MANIAC: A Next Generation Neurally Based Autonomous Road Follower

Todd M. Jochem

Dean A. Pomerleau

Charles E. Thorpe

The Robotics Institute, Carnegie Mellon University, Pittsburgh PA 15213

Abstract

The use of artificial neural networks in the domain of autonomous vehicle navigation has produced promising results. ALVINN [Pomerleau, 1991] has shown that a neural system can drive a vehicle reliably and safely on many different types of roads, ranging from paved paths to interstate highways. Even with these impressive results, several areas within the neural paradigm for autonomous road following still need to be addressed. These include transparent navigation between roads of different type, simultaneous use of different sensors, and generalization to road types which the neural system has never seen. The system presented here addresses these issue with a modular neural architecture which uses pre-trained ALVINN networks and a connectionist superstructure to robustly drive on many different types of roads.

1. Introduction

ALVINN (Autonomous Land Vehicle In A Neural Network) [Pomerleau, 1992] has shown that neural techniques hold much promise for the field of autonomous road following. Using simple color image preprocessing to create a grayscale input image and a 3 layer neural network architecture consisting of 960 input units, 4 hidden units, and 50 output units, ALVINN can quickly learn, using back-propagation, the correct mapping from input image to output steering direction. See Figure 1. This steering direction can then be used to control our testbed vehicles, the Navlab 1 [Thorpe, 1991] and a converted U.S. Army HMMWV called the Navlab 2.

ALVINN has many characteristics which make it desirable as a robust, general purpose road following system. They include:

- ALVINN learns the features that are required for driving on the particular road type for which it is trained.
- ALVINN is computationally simple.
- ALVINN learns features that are intuitively plausible when viewed by a human.
- ALVINN has been shown to work in a variety of situations.

These features make ALVINN an excellent candidate as the building block of a neural sys-

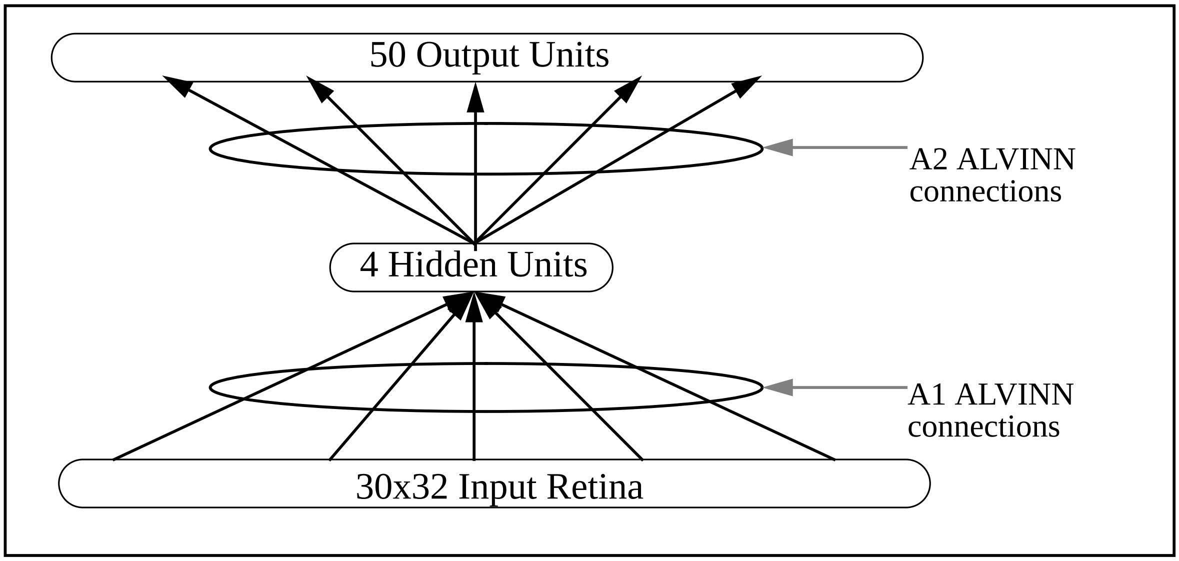

Figure 1: ALVINN network architecture.

tem which can overcome some of the problems which limit its use. The major problem this research addresses is ALVINN's lack of ability to learn features which would allow the system to drive on road types other than that on which it was trained. In addition to overcoming this problem, the system must meet the current needs of the autonomous vehicle community which include:

- The ability to robustly and transparently navigate between many different road types.
- Graceful performance degradation.
- The ability to incorporate many different sensors which can lead to a much wider range of operating conditions.

From these requirements we have begun developing a modular neural system, called MANIAC for Multiple ALVINN Networks In Autonomous Control. MANIAC is composed of several ALVINN networks, each trained for a single road type that is expected to be encountered during driving. See Figure 1. This system will allow for transparent navigation between roads of different types by using these pretrained ALVINN networks along with a connectionist integrating superstructure. Our hope is that the superstructure will learn to *combine* data from each of the ALVINN networks and not simply *select* the best one. Additionally, this system may be able to achieve better performance than a single ALVINN network because of the extra data available from the different ALVINN networks.

2. The MANIAC System

The MANIAC system consists of multiple ALVINN networks, each of which has been pretrained for a particular road type. They serve as road feature detectors. Output from each of the ALVINN networks is combined into one vector which is placed on the input units of the MANIAC network. The output from the ALVINN networks can be taken from either their output or hidden units. We have found that using activation levels from hidden units provides better generalization results and have conducted all of our experiments with this connectivity. The MANIAC system is trained off-line using the back-propagation learning algorithm [Rumelhart, 1986] on image/steering direction pairs stored from prior ALVINN training sessions.

2.1. MANIAC Network Architecture

The architecture of a MANIAC system which incorporates multiple ALVINN networks consists of a 30x32 input unit retina which is connected to two or more sets of four hidden units. (The M1 connections in Figure 2.) This hidden layer is connected to a second hidden layer by the M2 connections. The second hidden layer contains four units for every ALVINN network that the system is integrating. Finally, the second hidden layer is connected to an output layer of 50 units through the M3 connections. All units in a particular layer are fully connected to the units in the layer below it and use the hyperbolic tangent function as their activation function. Also, a bias unit with constant activation of 1.0 is connected to every hidden and output unit. The architecture of a MANIAC system incorporating two ALVINN networks is shown in Figure 2.

The topology of the input retina and M1 connections of MANIAC system is identical to that of the A1 connection topology of an ALVINN network. See Figure 1. This allows us to incorporate an entire MANIAC system into one compact network because the A1 connection weights can be directly loaded onto the M1 connections for a particular set of first layer hidden units of the MANIAC network. Simulating the entire MANIAC system, then, does not entail data transfer from ALVINN hidden units to MANIAC input units, but only a basic forward propagation through the network.

It is the A1 connection weights of the ALVINN network that extract vital features from the input image for accurate driving. So in addition to allowing easy implementation of the MANIAC network, the network topology of the M1 connections allows us to capture important weight information in the MANIAC system that the ALVINN hidden units have learned. These features can be interpreted graphically in two dimensional views of the A1 connection weight values. Typically, a network trained for one lane roads learns a matched filter that looks for the road body, while a network trained on multi-lane roads is sensitive to painted lines and shoulders.

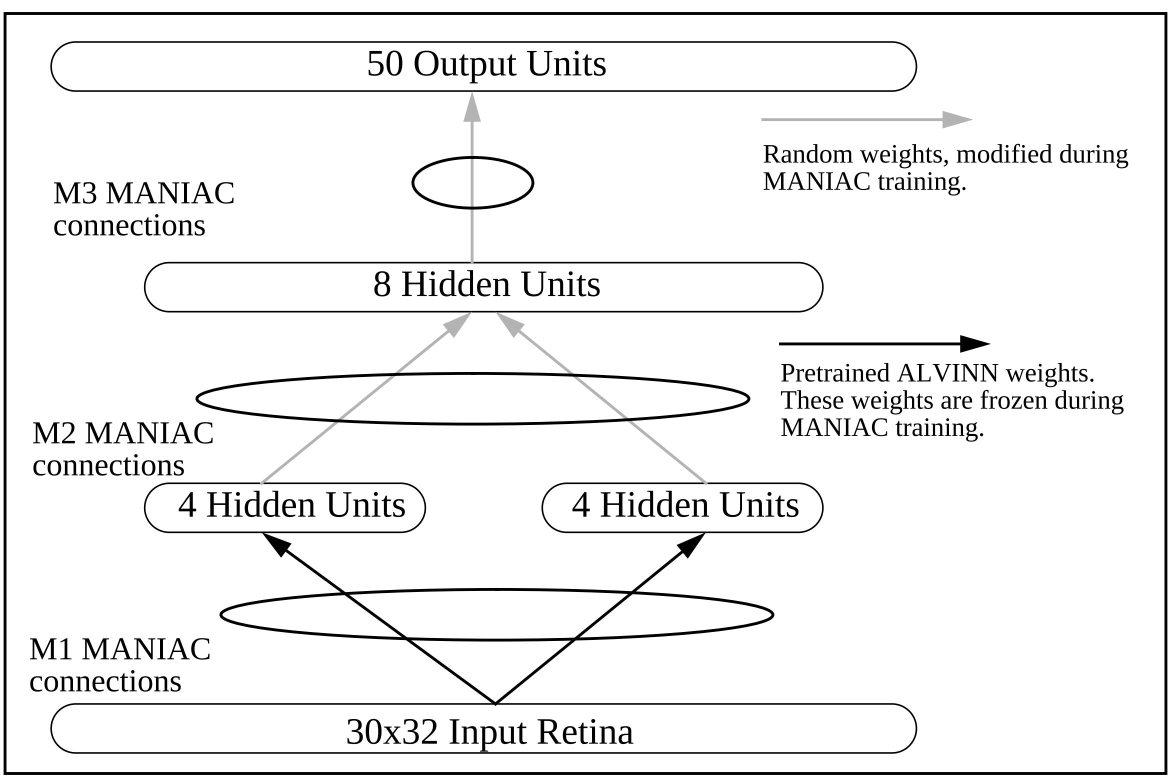

Figure 2: MANIAC network built using two ALVINN networks.

2.2. Training the MANIAC Network

To train the MANIAC network, stored image/steering direction pairs from ALVINN training runs are collated into a large training sequence. These pairs consist of a preprocessed 30x32 image which has been shifted and rotated to create multiple views of the original image along with the appropriate steering direction as derived by monitoring the human driver during ALVINN training. See [Pomerleau, 92] for an in-depth discussion of the image preprocessing and transformation techniques. After collation, the sequence of pairs is randomly permuted so that all exemplars of a particular road type are not seen consecutively. The current size of this training sequence for a two ALVINN MANIAC network is 600. If additional ALVINN networks are used, 300 images per new ALVINN network are added to the training sequence. This sequence is stored for use in our neural network simulator.

Next, weights on each of the connections in the MANIAC network must be initialized. Because the MANIAC M1 connections consist of precomputed ALVINN A1 connection weights, they must be loaded from stored weight files. After this is done, the M2 and M3 connection weights in the MANIAC network are randomized. This weight set is then ready for use as the initial starting point for learning.

To do the actual training, the network architecture along with the weight set created as discussed in the previous paragraph and the stored training sequence, are loaded into our neural network simulator. Because the MANIAC M1 connection weights are actually the pre-trained ALVINN weights who serve as feature detectors, the M1 connections are frozen so that no modification during training can occur to them. See Figure 2.

Initially, training is done using small learning and momentum rates. These values are used for 10 epochs. At this point they are increased (approximately doubled) for the remainder of training. This technique seems to prevent the network from getting stuck in local minima and is an adaption of a technique used in ALVINN training.

The back-propagation learning algorithm is used to train the network. The stored images are placed on the input units of the MANIAC network while a gaussian peak of activation is centered at the correct steering direction on the 50 output units of the network. After about 60 epochs, the network has converged to an acceptable state and its weights are saved. This takes approximately10 minutes on a Sun Sparcstation 2.

It should be noted that MANIAC uses the same output vector representation as ALVINN. This allows the output of the MANIAC network to easily be compared with that of ALVINN for quantitative study and also allows for the use of existing software in the MANIAC-vehicle interface.

2.3. Simulating the MANIAC network

Once the network has been trained, we use it in our existing neural network road following system to produce output steering directions at approximately 10 Hz.

3. Results

Empirical results of a MANIAC system composed of two ALVINN networks have been encouraging. For this system, one ALVINN network was trained to drive the vehicle on a one lane path while the other learned to drive on a two lane, lined, city street. The resultant MANIAC network was able to drive on both of these road types satisfactorily.

To determine more quantitative results, image/steering direction pairs from the same two road types as well as from a four lane, lined, city street were captured. See Figure 3. Using

One Lane Path   Two Lane Road   Four Lane Road

Figure 3: Typical road input images.

these stored images, ALVINN networks were trained in the lab to drive on the one lane paved path and the two lane, lined city street. Also, a MANIAC network integrating the same two ALVINN networks was trained. The results of these experiments are summarized in Table 1. In Table 1 the columns represent the average error per test image for a particular road type and the rows represent the type of network that is being used. The errors computed are of two types, SSD error and Output Peak error. SSD error is the sum of squared differences error while Output Peak error is the absolute distance between the position of the gaussian peak in the desired output activation and the peak in the actual output activation. SSD error can be thought of as a measure of the network's ability to accurately reproduce the target vector while Output Peak error is a measure of the ability of the network to produce the correct steering direction.

The initial comparison to notice in the table is that the ALVINN network trained for a partic-

	One Lane Path		Two Lane Road		Four Lane Road	
	SSD	Output Peak	SSD	Output Peak	SSD	Output Peak
One Lane Path ALVINN	5.913	2.045	5.570	2.228	5.469	2.225
Two Lane Road ALVINN	11.360	3.076	3.621	1.375	1.287	0.823
MANIAC	6.263	2.167	3.907	1.532	1.243	0.774

Table 1: The average output error of ALVINN and MANIAC systems are shown for a variety of road types using two different metrics. The lower the value in the table, the better the accuracy of the network.

ular road type always performs significantly better (> 50%) than the ALVINN network trained for the other road type when presented test images of the type of road on which it is trained. This is to be expected. Also notice that the *single* MANIAC network, which has been trained to respond properly to *both* road types, typically compares well to the correct ALVINN network (within 11% in all cases). As mentioned earlier, this amount of error is acceptable to properly drive the vehicle.

The case of the four lane road is unique in that neither of the ALVINN networks nor the

MANIAC network saw a road of this type. In this case, the response of the one lane path ALVINN network is nearly identical to when it was presented a two lane, lined, city street. Because this type of network typically responds to the body of the road and the fact that the two and four lane roads are both significantly wider than the one lane path, i.e. have a larger body area, this response was expected. A more interesting response is that of the two lane road ALVINN network. It seems to respond better to the four lane road images than it does to the two lane road test images. A possible explanation of why this is occurring can be seen in Figure 3. The four lane road and the two lane road look almost identical. One slight difference, though, is that the contrast of the road/offroad boundary is slightly higher in the four lane road case than it is in the two lane road case. This difference could help the network localize the road better, and because we want the vehicle to drive in the left lane, close to the yellow line, the correct output is identical to the two lane road case.

The most interesting result, though, is that when presented with four lane road images, the MANIAC network actually performs better than either the one lane path ALVINN network or the two lane road ALVINN network. In both the prior cases, the MANIAC network performed slightly worse than the best ALVINN network for a particular road. This could imply that the MANIAC network is using information from both networks to create a reasonable steering direction at its output. This will be discussed more in the following section.

4. Discussion

A central idea that this research is trying to examine is that of improving performance and making connectionist systems more robust by using multiple networks - some of which might be producing incorrect results. In our system the key point to notice is that although a particular ALVINN network may not be able to drive accurately in a given situation, its hidden units still detect useful features in the input image. For example, consider an ALVINN network that was trained to drive on a two lane, lined road. The features that it learns are important for accurate driving are the lines on the road and the road/non-road division. Now present this network with a paved, unlined bike path. The ALVINN network will respond in its output vector with two steering direction peaks. The reason for this is that one of the features that the network is looking for in the input image is the delineation between road and non-road. Because this occurs at two places in the image of the paved bike path, the feature detecting hidden units produce a response which indicates that the road/non-road edge is present at two locations. If these hidden unit activations were allowed to propagate to the output of the network, the characteristic two peak response would appear. Although in reality this is the incorrect response, it is a *consistent* response to this input stimulus. A similar scenario holds for other ALVINN networks given input images of road types for which they haven’t been trained. Because the response of particular ALVINN network is consistent when presented with similar images, the MANIAC network can use this ‘extra’ data to produce a correct, perhaps better, steering direction than a single ALVINN network. It is possible that this is what is happening in the case of MANIAC driving better on the four lane road than either of the ALVINN networks.

5. Future Work

There are many directions this research can take but perhaps the most interesting is that of developing *self-training* systems. In the current implementation of the MANIAC system, ALVINN networks must be trained separately on their respective roads types and then the MANIAC system must be trained using stored exemplars from the ALVINN training runs. If a new ALVINN network is added to the system, MANIAC must be retrained. It would be desirable to have a system that, when given initial or new ALVINN networks, created its own training exemplars and was able to automatically learn the correct MANIAC network weights. Creating training exemplars from existing network weights is essentially the network inversion problem. Techniques such as those developed by [Linden, 1989] may provide clues of how to do this one to many mapping that can create an input exemplar from an output target. It can be argued that this task is extremely difficult, even impossible, due to the high dimensionality of most networks, but perhaps it is worth taking a hard look at implementing some network inversion techniques because of the benefits that can be obtained by having self training modular neural networks.

Another area in which modular neural systems such as MANIAC may be useful is that of incorporating information from different sources. An example of this idea is to use MANIAC as a framework in which to add sensing modalities other than video. In addition to a video camera, our testbed vehicle, the Navlab 2, is equipped with an infrared camera and two laser rangefinders. If these devices can be used as input to ALVINN-like systems which produce a steering angle as output, it is reasonable to assume that a training technique similar to the one used in the current video-only MANIAC system will result in a network which will be robust in all of the component network domains. This could lead to highly robust autonomous systems which could operate in a variety of situations in which current systems fail. Driving with the same system in both daylight and at night is an example. In this scenario video images provide sufficient information to drive in the daytime but at night sensors such as infrared cameras would be necessary. The infrared cameras need not go unused in the day though, as their output would provide addition information to the modular network.

In addition to the previous areas of work, there is much to be done with developing systems which can allocate their resources and group relevant features together. It has been shown that modular neural networks can learn to allocate their resources to match a given problem, such as locating and identifying objects in an input retina [Jacobs, 1990], while the cascade correlation algorithm provides a way to produce appropriately sized networks. [Fahlman, 1990] By using similar techniques in a MANIAC-like system, the need to pretrain ALVINN networks would be eliminated. It is not clear, though, how new information would be incorporated into this type of system once it has been trained.

6. Conclusion

This research has focused on developing a modular neural system which can transparently navigate different road types by incorporating knowledge stored in pretrained networks. Initial results from the autonomous navigation domain are promising. Although the system is simplistic, it provides a starting point from which we can explore many different areas of the connectionist paradigm such as self-training modular networks and network resource allocation. In addition to these areas, autonomous navigation tasks such as multi-modal perception can be studied.

7. Acknowledgments

This research was partly sponsored by DARPA, under contracts “Perception for Outdoor Navigation” (contract number DACA76-89-C-0014, monitored by the US Army Topographic Engineering Center) and “Unmanned Ground Vehicle System” (contract number DAAE07-90-C-R059, monitored by TACOM) as well as a DARPA Research Assistantship in Parallel Processing administered by the Institute for Advanced Computer Studies, University of Maryland. Many thanks also go to the Semiautonomous Surrogate Vehicle group at Martin Marietta, Denver, where this research began.

8. References

[Fahlman, 1990] Fahlman, S. E. and Lebiere, C., “The Cascade-Correlation Learning Architecture”, Technical Report CMU-CS-90-100.

[Jacobs, 1990] Jacobs, R. A., Jordan, M. I., and Barto, A. G., “Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks”, COINS Technical Report 90-27.

[Linden, 1989] Linden, A. and Kindermann, J. “Inversion of Multilayer Nets”, in *Proceedings of the First International Joint Conference on Neural Networks*, Washington D.C.

[Pomerleau, 1991] Pomerleau, D.A. “Efficient Training of Artificial Neural Networks for Autonomous Navigation”, *Neural Computation 3:1*, Terrence Sejnowski (Ed).

[Pomerleau, 1992] Pomerleau, D. *A. Neural Network Perception for Mobile Robot Guidance*. Ph.D. dissertation, Carnegie Mellon University, February, 1992.

[Rumelhart, 1986] Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986) “Learning Internal Representations by Error Propagation” in Rumelhart, D.E. and McClelland, J.L. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press.

[Thorpe, 1991] Thorpe, C., Hebert, M., Kanade, T., and Shafer, S. *Toward Autonomous Driving: The CMU Navlab*. IEEE Expert, V 6 #4, August 1991.