Deep Learning with RNNs and LSTMs

Randil Pushpananda

University of Colombo School of Computing

No 35, Reid Avenue, Colombo 07

rpn@ucsc.cmb.ac.lk

Natural Language Processing

Examples of Sequence Models (RNNs)

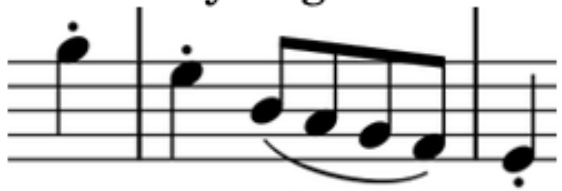
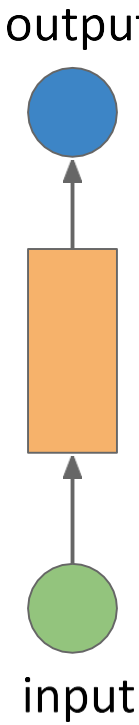
Speech recognition		"The quick brown fox jumped over the lazy dog."
Music generation	∅	
Sentiment classification	"There is nothing to like in this movie."	
DNA sequence analysis	AGCCCCTGTGAGGAACTAG	AGCCCCTGTGAGGAACTAG
Machine translation	Voulez-vous chanter avec moi?	Do you want to sing with me?

Image by Andrew Ng on Coursera

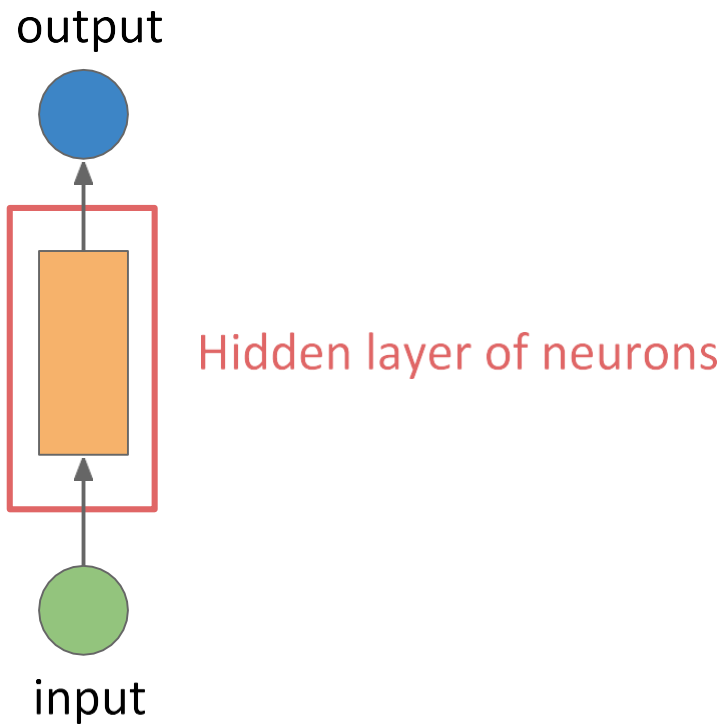
Feed Forward Neural Network



The simplest form of neural network

1-layer Feedforward Neural network

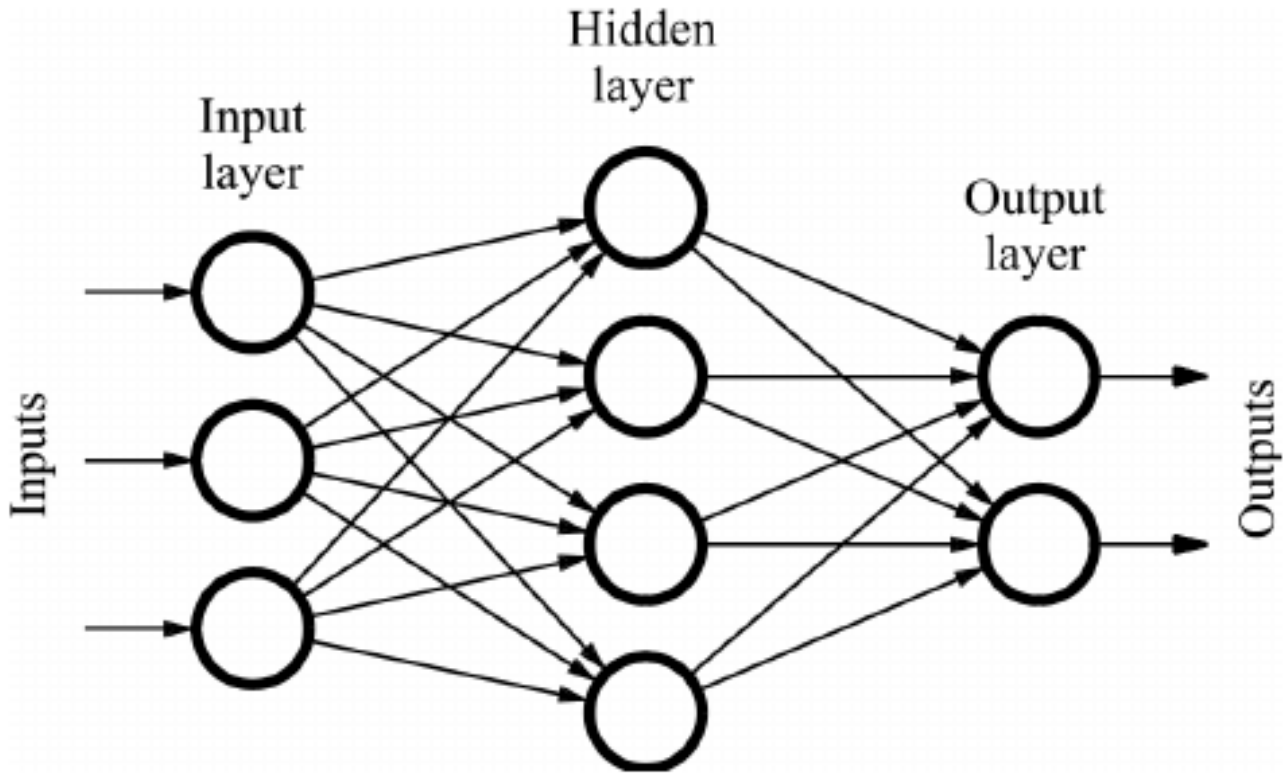
Feed Forward Neural Network



Information is only processed in one direction.

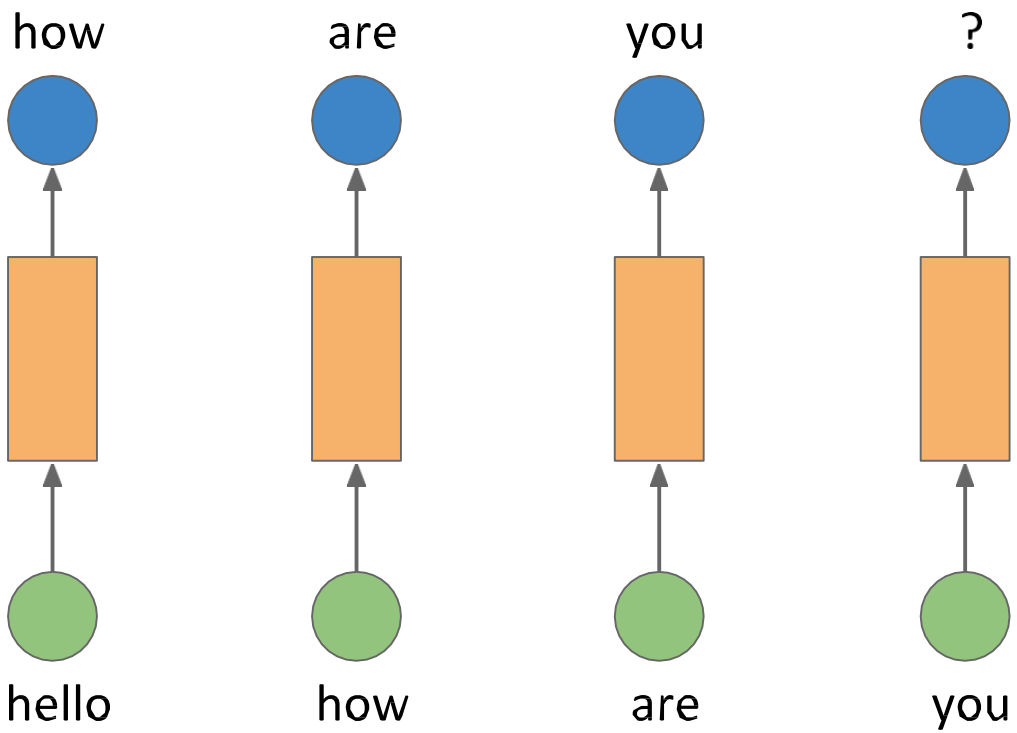
1-layer Feedforward Neural network

Feed Forward Neural Network

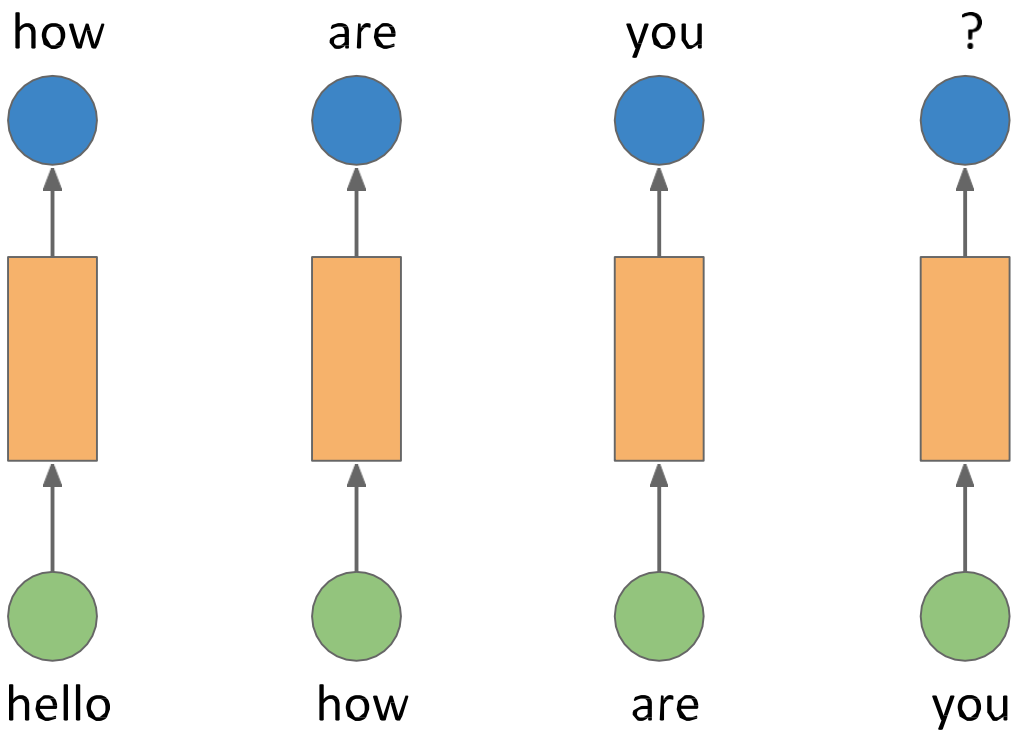


While the data may pass through multiple hidden nodes, it always moves in one direction and never backwards.

Feed Forward Neural Network

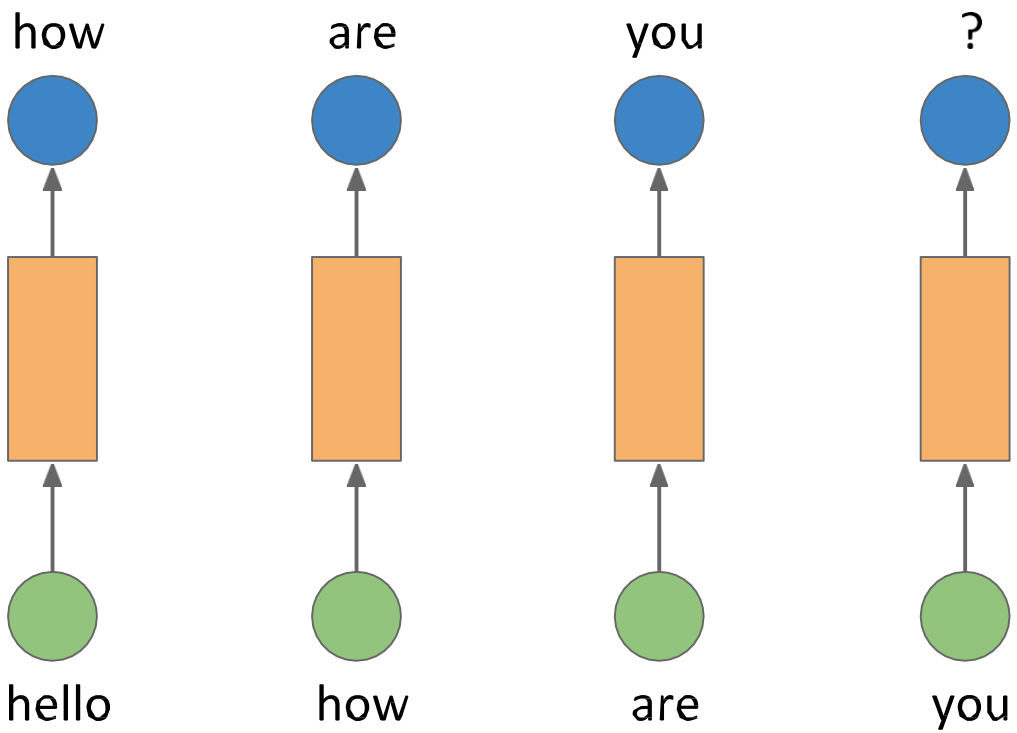


Feed Forward Neural Network



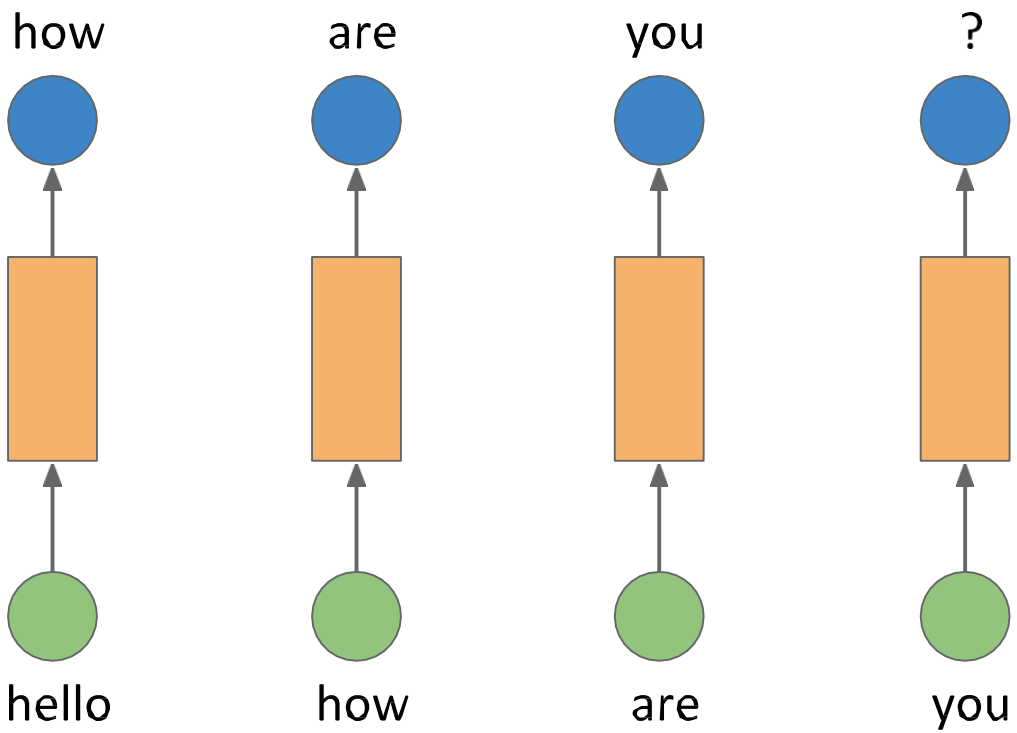
In the real world, we remember some history of previous words

Feed Forward Neural Network



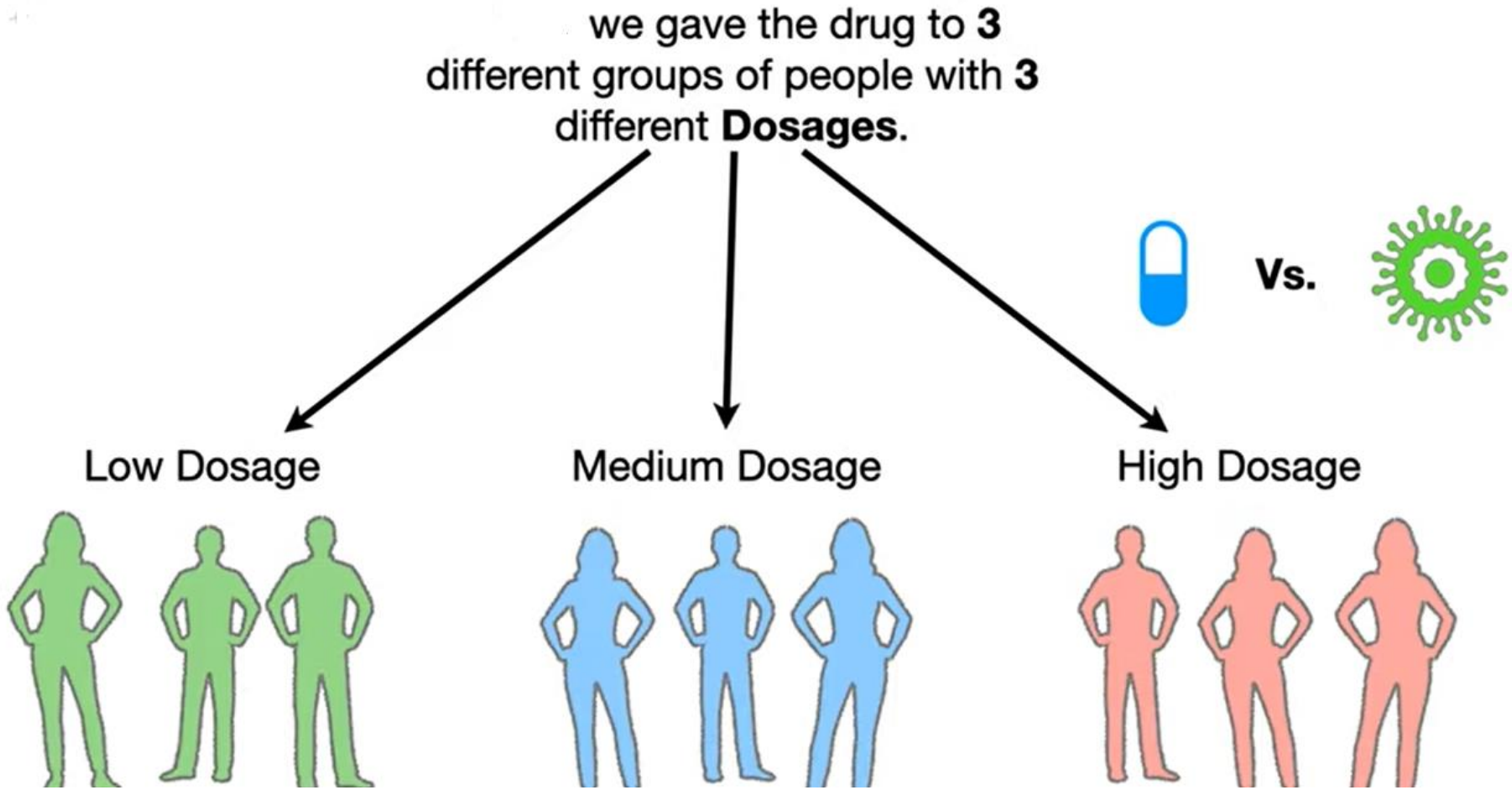
In our network here, each step is independent of the previous steps

Feed Forward Neural Network

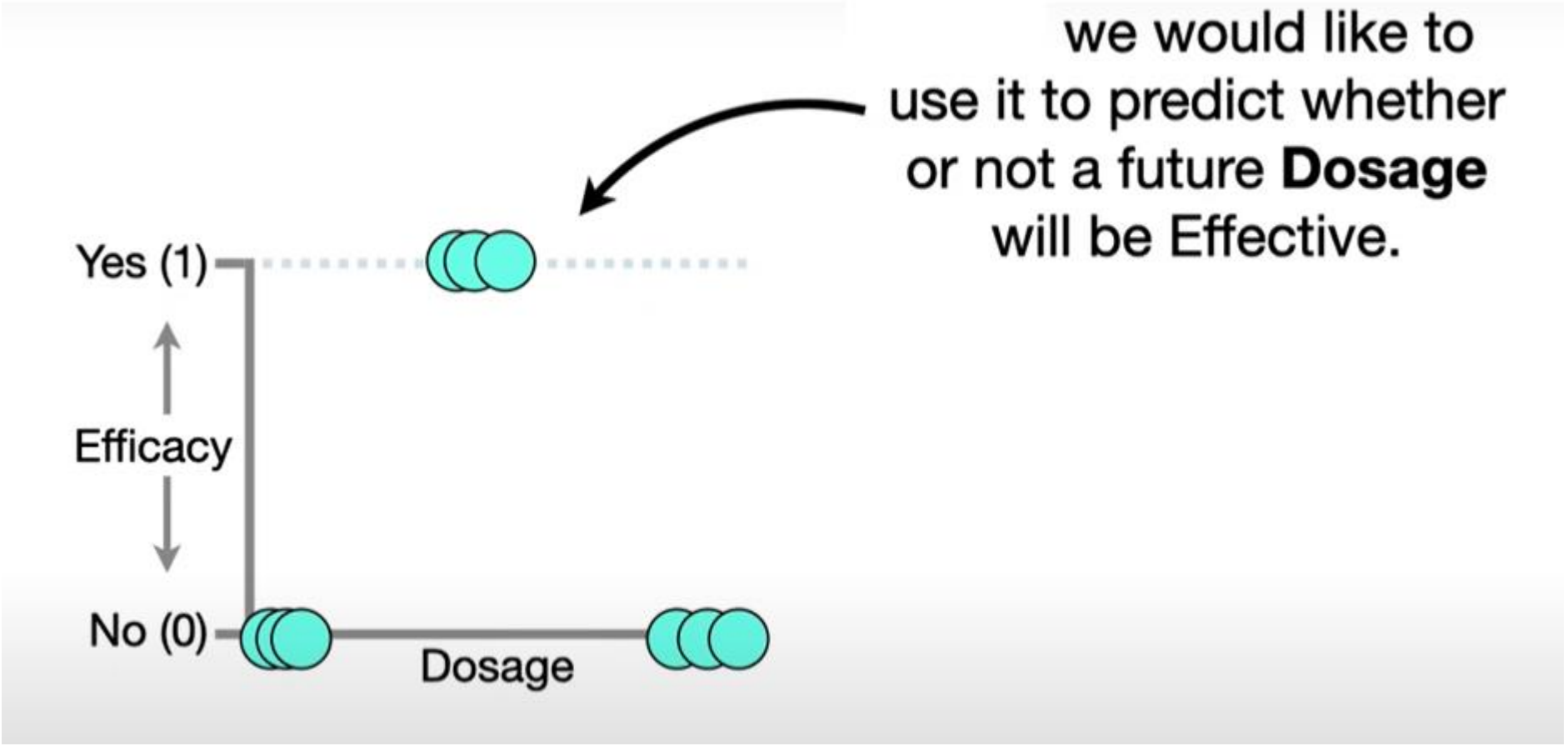


The only context available at every step is the input we provide to the network (bigram, trigram etc)

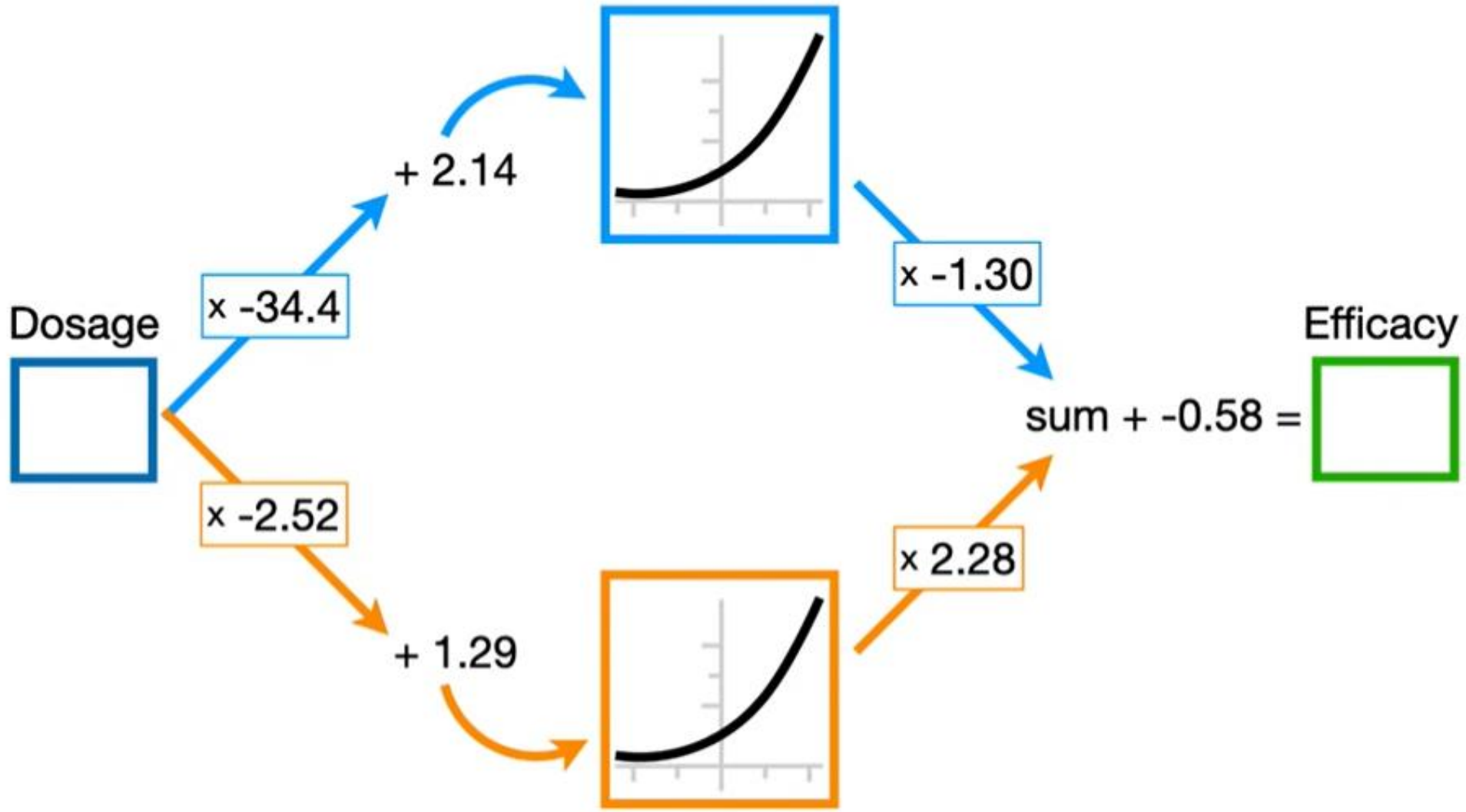
Problem

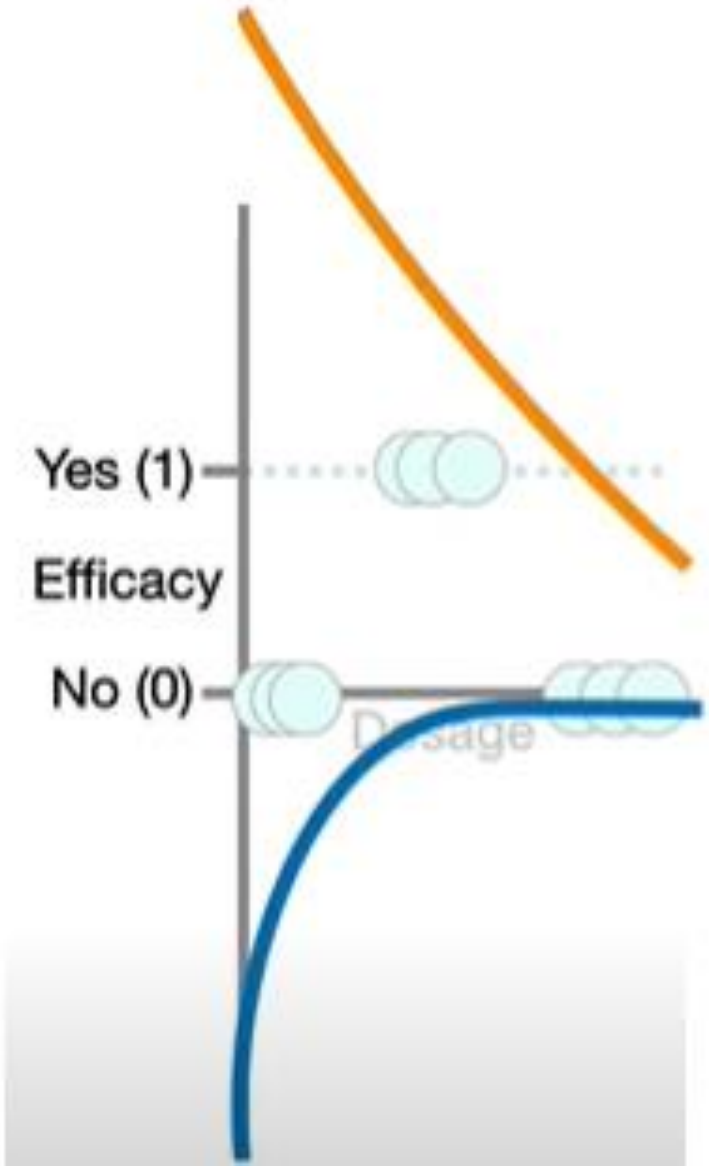

we gave the drug to **3** different groups of people with **3** different **Dosages**.
Vs.
Low Dosage
Medium Dosage
High Dosage

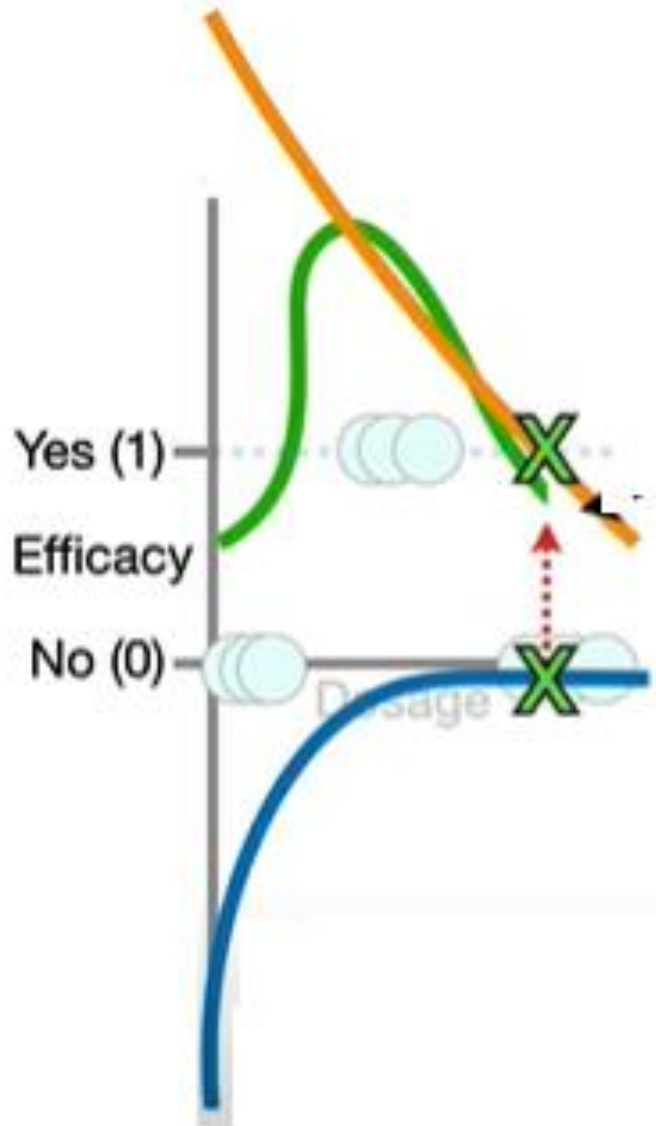
Problem

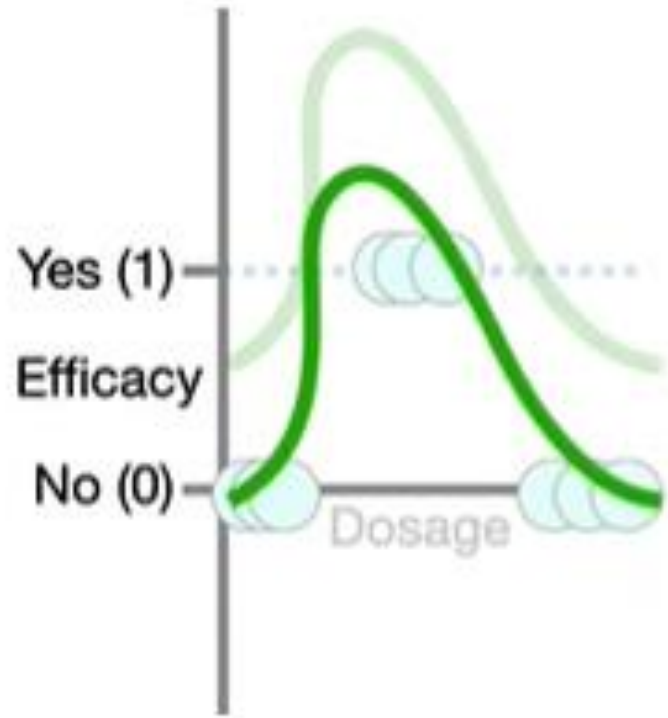

we would like to use it to predict whether or not a future **Dosage** will be Effective.
Yes (1)
Efficacy
No (0)
Dosage

Problem



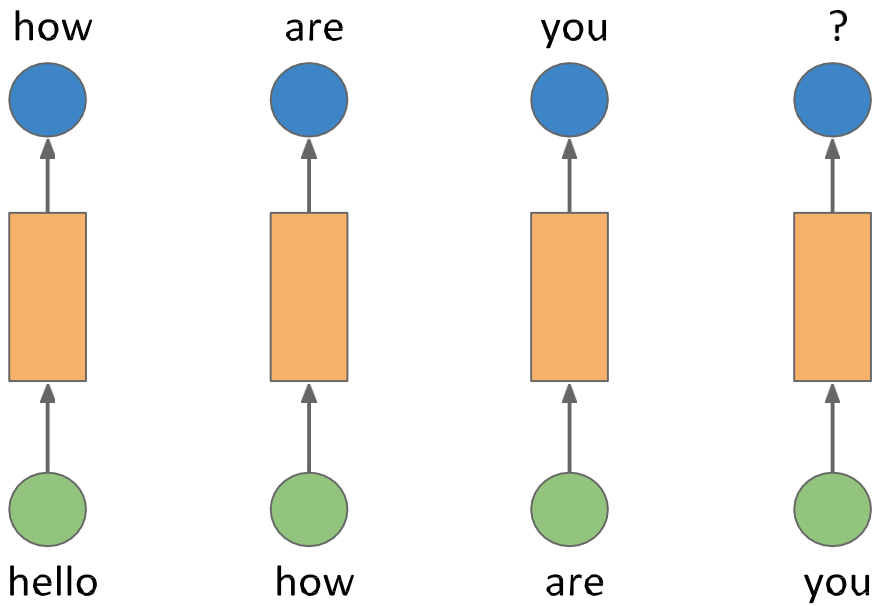

Yes (1)
Efficacy
No (0)
Dosage


Yes (1)
Efficacy
No (0)
Dosage

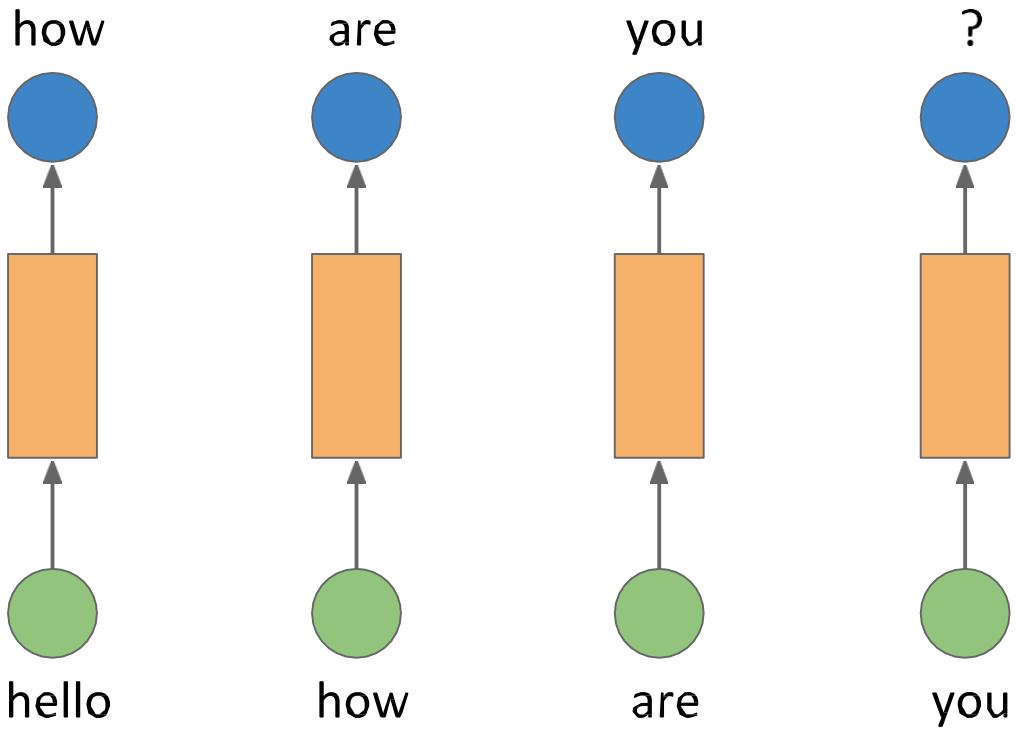

Yes (1)
Efficacy
No (0)
Dosage

Problems of Feed Forward Neural Network

- Loss of neighborhood information.
- More parameters to optimize.

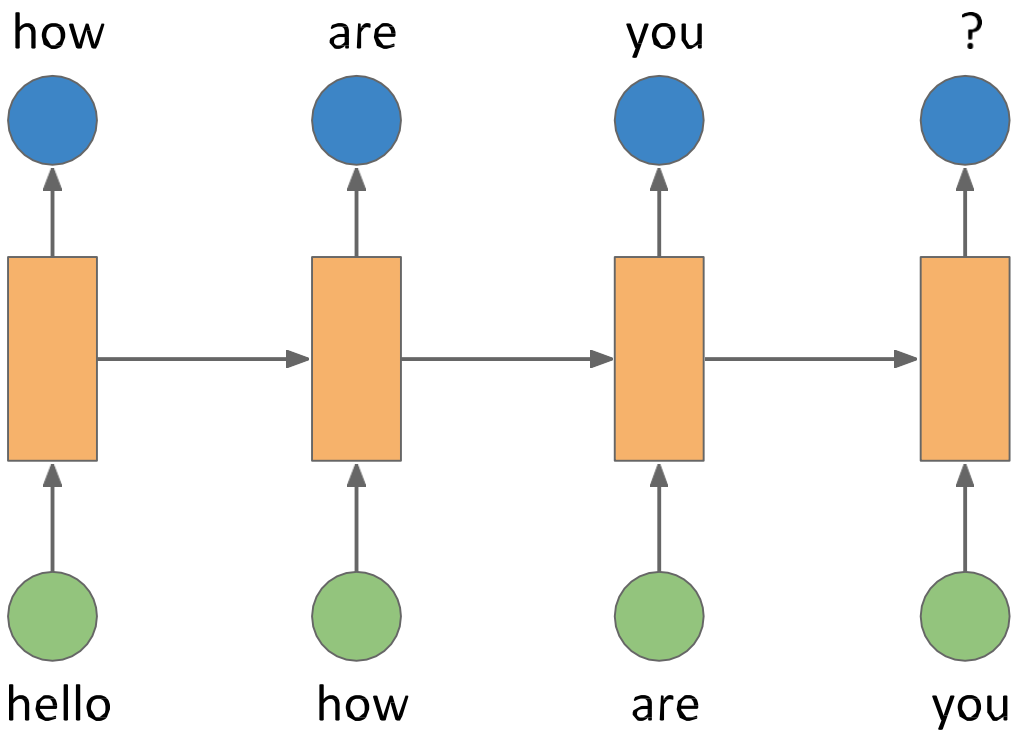


Feed Forward Neural Network



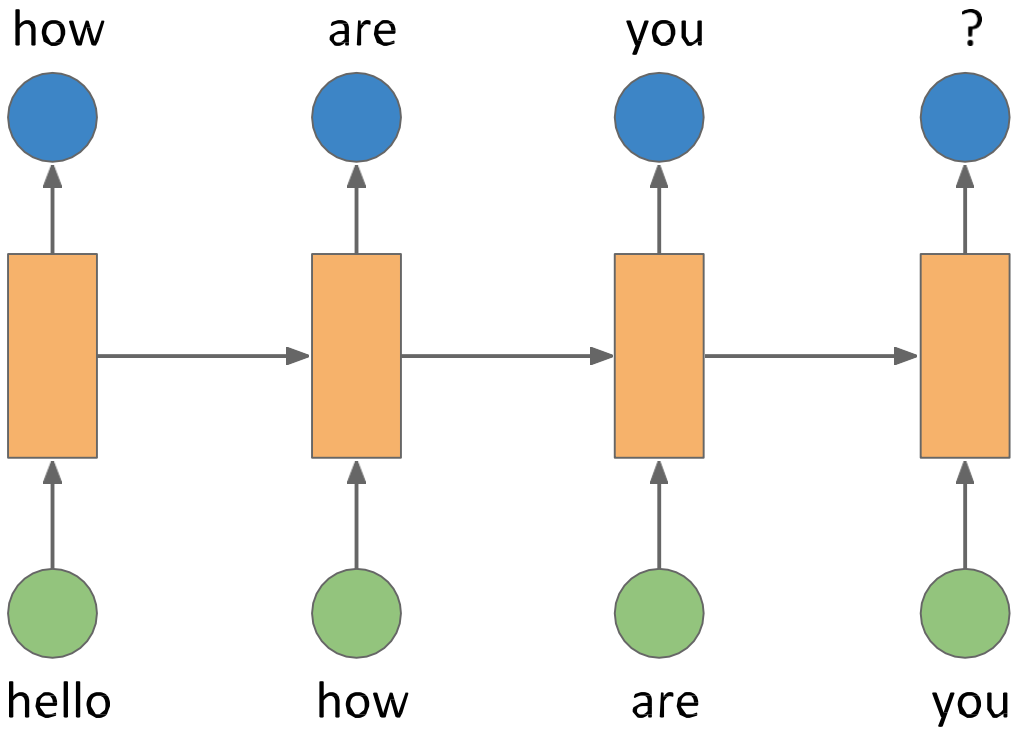
Why not connect these networks?

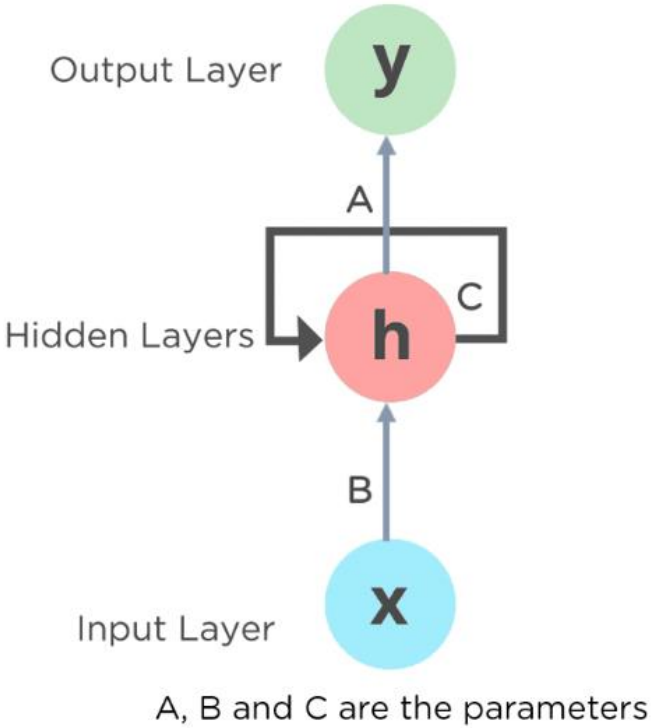
Recurrent Neural Network



Output of a particular layer is feeding back to the input in order to predict the output of the layer.

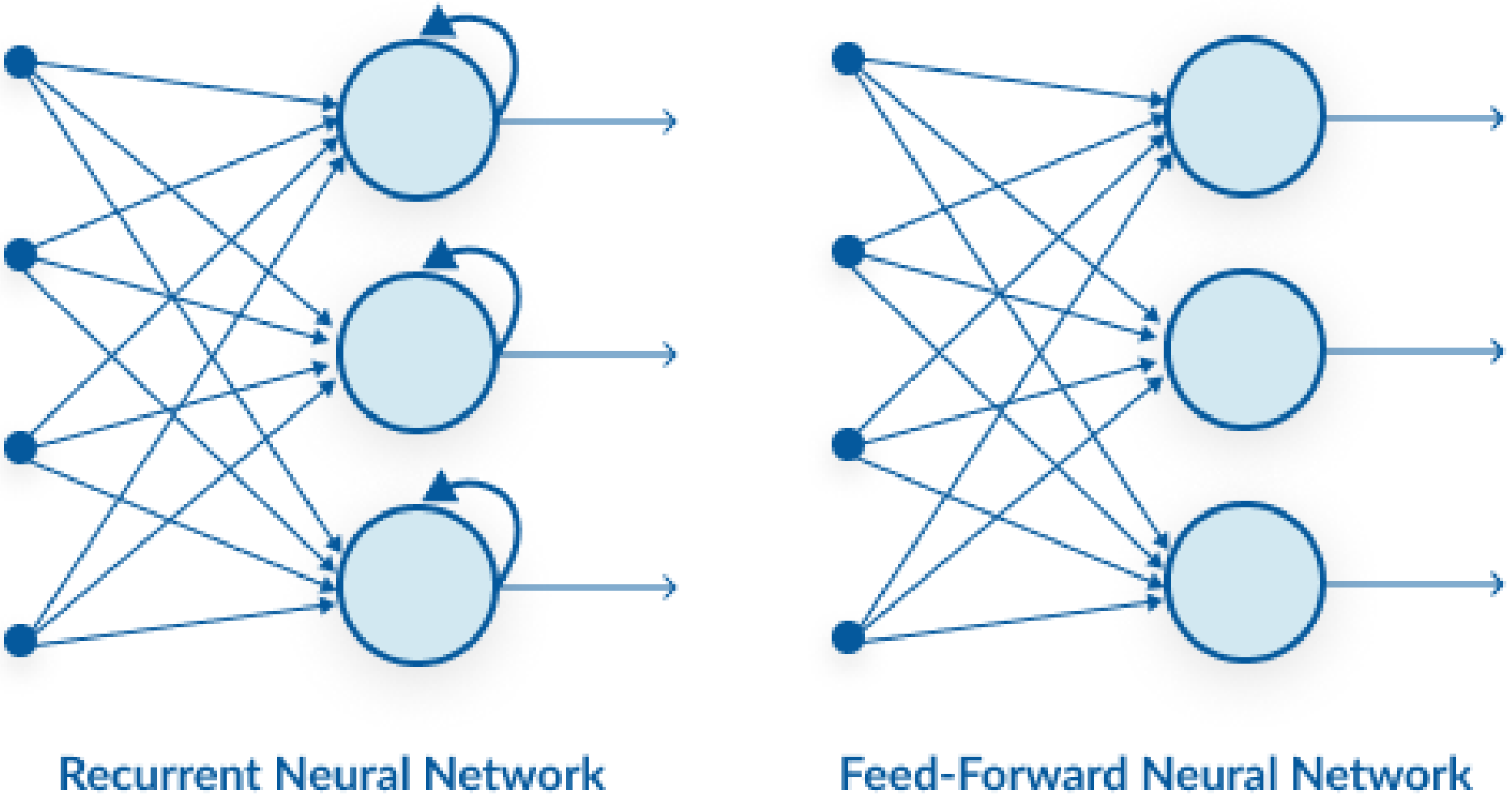
Recurrent Neural Network



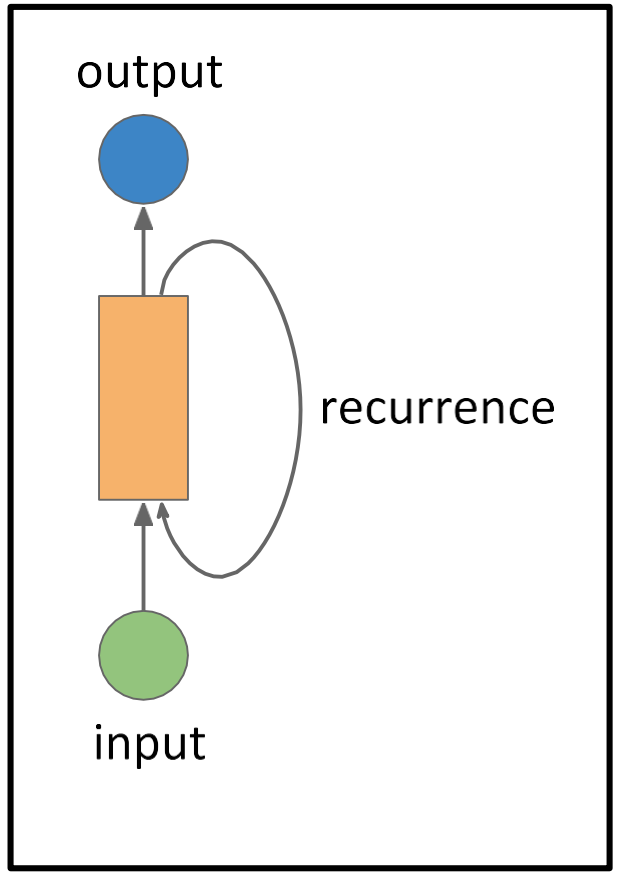


This is what recurrent neural networks do

Recurrent Neural Network

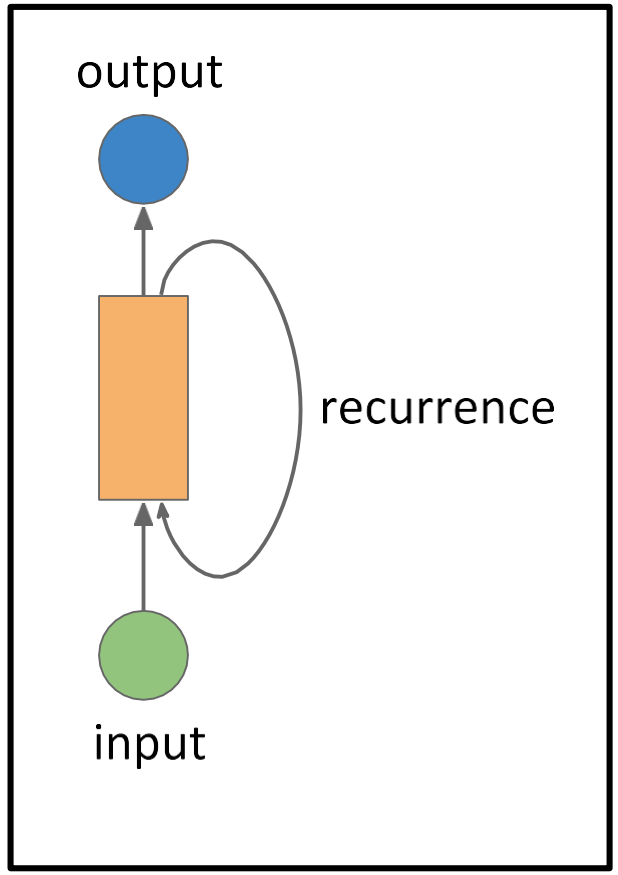


Recurrent Neural Network

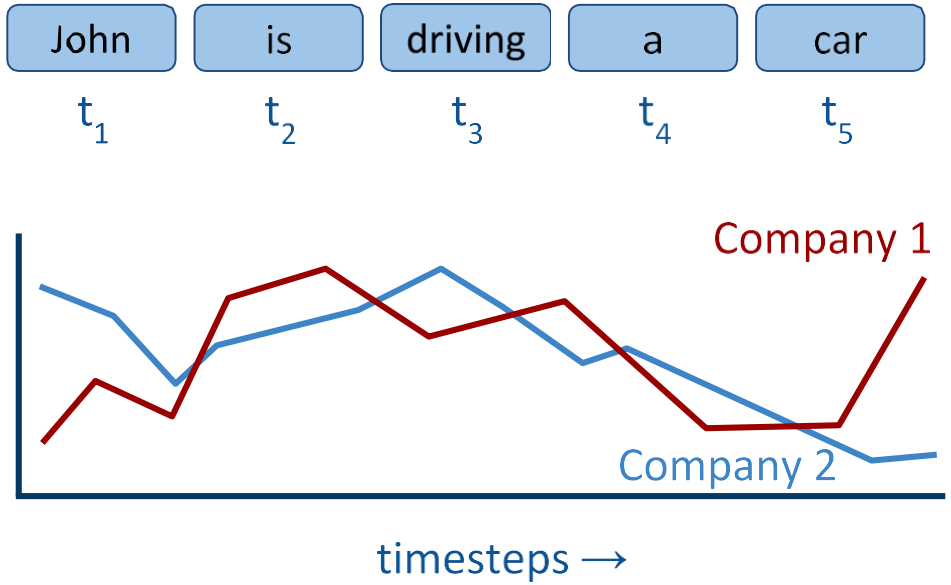


Recurrent units work very well for sequential information like a series of words, or knowledge across *timesteps*

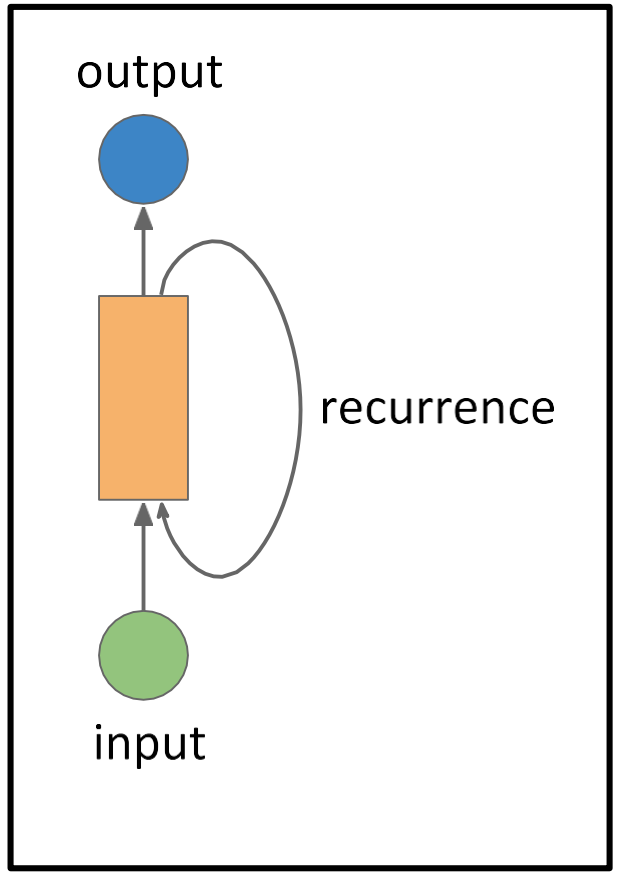
Recurrent Neural Network



Recurrent units work very well for sequential information like a series of words, or knowledge across *timesteps*



Recurrent Neural Network



Recurrent units work very well for sequential information like a series of words, or knowledge across *timesteps*

The recurrence unit has two inputs:

1) x_i (input at time i)
2) h_{i-1} (input from previous state)

Recurrent Neural Network

Mathematically,

$$h = Wx + b \longrightarrow h_t = Wx + W_h h_{t-1} + b$$

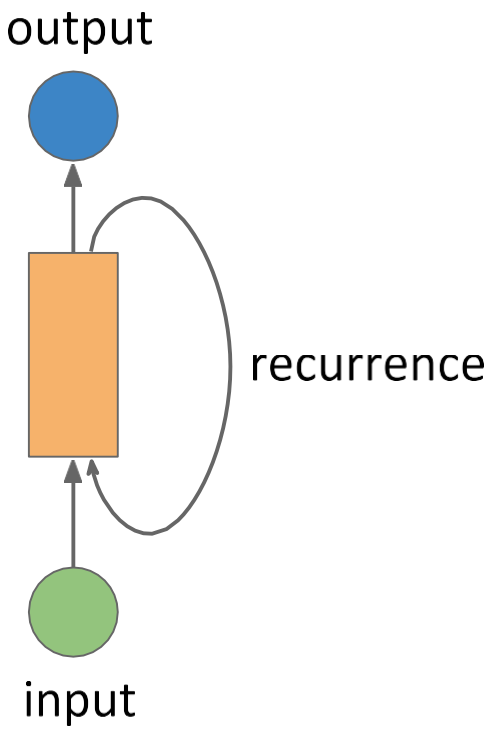
Linear

Recurrent

We have one additional set of parameters: W_h which deals with the information transferred from the previous step

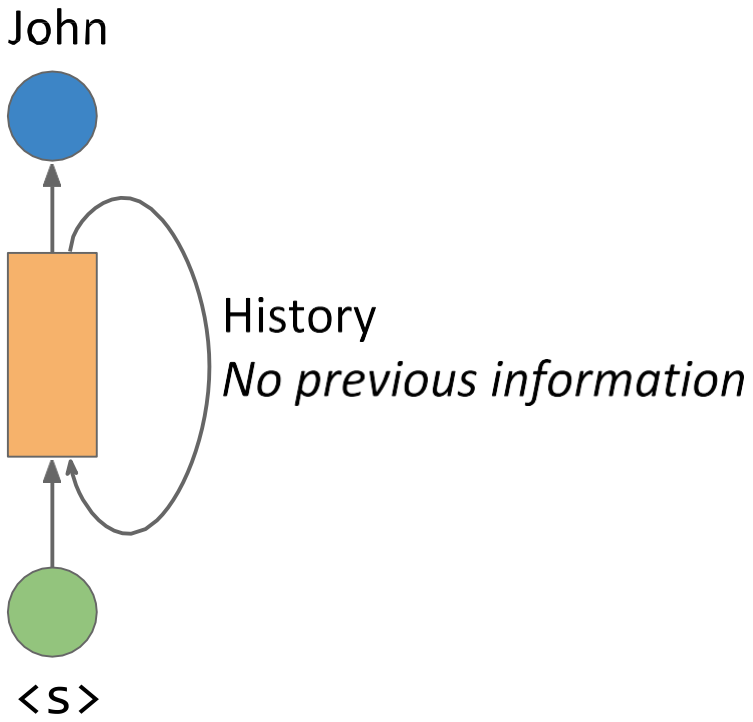
Recurrent Neural Network

Consider an example: <s> John is driving a car



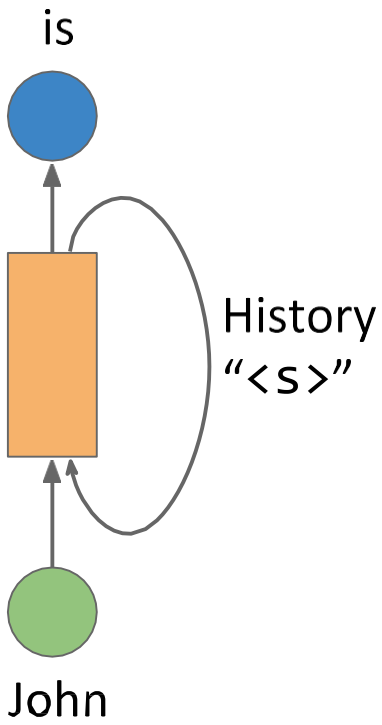
Recurrent Neural Network

Consider an example: <s> John is driving a car



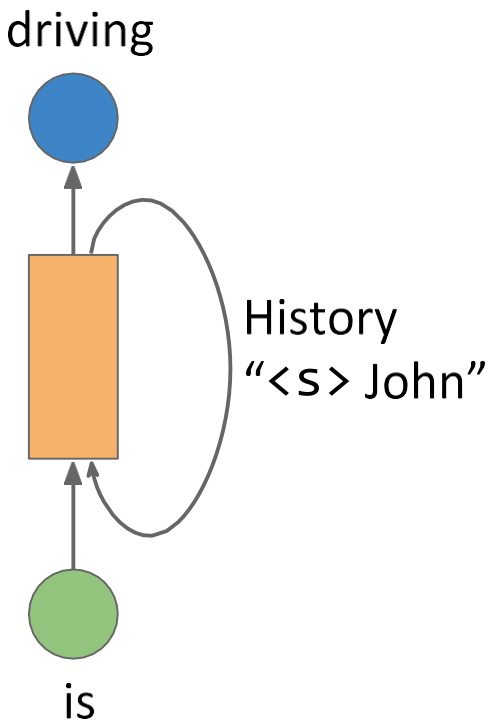
Recurrent Neural Network

Consider an example: <s> John is driving a car



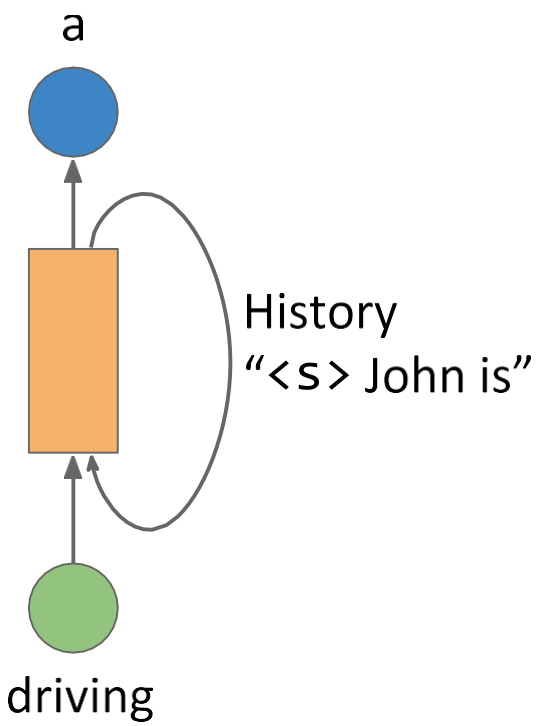
Recurrent Neural Network

Consider an example: <s> John is driving a car



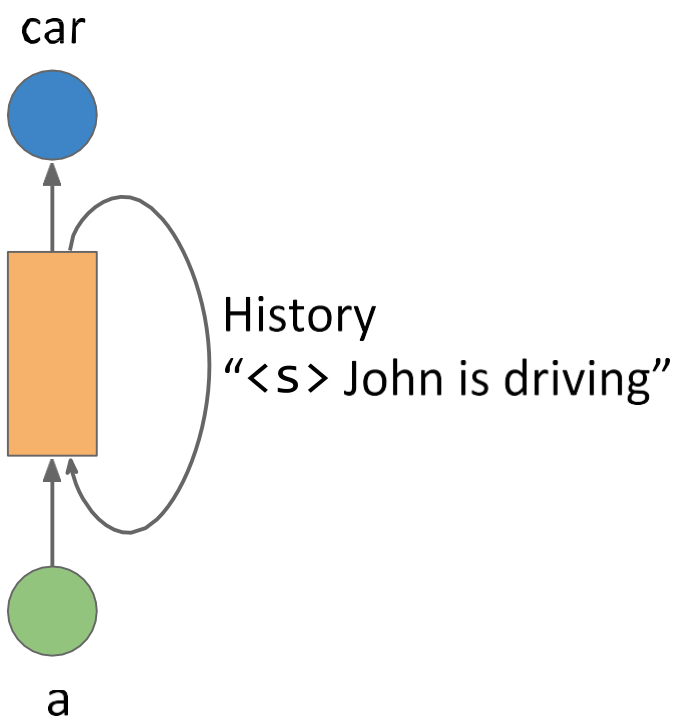
Recurrent Neural Network

Consider an example: <s> John is driving a car



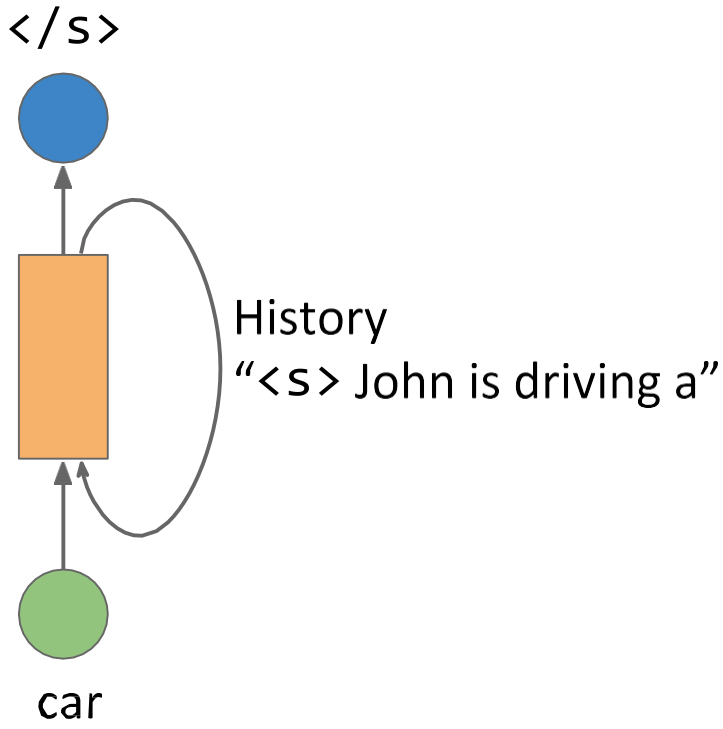
Recurrent Neural Network

Consider an example: <s> John is driving a car



Recurrent Neural Network

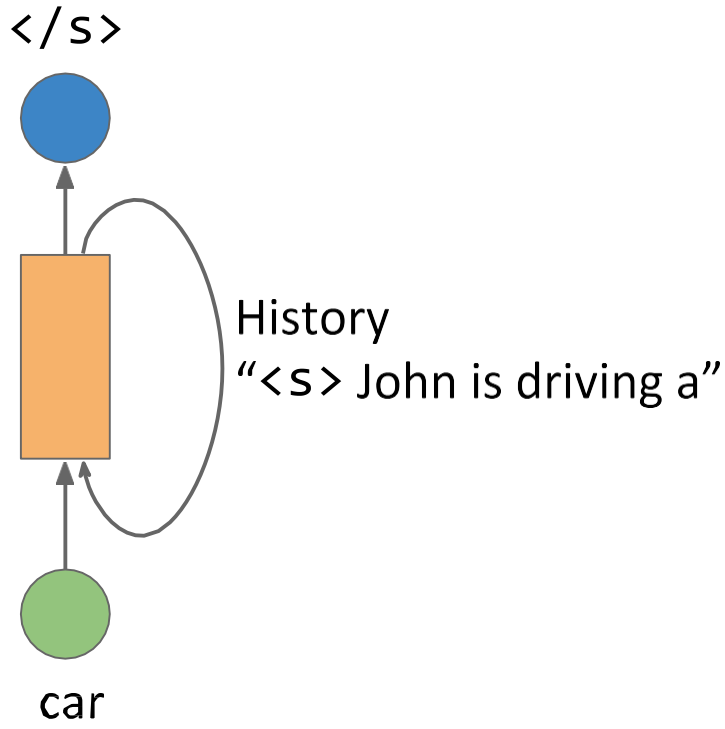
Consider an example: <s> John is driving a car



Recurrent Neural Network

Consider an example: <s> John is driving a car

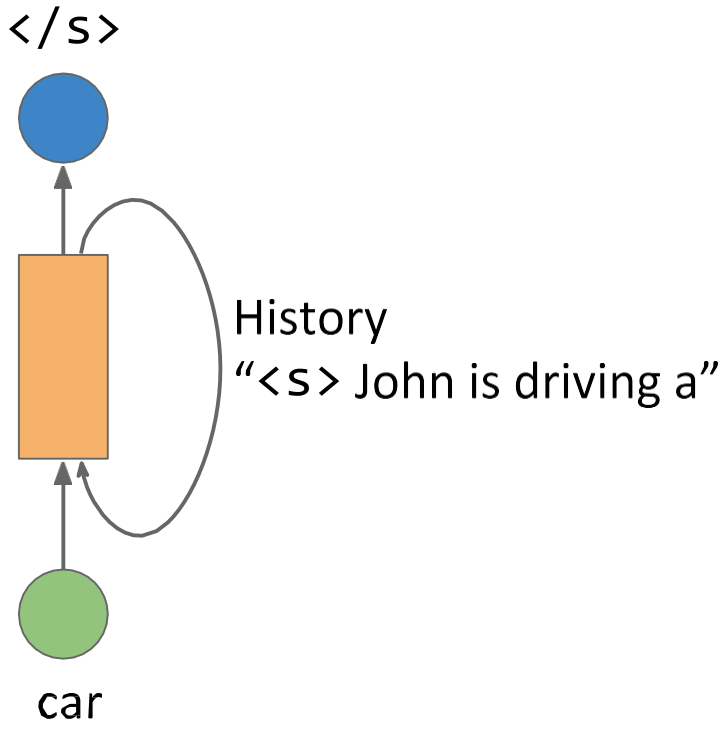
At the last timestep, the hidden state will have information about the entire sentence: **“John is driving a”** from history and **“car”** from the input



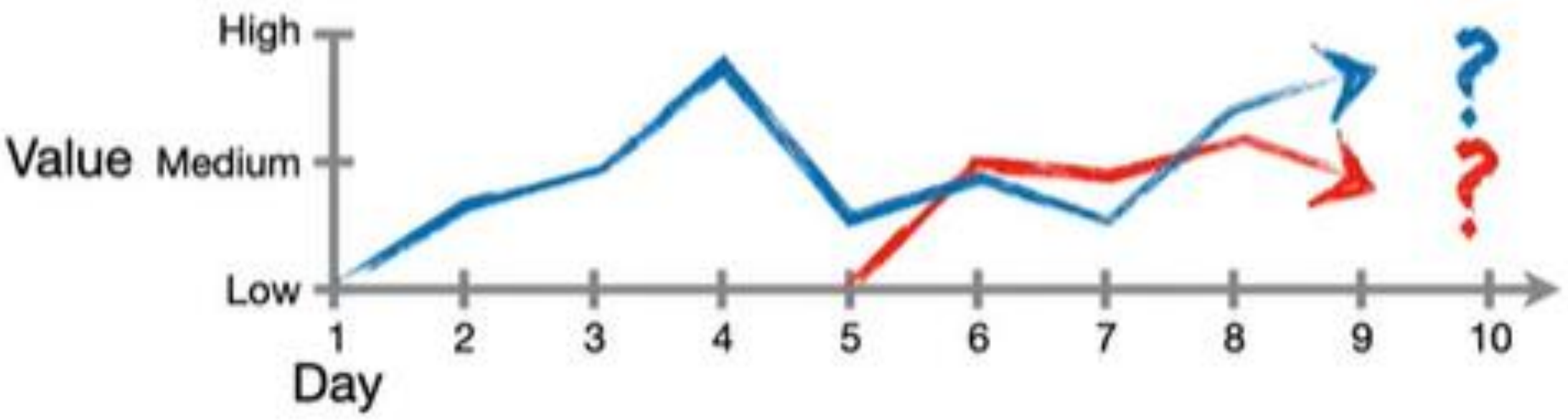
</s>

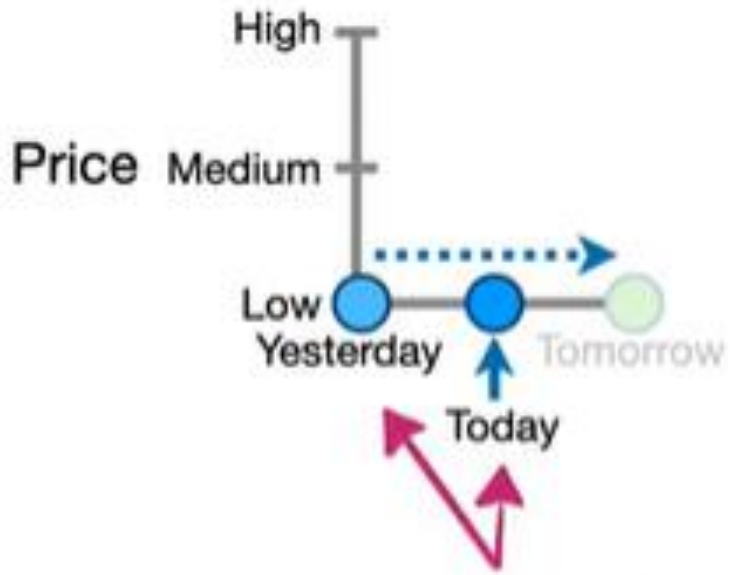
Consider an example: <s> John is driving a car

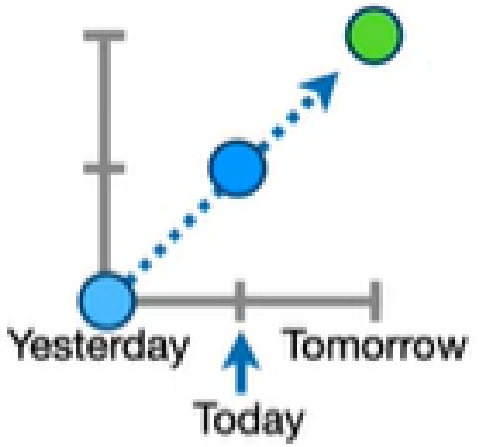
This hidden state can be considered as a "**summary**" of the entire sentence represented as a vector

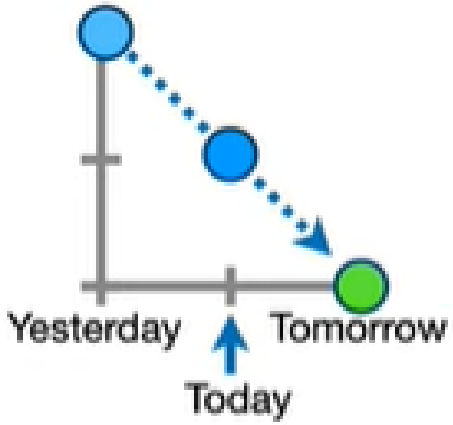


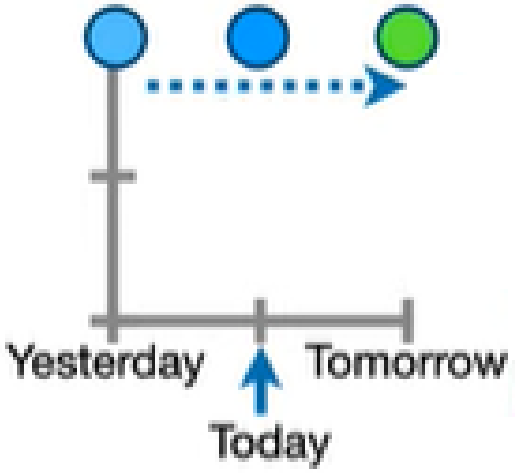
Problem – Predicting Stock Prices



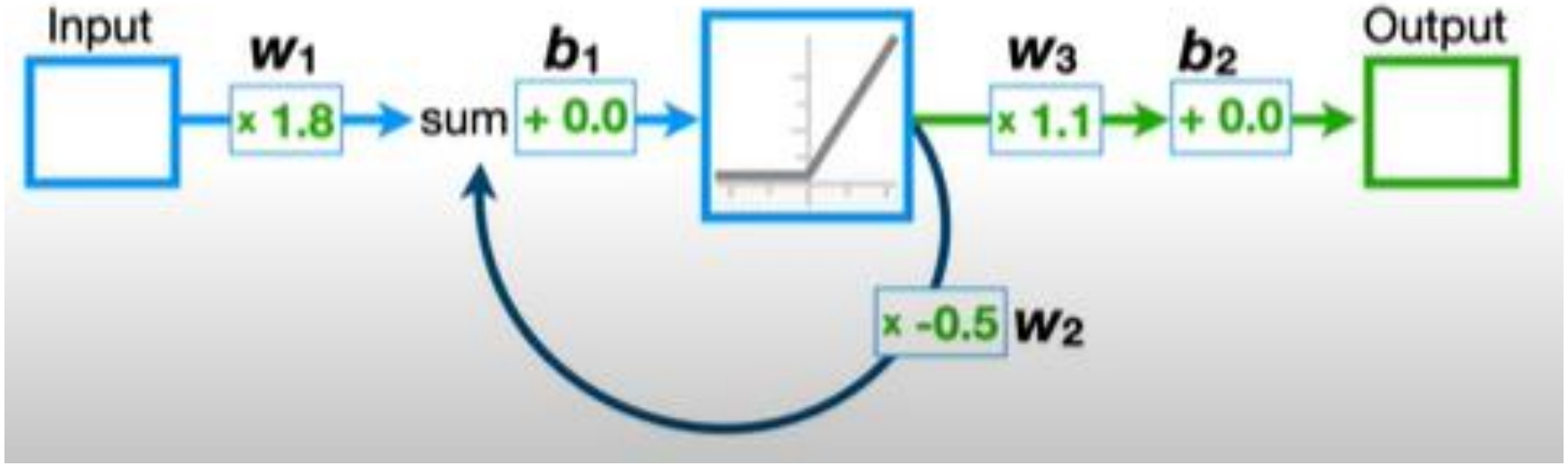

High
Price
Medium
Low
Yesterday
Tomorrow
Today


Yesterday
Tomorrow
Today

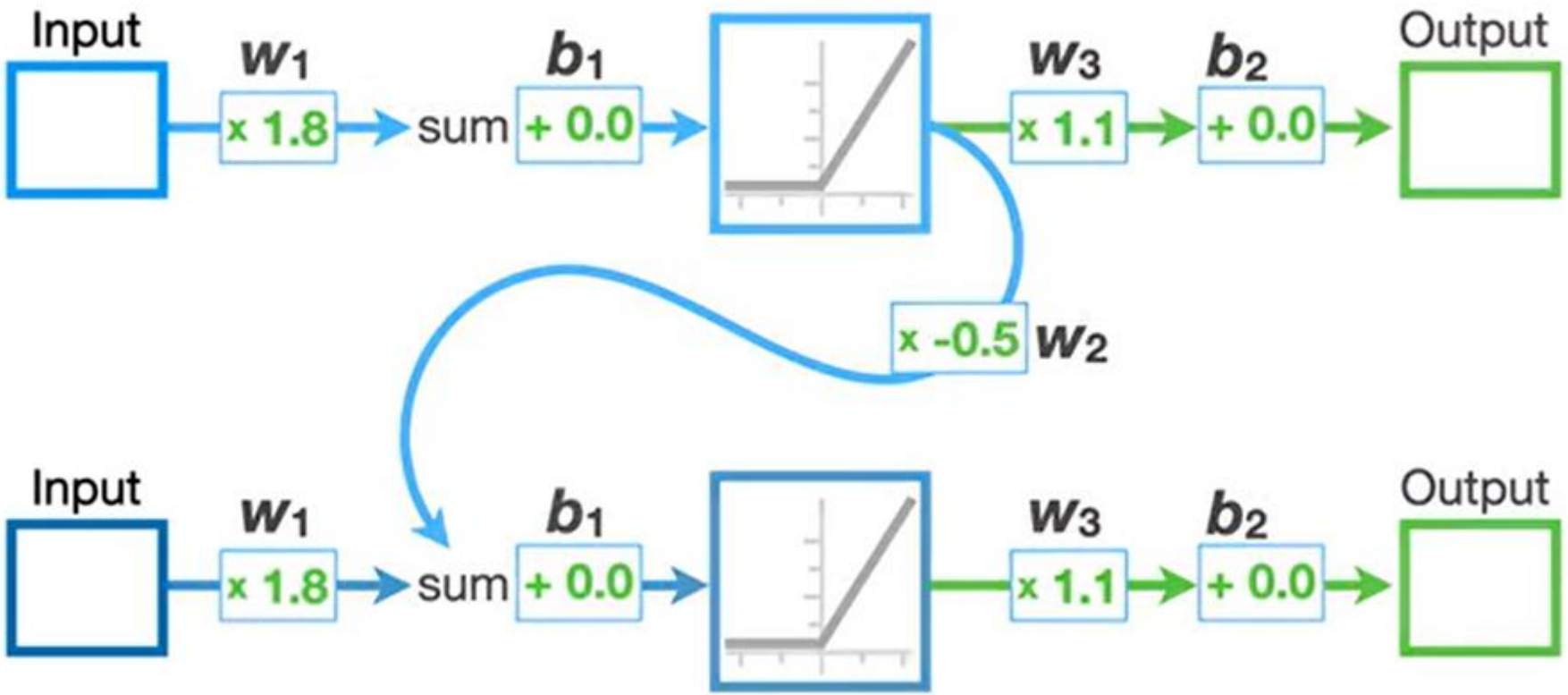

Yesterday
Tomorrow
Today


Yesterday
Tomorrow
Today

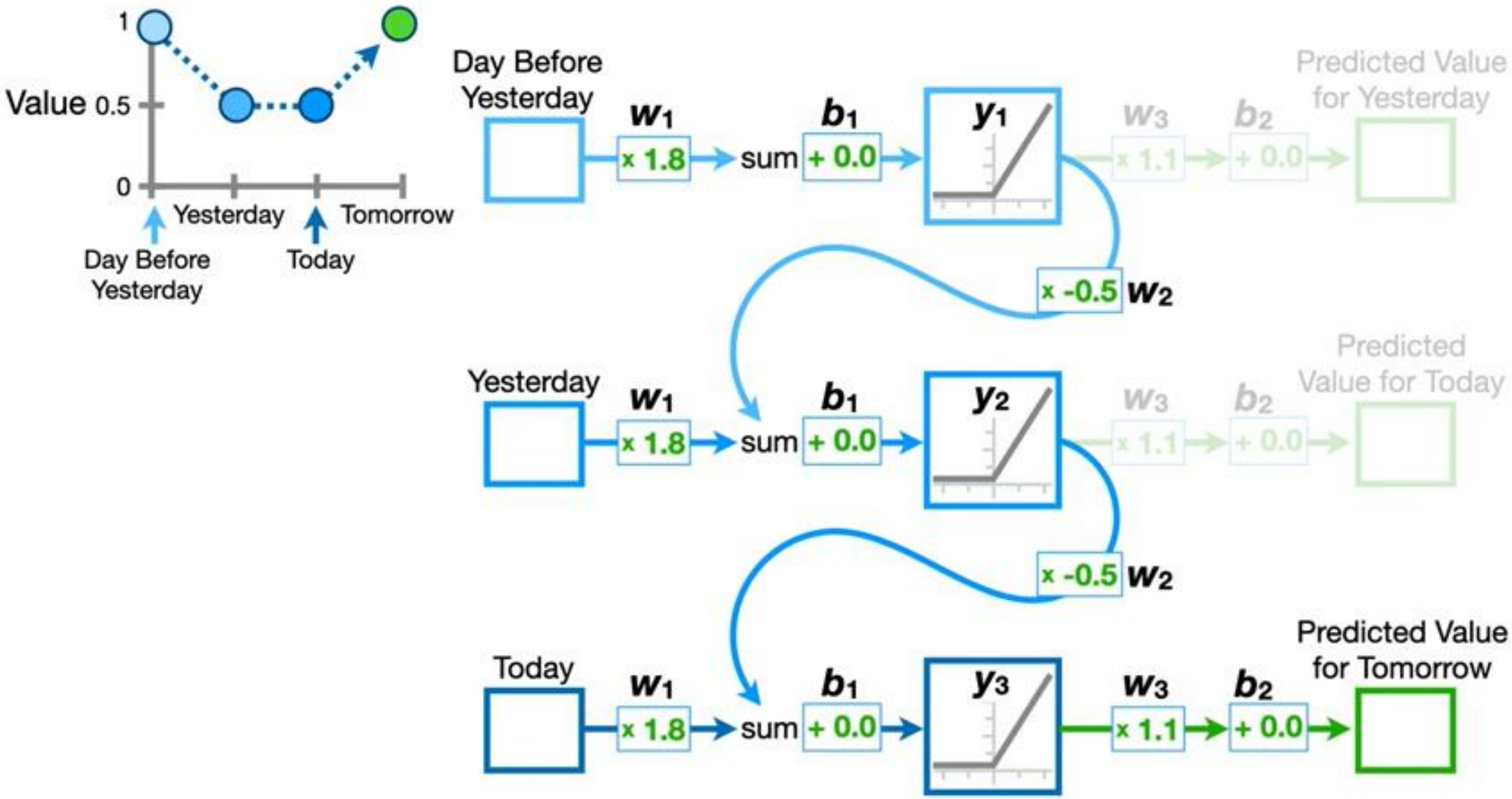
RNN - Solution



RNN - Solution



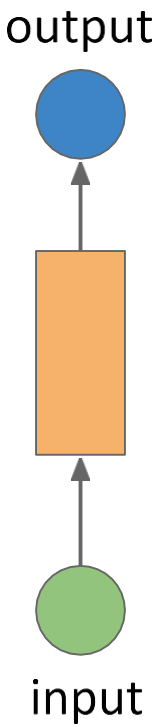
3 Inputs


1
Value 0.5
0
Yesterday
Tomorrow
Day Before Yesterday
Today
Day Before Yesterday
w1
x 1.8
sum
b1
+ 0.0
y1
w3
x 1.1
b2
+ 0.0
Predicted Value for Yesterday
x -0.5
w2
Yesterday
w1
x 1.8
sum
b1
+ 0.0
y2
w3
x 1.1
b2
+ 0.0
Predicted Value for Today
x -0.5
w2
Today
w1
x 1.8
sum
b1
+ 0.0
y3
w3
x 1.1
b2
+ 0.0
Predicted Value for Tomorrow

Loss computation

in recurrent neural networks

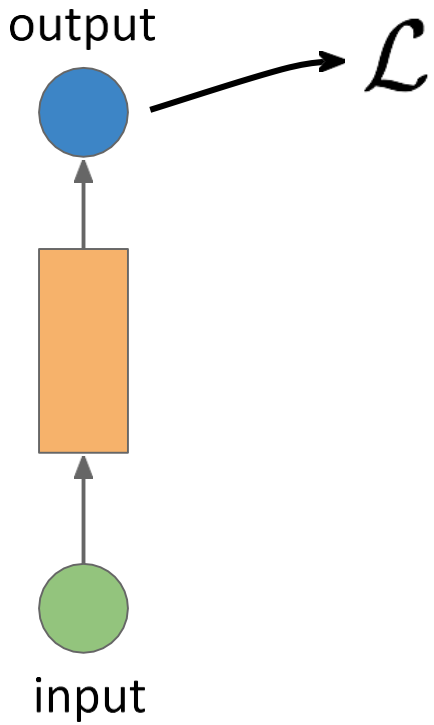
Loss Computation



Recall that in a feed forward network, we have a

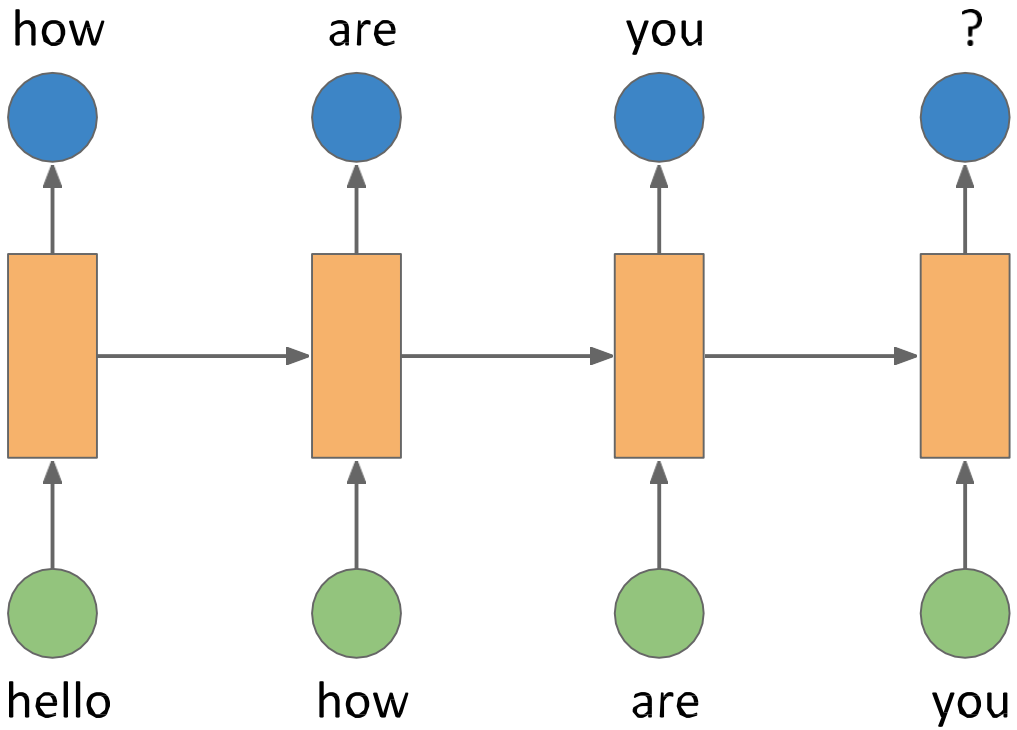
single output

Loss Computation



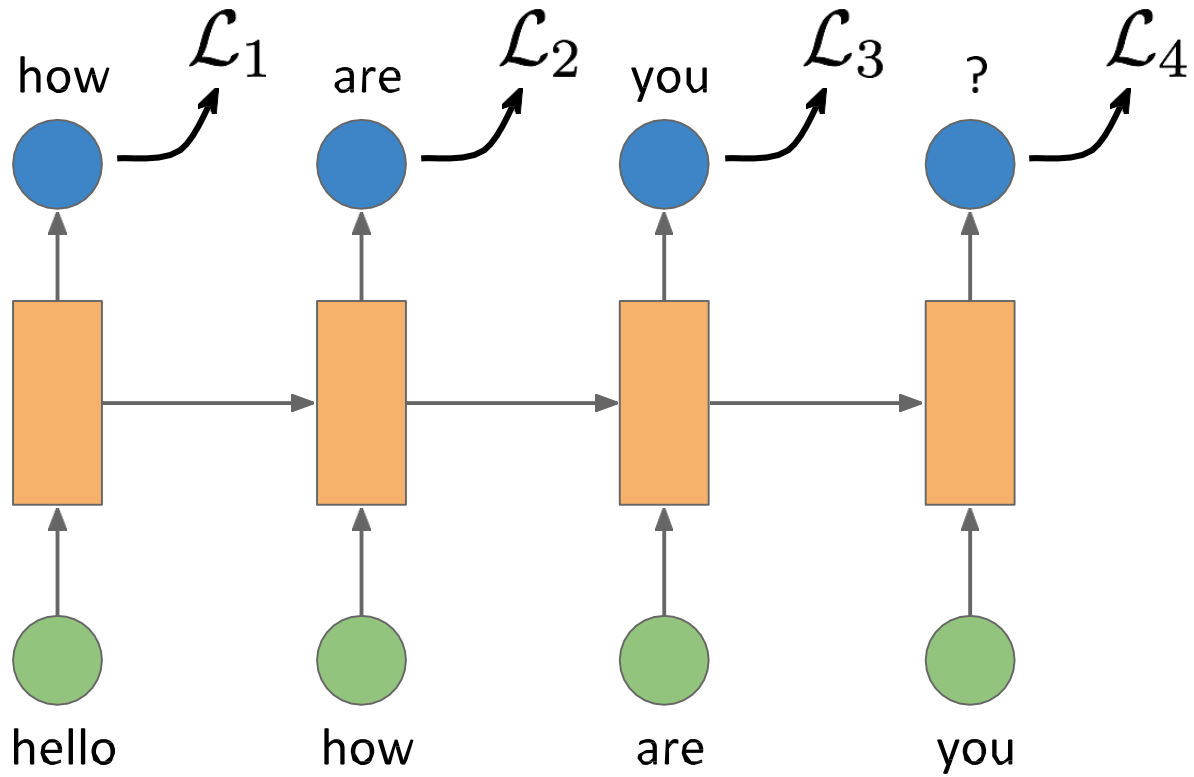
We compare this single output with the true label to get a loss value

Loss Computation



In the case of recurrent neural networks, we have an output per **timestep**

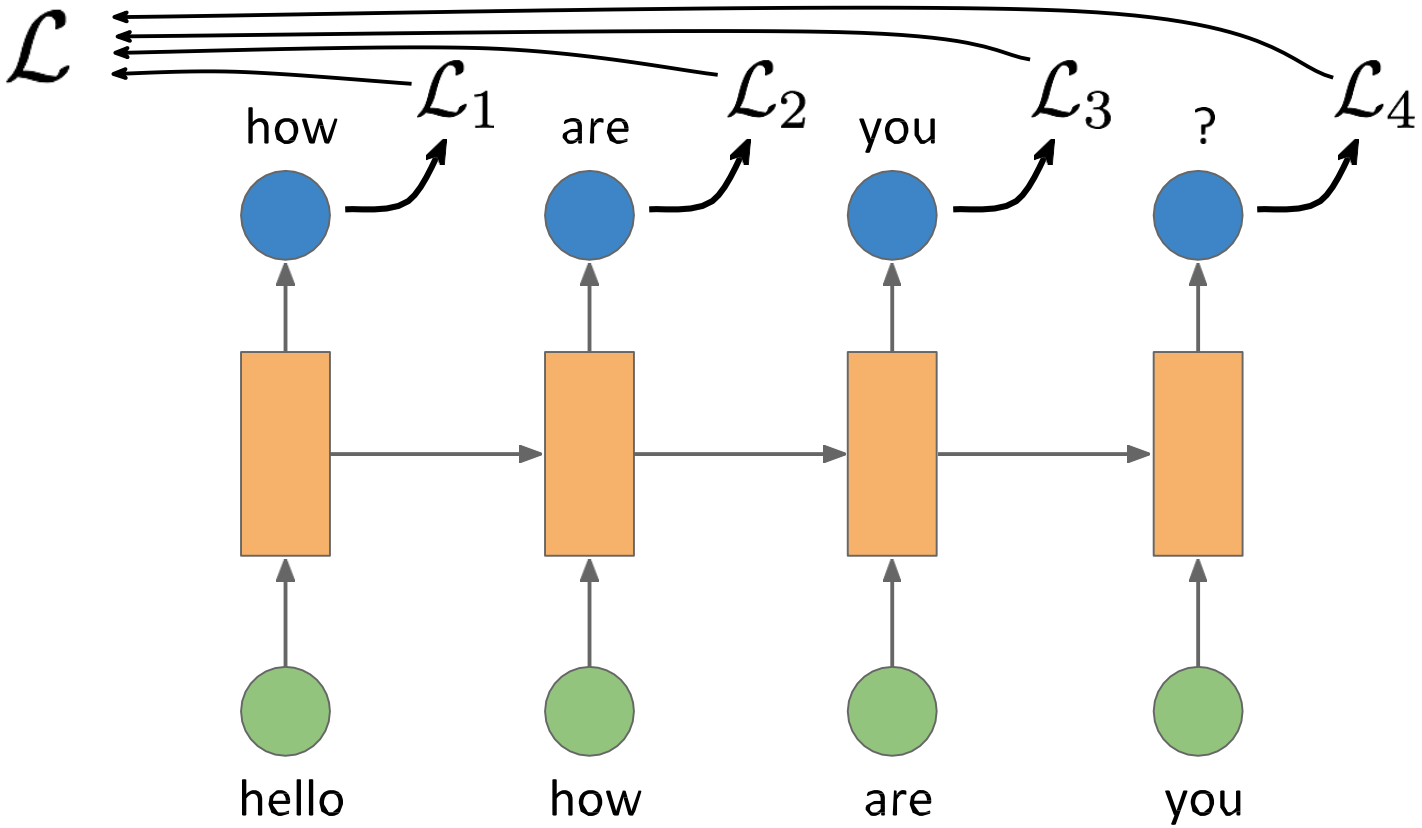
Loss Computation



Each of these outputs can be used to get

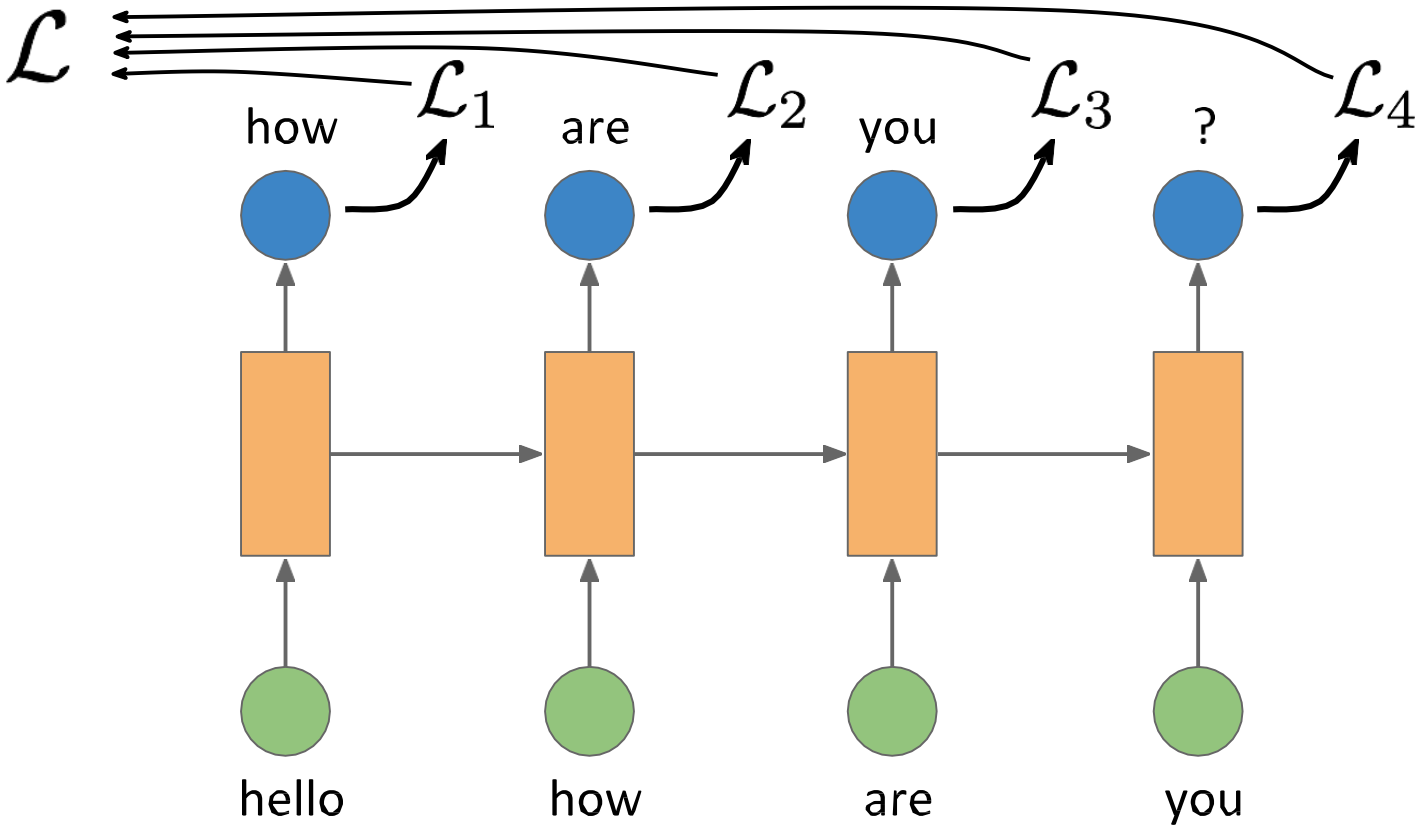
one loss per timestep

Loss Computation



Individual losses are still calculated as before - e.g. using cross entropy loss

Loss Computation

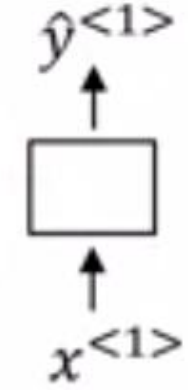


We *add all of these losses* together to get a single loss for our optimization algorithm

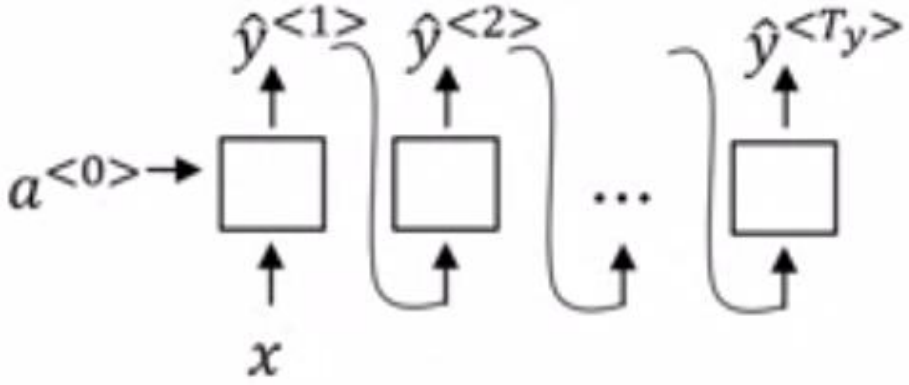
Applications of RNN

- Speech Recognition
- Language Modeling
- Machine Translation
- Image Captioning

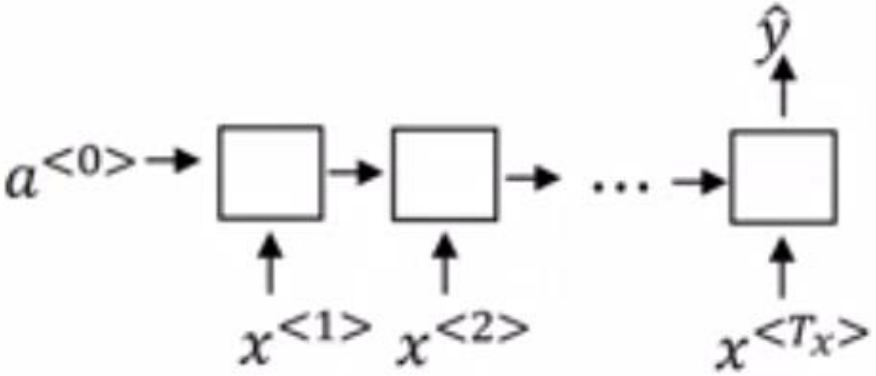
Different Types of RNNs



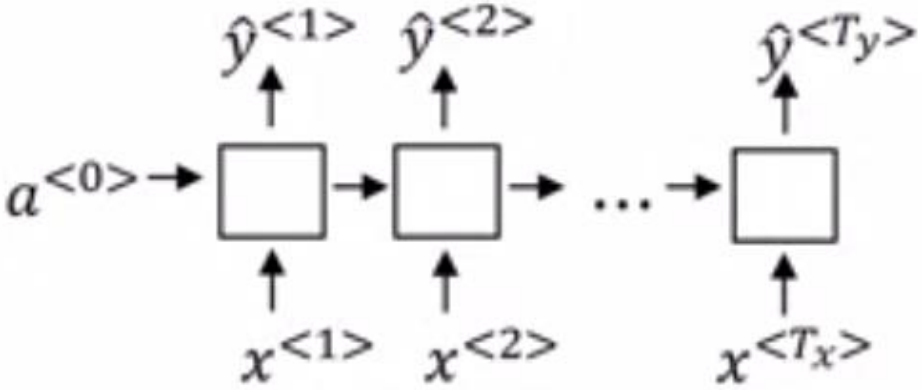
One to one



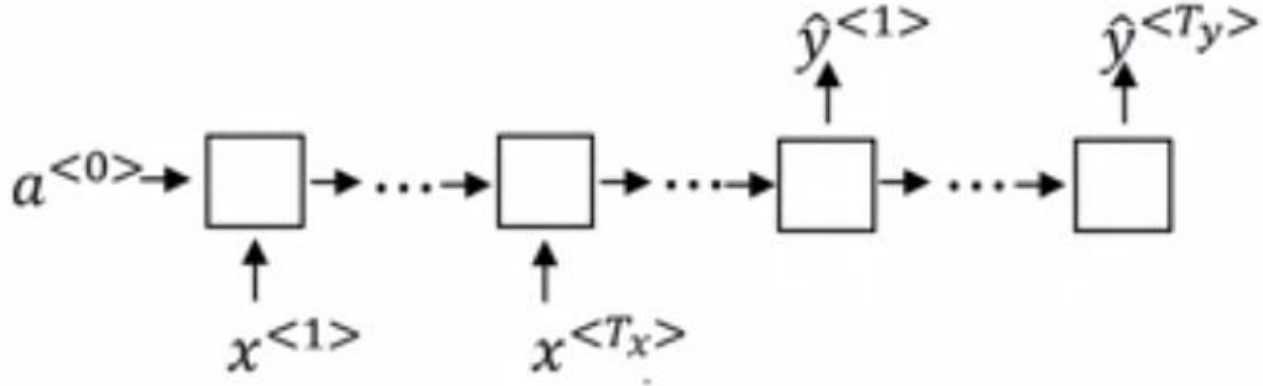
One to many



Many to one

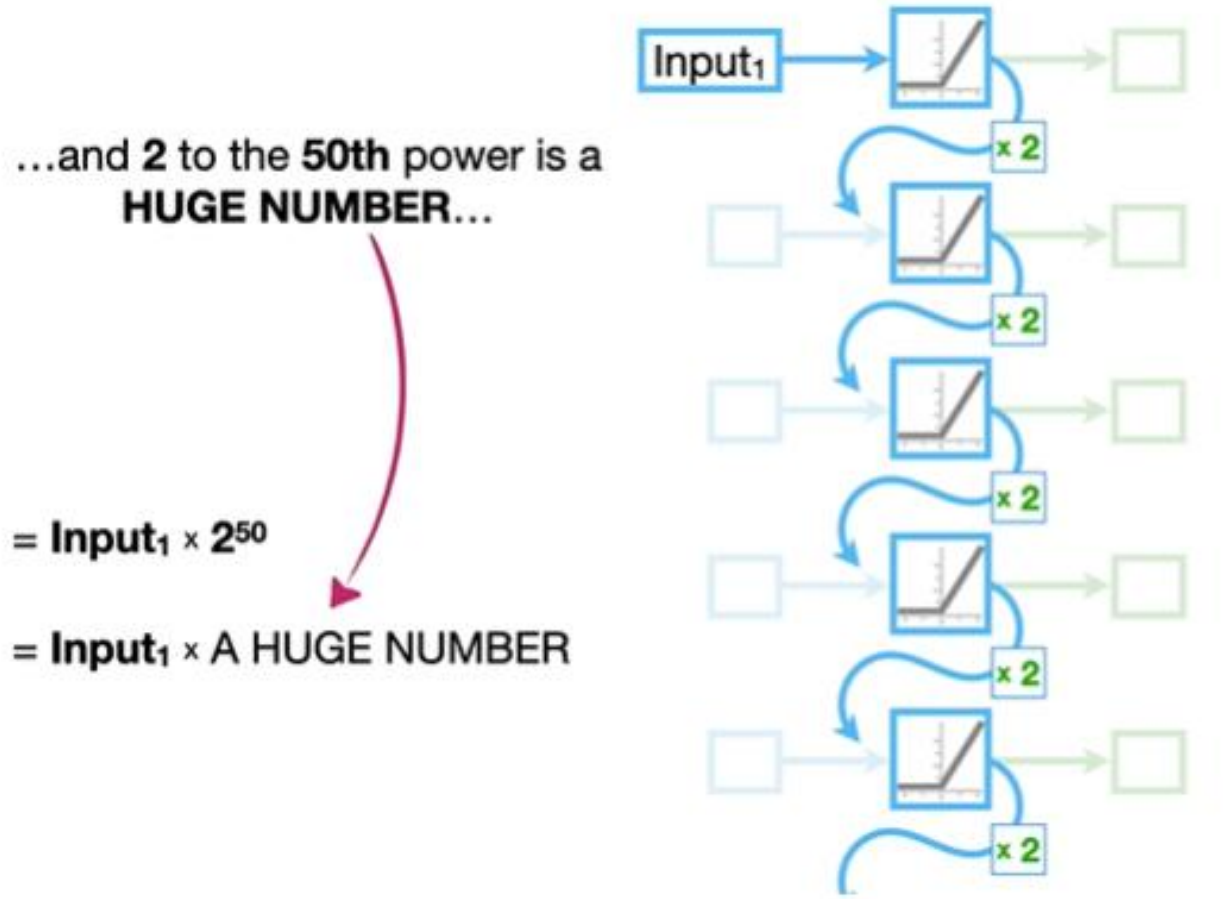


Many to many



Many to many

Problems in RNN

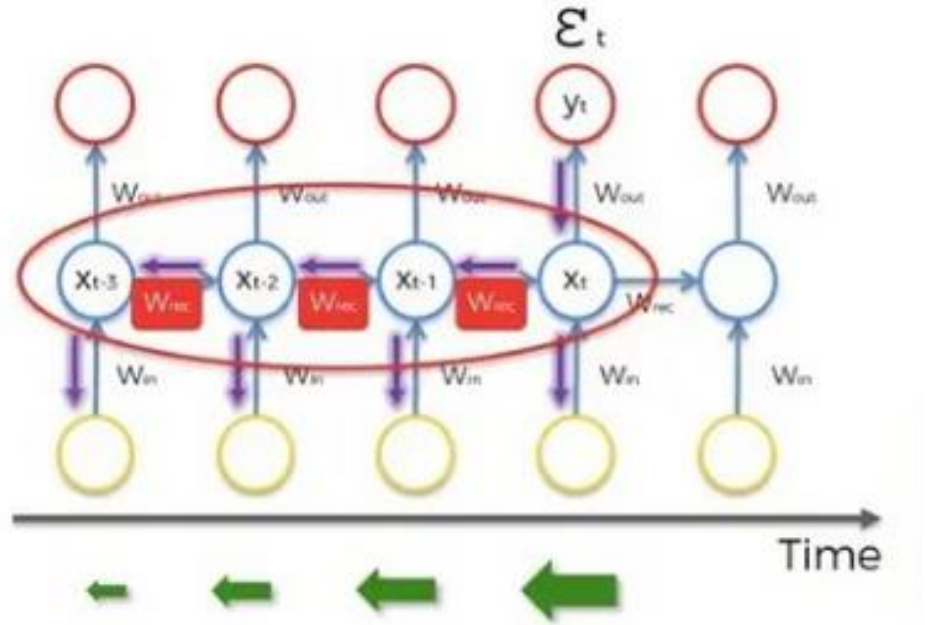


- Exploding Gradient Problem

Problems in RNN

Recurrent Neural Networks are hard to train because the gradients can **explode**, or **vanish**.

$= \textbf{Input}_1 \times w_2^{\text{Num. Unroll}}$



Vanishing Gradient

W_{rec} ~ small ⇨ Vanishing

W_{rec} ~ large ⇨ Exploding

- Vanishing Gradient Problem

Problem

- The clouds are in the ……. (sky)

- I grew up in France w w w w w w w w w w ww I speak fluent …… (French)

Problem

- The clouds are in the ……. (sky)
- I grew up in France w w w w w w w w w w ww I speak fluent …… (French)

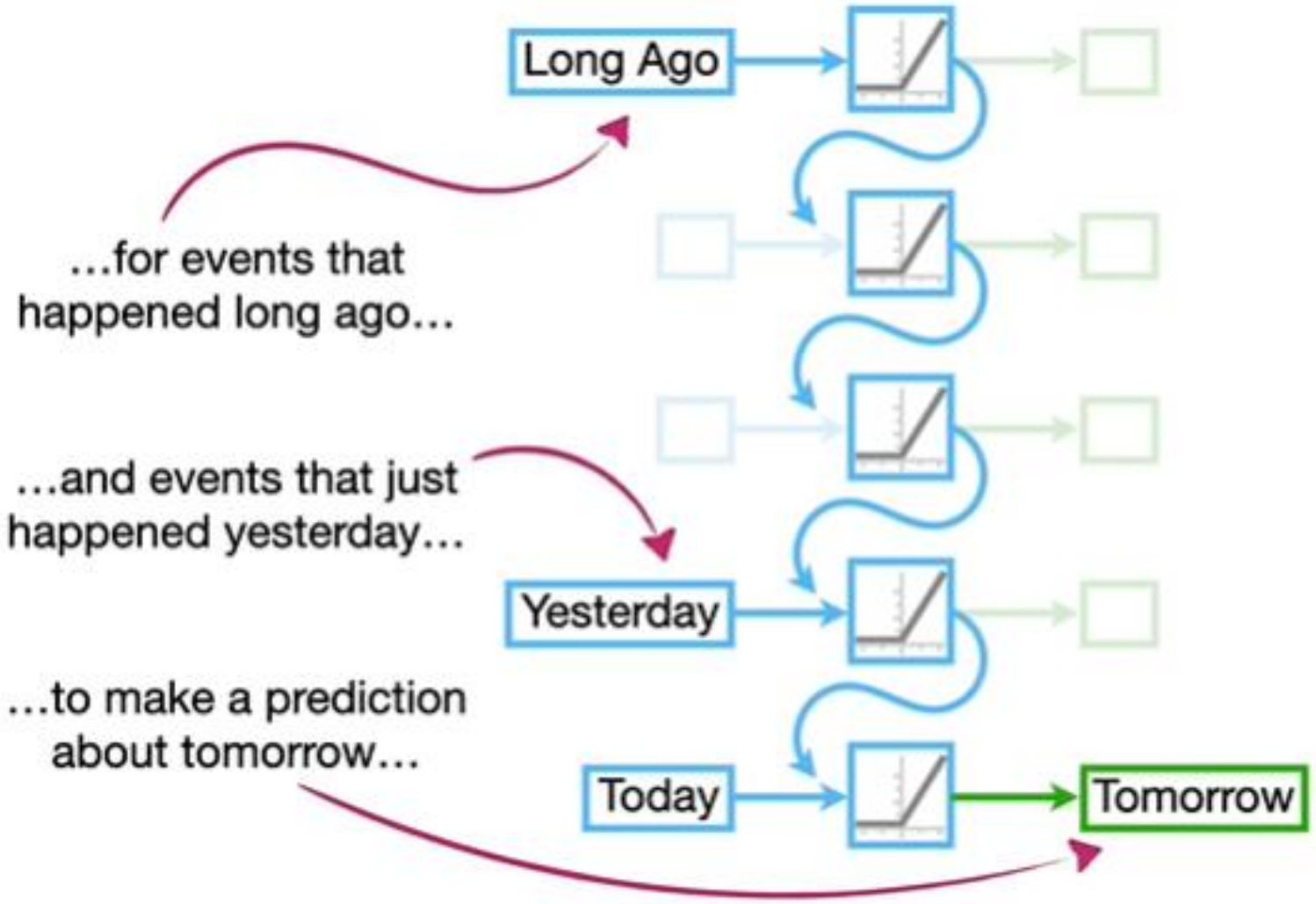
Solution

Long Short Term Memory (LSTM)

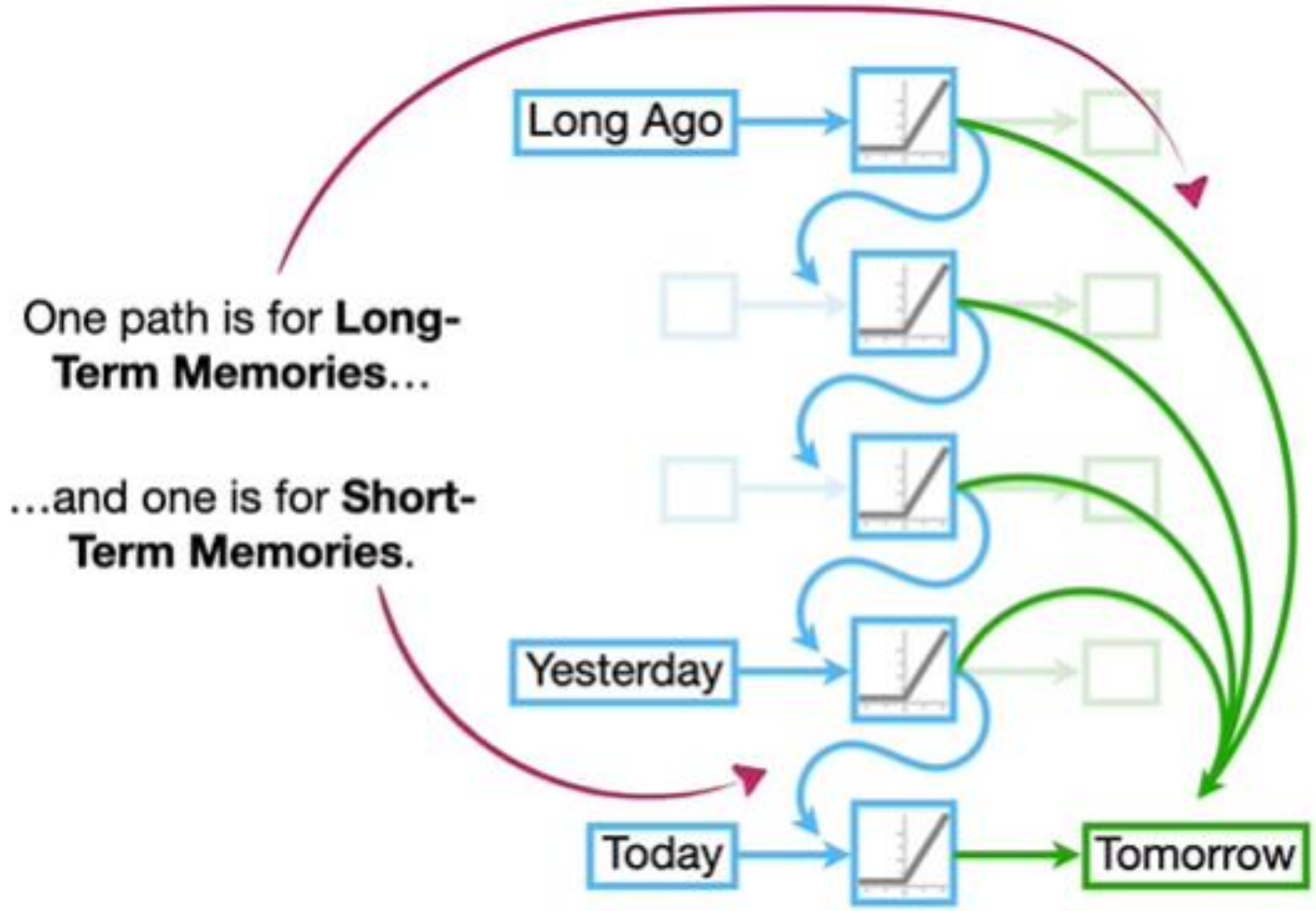
What is LSTM?

- Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) architecture designed to address the vanishing gradient problem
- LSTM networks are widely used in various natural language processing (NLP) tasks, including text classification, machine translation, speech recognition, and sentiment analysis.
- The key feature of LSTM is its ability to capture long-term dependencies in sequential data by incorporating memory cells and gating mechanisms

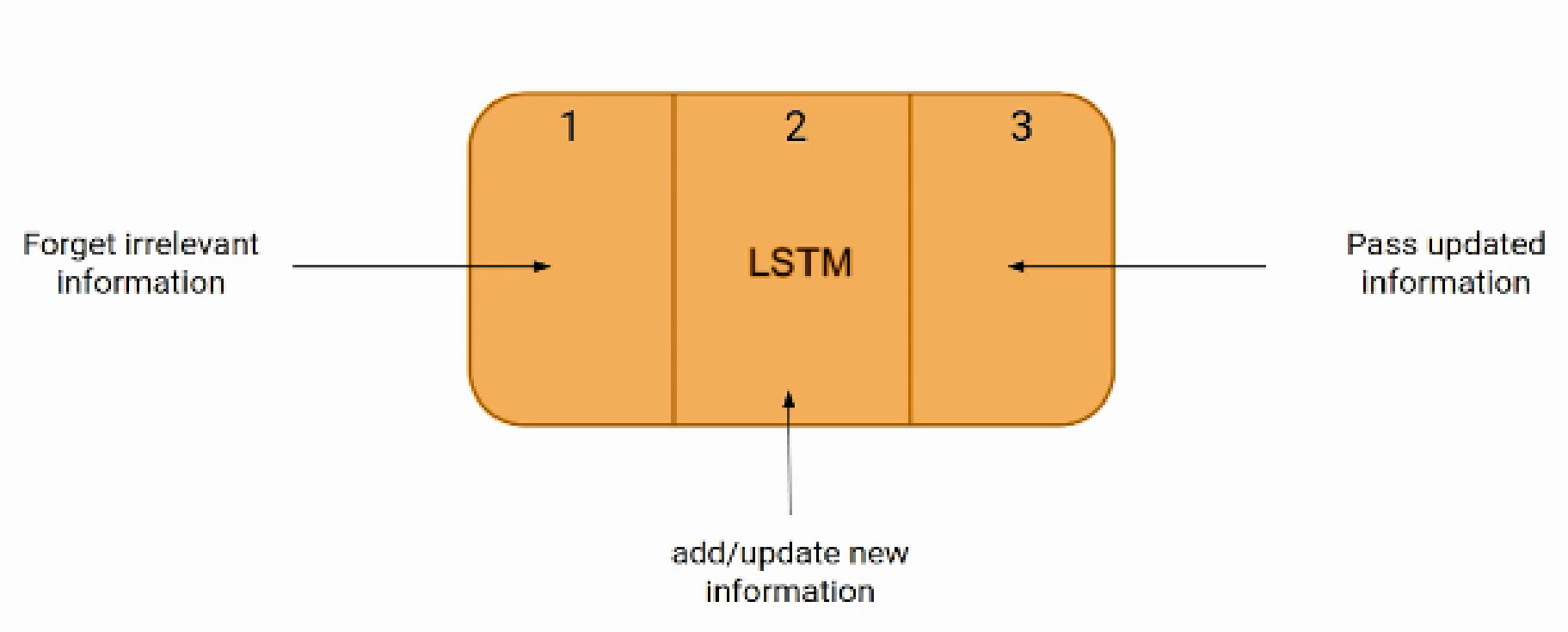
LSTM


Long Ago
…for events that happened long ago…
…and events that just happened yesterday…
Yesterday
…to make a prediction about tomorrow…
Today
Tomorrow

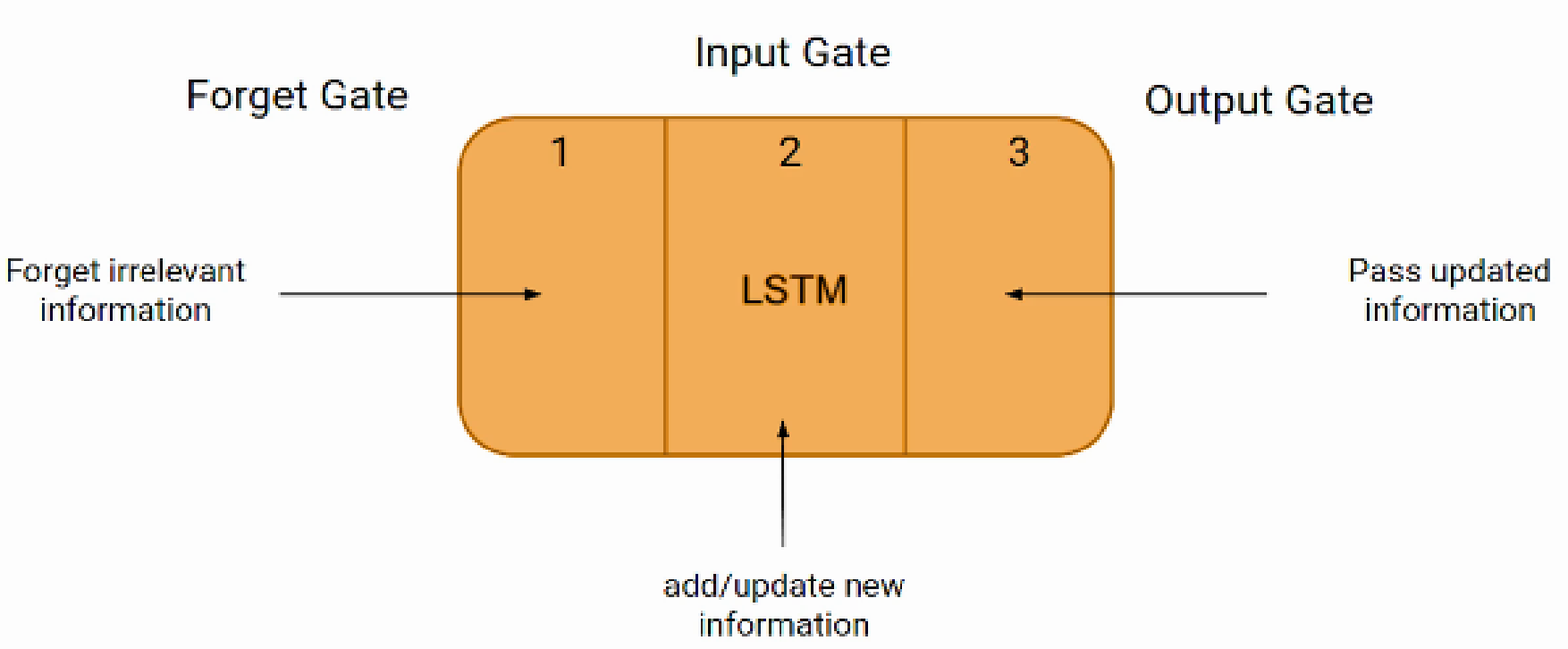
LSTM


One path is for **Long-Term Memories**...
...and one is for **Short-Term Memories**.
Long Ago
Yesterday
Today
Tomorrow

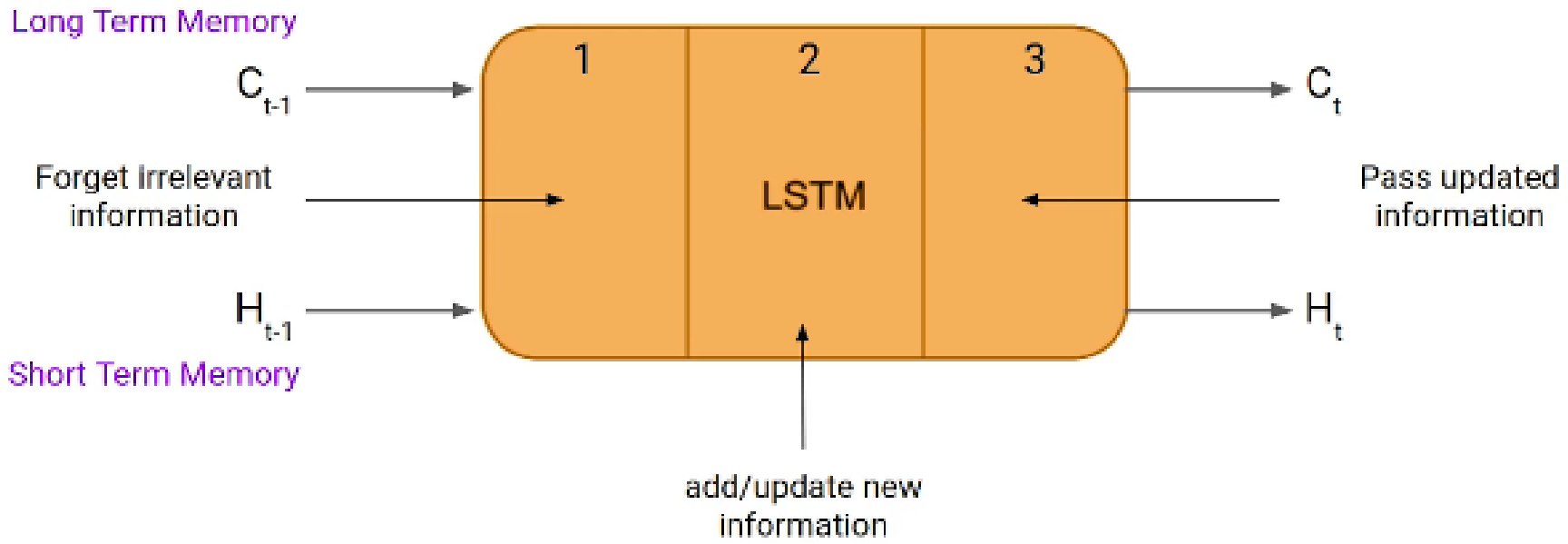
LSTM Architecture



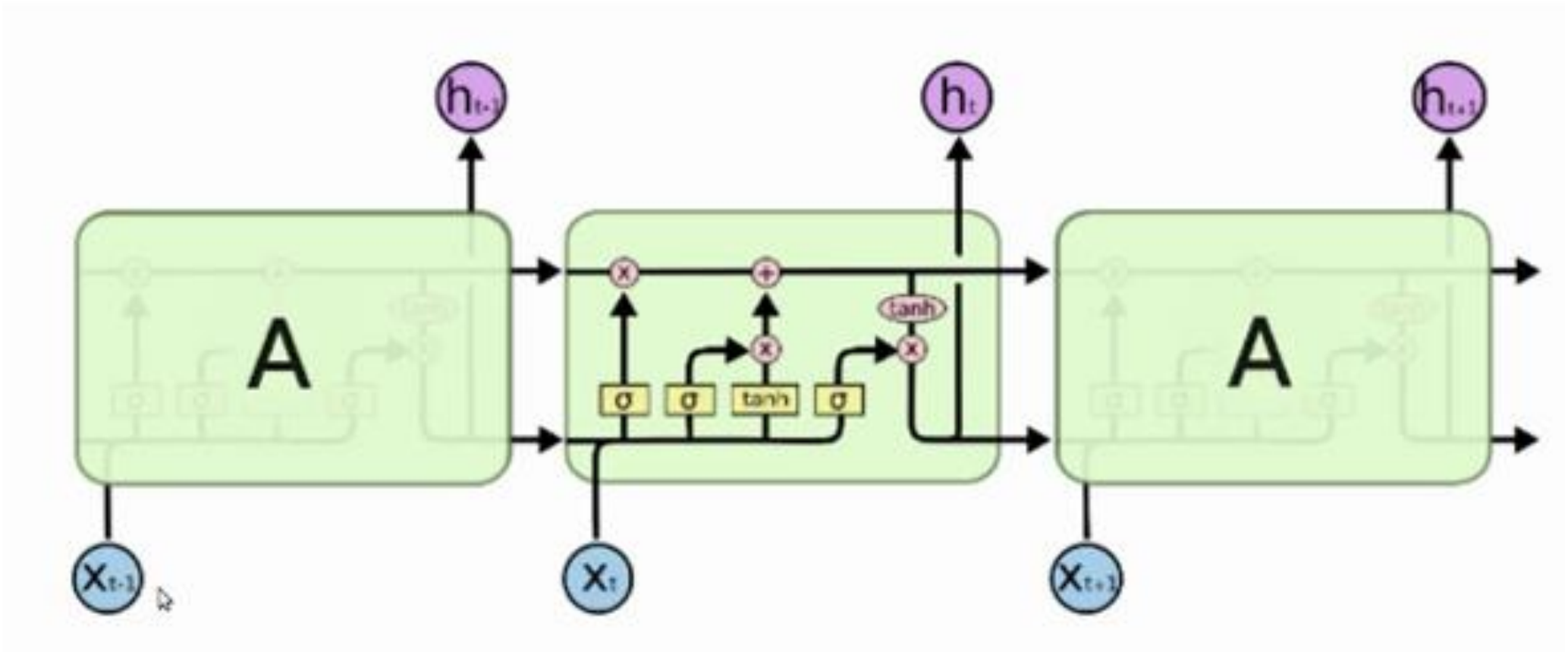
LSTM Architecture



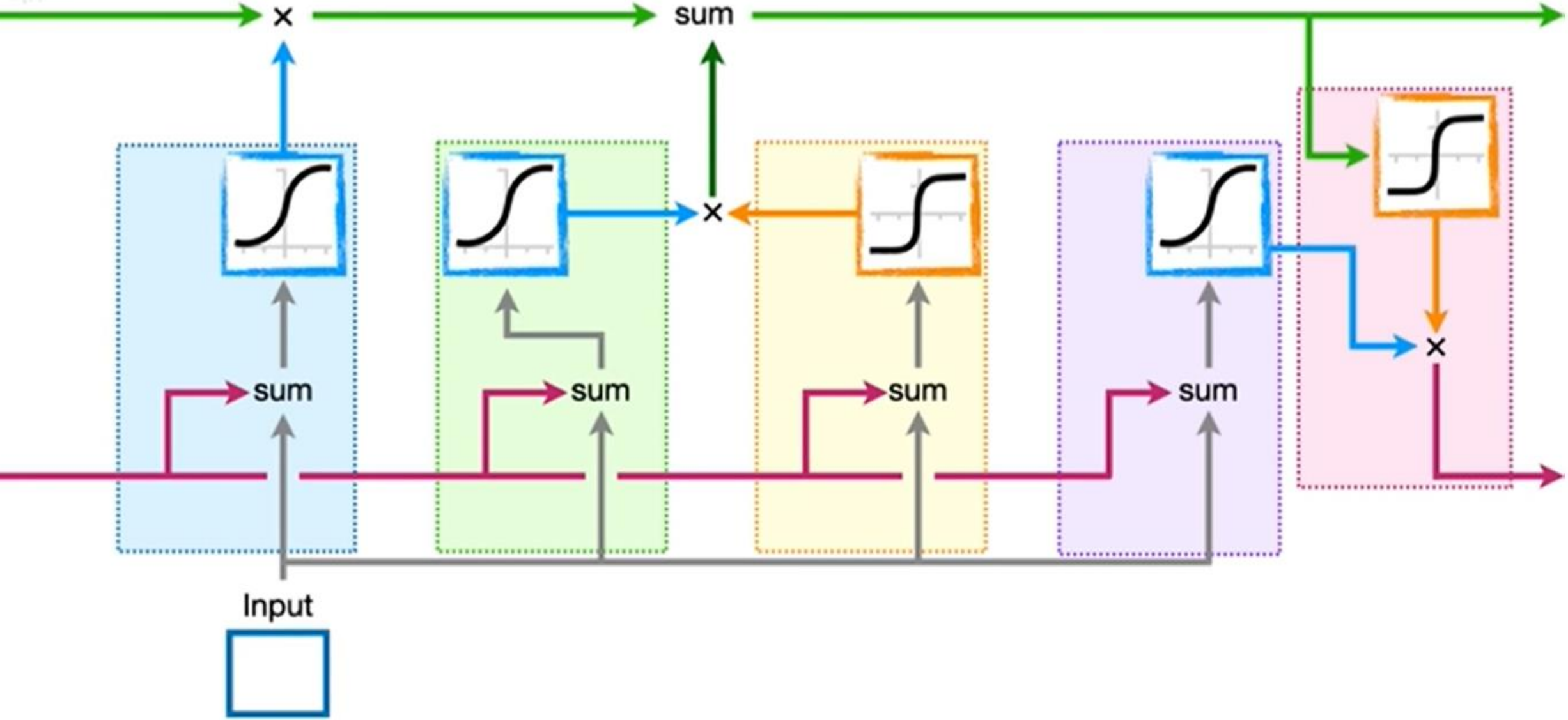
LSTM Architecture



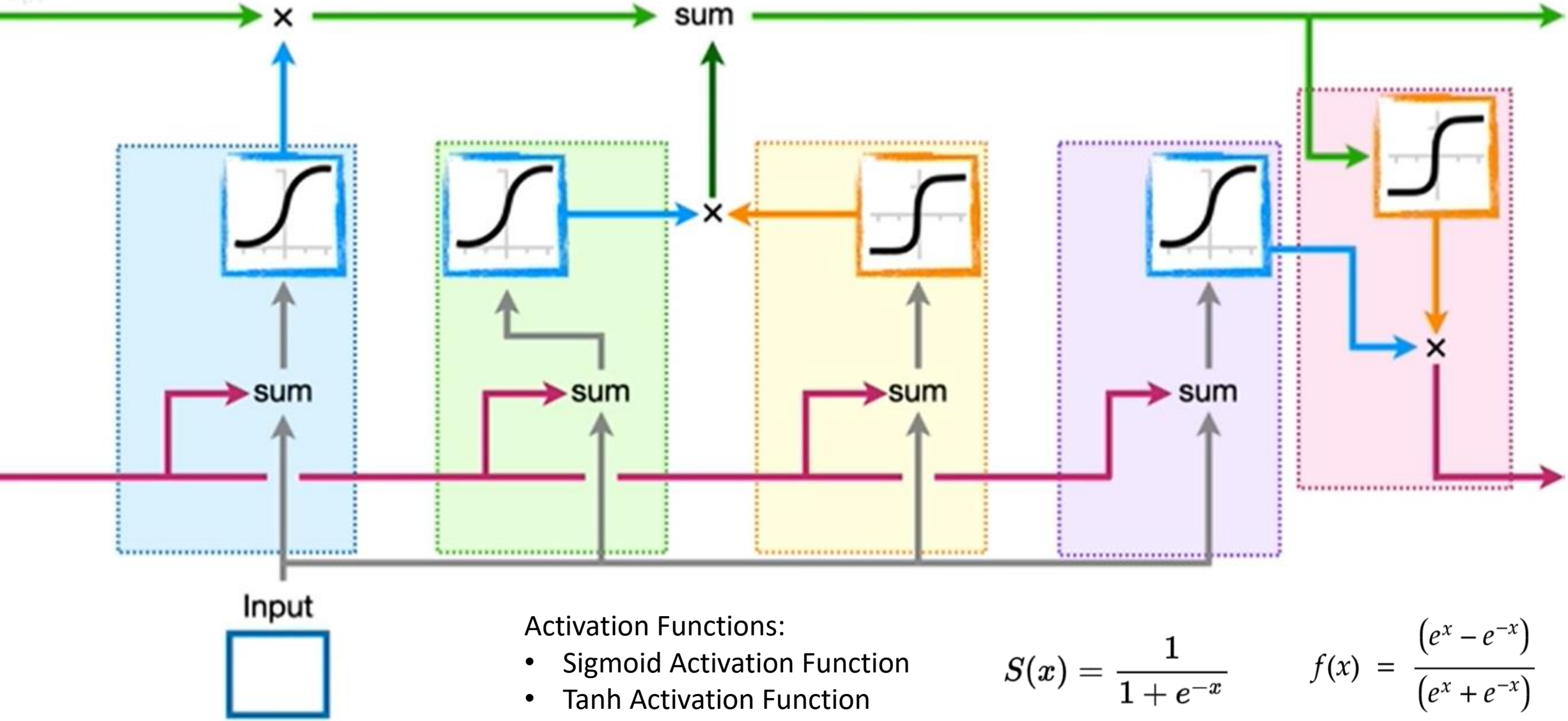
LSTM Architecture



…**Long Short-Term Memory** is based on a much more complicated unit.



…Long Short-Term Memory is based on a much more complicated unit.



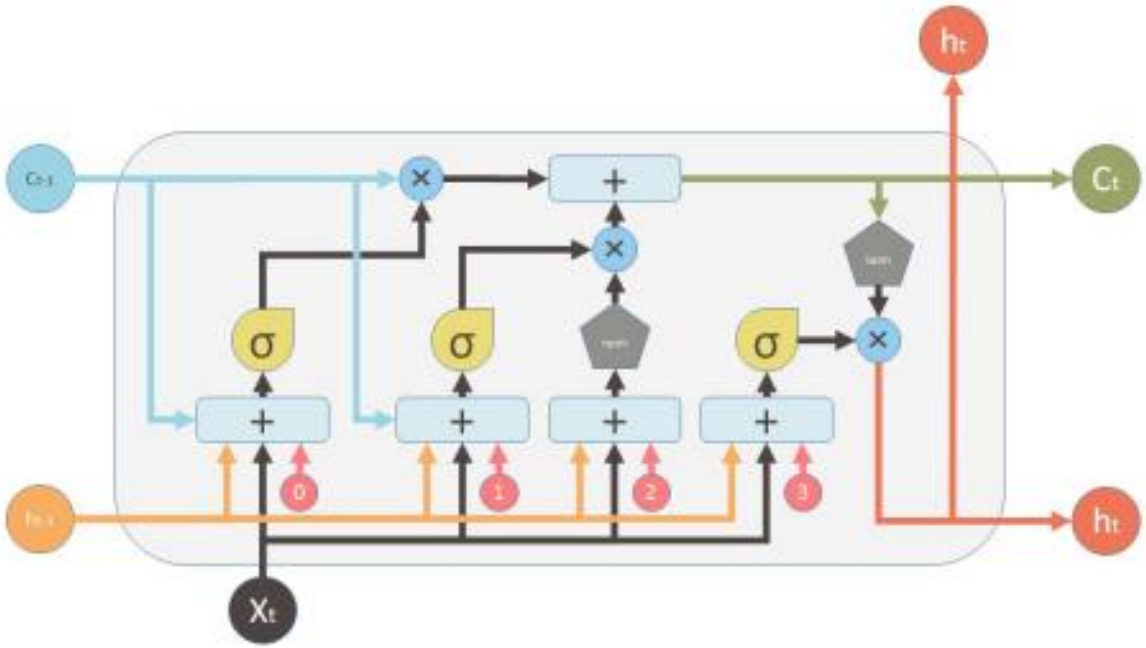
Activation Functions:

- Sigmoid Activation Function
- Tanh Activation Function

$$S(x) = \frac{1}{1 + e^{-x}}$$

$$f(x) = \frac{\left(e^{x} - e^{-x}\right)}{\left(e^{x} + e^{-x}\right)}$$

LSTM Memory Cell



Inputs:

- X_t Input vector
- C_{t-1} Memory from previous block
- h_{t-1} Output of previous block

outputs:

- C_t Memory from current block
- h_t Output of current block

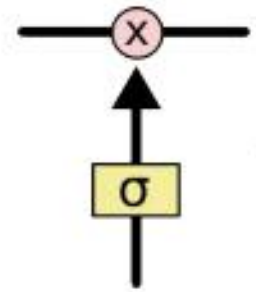
Nonlinearities:

- σ Sigmoid
- tanh Hyperbolic tangent

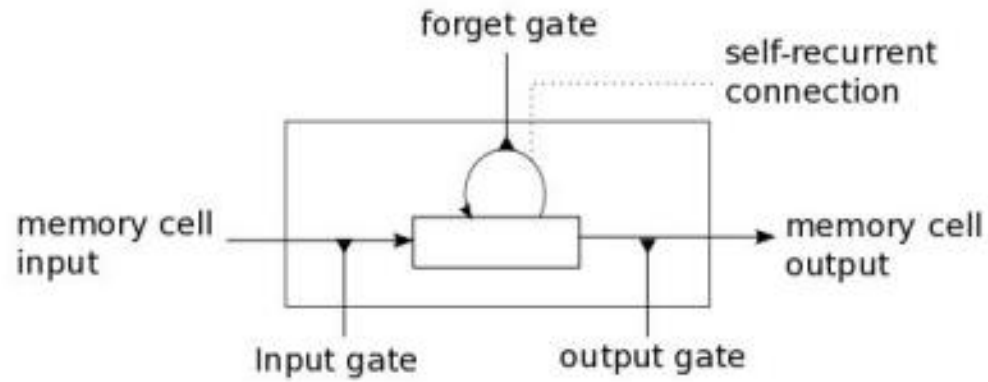
Bias: 0

Vector operations:

- × Element-wise multiplication
- \+ Element-wise Summation / Concatenation



Gate (sigmoid layer followed by pointwise multiplication)



Simplified schematic for reference

Gates

- Forget Gate:

Determines which information from the prior step is needed.

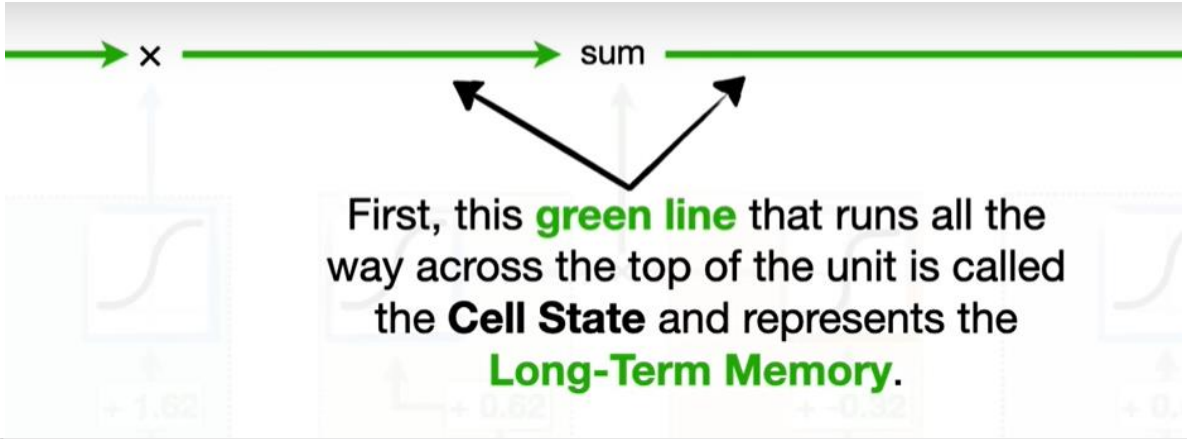
- Input Gate:

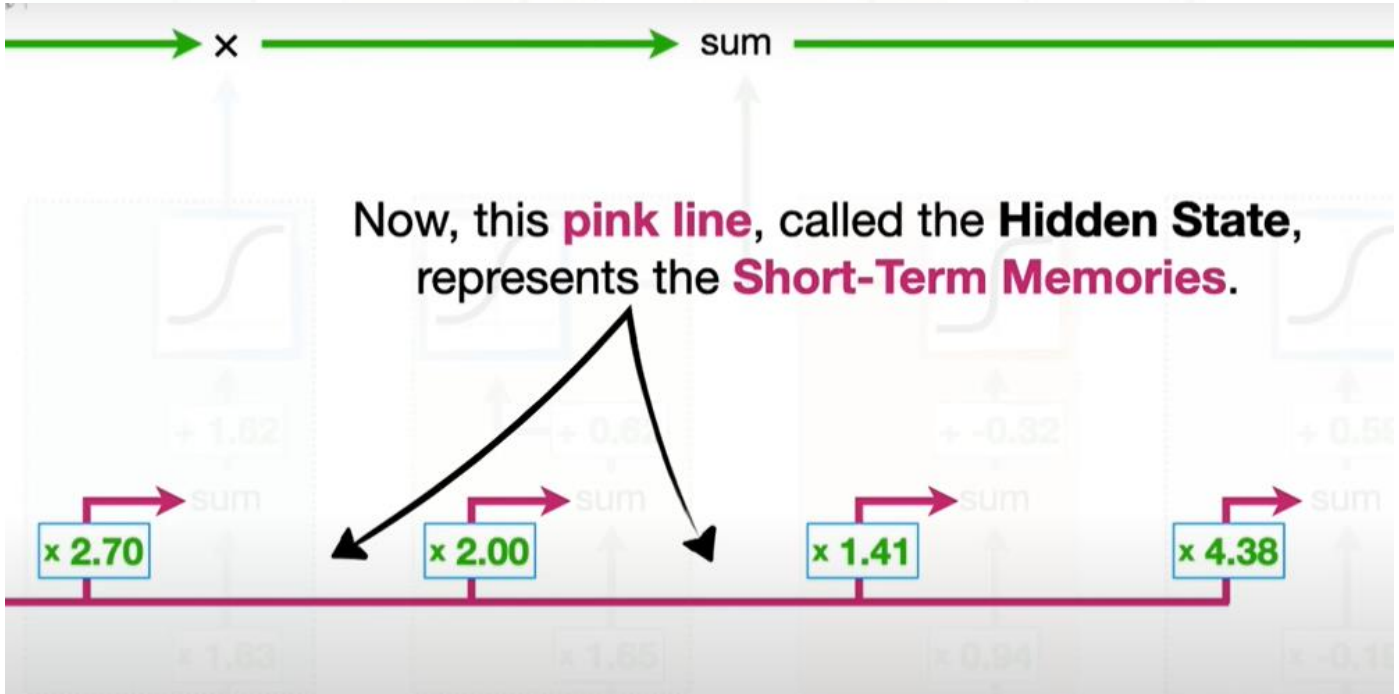
Determines what relevant information can be added from the current step.

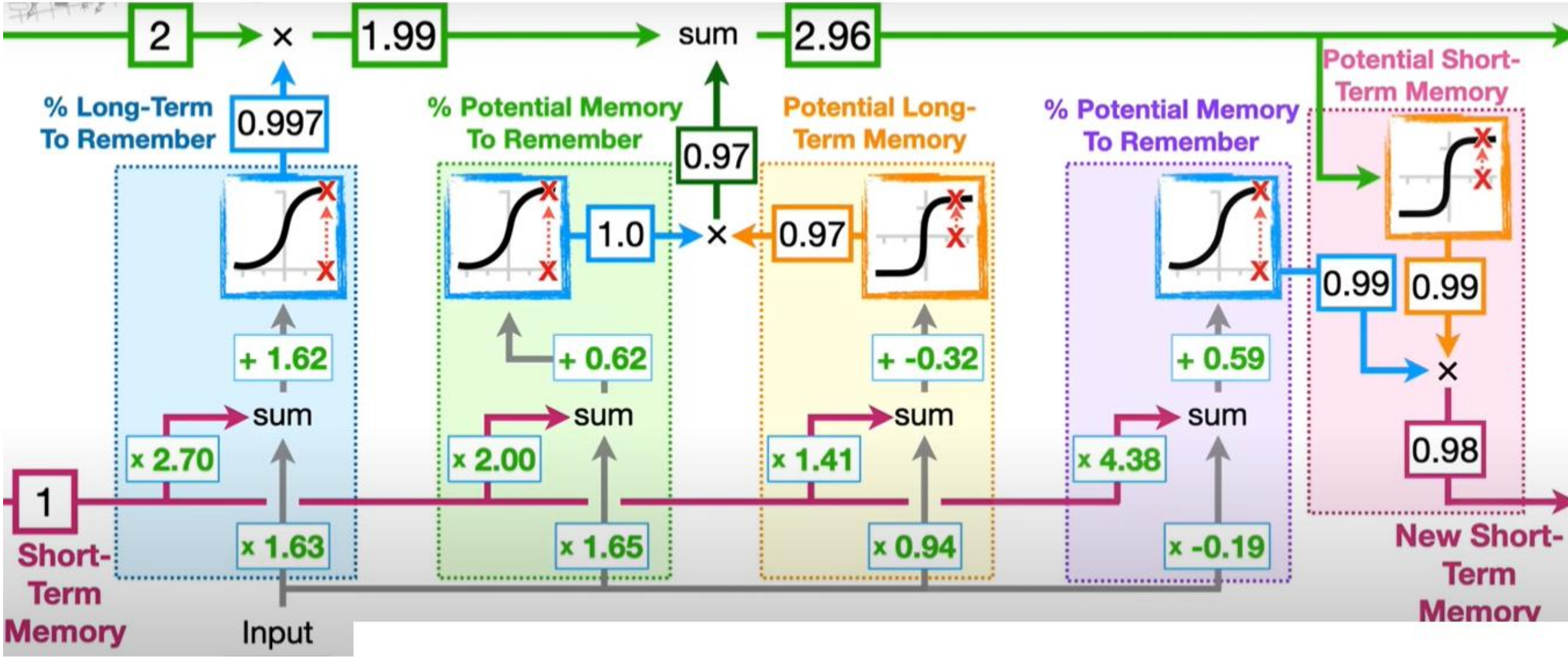
- Output Gate:

Finalize the next hidden state.

LSTM Calculations






2
×
1.99
sum
2.96
% Long-Term To Remember
0.997
% Potential Memory To Remember
0.97
Potential Long-Term Memory
% Potential Memory To Remember
Potential Short-Term Memory
1.0
×
0.97
0.99
0.99
+ 1.62
+ 0.62
+ -0.32
+ 0.59
sum
sum
sum
sum
×
x 2.70
x 2.00
x 1.41
x 4.38
0.98
1
x 1.63
x 1.65
x 0.94
x -0.19
Short-Term Memory
Input
New Short-Term Memory

Links

Introduction-to-long-short-term-memory-lstm

StatQuest - LSTM