Engineering 90 Topic Selection Memo
Computing Hardware for Accelerated Training of Neural Networks

David German '08

September 27, 2007

1 Introduction

This document proposes an Engineering 90 Senior Design Project topic. The objective of the project is to develop computing hardware that accelerates the training of an artificial neural network. This memo will explain the value of such a device, discuss anticipated technical challenges, and estimate the resources required to undertake the project. It is organized for easy expansion into a final project proposal.

2 Technical Discussion

2.1 Neural Networks

This section presents a conceptual overview of neural network theory that is common knowledge in the field of artificial intelligence. It does not present the underlying mathematics in detail, since such discussion is readily found in reference material. [1]

An artificial neural network is a function approximator composed of nodes that mimic the function of biological neurons. Each node receives some input stimulus, p_{net}, and generates an output stimulus determined by its *activation function*, $f_a(p_{net})$. The input stimulus is the weighted sum of the output of nodes that *feed forward* into it. If the function to be approximated has m inputs and n outputs, the neural network will have m special input nodes that output an element of the input vector, and n special output nodes that do not feed forward to any nodes. Nodes that are neither input nor output are *hidden*.

The process of adjusting a neural network to approximate the desired function is called *training*. Training requires a set of tuples $(\vec{i}, \vec{o})$ that are in the function to be approximated. Input $\vec{i}$ is applied to the network as an input, the output is computed, and the error between the output and $\vec{o}$ is used to modify the feed-forward weights by gradient descent. This process is repeated with each tuple in the training set until the change in the network falls below an arbitrary threshold. Under some models, the network may incrementally train and grow. If the network has learned the function of interest well, the application of an input that was not in the training set will yield a good approximation of the desired output. Because the function is entirely learned by example, it need not have a convenient mathematical form; a neural network could, for instance, be trained to output a 1 for inputs representing images that contain a pizza, and a 0 for all other inputs.

2.2 Possible Advantages of Specialized Hardware

General-purpose microprocessors are designed to perform computations in serial. In the past few years, CPUs with two or four parallel execution cores have been commercialized, but even so the bulk of their computational power rests in completing each operation quickly, not in completing many operations at once. This is appropriate for most software applications, in which an input to one instruction is very often the output of its predecessor. Massively parallel execution of such code is impractical: the overhead cost of hardware to check for dependencies and schedule parallel execution correctly is prohibitive.

Neural network training, however, contains a large number of computations that are guaranteed to be free of dependency. We have already seen that computing the input stimulus to node j requires a multiply-accumulate operation, the result of which does not affect the input stimulus to any node not fed by j. In fact, this computation is simply the convolution sum familiar to DSP filter designers, and widely performed with specialized parallel hardware.

In two training algorithms of particular interest to this project, the *back-propagation algorithm* and the *cascade correlation algorithm*, the computation of weight adjustments is also highly parallelizable. If a back-propagation network is divided into layers that are fully connected, such that every node in a given layer feeds every node in the next, error adjustment for back-propagation is matrix arithmetic. Cascade correlation adds one hidden node to the network at a time. The node is fed by all pre-existing input and hidden nodes, and feeds all output nodes. Its input weights are trained by gradient descent to maximize correlation with

[1] `http://www.willamette.edu/~gorr/classes/cs449/backprop.html` (Orr, accessed 9/25/2007) offers a summary that is both straightforward and rigorous, and was quite helpful in the preparation of this memo.

remaining error; all output weights are then retrained to minimize overall error. The correlation operation in the input training step is another task familiarly parallelized in DSP. Since input weights are frozen for the output retraining step, the error computation and weight adjustment is simply parallel scalar arithmetic.

2.3 Feature Overview of Proposed Hardware

The parameters that define a neural network are numerous, and a single piece of hardware cannot implement all combinations of interest to AI researchers. This following list identifies the key classes of constraint on project scope. Appropriate values for each class remain to be determined - the timeline for finalizing them is discussed in Section 3.1 - but once selected, these constraints should ensure that the project can realistically be accomplished in the available time. Note that it has not yet been decided whether the circuit will perform digital or analog arithmetic. This is an implementation question, not a feature constraint, but it does complicate the specification of the feature constraints.

Algorithm Specific The device will implement back-propagation or cascade correlation, but not both.

Depth Limited If back-propagation is implemented, there will be a fixed number of layers and layers per node, and the network will be fully connected. If cascade correlation is implemented, there will be a fixed number of hidden nodes allowed.

Fan Limited The device will accept a fixed, finite number of inputs and outputs. If the function to be approximated has a smaller fan, the device will learn it correctly if the excess inputs and outputs are always 0 in the training data.

Single Product The project timeframe does not allow for VLSI fabrication. Therefore, if a custom VLSI design (analog or digital) is pursued, the device will be developed and tested entirely in simulation. Alternatively, the device could be developed using off-the-shelf analog components and/or an FPGA. In this case, simulation will be used to validate subsytems, but the device will actually be produced, connected to a computer interface, and validated by actually training neural nets.

Precision Limited If the device performs analog arithmetic, precision will only be guaranteed to a threshold determined by the components used. If the device performs digital arithmetic, a bit width will be specified, and all arithmetic will be fixed-point.

No constraint is specified for the activation function. It is a design requirement that the device user be able to load an arbitrary activation function (and activation function derivative, if required for the algorithm) into a table.

3 Project Plan

3.1 Finalize Feature Set and Implementation Strategy

Objective Fully specify all the feature constraints discussed in section 2.3. Select a primarily analog or primarily digital design strategy.

Approach Read existing literature that investigates neural network hardware. There appears to be literature discussing both algorithms of interest, and addressing depth, fan, and precision limitations. Informed by this background study, select features for a feasible, but significant and challenging task.

Output Final specifications and a critical-path timeline for implementation.

Deadline Draft to advisor Friday, November 9 by 5:00 pm. Final version to department Friday, November 16 by 3:00 pm per customer requirements.

3.2 Make Supply Decisions

Objective Given final feature set, determine specific parts to purchase.

Approach If the project is to be completed in simulation, only office supplies are required and the total cost is minimal. If actual hardware will be constructed, the costs are potentially sizable. If they exceed the nominal $200 budget, I will attempt to convince the department that the materials will be useful to it in the future.

Output A final bill of materials and total cost.

Approximate Deadline Wishlist to advisor and appropriate authorities by Monday, November 26. Final output to department Wednesday, December 5 by 12:00 pm per customer requirements.

3.3 Design and Test Small-Scale Prototype

The details of this step will be included in the final proposal due November 16.

3.4 Productize and Deliver

The details of this step will be included in the final proposal due November 16.

4 Project Qualifications

I am a double major in Engineering and Computer Science with a particular interest in embedded systems and digital hardware. My past and present coursework includes Digital Systems, Computer Architecture, Compiler Design and Construction, Electronic Circuit Applications, VLSI Design, and Operating Systems, a strong academic background for this project. My practical experience with embedded systems also includes a three-month internship with The Boeing Company in 2006, where I developed customer-requested modifications to the embedded software used for command and control of the International Space Station.

5 Project Cost

5.1 Labor

I estimate this project will consume 500 hours of my own time. If an actual hardware implementation is undertaken, I will additionally request about 3 to 5 hours of Ed Jaoudi's time in small increments for assistance finding components and deploying tools. Prof. Tali Moreshet has agreed to advise my project. The amount of her time I will request depends largely on the magnitude of unforseen difficulties, but is unlikely to exceed 5% of the time I invest myself. I may occasionally request input of other professors of the College, especially Prof. Erik Cheever and Prof. Lisa Meeden, in their areas of expertise.

5.2 Materials

The details of this section will be included in the final bill of materials due December 5.

6 Acknowledgments

I thank Prof. Tali Morshet for agreeing to advise this project, Prof. Lisa Meeden for offering to lend her expertise in artificial intelligence when necessary, Prof. Nelson Macken for instruction in writing a project proposal, and George Dahl for suggesting the project objective.