Deep Learning
- Recurrent Neural Network -

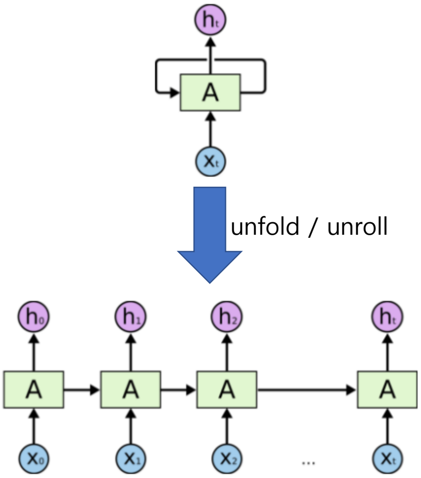
Aug. 2019

Ando Ki, Ph.D.
adki@future-ds.com

Contents

Recurrent Neural Network

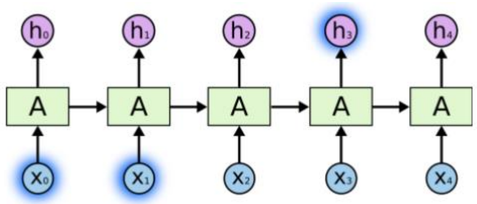
- RNN: neural network with loop
  - ▶ A RNN can be thought of as multiple copies of the same network, each passing a message to a successor



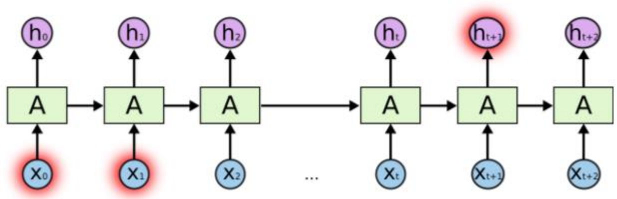
- It is called “recurrent” because it performs the same task for every element of a sequence, with the output being depended on the previous computations.
- It also can be seen as it has a “memory” which captures information about what has been calculated so far.

Predicting last word

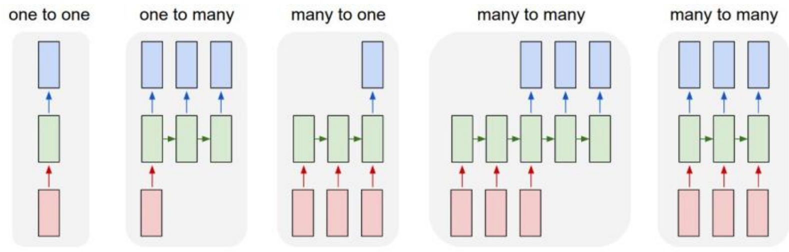
- Short term relationship
  - ▶ “the clouds are in the (   )”
    - ➲ ➔ “sky”



- Long term relationship: “I grew up in France……. I speak fluent (   )”
  - ➲ ➔ “French”



RNN applications and type of sequences

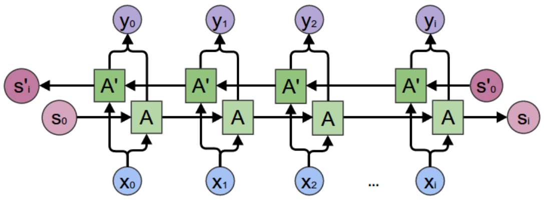


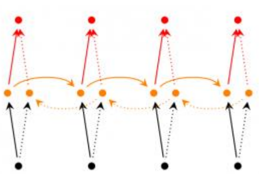
- (1) Vanilla mode of processing without RNN, from fixed-sized input to fixed-sized output (e.g. image classification). ➔ traditional neural network, e.g., MLP, CNN, ...
- (2) Sequence output (e.g. image captioning takes an image and outputs a sentence of words; music generation).
- (3) Sequence input (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).

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

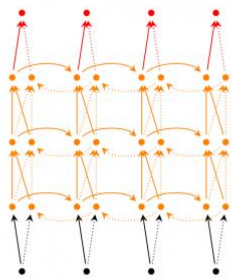
Variants of RNN: BRNN and DRNN

- **Bidirectional** (BRNN)
  - the output at time t may not only depend on the previous elements in the sequence, but also future elements.



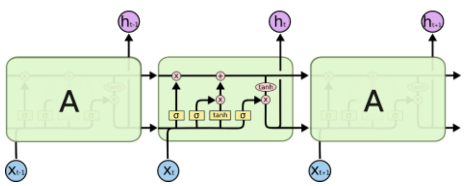


- **Deep** (DRNN)



Variants of RNN: LSTM

- LSTM
  - Long Short Term Memory network is a special kind of RNN, capable of learning long-term dependencies.
  - It fixes vanishing gradient problem of the original RNN



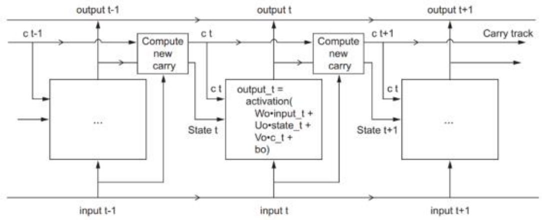
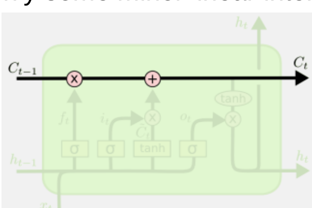


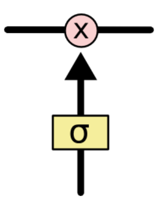
Figure 6.15 Anatomy of an LSTM

Variants of RNN: LSTM

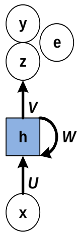
- It runs straight down the entire chain, with only some minor linear interactions.



- Gate using Sigmoid (0 ~ 1)



RNN



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

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

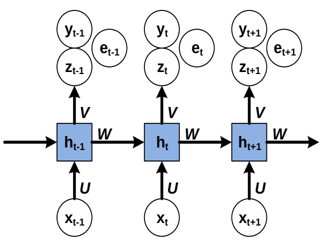
$$z_t = W_{hy} \cdot h_t$$

$$e_t = loss(z_t, y_t)$$

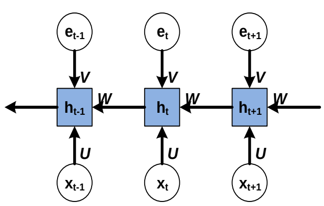
- x_t: input at time t
- z_t: output ot time t
- y_t: output at time t
- e_t: cost or error at time t
- h_t: internal (hidden) state at time t
- W_{hh}: weight for hidden state (W)
- W_{xh}: weight for input (U)
- W_{hy}: weight for output (V)
- f_W: function with weight W

BPTT: backpropagation through time

- Forward



- Backward



- Backpropagation on hidden to output
- Backpropagation on input to hidden
- Backpropagation on hidden to hidden

㈜퓨쳐디자인시스템
34051 대전광역시 유성구 문지로 193, KAIST 문지캠퍼스, F723호
(042) 864-0211~0212 / contact@future-ds.com / www.future-ds.com

Future Design Systems, Inc.
Faculty Wing F723, KAIST Munji Campus, 193 Munji-ro, Yuseong-gu, Daejeon 34051, Korea
+82-042-864-0211~0212 / contact@future-ds.com / www.future-ds.com

FUTURE
Design Systems