Introduction

Restricted Boltzmann Machines (RBMs) - the building block for newly popular Deep Belief networks (DBNs) - are a promising new tool for machine learning practitioners. However, future research in applications of RBMs is hampered by the considerable computation that training requires. We have designed a novel architecture and FPGA implementation that accelerates the training of general RBMs in a scalable manner, with the goal of producing a system that machine learning researchers can use to investigate even larger networks. Our current (single FPGA) design uses a highly efficient, fully-pipelined architecture based on 16-bit arithmetic for performing RBM training on an FPGA. Single-board results show a speedup of 25-30X achieved over an optimized software implementation on a high-end CPU. Current design efforts are for a multi-board implementation.

Restricted Boltzmann Machine

![RBM Structure](image)

An RBM is a two layer neural network with all-to-all connections between the layers.

Figure 1. RBM Structure (above)

A Deep Belief Network is a multi-layer generative model. The network is first learned with all the weights tied, which is equivalent to an RBM. Then it freezes the first layer and learns the remaining weights, which is also equivalent to another RBM.

RBM Training Procedure

Contrast-divergence learning to simplify infinite alternating Gibbs sampling

\[ \Delta w_{ij} = \varepsilon (\langle v_j h_i \rangle - \langle v_j h'_i \rangle) \]

Figure 3. RBM Training (from Hinton’s tutorial at NIPS’07)
The training begins with the data at the visible layer computing the probabilities of the hidden layer. The hidden layer is updated and stochastically fires to reconstruct the visible layer. The hidden layer is recomputed from the reconstructed visible layer. The weights are updated based on the difference between the visible-hidden product of the original training example and the visible-hidden product of the reconstructed data.

Experimental Platform

Stackable Altera Stratix III FPGA board with DDR2 SDRAM interface.

![Experimental Platform](image)

Figure 4. Terasic DE3 Fast Prototyping Board

The left image shows the DE3 board. It has an Altera Stratix III FPGA with high speed I/O interface for communications with multiple other boards. It also includes a DDR2 SO-DIMM interface, USB JTAG interface, and USB 2.0 interface. The right image illustrates multiple DE3 boards connected in a stacked manner.

Implementation Details

Single FPGA implementation of RBM is developed. Multi-FPGA version is being designed at the moment.

Figure 5. Overall System

Single FPGA consists of a CPU, DDR2 controller, and an RBM module. These components are connect- ed via Altera’s Avalon interface.

Performance Results

The RBM module runs at 200MHz. Comparison was made against one core of Intel Core2 2.4GHz processor running on MATLAB.

![Performance Results](image)

Extending to Multiple FPGA System

Two issues being tackled

1. The weight matrix increases as \(O(n^2)\) with the number of nodes. Thus, the weight matrix no longer fits on-chip and will have to be streamed from DRAM. Using a batch size of greater than 16, we can exploit the data parallelism to reduce the required bandwidth that is feasible for DDR SDRAM.

2. The single FPGA scheme issues a broadcast of visible data every cycle. If we extend this to multiple FPGAs, the height of the board stack may limit the maximum clock rate driving the RBM module.

Conclusion

Deep Belief Nets are an emerging area where Restricted Boltzmann Machine is at the heart of it. FPGAs can effectively exploit the inherent fine-grain parallelism in RBMs to reduce the computational bottleneck for large scale DBN research. As a prototype of building a fast DBN research machine, we implemented a high-speed configurable RBM on a single FPGA. The RBM has shown approximately 25X speedup compared to a single precision software implementation of the RBM running on an Intel Core2 processor.

Interacting with the Stanford’s AI research group, our future implementation of multiple FPGA boards is expected to provide enough speedup to attack large problems that remained unsolved for decades.

Sang Kyun Kim, Lawrence McAfee, Peter McMahon, Kunle Olukotun