CSCM77
RNN & LSTM

Prof. Xianghua Xie
x.xie@swansea.ac.uk
http://csvision.swan.ac.uk

A brief overview of all networks so far

- Fully connected feedforward neural network

- Convolutional neural network

- Region proposal network and RCNN

  - Fixed length input
  - Fixed length output
  - Data and error propagate uni-directionally

Sequential data

- Consider automated image captioning

A cat sitting on a suitcase on the floor

A cat is sitting on a tree branch

A dog is running in the grass with a frisbee

A white teddy bear sitting in the grass

Two people walking on the beach with surfboards

A tennis player in action on the court

Two giraffes standing in a grassy field

A man riding a dirt bike on a dirt track

Sequential data

- Further examples
  - Text translation
  - Stock price prediction
  - Frame-wise video classification
  - Sentiment classification
- Sequences might be of arbitrary or even infinite length

- Data inside a sequence are non identically, independently distributed (iid)
  - Context is important: the next word depends on (all) previous words
  - Thus, we need memory to model context

Memory

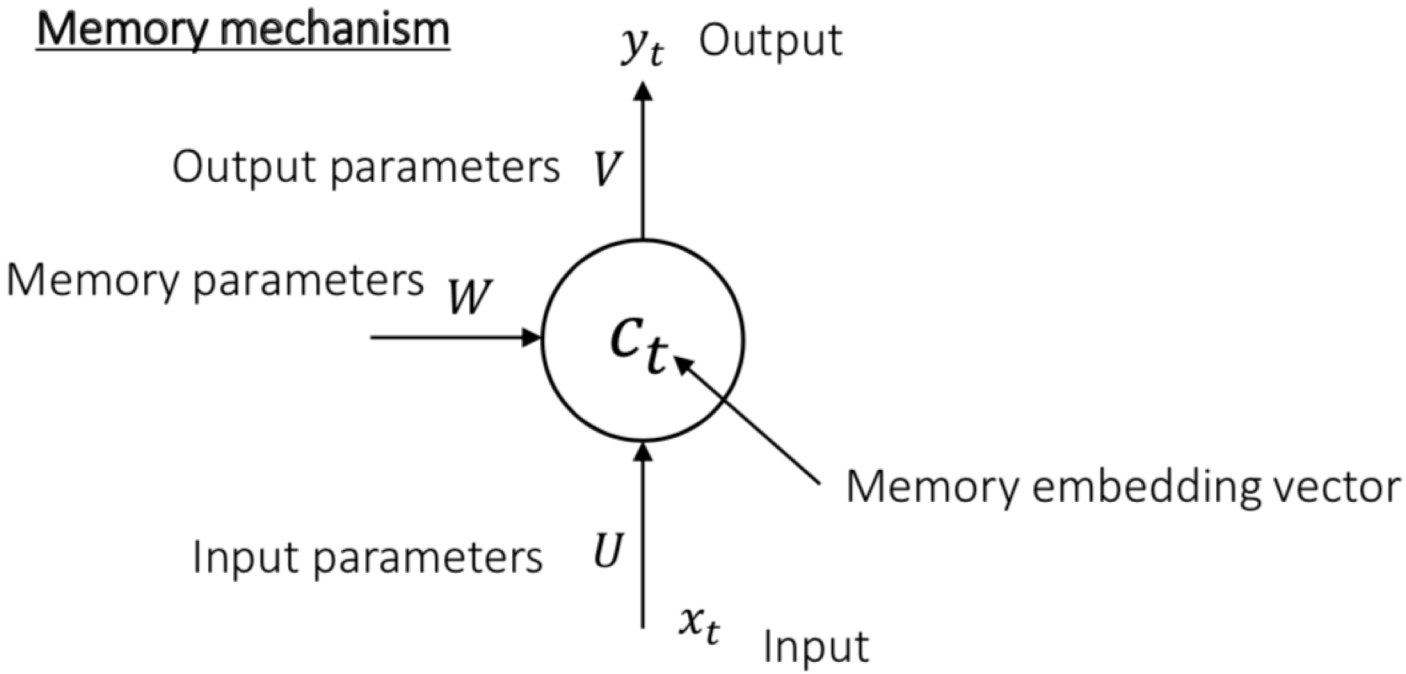
- Memory is a mechanism that learns a representation of the past
- At timestep t project all previous information 1, … , t onto a latent/hidden space c_t
  - Memory controlled by a neural network h_θ with shared parameters θ
  - Then, at timestep t + 1 re-use the parameters θ and the previous c_t

$$c_{t+1} = h_\theta(x_{t+1}, c_t)$$

$$c_{t+1} = h_\theta(x_{t+1}, h_\theta(x_t, h_\theta(x_{t-1}, \dots h_\theta(x_1, c_0))))$$

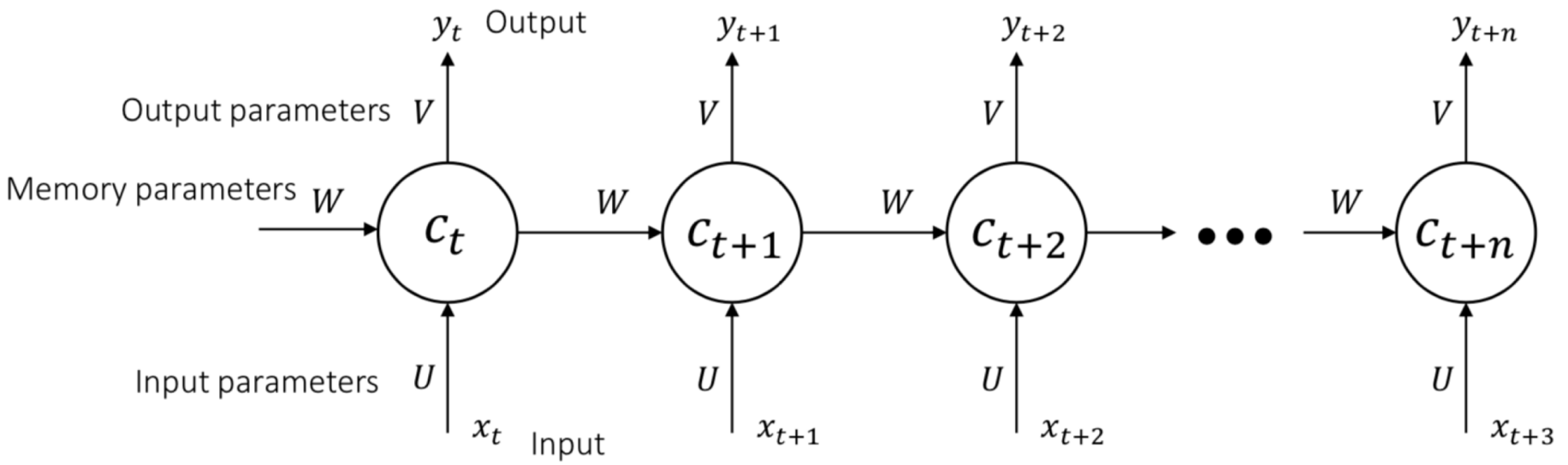
Memory: graphical representation

- Sequence inputs: we model them with parameters U
- Sequence outputs: we model them with parameters V
- Memory I/O: we model it with parameters W



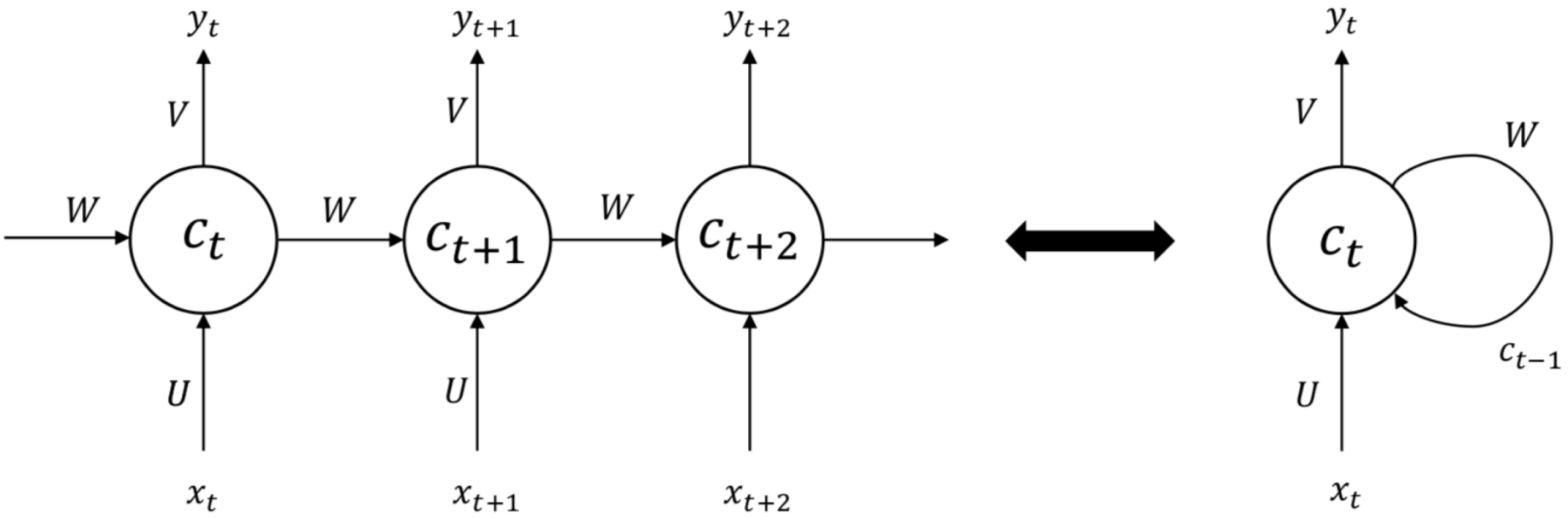
Memory: graphical representation

- Sequence inputs: we model them with parameters U
- Sequence outputs: we model them with parameters V
- Memory I/O: we model it with parameters W



Memory: graphical representation

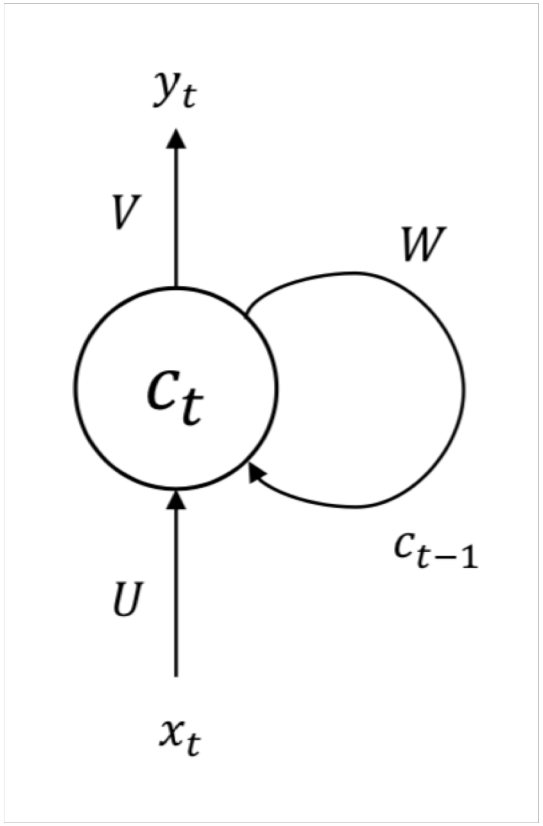
- Since the parameters (U, V, W) are shared, we can fold the memory units/cells



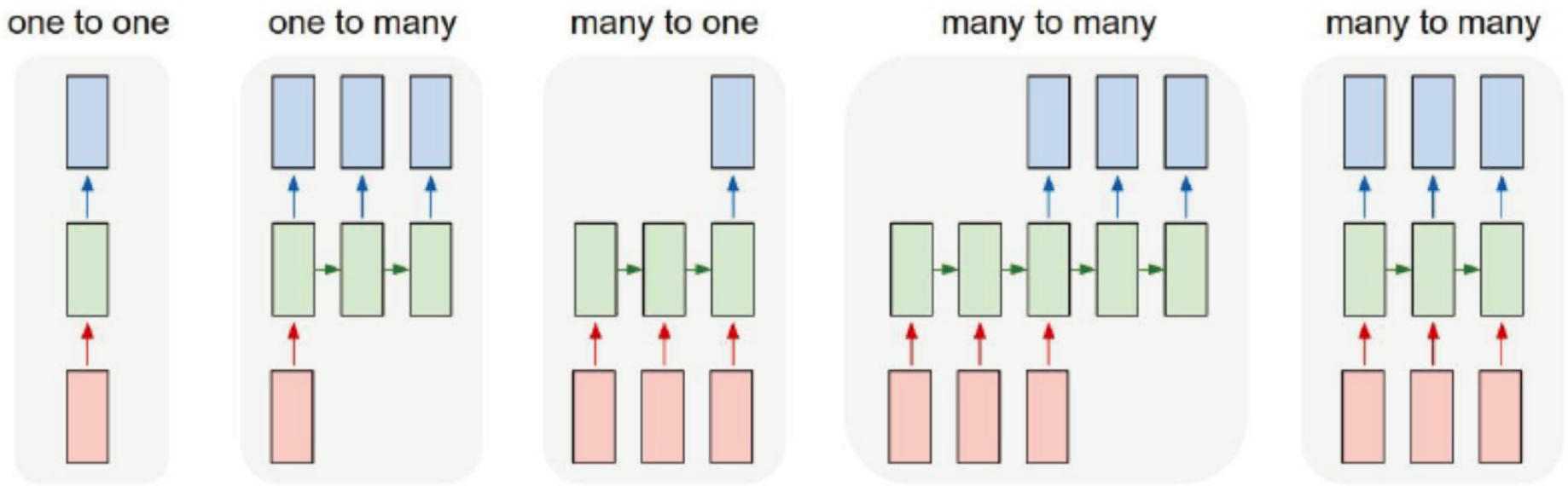
RNN: Recurrent Neural Network

- RNN
  - Have memory that keeps track of information observed so far
  - Map from the entire history of previous inputs to each output
  - Handle sequential data

$$c_t = \tanh(U\, x_t + W c_{t-1})$$
$$y_t = \text{softmax}(V\, c_t)$$



RNN: Recurrent Neural Network



Vanilla Neural Network

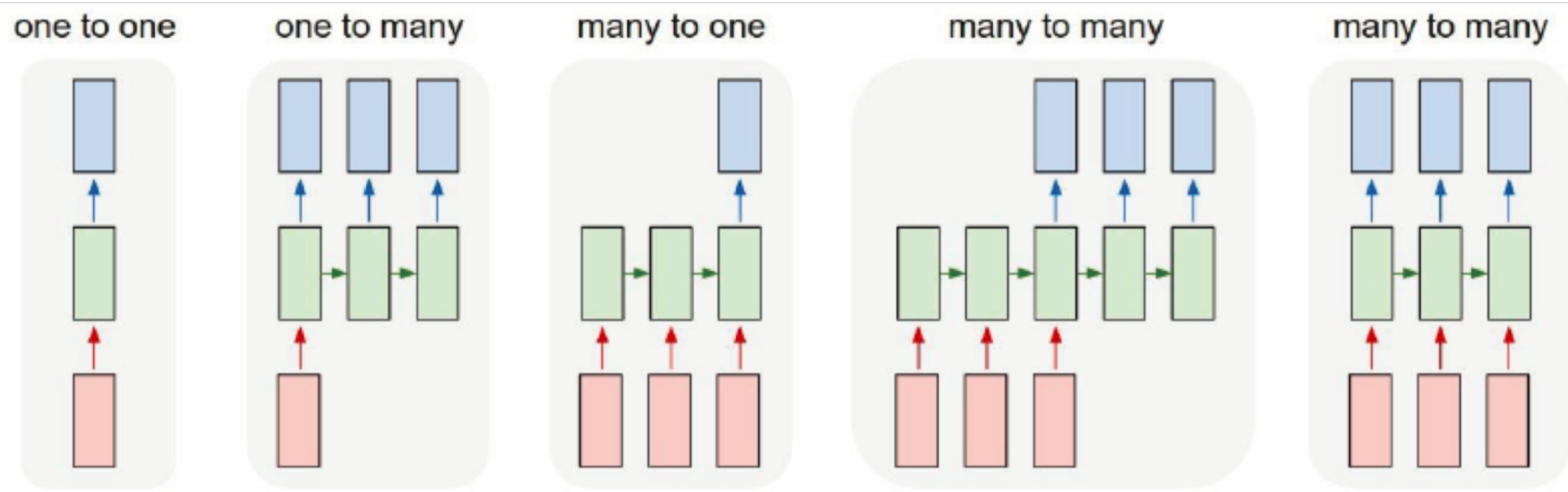
fixed-sized input -> fixed-size output

e.g. image classification

Image credit: F. Li et al.

Vector to vector

RNN: Recurrent Neural Network



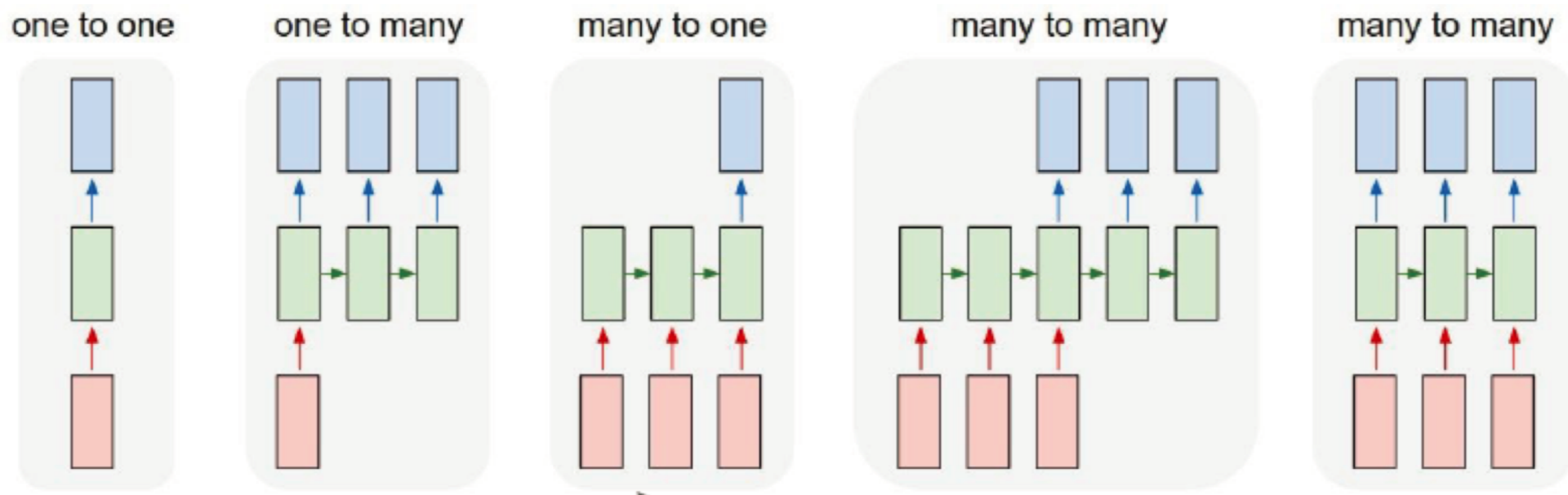
sequence output

e.g. image captioning

image -> sequence of words

Vector to sequence

RNN: Recurrent Neural Network



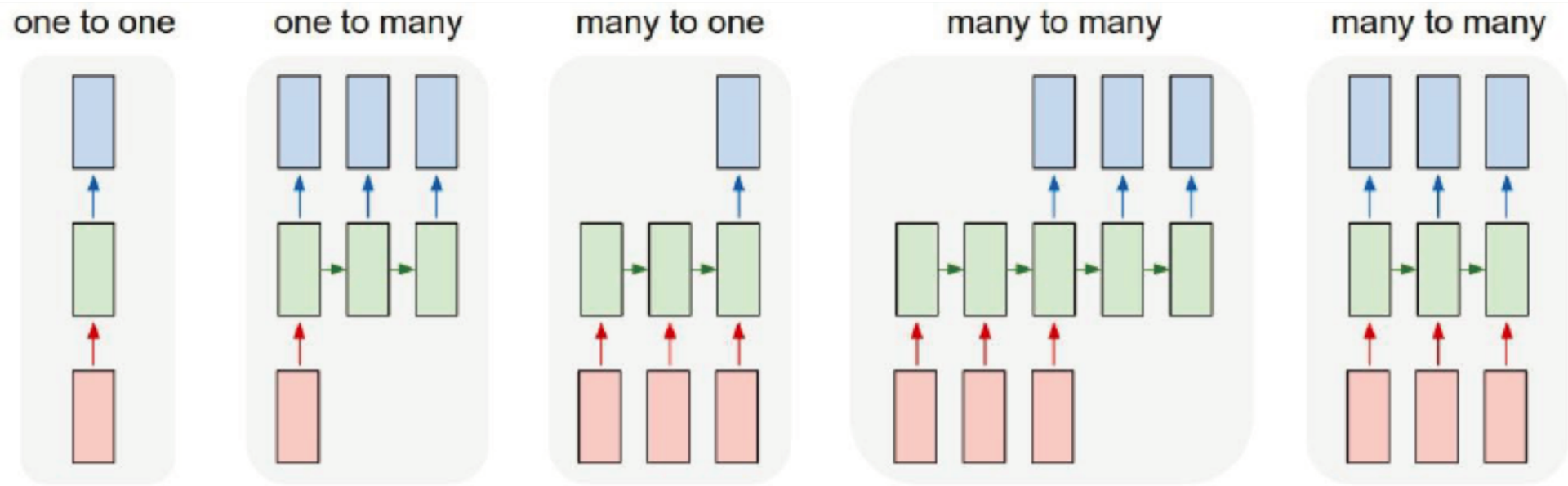
sequence input

e.g. sentiment classification

sequence of words -> sentiment

Sequence to vector

RNN: Recurrent Neural Network



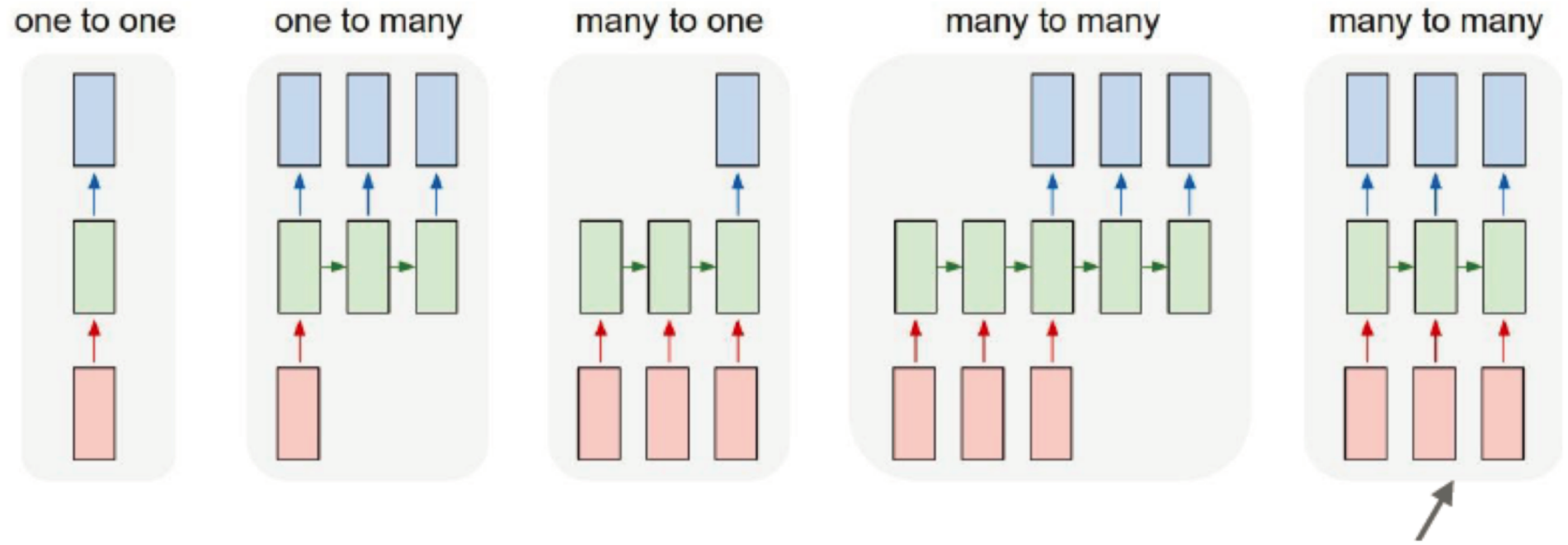
sequence input and sequence output

e.g. machine translation

seq of words -> seq of words

Sequence to sequence

RNN: Recurrent Neural Network



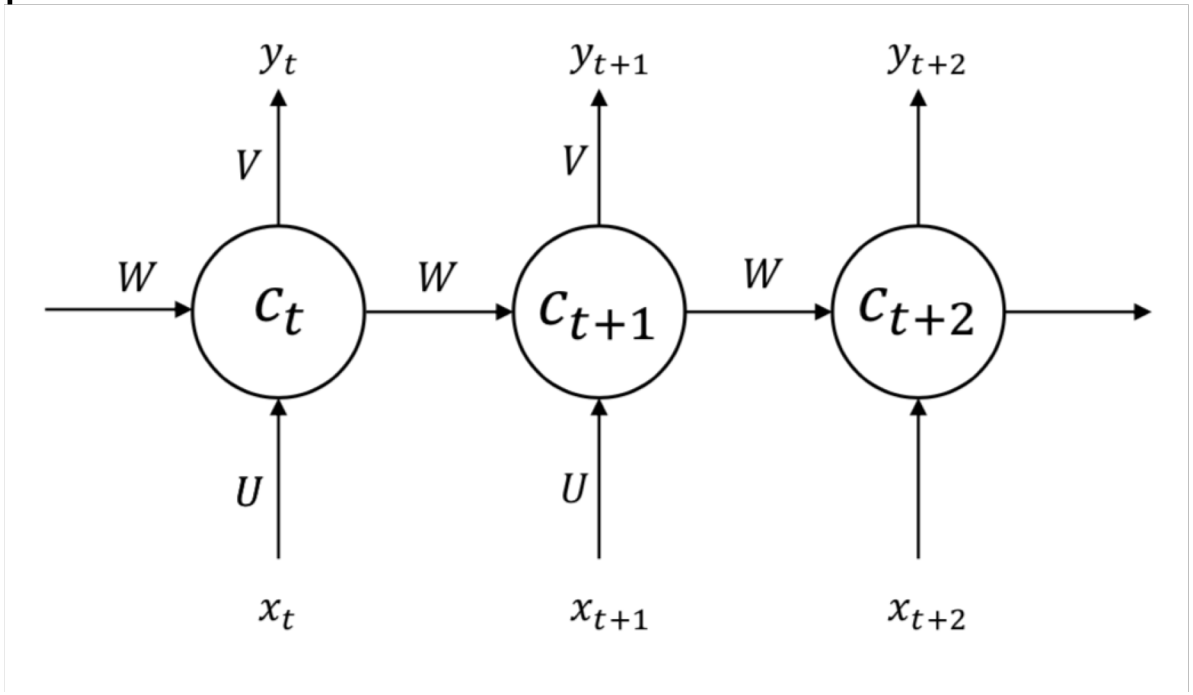
synced sequence input and output

e.g. video classification on frame level

Sequence to sequence

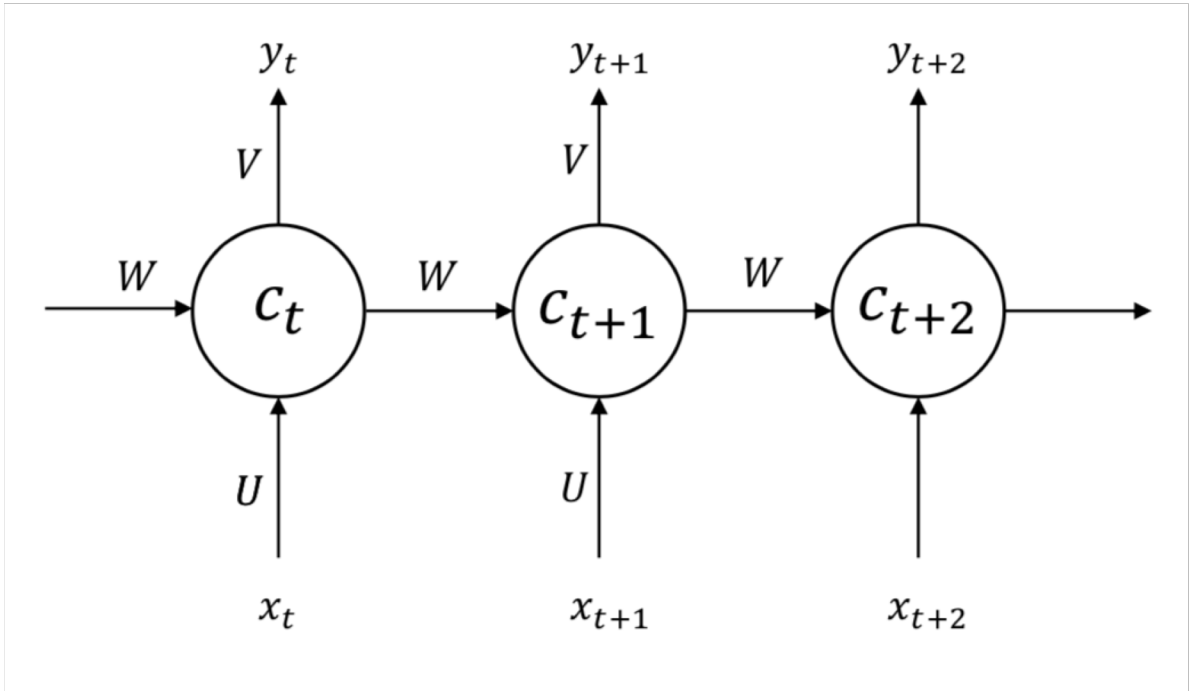
RNN: Recurrent Neural Network

- RNNs can be thought of as multiple copies of the same network, each passing a message to a successor.
- The same function and the same set of parameters are used at every time step.
  - Are called recurrent because they perform the same task for each input.



Back-Propagation Through Time (BPTT)

- Using the generalised back-propagation algorithm one can obtain the so-called **Back-Propagation Through Time** algorithm.
- The recurrent model is represented as a multi-layer one (with an unbounded number of layers) and backpropagation is applied on the unrolled model.



Vanishing/Exploding Gradients

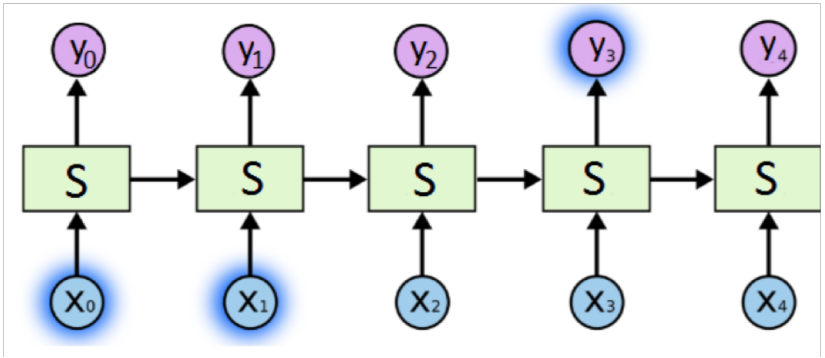
- In RNNs, during the gradient back propagation phase, the gradient signal can end up being multiplied many times.

- If the gradients are large
  - Exploding gradients, learning diverges
  - Mitigation: clip the gradients to a certain max value

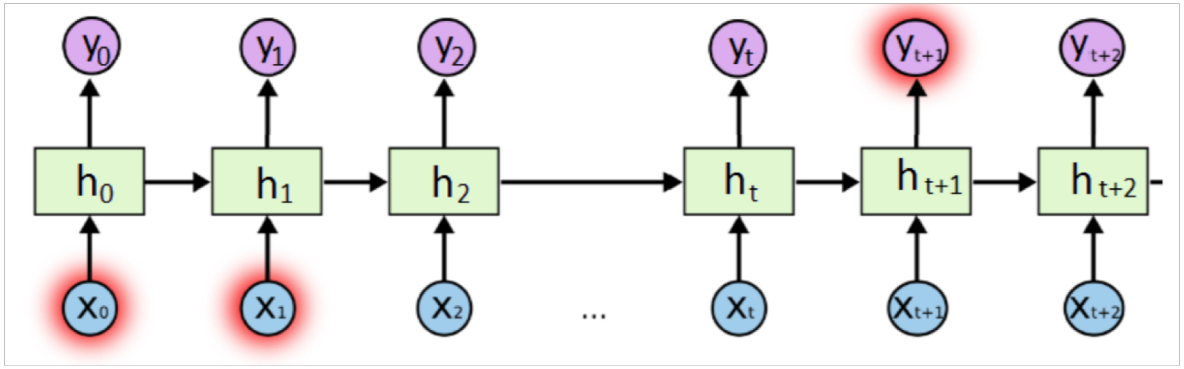
- If the gradients are small
  - Vanishing gradients, learning very slow or stops
  - Alternative architectures required

Long-term Dependencies

- RNNs connect previous information to present task:
  - may be enough for predicting the next word for ”the clouds are in the sky”



  - may not be enough when more context is needed: ”I grew up in France ... I speak fluent French”

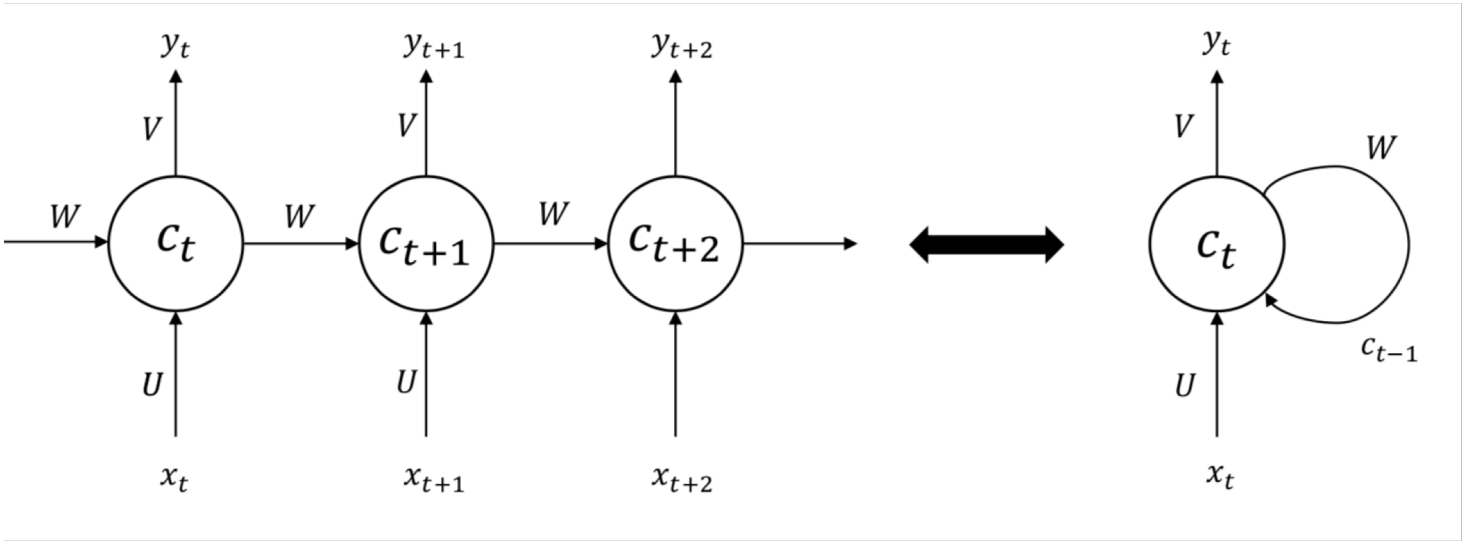


Credit: C. Olah

State Update and Memory

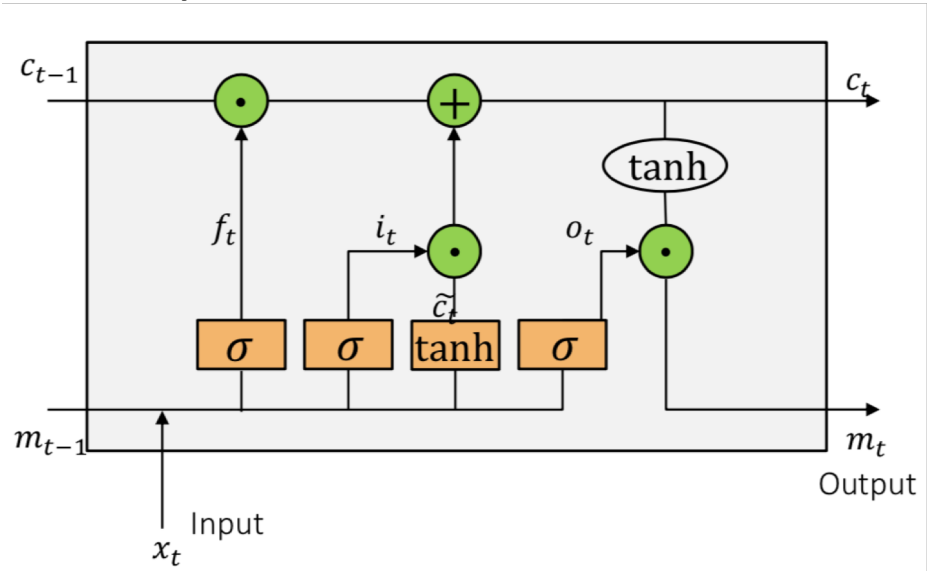
- Over time the state change is c_t+1 = c_t + Δc_{t+1}
  - This constant over-writing over long time steps leads to chaotic behaviour

- Are all inputs important enough to write them down?
- Are all outputs important enough to be read?
- Is all information important enough to be remembered over time?



LSTM: Long Short-Term Memory Networks

- Long Short-Term Memory (LSTM) networks are RNNs capable of learning long-term dependencies

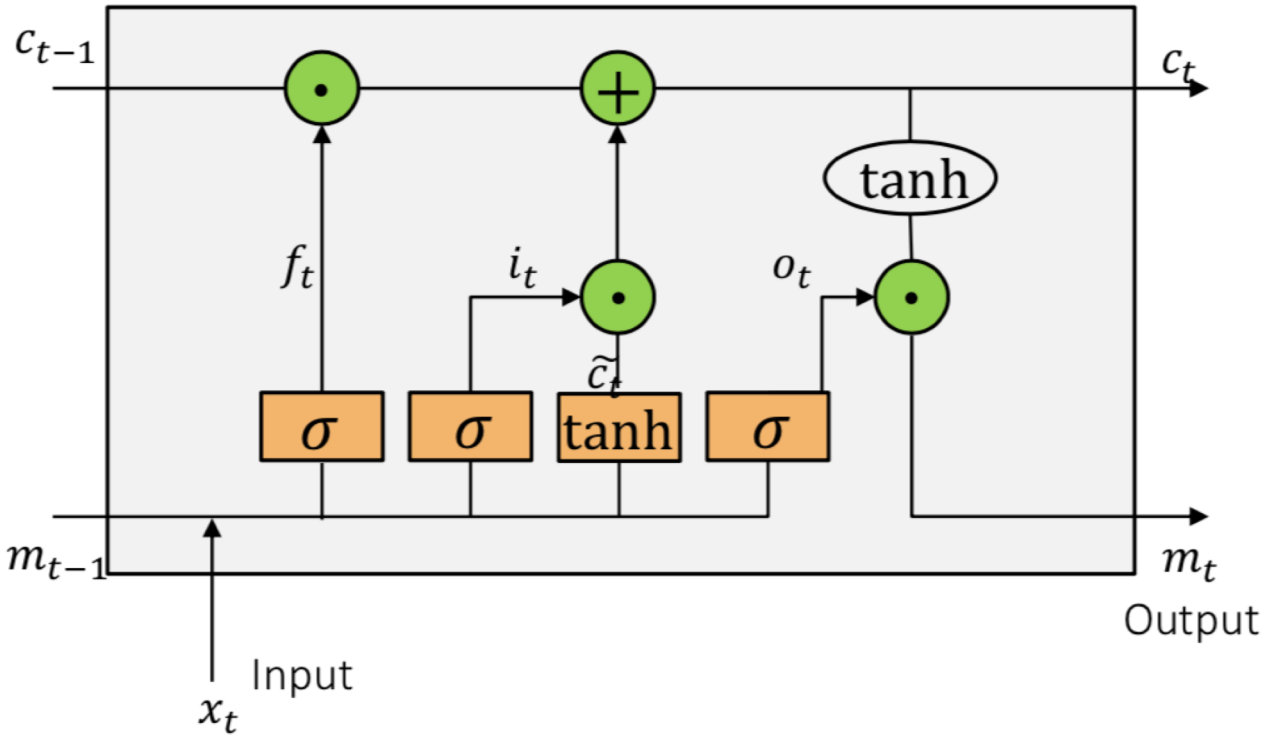


- A memory cell using logistic and linear units with multiplicative interactions:
  - Information writes into the cell when its input gate is on.
  - Information is thrown away from the cell when its forget gate is off.
  - Information can be read from the cell by turning on its output gate.

LSTM nonlinearities

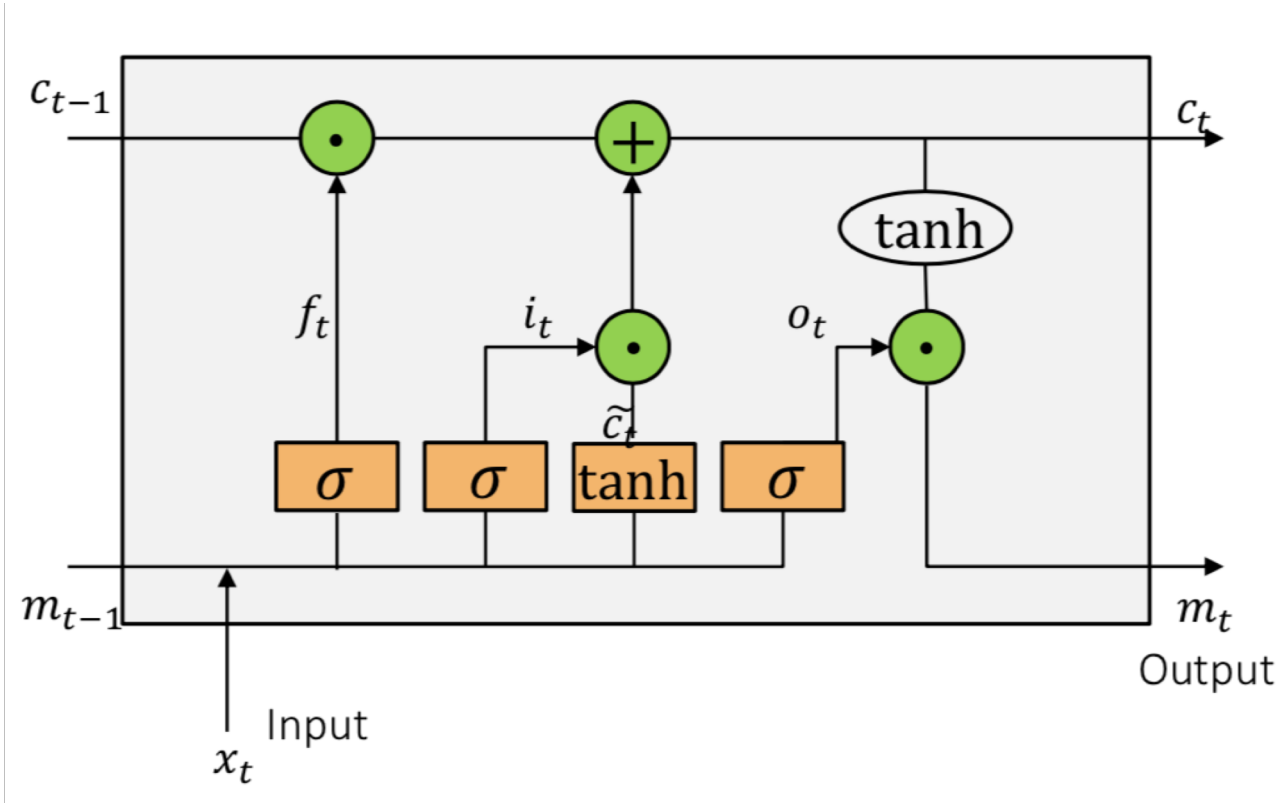
- $\sigma \in (0, 1)$: control gate – something like a switch
  - Sigmoid function

- tanh $\in (-1, 1)$: recurrent nonlinearity



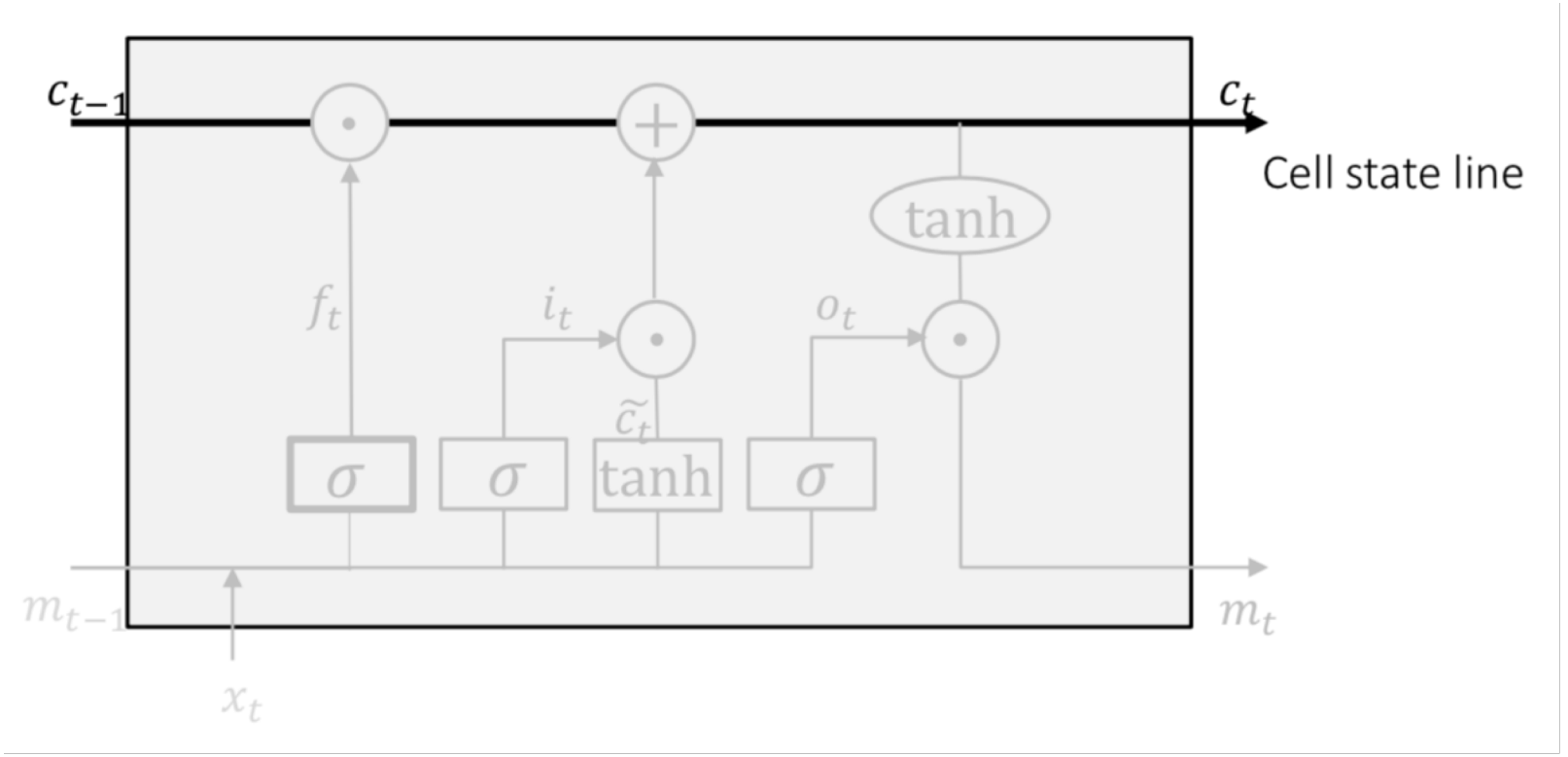
Gates

- Gates are a way to optionally let information through.
  - A sigmoid layer outputs number between 0 and 1, deciding how much of each component should be let through.
  - A pointwise multiplication operation applies the decision.



Cell State

- LSTM cell state line
  - Passing two operators:
    - Element by element multiplication
    - Addition

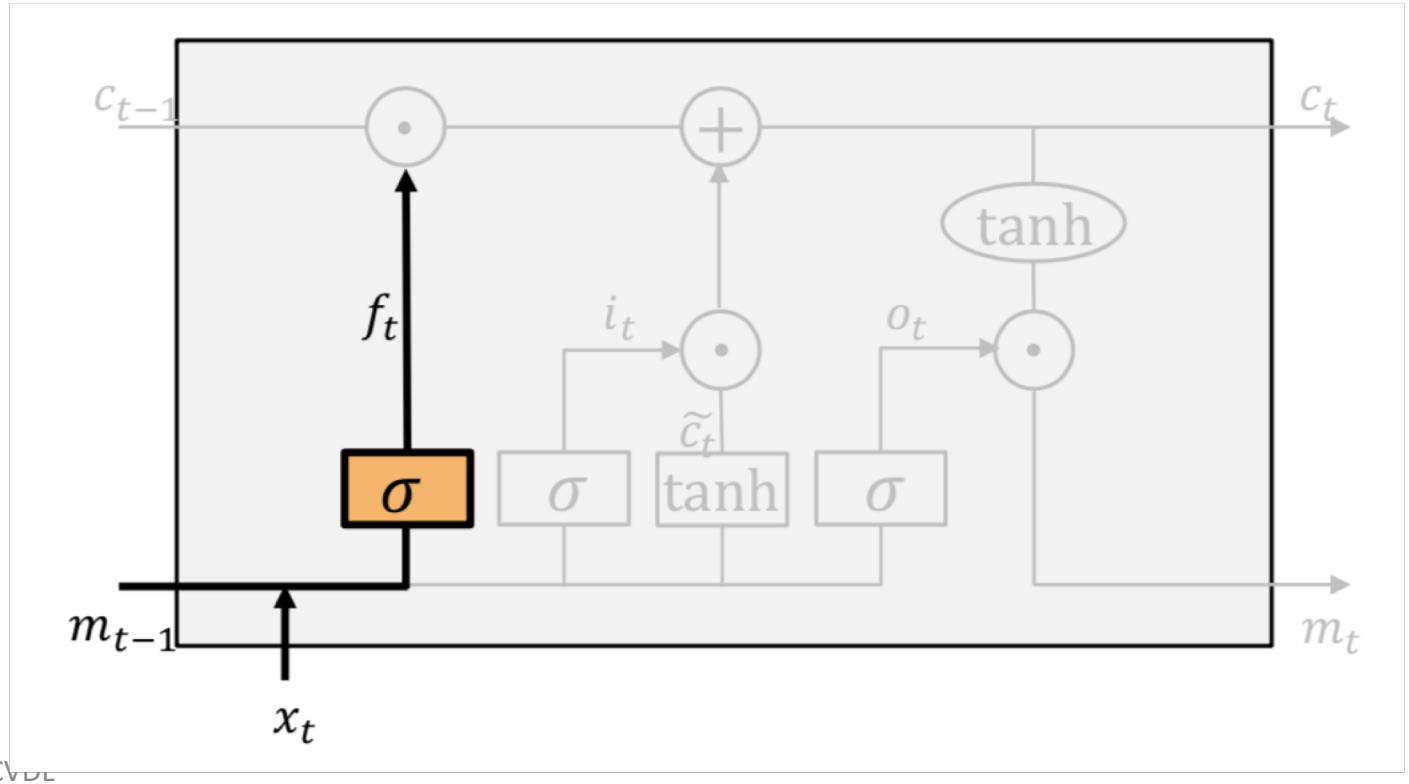


Forget gate

- A sigmoid layer, forget gate, decides which values of the memory cell to reset.

$$f = \sigma\left(x_t U^{(f)} + m_{t-1} W^{(f)}\right)$$

- m_{t-1} denotes previous output, $W^{(f)}$ and $U^{(f)}$ are parameters associated with forget gate

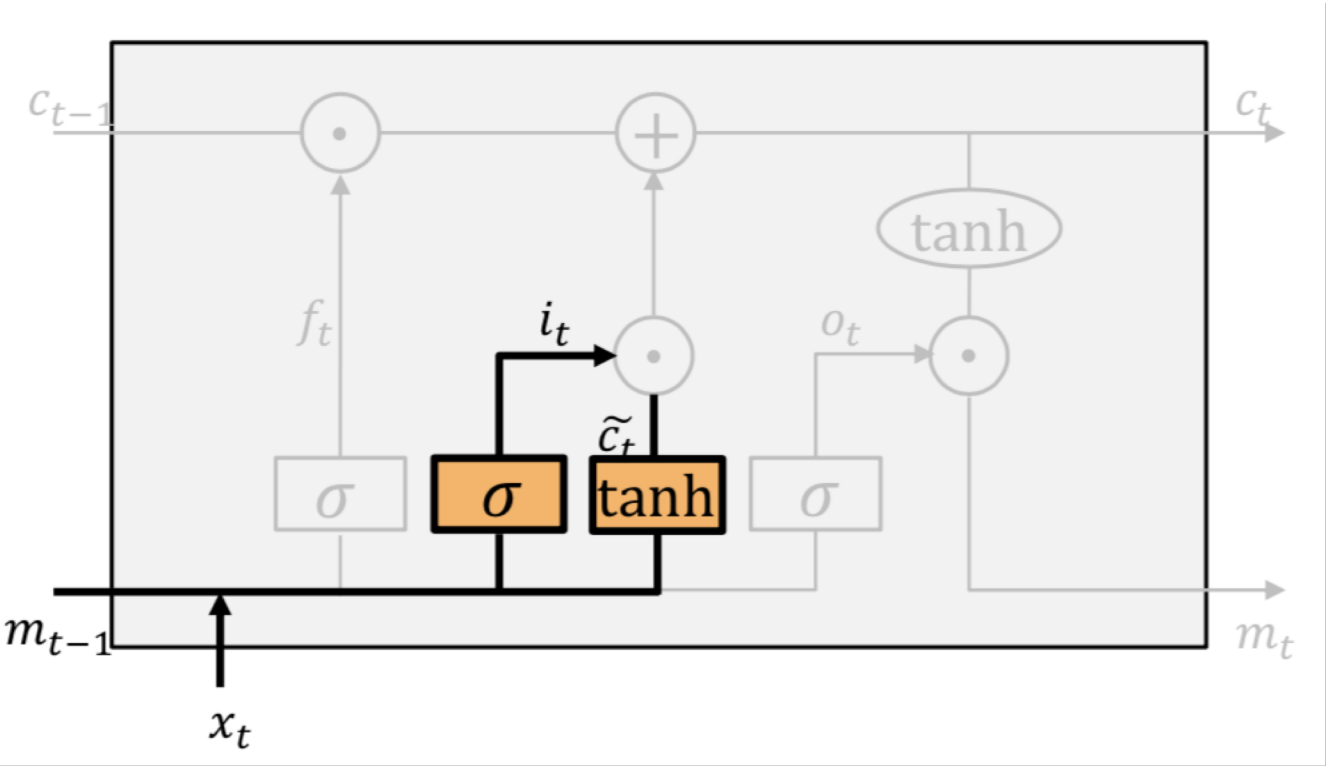


Input gate

- Decide what new information is relevant from the new input and should be added to the new memory
  - Modulate the input i_t
  - Generate candidate memories $\tilde{c}_t$

$$i = \sigma\left(x_t U^{(i)} + m_{t-1} W^{(i)}\right)$$

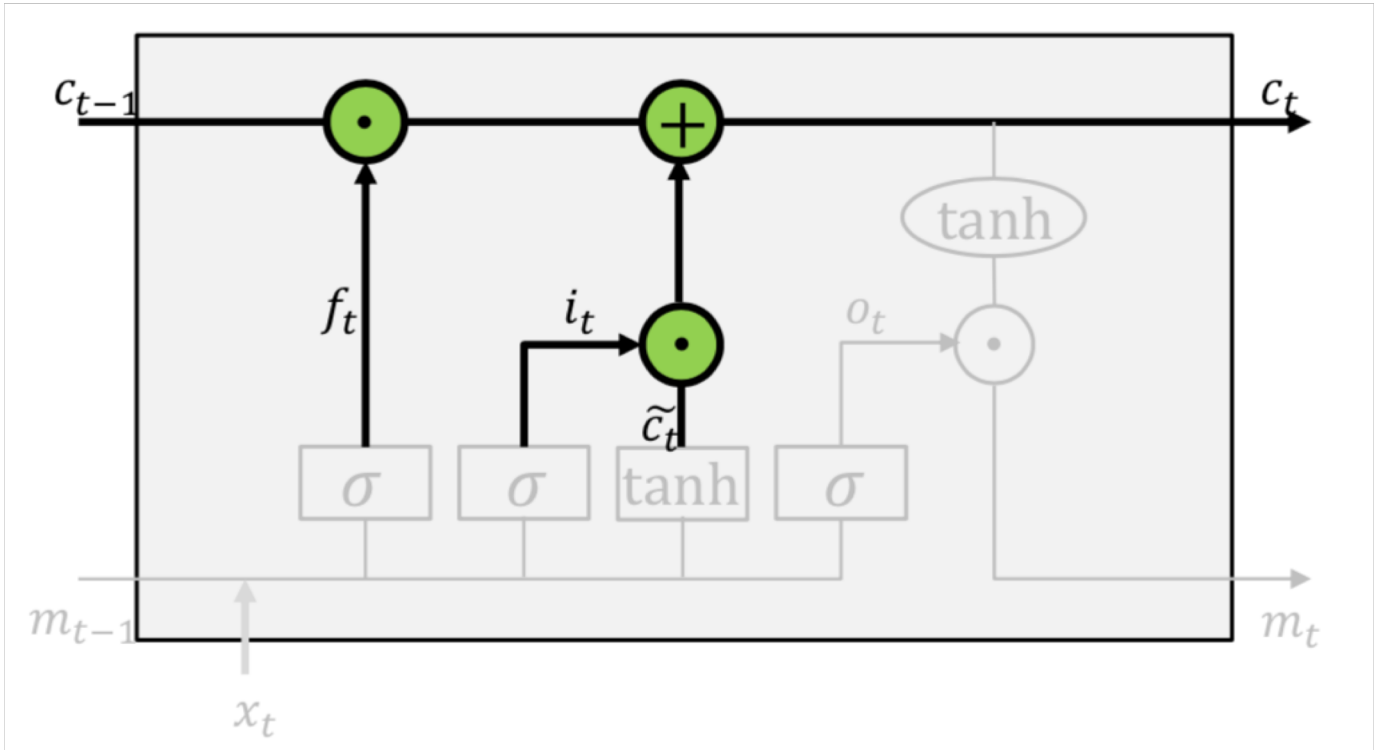
$$\tilde{c}_t = \tanh(x_t U^{(g)} + m_{t-1} W^{(g)})$$



Cell state update

- Compute and update the current cell state c_t
  - Depends on the previous cell state
  - What we decide to forget
  - What inputs we allow
  - The candidate memories

$$c_t = c_{t-1} \odot f + \tilde{c}_t \odot i$$

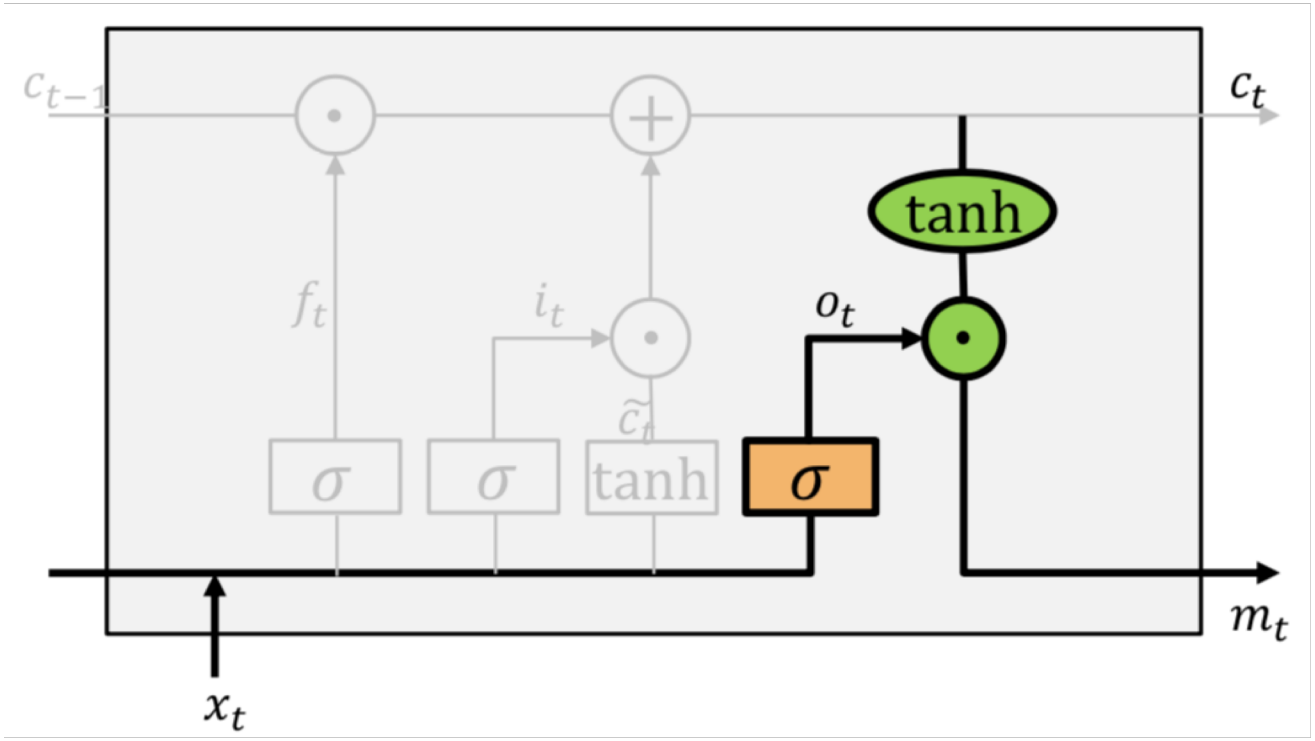


Output gate

- A sigmoid layer, output gate, decides which values of the memory cell to output
  - Modulate the output:
    - is the new cell state relevant? Sigmoid -> 1
    - If not Sigmoid -> 0

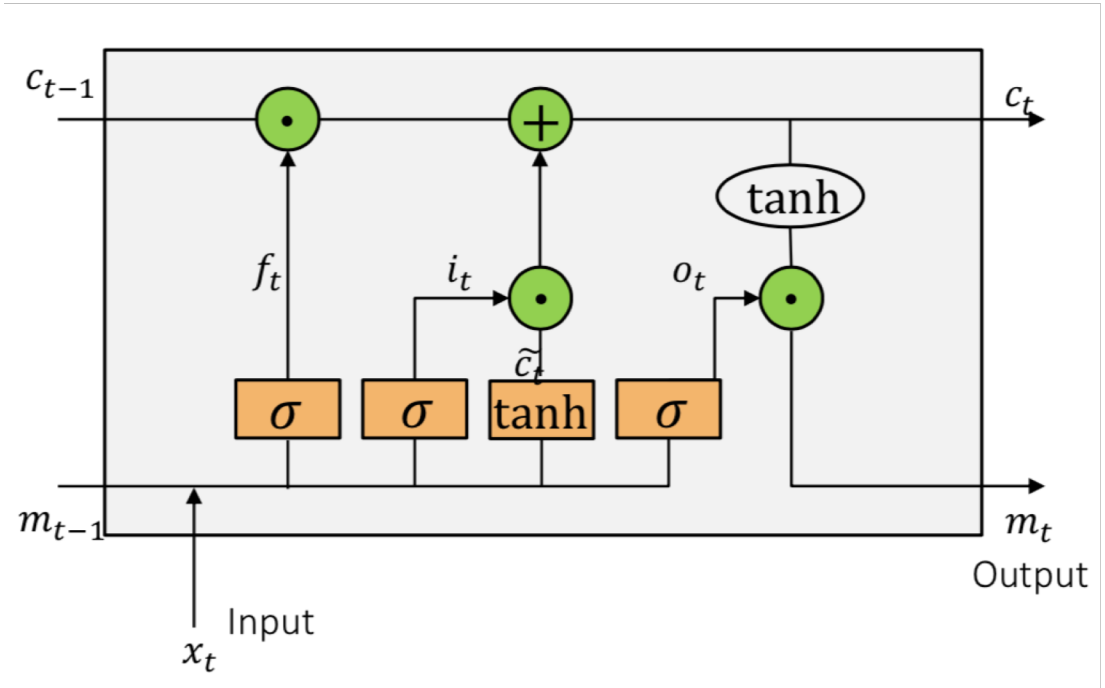
$$o = \sigma\left(x_t U^{(o)} + m_{t-1} W^{(o)}\right)$$

$$m_t = \tanh(c_t) \odot o$$



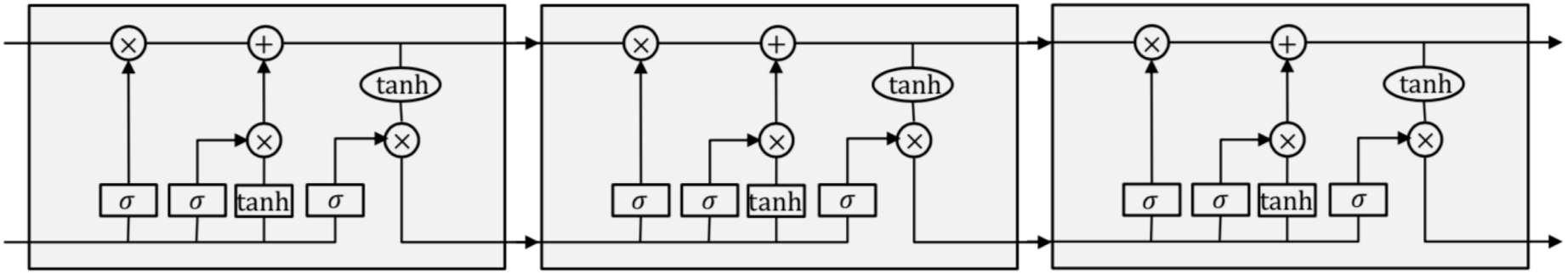
LSTM

$$
\begin{aligned}
i &= \sigma\left(x_t U^{(i)} + m_{t-1} W^{(i)}\right) \\
f &= \sigma\left(x_t U^{(f)} + m_{t-1} W^{(f)}\right) \\
o &= \sigma\left(x_t U^{(o)} + m_{t-1} W^{(o)}\right) \\
\tilde{c}_t &= \tanh\left(x_t U^{(g)} + m_{t-1} W^{(g)}\right) \\
c_t &= c_{t-1} \odot f + \tilde{c}_t \odot i \\
m_t &= \tanh(c_t) \odot o
\end{aligned}
$$



LSTM

- Unrolling: just the same like for RNNs
  - more complicated
  - Because of their gates LSTMs capture long and short term dependencies



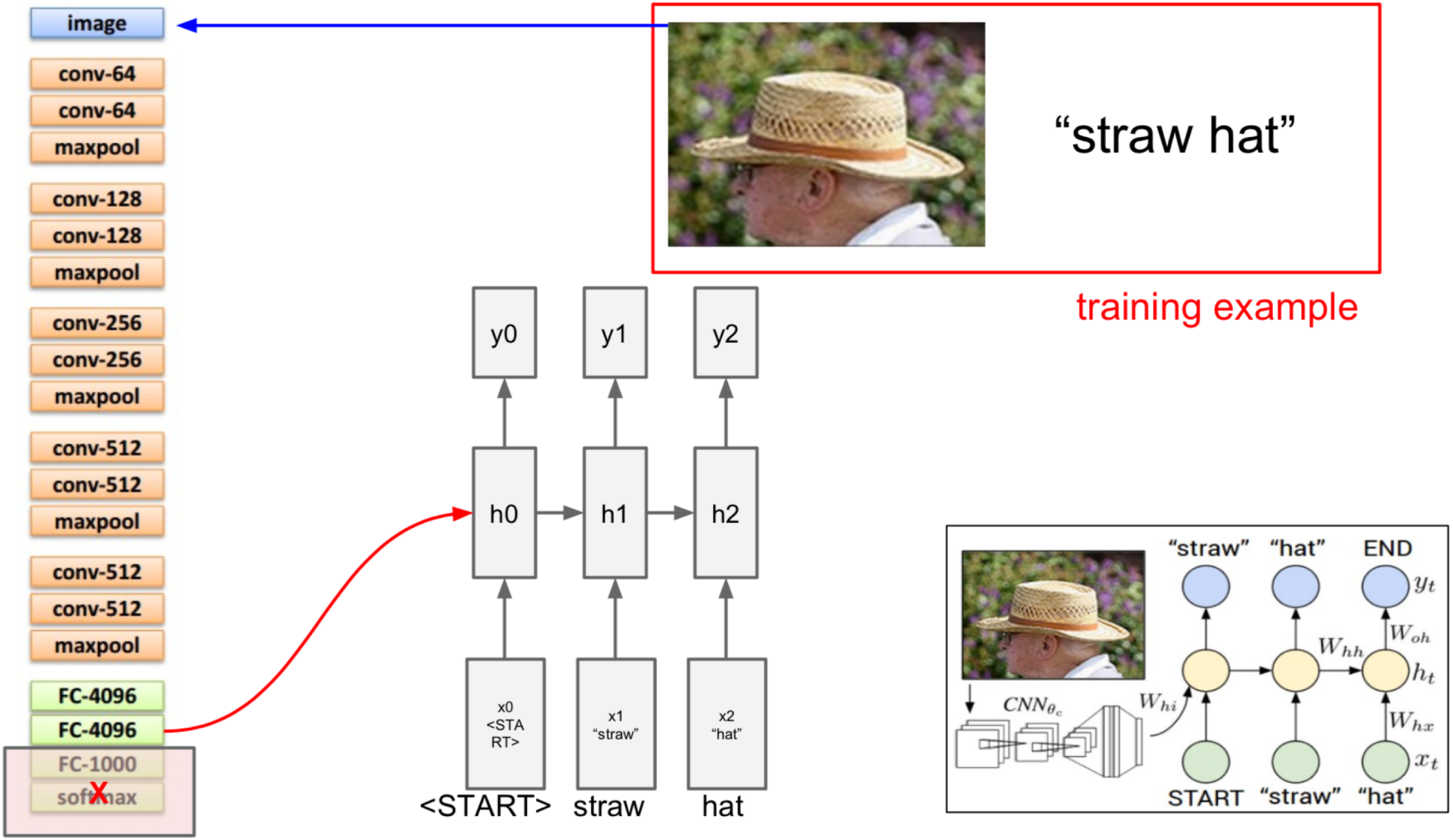
- Many variants, e.g.
  - Gated Recurrent Units (GRU)
  - Bi-directional recurrent networks
  - Gates have access also to the previous cell states $c_{(t-1)}$ (not only memories)

Example: image captioning

- Image sentence dataset, e.g.
  - Microsoft COCO, [Tsung-Yi Lin et al. 2014], mscoco.org
  - ~120K images
  - ~5 sentences each

Training

- Using pre-trained model on ImageNet (transfer learning)



Successful cases

A cat sitting on a suitcase on the floor

A cat is sitting on a tree branch

A dog is running in the grass with a frisbee

A white teddy bear sitting in the grass

Two people walking on the beach with surfboards

A tennis player in action on the court

Two giraffes standing in a grassy field

A man riding a dirt bike on a dirt track

Failure cases

A woman is holding a cat in her hand

A person holding a computer mouse on a desk

A woman standing on a beach holding a surfboard

A bird is perched on a tree branch

A man in a baseball uniform throwing a ball