A

Adaptive Neuro-Fuzzy Systems

Larbi Esmahi
Athabasca University, Canada

Kristian Williamson
Statistics Canada, Canada

Elarbi Badidi
United Arab Emirates University, UAE

INTRODUCTION

Fuzzy logic became the core of a different approach to computing. Whereas traditional approaches to computing were precise, or hard edged, fuzzy logic allowed for the possibility of a less precise or softer approach (Klir et al., 1995, pp. 212-242). An approach where precision is not paramount is not only closer to the way humans thought, but may be in fact easier to create as well (Jin, 2000). Thus was born the field of soft computing (Zadeh, 1994). Other techniques were added to this field, such as Artificial Neural Networks (ANN), and genetic algorithms, both modeled on biological systems. Soon it was realized that these tools could be combined, and by mixing them together, they could cover their respective weaknesses while at the same time generate something that is greater than its parts, or in short, creating synergy.

Adaptive Neuro-fuzzy is perhaps the most prominent of these admixtures of soft computing technologies (Mitra et al., 2000). The technique was first created when artificial neural networks were modified to work with fuzzy logic, hence the Neuro-fuzzy name (Jang et al., 1997, pp. 1-7). This combination provides fuzzy systems with adaptability and the ability to learn. It was later shown that adaptive fuzzy systems could be created with other soft computing techniques, such as genetic algorithms (Yen et al., 1998, pp. 469-490), Rough sets (Pal et al., 2003; Jensen et al., 2004, Ang et al., 2005) and Bayesian networks (Muller et al., 1995), but the Neuro-fuzzy name was widely used, so it stayed. In this chapter we are using the most widely used terminology in the field.

Neuro-fuzzy is a blanket description of a wide variety of tools and techniques used to combine any aspect of fuzzy logic with any aspect of artificial neural networks. For the most part, these combinations are just extensions of one technology or the other. For example, neural networks usually take binary inputs, but use weights that vary in value from 0 to 1. Adding fuzzy sets to ANN to convert a range of input values into values that can be used as weights is considered a Neuro-fuzzy solution. This chapter will pay particular interest to the sub-field where the fuzzy logic rules are modified by the adaptive aspect of the system.

The next part of this chapter will be organized as follows: in section 1 we examine models and techniques used to combine fuzzy logic and neural networks together to create Neuro-fuzzy systems. Section 2 provides an overview of the main steps involved in the development of adaptive Neuro-fuzzy systems. Section 3 concludes this chapter with some recommendations and future developments.

NEURO-FUZZY TECHNOLOGY

Neuro-fuzzy Technology is a broad term used to describe a field of techniques and methods used to combine fuzzy logic and neural networks together (Jin, 2003, pp. 111-140). Fuzzy logic and neural networks each have their own sets of strengths and weaknesses, and most attempts to combine these two technologies have the goal of using each techniques strengths to cover the others weaknesses.

Neural networks are capable of self-learning, classification and associating inputs with outputs. Neural networks can also become a universal function approximator (Kosko, 1997, pp. 299; Nauck et al., 1998, Nauck et al. 1999). Given enough information about an unknown continuous function, such as its inputs and outputs, the neural network can be trained to approximate it. The disadvantages of neural networks are they are not guaranteed to converge, that is to be trained properly, and after they have been trained they cannot give any information about why they take a particular course of action when given a particular input.

Fuzzy logic Inference systems can give human readable and understandable information about why a particular course of action was taken because it is governed by a series of IF THEN rules. Fuzzy logic systems can adapt in a way that their rules and the parameters of the fuzzy sets associated with those rules can be changed to meet some criteria. However fuzzy logic systems lack the capability for self-learning, and must be modified by an external entity. Another salient feature of fuzzy logic systems is that they are, like artificial neural networks, capable of acting as universal approximators.

The common feature of being able to act as a universal approximator is the basis of most attempts to merge these two technologies. Not only it can be used to approximate a function but it can also be used by both neural networks, and fuzzy logic systems to approximate each other as well. (Pal et al., 1999, pp. 66)

Universal approximation is the ability of a system to replicate a function to some degree. Both neural networks and fuzzy logic systems do this by using a non-mathematical model of the system (Jang et al., 1997, pp. 238; Pal et al., 1999, pp. 19). The term approximate is used as the model does not have to match the simulated function exactly, although it is sometime possible to do so if enough information about the function is available. In most cases it is not necessary or even desirable to perfectly simulate a function as this takes time and resources that may not be available and close is often good enough.

Categories of Neuro-Fuzzy Systems

Efforts to combine fuzzy logic and neural networks have been underway for several years and many methods have been attempted and implemented. These methods are of two major categories:

- Fuzzy Neural Networks (FNN): are neural networks that can use fuzzy data, such as fuzzy rules, sets and values (Jin, 2003, pp.205-220).
- Neural-Fuzzy Systems (NFS): are fuzzy systems “augmented” by neural networks (Jin, 2003, pp.111-140).

There also four main architectures used for implementing neuro-fuzzy systems:

- Fuzzy Multi-layer networks (Jang, 1993; Mitra et al., 1995; Mitra et al., 2000; Mamdani et al., 1999; Sugeno et al., 1988, Takagi et al., 1985).
- Fuzzy Self-Organizing Map networks (Drobics et al., 2000; Kosko, 1997, pp. 98; Haykin, 1999, pp. 443)
- Black-Box Fuzzy ANN (Bellazzi et al., 1999; Qiu, 2000; Monti, 1996)
- Hybrid Architectures (Zatwarnicki, 2005; Borzemski et al., 2003; Marichal et al., 2001; Rahmoun et al., 2001; Koprinska et al., 2000; Wang et al. 1999; Whitfort et al., 1995).

DEVELOPMENT OF ADAPTIVE NEURO-FUZZY SYSTEMS

Developing an Adaptive Neuro-fuzzy system is a process that is similar to the procedures used to create fuzzy logic systems, and neural networks. One advantage of this combined approach is that it is usually no more complicated than either approach taken individually.

As noted above, there are two methods of creating a Neuro-fuzzy system; integrating fuzzy logic into a neural network framework (FNN), and implementing neural networks into a fuzzy logic system (NFS). A fuzzy neural network is just a neural network with some fuzzy logic components; hence is generally trained like a normal neural network is.

Training Process: The training regimen for a NFS differs slightly from that used to create a neural network and a fuzzy logic system in some key ways, while at the same time incorporating many improvements over those training methods.

The training process of a Neuro-fuzzy system has five main steps: (Von Altrock, 1995, pp. 71-75)

- **Obtain Training Data:** The data must cover all possible inputs and output, and all the critical regions of the function if it is to model it in an appropriate manner.