LECTURER: Nghia Duong-Trung

NEURAL NETS AND DEEP LEARNING

TOPIC OUTLINE

- Course book: DLBDSNNDL01_Neural Nets and Deep Learning, provided by IU, myStudies
- Reading list DLBDSNNDL01, provided by IU, myStudies
- This slide is a summarization of important contents in the course book.
- Additional teaching materials:

https://github.com/duongtrung/IU-DLBDSNNDL01_Neural_Nets_and_Deep_Learning

DISCLAIMER

- This is the modified version of the IU slides.
- I used it for my lectures at IU only.

UNIT 6

RECURRENT NEURAL NETWORKS

STUDY GOALS

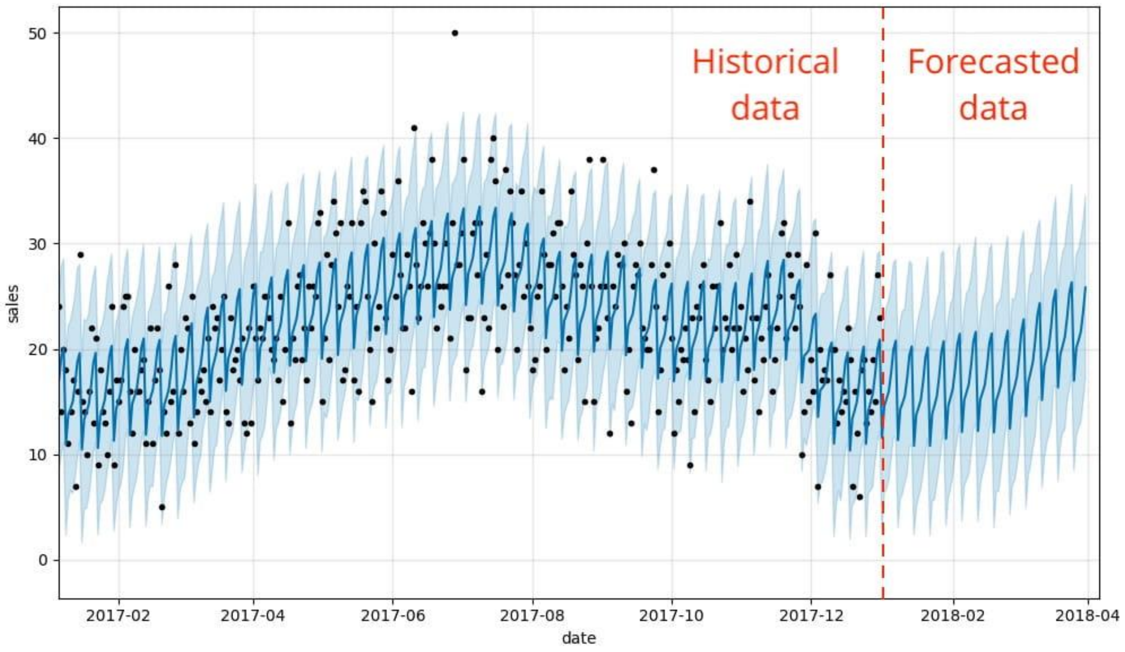
On completion of this unit, you will be able to ...

- understand what recurrent neural networks are and how they work.
- know what memory cells are and the role they play in a recurrent neural network.
- identify the major challenges encountered when training recurrent neural networks.
- comprehend what a long short-term memory unit is and why it performs better on long sequences than a naive recurrent neural network.
- develop a simple recurrent neural network to process sequence data.

EXPLAIN SIMPLY

- What are recurrent neural networks (RNNs) and their functionalities?
- How do LSTM units work?
- How to develop a basic RNN for sequential data processing?

- Any continuous-valued measurement taken periodically. Go-to examples: a company's stock price, weather: forecasting task is very useful.
- Speech / audio recognition
- Text can be treated as a sequence.
- Usually, we think of a 1-D time series signal. We can extend the concept to multiple dimensions, e.g., NxD matrix.
  - D = #features -> T = sequence length or #time steps in the sequence



Source of the image: https://www.databricks.com/blog/2020/01/27/time-series-forecasting-prophet-spark.html

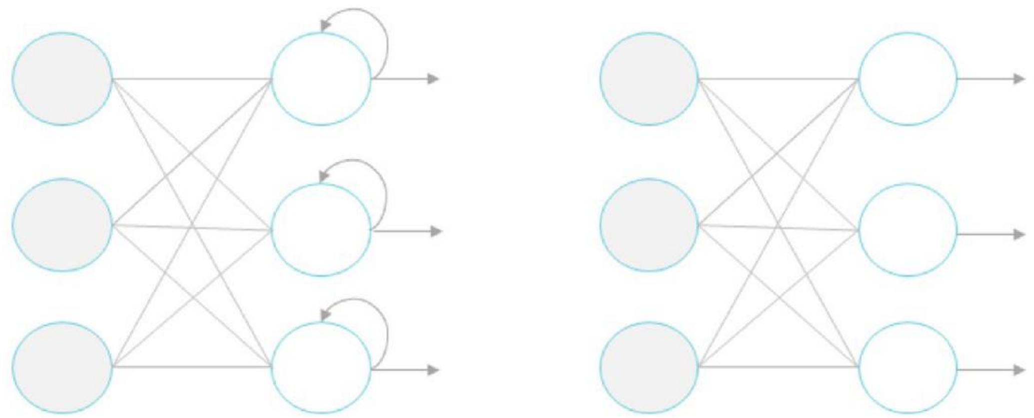
Recurrent Neural Networks (RNNs)…

- are distinguished by their "**memory**," which allows them to maintain a state of what the network has observed thus far.
- **iterate** through elements and can **maintain** a state of what the network has observed so far.
- can process a **sequence** of information and use past inputs to affect the final output.
- are particularly useful for problems where context is important.
- can learn to form a **deeper understanding** of the sequence of input data by leveraging past inputs.

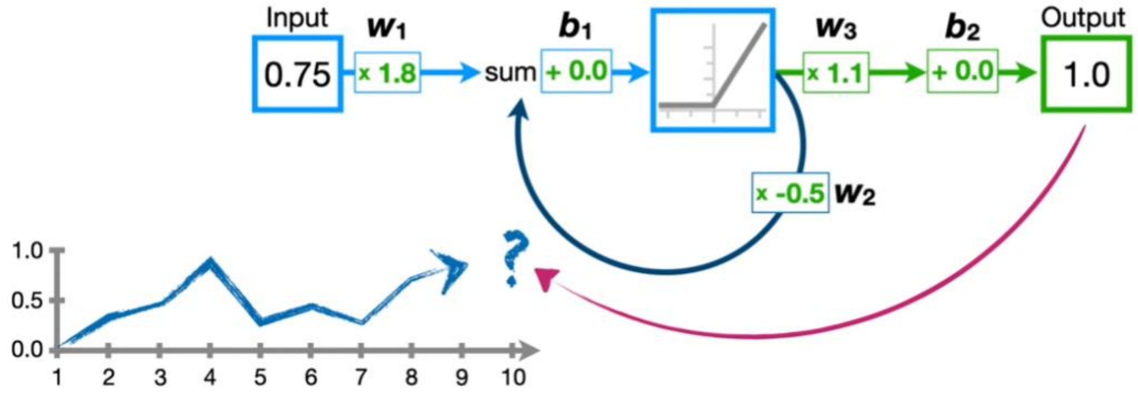
Source of the text: Zöller, 2023
Source of the image: Own Creation

Information flow in RNNs

- RNNs are structured in a way that facilitates the transmission of information through a **cycle**.
- To make a decision, RNNs consider not only the **current** input, but also what it has learned from **prior** inputs.
- Because of their internal **memory**, RNNs are designed to pass information in **two directions**.



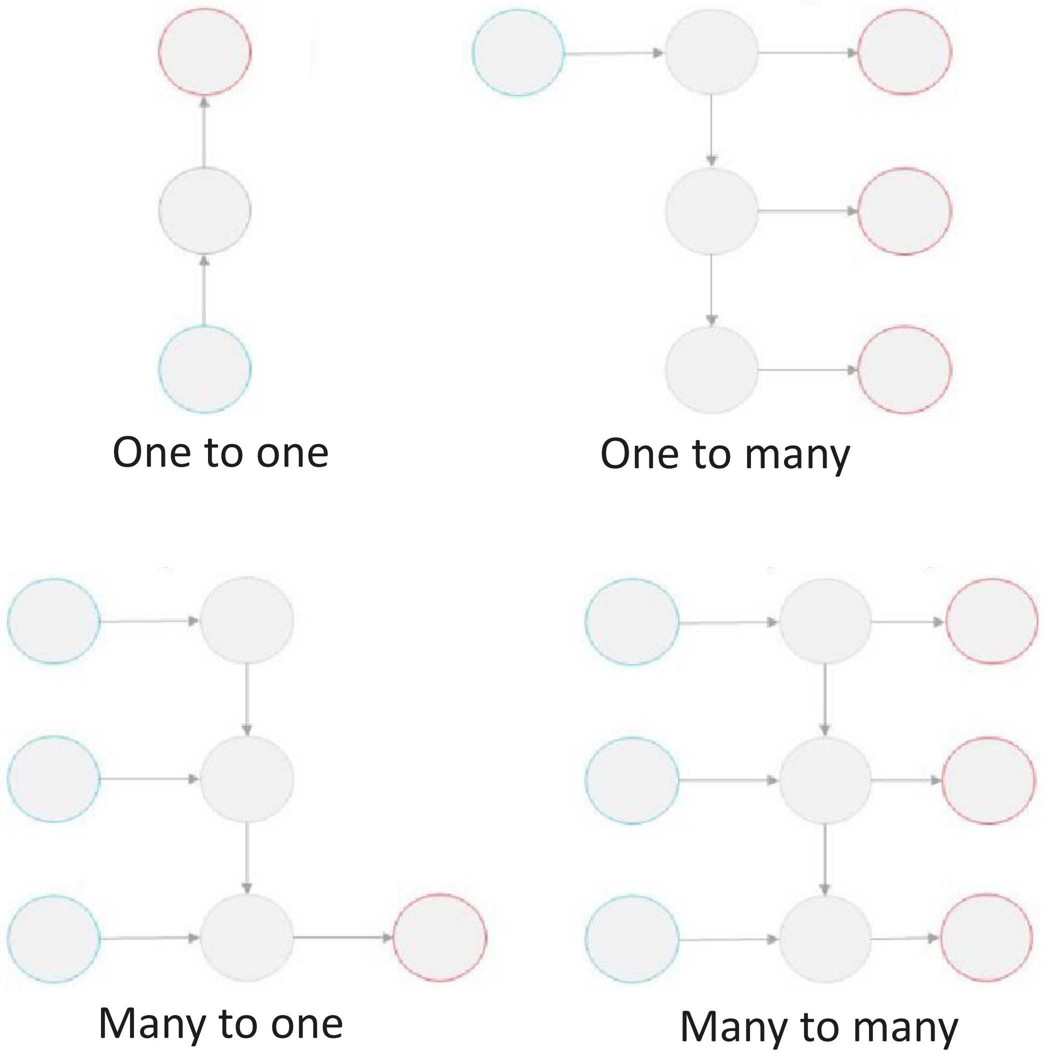
RNN versus Feed-forward neural networks



Source of the text and the image: Zöller (2023)
Source of the image: https://www.youtube.com/watch?v=AsNTP8Kwu80

Architectural types of RNNs

- **One-to-one RNN** maps one input to one output, similar to a regular neural network.
- **One-to-many RNN** allows a single input to be mapped to multiple outputs.
- **Many-to-one RNN** generates a single output from a series of inputs.
- **Many-to-many RNN** uses a series of input data to generate a series of outputs.



Source of the text and the image: Zöller (2023)

Issues of RNNs

- **Exploding gradients …**
  - occur when **large error** gradients accumulate, resulting in unreasonably **large updates** to the weights of the network.
  - cause the network to become **unstable** and ultimately **unable to learn** well from the training data.
  - are addressed by applying **truncation** and **clipping** techniques.
- **Vanishing gradients …**
  - occur when the networks become **unable to propagate** useful gradient information from the output of the model back to the layers **near the input** of the model.
  - cause the network to become **difficult to learn** meaningful patterns out of the input data.
  - are addressed by applying **weight initialization** technique.

Source of the text and the image: Zöller (2023)

Memory cell

- Recurrent neuron preserves information in a state (a **memory** cell) across different timesteps.
- At a given time step t, the output of a recurrent neuron is **dependent on inputs** from previous timesteps.

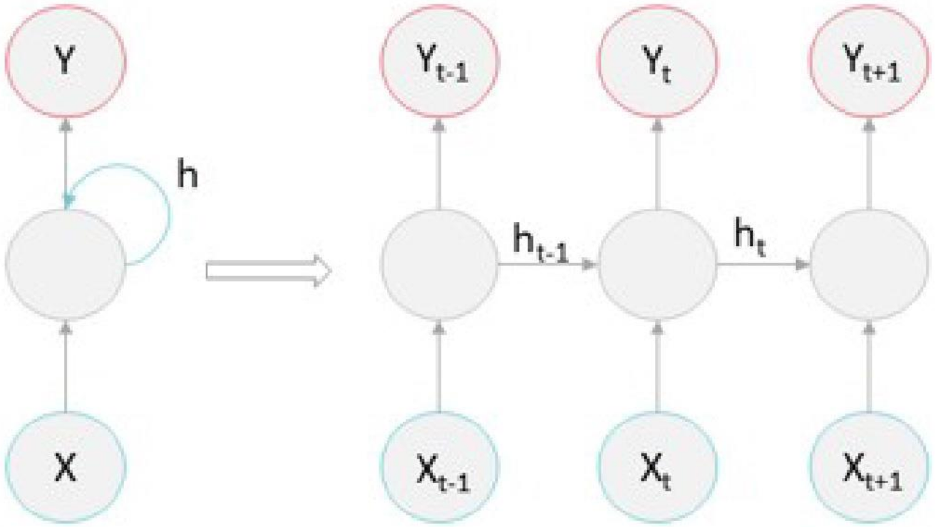
  State of the cell: $h_t = f(h_{t-1}, X_t)$

  where:

  h_t- cell state at timestep t

  h_{t-1}- cell state at timestep $t - 1$

  X_t - input at timestep t

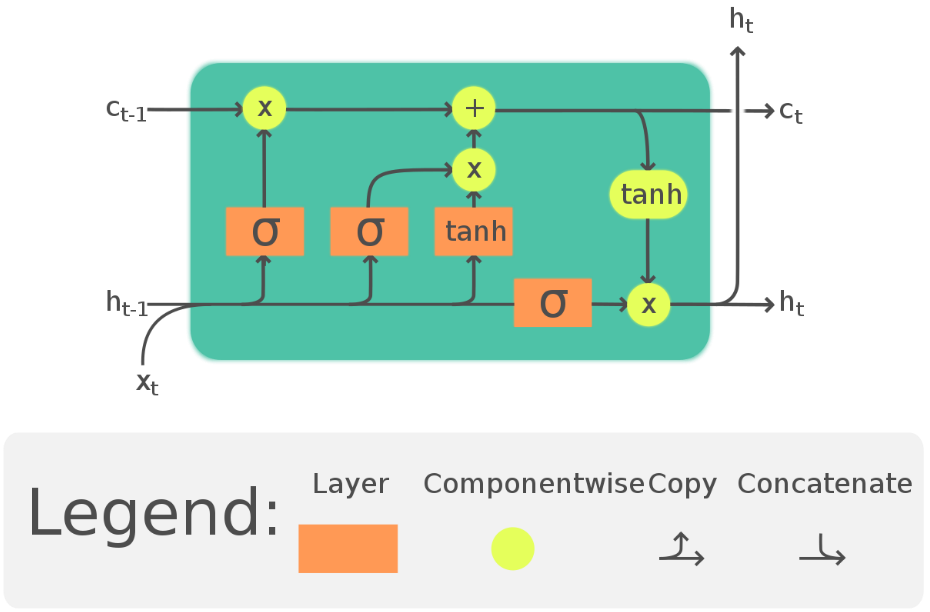


A memory cell with hidden states

Source of the text and the image: Zöller (2023)

Long Short-Term Memory (LSTM) ...

- is a special type of RNNs capable of handling **long-term dependencies**.
- addresses the **vanishing gradient** problem in RNNs.
- preserves relevant information from **earlier sequences** and carries it forward.
- can **add** or **remove** information from the cell state based on the regulations of the gates.
- can learn from events that have a significant **time lag**.



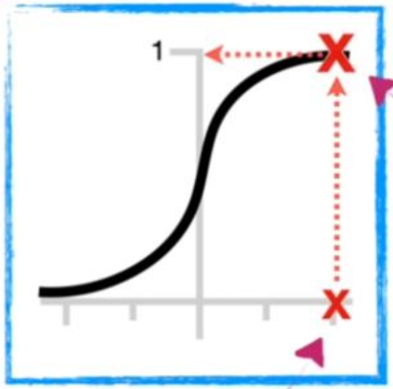
A LSTM cell

Source of the text: Zöller (2023)
Source of the image: File:LSTM_Cell.svg

SIGMOID AND TANH ACTIVATION

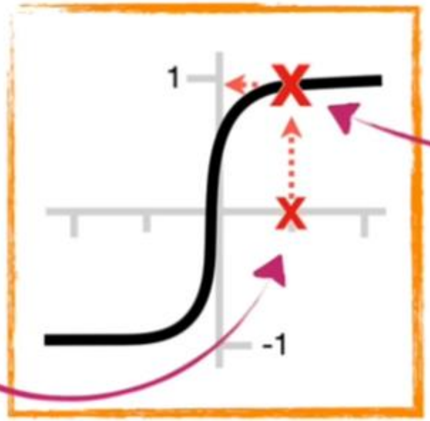
In a nutshell, the **Sigmoid Activation Function** takes any x-axis coordinate…

…and turns it into a y-axis coordinate between **0** and **1**.

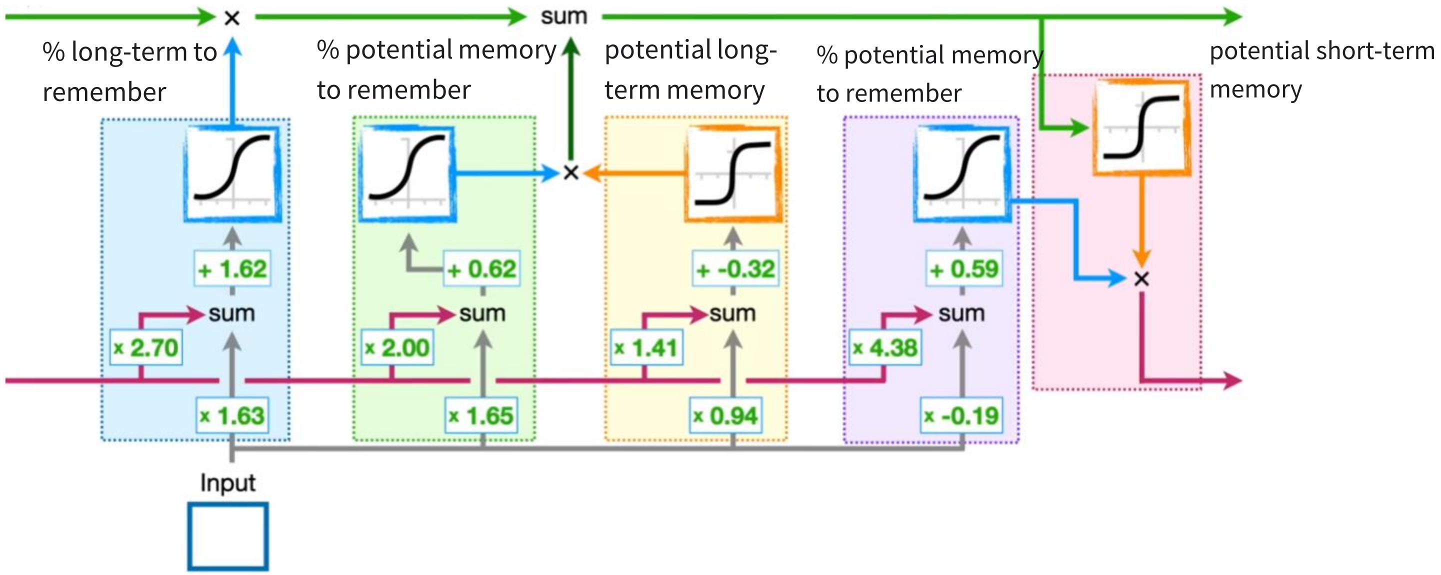


In contrast, the **tanh**, or **Hyperbolic Tangent**, **Activation Function** takes any x-axis coordinate…

…and turns it into a y-axis coordinate between **-1** and **1**.



- The top green line that runs all the way across the top of the unit is called the **Cell State** and represents the **Long-Term Memory**.
- The **lack of weights** allows the Long-Term Memories to flow through a series of unrolled units without causing the gradient to explode or vanish.

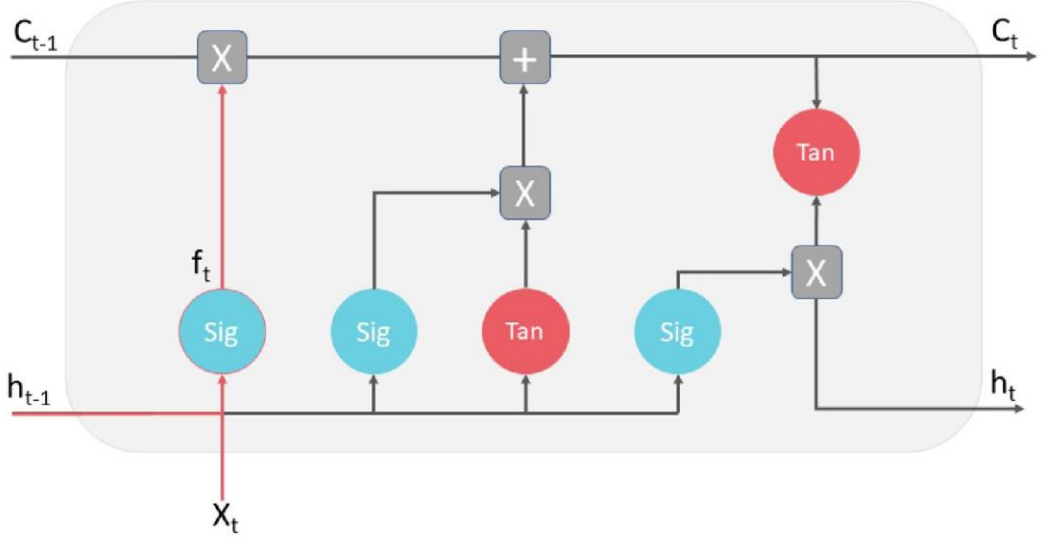


- The pink line, called the **Hidden State**, represents the **Short-Term Memories**.
- Short-Term Memories are directly connected to **weights** that can modify them.

Source of image: https://www.youtube.com/watch?v=YCzL96nL7j0

LSTM Forget Gate ...

- output determines which information is **discarded**, **remembered**, or **partially remembered** for further steps.
- considers the **input** at a given timestep and the **hidden state** from the previous timestep.

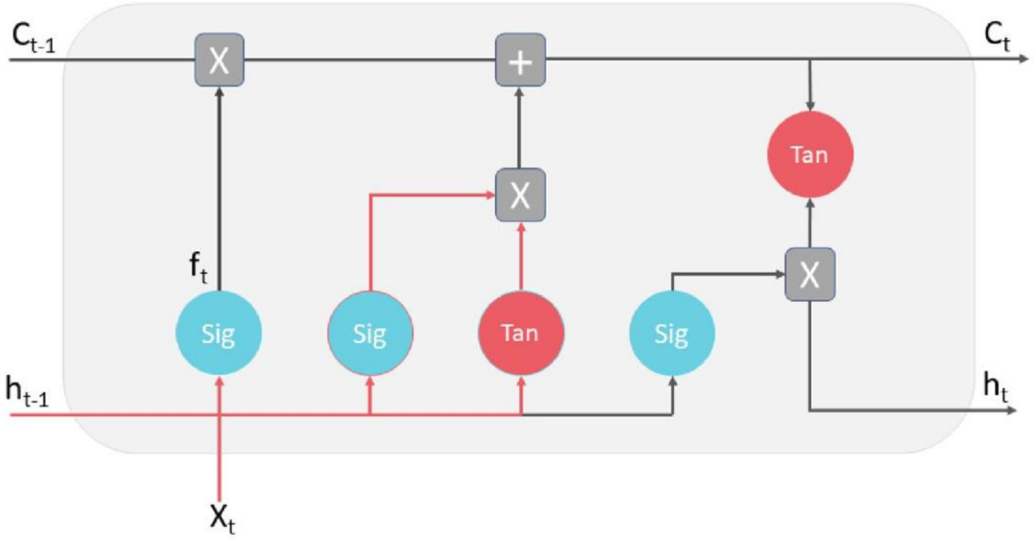


LSTM Forget Gate

Source of the text and the image: Zöller (2023)

LSTM Input Gate …

- determines which values will be **added** to the cell state and long-term memory.
- multiplies the relevant information with the forget vector and performs **pointwise addition** to update the cell state.

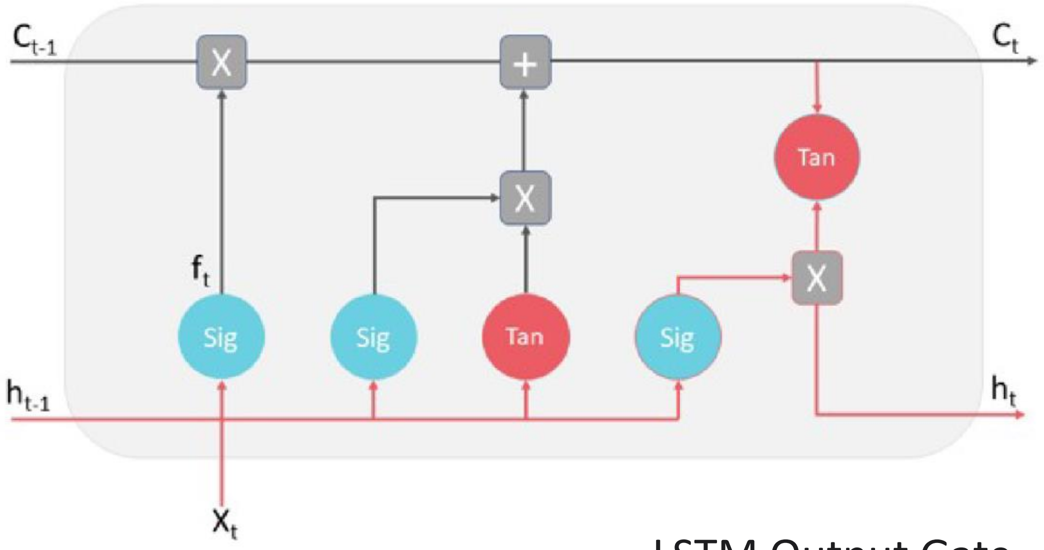


LSTM Input Gate and Update State

Source of the text and the image: Zöller (2023)

LSTM Output Gate ...

- determines what information to **output** from the cell.
- combines the hidden state and current input to modify the state.

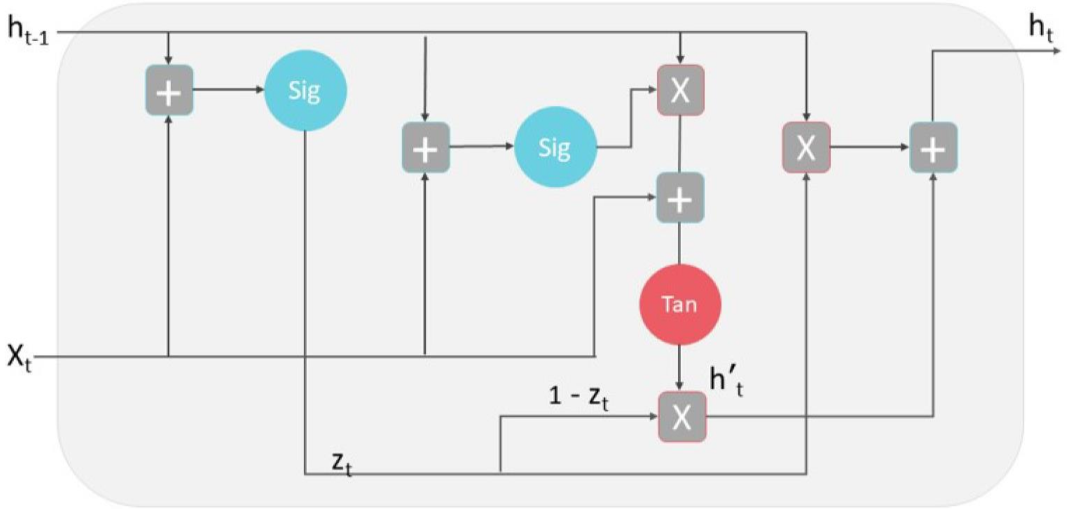


LSTM Output Gate

Source of the text and the image: Zöller (2023)

Gated Recurrent Unit network (GRU) ...

- is leverages **connections** through a sequence of nodes.
- addresses the **vanishing gradient** problem.
- has a **simpler** architecture, uses **less memory**, and requires **less time** to train.
- uses the **hidden state** to pass information.
- relies on **gates** to control information flow.
- has two gate operating **mechanisms**:
  - **Update gate:** decides the amount of previous information that will be **passed** into the next state.
  - **Reset gate:** decides how much of the information from earlier timesteps it should be **forgot.**

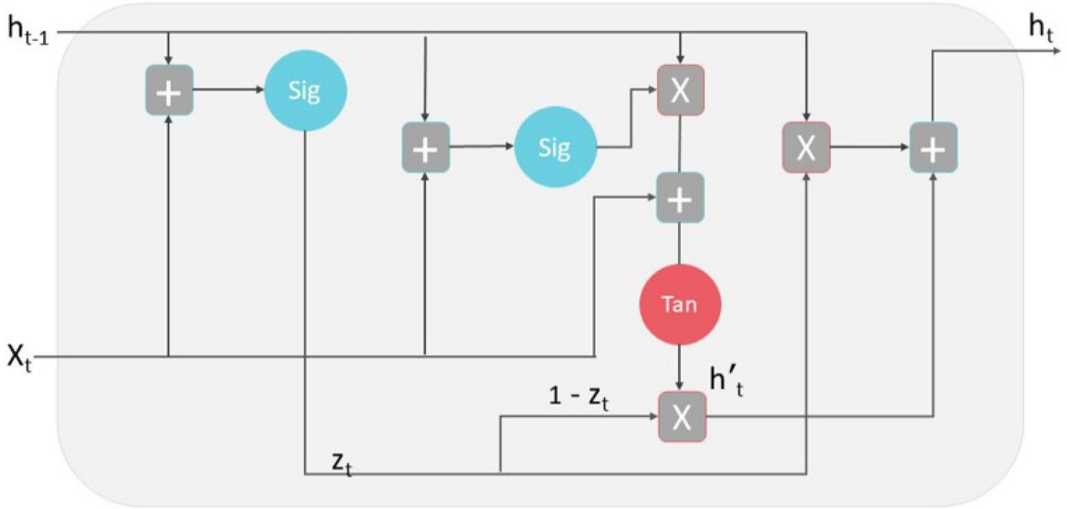


LSTM Output Gate

Source of the text and the image: Zöller (2023)

Training RNN

- **Backpropagation through time** BPTT is used for training certain types of RNNs.
- All input timesteps need to be **unrolled**.
- At each timestep, the algorithm will consider **one input** and **one output** timestep and a **copy** of the RNN.
- At each timestep, the **errors** will be calculated, then **accumulated** together.
- **Forward Pass:** generates the input for the next timestep.
- **Backward Pass:** starts from the end and moves backwards to compute the gradient.



LSTM Output Gate

Source of the text and the image: Zöller (2023)

Implementation of an LSTM Layer in Python

```
# import & install libraries
import keras
from keras import activations
from keras.layers import Embedding, Dense, LSTM

# create LSTM layer
embedding_input_size = 32
embedding_output_size = 16
lstm_size = 16
network_output_size = 1
network = keras.models.Sequential()
network.add(Embedding(embedding_input_size, embedding_output_size))
network.add(LSTM(output_size))
network.add(Dense(units=network_output_size,activation=activations.sigmoid))
```

Source of the text: Zöller (2023)

REVIEW STUDY GOALS

- understand what recurrent neural networks are and how they work.
- know what memory cells are and the role they play in a recurrent neural network.
- identify the major challenges encountered when training recurrent neural networks.
- comprehend what a long short-term memory unit is and why it performs better on long sequences than a naive recurrent neural network.
- develop a simple recurrent neural network to process sequence data.

SESSION 6

RECURRENT NEURAL NETWORKS

TRANSFER TASK
PRESENTATION OF THE RESULTS

Please present your results.

The results will be discussed in plenary.

Task:

Suppose you want to train a deep learning model for weather forecasting.
Based on the characteristics of RNNs, explain briefly why they can be effectively applied for weather forecasting? Support your answer with 4 arguments.

Sample solution:

1. Weather data exhibits **sequential** dependencies, and RNNs are well-suited for modeling and predicting sequences, allowing forecasters to capture patterns over time.
2. RNNs can handle **variable-length** input sequences, meaning they can work with weather data that varies in terms of length and time intervals, making them adaptable for different forecasting scenarios.
3. RNNs can process past weather observations to predict future weather conditions, leveraging the ability of the network's hidden state to **retain information** from previous time steps.
4. RNNs can provide **probabilistic forecasts** by incorporating uncertainty estimation techniques, enabling weather forecasters to assess the reliability of their predictions and communicate potential risks accurately.

1. Which of the following best describes the key idea behind recurrent neural networks (RNNs)? Select one.
   a) RNNs are a type of neural networks that use recurrent features from a dataset to find the best answers.
   b) RNNs are a type of neural networks that use loops between the most important features to predict the next output.
   c) RNNs are a type of neural networks that have no memory, and therefore can perform computations much faster and more effectively.
   d) RNNs are a type of neural network distinguished by their “memory” as they leverage prior inputs to influence current inputs and output.

2. Which of the following statements is true about Long-Short Term Memory (LSTM)? Select one

a) A long short-term memory unit is a special type of a recurrent neural network capable of handling well long-term dependencies.

b) LSTMs differ from regular recurrent neural networks in the fact that they leverage feedback loops to perform computations faster.

c) LSTMs address the vanishing gradient problem by using two gates commonly known as the “update” and “reset” gates which filter information based on previous inputs.

d) LSTMs provide only a theoretical advantage over regular recurrent neural networks, as their usage in practice does not show important results.

3. Which of the following statements is true about backpropagation through time (BPTT) training algorithm? Select one
    a) During BPTT the network considers forward connections from the current input to the target output in order to make a prediction.
    b) BPTT works well only with gated recurrent unit (GTU) networks, but not with long short-term memory units (LSTM) networks because of their architecture.
    c) BPTT is a gradient-based technique used for training certain types of recurrent neural networks such as long short-term memory networks.
    d) Though BPTT offers a robust framework for training a recurrent neural network, it is rarely used in practice because of the heavy computational load of the algorithm.

Answers

1. d)
2. a)
3. c)

LIST OF SOURCES

Text

Zöller, T. (2023). Neural Nets and Deep Learning Course Book. IU International University of Applied Sciences.

Images

Zöller (2023)

File:LSTM_Cell.svg (2023, September). Long Short-Term Memory. *Wikipedia Commons.* https://en.wikipedia.org/wiki/Long_short-term_memory