2009 Ninth International Conference on Intelligent Systems Design and Applications

Tutorial: About Industrial Acceptance of Intelligent Systems

Leonardo M. Reyneri
Politecnico di Torino, Torino, Italy. Email: leonardo.reyneri@polito.it

***Abstract*—This paper analyzes the stage of maturity that neurofuzzy systems (and soft computing in general) have recently reached and tackles the several reasons why they have not yet reached a widespread acceptance in industrial and agronomic applications, despite the good performance they can offer with a reduced design effort.**

I. Introduction

Neural networks and *fuzzy systems* are well known *soft computing* techniques, which already date back several decades since the preliminary works of Mc Culloch and Pitts, Grossberg, Zadeh, and tens of other precursors.

During the first decades, neural networks were believed to become "the simple and viable solution" to all tough problems one might be faced with, therefore they gave rise to the interest of researches all over the world and gathered a lot of funding. During this preliminary period, a lot of theories have been developed, analyzed and applied.

Later on, people discovered that most simple problems (the so called "toy problems") actually found simple solutions using neural networks (and, more recently, fuzzy logic). Unfortunately, the really tough problems (for instance, handwriting recognition) still could not be (completely) solved, although neural networks and fuzzy logic helped to simplify their (partial) solution.

After several decades of alternating interest of the scientific and industrial community, after publishing tens of thousands of theoretical and practical studies and after several attempts to apply them in a large number of application domains, neural networks and fuzzy systems are nowadays reaching a rather *mature stage*.

In fact, people now start understanding the real capabilities, potentials, limitations and drawbacks of neurofuzzy techniques in general, therefore they begin adopting soft computing methods appropriately, without excessive enthusiasm but also, more important, with a good rationale for their use.

This paper attempts to analyze the level of maturity and acceptance that neurofuzzy techniques have now reached and tries to assess how easily they are (or can be) accepted in the industrial domain. Note that I will mainly quote *industrial applications*, although I generally refer to both industrial, agronomic, economic, mathematic, forecasting, etc. applications.

II. Apparent Diversity of Neurofuzzy Paradigms

At the beginning, neural networks, fuzzy systems and other soft computing techniques like *wavelet networks*, *Bayesian classifiers*, *clustering methods*, etc., were believed to be independent, although complementary, methods, which had to be analyzed and studied independently of each other. This caused an excessive effort to study, analyze, get familiar (and train personnel) with a huge variety of methods, each one apparently having its characteristics and preferred application domains. For instance, fuzzy logic was usually selected in control applications, were expert practitioners were able to express their knowledge in linguistic form; neural networks were often chosen in complex, model-free classification tasks, like for instance, handwriting interpretation; wavelet networks were often chosen in signal processing applications.

Instead it has recently been proved [1] that all those soft computing techniques are nothing but different *languages* for the very same paradigm, therefore they need not any more to be studied apart. As a consequence, we can now talk of *neurofuzzy systems* as a whole, and consider the neural, fuzzy, wavelet, Bayesian, etc. nothing else than *neurofuzzy languages*, each one being more appropriate to any given application.

As a consequence, the number of independent paradigms significantly reduces to as few as four [1]. That is, all known topologies for neural, fuzzy, wavelet, Bayesian, clustering paradigms, etc. and supervised or unsupervised training algorithms are, in practice, just particular implementations and interconnections of four elementary blocks (namely, *computing elements* and *layers*, *normalization layers* and *sensitivity layers*).

The consequences of *neurofuzzy unification* are quite important for a wider acceptance of neurofuzzy techniques, mainly in industry, were training costs and

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.133

design effort are one of the most relevant matters in the selection phase of a new technique. In fact, in the early times of neural networks and fuzzy systems, practitioners were asked to study and become familiar with so many (apparently) different paradigms, each one with a set of different design criteria, training algorithms etc.. This training phase was rather time consuming, therefore costly for industry, and it often happened that it could not be afforded, especially without being sure to get appropriate returns. It must be remembered that adopting any novel method *may* offer advantages, but it *surely* costs money.

Instead an appropriate use of unification allows, on the one hand, the users to quickly learn and get familiar with the very few basic paradigms (therefore reducing training costs to a very minimum, such that anybody can easily afford costs and risks) and, on the other hand, augments flexibility and performance of neurofuzzy techniques (therefore increasing the economical return and reducing the corresponding risks).

III. Maturity of Neurofuzzy Techniques

After decades of developments, researches, studies, application attempts, published papers and books, nowadays neurofuzzy techniques have reached a *maturity stage* never reached before.

Most of the original theories have recently been nearly abandoned (like, for instance, *Hopfield networks* and *Boltzmann machines*, *glass spin theories*, *stochastic networks*, etc.) either because they could not offer reasonable performance or because they were too cumbersome to use, while others (like, for instance, *perceptrons*, *radial basis functions* and *fuzzy systems*) were clearly more viable, therefore they eventually reached widespread acceptance.

In particular, the maturity of techniques like soft computing can be assessed from a number of clues like, among others:

- the number of theories and paradigms which have survived, which should be as low as possible, to simplify as much as possible the knowledge one must have in order to apply them (see sect. II);
- the number of theories and paradigms which have been created, which should be as high as possible, to be sure that no possibility has been forgotten;
- the level of acquaintance a designer has with these techniques
- the number of *accepted* industrial applications

As a consequence of the maturity level they reached, nowadays neurofuzzy systems deserve to be in the *knowledge briefcase* of each engineer, economist, agronomist, scientist, etc. *together with, and a the same acceptance level of* several other basic techniques like algebra, statistics, geometry, etc. Neurofuzzy techniques cannot any more be considered as a discipline for a few experts.

As a consequence of this new approach (for me, the only viable to reach a widespread industrial acceptance), nobody shall any more use statements like:
I have used/developed a neural network for...
but, instead, for instance:
I have just developed a complex system with interacting signal preprocessor, neural network, user interface, a differential equation solver, a post processor, some sensor and actuator interface, etc.
The difference between the two approaches is that, while in the former statement the stress in on the presence of a neural network which therefore improperly becomes the most relevant block within the systems, in the latter statement the neural network takes its proper place, that is, at the same relevance level as all the other system blocks.

In several applications, the surrounding blocks are even more complex to design and to properly use than a neural networks, which does not deserve any more the highest consideration it had (inappropriately) until a few year ago. An example for this is in the field of image processing and handwriting recognition, where a successful application relies much more on a proper image preprocessing (filtering, contrast enhancement, segmentation, labeling, skeletonization, etc.) than on the neurofuzzy processing itself.

Unfortunately, despite the level of maturity they reached, neurofuzzy systems still experience a lot of difficulties to be *accepted* by the industrial community, which still sees them as academic experiments or bizarre techniques and not as a powerful tool to solve their problems.

IV. Relevant Characteristics For Industry

I try to analyze here some of the reasons why neural networks and fuzzy systems still experience difficulties in being accepted as an industrial standard.

A. Crypticity

Soft computing techniques and, in particular, neural networks and wavelet networks are still often felt as being rather *cryptic*, as nobody can really understand why and how a trained network can solve a given problem.

All the knowledge of a trained neural (or wavelet) network is hidden within a chunk of numbers usually arranged into one or more matrices (e.g. either *weights* for perceptrons and wavelons or *centers* for radial basis or Kohonen networks). There is usually no clue on how to interpret such numbers, therefore engineers are still skeptical in regards to the correctness of those numbers.

In reality correctness of weights is based on a successful training, although it is often difficult to guarantee that training has properly succeeded. Training is measured on the amount of a residual *error measure*, but there is often no clue for an appropriate value for this error, especially when *sum-of-errors* measures are used, as in several commercial simulation tools. The neurofuzzy designer cannot therefore reliably argue that a trained model is really representative of the desired system/function.

On the other hand, traditional (namely, non-neurofuzzy) design methods are based on some analytical or empirical model which is *chosen by the designer*, together with its parameters. Designer's *knowledge and experience* provide enough information to properly solve a problem, even though seldom in the optimal way. Usually nothing is left to randomness. The only items which are not chosen directly by the designer are the parameters of parametric models, when they are empirically adapted to match a given system.

In reality, the process of empirical adaptation of a given parametric model to a given system is rather similar to the soft computing approach of training a neural or wavelet network (which is nothing but a highly generic parametric model) based on a set of training data. Yet the former is considered as normal and straightforward by nearly any designer, while the latter still makes several designers skeptical. Why is that so?

One of the reasons is that non-neural parametric models currently used in practice are much less generic than soft computing models, therefore they are always under control of the engineer, which usually can properly interpret parameter values. For instance, the model of an electric motor can model nothing else than an electric motor, and its parameters represent, for instance, winding resistance and inductance, rotor inertia, friction, etc. which are directly measurable and for which the designer can feel if they assume reasonable values or not.

By appropriate varying these parameters, the model will be adapted to either large or small motors, either fast or slow, but it will never be able to model, for instance, a chemical process. The designer can therefore become aware that, for instance, an improperly tuned model has a too large or too narrow winding resistance in comparison with the size of the motor under examination. He can therefore immediately become aware of the improper tuning and correct it accordingly.

On the other hand, neural networks are so generic that they can adapt to virtually any system, either electrical or chemical or economical or mechanical or agronomic, etc. The same parameters can therefore mean anything, depending on the actual use of the network; in addition parameters are interchangeable and there is no clue to understand what a parameter represents in practice. As a consequence, nobody will ever be aware that training has not been done correctly and whether the model really represents that system.

B. How to Avoid Crypticity

The use of modern unification paradigms [1] allows to easily convert neural and wavelet networks into fuzzy systems and viceversa, with several advantages, among which, for instance:

- a given neuro/wavelet network can be converted into fuzzy language, thus interpreted linguistically by experts, who are then able to "validate" and consequently "accept" an otherwise cryptic neuro/wavelet model;
- human experience, usually expressed as a set of fuzzy rules, can be converted into a neural network and then empirically tuned by means of an appropriate training set. This technique is much more similar to the traditional approach described above, where a user empirically optimizes a given analytical model. The only difference is that the user-defined model is expressed in terms of fuzzy rules instead of analytical methods and this is only slightly tuned to optimize performance. The advantages are that: i) the size of training set is much smaller; ii) the model cannot vary too much, therefore it cannot differ too much from what has been defined by the designer, who therefore keeps total control of the model.

It is therefore mandatory to abandon all the older approaches who were more like "magic formulas" than real engineering methods and concentrate on modern approaches who consider neural, wavelet, fuzzy, Bayesian, regressor, clustering techniques as a whole, that is, as a set of *interchangeable paradigms*.

The ever lasting fight among neural- and fuzzy-people is so detrimental, as it helps to keep the level of crypticity high, which prevents a widespread acceptance of neurofuzzy methods.

The choice between *neural networks* and *fuzzy logic* should therefore be converted into a more appropriate selection between a *neural* and a *fuzzy language*, which should be chosen depending on: i) the available knowledge from human experts; ii) the size of available training set; iii) the level of crypticity which is accepted; iv) if and how the model has to be interpreted by humans or only processed by computers.

C. Gathering Data for Network Training

Neural techniques rely on the availability of a large enough training set, which is often too expensive to obtain. Each training point is an appropriate *measurement* of a mechanical or chemical or a biological or an economical process. Several processes are so slow that one single point may require up to several days to be measured. In some particular cases, computer simulations can substitute direct measurements, although this requires an accurate numerical model.

Other soft computing techniques (in particular those based on fuzzy and Bayesian languages) may require much smaller training sets, as they rely on a predefined model described in linguistic terms according to the previous human experience. This is the main reason why fuzzy logic has so far been accepted more quickly and extensively by industry than neural networks.

An industrial manager has to consider attentively the trade-off between the cost of gathering a large training set and the reliability of the trained neurofuzzy network. As already said, this trade-off pushes toward the use of fuzzy logic (namely, fuzzy language) whenever possible and bounds the use of neural networks (namely, neural language) in a limited set of industrial applications.

D. Performance are Always Optimistic...

Virtually any paper published in literature shows that, for a wide range of applications, neural networks and fuzzy systems offer tremendously good performance.

Unfortunately, more than 50% of them does not even try to afford a performance comparison with other techniques, therefore it becomes rather difficult to feel how good such performance really are. Just as an example, I once found *a neural model of a biochemical process which is 90% accurate* and the author was *enthousiast* of that *incredible result*. I had no experience on modeling such processes, therefore I could not do anything else than blindly accept author's statement. But, when I met an experienced colleague, he told me that state of the art had already achieved about 95% since a few years, making the neurofuzzy result useless for industry.

I am quite sure the the author was really convinced of the optimality of his result, due perhaps to the same limited experience I had, which was not enough to judge. What the author surely did was to try a number of different topologies, paradigms, network sizes, training algorithms and found that his own network was offering the very best performance among all those tests.

This method is (partially) correct to evaluate the performance of a novel neurofuzzy paradigm, but **not** to evaluate the appropriateness of a neurofuzzy model compared with a non-neurofuzzy one.

What was true for that specific problem, is that a hybrid empirical/analytical model developed by a team of experienced engineers and biologists offered a much better performance than the best neural network, and even for a comparable computation complexity.

The reason for that (which happens much more frequently than one can even imagine) is that human experience, knowledge and mental capacities, which are used to develop a given model, boost so much the overall performance of a given model than even an optimal neurofuzzy system trained in the best known way cannot compensate the absence of knowledge present in the neural network.

E. How to Avoid Optimism

The advantage of neurofuzzy networks is that a given analytical/empirical model is by definition specific and cannot be tailored to a different problem, while neural networks are. Furthermore an analytical/empirical model usually comes after years of improvements, while neural networks are trained in a short time.

This is theory, while in practice this is also not completely true, as a purely analytical model can be developed without any field measurement, while an analytical/empirical models also requires a limited amount of field measurement. Instead neurofuzzy networks always require an often huge amount of field measurements which, in several cases, can take years to gather.

Last but not least, the amount of field measurements which is required (that is roughly the development time) is a function of the *reliability* which is asked to the model (see also sect. IV-C). A large training set is in fact mandatory in industry to offer an adequately high reliability, while reliability of analytical models is often independent of field measurements but relies on designer's experience.

Yet, in most industrial applications that I have encountered so far, very few neural networks offered such better performance with respect to other techniques to really convince the skeptical user. I obviously compare the best neural network or fuzzy system with the correspondingly *best* non-neurofuzzy technique.

V. How to Help Industry Accepting Neurofuzzy Techniques

I personally believe that industry strongly needs to be helped to accept neurofuzzy techniques, as this is can be a major role for universities and research institutions. But these have to do it in the more appropriate way, that is, to show them unambiguously if, where and when neurofuzzy techniques offer significant advantages or, more realistically, more advantages than drawbacks.

This is one of the major reasons for the Special Session on Industrial and Agronomic applications at ESANN 2003. Authors were requested to present their neurofuzzy ideas but, more important, to prove that they were either comparable or significantly better than other standard techniques.

But such a comparison has to be as fair as possible, as it is not normally the case. In practice, in most papers, neurofuzzy techniques are usually compared among themselves, but the expert reader is left with the question: *are you sure that other techniques would not be even better or simpler?*.

Or, when a comparison is attempted with standard techniques, these are usually much older, that is, the paper demonstrates that an *up-to-date neurofuzzy network* is much better than an *older-than-my-father standard technique*, which is rather obvious, as technology keeps improving, independent of neural networks.

One of the major reasons for this lack of fair comparisons is that comparing with an *up-to-date standard method* requires developing by scratch an appropriate demonstrator, which often requires either a lot of specific experience or a lot of time, and nobody wants to afford it.

Only those research groups who tightly cooperate with an industrial group can merge industrial and academic experiences to develop both techniques appropriately, although these are seldom done together, due to unaffordable additional costs.

VI. Conclusion

According to my personal experience, I *cannot* state that neurofuzzy techniques are so advantageous with respect to traditional techniques to be universally accepted for industrial applications. Or better, advantages exist but are often too limited when compared with the additional risks, training costs, design time, and documentation/maintenance effort. There are surely applications were they provide advantages, especially in tough problems, but these are rather limited therefore they do not justify a universal acceptance.

Yet there is one advantage (perhaps only one?) which makes neurofuzzy techniques attractive in a wider range of applications. That is, they are
generic approximation and modeling techniques which allow accurate system modeling/forecasting/-approximation/classification/etc. **without** *any specific experience of the designer*.

In practice anybody without any experience in a specific subject can afford solving a problem which could otherwise (namely, with traditional techniques) be afforded only by an expert (or a team of experts) in that field. Perhaps the expert, with appropriate knowledge of the problem and of a bunch of more specific methods would achieve a better result, but this would be far more expensive for an industry, both because of the higher cost of the expert and for the longer development time.

This is a rather interesting advantage, even when neurofuzzy techniques are suboptimal, as it significantly reduces training costs of unexperienced personnel.

A. A Critical Question

I have so far encountered very few applications were neural networks provided such better performance with respect to other techniques to really convince even the most skeptical user. In most cases they can either offer a slightly better performance (when compared with an alternative well-designed method) with a shorter design time but, on the other hand, design risks are often so critical that they definitely impair the advantages.

It is therefore time for a critical question: *In which applications are neural networks have fuzzy logic a higher chance of being accepted?*

I think that, at present, the most promising areas are, for instance:

- *data mining*, *knowledge based systems*, where information, data, knowledge and models are *valuable items*, but they are often *hidden* in a huge amount of noise, ambiguous, contradicting data. Data is so wide, contradicting, ambiguous, that no method can be accurate and predictable, therefore neural networks may provide advantages, without needing to be 100
- *prediction/classification of partially random processes*, like *time-series prediction*, *forecasting*, *complex pattern classification*, *semantic Web* etc., where the randomness of the process/patterns prevents a 100the errors of the neurofuzzy systems can be accepted at no cost;
- *modeling of complex systems*, where any other modeling technique would be as incomprehensible as a neurofuzzy model;

- *consumer applications* where the appeal of the ”fuzzy label” increases the market of an appliance.

Whereas, on the other hand, several applications should be considered nothing more than *toy problems*, which may be interesting from the academic point of view, just to prove the validity of a new method, but which will never have any chance of being accepted industrially.

REFERENCES

[1] L.M. Reyneri, “Unification of Neural and Wavelet Networks and Fuzzy Systems”, in *IEEE Trans. on Neural Networks*, Vol. 10, no. 4, July 1999, pp. 801-814.

[2] S. Haykin, “Neural Networks: A Comprehensive Foundation”, *Mc Millan College Publishing Company*, New York, 1994.

[3] Q. Zhang: “Using Wavelet Network in Non-parametric Estimation,” *IEEE Transactions on Neural Networks*, Vol. 8, No. 2, pp. 227-236, March 1997.

[4] V. Colla, M. Sgarbi, L.M. Reyneri: “A Comparison Between Weighted Radial Basis Functions and Wavelet Networks,” in *Proc. of ESANN 98*, Bruges, Belgium, 22-24 April 1998, pp. 13-19.

[5] V. Colla, L.M. Reyneri, M. Sgarbi, “Neuro-Wavelet Characterization of Jominy Profiles of Steels”, in *Journal of Integrated Computer-Aided Engineering*, 1999, John Wiley & Sons, New York (NY).