Non-Fixed Time Scale Recurrent Neural Net Paradigms and their Applications

1 Readings

- Li et al., Approximation of Dynamical Time Variant Systems by Continuous Time Recurrent Neural Networks, IEEE Transactions On Circuits And Systems II: Express Briefs Vol. 52, 2005, `http://www.ee.cityu.edu.hk/~twschow/pubs/papers/74.pdf`
- Ba et al., Using Fast Weights to Attend to the Recent Past, CoRR, 2016, `https://arxiv.org/abs/1610.06258`
- Neil et al., Phased LSTM: Accelerating Recurrent Network Training for Long or Event-based Sequences, arXiv, 2016, `https://arxiv.org/abs/1610.09513`

2 Abstract

Recurrent Neural Networks (RNNs) have become a state of the art choice for modelling dynamic systems and identifying patterns in the sequential data streams they produce. Current RNN models, however, operate at fixed time scales, and are therefore ill-suited for processing data produced by real continuous time dynamical systems which can produce sparse asynchronous streams of updates. In this presentation, we will examine three different RNN paradigms that break free of the standard fixed time update pattern and discuss their advantages and use cases.

Continuous RNNs: We will begin by discussing “Approximation of Dynamical Time-Variant Systems by Continuous-Time Recurrent Neural Networks”, which examines continuous RNNs in the context of RNN based automatic control systems. RNNs are increasingly commonly used for applications in signal processing, system identification, and intelligent control systems. The performance of these RNN-based controllers, however, depend on the ability of an RNN to model the underlying dynamical system. First. we show that the time trajectory of any time variant dynamic system can be approximated by a continuous RNN to any degree of accuracy. Subsequently, we will see several special forms of dynamical time variant systems which can be approximated by standard discrete time RNNs.

“Fast weight” RNNs: Next, we turn to “fast weights” RNNs, introduced in “Using Fast Weights to Attend to the Recent Past”. This paradigms introduces network variables which change more slowly than the network’s inputs but more rapidly than traditional network weights. These “fast weights” are computed along with the ordinary network updates at each update, and undergo a brief iterative settling process between updates while the remainder of the network’s hidden state is maintained as a sustained boundary condition. We will see

that these fast weights can provide a detailed memory of the recent past which is very helpful in certain sequence to sequence modelling.

Phased LSTM: Finally, we examine phased LSTM in "Phased LSTM: Accelerating Recurrent Network Training for Long or Event-based Sequences", also designed for the asynchronous event driven streams of information produced by dynamical systems. Phased LSTM extends the LSTM model by adding a new time gate. This time gate only allows memory cell state to be updated during certain periods of time. These periods of time are learned during the training process. By imposing sparse updates on cell state, phased LSTM substantially decreases decreases the total number of updates, allowing an undecayed gradient to be back-propagated through time, allowing for faster learning convergence.

Spotlight question: What are some practical applications of non-fixed time RNNs? Are the RNNs above capable of realizing these applications, and if not what capabilities do they lack?