

## Sistemas Neuro-Difusos: Técnicas de Modelização Não Linear

José A. B. Vieira, Alexandre M. N. Mota

**Resumo** - As técnicas de inteligência computacional baseadas em lógica difusa e redes neuronais são muitas vezes aplicadas em conjunto. As razões de se combinar esses dois paradigmas vem das dificuldades e limitações inerentes a cada uma das técnicas isoladas. Genericamente, quando são utilizadas de forma combinada, são chamadas de sistemas Neuro-Difusos. Este termo, porém, é muitas vezes utilizado para designar um tipo específico de sistema que integra ambas as técnicas. Esse tipo de sistema caracteriza-se por um sistema difuso cujos conjuntos e regras difusas são ajustadas mediante a apresentação de pares de treino. Existem diversas implementações de sistemas neuro-difusos, pois cada autor tem definido o seu próprio modelo. Este artigo apresenta uma visão geral da área descrevendo, resumidamente, as técnicas neuro-difusas híbridas mais conhecidas, suas vantagens e desvantagens. Existem inúmeros exemplos de aplicação bem sucedidos destas arquiteturas em quase todas as áreas de conhecimento.

**Abstract** - The artificial intelligence techniques based in fuzzy logic and neural networks are many times applied together. The reasons to combine these two paradigms come out of the difficulties and inherent limitations of each isolated technique. Generically, when they are used in a combined way, they are called Neuro-Fuzzy systems. This term, however, is many times used to assign a specific type of system that integrates both techniques. This type of system is characterised for a fuzzy system where fuzzy sets and fuzzy rules are adjusted using training data. There are several different implementations of Neuro-Fuzzy systems; therefore each author has defined its specific model. This article summarizes a general vision of the area describing the most known hybrid neuro-fuzzy techniques, its advantages and disadvantages.

### I. INTRODUÇÃO

As modernas técnicas de inteligência computacional têm encontrado aplicação em quase todos os campos do conhecimento humano. Porém, um destaque muito maior é dado nas áreas de ciências exactas. Talvez a maior expressão do sucesso dessas abordagens esteja no campo da engenharia. Redes neuronais e lógica difusa são duas dessas técnicas muitas vezes aplicadas em conjunto para a resolução de problemas de engenharia para as quais as teorias as teorias clássicas não fornecem uma solução viável. O termo sistema Neuro-Difuso nasceu naturalmente da fusão dessas duas técnicas. Como cada

investigador combina essas duas ferramentas de diferentes maneiras, criou-se uma certa confusão sobre que tipo de sistema este termo designava. Ainda não há consenso absoluto mas, em geral, o termo Neuro-Difuso refere-se a um tipo de sistema caracterizado por uma estrutura semelhante a um controlador difuso em que os termos (conjuntos difusos) e regras são ajustados mediante a apresentação de pares de treino. Tal sistema exhibe duas formas distintas de comportamento. Numa primeira fase, fase de aprendizagem, comporta-se como uma rede neuronal, aprendendo alguns parâmetros e características essenciais ao seu funcionamento futuro. Posteriormente, na fase de processamento, comporta-se como um sistema difuso. Isoladamente, cada uma dessas técnicas possui vantagens e desvantagens que, quando combinadas, compensam-se e proporcionam soluções melhores que aquelas resultantes da aplicação de apenas uma delas.

#### A. *Sistemas Difusos*

Os sistemas difusos propõem uma abordagem matemática para traduzir e processar o conhecimento subjectivo obtido de um especialista, ou seja, uma maneira de manipular conhecimento com um determinado nível de incerteza. A teoria dos conjuntos difusos foi desenvolvida, a partir de 1965, por Lofti A. Zadeh [1].

O comportamento de tais sistemas é descrito por um conjunto de regras difusas do tipo:

**SE <premissa> ENTÃO <consequência>**

utilizando variáveis linguísticas com termos simbólicos. Cada termo representa um conjunto difuso. Os termos do espaço de entrada (tipicamente 5-7 por variável linguística) compõem uma partição difusa.

O processamento consiste de três etapas: Na primeira etapa, os valores das entradas numéricas são mapeados por uma função de pertinência de acordo com o grau de compatibilidade (pertinência) com os respectivos conjuntos a esta operação pode-se chamar “fuzificação”. No segundo passo, o sistema difuso processa as regras de acordo com os graus de compatibilidade. No terceiro passo, os valores difusos resultantes são transformados novamente em valores numéricos, a esta operação pode-se chamar “desfuzificação”. Essencialmente, este procedimento torna possível usar categorias difusas como representação de palavras e ideias abstractas dos seres humanos para a descrição do procedimento de tomada de decisão.

As vantagens dos sistemas difusos são:

- [1] capacidade de representar incertezas inerentes ao conhecimento humano através da utilização de variáveis linguísticas;
- [2] interacção mais simples do especialista da área com o projectista do sistema;
- [3] fácil interpretação dos resultados, devido a representação quase natural das regras;
- [4] fácil extensão da base de conhecimento através da adição de novas regras, sem perda do conhecimento anterior;
- [5] possibilidade de aplicação sem necessidade de modelos matemáticos do processo, apenas conhecimento empírico;
- [6] robustez em relação a perturbações no sistema (esta afirmação pode ser confirmada na prática, mas é difícil comprová-la teoricamente).

As desvantagens destes sistemas são:

- [1] não é capaz de generalizar, ou seja, só responde ao que está descrito em sua base de regras;
- [2] não é robusto em relação a mudanças topológicas do sistema ao qual é aplicado, pois tais mudanças exigiriam alterações na base de regras;
- [3] depende da existência de um especialista para determinar as regras de inferência lógica;
- [4] é sensível à subjectividade do especialista.

### B. Redes Neurais

As redes neurais tentam modelar directamente as funções biológicas do cérebro humano. Isto leva à idealização dos neurónios como unidades discretas de processamento distribuído. As suas conexões locais ou globais dentro de uma rede são também idealizadas, levando assim à capacidade do sistema nervoso em assimilar, aprender ou prever reacções ou decisões a serem tomadas. O primeiro modelo de Rede Neuronal foi descrito por W. S. McCulloch e W. Pitts e o primeiro algoritmo de treino foi desenvolvido por F. Rosenblatt (Perceptron) e B. Widrow (Adaline).

A principal característica das redes neurais é o facto destas estruturas aprenderem com exemplos (pares de entradas e saídas). As redes neurais modificam a sua estrutura interna, os pesos das conexões entre os seus neurónios artificiais, no sentido de mapear, com um nível de erro aceitável para a aplicação, as relações entre entradas e saídas que representam o comportamento do sistema a ser modelado.

As vantagens das redes neurais são:

- [1] capacidade de aprendizagem ;
- [2] capacidade de generalização: responde correctamente a novos casos desde que esses sejam semelhantes aos casos aprendidos;
- [3] robustez em relação a perturbações.

E suas desvantagens são:

- [4] interpretação da funcionalidade não é possível, pois a informação está armazenada de forma distribuída;
- [5] dificuldade em determinar o número de camadas e o número de neurónios ideais para um dado problema (a escolha é tipicamente feita através de tentativa e erro).

## II. SISTEMA NEURO-DIFUSO

Desde o momento em que os sistemas difusos se tornaram uma aplicação comum no meio industrial, os projectistas perceberam que o desenvolvimento de um sistema difuso de boa performance não é uma tarefa fácil. O problema de encontrar funções de pertinência e regras apropriadas é, frequentemente, um processo cansativo de tentativa e erro. Assim, surgiu a ideia de aplicar algoritmos de aprendizagem aos sistemas difusos. As redes neuronais, eficientes algoritmos de aprendizagem, apresentaram-se como uma alternativa adequada para automatizar ou dar suporte ao desenvolvimento de sistemas difusos.

Os primeiros estudos dos sistemas Neuro-Difuso datam do início da década de 90, com Jang, Takagi e Hayashi e ainda Lin e Lee todos em 1991, Berenji em 1992 e Nauck a partir de 1993, etc. A maioria das aplicações desta época era voltada para controlo. Gradualmente foram aparecendo aplicações voltadas para outros temas, como a análise e classificação de dados, detecção de falhas e apoio à decisão, etc.

As redes neurais e os sistemas difusos podem ser integrados (ou combinados) para unirem as suas vantagens e sanarem as suas deficiências individuais. As redes neurais introduzem suas características computacionais e de aprendizagem nos sistemas difusos e recebem destes o poder de representação e interpretação. Assim, as desvantagens dos sistemas difusos são compensadas pelas capacidades das redes neuronais e as vantagens dos sistemas difusos contornam as limitações das redes neuronais. As técnicas são complementares, o que justifica a combinação das duas na solução de problemas aproveitando o máximo das potencialidades de cada uma delas e evitando as suas limitações.

## III. TIPOS DE SISTEMAS NEURO-DIFUSO

De uma forma geral, todas as combinações de técnicas baseadas em redes neuronais e lógica difusa podem ser chamadas Neuro-Difusas. As diferentes combinações dessas técnicas podem, em geral, ser divididas de acordo com a seguinte classificação [2]:

**Sistema Neuro-Difuso Cooperativo:** No modelo cooperativo existe um pré-processamento onde os mecanismos de aprendizagem das redes neuronais determinam sub-blocos do sistema difuso com os dados de treino, como os conjuntos difusos e/ou regras difusas (memórias associativas difusas [3] ou o uso de classificadores para determinar as regras e seu posicionamento[4]). Depois dos sub-blocos difusos calculados a rede neuronal é posta de parte ficando só o sistema difuso.

**Sistema Neuro-Difuso Concorrente:** No modelo concorrente a rede neuronal e o sistema difuso trabalham em conjunto continuamente na mesma tarefa sem que um sistema seja utilizado para determinar as características do outro. Em geral, a rede neuronal pré-processa as entradas (ou pós-processa as saídas) do sistema difuso.

**Sistema Neuro-Difuso Híbrido:** Nesta categoria, uma rede neuronal é usada para aprender vários parâmetros do sistema difuso (conjuntos difusos, regras difusas e, algumas vezes, pesos das regras) de um sistema difuso de uma forma iterativa.

A maioria dos investigadores usa o termo neuro-difuso para designar apenas o sistema neuro-difuso híbrido [5], [4], [6] e [7].

#### IV. SISTEMA NEURO-DIFUSO COOPERATIVO

Num sistema cooperativo estamos perante um sistema neuro-difuso, no entanto a rede neuronal só é usada numa fase inicial. Neste caso as redes neuronais determinam sub-blocos do sistema difuso com os dados de treino, depois desta operação a rede neuronal é posta de parte ficando só o sistema difuso a ser executado.

Nos sistemas neuro-difusos cooperativos, a estrutura não é totalmente interpretável o que seria considerado uma desvantagem.

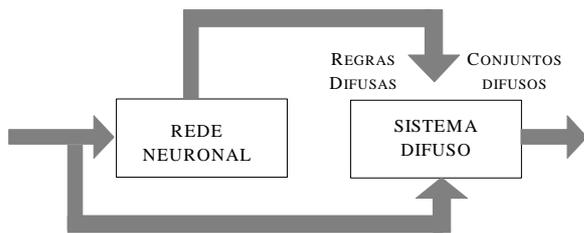


Figura 1. Sistema Cooperativo

#### V. SISTEMA NEURO-DIFUSO CONCORRENTE

Um sistema concorrente não é um sistema neuro-difuso no sentido estrito, porque a rede neuronal não afecta o sistema difuso. A rede neuronal modifica a saída do sistema difuso, isto é, os dados entram no sistema difuso, são pré-processados e, a seguir, a rede neuronal processa a saída do sistema difuso, em conjunto com os dados iniciais ou ao contrario, isto é, o sistema difuso modifica a

saída da rede neuronal, isto é, os dados entram na rede neuronal, são pré-processados e, a seguir, o sistema difuso processa a saída da rede neuronal.

Nos sistemas neuro-difusos concorrentes, os resultados não são completamente interpretáveis, o que pode ser considerado uma desvantagem.

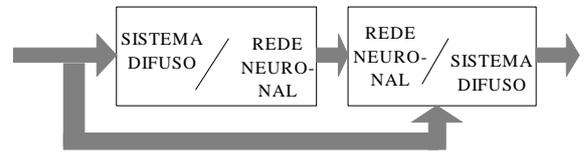


Figura 2. Sistema Concorrente

#### VI. SISTEMA NEURO-DIFUSO HÍBRIDO

Na definição de Nauck [2]: “Um sistema Neuro-Difuso é um sistema difuso que usa um algoritmo de aprendizagem com base nas derivadas ou inspirado pela teoria de redes neuronais (estratégias de aprendizagem heurísticas) para determinar seus parâmetros (conjuntos difusos e regras difusas) através do processamento de dados de treino (entrada e saída)”.

Um sistema neuro-difuso pode ser interpretado como um conjunto de regras difusas. Este sistema pode ser criado totalmente a partir de dados de entrada e saída, ou inicializado com o conhecimento que se tenha *a priori* em forma de regras difusas.

O sistema resultante da fusão de sistemas difusos e redes neuronais tem como vantagens a aprendizagem através de dados de treino (característica das redes neuronais) e a fácil interpretação da funcionalidade (característica dos sistemas difusos).

Existem diversas abordagens para o desenvolvimento de sistemas neuro-difusos híbridos, pois, por ser um tema de pesquisa bastante recente, cada investigador tem definido o seu modelo particular. Estes modelos são semelhantes na sua essência, mas apresentam diferenças fundamentais.

Muitos tipos de sistemas neuro-difusos são representados por uma rede neuronal que implementa funções lógicas. Isto não é necessário para a aplicação de um algoritmo de aprendizagem a um sistema difuso. Entretanto, a representação por redes neuronais é mais conveniente porque permite visualizar o fluxo de dados através do sistema e os sinais de erro que são utilizados para actualizar os seus parâmetros. Um benefício adicional é permitir que os diferentes modelos sejam comparados e que as diferenças estruturais sejam facilmente percebidas.

As arquitecturas neuro-difusas mais conhecidos são:

*Fuzzy Adaptive Learning Control Network* (FALCON) C. T. Lin e C. S. Lee [8];

*Adaptive Network based Fuzzy Inference System* (ANFIS) R. R. Jang [9];

*Generalized Approximate Reasoning based Intelligence Control* (GARIC) H. Berenji [10];

**Neuronal Fuzzy Controller** (NEFCON) D. Nauck [11];  
**Fuzzy Inference and Neural Network in Fuzzy Inference Software** (FINEST) Tano, O. e Arnould [12];  
**Fuzzy Net** (FUN) S. Sulzberger, N. Tschichold e S. Vestli [13];

**Fuzzy Neural Network** (NFN) Figueiredo e Gomide [14];

**Dynamic/Evolving Fuzzy Neural Network** (EFuNN and dmEFuNN) Kasabov e Song [15];

**Self Constructing Neural Fuzzy Inference Network** (SONFIN) Juang e Lin [16].

De seguida são descritas as várias arquitecturas mencionados de uma forma resumida.

#### A. *Arquitectura FALCON*

A Arquitectura **Fuzzy Adaptive Learning Control Network** FALCON [8] é uma arquitectura de cinco camadas como mostra a figura 3. Existem dois nós linguísticos para cada variável de saída. Um é para os dados de treino e o outro é para a saída real da arquitectura FALCON. A primeira camada escondida é responsável pelo mapeamento das variáveis de entrada relativamente às funções de pertença. Cada nó pode ser um nó simples que representa uma simples função de pertença ou um composto de nós multi-camada que calculam uma função de pertença complexa. A segunda camada escondida define os antecedentes das regras seguido dos consequentes na terceira camada escondida. A arquitectura FALCON usa um algoritmo de aprendizagem híbrido composto por uma aprendizagem não supervisionada para definir as funções de pertença e base de regras iniciais e o algoritmo de aprendizagem baseado no gradiente descendente para otimizar e ajustar os parâmetros das funções de pertença para produzir a saída desejada.

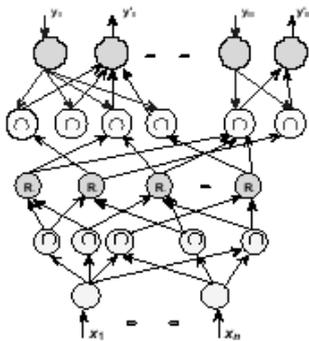


Figura 1. Arquitectura FALCON.

#### B. *Arquitectura ANFIS*

A arquitectura **Adaptive Network based Fuzzy Inference System** ANFIS [9] implementa um sistema difuso de inferência do tipo Takagi Sugeno e é, como mostra a figura 4, constituído por cinco camadas. A primeira camada escondida é responsável pelo mapeamento das

variáveis de entrada relativamente às funções de pertença. O operador norma – T é aplicado na segunda camada escondida para calcular os antecedentes das regras. A terceira camada escondida normaliza o grau de compatibilidade/pertença das entradas seguida da quarta camada escondida onde os consequentes das regras são determinados. A camada de saída calcula a saída global como a soma de todos os sinais que chegam a esta camada. A arquitectura ANFIS usa o algoritmo de aprendizagem de retropropagação para determinar os parâmetros das funções de pertença de entrada e o algoritmo do método dos mínimos quadrados para determinar os parâmetros nos consequentes. Cada passo do algoritmo de aprendizagem tem duas partes. Na primeira os pares de dados de entrada são propagados e os parâmetros dos consequentes são calculados usando o algoritmo iterativo do método dos mínimos quadrados, enquanto os parâmetros das premissas são considerados fixos. Na segunda parte os pares de dados de entrada são propagados novamente e, em cada iteração, o algoritmo de aprendizagem de retropropagação é usado para modificar os parâmetros das premissas, enquanto os consequentes se mantêm fixos. Este procedimento é iterativo.

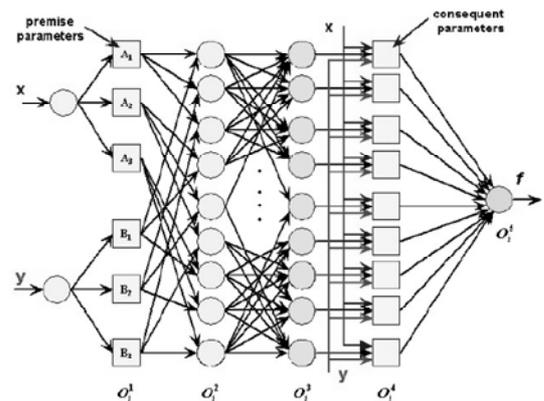


Figura 4. Arquitectura ANFIS

#### C. *Arquitectura GARIC*

A arquitectura **Generalized Approximate Reasoning based Intelligence Control** GARIC [10] implementa um sistema Neuro-Difuso usando dois módulos de redes neuronais, o ASN (Action Selection Network) e o AEN (Action State Evaluation Network). O AEN é um avaliador adaptativo das acções do ASN. O ASN do GARIC é uma rede avançada de cinco camadas. A figura 5 ilustra a estrutura GARIC – ASN. As ligações entre as camadas não são pesadas. A primeira camada escondida armazena os valores linguísticos de todas as variáveis de entrada. Cada unidade de entrada só liga aos nós da primeira camada, os quais representam os seus valores linguísticos associados. A segunda camada escondida representa os nós das regras difusas que determinam o grau de compatibilidade de uma regra usando um operador softmin. A terceira camada escondida representa os valores linguísticos da variável de saída do sistema. As

conclusões de cada regra são calculadas dependendo do peso dos antecedentes das regras calculados no nó da camada de regras. A arquitectura GARIC usa o método da media dos máximos locais para calcular a saída das regras. Este método precisa de um valor numérico da saída de cada regra. Assim, as conclusões tem de ser transformadas de valores difusos para valores numéricos antes de serem acumuladas no valor final de saída do sistema. GARIC usa uma mistura do gradiente descendente com uma aprendizagem de reforço para um ajuste fino dos parâmetros dos nós.

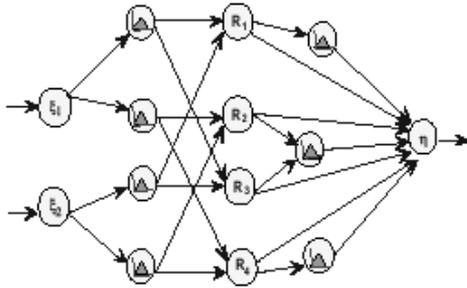


Figura 5. Arquitectura GARIC.

A arquitectura Neuronal Fuzzy Controller NEFCON [17] foi desenhada para implementar um sistema difuso de inferência do tipo Mamdani como ilustra a figura 6. As ligações nesta arquitectura são pesadas com conjuntos difusos e regras usando os mesmos antecedentes chamado pesos partilhados, que são representados por elipses desenhadas á volta das ligações. Elas asseguram a integridade da base de regras. As unidades de entrada assumem a função de interface de fuzificação, a interface lógica é representada pela função de propagação e a unidade de saída é responsável pela interface de desfuzificação. O processo de aprendizagem na arquitectura NEFCON é baseado na mistura de aprendizagem de reforço com o algoritmo de retropropagação. Esta arquitectura pode ser usada para aprender uma base de regras do início, se não houver qualquer conhecimento á priori do sistema, ou otimizar uma base de regras definida manualmente. NEFCON tem duas variantes NEFPXOX (para aproximação de funções) e NEFCCLASS (para tarefas de classificação) [17].

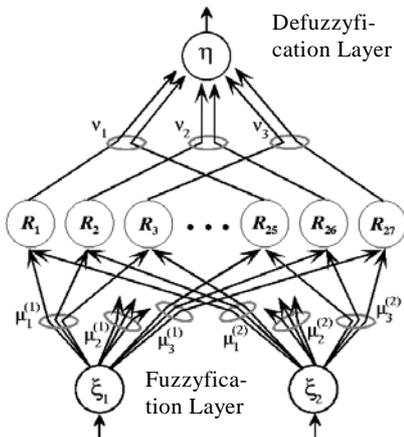


Figura 6. Arquitectura NEFCON.

E. *Arquitectura FINEST*

A arquitectura *Fuzzy Inference and Neural Network in Fuzzy Inference Software* FINEST [12] tem capacidade de implementar dois processos de ajuste, ajuste dos predicados difuso, funções de combinação e ajuste das funções de implicação. O generalizado modus ponens é melhorado das seguintes quatro formas: (1) operadores de agregação que tem sinergia e cancelamento natural (2) uma função de implicação parametrizada (3) uma função de combinação que reduz a *fuzzyness* (4) encadeamento inverso baseado no generalizado modus ponens. A arquitectura FINEST usa o algoritmo de retropropagação para o ajuste fino dos parâmetros. A figura 7 mostra a arquitectura FINEST em camadas e o processo de cálculo da inferência difuso. Esta arquitectura oferece uma ferramenta de ajuste para qualquer parâmetro que apareça nos nós da rede representando o processo de cálculo dos dados difusos se a função derivativa com respeito aos parâmetros for dada.

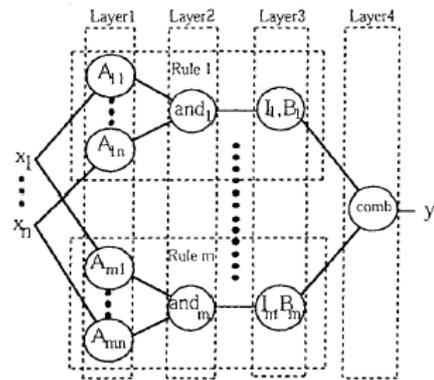


Figura 7. Arquitectura FINEST.

F. *Arquitectura FUN*

Na arquitectura *Fuzzy Net* FUN [13], os neurónios na primeira camada escondida contem funções de pertença e efectuando a fuzificação das variáveis de entrada. Na segunda camada escondida, a conjunção (difusa-AND) é calculada. As funções de pertença das variáveis de saída são armazenadas na terceira camada escondida. As suas funções de activação são o (difuso-OR). Finalmente, o neurónio de saída realiza a desfuzificação. A rede é inicializada com uma base de regras difusas e as correspondentes funções de pertença e usa uma técnica de aprendizagem estocástica que muda os parâmetros das funções de pertença e as ligações dentro da estrutura da rede de uma forma aleatória. O processo de aprendizagem é baseado numa função de custo, que é avaliada após cada modificação aleatória de parâmetros. Se da modificação resultar um aumento de performance a modificação é

guardada, se não, é desfeita e são mantidos os valores anteriores.

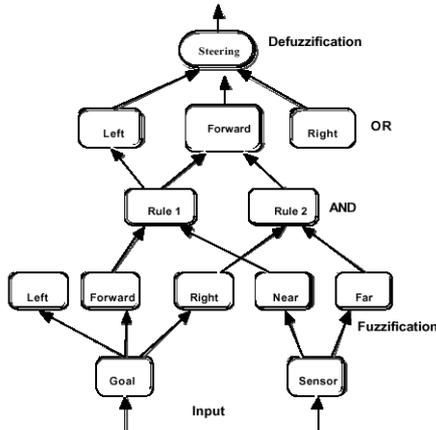


Figura 8. Arquitetura FUN.

G. *Arquitetura NFN*

A arquitetura *Neuro Fuzzy Network* NFN [14] implementa um sistema difuso com aprendizagem ilustrado na figura 9. Este sistema usa neurónios lógicos, que implementam funções do tipo t-norma (T) e t-conorma (S). As entradas são fuzzificadas e depois de uma forma pesada são ligadas a neurónios and. São colocados neurónios do tipo and para que todas as combinações possíveis das variáveis de entrada sejam formadas e são colocados neurónios do tipo or para todas as classes de saída. Cada neurónio and é ligado a somente um neurónio or. Os pesos  $w_{kij}$  (pesos das conexões que ligam os termos difusos aos neurónios and) são inicializados como zero e os pesos  $v_{ij}$  (pesos das conexões que ligam os neurónios and aos neurónios or), como um. Estes pesos representam graus de incerteza das regras. O treino/aprendizagem é feita por um algoritmo com duas fases. Primeiro, uma fase de auto-organização para definição dos neurónios da segunda camada e adaptação dos pesos  $w_{kij}$ . Isto significa a aprendizagem das funções de pertinência dos antecedentes de cada regra. A segunda fase usa um esquema de supervisão para a adaptação dos consequentes das regras, pesos  $v_{ij}$  através do método do gradiente descendente.

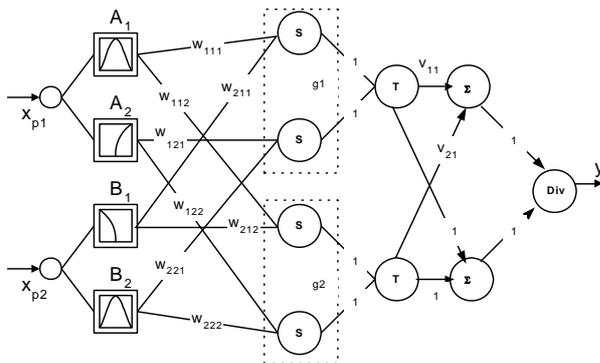


Figura 9. Arquitetura NFN.

H. *Arquitetura EFuNN*

Na arquitetura *Evolving Fuzzy Neural Network* EFuNN [15] todos os nós são criados durante a aprendizagem. A camada passa os dados para a segunda camada, que calcula os graus das funções de pertinência aos quais os valores de entrada pertencem a uma função de pertinência difusa pré-definida. A terceira camada possui os nós das regras difusas que representam protótipos dos dados de entrada/saída como uma associação de hiper-esferas do universo difuso de entrada e do universo difuso de saída. Cada nó de regras é definido por dois vectores de pesos de ligação, que são ajustados através de uma técnica de aprendizagem híbrida. A quarta camada calcula o grau aos quais as saídas das funções de pertinência são combinadas com os dados de entrada e a quinta camada realiza a desfuzzificação e calcula o valor numérico para as variáveis de saída. *Dynamic Evolving Fuzzy Neural Network* (dmEFuNN) [15] é uma versão modificada da EFuNN com a ideia de que não só a activação dos nós das regras vencedoras são propagados mas um grupo de regras dinamicamente escolhido para cada novo par de dados de entrada e os respectivos valores de activação são usados para calcular os parâmetros dinâmicos da função de saída. Enquanto que EFuNN implementa regras do tipo difuso Mamdani, dmEFuNN estima regras difusas do tipo Takagi Sugeno baseadas no algoritmo dos mínimos quadrados.

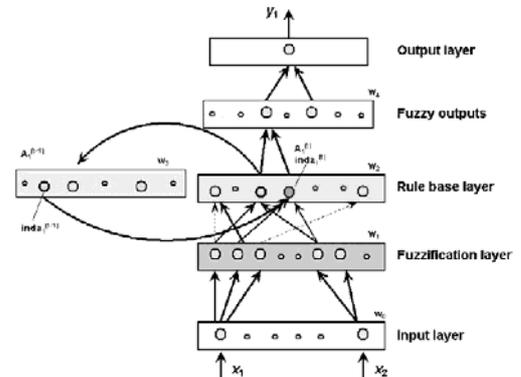


Figura 10. Arquitetura EFuNN.

I. *Arquitetura SONFIN*

A arquitetura *Self Constructing Neural Fuzzy Inference Network* SOFIN [16] implementa um sistema difuso de inferência do tipo Takagi-Sugeno modificado que está ilustrado na figura 10. Na estrutura de identificação dos antecedentes, o espaço das entradas é particionado de uma forma flexível de acordo com um algoritmo baseado no alinhamento de classes. Na identificação da estrutura dos consequentes, só um valor numérico seleccionado por um método de classificação e é atribuído a cada regra inicialmente. Mais tarde, alguns

termos adicionais significativos (variáveis de entrada) seleccionados através de uma projecção baseada na medida da correlação para cada regra são adicionados aos consequentes (formando uma equação linear das variáveis de entrada) incrementalmente enquanto o procedimento de aprendizagem segue. Para a identificação dos parâmetros, os parâmetros consequentes são ajustados e otimizados pelo método da média dos mínimos quadrados ou o algoritmo recursivo do método da média dos mínimos quadrados e os parâmetros dos antecedentes são ajustados pelo algoritmo de retropropagação.

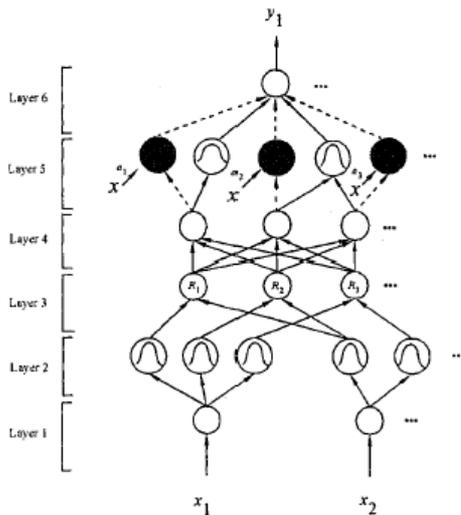


Figura 11. Arquitectura SONFIN.

Para ter uma ideia mais aprofundada desta arquitectura e suas aplicações além das específicas referências apontadas existe um levantamento mais vasto e profundo feito em 2000 por Abraham [18] em que se podem ver com mais pormenor as várias arquitecturas e respectivos algoritmos de aprendizagem.

## VII. DISCUSSÃO E APLICAÇÕES

Como era de esperar, ambos os sistemas concorrentes e cooperativos não são totalmente interpretáveis devido à presença da rede neuronal (conceito de caixa negra). Os sistemas neuro-difusos fundidos ou híbridos apresentam um modelo interpretável e com capacidade de aprendizagem de um modo supervisionado. Nas arquitecturas FALCON, NFN, GARIC, ANFIS, NEFCON, SONFIN, FINEST e FUN o processo de aprendizagem só se ocupa com a adaptação de parâmetros com estruturas do sistema fixas. Para problemas complexos, em grande escala, será muito complicado determinar os parâmetros das premissas e dos consequentes, número de regras, etc. O projectista tem que fornecer os detalhes da arquitectura (tipo e quantidade de funções de pertinência para as variáveis de entrada e saída, tipo de operadores difuso, etc). A arquitectura FINEST fornece um mecanismo baseado no generalizado

modus ponens melhorado para o ajuste fino dos predicados difuso, combinação de funções e ajuste das funções de implicação. Uma importante característica da arquitectura dmEFuNN e EFuNN é fazer o treino só numa passagem (uma iteração) permitindo assim adaptação/aprendizagem on-line de uma forma simples. Já que FUN usa um algoritmo de aprendizagem com base em métodos estocásticos, pode ser questionável se ainda assim pode ser considerado um sistema neuro-difuso. À medida que os sistemas se tornam mais complicados a definição manual da estrutura/parâmetros torna-se muito complicada. Existe quem defenda [5] que uma abordagem evolucionária com base nos algoritmos genéticos para a optimização de todas as variáveis (tipo de sistema difuso, estrutura, parâmetros, operadores de inferência, regras e funções de pertinência) de um sistema neuro-difuso seja a melhor solução a utilizar.

Ao nível da aplicação destas arquitecturas observou-se que inicialmente elas começaram por ser desenvolvidas na área da engenharia da modelização e controlo de processos não lineares, no entanto, actualmente, estas arquitecturas são usadas quase em todas as áreas onde se pretendam modelar funções não lineares. As actuais áreas de aplicação vão desde a medicina, economia, controlo, mecânica, física, química, etc.

## VIII. CONCLUSÕES

Neste artigo é apresentada, de uma forma resumida, a última década de investigação na área da modelização de funções não lineares através de sistemas neuro-difusos. Devido ao vasto leque de ferramentas comuns continua a ser difícil comparar conceptualmente as diferentes arquitecturas e avaliar comparativamente a sua performance. Em termos genéricos a bibliografia aponta que as arquitecturas que implementam sistemas de inferência difusa do tipo Takagi-Sugeno obtêm melhores resultados nas aproximações isto relativamente as arquitecturas que implementam sistemas de inferência difusa do tipo Mamdani, embora sejam computacionalmente mais exigentes. Como guia, para que os sistemas neuro-difusos sejam eficientes, executáveis e inteligentes alguns dos requisitos devem ser conseguidos:

- [1] Aprendizagem rápida (baseada em memória, armazenamento eficiente e capacidade de recuperação);
- [2] Adaptabilidade on-line (adaptar-se a novas características como novas entradas, saídas, ligações, etc...);
- [3] Ajustar-se com o fim de conseguir atingir um erro global baixo;
- [4] Complexidade computacional o mais baixa possível;

A aquisição dos dados e o pré-processamento dos dados de treino é também muito importante para o sucesso da aplicação das arquitecturas neuro-difusas.

Todas as arquiteturas Neuro-Difusas usam as técnicas de gradiente descendente para a aprendizagem dos parâmetros das funções de pertinência. Para uma mais rápida convergência destes parâmetros seria interessante explorar outros eficientes algoritmos de aprendizagem para redes neuronais como a procura gradiente conjugada em vez do algoritmo de retropropagação.

#### REFERÊNCIAS

- [1] L. A. Zadeh; "Fuzzy Sets", Information and Control, 1965, Vol. 8, pp. 338-353.
- [2] D. Nauck, F. Klawonn; R. Kruse, "Foundations of Neuro-Fuzzy Systems", John Wiley & Sons, 1997.
- [3] B. Kosko, "Neural Networks and Fuzzy Systems: A Dynamical System Approach to Machine Intelligence", Prentice Hall, Englewood Cliffs, New Jersey, 1992.
- [4] E. Czogala and J. Leski, "Neuro-Fuzzy Intelligent Systems, Studies in Fuzziness and Soft Computing", Springer Verlag, Germany, 2000.
- [5] Abraham, "Neuro-Fuzzy Systems: State-of-the-Art Modelling Techniques", Connectionist Models of Neurons, Learning Processes and Artificial Intelligence, LNCS 2084, Mira J. and Prieto A. (Eds.), Springer-Verlag Germany, 2001, pp. 269-276.
- [6] L. C. Jain e N. M. Martin (Eds), "Fusion of Neural Networks, Fuzzy Logic and Evolutionary Computing and their Application", CRC Press USA, 1999.
- [7] J S. R. Jang, C.-T. Sun, E. Mizutani, "Neuro-Fuzzy and Soft Computing A Computation Approach to Learning and Machine Intelligence", Matlab Curriculum Series, Prentice Hall, 1997.
- [8] T. C. Lin, C. S. Lee, "Neural Network Based Fuzzy Logic Control and Decision System", IEEE Transactions on Computers, 1991, Vol. 40, no. 12, pp. 1320-1336.
- [9] R. Jang, "Neuro-Fuzzy Modelling: Architectures, Analysis and Applications", PhD Thesis, University of California, Berkley, July 1992.
- [10] H. R. Berenji and P. Khedkar, "Learning and Tuning Fuzzy Logic Controllers through Reinforcements", IEEE Transactions on Neural Networks, 1992, Vol. 3, pp. 724-740.
- [11] D. Nauck, Neuro-Fuzzy Systems: Review and Prospects. Proceedings of the Fifth European Congress on Intelligent Techniques and Soft Computing (EUFIT'97), Aachen, 1997.
- [12] S. Tano, T. Oyama, T. Arnould, " Deep Combination of Fuzzy Inference and Neural Network in Fuzzy Inference", Fuzzy Sets and Systems, 1996, Vol. 82(2), pp. 151-160.
- [13] S. Sulzberger, N. Tschichold e S. Vestli, "FUN: Optimization of Fuzzy Rule Based Systems Using Neural Networks", Proceedings of IEEE Conference on Neural Networks, San Francisco, March 1993, pp. 312-316.
- [14] M. Figueiredo and F. Gomide; "Design of Fuzzy Systems Using Neuro-Fuzzy Networks", IEEE Transactions on Neural Networks, 1999, Vol. 10, no. 4, pp.815-827.
- [15] N. Kasabov e Qun Song, "Dynamic Evolving Fuzzy Neural Networks with 'm-out-of-n' Activation Nodes for On-Line Adaptive Systems", Technical Report TR99/04, Departement of Information Science, University of Otago, 1999.
- [16] F. C. Juang, T. Chin Lin, "An On-Line Self Constructing Neural Fuzzy Inference Network and its applications", IEEE Transactions on Fuzzy Systems, 1998, Vol. 6, pp. 12-32.
- [17] D. Nauck, R. Kurse, "Neuro-FuzzySystems for Function Approximation", 4th International Workshop Fuzzy-Neuro Systems, 1997.
- [18] Abraham and Baikunth Nath, " Hybrid Intelligent Systems: A Review of a decade of Research", School of Computing and Information Technology, Faculty of Information Technology , Monash University, Australia, Technical Report Series, 5/2000, 2000, pp. 1-55.
- [19] D. Nauck, "Beyond Neuro-Fuzzy Systems: Perspectives and Directions". Proceedings of the Third European Congress on Intelligent Techniques and Soft Computing (EUFIT'95), Aachen, 1995.
- [20] D. Nauck; "A Fuzzy Perceptron as a Generic Model for Neuro-Fuzzy Approaches". Proc. Fuzzy-Systems, 2nd GI-Workshop, Munich, 1994.