Fifth IFSA World Congress (1993), 1118-1120

Fuzzy Neural Controller with Additive Hybrid Operators

Yoichi Hayashi+, James M. Keller*, and Zhihong Chen*

+ **Dept. of Computer and Information Sciences**
Ibaraki University
Hitachi-shi, Ibaraki 316, Japan
TEL: (+81) 294-25-7111
FAX: (+81) 294-34-3254
e-mail: hayashi@cis.ibaraki.ac.jp

* **Electrical and Computer Engineering**
University of Missouri-Columbia
Columbia, MO. 65211, USA
TEL: (314) 882-7339
FAX: (314) 882-0397
e-mail: ecejk@mizzou1.bitnet

Abstract

Fuzzy logic places a considerable burden on an inference engine for applications such as control or approximate reasoning. Various neural network architectures have been proposed to deal with the computational task, and yet, maintain flexibility in the desired traits of the final system. Recently, we introduced a trainable network architecture whose nodes implement weighted Yager additive hybrid operators for fuzzy logic inference in an approximate reasoning setting. In this paper we examine the utility of such networks for control situations. We show that they are capable of learning control functions which are piece-wise monotonic in each of the variables. The learning ability is demonstrated through an example.

Introduction

Fuzzy logic and neural networks have gained considerable attention in recent years as mechanisms to solve control problems [1-6]. Both technologies, and combinations of the two, have been shown to deal effectively with nonlinear situations, especially in those cases where the underlying model is only known heuristically, or through a collection of training data. Recently, we have introduced a new class of fuzzy neural networks, based on additive hybrid fuzzy set theoretic connectives for pattern recognition, general multicriteria decision-making, and fuzzy logic inference[7-11]. The networks are trainable and the individual nodes can be linguistically interpreted as "mini-rules" after training [9]. Also, unimportant features can be detected during training [10]. The purpose of this paper is to demonstrate how these networks can be applied to control problems.

Weighted Yager Additive Hybrid Operator Networks for Control

Fuzzy set theoretic connectives, i.e., unions, intersections, generalized means, and hybrid operators, are useful for aggregating memberships functions. The resulting membership depends on the type of aggregation connective used, and this type is dictated by the kind of attitude that we expect from the aggregation connective. These connectives are very useful in decision analysis and making. The reader is referred to [7,12] for a more complete description of such fuzzy set connectives.

In the hybrid type of connective, the high input values are allowed to compensate for the low ones. The additive γ operators are defined as weighted arithmetic mean of union and intersection operators respectively:

$$A \oplus_\gamma B = (1 - \gamma)(A \cap B) + \gamma (A \cup B).$$

It is clear that this operator can act as a pure intersection or union at the extremes: $\gamma = 0$ and 1 respectively. But it allows the intersection and union to compensate for each other when $0 < \gamma < 1$. Thus γ can be regarded as the parameter that controls the degree of compensation. There are many different forms of both the additive and a corresponding multiplicative hybrid connective which are useful in automated decision analysis[7,12].

In order to match more closely the way in which a traditional neural network handles inputs and weights, exponential weights were incorporated into the new operators. Here, the output value had the form

$$y = (1 - \gamma)\, y_1 + \gamma\, y_2, \quad \text{where}$$

$$y_1 = 1 - \min\{1, f_1(x_i^{\delta_i}, p)\}, \quad \text{with}$$

$$f_1(x_i^{\delta_i}, p) = \Big[\sum_{i=1}^{m} (1 - x_i^{\delta_i})^p\Big]^{1/p}, \; p \in [0, \infty);$$

and

$$y_2 = \min\{1, f_2(x_i^{\delta_i}, p)\}, \quad \text{with}$$

$$f_2(x_i^{\delta_i}, p) = \Big[\sum_{i=1}^{m} (x_i^{\delta_i})^p\Big]^{1/p}, \; p \in [0, \infty).$$

The union part (y_2) is just a saturated p-norm of the exponentially weighted inputs, and the intersection is obtained through DeMorgan's Law. In this case, the constraint $\sum_{i=1}^{m} \delta_i = m$ (which ensures that the union part is always greater than the intersection part in the multiplicative γ-model) is no longer needed. All that is required is that each weight is greater than or equal to zero, and that at least one $\delta_i \geq 0$, [7]. The $x_i \in [0,1]$ are the inputs or criteria to be aggregated, δ_i represents the weight associated with the input x_i and is related to the importance of that input, and $\gamma \in [0,1]$ controls the degree of compensation between the union and intersection parts of the operator.

Since each node computes a value which is a differentiable function of its inputs, this network can be trained using any neural network training technique, such as backpropagation. To demonstrate the use of weighted Yager additive hybrid networks to control, we applied the network shown in Figure 1 to the data found in [3]. These data were generated from the equation $y = (1.0 + x_1^{0.5} + x_2^{-1} + x_3^{-1.5})^2$ and random noise x_4. Because these operators are monotonic non-decreasing functions of their inputs (or monotonic non-increasing functions of the complements of the inputs), it is necessary to determine, from the training data, regions where the output is increasing and decreasing with respect to each input variable. Then, over those regions, the input variable or its complement is used for the network. The data, both inputs and outputs, were scaled into the open interval (0,1) for use in this network. The first six columns of Table 1 represent the training data ("Y" represents the target; "y" corresponds to the output of the network after training). The last six columns display the testing data.

As can be seen, the expected values ("Y") and the generated values ("y") are in close agreement. The network converged with a mean squared error of 0.0002 on the 20 training samples. We note that in [3], both the training data and the testing data were used to determine the appropriate number of training iterations to prevent over learning of their network. Another noteworthy advantage of this approach is that unimportant features can be detected during training [10]. In this case, the weight for the input x_4 stabilized at 0.003, with $\gamma = 0.484$ for that node. Since the node in question is more intersection than union, very small weights signify unimportance of the feature for the training data, i.e., even a very low membership value does not contribute much to the calculation of the intersection.

Conclusions

In this paper, we demonstrated how networks of weighted Yager additive hybrid operators could be used for control. The advantages of this approach are that the networks are trainable, the operators possess useful theoretical properties, unimportant features can be detected during training, and after training, individual nodes can be interpreted in a linguistic fashion as "mini-rules".

References

1. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy logic Controller, Parts I & II", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-20, No. 2, 1990, pp. 404-435.

2. Y. Hayashi, E. Czogala, and J. Buckley, "Fuzzy Neural Controller", *Proceedings, IEEE International Conference on Fuzzy Systems*, San Diego, CA, March, 1992, pp. 197-202.

3. H. Takagi and I. Hayashi, "NN-Driven Fuzzy Reasoning", *International Journal of Approximate Reasoning*, Vol. 5, 1991, pp. 191-212.

4. D. White and D. Sofge, Eds., *Handbook of Intelligent Control*, Van Nostrand, 1992.

5. J. Nie and D. Linkens, "Fuzzy Reasoning Implemented by Neural Networks", *Proceedings, International Joint Conference on Neural Networks*, Vol II, Baltimore, MD, June, 1992, pp. 702-706.

6. I. Hayashi, H. Nomura, H. Yamasaki, and N. Wakami, "Construction of Fuzzy Inference Rules by NDF and NDFL", *International Journal of Approximate Reasoning*, Vol. 6, 1992, pp. 241-266.

7. J. Keller, R. Krishnapuram, Z. Chen, and O. Nasraoui, "Fuzzy Additive Hybrid Operators for Network-Based Decision Making", submitted for publication.

8. J. Keller and Z. Chen, "Learning in Fuzzy Neural Networks Utilizing Additive Hybrid Operators", *Proceedings, Second International Conference on Fuzzy Logic and Neural Networks*, Iizuka, Japan, July, 1992, pp. 85-87.

9. J. Keller, Y. Hayashi, and Z. Chen, "Interpretation of Nodes in Neural Networks for Fuzzy Logic", *Proceedings, Second IEEE International Conference on Fuzzy Systems*, San Francisco, CA, March/April, 1993.

10. J. Keller and Z. Chen, "Detecting Unimportant Features in Additive Hybrid Operator Networks", to appear, *Proceedings, ANNIE'92*, St. Louis, MO., Nov., 1992.

11. J. Keller, Y. Hayashi, and Z. Chen, "Additive Hybrid Networks for Fuzzy Logic Inference", submitted for publication, 1992.

12. R. Krishnapuram and J. Lee, "Fuzzy-Connective-Based Hierarchical Aggregation Networks for Decision Making", *Fuzzy Sets and Systems*, Vol. 46, 1992, pp. 1-17.

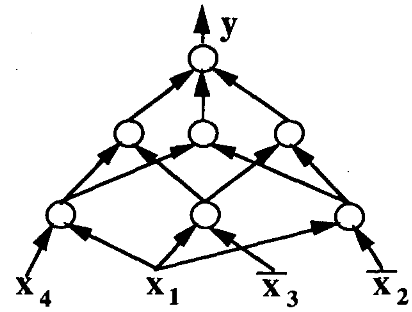

Figure 1. Additive Hybrid Network for Fuzzy Control Example.

Table 1. Training and Testing Results for Additive Hybrid Network

Training Data and Results (left six columns) — **Testing Data and Results** (right six columns)

x_1	$\bar{x}_2$	$\bar{x}_3$	x_4	Y	y	x_1	$\bar{x}_2$	$\bar{x}_3$	x_4	Y	y
.160	.520	.840	.160	.370	.356	.160	.840	.200	.160	.318	.329
.160	.200	.680	.160	.217	.230	.160	.520	.360	.160	.201	.209
.160	.840	.520	.800	.340	.388	.160	.200	.520	.800	.191	.194
.160	.520	.360	.800	.201	.204	.160	.840	.680	.800	.375	.374
.160	.200	.200	.160	.175	.173	.160	.520	.840	.160	.370	.361
.800	.840	.360	.160	.634	.625	.800	.200	.680	.160	.479	.494
.800	.520	.520	.800	.472	.470	.800	.840	.520	.800	.654	.660
.800	.200	.680	.800	.479	.487	.800	.520	.360	.800	.455	.449
.800	.840	.840	.160	.914	.863	.800	.200	.200	.160	.414	.422
.800	.520	.680	.160	.513	.531	.800	.840	.360	.160	.634	.630
.160	.200	.520	.800	.191	.190	.160	.520	.520	.800	.213	.222
.160	.840	.360	.800	.326	.326	.160	.200	.680	.800	.217	.235
.160	.520	.200	.160	.196	.202	.160	.840	.840	.160	.533	.422
.160	.200	.360	.160	.180	.175	.160	.520	.640	.160	.241	.247
.160	.840	.520	.800	.340	.388	.160	.200	.520	.800	.191	.194
.800	.520	.680	.800	.513	.513	.800	.840	.360	.800	.634	.630
.800	.200	.840	.160	.656	.640	.800	.520	.200	.160	.446	.438
.800	.840	.680	.160	.702	.721	.800	.200	.360	.160	.423	.422
.800	.520	.520	.800	.472	.470	.800	.840	.520	.800	.654	.660
.800	.200	.360	.800	.423	.422	.800	.520	.680	.800	.513	.537