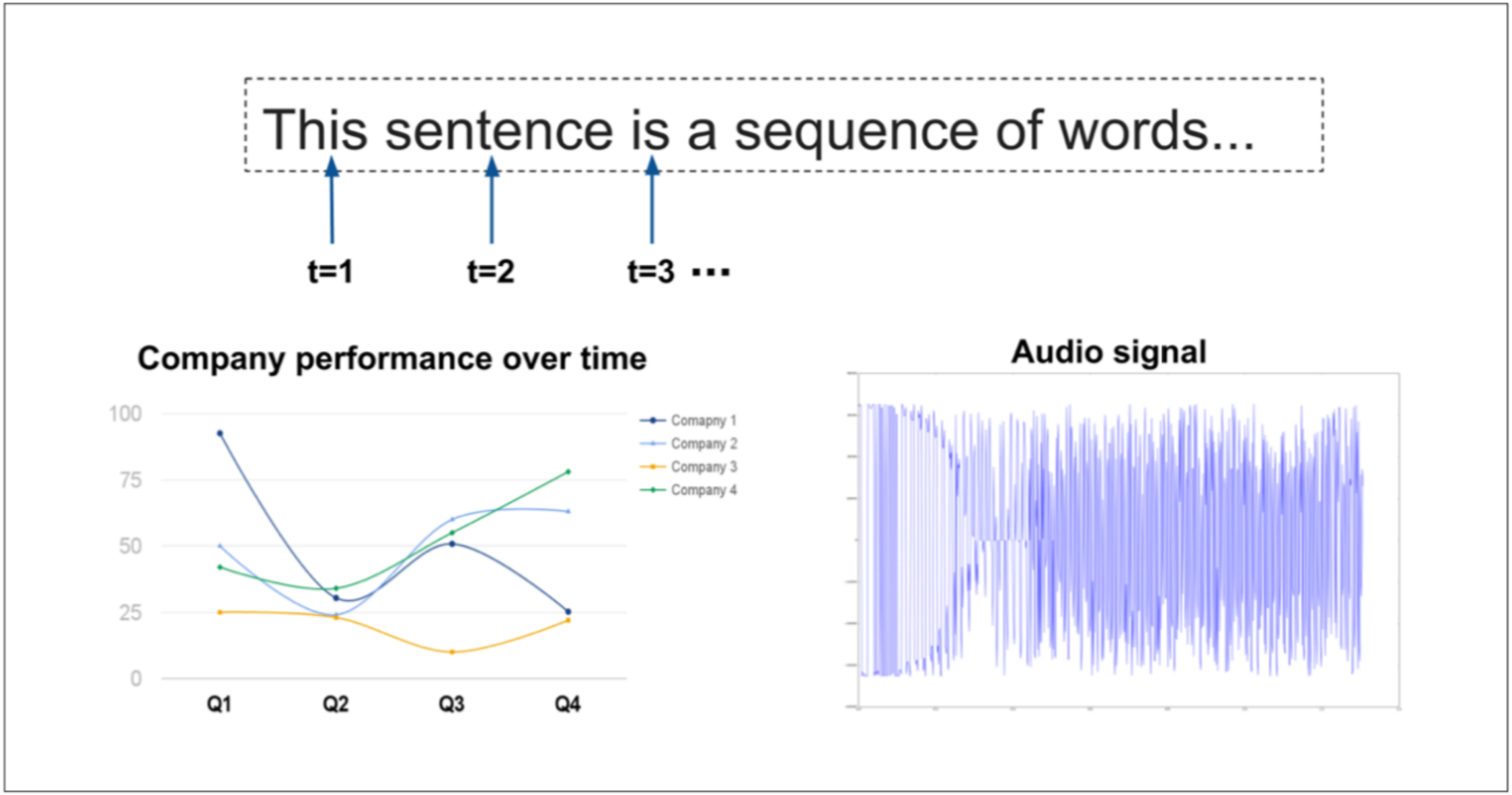
6. Recurrent Neural Networks (RNN)

RNNs for Sequences / Time-series Datasets



From : Tom Hope, Yehezkel S. Resheff, and Itay Lieder,
"Learning TensorFlow : A Guide to Building Deep Learning Systems"
Chapter 5 : Fig. 5-1, 2017,

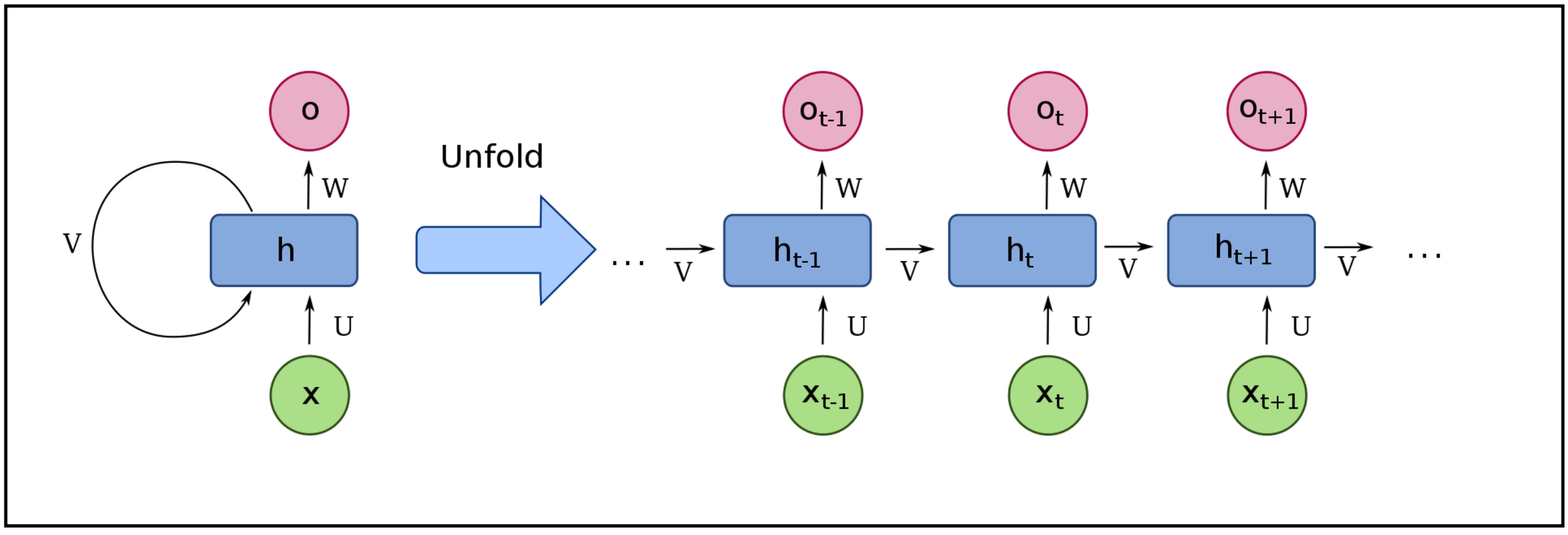
RNN

(cont'd)

RNN — Hidden Markov Model

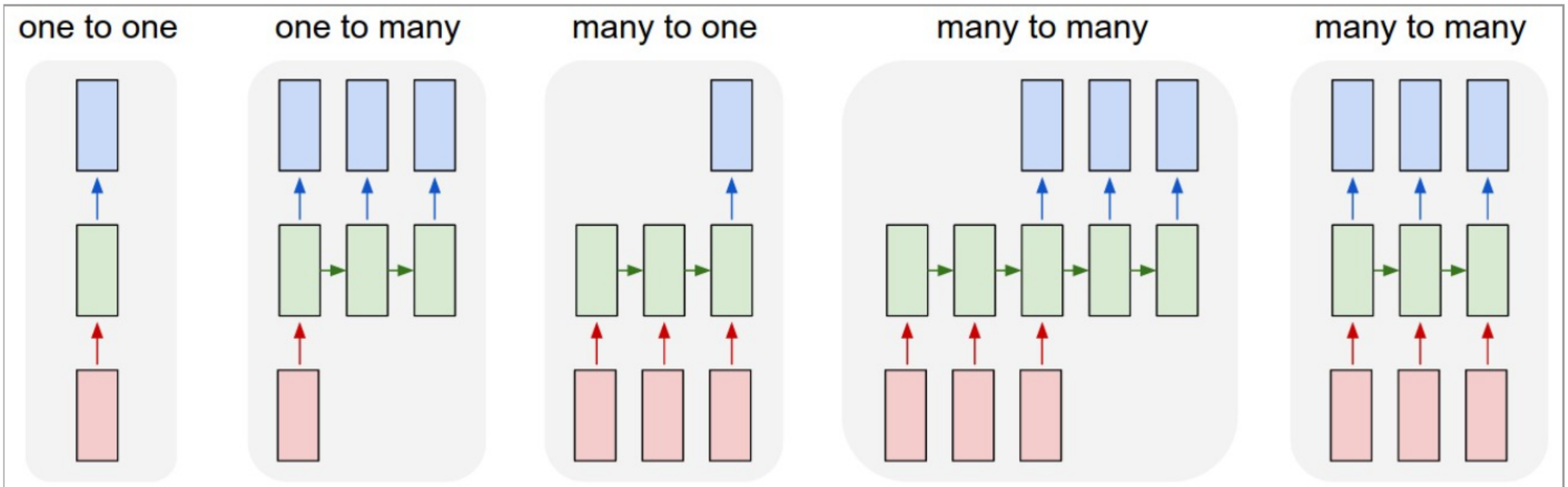
$$h_t = AF(W_x x_t + W_h h_{t-1} + b)$$

(AF : Activation Function)



from Wikipedia: Recurrent Neural networks
https://en.wikipedia.org/wiki/Recurrent_neural_network#/media/File:Recurrent_neural_network_unfold.svg

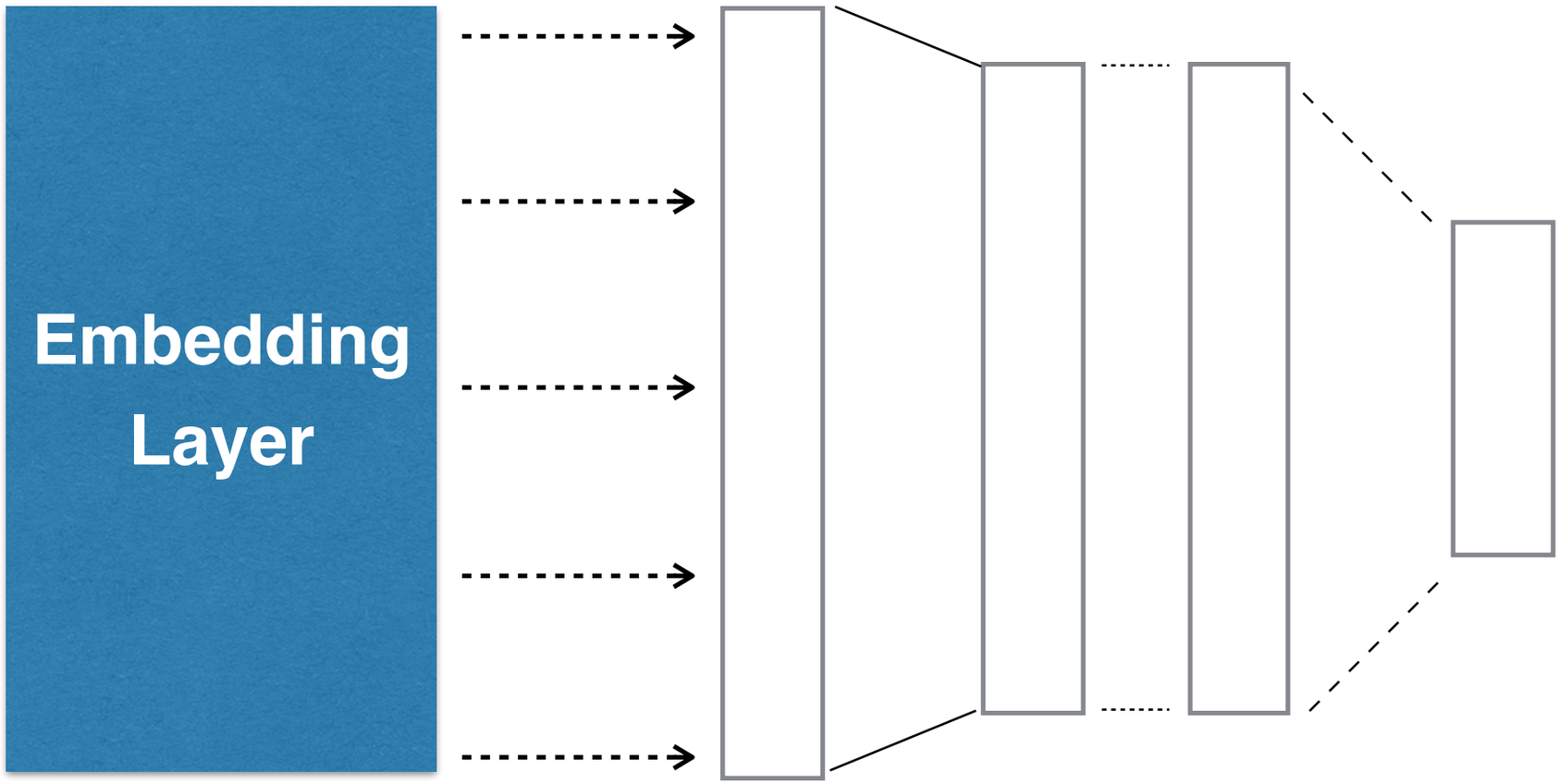
RNN Models for **Sequences / Time-series Datasets**



Each rectangle is a vector and arrows represent functions (e.g. matrix multiply). Input vectors are in red, output vectors are in blue and green vectors hold the RNN's state (more on this soon). From left to right: **(1)** Vanilla mode of processing without RNN, from fixed-sized input to fixed-sized output (e.g. image classification). **(2)** Sequence output (e.g. image captioning takes an image and outputs a sentence of words). **(3)** Sequence input (e.g. sentiment analysis where a given sentence is classified as expressing positive or negative sentiment). **(4)** Sequence input and sequence output (e.g. Machine Translation: an RNN reads a sentence in English and then outputs a sentence in French). **(5)** Synced sequence input and output (e.g. video classification where we wish to label each frame of the video). Notice that in every case are no pre-specified constraints on the lengths sequences because the recurrent transformation (green) is fixed and can be applied as many times as we like.

From : Andrej Karpathy blog
"The Unreasonable Effectiveness of Recurrent Neural Networks"
https://karpathy.github.io/2015/05/21/rnn-effectiveness/

Example : **Natural Language Processing (NLP)**

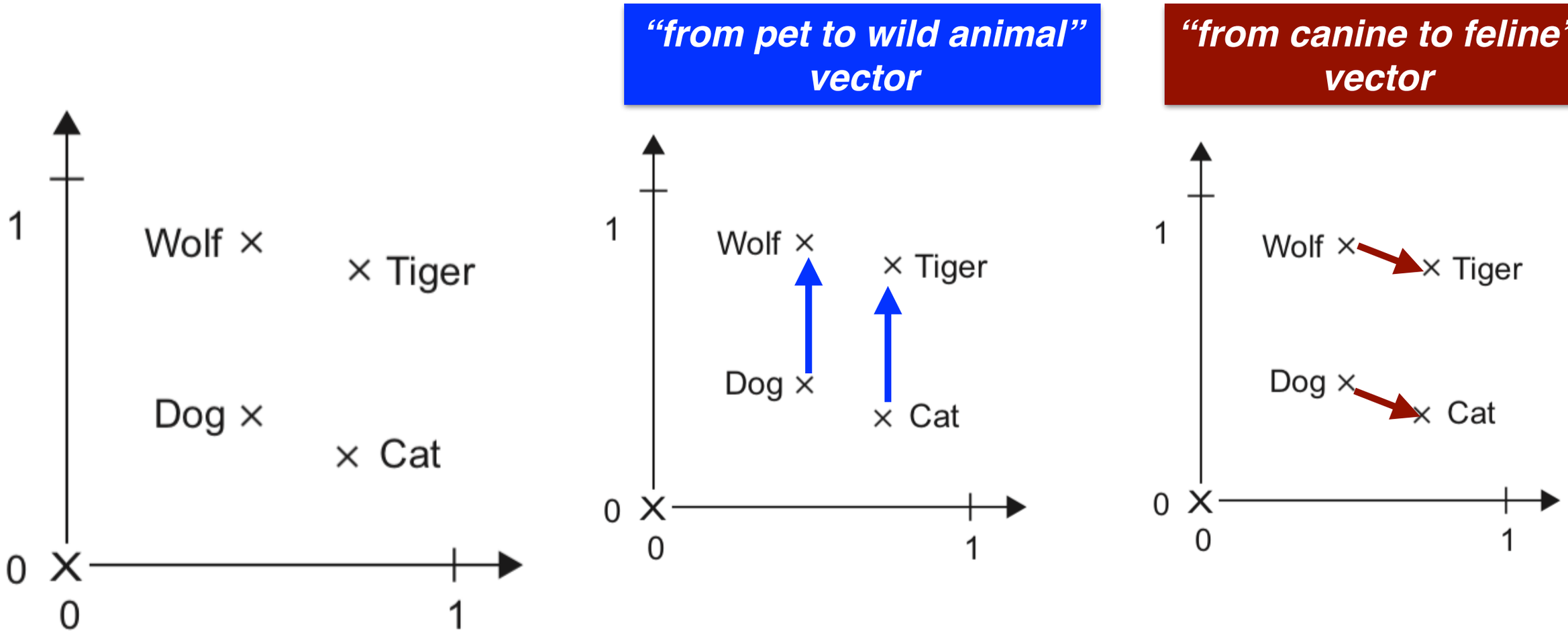


Word Embeddings for 1D Text Dataset + **Recurrent NNs**

Example : **Natural Language Processing (NLP)**

Embedding Layer for1D Text Applications

- One-hot encoding
- Word embeddings



Learning word embeddings with Embedding Layer — *WORD-to-Vector*

RNN

(cont'd)

Implementation of a simple RNN

$$h_t = AF(W_x x_t + W_h h_{t-1} + b)$$

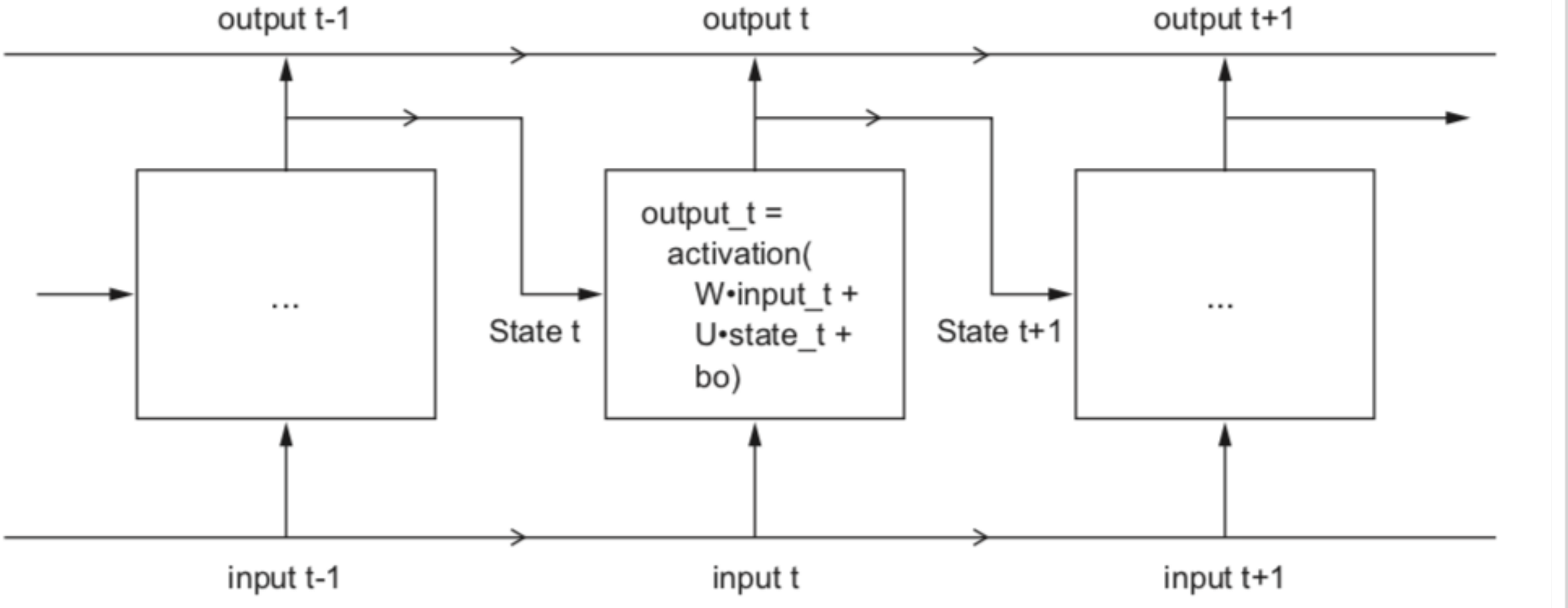


Figure from ***Deep Learning with Python***, **Chapter 6**, 2018 by Chollet

RNN

(cont'd)

RNN Implementation with LSTM (Long-Short-Term-Memory)

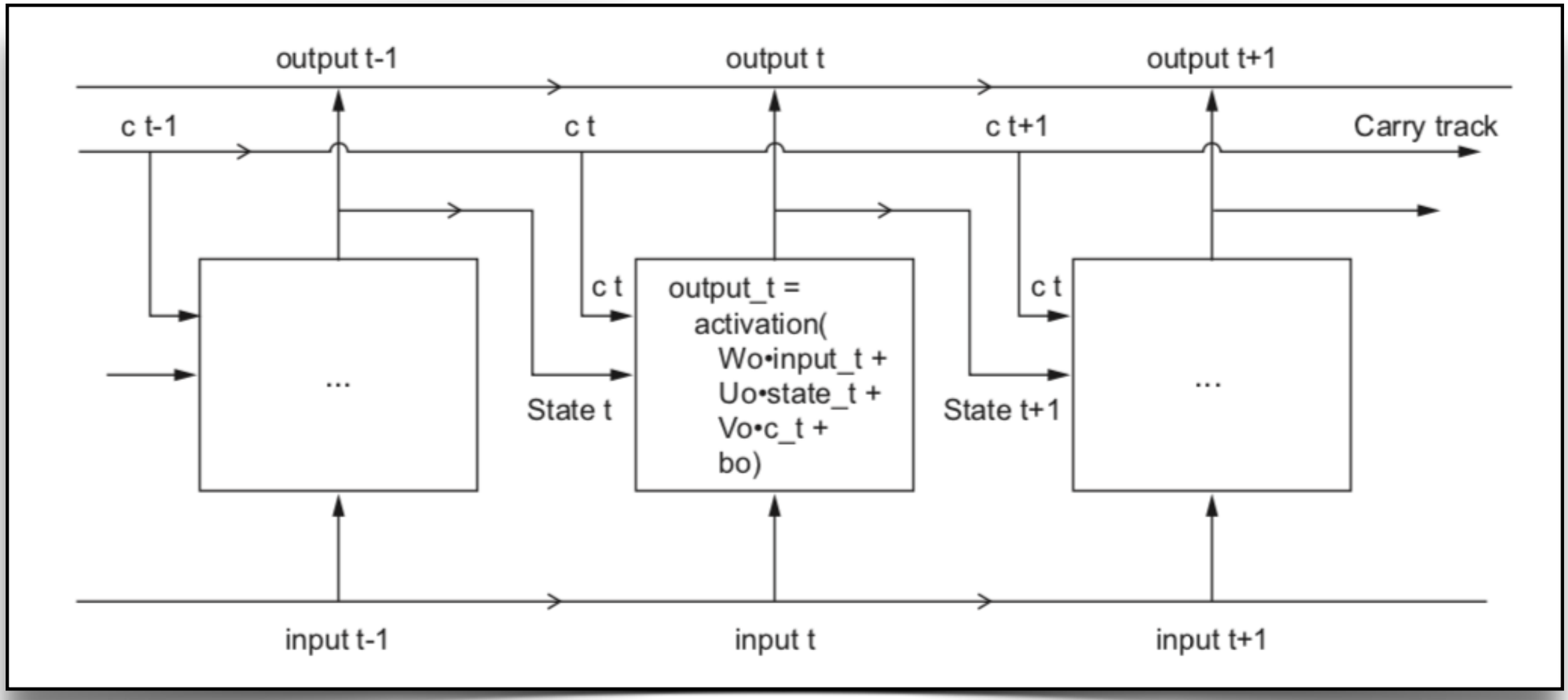


Figure from ***Deep Learning with Python***, **Chapter 6**, 2018 by Chollet