Why Hardware NNWs

- The performance of conventional von Neuman processors, e.g. the Intel Pentium series, continues to improve dramatically.
- So why bother to implement neural network algorithms in special hardware??

SPEED!

- Even the fastest sequential processor cannot provide *real-time* response and learning for networks with large numbers of neurons and synapses.
- Parallel processing with multiple simple processing elements (PE's) can provide tremendous speedups.
- When the particular task at hand does not require super fast speed, most designers of neural network solutions find a software implementation on a PC or workstation a satisfactory solution. .

Specialized chips

- Specialized applications,however, can also motivate the use of hardware NNs:
- Cheap dedicated devices, such as those for speech recognition in consumer products.
- Analog/neuromorphic devices, such as silicon retinas, that directly implement the desired functions

Applications

- NN's, despite all appearances to the contrary, are appearing in ever increasing numbers of real world applications and are making real money:
- OCR (Optical Character Recognition)
- Caere Inc ($3M profit on $55M revenue in 1997) "*OmniPage Pro 6.0 significantly increases accuracy with its exclusive Quadratic Neural Network(TM) (QNN) technology, an enhancement to its industry-leading OCR engine...*"
- Data Mining
- HNC ($23M profit on $110M revenue in 1997). Their flagship product is Falcon. "*Falcon is a neural network-based system that examines transaction, cardholder, and merchant data to detect a wide range of credit card fraud...*".

OCR

- These days a purchase of a new scanner typically includes a commercial OCR program.
- The algorithms used are proprietary but most OCR programs are believed to use NNWs. (Calera, started in 1986, did not admit to using NNW in its OCR programs until 1992 when Caere began advertising the use of them in its OCR products).
- However, the OCR example also illustrates why one cannot claim NNWs are conquering the world.
- One does not feed the pixels of the picture file into a single giant NNW and out pops the text.

Amdahl's Law

- The OCR application discussed above is a good example of the problems facing the designer of NNW hardware.
- Many steps must be executed to achieve the goal of converting the image of text to a text file.
- Amdahl showed that only when a substantial part of a task can be parallelized, is it worth doing.
- For example, suppose that 50% of the operations in your OCR program could be executed in a parallel system of infinite speed.
- The total speedup of the program is still only a factor of 2.
- Only when 90% or more of the program execution can be made parallel, do speedups of 10 or greater occur.

Hardware vs Software

- Implementing your Neural Network in special hardware can entail a substantial investment of your time and money:
- the cost of the hardware
- cost of the software to execute on the hardware
- time and effort to climb the learning curve to master the use of the hardware and software.
- Before making this investment, you would like to be sure it is worth it.

Hardware vs Software

- A scan of applications in a typical NNs conference proceedings will show that many, if not most, use feedforward networks with 10-100 inputs, 10-100 hidden units, and 1-10 output units.
- A forward pass through networks of this size will run in millisecs on a Pentium.
- Training may take overnight but if only done once or occasionally, this is not usually a problem.
- Most applications involve a number of steps, many not NNs related, that cannot be made parallel. So Amdahl's law limits the overall speedup from your special hardware.
- Intel series chips and other von Neuman processors have grown rapidly in speed, plus one can take advantage of huge amount of readily available software.
- One quickly begins to see why the business of Neural Network hardware has not boomed the way some in the field expected back in the 1980's.

The Hardware Designer's Dilemma

- The company, or research group, deciding whether to build a hardware NNW system clearly faces some tough questions. It will typically take at least 2 years to design, manufacture and debug a chip and a card to run it.
- It may look great on paper now, but will the system still be several times the speed of Intel's processor 2 years from now?
- The real dilemma, though, is how to fight Amdahl:
- Build a general, but probably expensive, system that can be re-programmed for many kinds of tasks? - e.g. Adaptive Solutions CNAPS
- Or build a specialized but cheap chip to do one thing very quickly and efficiently? - e.g. IBM ZISC
- Different designers, such as IBM and Adaptive Solutions, have had different answers.

The User's Dilemma

- So the user of a NNW must decide if the benefits of implementation in hardware are sufficient to overcome Intel and Amdahl.
- Hardware NNW are of greatest benefit for applications that:
- Need high speed, especially for the learning phase.
- Need a cheap, simple dedicated NNW to be embedded in large number of systems.
- Need the special functional capabilities obtained from close emulation of biological systems.

Applications

- While not yet as successful as NNs in software, there are in fact hardware NNs hard at work in the real world. For example:
- OCR (Optical Character Recognition)
- Adaptive Solutions high volume form and image capture systems.
- Ligature Ltd. OCR-on-a-Chip
- Voice Recognition
- Sensory Inc. RSC Microcontrollers and ASSP speech recognition specific chips.

NN Chips

- Though NNW's have been built with discrete components, the heart of the modern hardware NNW is a VLSI chip.
- The basic categories are:
- Digital
- Analog
- Hybrid
- Other major distinguishing features:
- Neural Network architecture(s)
- Programmable or hardwired network(s)
- On-chip learning or chip-in-the-loop training
- Low, medium or high number of parallel processing elements (PE's)
- Maximum network size.
- Can chips be chained together to increase network size.
- Bits of precision (estimate for analog)
- Transfer function on-chip or off-chip, e.g. in lookup table (LUT).
- Accumulator size in bits.
- Expensive or cheap

Ratings

- Comparing hardware NNW performance can be tricky.
- The most common performance measure is the Connection-Per-Sec (CPS) rate, defined as the number of multiply and accumulate operations per sec during recall, or forward, processing.
- However, a device with only 4 bit weights and inputs may not always be considered superior to another device that has a lower CPS but, say, 16 bit weights and inputs.
- For a measure of training speed, the Connection-Update-Per-Sec (CUPS) rate is sometimes provided.
- The learning rate also depends, of course, on the algorithm implemented. A chip with a RBF algorithm could have a slower learning pass than a feed-forward chip trained with back-propagation, but learns with far fewer passes.
- Unfortunately, just as for software network algorithms, there is no standard benchmark datasets on which hardware networks are tested.