Deep Learning Basics

Recurrent Neural Networks

Francis Steen and Xinyu You
Red Hen Lab
August 2019

A Question

You should take your ___?___.

What is the most likely word in the blank?

A Question

It's raining outside. You should take your __?__.

What is the most likely word in the blank?

A Question

It's raining outside. You should take your umbrella.

How?

A Question

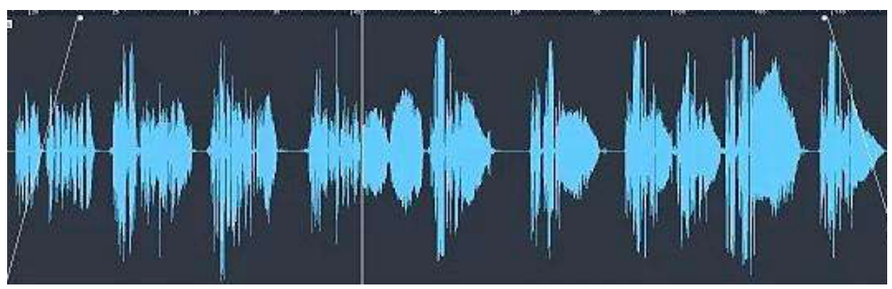
It's raining outside. You should take your umbrella.

information

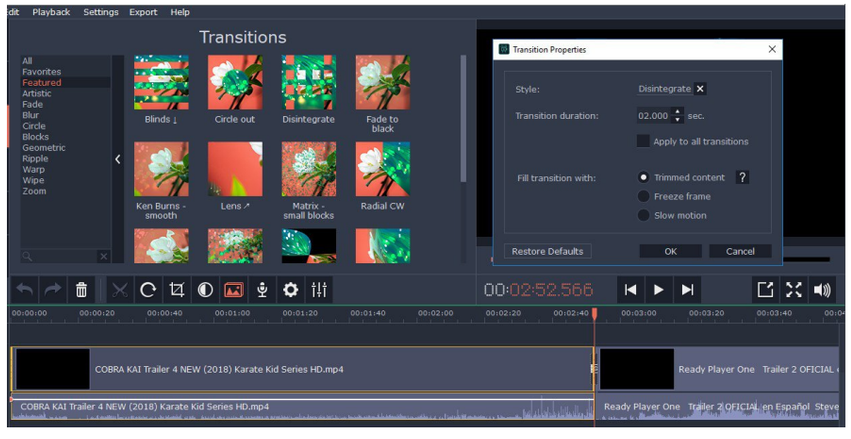
We need information from the distant past
to accurately predict the correct word.

Sequence Data

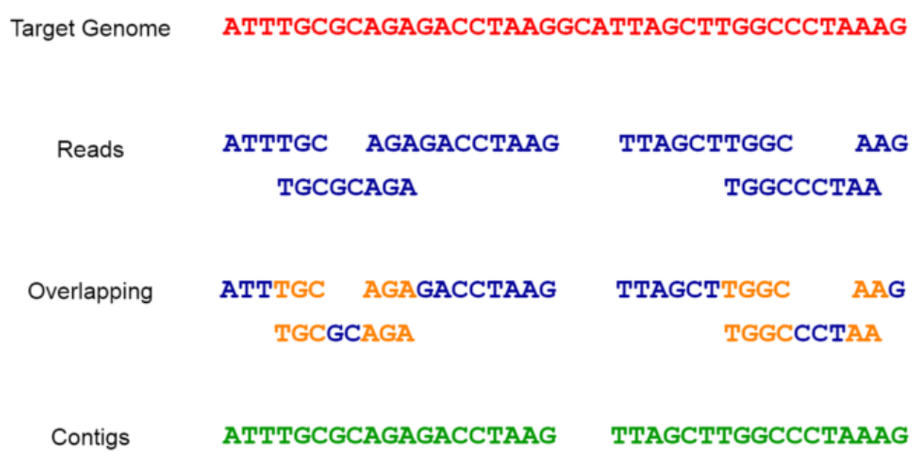
- Audio



- Text
- Video



- DAN Sequence

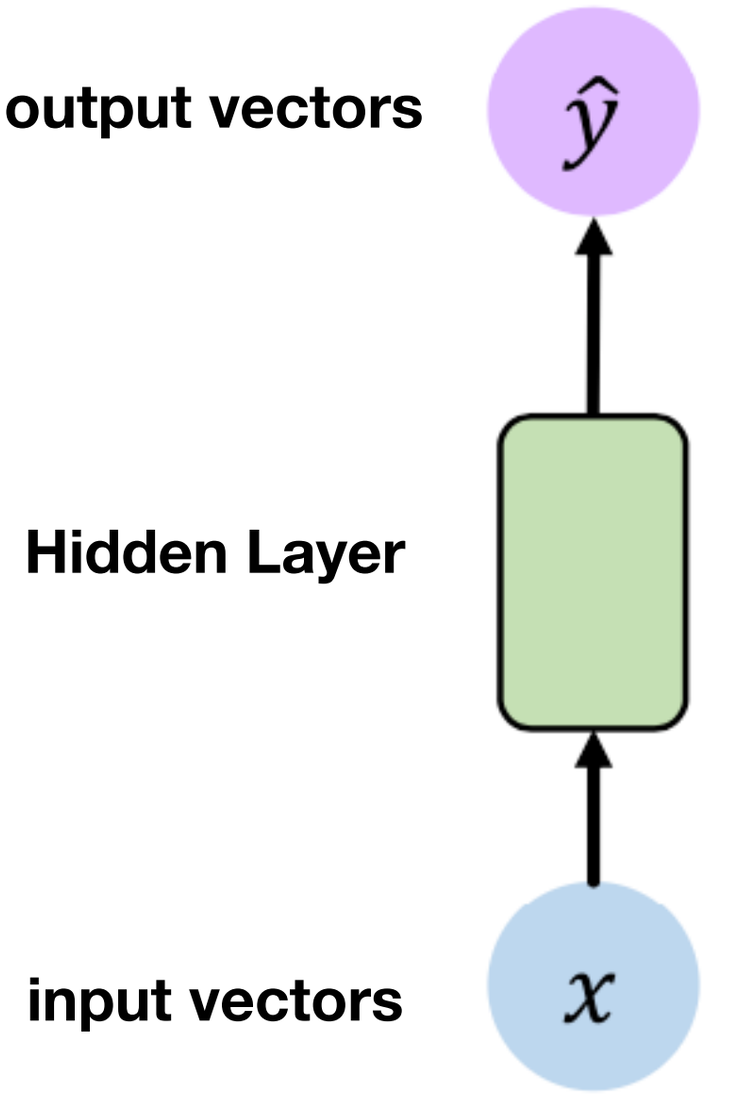


Sequence modeling

To model sequences, we need to:

1. Handle variable-length sequences
2. Track long-term dependencies
3. Maintain information about order
4. Share parameters across the sequence

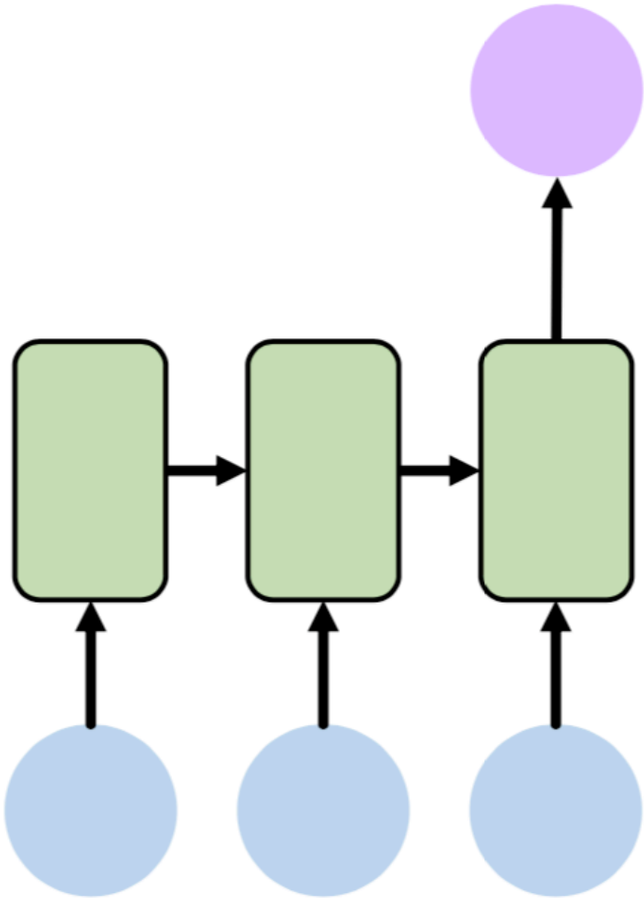
Deep forward neural networks



One to One

- x is the input vectors which in are in blue.
- y is the output vectors which are in purple.
- And green vectors hold the hidden layer.
- Assuming that each input is **independent**, that is, the output of each network depends only on the current input.

Recurrent neural networks



Many to One

Many to One

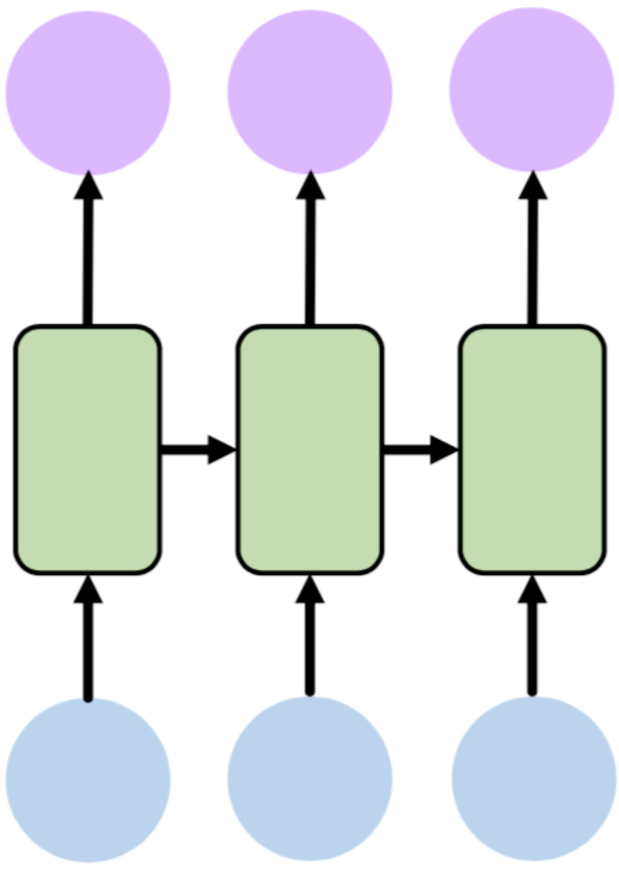
For example:

Input a sequence of words and predict the sentiment associated with it.

Input: the summer holiday begins, I am so happy.

Output: Happy.

Recurrent neural networks



Many to Many

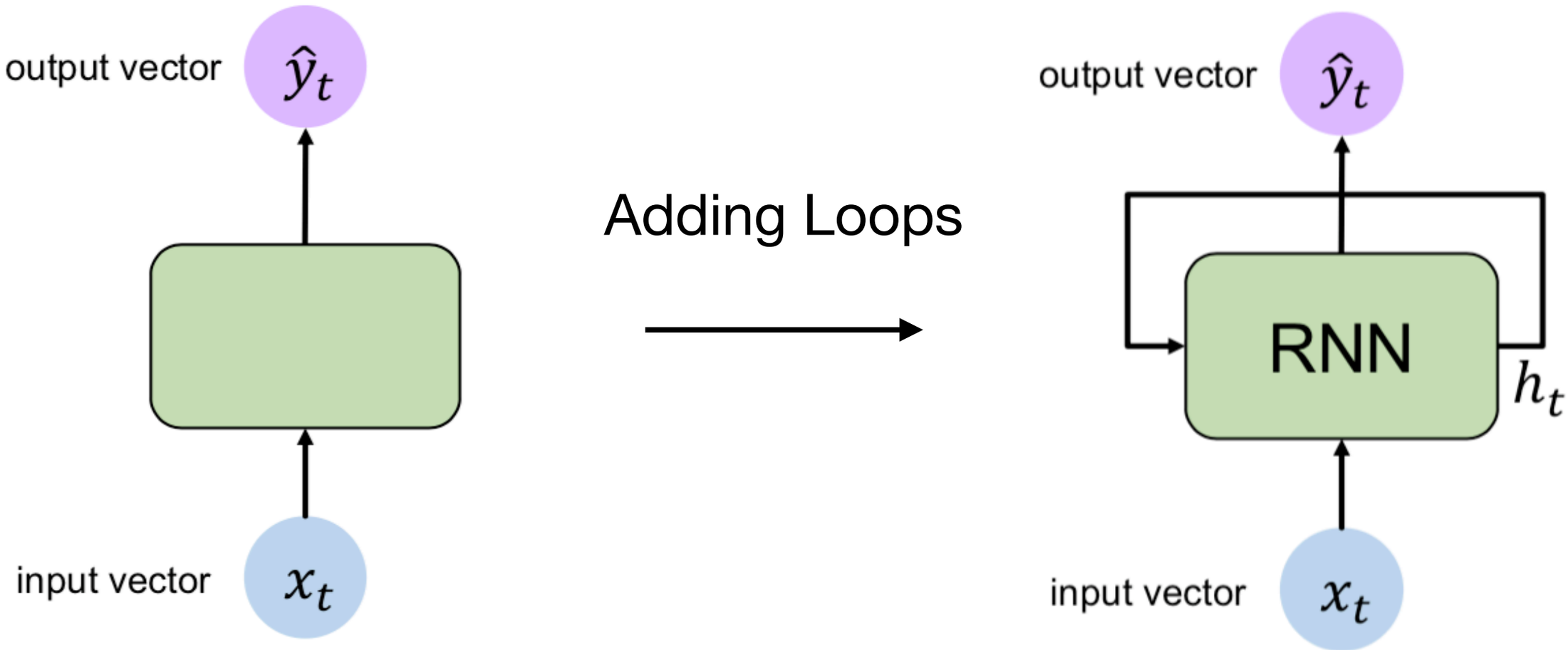
Many to May

For example: Q and A system.

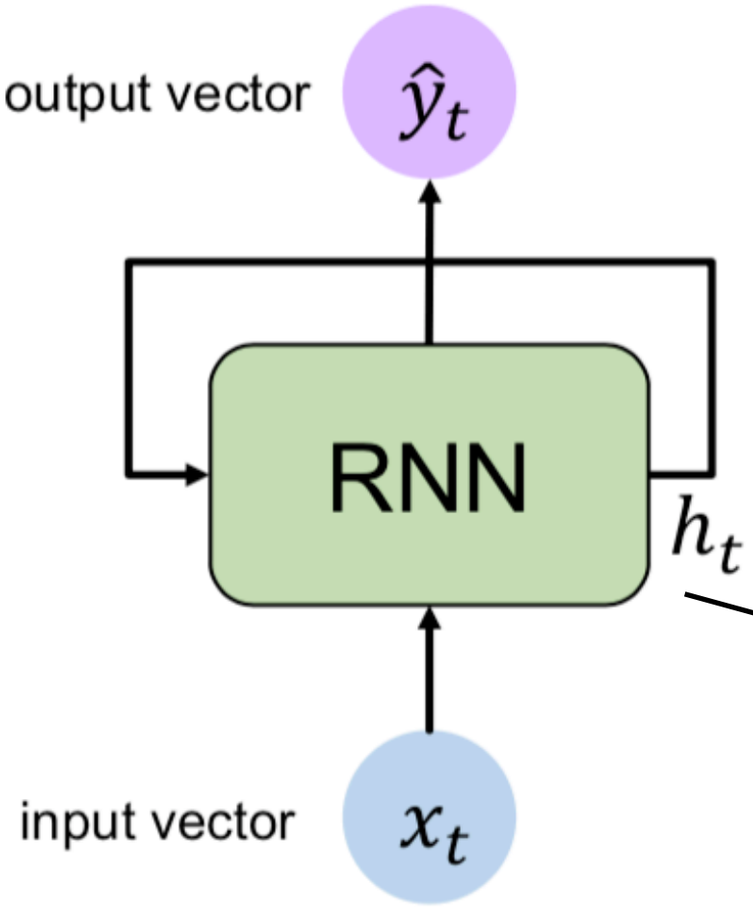
Input: which city is the capital of China?

Output: the capital of China is Beijing.

Recurrent neural networks



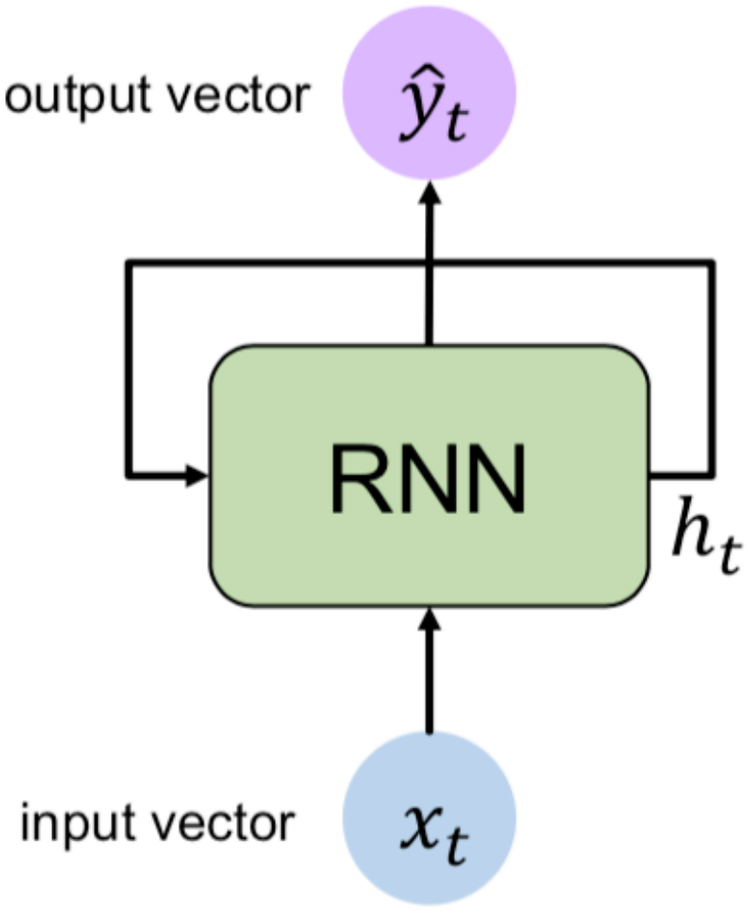
Recurrent neural networks



$$h_t = f_w(h_{t-1}, x_t)$$

- Recurrent cell
- h_t : hidden state at time t step

Recurrent neural networks

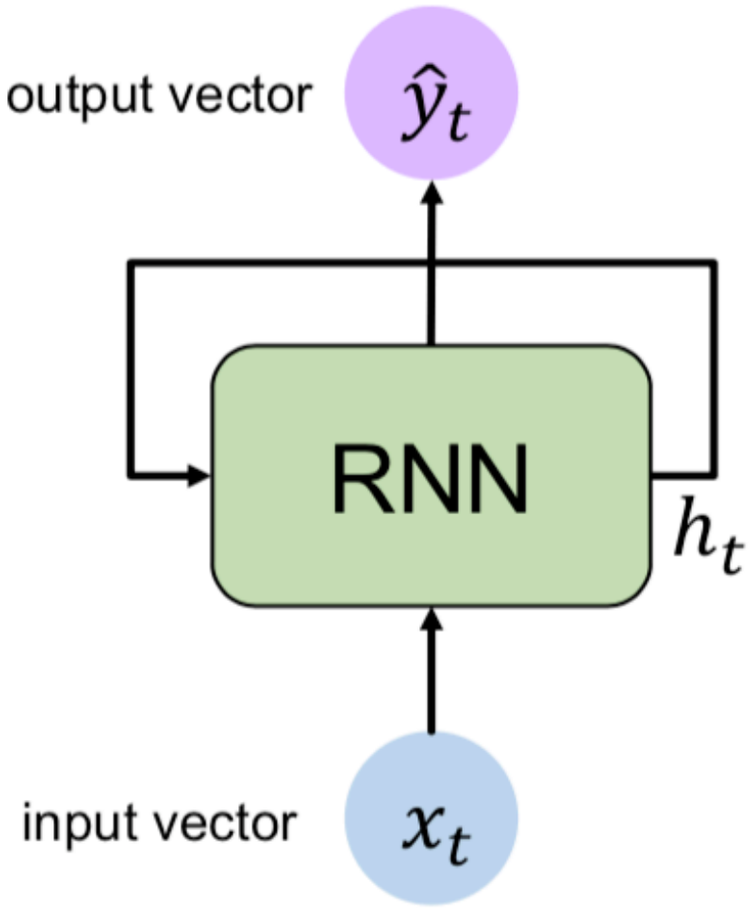


$$h_t = f_W(h_{t-1}, x_t)$$

new state — function parameterized by W — old state — input vector at time step t

h_t is the "memory" of the network. It's calculated based on the previous hidden state and the input at the current step.

Recurrent neural networks

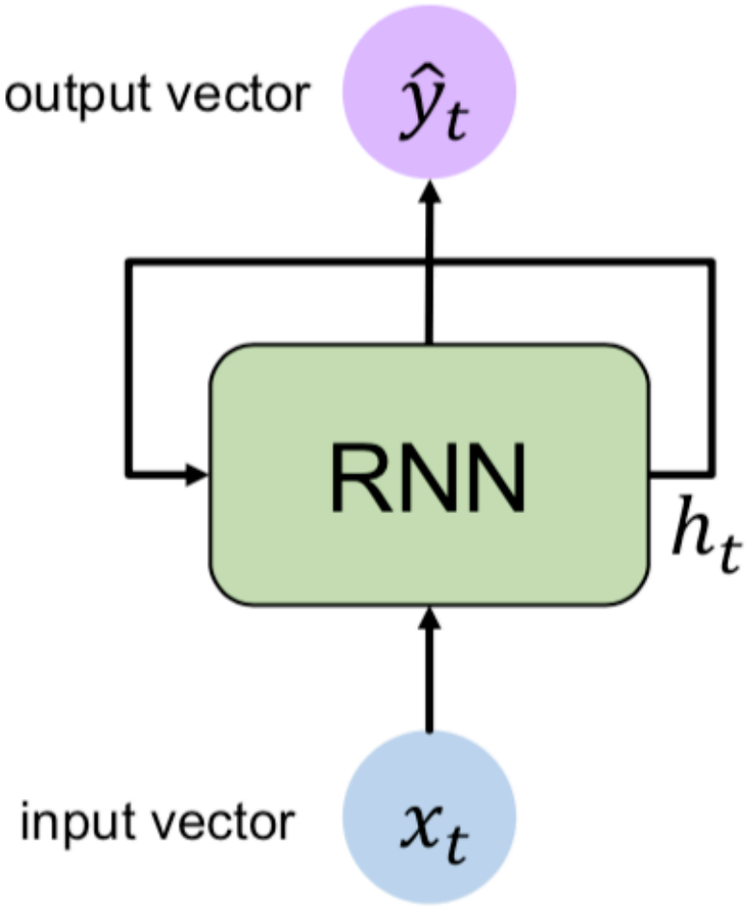


$$h_t = f_W(h_{t-1}, x_t)$$

new state — function parameterized by W — old state — input vector at time step t

The same function f_w and set of parameters W are used at every time step

Recurrent neural networks

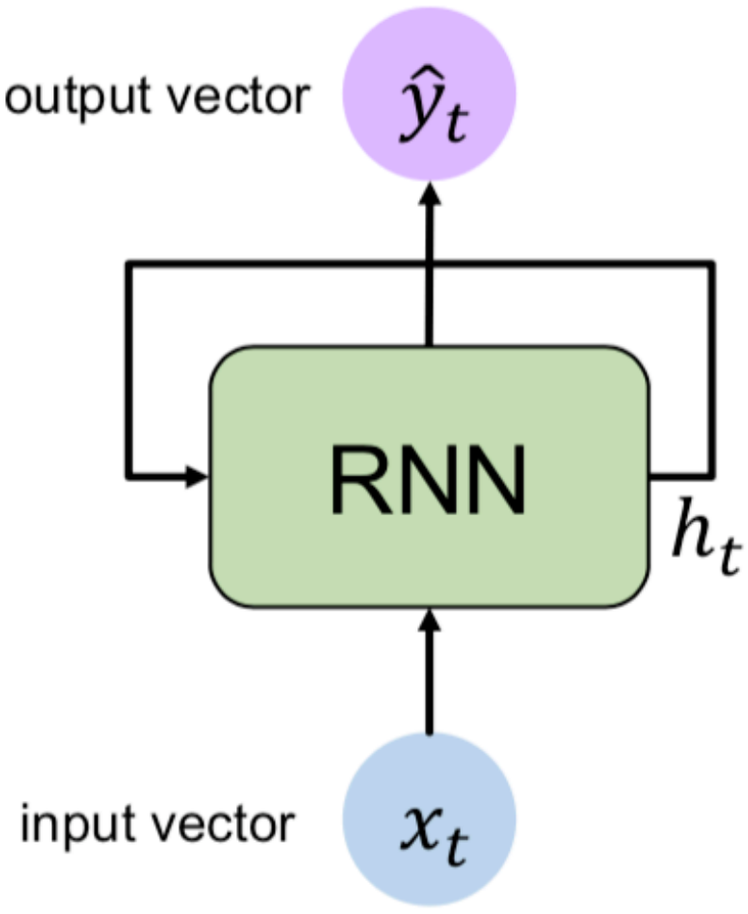


$$h_t = \tanh(\boldsymbol{W_{hh}} h_{t-1} + \boldsymbol{W_{xh}} x_t)$$

Two weight matrices

- Input vector x_t
- Previous state h_{t-1}

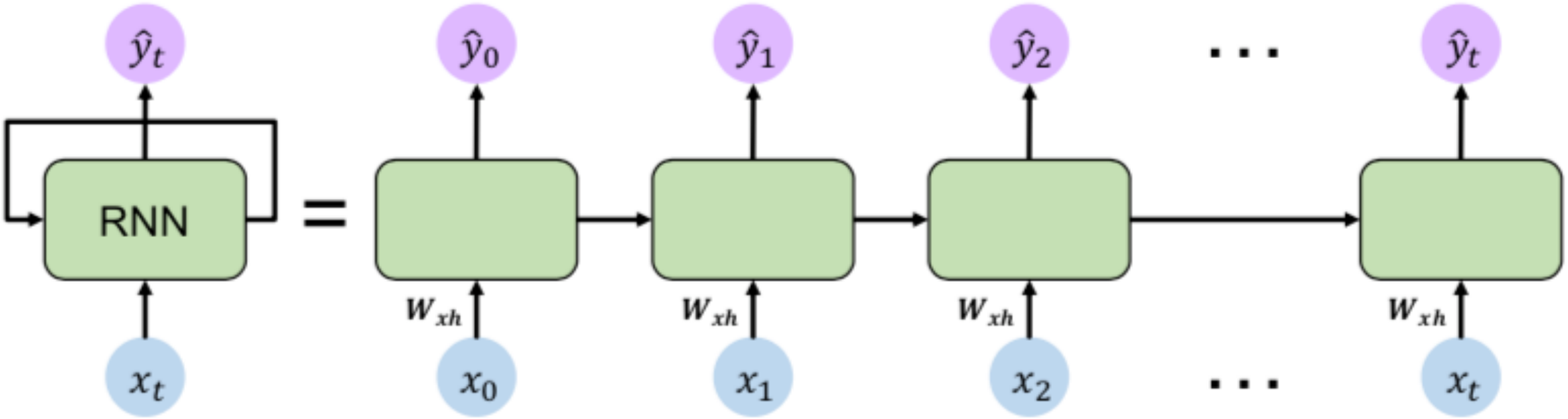
Recurrent neural networks



$$\hat{y}_t = \boldsymbol{W}_{\boldsymbol{hy}} h_t$$

$$h_t = \tanh(\boldsymbol{W}_{\boldsymbol{hh}} h_{t-1} + \boldsymbol{W}_{\boldsymbol{xh}} x_t)$$

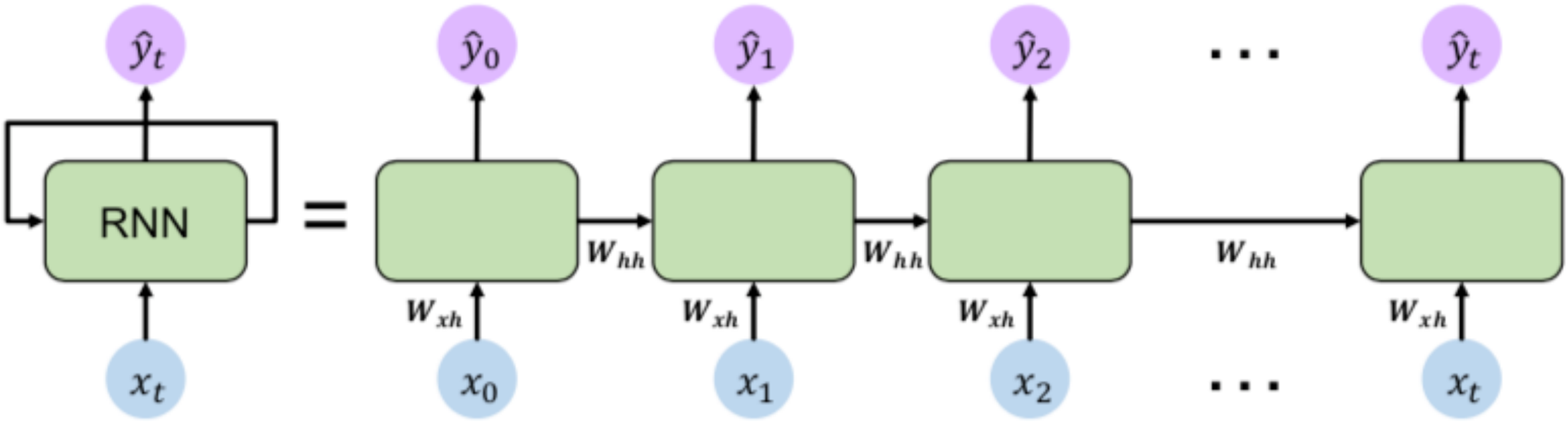
RNNs: computational graph across time



x0, x1, x2 … xt are the inputs

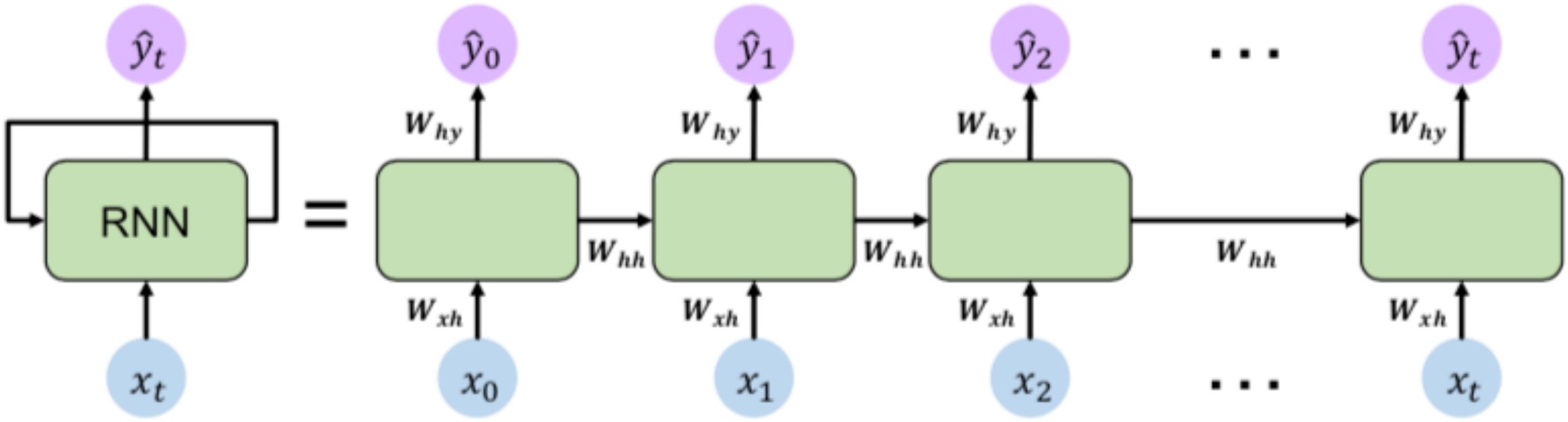
W_xh is the weight of inputs

RNNs: computational graph across time



W_hh is the weight of hidden state cell.

RNNs: computational graph across time



W_hy is the weight of outputs.

Recurrent neural networks

- RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations.

- With hidden state cell, they have a “memory” which captures information about what has been calculated so far.

- Unlike a traditional deep neural network, which uses different parameters at each layer, a RNN shares the same parameters (W_hh, W_xh, W_hy) across all steps. This means we are performing the same task at each step, just with different inputs. This greatly reduces the total number of parameters we need to learn.

The Problems Of RNNs

- RNNs can learn to use the past information when the gap between the relevant information and the place that it's needed is small.

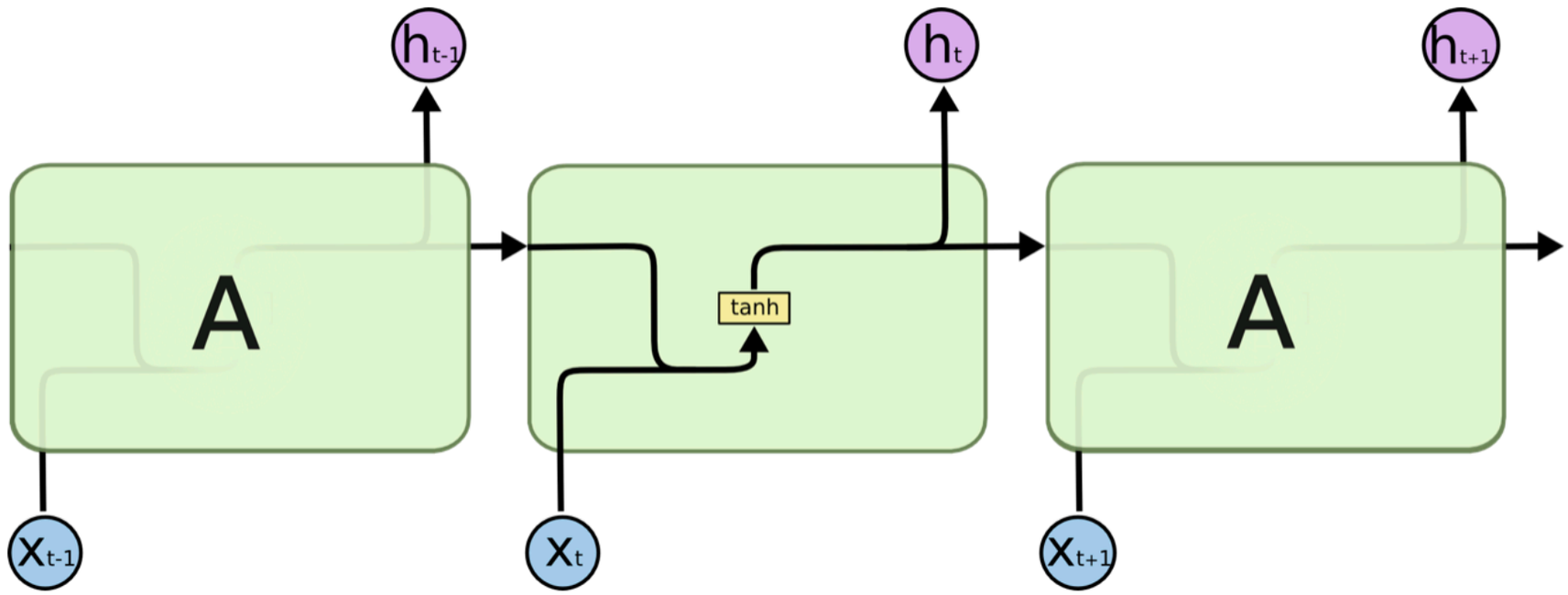
- But when the gap between the relevant information is large, RNNs are unable to learn to connect the relevant information!

LSTM networks

Long Short Term Memory networks (LSTMs) are a special kind of RNN, are a special kind of RNN, capable of learning **long-term dependencies.**

Standard RNNs VS LSTMs

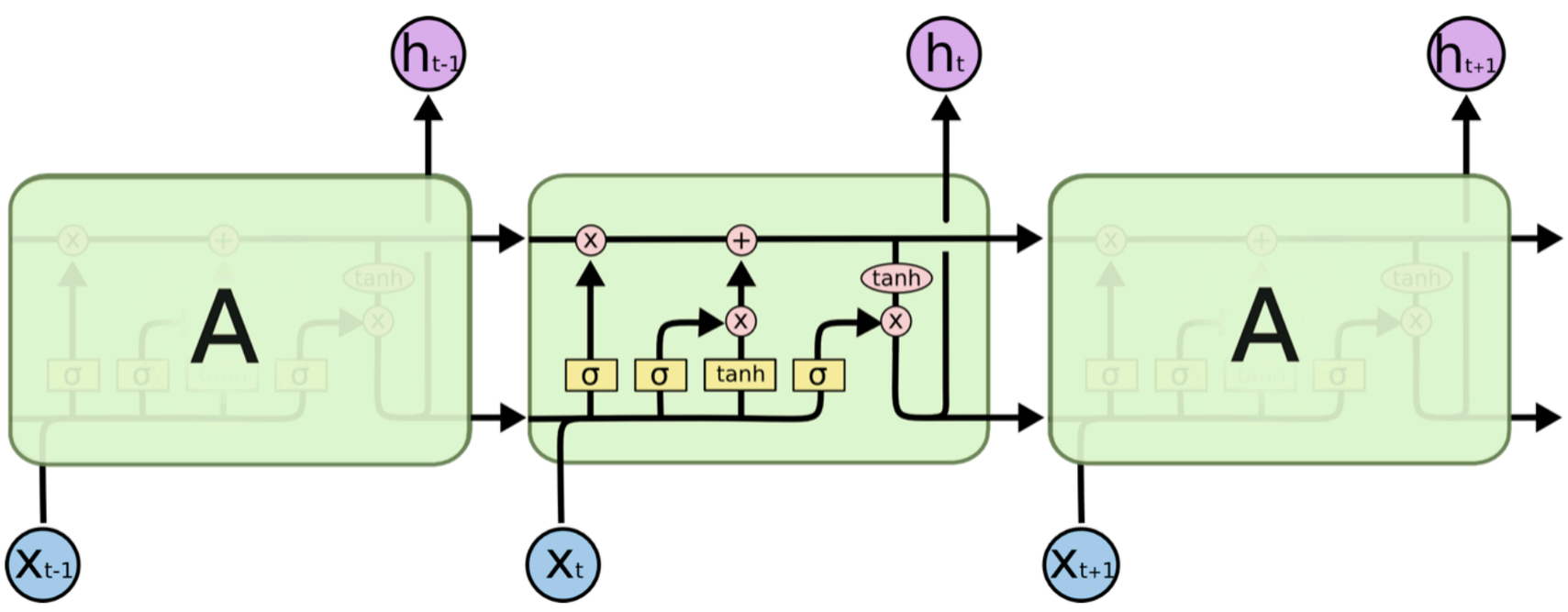
In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.



The repeating module in a standard RNN contains a single layer.

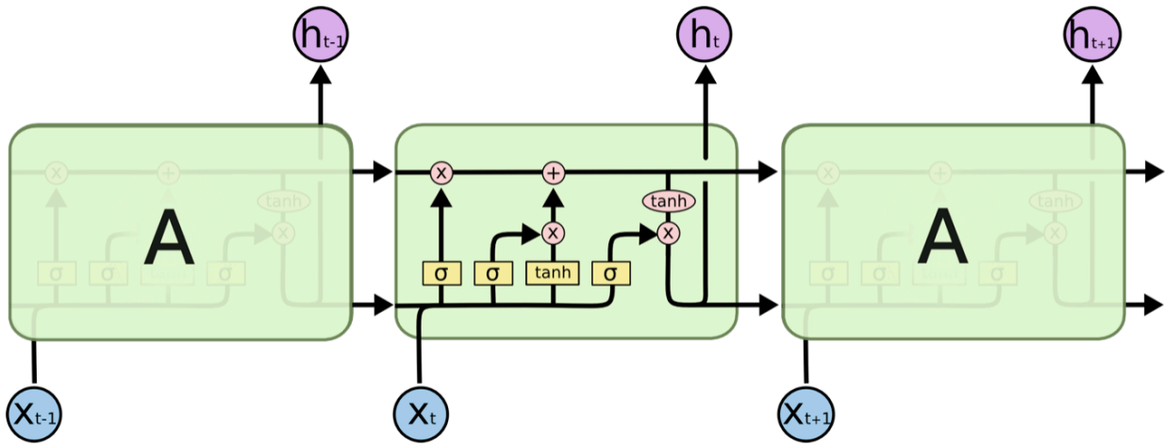
Standard RNNs VS LSTMs

- In LSTMs, the repeating module has four neural network layers, instead of one, interacting in a very special way.

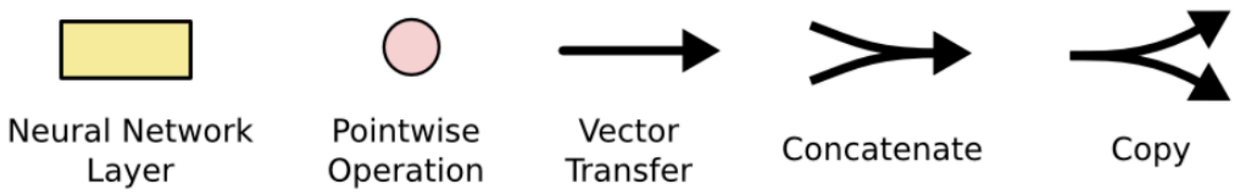


The repeating module in an LSTM contains four interacting layers.

Symbols Of LSTMs



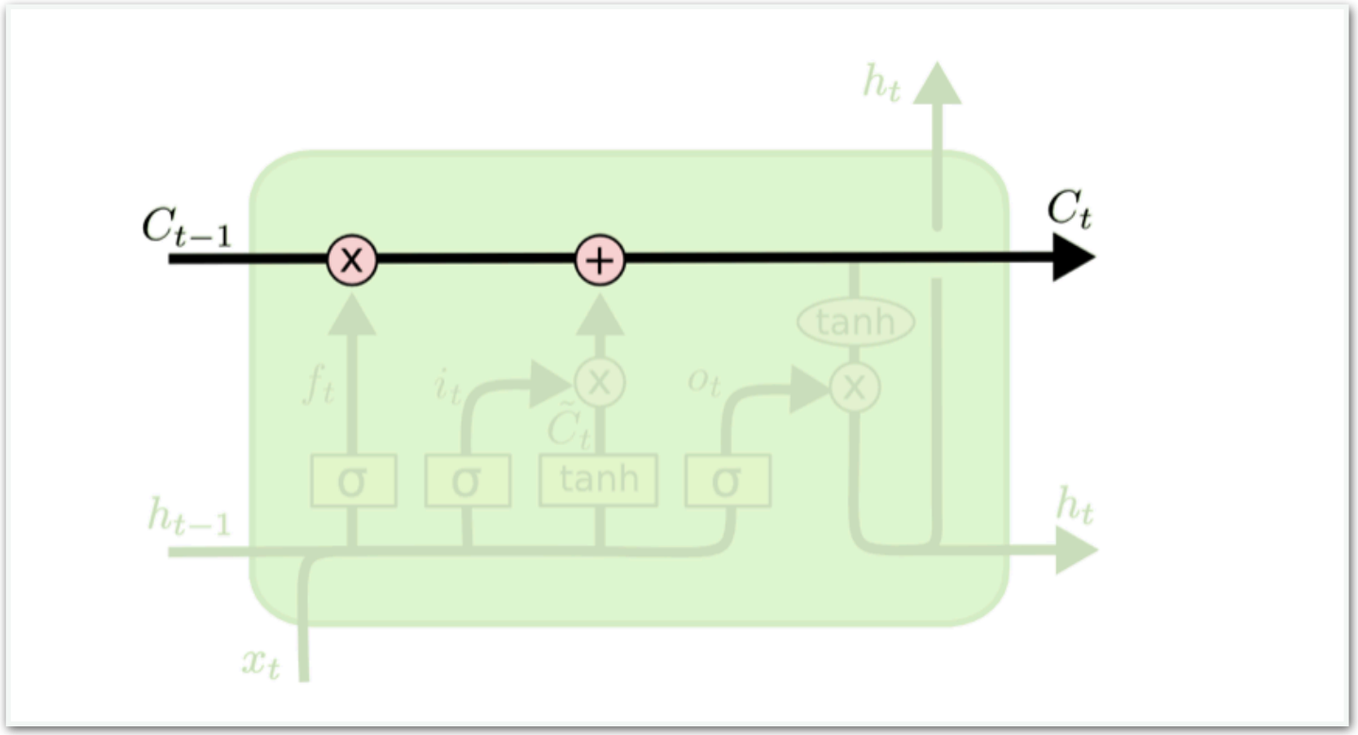
The repeating module in an LSTM contains four interacting layers.



- Each line carries an entire vector, from the output of one node to the inputs of others.
- The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers.
- Lines merging denote concatenation, while a line forking denote its content being copied and the copies going to different locations.

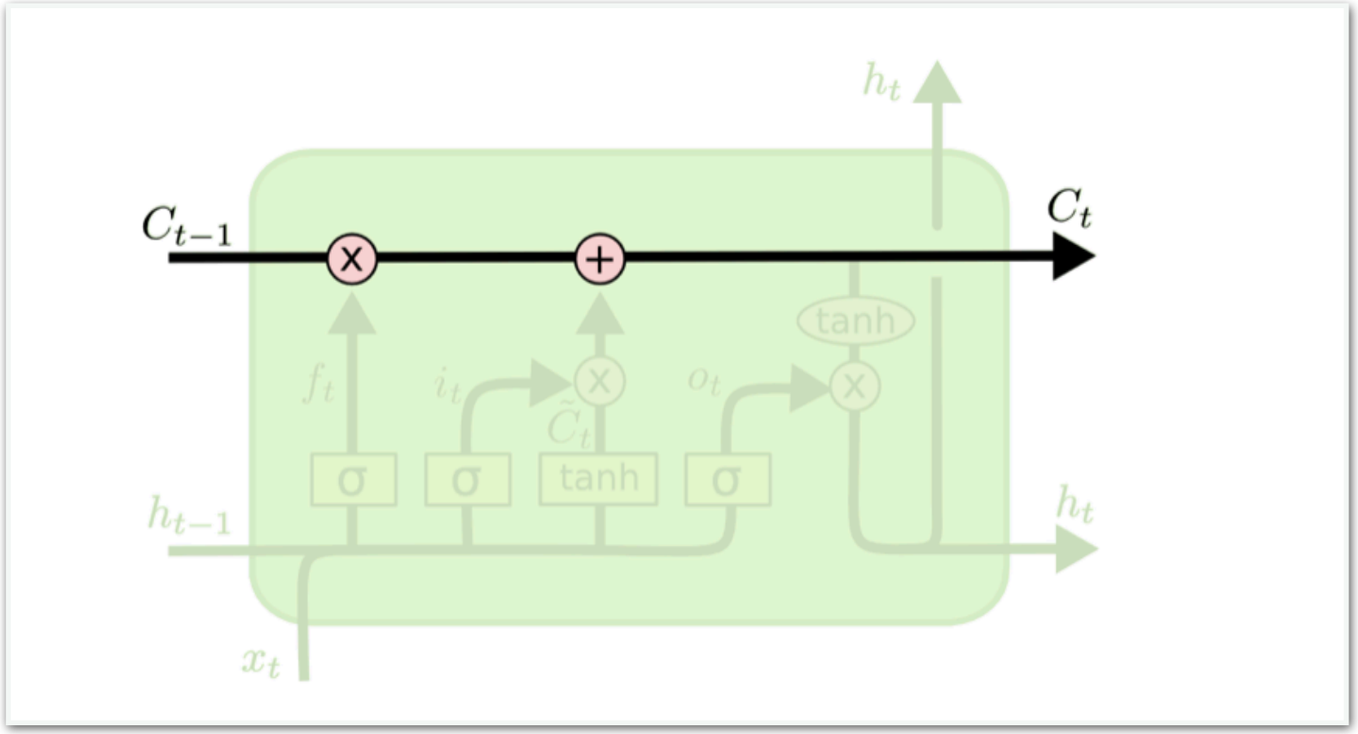
The Core Idea Behind LSTMs

- **Cell state** is the horizontal line running through the top of the diagram, which is the **key** to LSTMs



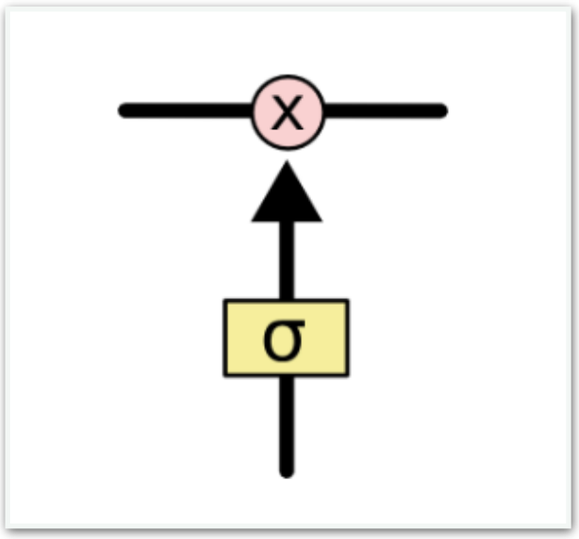
The Core Idea Behind LSTMs

- Cell state runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged.



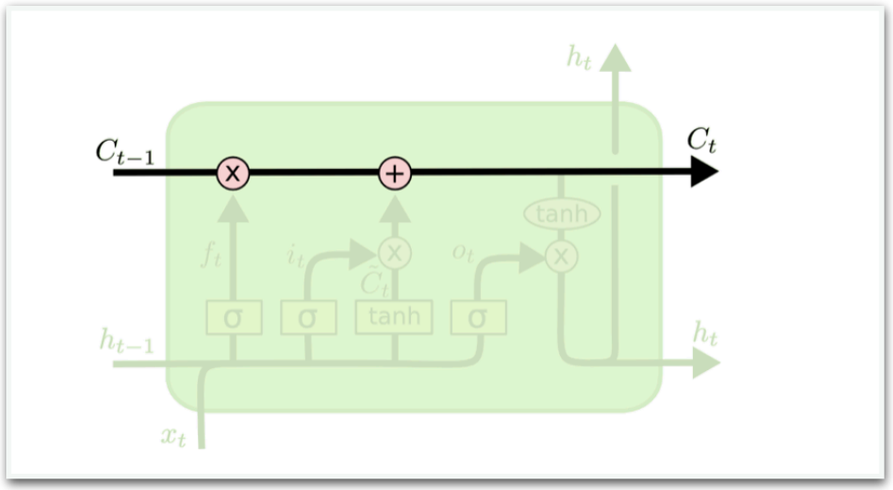
The Core Idea Behind LSTMs

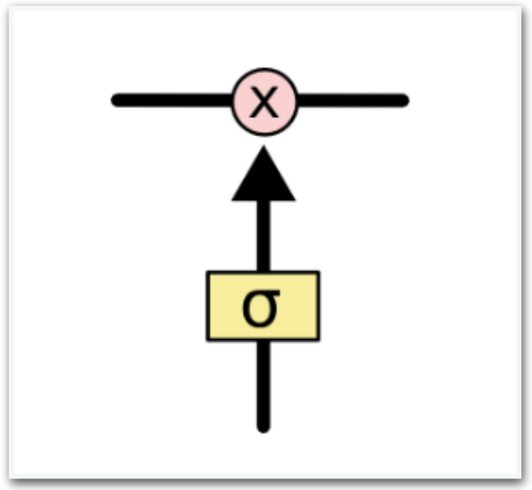
- **Gates** are a way to optionally let information through. They are composed out of a **sigmoid** neural net layer and a pointwise **multiplication** operation.



The Core Idea Behind LSTMs

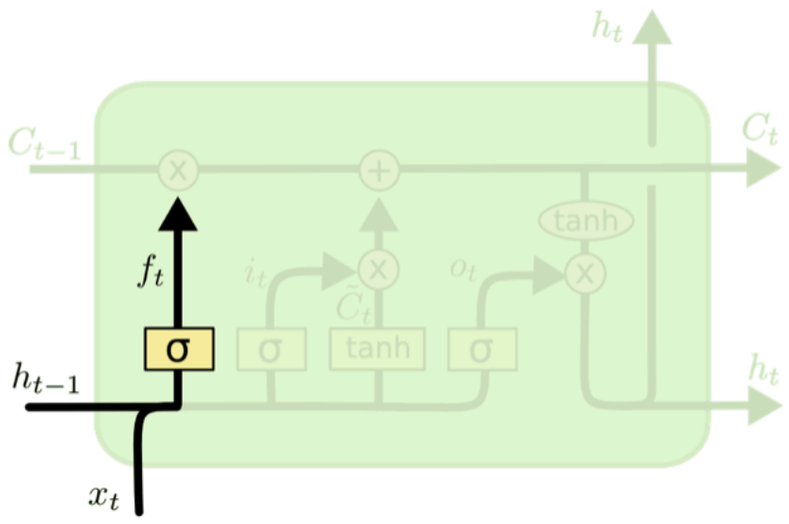
- The LSTM use gates remove or add information to the cell state.





Step-by-Step LSTM

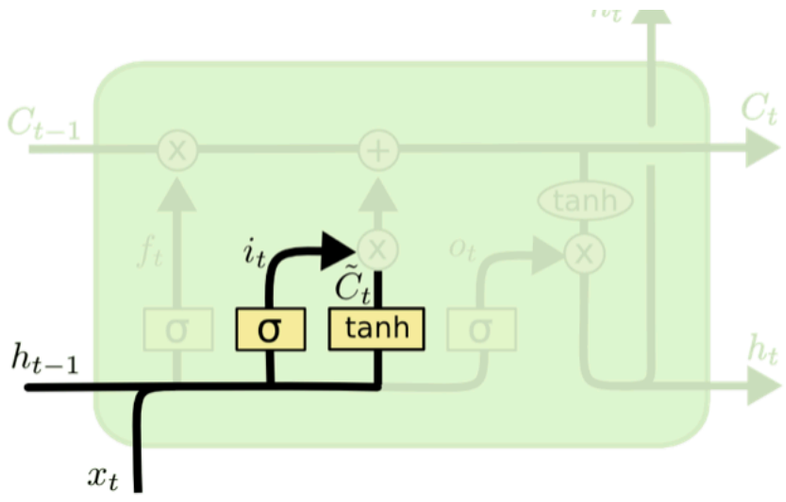
- The first step in our LSTM is to decide what information we're going to throw away from the cell state.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Step-by-Step LSTM

- The second step is to decide what new information we're going to store in the cell state.

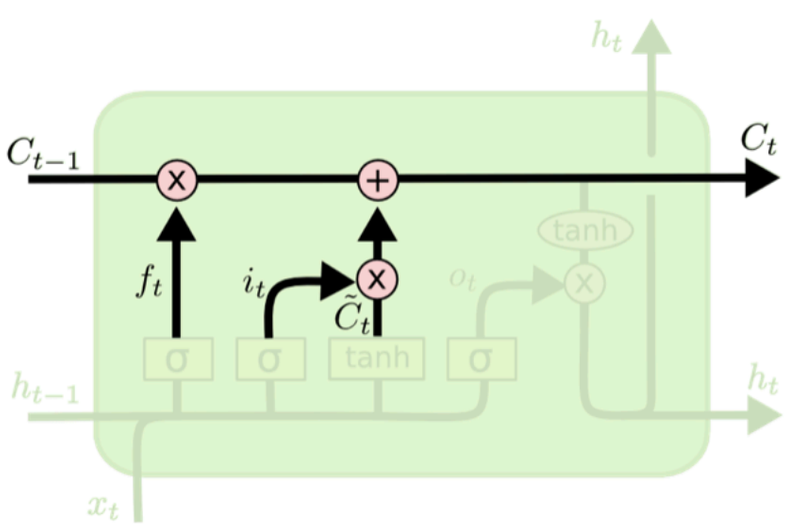


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Step-by-Step LSTM

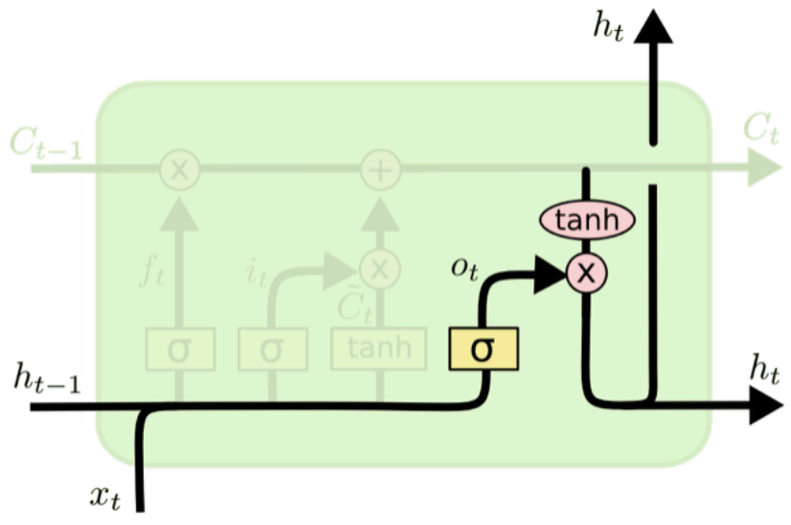
- The third step is update the old cell state into the new cell state



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Step-by-Step LSTM

- Finally, we need to decide what we're going to output



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

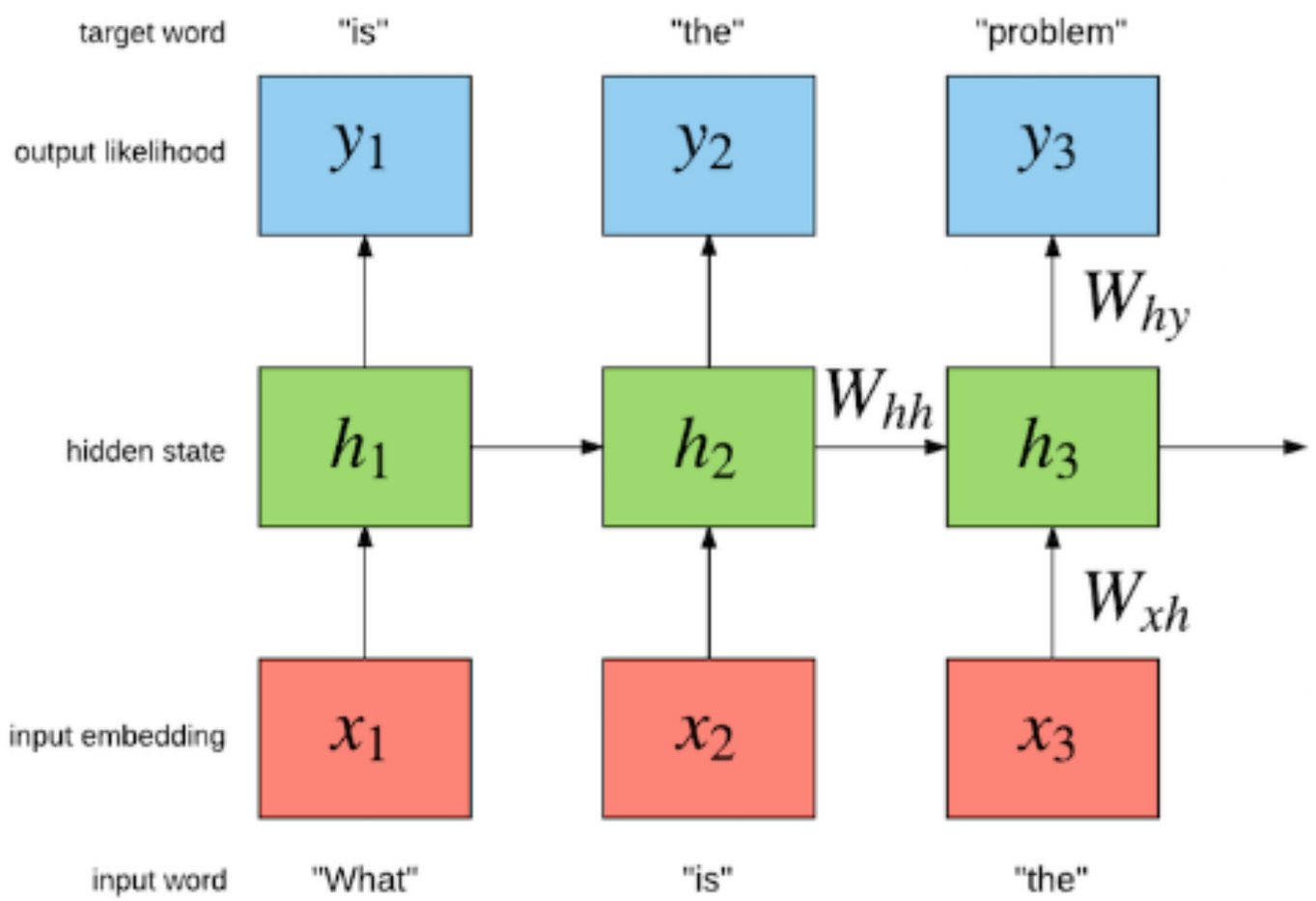
LSTMs: key concepts

1. Maintain a separate cell state from what is outputted
2. Use **gates** to control the flow of information
   - Forget gate gets rid of irrelevant information
   - Selectively update cell state
   - Output gate returns a filtered version of the cell state

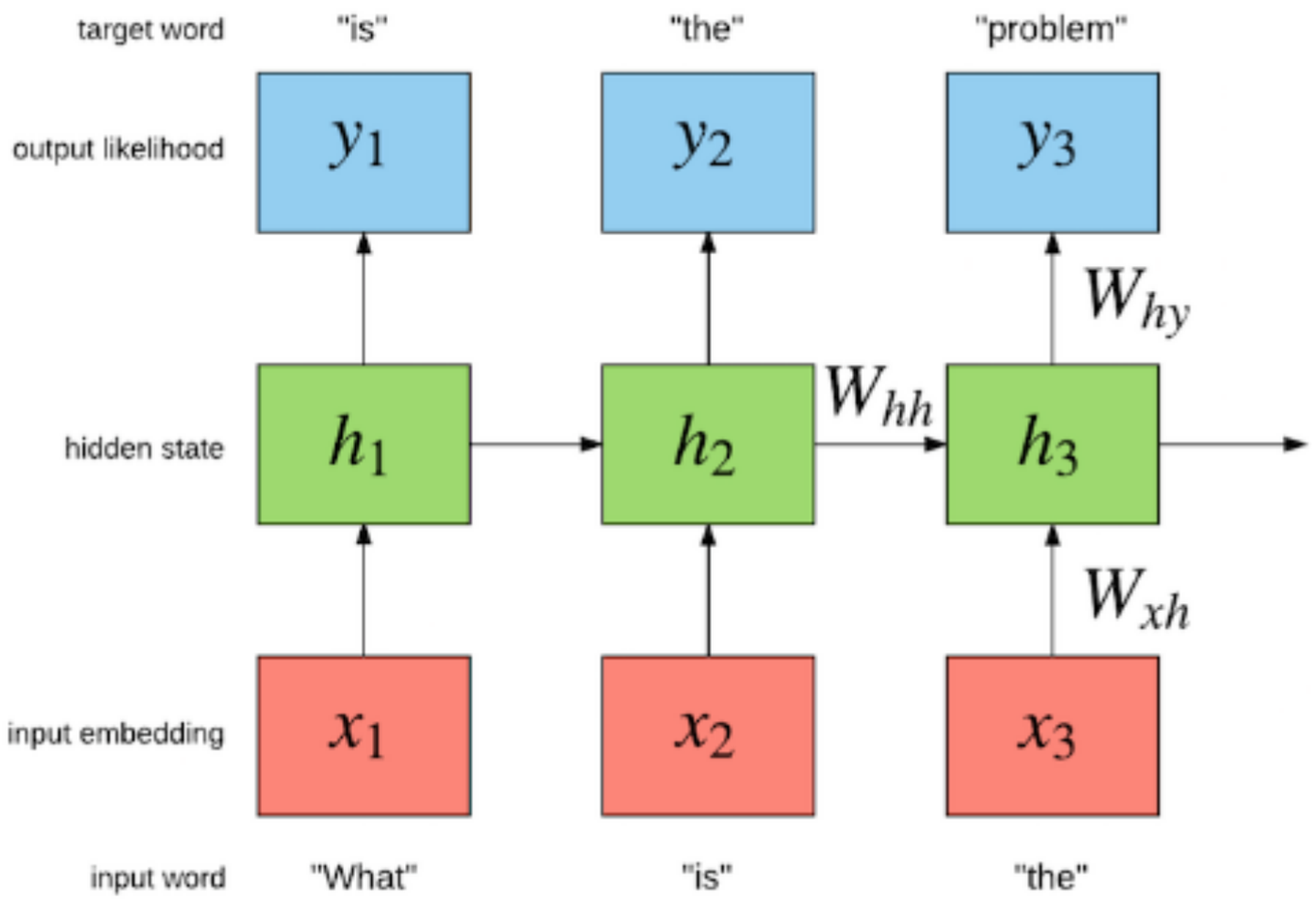
Application of RNNs

- Language Model

- Language Model



Language Model

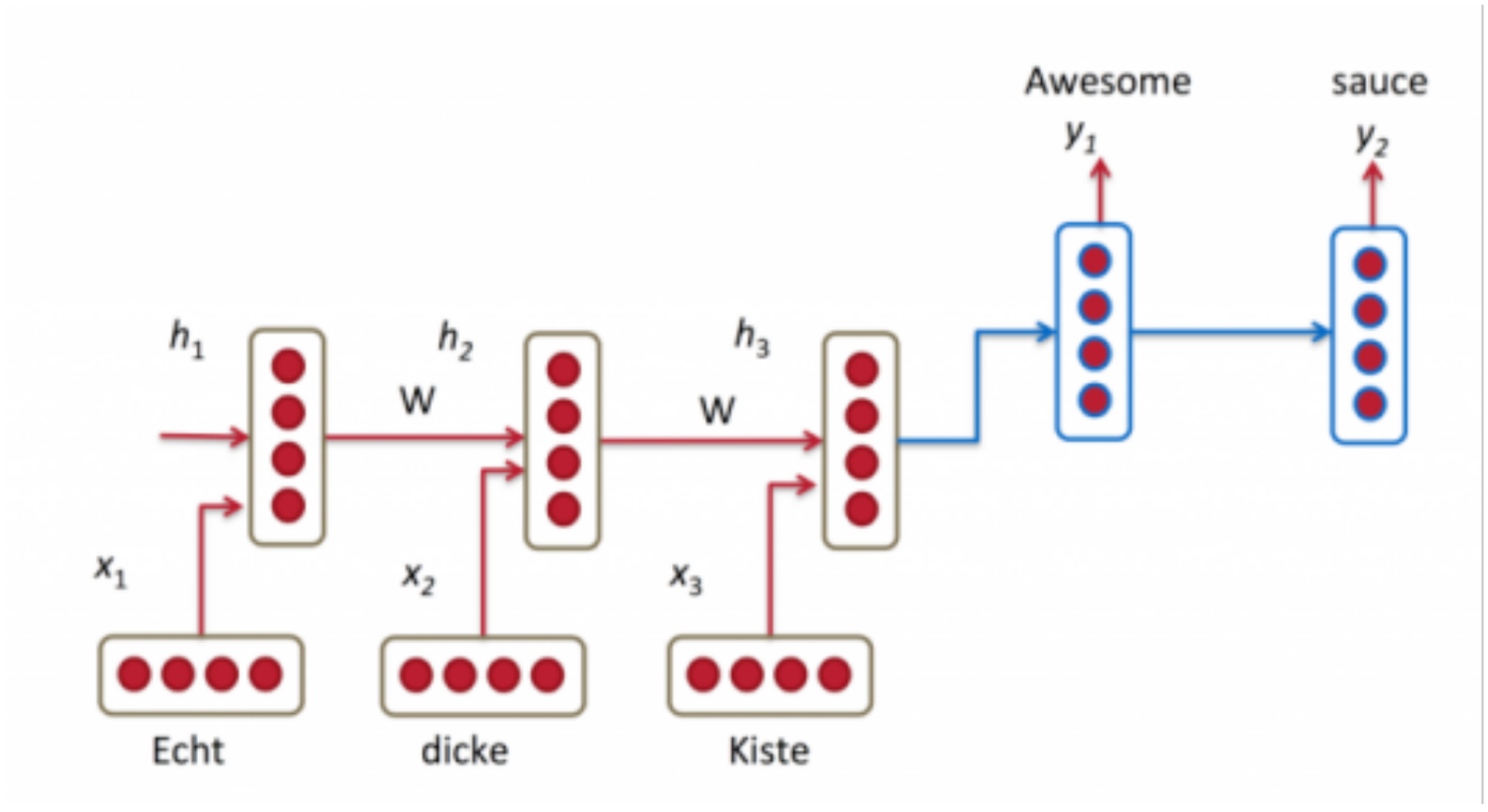


Writing a poem

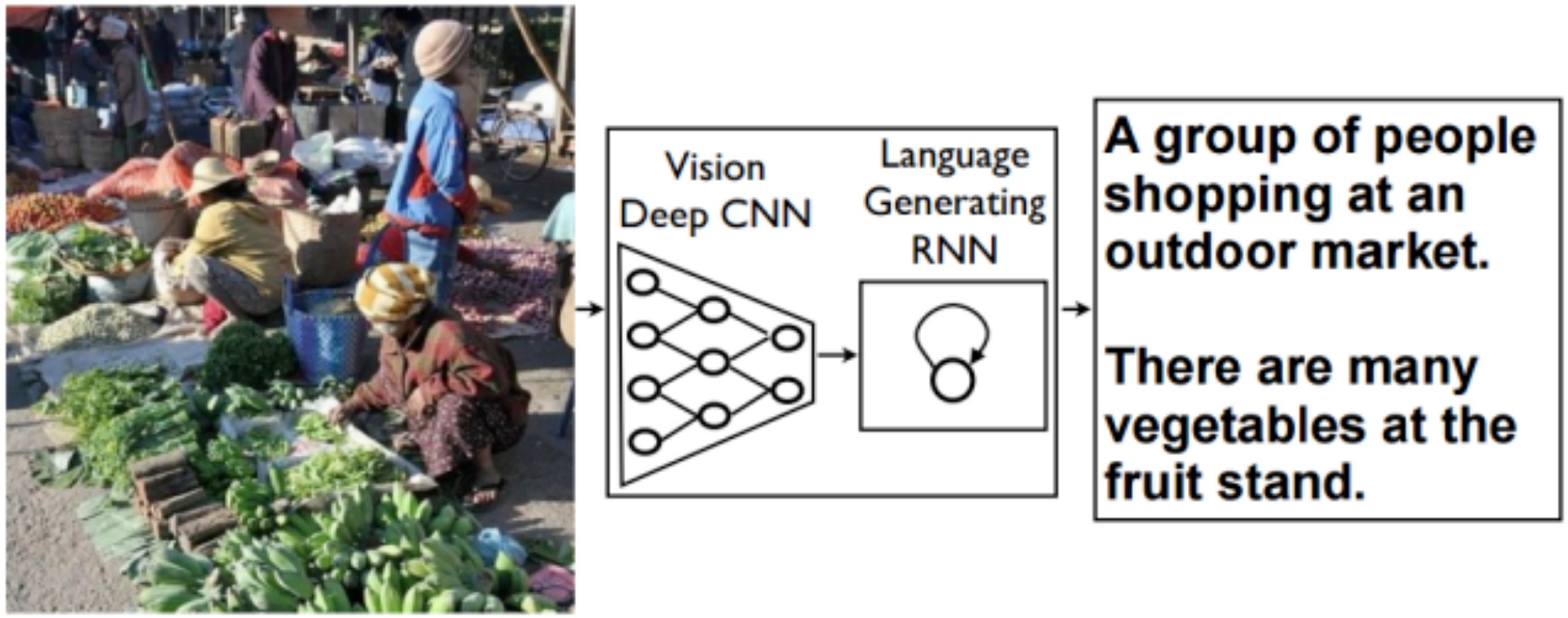
白鹭窥鱼立, Egrets stood, peeping fishes. 青山照水开. Water was still, reflecting mountains. 夜来风不动, The wind went down by nightfall, 明月见楼台. as the moon came up by the tower.	满怀风月一枝春, Budding branches are full of romance. 未见梅花亦可人. Plum blossoms are invisible but adorable. 不为东风无此客, With the east wind comes Spring. 世间何处是前身. Where on earth do I come from?

Machine Translate

- An RNN for coding
- Another RNN for decoding

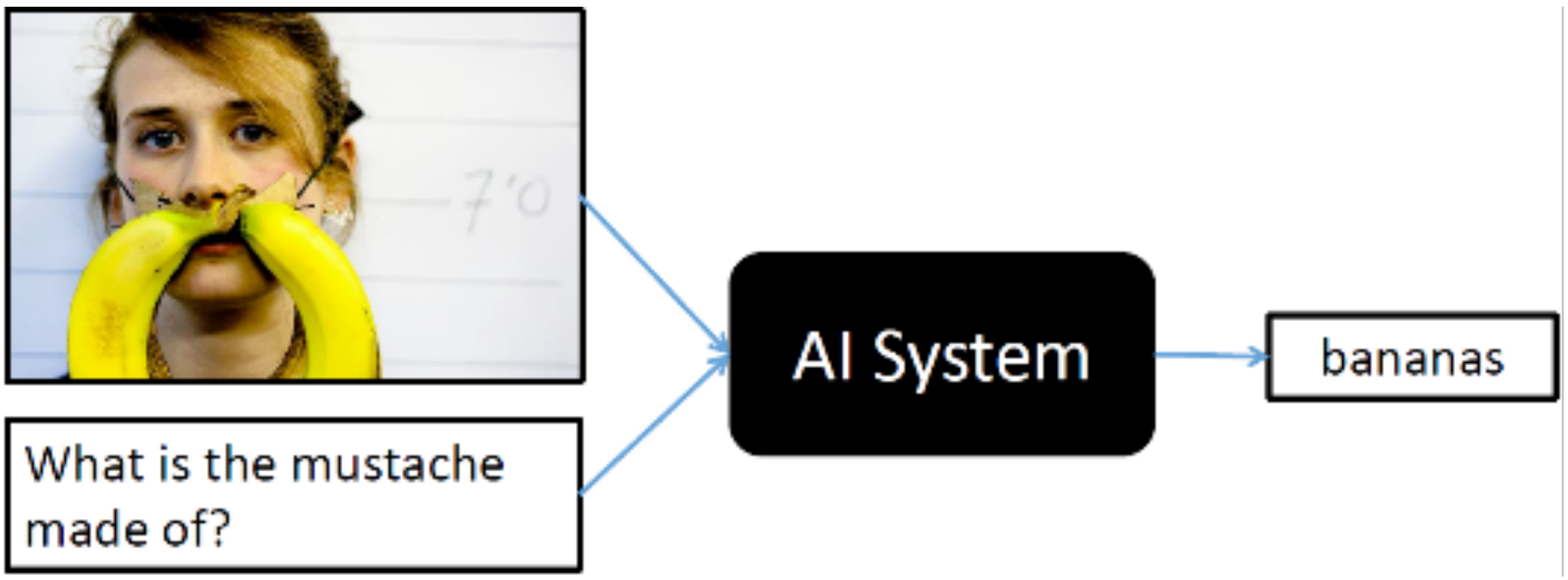


Talking about pictures



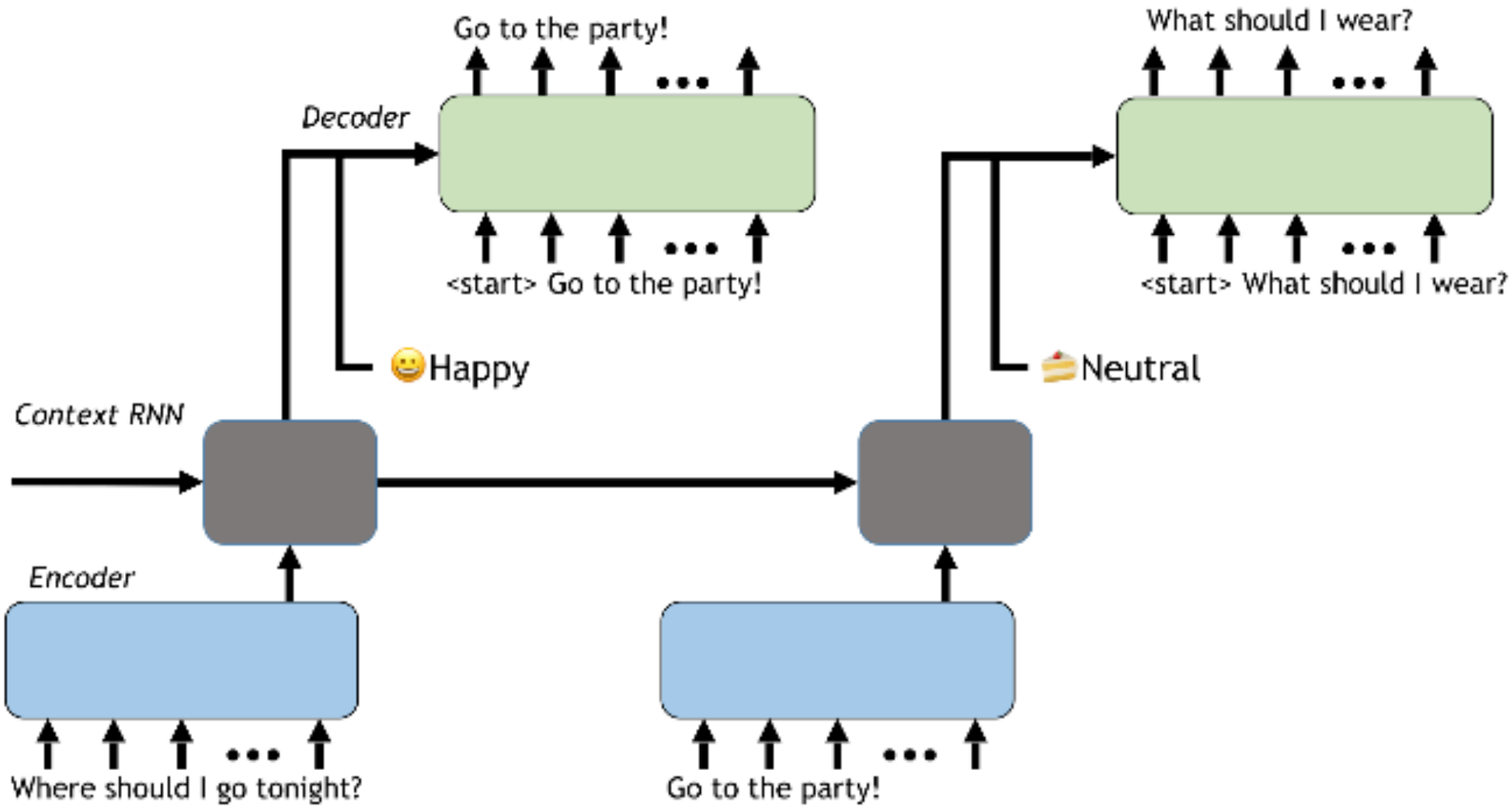
Visual Question Answering (VQA)

VQA: Given an image and a natural language question about the image, the task is to provide an accurate natural language answer



Dialogue system

https://github.com/lukalabs/cakechat



Thank you!