Coding-based tutorial about recurrent neural network (RNN)

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 RNN

The recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a temporal sequence [1], which allows it to exhibit temporal dynamic behavior. Derived from the feed-forward neural network, RNN can use its internal state (memory) to process variable length sequences of inputs. RNN was originally designed for language processing tasks, but it can also be applied to other circumstances. In this tutorial, we will employ RNN to solve the classification problem. We still choose MNIST as our dataset.

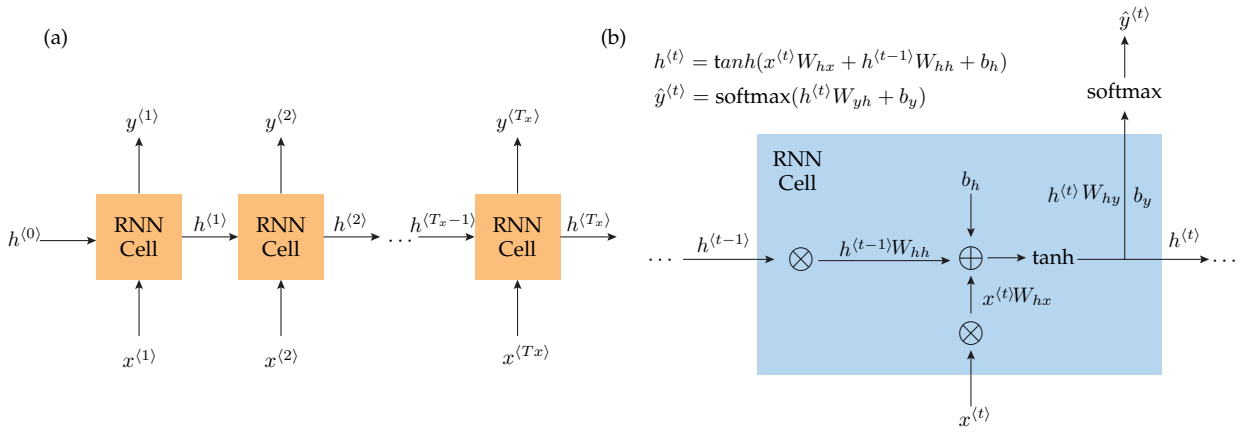

Figure 1: Structure of RNN.

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)
- $y^{\langle t \rangle}$.shape = (N,M)
- W_{hx}.shape = (D,H)
- W_{hh}.shape = (H,H) b_h.shape = (1,H)
- W_{hy}.shape = (H,M) b_y.shape = (1,M)

For the MNIST dataset, the ith row can be considered as a feature at time step t. Specifically, the first row is actually $x^{\langle 1 \rangle}$, the 28th row is $x^{\langle 28 \rangle}$. Therefore, in MNIST dataset, $N = 2000$, $D = 28$, and $M = 10$.

References

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

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