Recurrent neural networks

Liangjie Cao

13 May 2018

Today I learn Recurrent neural networks in Deep learning. The paper when backpropagation was first introduced, its most exciting use was for training recurrent neural networks (RNNs in Table. 1). For tasks that involve sequential inputs, such as speech and language, it is often better to use RNNs (Fig. 1 1). RNNs are very powerful dynamic systems, but training them has proved to be problematic because the backpropagated gradients either grow or shrink at each time step, so over many time steps they typically explode or vanish. [1]

RNNs, the paper also says once unfolded in time (Fig. 1), can be seen as very deep feedforward networks in which all the layers share the same weights. Although their main purpose is to learn long-term dependencies, theoretical and empirical evidence shows that it is difficult to learn to store information for very long. To correct for that, one idea is to augment the network with an explicit memory. The first proposal of this kind is the long short-term memory (LSTM) networks that use special hidden units, the natural behaviour of which is to remember inputs for a long time. Translating language, controlling robots, image analysis, document summarization, recognition of speech recognition, handwriting recognition, control of chat robots, prediction of disease, click rate and stock and synthetic music. That's all the function of LSTM.

LSTM networks have subsequently proved to be more effective than conventional RNNs, especially when they have several layers for each time step, enabling an entire speech recognition system that goes all the way from acoustics to the sequence of characters in the transcription. That's

amazing. How can we do next step? Only time will tell us.

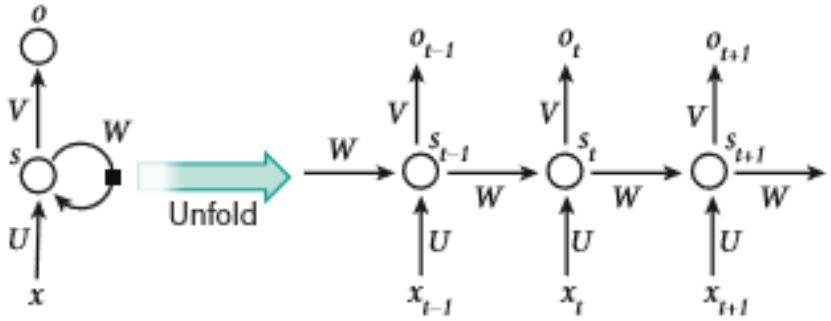

Figure 1: Model

1	Back Propagation Through Time(BPTT)
2	Real-time Recurrent Learning(RTRL)
3	Extended Kalman Filter(EKF)

Table 1: **Regular training algorithms**

References

[1] Yann LeCun *etal*. Deep learning. *Nature*, 521(28):9, 2015.