Approximation Paper: Part I

J. J. Buckley
Mathematics Department
University of Alabama at Birmingham
Birmingham, AL 35294

Abstract: In this paper we discuss approximations between neural nets, fuzzy expert systems, fuzzy controllers, and continuous processes.

1. Introduction.

In this section we will first present the definition of a continuous process (system). The following sections discuss neural networks, fuzzy expert systems, the fuzzy controller, and approximations between all four objects. The last section has a brief summary and suggestions for future research.

A system (process) S has m inputs x_i and n outputs y_j. Let $x = (x_1, \cdots, x_m)$ and $y = (y_1, \cdots, y_n)$. The inputs are all bounded so assume that each input is scaled to belong to $[0,1]$. This means that S will be a mapping from $[0,1]^m$ into R^n written as $y = S(x)$. We assume that S is continuous and let $\mathcal{S}$ denote the set of all continuous mappings from $[0,1]^m$ into R^n. By a continuous process (system) we will mean any S in $\mathcal{S}$.

2. Neural Nets.

The neural network will be a layered, feedforward, net with m input neurons and n output neurons. The net can have any number of hidden layers. Input to the net will be a vector $x = (x_1, \cdots, x_m)$, x_i in $[0,1]$ all i, and the output is also a vector $y = (y_1, \cdots y_n)$. We assume that the activation function[1] within a neuron is continuous. Therefore, the neural net is a continuous mapping from input x in $[0,1]^m$ to output y in R^n denoted as $y = F(x)$. We note that F belongs to $\mathcal{S}$.

The following result comes from recent publications in the neural network literature ([1], [9], [14], [15], [16]) where it was shown that multilayer feedforward nets are universal approximators. Given S in $\mathcal{S}$ and $\epsilon > 0$ there is a neural net F so that $|S(x) - F(x)| < \epsilon$ for all x in $[0,1]^m$, see [8].

3. Fuzzy Expert Systems.

The fuzzy expert system will contain one block of rules written as[2]

$$\mathcal{R}_i \; : \; \text{If } X = \bar{A}_i, \text{ then } Z = \bar{C}_i,$$

$1 \leq i \leq N$. $\bar{A}_i$ and $\bar{C}_i$ represent fuzzy subsets of the real numbers. If $\bar{A}$ denotes any fuzzy subset of the reals, then $\bar{A}(x)$ is its membership function evaluated at x.

Let $X = \bar{A}'$ be the input to the fuzzy expert system. The rules are evaluated using some method of approximate reasoning (fuzzy logic) producing final conclusion (output) $Z = \bar{C}'$. Let $\mathcal{A}$ denote the type of approximate reasoning employed by the fuzzy expert system.

We now discretize all the fuzzy sets. Let $x_0, \cdots, x_{N_1}$ be numbers covering the support of all the $\bar{A}_i$ and $\bar{A}'$ and let $z_0, \cdots, z_{N_2}$ be numbers covering the support of all the $\bar{C}_i$ and $\bar{C}'$. Let $x = \left(\bar{A}'(x_0), \cdots, \bar{A}'(x_{N_1})\right)$ in $[0,1]^m$ if $m = N_1 + 1$ and let $y = \left(\bar{C}'(z_0), \cdots, \bar{C}'(z_{N_2})\right)$ in R^n if $n = N_2 + 1$. Then x is the input to the fuzzy expert system and y is its output. So, the fuzzy expert system is a mapping from x in $[0,1]^m$ to y in R^n which we write as $y = G(x)$. We assume that we have selected an $\mathcal{A}$ so that this mapping is continuous. Hence, G also belongs to $\mathcal{S}$.

The first papers discussing the approximation of a neural net by a fuzzy expert system were ([7], [10]) but the main result was proven in [8]. Given a neural net F and $\epsilon > 0$ there exists a fuzzy expert system (block of rules and $\mathcal{A}$) so that $|F(x) - G(x)| < \epsilon$ for all x in $[0,1]^m$. In [8] we found only one $\mathcal{A}$ that will do the job. From the second section we may conclude that given any S in $\mathcal{S}$ and $\epsilon > 0$ there is a fuzzy expert system G so that $|S(x) - G(x)| < \epsilon$ for all x in $[0,1]^m$.

4. Fuzzy Controller.

It will be easier now if we restrict $m = 2$ and $n = 1$, however we can generalize to other values of m and n. Let us assume that the fuzzy controller has only two inputs error = e and change in error = Δe, and only one defuzzified output δ. We assume that the inputs have been scaled to lie in $[0,1]$. The fuzzy control rules are of the form

$$\begin{aligned} \mathcal{R} \; : & \text{ If Error } = \bar{A}_i \text{ and Change in Error } = \bar{B}_j, \\ & \text{ then Control } = \bar{C}_k. \end{aligned}$$

Once a method of evaluating the rules has been chosen and a procedure for defuzzification is adopted, the fuzzy controller is a mapping H from $(e, \Delta e)$ in $[0,1]^2$ to δ in R. We assume that the internal operations within the controller are continuous so that H belongs to $\mathcal{S}$ for $m = 2$, $n = 1$. In general, we can have H in $\mathcal{S}$ for any m and n.

Different types of fuzzy controllers are discussed in ([2], [3]). In [5] and [6] we identified two types of fuzzy controller, now labeled $\mathcal{T}_1$ and $\mathcal{T}_2$, that can approximate any S in $\mathcal{S}$ to any degree of accuracy. A different type of approximation result of S in $\mathcal{S}$, by fuzzy controllers, is presented in [4]. Let this third type of controller be called $\mathcal{T}_3$. So, given S in $\mathcal{S}^3$, $\epsilon > 0$ and i in $\{1,2,3\}$, there is a fuzzy controller H in $\mathcal{T}_i$ so that $|S(x) - H(x)| < \epsilon$ for all x in $[0,1]^2$. Hence, from the previous two sections, we may approximate fuzzy expert systems and neural nets, to any degree of accuracy, by fuzzy controllers.

5. Conclusions.

The results discussed in this paper may be summarized as follows: given any two objects E_1 and E_2 from the set {continuous process, neural net, fuzzy expert system, fuzzy controller}, we can use an E_1 to approximate an E_2 to any degree of accuracy. Assumptions needed to obtain this result are discussed within the paper.

Future research is needed to extend these results in many directions including: (1) fuzzy neural nets ([11], [12]); (2) neural nets that employ t-norms and t-conorms to process information [13]; (3) finding more fuzzy expert systems ($\mathcal{A}$'s) that can be used to approximate neural nets; and (4) discovering other types of fuzzy controllers that approximate continuous systems.

6. References.

1. E. K. Blum and L. K. Li, Approximation Theory and Feedforward Networks, Neural Networks 3 (1991) 511–515.
2. J. J. Buckley, Theory of the Fuzzy Controller: An Introduction, Fuzzy Sets and Systems. To appear.
3. J. J. Buckley, Theory of the Fuzzy Controller: A Brief Survey, in: C. V. Negoita (ed.), Handbook of Cybernetics and Systems, Marcel Dekker, N. Y.. To appear.

4. J. J. Buckley, Controllable Processes and the Fuzzy Controller, Fuzzy Sets and Systems. Submitted.

5. J. J. Buckley, Sugeno Type Controllers are Universal Controllers, Fuzzy Sets and Systems. Submitted.

6. J. J. Buckley, Universal Fuzzy Controllers, Automatica. Submitted.

7. J. J. Buckley and E. Czogala, Fuzzy Models, Fuzzy Controllers and Neural Nets, Frontiers of Applied Mathematics, SIAM. To appear.

8. J. J. Buckley, Y. Hayashi and E. Czogala, On the Equivalence of Neural Nets and Fuzzy Expert Systems, Fuzzy Sets and Systems. Submitted.

9. G. Cybenko, Approximation by Superpositions of a Sigmoidal Function, Math. of Control, Signals, and Systems 2 (1989) 303–314.

10. Y. Hayashi, J. J. Buckley and E. Czogala, Fuzzy Expert Systems Versus Neural Networks, Int. J. Approximate Reasoning. Submitted.

11. Y. Hayashi, J. J. Buckley and E. Czogala, Fuzzy Neural Network with Fuzzy Signals and Weights, Int. J. Intelligent Systems. Submitted.

12. Y. Hayashi, J. J. Buckley and E. Czogala, Systems Engineering Applications of Fuzzy Neural Networks, J. of Systems Engineering. Submitted.

13. Y. Hayashi, E. Czogala and J. J. Buckley, Fuzzy Max-Min Neural Controller, Proceedings FUZZ IEEE, San Diego, March, 1992. To appear.

14. K. Hornik, Approximation Capabilities of Multilayer Feedforward Networks, Neural Networks 4 (1991) 251–257.

15. K. Hornik, M. Stinchcombe, and H. White, Multilayer Feedforward Networks are Universal Approximators, Neural Networks 2 (1989) 359–366.

16. V. Y. Kreinovich, Arbitrary Nonlinearity is Sufficient to Represent all Functions by Neural Networks: A Theorem, Neural Networks 4 (1991) 381–383.

7. Notes

[1] In general, we assume that the mapping from input to output, for any neuron in the net, is a continuous operation.

[2] We could consider more complicated rules and/or more blocks of rules.

[3] $m = 2$ and $n = 1$. Can generalize.