Machine Learning
For Natural Language Processing

Abdelhak Mahmoudi
abdelhak.mahmoudi@um5.ac.ma

2020

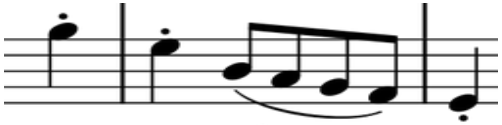
Content

1. Introduction
2. Machine Learning
   1. Supervised Learning, 2. Unsupervised Learning
3. Natural Language Pre-Processing
   1. Regular Expressions, 2. Tokenization, 3. Character Encoding, 4. Part-of-Speech Tagging, 5. Chunking, 6. Stemming and Lemmatization, 7. Parsing, 8.
4. Vector Representation of Text
   1. One hot vector, Word Embeddings, Tf-Idf, Word2Vec, CloVe
5. **Introduction to Deep Learning for NLP**
   1. **Sequence models, 2. BERT models**
6. NLP Applications
   1. Named Entity Recognition, 9. Topic Segmentation

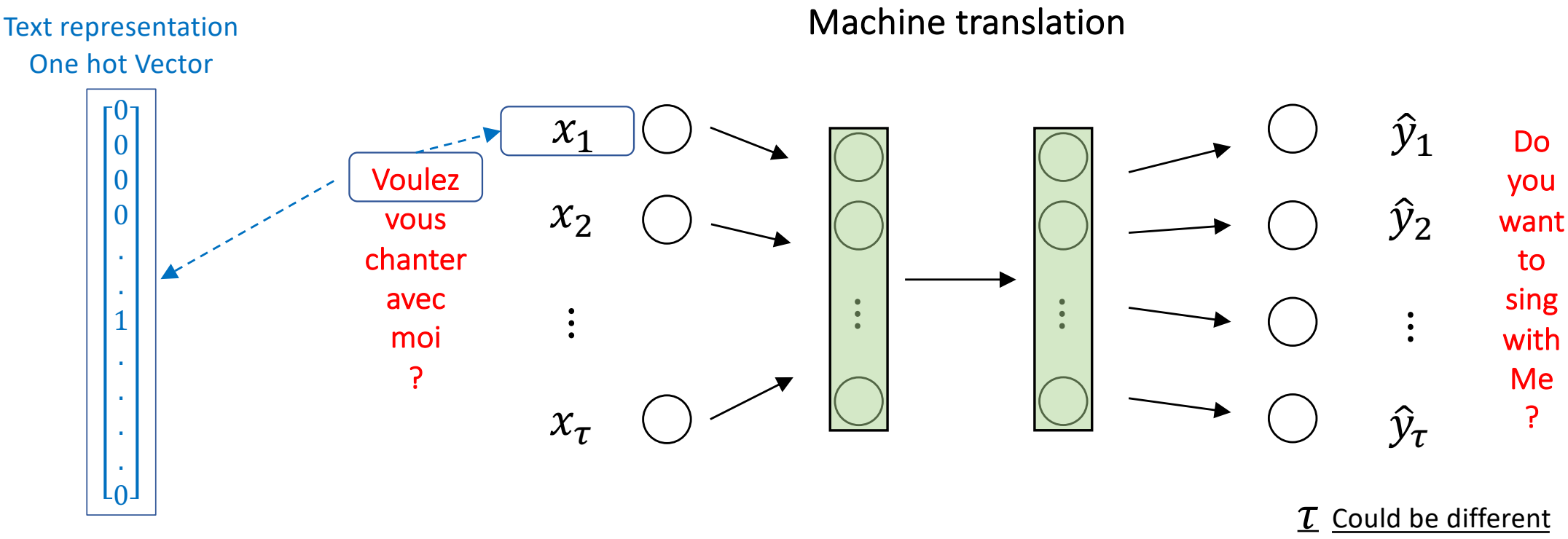
Sequence Models

- Applications
- Why not simple Deep NN ?
- Recurrent Neural Networks (RNN)
  - Architectures
  - Vanishing/Exploding gradients
- Long Short Term Memory Nets (LSTMs)
- Gated Recurrent Units (GRUs)
- Transformers

Applications

	Input x		Output y
Speech recognition		→	"The quick brown fox jumped over the lazy dog."
Music generation	∅	→	
Sentiment classification	"There is nothing to like in this movie."	→	
DNA sequence analysis	AGCCCCTGTGAGGAACTAG	→	AGCCCCTGTGAGGAACTAG
Machine translation	Voulez-vous chanter avec moi?	→	Do you want to sing with me?
Video activity recognition		→	Running
Name entity recognition	Yesterday, Harry Potter met Hermione Granger.	→	Yesterday, Harry Potter met Hermione Granger.

Why not simple Deep NN ?



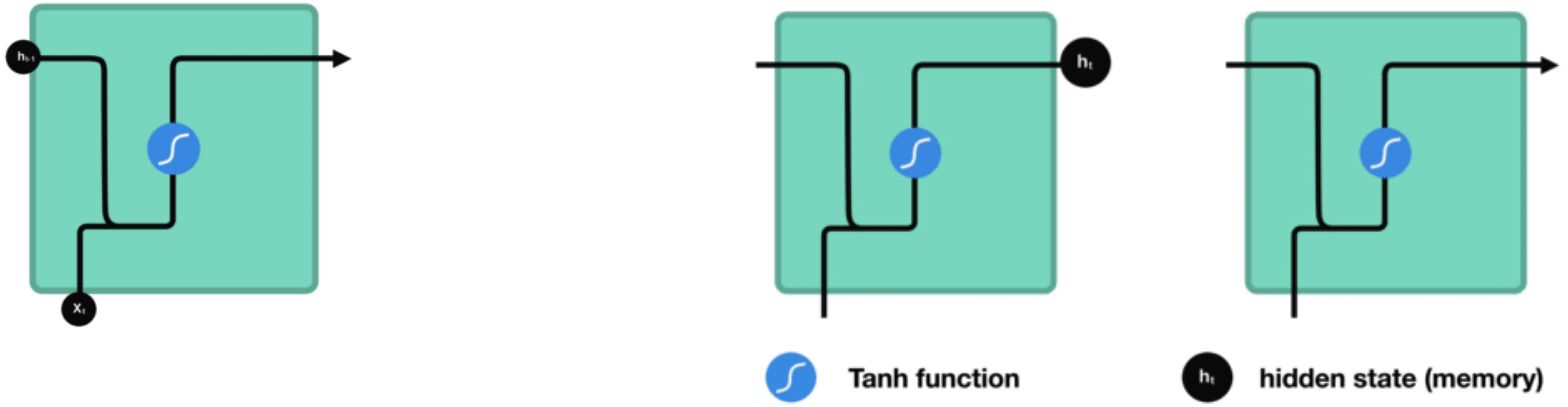
Why not simple Deep NN ?

- Process sequences of variable length
  - Use a fixed window?
  - What about dependencies?
- Handle long-term dependencies
  - Still not taking into account the order in the sequence !
- Maintain order's information
  - How?
- Sharing parameters

"I'm Moroccan, I speak fluent……."

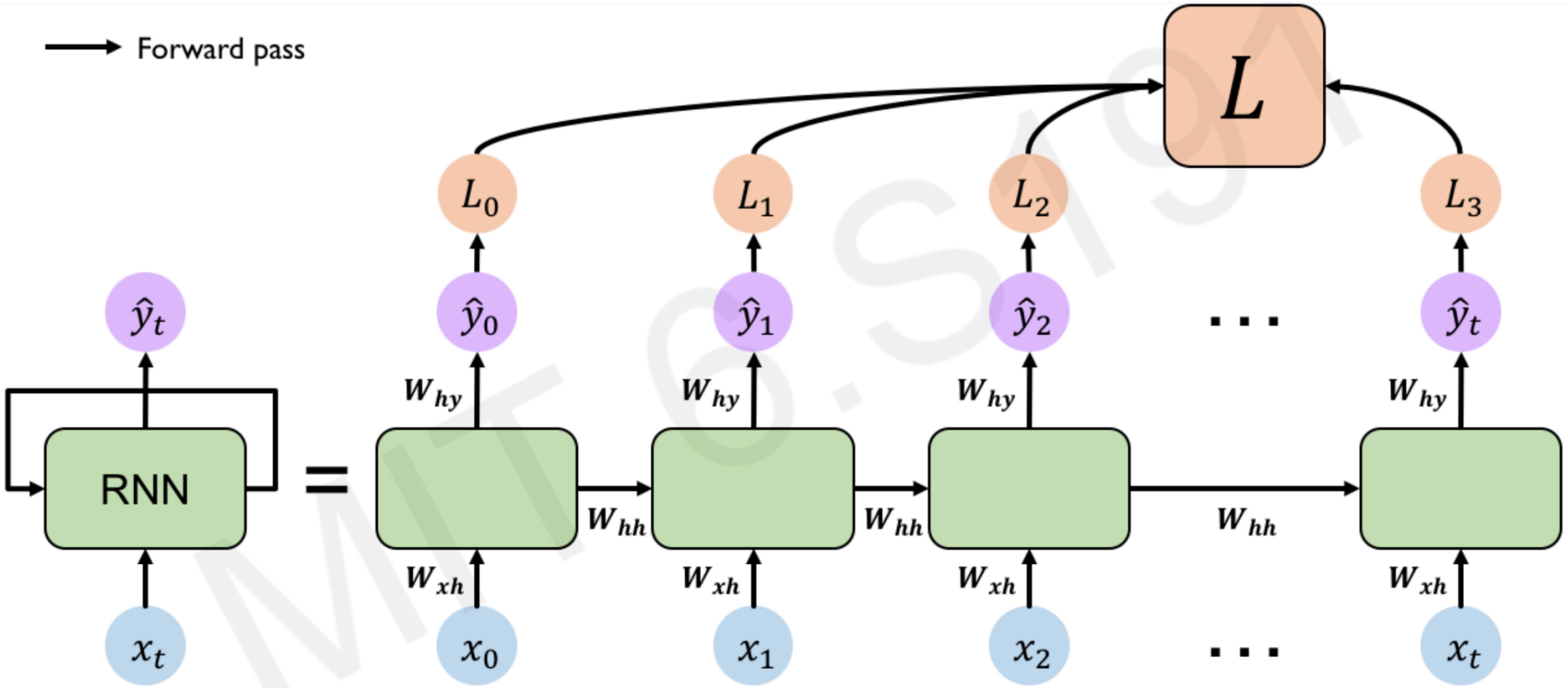
"All people love Moroccan Couscous."

RNN intuition

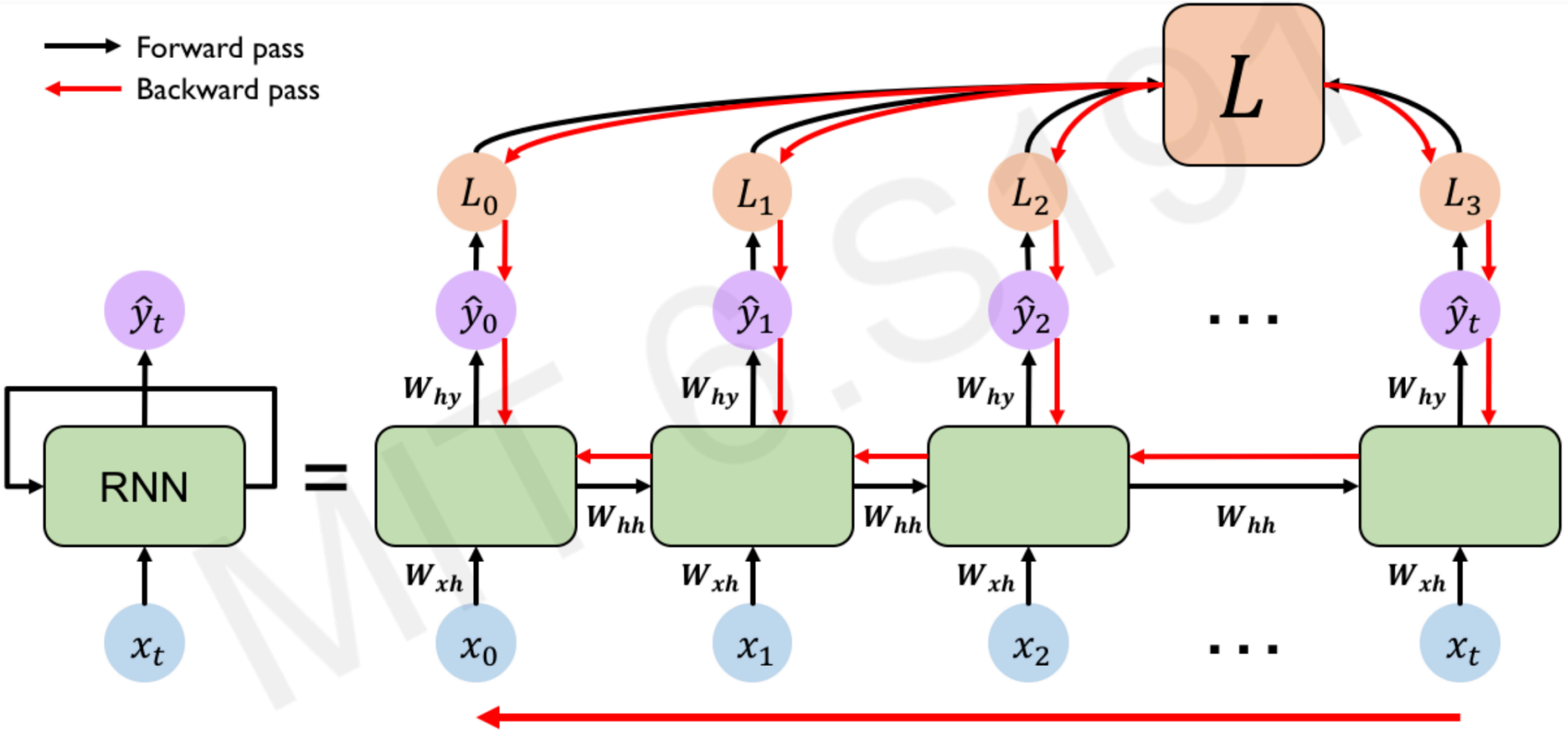


https://towardsdatascience.com/@learnedvector

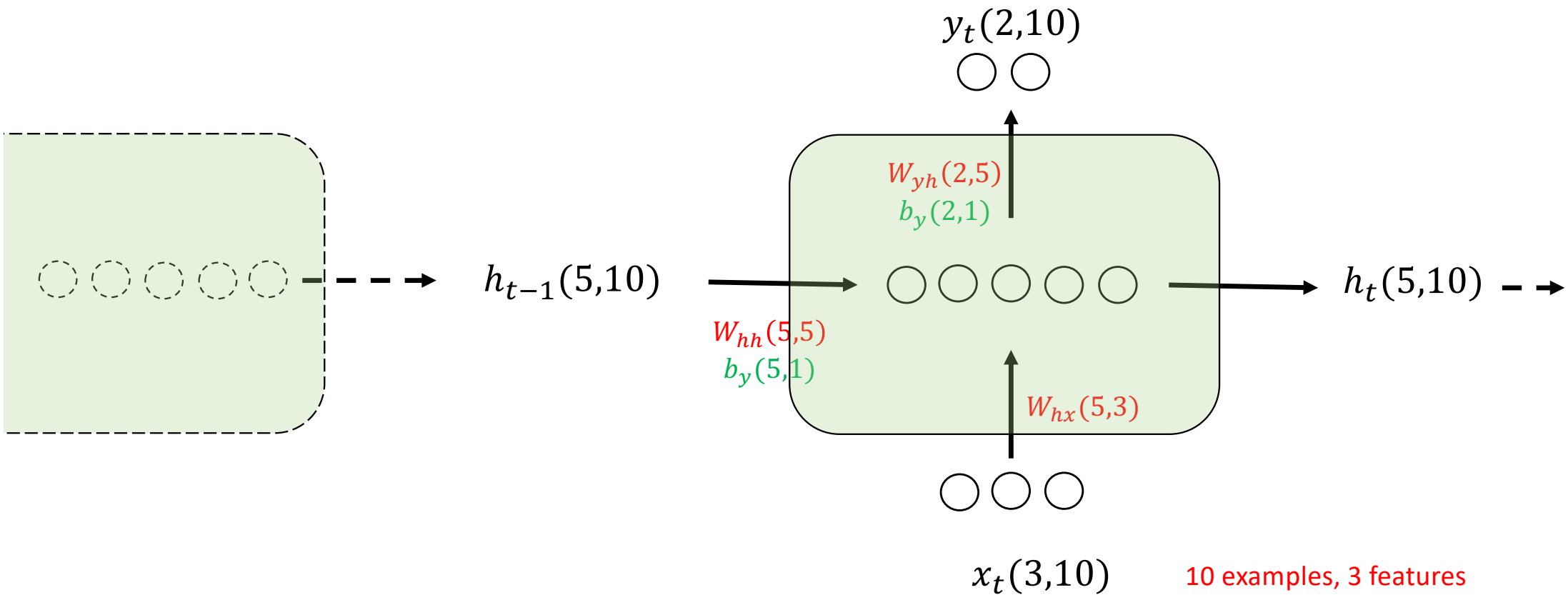
RNN Architecture



RNN : Backpropagation through time

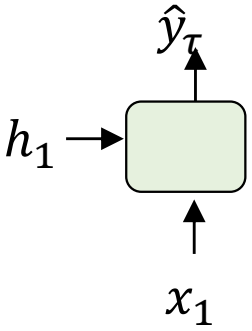


RNN - Dimensions

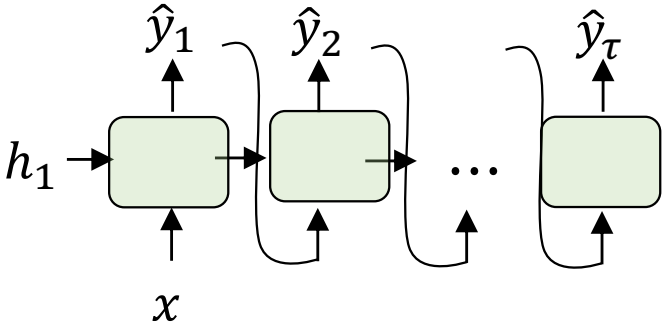


RNN – Different Architectures

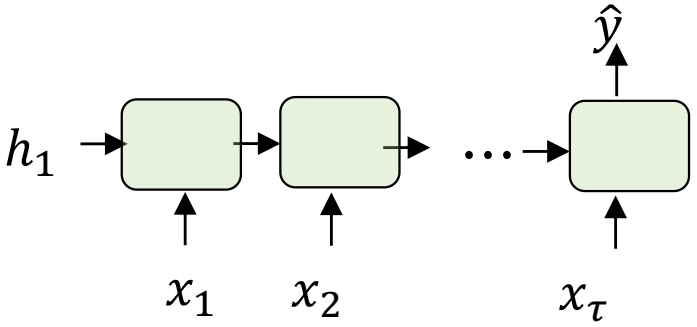
One to one

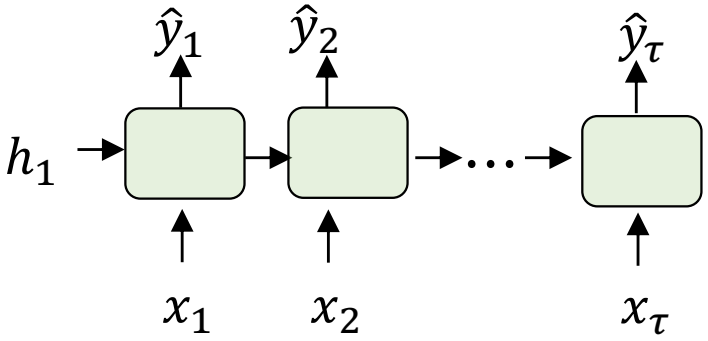


One to many
(Music/ Text Generation)

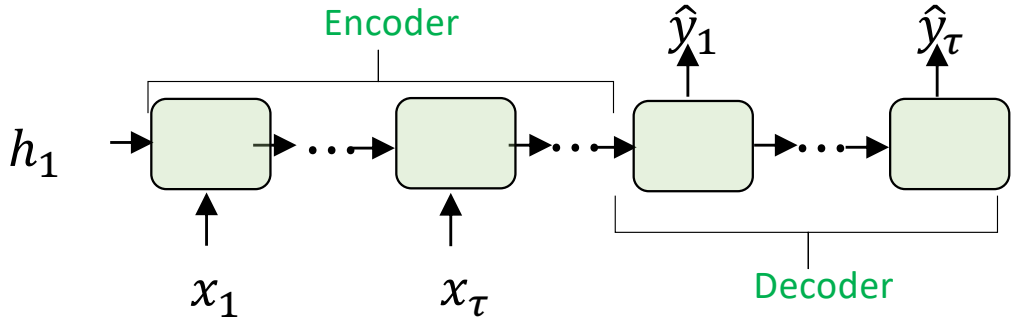


Many to one
(Sentiment)





Many to many

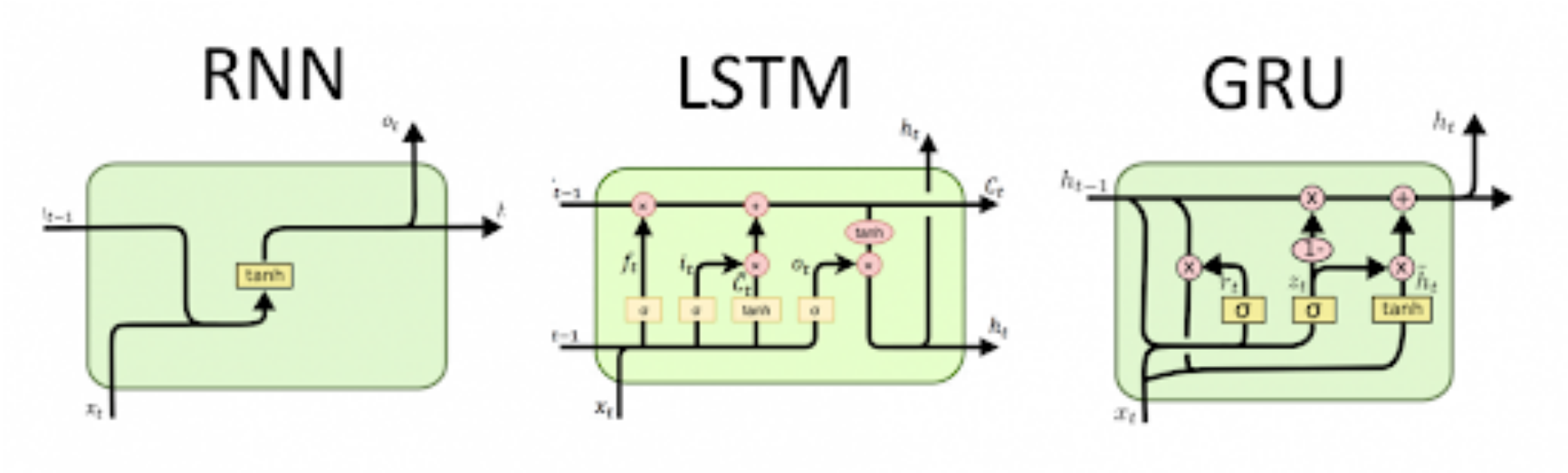


Many to many
(Translation)

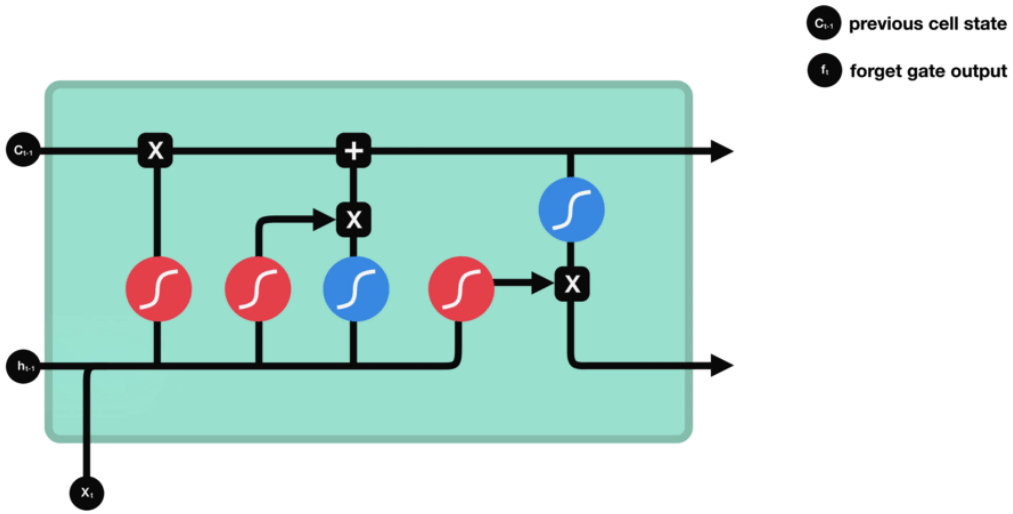
Vanishing/Exploding gradients

- $W^{[l]} =: W^{[l]} - \alpha \frac{\partial L}{\partial W^{[l]}}$
- $W^{[l]}$ <1 → $\frac{\partial L}{\partial W^{[l]}}$ <1→ Vanishing → slow down training
- $W^{[l]}$ >1 → $\frac{\partial L}{\partial W^{[l]}}$ >1→ Exploding → divergence
- Solution
  - Batch normalization
  - Random Weights Initialization
  - Use Gradient Clipping
  - **Use gated cells GRU, LSTM**

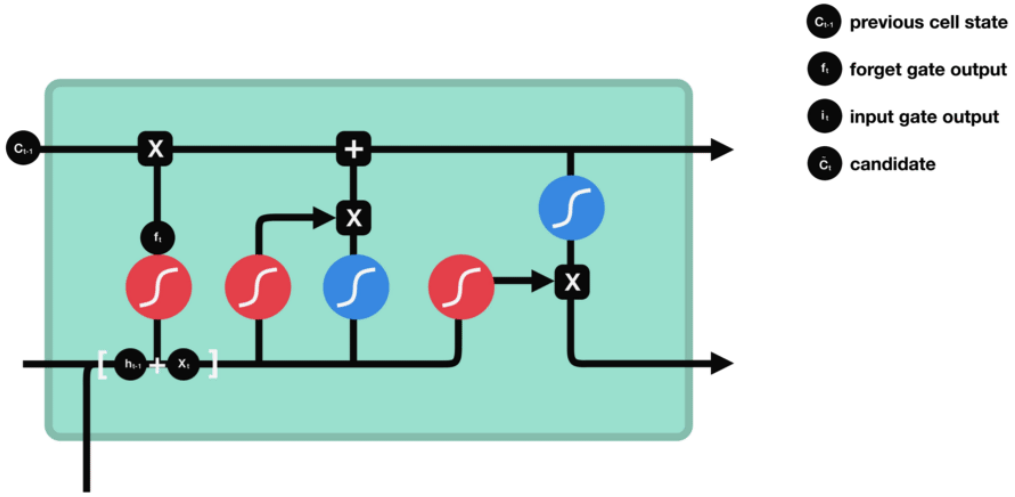
RNN, LSTM, GRU



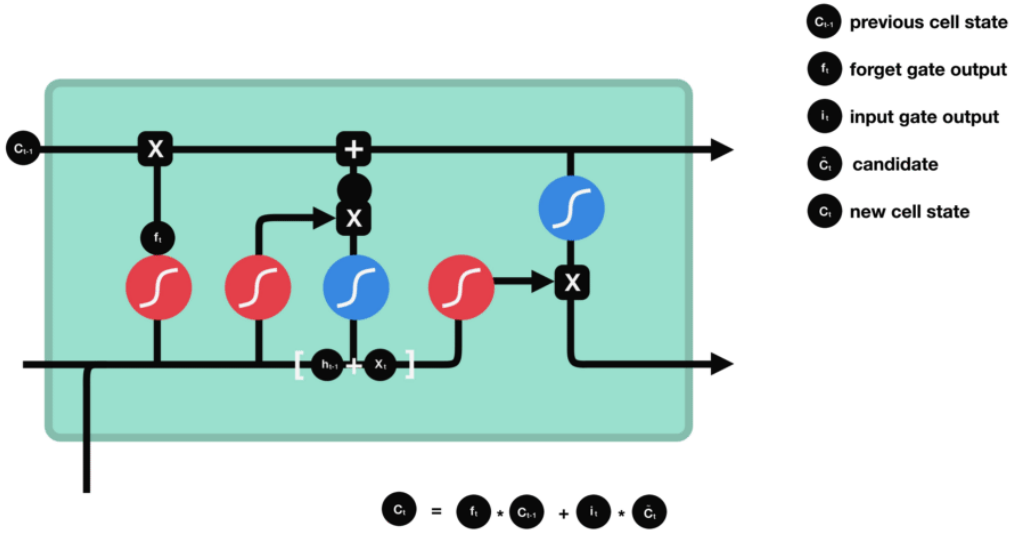
Long Short Term Memory (LSTM)



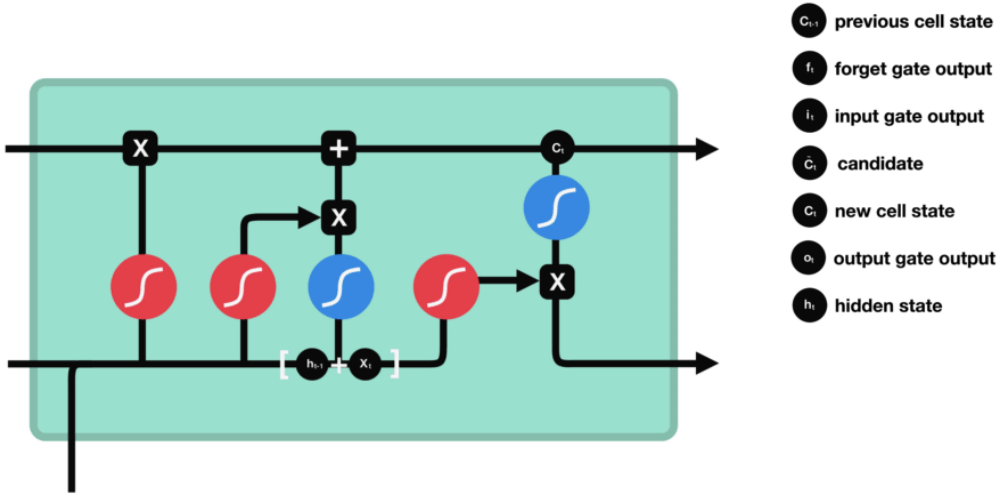
Long Short Term Memory (LSTM)



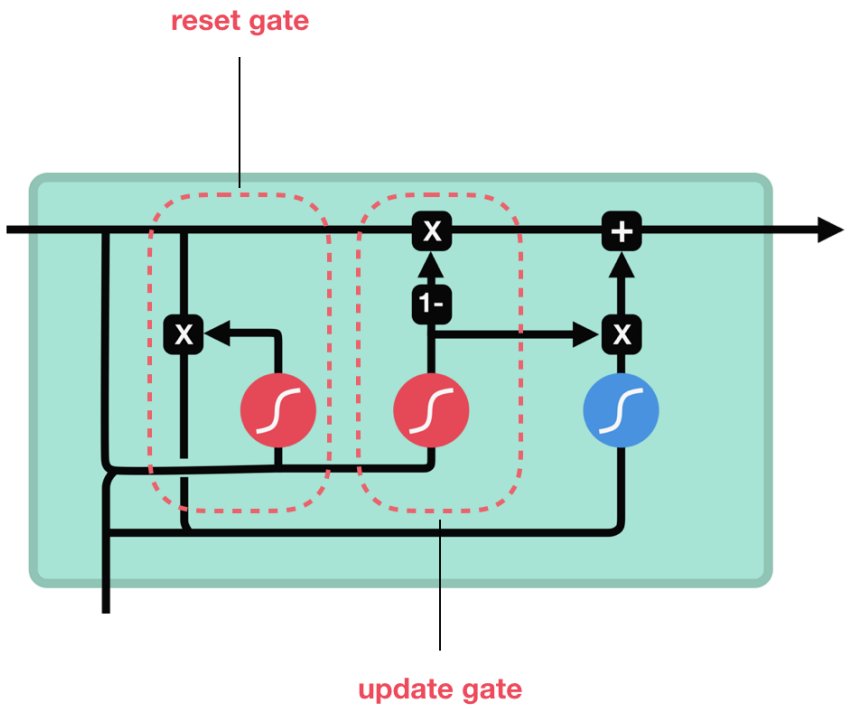
Long Short Term Memory (LSTM)



Long Short Term Memory (LSTM)

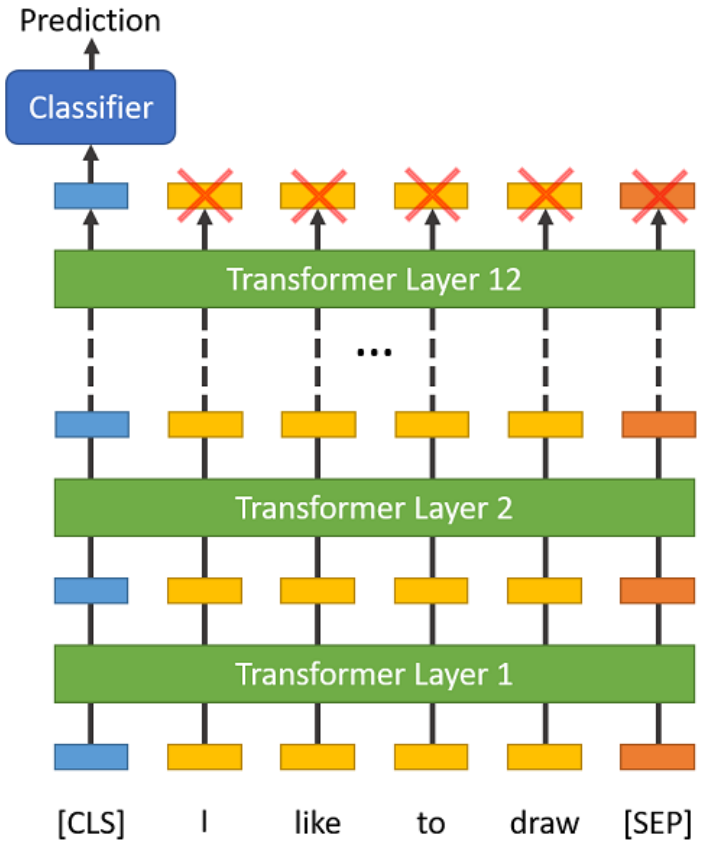


Gated Recurrent Units



Seq2Seq for Natural Language

- RNN
- LSTM
- Word2Vec
- GRU
- Attention
- Bidirectional LSTM
- Transformer (Attention is all what you need!)



Transformers

- BERT (Bidirectional Encoder Representations from Transformers)
  - A Lite BERT (ALBERT)
- Generative Pre-Training (GPT)
  - GPT2
- ELMo (Embeddings from Language Models)