Literature Review of Applications of Neural Network in Control Systems

[1]**Lalithamma.G.A**, [2]**Dr. P.S. Puttaswamy**

[1]Associate professor , Dept of Electrical & Electronics Eng, SJBIT, Bangalore
[2]Professor , Dept of Electrical & Electronics Eng, PESCE, Mandya

Abstract: In this literature review the literature available for the neural networks in general, feed forward neural networks in particular are briefly discussed. The application of neural networks in the control systems, direct and Indirect Adaptive Controls, closed loop and fixed gain controller are also reviewed. Important conclusions are drawn and the advantages and limitation are highlighted.

Index Terms- Neural network, back propagation, feed forward neural network, perceptron, learning, weights, training, adaptive control

I. Introduction

By the derivation of back propagation, the modern era of neural networks started in 1986. A good amount of literature survey has been carried out on neural networks [1]. Neural networks find applications in variety of subjects like control systems, weather forecast, etc. It is a tedious job to take the deep depth of available material. To understand the evolution of neural networks brief history of neural networks has been written. To study the back propagation algorithm and feed forward neural network a detailed review has been written. In this study, papers on various topics are detailed to explain the need for the proposed work. There are limited numbers of books in the area of neural networks, which are distinguished itself as the leading authority in the past ten years.

II. Neural Networks history

By modeling the neuron, neural network history can be traced back. The first model of a neuron was used by physiologists, McCulloch and Pitts. The first modeled neuron was with one output and two inputs. They found that with only one input active, the neuron would not become active. Binary output was found for input were of equal weight, the output found to be zero till the inputs summed up to a thresh hold value. The neuron developed by McCulloch and Pitts' neuron today is the logic circuit [2].

Rosenblatt developed p*erceptron* as the next model to achieve "learning." Rosenblatt used trial and error method and interconnected perceptrons randomly to change the weights. The better model for the electrochemical process is the model developed by McCulloch and Pitts'. This neuron is the basis in the field of modern day neural networks. Neuron works like a voltage-to-frequency translator due to electrochemical process of a neuron. Neuron discharges due to chemical reaction when the certain threshold is build by the neuron, then fires at a higher frequency. Though higher inputs come into the neuron but the magnitude of the output from the neuron is the same [3].

Model of the perceptron can be traced back to the model of the neuron with multiple inputs and a single output. Perceptron seems to resemble the neuron but perceptron similarity does not model the complex electrochemical processes that actually go on inside a neuron. Very simple mathematical representation of the neuron is the perceptron. By trial-and-error method weights in a neuron are adjusted by Rosenblatt. weight space idea to the perceptron was brought by Selfridge [4]. By choosing a direction vector Selfridge adjusted weights of neuron. Weights were returned to their previous values, if the performance did not improve, and a new random direction vector was chosen. The above is the process which is referred as *climbing the mountain. S*ince energy is minimized it is referred as *descending on the gradient*. A mathematical method was developed for adapting the weights by Nguyen and Hoff [5]. Based on minimizing the error squared and Assuming a desired response existed, a gradient search method was implemented. Later this algorithm is known as

Least Mean Squares (LMS). In the last few years LMS and its variations are extensively used in a variety of applications. For minimizing the error mathematical method was provided. By gradient search method, trial-and-error process is not learning. With Selfridge's work computational time decreased, but the amount of computational time decreased even more by LMS method by making use of feasible perceptrons.

In 1960’s, articles about robots that could think, filled the newspapers. Any problem can be solved by perceptron, it seems, by one book by Minsky and Papert [6].They brought the research to an halt. Linearly separable problems could be solved by perceptrons. n-1 nodes are needed to solve n separable problems and this was shown by Perceptions’. 2-separable problem or a linearly separable problem can be solved by a perceptron. Research went unfunded in the field of neural networks after Perceptrons was published, it would continue until to solve n-separable problems a method develops. Back propagation algorithm was first developed by Werbos Parker and Rumelhart and McClelland [7] simultaneously rediscovered independently. N-1 node neural network could be constructed and trained by training perceptrons in a multilayer configuration, Widrow-Hoff LMS algorithm by knowing the error between the some known desired output and output. The weights that are starting with the output layer weights are adjusted according to back propagation through the hidden layer till the input layer is reached. Schematic of the perceptron can be changed by back propagation algorithm using sigmoidal function as the squashing function. Signum function was used by the perceptron of earlier versions. Sigmoidal function is differentiable where as signum function has the advantage to allow the neural network to converge to a local minimum. Gradient information can be transferred using nonlinear squashing function by back propagation algorithm. Anderson and Rosenfeld is an excellent source of the work [8].

In the field of neural networks the collection of papers is very good. About 25 years ago golden age of neural network research ended. Now the research in this area is re-energized after the discovery of back propagation. Interconnection of perceptrons is used by the feed-forward neural network and many reviewers used this. Neural network will be the cornerstone of the work done in this study for the feed-forward neural network and the back propagation algorithm.

III. Feed-Forward Neural Network

With a differentiable squashing function usually the sigmoidal function, the feed-forward neural network is a network of perceptrons. Using the idea of minimizing the error squared, the back propagation algorithm adjusts the weights. For adjusting the weights across multiple hidden layers the back propagation algorithm allows the differentiable squashing function. To adjust the weights n-separable problems can be solved. Exclusive-OR, or the XOR problem can be solved by having multiple nodes on each layer. From input to output feed-forward neural network is connected and on the adjacent layers each node is connected to every node.

The representative of the neuron is the individual perceptrons or nodes. The output from a previous layer is the input to the node if the node is on the output layer or hidden layer and input to the neural network is the input to the node. To train the neural network key is the node. At a time, the weights of one individual node can be changed by changing the weights using back propagation algorithm. The difference between output and the desired response is calculated in each iteration. The output of one individual node depends on inputs which are nothing but the outputs of nodes on the previous layer of the entire neural network. In the case of a single output, manageable training process becomes possible by the neural network, by breaking neural network down to the nodes. For updating the weights, the back propagation algorithm is an LMS-like algorithm. Rumelhart and McClell minimized the square error by derivation of the back propagation algorithm.

At the first layer of nodes the inputs get summed after entering at the neural network. During the training process at the second layer of nodes the first layer outputs get summed up until the output comes from the neural network process. The error is calculated by comparing the desired output with the actual output. After output comes from the output node for adjusting the weights backwards through the neural network, the error is used. One shortcoming weight adjustment equation is that the weights of all the nodes on the same layer cannot be the same, because the weights will be adjusted with the weight for each node that has identical weights on each layer. on each layer the weights would be adjusted If all of the neural network's weights were initially set at zero. Equivalent Mathematical model is having per layer a single node. Hence neural network weights, when neural network with multi-layer, need to be randomly initialized. To search the weight space properly, initializing the weights randomly is the other reason .The randomly initialized weights makes it very difficult to estimate the initial performance of the control system .

IV. Neural Networks in Control Applications

Neural networks find its application in the area of control systems. When any methodology is applied controls has its own unique set of problems to solve. To conform to a set of specifications the performance needs to be changed, is the principle behind controls. Due to non-linearity and uncertainties the goal of the control becomes complicated. Methodologies are developed to handle uncertainties is tried by Control theory. When there is no system priori knowledge, neural networks finds its applications. System partial model is available often, where as the system complete model is available rarely.

For nonlinear plants adaptive control schemes are initiated by Narendra and Parthasarathy [9]. Identification and control of neural networks applications are dealt by Many of Narendra's papers. There was work done on self-learning control systems by Nguyen and Widrow. Using neural networks identification problems and control problems a Long history was given by Windrow’s papers. Windrow’s work was on open-loop control and with no priori knowledge. Closed-loop feedback was not included in Windrow’s work. To achieve the specified performance, he stabilized an unstable system with feedback and then used neural network. Application of neural network is the suggestion given by Windrow’s work. In the field of control based on neural network, ground work is done by Narendra and Widrow. Work done by them on neural network was reviewed in most of the papers in control applications. Neural networks topics are given in the following sections. About any system limited amount information going to be known. Any knowledge about the priori system leaving is not a good design.

In several places it has been suggested the use of a priori information and integration of the information about the system to control by neural network, before applying to the actual system. Neural network controller trained the neural network off-line with known dynamics of the system according to Selinsky and Guez [10]. Through the training algorithm modification, Joerding and Meador[11], using a priori knowledge, constrained the weights of the neural network through a parallel control path. A priori knowledge was incorporated by Nordgren and Meckl [12] . Neural network structure to the topographical weight map was developed by Miller, Sutton and Werbos [13]. Functional links are used to replace the linear inputs by Pao [14]. He developed a technique for enhancing the initial representation of the data to the neural network. A priori knowledge was incorporated as the output layer of the neural network into the system by Brown, Ruchti, and Feng [15]. To design a fast, effective controller it is very important to have priori knowledge linsky and Guez. In neural network it is very common practice to train the neural network off-line, use of a priori knowledge are found in papers by Selinsky's and Iiguni's. Their idea is to train the neural network by creating a model with more details available. The colored noise is the input for the training set. Neural network after training is connected to the actual system. The neural network may not perform as expected due to the nonlinearities is the problem with this method, if the model is not precisely correct.

Narendra and Parthasarathy used same idea and it is used by many other researchers. Model of the neural network was created by the assumed access to the actual system. Neural network controller is trained by the model of the neural network. Training, using the actual system, works well using a neural network. Creating a good model of the system is the first problem. Finding correlation between the outputs and the inputs is relied on the neural network. Correlation is the second problem to influence the working of the controller with neural network. Priori information was not used by the above two methods. Using a priori knowledge to the constrained weights of the neural network was used by Joerding and Meador. The problem of priori knowledge incorporating about an output optimal function into specific constraints is addressed by them. Weight Constraint method and an Architecture Constraint method are the two general approaches. Monotonic and concavity which are the forms of the optimal output function was assumed by both the methods. Slope does not change sign in monotonic function where as slope that increases (or decreases) as the function arguments increase the neural network desired output is constrained to these function types. With a hyperbolic tangent squashing function for feed-forward neural network to exploit the mathematical nature, two methods are used. Hyperbolic tangent sign is the same as argument sign. Hyperbolic tangent is one which is monotonic and concave optimal function derivative plus back propagation constitute the modified training algorithm. To the neural network, priori information encoding idea is interesting for system modeling. Thus the above methods work well but for controller application they are not translatable directly.

For representing a nonlinear function, complex function f(s) neural network is a table look-up technique according to Miller, Glanz, and Kraft[16]. Albus did the original work on CMAC neural network by using functional links to replace the linear inputs. A technique was developed by Pao for enhancing neural network initial data representation. To find simple mathematical correlations between the output and input, such as higher-order terms or periodicity an attempt is made with functional links. For the neural network data preprocessing, functional links are very important. A functional link is also sometimes called *conditioning the input* between neural networks and adaptive control. There is a parallel input to the adaptive controller, which is limited to an order of magnitude of zero and this method in adaptive control is called *the MIT rule* . Without the magnitude of the input overwhelming the coefficients to adaptive scheme, the MIT rule can constrain the input of the neural network. A functional link in its simplest form is used. The functional link makes the neural network input more usable if the neural network input is ill-conditioned by structuring the input. The amount of work done by the neural network can be decreased by functional links such that it is easier to see by the neural network the correlation between the output and input. Function link is very useful if the prior knowledge of the system contains information. The functional link is limited if a priori knowledge does not exists.

A gray layer is a method which is developed by Brown, Ruchti, and Feng to incorporate a priori information of the system. Output of the neural network was used by the gray layer to propagate the error through the gray layer to the weights. Changes in the training method are included in Brown, Ruchti, and Fengs paper. To converge neural network weights through the gray layer, the error needs to be propagated. In the identification of nonlinear uncertain systems advantage is decided by the gray layer. Such information exploitation is usually beneficial, resulting in the selection of more accurate identification models and a faster rate of parameter convergence exerted by the authors. The most difficult part of a model is to obtain knowledge of the non-linearity, which is required by the gray layer. CMAC neural networks and functional links in the appropriate situations are all very useful. Incorporating a priori knowledge into the neural network, many methods are available. Specific kind of knowledge seems to be needed by each method. Each can be used to a limited degree in many situations to reduce the pseudo-linear problems. Gray layers, CMAC and functional links can be used if the system nonlinearities are known. Often system linear parameters are known. It is possible to add the neural network controller parallel to the classical controller in the control of the system.

V. Direct and Indirect Adaptive Control

Neural networks apply two methods, namely, indirect adaptive control and direct adaptive control. When a plant viable model exists direct adaptive control can be applied by a second neural network. When a model must be developed, indirect adaptive control is applied. Parthasarathy and Narendra originally did the work for the two methods using neural networks. The above work has followed up by several other researchers to the inverted pendulum problem. Tanomaru and Omatu [17] applied two methods to a two link robot arm .Indirect adaptive control was applied by Greene and Tan [18] .To adjust the weights of the neural networks Back propagation was used by both the method. To develop a gradient for convergence Jacobian was used in the update algorithm model adapted by the controller. Error must be back propagated through the plant's Jacobian matrix because the plant lies between output error that is to be minimized and the adaptive neural network. Jacobian knowledge is required for this procedure. To replace Jacobian for SISO plants the partial derivatives can be used. Some knowledge requirement about plant is the direct adaptive control's serious drawback. The same is not required in the indirect adaptive control scheme. Two neural networks: a plant emulator and a controller are required in indirect adaptive control scheme. Emulator is a feed-forward neural network, and plant emulator should be trained off-line with a sufficiently large data. To allow for identification via back propagation, and to calculate the plants derivatives, an efficient way is provided by the emulator. By considering bigger ones as two networks, the parameters of the controller need to be adjusted. On-line training process can be performed on both networks. With dynamic and static back propagation, Parthasarathy and Narendra developed convergence process. For the dynamic back propagation, the derivatives of the output of the plant are calculated. When there is no plant model control approach static back propagation can be used to update the controller weights. The indirect adaptive control is particularly interesting for a wide variety of control problems. Indirect adaptive control and the direct adaptive control methods finds application if the plant and plant model have Jacobian matrices which are similar. The control by direct adaptive works very well to create a model off-line. If sufficient data is available, then the indirect method works well. To converge neural networks weights both indirect method and direct adaptive rely on back propagation. Research in the area of direct adaptive on control was carried out inside a closed loop with the addition of fixed-gain controller.

VI. Closed-Loop, Fixed-Gain Controller

In a parallel path to the neural network controller classical PID controller is used. To control the actual system, a classical controller is built using priori information to create a system model, and from that model it is built to increase system performance, and to supplement the classical controller, along with a parallel path to the classical controller a neural network controller is placed. On the plant, as well as on the systems priori knowledge, the adaptive law is based. In Chen and Chang Jin, Pipe and Winfield [19] the above idea can be seen. Method of priori knowledge incorporated in the classical controller. By giving neural network an unknown initial gain, neural network is randomly initialized. It is difficult to predict systems stability and performance. The controller with the neural network has to find correlation in the data of its own Meckl and Nordgrento.

Nonlinear plant control neural networks use three independent neural networks. Pre-filter, a feed-forward controller and a feedback controller are the three neural networks. Neural networks are trained by different learning techniques. To teach feed-forward controller, the pre-filter general learning architectures and indirect learning are used. Feedback controller is taught using specialized learning architecture. Neural network is trained using a new algorithm developed by them with the nonlinearities of the plant knowledge. The new algorithm thinks of the plant as another layer to the neural network, and the partial derivatives of the plant at its operating point are used to train through the plant. This method requires knowledge of the nonlinearities of the plant [20].

Conclusions:

In this work, an extensive literature survey is conducted for the application of neural networks in applications related to control systems. Various models are discussed in detail and its suitability for specific application is highlighted. The importance of back propagation algorithm is emphasized. The algorithms available to control non-linear plants are also discussed. The gaps in these algorithms are identified and remedies are discussed.

References

[1] Rumelhart, D. E. and J. L. McClelland, 1986, Parallel Distributed Processing:
Explorations in the Microstructure of Cognition, MIT Press, Cambridge, MA
[2] McCulloch, W. S. and W. Pitts, 1943, "A logical calculus of ideas immanent in
nervous activity," Bulletin of Mathematical Biophysics, vol. 5 pp. 115-133
Neural Networks, pp. 2476-2481
[3] Rosenblatt, F., 1958, "The perceptron: a probabilistic model for information
storage and organization in the brain," Psychological Review, vol. 65, pp. 386-
408
[4] Selfridge, O. G., 1958, "Pandemonium: a paradigm for learning," Mechanisation

of Thought Processes: Proceedings of a Symposium Held at the National
Physical Laboratory, London: HMSO, pp. 513-526
[5]Nguyen, D. and B. Widrow, 1990, "Improving the Learning Speed of 2-layer Neural Networks by Choosing Initial Values of the Adaptive Weights,"
Proceedings of the IEEE International Joint Conference on Neural Networks,
Vol. 3, pp. 21-26
[6]Minsky, M. and S. Papert, 1969, Perceptrons, MIT Press, Cambridge, MA
[7]Parker, D., 1985, "Learning Logic," Technical Report TR-87, Center for
Computational Research in Economics and Management Science, MIT,
Cambridge, MA
[8] Anderson, J. A. and E. Rosenfeld, 1987,
Neurocomputing: Foundations of
Research, MIT Press, Cambridge, MA
[9] Narendra, K. S. and K. Parthasarathy, 1990, "Identification and Control of
Dynamical Systems using Neural Networks," IEEE Transactions on Neural
Networks, Vol. 1, No. 1, pp. 4-27
[10] Selinsky, J. W. and A. Guez, 1989, "The Role of A Priori Knowledge of Plant
Dynamics in Neurocontroller Design," Proceedings of the 28th Conference on
Decision and Control, Vol. 2, pp. 1754-1758
[11]Joerding, W. H. and J. L. Meador, 1991, "Encoding A Priori Information in Feed-
Forward Networks," Neural Networks, Vol. 4, pp. 847-856
Proceedings of the IEEE International Joint Conference on Neural Networks,
Vol. 3, pp. 21-26
[12]Nordgren, R. E. and P. H. Meckl, 1993, "An Analytical Comparison of a Neural
Network and a Model-Based Adaptive Controller," IEEE Transactions on Neural
Networks, Vol. 4, No. 4, pp. 685-694
[13] Miller, W. T., R. S. Sutton, and P. J. Werbos, 1990, Neural Networks for Control,
MIT Press, Cambridge, MA
[14] Pao, Y. H., 1989, Adaptive Pattern Recognition and Neural Networks
[15]Brown, R. H., T. L. Ruchti, and X. Feng, 1993, "Artificial Neural Network
Identification of Partially Known +Dynamic Nonlinear Systems," Proceedings of
the 32nd Conference on Decision and Control, vol. 4, pp. 3694-3699
[16]Miller, W. T., F. H. Glanz, and L. G. Kraft, 1987, "Application of a General
Learning Algorithm to the Control of Robotic Manipulators," The International
Journal of Robotics Research, Vol. 6, No. 2, pp. 84-98
[17] Tanomaru, J. and S. Omatu, 1991, "On the Application of Neural Networks to
Control and Inverted Pendulum: an Overview," Proceedings of the 30th SICE
[18]Greene, M.E. and H. Tan, 1991, "Indirect Adaptive Control of a Two-Link Robot
Arm Using Regularization Neural Networks," Proceedings of the International
Conference on Industrial Electronics, Control and Instrumentation, Vol. 2, pp.
952-956
135
[19] Jin, Y., T. Pipe, and A. Winfield, 1993, "Stable Neural Network Control for
Manipulators," Intelligent Systems Engineering, Winter, Vol. 2, No. 4, pp. 213-
222
[20]Psaltis, D., A. Sideris, and A. A. Yamamura, April 1988, “A Multilayered Neural
Network Controller,” IEEE Control Systems Magazine, Vol. 8, No. 2, pp. 17-21