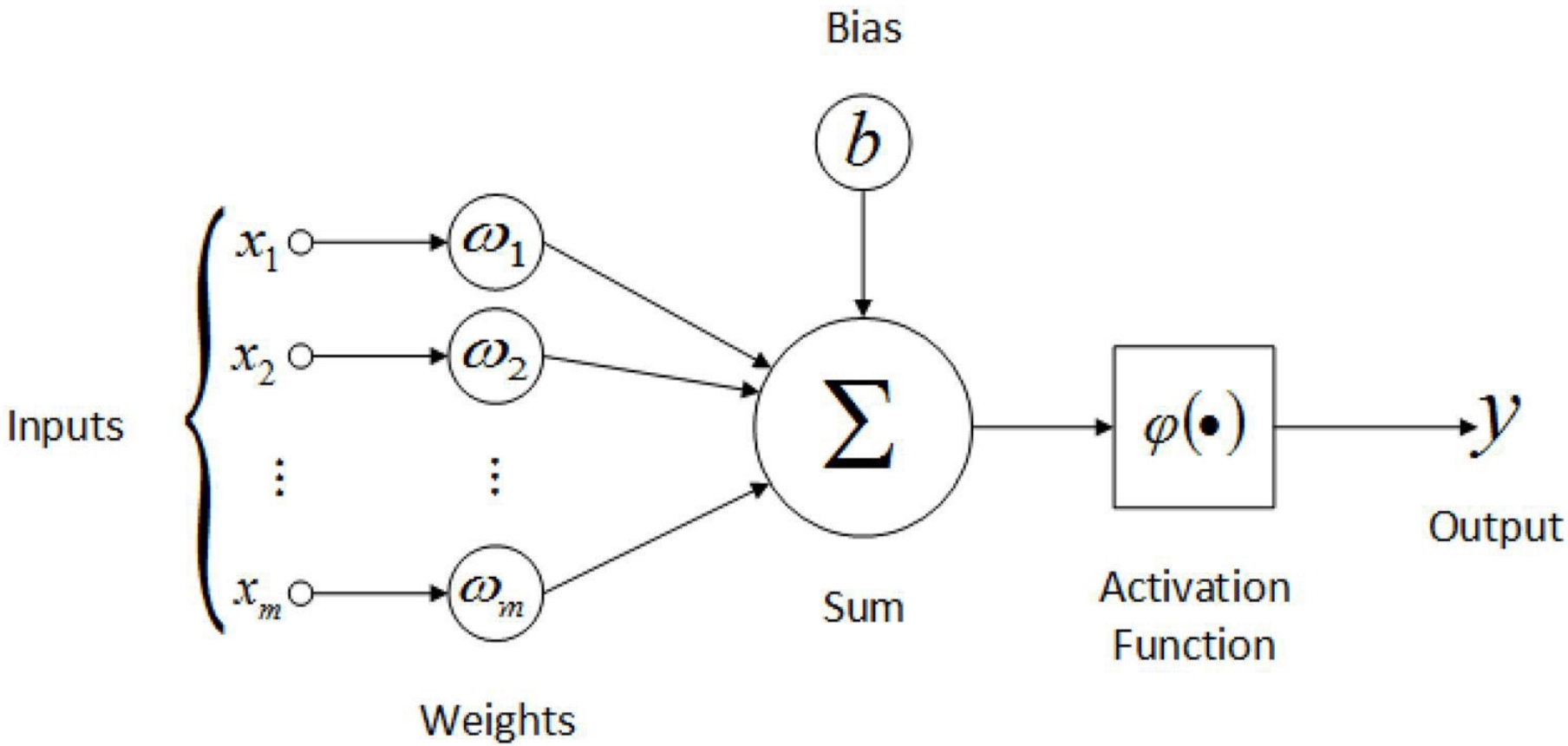
# Recurrent Neural Networks

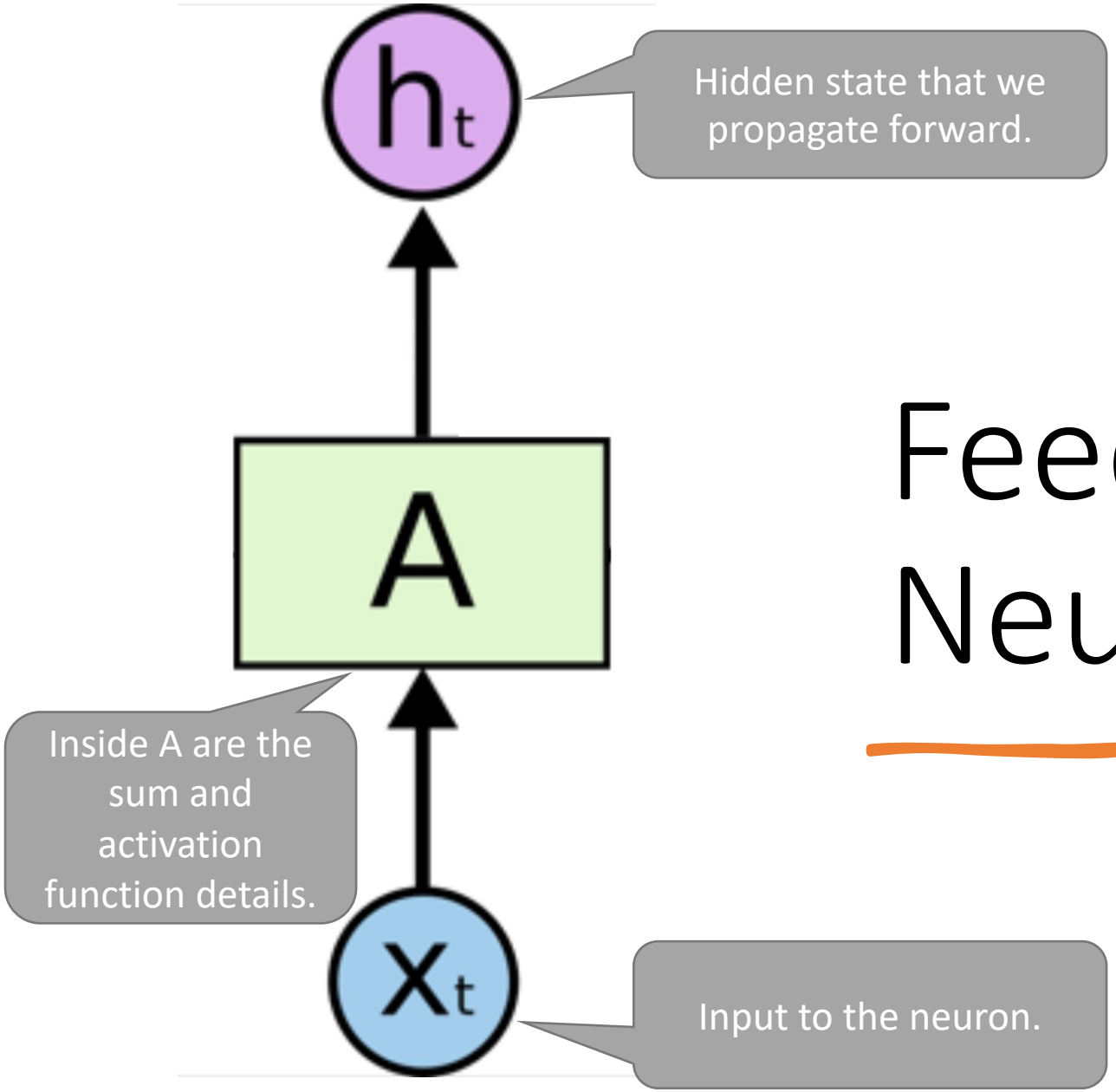
Christan Grant

oudatalab.com

AMLI 2022

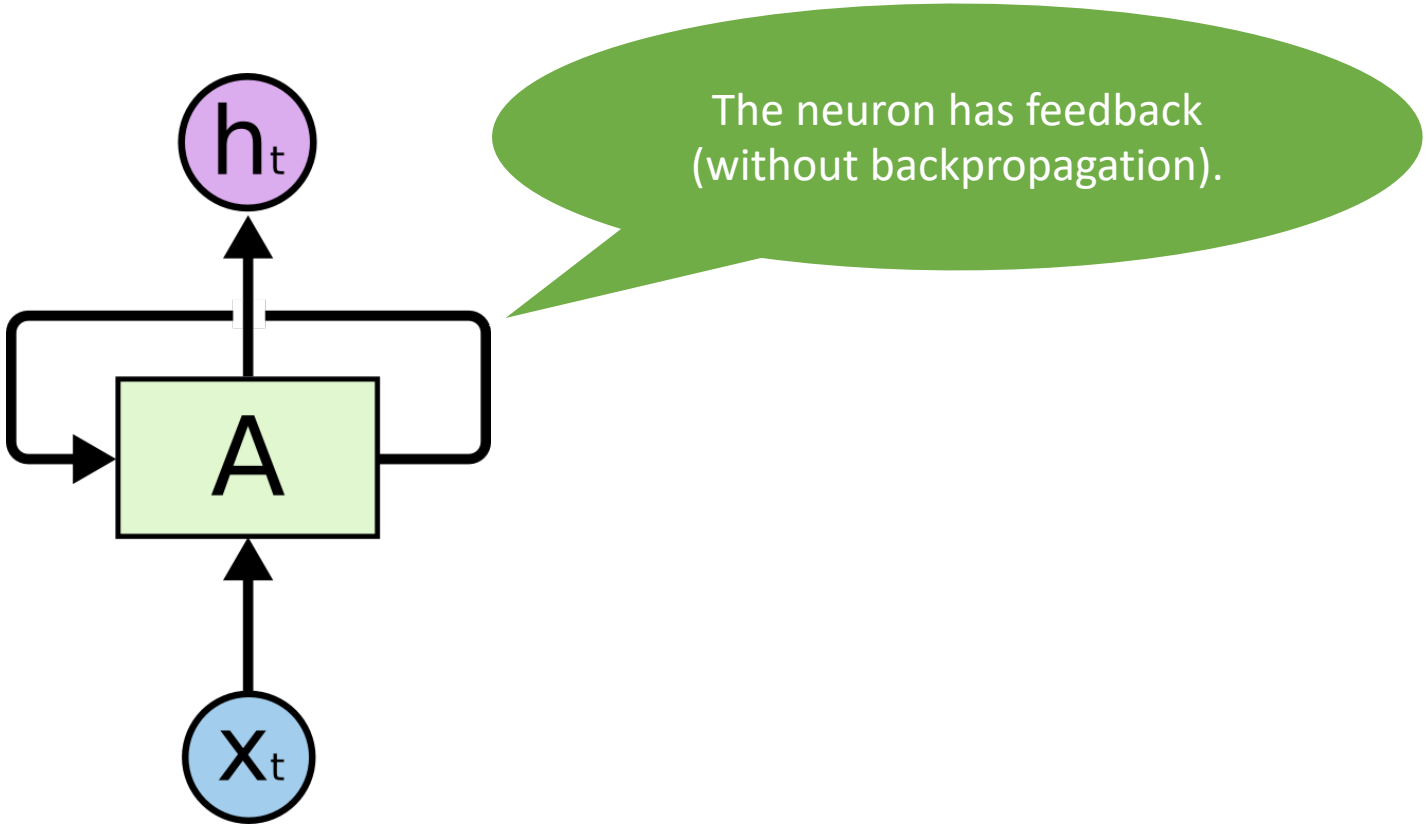
# Feed forward Neuron





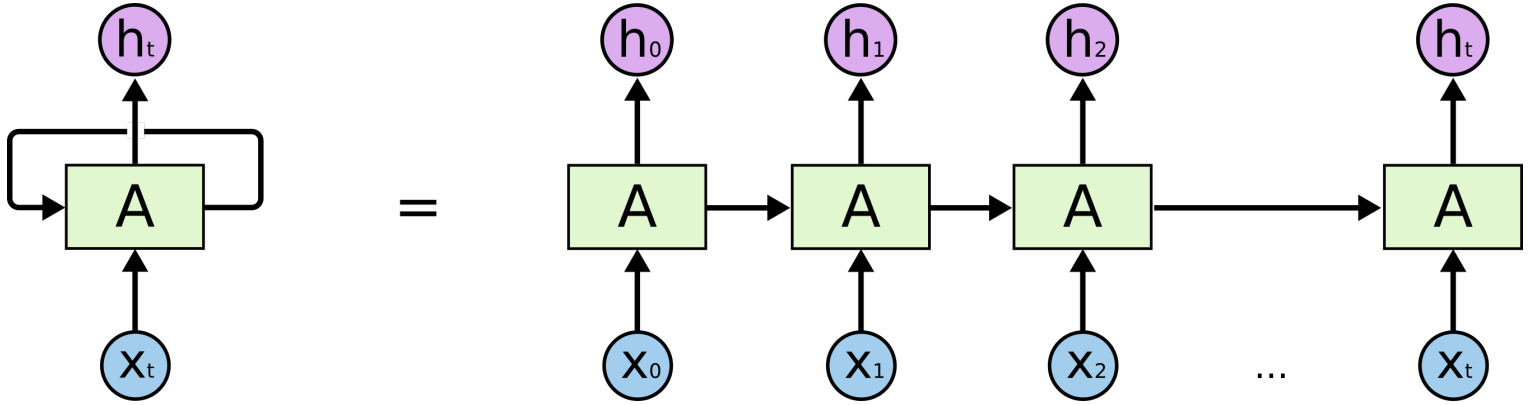

# Feedforward Neuron

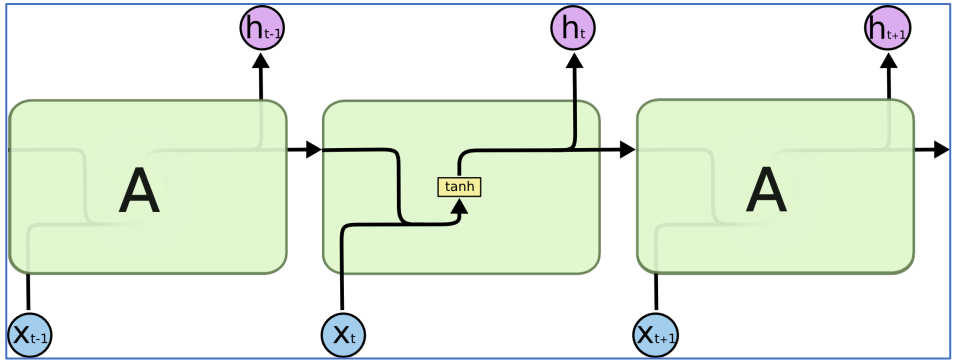
# Recurrent Neuron




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

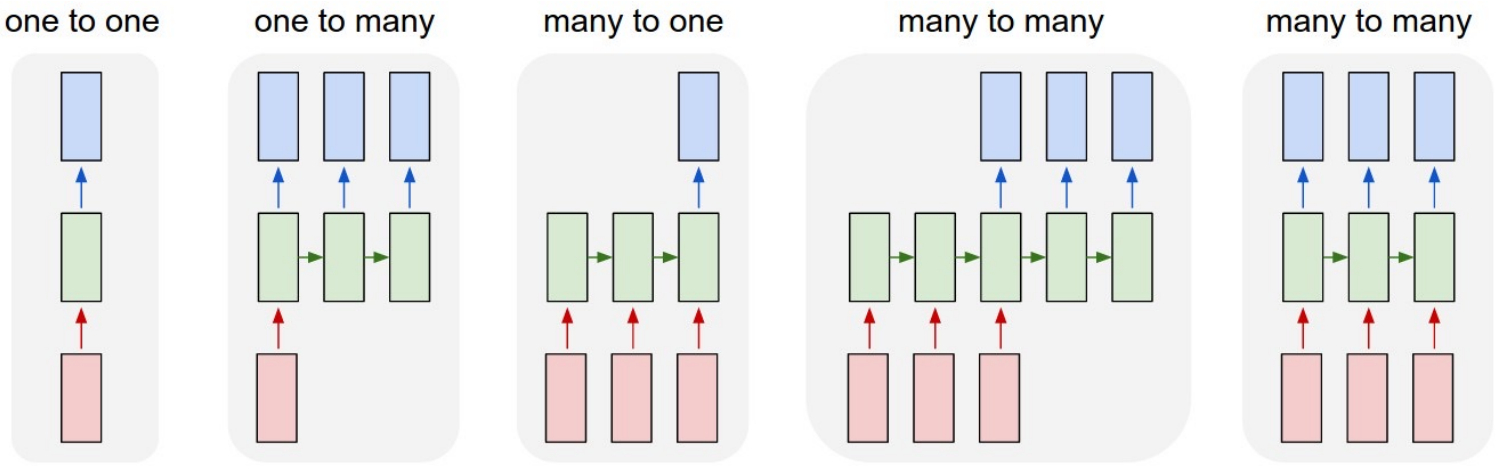
# Recurrent Neuron Over Time




Previous inputs and new inputs are combined and passed through an activation function.

# Designing RNNs



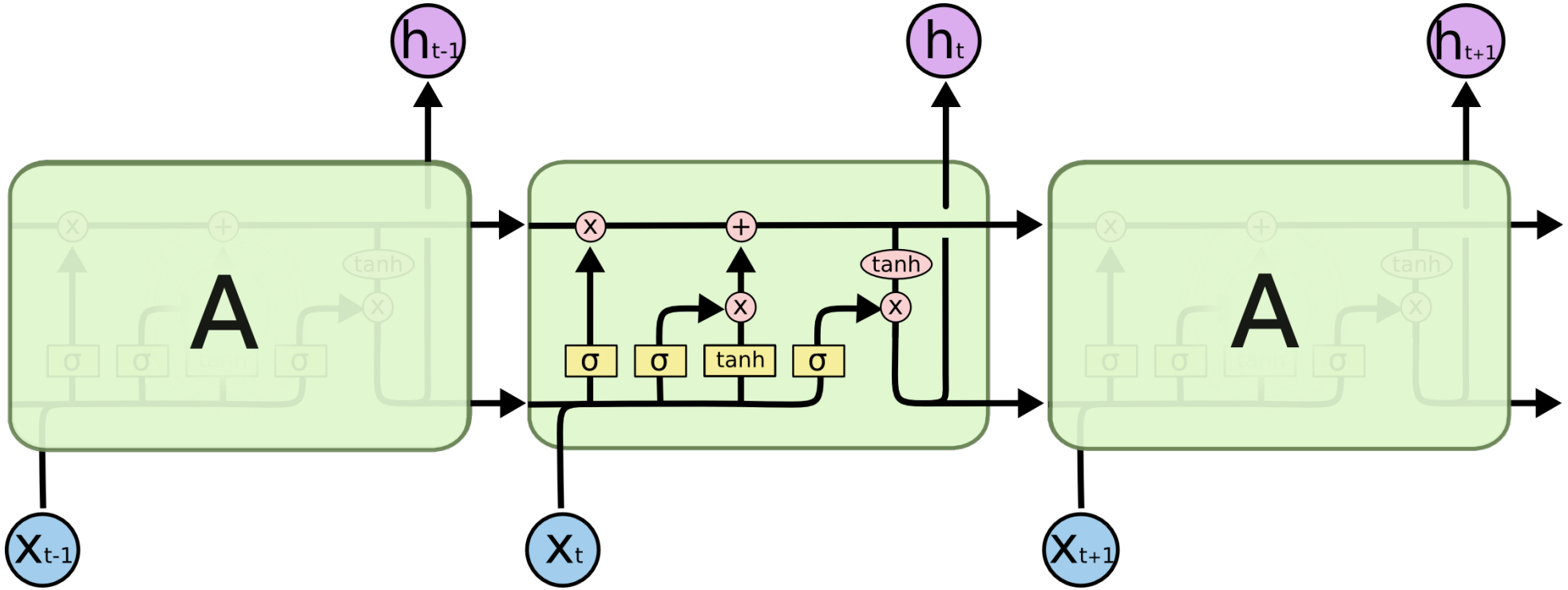

Via https://karpathy.github.io/2015/05/21/rnn-effectiveness/

# Long Short-Term Memory (LSTM) Neuron

RNNs have problems with *long-term dependencies*. As they get longer, they start to forget.

LSTMs adds a memory gates to the neuron.

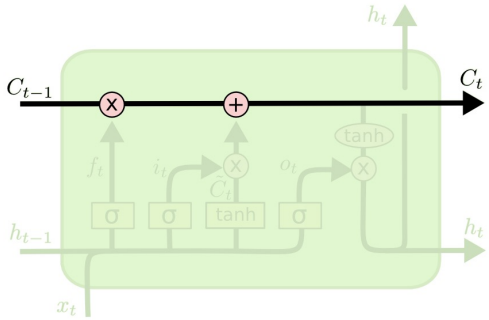
The neuron passes two weights back to itself. One represents the long-term member, and the other represents the short-term member.



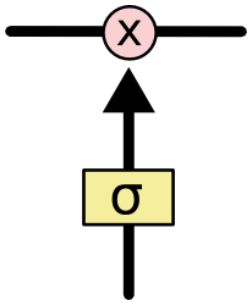

More info: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

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

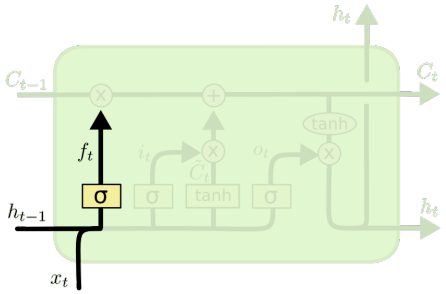
# LSTM Deconstructed



Propagates information forward.



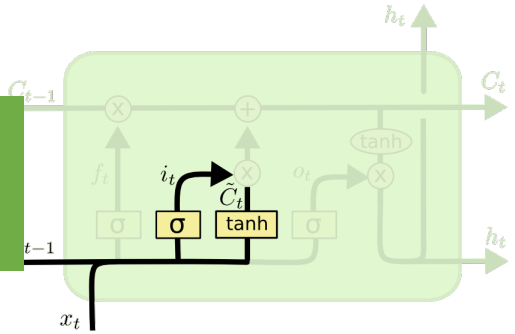

Gates control information flow. (A zero from the activation function closes the gate.)



**Forget gate** decides if we want to keep previous information.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

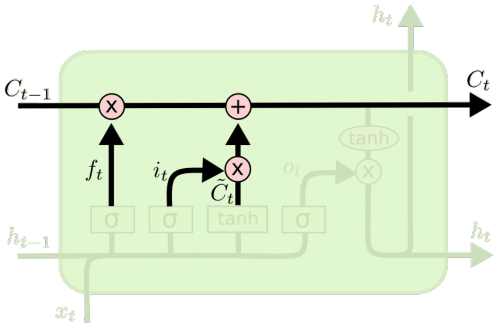
**Input gate** decides if we want to keep the current information.




$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

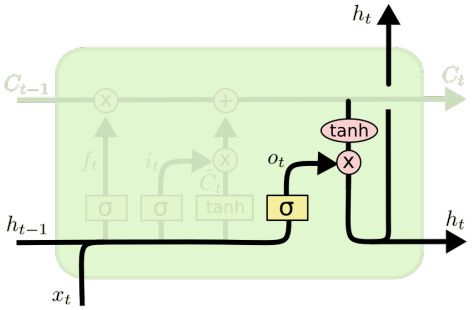
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Combine old and current state to create the new state.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Decide what to output based on the current state.




$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

# Other RNNs

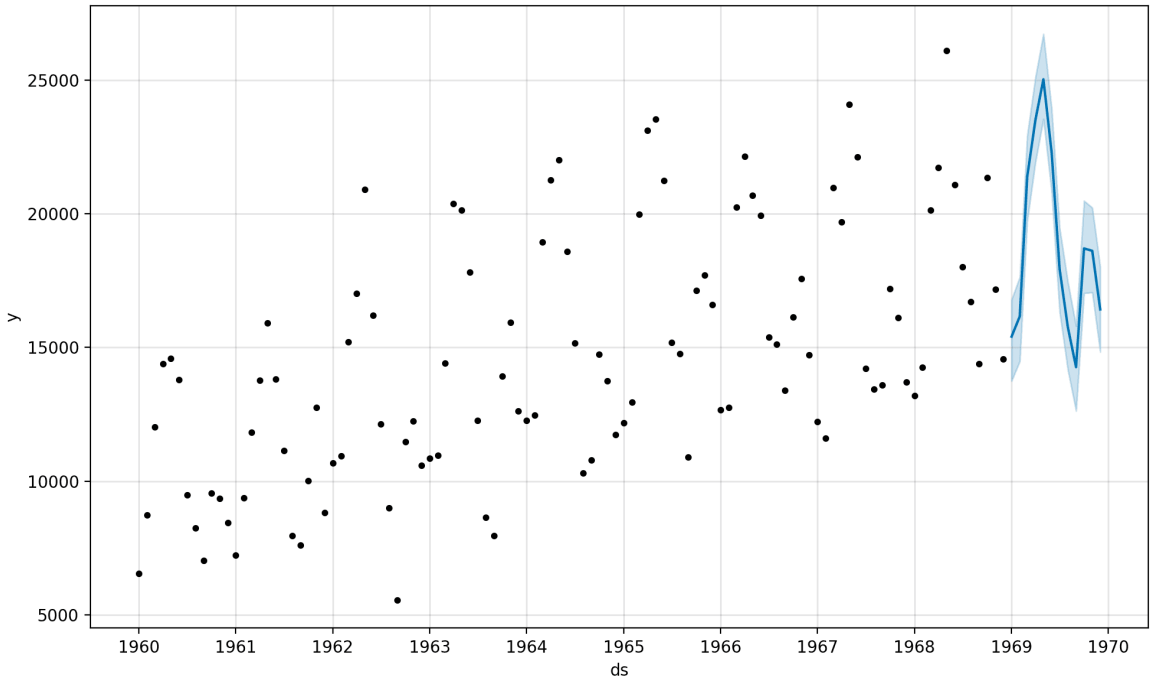
## Gated Recurrent Unit (GRU)

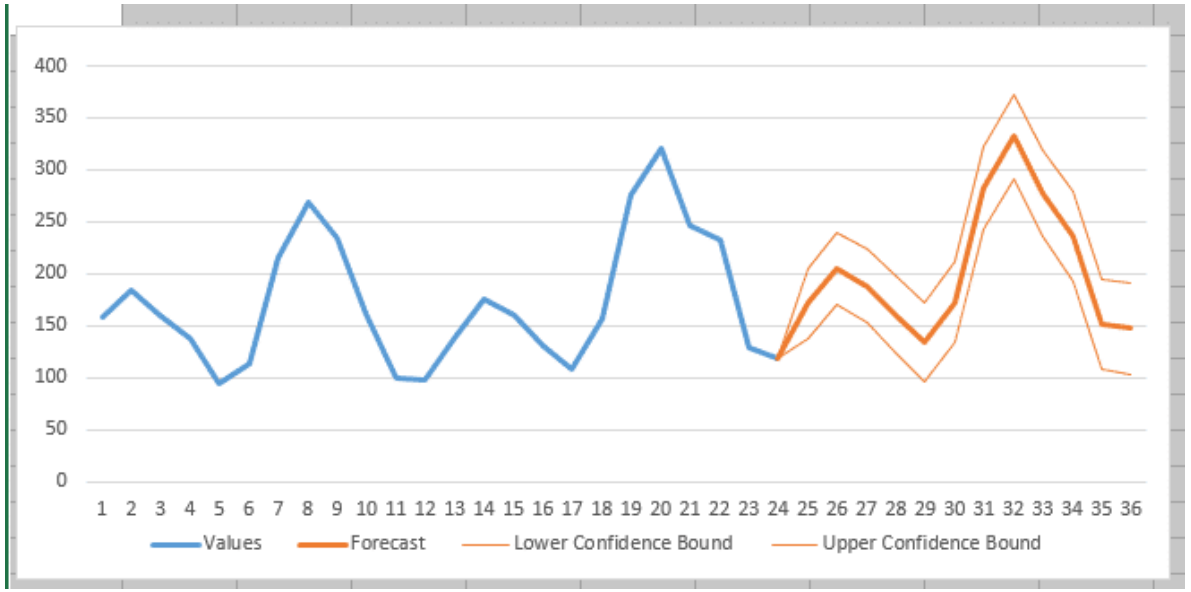
- Simpler than an LSTM with comparable performance in practice.

## 1D Convolutional Net

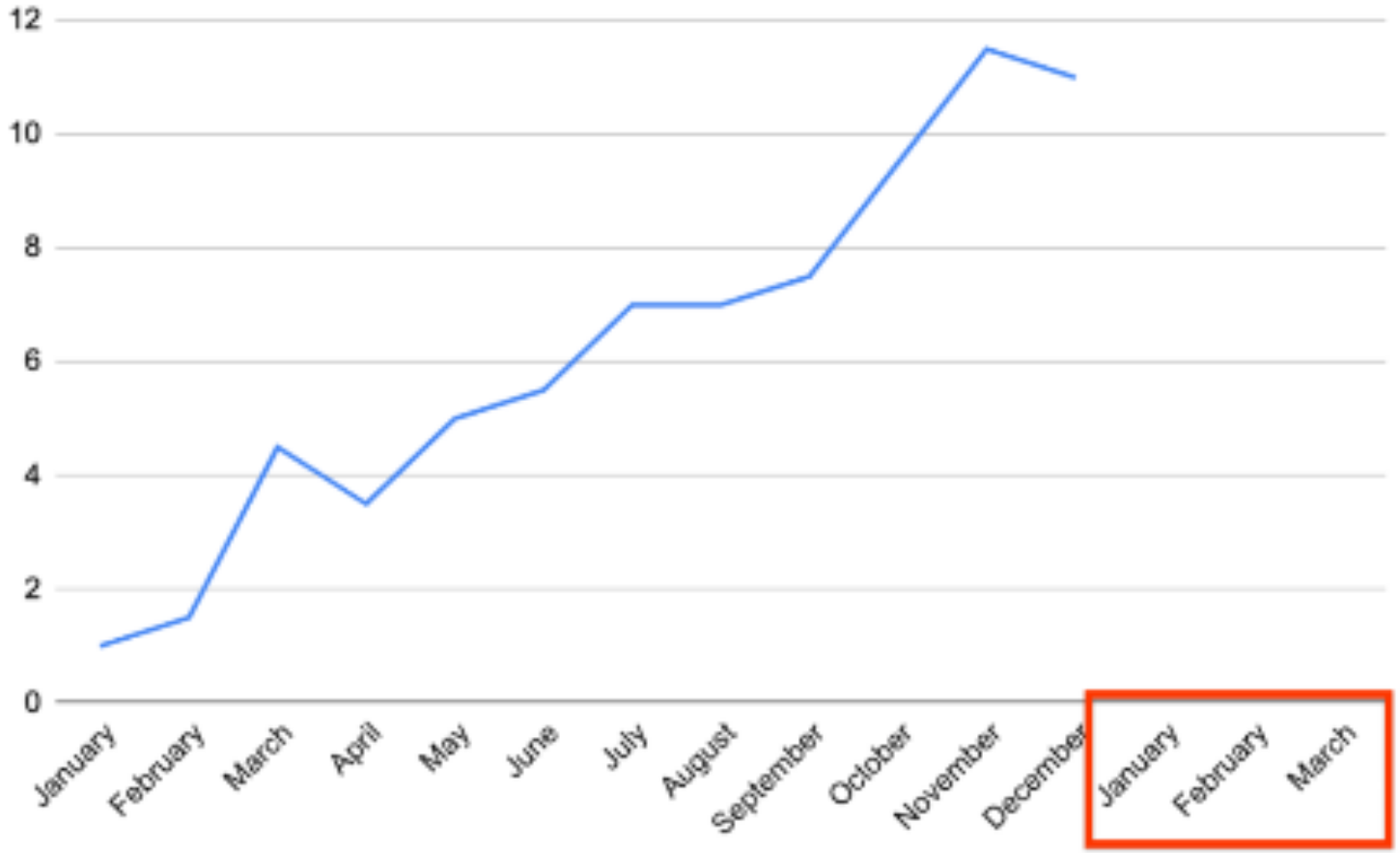
- Sliding window with some kernel size of k.

# Sequence Prediction --- Time Series Data



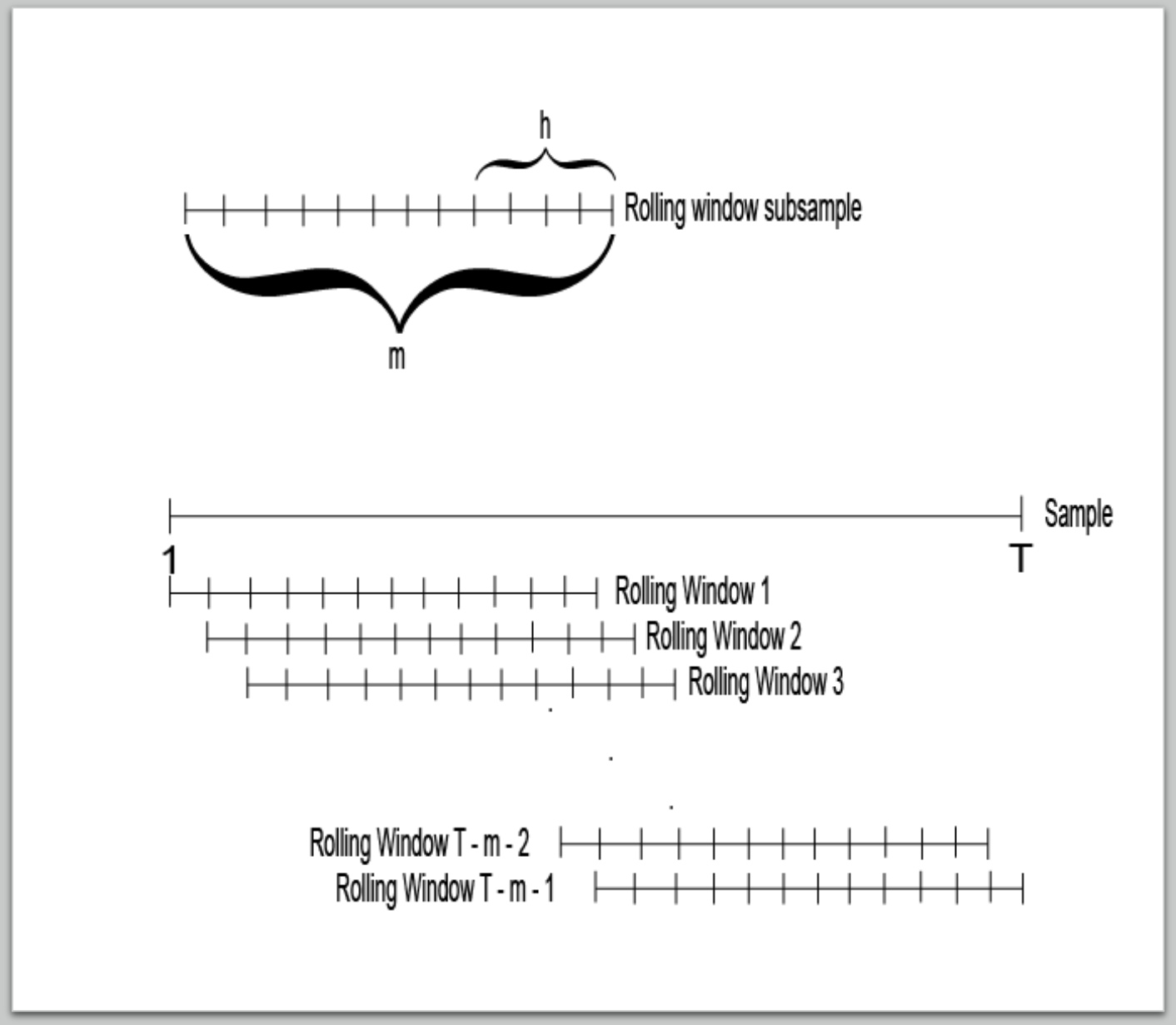



# What are we predicting?

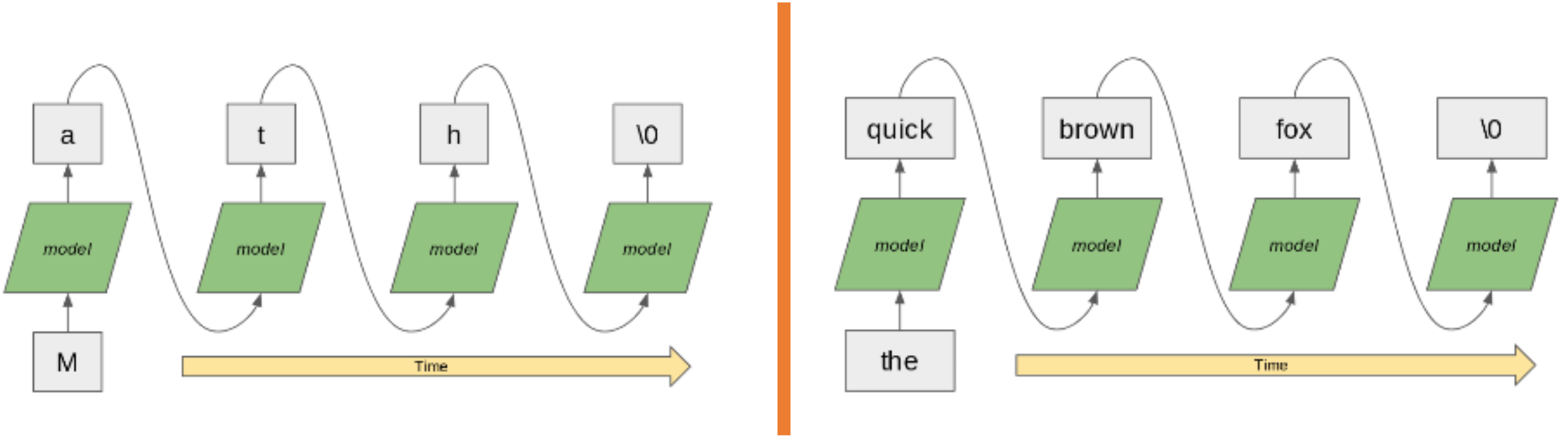


# What does “over time” mean?

- Similar to “ngrams”
- Take iterative groups of sequences as input and the final part of the sequence as output.
- Rolling window sizes may vary



# Character-level models vs. Word-level models



# RNNs for Sequence Prediction

- Statistical methods are traditional used for sequence prediction
  - E.g. Markov Chains Models, ARIMA, others.
- RNNs do require a lot of data to perform well.
- We will look at the performance of RNNs in the colab.

# Why use RNNs?

- Language Translation
- Sequence Predictions
- Sequence Generation
- Word Tagging
- Text Summarization
- Stock Prediction
- Human Activity Recognition
- … and more!