Introduction

• What is a neural network?
  ○ Collection of interconnected neurons that compute and generate impulses
  ○ Components of a neural network include neurons, synapses, and activation functions
  ○ Neural networks modeled by mathematical or computer models are referred to as artificial neural networks
  ○ Neural networks achieve functionality through learning/training functions
A neuron is commonly modeled by synapses, represented by $W_i$, connected to input signals, represented by $X_i$. These inputs are either inputs to the network, or outputs from previous neurons. A neuron will sum these inputs and calculate the output based on an activation function.
Network Topologies

- Network topologies describe how neurons map to each other. Many artificial neural networks (ANNs) utilize network topologies that are very parallelizable.
Most Commonly Used Algorithms

- **Multilayer feedforward networks**
  - Layered network where each layer of neurons only outputs to next layer

- **Feedback networks**
  - Single layer of neurons with outputs that loops back to the inputs

- **Self-organizing maps (SOM)**
  - Forms mappings from a high dimensional space to a lower dimensional space (one layer of neurons)

- **Sparse distributed memory (SDM)**
  - Special form of a two layer feedforward network that works as an associative memory
Learning/Training

- The key to neural networks is weights. Weights determine the importance of signals and essentially determine the network output.
- Training cycles adjust weights to improve the performance of a network. The performance of an ANN is critically dependant on training performance.
- An untrained network is basically useless.
- Different training algorithms lend themselves better to certain network topologies, but some also lend themselves to parallelization.
Why Neural Networks?

There are about $10^{16}$ synapses in the human brain each performing multiple operations/s. This means that the brain is processing approximately $10^{16}$ operations/s while only dissipating a few watts. ANNs aim to mimic this behavior in hopes of achieving similar performance. The brain's ability to interpret raw sensory data and perform behavior based on it outperform any modern design.
Why Parallel Computing?

- Training and evaluation of each node in large networks can take an incredibly long time
- However, neural networks are "embarrassingly parallel"
  - Computations for each node are generally independent of all other nodes
Possible Levels of Parallelization

Typical structure of an ANN:

- For each training session
  - For each training example in the session
    - For each layer (forward and backward)
    - For each neuron in the layer
      - For all weights of the neuron
        - For all bits of the weight value
Possible Levels of Parallelization

So there are at least 6 possible ways to achieve parallelism, from coarse grain to fine grain:

- Training session parallelism
  - Training example parallelism
    - Layer parallelism or Forward-Backward parallelism
  - Neuron parallelism
    - Weight parallelism
    - Bit parallelism

Even coarser parallelism: Network level [4]
Matching the Degree of Parallelism to the Network

- The multiple layer structure of many feedforward networks and their back-propagation training technique severely limits the speedup of parallel neuron computations
  - Training set or network level parallelism more practical
- Neuron level parallelism more effective for single layer networks
  - SOM or some recurrent networks
Network Level vs Neuron Level Parallelism

Network Level Parallelism [7]

Neuron Level Parallelism [7]
Computational Considerations

- **Main operation is matrix-vector multiplication**
  - For an N-node layer, there are $N^2$ scalar multiplications and N sums of N numbers
  - Need good support for multiply or multiply-and-add

- **Sigmoid function**
  - In many models, neuron output is determined by a function such as $f(x) = 1/(1+e^{-x})$
  - For efficiency, needs to be approximated via lookup table or a piecewise linear model
Communication Considerations

- High degree of connectivity and large data flows are characteristic features of ANNs.
- Structure and bandwidth of communication are very important:
  - Depends on degree of parallelism.
- Broadcast or ring communication is very efficient for ANNs.
More processors is not always better

<table>
<thead>
<tr>
<th>Used Processors</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>neuron distribution among available processors</td>
<td>1x25</td>
<td>1x13</td>
<td>1x9</td>
<td>1x7</td>
<td>5x5</td>
<td>1x5</td>
<td>4x4</td>
<td>1x4</td>
<td>7x3</td>
<td>7x3</td>
<td>5x3</td>
<td>3x3</td>
<td>1x3</td>
</tr>
<tr>
<td>neuron distribution among available processors</td>
<td>1x12</td>
<td>2x8</td>
<td>3x6</td>
<td>5x4</td>
<td>3x3</td>
<td>7x3</td>
<td>2x2</td>
<td>5x2</td>
<td>8x2</td>
<td>11x2</td>
<td>1x1</td>
<td>12x2</td>
<td>1x1</td>
</tr>
</tbody>
</table>

Division of labor for a 5x5 SOM (25 neurons). Bold numbers show the required number of sequential loops [5]

Additionally, the effort and time to transfer data between processors increases with each step.
Example: Parallel Back-propagation training [7]

Training-set parallelization
- Each processor performs full sequential training of the network for one portion of the training set
- Main processor performs averaging procedure on results to extract final weight table
Example: Parallel SOM [7]

Partitions the neuron grid into a set of nodes that can be processed independently of each other.
Results

Execution time vs. number of processors in MPI: Neuron-level parallelism[3]

Execution time vs. number of processors in MPI: Training-Set Parallelism[3]
Results (cont...)

Training Time of 25-node SOM depending on the number of used processors [5]
Results (cont...)
References