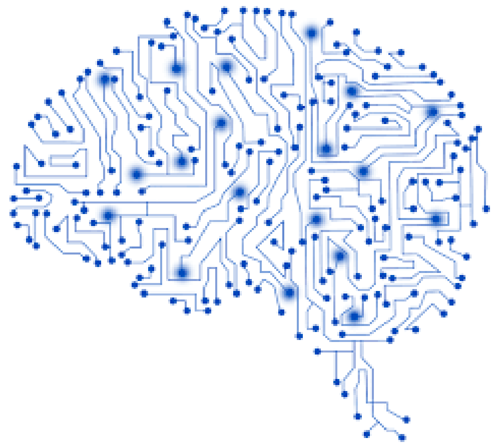
University of Minho
School of Engineering



Recurrent Neural Networks

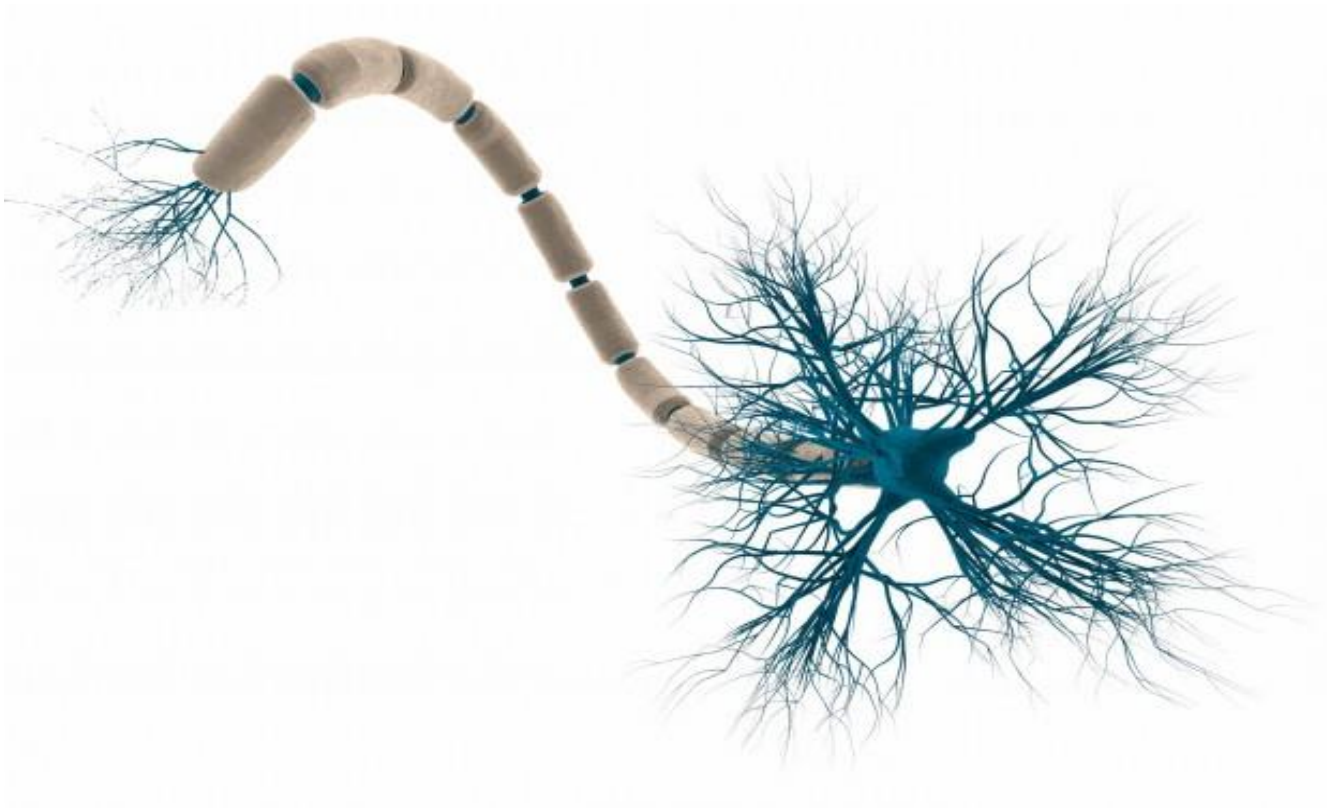
Connective Systems and Classifiers

Perfil ML:FA @ MiEI/4º ano - 2º Semestre
Bruno Fernandes, Victor Alves, Cesar Analide

12/03/2020

Contents

Neurons

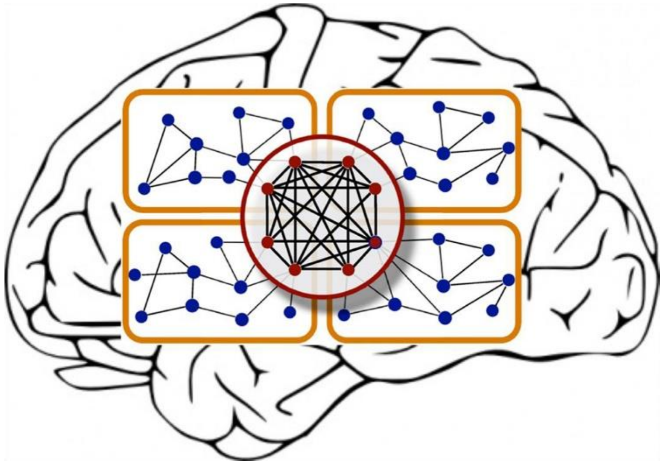


You may want to watch:
https://www.youtube.com/watch?v=3JQ3hYko51Y

Artificial Neural Networks

Definition

- An Artificial Neural Network (ANN) is a computational system based on connections for problem solving
- An ANN is conceived as a simplified model of the central nervous system of human beings!
- ANNs are defined by a interconnected structure of computational units, called neurons, with learning abilities

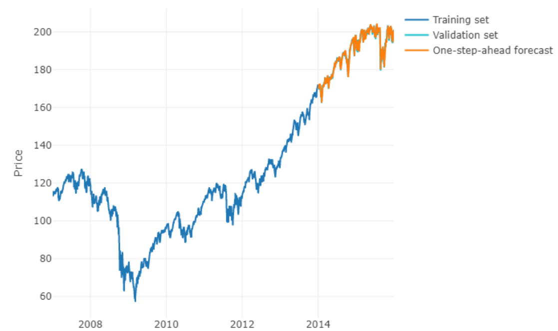


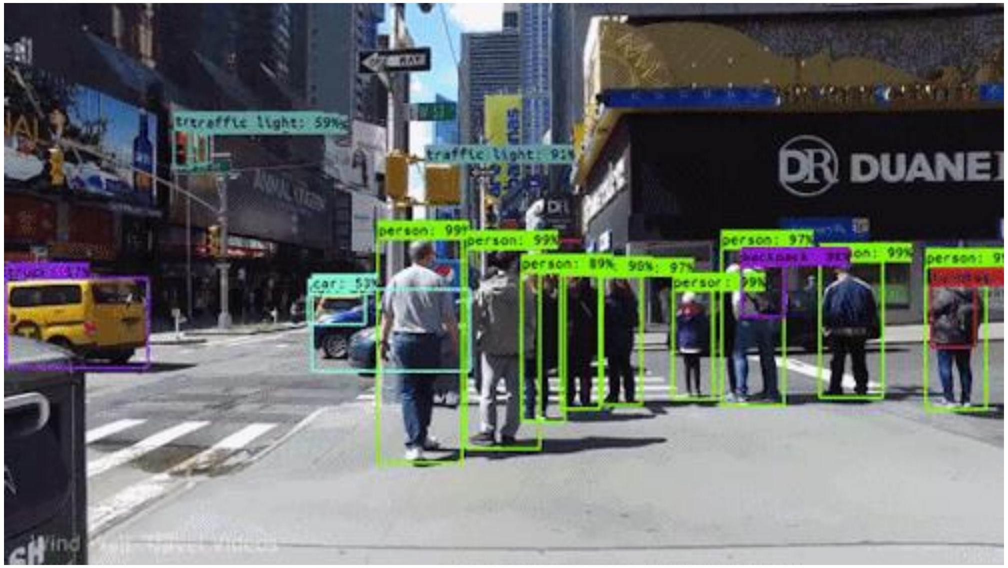
Artificial Neural Networks

Examples

ANNs are being used for:

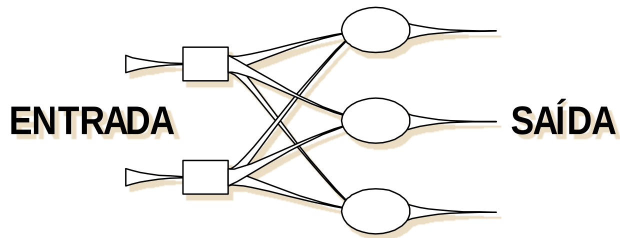
- Fraud Detection
- Audio recognition
- Text-to-speech
- Text translation
- Image Classification
- Object Detection
- Time Series
- …

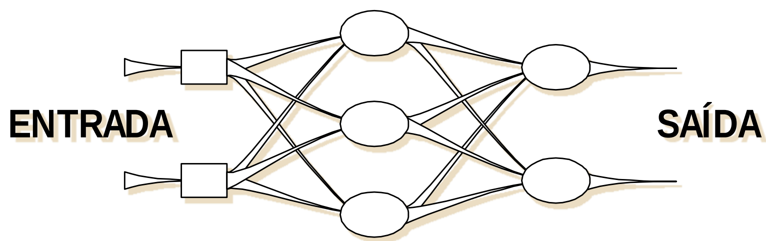


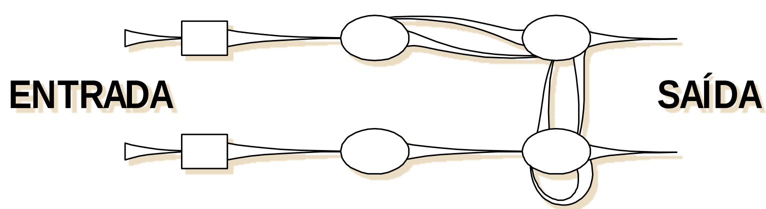


Artificial Neural Networks

Architectures



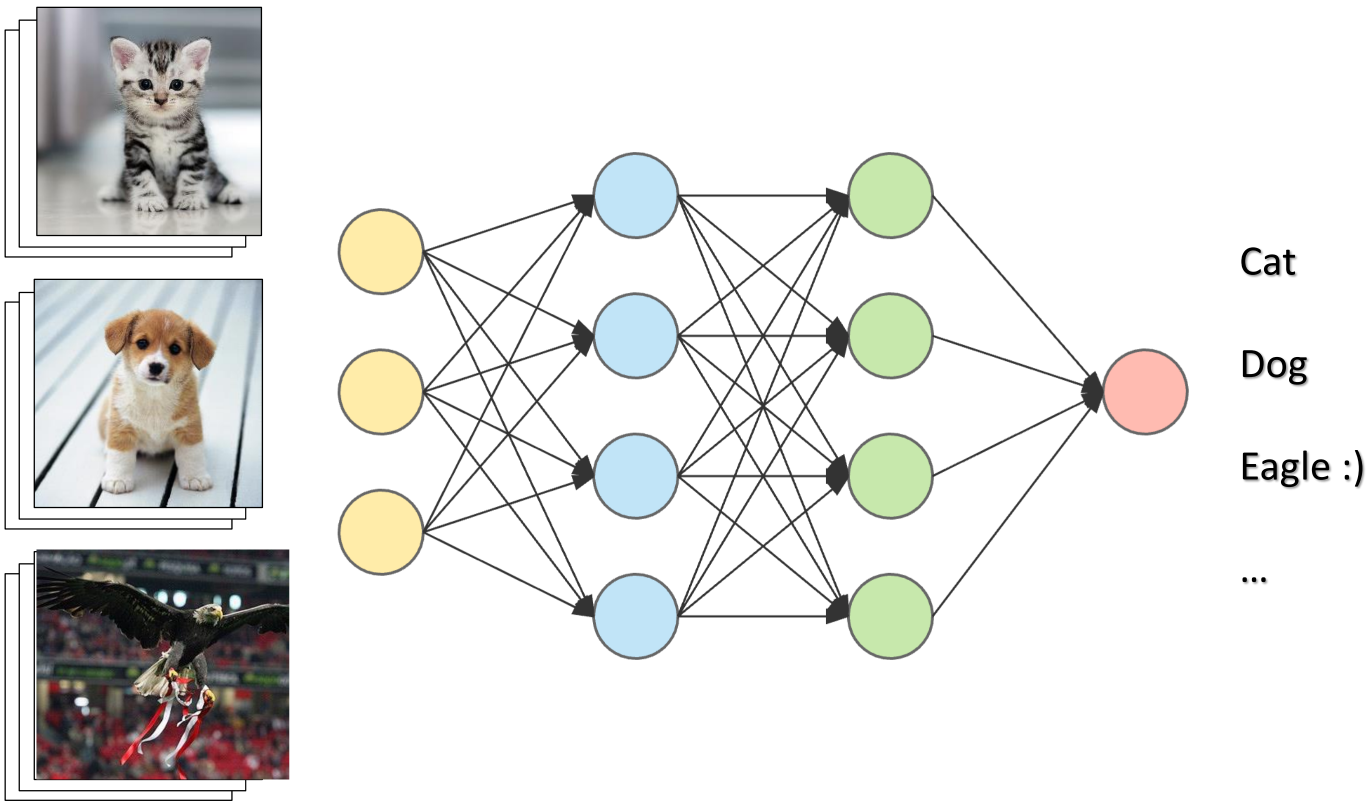




Recurrent Architecture

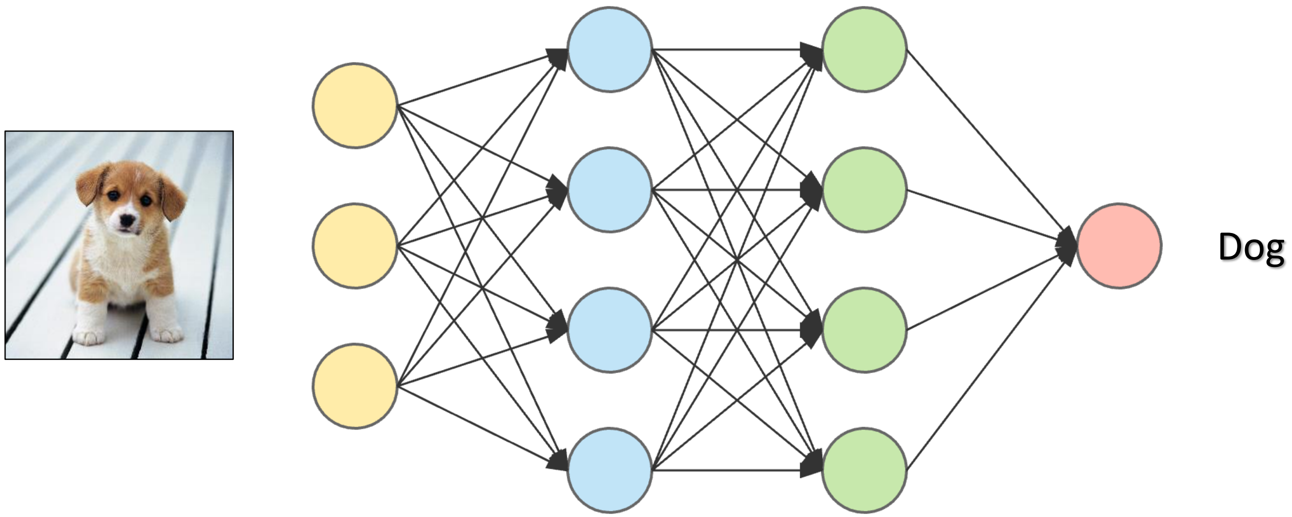
Artificial Neural Networks

MLPs



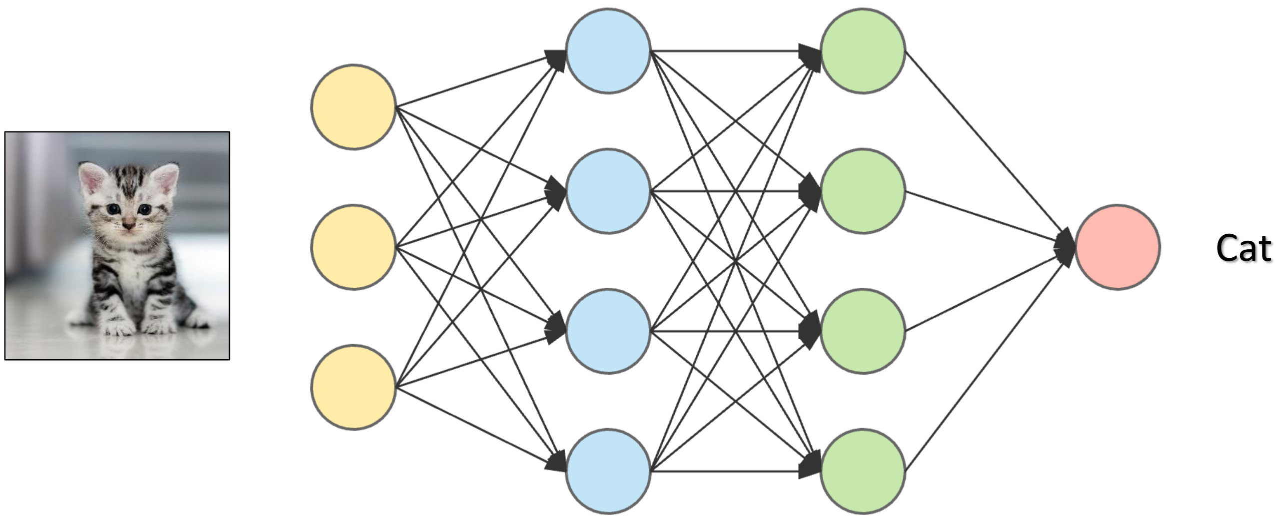
Artificial Neural Networks

MLPs



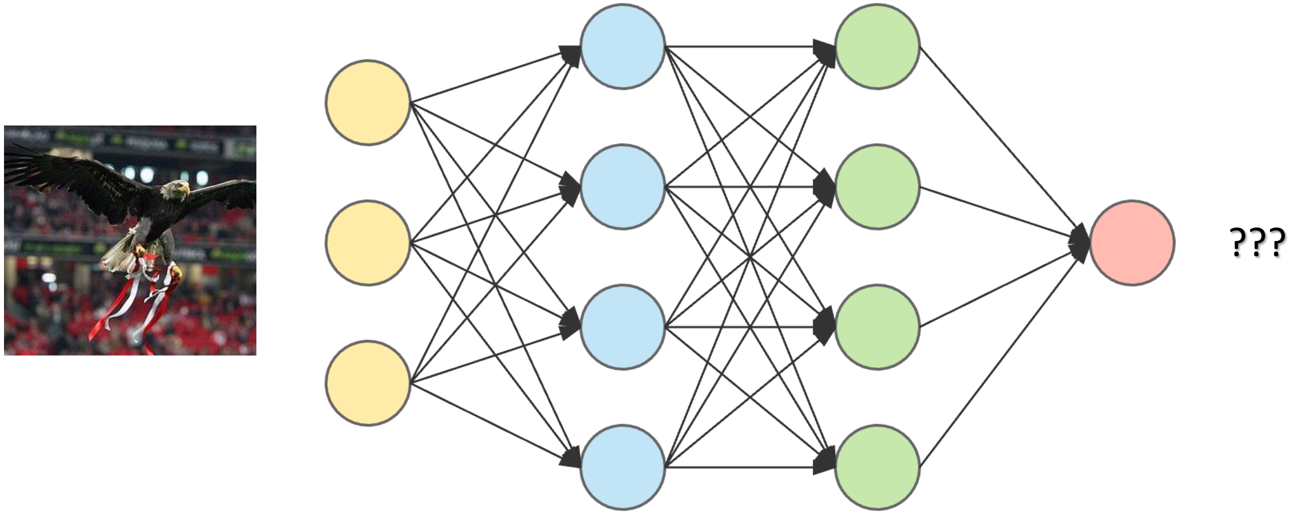
Artificial Neural Networks

MLPs



Artificial Neural Networks

MLPs



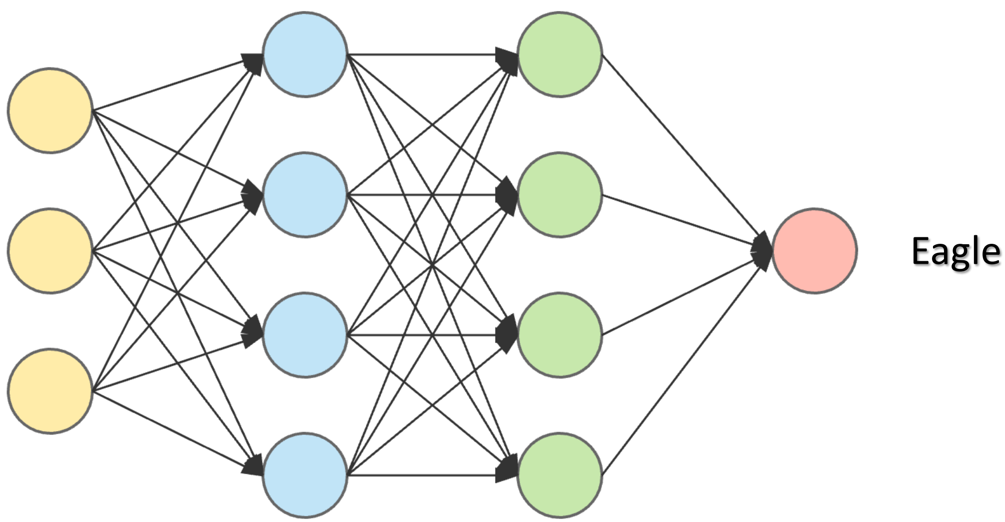
Artificial Neural Networks

MLPs

Does the answer to a problem depends on a previous value/input/tensor?

If it does, a simple MLP:

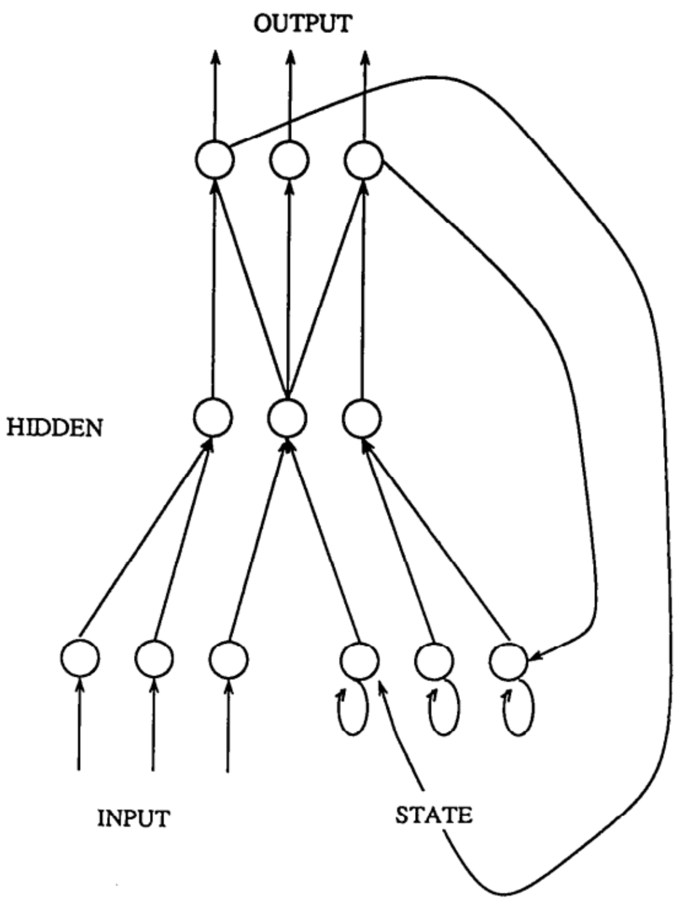
- Does not have any notion of order, sequence or time!
- Are "amnesic" when it comes to solve past problems
- Only remember what they learned during training



Recurrent Neural Networks

Recurrent Neural Networks (RNNs) where introduced in the 90's

- Simple Recurrent Network (SRN)
- In https://crl.ucsd.edu/~elman/Papers/fsit.pdf



In a RNN, the input is made of two components:

- The example in itself
- Previous perceptions

A previous perception will influence the decision of the current iteration!

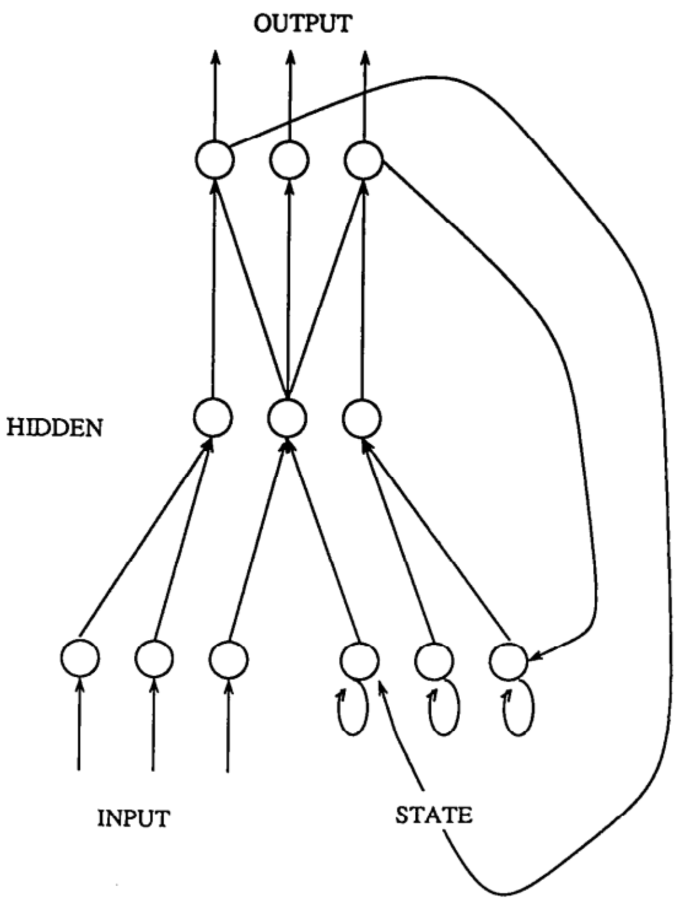
Aka, memory!

Recurrent Neural Networks

Applications

Problems dealing with sequential occurrences:

- Speech Recognition
- Stock market prediction
- Music generation
- Traffic flow forecasting
- …



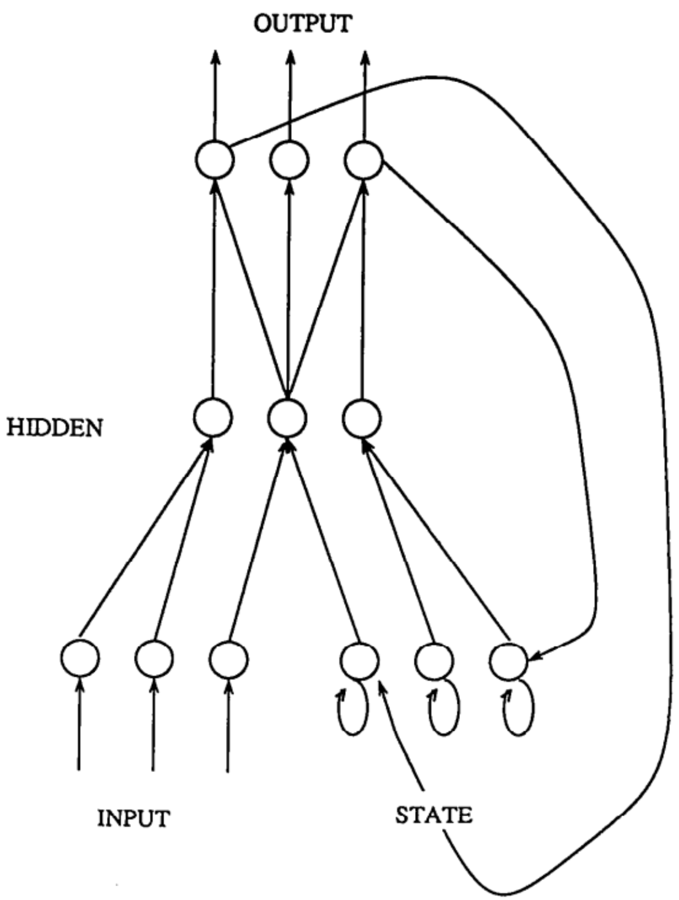
1. Multiplication table: how much is 7 x 8?

Recurrent Neural Networks
Applications

Problems dealing with sequential occurrences:

- Speech Recognition
- Stock market prediction
- Music generation
- Traffic flow forecasting
- …

1. Multiplication table: how much is 7 x 8?
2. Alphabet: name the 5 letters that appear after Q?

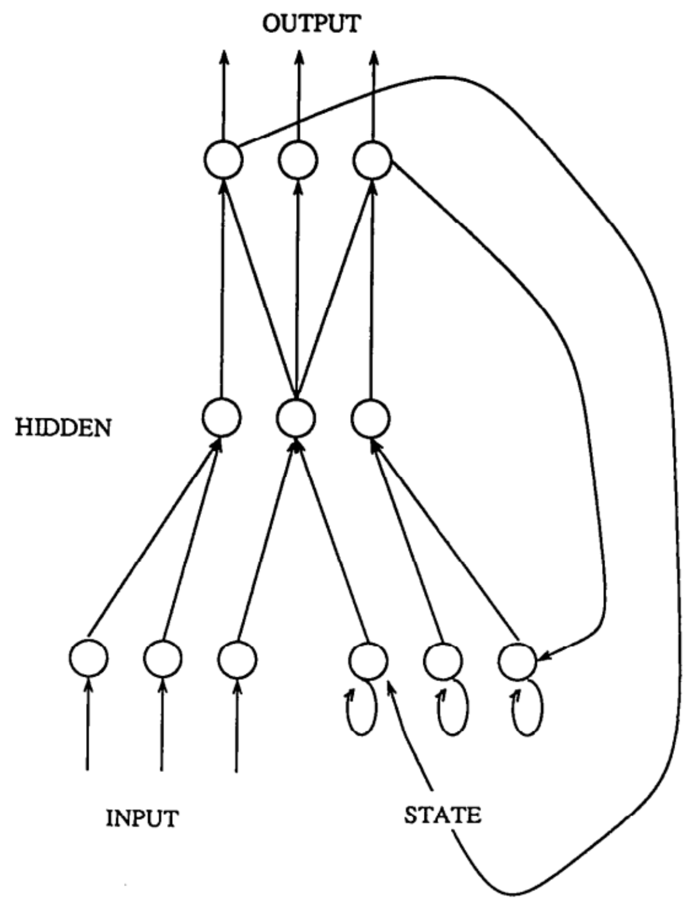


Recurrent Neural Networks

Applications

Problems dealing with sequential occurrences:

- Speech Recognition
- Stock market prediction
- Music generation
- Traffic flow forecasting
- …

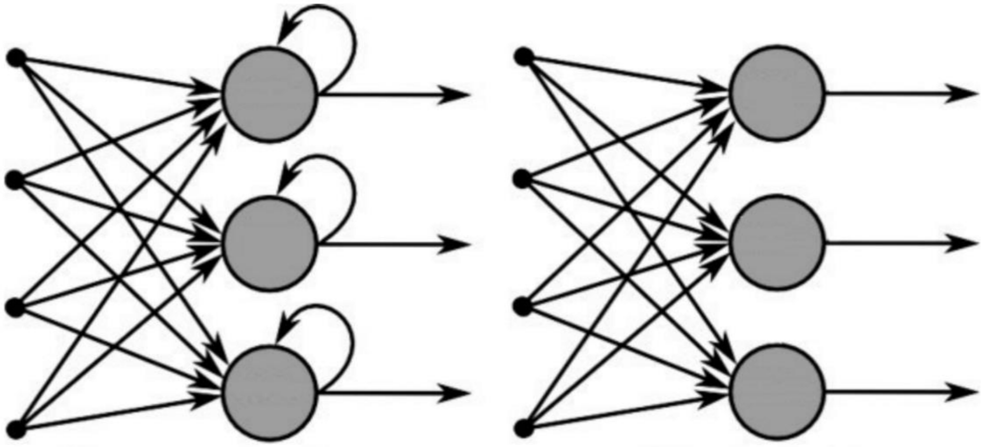


1. Multiplication table: how much is 7 x 8?
2. Alphabet: name the 5 letters that appear after Q?
3. Spell the word "*ALFABETICAMENTE*" from the last letter to the first

Recurrent Neural Networks

Recurrent vs Feed-Forward

- With RNNs, previous perceptions affect the current computation!
- This is what gives the network its memory and the notion of order/time!
- It is this notion of order/time that enables RNNs to solve problems with occurrence characteristics!

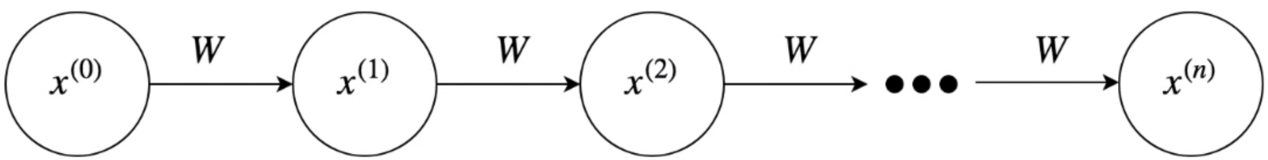


Recurrent Neural Network Feed-Forward Neural Network

Recurrent Neural Networks

Vanishing/Exploding Gradients

- The temporal dimension of RNNs is presented recursively throughout its execution!
- The error, at a given timestep t, depends on the error at the timestep t-1, and so on through all iterations:



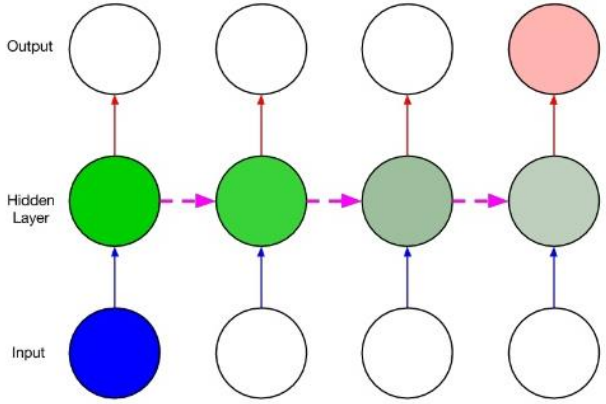
$$x^{(n)} = W^n x^{(0)} \qquad x^{(i)}, W \in \mathbb{R}, \quad i \in [0, n]$$

$$W^n x^{(0)} \rightarrow \begin{cases} \infty; & W > 1 \\ 0; & W < 1 \end{cases}$$

Recurrent Neural Networks

Vanishing/Exploding Gradients

- The temporal dimension of RNNs is presented recursively throughout its execution!
- The error, at a given timestep t, depends on the error at the timestep t-1, and so on through all iterations:
  - Exploding Gradients: when great importance is given to the weights, an “explosion” of their values can occur, which leads to high instability when learning!
  - Vanishing Gradients: When the gradient values are too small, their propagation tends to lose influence, which leads to a loss of learning capacity (learning freezes)!
- To address these issues, new RNNs proposals have emerged! We will consider two of those!



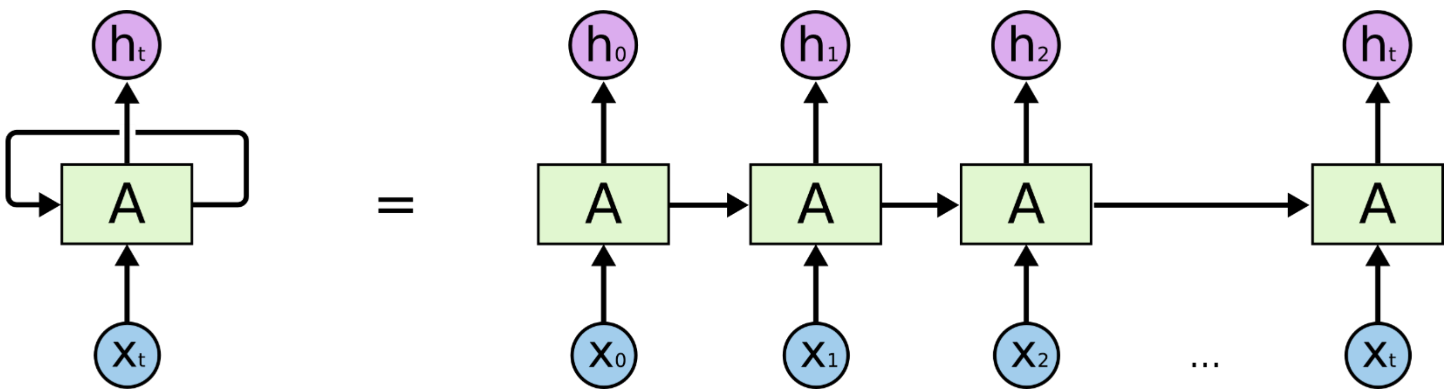
Long Short-Term Memory Networks (LSTMs)

Reading it slowly…

1. Long
2. Short-term
3. Memory units

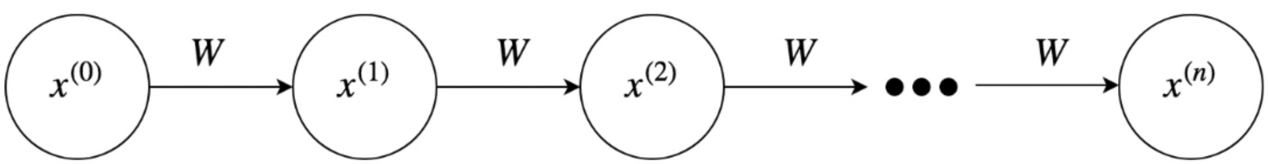
Ou:

1. *Muitas*
2. *Unidades de Mémoria*
3. *De curta-duração*



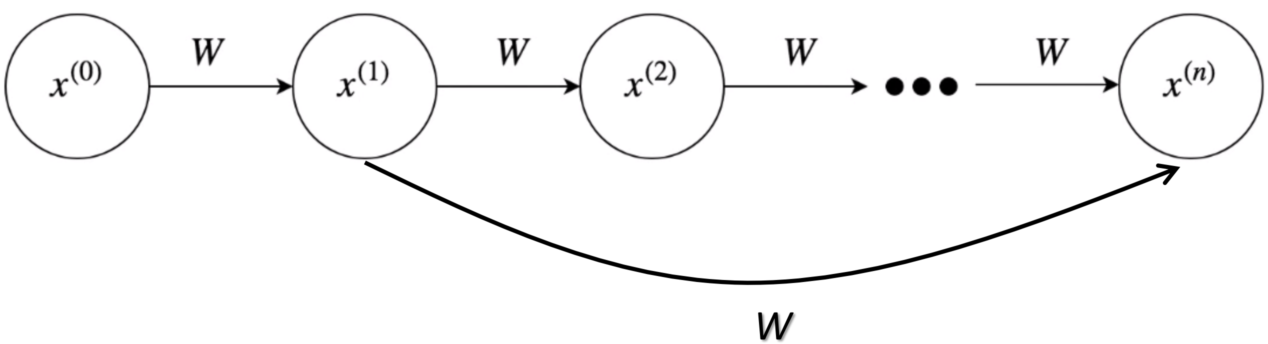
Long Short-Term Memory Networks (LSTMs)

- A special kind of RNN, capable of learning long-term dependencies
- Introduced by Hochreiter & Schmidhuber in 1997, but many have contributed to the current version of LSTMs
- The goal of LSTMs is to preserve the error estimation, which is backpropagated through time (through previous timesteps!)
- With a "constant" error value variation, LSTMs train deeper and during more iterations



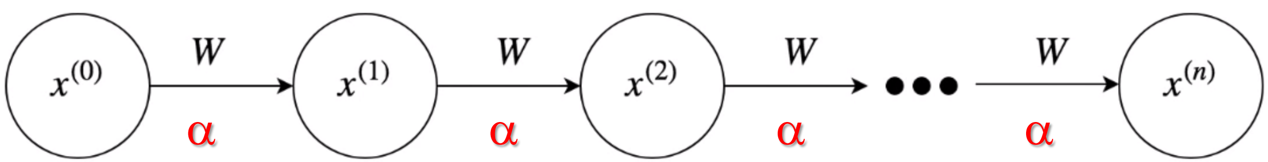
Long Short-Term Memory Networks (LSTMs)

- The goal of LSTMs is to preserve the error estimation, which is backpropagated through time (through previous timesteps!)
- With a “constant” error value variation, LSTMs train deeper and during more iterations
- It may skip connections!



Long Short-Term Memory Networks (LSTMs)

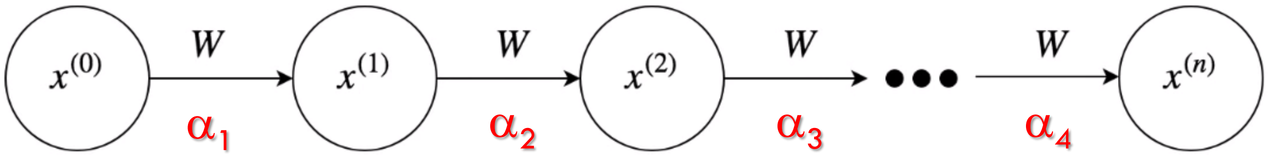
- The goal of LSTMs is to preserve the error estimation, which is backpropagated through time (through previous timesteps!)
- With a “constant” error value variation, LSTMs train deeper and during more iterations
- Control weights proportion (*Leaky Recurrent Units*)



with $0 < \alpha < 1$

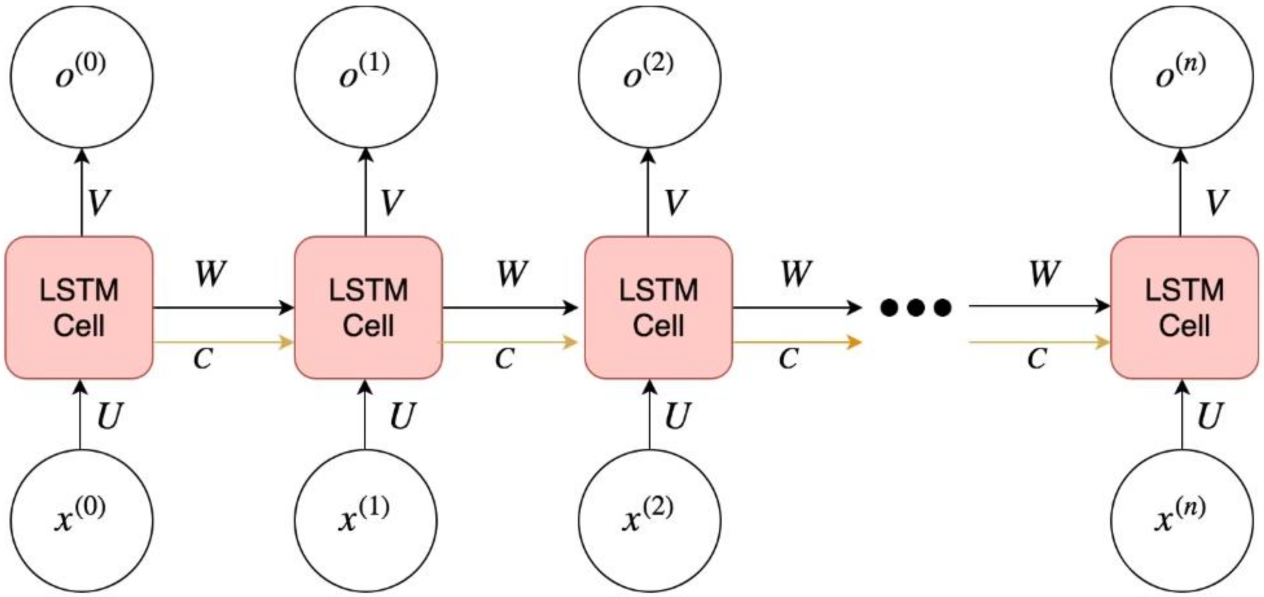
Long Short-Term Memory Networks (LSTMs)

- The goal of LSTMs is to preserve the error estimation, which is backpropagated through time (through previous timesteps!)
- With a “constant” error value variation, LSTMs train deeper and during more iterations
- Individually control weights proportion (*Gated Recurrent Networks*)



with $0 < \alpha_i < 1$

Long Short-Term Memory Networks (LSTMs)

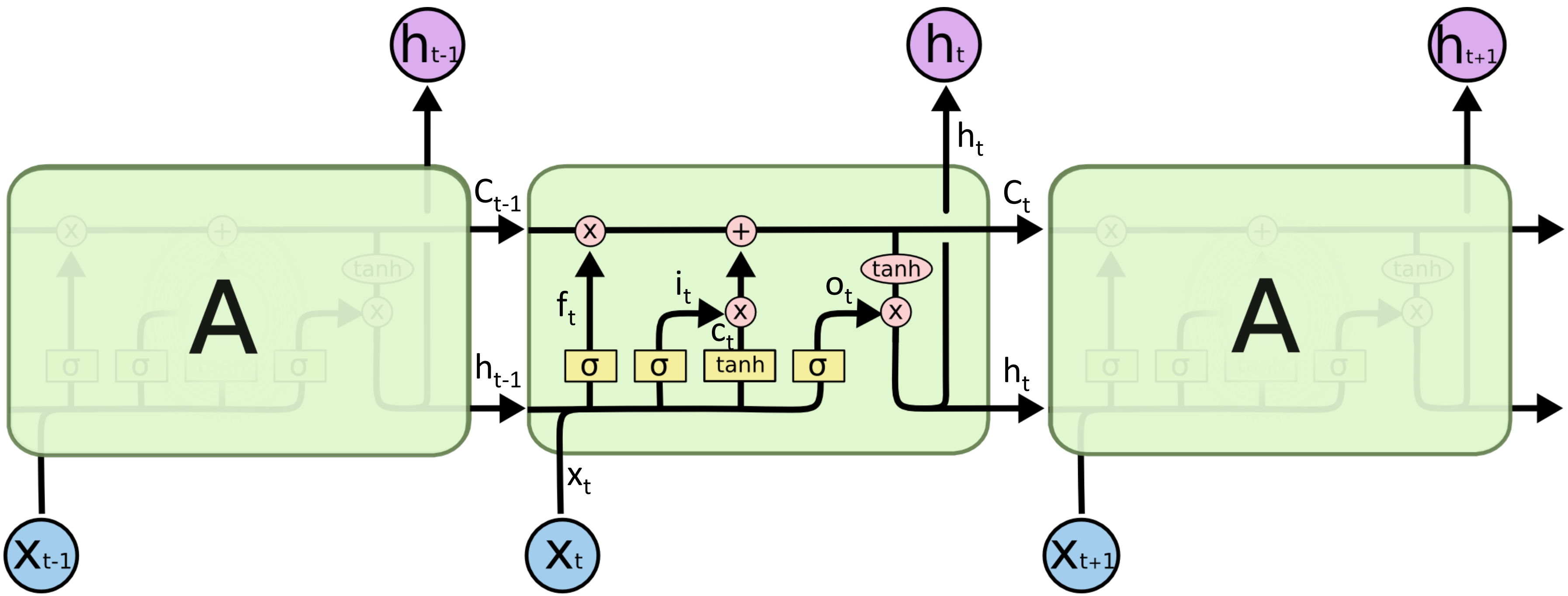


Long Short-Term Memory Networks (LSTMs)

- LSTMs are one of the most used type of networks today when handling sequences
- Each LSTM cell works as a “switch” the network must learn to enable/disable
- Each LSTM cell is characterized by its internal state (aka cell state) - C
- It uses three specific types of gates:
  - Forget Gate: controls what to forget from the cell’s internal state
  - Input Gate: controls what to add to the cell’s internal state
  - Output Gate: controls what to output from the cell’s internal state
- The “switches” are typically sigmoid functions (keeping the values between 0 and 1)
- The internal state (C) is updated based on the weights, the input and the previous states, going through a tanh function (keeping the values between -1 and 1)

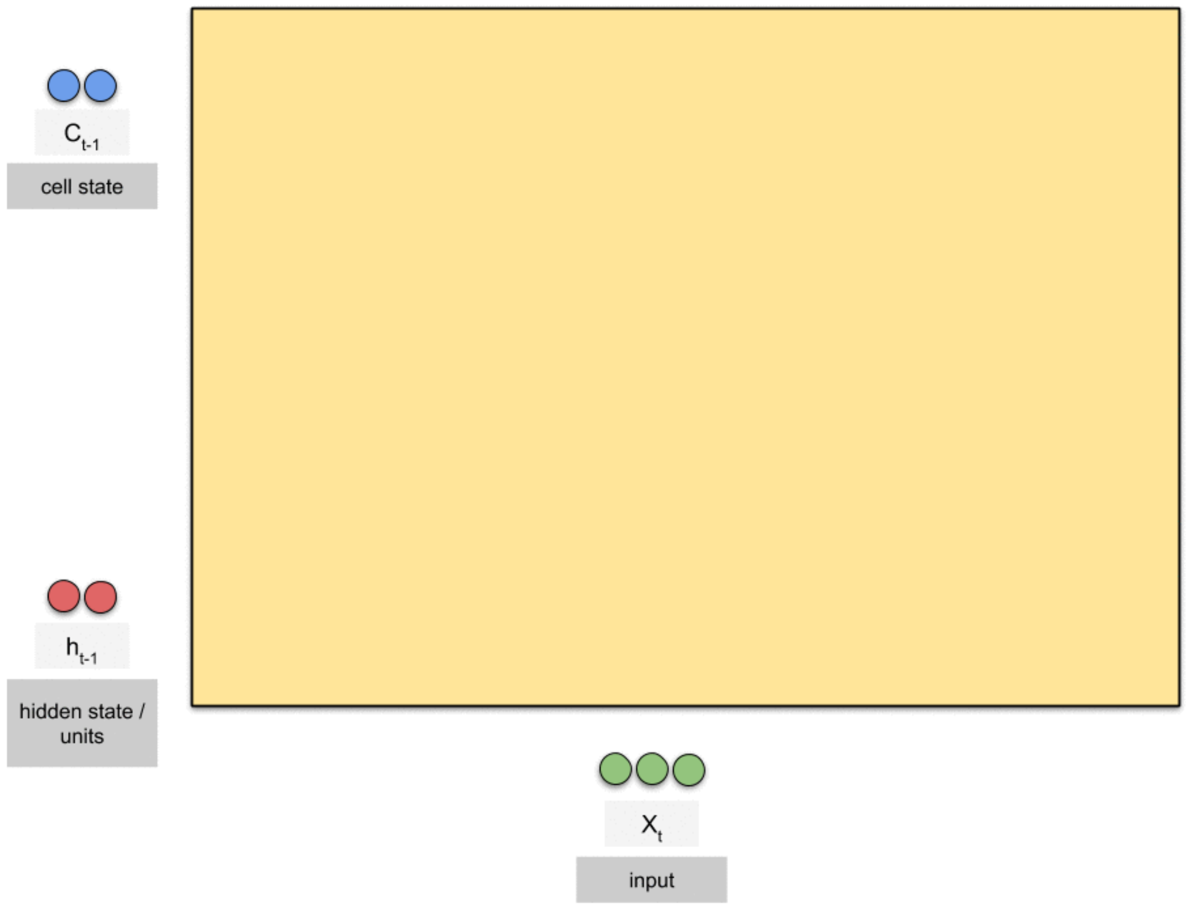
You may want to watch:
https://www.youtube.com/watch?v=8HyCNIVRbSU

Long Short-Term Memory Networks (LSTMs)



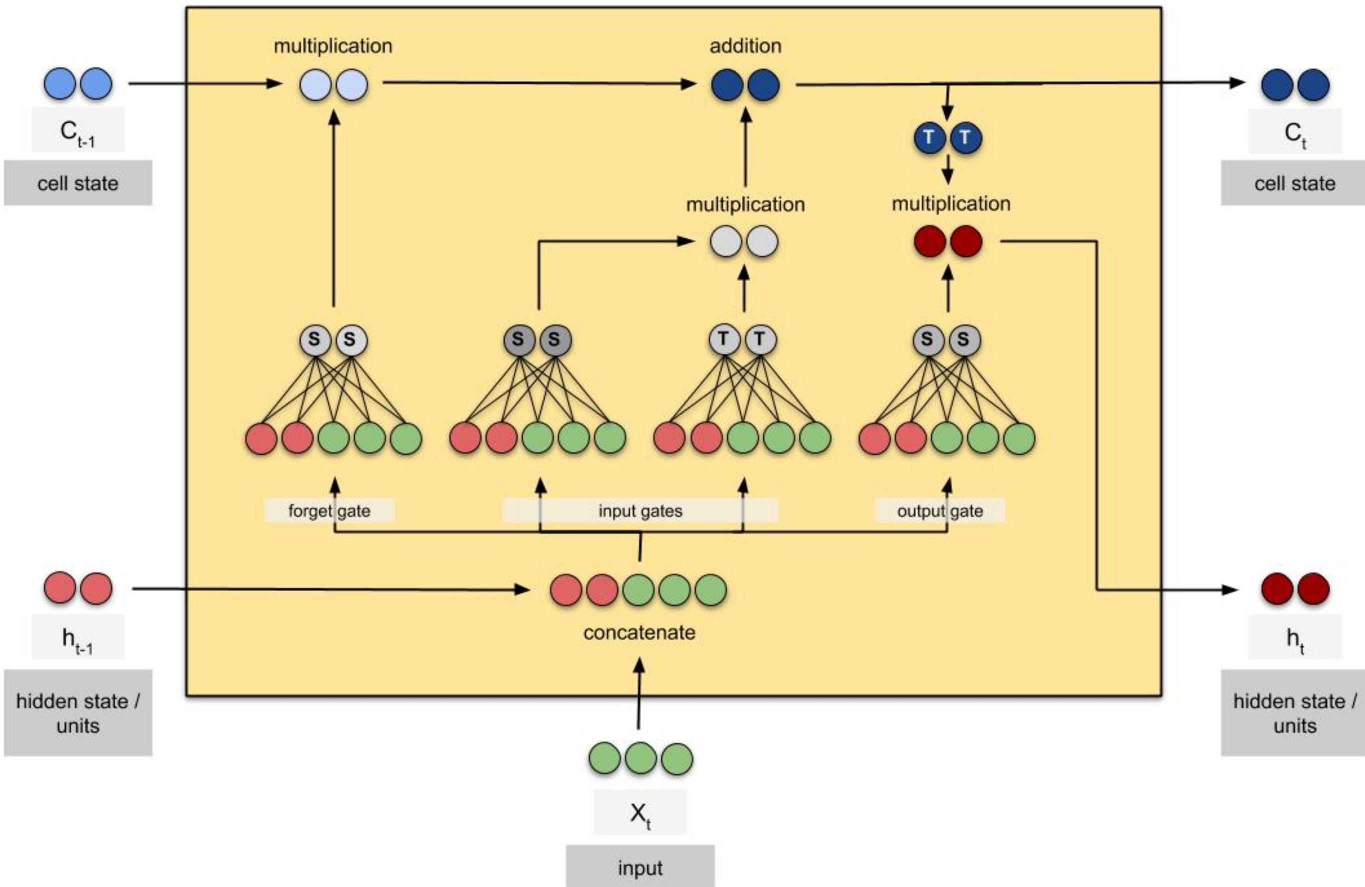
(https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Long Short-Term Memory Networks (LSTMs)



(https://towardsdatascience.com/animated-rnn-lstm-and-gru-ef124d06cf45)

Long Short-Term Memory Networks (LSTMs)

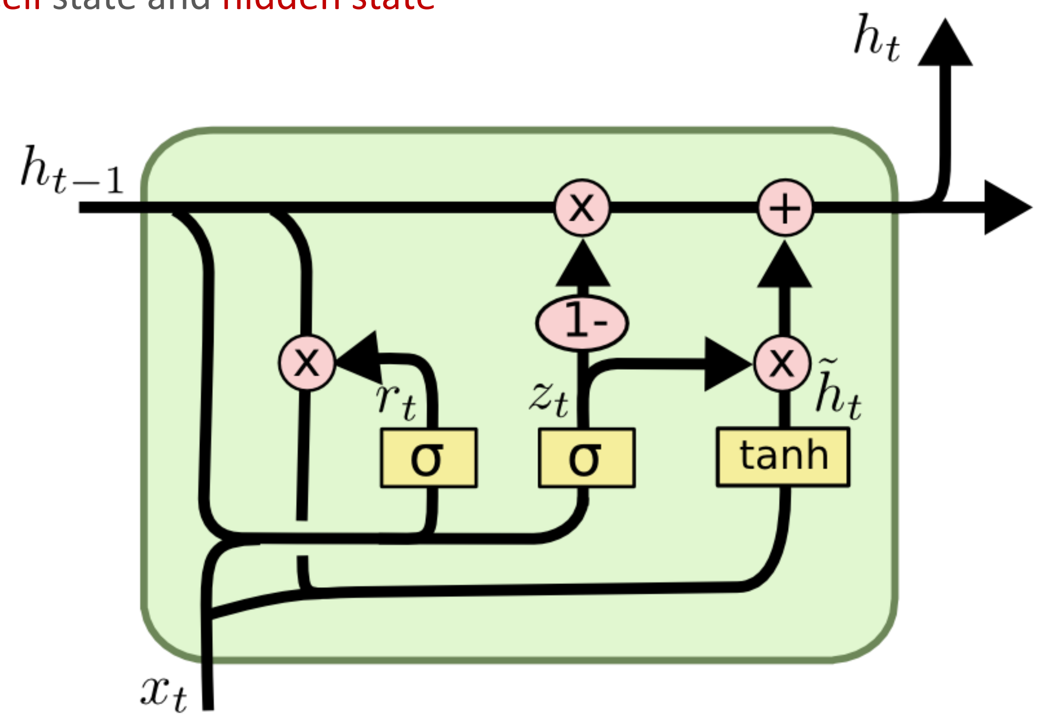


(https://towardsdatascience.com/animated-rnn-lstm-and-gru-ef124d06cf45)

Gated Recurrent Units (GRUs)

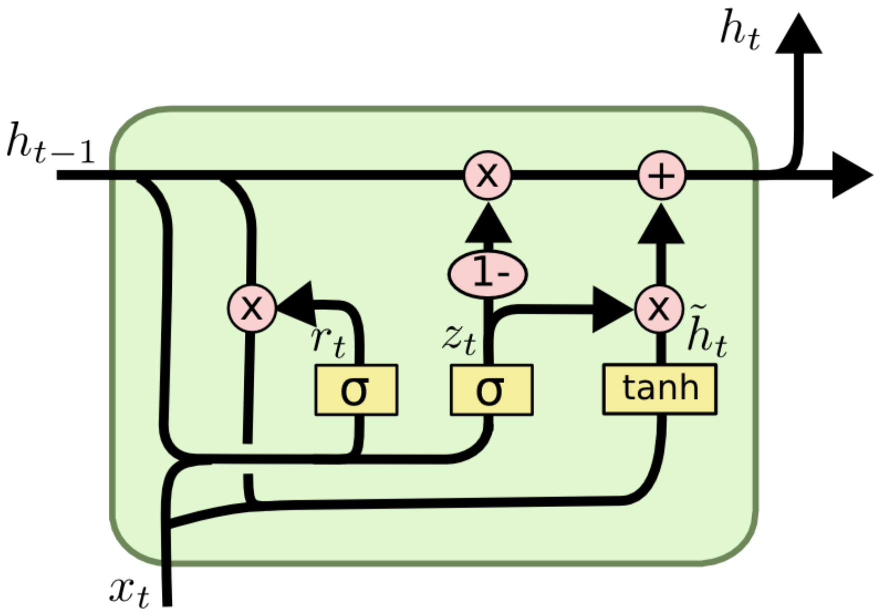
Introduced by Cho, et al. (2014). It essentially:

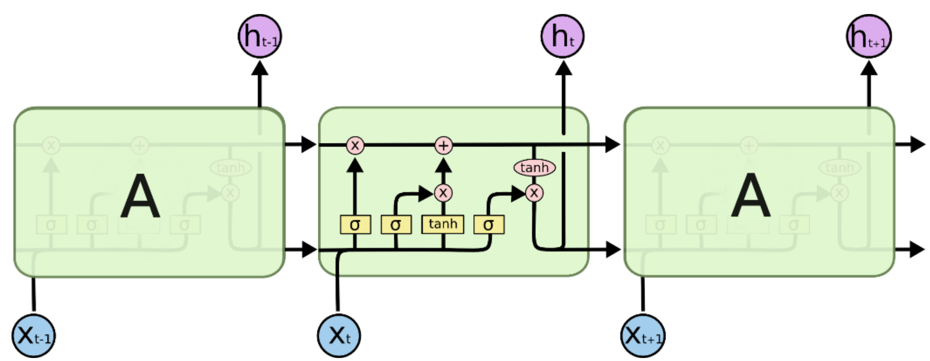
- Combines the forget and input gates into an update gate
- Merges the cell state and hidden state



(https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Gated Recurrent Units (GRUs)



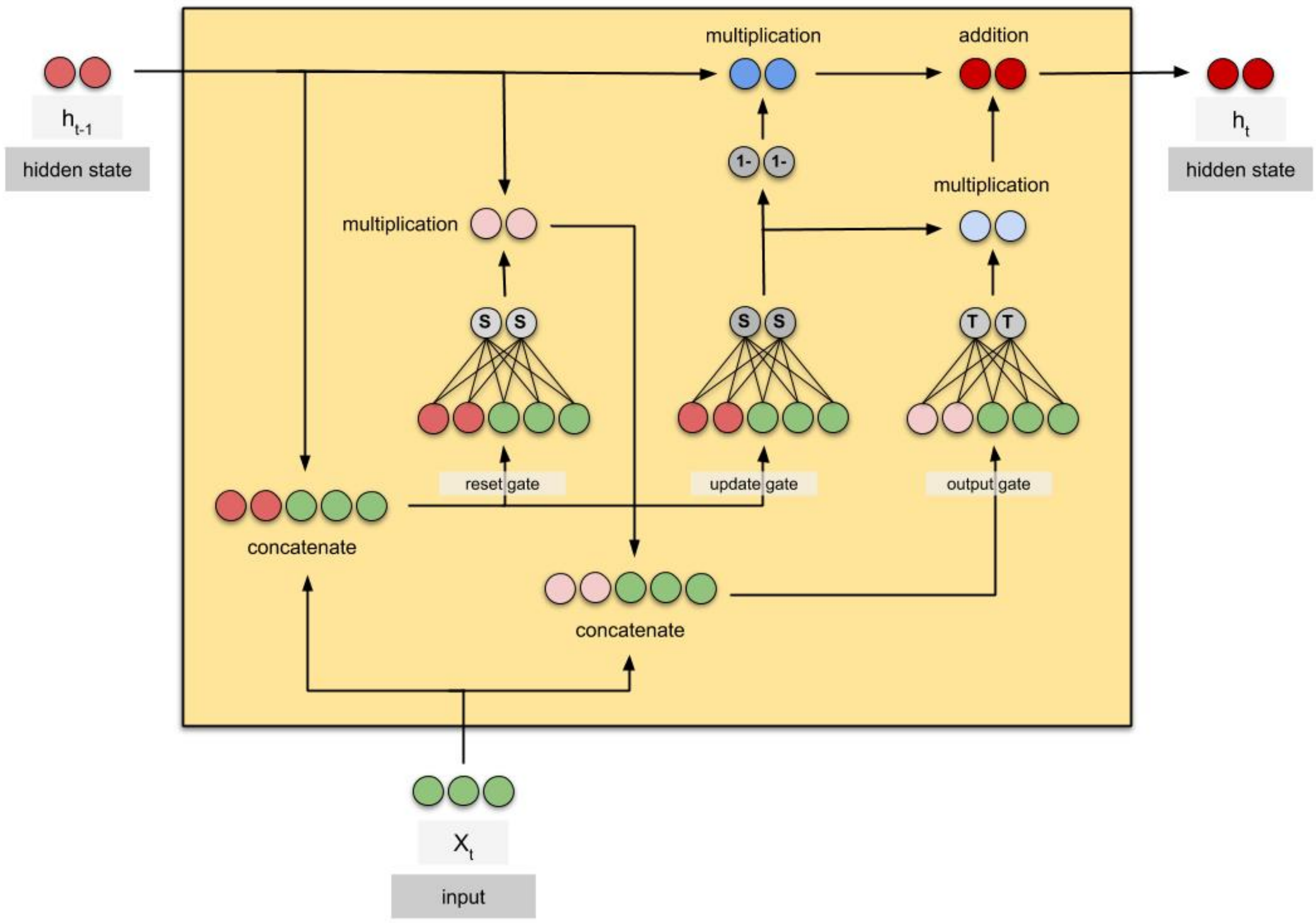


(https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Gated Recurrent Units (GRUs)

(https://towardsdatascience.com/animated-rnn-lstm-and-gru-ef124d06cf45)

Gated Recurrent Units (GRUs)



(https://towardsdatascience.com/animated-rnn-lstm-and-gru-ef124d06cf45)

Conclusions

- RNNs work as a chain of Feed-Forward networks
- They learn by backpropagating in depth and through time (BPTT)
- Have memory!
- Specially useful for problems involving sequences
- Base RNNs may suffer from the Vanishing/Exploding gradient problem
- GRUs and LSTMs, among others, appeared to minimize these problems!

Glossary

- Gated Recurrent Units, Recurrence, Long Short-Term Memory Networks, and so on…

Check the previous slides for the definition of each and every one of the terms we saw today.

Resources

- Papers, Books, online courses, tutorials…
  - Elman, J., “Finding Structure in Time”, Cognitive Science 14, pp. 179-211, 1990. https://crl.ucsd.edu/~elman/Papers/fsit.pdf
  - Haykin, S., “Neural Networks - A Comprehensive Foundation”, Prentice-Hall, New Jersey, 2nd Edition, 1999.
  - James McClelland (2015), “Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises”, “Chapter 7 – The Simple Recurrent Network: A Simple Model that Captures the Structure in Sequences” https://web.stanford.edu/group/pdplab/pdphandbook/handbook.pdf
  - Hochreiter, S. & Schmidhuber, J., “Long Short-Term Memory”, Neural Computation 9(8), pp. 1735-1780, 1997. http://www.bioinf.jku.at/publications/older/2604.pdf
  - Cho, K., et al., “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”. https://arxiv.org/abs/1406.1078
  - https://colah.github.io/posts/2015-08-Understanding-LSTMs
  - https://towardsdatascience.com/animated-rnn-lstm-and-gru-ef124d06cf45