Coding-based tutorial about Long Short-Term Memory (LSTM)

Guo-Wei Wei[1,2,3] and Rui Wang[1]
[1] Department of Mathematics, Michigan State University, MI 48824, USA
[2] Department of Biochemistry and Molecular Biology
Michigan State University, MI 48824, USA
[3] Department of Electrical and Computer Engineering
Michigan State University, MI 48824, USA

Contents

1 Structure of LSTM

Long Short-Term Memory (LSTM) [1] is designed to avoid the vanishing gradient problem.

1. The forget gate decides what information should be thrown away or not (The value in this gate is close to 0 means forget, close to 1 means to keep/remember).
2. The input gate (update gate + tanh) is designed to update the cell state. Specifically, The update gate decides which value will be updated (0 is not important, 1 means important). Tanh function helps regulate the network by squish values between - 1 and 1.
3. The cell/memory state accepts information from the forget gate and the input gate.
4. The output gate decides what net hidden state should be.

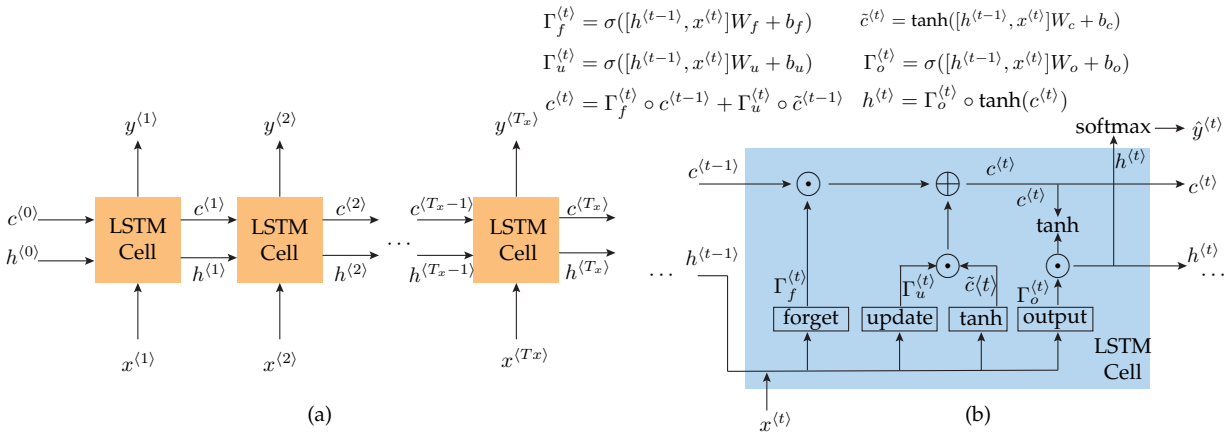

Figure 1: Structure of LSTM [2].

Assume the number of samples is N, the number of features is D, and the number of categories is M.

- $x^{\langle t \rangle}$.shape = (N, D)
- $h^{\langle t \rangle}$.shape = (N, H)
- $c^{\langle t \rangle}$.shape = (N, H)
- $y^{\langle t \rangle}$.shape = (N, M)
- W_f.shape = ($H + D$, H) b_f.shape = (1, H)
- W_i.shape = ($H + D$, H) b_i.shape = (1, H)
- W_o.shape = ($H + D$, H) b_o.shape = (1, H)
- W_c.shape = ($H + D$, H) b_c.shape = (1, H)
- W_y.shape = (H, M) b_y.shape = (1, M)

References

[1] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[2] Andrew Ng. CS230 Deep Learning. *Stanford University*, 2018