Optimizing Recurrent Neural Networks (RNN) for Xeon Phi Cluster Totient

CS5220 Final Project Proposal for Amiraj Dhawan (ad867), Saurabh Netravalkar (sn575), Sania Nagpal (sn579)

Recurrent Neural Network:
Recurrent Neural Network (RNN) is one of an algorithm of Deep Learning. It differs from the standard feed-forward neural networks because it can include directed cycles between neurons which gives it the capability of maintaining internal state of the network and allows it model dynamic temporal behavior. The biggest advantage is that RNN's can work with arbitrary sequence of inputs unlike the normal rigid neural networks. These networks have been shown to work better on Handwritten recognition, Language Learning applications than other network architectures.

Architectures: There are numerous architectures proposed till date for Recurrent Neural networks but for the scope of the project I will mostly focus on the Long Short-Term Memory network (LSTM) which have been shown to perform slightly better than the other architectures.

Training: Mostly 2 methods are used Gradient Descent or Non-Linear Global Optimization problem (usually optimizing mean squared error). This can be optimized and parallelized.

Motivation: RNN's are renowned to be difficult to scale up and practically used. The training becomes really costly because of the cycles present in the neurons. This will be an interesting problem to parallelize and optimize.

Testing: A library called DL4J (for Hadoop and Spark) is available and contains an implementation of LSTM RNN which can be used as a target to beat. Though I am not sure how this will be possible to setup up on the cluster and execute.