A generalized LSTM-like training algorithm for second-order recurrent neural networks

The paper "A generalized LSTM-like training algorithm for second-order recurrent neural networks" by Derek Monner and James A. Reggia, published in *Neural Networks* in January 2012, introduced the Generalized Long Short-Term Memory(LSTM-g) training algorithm, which provides LSTM-like locality while being applicable without modification to a much wider range of second-order network architectures.

Different from LSTM, each unit in a network trained by LSTM-g has the same set of operating instructions, which only depends on its local network environment to determine if the role of memory cell, gate unit, both, or neither will be fulfilled. As every unit is trained in the same way in LSTM-g, the gate units are viewed as modulating the weights on connections between units. Therefore, LSTM-g provides more flexibility to network designers who want to explore arbitrary architectures where gates can temporarily isolate one part of the network from another.

Based on the algorithm and calculation, an LSTM-g network with the same architecture as an LSTM network will produce the same weight changes as LSTM training would, provided that peephole connections are not present. When peephole connections are added to the LSTM architecture, however, LSTM-g utilizes a source of error that LSTM training neglects: error responsibilities back-propagated from the output gates across the peephole connections to the associated memory cells, and beyond, which can enable better performance than the original algorithm.

Then, experiments are conducted to test the effectiveness of LSTM-g with original LSTM algorithm and architecture as a control comparison. The Distracted Sequence Recall task requires the network to recognize the symbols as targets and preferentially save them, along with temporal order information, in order to produce the correct output sequence. Using the standard architecture, LSTM-g trains the LSTM network architecture significantly faster than the original algorithm as both networks reached the performance criterion as expected. Then, the task is tested with three other networks with the same approach and parameters as in the previous experiment. The result present a speed advantage and the potential benefits of the wide range of customized architectures for LSTM-g over LSTM. At last, a Language Understanding task, in which a network is given an English sentence as input and is expected to produce a set of predicates that signifies the meaning of that sentence as output,is used to test LSTM-g. The result shows that the two-stage network trained with LSTM-g was able to achieve significantly better generalization performance and much more quickly than the standard LSTM network on average, reflecting the value of using LSTM-g to train customized architectures that traditional LSTM cannot.

Overall, LSTM-g provides the power, speed, and spatial and temporal locality of the LSTM algorithm, but unlike said algorithm is applicable to arbitrary second-order recurrent neural networks. Also, LSTM-g makes use of extra back-propagated error when applied to the canonical LSTM network architecture with peephole connections. The authors believe that it seems plausible that the locality properties of LSTM-g would lead to a better performance-to-computation ratio, resulting in faster convergence in terms of required computation time.