Paper Submission to the:
IEEE Meditteranean Symposium on new Directions in Control Theory and Applications
June 21-23, 1993

A Novel Neural Network Structure with Application to Robot Trajectory Control

Athanasios Sideris[1]
University of California, Irvine
Dept. of Mechanical Eng.
Irvine, CA 92717, USA

Kazuyuki Orita
Nippon Steel Corporation
2-6-3 Otemachi, Chiyoda-ku
Tokyo 100, Japan

January 14, 1993

Keywords: Neural Networks, Intelligent Control, Robotics.

Extended Abstract

During the past few years, there has been an increasing research activity in the area of neural networks and their application to various fields such as pattern recognition, signal processing and control, robotics, etc. The function of neural networks is to perform a transformation on a set of input patterns or signals and perhaps their most appealing characteristic is their capability to "learn" their function on the basis of input instances for which the correct answer is known. The most commonly used neural network structure is that of Feedforward Neural Networks (FFNN) with Backward Error Propagation (BEP) as the learning rule. In our experience, as well as the experience of other researchers, learning in FFNN with BEP can be very effective

[1]Author to whom all correspondence should be addressed. e-mail: sideris@euclid.eng.uci.edu, phone: (714)-856-8139, fax: (714)-856-8585.

especially when the number of neurons in the hidden layer(s) and certain parameters determining the rate and accuracy of learning are chosen appropriately. However, there is no guarantee that these networks with have good generalization properties. i.e that they can produce the correct answer to inputs that have not been trained on.

In this paper, we first propose a method for structured learning in FFNN's which results in improved generalization properties and also faster training. We will refer to the resulting neural network as Structured Neural Network (SNN). We then apply SNN to the task of controlling the motion of a two-link robotic manipulator over a desired trajectory. The latter problem has been used extensively by many researchers as a benchmark problem for testing the neural network concepts for the control of dynamic systems. We compare the performance of SNN to a standard FFNN and also to the Cerebellar Model Articulation Controller (CMAC) of Albus'. We evaluate performance of the neural network based control system in terms of 1) its ability to track a given trajectory, 2) its ability to generalize to nearby trajectories before any additional training, 3) its ability to "remember" previous trajectories that it has been trained on after training on new trajectories, and 4) its ability to tolerate measurement noise.

In applications of neural networks to control, there are some additional issues to be resolved besides effective learning on the training set and good generalization. Namely, first, FFNN implement nondynamic maps while the input-output behavior of the system to be controlled is dynamic, and secondly, it is desirable that learning should take place during system operation and this implies that the error signals are not readily available at the output of the neural network as supervised training procedures require. We employ a control system configuration in which these issues are resolved. Our configuration consists of a combination of a conventional controller and an neural network configured as a state feedback controller. The basic rationale behind such a scheme is that the conventional controller should be designed with a minimal effort so that only a basic level of control system performance is assured, and the neural network is used to improve system performance during operation. We tested this scheme via computer simulation and we found that SNN achieves excellent performance with respect to the above criteria, outperforming both FFNN and CMAC.

We next briefly explain the key concept behind CNN. Generalization in neural networks, especially in signal processing and control applications in which the input space is a continuum, can be interpreted as the ability of the network to interpolate and/or extrapolate on the basis of the training set. It is a common experience of several researchers that increasing the number of hidden neurons in FFNN leads to poorer generalization properties. The basic idea behind structured learning and SNN is to use only a fraction of the neurons in the hidden layer of a three-layered FFNN both during training and computation of the response to a particular input. In this manner, the conflicting objectives of having a lot of hidden neurons to improve the capacity of the network and having fewer hidden neurons to improve its generalization properties are resolved.

The hidden neurons which are activated for a particular input are determined from the first stage of a Cerebellar Model Articulation Controller (CMAC) driven by the same input as FFNN. The key feature of CMAC is a layer of hard thresholding neurons or cells called the association layer. The first stage of CMAC consists of fixed connections from the (discretized) input space to the association layer. These connections are configured so that inputs which are "close" excite a lot of common cells in the association layer and thus result in responses that are likely to be "close". Therefore, CMAC exhibits a natural tendency to generalize and indeed it can be shown that it performs a kind of linear interpolation on the training set. Since the size of the association layer turns out to be of the order of the discretized input space, in order to make this scheme practical, Albus uses hash-coding to map the association layer onto the available neurons of a hidden layer between input and output layers. He showed that for practical sizes of the hidden layer, the probability for the so-called collisions can be quite low.

In SNN the association layer of CMAC driven by the same input as the FFNN is hash-coded onto the hidden layer of FFNN. Therefore, the basic idea of CMAC carries on, i.e. close (or similar) inputs enable a lot of common hidden neurons and far (or dissimilar) inputs enable normally disjoint sets of hidden neurons. This, as in CMAC, results in a tendency to interpolate among the patterns in the training set and thus in improved generalization properties. Despite the large size of the FFNN network used in SNN, we obtain significantly better loading times as compared to standard FFNN, in terms of both number of passes through the training set and computations during each pass. This is because only a few neurons are involved during training for a particular input, thus reducing the effective size of the network and also avoiding interference between inputs in the training set during learning.