Winter School on GenAI
CSE, IIT Jodhpur

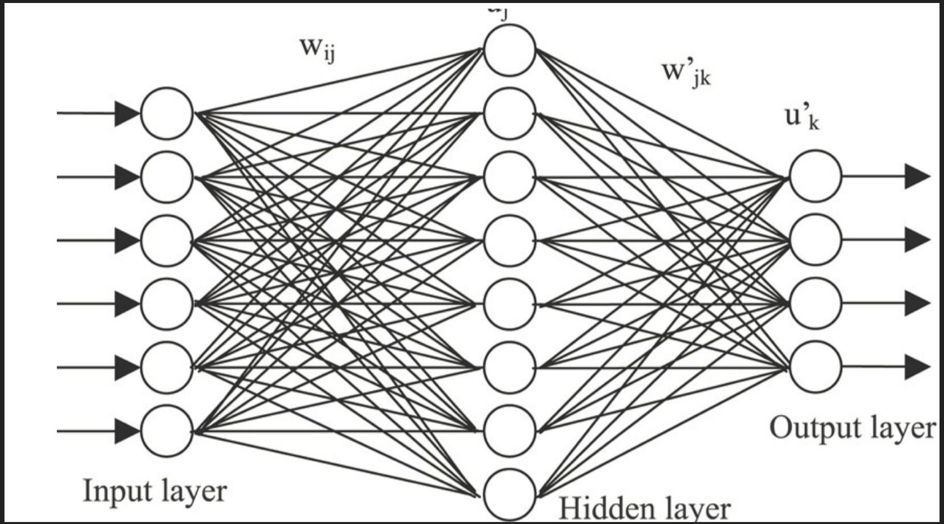
Neural Networks for Sequential Data

Anand Mishra

भारतीय प्रौद्योगिकी संस्थान जोधपुर
Indian Institute of Technology Jodhpur

Recap: Feed Forward Neural Network

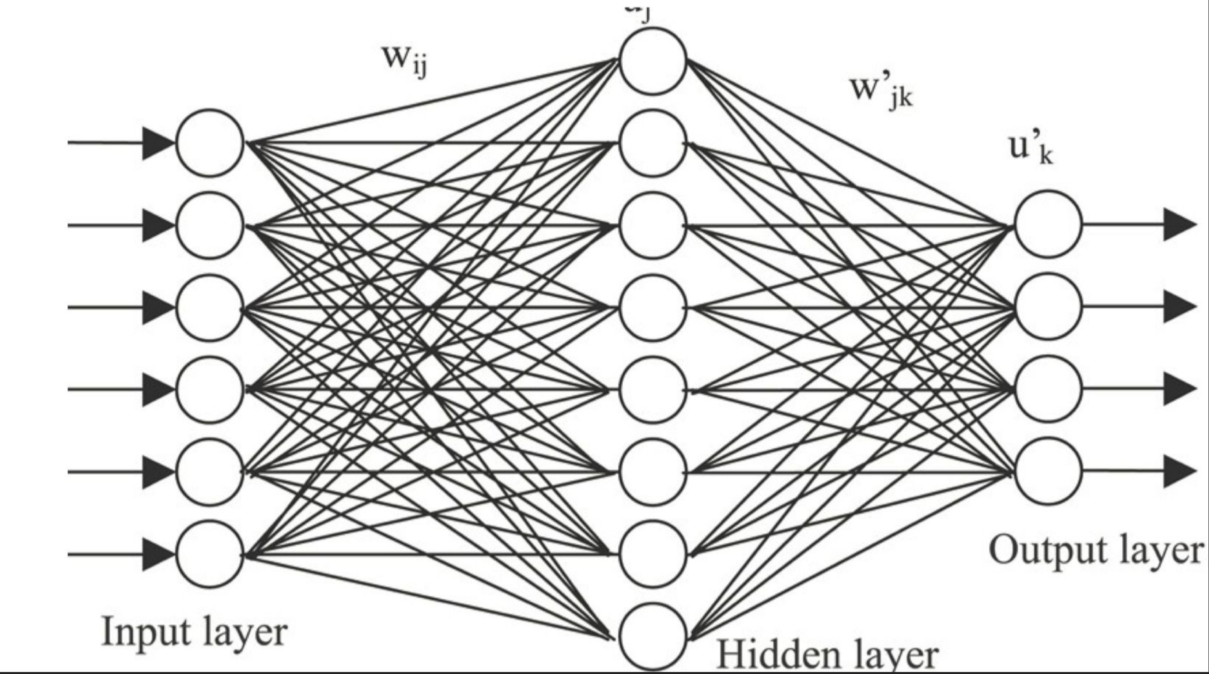
Recap



- Neural network (Network of Perceptron/MLP) can:
- model any Boolean function
- model any decision boundary
- Model any continuous valued function

Recap

Input

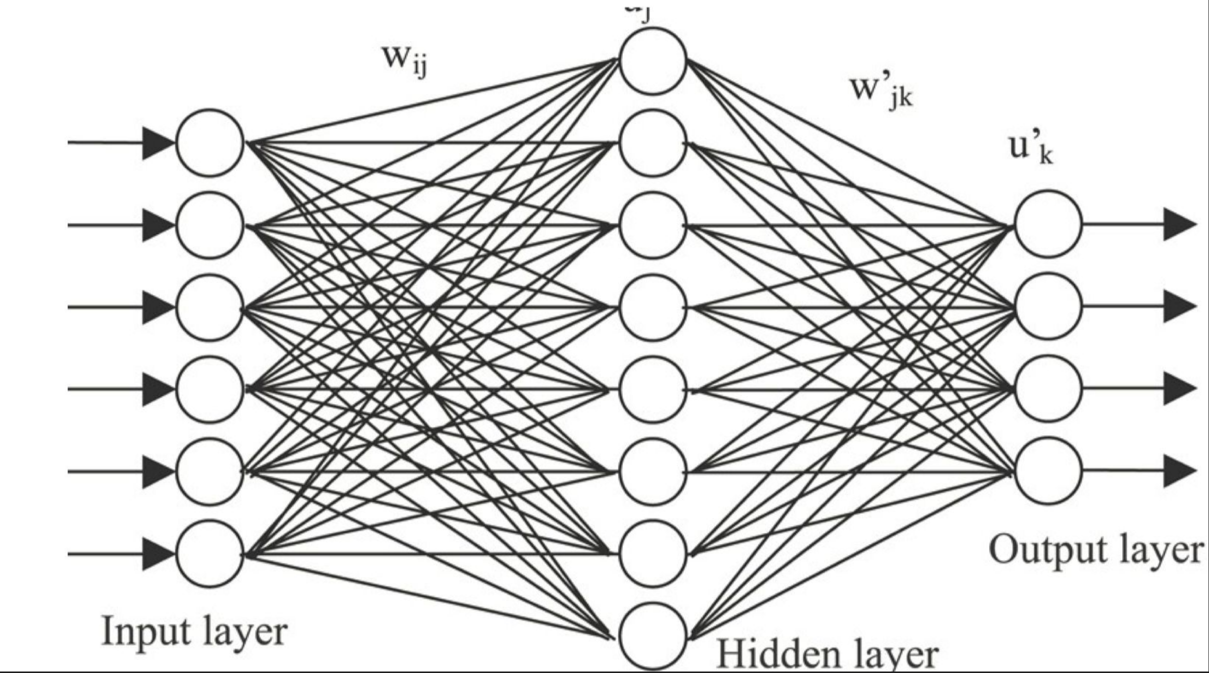


Output

157

Recap

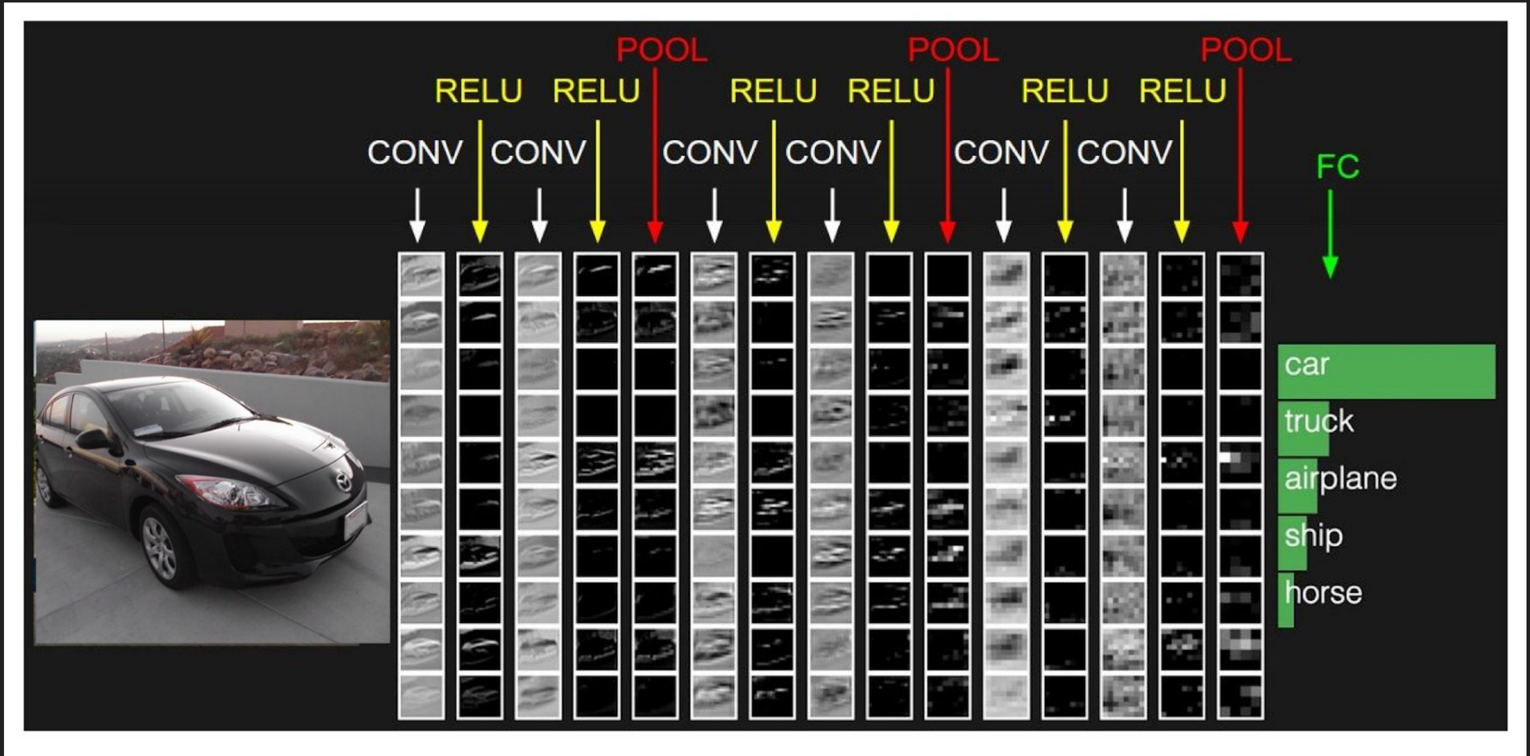
Input



Output

158

Recap: CNN

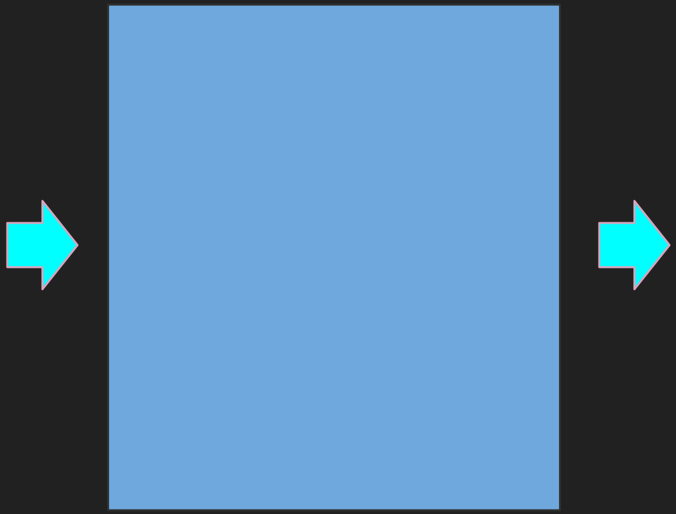

POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

Why worry about sequential data?

Let us look into some exciting ML tasks

Let us look into some exciting ML tasks

Jodhpur is a beautiful city.



जोधपुर एक खूबसूरत शहर है।

Sequence as Input, Sequence as Output

Let us look into some exciting ML tasks

a woman riding a horse jumping over an obstacle.

Sequence as Output

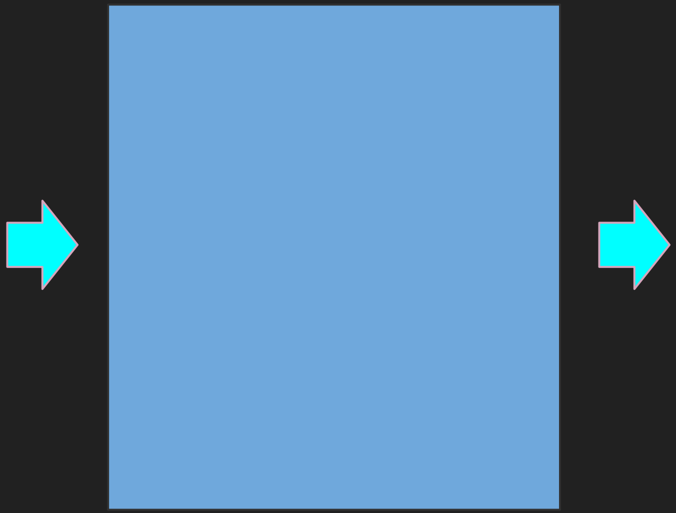
Let us look into some exciting ML tasks

Person riding horse

Sequence as input, Sequence as Output

Let us look into some exciting ML tasks

IIT J is becoming an awesome place to be.



Positive

Sequence as input

Let us look into some exciting ML tasks

Current Indian team is the worst in the history of cricket.

Negative

Sequence as input

slido

Give an example of sequential data problem that we have not discussed so far?

ⓘ Start presenting to display the poll results on this slide.

A Sequence Modeling Problem: Next Word Prediction

Next Word Prediction

I woke up in the morning.

Next Word Prediction

I woke up in the morning.

Let us try solving this problem using simple feed forward neural networks.

Next Word Prediction

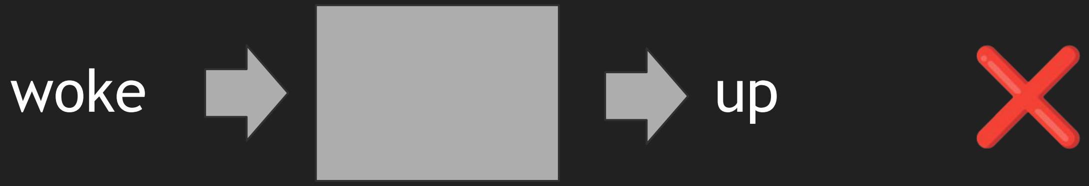
I woke up in the morning.

Let us try solving this problem using simple feed forward neural networks.

How to represent language to a neural network?

Representing Language

Representing Language



Representing Language

Representing Language

1. Define a vocabulary

Vocab = {“I”, “woke”, “up”, “in”, “the”, “morning”}

Representing Language

1. Define a vocabulary
2. Assign unique index to each word in the vocabulary

Vocab = {“I”, “woke”, “up”, “in”, “the”, “morning”}

Index={“I”(1), “woke”(2), “up”(3), “in”(4), “the”(5), “morning”(6)}

Representing Language

1. Define a vocabulary
2. Assign unique index to each word in the vocabulary
3. Represent words using one-hot or learned vector

Vocab = {“I”, “woke”, “up”, “in”, “the”, “morning”}

Index={“I”(1), “woke”(2), “up”(3), “in”(4), “the”(5), “morning”(6)}

Morning: [0 0 0 0 0 1]

Woke: [0 1 0 0 0 0]

Next Word Prediction

I woke up in the morning.

Idea 1: Input words from fixed window

Next Word Prediction

I woke up in the morning.

Idea 1: Input words from fixed window

Problem: Does not capture long-term dependencies.

Odisha is where I grew up, but now I live in Rajasthan. I speak fluent Rajasthani as well as ????

Next Word Prediction

I woke up in the morning.

Idea 2: Input entire string.

Next Word Prediction

I woke up in the morning.

Idea 2: Input entire string.

BoW representation:
[0 1 0 0 1 0 0 1 1 1 1 0 0 0 0 0 0]

slido

What will be the BoW representation for the following sentences: Ram is a good boy. Sita is a good girl. Assume vocabulary=[Ram, Sita, is, a, good, boy, girl]

ⓘ Start presenting to display the poll results on this slide.

Next Word Prediction

I woke up in the morning.

Idea 2: Input entire string.

Problem: Representation can be misleading

Next Word Prediction

I woke up in the morning.

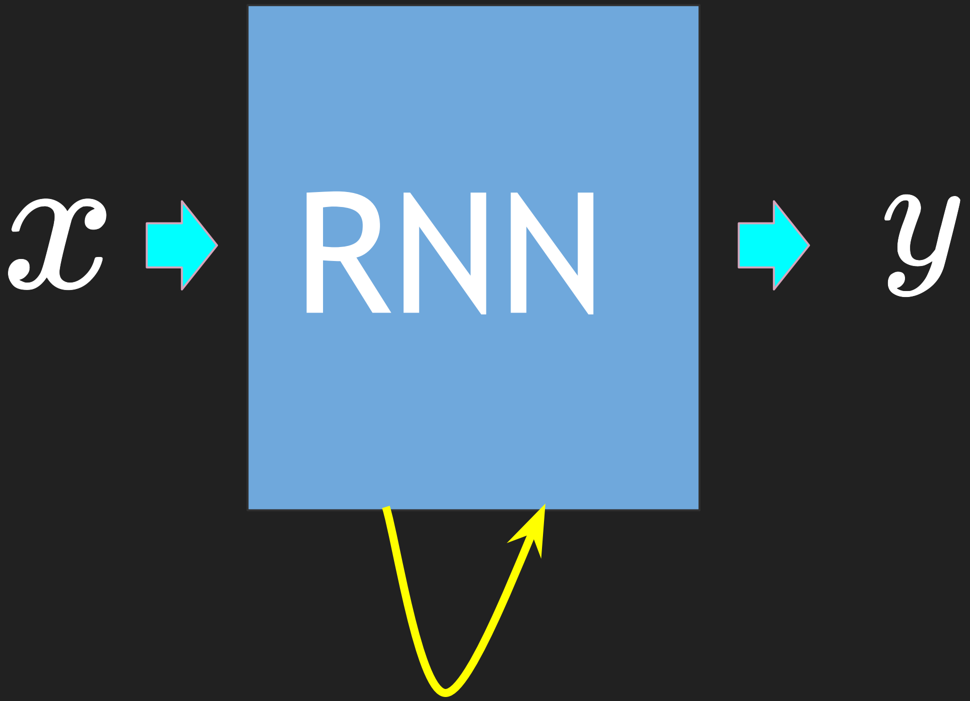
Idea 2: Input entire string.
Problem: Representation can be misleading

Example:
Good movie, not bad at all
Vs
Bad movie, not good at all

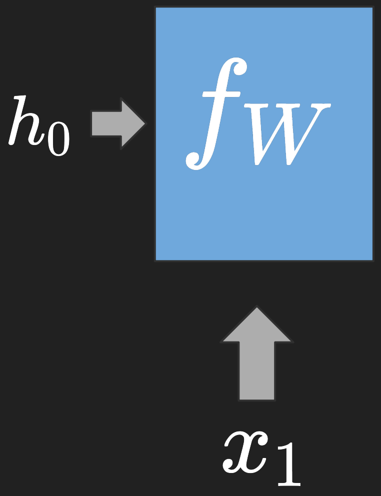
Sequence Modeling: Design Criteria

1. Variable-length sequence
2. Parameter-sharing
3. Long-term dependencies
4. Order of words

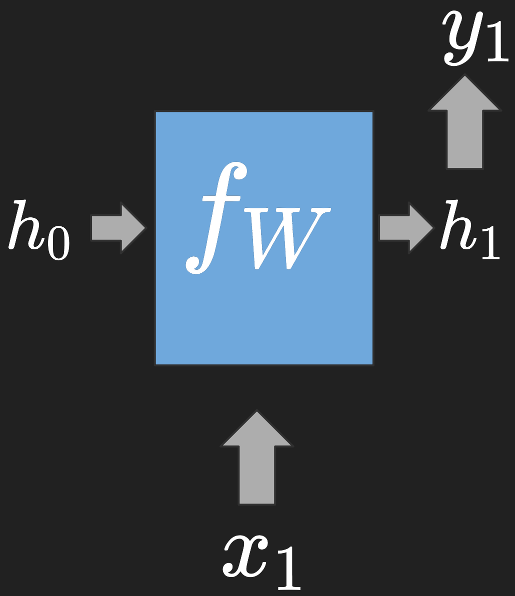
Recurrent Neural Network



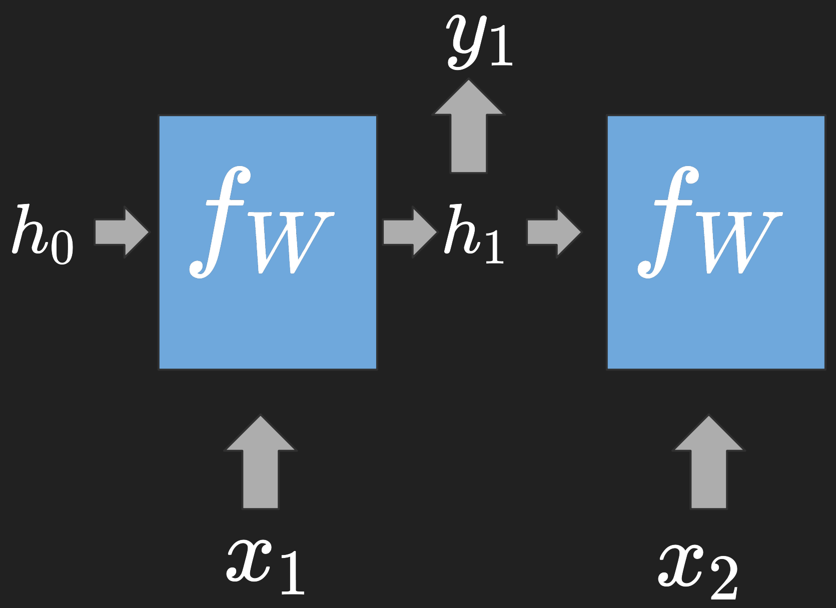
Recurrent Neural Network



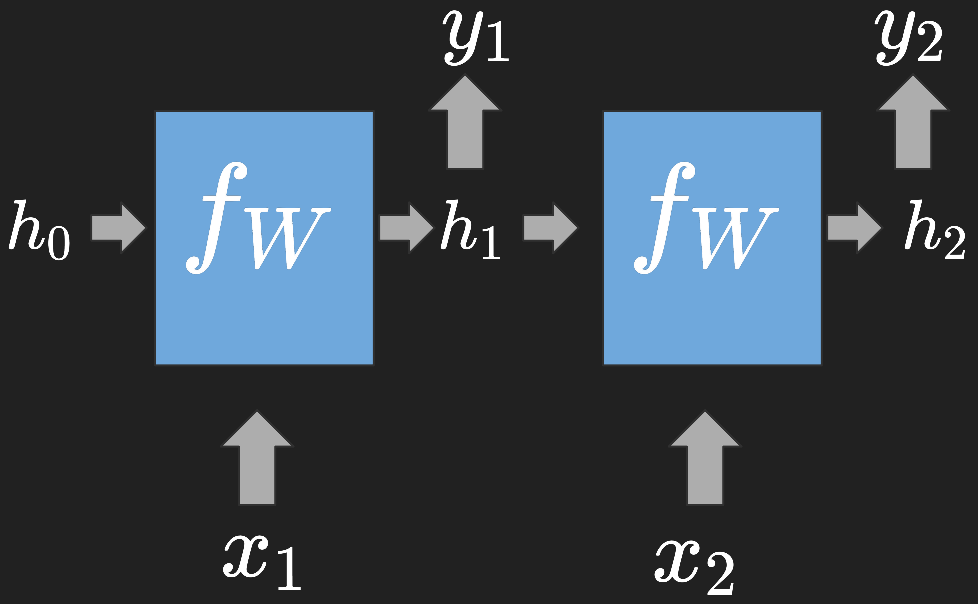
Recurrent Neural Network



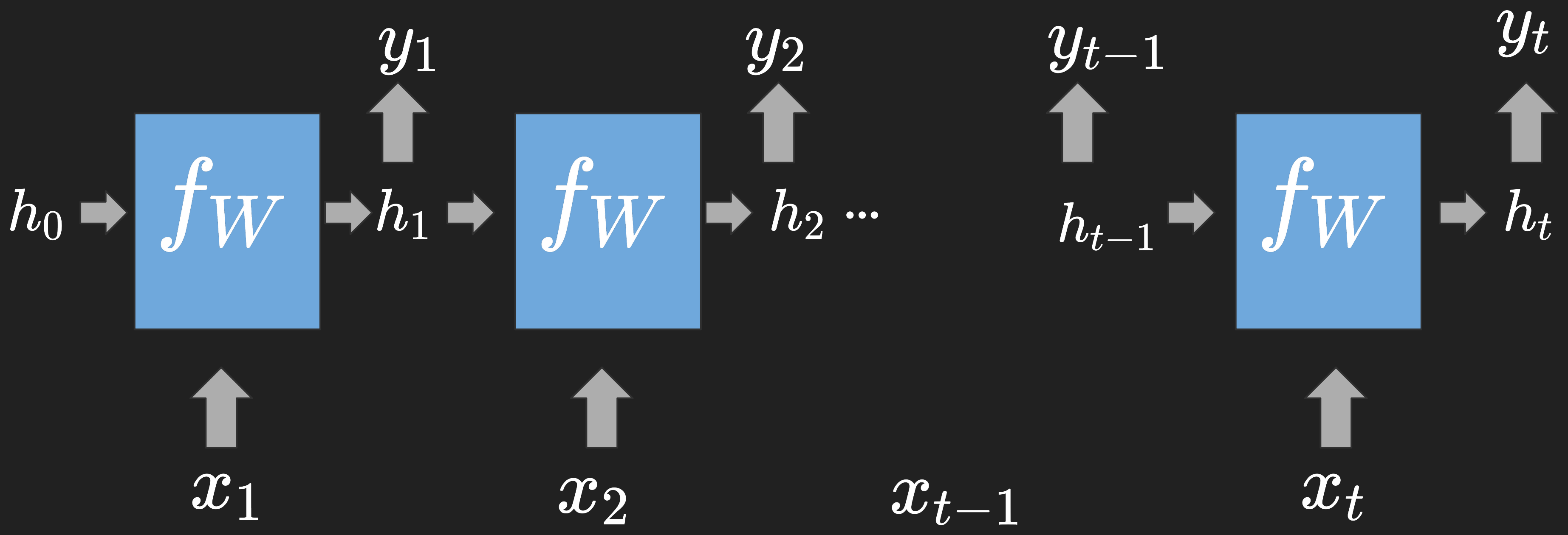
Recurrent Neural Network



Recurrent Neural Network



Recurrent Neural Network



RNN as Recurrence Function

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

RNN as Recurrence Function

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

Cell state

RNN as Recurrence Function

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

Cell state

Parameterized function

RNN as Recurrence Function

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

Cell state

Parameterized function

Previous cell state

RNN as Recurrence Function

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

Cell state

Parameterized function

Previous cell state

Input vector at time t

RNN as Recurrence Function

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

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

RNN as Recurrence Function

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

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

Non-linearity

Learnable Matrix

slido

Suppose hidden states has size h X 1 and input has size d X 1. Then, what should be the size of W_{hh} and W_{xh} respectively?

ⓘ Start presenting to display the poll results on this slide.

RNN as Recurrence Function

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

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

$$\hat{y}_t = W_{hy}^T h_t$$

In the context of h_t consider the following statements:

(i) size of h_{t-1} and h_{t} are same.

(ii) h_{t} contains values between -1 to 1.

(iii) h_{t} is scalar.

(iv) size of h_{t} and x_{t} has to be same.

Which among the above are TRUE?

RNN: few line of implementation

```
my_rnn=RNN()
hidden_state=[0,0,0,0]
sentence = ["I", "love", "recurrent", "neural"]

for word in sentence:
    pred, hidden_state = my_rnn(word,hidden_state)

next_word = pred
```

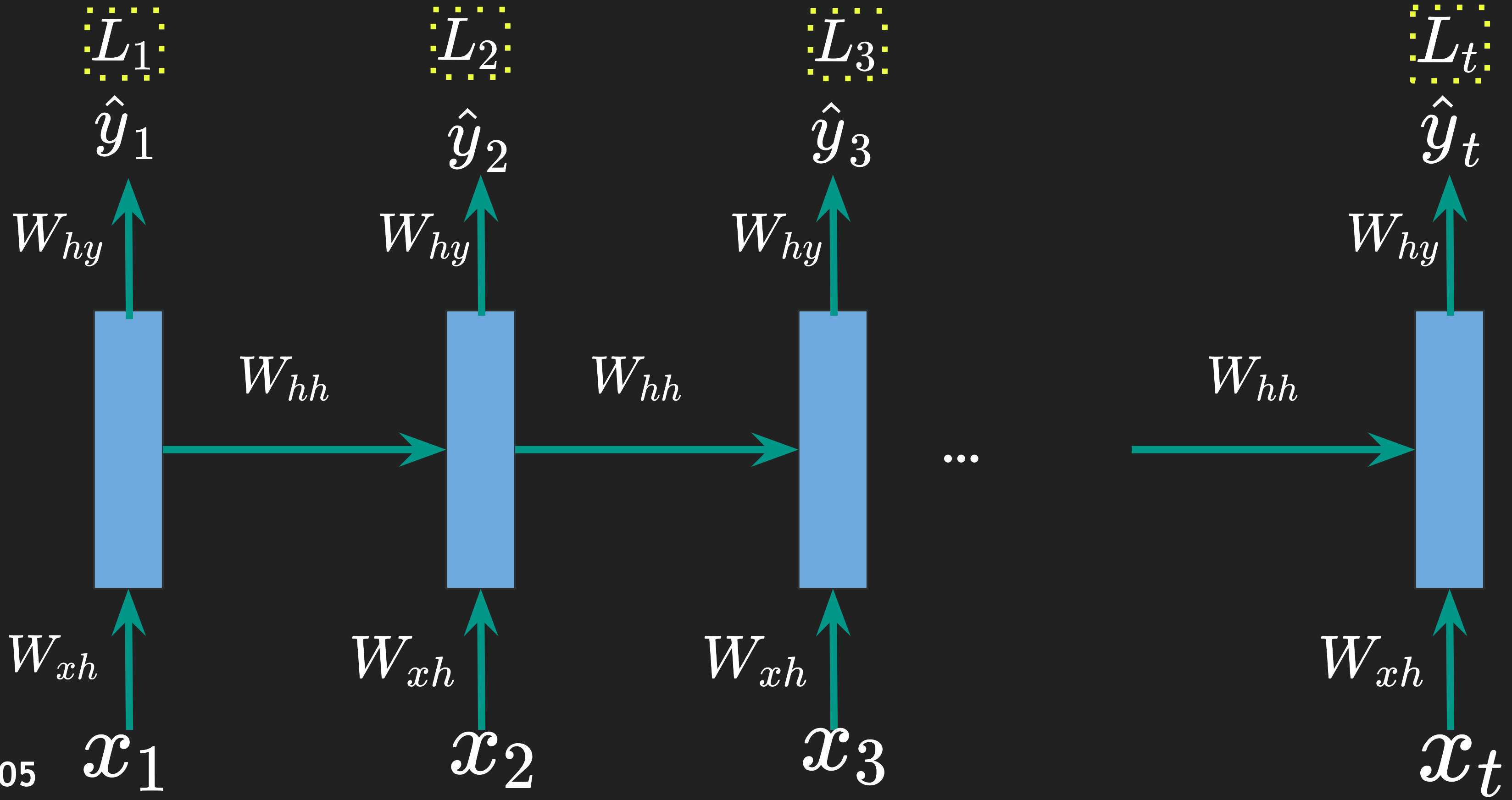
RNN as python code

```
class myrnn(tf.keras.layers.Layer):
    def __init__(self,rnn_units,input_dim,output_dim):
        super(myrnn,self).__init__()

        #initalize weight matrix
        self.W_xh=self.add_weight([rnn_units, input_dim])
        self.W_hh=self.add_weight([rnn_units, rnn_units])
        self.W_hy=self.add_weight([output_dim, rnn_units])

        #initialize hidden state with zeros
        self.h=tf.zeros([rnn_units, 1])
```

RNN as python code

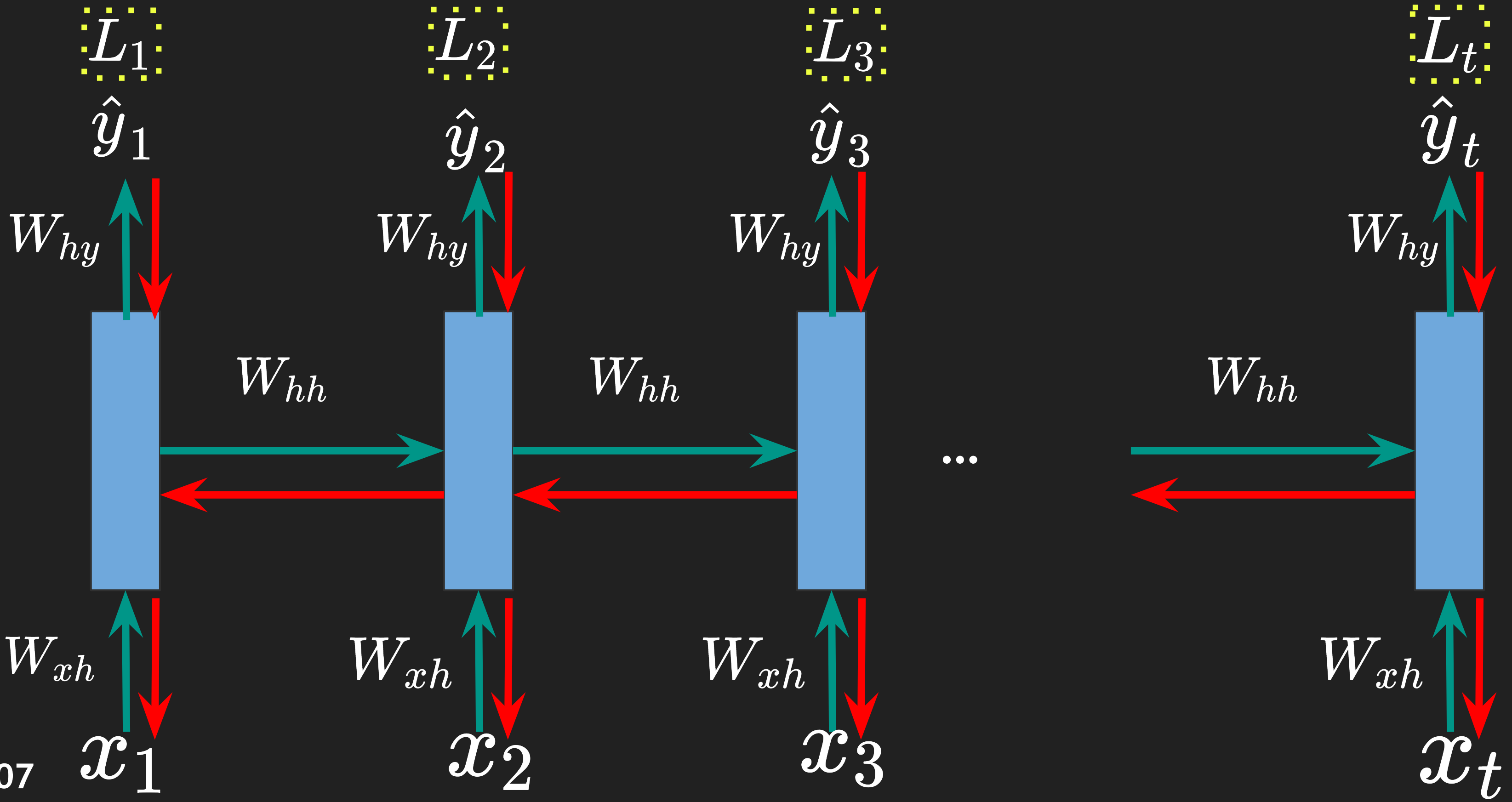
```
def call(self,x):
    #update hidder state
    self.h=tf.math.tanh(self.W_hh*self.h + self.W_xh*x
        )
    #comput output
    output=self.W_hy*self.h

    return output,self.h
```

RNN as Computation Graph



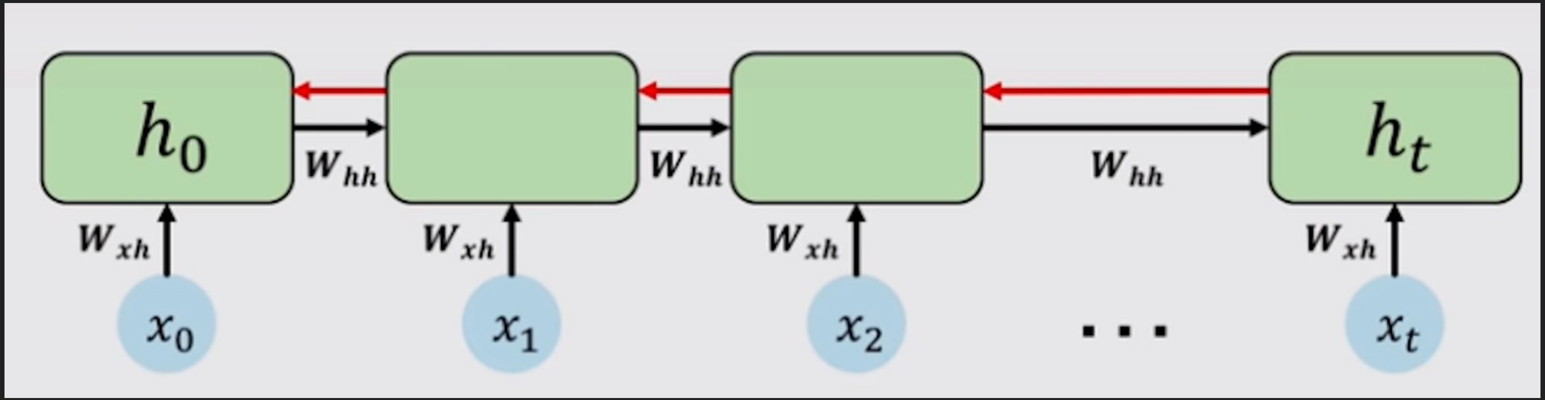
Backpropagation Through Time (BPTT)

RNN: Backprop through Time (BPTT)



Problem with RNN

Exploding Gradient

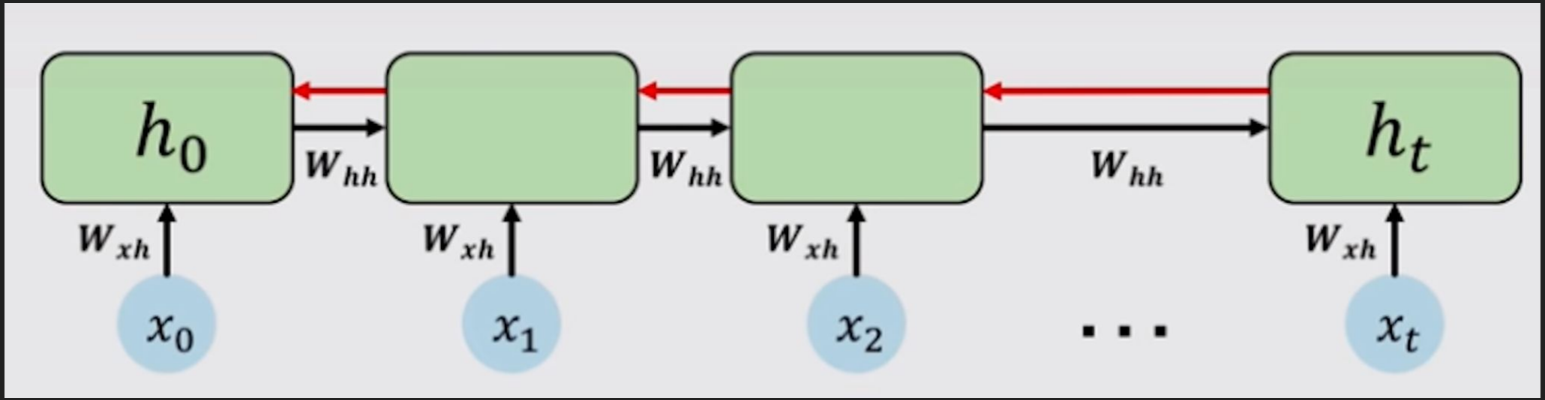


1. Computing gradient wrt h_0 requires many factors of W_{hh} and repeated gradient computation

   If Many values > 1

   exploding gradients

Exploding Gradient

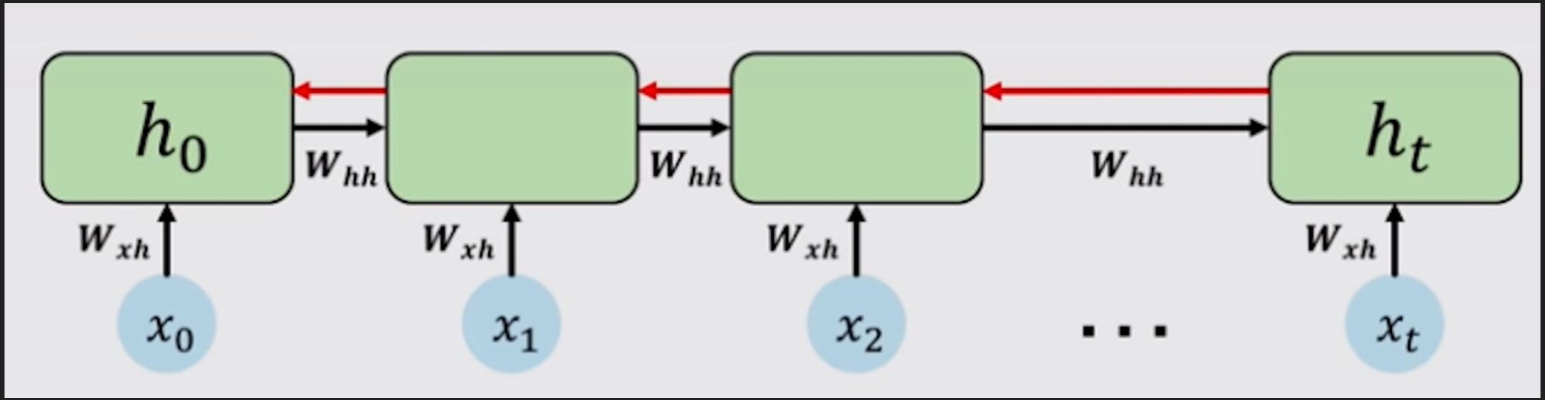


1. Computing gradient wrt h_0 requires many factors of W_{hh} and repeated gradient computation

   If Many values > 1 exploding gradients

   Solution: Gradient Clipping

Vanishing Gradient



1. Computing gradient wrt h_0 requires many factors of W_{hh} and repeated gradient computation

   If Many values < 1 = vanishing gradients

Vanishing Gradient

Why Vanishing gradient is a problem?

Vanishing Gradient

Why Vanishing gradient is a problem?

Multiply many small numbers together.

Vanishing Gradient

Why Vanishing gradient is a problem?

Multiply many small numbers together.

$\longrightarrow$ Errors due to further back time steps have smaller and smaller gradients

Vanishing Gradient

Why Vanishing gradient is a problem?

Multiply many small numbers together.

$\longrightarrow$ Errors due to further back time steps have smaller and smaller gradients

$\longrightarrow$ Bias parameters to capture short-term dependencies.

Vanishing Gradient

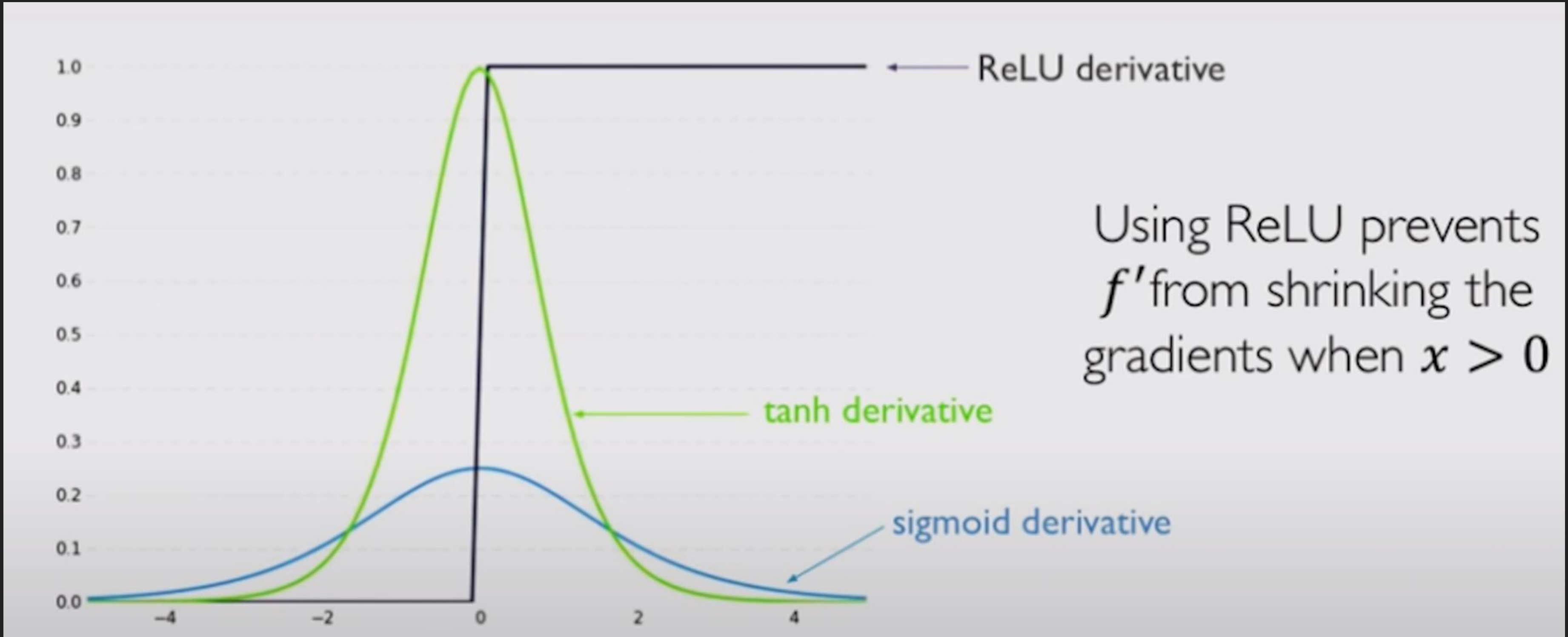
Short-term dependency is not always bad.

For example: I wake up early in the morning.

But also recall the following example:

Odisha is where I grew up, but now I live in Rajasthan. I speak fluent Rajasthani as well as ????

Solution-1 (activation function)



Solution-2 (Initialization)

Initialize weight matrix to Identity matrix and bias to zero.

$\rightarrow$ This helps prevent weights from shrinking to zero.

Solution-3 (Gated Cell)

Solution-3 (Gated Cell)

Gates in recurrent cells control the information flow.

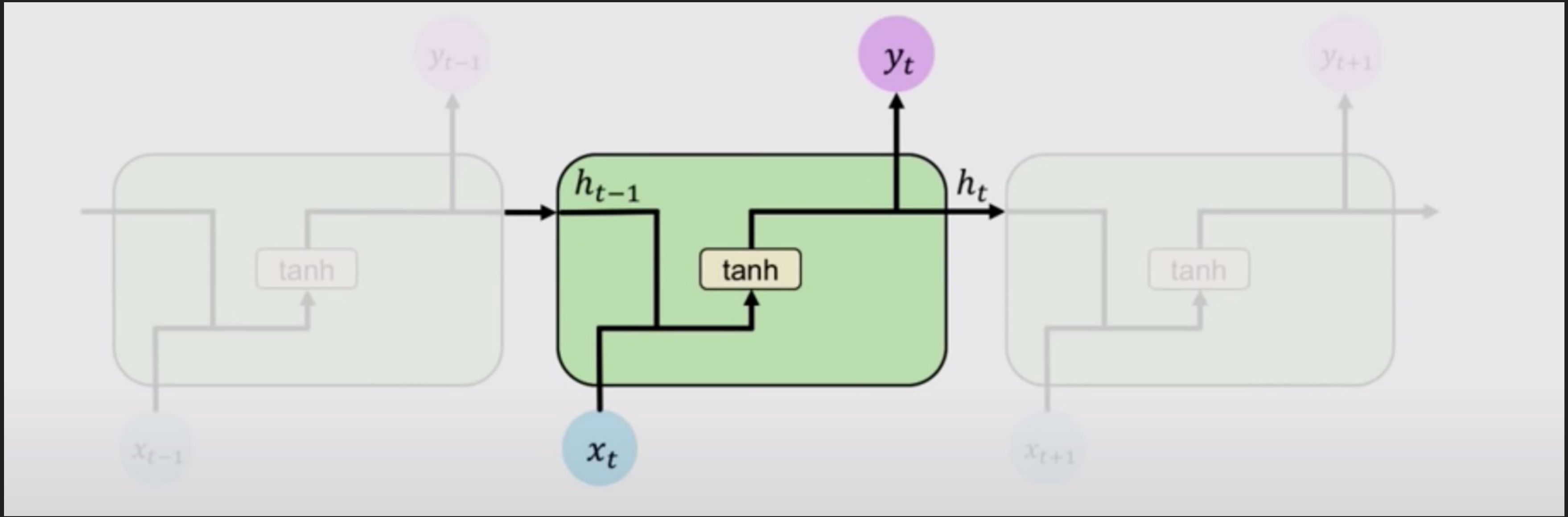
$\rightarrow$ LSTMs, GRU, etc.

LSTM

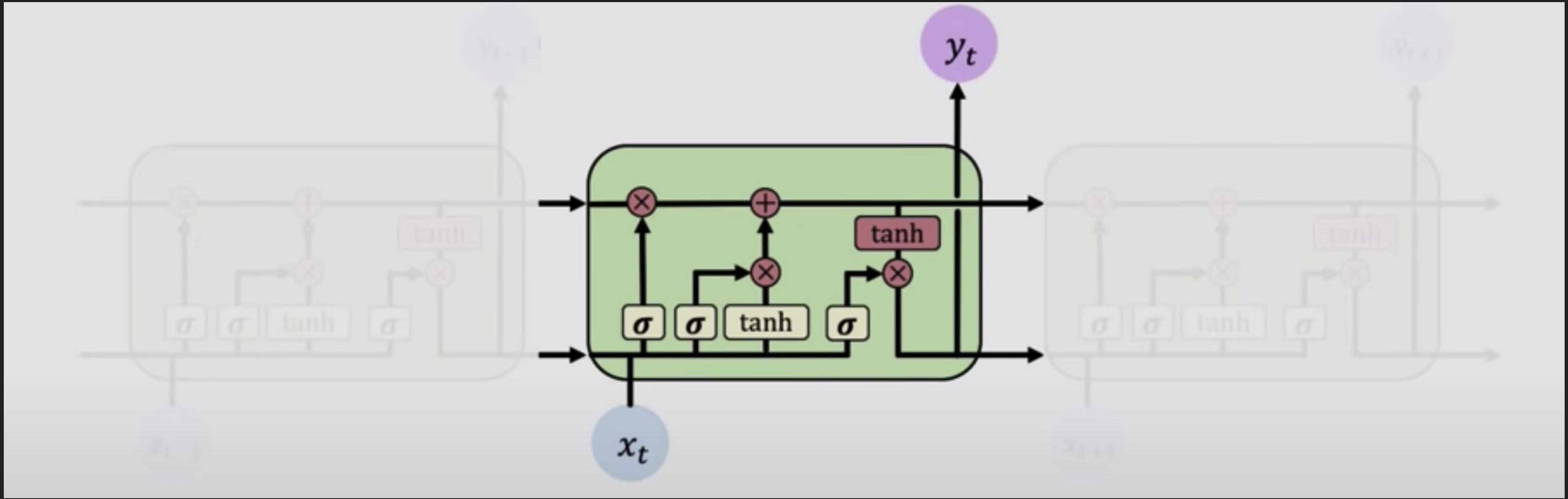
‘

Long Short Term Memory Network

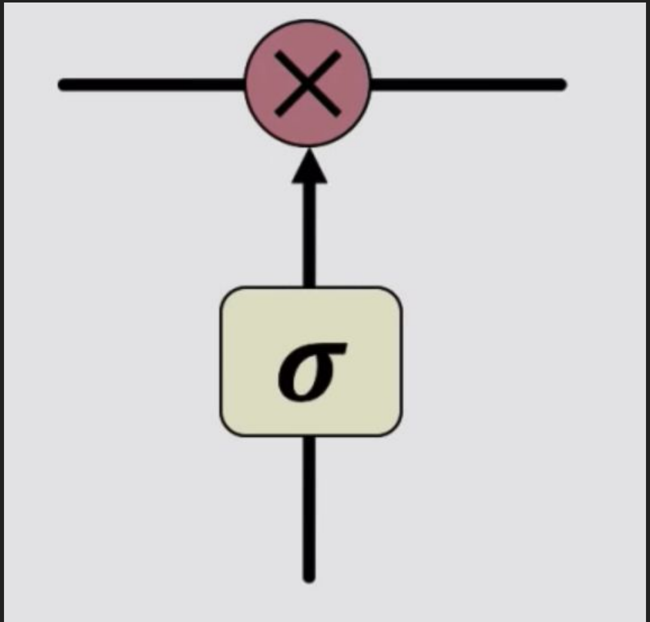
Back to RNN (as a computational cell)



LSTM as Gated Cell

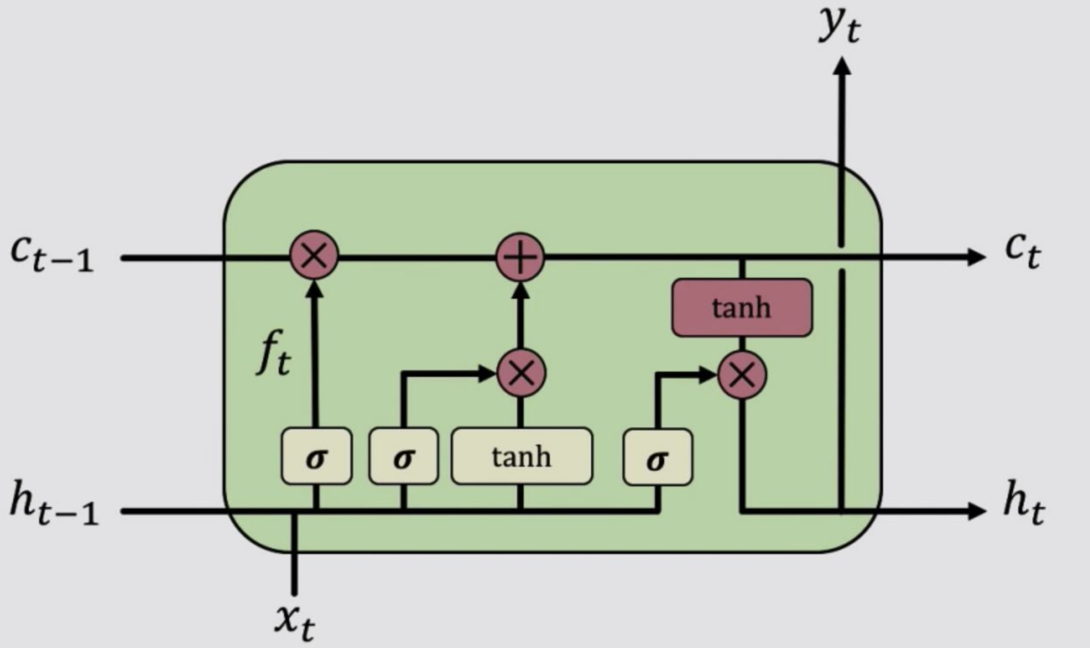


What is Gate?



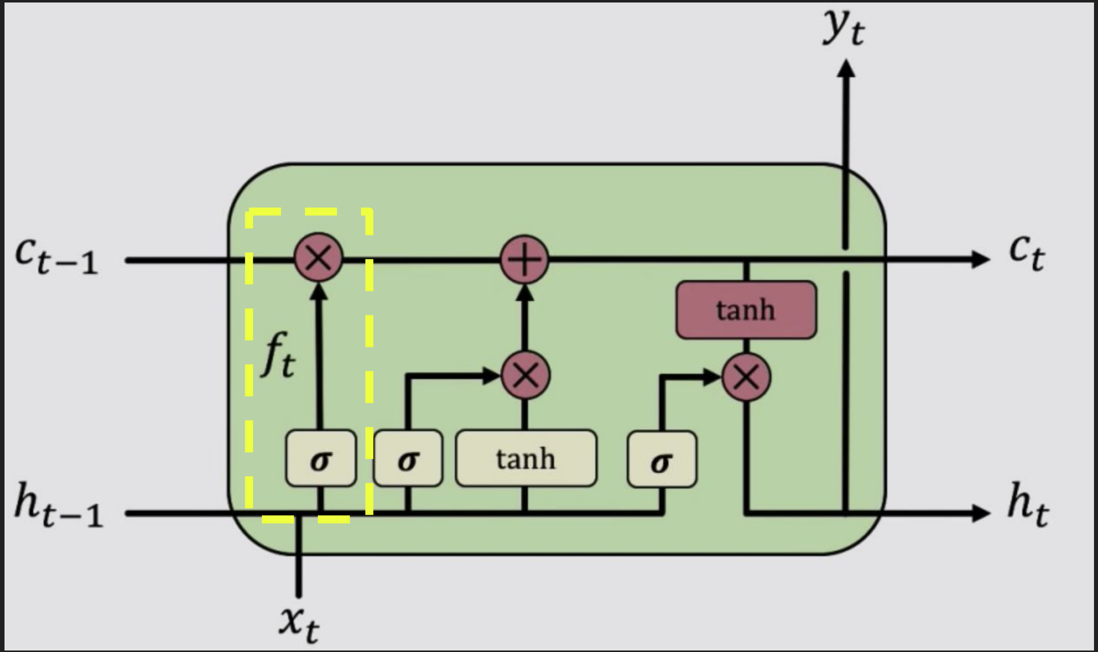
Information either pass or obstructed through a gate

How LSTMs work?



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

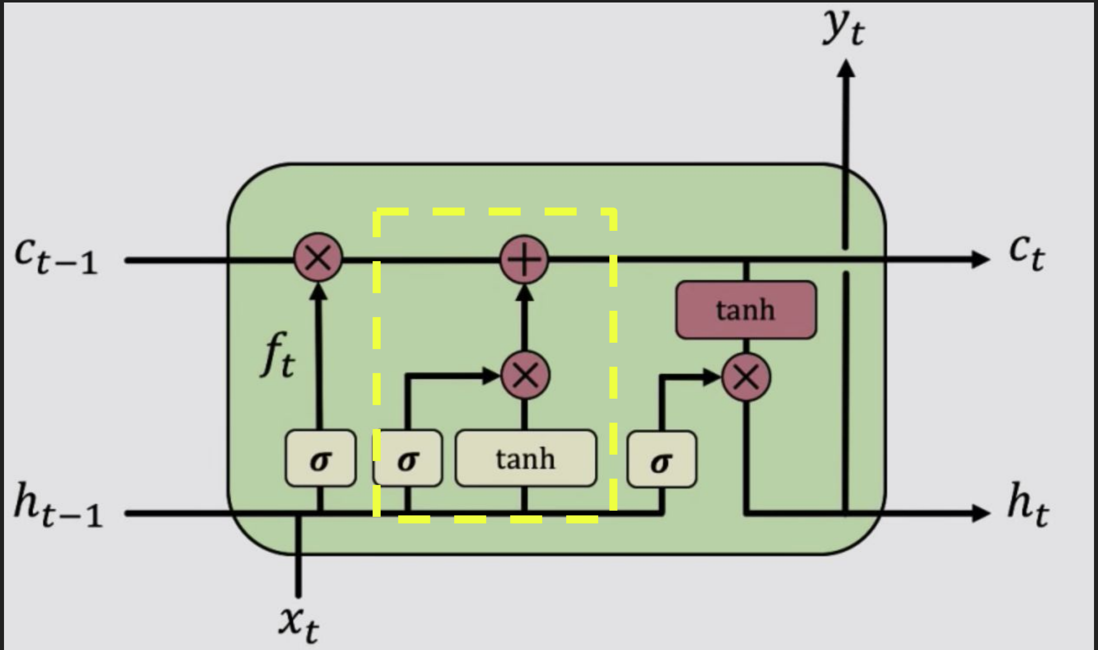
How LSTMs work?



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

LSTMs forget irrelevant part of previous hidden state

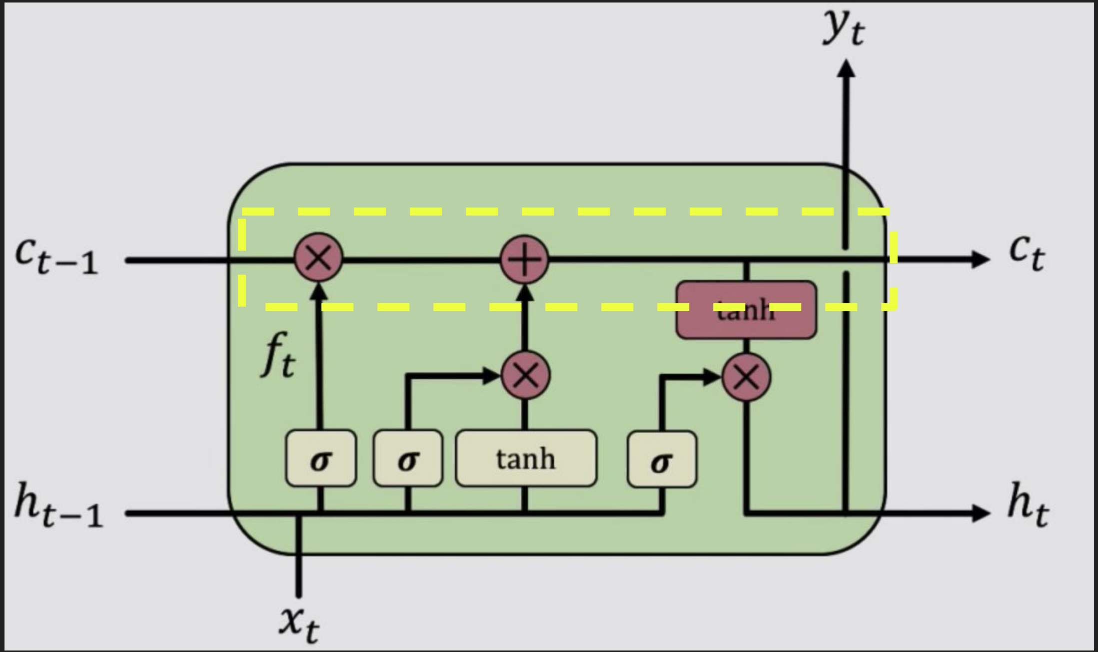
How LSTMs work?



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

LSTMs stores relevant new information into cell state

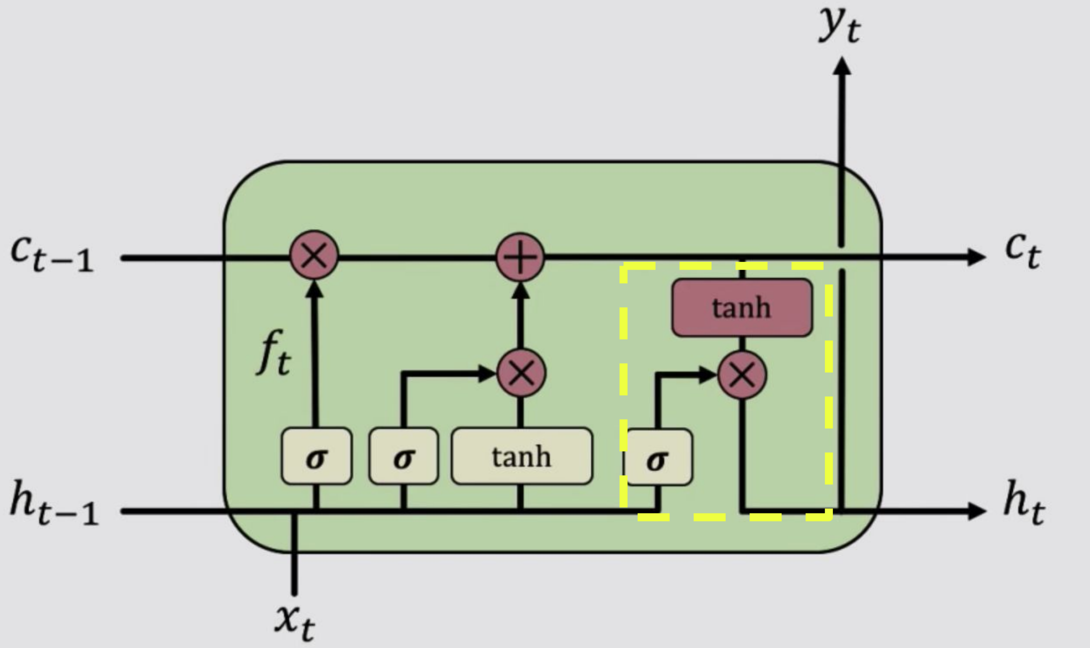
How LSTMs work?



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

LSTMs selectively update cell state values

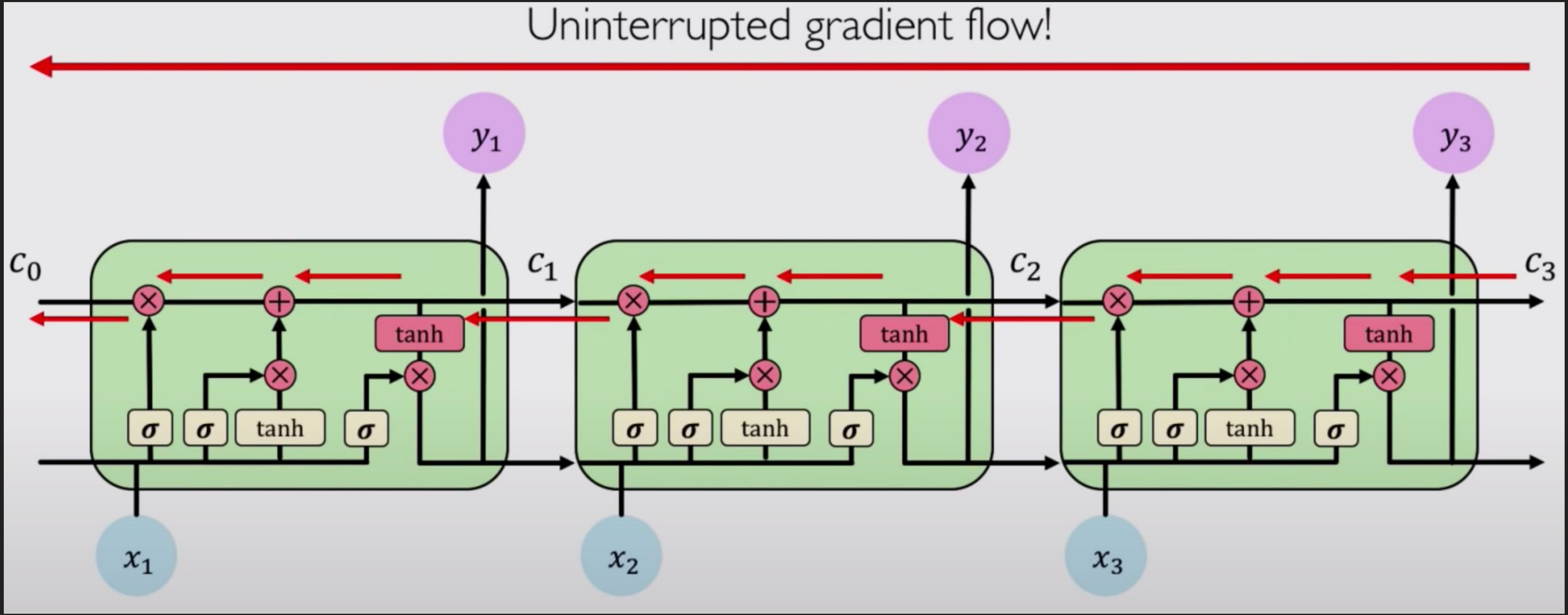
How LSTMs work?



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

Output gate controls what information to send to next time stamp

Gradient Flow



LSTMs: Summary

1. Use gates to control the flow of information

   (i) Forget

   (ii) Store

   (iii) Update

   (iv) Output

2. BPTT with uninterpreted gradient flow

Summary

- Why sequential data
- RNNs for sequence modeling
- Backpropagation through time
- LSTMs for long-term dependencies

References

https://d2l.ai/chapter_recurrent-neural-networks/

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Few slide contents are taken from : Ava Amini (MIT) and CS231n

Thank you!

https://anandmishra22.github.io/