Introducción a Redes Neuronales Artificiales

Mauricio A. Álvarez PhD,
H.F. Garcia C. Guarnizo (TA)

Universidad Tecnológica de Pereira, Pereira, Colombia

Outline

Outline

Sequence Learning

1. Sequence learning is the study of machine learning algorithms designed for sequential data.
2. Discrete dynamical systems based on delay steps.
3. Language model is one of the most interesting topics that use sequence labeling.
   - Language modeling.
   - Machine Translation.

Types of RNNs

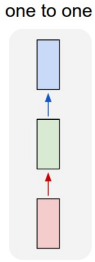


Image Classification
Vanilla RNN

Types of RNNs

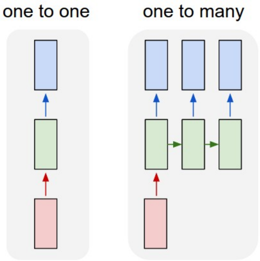
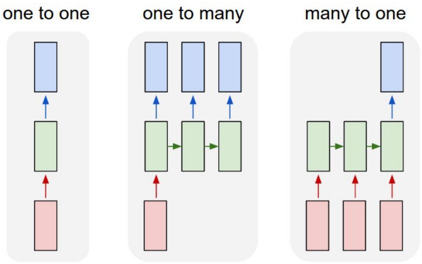


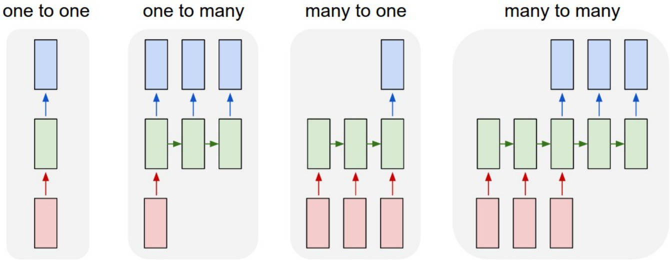
Image Captioning

Types of RNNs



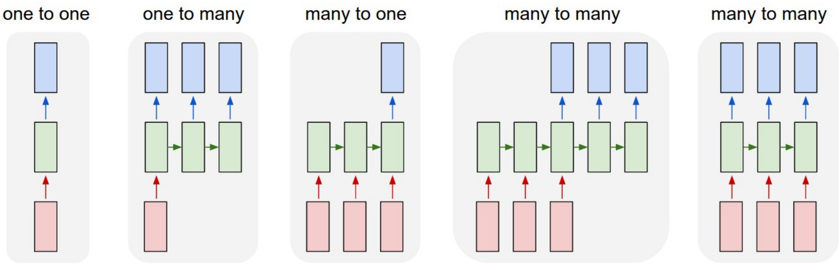
Sentiment Classification

Types of RNNs



Machine Translation

Types of RNNs

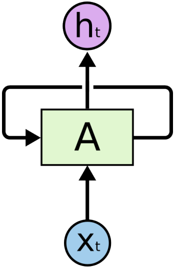


Video Labeling (frame)

Outline

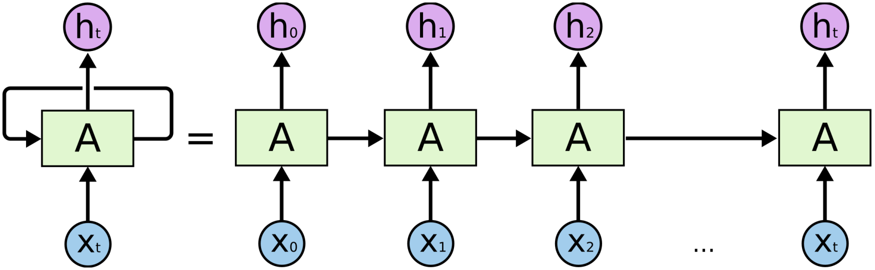
Recurrent Neural Network - RNN

$h_t \in \mathbb{R}^N, x_t \in \mathbb{R}^M \rightarrow h_t = f_W(h_{t-1}, x_t), y_t = g_{W_o}(h_t)$



Recurrent Neural Network - RNN

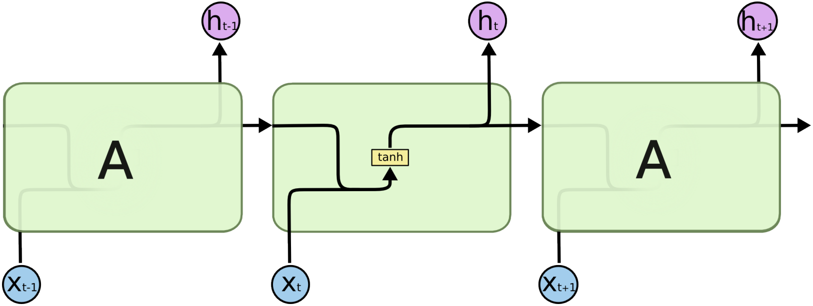
$$h_t \in \mathbb{R}^N, \, x_t \in \mathbb{R}^M \rightarrow h_t = f_W(h_{t-1}, x_t), \, y_t = g_{W_o}(h_t)$$



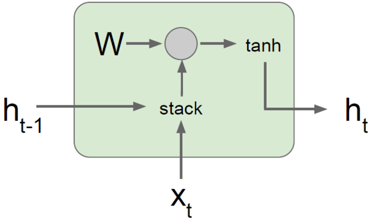
Ountline

Vanilla RNN

$$h_t \in \mathbb{R}^N,\, x_t \in \mathbb{R}^M \rightarrow h_t = \tanh(W_{hh}h_{t-1} + W_{hx}x_t),\, y_t = \mathbf{W}_o h_t$$

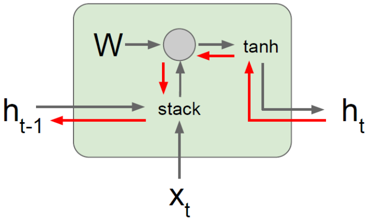


Vanilla RNN



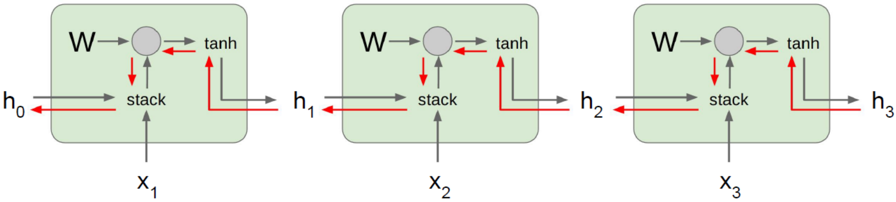
$$
\begin{aligned}
h_t &= \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \\
&= \tanh\left(\begin{pmatrix} W_{hh} & W_{hx} \end{pmatrix}\begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \\
&= \tanh\left(W\begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)
\end{aligned}
$$

Vanilla RNN



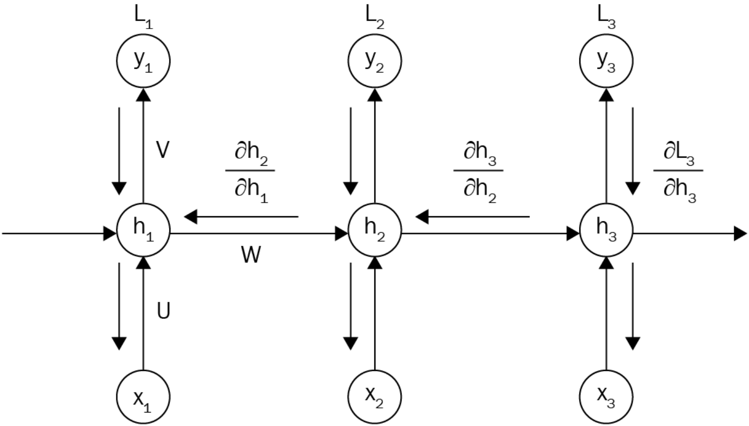
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$
$$= \tanh\left(\begin{pmatrix} W_{hh} & W_{hx} \end{pmatrix} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$
$$= \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$

Vanilla RNN



$$h_1 = \tanh(W_{hh}h_0 + W_{hx}x_1)$$
$$h_2 = \tanh(W_{hh}(\tanh(W_{hh}h_0 + W_{hx}x_1) + W_{hx}x_2)$$
$$h_3 = \tanh(\color{red}{W_{hh}}(\tanh(\color{red}{W_{hh}} \tanh(\color{red}{W_{hh}}h_0 + W_{hx}x_1) + W_{hx}x_2) + W_{hx}x_3)$$

Vanilla RNN: Gradients



Vanilla RNN: Gradient problems

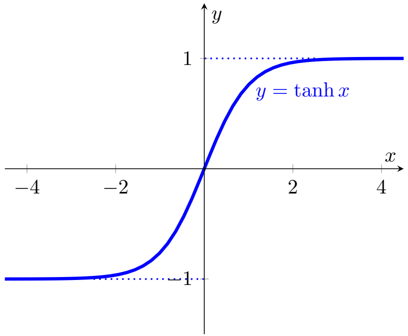
$$\frac{\partial E}{\partial \theta} = \sum_{1 \leq t \leq T} \frac{\partial E_t}{\partial \theta}$$

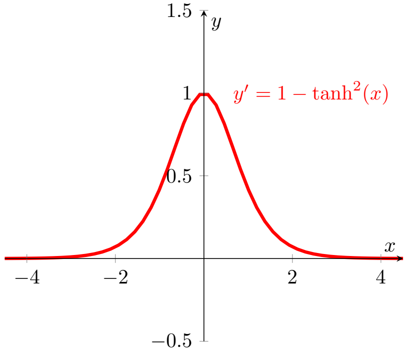
$$\frac{\partial E_t}{\partial \theta} = \sum_{1 \leq k \leq t} \left(\frac{\partial E_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial^+ h_k}{\partial \theta} \right)$$

$$\frac{\partial h_t}{\partial h_k} = \prod_{t \geq i > k} \frac{\partial h_i}{\partial h_{i-1}} = \prod_{t \geq i > k} W_{hh}^\top \operatorname{diag}(\sigma'(h_{i-1}))$$

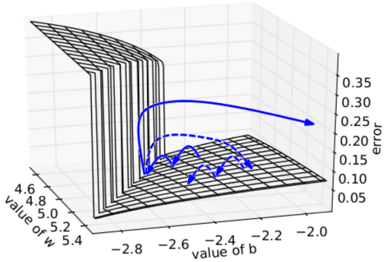
R. Pascanu - On the difficulty of training recurrent neural networks, 2013.

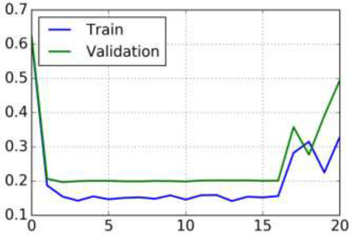
Vanilla RNN: Vanishing Gradients





Vanilla RNN: Exploding Gradients

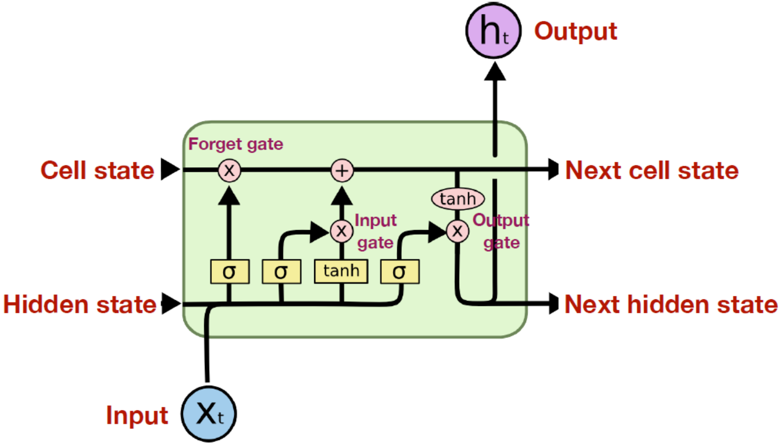




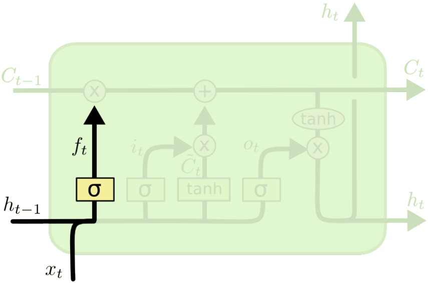
$$\frac{\partial h_t}{\partial h_k} = \prod_{t \geq i > k} \frac{\partial h_i}{\partial h_{i-1}} = \prod_{t \geq i > k} W_{hh}^{\top} \operatorname{diag}(\sigma'(h_{i-1}))$$

Ountline

LSTM - Long-Short Term Memory



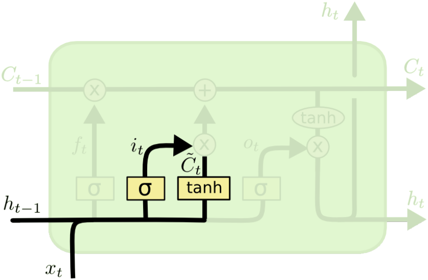
http://colah.github.io/posts/2015-08-Understanding-LSTMs/



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

Decide which information to throw away from the cell state.

LSTM - Input Gate

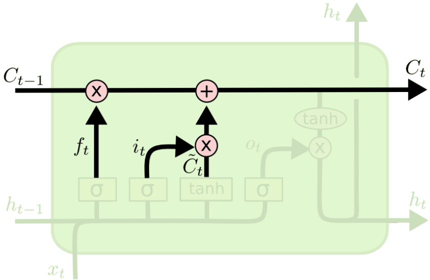


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

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

Decide which information to store to the cell state.

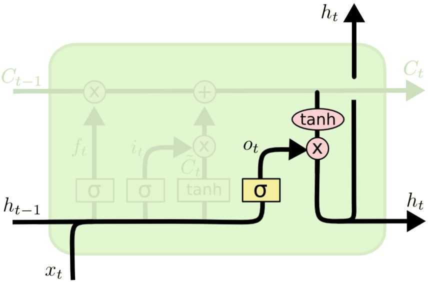
LSTM - Update Cell State



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

Update the cell state scaled by input and forget gates.

LSTM - Output Gate

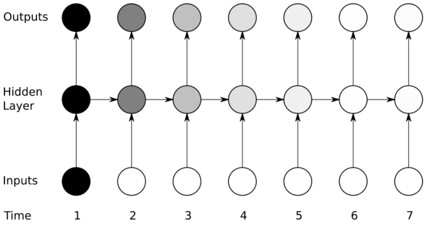


$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

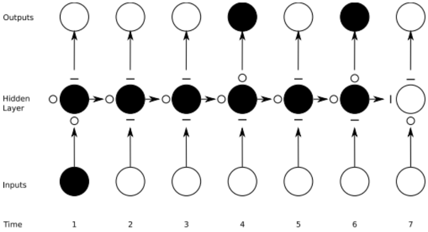
$$h_t = o_t * \tanh\left(C_t\right)$$

Output based on the updated cell state.

LSTM vs. Vanilla RNN

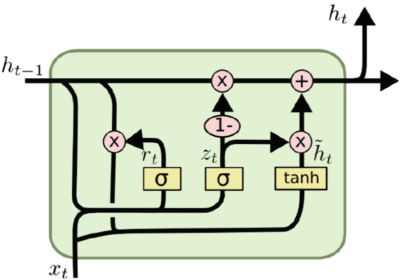


Vanilla RNN



LSTM

Ountline



$$z_t = \sigma \left(W_z \cdot [h_{t-1}, x_t] \right)$$

$$r_t = \sigma \left(W_r \cdot [h_{t-1}, x_t] \right)$$

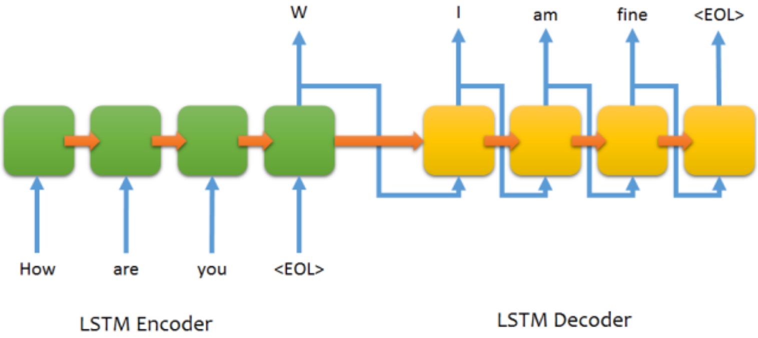
$$\tilde{h}_t = \tanh \left(W \cdot [r_t * h_{t-1}, x_t] \right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be

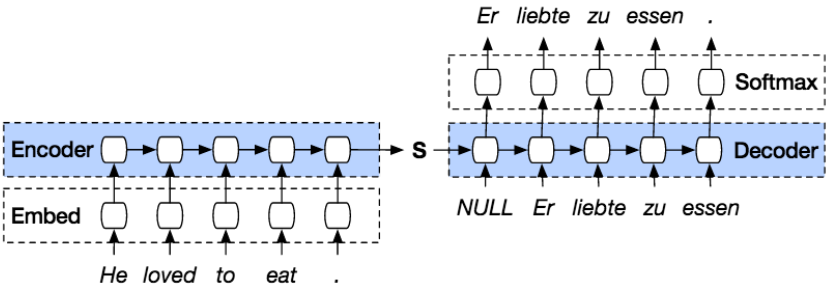
Outline

Chatbots



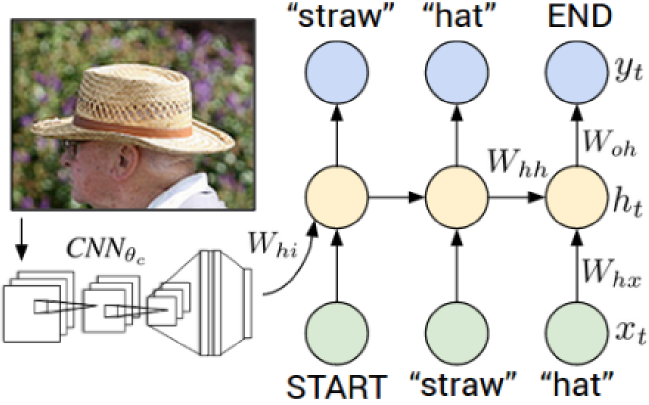
https://github.com/farizrahman4u/seq2seq

Machine Translation



https://github.com/farizrahman4u/seq2seq

Image Captioning



http://bit.ly/neuraltalkdemo

Summary

1. Vanilla RNNs presents two main issues during its learning based on backprop, i.e. *Vanishing* and *Exploding* gradients.
2. LSTM and GRU networks are able to overcome the above problems, and are the most used RNNs.