NOW COMES THE TIME TO DEFUZZIFY NEURO-FUZZY MODELS

Hugues Bersini and Gianluca Bontempi
IRIDIA-CP 194/6
Université Libre de Bruxelles
50, av. Franklin Roosevelt
1050 Bruxelles - Belgium

Fuzzy models present a singular Janus-faced: On the one hand, they are knowledge-based software environments constructed from a collection of linguistic IF-THEN rules, and on the other hand, they realize nonlinear mappings which have interesting mathematical properties like "low-order interpolation" and "universal function approximation". Neuro-fuzzy basically provides fuzzy models with the capacity, based on the available data, to compensate for the missing human knowledge by an automatic self-tuning of the structure and the parameters. A first consequence of this hybridization between the architectural and representational aspect of fuzzy models and the learning mechanisms of neural networks has been to progressively increase and fuzzify the contrast between the two Janus faces: readability or performance.

1 Introduction

Fuzzy models present a singular Janus-faced: On the one hand, they are knowledge-based software environments constructed from a collection of linguistic IF-THEN rules, and on the other hand, they implement nonlinear mappings which have interesting mathematical properties like "low-order interpolation" and "universal function approximation". In process control, fuzzy models have been recognized as an attractive alternative to classical control schemes for the low-cost and facilitated design of the control laws of partially known, nonlinear and complex processes. However, the “locality” inherent to the way fuzzy models decompose a problem into fuzzy regions makes the design of fuzzy models a time-consuming procedure involving knowledge acquisition and/or the automatic tuning of an exponential number of parameters, especially when facing multi-variables problems. Neuro-fuzzy models have been perceived by a growing number of researchers in the fuzzy community as a promising road to provide fuzzy models with the capacity to automatically self-tune their parameters. This added facility for the automatic tuning of parameters, like for instance the center and variance of membership functions, is often and confusingly designated as a transposing of the backpropagation algorithm instead of, what it really is, a simple implementation of classical optimization methods for a particular parametrized structure.

Clearly the use of optimisation methods for the automatic tuning of fuzzy models is primarily intended, on the basis of the available numerical data, to improve the behaviour of these models for whatever use: automatic function identification, control or classification. Consequently, a first consequence of this hybridization between the architectural and representational aspect of fuzzy models and the learning mechanisms of neural networks has been to progressively increase the contrast between two visions of fuzzy models, which although was existing for ten years, was never as pronounced as today. These two visions occur by lighting respectively one or the other of the two Janus faces: readability or performance.

The first vision of fuzzy models, and by far the most innovating one, is as an original way of representing a mathematical input/output mapping by means of expert-system-like production rules and by harmonizing the continuous nature of the mathematical reality with the discrete nature of our human language. Mamdani's fuzzy controller is the most known and representative instance of such a vision. Adopting this perspective, the emphasis is put, less on the computational cost, the size of the model and the performance of the function (i.e. fine quality of the approximation, classification or control), but essentially on the readability of the model. Computing with words should be according to fuzzy logic's father the ultimate purpose of fuzzy logic.

As it is well known, unquestionable, and abundantly treated in the literature, readability has two fundamental purposes and advantages: improving the human controller/automatic controller interface (they now speak a common language) and making possible to connect low level automatisms with high-level planning decision. The interaction between these two cognitive levels is a key aspect of human psychology, and fuzzy logic should contribute a lot to reproduce such an interaction in artificial systems. Roughly speaking, readability is required both at the input and output sides of the model and a decomposition of the model on each salient variable together with a unique definition of all linguistic term turns necessary. Although readable, but somewhat paradoxically, because readable, fuzzy models can become heavy to run, to maintain and to manually tune. The models tend to complexify, requiring too much parameters. This in turn entails a large consumption of computer time and memory. Besides, bad validation performance, above all when noise is pervading the training data, is a typical outcome of overparametrized architecture. The automatic tuning of a readable non-linear mapping leaves few connections with neural networks since nowhere in a bunch of neurons is readibility required. Fuzzy logic traces its origins back to the cognitive and declarative side of AI while neural networks to the procedural and behaviourist side of AI. The second section will discuss in great details how adaptation can be supplied to fuzzy models when first priority is to preserve readability.

The second vision, closer to the type of fuzzy models originally proposed by Takagi and Sugeno [18], is as an interesting way to approximate unknown non-liner mapping by a smooth combination of simple and local models (generally low-order polynomials). It is a new instance of the divide-and-conquer strategy where the problem space is decomposed into fuzzy regions and the data are fitted in every region by simple models. When the local models are linear, this can be seen as a new instance of piecewise linear regression characterized by a softened transition between the pieces. Adopting such a perspective, readability is pushed into the shade or at least relaxed to an easier structural and parametric initialisation of the black-box (as compared with a multilayer perceptron). A main aspect which distinguishes this vision of fuzzy models from other non-linear approximation methods is their potentiality for an easy pre-structuration and for a convenient integration of a priori knowledge under the form of first principle models. A further characteristic is their relying on simple local models while still leaving room for a possible use of data-driven learning technics.

Since this second vision perceives fuzzy models as new black-box way to approximate, control or classify, only performance should be the ultimate purpose. When crude performance is at stake, readability may turn to be something to avoid due to its responsibility in overparametrizing the model, entailing then a high computational cost and a degraded validation behaviour. Readability and performance might thus appear as antagonist objectives with different possible attitude in the search for some form of compromise. This second vision of fuzzy models, as a natural way to smoothly switch between simple operations acting locally in the problem space, although challenged by a large number of fuzzy practitioners, is taking more an more importance these days and is here to stay. In process control, it is a classical strategy (called gain scheduling) that the influence of fuzzy control revives in a growing number of works, like “the composition of heterogeneous control law” [11] and the “local controller network” proposed by [9]

[14] [19]. It is clear that this decomposition into linear actors is a fundamental achievement for control where a great amount of theoretical works tackling robustness, optimality and stability only apply for linear controllers.

What remains neuro-fuzzy in this second vision? Again not a great deal. Viewing fuzzy models as interesting black-box or grey-box to approximate, control or classify, boil them down to an added member (whose originality has to be finely investigated) in the large family of function approximators, neural or not, based on a smooth aggregation of local effects. As a matter of fact closely related approximators are Radial Basis Functions (RBF) [6] [13] (see also [12] for a different perspective on this comparative analysis), GRNN [17], CMAC, B-splines network (see [4] for an interesting overview of this family of models), Locally Weighted Learning or Regression [1] [5], Kernel Regression Estimator [21]. We attend today a strong tendency in the neural net community to privilege the use of local actors and local learning (i.e. learning to behave properly only on the basis of the data present in the restricted region occupied by the local actor) [1] [2] at the expense of global approximators and global learning (the best known being multilayer perceptron). Also the idea that by means of a fuzzy composition you can aggregate information provided by simple actors acting only in part of the problem space is at the basis of Jordan and Jacob's work on "Mixture of Experts" [8]. The third section will discuss this vision of fuzzy model and compare them in turn with closely related models. Finally, the fourth section will present results of a preliminary experimental comparison on a two dimensional function benchmark, a kind of practical attitude that we strongly encourage and which has to become more and more common provided this second vision is adopted.

2 Supplying adaptation to readable fuzzy models

Whatever adopted vision of fuzzy model, two types of tuning are required, designated as structural and parametric tuning. Structural tuning concerns the structure of the rules: which variables to account for in the rules, how to partition each variable universe of discourse so as to derive the linguistic labels, how many rules, and for each rule what type of logical operation to compose it. Once available a satisfactory structure, the parametric tuning must follow with main concern the fine adjustment of the position of all membership functions together with their shape. Metaphorically, the performance of a fuzzy model can be represented as a very rugged two-dimensional mapping: one input dimension coding the structure and the other coding the parameters. There may be a lot of structure/parameter combinations which make the fuzzy model behaving in a satisfactory way. So the search in this metaphorical two dimensional space is not that easy to conduct. As a rule, simple fuzzy models are generally preferred to complex ones so that, in our metaphorical graph, a first reduction of the optimization problem consists in restricting the search to region coding for simple fuzzy models. This is implicitly achieved by turning the cost to optimize into a combination of two objectives: good performance and system of minimal complexity. Finally two types of optimisation mechanisms are required: one for searching in the real space of parameters and another one for searching in the combinatorial space of possible structures.

Now when a first priority is to preserve readability, additional structural and parametric constraints are needed. The definition of these constraints is not an easy task and it is somewhat surprising that despite the importance and originality of the issue, it is hard to find an explicit and exhaustive list of these constraints which when violated makes the model loosing its readability. Perhaps one reason for such a missing list is the awareness that "readability" might itself be a fuzzy concept, taking values on a scale instead of just yes or not. For instance, Takagi and Sugeno's fuzzy models, due to the presence of polynomial consequence instead of linguistic one, can be said to be much less readable than Mamdani's fuzzy models. Often readability first implies a possible decomposition of the models on all the relevant variables in order to generate conjunctive production rules. A n-dimensional Gaussian, whose axis are parallel to the coordinate axis, is straightforward to decompose, while an "oblique" Gaussian is not. Additionally, the fuzzy regions must be homogeneously distributed in the problem space i.e. there must be an unique definition of any linguistic variable which does not change as a function of the other variables (exit the type of space partition provided by decision tree). Too many linguistic labels for each variable is preferentially to avoid. A complete coverage must be guaranteed while too much overlapping between the membership functions is again something to prevent. Linguistic labels must be associated with local and bounded function supports. While it is easy to supervene linguistic rules on fuzzy models with singleton consequence, it becomes harder to read, for instance, fuzzy models with linear consequence.

In order to reduce the complexity of certain fuzzy models, the presence of algebraic operations (for instance turning a multivariable fuzzy model into the addition or multiplication of monovariable fuzzy models) has been recommended [4], which again deteriorates the readability of these models. Hierarchical architectures also contribute to reduce the complexity of the models but again at the expense of readability [15]. Another interesting addition is the presence of internal variables (capturing the memory of recurrent system [7] or akin as macrovariable to decrease the number of influent variables) but again and again at the risk of a great lost of readability and control of the model. As a rule, all welcome modifications achieved in order to reduce the complexity i.e. the number of parameters, then keeping things manageable and decreasing the overfitting risk, often contribute to weaken the readability of the fuzzy model.

Consequently, first to use whatever optimization algorithms you like (GA, gradient descent, off-line or on-line clustering, on-line self-structuring) for automatically tuning the structure and the parameters of the fuzzy model, a clear definition of the "degree" of readability characterizing your fuzzy model must be given. Once provided with this definition, preserving readability requires either an ingenious selection of the degree of freedom of the models subject to the optimisation or the use of an optimization method under constraint which penalizes all solutions which are not readable.

3 Fuzzy models as smooth combinations of simple local models

This second vision of fuzzy models can be integrated in the more general family of methods based on the combination of multiple models. Suppose for simplicity to approximate a multi-input-mono-output mapping based on some trainig data (N will be their number), a multimodel approach can be formalized in the following way:

$$y = \sum_{k=1}^{m} \alpha_k(x) f_k(x) \qquad (1)$$

where m is the number of models, the functions α_k are called the mixing coefficients and f_k are the local models.

The $\alpha_k(\mathbf{x})$ must further respect the following constraint: $\sum_{k=1}^{m} \alpha_k(x) = 1$ (2)

Among other things, this constraint allows to install the multimodelling approach in a probabilistic framework i.e. $f_k(\mathbf{x})$ has probability $\alpha_k(\mathbf{x})$ to be the right model for acting in position $\mathbf{x}$. In the case of fuzzy models, the mixing coefficients are defined by means of the membership functions $G_k(\mathbf{x})$. $G_k(\mathbf{x})$ are functions with bounded support, $G_k(\mathbf{x}_k^c)$ = 1 for $\mathbf{x}_k^c$ the center of the ith support, $0 \le G_k(\mathbf{x}) \le 1$ and $G_k(\mathbf{x})$ decreases when getting away from the center.

$$\alpha_k(x) = \frac{G_k(x)}{\sum_{k=1}^{m} G_k(x)} \tag{3}$$

As discussed in the previous chapter, and relating to the "readability issue", $G_k(\mathbf{x})$ can be yes or not factorizable (or decomposable) on the coordinate axis. In a large amount of applications, $G_k(\mathbf{x})$ are taken to be Gaussians. Besides Takagi and Sugeno's fuzzy model, numerous non-linear approximation methods share this same decomposition strategy: Normalized RBF [6] [12] [13], Specht's General Regression Neural network (GRNN) [17], B-splines network [4], Kernel Regression Estimator [21], Schaal and Atkeson's Receptive Field Weighted Regression (RFWR) [16], Jordan and Jacob's Mixture of Experts Network [8]. When adopting linear function for the local models, the approach can be perceived as a softening version of piecewise linear regression. The learning of these multimodelling approaches must be performed at two levels:1) How to decompose the problem space into a certain number of sub-spaces to be treated restrictively i.e. the number and the nature of the mixing coefficients $\alpha_k(\mathbf{x})$ and 2) What's the nature of all local models i.e. the nature of the $f_k(\mathbf{x})$. In the following, we will briefly attempt a presentation of several multimodelling approaches characterized by different answers given to these two questions.

3.1 The nature and the learning of the mixing coefficients

In the original definition of RBF [6], there are as many $\alpha_k(\mathbf{x})$ as data in the training set (m=N) and a Gaussian center is located on each input data. This turns also to be the case for Kernel Regression Estimator where, for every new point to estimate, each data of the training set exerts an influence on this estimation which is weighted by its distance to this point. In more economical versions of RBF (and similarly for GRNN), mainly for multivariable problems and when the training set is riched in data, the number of mixing coefficients is much smaller than the number of data, although sufficient to cover the problem space. Their position can then be learned by means of a clustering or an unsupervised learning algorithm (only on the basis of the input data) or by means of a supervised clustering (on the basis of the input together with the output data).

The unsupervised learning mechanism locates the mixing coefficient only as a function of the density of points in the input space with no account for the local-model fitting at the output level. For instance, it is very likely to have region of the input space well fitted by a unique parametric mapping although needing more than one cluster to be rightly covered. The reciprocal situation is even more likely, namely to have a very homogeneous region of points the fitting of which is highly non-linear. This explains the better performance often obtained with the supervised clustering procedure which naturally follows from using a gradient-based automatic tuning of the mixing coefficients (above all for tuning the centers, the resulting RBF are called "Generalized RBF" in [6] [20]). Similarly, in the last section, the best experimental results are obtained when initializing the fuzzy architecture by means of such a supervised clustering procedure, namely the fuzzy hyperllipsoidal clustering [2] which acts in the input-output space. The nature of the mixing coefficients will not only depend on the density of the input data but also, and more properly, on the linear models hidden in the training set. Instead of automatically learned, it is often advantageous to manually tune the variance so as to guarantee a complete and homogeneous coverage.

In the Jordan and Jacob's Mixture of Experts (ME) approach, the mixing coefficients α_k are given by a softmax distribution:

$$\alpha_k(x) = \frac{e^{\xi_k(x)}}{\sum_{k=1}^{m} e^{\xi_k(x)}} \tag{4}$$

$\xi_k(x)$ is the ith output of a gating neural network (so it can be whatever function of the input) associated with the kth linear expert. In contrast, the previous models: fuzzy, RBF, GRNN and Kernel-based are more restrictive since the experts act locally in the input space (with $\xi_k(x)$ just function of the euclidian distance) and not just locally in the data set. In the previous models, locality had a spatial meaning. In ME (and in its hierarchical extension HME) a same expert might act in very separated regions of the input space. Now, in function approximation and control, spatial locality and simplicity are two sides of a same coin since any low order approximation takes place in the local neighborhood of a zone of interest. So this further restriction proper to fuzzy models, RBF and GRNN makes indeed a lot of sense.

3.2 The nature and the learning of the local models

These local models can be whatever function of the input kept simple and low dimensional due to their localized field of competence. The most frequent ones are constant or linear. They are constant in Takagi and Sugeno's model with singleton output, in Kosko's additive fuzzy systems [10], Normalized RBF, GRNN and Kernel Based Estimator. They are linear in Takagi and Sugeno's model with linear output, Schaal and Atkeson's RFWR, Jordan and Jacob's ME. In the constant case, a very easy way to learn them is to base their tuning upon the output values of the training set. For instance, in GRNN, all the data belonging to the receptive field of a given Gaussian are averaged to give the constant parameter associated with this Gaussian. Since, once fixed the mixing coefficients, the remaining structure is linear in the parameters to learn (both when the local models are constant and linear), another easy way to learn the local models is to rely on a global one shot pseudoinverse method (like Singular Value Decomposition) which garantees to find the global maximum.

When the local models are taken to be linear, two ways of learning the local models become possible: one local - learning every local model separately on the basis of the most influent data, one global - learning all local models simultaneously accounting thus for the approximation benefits of their overlapping (or cooperation). In the case of the local learning, the least-square cost to minimize for each kth model is as follows:

$$E = \sum_{i=1}^{N} \left[(y_i - f_k(\boldsymbol{x}_i))^2 \alpha(\boldsymbol{x}_i, \boldsymbol{x}_k) \right] \tag{5}$$

with N the number of points in the training set, $\mathbf{x}_k$ the center of the kth Gaussian and f_k the linear model associated to this Gaussian. In the case of the global learning, the global least-square cost to minimize is as follows:

$$E = \sum_{i=1}^{N} \left[(y_i - \sum_{k=1}^{m} \alpha(x_i, x_k) f_k(x_i))^2 \right] \tag{6}$$

Since even in the global learning case, the local nature of the mixing coefficients will finally localize the learning, which procedure to adopt is not an easy choice. Using global learning might appear to be more realistic since the least-square error to minimize really includes the exact global model while local learning can benefit from the use of local performance criterion such as a local study of the cross validation estimate which is easy and fast to implement in the linear case. Once again, learning the linear f_k functions can be done by the simple use of the SVD method.

3.3 Comparison with the lazy methods

Locally weighted Learning, a very classical approach in the statistical and economical literature [5] (see [1] for an interesting overview), but recently imported in the connectionist and machine learning communities, works in the following simple way (said "lazy" in [1]). The training set extracted from an unknown mapping y(**x**) is at your disposal. As soon as you receive a query i.e. an input $\mathbf{x}_k$ for which you want to deduce a good approximation of the output, you fit a local model $f_k(\mathbf{x})$ through the closest points of your query. The approximated answer to your query is obtained by computing the output of this local model for the input: $y_k=f_k(\mathbf{x}_k)$. Each point $(\mathbf{x}_i,y_i)$ (i=1..N) in the data set participates in determining the local model but weighted by a distance measure $\alpha_k(\mathbf{x}_i,\mathbf{x}_k)$. In the simplest Kernel based estimation case, each local model just consists in the value of the output for each input of your data set.

Similarly to previous models, the local model can be whatever mathematical function $f_k(\mathbf{x})$, in general kept simple (linear is the most frequent case), since, like for all approaches discussed up to now, the decomposition makes sense only if it allows the use of simplified local actors. With local model given by $f_k(\mathbf{x})$, the first step is to determine the specific local model to be exploited for answering the query. It is given by minimizing the following error measure:

$$E=\sum_{i=1}^{N}\left[(y_i-f_k(x_i))^2\alpha(x_i,x_k)\right] \qquad (7)$$

This expression is perfectly similar to (5) with α centered in each query point instead of resulting from a clustering algorithm. Thus the great similarity between weighted local learning and fuzzy models resides in their share emphasis on decomposing a problem into a set of local simpler sub-problems. However, a capital difference has to be pointed out. Fuzzy models propose to approximate the function by means of an explicit model (all $f_k(\mathbf{x})$ are explicitly defined) while locally weighted learning techniques enter in the family of interpolation method (one $f_k(\mathbf{x})$ is temporarily obtained for a specific query). No explicit model is derived but just an algorithmic way of answering a query. The method is said lazy because any processing of data is deferred until a new input query is presented. Furthermore all the training set is needed from the beginning. For certain applications such as adaptive process control, for which fuzzy models are largely used, an initial explicit approximation of the controller must exist even in the absence of an initial training set.

4. A Preliminary Experimental Comparison Between Seven MultiModelling Approaches

Seven multimodelling approaches, among which a “lazy” one, have been experimentally compared on a two-dimensional non linear benchmark function

$$[-1,1]\times[-1,1] \rightarrow R : f(x1,x2)= 4\sin(\pi x_1)+2\cos(\pi x_2)+N(0,0.5)$$

with N(0,0.5) a Normal noise with zero-mean and standard deviation of 0.5.

A learning set of 200 pairs (x_{1i}, x_{2i}) i=1...200, sampled in an uniform random manner in the domain [-1,1]×[-1,1], was used for training the first four models. We tested the final performance of the models on an independent set of 1024 points regularly sampled in the input domain. The seven approaches are:

1. a fuzzy system with triangular membership functions, linear output, a normalised composition of the outputs and an unsupervised K-mean initialization of the centers and bases
2. a fuzzy system with gaussian membership functions, linear output , a non-normalised composition of the outputs and an unsupervised k-mean initialization of the centers and variances
3. a fuzzy system with gaussian membership functions, linear output , a non-normalised composition of the outputs and a supervised Hyperllipsoidal Fuzzy Clustering [2] initialization of centers, variances and parameters of the output models
4. a RBF with gaussian membership functions, constant output, a non-normalised composition of the outputs and an unsupervised K-mean initialization of the centers and variances.
5. a lazy learning method based on a locally weighted regression , with a tricubic kernel and a mixed polinomial approach
6. a mixture of linear experts (ME) whose tree has just a single level but developed in width.
7. a hierarchical mixture of linear experts (HME) whose tree is binary but developed in depth (i.e. can have several levels).

For all non lazy approaches, the number of local models was incremented until to obtain the best performance on a 10th fold cross-validation procedure. It is important that any procedure which aims at the optimal number of local models not only depends on the learning set but also on the cross-validation successive step. In table 1, this 10th fold cross-validation results are indicated in the column called: “Estimated RMSE (Root Mean Squared Error)”. For the first four apporaches, and for whatever number of local models, a structural initialisation was performed for locating the local models: unsupervised in the cases 1, 2 and 4, and supervised in the case 3. Then we applied an alternate running of the gradient-based Levenberg Marquadt optimisation method for a fine tuning of the parameters coding the α_k, followed by the SVD routine for the linear parameters coding the f_k. For the approaches 6 and 7, the parametric learning was done by the EM algorithm described in [22].

For each tested structure, a statistic analysis was performed over 10 different random initializations of the system. For the fifth and lazy approach, the only indicated result is the performance obtained on the 1024 points indicated in the column: “Measured RMSE”. In the first colum of table 1 is indicated the best complexity found (i.e. which ends the incremental procedure). For approaches 1-2-3-4-6, this complexity amounts to the number of local models. For the 7th approach, the complexity is a function of the number of hierarchical levels: $2^{(\text{number of levels + 1})}$. The best result on the 1024 points testing set was achieved by the third approach i.e. the fuzzy model with gaussian membership functions and linear outputs. However, we strongly suspect the structural initialisation, supervised in this case, to be a key factor for explaining this good performance.

Table 1: Results of the Experimental Comparison

Model	**Best complexity found**	**Estimated RMSE**	**Measured RMSE (mean ± std)**
Model 1 (fuzzy)	9	1.0863	1.404 ± 0.335
Model 2 (fuzzy)	21	0.608140	0.635 ± 0.025
Model 3 (fuzzy)	13	0.580665	0.552 ± 0.014
Model 4 (RBF)	22	0.760997	0.821 ± 0.112
Model 5 (lazy)	-------------	---------------	0.735
Model 6 (ME)	20	0.800	1.005 ± 0.015
Model 7 (HME)	16	0.699	0.6776 ± 0.009

5 Conclusions

The most important message delivered in this paper is the today existence of two visions of fuzzy models, first as a linguistic translation of a numerical mapping and then as a powerful multi-local-model black-box approximator. These two visions can be incompatible to some extent i.e. the search for a readable model can be done at the expense of the computational cost, and above all at the expense of the performance in approximating real data. The growing interest in neuro-fuzzy models assumes an important responsibility not only for deepening the gap between these two visions but also for "fuzzyfying" in which context and for what reason fuzzy models are used. This is the main reason for the title of the paper. Today, the time has come to help readers of the neuro-fuzzy literature (growing so fast today) to better understand what is the context and the motivations of the work described in the neuro-fuzzy papers. This often can easily be done by a friendly divorce between the just married methods.

If readability is the main objective, first a precise understanding of what is meant by "readability" is necessary, then a whole new set of optimization methods, very dissimilar from their implementation in neural models, must still be derived to complement the human expert knowledge in the automatic generation of fuzzy models. Also more discussion would be welcome on the necessary compromise to reach between a satisfactory approximation and a readable model. If optimal performance in control, function identification or classification is the main objective, then an exhaustive and deep comparison with closely related methods (local-based and multi-models) has to be done, for realistic problems, in terms of pure performance, computational cost and practicability.

References

[1] Atkeson, C.G., Moore, A.W. and S. Schaal. 1995. Locally Weighted Learning. *Carnegie Mellon University Technical Report submitted to Artificial Intelligence Review*.

[2] Bezdek, J.C. and J.M. Anderson. 1985. An application of the c-varieties clustering algorithms to polygonal curve fitting. In IEEE Trans. SMC, vol. 15, No 5 - pp. 637-641.

[3] Bottou, L. and V. Vapnik. 1992. Local Learning Algorithms - *Neural Computation* 4 - pp. 888-900.

[4] Brown, M. and C.J. Harris 1994. *Neurofuzzy adaptive modelling and control*. Prentice Hall, Hemel Hempstead.

[5] Cleveland, W.S., Devlin, S.J. and S.J Grosse, 1988. Regression by Local Fitting: Methods, Prospectives and Computational Algorithms. *Journal of Econometrics* 37 - pp. 87-114. North Holland.

[6] Girosi, F., Jones, M. and T. Poggio 1995. Regularization Theory and Neural Networks Architectures - *Neural Computation* - Vol. 7 - No 2 - pp. 219-269.

[7] Gorrini V. & Bersini H., 1994, "Recurrent Fuzzy Systems". In *Proceedings of the Third IEEE International Conference on Fuzzy Systems*, pp. 193-198.

[8] Jacobs, R.A., Jordan, M.I., Nowlan, S.J. and G.E. Hinton 1991. Adaptive Mixture of Local Experts - *Neural Computation*, Vol.3, pp.79-87.

[9] Johansen, T.A. and Bjarne A. Foss 1995. Semi-Empirical Modeling of Non-Linear Dynamic Systems through Identification Operating Regimes and Local Models - *Neural Network Engineering in Dynamic Control Systems* - K.J. Hunt, G.R.Irwin and K. Warwick (Eds.) - Springer. pp. 106-126.

[10] Kosko, B., 1996. Additive fuzzy systems: from function approximation to learning. Ch. 9 in *Fuzzy Logic and Neural Network Handbook*, ed. by C.H. Chen, Mc Graw-Hill, Inc., NY.

[11] Kuipers, B. and K. Astrom, 1994. The Composition of Heterogeneous Control Laws - in *Fuzzy Sets, Neural Networks and Soft Computing* - R. Yager and L.A. Zadeh (Eds.) - Von Nostrand Reinhold - New York. pp. 45-62.

[12] Mendel, J. 1995. Fuzzy logic systems for engineering: a tutorial. *IEEE Proc*. vol. 83, pp. 345-377.

[13] Moody J. & Darken, C.J. 1989. Fast learning in networks of locally tuned processing units. *Neural Computation*, **1**, 281-294.

[14] Murray-Smith, R. and K. Hunt. 1995. Local Model Architectures for Nonlinear Modelling and Control - in *Neural Network Engineering in Dynamic Control Systems* - K.J. Hunt, G.R.Irwin and K. Warwick (Eds.) - Springer. pp.61-82

[15] Raju, G.V.V, Zhou, J. and R.A. Kisner - 1991. Hierarchical Fuzzy Control - *International Journal of Control* - 54(5) - pp. 1201-1216.

[16] Schaal, S. and C.G. Atkeson - 1996. From Isolation to Cooperation: An Alternative View of a System of Experts - In Proceedings of NIPS 8.

[17] Specht, D.E. 1991. A General Regression Neural Network. IEEE Transactions on Neural Networks, 2(6) - pp.568-576.

[18] Tagaki, T. and M. Sugeno 1985: Fuzzy identification of systems and its application to modelling and control. *IEEE Trans. Syst. Man Cybern.*, vol SMC-15, no. 1, pp 116-132.

[19] Tanaka, K. and M. Sano. 1994. A Robust Stabilisation Problem of Fuzzy Control System and Its Application to BAcking up Control of a Truck-Trailer - *IEEE Trans. on Fuzzy Systems* - pp. 119-134.

[20] Wettschereck D. & Dietterich T. (1992). Improving the performance on the radial basis function networks by learning center locations. In J.E. Moody, S.J. Hanson & R.P. Lippman (Eds.), *Advances in Neural Information Processing Systems 4.* San Mateo, CA: Morgan Kaufmann, pp 1133-1140.

[21] Xu, L., Krzyzak, A. and A. Yuille. 1994. On Radial Basis Function Nets and Kernel Regression: Statistical Consistency, Convergence Rates, and Receptive Fields - *Neural Networks* - Vol. 7 - No 4 - pp. 609-628.

[22] Xu, L., Jordan, M.I. and Hinton G. E. 1995. An Alternative Model for Mixtures of Experts. *Advances in Neural Information Processing Systems 7*, edited by G. Tesauro, D.S. Touretzky and T.K. Leen, pp. 633-640