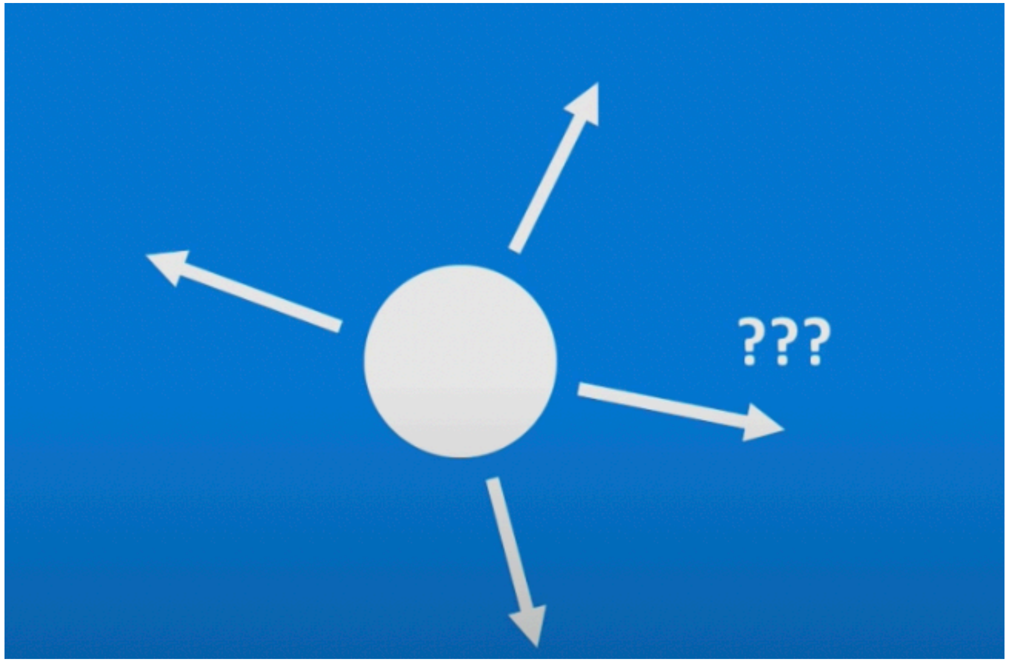
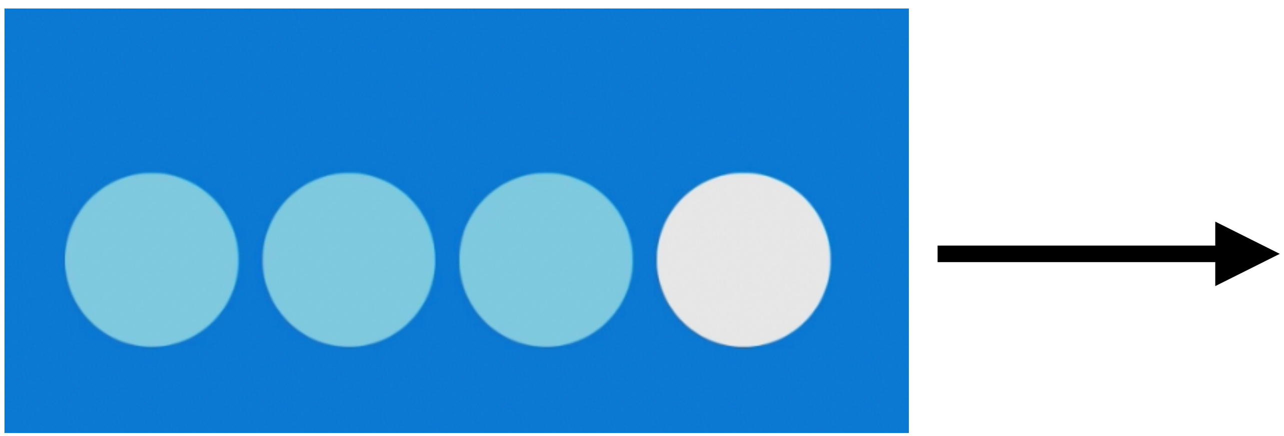
SEQUENCE MODELLING

*based on MIT Lecture by Ava Soleimany

GIVEN AN IMAGE OF A BALL, CAN YOU PREDICT WHERE IT WILL GO NEXT?

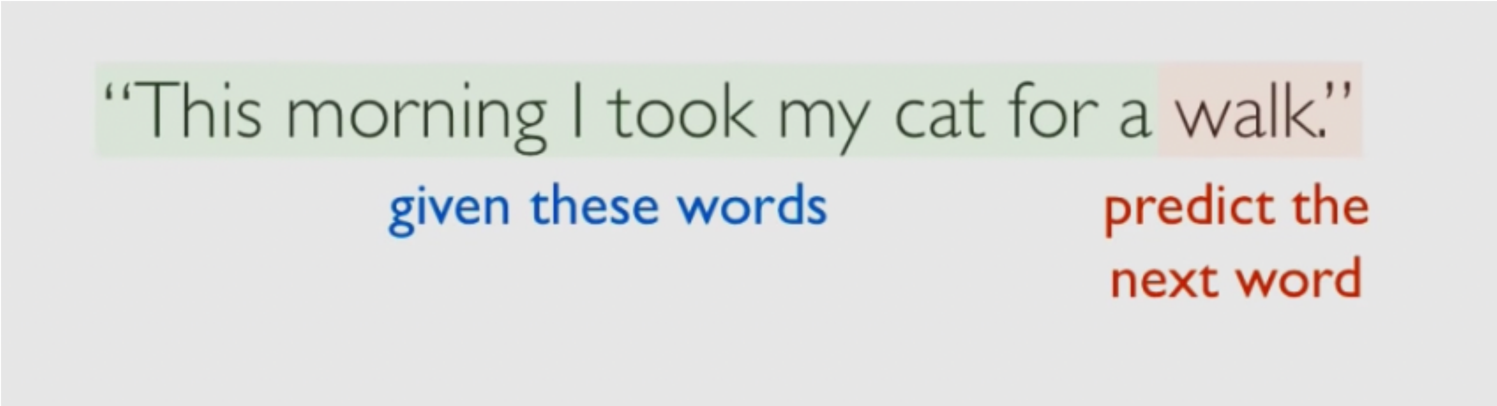


GIVEN AN IMAGE OF A BALL, CAN YOU PREDICT WHERE IT WILL GO NEXT?

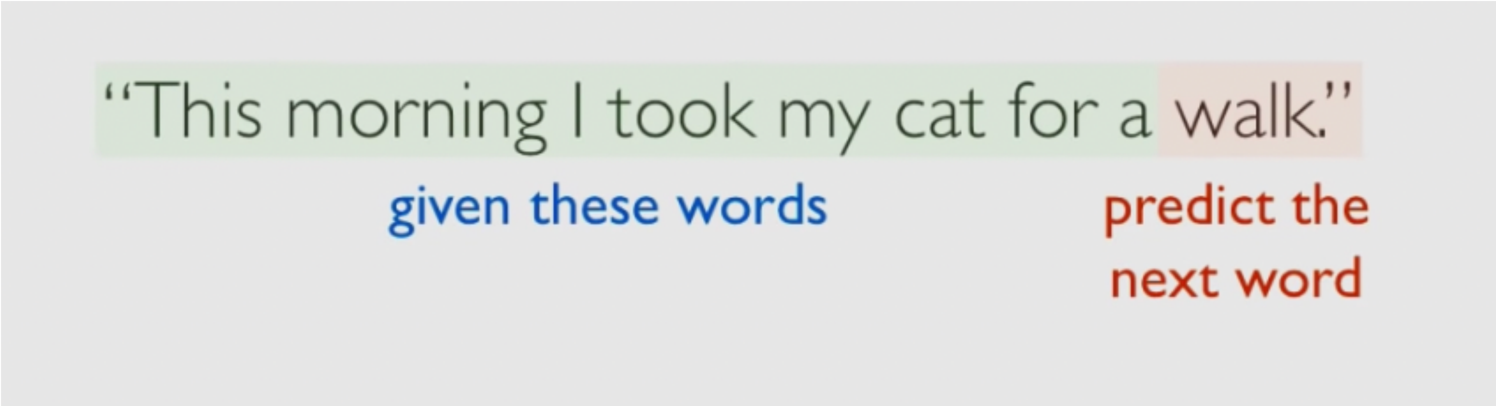


Previous positons help guessing the future

LET’S TAKE A SIMPLE EXAMPLE OF LANGUAGE MODELLING

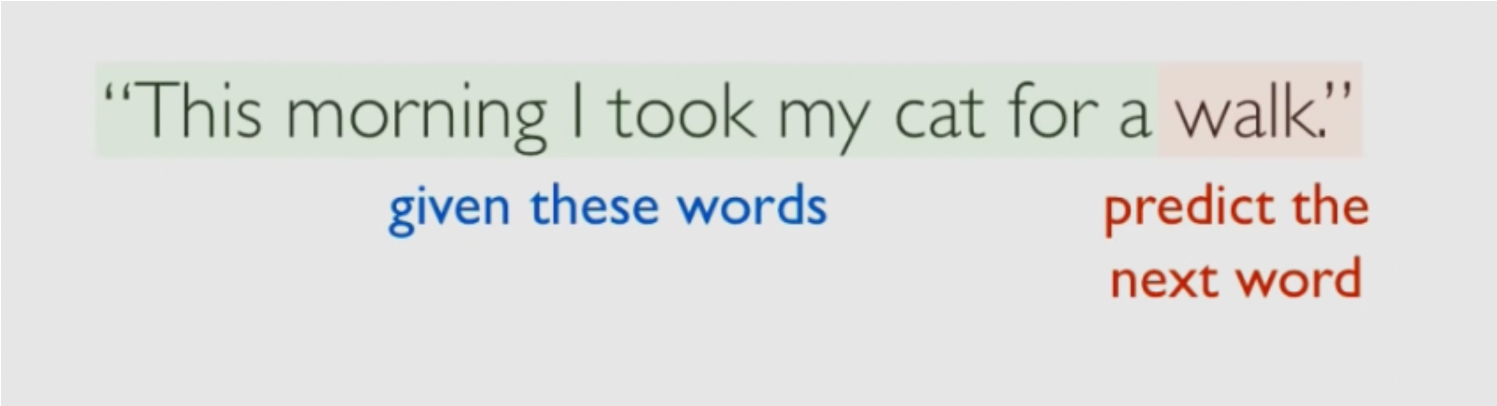


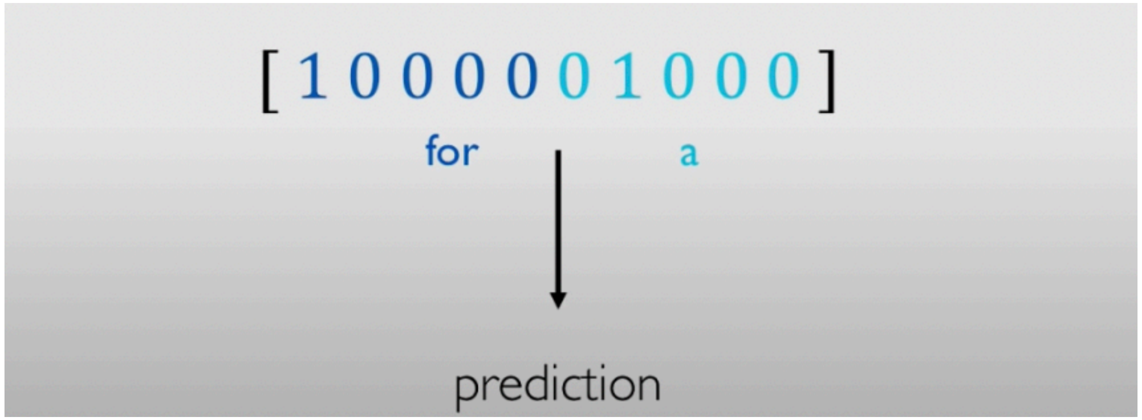
LET'S TAKE A SIMPLE EXAMPLE OF LANGUAGE MODELLING



Normal ANNs and CNNs cannot handle variable length inputs …

WE COULD SIMPLY USE A FIXED WINDOW…

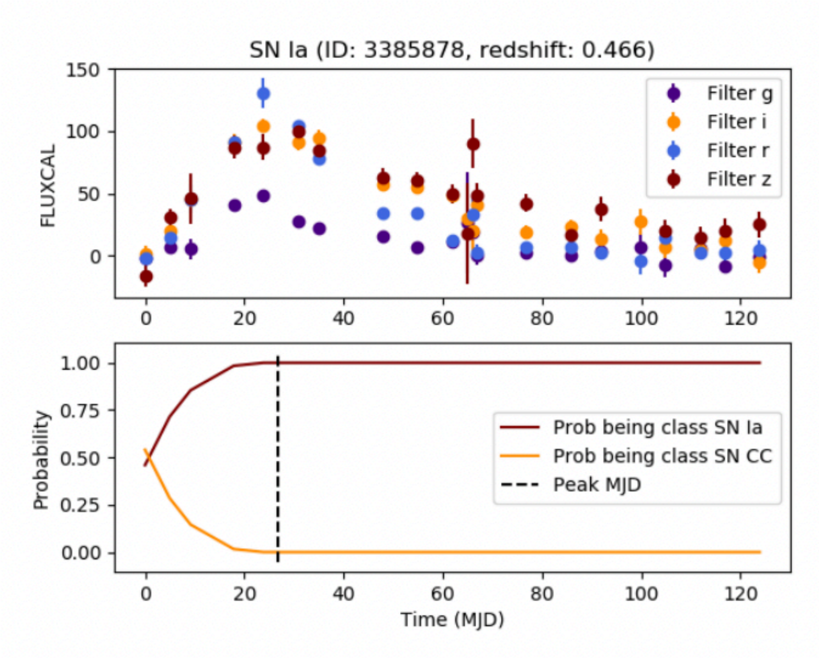




HOWEVER THIS CANNOT HANDLE LONG TERM MEMORY

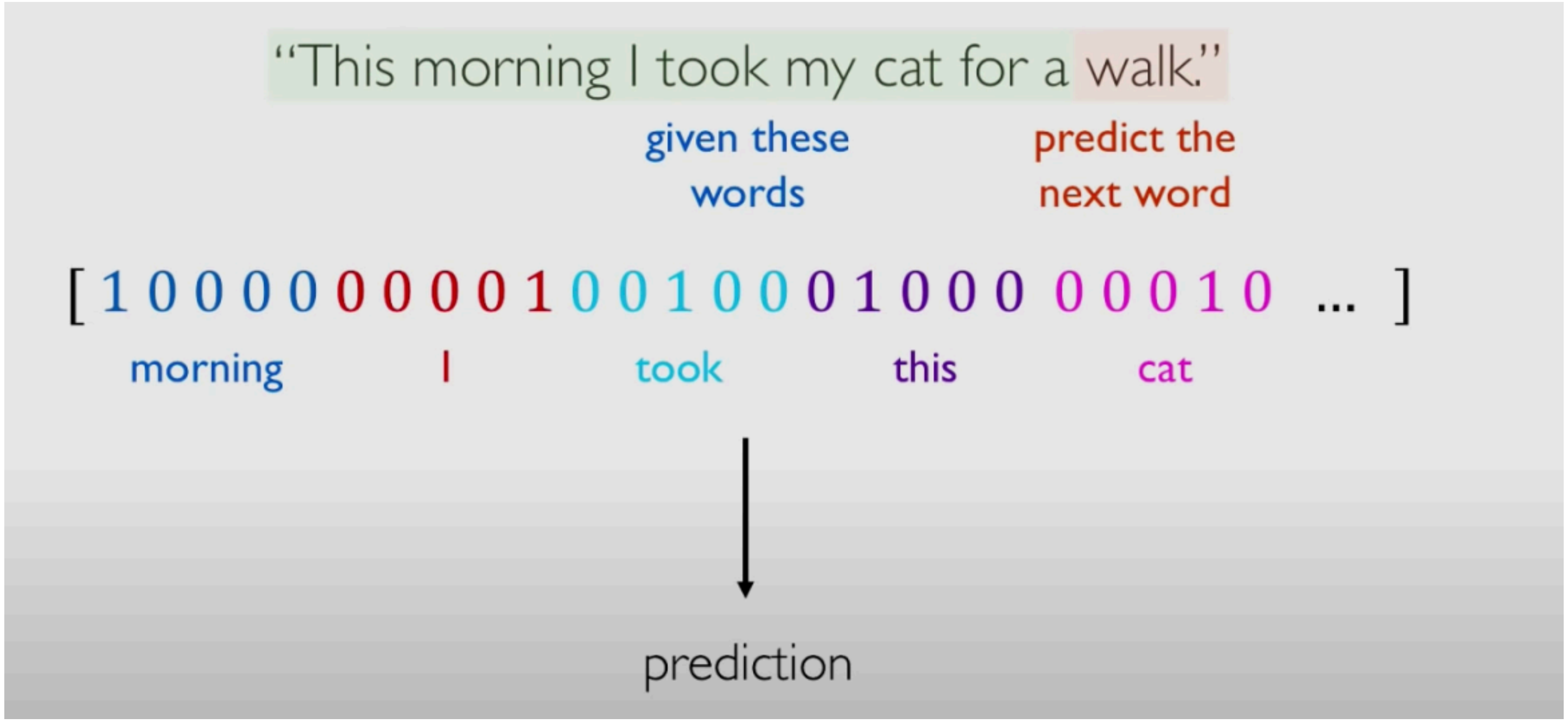
"**France** is where I grew up, but I now live in Boston. I speak fluent ____."

IN SOME CASES, INFORMATION FROM THE DISTANT PAST IS NEEDED FOR A CORRECT PREDICTION



Moller+19

ANOTHER ALTERNATIVE COULD BE TO FEED A REALLY LARGE WINDOW



BUT WE STILL HAVE THE PROBLEM THAT THERE IS NO PARAMETER SHARING (SAME AS PIXELS)

RECURRENT NEURAL NETWORKS (RNNs)

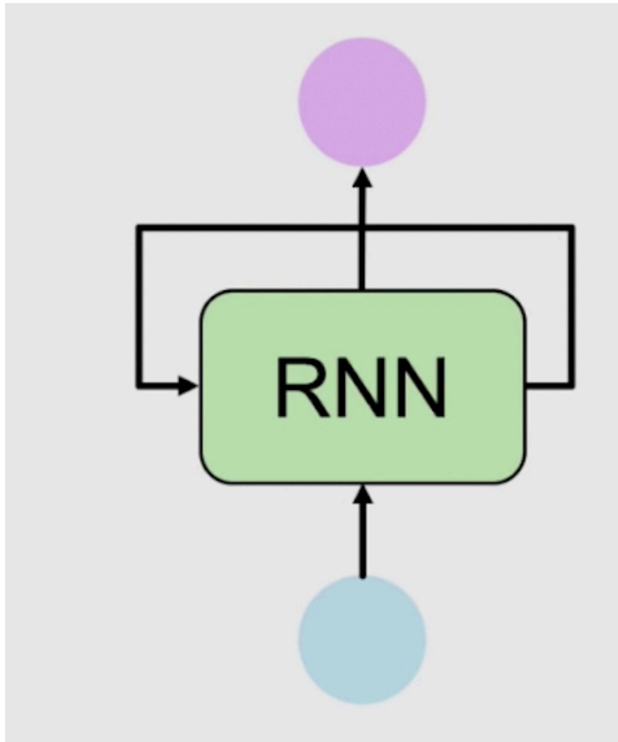
OUR WISH LIST:

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

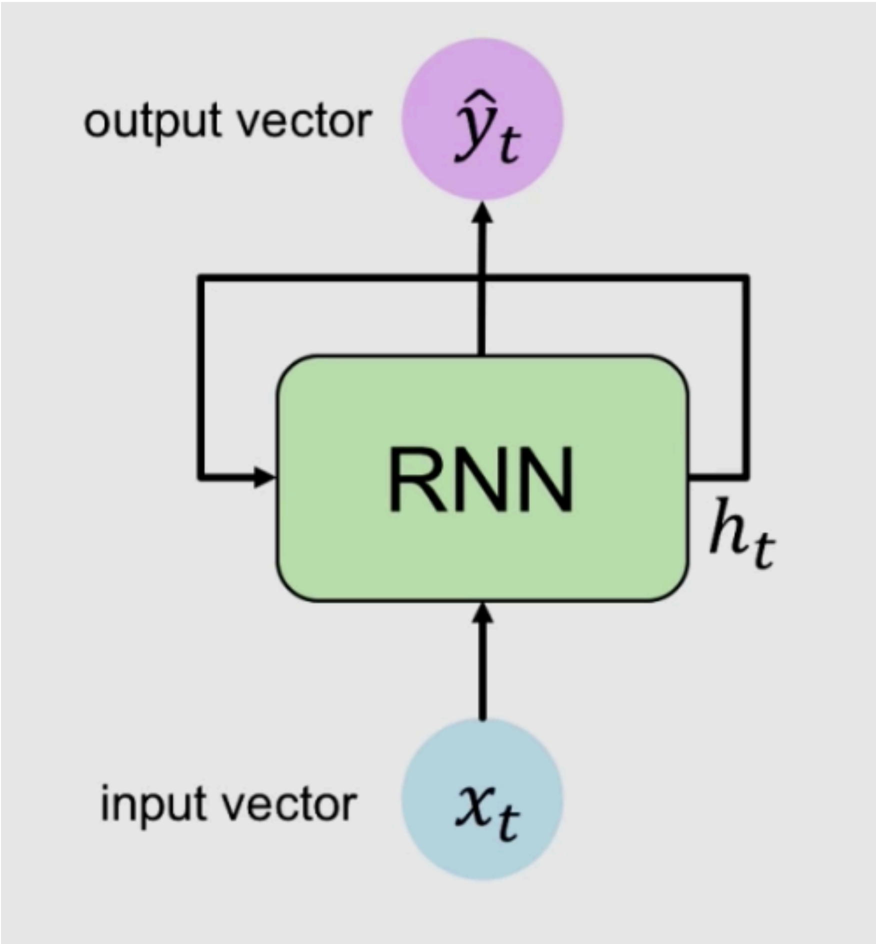
OUR WISH LIST:

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

Recurrent Neural Networks offer a first solution to this problem

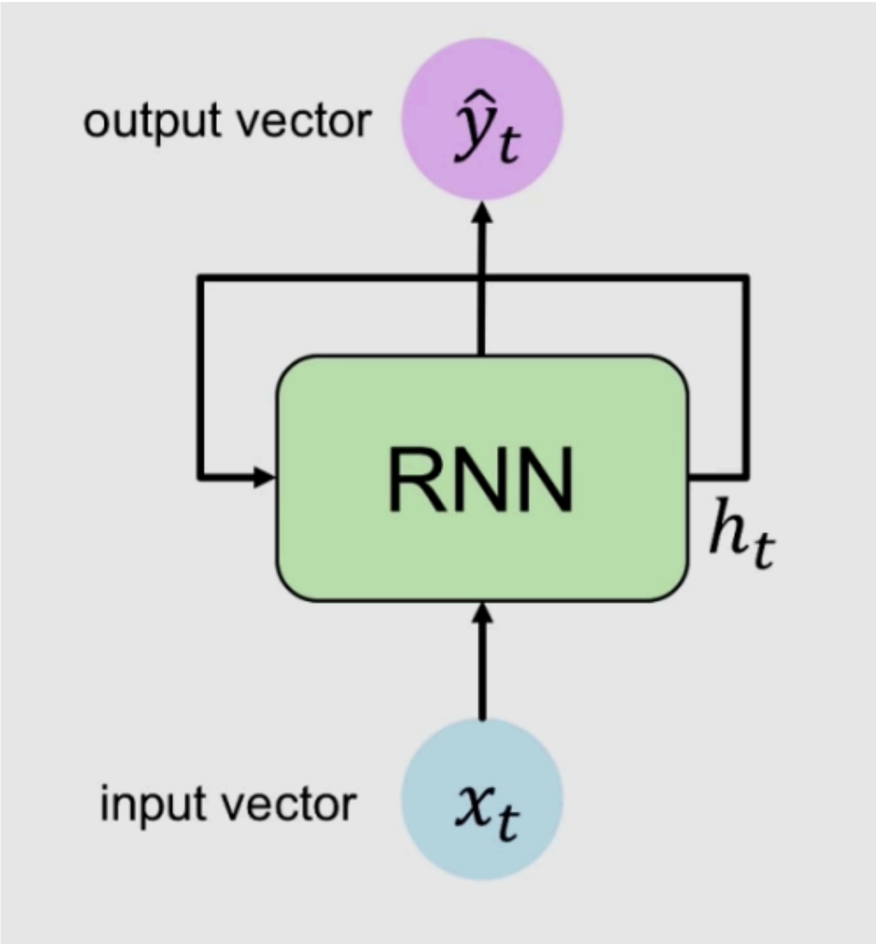


The RNN Block



$$\underset{\text{cell state}}{\boxed{h_t}} = \underset{\text{function parameterized by W}}{\boxed{f_W}}\left(\underset{\text{old state}}{\boxed{h_{t-1}}}, \underset{\text{input vector at time step } t}{\boxed{x_t}}\right)$$

The RNN Block



Output Vector

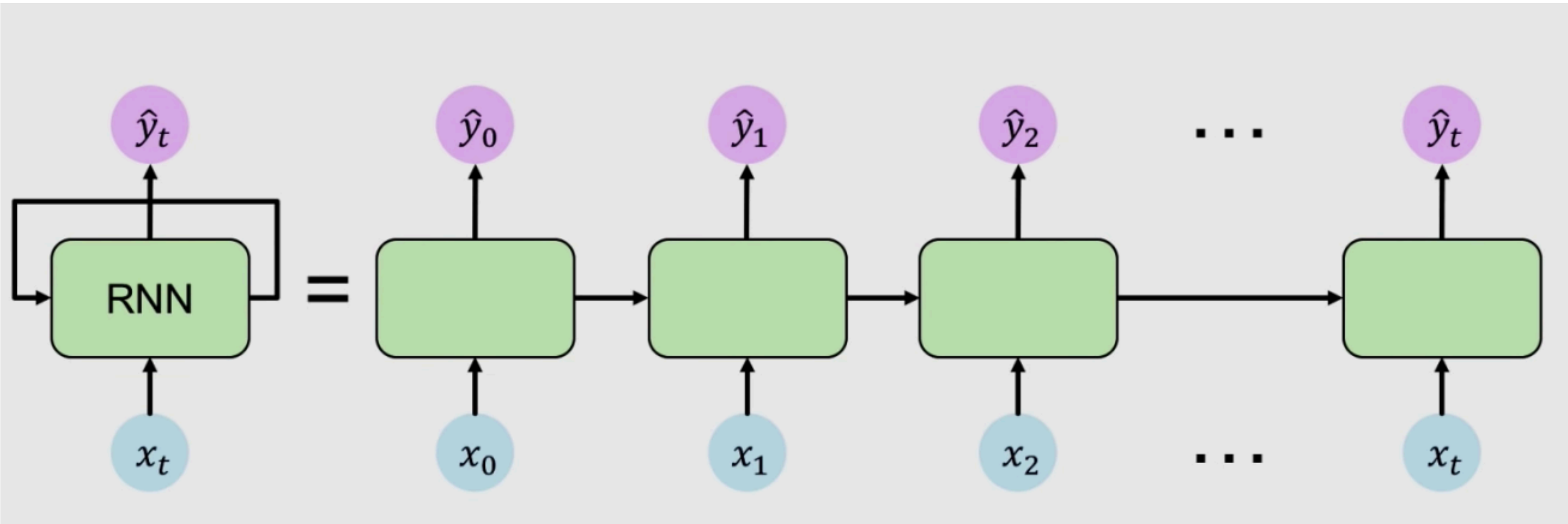
$$\hat{y}_t = \boldsymbol{W}_{\boldsymbol{hy}}^{\boldsymbol{T}} h_t$$

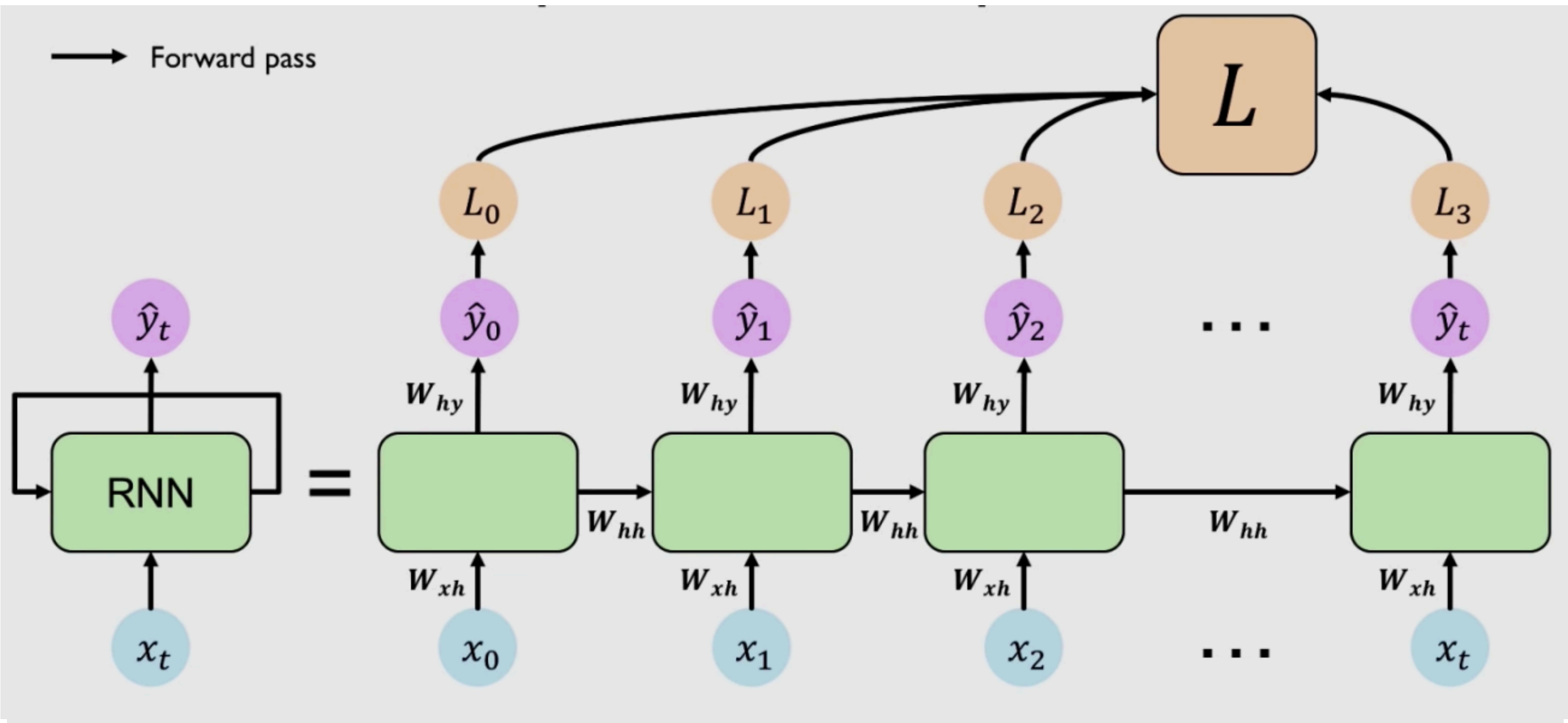
Update Hidden State

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

Input Vector

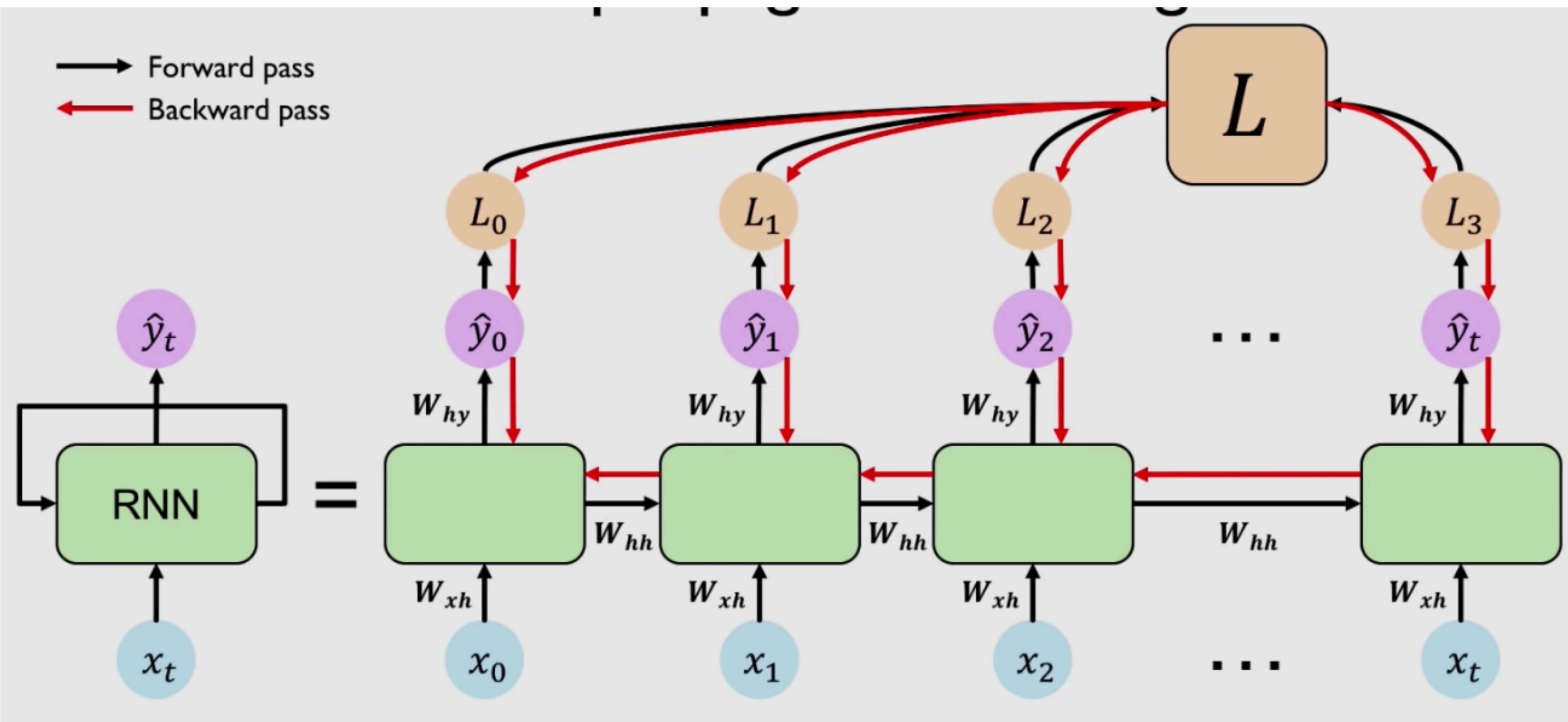
$$x_t$$

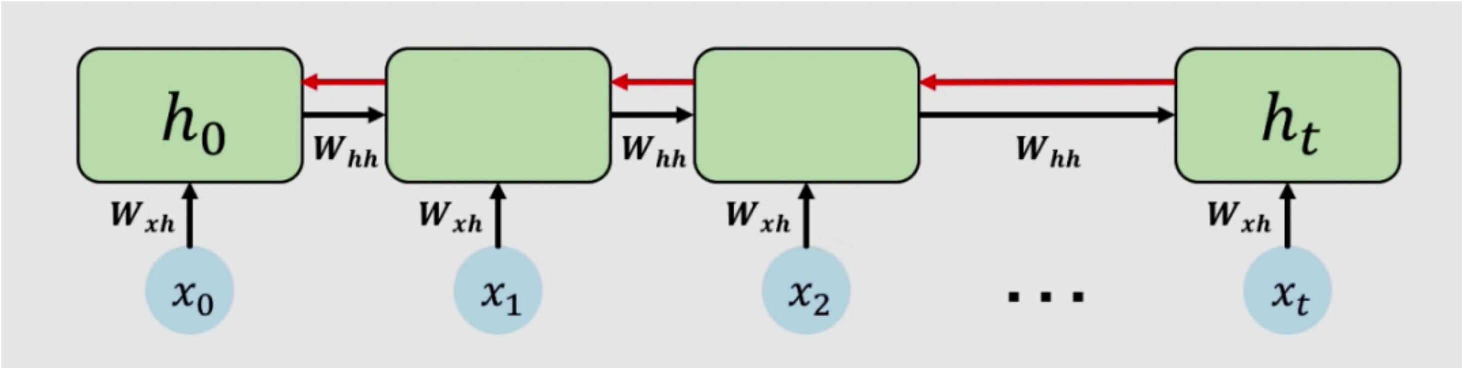

$\hat{y}_t$
RNN
x_t
=
$\hat{y}_0$
$\hat{y}_1$
$\hat{y}_2$
...
$\hat{y}_t$
x_0
x_1
x_2
...
x_t



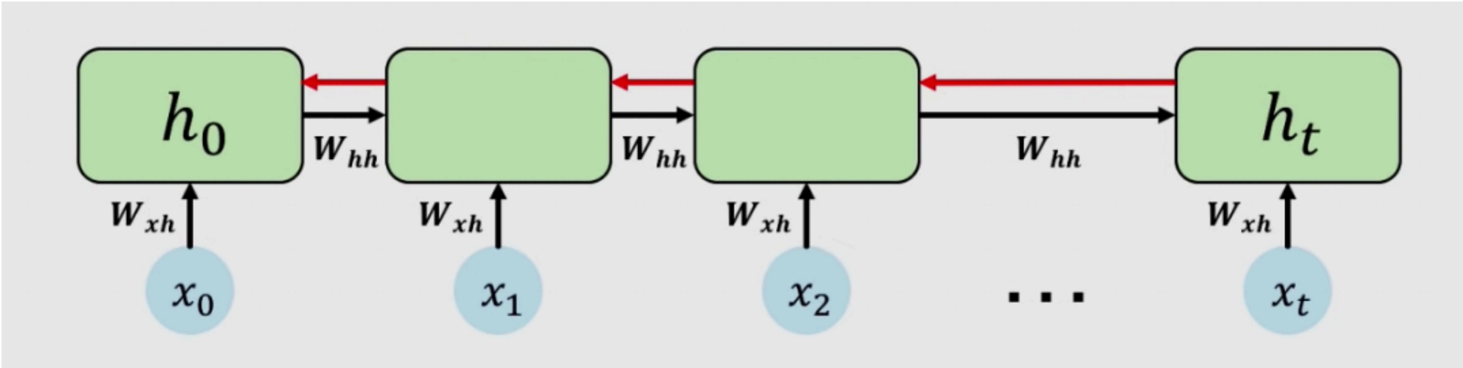
Forward pass
L
L_0
L_1
L_2
L_3
$\hat{y}_t$
$\hat{y}_0$
$\hat{y}_1$
$\hat{y}_2$
...
$\hat{y}_t$
W_{hy}
W_{hy}
W_{hy}
W_{hy}
RNN
=
W_{hh}
W_{hh}
W_{hh}
W_{xh}
W_{xh}
W_{xh}
W_{xh}
x_t
x_0
x_1
x_2
...
x_t

BACKPROPAGATION THROUGH TIME (BPTT)





BPTT implies a large amount of weight multiplications: if we want to take into account long term memory, networks become quickly very deep and so are subjected to vanishing and exploding gradients problems (see previous lecture)

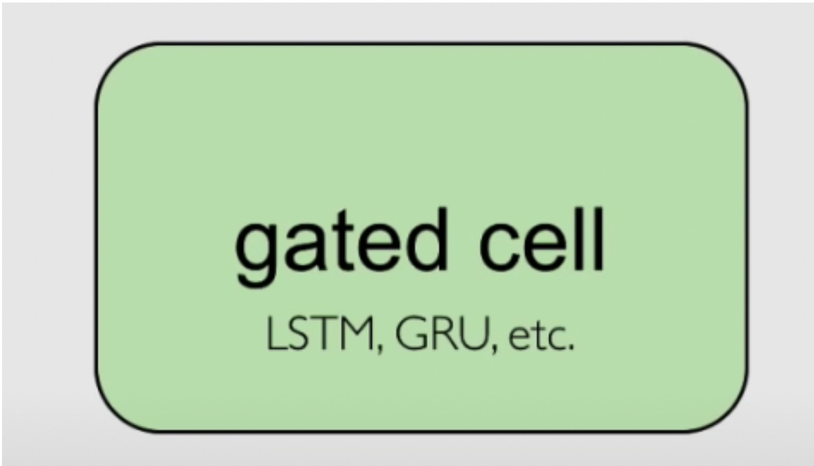


BPTT implies a large amount of weight multiplications: if we want to take into account long term memory, networks become quickly very deep and so are subjected to vanishing and exploding gradients problems (see previous lecture)

ONE CAN USE THE SAME TRICKS THAT WE DISCUSSED FOR DEEP CNNs (WEIGTH INITIALISATION, RELU ACTIVATION FUNCTION …)

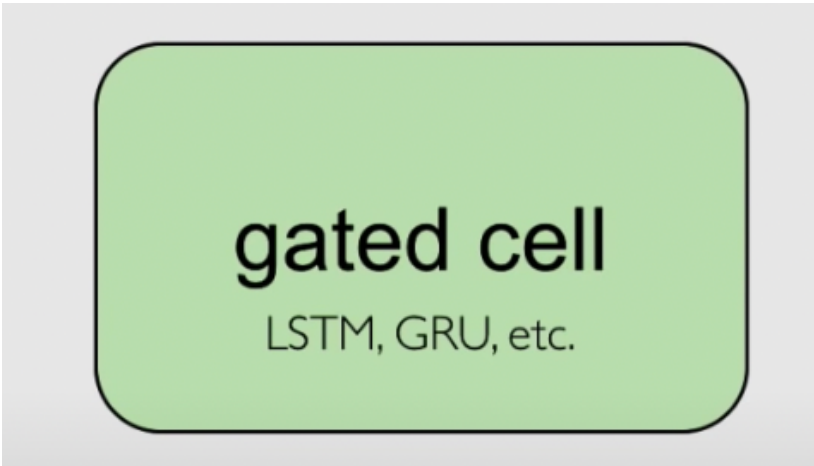
ANOTHER SPECIFIC SOLUTION: GATED CELLS

Use a more complex recurrent unit with gates to control what information is passed through…



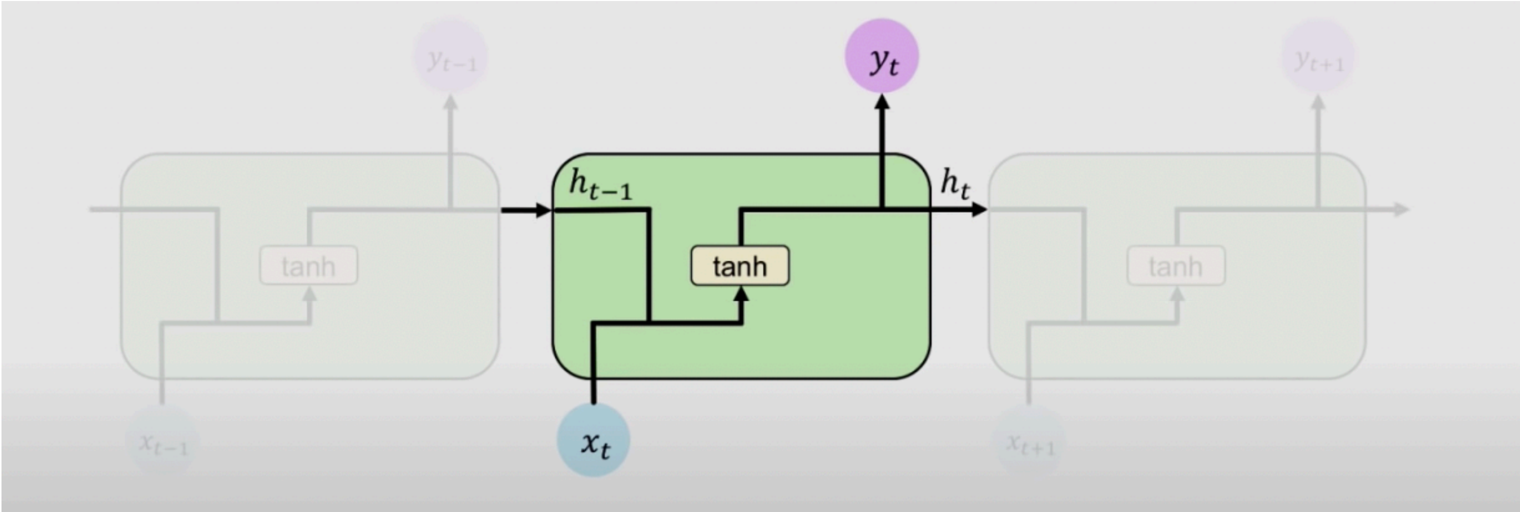
ANOTHER SPECIFIC SOLUTION: GATED CELLS

Use a more complex recurrent unit with gates to control what information is passed through…

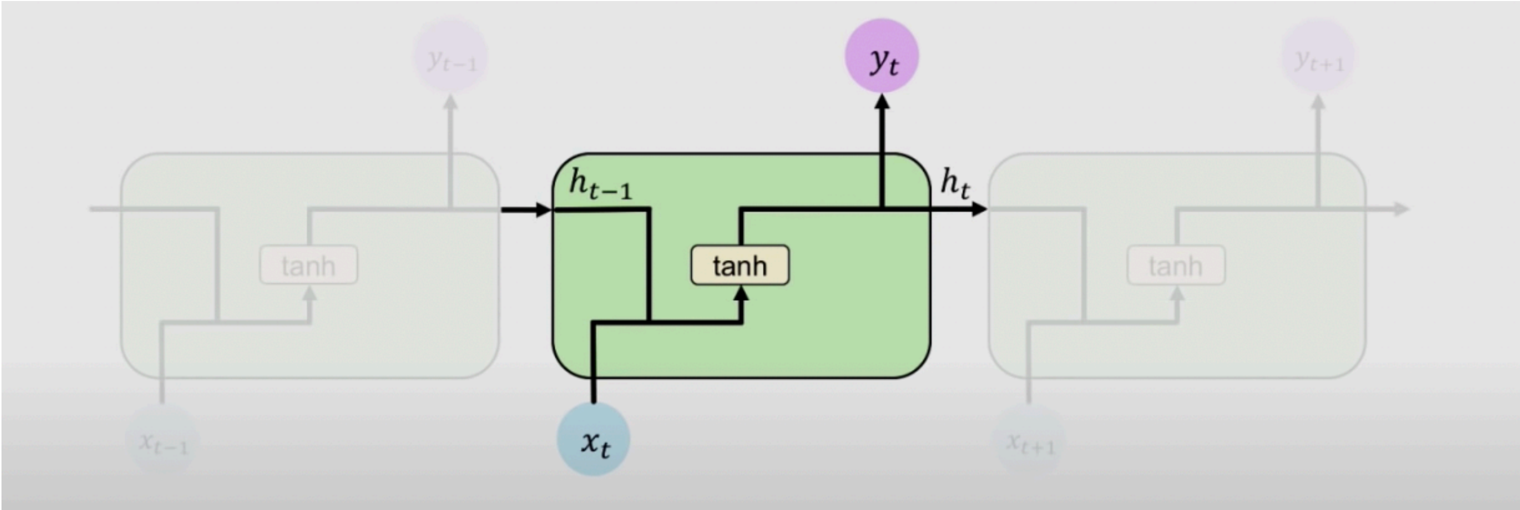


Long Short Term Memory (LSTMs) networks are an example of this

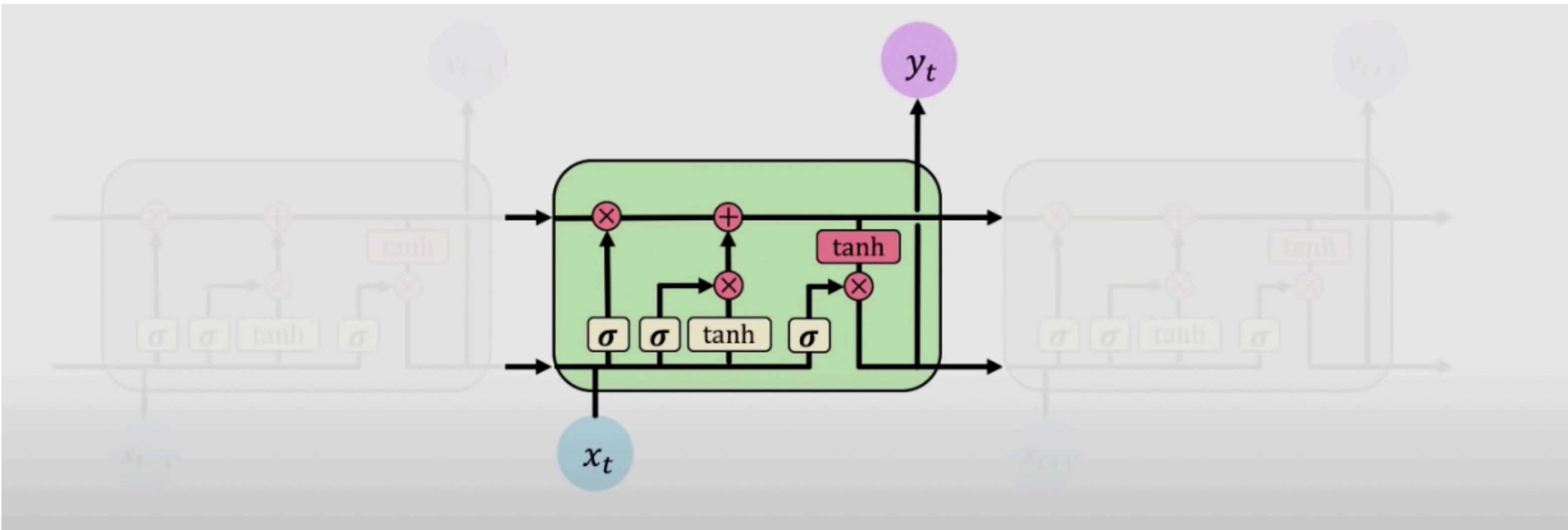
Standard RNN



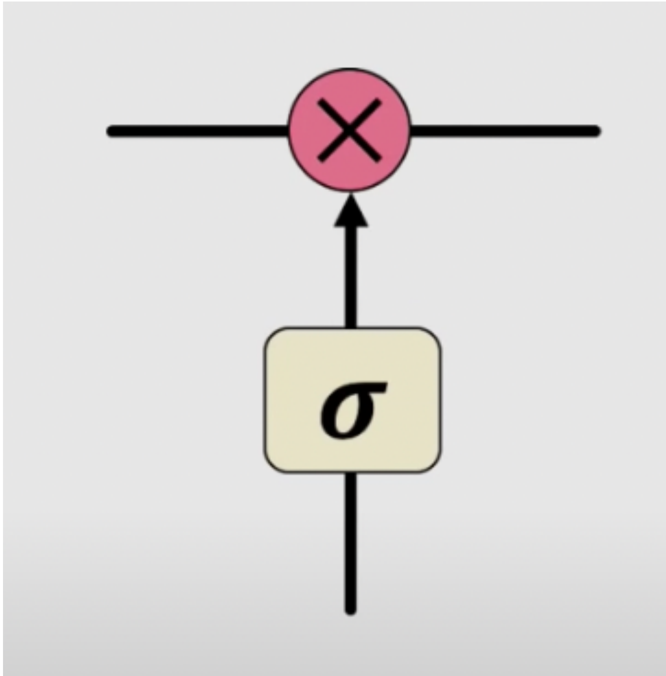
Standard RNN



LSTM unit

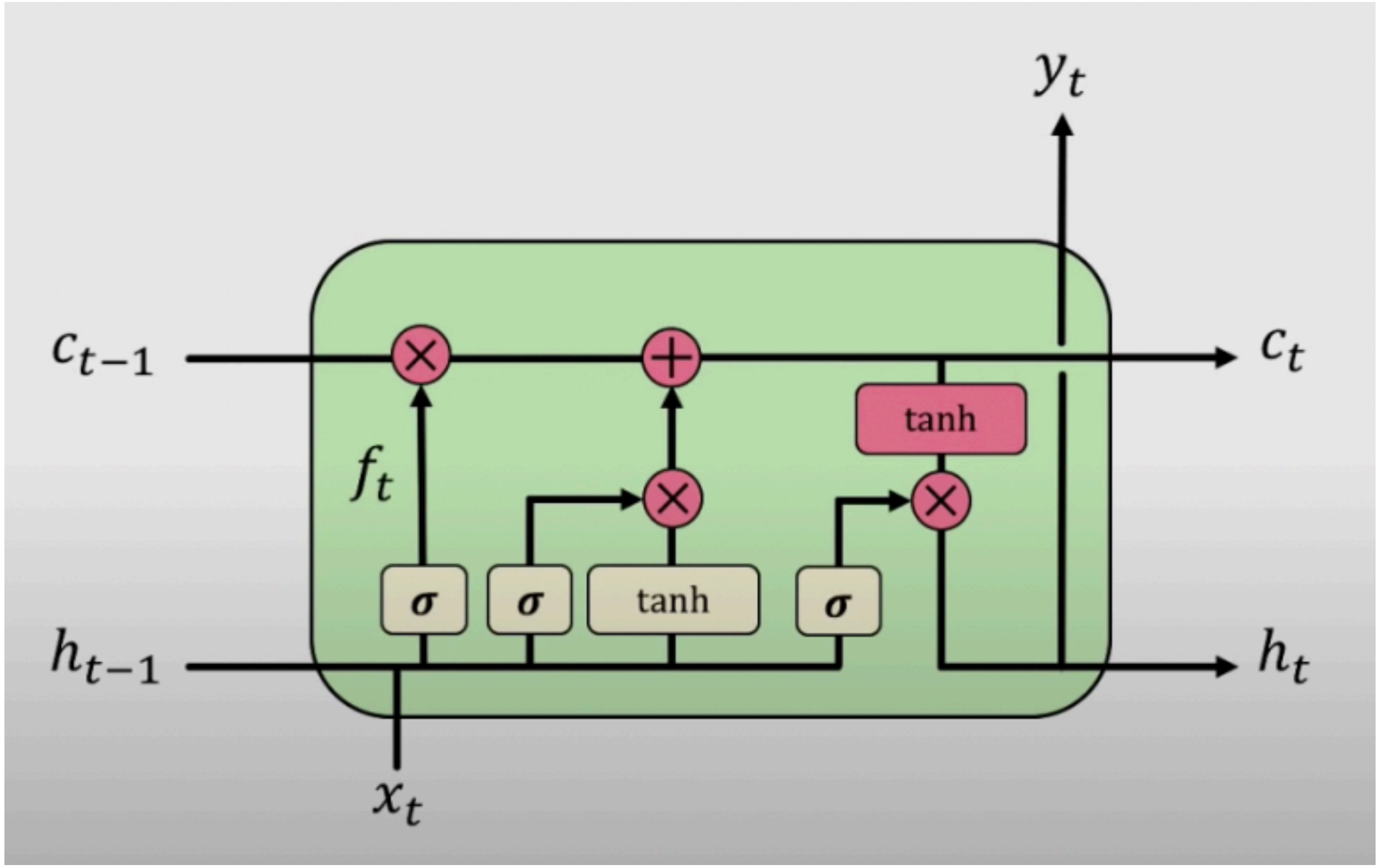


The key building block are the so-called gates



INFORMATION IS PASSED OR NOT WITH A COMBINATION OF A SIGMOID NN AND A POINTWISE MULTIPLICATION

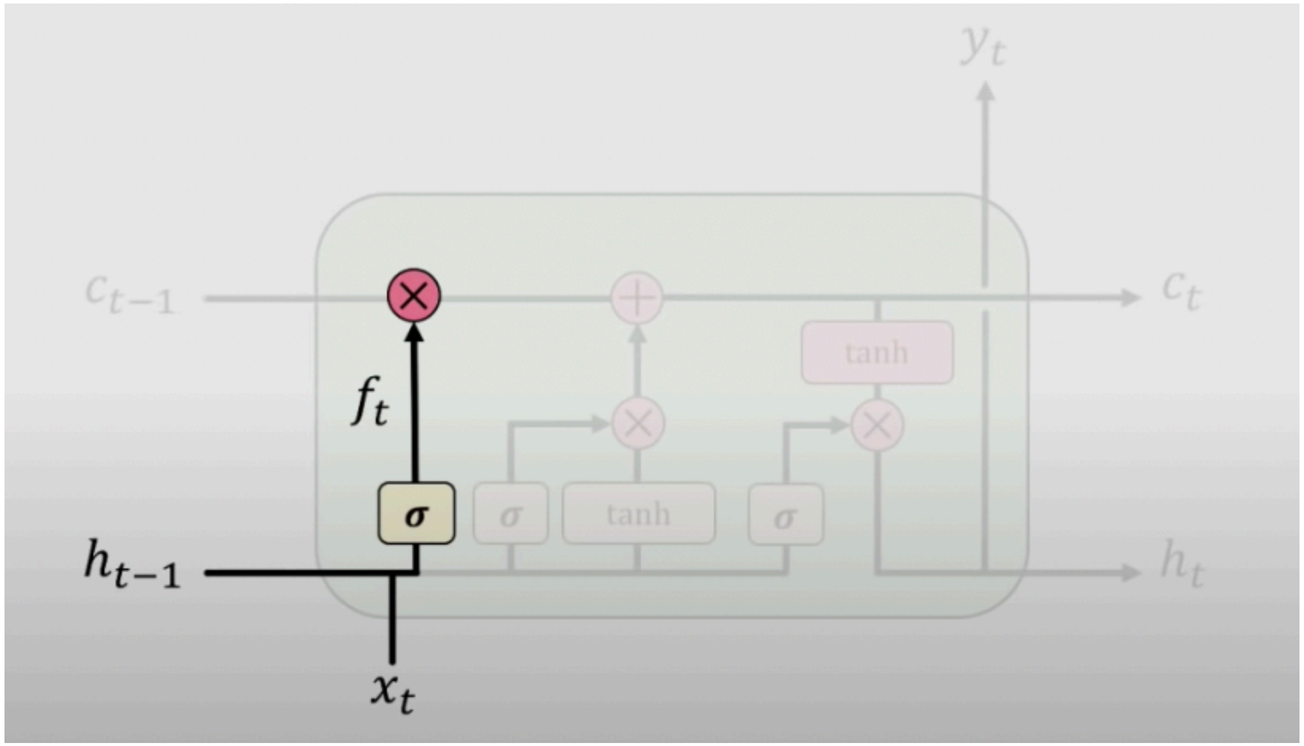
Long Short Term Memory (LSTMs)



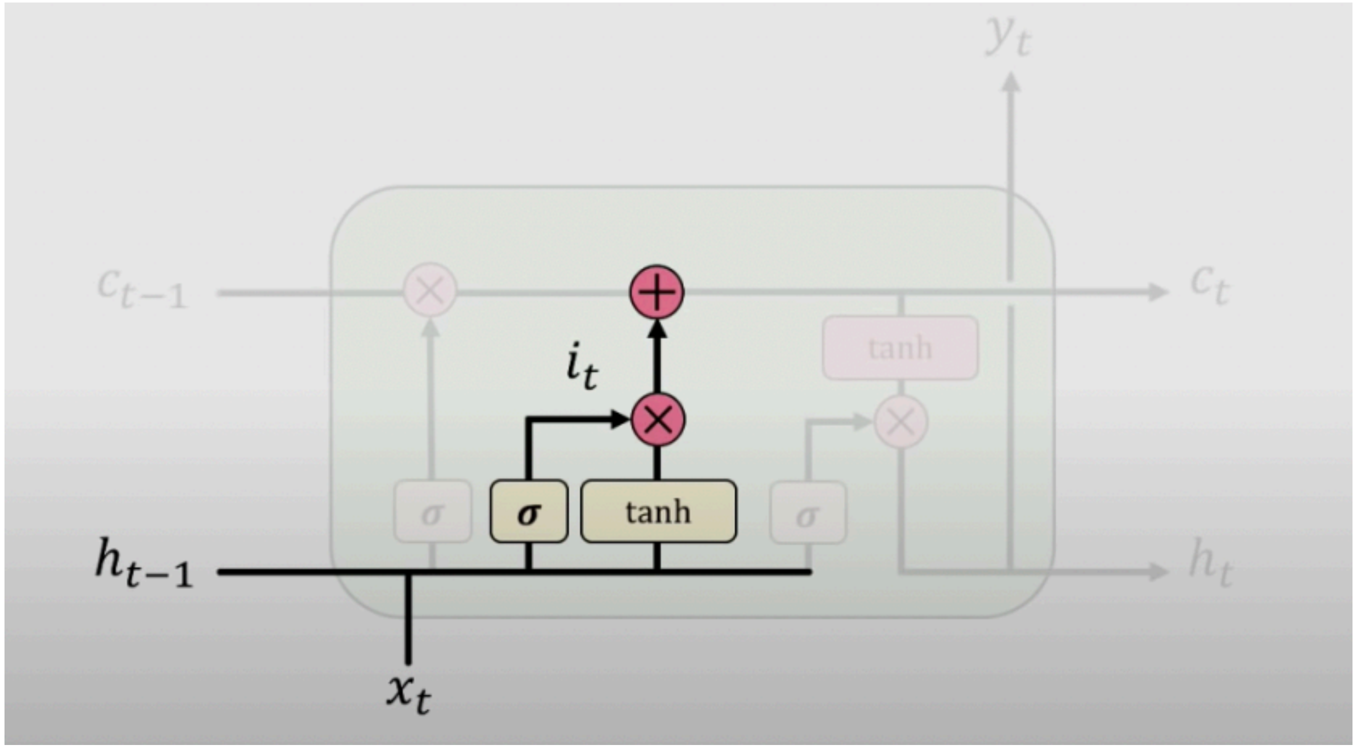
<u>4 main steps:</u>

1-Forget 2-Store 3-Update 4-Output

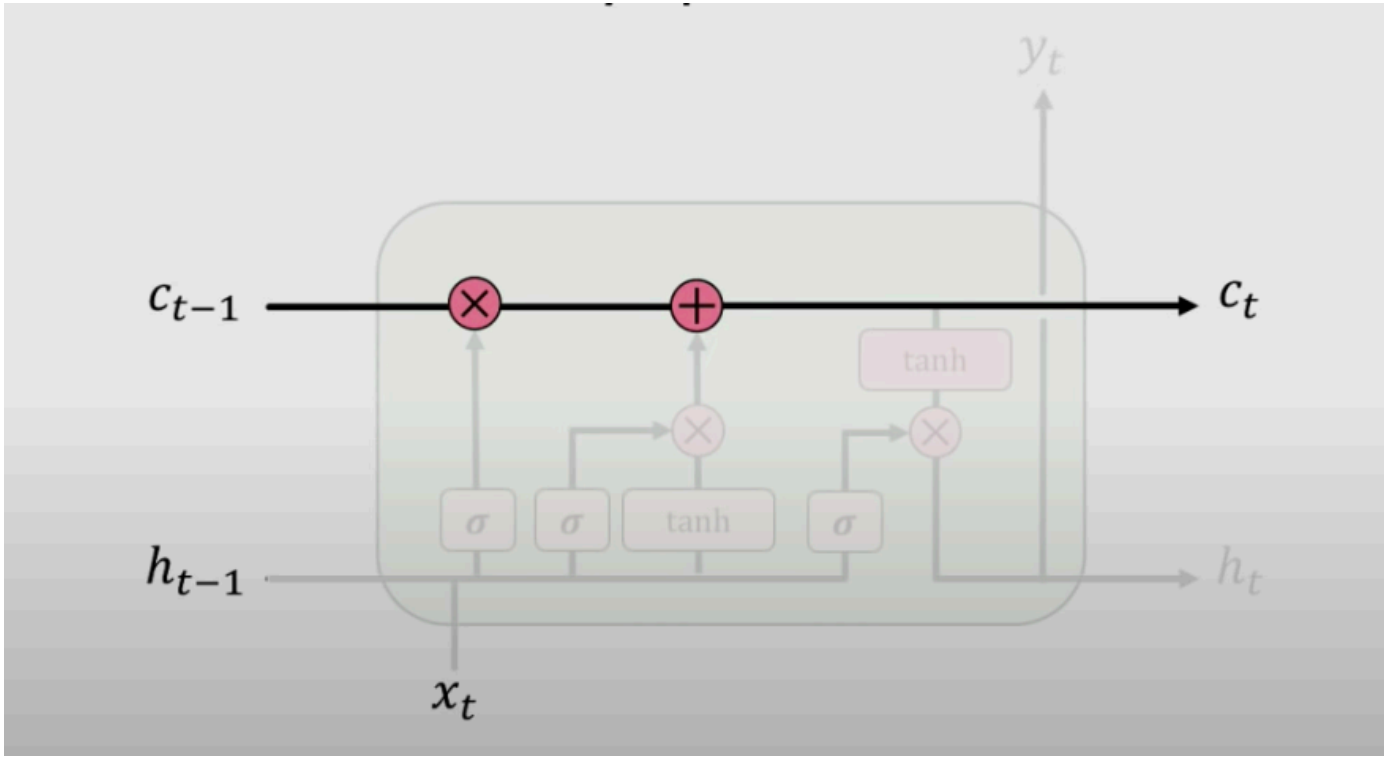
STEP 1: FORGET



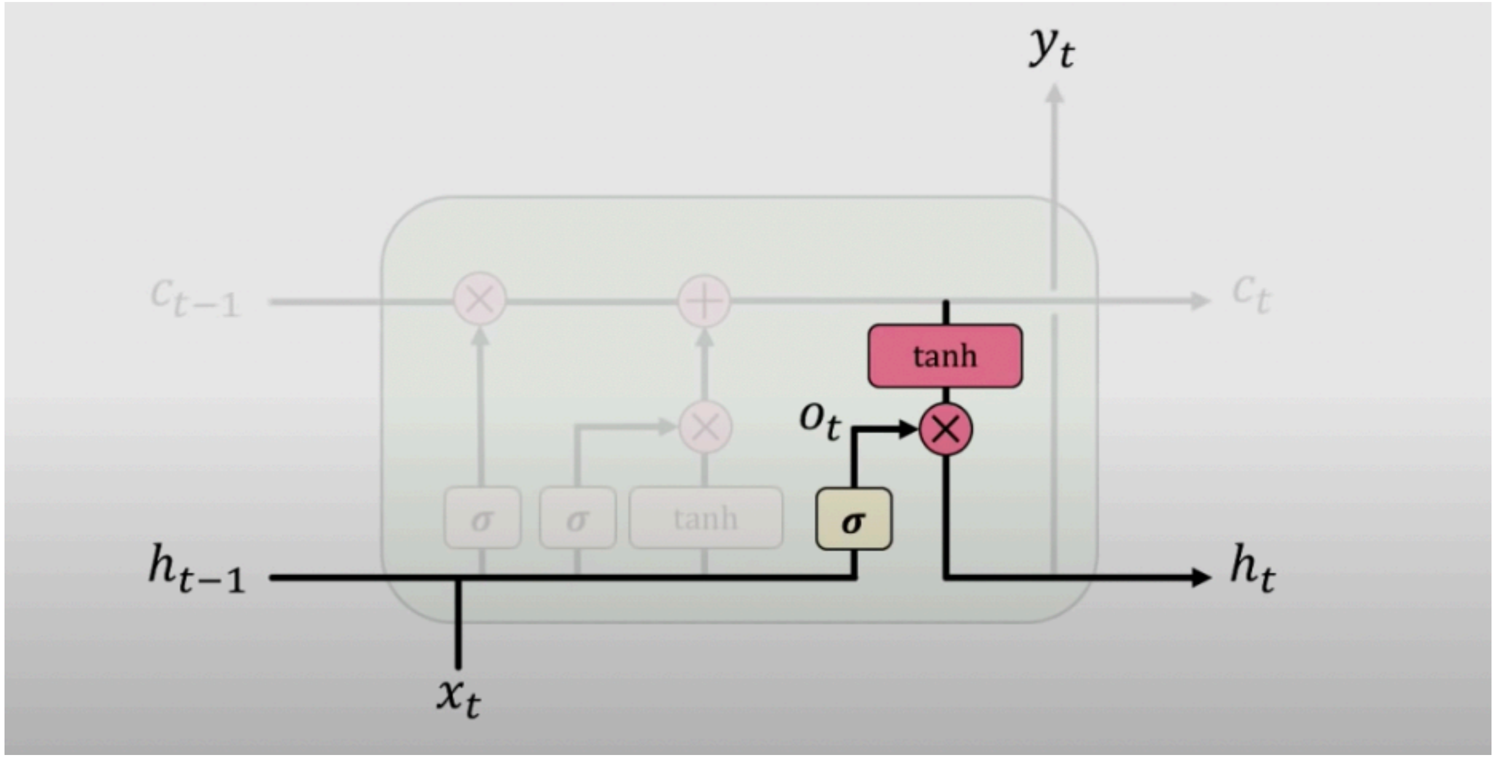
STEP 2: STORE



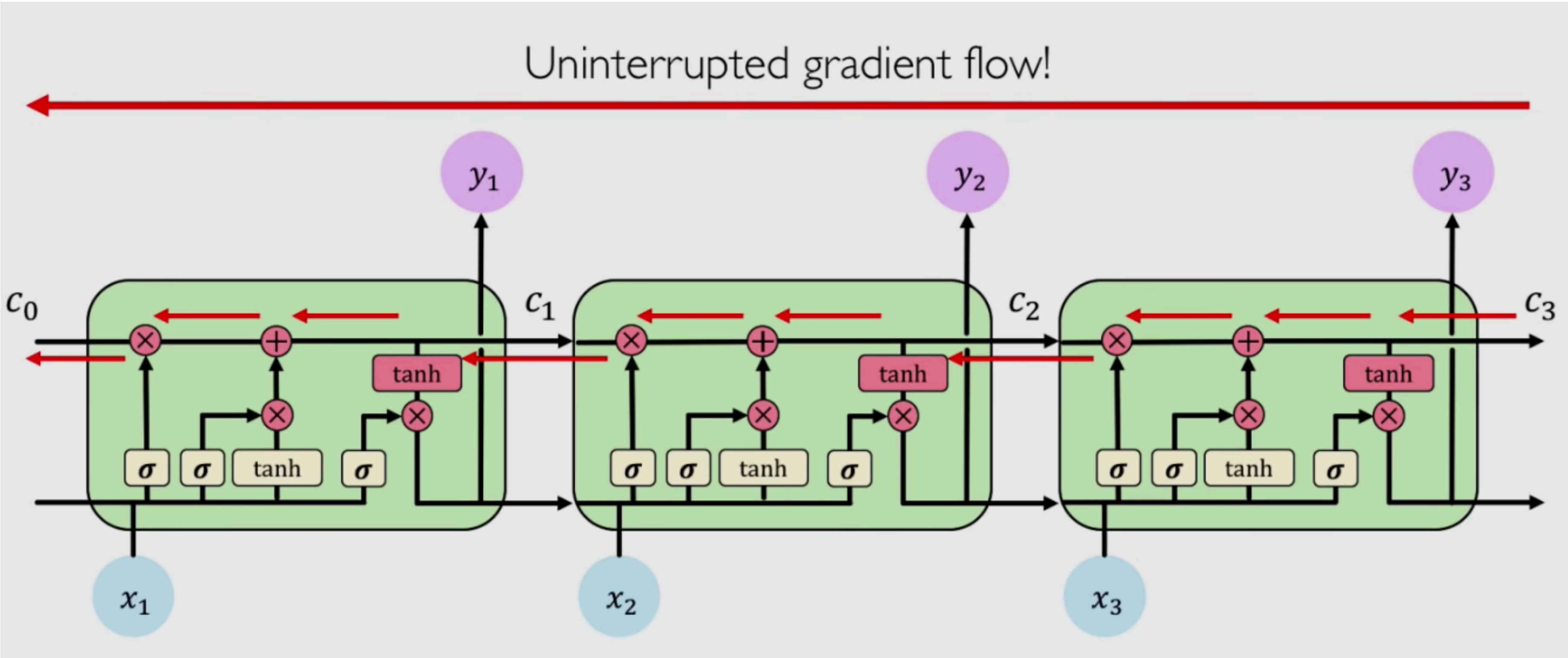
STEP 3: UPDATE



STEP 4: OUTPUT



LSTMs help mitigating the vanishing gradient problem



LSTMs KEY CONCEPTS

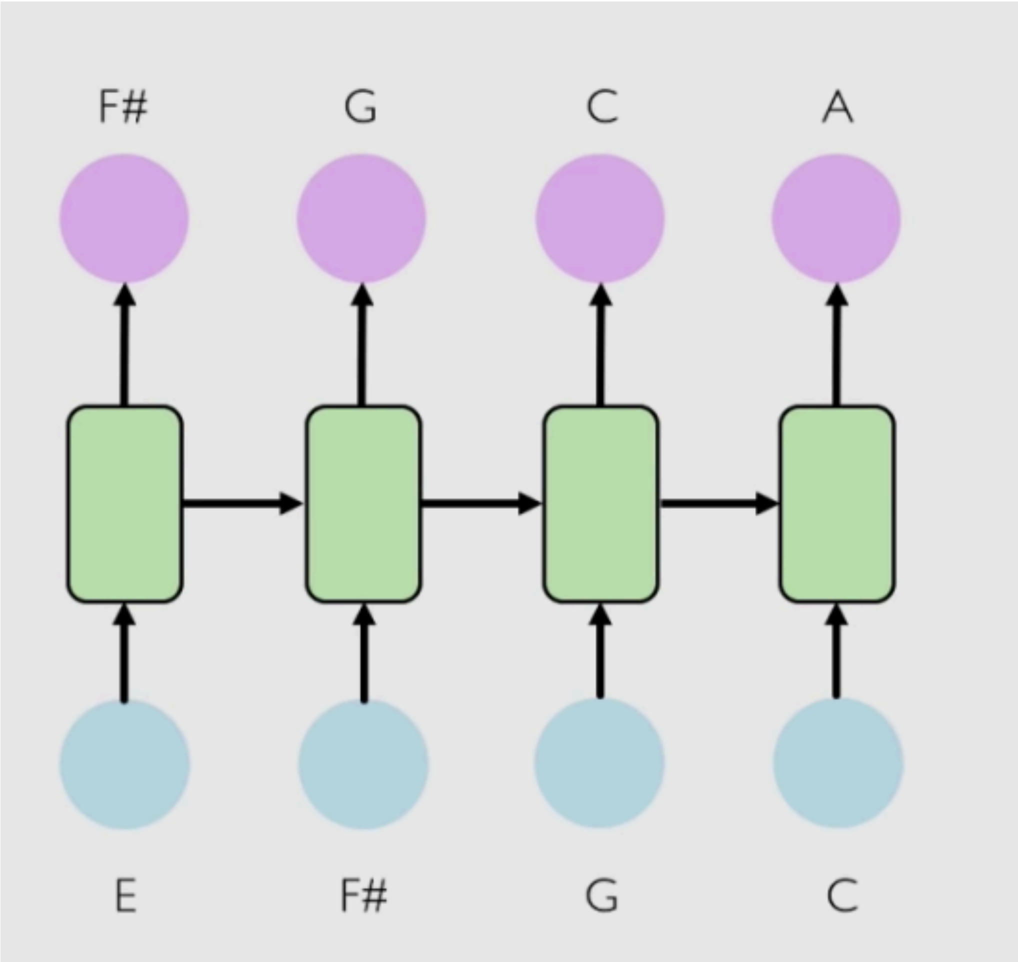
1. Maintain a separate cell state from what is outputted

2. Use gates to control the flow of information

- Forget gates get rid of irrelevant information
- Store relevant information from current input
- Selectively update cell state
- Output gate returns a filtered version of the cell state

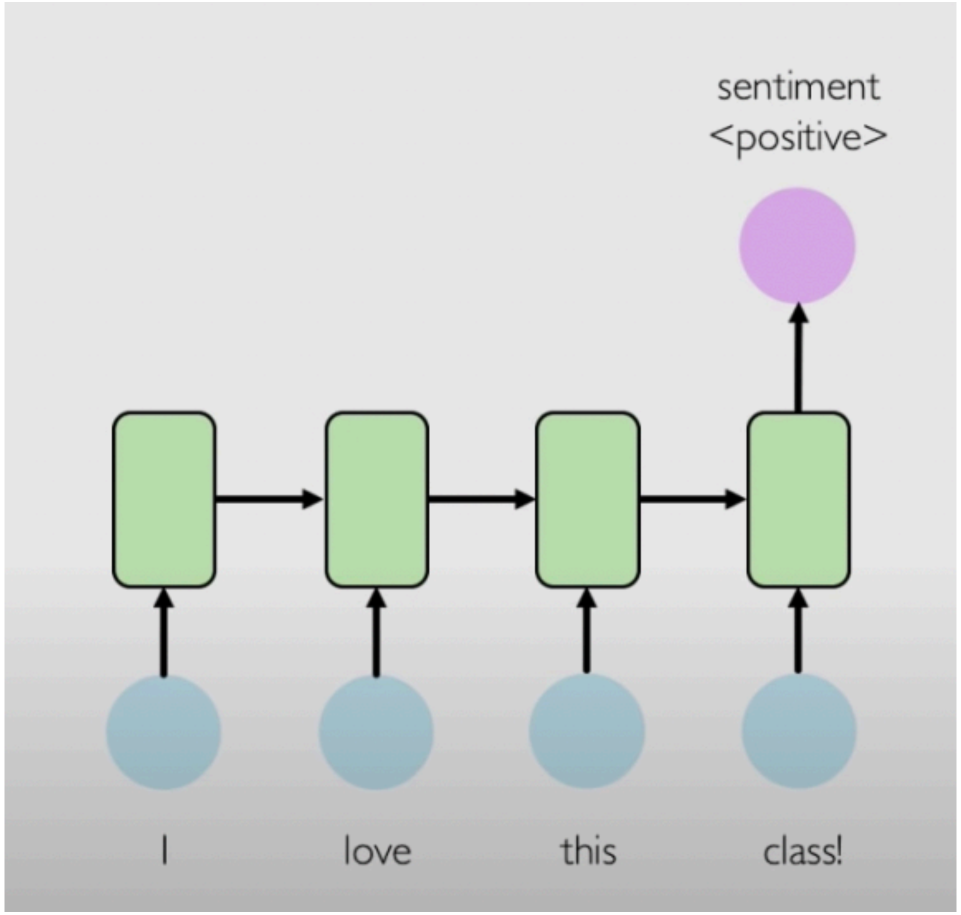
3. Backropagation through time with uninterrupted gradient flow

SOME APPLICATIONS (OUTSIDE ASTRONOMY)



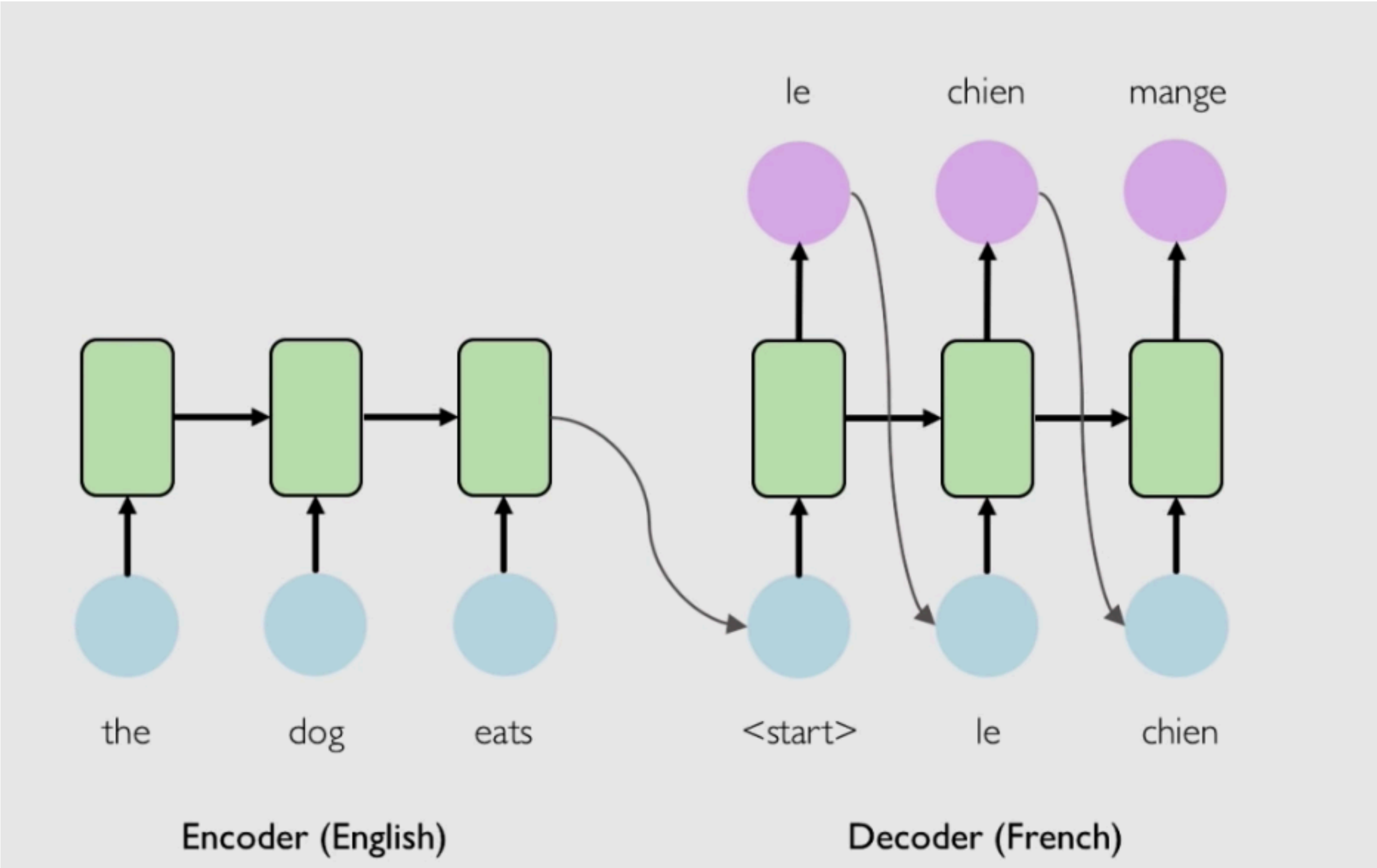
Automatic music generation

SOME APPLICATIONS (OUTSIDE ASTRONOMY)

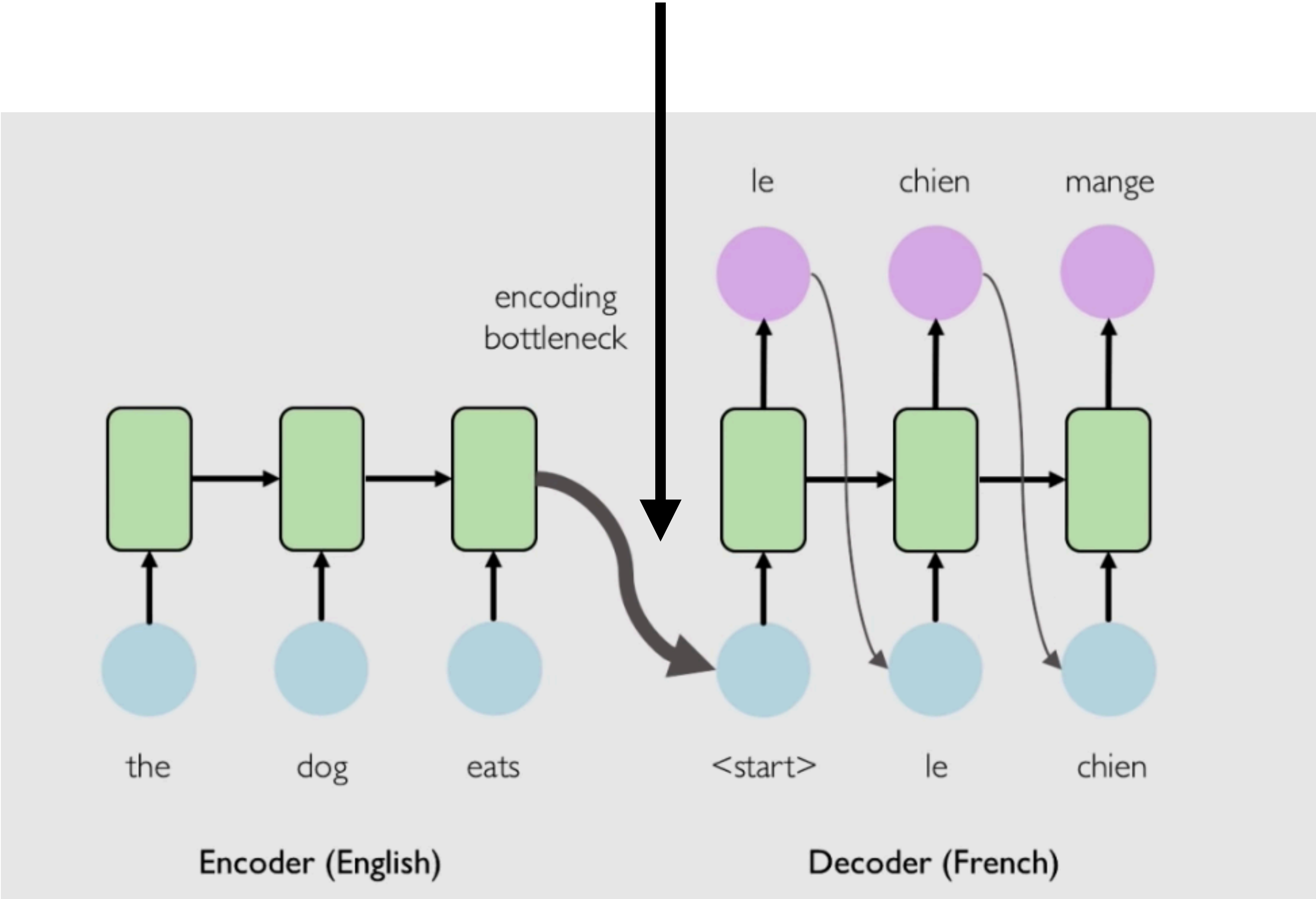


SENTIMENT CLASSIFICATION

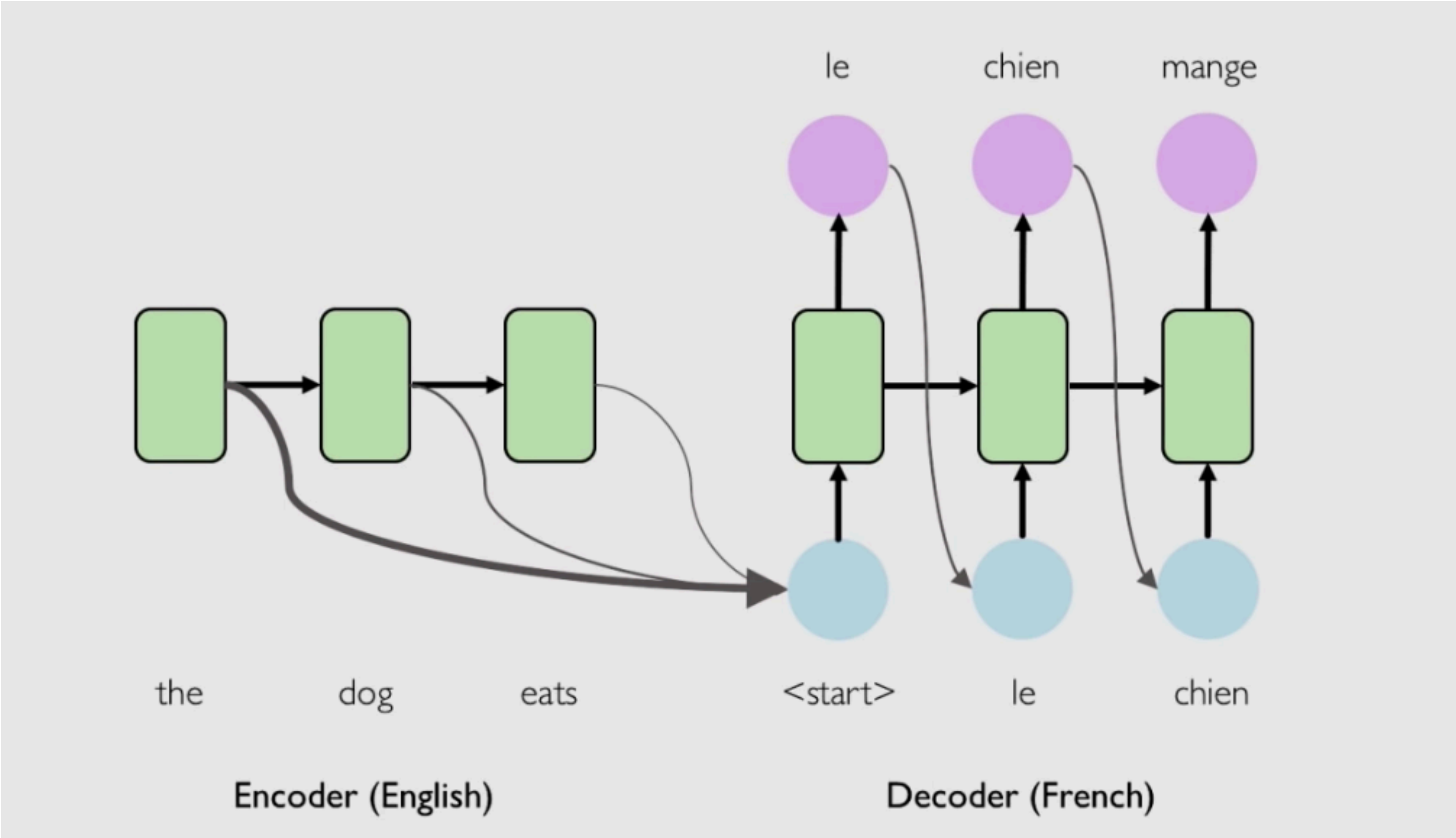
MACHINE TRANSLATION

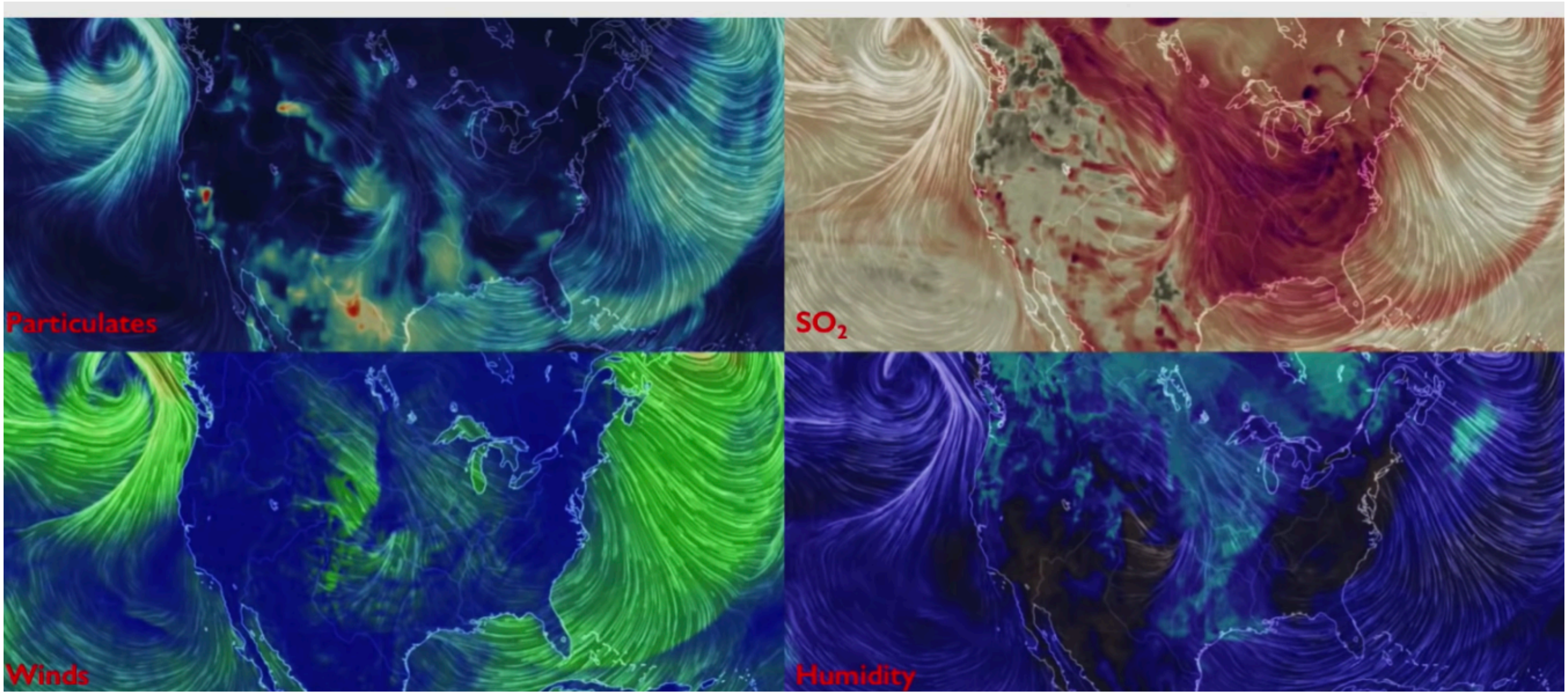


The entire content is encoded into a single vector
(Large information bottleneck)



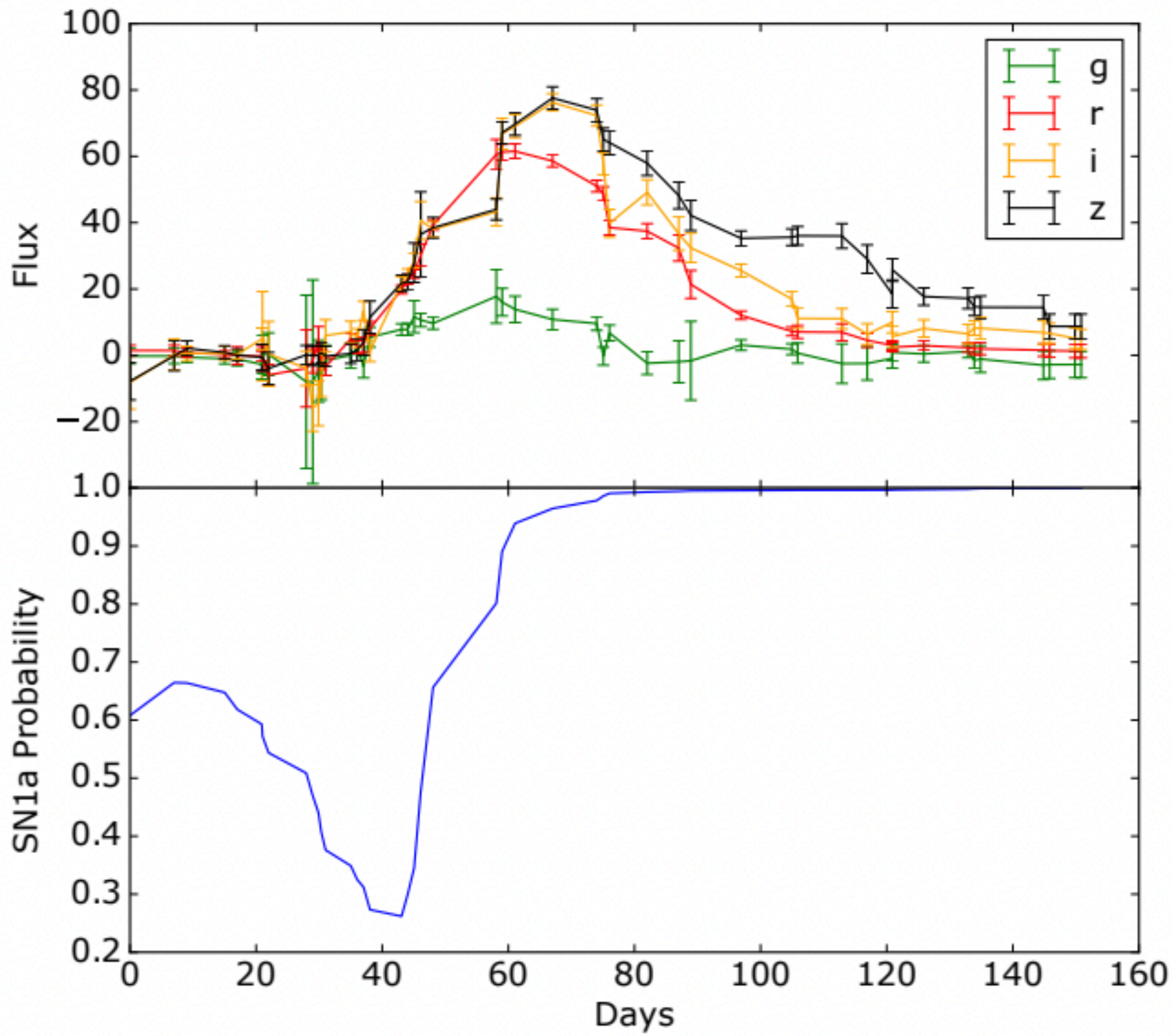
ALL YOU NEED IS ATTENTION: ATTENTION MECHANISMS





<u>ENVIRONMENTAL PREDICTION</u>

SN light curve classification with RNNs



Charnock+17