Communicated by Dana Ballard

Efficient Training of Artificial Neural Networks for Autonomous Navigation

Dean A. Pomerleau
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA

The ALVINN (Autonomous Land Vehicle In a Neural Network) project addresses the problem of training artificial neural networks in real time to perform difficult perception tasks. ALVINN is a backpropagation network designed to drive the CMU Navlab, a modified Chevy van. This paper describes the training techniques that allow ALVINN to learn in under 5 minutes to autonomously control the Navlab by watching the reactions of a human driver. Using these techniques, ALVINN has been trained to drive in a variety of circumstances including single-lane paved and unpaved roads, and multilane lined and unlined roads, at speeds of up to 20 miles per hour.

1 Introduction

Artificial neural networks sometimes require prohibitively long training times and large training data sets to learn interesting tasks. As a result, few attempts have been made to apply artificial neural networks to complex real-world perception problems. In those domains where connectionist techniques have been applied successfully, such as phoneme recognition (Waibel *et al.* 1988) and character recognition (LeCun *et al.* 1989; Pawlicki *et al.* 1988), results have come only after careful preprocessing of the input to segment and label the training exemplars. In short, artificial neural networks have never before been successfully trained using sensor data in real time to perform a real-world perception task.

The ALVINN (Autonomous Land Vehicle In a Neural Network) system remedies this shortcoming. ALVINN is a backpropagation network designed to drive the CMU Navlab, a modified Chevy van (see Fig. 1). Using real time training techniques, the system quickly learns to autonomously control the Navlab by watching a human driver's reactions. ALVINN has been trained to drive in a variety of circumstances including

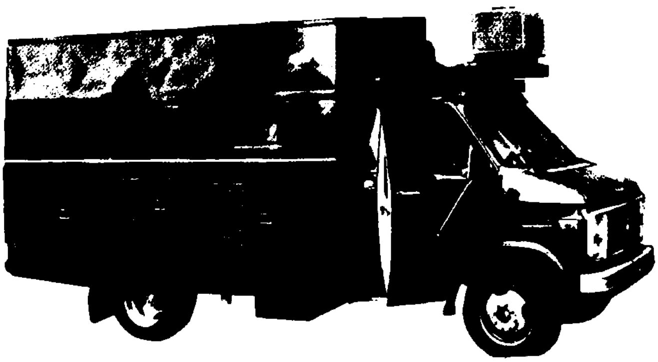

Figure 1: The CMU Navlab autonomous navigation testbed.

single-lane paved and unpaved roads, and multilane lined and unlined roads, at speeds of up to 20 miles per hour.

2 Network Architecture

ALVINN's current architecture consists of a single hidden layer back-propagation network (see Fig. 2). The input layer of the network consists of a 30×32 unit "retina" onto which a video camera image is projected. Each of the 960 units in the input retina is fully connected to the hidden layer of 5 units, which in turn is fully connected to the output layer. The output layer consists of 30 units and is a linear representation of the direction the vehicle should travel in order to keep the vehicle on the road. The centermost output unit represents the "travel straight ahead" condition, while units to the left and right of center represent successively sharper left and right turns.

To drive the Navlab, a video image from the onboard camera is reduced to a low-resolution 30×32 pixel image and projected onto the input layer. After completing a forward pass through the network, a steering command is read off the output layer. The steering direction dictated by the network is taken to be the center of mass of the "hill" of activation surrounding the output unit with the highest activation level. Using the center of mass of activation instead of the most active output unit when determining the direction to steer permits finer steering corrections, thus improving ALVINN's driving accuracy.

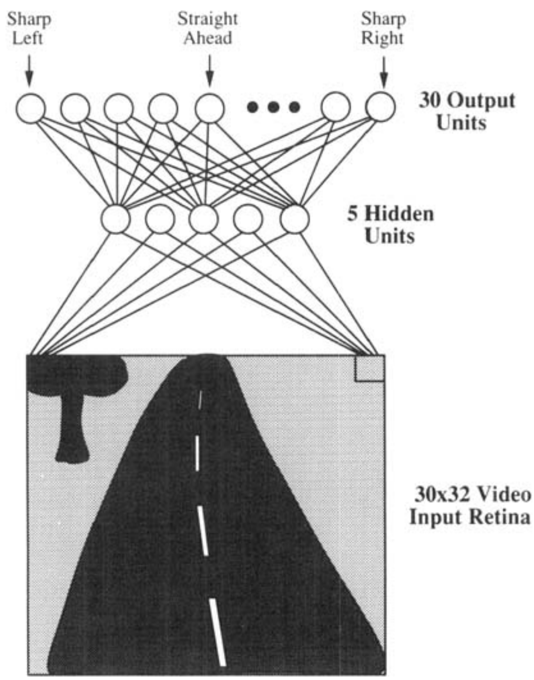

Figure 2: ALVINN architecture.

3 Training

To train ALVINN, the network is presented with road images as input and the corresponding correct steering direction as the desired output. The weights in the network are altered using the backpropagation algorithm so that the network's output more closely corresponds to the correct steering direction. The only modifications to the standard backpropagation algorithm used in this work are a weight change momentum factor that is steadily increased during training, and a learning rate constant for each weight that is scaled by the fan-in of the unit to which the weight projects. ALVINN's ability to learn quickly results from the output representation and the exemplar presentation scheme.

Instead of training the network to activate only a single output unit, ALVINN is trained to produce a gaussian distribution of activation centered around the steering direction that will keep the vehicle centered on the road. As in the decoding stage, this steering direction may fall between the directions represented by two output units. The following approximation to a gaussian equation is used to precisely interpolate the correct output activation levels:

$$x_i = e^{-d_i^2/10}$$

where x_i represents the desired activation level for unit i and d_i is the ith unit's distance from the correct steering direction point along the output vector. The constant 10 in the above equation is an empirically determined scale factor that controls the number of output units the gaussian encompasses.

As an example, consider the situation in which the correct steering direction falls halfway between the steering directions represented by output units j and $j + 1$. Using the above equation, the desired output activation levels for the units successively farther to the left and the right of the correct steering direction will fall off rapidly with the values 0.98, 0.80, 0.54, 0.29, 0.13, 0.04, 0.01, etc.

This gaussian desired output vector can be thought of as representing the probability density function for the correct steering direction, in which a unit's probability of being correct decreases with distance from the gaussian's center. By requiring the network to produce a probability distribution as output, instead of a "one of N" classification, the learning task is made easier since slightly different road images require the network to respond with only slightly different output vectors. This is in contrast to the highly nonlinear output requirement of the "one of N" representation in which the network must significantly alter its output vector (from having one unit on and the rest off to having a different unit on and the rest off) on the basis of fine distinctions between slightly shifted road scenes.

3.1 Original Training Scheme. The source of training data has evolved substantially over the course of the project. Training was originally performed using simulated road images designed to portray roads under a wide variety of weather and lighting conditions. The network was repeatedly presented with 1200 synthetic road scenes and the corresponding correct output vectors, while the weights between units in the network were adjusted with the backpropagation algorithm (Pomerleau *et al.* 1988). The network required between 30 and 40 presentations of these 1200 synthetic road images in order to develop a representation capable of accurately driving over the single-lane Navlab test road. Once trained, the network was able to drive the Navlab at up to 1.8 m/sec (3.5 mph) along a 400-m path through a wooded area of the CMU campus under a variety of weather conditions including snowy, rainy, sunny, and cloudy situations.

Despite its apparent success, this training paradigm had serious drawbacks. From a purely logistical standpoint, generating the synthetic road scenes was quite time consuming, requiring approximately 6 hr of Sun-4 CPU time. Once the road scenes were generated, training the network required an additional 45 min of computation time using the Warp systolic array supercomputer onboard the Navlab. In addition, differences between the synthetic road images on which the network was trained and the real images on which the network was tested often resulted in poor performance in actual driving situations. For example, when the network was trained on synthetic road images that were less curved than the test road, the network would become confused when presented with a sharp curve during testing. Finally, while effective at training the network to drive under the limited conditions of a single-lane road, it became apparent that extending the synthetic training paradigm to deal with more complex driving situations like multilane and off-road driving, would require prohibitively complex artificial road generators.

3.2 Training "On-the-fly". To deal with these problems, I have developed a scheme, called training "on-the-fly," that involves teaching the network to imitate a human driver under actual driving conditions. As a person drives the Navlab, backpropagation is used to train the network with the current video camera image as input and the direction in which the person is currently steering as the desired output.

There are two potential problems associated with this scheme. First, since the human driver steers the vehicle down the center of the road during training, the network will never be presented with situations where it must recover from misalignment errors. When driving for itself, the network may occasionally stray from the road center, so it must be prepared to recover by steering the vehicle back to the center of the road. The second problem is that naively training the network with only the current video image and steering direction runs the risk of overlearning from repetitive inputs. If the human driver takes the Navlab down a straight stretch of road during part of a training run, the network will be presented with a long sequence of similar images. This sustained lack of diversity in the training set will cause the network to "forget" what it had learned about driving on curved roads and instead learn to always steer straight ahead.

Both problems associated with training on-the-fly stem from the fact that backpropagation requires training data that are representative of the full task to be learned. To provide the necessary variety of exemplars while still training on real data, the simple training on-the-fly scheme described above must be modified. Instead of presenting the network with only the current video image and steering direction, each original image is laterally shifted in software to create 14 additional images in which the vehicle appears to be shifted by various amounts relative to the road center (see Fig. 3). The shifting scheme maintains the correct perspective by shifting nearby pixels at the bottom of the image more than far away pixels at the top of the image as illustrated in Figure 3.

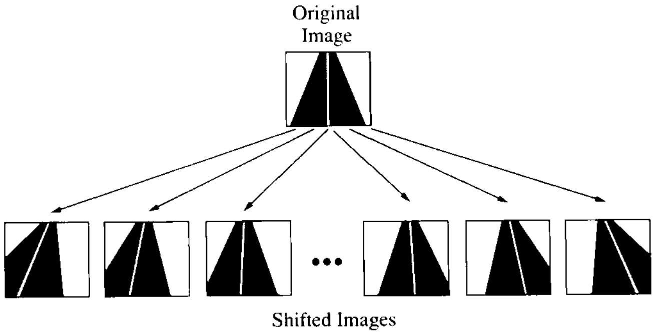

Figure 3: The single original video image is laterally shifted to create multiple training exemplars in which the vehicle appears to be at different locations relative to the road.

The correct steering direction as dictated by the driver for the original image is altered for each of the shifted images to account for the extra lateral vehicle displacement in each. The use of shifted training exemplars eliminates the problem of the network never learning about situations from which recovery is required. Also, overtraining on repetitive images is less of a problem, since the shifted training exemplars add variety to the training set. However as additional insurance against the effects of repetitive exemplars, the training set diversity is further increased by maintaining a buffer of recently encountered road scenes.

In practice, training on-the-fly works as follows. A video image is digitized and reduced to the low resolution image required by the network. This single original image is shifted 7 times to the left and 7 times to the right in 0.25-m increments to create 15 new training exemplars. Fifteen old patterns from the current training set of 200 road scenes are chosen and replaced by the 15 new exemplars. The 15 patterns to be replaced in the training set are chosen in the following manner. The 10 tokens in the training set with the lowest error are replaced in order to prevent the network from overlearning frequently encountered situations such as straight stretches of road. The other 5 exemplars to be replaced are chosen randomly from the training set. This random replacement is done to prevent the training set from becoming filled with erroneous road patterns that the network is unable to correctly learn. These erroneous exemplars result from occasional momentary incorrect steering directions by the human driver.

After this replacement process, one forward and one backward sweep of the backpropagation algorithm is performed on these 200 exemplars to incrementally update the network's weights, and then the process is repeated. The network requires approximately 50 iterations through this digitize–replace–train cycle to learn to drive on the roads that have been tested. Running on a Sun-4, this takes approximately 5 min during which a person drives at about 4 miles per hour over the test road. After this training phase, not only can the network imitate the person's driving along the same stretch of road, it can also generalize to drive along parts of the road it has never encountered, under a wide variety of weather conditions. In addition, since determining the steering direction from the input image merely involves a forward sweep through the network, the system is able to process 25 images per second, allowing it to drive at up to the Navlab's maximum speed of 20 miles per hour.[1] This is over twice as fast as any other sensor-based autonomous system has driven the Navlab (Crisman and Thorpe 1990; Kluge and Thorpe 1990).

4 Discussion

The training on-the-fly scheme gives ALVINN a flexibility that is novel among autonomous navigation systems. It has allowed me to successfully train individual networks to drive in a variety of situations, including a single lane dirt access road, a single-lane paved bicycle path, a two-lane suburban neighborhood street, and a lined two-lane highway (see Fig. 4). ALVINN networks have driven in each of these situations for up to 1/2 mile, until reaching the end of the road or a difficult intersection. The development of a system for each of these domains using

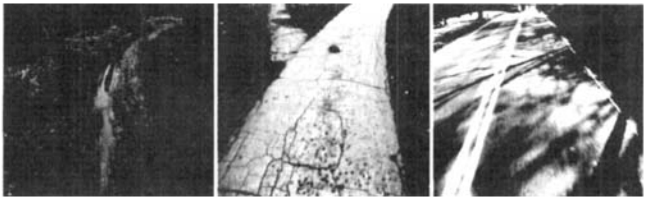

Figure 4: Video images taken on three of the test roads ALVINN has been trained to drive on. They are, from left to right, a single lane dirt access road, a single lane paved bicycle path, and a lined two-lane highway.

[1]The Navlab has a hydraulic drive system that allows for very precise speed control, but that prevents the vehicle from driving over 20 miles per hour.

the "traditional approach" to autonomous navigation would require the programmer to (1) determine what features are important for the particular task, (2) program detectors (using statistical or symbolic techniques) for finding these important features, and (3) develop an algorithm for determining which direction to steer from the location of the detected features.

An illustrative example of the traditional approach to autonomous navigation is the work of Dickmanns (Dickmanns and Zapp 1987) on high-speed highway driving. Using specially designed hardware and software to track programmer chosen features such as the lines painted on the road, Dickmanns' system is capable of driving at up to 60 miles per hour on the German autobahn. However, to achieve these results in a hand-coded system, Dickmanns has had to sacrifice much in the way of generality. Dickmanns emphasizes acccurate vehicle control in the limited domain of highway driving, which, in his words, "put relatively low requirements on image processing."

In contrast, ALVINN is able to *learn* for each new domain what image features are important, how to detect them, and how to use their position to steer the vehicle. Analysis of the hidden unit representations developed in different driving situations shows that the network forms detectors for the image features that correlate with the correct steering direction. When trained on multilane roads, the network develops hidden unit feature detectors for the lines painted on the road, while in single-lane driving situations, the detectors developed are sensitive to road edges and road shaped regions of similar intensity in the image. Figure 5 illustrates the evolution of the weights projecting to the 5 hidden units in the network from the input retina during training on a lined two-lane highway. For a more detailed analysis of ALVINN's internal representations see Pomerleau (1989, 1990).

As a result of this flexibility, ALVINN has been able to drive in a wider variety of situations than any other autonomous navigation system. ALVINN has not yet achieved the impressive speed of Dickmanns' system on highway driving, but the primary barrier preventing faster driving is the Navlab's physical speed limitation. In fact, at 25 frames per second, ALVINN cycles twice as fast as Dickmanns' system. A new vehicle that will allow ALVINN to drive significantly faster is currently being built at CMU.

Other improvements I am developing include connectionist and nonconnectionist techniques for combining networks trained for different driving situations into a single system. In addition, I am integrating symbolic knowledge sources capable of planning a route and maintaining the vehicle's position on a map. These modules will allow ALVINN to make high-level, goal-oriented decisions such as which way to turn at intersections and when to stop at a predetermined destination.

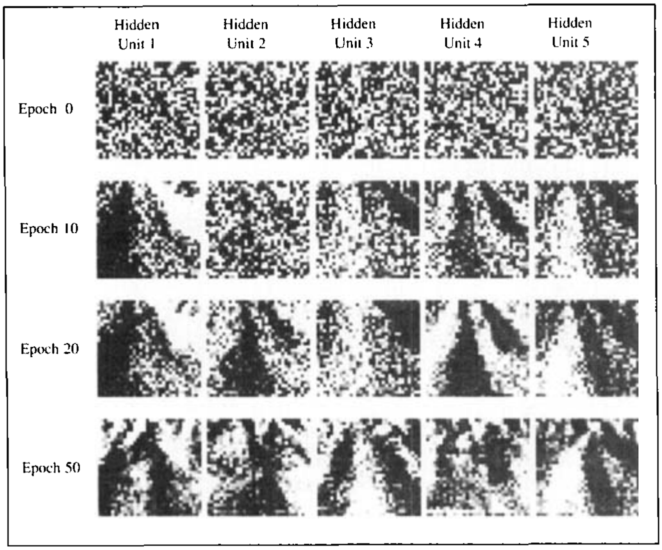

Figure 5: The weights projecting from the input retina to the 5 hidden units in an ALVINN network at four points during training on a lined two-lane highway. Black squares represent inhibitory weights and white squares represent excitatory weights. The diagonal black and white bands on weights represent detectors for the yellow line down the center and the white line down the right edge of the road.

Acknowledgments

This work would not have been possible without the input and support provided by Dave Touretzky, John Hampshire, and especially Charles Thorpe, Omead Amidi, Jay Gowdy, Jill Crisman, James Frazier, and the rest of the CMU ALV group.

This research was supported by the Office of Naval Research under Contracts N00014-87-K-0385, N00014-87-K-0533, and N00014-86-K-0678, by National Science Foundation Grant EET-8716324, by the Defense Advanced Research Projects Agency (DOD) monitored by the Space and Naval Warfare Systems Command under Contract N00039-87-C-0251, and by the Strategic Computing Initiative of DARPA, through contracts DACA76-85-C-0019, DACA76-85-C-0003, and DACA76-85-C-0002, which are monitored by the U.S. Army Engineer Topographic Laboratories.

References

Crisman, J. D., and Thorpe C. E. 1990. Color vision for road following. In *Vision and Navigation: The CMU Navlab*, Charles Thorpe, ed., pp. 9–23. Kluwer Academic Publishers, Boston, MA.

Dickmanns, E. D., and Zapp, A. 1987. Autonomous high speed road vehicle guidance by computer vision. In *Proc. 10th World Congress Automatic Control*, Vol. 4, Munich, West Germany.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Comp.* **1**(4), 541–551.

Kluge, K., and Thorpe, C. E. 1990. Explicit models for robot road following. In *Vision and Navigation: The CMU Navlab*, Charles Thorpe, ed., pp. 25–38. Kluwer Academic Publishers, Boston, MA.

Pawlicki, T. F., Lee, D. S., Hull, J. J., and Srihari, S. N. 1988. Neural network models and their application to handwritten digit recognition. *Proc. IEEE Int. Conf. Neural Networks*, San Diego, CA.

Pomerleau, D. A. 1989. ALVINN: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems, 1*, D. S. Touretzky, ed., pp. 305–313. Morgan Kaufmann, San Mateo, CA.

Pomerleau, D. A. 1990. Neural network based autonomous navigation. In *Vision and Navigation: The CMU Navlab*, Charles Thorpe, ed., pp. 83–92. Kluwer Academic Publishers, Boston, MA.

Pomerleau, D. A., Gusciora, G. L., Touretzky, D. S., and Kung, H. T. 1988. Neural network simulation at Warp speed: How we got 17 million connections per second. *Proc. IEEE Int. Joint Conf. Neural Networks*, San Diego, CA.

Thorpe, C., Herbert, M., Kanade, T., Shafer S., and the members of the Strategic Computing Vision Lab. 1987. Vision and navigation for the Carnegie Mellon Navlab. In *Annual Review of Computer Science*, Vol. II, Joseph Traub, ed., pp. 521–556. Annual Reviews, Palo Alto, CA.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. 1988. Phoneme recognition: Neural networks vs. hidden Markov models. *Proc. Int. Conf. Acoustics, Speech and Signal Process.*, New York, NY.

Received 23 April 1990; accepted 8 October 90.