BABD

Masters in Business Analytics and Big Data

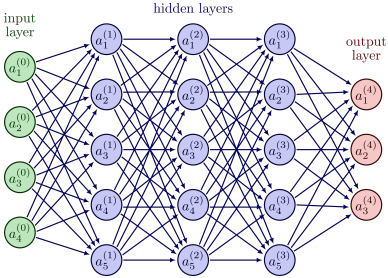
Introduction to NN Construction

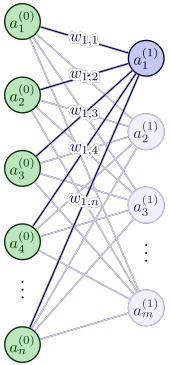
Mauricio Soto - mauricioabel.soto@polimi.it

Contents

- Fast Forward
- Convolutional NN
- RNN - Long Short Term Memory networks (LSTM)

Fast Forward





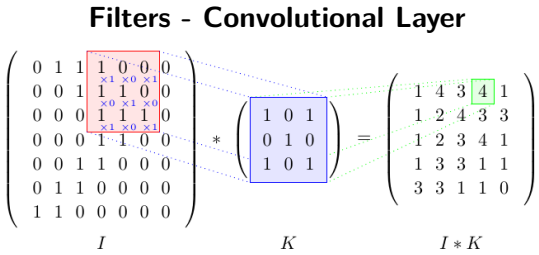
$$= \sigma\left(w_{1,0}a_0^{(0)} + w_{1,1}a_1^{(0)} + \ldots + w_{1,n}a_n^{(0)} + b_1^{(0)}\right)$$

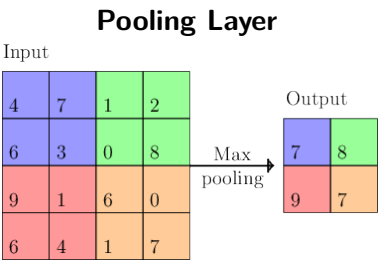
$$= \sigma\left(\sum_{i=1}^{n} w_{1,i}a_i^{(0)} + b_1^{(0)}\right)$$

$$\begin{pmatrix} a_1^{(1)} \\ a_2^{(1)} \\ \vdots \\ a_m^{(1)} \end{pmatrix} = \sigma\left[\begin{pmatrix} w_{1,0} & w_{1,1} & \ldots & w_{1,n} \\ w_{2,0} & w_{2,1} & \ldots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,0} & w_{m,1} & \ldots & w_{m,n} \end{pmatrix}\begin{pmatrix} a_1^{(0)} \\ a_2^{(0)} \\ \vdots \\ a_n^{(0)} \end{pmatrix} + \begin{pmatrix} b_1^{(0)} \\ b_2^{(0)} \\ \vdots \\ b_m^{(0)} \end{pmatrix}\right]$$

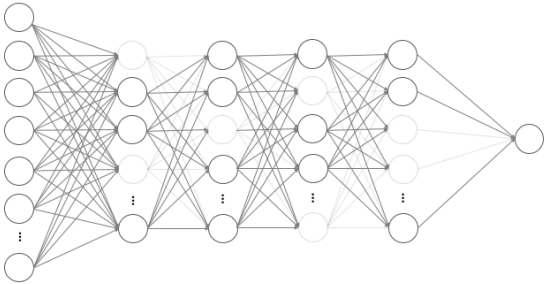
$$a^{(1)} = \sigma\left(\mathbf{W}^{(0)}a^{(0)} + \mathbf{b}^{(0)}\right)$$

Convolutional Neural Neural



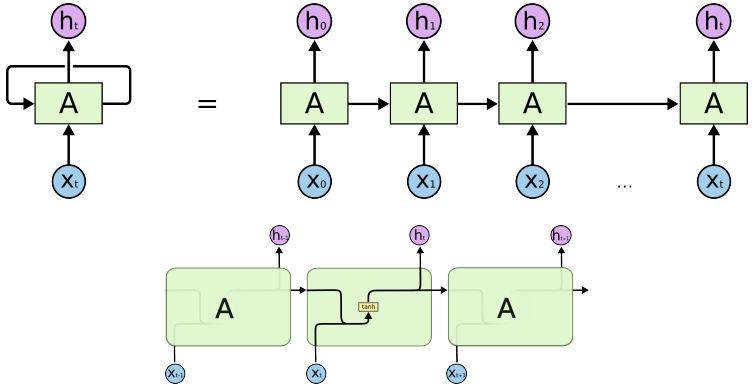


Dropout Layer



RNN: Recurrent Neural Networks

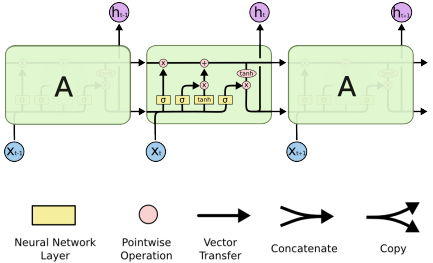
Given a sequence (of words): $x = x_1 x_2 \cdots x_t$



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

LSTM: Long Short Term Memory networks

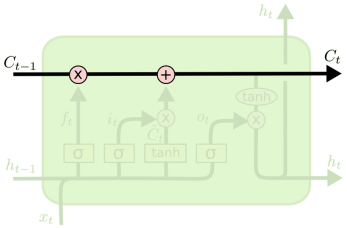
1. We keep a cell state across the sequence C_t
2. After each step t we:
   - forget something: f_t
   - include something : i_t
   - update the cell state: C_t
   - output something to the next step: h_t



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

LSTM: Keep Global state

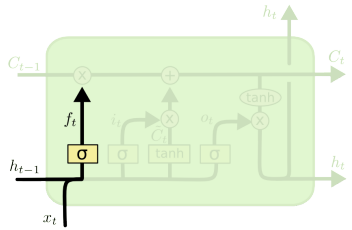
1. **We keep a cell state across the sequence** C_t
2. After each step t we:
   - forget something: f_t
   - include something : i_t
   - update the cell state: C_t
   - output something to the next step: h_t



http://colah.github.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

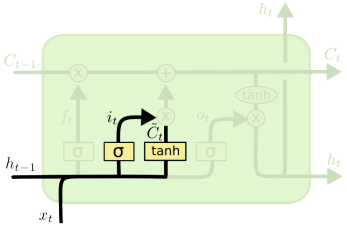


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

http://colah.github.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



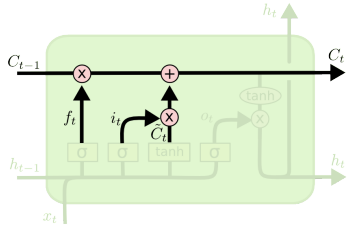
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

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

http://colah.github.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

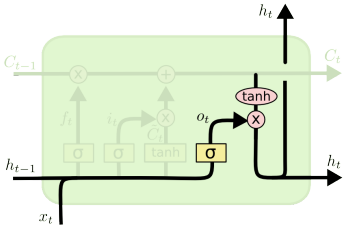


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

http://colah.github.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



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

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

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.