An optical face recognition system

Demetri Psaltis, Yong Qiao, Hsin-Yu Li

Electrical Engineering
California Institute of Technology
Pasadena, California 91125

ABSTRACT

We describe a two-layer neural network using holographic optical disks as the interconnection weights. Such a system can be used to implement one two-layer with large number of hidden units, or several two-layer networks with smaller number of hidden units.

1. INTRODUCTION

There has been much interest in recent years in using neural networks to solve problems that are not "well defined" and therefore difficult to deal with using conventional algorithmic approaches. Instead of trying to having explicit instructions in the program on what to do, the system "learns," or "generalizes," from examples and adjusts parameters (weights) according to some rule until the system gives the desired input/output relations. Roughly speaking, a neural network consists of many simple processing units (neurons) and a large number of interconnections between the processing units. Much work has been done on both theoretical studies and hardware implementation, the three main approaches for implementation being: computer simulation, (micro) electronics, and optics.

In spite of the promises of neural networks, there have been many difficulties in building systems for "real-life" applications. One of the reasons is because of the difficultly in doing simulations on computers or building electronic hardware as the number of neurons and interconnections become larger. In addition it is desirable to have real-time responses, which is also hard to implement. Having large number of interconnections between neurons is difficult to achieve using VLSI technology on semiconductor chips, but is very well suited with optical interconnects using holography.

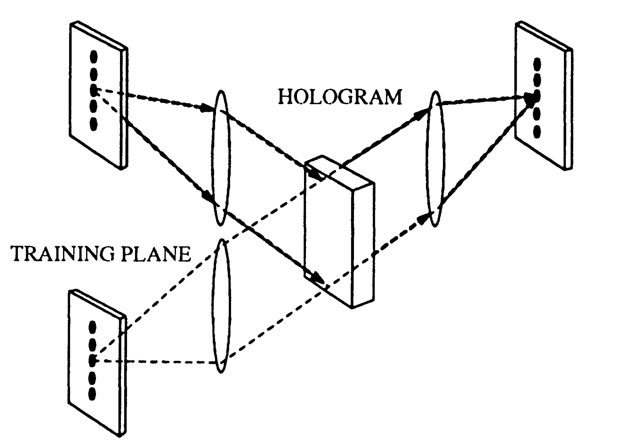

Figure 1. Optical interconnection using holograms.

Figure 1 shows the basic setup for a holographic interconnect.[1] By recording a hologram between the light pattern coming from the training plane and the input plane, light emitting from the input plane diffracts into the output plane according to the hologram written on the recording media. In this way, interconnection strengths are stored on the holographic media, and can be modified if the recording media is alterable (e.g., photorefractive crystals). Very large number of interconnections can be programmed this way, and the propagation of signals from one layer of neurons to the other is done in parallel.

2. MULTILAYER NEURAL NETWORKS

Given the building block shown in Figure 1, they can be cascaded to build multilayer networks[2]. The system we envision uses optoelectronic chips with detectors and light modulators or emitters (laser diodes or LEDs) for neurons, and holograms (using photorefractive crystals[3] or photopolymers) as the interconnections. The number of connections that can be stored from one neural plane to the other will depend on the space-bandwidth product of the optical components, the dynamic range of the recording media, and also the geometry of the recording media.

As the problems that we want to solve become more complicated, it is often necessary to have more and more hidden units and connections than possible on one particular location, while the number of input units and output units remain the same. Note that for practical hardware implementations, the maximum number of hidden unit devices will be the same order as that of the input and output units. One way around this problem is to handle a relatively small number of hidden units at a time.

The situation is shown in Fig. 2. The input units broadcast their signal through a set of connection weights to the set of hidden units, labeled A, which in turn send the processed (summed and thresholded) signal through a second layer of weights to the output units. The output units now store the received signal in some internal memory. Next the system switch to another set of hidden units labeled B. The input units broadcast their signal through another set of weights to hidden unit set B, and the processed signal is again sent through a second layer of weights to the output units. The signal arriving at the output units are then added to the values already stored in the internal memory. This process can be continued till all the sets of hidden units (A, B, C, ...) are exhausted. The final accumulated signal in the internal memory of the output units are the same as if all the hidden units were processed at the same time. In this way, it is possible to handle arbitrary number of hidden units by sacrificing parallelism.

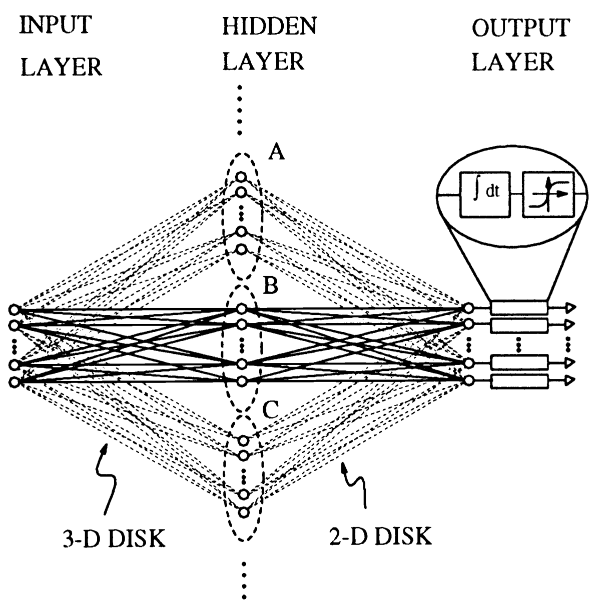

Figure 2. Two-layer neural network with arbitrary number of hidden units.

3. OPTICAL IMPLEMENTATION

It is of course not necessary to physically have all the hidden units present. All that is necessary is to be able to recall the sets of weights one after the other. Note that information needs to be stored in the output units, but

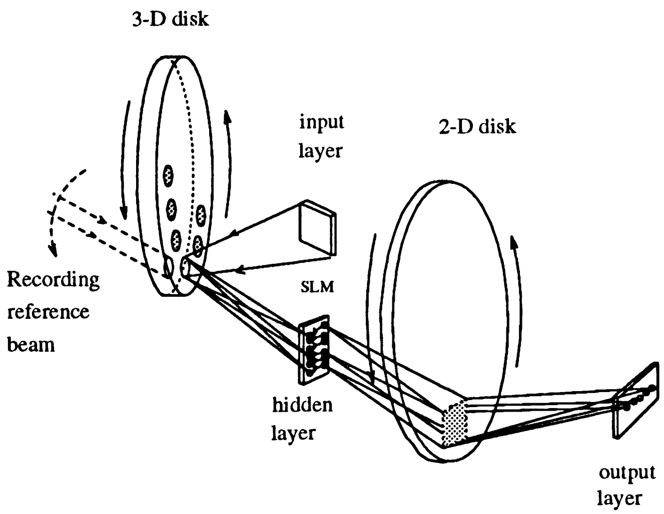

Figure 3. Optical implementation of a two-layer network.

not in the hidden units. The architecture we propose uses the structure in Fig. 1 as the building block to implement a two-layer optical neural network with large number of hidden units. The setup is shown in Fig. 2.

As in Fig. 1, the weights are stored as holograms (volume or planar). But here different sets of weights are stored on different locations of the recording media, which is made in the form of a disk. To access a different set of weights, the disk rotates to the appropriate location. The number of weights that can be stored depends on the number of recording locations on the disk. Given a disk size of 5 cm diameter and recording size of 2×2 mm^2, we can store about 100 locations along the periphery of the disk. Assuming that we can have 1000×1000 units on our neural planes, we can store 1000^3 independent weights[4] per location (assuming we record volume holograms). This gives us 10^{11} independent weights per layer.

One of the advantages of neural networks is the speed gained by having parallel operations. In the architecture we propose, some speed is sacrificed by having part of the operations done sequentially. The more groups of hidden units we process, the slower the system. The speed at which the system operates will be limited mainly by the speed at which we can rotate the disk.

Consider a rotational speed of 1,800 rpm, which is typical for rotating polygon mirrors. This requires that the input neural plane changes data no faster than 30 frames/sec, which is standard video rate. The neurons at the hidden layer and output layer, however, have to process data faster than about 0.3 msec since we are assuming that we go through 100 recording locations per revolution. We will assume that the diffracted light from the holograms give enough energy in this interval for the detectors to detect. We then have 3×10^{10} weights processed per second.

4. DISCUSSION AND CONCLUSION

For many applications, the input is an image. For example, if the network were trained to recognize a face, we would want the system to have the same response if the face changed position in the scene. Shift invariance is therefore a desirable feature to have. A natural way of implementing this is to use planar (2-D) holograms, instead of volume (3-D) holograms. The trade-off is that the storage capacity of the holograms decreases by a factor of N, where N^2 is the number of pixels or neurons on the neural plane.

In conclusion, the proposed system implements a two-layer network with very large number of hidden units. Note however that this system can also be used to implement multiple two-layer networks where we use only a few different recording locations to store the weight of one network. By controlling the time at which we read out the accumulated data at the output neurons, the system can be configured to simulate one large two-layer networks, or several smaller two-layer networks (with less number of hidden units). This extra flexibility enables us to implement different systems as required.

5. ACKNOWLEDGEMENT

The support for this work by DARPA and AFOSR is gratefully acknowledged.

6. REFERENCE

1. D. Psaltis, D. Brady, X.-G. Gu, and S. Lin, "Holography in Artificial Neural Networks," *Nature* **343**, No. 25, 325–330, 1990.
2. K. Wagner and D. Psaltis, "Multilayer Optical Learning Networks," *Appl. Opt*, **26**(23), 5061–5076, 1987.
3. D. Psaltis, D. Brady, and K. Wagner, "Adaptive Optical Networks Using Photorefractive Crystals," *Appl. Opt.*, **27**(9), 1752–1759, 1988.
4. H. Lee, X.-G. Gu, and D. Psaltis, "Volume Holographic Interconnections With Maximal Capacity and Minimal Cross Talk," *J. Appl. Phys.* **65**(6), 2191–2194, 1989.
5. D. Psaltis, A. Yamamura, H.-Y. Li, "Mass storage for digital optical computers," SPIE Critical Reviews Series, Digital Optical Computing, vol. 1214-08, Los Angeles, Jan. 1990.