Recurrent Neural Networks

Scratching the Surface

Roberto Souza
Assistant Professor
Electrical and Computer Engineering
Schulich School of Engineering

March 2021

UNIVERSITY OF CALGARY

Outline

- Learning Goals
- Motivation
- Recurrent Neural Networks (RNNs)
  - Traditional RNNs
  - Long short-term memory (LSTM)
- Summary

Additional Resources

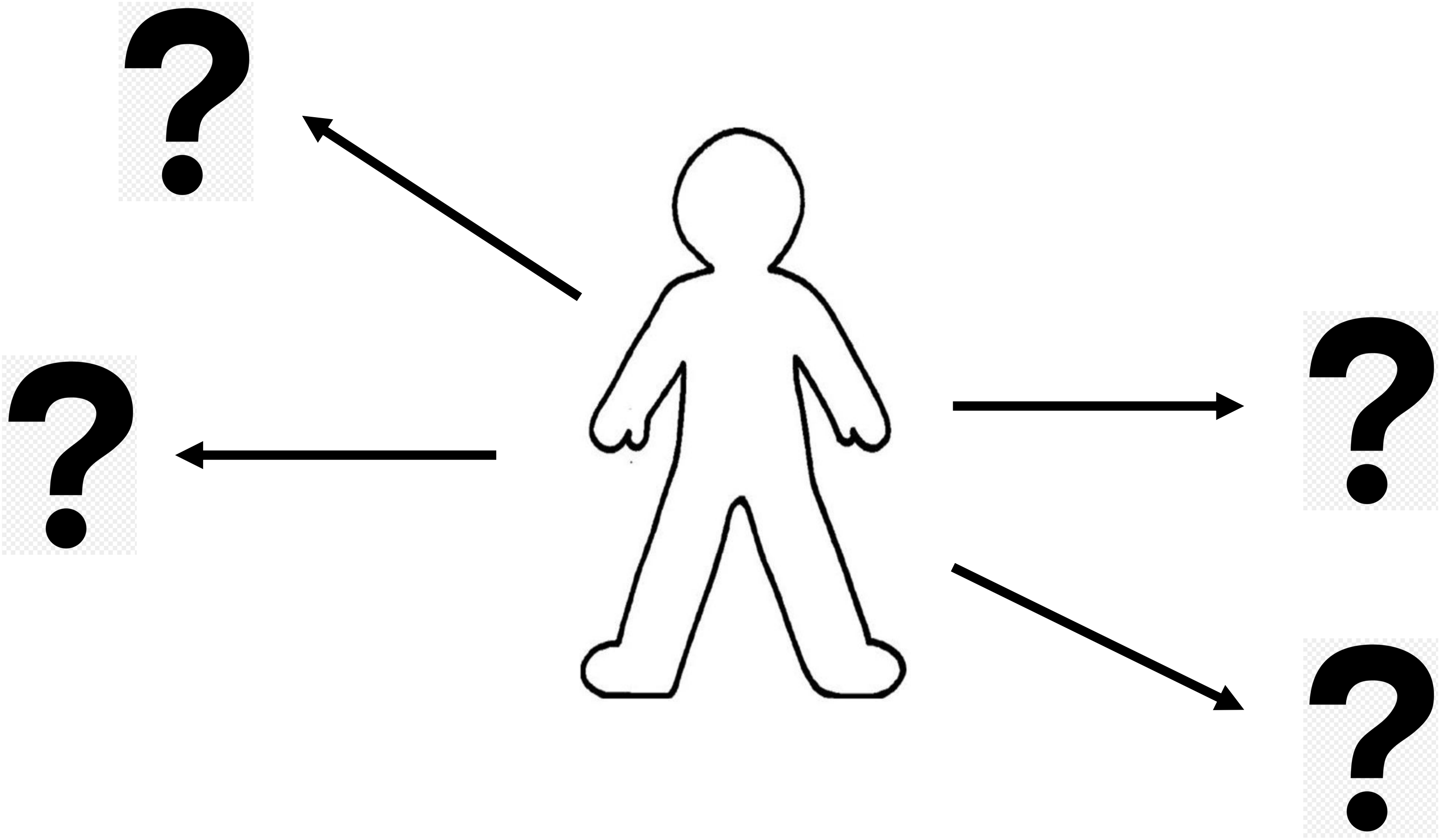
- Stanford cheatsheet:
  - https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks
- Colah’s blog:
  - https://colah.github.io/posts/2015-08-Understanding-LSTMs/
- Illustrated guide of LSTM:
  - https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Learning Goals

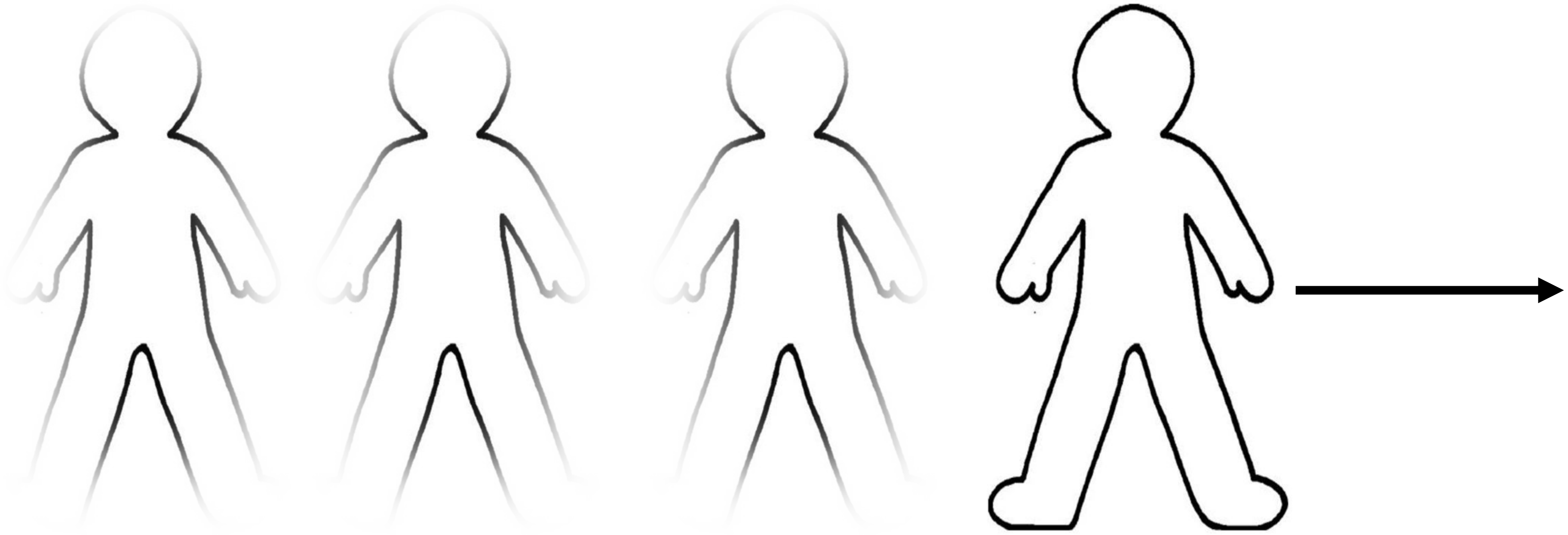
- Learn the intuition behind RNNs

- Get familiar with the most common types of RNNs (traditional and LSTM)

Where is the person going?



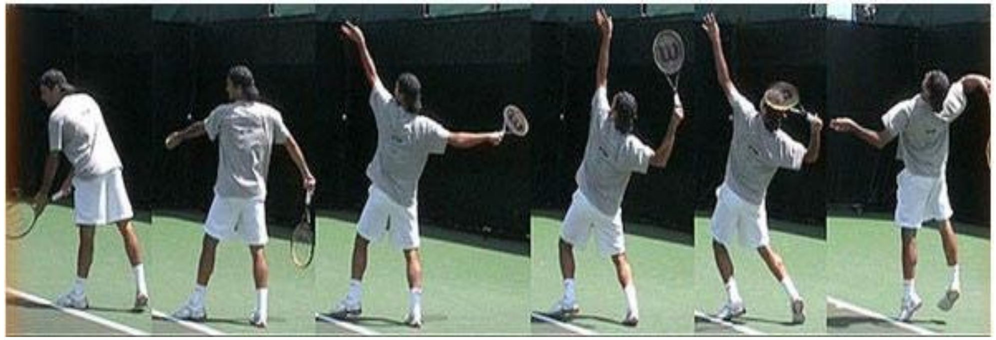
Where is the person going?

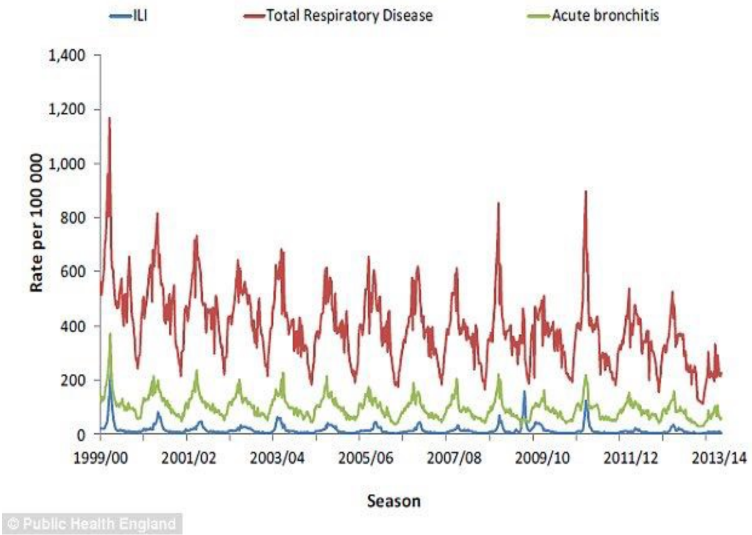


Motivation - Data is often sequential in nature

Steph Curry releases the ball and as it moves you know it is going to be 3 points to Golden State…

Motivation - Data is often sequential in nature





Introduction

- Building models of sequential data is important: automatic speech recognition, machine translation, natural language, …

- Recurrent neural networks (RNNs) are a simple and general framework for this type of tasks

Introduction

```
A B C A B C A B _
```

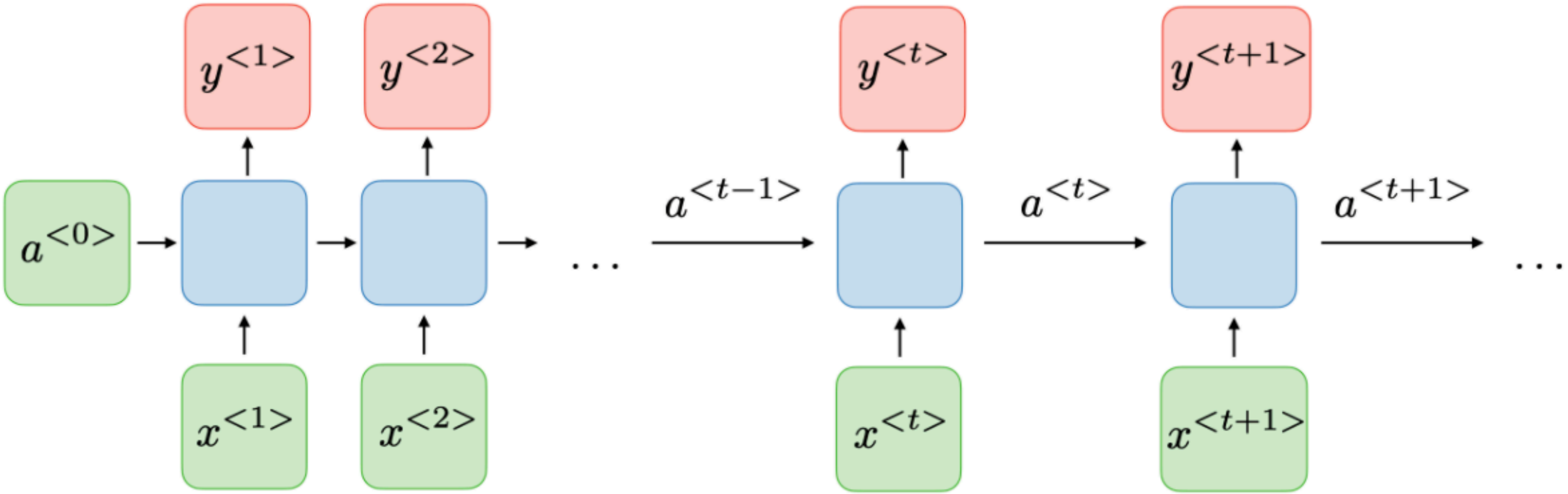
- What symbol comes next?

```
Yesterday it was Sunday, so today it must be _
```

- How to predict the next word?

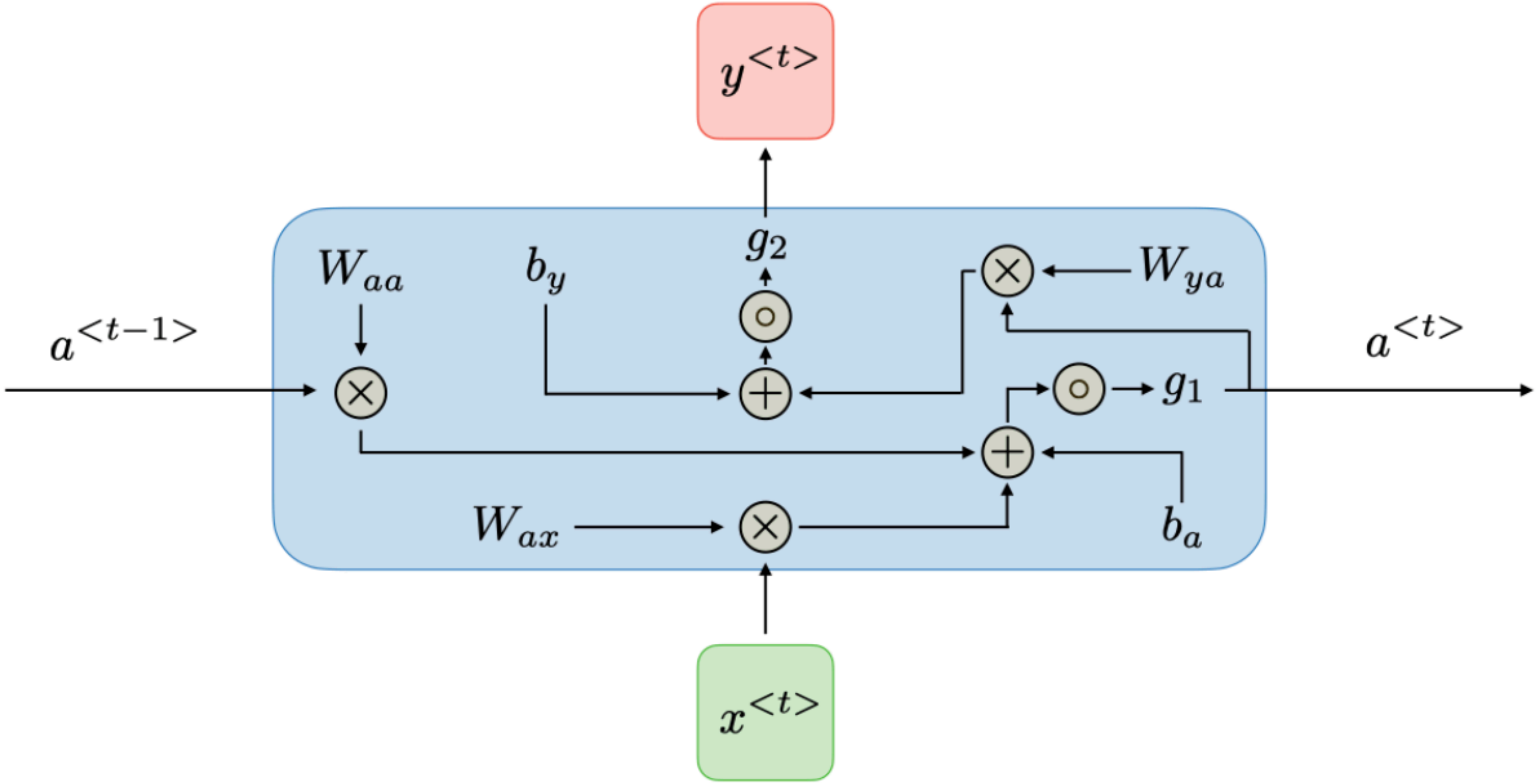
Traditional RNN

$$y^t = f(x^t, a^{t-1})$$



- Type equation here.RNNs can be seen as a (very deep) feedforward network with shared weights
- Model is trained using backpropagation through time

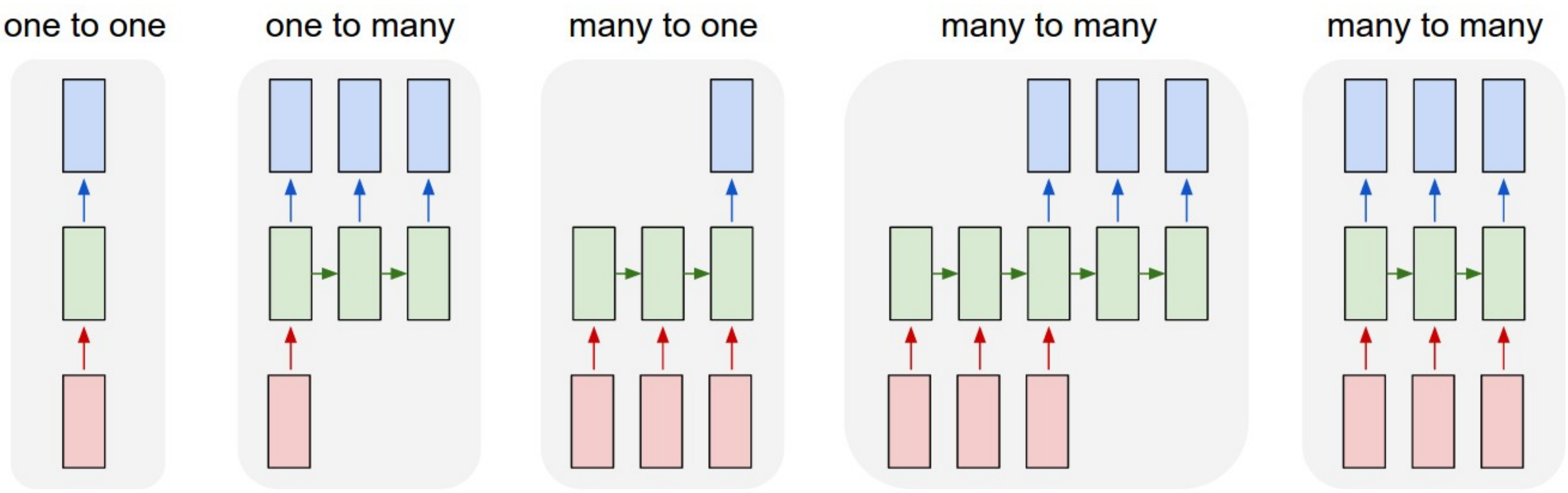
Traditional RNN



$$\boxed{a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)}$$

$$\boxed{y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)}$$

Types of RNNs



RNNs Advantages and Disadvantages

Advantages	Disadvantages
Possibility of processing input of any length	Computation being slow
Model size not increasing with size of input	Difficulty of accessing information from a long time ago
Computations take into account historical information	Cannot consider any future input for the current state
Weights are shared across time	

Major shortcomings

- Handling of complex non-linear interactions
- Difficulties using BPTT to capture long-term dependencies exploding gradients
- Vanishing gradients

Vanishing gradients

- As we propagate the gradients back in time, usually their magnitude quickly decreases: this is called “vanishing gradient problem”

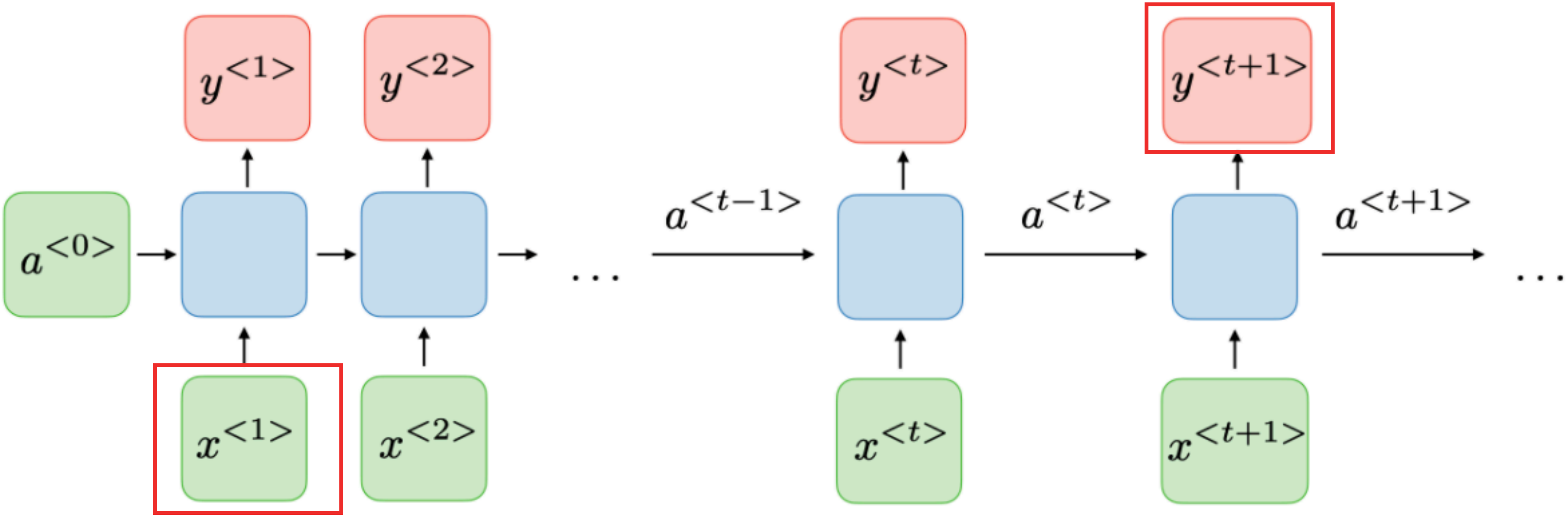
- In practice this means that learning long term dependencies in data is difficult for simple RNN architecture

- Special RNN architectures address this problem:
  - *Exponential trace memory* (Jordan 1987, Mozer 1989)
  - *Long Short-term Memory* (Hochreiter & Schmidhuber, 1997))

Exploding gradients

- Sometimes, the gradients start to increase exponentially during backpropagation through the recurrent weights
- Happens rarely, but the effect can be catastrophic: huge gradients will lead to big change of weights, and thus destroy what has been learned so far
- One of the main reasons why RNNs were supposed to be unstable
- Simple solution: clip or normalize values of the gradients to avoid huge changes of weights

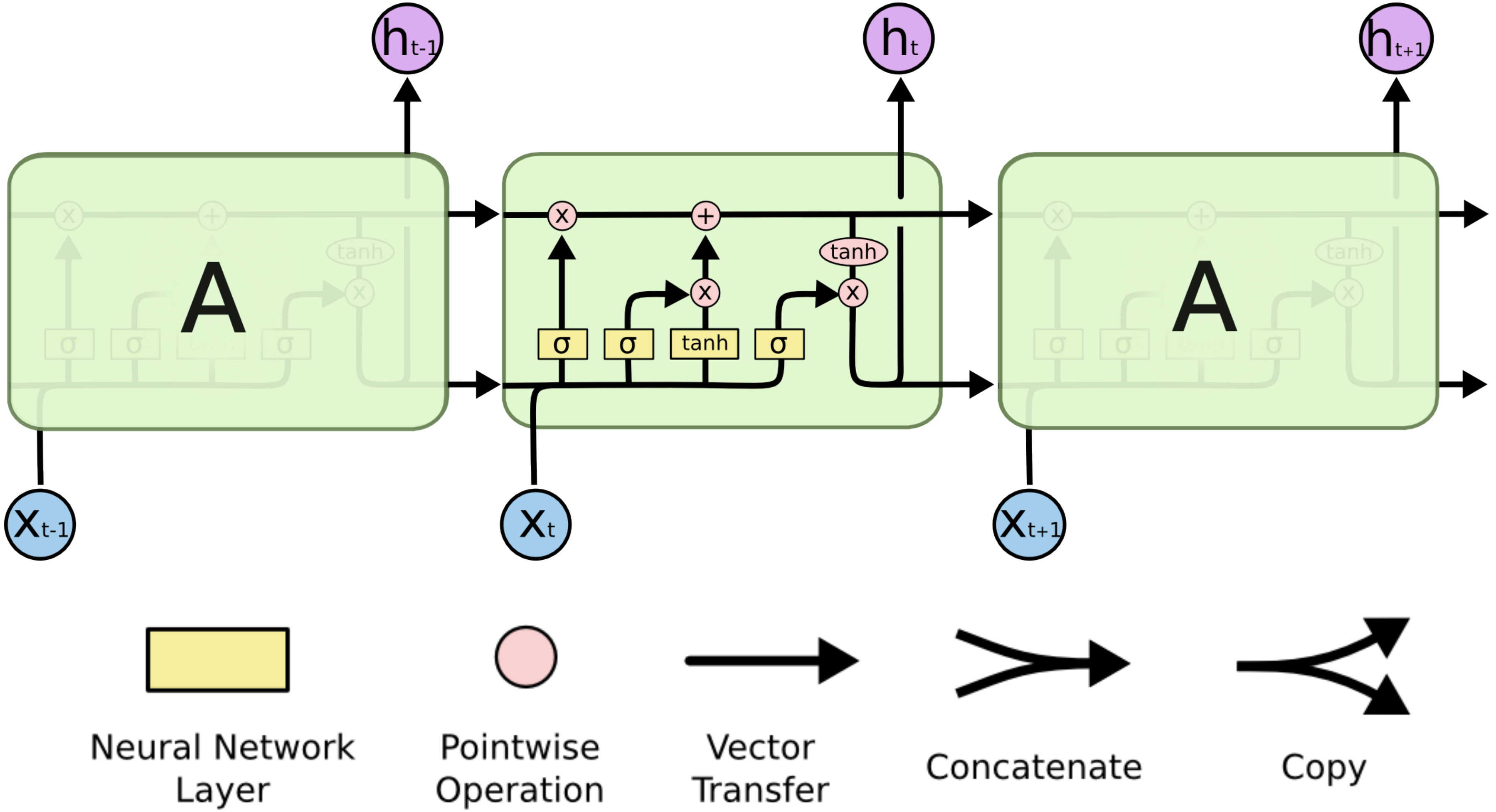
The Problem of Long-Term Dependencies



"In theory, RNNs are absolutely capable of handling such "long-term dependencies." A human could carefully pick parameters for them to solve toy problems of this form. Sadly, in practice, RNNs don't seem to be able to learn them."

Long Short-Term Memory (LSTM)

- LSTM is a type of RNN capable of learning long-term dependencies



UNIVERSITY OF CALGARY

Summary

- RNNs are capable of handling sequences of arbitrary lengths
- Traditional RNNs are not capable in practice to model long-term dependencies in data
- The LSTM model allows you to model these long-term dependencies
- More details in the tutorials…

Thank you!