AI in Built Environment
DCP4300

Week 11: Natural Language Processing

Part B: Language Models

Dr. Chaofeng Wang
Jianhao Gao (TA)

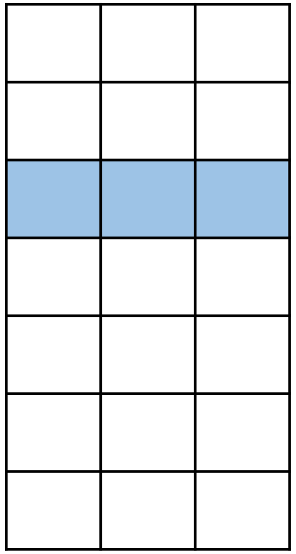
University of Florida
College of Design Construction and Planning

Word Embeddings

Words are represented using dense continuous vectors

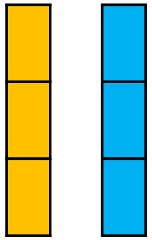
The embedding vectors can be learned using neural networks (supervised or unsupervised?)

The embeddings vectors can be used in other NLP tasks for improved performance
(much better than one-hot encoding)



E

Language modeling



I would like to have an apple for breakfast.

$$P(t|c) = \frac{\exp(e_t' \cdot e_c)}{\sum_{i=1}^{n} \exp(e_i' \cdot e_c)}$$

It’s a probability.

Context -> Target

Language modeling

N-gram

I would like to have $\boxed{\text{an apple for}}$ breakfast.

$$P(\text{breakfast} \mid \text{an apple for}) = \frac{Count(\text{an apple for breakfast})}{Count(\text{an apple for})}$$

Language modeling

N-gram

I would like to have an apple for

breakfast 0.82
lunch 0.03
supper 0.05
…

Language modeling

N-gram

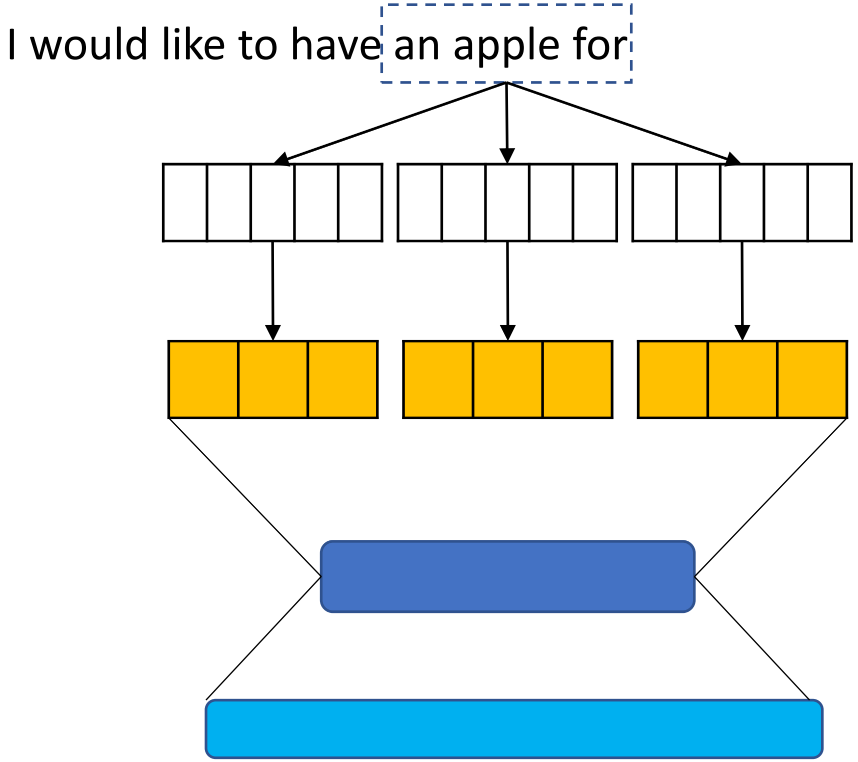
I would like to have $\boxed{\text{an apple for}}$ breakfast.

$$P(\text{breakfast} \mid \text{an apple for}) = \frac{Count(\text{an apple for breakfast})}{Count(\text{an apple for})}$$

Issue:
Sparsity of data in a corpus makes the likelihood estimation not accurate

Language modeling

Neural networks



a fixed window neural network

Sparsity problem mitigated, but

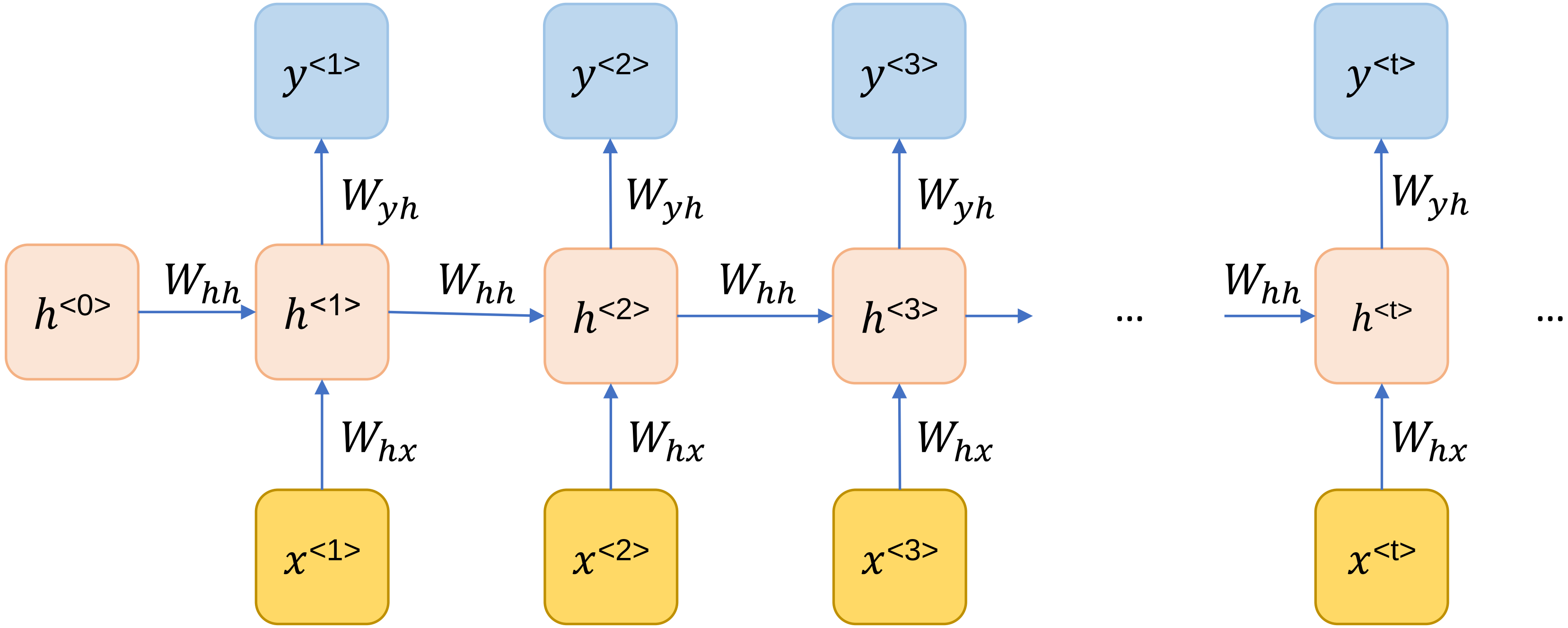
Fixed window is small

Enlarging window increases size of model

Each word vector has its own weights

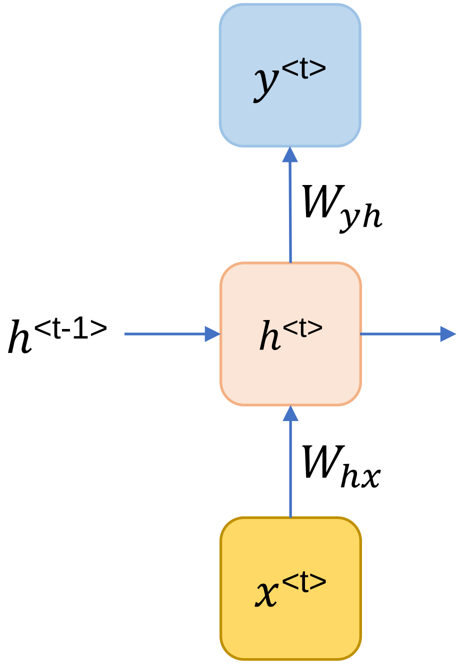
Recurrent Neural Network (RNN)

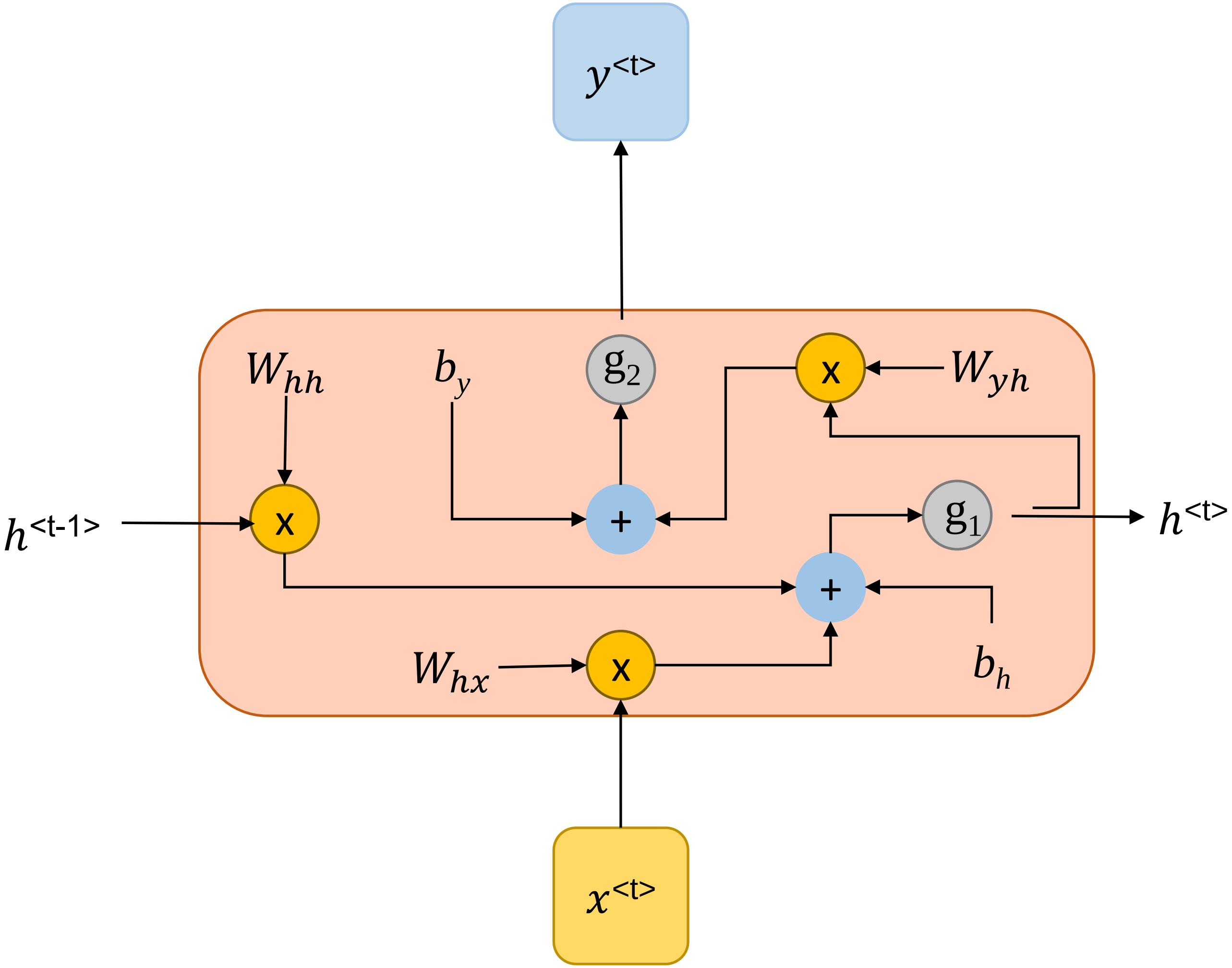
The same weight is shared across all units in the same layer.
The operation is repeated along the sequence.



Recurrent Neural Network (RNN)

What's inside a RNN cell





Recap

Language model: a system that can predict or generate words based on context

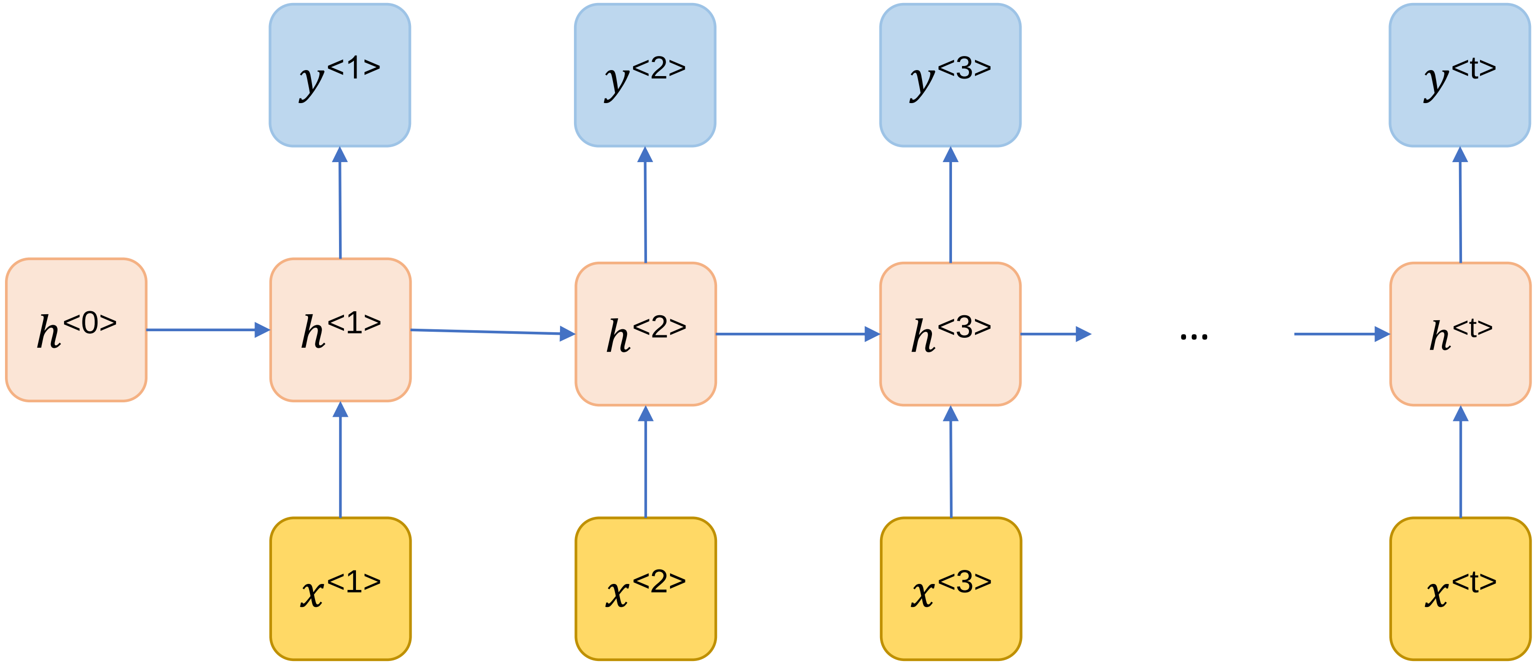
Recurrent Neural Network:

- Input can be a sequence of words of any length
- The same weights is applied to all words in the sequence
- Can output on selected step

Recurrent Neural Network (RNN)

Types and applications

Many to many

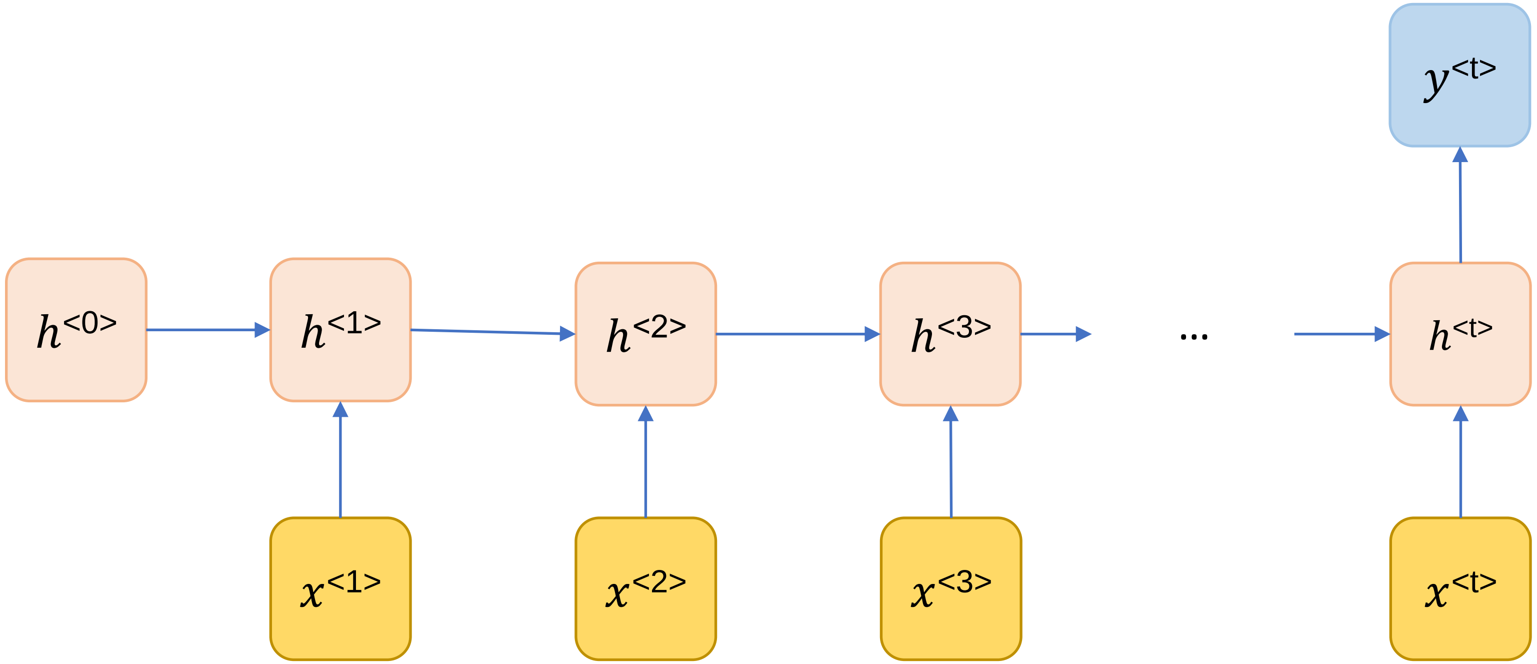


Can be used for: sequence to sequence taks (e.g., machine translation …)

Recurrent Neural Network (RNN)

Types and applications

Many to one

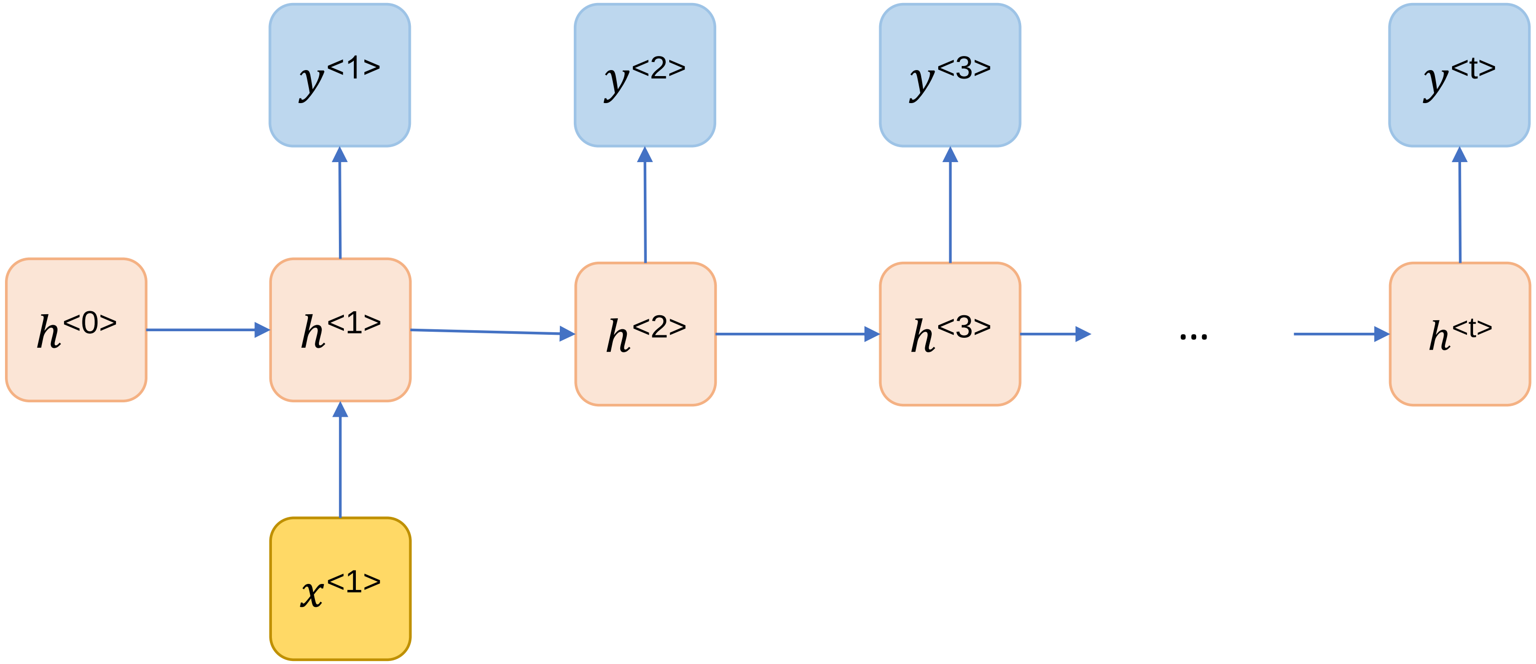


Can be used for: Classification (e.g., sentiment analysis)

Recurrent Neural Network (RNN)

Types and applications

One to many

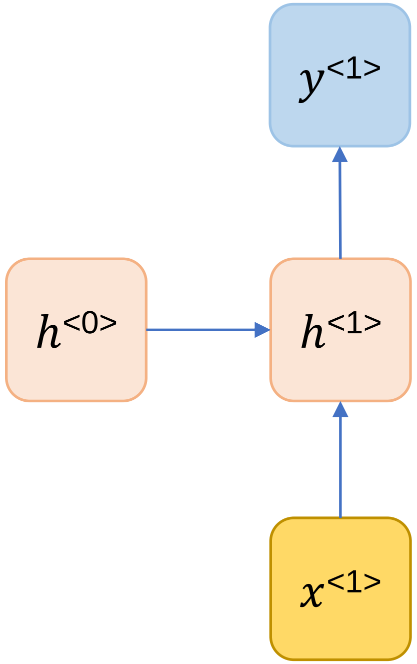


Can be used for: Generation (e.g., image captioning)

Recurrent Neural Network (RNN)

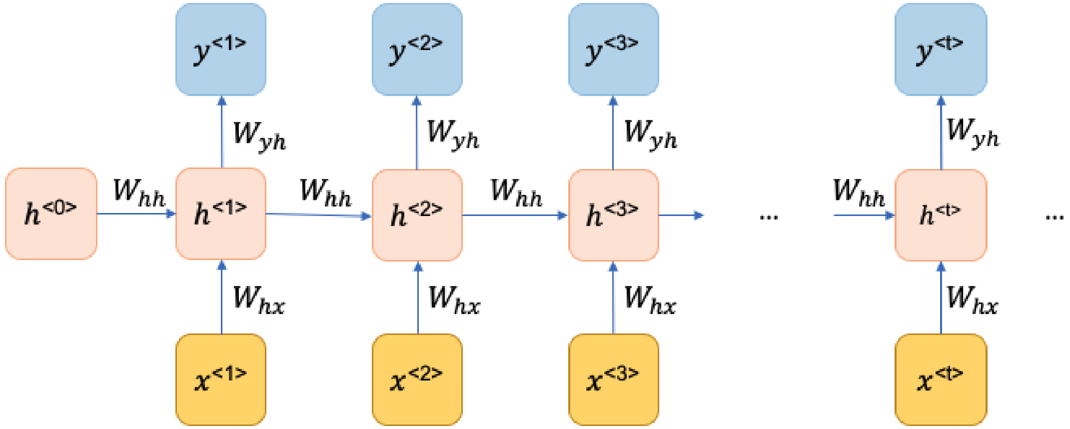
Types and applications

One to one



It’s a traditional neural network.

Recurrent Neural Network (RNN)



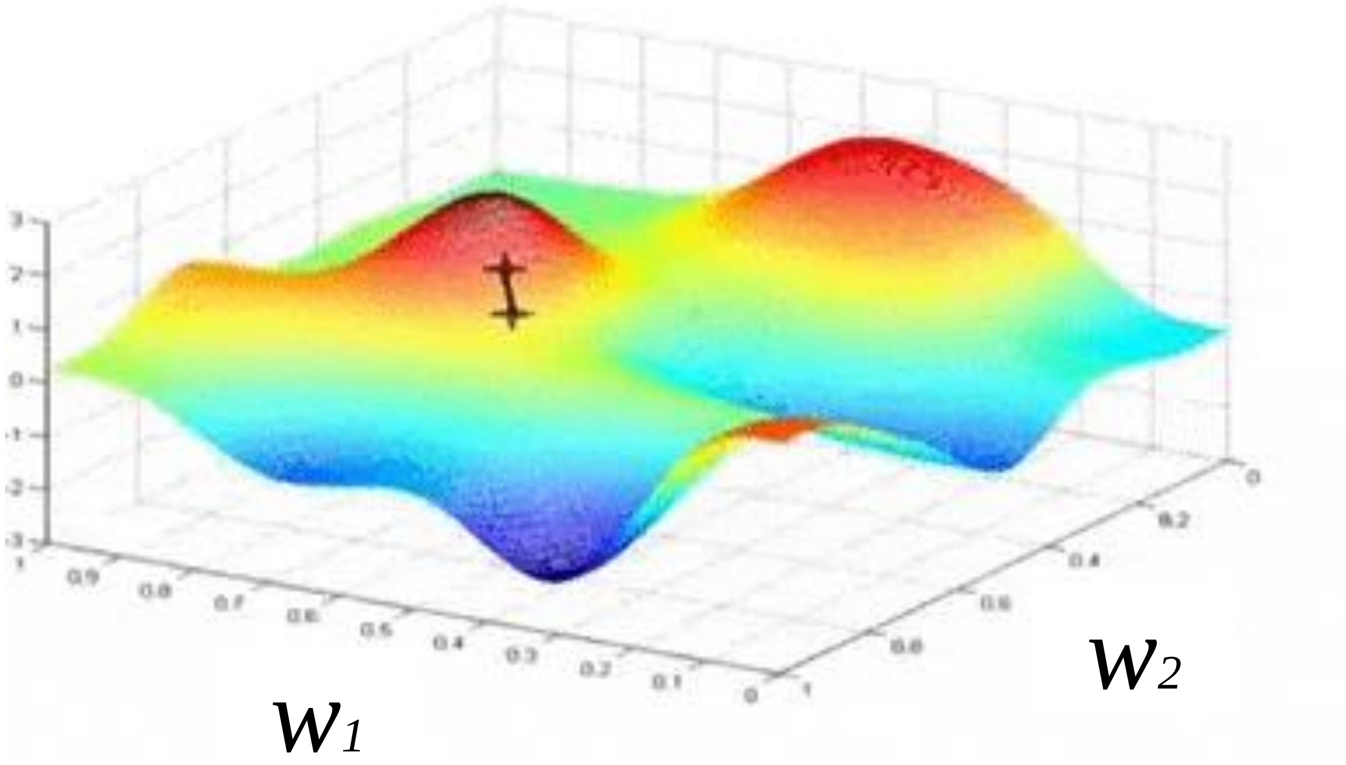
Pros	Cons
• Can process input of any length • Model size (weights size) not increasing with input size, because weights are shared and used repeatedly • Historical information is counted in	• Computation is slow • Difficult to count in long time history

Evaluate the performance of a language model

Perplexity = $P(w_1\ w_2\ \dots\ w_n)^{-1/n} = \sqrt[n]{\prod_{\mathrm{i=1}}^{\mathrm{n}} \frac{1}{P(w_i|w_1 w_2 \dots w_{i-1})}}$

A lower perplexity indicates a better predictive performance, meaning the model is less "perplexed" by the sequence of words

Exploding gradient and Vanishing gradient



Gradient descent:

1. Compute the slope (gradient) at the current step $\frac{\partial J}{\partial w}$

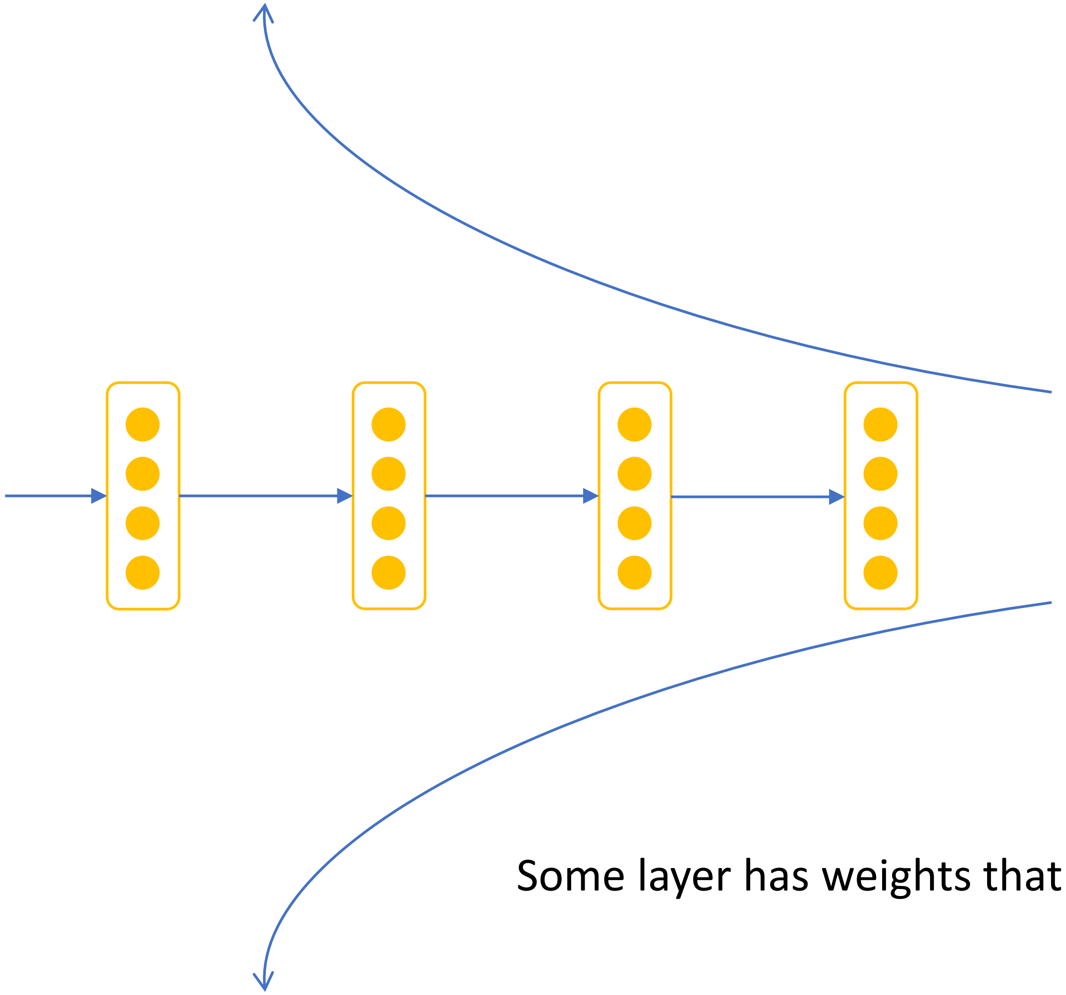
2. Make a move in the direction opposite to the slope

The move = $-\eta \frac{\partial J}{\partial w}$

Learning rate

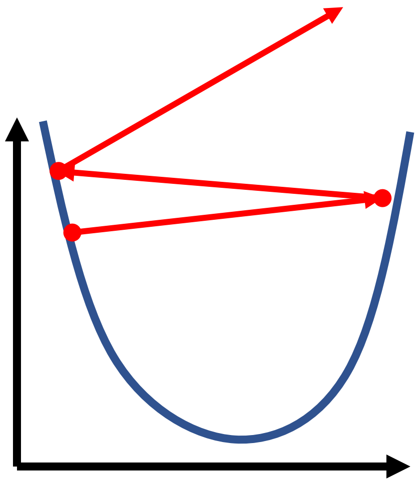
Gradient exploding in deep neural net

In backpropagation, the error signal increases exponentially with the deep of the neural nets



Some layer has weights that cause the gradients to be large

Gradient exploding



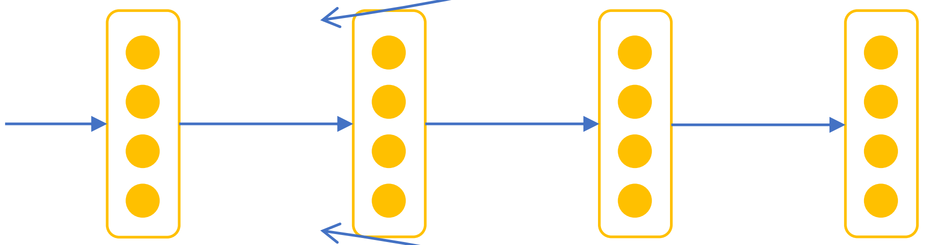
Overshoot

Network not stable
Can’t learn
NaN in weights

Vanishing gradient in deep neural net

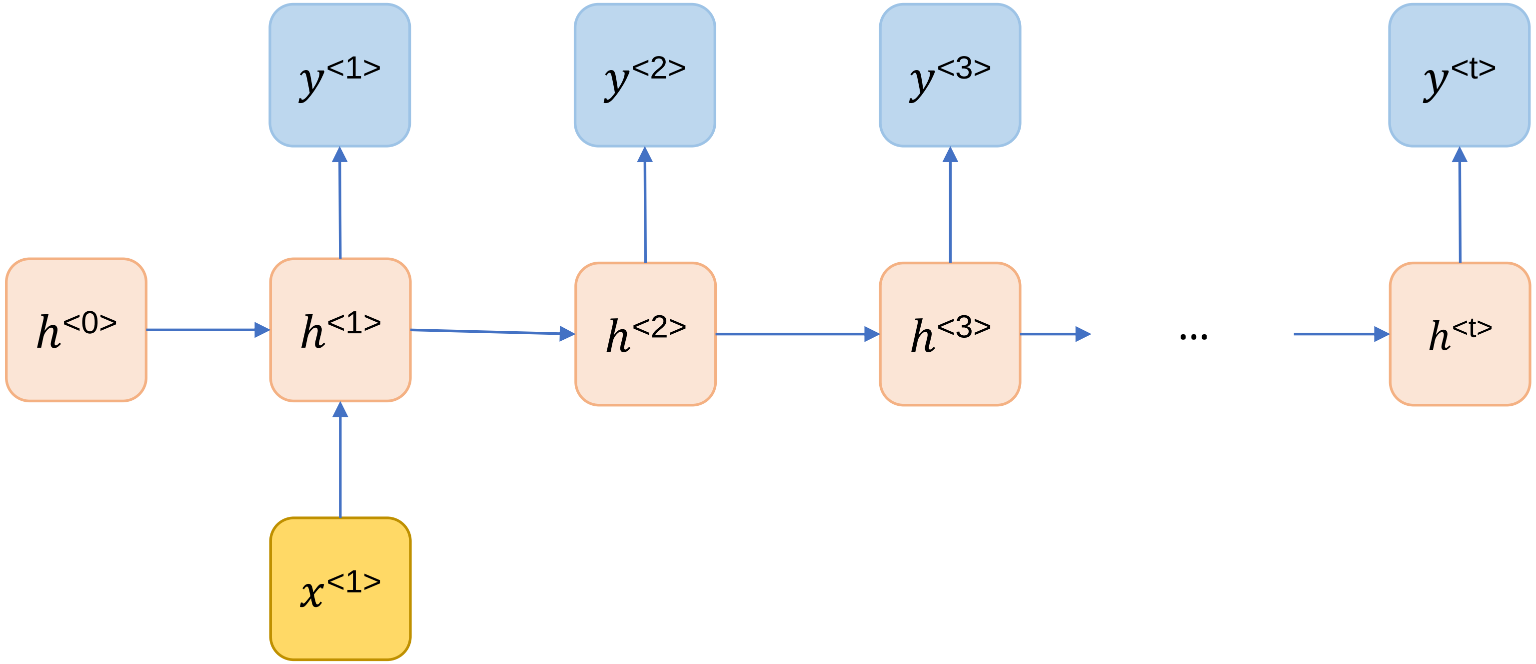
In backpropagation, the error signal declines exponentially with the deep of the neural nets

the earlier layers of the network hardly change



Derivatives are small, multiplying many of these small numbers
can lead to an exponentially smaller gradient as we move
backward through the layers

Vanishing gradient in RNN



Far away gradient is much smaller than the close-by -> so weights are more likely to be updated by nearest words.

Solutions:

Gradient clipping: if the gradients exceed this threshold during backpropagation, they are clipped or scaled down to be within a defined range.

Weight regularization: e.g. L2 regularization, which adds the squared values of the weights, to the loss function.

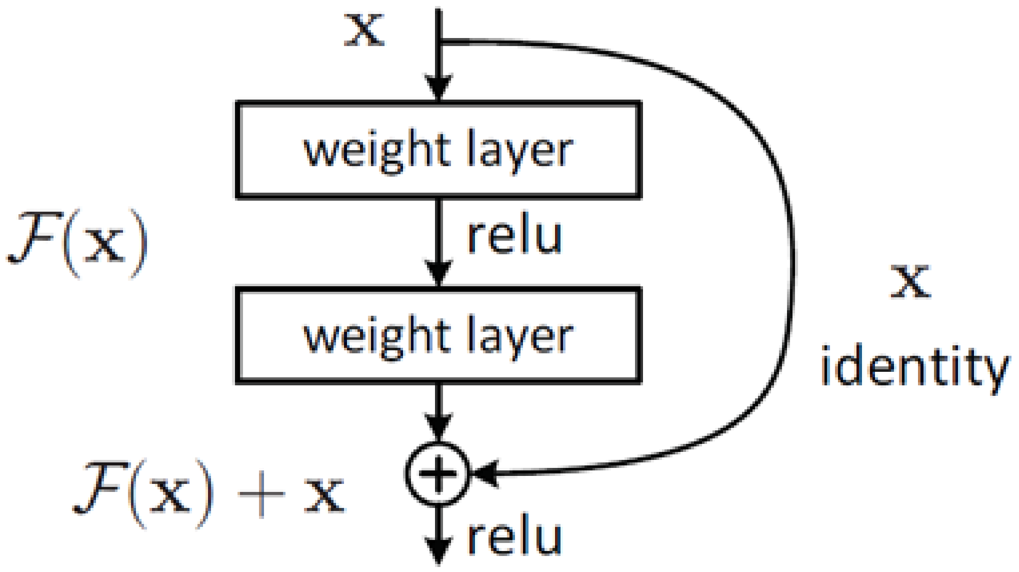
Use ReLU for activation: gradient to be 1 or 0.

Add residual: skip connections.

More advanced algorithms such as LSTM

Residual

help mitigate the vanishing and exploding gradient problems by providing an alternate shortcut path for the gradient to flow through



ResNet

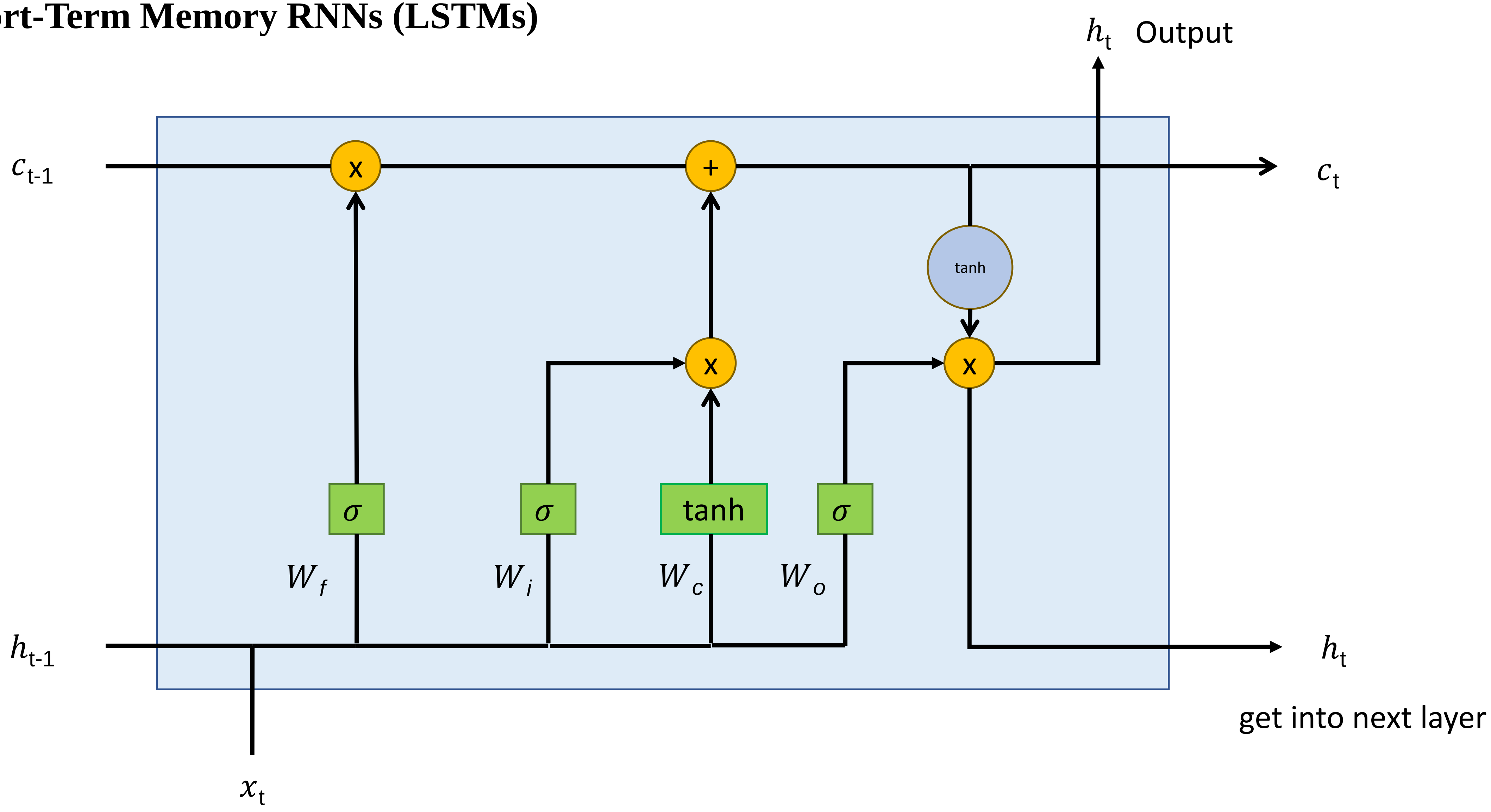
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778). https://arxiv.org/abs/1512.03385

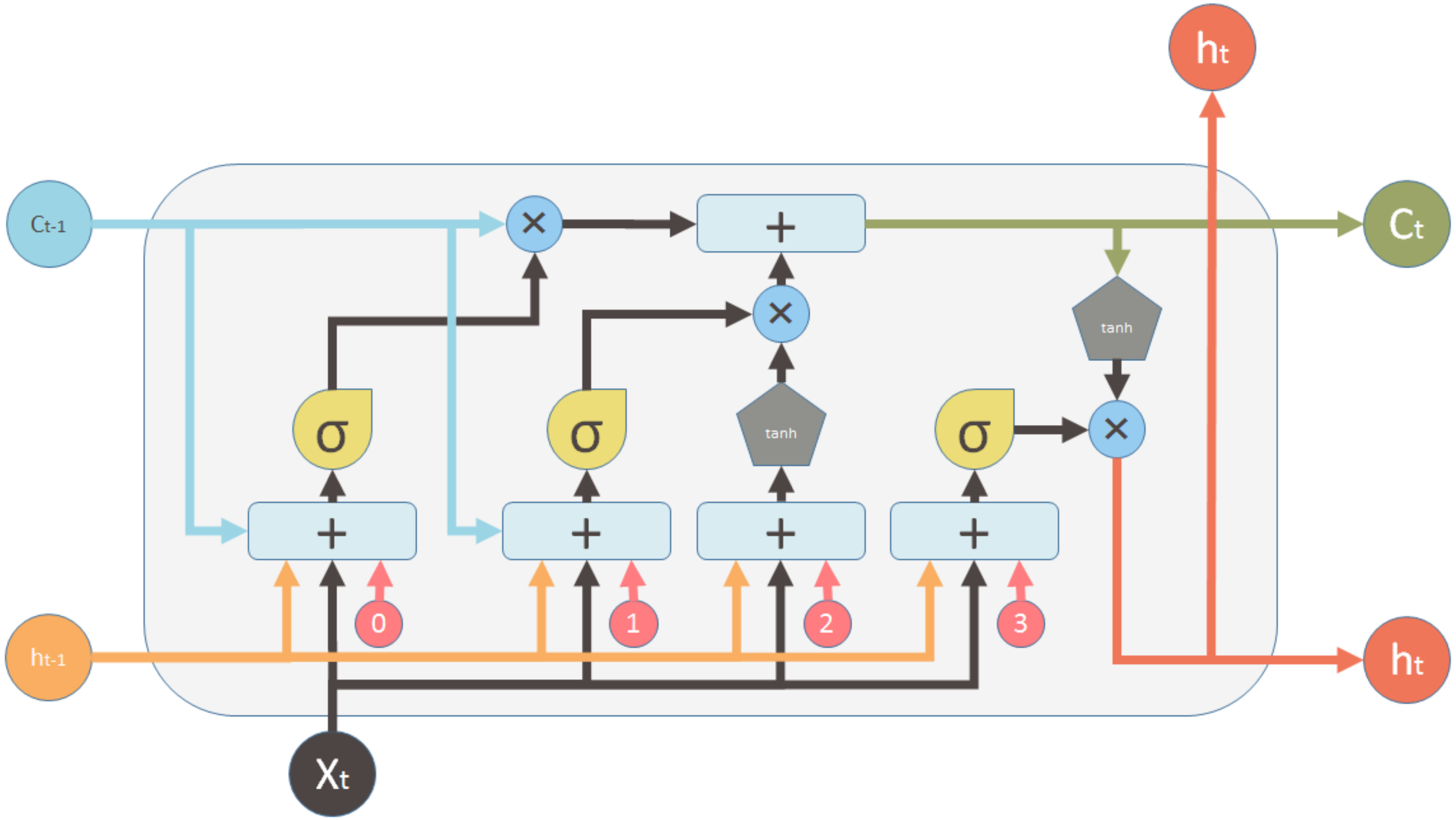
Long Short-Term Memory RNNs (LSTMs)

In a LSTM cell, there's a hidden state $h^{<t>}$ and a cell state $c^{<t>}$
A state is a vector. $h^{<t>}$ *and* $c^{<t>}$ *have the same size*
The LSTM cell can store long-term history
The LSTM cell has gates that can be used to read, erase and write information from cell
Gates are also vectors with the same size, their values are calculated based on the context.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780. https://www.bioinf.jku.at/publications/older/2604.pdf
Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural computation, 12(10), 2451-2471. https://dl.acm.org/doi/10.1162/089976600300015015

Long Short-Term Memory RNNs (LSTMs)





Inputs:

- X_t Input vector
- C_{t-1} Memory from previous block
- h_{t-1} Output of previous block

outputs:

- C_t Memory from current block
- h_t Output of current block

Nonlinearities:

- σ Sigmoid
- tanh Hyperbolic tangent

Vector operations:

- × Element-wise multiplication
- + Element-wise Summation / Concatenation

Bias: 0

https://blog.mlreview.com/understanding-lstm-and-its-diagrams-37e2f46f1714

Long Short-Term Memory RNNs (LSTMs)

Vanishing/exploding might still happen in LSTM

But LSTM can learn long-distance dependencies