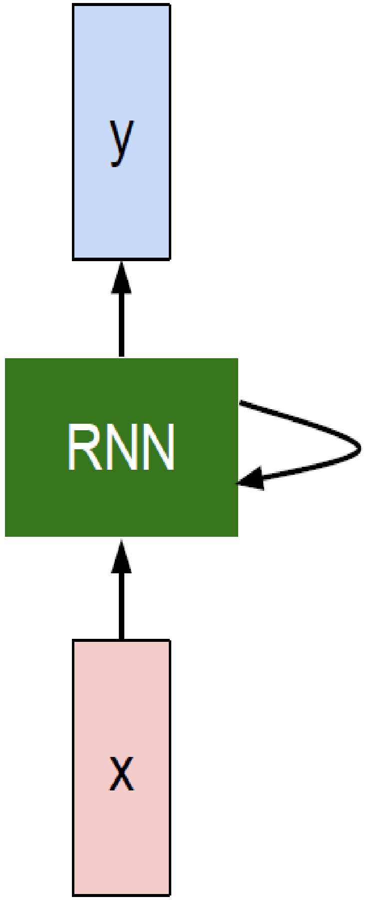
Simple RNN Cell)

Where is the **memory**?

- In recurrent neural networks, the **hidden state h** represents the **memory** of the network**.**
- The **hidden state h** is the **indicator** of the **short-term** memory, a.k.a. the **working** memory, of the neural network**.**

Slide 18: Long Short-Term Memory Neural Networks (LSTM)

RNN: Problem of Short-Term Memory

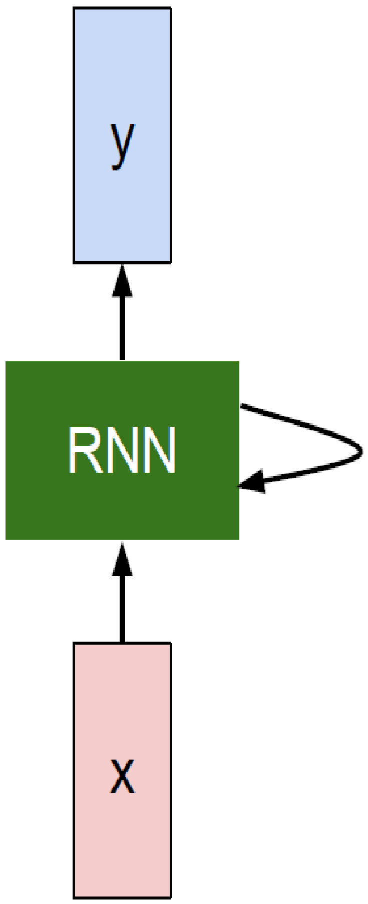


Recurrent Neural Network
(Source: Stanford.edu)

- The **Simple RNN** only has the **short-term** memory.
- It **cannot "memorize"** the pieces of information that show up **early** in the sequence chain.

Slide 19: Long Short-Term Memory Neural Networks (LSTM)

RNN: Problem of Short-Term Memory



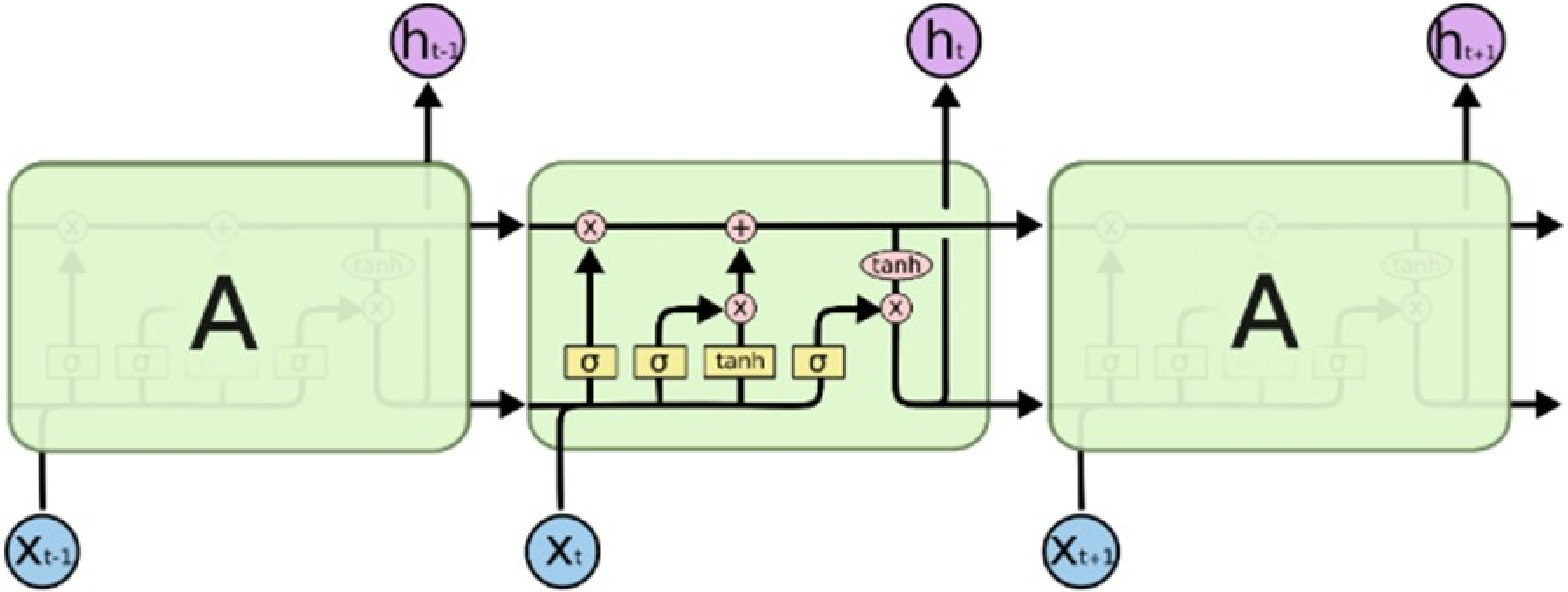
Recurrent Neural Network
(Source: Stanford.edu)

Vanishing Gradient Problem and **Short-Term Memory** of Simple RNN

What is the Cause-Effect relationship between them?

Slide 20: Long Short-Term Memory Neural Networks (LSTM)

RNN: LSTM: A Powerful Solution to RNN’s Short-Term Memory



Unrolled LSTM Neural Network (Source: Colah Blogs)

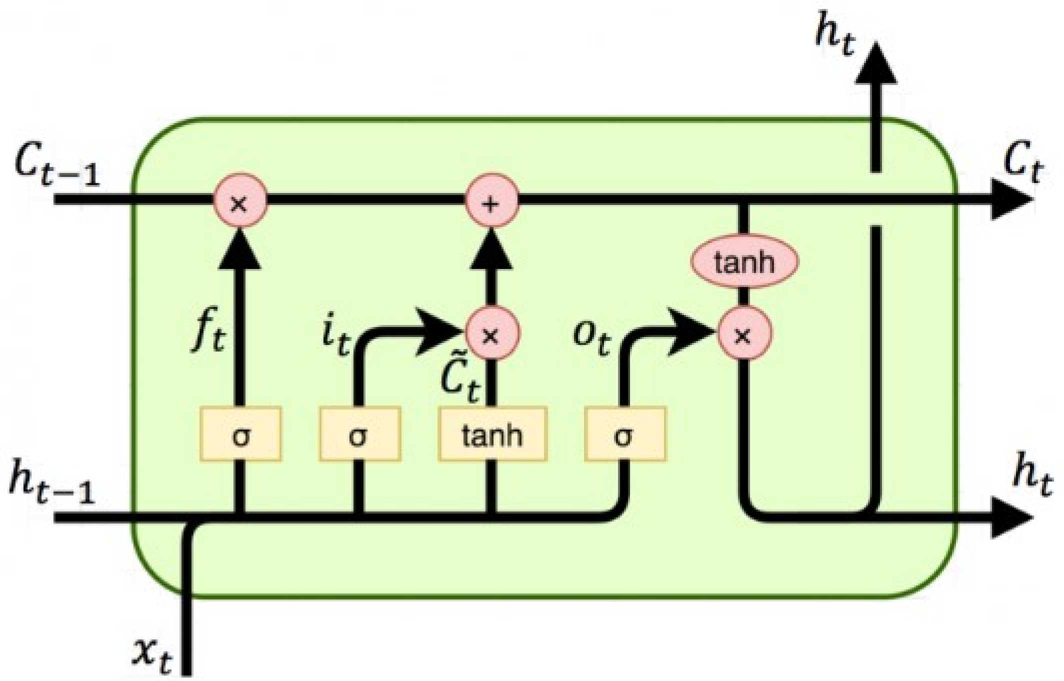
Slide 21: Long Short-Term Memory Neural Networks (LSTM)

RNN: LSTM: A Powerful Solution to Vanishing Gradient Problem

A Brief Introduction

- In the mid-90s, Sepp Hochreiter and Juergen Schmidhuber (German researchers) proposed the **Long-Short-Term Memory** (LSTM) neural network as a **solution** to the vanishing gradient problem.

- The **LSTM** network can **preserve** the **gradient** along the path of backpropagation through time and layers.
  - It **maintains** a more constant gradient while backpropagation is done.
  - The neural network can **continue learning** over many time steps.



LSTM Cell (Source: Colah's) Blog