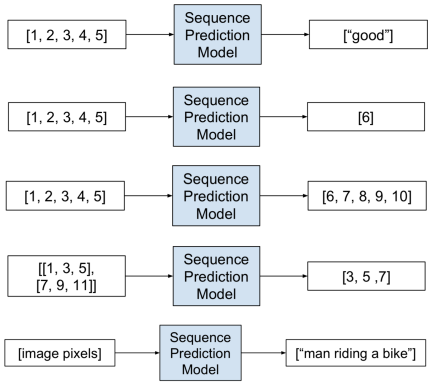
Modeling sequences with recurrent networks

Julien VELCIN

Université Lumière Lyon 2

Master 2 Data Mining

Multiple tasks



Some history

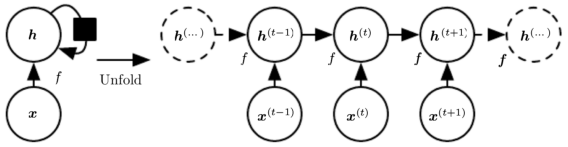
- aims to model more appropriately sequence structure
- foundational papers in both cognitive science and computationnal neuroscience journals:
  - Hopfield, 1982
  - Jordan, 1987
  - Elmann, 1990
- new era
  - LSTM
  - GRU
  - Bidirectional RNN

Simple RNN

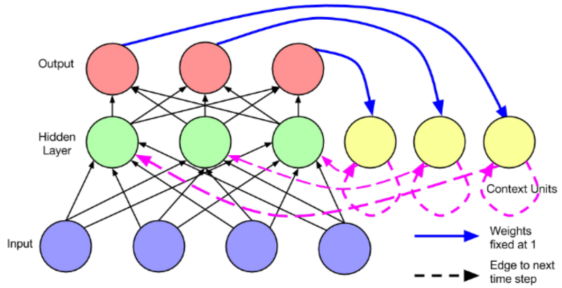
Forward pass of a simple RNN at time t :

$$\mathbf{h}^{(t)} = f(W^{hx}\mathbf{x}^{(t)} + W^{hh}\mathbf{h}^{(t-1)} + \mathbf{b})$$

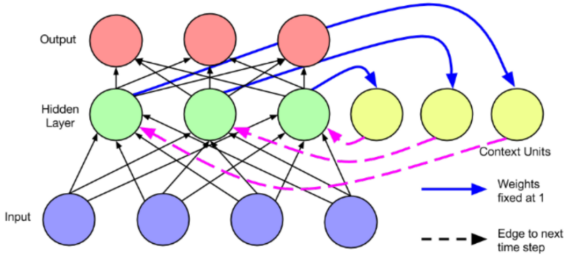
$$\hat{\mathbf{y}}^{(t)} = softmax(W^{yh}\mathbf{h}^{(t)} + \mathbf{c})$$



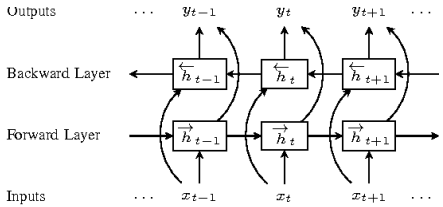
Jordan's net (1987)



Elan's net (1990)



Bidirectional RNN (Schuster and Paliwal, 1997)

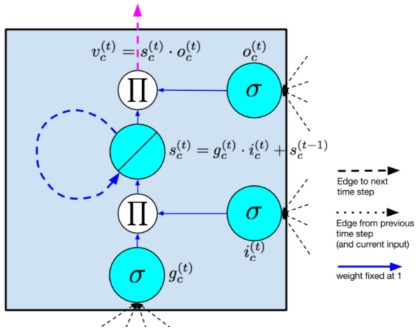


Picture taken from "A Unified Tagging Solution: Bidirectional LSTM Recurrent Neural Network with Word Embedding" (Wang et al., ArXiv 2015)

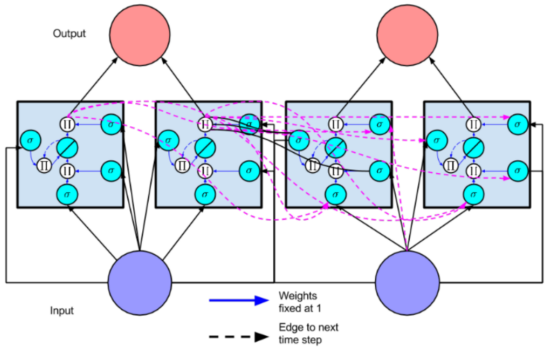
Problems of (past versions of) RNN

- training is difficult because optimisation is NP-complete
- long-range dependencies induces vanishing or exploding gradients
- truncated backprop through time can be one solution
  - cuts the time span influence
  - but kill the long-term memory
- a reborn of RNN was the introduction of gated architectures

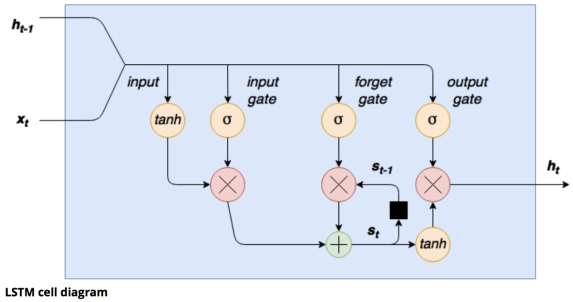
Long Short-Term Memory (Hochreiter and Schmidhuber, 1997)



Long Short-Term Memory (unfolded)

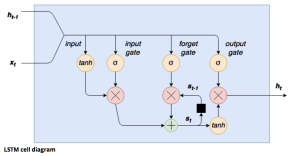


Long Short-Term Memory (another view)



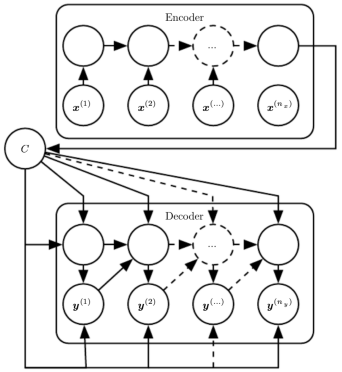
Taken from: http://adventuresinmachinelearning.com/keras-lstm-tutorial/

LSTM (math)

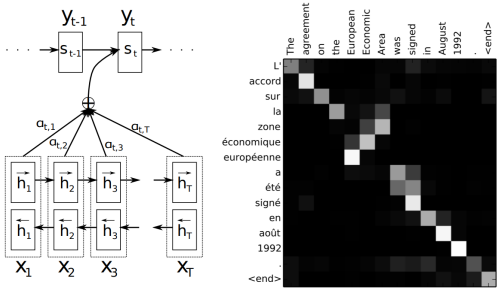


$$g = tanh(b^g + x_t.U^g + h_{t-1}.V^g)$$
$$i = \sigma(b^i + x_t.U^i + h_{t-1}.V^i)$$
$$f = \sigma(b^f + x_t.U^f + h_{t-1}.V^f)$$
$$s_t = s_{t-1} \odot f + g \odot i$$
$$o = \sigma(b^o + x_t.U^o + h_{t-1}.V^o)$$
$$h_t = tanh(s_t) \odot o$$

seq2seq architectures (Cho et al., 2014; Sutskever et al., 2014)



All you need is attention? (Bahdanau et al., 2015)



Some references

- Elman, Jeffrey L. (1990). Finding Structure in Time. Cognitive Science. 14 (2): 179–211.
- Hochreiter, S. and Schmidhuber, J. (1997). LSTM can solve hard long time lag problems. In Advances in neural information processing systems (pp. 473-479).
- Bahdanau, D., Cho, K. and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. ICLR 2015.
- Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y. (2016). Deep learning. Cambridge: MIT press.
- Brownlee, J. (2017). Long Short-Term Memory Networks with Python Develop Sequence Prediction Models with Deep Learning. Machine Learning Mastery, EBook.