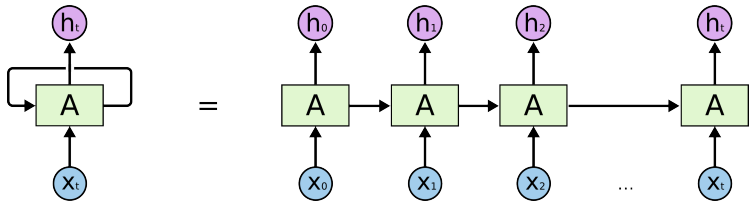
POLIMI GRADUATE SCHOOL OF MANAGEMENT

INTRODUCTION TO NATURAL LANGUAGE PROCESSING - 2

Andrea Mor - andrea.mor@polimi.it

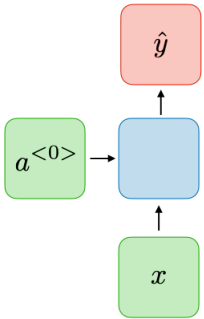
RNN: RECURRENT NEURAL NETWORKS

Given a sequence (of words): $x = x_1 x_2 \cdots x_t$

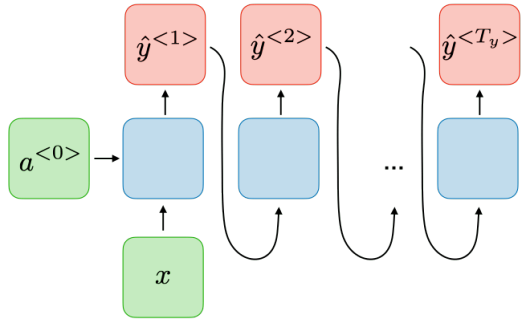


https://colah.github.io/posts/2015-08-Understanding-LSTMs/

RNN: RECURRENT NEURAL NETWORKS

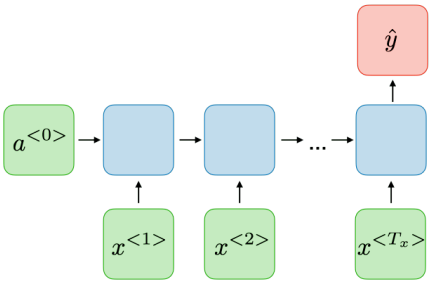


Traditional NN

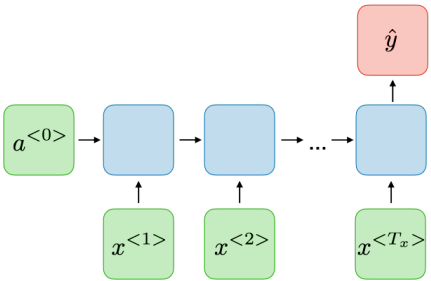


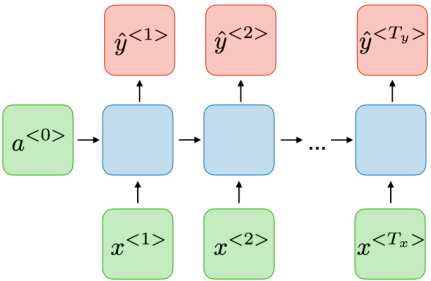
Music generation

RNN: RECURRENT NEURAL NETWORKS



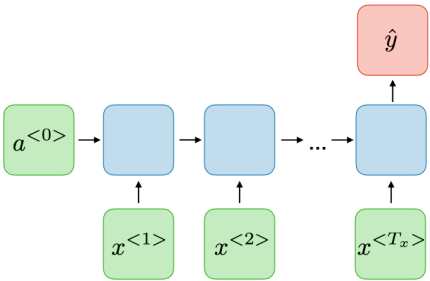
RNN: RECURRENT NEURAL NETWORKS



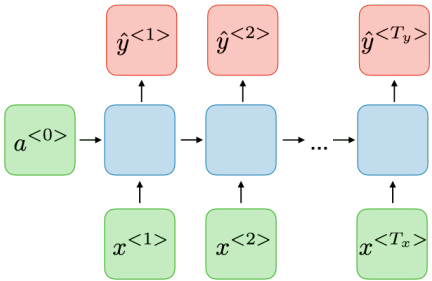


Sentiment classification

RNN: RECURRENT NEURAL NETWORKS

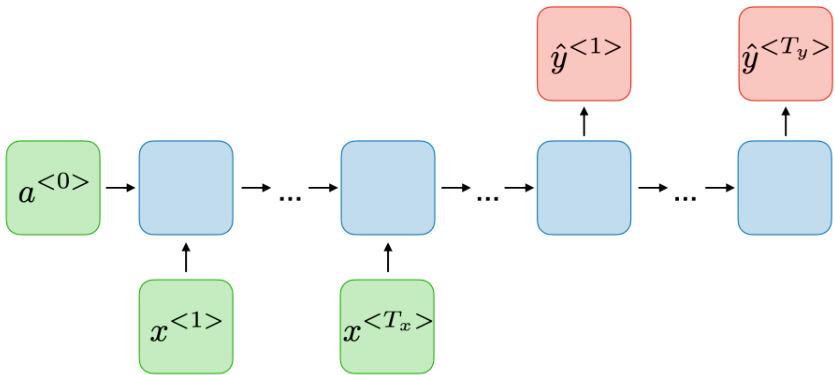


Sentiment classification

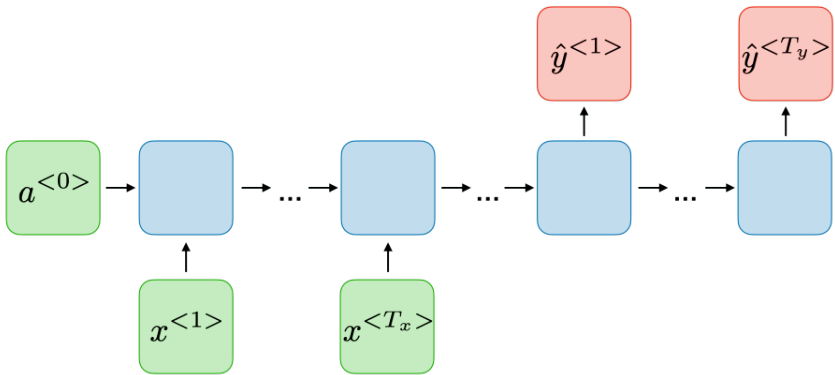


Name entity recognition

RNN: RECURRENT NEURAL NETWORKS



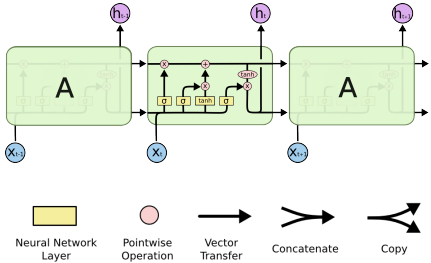
RNN: RECURRENT NEURAL NETWORKS



Text translation

LSTM: LONG SHORT TERM MEMORY NETWORKS

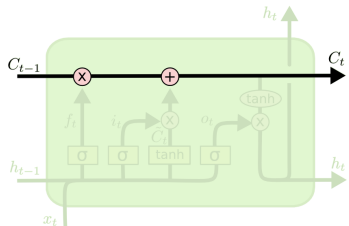
1. We keep a cell state across the sequence C_t
2. After each step t we:
   - forget something: f_t
   - include something : i_t
   - update the cell state: C_t
   - output something to the next step: h_t



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

LSTM: KEEP GLOBAL STATE

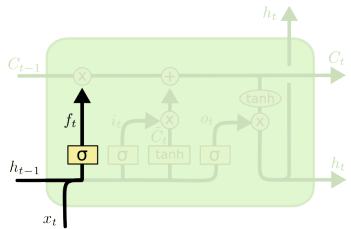
1. **We keep a cell state across the sequence** C_t
2. After each step t we:
   - forget something: f_t
   - include something : i_t
   - update the cell state: C_t
   - output something to the next step: h_t



b.io/posts/2015-08-Understanding-LSTMs/

LSTM: FORGET GATE STATE

1. We keep a cell state across the sequence C_t
2. After each step t we:
   - **forget something**: f_t
   - include something : i_t
   - update the cell state: C_t
   - output something to the next step: h_t

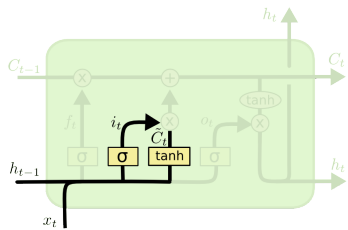


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

b.io/posts/2015-08-Understanding-LSTMs/

LSTM: INPUT GATE STATE

1. We keep a cell state across the sequence C_t
2. After each step t we:
   - forget something: f_t
   - **include something** : i_t
   - update the cell state: C_t
   - output something to the next step: h_t



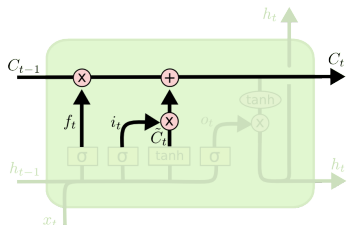
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

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

b.io/posts/2015-08-Understanding-LSTMs/

LSTM: UPDATE CELL STATE

1. We keep a cell state across the sequence C_t
2. After each step t we:
   - forget something: f_t
   - include something : i_t
   - **update the cell state**: C_t
   - output something to the next step: h_t

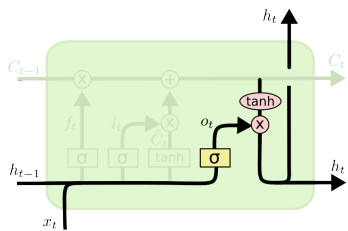


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

b.io/posts/2015-08-Understanding-LSTMs/

LSTM: CELL OUTPUT

1. We keep a cell state across the sequence C_t
2. After each step t we:
   - forget something: f_t
   - include something : i_t
   - update the cell state: C_t
   - **output something to the next step**: h_t



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

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

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

GENERATING TEXT

1. From the text, we create a training set form by couples $([x_1, \ldots, x_t], y_t)$ where:
   - $[x_1, \ldots, x_t]$ is a sequence of t elements (letters, words)
   - y_t is the element to be predicted
2. From a seed sequence we sequentially generate the text consider as input the last sequence.

THANK YOU