CommNets: Communicating Neural Network Architectures for Resource Constrained Systems

Prince M Abudu
Department of Computer Science
University of Oxford
15 Parks Rd, Oxford OX1 3QD, UK
prince.abudu@balliol.ox.ac.uk

Abstract

Applications that require heterogeneous sensor deployments continue to face practical challenges owing to resource constraints within their operating environments (i.e. energy efficiency, computational power and reliability). This has motivated the need for effective ways of selecting a sensing strategy that maximizes detection accuracy for events of interest using available resources and data-driven approaches. Inspired by those limitations, we ask a fundamental question: whether state-of-the-art Recurrent Neural Networks can observe different series of data and communicate their hidden states to collectively solve an objective in a distributed fashion. We realize our answer by conducting a series of systematic analyses of a Communicating Recurrent Neural Network architecture on varying time-steps, objective functions and number of nodes. The experimental setup we employ models tasks synonymous with those in Wireless Sensor Networks. Our contributions show that Recurrent Neural Networks can communicate through their hidden states and we achieve promising results.

Motivation

Resource Constrained Systems i.e. Wireless Sensor Networks and Embedded Systems are an integral part of human life owing to their successes in patient, environmental and wildlife monitoring (Bell et al. 2014; Lazarescu 2013; Dyo et al. 2012), among other application domains. Despite their demonstrated value, these systems are plagued with constraints on computing speed, memory size, energy efficiency and communication bandwidth.

We consider Recurrent Neural Networks (RNNs) in solving the problem of distributed communication. Current research has shown that RNNs can effectively model sequential data using their predictive power (Graves 2013; Sutskever, Vinyals, and Le 2014).

Owing to the great success of RNNs in various applications, many works have been motivated to make RNN architectures better suited for their implementation, for instance; Light RNN (Li et al. 2016) and Hierarchical Multiscale RNN (Chung, Ahn, and Bengio 2016). Work by (Zhang et al. 2016) introduces connecting architectures for RNNs. We harness the power of state-of-the-art RNNs to provide a communicating neural network architecture that can collective solve tasks and objectives in distributed settings - a common characteristic of Resource Constrained Systems, with an example shown in Figure 1.

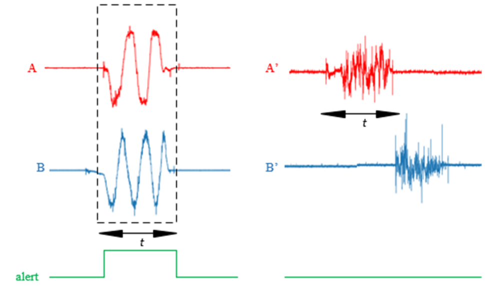

Figure 1: An illustration of event detection. Two (2) sets of sensor signals *A* and *B*, *A'* and *B'* depicting temporally varying phenomena are shown. An alert signal is generated for each set based on the occurrence of an episode of interest. In this work, we demonstrate that distributed nodes observing different signals characterized by dynamic phenomena can communicate - a capability which enables event detection applications in Resource Constrained Systems.

Proposed Contribution

We propose a data-driven, decentralized approach to communication that employs nodes observing unique data over defined periods of time, and at each processing step sharing their hidden states with neighbouring nodes to ultimately learn the dependencies in the data and the observed function. To achieve this, we alter the architecture of traditional RNNs on the lowest level of implementation possible, to enable communication through time *(t)* and space (number of nodes) and empirically assess our results.

Our proposed architecture implements homologous and unidirectional RNNs with shared weights implemented in PyTorch. All layers are ReLu activated and parameters are optimized using SGD (Rumelhart, Hinton, and Williams 1986) with a learning rate of 0.0005. The model uses the default Adam (Kingma and Ba 2014) optimizer in PyTorch. Communicating nodes have 8 latent states and are fully connected.

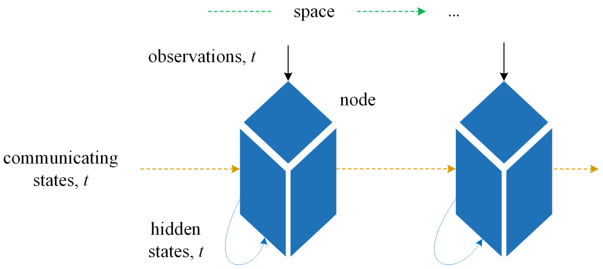

Figure 2: Nodes in a Communicating RNN architecture

Results

Our results are promising as we are able to demonstrate the effectiveness of recurrent computation in a distributed sense for our defined objectives. We make empirical analyses of our Communicating RNN architecture for Objective functions; *Value (T1)*, *Range (T2)* and *Compare (T3)*, with convergence shown in Figure 3 and results in Figure 4.

$$y_t = \begin{cases} 1 & \text{if } x_t^1 > p \;\; AND \;\; x_t^2 < q \\ 0 & \text{otherwise.} \end{cases} \qquad (T1)$$

$$y_t = \begin{cases} 1 & \text{if } (x_t^1 < a \;\; AND \;\; x_t^2 < a) \\ & OR \quad (x_t^1 > b \;\; AND \;\; x_t^2 > b) \\ 0 & \text{otherwise.} \end{cases} \qquad (T2)$$

$$y_t = \begin{cases} 1 & \text{if } x_t^1 > x_t^2 \\ 0 & \text{otherwise.} \end{cases} \qquad (T3)$$

where x_t^1, x_t^2 are observations in different sequences, y_t is the network target, *p, q* are pre-defined threshold values, and *a, b* are pre-defined range values.

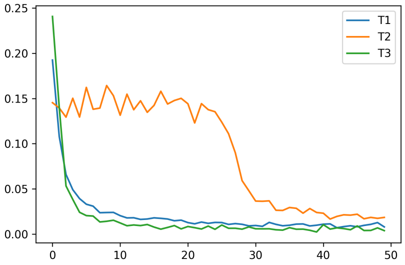

Figure 3: Training losses for all tasks up to *M*=50000 samples

Conclusion

Our evaluations show that we can solve objective tasks in a distributed fashion using a data-driven communication paradigm that harnesses the power of RNNs. In addition to our experimental results, we provide empirical basis for the instatiation of Communicating RNN architectures. Our next steps will investigate more complex tasks and consider the impact of scale, both in space and time.

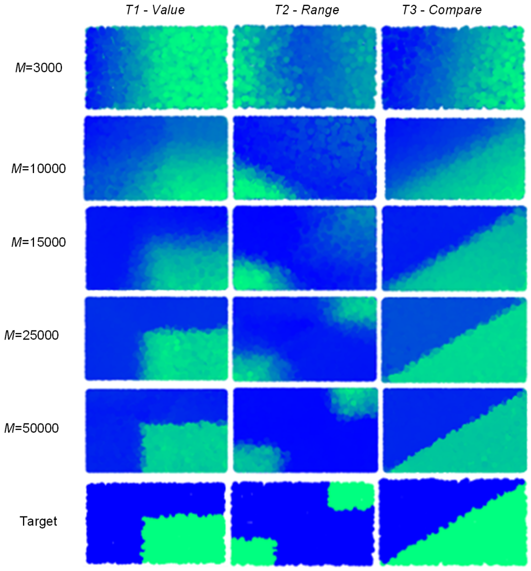

Figure 4: Scatter plots of targets and model predictions showing color-valued dependencies between x_t^1, x_t^2 and predictions.

References

Bell, A.; Rogers, P.; Farnell, C.; Sparkman, B.; and Smith, S. C. 2014. Wireless patient monitoring system. In *2014 IEEE Healthcare Innovation Conference (HIC)*, 149–152. IEEE.

Chung, J.; Ahn, S.; and Bengio, Y. 2016. Hierarchical Multiscale Recurrent Neural Networks. *CoRR* abs/1609.01704.

Dyo, V.; Yousef, K.; Ellwood, S. A.; Macdonald, D. W.; Markham, A.; Trigoni, N.; Wohlers, R.; Mascolo, C.; Pásztor, B.; and Scellato, S. 2012. WILDSENSING. *ACM Transactions on Sensor Networks* 8(4):1–33.

Graves, A. 2013. Generating Sequences With Recurrent Neural Networks. *CoRR* abs/1308.0850.

Kingma, D. P., and Ba, J. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980.

Lazarescu, M. T. 2013. Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 3(1):45–54.

Li, X.; Qin, T.; Yang, J.; and Liu, T.-Y. 2016. LightRNN: Memory and Computation-Efficient Recurrent Neural Networks. *CoRR* abs/1610.09893.

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

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks.

Zhang, S.; Wu, Y.; Che, T.; Lin, Z.; Memisevic, R.; Salakhutdinov, R.; and Bengio, Y. 2016. Architectural Complexity Measures of Recurrent Neural Networks. *CoRR* abs/1602.08210.