Approximations Between Fuzzy Expert Systems and Neural Networks

Yoichi Hayashi
Ibaraki University, Hitachi-shi, Ibaraki 316, Japan
James J. Buckley
University of Alabama at Birmingham, Alabama

ABSTRACT

The fuzzy expert system we are concerned about in this paper is a rule-based fuzzy expert system using any method of approximate reasoning to evaluate the rules when given new data. In this paper we argue that: (1) any continuous fuzzy expert system may be approximated by a neural net; and (2) any continuous neural net (feedforward, multilayered) may be approximated by a fuzzy expert system. We show how to train the neural net and how to write down the rules in the fuzzy expert system.

KEYWORDS: ***Neural networks, fuzzy expert systems, approximations***

1. INTRODUCTION

The fuzzy expert system we are concerned about in this paper is a continuous rule-based fuzzy expert system using any method of approximate reasoning to evaluate the rules when given new data. There will be no uncertainties in the data or the rules, and there is no thresholding in the firing of a rule. That is, all rules fire given new data. The neural nets will be continuous (no thresholding within neurons) feedforward, multilayered, employing any learning algorithm [1].

In the next section we begin by showing that a three-layered neural net can be trained to approximate a given discrete, continuous, fuzzy expert system uniformly, to any degree of accuracy, over all inputs. The motiva-

---

Address correspondence to Dr. Yoichi Hayashi, Department of Computer & Information Sciences, Ibaraki University, Hitachi-shi, Ibaraki 316, Japan.

A Preliminary version of this paper was presented at the Second International Conference on Fuzzy Logic and Neural Networks (IIZUKA'92), Iizuka, Japan, July 17–22, 1992.

Received November 17, 1992; accepted August 15, 1993.

International Journal of Approximate Reasoning 1994; 10:63–73

655 Avenue of the Americas, New York, NY 10010 0888-613X/94/$7.00

tion for this part of our research is to build fast parallel computation (a neural net) for discrete fuzzy expert systems. In [2] we discussed approximating fuzzy expert systems by neural nets, but the discussion was based on an existence proof. That is, used results that state continuous function can be approximated uniformly on compact sets by neural networks. In this study we show how to train a three-layered neural net, having a sufficient number of neurons in the hidden layer, to approximate a given continuous, discrete, fuzzy expert system. In [3] the authors also train neural nets to approximate fuzzy rules in a fuzzy expert system. However, they train a neural net to approximate one or two explicit fuzzy rules. Our results are more general in that our continuous, discrete, fuzzy expert system has any number of rules and uses any method of approximate reasoning to infer its final conclusion.

In the second part of the next section we explain how to construct a continuous, discrete, fuzzy expert system to approximate a given neural net uniformly, to any degree of accuracy, over all inputs. We are not suggesting replacing neural nets by fuzzy expert systems. This is an important theoretical result showing that fuzzy expert systems can be computationally equivalent to neural nets. In [2] we also discussed this approximation result, but there we presented more of an existence proof. We showed that given the neural net, we can build a discrete fuzzy expert system, in particular we showed that there exists a method of approximate reasoning, to approximate the net. In this paper we present a more general discussion on the construction of the discrete fuzzy expert system.

We will use a bar over a symbol to represent a fuzzy set. So $\overline{A}, \overline{B}, \overline{C}, \ldots$ are all fuzzy sets. Also, all our fuzzy sets will be subsets of the real numbers. If $\overline{A}$ is a fuzzy set, then $\overline{A}(x)$ denotes its membership function evaluated at real number x. Similarly, $\overline{B}(x), \overline{C}(x)$ are the membership functions for $\overline{B}, \overline{C}$, respectively.

2. APPROXIMATIONS

We will first discuss how to approximate a fuzzy expert system with a neural net and then use a fuzzy expert system to approximate a neural net.

2.1. Fuzzy Expert System

Consider a fuzzy expert system (abbreviated FES) having one block of rules

$$\mathscr{R}_r\text{: If } X = \overline{A}_i \text{ and } Y = \overline{B}_j, \text{ then } Z = \overline{C}_p, \tag{1}$$

for $1 \le r \le n$. For simplicity we have assumed that each rule has only two simple clauses in its antecedent. Also, we are using fuzzy sets directly in the rules instead of having their equivalent linguistic values. In equation (1) we connected the two clauses with an "and," but one can use "and" or "or" in the rules. We will assume that: (1) the interval $[a_1, b_1]$ contains the support of all the fuzzy sets than can be used for X; (2) $[a_2, b_2]$ contains all the fuzzy sets that may be used for Y; and (3) $[a_3, b_3]$ has the support of all the fuzzy sets that can be identified with variable Z.

The expert system will use some method of approximate reasoning to evaluate the rules when given data $X = \overline{A}'$ and $Y = \overline{B}'$. One method of approximate reasoning involves: (1) choosing an implication operator; (2) picking a method of composing the data $X = \overline{A}', Y = \overline{B}'$, with the information in a rule; and (3) deciding on how to combine the results of each rule into a final conclusion [4]. There are other methods like first combining all the rules into one fuzzy relation [4], but we need not exactly specify any particular procedure of approximate reasoning in this paper. We now assume some method of approximate reasoning has been chosen, and we will denote this method by $\mathscr{AR}$. So, given the data $X = \overline{A}'$ and $Y = \overline{B}'$, the block of rules $\mathscr{R}_r$, $1 \le r \le n$, and $\mathscr{AR}$, the fuzzy expert system produces its conclusion $Z = \overline{C}'$.

Now assume we run this fuzzy expert system on some test data $X = \overline{A}'_k, Y = \overline{B}'_k$, $1 \le k \le K$. Let the corresponding conclusions be $Z = \overline{C}'_k$, $1 \le k \le K$. In a computer one usually uses discrete versions of the continuous fuzzy sets $\overline{A}_i, \overline{B}_j, \overline{C}_p, \overline{A}', \overline{B}', \overline{C}'$, etc. So let $\mathscr{D}$ be a discretization of the intervals $[a_i, b_i]$, $1 \le i \le 3$. In this paper we will choose the following discretization: (1) pick x_i in $[a_1, b_1]$ as $x_0 = a_1, x_i = a_1 + i(b_1 - a_1)/N_1$ $1 \le i \le N_1$, for positive integer N_1; (2) choose y_i in $[a_2, b_2]$ as $y_0 = a_2, y_i = a_2 + i(b_2 - a_2)/N_2$, $1 \le i \le N_2$, for positive integer N_2; and (3) let z_i be in $[a_3, b_3]$ so that $z_0 = a_3, z_i = a_3 + i(b_3 - a_3)/N_3$, $1 \le i \le N_3$, N_3 a positive integer. $\mathscr{D}$ consists of $x_0, \dots, x_{N_1}, y_0, \dots, y_{N_2}, z_0, \dots, z_{N_3}$. Let $M = N_1 + N_2$. Then we input the numbers $\overline{A}'_k(x_i)$, $0 \le i \le N_1$, for $X = \overline{A}'_k$ and the numbers $\overline{B}'_k(y_i)$, $0 \le i \le N_2$, and $Y = \overline{B}'_k$, into the fuzzy expert system and obtain the numbers $\overline{C}'_k(z_i)$, $0 \le i \le N_3$, for $Z = \overline{C}'_k$, $1 \le k \le K$.

We now construct, and train, a neural network that will compute (approximately) the same results as the fuzzy expert system for the inputs $\overline{A}'_k(x_i)$, $0 \le i \le N_1$, $\overline{B}'_k(y_i)$, $0 \le i \le N_2$, for $1 \le k \le K$. That is, the neural net computes the same as the fuzzy expert system, with respect to $\mathscr{D}$, for the test data $X = \overline{A}'_k, Y = \overline{B}'_k$, $1 \le k \le K$.

It is well-known that neural nets are universal approximators. What this means is that given a continuous F: $\boldsymbol{R}^d \to \boldsymbol{R}$ there is a neural net that can uniformly approximate F, to any degree of accuracy, on compact subsets of $\boldsymbol{R}^d$. See [5–17] for a survey of this literature. From the discussion above we see that a discrete fuzzy expert system FES will be a mapping from $[0, 1]^d$ into $[0, 1^e$, for $d = M + 2, e = N_3 + 1$. We now argue that we would normally expert FES to be a continuous mapping.

In [18] it is shown that Zadeh's compositional rule of inference is a continuous operation on discrete fuzzy sets, when it is based on a continuous t-norm. We would therefore expect that the method of approximate reasoning, used within the fuzzy expert system is also continuous. This then implies that the FES is a continuous mapping from $[0, 1]^d$ into $[0, 1]^e$. It can be shown that there is a neural net that can approximate FES, uniformly to any degree of accuracy, on compact $[0, 1]^d$. However, this argument is only an existence argument and it does not tell you how to construct and train the neural net. So, for the rest of this subsection we will discuss how to obtain a neural net that will approximate the given FES.

The neural net will have $M + 2$ input neurons, one hidden layer, and $N_3 + 1$ output neurons. There are different methods of specifying a sufficient number of neurons in the hidden layer ([19–21]), and we assume that one such method has been chosen so that the hidden layer has a sufficient number of neurons to learn the training set. Label the input neurons $I_1, I_2, \ldots, I_{M+2}$, and the output neurons $O_1, \ldots, O_{N_3+1}$. We now describe how to train the neural network. We input $\overline{A}'_k(x_0)$ to $I_1, \ldots, \overline{A}'_k(x_{N_1})$ to $I_{N_1+1}, \overline{B}'_k(y_0)$ to $I_{N_1+2}, \ldots, \overline{B}'_k(y_{N_2})$ to I_{M+2} and the desired output is $\overline{C}'_k(z_0)$ from $O_1, \ldots, \overline{C}_k(z_{N_3})$ from O_{N_3+1}. That is, the training set has K pairs of inputs-outputs with $\{\overline{A}'_k(x_i) | 0 \leq i \leq N_1\} \cup \{\overline{B}'_k(y_i) | 0 \leq i \leq N_2\}$ the input set and $\{\overline{C}'_k(z_i) | 0 \leq i \leq N_3\}$ the output set. Figure 1 shows how the neural net will approximate the fuzzy expert system for the special case of $N_1 = N_2 = N_3 = 10$. Then the neural net and the fuzzy expert system compute the same output, with respect to $\mathscr{D}$, for the test data $X = \overline{A}'_k, Y = \overline{B}'_k, 1 \leq k \leq K$.

Now one wonders how the outputs, from the two systems, compare if we input new data $X = \overline{A}'$ and $Y = \overline{B}'$ where these fuzzy sets do not belong to the test data set. The result depends on how we picked the test data set. Suppose first that we chose $\overline{A}'_k = \overline{A}_i, \overline{B}'_k = \overline{B}_j, \overline{C}'_k = \overline{C}_p$, the fuzzy sets in the rules. That is, we do not run the fuzzy expert system to compute $Z = \overline{C}'_k$ but set the input pair to be the fuzzy sets in the antecedent of a rule and the output fuzzy set is the fuzzy set in the rule's consequence. The test set has all the fuzzy sets in the rules and no other fuzzy sets. We then train the neural net only on the knowledge base (rules) of the fuzzy expert system and the net will know nothing about $\mathscr{A}\mathscr{R}$. So, the two systems can differ considerably for new data $X = \overline{A}'$ and $Y = \overline{B}'$. However, if the test data set has a number of pairs $\{\overline{A}'_k, \overline{B}'_k\}$, where these fuzzy sets do not belong to some rule's antecedent, and use $\mathscr{A}\mathscr{R}$ to compute $Z = \overline{C}'_k$, then the net trained on this information will incorporate some of $\mathscr{A}\mathscr{R}$ into its

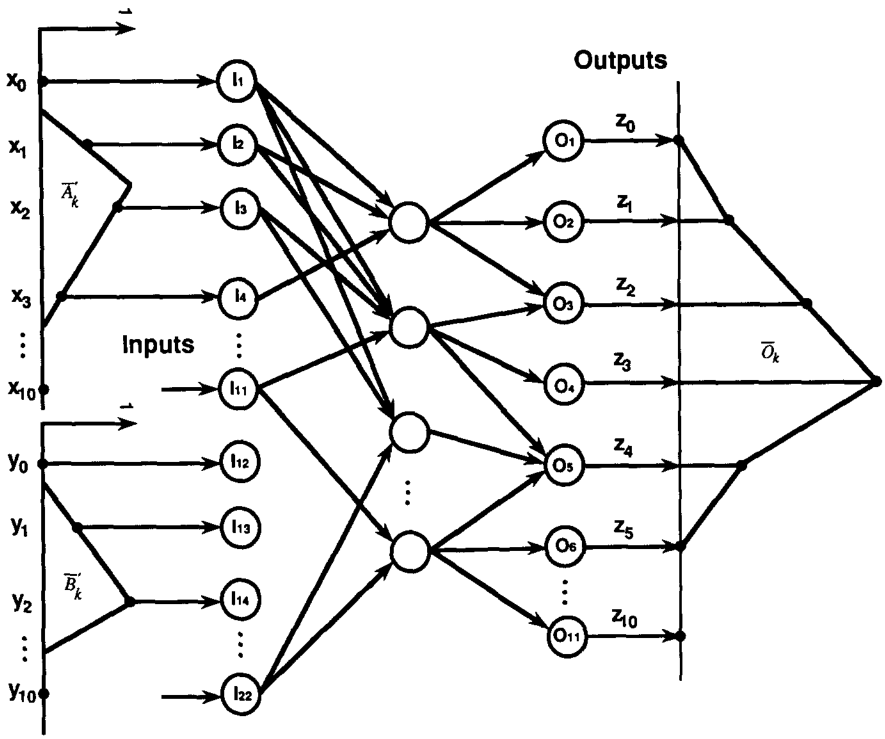

Figure 1. Neural net approximating fuzzy expert system.

weights. Then the two systems can produce similar results for new data $X = \overline{A}'$ and $Y = \overline{B}'$. So, one should pick the test data so that the input pairs $(\overline{A}'_k, \overline{B}'_k)$ broadly cover applications of the fuzzy expert system and then train the neural net on this information. Then the net will better approximate the FES on new data.

In the appendix we present a more formal argument (mathematical) on why you should not use only the rules in the FES as the training set for the neural network.

2.2. Neural Net

Consider a continuous neural net (NN) with m input neurons, any number of hidden layers, and n output neurons. Assume that all the input, and output, signals are bounded between zero and one. Therefore, NN is a continuous mapping from $[0, 1]^m$ into $[0, 1]^n$.

Recently, there have been a number of papers showing that certain types of fuzzy systems are universal approximators ([22–28]). All of these fuzzy systems resemble a fuzzy controller in that they have singleton (crisp) inputs and defuzzified (crisp) output. We may use these results to obtain a generalized fuzzy system (multiple outputs) that can approximate NN uniformly, to any degree of accuracy, over $[0,1]^m$. However, we shall not pursue that idea in this paper but instead we will be interested in building a fuzzy expert system to approximate NN. This FES will have one block of rules, we will use a fuzzy relation to model the implication in each rule, use Zadeh's compositional rule of inference to evaluate each rule given new data, and finally combine the outputs from all the rules into one final conclusion. Input, and output, from the FES will be discrete versions of continuous fuzzy sets. The arguments that fuzzy systems are universal approximators are existence proofs, they do not show you how to build the approximating fuzzy system. Our method is more constructive in that we show how to construct rules and we can specify the method of approximate reasoning to be used to evaluate the rules.

We first choose w_j, $1 \leq j \leq J$, uniformly spread around $[0,1]^m$ and let $NN(w_j) = q_j$, $1 \leq j \leq J$. We are using functional notation where $w_j \in [0,1]^m$ is input to the neural net and $NN(w_j)$ is its output, a vector q_j in $[0,1]^n$. The FES will have one block of rules

$$\mathscr{R}_j\text{: If } X = \overline{A}_j, \text{ then } Z = \overline{C}_j, 1 \leq j \leq J. \tag{2}$$

The interval that contains all the fuzzy sets for $X(Z)$ is $[1,m]([1,n])$. The discretization of these intervals is: (1) $x_0 = 1, x_1 = 2, \ldots, x_{m-1} = m$; and (2) $z_0 = 1, z_1 = 2, \ldots, z_{n-1} = n$. We define $\overline{A}_j$ and $\overline{C}_j$ with respect to this discretization as follows: (A) $\overline{A}_j^{(i)}$ = the i^{th} component of w_j, $1 \leq i \leq m$; and (2) $\overline{C}_j^{(i)}$ = the i^{th} component of q_j, $1 \leq i \leq n$. The fuzzy sets $\overline{A}_j$ and $\overline{C}_j$ have no physical meaning nor do they represent linguistic variables. They are defined to match the input (w_j)-output (q_j) pairs from NN so that the FES will be able to uniformly approximate the neural net.

Next we need to specify the method of approximate reasoning used to evaluate the rules given input ν in $[0,1]^m$ on X. That is, data on X will be $\overline{A}'$, a fuzzy subset of $[0,m]$, and the fuzzy expert system then concludes $Z = \overline{C}'$, a fuzzy subset of $[0,n]$. However, discrete versions of these fuzzy sets will be used so that the input data will be $X = \nu$ in $[0,1]^m$, where $\overline{A}'(i)$ = the i^{th} component of ν for $1 \leq i \leq m$, and the output from the FES will be $Z = u$ in $[0,1]^n$, where $\overline{C}'^{(i)}$ = the i^{th} component of u for $1 \leq i \leq n$. In functional notation $FES(\nu) = u$.

The main requirement of the method of approximate reasoning is $FES(w_j) = q_j$ all j, or $FES(w_j) = NN(w_j)$ all j. What this means is if

$X = \overline{A}' = \overline{A}_j$ (discrete version w_j) in rule $\mathscr{R}_j$, then the final conclusion from the expert system is $Z = \overline{C}' = \overline{C}_j$ (discrete version q_j), for all rules. We first must guarantee that this will happen for each rule because the rules will all fire separately, and then we will combine their results to get final output $Z = \overline{C}'$.

We first combine the data ($\overline{A}_i$ and $\overline{C}_i$) in each rule $\mathscr{R}_i$ into a discrete fuzzy relation $\overline{R}_i$ on $[0, m] \times [0, n]$. Given input data $X = \nu$ in $[0,1]^m$ each rule fires producing conclusion $Z = \nu \circ \overline{R}_j$, for some composition operator "$\circ$." Then the FES combines the results $\nu \circ \overline{R}_j$ across all rules ($1 \leq j \leq J$) into its final conclusion (output) $Z = u$ in $[0,1]^n$. We need to choose the $\overline{R}_j$ and "$\circ$" so that $w_j \circ \overline{R}_j = q_j$ all j. There are a number of different choices if the fuzzy sets are normalized (at least one component of w_j is equal to one, all j). However, our (discrete) fuzzy sets are not necessarily normalized since $w_j = 0$ (all components zero) could be a choice for a w_j. But, there are methods of approximate reasoning ([2]) with the property $w_j \circ \overline{R}_j = q_j$ all j, for any w_j in $[0,1]^m$. Then we have $w_j \circ \overline{R}_j$ equal to q_j for each rule $\mathscr{R}_j$. All that is left to do is to specify how the FES combines the results into its final output. For any ν in $[0,1]^m$, the FES averages the results $w_j \circ \overline{R}_j$, for those w_j nearest to ν in $[0,1]^m$.

Then given $\varepsilon > 0$, we choose the w_j, $1 \leq j \leq J$, uniformly spread around $[0,1]^m$, construct the FES as described above, and obtain $|NN(\nu) - FES(\nu)| < \varepsilon$ for all ν in $[0,1]^m$. That is, we may build an FES to approximate a neural net, uniformly to any degree of accuracy, over all inputs from $[0,1]^m$.

3. SUMMARY AND CONCLUSIONS

In this paper we first argued that given any continuous, discrete, rule-based fuzzy expert system using any method of approximate reasoning to evaluate the rules, we can build and train a neural net to uniformly approximate, to any degree of accuracy, the fuzzy expert system. We also argued that you should not train the neural net only on the rules in the fuzzy expert system. We showed (in the Appendix) how to train the net to uniformly approximate the fuzzy expert system. A practical application of this result is to substitute fast parallel computation (the neural net) for a fuzzy expert system.

Next we argued that given any continuous, multilayered, feedforward neural net, one can construct a fuzzy expert system to uniformly approximate, to any degree of accuracy, the neural net. We showed how to construct the rules in the fuzzy expert system and discussed the properties needed in the method of approximate reasoning to obtain the desired result. Mathematical details may be found in [2].

Both approximation results together say that certain neural nets and fuzzy expert systems are computationally equivalent.

APPENDIX

In this appendix we first show how to train the neural net so that it can uniformly approximate the FES over all possible inputs. Then we argue that one should not train the network only on the rules in the fuzzy expert system.

Let $\nu \in [0,1]^d$ be a possible input to the FES or to the neural net (NN). That is, $\nu = (\bar{A}'(x_0), \bar{A}'(x_1), \ldots, \bar{B}'(y_{N_2}))$. We have assumed that the FES is a continuous mapping from $[0,1]^d$ into $[0,1]^e$ so it is uniformly continuous. What this means is that given $\varepsilon > 0$ there is a $\delta_1 > 0$ so that $|FES(\nu_1) - FES(\nu_2)| < \varepsilon/2$ if $|\nu_1 - \nu_2| < \delta_1, \nu_1, \nu_2$ in $[0,1]^d$. We are using the functional notation of ν_i input to FES producing $FES(\nu_i)$ as (discrete) output, a vector in $[0,1]^e$.

Let NN be a continuous three-layered, feedforward, neural net with d input neurons, e output neurons, and a sufficient number of neurons in the hidden layer so that it can learn a data set of size L. Suppose that u_l, $1 \leq l \leq L$, is a set of vectors spread around $[0,1]^d$. Let $u_l' = FES(u_l)$ a vector in $[0,1]^e$, $1 \leq l \leq L$. We have assumed that NN can learn (u_l, u_l'), $1 \leq l \leq L$, which means that the weights can be adjusted so that $NN(u_l) = u_l'$, $1 \leq l \leq L$.

NN is a continuous mapping from $[0,1]^d$ into $[0,1]^e$, all signals are in the interval $[0,1]$, so it is uniformly continuous. Therefore, there is a $\delta_2 > 0$ so that $|NN(\nu_1) - NN(\nu_2)| < \varepsilon/2$ if $|\nu_1 - \nu_2| < \delta_2, \nu_1, \nu_2$ in $[0,1]^d$. Again we are using functional notation with ν_i input to the neural net and $NN(\nu_i)$ its output vector in $[0,1]^e$. Define δ to be the minimum of δ_1 and δ_2.

Choose w_j, $1 \leq j \leq J$, uniformly spread around $[0,1]^d$ with the property that given any ν in $[0,1]^d$ there is a w_j so that $|\nu - w_j| < \delta$. Assume that $J \leq L$.[1]

Let $FES(w_j) = q_j$, $1 \leq j \leq J$. The learning data for NN will be (w_j, q_j), $1 \leq j \leq J$. That is, train the NN so that $NN(w_j) = q_j$ all j.

We now argue that this NN will uniformly approximate the FES. Let ν be any vector in $[0,1]^d$ and choose w_j also in $[0,1]^d$ so that $|\nu - w_j| < \delta$.

[1]We know there is an NN (sufficient number of neurons in the hidden layer) that will approximate FES, uniformly over all inputs ν in $[0,1]^d$, to any degree of accuracy $\varepsilon > 0$. So this NN will have L large enough.

Then

$$|FES(\nu) - NN(\nu)| = |FES(\nu) - FES(w_j) + NN(w_j) - NN(\nu)|$$
$$\leq |FES(\nu) - FES(w_j)| + |NN(w_j) - NN(\nu)|$$
$$< \varepsilon/2 + \varepsilon/2 = \varepsilon$$

because: (1) $FES(w_j) = NN(w_j)$; (2) $|\nu - w_j| < \delta_1$; and (3) $|\nu - w_j| < \delta_2$. So, this neural net will compute, within ε, the same as the FES, across all possible inputs.

Now suppose you train the NN only on the rules (knowledge base). Let $s_r \in [0,1]^d$ denote the discretization of A_i and B_j in the antecedent of rule $\mathscr{R}_r$, $1 \leq r \leq n$. Next let t_r in $[0,1]^e$ be the discretization of $\overline{C}_p$, the fuzzy set in the conclusion of rule $\mathscr{R}_r$, $1 \leq r \leq n$. Train the NN on the set (s_r, t_r), $1 \leq r \leq n$, so that $NN(s_r) = t_r$ all r. Now pick any ν in $[0,1]^d$. We would not expect $|FES(\nu) - NN(\nu)|$ to be small because: (1) the s_r, $1 \leq r \leq n$ does not necessarily form a uniform span of the input space $[0,1]^d$; and (2) it may happen, depending on $\mathscr{AR}$, that $FES(s_r) \neq t_r$ for some rules. That is, if ν is not near any s_r, then we would not expect $NN(\nu)$ to be close to $FES(\nu)$. Also, NN may differ from FES even on the training set. For these reasons we do not recommend training the neural net only on the rules of the fuzzy expert system.

References

1. Rumelhart, D. D., Hinton, G. E., and Williams, R. J., Learning internal representations by error propagation, in *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Vol. 1, (D. E. Rumelhart and J. L. McClelland, eds.), MIT Press, Cambridge, MA, 675–695, 1986.

2. Buckley, J. J., Hayashi, Y., and Czogala, E., On the equivalence of neural nets and fuzzy expert systems, *Fuzzy Sets and Systems* 53, 129–134, 1993.

3. Keller, J. M., and Tahani, H., Implementation of conjunctive and disjunctive fuzzy logic rules with neural networks, *Int. J. Approx. Reasoning* 6, 221–240, 1992.

4. Dubois, D., and Prade, H., Fuzzy sets in approximate reasoning, part I: inference and possibility distributions, *Fuzzy Sets and Systems* 40, 143–202, 1991.

5. Blum, E. K., and Li, L. K., Approximation theory and feedforward networks, *Neural Networks* 4, 511–515, 1991.

6. Cardaliaguest, P., and Euvrard, G., Approximation of a function and its derivative with a neural network, *Neural Networks* 5, 207–220, 1992.

7. Cotter, N. E., and Guillerm, T. J., The CMAC and a theorem of Kolmogorov, *Neural Networks* 5, 221–228, 1992.

8. Cybenko, G., Approximation by superpositions of a sigmoidal function, *Math. of Control, Signals, and Systems*, 2, 303–314, 1989.

9. Geva, S., and Sitte, J., A constructive method for multivariate function approximation by multilayer perceptrons, *IEEE Trans. on Neural Networks*, 3, 621–623, 1992.

10. Hornik, K., Approximation capabilities of multilayer feedforward networks, *Neural Networks* 4, 251–257, 1991.

11. Hornik, K., Stinchcombe, M., and White, H., Multilayer feedforward networks are universal approximators, *Neural Networks* 2, 359–366, 1989.

12. Ito, Y., Approximation of functions on compact sets by finite sums of a sigmoid function without scaling, *Neural Networks* 4, 817–826, 1991.

13. Ito, Y., Approximation of continuous functions on R^d by linear combinations of shifted rotations of a sigmoid function with and without scaling, *Neural Networks* 5, 105–115, 1992.

14. Kreinovich, V. Y., Arbitrary nonlinearity is sufficient to represent all functions by neural networks: a theorem, *Neural Networks* 4, 381–383, 1991.

15. Kůrková, V., Kolmogorov's theorem is relevant, *Neural Computation* 3, 617–622, 1991.

16. Kůrková, V., Kolmogorov's theorem and multilayer neural networks, *Neural Networks* 5, 501–506, 1992.

17. Park, J., and Sandberg, I. W., Universal approximation using radial-basis-function networks, *Neural Computation* 3, 246–257, 1991.

18. Fullér, R., and Zimmermann, H.-J., On Zadeh's compositional rule of inference, *Proc. Fourth IFSA World Congress*, Vol. Artificial Intelligence, Brussels, 41–44, July 7–12, 1991.

19. Huang, S.-C., and Huang, Y.-F., Bounds on the number of hidden neurons in multilayer perceptrons, *IEEE Trans. on Neural Networks* 2, 47–55, 1991.

20. Mirchandani, G., On hidden nodes for neural nets, *IEEE Trans. on Circuits and Systems* 36, 661–664, 1989.

21. Sartori, M. A., and Antsaklis, P. J., A simple method to derive bounds on the size and to train multilayer neural networks, *IEEE Trans. on Neural Networks* 2, 467–471, 1991.

22. Buckley, J. J., Sugeno type controllers are universal controllers, *Fuzzy Sets and Systems* 53, 299–304, 1993.

23. Buckley, J. J., Universal fuzzy controllers, *Automatica* 28, 1245–1248, 1992.

24. Buckley, J. J., Controllable processes and the fuzzy controller, *Fuzzy Sets and Systems* 53, 27–32, 1993.

25. Buckley, J. J., Applicability of the fuzzy controller, in *Advances in Fuzzy Systems: Applications and Theory*, (P.-Z. Wang and K. F. Loe, Eds.) World Scientific Press, Singapore, in press.

26. Kosko, B., Fuzzy systems as universal approximators, *Proc. of IEEE Inter. Conf. on Fuzzy Systems*, San Diego, 1153–1162, March 8–12, 1992.

27. Nguyen, H. T., and Kreinovich, V., On approximation of controls by fuzzy systems. Unpublished manuscript.

28. Wang, L.-X., Fuzzy systems and universal controllers, *Proc. of IEEE Inter. Conf. on Fuzzy Systems*, San Diego, 1163–1170, March 8–12, 1992.