



Utilizando Informações de Tendência *Fuzzy* para Previsão de Vazões com Redes Neurais

Trabalho de Conclusão de Curso

Engenharia da Computação

Aluno: Tarcísio Daniel Pontes Lucas

Orientador: Prof. Dr. Mêuser Jorge Silva Valença



Tarcísio Daniel Pontes Lucas

Utilizando Informações de Tendência *Fuzzy*
para Previsão de Vazões com Redes Neurais

Monografia apresentada
como requisito parcial
para obtenção do diploma de Bacharel em
Engenharia da Computação pela Escola
Politécnica de Pernambuco - Universidade de
Pernambuco.

Recife, dezembro de 2009.

Seja aquilo que você deseja para o mundo!

Agradecimentos

Agradeço a Deus por ter me dado saúde e oportunidade para chegar até esse momento.

Agradeço a meu pai, José Lucas da Silva, a minha mãe, Terezinha de Jesus Pontes Lucas, e aos meus irmãos, Rodrigo Alexandre Pontes Lucas, André Felipe Pontes Lucas e Andreza Daniela Pontes Lucas por tudo! Vocês são minha base e inspiração para seguir adiante.

Agradeço a minha namorada e futura mãe dos meus filhos, Marianna Cavalcanti Rocha, que participa intensamente de tudo na minha vida.

Agradeço também aos amigos Anderson Berg, Antônio Higor, Hélio Bentzen, Liliane Sheyla, Rafael Barros, Robson David e demais colegas pelo apoio mútuo e a amizade durante esses cinco anos de luta.

Agradeço ao Prof. Dr. Méuser Jorge Silva Valença, pela grande disponibilidade e boa vontade durante toda orientação desse trabalho e ao graduando Cristóvão Rufino, pela parceria em parte das simulações.

Agradeço ainda aos outros orientadores que tive durante o curso, Prof. Ph.D Fernando Buarque de Lima Neto e o mestre Diogo Ferreira Pacheco.

Por fim, agradeço aos meus professores pela boa formação que recebi: Abel Guilherme, Adriano Lorena, Byron Bezerra, Carlos Alexandre, Carmelo Bastos, Cristine Gusmão, Eduardo Calábria, Luis Menezes, Maria Lencastre, Renato Moraes, Ricardo Massa, Sérgio Campello, Sérgio Murilo, Sérgio Soares, Simone Cristian, Tiago Massoni e Wellington Pinheiro.

Resumo

O sistema brasileiro de geração de energia elétrica é basicamente formado por hidroelétricas (aproximadamente 90%) e termoeletricas, de complementação. A energia advinda de usinas termoeletricas tem um elevado custo financeiro e ecológico, se comparada a fontes hidroelétricas. Dessa forma, o Operador Nacional de Sistema Elétrico (ONS) realiza o planejamento energético brasileiro buscando maximizar o potencial hidroelétrico e minimizar o uso de termoeletricas. No entanto, usinas hidroelétricas têm seu potencial diretamente ligado às vazões das bacias hidrográficas onde atuam. Assim, ter boas estimativas das vazões nesses pontos é fundamental na construção de um planejamento seguro e preciso. A ONS utiliza basicamente métodos lineares estatísticos para tal. No entanto, o uso de Redes Neurais Artificiais tem conseguido resultados mais precisos. Esse trabalho tem como objetivo obter previsões de vazões mais precisas no contexto das bacias hidrográficas brasileiras a partir do uso de técnicas de Computação Inteligente, tais como Sistemas Neurais Híbrido, Máquinas de Comitê e Busca Harmônica. Com isso, espera-se que o planejamento brasileiro de geração elétrica torne-se mais preciso e seguro, vindo a beneficiar financeira e ecologicamente o país.

Abstract

The electric generate Brazilian system is basically constituted by hydroelectric (almost 90%) and thermoelectric complementation. The energy arising out thermoelectric plant is more expensive than hydroelectric, in terms of money and ecology. Thus, the national operator of electric system (NOS) gives the Brazilian energy planning one goal: maximizing the hydroelectric power to minimize the use of thermoelectric. However, hydroelectric plants power has a strong connection with the flow basin when it is. Therefore, having good estimate of flow in these points is fundamental in the construction for a safe and accurate planning. The NOS uses linear and statistical methods for this, but Artificial Neural Networks have got better results. This work tries to upgrade the accuracy of the predict flow, in the context of Brazilian basin, by Hybrid Neural System, Committee Machine and Harmonic Search. Thereby, expected that Brazilian generation electrical planning will become more safe and accurate, providing financial and ecological benefits to the country.

Sumário

Lista de Figuras

Lista de Tabelas

Tabela de Símbolos e Siglas

p. 14

1 Introdução

p. 15

1.1 Motivação e problema p. 15

1.2 Objetivos p. 17

1.2.1 Objetivo geral p. 17

1.2.2 Objetivos específicos p. 17

1.3 Estrutura da monografia p. 17

2 Revisão Bibliográfica

p. 19

2.1 Planejamento energético brasileiro p. 19

2.1.1 Modelos de médio prazo p. 19

2.1.2 Modelos de curto prazo p. 20

2.1.3 Modelo diário de operação p. 20

2.2 Redes Neurais Artificiais (RNAs) p. 21

2.2.1 Neurônio biológico p. 21

2.2.2 Neurônio artificial p. 21

2.2.3 RNA *Multilayer Perceptron* (MLP) p. 23

2.2.4 *Back-Propagation* p. 24

2.2.5	Critério de parada: validação cruzada	p. 25
2.3	Lógica <i>Fuzzy</i>	p. 26
2.3.1	Fuzzificação	p. 26
2.3.2	Processo de inferência	p. 28
2.3.3	Defuzzificação	p. 29
2.4	Busca Harmônica (BH)	p. 30
2.4.1	Exemplo	p. 31
2.5	Sistemas Neurais Híbridos (SNHs)	p. 34
2.6	Máquinas de Comitê (MC)	p. 34
3	Metodologia	p. 36
3.1	Bases de dados	p. 36
3.2	Pré-processamento	p. 38
3.3	Predizendo vazões diárias com a RNA MLP	p. 39
3.4	Realizando previsões a partir de um SNH	p. 40
3.4.1	Módulos <i>Fuzzy</i>	p. 43
3.4.2	Módulo neural	p. 44
3.4.3	Método de treinamento e avaliação	p. 44
3.5	RNA MLP com padrões de entrada mistos	p. 45
3.5.1	Definição do atributo <i>Fuzzy</i>	p. 46
3.6	Realizando previsões a partir de uma MC	p. 47
3.6.1	Aprendizado dos especialistas	p. 48
3.6.2	Módulo Fuzzy	p. 48
3.7	Método de seleção de variáveis	p. 49
4	Resultados	p. 51
4.1	Resultados da RNA MLP	p. 51

4.2	Resultados do SNH	p. 52
4.3	Resultado da RNA MLP com padrões de entrada misto	p. 53
4.4	Resultado da MC	p. 55
4.5	Resultado da BH	p. 56
5	Conclusão	p. 59
	Referências	p. 62

Lista de Figuras

1	Neurônio biológico.	p. 21
2	Modelo matemático de um neurônio biológico.	p. 22
3	Exemplo de uma Rede Neural Artificial de múltiplas camadas.	p. 23
4	Exemplo ilustrativo de cálculo do erro δ para um neurônio de saída (em azul) e um escondido (em vermelho), usado na atualização do peso w de cor correspondente.	p. 25
5	Validação cruzada: ponto de parada.	p. 26
6	Função-pertinência triangular para a classe <i>Regular</i>	p. 27
7	Função-pertinência trapezoidal para a classe <i>Regular</i>	p. 27
8	Função-pertinência parabólica para a classe <i>Regular</i>	p. 27
9	Exemplo de fuzzificação para classes de altura de um indivíduo.	p. 28
10	Exemplo que ilustra a dificuldade no processo de defuzzificação.	p. 29
11	Exemplo de defuzzificação da massa de um indivíduo a partir de suas pertinências, pelo método de centro de gravidade.	p. 30
12	Exemplo de uma Memória Harmônica inicializada.	p. 31
13	Exemplo de geração de uma nova harmonia a partir de uma Memória Harmônica.	p. 32
14	Exemplo de uma Memória Harmônica após a substituição da pior harmonia pela gerada.	p. 33
15	Exemplo de uma Memória Harmônica após algumas iterações.	p. 33
16	Exemplo de uma Máquina de Comitê de estrutura estática.	p. 35
17	Exemplo de uma Máquina de Comitê de estrutura dinâmica.	p. 35
18	Gráfico da vazão em função do tempo para a usina do Foz do Areia.	p. 37

19	Gráfico da vazão em função do tempo para a usina de Tucuruí.	p.37
20	Arquitetura da RNA aplicada ao problema de predição de vazões.	p.40
21	Modelo do SNH com Lógica <i>Fuzzy</i> , onde os <i>V</i> 's são vazões do sistema e <i>n</i> corresponde ao número de neurônios escondidos.	p.42
22	Curva com quatro comportamentos distintos: decrescimento acentuado (vermelho), decrescimento atenuado (azul), crescimento acentuado (verde) e crescimento atenuado (laranja).	p.43
23	Passos percorridos por um conjunto de vazões de entrada de um determinado padrão até a obtenção de suas pertinências médias quanto às classes <i>Fuzzy crescimento</i> e <i>decrescimento</i>	p.46
24	Classes <i>Fuzzy</i> para <i>decrescimento</i> , em azul e <i>crescimento</i> , em vermelho.	p.47
25	Arquitetura de uma MC com dois especialistas e um módulo de integração <i>Fuzzy</i>	p.47
26	Áreas de atuação dos especialistas.	p.48
27	Atributo utilizado no módulo <i>Fuzzy</i> da MC.	p.49
28	Resultados do EPMA para 30 simulações distintas, com a escolha do padrão de entrada por parte do Especialista (em vermelho), e pela BH (em azul), para a base de dados do <i>Foz do Areia</i>	p.58
29	Resultados do EPMA para 30 simulações distintas, com a escolha do padrão de entrada por parte do Especialista (em vermelho), e pela BH (em azul), para a base de dados de <i>Tucuruí</i>	p.58
30	Comparativo dos resultados entre os métodos de predição: ONS, RNA, RNA Misto, MC e BH.	p.60

Lista de Tabelas

1	Participação das fontes primárias de energia na geração de eletricidade. Fonte: <i>Electricity in World in 2003 - International Energy Agency Statics</i> - http://www.iea.org/textbase/stats	p. 15
2	Principais características dos modelos de predição de vazão desenvolvidos pelo CEPEL.	p. 20
3	Asmostra da base de dados da usina de Tucuruí.	p. 37
4	Amostra de padrões obtida a partir da base de dados da usina de Tucuruí.	p. 38
5	Exemplos de padrões formados por 16 entradas e sete saídas. As entradas são compostas por 14 vazões e duas pertinências, uma quanto à classe <i>decrescimento</i> ($P_{decrescimento}$) e outra quanto à classe <i>crescimento</i> ($P_{crescimento}$).	p. 45
6	Associação entre os parâmetros taxa de aprendizado e número máximo de ciclos para o experimento com RNA MLP.	p. 52
7	Valores dos parâmetros variados no treinamento da RNA.	p. 52
8	Valores dos parâmetros de treinamento da RNA MLP que trouxeram melhores resultados para as bases de dados de <i>Foz do Areia</i> e <i>Tucuruí</i>	p. 52
9	EPMA para primeira semana de predição por RNAs e pelos padrões vigentes na ONS.	p. 52
10	Parametrização da RNA MLP utilizada dentro do SNH proposto	p. 53
11	Parâmetros para simulação da RNA.	p. 53
12	Valores dos parâmetros que compõem as classes <i>Fuzzy Crescimento</i> e <i>Decrescimento</i> de acordo com a Figura 24, para as bases <i>Foz do Areia</i> e <i>Tucuruí</i>	p. 54
13	Parametrização da RNA MLP utilizada dentro do SNH proposto	p. 54

14	Melhores resultados obtidos e correspondente parametrização da MLP com padrão misto para as bases do <i>Foz do Areia</i> e de <i>Tucuruí</i>	p. 54
15	Parâmetros para simulação da RNA.	p. 55
16	Valores dos parâmetros que compõem o atributo <i>Fuzzy</i> definido na Figura 27.	p. 55
17	Parametrização da RNA com padrões de treinamento com redundância <i>Fuzzy</i>	p. 56
18	Predições mais precisas e respectiva parametrização da MC para as bases de dados de <i>Foz do Areia</i> e <i>Tucuruí</i>	p. 56
19	Parametrização da BH.	p. 56
20	Parametrização da RNA utilizada como geradora de <i>fitness</i> para a BH.	p. 57
21	Padrões de entrada encolhidos pela BH.	p. 57
22	Melhores resultados obtidos e correspondente parametrização da MLP com padrão de entrada escolhido pelo algoritmo de BH, para as bases do <i>Foz do Areia</i> e de <i>Tucuruí</i>	p. 57

Tabela de Símbolos e Siglas

Em ordem alfabética:

- **BH:** Busca Harmônica
- **CEPEL:** Centro de Pesquisas de Energia Elétrica
- **CI:** Computação Inteligente
- **EPMA:** Erro Probabilístico Médio Absoluto
- **MC:** Máquinas de Comitê
- **MH:** Memória Harmônica
- **MLP:** Mult-layer Perceptron (Perceptron de Múltiplas Camadas)
- **ONS:** Operador Nacional de Sistema Elétrico
- **PMO:** Programa Mensal da Operação Energética
- **RNAs:** Redes Neurais Artificiais
- **SH:** Sistemas Híbridos
- **SIN:** Sistema Hidrotérmico Interligado Brasileiro
- **SNH:** Sistemas Neurais Híbridos

1 *Introdução*

Esse capítulo se inicia descrevendo o problema e a motivação do desenvolvimento deste trabalho. Posteriormente são expostos seus principais objetivos. Por fim, ele é encerrado mostrando o conteúdo abordado nos capítulos seguintes.

1.1 Motivação e problema

A geração de energia elétrica no Brasil depende basicamente das vazões que naturalmente afluem em 12 bacias hidrográficas do país. Nelas encontram-se instaladas hidroelétricas que respondem por mais de 90% de toda energia produzida no país. No mundo, essa importância é menor, mas ainda relevante (pouco mais de 16% das fontes geradoras de eletricidade), perdendo apenas para o carvão e o gás natural [1], como mostra a Tabela 1, abaixo. Entretanto estipula-se que a energia elétrica advinda de hidroelétricas e de outras fontes renováveis tenha um crescimento de 56% nos próximos 24 anos, segundo a edição de 2007 da *International Energy Outlook* dos Estados Unidos [2].

Tabela 1: Participação das fontes primárias de energia na geração de eletricidade. Fonte: *Electricity in World in 2003 - International Energy Agency Statics* - <http://www.iea.org/textbase/stats>

Fonte	Participação (%)
Óleo	6,9
Carvão	39,9
Gás Natural	19,3
Hidroelétrica	16,3
Nuclear	15,7
Outras fontes	1,9

As vazões fluviais têm como uma de suas características a inconstância, que, no Brasil, é bastante acentuada. Dessa forma, para manter o fornecimento elétrico no país, algumas estratégias são aplicadas. Uma delas é a utilização de grandes reservatórios, que acumulam água em períodos de grande afluência fluvial, posteriormente utilizadas em períodos de

baixa afluência. A outra se dá através de usinas termoelétricas de complementação, que atuam sempre que as hidroelétricas não dão conta da demanda [1].

Usinas termoelétricas geralmente funcionam com algum tipo de combustível fóssil, como petróleo, gás natural e carvão. O alto preço desses combustíveis causa maior impacto na economia nacional, se comparado a fontes hidroelétricas. Além disso, por serem fontes não renováveis, se tornaram cada vez mais escassas e caras. Outra desvantagem é o impacto ambiental, uma vez que contribui para o aquecimento global, através do efeito estufa e de chuvas ácidas [1].

Tendo em vista que a matriz de energia elétrica brasileira é formada basicamente por usinas hidroelétricas, completada pela atuação de termoelétricas, e que a segunda causa maior impacto econômico e ambiental que a primeira, nota-se que a maximização dos recursos energéticos do Brasil passa pela minimização do uso de suas usinas termoelétrica. No atual contexto, isso pode ser alcançado com a maximização das já instaladas usinas hidroelétricas [1].

A coordenação de operações dos sistemas termoelétricos e hidroelétricos brasileiro é realizada no âmbito do chamado Planejamento da Operação do Sistema Interligado, atualmente executado no Operador Nacional de Sistema Elétrico (ONS). Esse tipo de planejamento é diretamente afetado pela qualidade das previsões hidrológicas [1]. Devido a sua complexidade, ele é atualmente feito em etapas e conta com um conjunto de modelos desenvolvidos no Centro de Pesquisas de Energia Elétrica (CEPEL) [3].

Os modelos construídos pelo CEPEL podem ser classificados como de médio prazo, curto prazo e programação da operação. Dentro da última são consideradas as previsões de afluências diárias de normalmente sete dias. A partir dessas previsões, é feita a programação diária dentro do âmbito do sistema hidrotérmico brasileiro. Dessa forma, o aprimoramento desse tipo de modelo de previsão pode beneficiar largamente o sistema brasileiro de geração de energia elétrica [1].

Atualmente, a previsão das vazões nas bacias hidrográficas do Brasil é feita através de métodos lineares [1]. Entretanto, predições mais precisas já foram obtidas a partir de aplicações não lineares que fazem uso de técnicas de Computação Inteligente (CI), entre elas, algumas Redes Neurais Artificiais (RNAs) [4] e Algoritmos Genéticos [5]. A busca por melhores previsões se deu dentro de todos os contextos de predição hoje utilizados: anual [6] [7], mensal [8] [9], semanal [10] e diário [11] [12] .

O presente estudo se dedicou a melhorar os resultados de predições diárias do estado

da arte. Para tal, foram desenvolvidos novos modelos de predição a partir de combinações inéditas entre as já bem sucedidas RNAs e outras técnicas de CI, tais como Lógica *Fuzzy* [13].

1.2 **Objetivos**

1.2.1 **Objetivo geral**

- Obter predições de vazões mais precisas no contexto das bacias hidrográficas brasileiras a partir do uso de técnicas de CI.

1.2.2 **Objetivos específicos**

- Implementar uma RNA para predição de vazões.
- Realizar simulações em pelo menos duas bases de dados para sete dias à frente.
- Comparar desempenho entre os métodos clássicos, atualmente adotados, e a RNA.
- Desenvolver sistemas que incorporem tendências *Fuzzy* na RNA.
- Realizar simulações com os sistemas desenvolvidos.
- Comparar desempenho entre a RNA e os sistemas desenvolvidos.
- Buscar as entradas que maximizem a precisão das predições da RNA pela técnica de CI denominada Busca Harmônica (BH).
- Comparar os resultados obtidos a partir da técnica de CI BH com os obtidos pela escolha de um especialista.

1.3 **Estrutura da monografia**

O Capítulo 2 reúne o conteúdo teórico necessário para se compreender o trabalho proposto, a saber: planejamento energético brasileiro, RNAs, Lógica *Fuzzy*, BH, Sistemas Neurais Híbridos (SNH) e Máquinas de Comitê (MC). Em seguida, o Capítulo 3 mostra todas as abordagens desenvolvidas para atingir o objetivo geral do projeto, tais como o uso de RNA, SNH e MC. Logo depois, o Capítulo 4 mostra todos os resultados obtidos a partir das aplicações e simulações das técnicas desenvolvidas. Por fim, o Capítulo 5 resume o

trabalho final, reúne dificuldades encontradas, discussões geradas e as conclusões obtidas, além de melhorias e trabalhos futuros.

2 *Revisão Bibliográfica*

Este capítulo descreve todo conteúdo teórico que foi utilizado na tentativa de resolver o problema descrito no capítulo 1. A seção 2.1 mostra como funciona o planejamento energético brasileiro. Em seguida, a seção 2.2 discorre sobre RNAs, uma técnica de CI que vem mostrando ser uma alternativa mais precisa na predição de vazões. Já a seção 2.3 aborda a Lógica *Fuzzy*, fortemente utilizada no estudo corrente. Em seguida, a seção 2.4 descreve uma técnica de busca inspirada em teoria musical, denominada Busca Harmônica (BH). Por fim, as seções 2.5 e 2.6 introduzem dois tipos de sistemas que são formados a partir de outras técnicas de CI, são eles: SNH e MC, respectivamente.

2.1 Planejamento energético brasileiro

A matriz energética brasileira é formada, basicamente, por hidroelétricas (aproximadamente 90% da energia produzida) e termoelétricas, de complementação. Usinas termoelétricas geralmente funcionam com algum tipo de combustível fóssil, o que traz maior impacto econômico e ambiental, se comparado a fontes hidroelétricas. Dessa forma, o objetivo do planejamento energético brasileiro é maximizar o uso das fontes hidroelétricas para minimizar o uso de suas termoelétricas.

O planejamento da operação do sistema hidrotérmico interligado brasileiro (SIN), devido a sua complexidade, é feito em etapas e com a ajuda de uma cadeia de modelos desenvolvidos no âmbito do CEPEL [3]. Os modelos se dividem em: médio prazo, curto prazo e de operação. Quanto menor o intervalo de horizonte de planejamento, mais preciso e detalhado é o modelo.

2.1.1 Modelos de médio prazo

Os modelos de médio prazo constroem políticas a serem usadas no início de cada mês. A partir delas se define o quanto de energia térmica e hidroelétrica serão utilizadas

para atender à demanda do país. O modelo hidrológico utilizado nele é o GEVAZP [14]. Seu horizonte de planejamento é de até 10 anos, mas, normalmente, são considerados apenas os 12 primeiros meses. As afluências hidrológicas nesse contexto são consideradas estocásticas.

2.1.2 Modelos de curto prazo

Nos modelos de curto prazo, os blocos de geração hidráulico e térmico definidos na política de médio prazo para o primeiro mês são divididos em metas semanais. Para cada uma dessas semanas, as vazões são consideradas determinísticas. Um dos modelos utilizados na previsão mensal é o PREVIVAZM [15]. Ainda dentro do modelo de curto prazo, são feitas previsões mensais a partir do modelo GEVAZP num horizonte de até 12 meses, mas, normalmente, utilizam-se apenas os 2 primeiros meses.

2.1.3 Modelo diário de operação

Na programação diária a previsão é considerada determinística. Seu horizonte é normalmente de 7 dias, mas pode chegar a 14. O modelo utilizado hoje é o PREVIVAZH [16], fortemente utilizado na elaboração do Programa Mensal da Operação Energética (PMO). O PREVIVAZH utiliza métodos lineares que consideram fatores externos (precipitações, entre outros) e fatores internos, como vazões em dias anteriores.

Todos os modelos de predição mostrados até então realizam suas previsões a partir de métodos estatísticos e lineares. A Tabela 2 resume as principais características dos modelos desenvolvidos pelo CEPEL.

Tabela 2: Principais características dos modelos de predição de vazão desenvolvidos pelo CEPEL.

Etapa	Médio prazo	Curto prazo	Diário
Modelo hidrológico	GEVAZP	PREVIVAZM/ GEVAZP	PREVIVAZH
Horizonte de planejamento	Até 10 anos	Até 1 ano	Até 14 dias
Discretização	Mensal	Semanal/Mensal	Horária/Diária
Afluências hidrológicas	Estocástico	Determinístico/ Estocástico	Determinístico

2.2 Redes Neurais Artificiais (RNAs)

É uma técnica de CI inspirada no funcionamento do cérebro humano e, como tal, é composta por unidades básicas de processamento (os neurônios artificiais) [4]. No entanto, para entender melhor a formação e o funcionamento do neurônio artificial, é importante ter conhecimento sobre os neurônios biológicos.

2.2.1 Neurônio biológico

O neurônio biológico (Figura 1) é a unidade básica do sistema nervoso humano. Eles recebem sinais de outros neurônios pelos seus dendritos através de ligações sinápticas (Figura 1). Essa informação é processada no corpo celular e posteriormente propagada até os terminais axônicos (Figura 1). No entanto, ela é transmitida para outros neurônios apenas se a intensidade do sinal for maior que o limiar excitatório (Lei do Tudo ou Nada). Dessa forma, os neurônios se conectam e formam um conjunto de grandes e poderosas cadeias de processamento paralelo, entre elas, o cérebro humano [17].

Observando a simplicidade do funcionamento de um neurônio e o quão poderoso podem se tornar, quando conectados, foi desenvolvido o primeiro modelo matemático que simula o seu comportamento: o neurônio artificial.

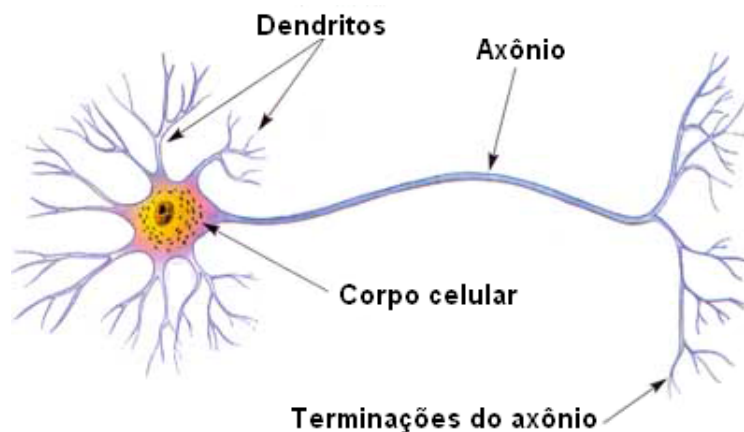


Figura 1: Neurônio biológico.

2.2.2 Neurônio artificial

A primeira proposta de modelo matemático do neurônio biológico foi feita por McCulloch Pitts em 1943 (Figura 2) [18]. O modelo é bastante simples, formado por um conjunto de entradas, uma unidade de processamento e uma ou mais unidades de saídas,

que correspondem, respectivamente, aos dendritos, corpo celular e os terminais axônicos do neurônio biológico [4].

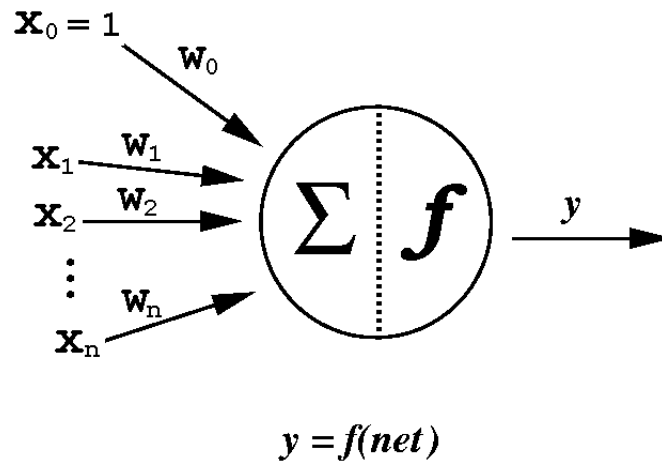


Figura 2: Modelo matemático de um neurônio biológico.

Como pode ser visto na Figura 2, as entradas, nesse tipo de neurônio, são formadas apenas por números. Cada entrada (x_i) é multiplicada por um peso (w_i), que representa como aquela entrada é relevante para o neurônio. A soma ponderada de todas as entradas com seus respectivos pesos (net), dado pela equação Equação 2.1, é usada como parâmetro para função de ativação $f(\text{net})$, a qual retorna o valor de saída do neurônio (y). Nota-se que o valor de x_0 é predefinido e, quando multiplicado por seu peso, representa o valor do limiar excitatório do neurônio [19]).

Os pesos do neurônio devem ter valores, de forma que, para cada entrada, a saída do neurônio se aproxime ao máximo da saída desejada, ou seja, a inteligência nesse modelo de neurônio artificial está nos pesos que o constitui [20].

$$\text{net}_i = \sum_{i=0}^n x_i \times w_i \quad (2.1)$$

Alguns dos primeiros neurônios criados foi o *Perceptron* e o *Adaline*. A diferença entre eles é que o primeiro lida apenas com saída discreta, e o segundo permite respostas no universo contínuo. Essa diferença se dá na escolha da função de ativação do neurônio. As Equações 2.2 e 2.3 são exemplos de funções de ativação dos neurônios *Perceptron* e *Adaline*, respectivamente [21].

$$y_i = \begin{cases} 1, & \text{para } \text{net}_i \geq 0 \\ 0, & \text{para } \text{net}_i < 0 \end{cases} \quad (2.2)$$

$$y_i = \frac{1}{1 + e^{-net_i}} \quad (2.3)$$

A partir da união de neurônios artificiais, com uma ou mais colunas, constroem-se as chamadas Redes Neurais Artificiais (RNAs). Existem vários tipos de RNAs, uma das mais famosas e utilizadas é a *Mult-layer Perceptron* (MLP), que quer dizer *Perceptron* com múltiplas camadas.

2.2.3 RNA *Multlayer Perceptron* (MLP)

Um conjunto de neurônios *Adaline* conectados, com apenas uma camada, formam uma rede neural *Adaline*. As redes MLP podem ser consideradas uma extensão desse tipo de rede, pois permitem também a conexão em múltiplas camadas (Figura 3). Isso é um grande diferencial, pois redes com apenas uma camada são capacitadas a resolver apenas problemas linearmente separáveis, ao passo que duas camadas intermediárias permitem a aproximação de qualquer função. No entanto, existe uma grande dificuldade dentro do contexto de redes de múltiplas camadas: o aprendizado [4].

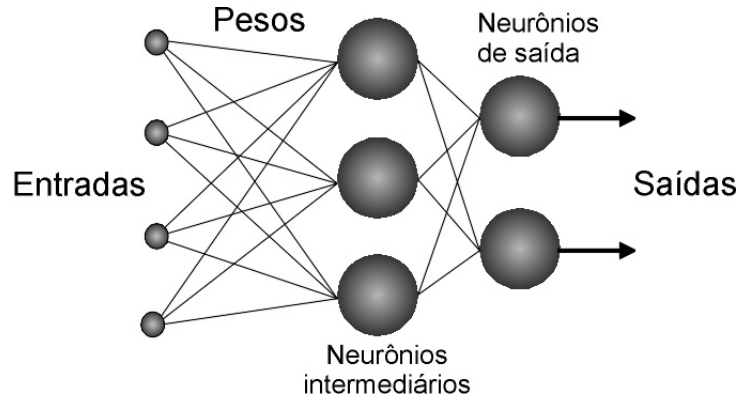


Figura 3: Exemplo de uma Rede Neural Artificial de múltiplas camadas.

Toda RNA tem a capacidade de aprender, decorrente de sua fase de treinamento. É nesse momento que os pesos da rede são modificados, de acordo com alguma abordagem, de forma a se adequar melhor ao problema. Existem três tipos de aprendizado: supervisionado, não-supervisionado e por reforço [4]. A RNA MLP aprende de forma supervisionada.

No aprendizado supervisionado existe o papel do professor, que conhece as respostas corretas para cada entrada da rede. Dessa forma, pode-se calcular o erro na saída da rede e, a partir dele, corrigir os pesos de forma que a rede se adeque melhor ao problema. A

dificuldade em redes de múltiplas camadas está em corrigir os pesos de neurônios que não se encontram na camada de saída [4].

No entanto, já existem algoritmos de aprendizado supervisionado com essa capacidade. Um dos mais bem sucedidos e utilizados em redes MLP é o *Back-Propagation*, que quer dizer propagação recursiva.

2.2.4 *Back-Propagation*

O algoritmo *Back-Propagation* corrige os pesos da rede na ordem inversa da propagação do sinal, ou seja, da camada de saída para camada de entrada [22]. Dois parâmetros importantes dessa correção são a *taxa de aprendizado* e o *momentum*.

A *taxa de aprendizado* revela o quão grande é o tamanho do passo na direção da correção do erro. Já o *momentum*, é utilizado para acelerar a convergência da rede e diminuir a incidência de mínimos locais [20].

A fórmula geral que atribui um novo valor a um determinado peso pode ser vista na Equação 2.4, onde $w_{i,j}(t+1)$ é o novo valor atribuído ao i -ésimo peso do neurônio j , $w_{i,j}(t)$ é o valor corrente do i -ésimo peso do neurônio j (instante t), α é a taxa de aprendizado, δ_j mede o erro no neurônio j , x_i é o sinal de entrada do i -ésimo neurônio, β é a taxa de *momentum* e $w_{j,i}(t-1)$ é o valor do i -ésimo peso do neurônio j no instante $t-1$ [4].

$$\Delta w_{j,i}(t+1) = w_{j,i}(t) + \alpha \delta_j x_i + \beta (w_{j,i}(t) - w_{j,i}(t-1)) \quad (2.4)$$

Se o neurônio de índice j for de saída, o erro δ_j é obtido pela Equação 2.5, onde d_j é a saída desejada, y_j é a saída da rede e $f'(net_j)$ é a derivada da função de ativação. Caso contrário, δ_j é dado pela Equação 2.6, onde δ_l é o erro propagado pelo l -ésimo neurônio da camada imediatamente a frente e $w_{l,j}$ o peso j do neurônio l [4]. A Figura 4 ilustra o cálculo do erro δ_j para um neurônio de saída (em azul) e um neurônio escondido (em vermelho), usado na atualização do peso w de cor correspondente.

$$\delta_j = (d_j - y_j) f'(net_j) \quad (2.5)$$

$$\delta_j = f'(net_j) \sum_l \delta_l w_{l,j} \quad (2.6)$$

No processo de aprendizagem supervisionada, o conjunto de padrões é, normalmente,

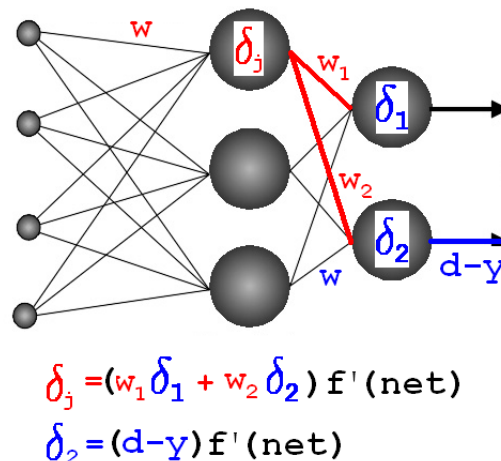


Figura 4: Exemplo ilustrativo de cálculo do erro δ para um neurônio de saída (em azul) e um escondido (em vermelho), usado na atualização do peso w de cor correspondente.

mostrado a RNA várias vezes, determinando o número de épocas do treinamento. Os pesos podem ser modificados a cada amostra, ou a cada época. Encontrar o momento de parada ideal não é uma tarefa tão simples, pois, se treinada demasiadamente, a RNA decora os padrões, perdendo sua capacidade de generalização. Um dos mais usados critérios de parada é o de validação cruzada.

2.2.5 Critério de parada: validação cruzada

O critério de parada de validação cruzada divide o conjunto de padrões em três partes: treinamento, validação e teste, onde apenas as saídas do terceiro não são normalizadas. O conjunto de treinamento é utilizado para correção dos pesos da rede, segundo o algoritmo de treinamento. Depois de cada época, a rede treinada é testada com o conjunto de validação cruzada. Como o conjunto de validação cruzada nunca é utilizado para corrigir os pesos, eles são sempre inéditos para a rede. Dessa maneira, enquanto o erro de validação cruzada estiver diminuindo, a rede continua generalizando.

Entretanto, a partir no momento que o erro de validação cruzada começa a aumentar seguidamente, enquanto simultaneamente o erro de treinamento diminui, é sinalizado que a rede pode estar decorando os padrões de treinamento. Esse é o momento em que o treinamento deve ser parado (Figura 5). Posteriormente, a rede é submetida ao conjunto de teste, onde se calcula a avaliação final da precisão da rede. Vale observar que o cálculo do erro no conjunto no teste é realizado com as variáveis desnormalizadas.



Figura 5: Validação cruzada: ponto de parada.

2.3 Lógica *Fuzzy*

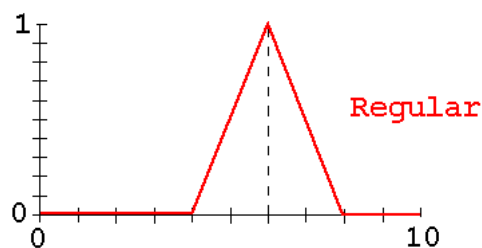
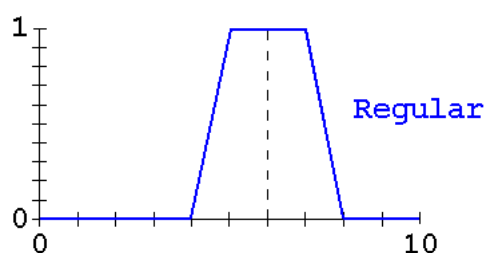
Lógica *Fuzzy* pode ser definida como uma forma de representar elementos do mundo real de forma imprecisa. Impreciso, a princípio, pode parecer algo não desejado no contexto de Ciência Computacional, no entanto, tal teoria mostra que o mundo real é impreciso e que a lógica clássica - Lógica Binária ou Booleana - é limitada em alguns contextos. Na verdade, a Lógica Fuzzy é uma extensão da Lógica Booleana. A Lógica Booleana admite apenas definições precisas, como falso ou verdadeiro; já a Lógica Difusa permite também que se assumam valores intermediários, tais como pouco falso, muito verdadeiro [13] [23].

O raciocínio Fuzzy consiste em implementar as etapas: fuzzificação, inferência e defuzzificação [24].

2.3.1 Fuzzificação

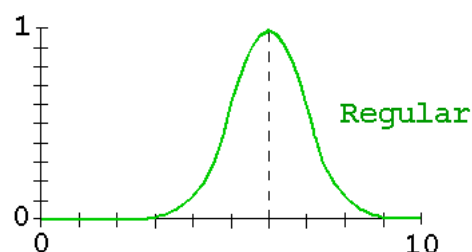
Fuzzificação é o processo em que se definem as variáveis linguísticas e suas respectivas classes e funções-pertinência. As variáveis linguísticas podem ser as mais diversas possíveis. Ao avaliar a qualidade ensino de escolas públicas, por exemplo, a variável linguística *Nota* poderia ser representada pelas classes *Ruim*, *Regular*, *Bom* e *Ótimo*. Para cada classe existe uma função pertinência que a descreve. A classe *Regular*, por exemplo, pode ser representado por uma função triangular (Figura 6), trapezoidal (Figura 7) ou mesmo por uma parábola (Figura 8).

Nota-se que todas as funções pertinência (Figuras 6, 7 e 8) têm, em seu eixo x , os valores reais da variável linguística. Uma nota, por exemplo, pode assumir valores no

Figura 6: Função-pertinência triangular para a classe *Regular*.Figura 7: Função-pertinência trapezoidal para a classe *Regular*

intervalo $[0, 10]$. Dessa forma, o eixo das coordenadas pode assumir qualquer intervalo de valor, dependendo apenas da natureza da variável a qual se pretende representar. Uma nota regular, por exemplo, pode estar entre 5 e 6. Já o eixo y , está no intervalo $[0, 1]$, que representa a pertinência de um determinado valor do eixo x àquela classe.

A ideia na Lógica *Fuzzy* é que, para um dado valor no eixo x , seja normal a existência de mais de uma classe associada a ele. As classes podem ser dispostas de várias formas no eixo da coordenadas. A Figura 9 mostra um exemplo de fuzzificação da variável linguística *Estatuta*, formada pelas classes *baixa*, *média* e *alta*. Observa-se que para certos valores, o indivíduo é considerado alto e médio ao mesmo tempo, mas com pertinências particulares.

Figura 8: Função-pertinência parabólica para a classe *Regular*

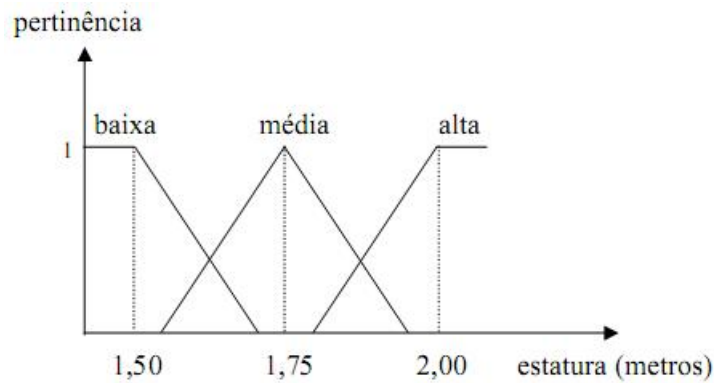


Figura 9: Exemplo de fuzzificação para classes de altura de um indivíduo.

2.3.2 Processo de inferência

O processo de inferência é comumente feito por regras $SE \rightarrow ENTÃO$. As possibilidades nesse contexto também são muitas. Seguem alguns exemplos de regras de inferência na escolha do posicionamento de um jogador de futebol com base na sua estatura e nível de agilidade.

- $SE (Estatura = Alto) E (Agilidade = Baixa) ENTÃO (Posição = Zagueiro)$
- $SE (Estatura = Mediano) E (Agilidade = Alta) ENTÃO (Posição = Meia ofensivo)$
- $SE (Estatura = Alto) E (Agilidade = Mediana) ENTÃO (Posição = Lateral)$

Como *Fuzzy* permite que um mesmo valor esteja vinculado a duas classes, um jogador, por exemplo, pode ser classificado como *Zagueiro* e *Lateral* ao mesmo tempo. Nesses casos deve-se criar algum critério de desempate. O processo de inferência ainda permite atribuir alguma pertinência à classe de saída da regra. Um jogador, por exemplo, pode ter pertinências quanto a habilidade como *Zagueiro* ou *Lateral*. Segue um exemplo de regra que atribui pertinência à classe de saída da regra de inferência, onde x_1 e x_2 são as pertinências quanto às classes de entrada *Alto* e *Baixa*, respectivamente, e z a pertinência quanto à classe *Zagueiro*. Nota-se que, para esse caso, a agilidade (peso 0,3) é mais importante que a estatura (peso 0,7), na qualificação do jogador como zagueiro.

- $SE [Estatura = Alto(x_1)] E [Agilidade = Baixa(x_2)] ENTÃO [Posição = Zagueiro(z = 0,3x_1 + 0,7x_2)]$

2.3.3 Defuzzificação

A defuzzificação é responsável por converter as variáveis linguísticas advindas do resultado de alguma regra de inferência em valores aceitáveis pelo sistema. Se realizadas sob diferentes variáveis, pode ser um processo bastante simples. Por exemplo, dado que o jogador foi qualificado como *Lateral*, focar seu treinamento no desenvolvimento da resistência física.

No entanto, no caso onde se deseja recuperar o valor do mundo real a partir das pertinências quanto a um determinado atributo, ocorrerá perda de informação. A Figura 10 mostra a dificuldade em recuperar a estatura de um indivíduo a partir das pertinências que a compuseram. Nesse caso, o indivíduo tem pertinências relacionadas às classes *baixa* (em azul) e *média* (em vermelho).

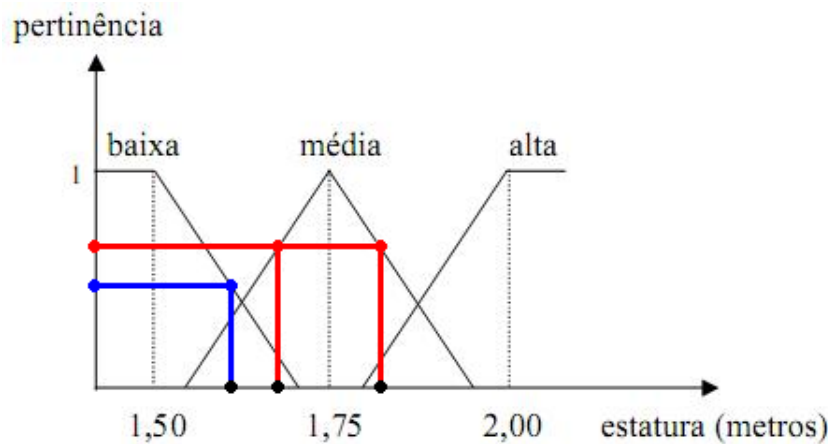


Figura 10: Exemplo que ilustra a dificuldade no processo de defuzzificação.

Foram criadas algumas abordagens na tentativa de recuperar um valor a partir de suas pertinências. Uma forma simples é escolher o maior entre os valores. Outra forma é assumir a média como valor final. No entanto, a forma mais utilizada, é a de recuperação por centro de gravidade [13].

O método de defuzzificação por centro de gravidade é calculado pela Equação 2.7, onde v_i é um dos valores referentes ao eixo das coordenadas, p_i uma das pertinências e n o número de intervalos que serão utilizados no cálculo do centro de gravidade. A Figura 11 mostra um exemplo de defuzzificação da variável *Massa* a partir de suas pertinências. Nesse caso, o valor recuperado foi 60, como mostra o cálculo a seguir.

$$\begin{aligned}
 [h!] \sum_{i=1}^7 v_i p_i / \sum v_i &= ((0,5 \times 50) + (0,5 \times 0) + \dots + (0,7 \times 80)) / (2 \times 0,5) + (5 \times 0,7) \\
 &= 270 / 4,5 \\
 &= 60
 \end{aligned}$$

$$\sum_{i=1}^n v_i p_i / \sum v_i \quad (2.7)$$

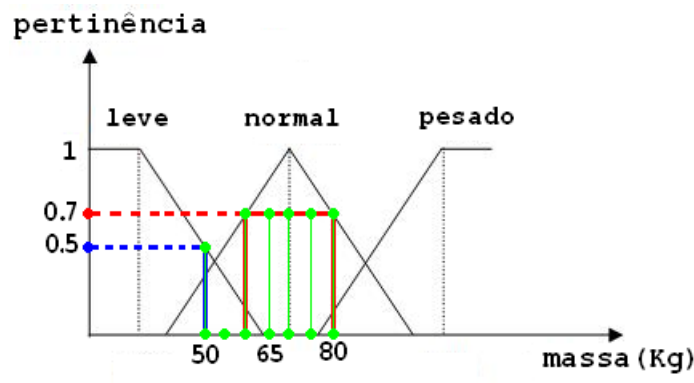


Figura 11: Exemplo de defuzzificação da massa de um indivíduo a partir de suas pertinências, pelo método de centro de gravidade.

2.4 Busca Harmônica (BH)

É uma técnica de CI inspirada na teoria da harmonia musical, designada a resolver problemas de busca e otimização. A harmonia musical pode ser definida como a combinação de sons considerados agradáveis do ponto de vista estético. A relação desse conceito com um algoritmo de busca pode ser explicada em vários aspectos [25].

No tocante à harmonia, um músico busca encontrar o mais rápido possível uma combinação de notas de forma a gerar o melhor acorde possível. Não muito diferente, algoritmos de otimização tentam encontrar o ótimo global de uma determinada função-objetivo o quanto antes. A avaliação da harmonia é determinada por um conjunto de notas tocadas em um determinado instrumento, assim como uma função-objetivo é avaliada a partir de um conjunto de valores atribuídos as suas variáveis. Além disso, para os dois casos, a melhora da busca vem com a prática [25].

O funcionamento do algoritmo BH pode ser definido através dos seguintes passos [25]:

1. Inicialização da Memória Harmônica (MH)
2. Criar uma nova harmonia a partir da MH.
3. Se a harmonia criada for melhor que a pior harmonia presente na MH, retirar a pior harmonia da MH e incluir a nova harmonia dentro dela.
4. Se o critério de parada não for atingido, ir para o passo 2.

2.4.1 Exemplo

O problema-exemplo trata-se de encontrar a melhor combinação entre 5 atributos, de forma a maximizar a resultante da função, representada pela variável denominada *fitness*. A MH será formada por 4 harmonias, podendo ser gerada de diversas formas. Nesse caso será dado 50% de chance de escolha a cada atributo.

A Figura 12 mostra um exemplo de uma MH inicializada, onde X representa um possível atributo a ser escolhido e h uma harmonia, formada pela escolha de um conjunto de atributos X . Os atributos escolhidos estão em vermelho. Na coluna mais a direita, estão os *fitness* correspondentes a cada harmonia h . Note que, apesar de existirem harmonias com diferentes quantidades de atributos setados, na média, metade deles foram escolhidos. O grande retângulo em verde representa o espaço onde o novo indivíduo será criado a partir dos já existentes.

	X_1	X_2	X_3	X_4	X_5	Fitness
h_1						0,6
h_2						0,5
h_3						0,3
h_4						0,1

Figura 12: Exemplo de uma Memória Harmônica inicializada.

A criação de uma nova harmonia é feita a partir da escolha dos atributos das diferentes harmonias existentes na MH (Figura 13). Os atributos escolhidos estão indicados com um X verde. Após o cálculo do *fitness* da nova harmonia gerada, elimina-se a pior harmonia (h_4) e a substitui pela nova harmonia gerada (h_5) (Figura 14). A Figura 15 mostra como pode ficar MH após algumas iterações. Nota-se que a tendência é que as harmonias presentes na MH fiquem parecidas. Um possível critério de parada para esse algoritmo se dá quando o *fitness* médio estiver muito próximo do *fitness* da melhor harmonia.

	X_1	X_2	X_3	X_4	X_5	Fitness
h_1	X					0,6
h_2				X		0,5
h_3		X			X	0,3
h_4			X			0,1
h_5						0,4

Figura 13: Exemplo de geração de uma nova harmonia a partir de uma Memória Harmônica.

	X_1	X_2	X_3	X_4	X_5	Fitness
h_1						0,6
h_2						0,5
h_5						0,4
h_3						0,3

Figura 14: Exemplo de uma Memória Harmônica após a substituição da pior harmonia pela gerada.

	X_1	X_2	X_3	X_4	X_5	Fitness
h_{13}						0,9
h_9						0,9
h_1						0,6
h_{19}						0,6

Figura 15: Exemplo de uma Memória Harmônica após algumas iterações.

BH é um tipo de algoritmo que tem baixo custo computacional, quando comparado a técnicas como Algoritmo Genético [5] [26]. Isso torna-o uma boa alternativa, quando a geração de *fitness* tem custo computacional elevado.

2.5 Sistemas Neurais Híbridos (SNHs)

SNHs são uma categoria de Sistemas Híbridos (SH). Um SH é formado pela junção de pelo menos duas técnicas de CI. A motivação para tal é que uma única técnica, devido as suas limitações e/ou deficiências, pode não ser suficiente para resolver certos tipos de problemas. Nesse contexto, um SH busca unir técnicas de forma a aglutinar seus potenciais, eliminando algumas de suas limitações individuais e gerando uma solução mais robusta e eficiente. Vale observar, entretanto, que tal união nem sempre converge para uma melhor solução [4] [27].

Os SHNs são formados por uma RNA e pelo menos mais uma técnica de CI. A outra técnica pode, por exemplo, ser utilizada na escolha dos parâmetros de uma RNA, como taxa de aprendizado, número de neurônios, função de ativação, ou mesmo na escolhas dos pesos iniciais [4].

2.6 Máquinas de Comitê (MC)

Um princípio bastante utilizado em engenharia é o de dividir para conquistar. A ideia é separar uma tarefa complexa em um conjunto de tarefas mais simples e, posteriormente, combiná-las de forma a atingir a solução para o problema. No contexto de aprendizagem supervisionada, a simplicidade computacional é alcançada distribuindo-se as tarefas de aprendizagem entre RNAs, constituindo redes especialistas. Uma MC é formada pela combinação desses especialistas [20]. As MC podem ser utilizadas tanto em problemas de predição [28], quando de classificação [29].

MC podem ser classificadas em duas grandes categorias: estrutura estática e estrutura dinâmica. Na estrutura estática, os especialistas são combinados por algum mecanismo que não envolve os sinais de entrada, como mostra a Figura 16. Já na estrutura dinâmica, os sinais de entrada interferem no método que combina as saídas dos especialistas em uma saída global, nesse caso, através de uma Rede de Passagem [20], como pode ser observado, na Figura 17.

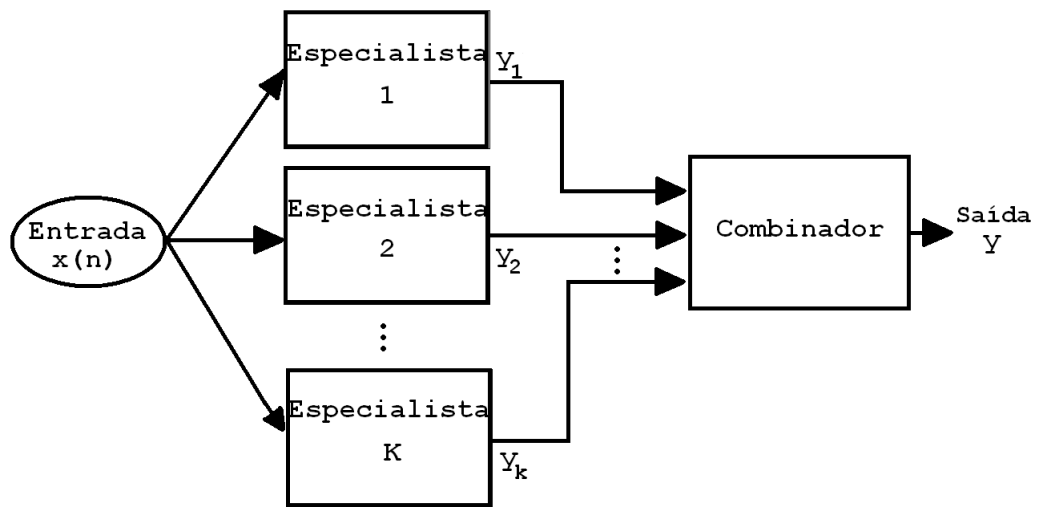


Figura 16: Exemplo de uma Máquina de Comitê de estrutura estática.

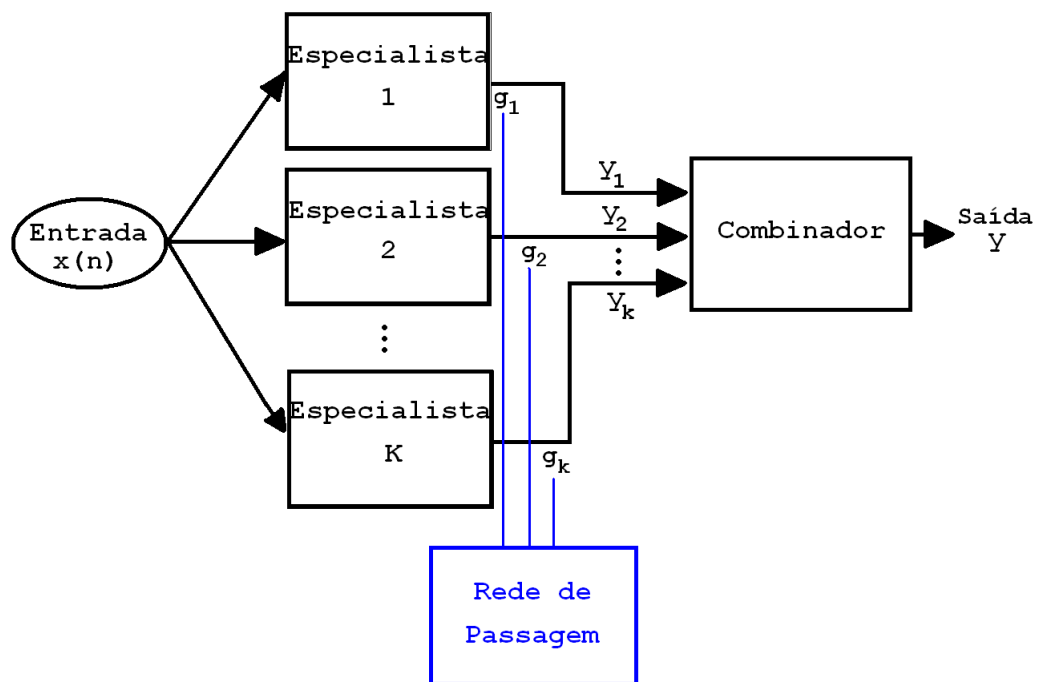


Figura 17: Exemplo de uma Máquina de Comitê de estrutura dinâmica.

3 *Metodologia*

Este capítulo descreve o processo de desenvolvimento e aplicação das teorias apresentadas no Capítulo 2, em cima do problema que o projeto corrente se propôs a resolver. As implementações foram feitas em *Java*, no ambiente de desenvolvimento *NetBeans*.

Inicialmente, na Seção 3.1, são descritas as características das bases de dados as quais se aplicam as técnicas. Em seguida, a Seção 3.2 discorre sobre o pré-processamento dos dados. Já a Seção 3.3, explica como a RNA foi aplicada ao problema de predição de vazões diárias.

A partir de então, as três seções seguintes buscam melhorar os resultados obtidos na Seção 3.3. A primeira delas, através de um SNH (Seção 3.4); já a segunda, pela inserção de novos dados na base de treinamento da RNA (Seção 3.5). A terceira (Seção 3.6), por fim, com a construção de uma MC. A última seção desse capítulo (Seção 3.7) aborda sobre uma forma automática de escolher os parâmetros de entrada utilizados no aprendizado de uma RNA.

3.1 Bases de dados

Os experimentos são realizados com duas das 10 mais importantes usinas hidroelétricas do Brasil. Uma delas é a de Tucuruí, a segunda mais potente, localizada no rio Tocantins, município de Tucuruí, estado do Pará. A outra é a do Foz do Areia, nona mais potente, localizada no rio Iguaçu, município de Pinhão, estado do Paraná.

A base de dados do Foz do Areia contem suas vazões diárias entre março de 1968 e dezembro de 2001. Elas formam uma série temporal sem sazonalidade (Figura 18), o que dificulta a precisão das estimativas das próximas vazões. Já a base de Tucuruí, é uma série temporal com sazonalidade (Figura 19), e tem suas vazões diárias entre setembro de 1969 e dezembro de 2001. A formatação das duas bases de dados é igual. A Tabela 3 mostra a disposição dos dados para a usina de Tucuruí.

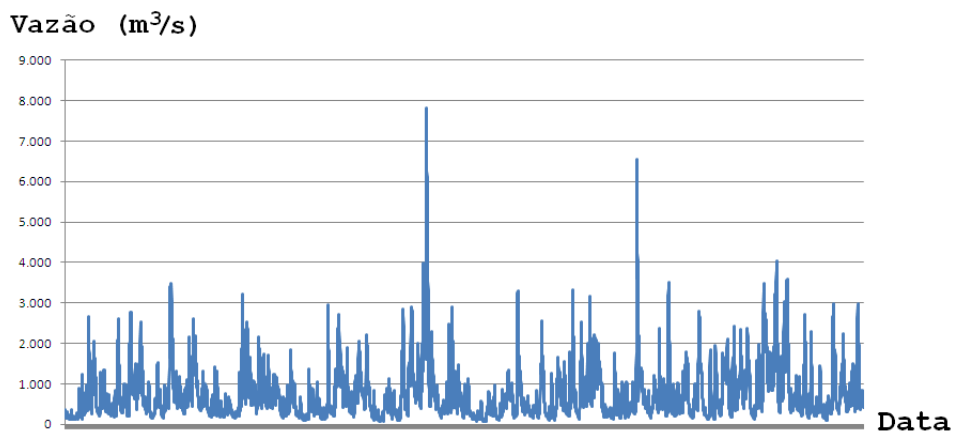


Figura 18: Gráfico da vazão em função do tempo para a usina do Foz do Areia.

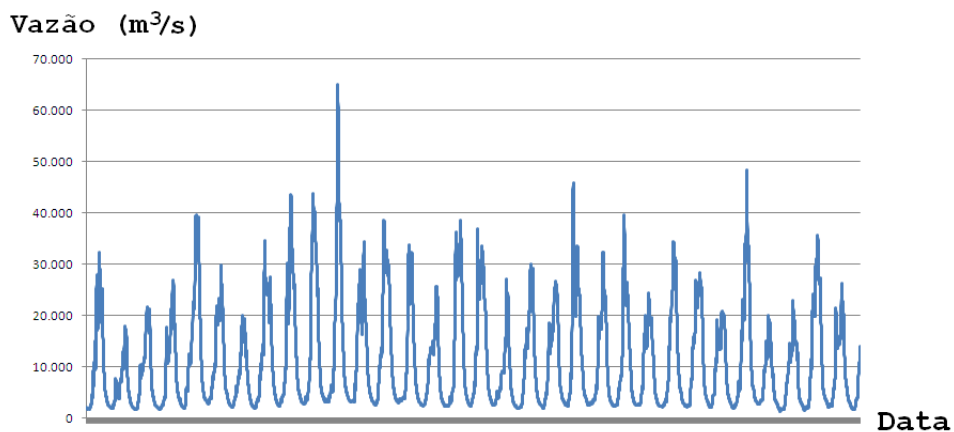


Figura 19: Gráfico da vazão em função do tempo para a usina de Tucuruí.

Tabela 3: Amostra da base de dados da usina de Tucuruí.

Data	Vazão (m^3/s)
01/09/1969	2110
02/09/1969	2089
03/09/1968	2050
...	...
29/12/2001	13036
30/12/2001	13547
31/12/2001	14013

Entretanto, para serem usados em alguma técnica de CI, os dados, normalmente, passam pela etapa de pré-processamento, momento em que são estruturados de acordo com a técnica.

3.2 Pré-processamento

A primeira etapa do pré-processamento é transpor os dados do formato original (Tabela 3) para a forma padrão do problema (Tabela 4). O padrão é formado por n vazões consecutivas, onde $n > 7$. Para cada padrão, os primeiros valores serão utilizados para estipular os sete seguintes, que correspondem ao que se quer prever. A quantidade de valores consecutivos por padrão é feita de forma automática, pois a implementação do pré-processamento foi feita de forma a permitir isso.

Tabela 4: Amostra de padrões obtida a partir da base de dados da usina de Tucuruí.

Índices de cada padrão									
1	2	...	n-6	n-5	n-4	n-3	n-2	n-1	n
2110	2089	...	2110	2010	1991	1971	1971	1971	2000
2089	2050	...	2010	1991	1971	1971	1971	2000	2000
2050	2110	...	1991	1971	1971	1971	2000	2000	2000
2110	2010	...	1971	1971	1971	2000	2000	2000	1981
2010	1991	...	1971	1971	2000	2000	2000	1981	1990

A etapa seguinte é a de normalização dos valores. Essa etapa é importante para que todas as variáveis recebam a mesma atenção da RNA na fase de treinamento. Além disso, as variáveis devem ter seus valores proporcionais aos limites da função de ativação usada nas camadas de saída da RNA [19].

A normalização é feita no intervalo $[0, 15; 0, 85]$. Algumas vezes os valores são normalizados no intervalo $[0; 1]$, no entanto, alguns algoritmos de treinamento, como o *Back-propagation*, não se comportam muito bem com valores muito próximos de zero. Além disso, é também deixada uma margem de tolerância para que valores inéditos e fora do intervalo dos dados presentes na base possam, ainda, serem representados pelo sistema [19]. Dessa forma, a normalização é feita através da fórmula descrita na Equação 3.1.

$$y = [0, 70(x_i - x_{min}) / (x_{max} - x_{min})] + 15 \quad (3.1)$$

Depois de organizar toda a base de dados em conjuntos de vazões consecutivas, e normalizá-los entre 0,15 e 0,85, os dados, agora estruturados como padrões, podem ser utilizados dentro do contexto de RNAs MLP.

3.3 Predizendo vazões diárias com a RNA MLP

Apesar das constantes pesquisas, MLPs ainda revelam certa complexidade em seu uso. A quantidade de parâmetros configuráveis, tanto de estruturação quanto de simulação, é uma delas [30]. Seguem os parâmetros que são definidos no uso da MLP e do *Backpropagation*, conteúdos teóricos vistos nas Seções 2.2.3 e 2.2.4, respectivamente.

- **Número de entradas.**
- Número de neurônios escondidos (apenas uma camada escondida).
- **Número de neurônios de saída.**
- **Função de ativação.**
- **Critério de parada.**
- **Forma de inicialização dos pesos da rede.**
- Taxa de aprendizado.
- Momentum.
- Número máximo de ciclos.

O negrito define a parametrização que não variou durante as simulações. Tais escolhas estão discriminadas nos parágrafos seguintes.

As previsões para sete dias são feitas com base nos 14 dias anteriores. Dessa forma, o padrão utilizado pela RNA MLP é composto de 21 vazões consecutivas, devidamente normalizadas, como na Seção 3.2. A quantidade de entradas foi escolhida por um especialista nesse tipo de problema, o Dr. Mêuser Valença. Os experimentos são feitos para a usina de Tucuruí e Foz do Areia. A arquitetura da rede pode ser vista na Figura 20, onde se nota que a quantidade de neurônios escondidos não é fixa.

O algoritmo de treinamento utilizado é o *Backpropagation* (Seção 2.2.4). A função de ativação escolhida é a Sigmoidal Logística, descrita na Equação 3.2, onde y_i e net_i são, respectivamente, a saída e a média ponderada dos pesos com as entradas do i -ésimo neurônio. Tal função retorna valores dentro do intervalo $[0; 1]$.

$$y_i = \frac{1}{1 + e^{-net_i}} \quad (3.2)$$

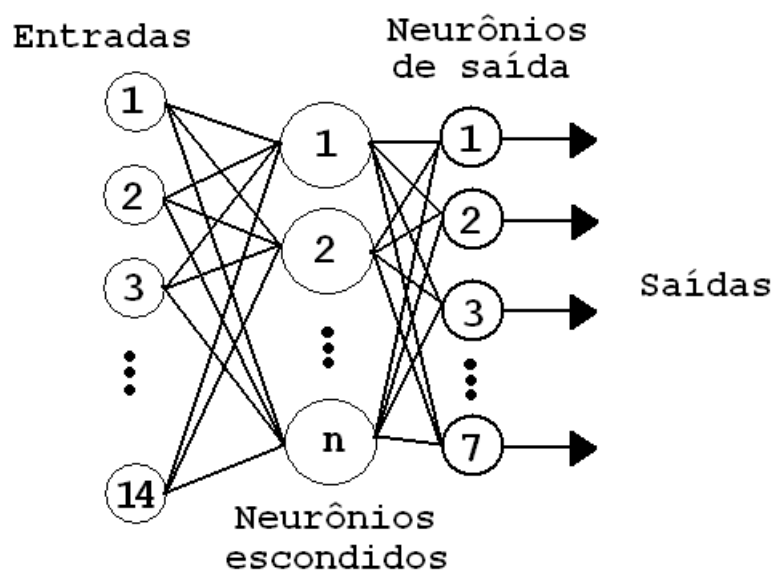


Figura 20: Arquitetura da RNA aplicada ao problema de previsão de vazões.

Já o critério de parada adotado é o de validação cruzada, com 50% do conjunto de padrões para treinamento, 25% para validação cruzada e os outros 25% para teste. A rede tem seu treinamento interrompido quando o erro da base de validação cruzada aumentar 5 vezes seguidas, ou quando atingir o número máximo de ciclos.

Os pesos da rede são inicializados com valores no intervalo $[-1; 1]$ por um gerador aleatório com semente, dando a possibilidade do experimento ser reproduzido. Como a semente pode vir a influenciar nos resultados da rede, foram feitos testes com 10 sementes, todos os números primos com pelo menos três dígitos. Para cada uma delas, realizaram-se cerca de 10 simulações diferentes. Depois de todos os testes, chegou-se à conclusão de que a rede convergia bem com a semente 719 e, a partir desse momento, todas as simulações com RNAs utilizaram-se dela para inicialização dos pesos.

Embora a uso de RNAs na previsão diária das vazões em bacias hidrográficas brasileiras não seja uma novidade, o objetivo maior é a busca por melhores resultados por meio de novas abordagens, que posteriormente podem ser aplicadas a outras usinas. Uma delas partiu do desenvolvimento de um SNH.

3.4 Realizando previsões a partir de um SNH

O sistema híbrido proposto nesta seção utiliza curvas de tendência *Fuzzy* para previsão de vazões. A Figura 21 mostra sua arquitetura. As vazões (v_1 a v_{14}) são as entradas do sistema, que, quando recebidas, são subtraídas aos pares ($v_i - v_{i-1}$) de forma a obter a

diferença entre cada vazão e o seu anterior, denominado *valor diferença*. Posteriormente, cada *valor diferença* é fuzzificado quanto a um atributo *Fuzzy* com quatro classes, passando a ser representado por quatro valores de pertinência. Em seguida, os 52 valores de pertinência são recebidos como entrada da RNA, destacada por um retângulo vermelho. A RNA, então, processa os valores e retorna um conjunto de 28 saídas. Logo depois, em grupos de 4, as saídas são defuzzificadas, retornando 7 *valores diferença*. Por fim, da vazão v_{14} , são calculadas as sete vazões de saída do sistema.

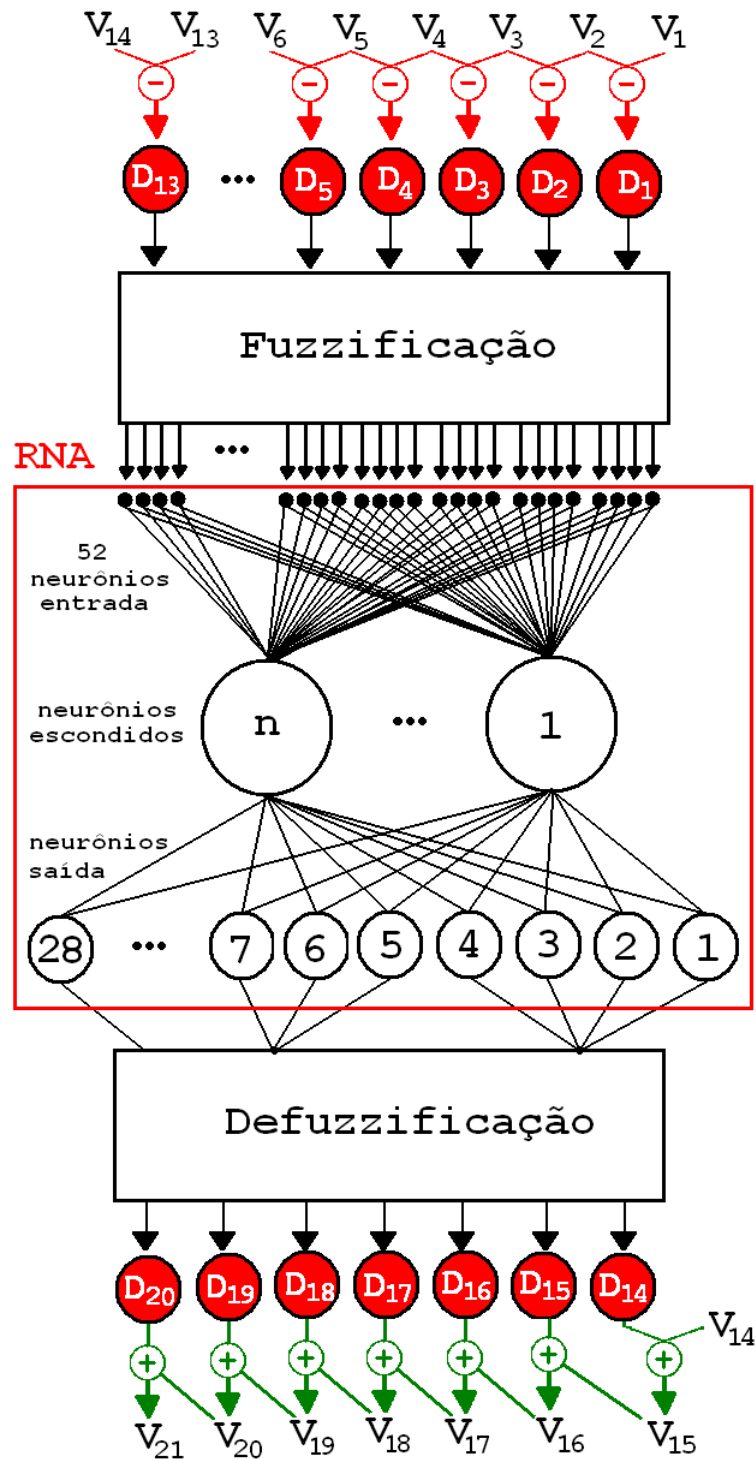


Figura 21: Modelo do SNH com Lógica *Fuzzy*, onde os V 's são vazões do sistema e n corresponde ao número de neurônios escondidos.

O objetivo do SNH é prever, com base nas 14 vazões de entrada, as sete seguintes. Os próximos tópicos discorrem sobre os módulos *Fuzzy*, o módulo neural e o método de treinamento e avaliação do sistema proposto.

3.4.1 Módulos *Fuzzy*

A incorporação de tendências de curva na RNA interna ao SNH proposto (Figura 21) é o principal objetivo dos módulos *Fuzzy*. Eles atuam antes (fuzzificação) e depois (defuzzificação) da participação da RNA no sistema.

O processo de fuzzificação é feito a partir de um *valor diferença*, que representa a intensidade de mudança entre uma vazão v_i e sua antecessora (v_{i-1}). A intenção é que a RNA absorva essa mudança de forma difusa. Dessa forma, é criado um atributo *Fuzzy* para representar cada *valor diferença* do sistema. O atributo é constituído de quatro classes, onde cada uma delas representa um tipo de curva, são elas:

1. *Crescimento acentuado*.
2. *Crescimento atenuado*.
3. *Decrescimento acentuado*.
4. *Decrescimento atenuado*.

Os quatro tipos de curvas podem ser vistos na Figura 22, onde o *decrecimento acentuado* está em vermelho, o *decrecimento atenuado* em azul, o *crescimento acentuado* em verde e, finalmente, o *crescimento atenuado* em laranja. A partir dessas curvas é possível aproximar qualquer função matemática. Cada uma das quatro classes é representada por uma função pertinência triangular.

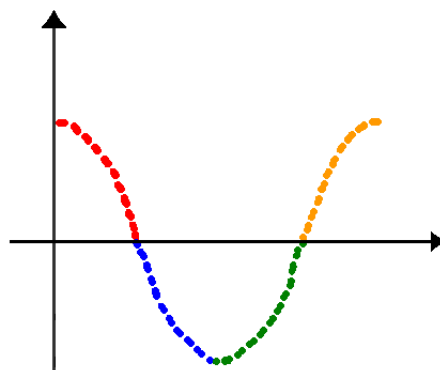


Figura 22: Curva com quatro comportamentos distintos: decrecimento acentuado (vermelho), decrecimento atenuado (azul), crescimento acentuado (verde) e crescimento atenuado (laranja).

Após o processo de fuzzificação, cada *valor diferença* é representado por quatro pertinências. Os 52 valores, retornados a partir dos 13 *valores-diferença*, são, então, encaminhados para a RNA, que retorna outras 28 pertinências, correspondentes a sete *valores diferença* de saída. A partir desse momento entra em ação o módulo de defuzzificação.

O método de defuzzificação tem por objetivo transformar os sete conjuntos de quatro pertinências em *valores-diferença* com um nível de precisão que não comprometa o desempenho do sistema. O método utilizado é o baseado no centro de gravidade, abordado na Seção 2.3.3.

Todo o processo realizado pelos modelos *Fuzzy* tem como objetivo maximizar o desempenho do "coração" do sistema, a RNA.

3.4.2 Módulo neural

A previsão do módulo neural é a parte mais importante do sistema. Espera-se que sua capacidade de generalização para o problema seja elevada, com a nova forma de representação dos dados.

O modelo é formado por uma RNA MLP constituída de 52 entradas e 28 saídas. A função de ativação de seus neurônios é a Sigmoidal Logística (Equação 3.2), e os pesos iniciais são atribuídos de forma aleatória, com valores no intervalo $[-1, 1]$.

Após ter visto o funcionamento dos módulos que compõem o SNH proposto, o passo seguinte é entender como se dá seu treinamento e avaliação de desempenho.

3.4.3 Método de treinamento e avaliação

O treinamento do sistema é, na realidade, o aprendizado da RNA interna. A cada conjunto de pertinências de entrada, a rede modifica seus pesos de forma a se aproximar ao máximo das pertinências de saída desejada. A rede utiliza o critério de parada de validação cruzada, mas o conjunto de testes não é utilizado para validar a rede, e sim o sistema híbrido como um todo.

Dessa forma, para cada entrada (14 vazões em sequência), o sistema dará a previsão das sete vazões subsequentes. A saída do sistema é, então, comparada com a saída desejada e o erro do sistema é calculado.

Depois do desenvolvimento do SNH, é proposto mais uma forma de melhorar as previsões obtidas a partir da RNA MLP. Ela se dá pela formação de um padrão de

entrada misto, constituído de 14 vazões consecutivas, mais o acréscimo de informações de tendência de curva. Tal abordagem foi denominada de RNA MLP com padrões de entrada mistos.

3.5 RNA MLP com padrões de entrada mistos

Na Seção 3.4 são utilizadas pertinências quanto a quatro classes, as quais representam quatro tipos de curvas. Na abordagem corrente, todo o processo de transformação, até chegar a um conjunto de padrões formados por pertinências, é repetido, mas, dessa vez, apenas para duas classes: *crescimento* e *decréscimento*. Feito isso, para cada padrão, é calculada a média das pertinências para os valores de entrada da rede, tanto para uma classe, quanto para a outra. Dessa forma, as duas pertinências médias representam a tendência de crescimento e decréscimento de todas as entradas de determinado padrão.

A Figura 23 mostra um exemplo do passo a passo na obtenção das pertinências médias de um padrão de entrada formado por sete vazões. Nele, depois de subtrair cada vazão com a imediatamente anterior, são obtidos os *valores-diferença*, no caso, variando entre -20 e 200 . Posteriormente, cada *valor-diferença* é fuzzificado quanto as classes *crescimento* (a direita) e *decréscimento* (a esquerda), passando a ser representado por duas pertinências. Por fim, é calculada a média de todas as pertinências quanto a cada uma das classes. O valor médio quanto às pertinências das classes *decréscimento* e *crescimento* é, nesse caso, respectivamente, $0,034$ e $0,34$.

Obtidas as pertinências médias, elas são introduzidas como mais duas entradas do padrão de treinamento da RNA. Com isso, os padrões utilizados na aprendizagem da rede seguem o formato dos exemplos exibido na Tabela 5.

Tabela 5: Exemplos de padrões formados por 16 entradas e sete saídas. As entradas são compostas por 14 vazões e duas pertinências, uma quanto à classe *decréscimento* ($P_{decréscimento}$) e outra quanto à classe *crescimento* ($P_{crescimento}$).

Entrada 1	...	Entrada 14	$P_{decréscimento}$	$P_{crescimento}$	Saída 1	...	Saída 7
2110	...	2310	0,14	0,4	2350	...	2450
2089	...	1991	0,45	0	1800	...	1750
2050	...	2450	0	0,6	2800	...	3000
2010	...	2071	0,14	0,35	2175	...	2100

Para que a busca pelas melhores predições com o novo conjunto de padrões seja mais precisa, é necessário definir o melhor atributo *Fuzzy* na determinação das pertinências médias, tanto para base de *Foz do Areia* quanto para de *Tucuruí*.

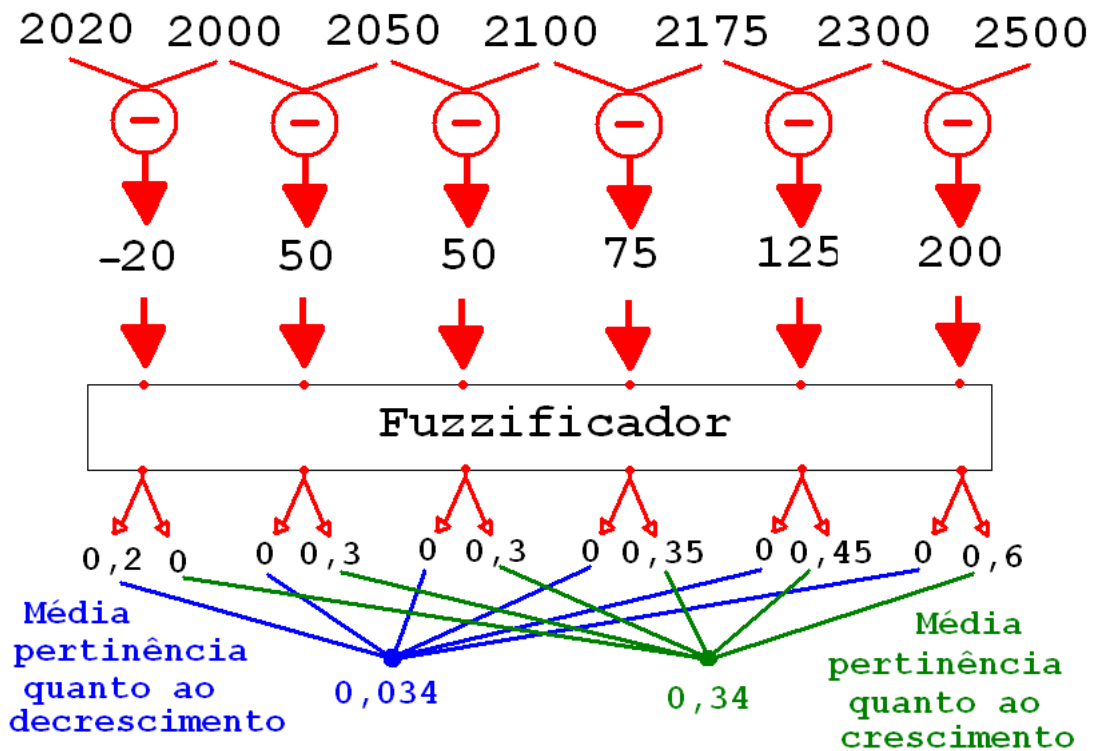


Figura 23: Passos percorridos por um conjunto de vazões de entrada de um determinado padrão até a obtenção de suas pertinências médias quanto às classes *Fuzzy crescimento* e *decrescimento*

3.5.1 Definição do atributo *Fuzzy*

As funções pertinências utilizadas no atributo *Fuzzy* são a triangular inicial e a triangular final, como mostra a Figura 24. Os parâmetros a_1 e a_2 (Figura 24) são os limites do espaço amostral de diferenças entre vazões vizinhas para cada base de dados. A Figura 24 também mostra que as classes não têm ponto de intersecção, o que é bastante aceitável, já que um ponto, numa curva, não pode estar crescendo e decrescendo ao mesmo tempo. Assim, os únicos parâmetros que faltam para formação do atributo *Fuzzy* são b_1 e b_2 (Figura 24).

A busca pelos parâmetros b_1 e b_2 (Figura 24) é feita com base no desempenho da RNA com diferentes configurações de tais variáveis. Dessa forma, a RNA tem todos os seus parâmetros de treinamento fixos, ao passo que dois tipos de configurações de b_1 e b_2 são testados.

No primeiro, a partir do ponto zero do eixo x são escolhidas diferentes constantes K , tal que $b_1 = -K$ e $b_2 = K$, onde $K < |a_1| \wedge K < |a_2|$ (Figura 24). Já no segundo,

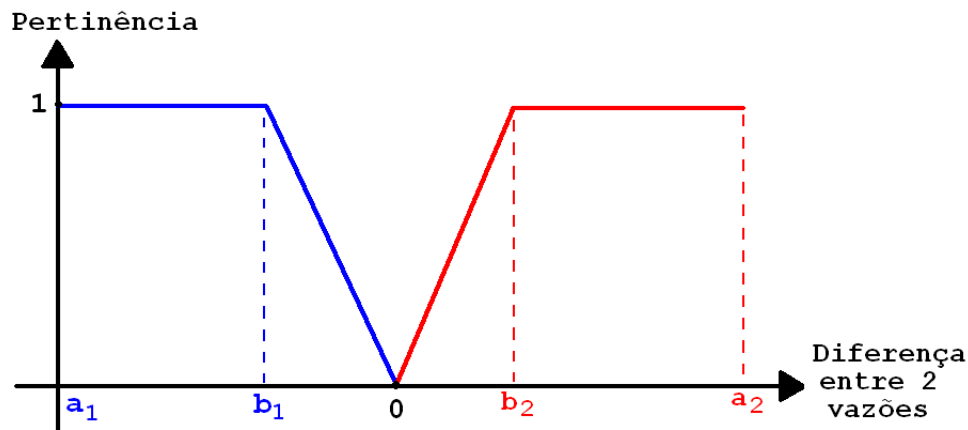


Figura 24: Classes *Fuzzy* para *decréscimo*, em azul e *crescimento*, em vermelho.

a distância em relação ao ponto zero é proporcional aos extremos, seguindo o formato $b_1 = Q \times a_1$ e $b_2 = (1 - Q) \times a_2$, onde $0 \leq Q \leq 1$ (Figura 24).

A abordagem utilizada nessa seção utiliza outra técnica de CI apenas para formar um padrão de entrada com mais informações. Já a próxima, lida com a divisão do conjunto de treinamento entre duas RNAs, formando uma MC.

3.6 Realizando previsões a partir de uma MC

A MC desenvolvida é formada por dois especialistas (RNA) e um módulo *Fuzzy*, que tem o papel de receber as saídas das duas redes e unir em uma única resposta, que é a saída do sistema (Figura 25).

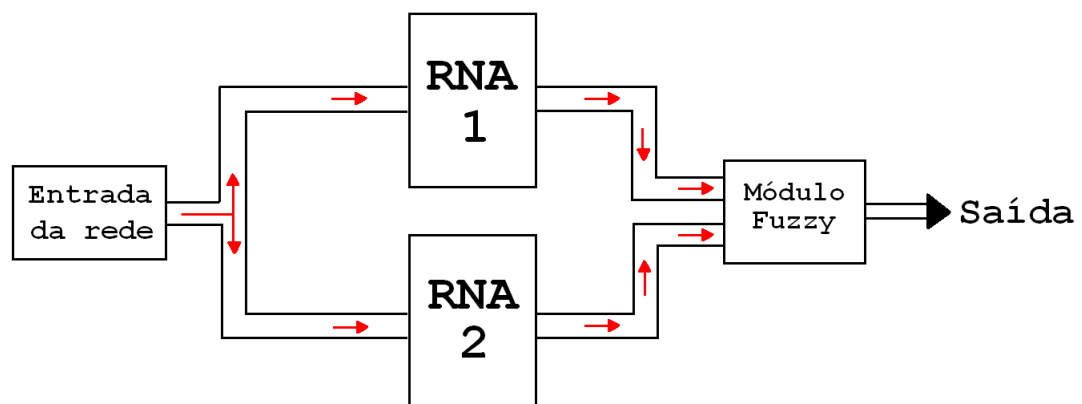


Figura 25: Arquitetura de uma MC com dois especialistas e um módulo de integração *Fuzzy*.

A ideia é que cada especialista seja qualificado em predizer apenas parte do universo amostral de entradas e que, com isso, passe a dar respostas mais precisas dentro do seu escopo de atuação. A formação dos especialistas acontece dentro do contexto de aprendizado supervisionado.

3.6.1 Aprendizado dos especialistas

A tarefa de aprendizado é dividida entre os especialistas como mostra a Figura 26. O primeiro especialista aprende a partir dos padrões que estiverem com todas as suas vazões entre a mínima e a delimitação em azul. Já o segundo, modifica seus pesos apenas com base nos padrões com vazões que estão entre a delimitação em vermelho e a vazão máxima.

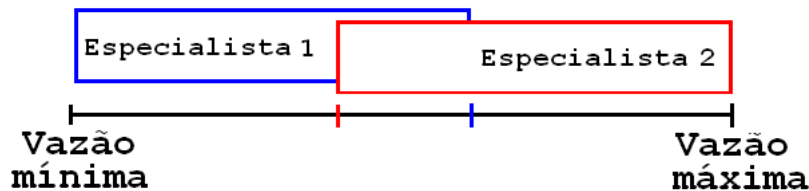


Figura 26: Áreas de atuação dos especialistas.

Ainda com relação à Figura 26, nota-se que existe uma área que os dois especialistas levam em consideração na fase de aprendizado, delimitada entre as marcas vermelha e azul. A forma como são aproveitadas as respostas dos especialistas, em especial, na região em que os dois aprendem, é definida pelo módulo *Fuzzy*.

3.6.2 Módulo Fuzzy

O módulo *Fuzzy* funciona de acordo com o atributo formado pelas classes *especialista 1* e *especialista 2*, como pode ser visto na Figura 27. Dessa forma, o valor de saída é dado pela Equação 3.3, onde $vazao_1$ e $vazao_2$ são, respectivamente, as vazões retornadas pela RNA_1 e RNA_2 (Figura 26) e, μ_1 e μ_2 , as pertinências associadas às classes *especialista 1* e *especialista 2*.

$$\frac{\mu_1 \times vazao_1 + \mu_2 \times vazao_2}{\mu_1 + \mu_2} \quad (3.3)$$

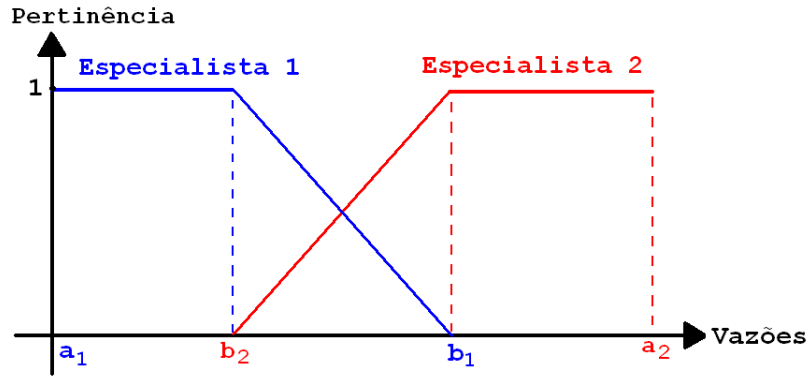


Figura 27: Atributo utilizado no módulo *Fuzzy* da MC.

Uma dificuldade encontrada nesse tipo de abordagem é definir para que valor de vazão as pertinências μ_1 e μ_2 serão calculadas. Para tal, são utilizadas três abordagens:

1. **Direto:** μ_1 e μ_2 são as pertinências das classes *especialista 1* e *especialista 2*, em relação aos valores retornados pela RNA_1 e RNA_2 , respectivamente.
2. **Médio:** μ_1 e μ_2 são as pertinências relacionadas às classes *especialista 1* e *especialista 2*, respectivamente, com relação a médias entre as vazões advindas das RNAs.
3. **Maior pertinência:** μ_1 e μ_2 são as maiores pertinências relacionadas às classes *especialista 1* e *especialista 2*, para os valores retornados pelas RNAs do sistema.

Os parâmetros do atributo *Fuzzy* são definidos para cada uma das bases de dados utilizadas nas simulações. As variáveis a_1 e a_2 (Figura 27) recebem, respectivamente, a menor e maior vazão presente na base de dados correspondente. Já os valores de b_1 e b_2 (Figura 27) são obtidos a partir de experimentos, onde os parâmetros de simulação das RNAs da MC são fixados, enquanto são testadas diferentes combinações entre b_1 e b_2 . Os valores atribuídos são escolhidos com base na média, mediana e valores extremos das vazões presentes em cada uma das bases de dados.

Após a realização de predições através de uma MC formada por dois especialistas e um módulo de integração, a próxima e última seção desse capítulo discorrem sobre uma forma de seleção de parâmetros de entrada de uma RNA de forma automática.

3.7 Método de seleção de variáveis

A previsão de vazões com RNAs MLP, na Seção 3.3, utiliza-se de padrões de entrada com 14 vazões. Essa escolha foi feita por um especialista. No entanto, existem diversas

técnicas inteligentes de busca que podem ser aplicadas na realização dessa escolha. Esta seção se destina a utilizar BH para tal e comparar o desempenho das redes treinadas pelos padrões de entrada escolhidos pela técnica e pelo especialista.

Na aplicação proposta, uma harmonia é formada por 30 notas. Cada nota é representada pela escolha ou não de uma vazão na formação do padrão de entrada da rede. As vazões escolhidas não precisam ser consecutivas, podendo, por exemplo, ser as de índices 1 e 30. A qualificação da harmonia gerada, o *fitness*, é o erro da rede quando submetida ao treino com o padrão de entrada escolhido pelo algoritmo. Assim, quanto menor for o seu valor, melhor.

Dessa forma, para cada nova harmonia criada, uma RNA é treinada. O aprendizado de uma MLP, na aplicação de predição de vazões, gasta, em média 200 minutos, dependendo do número máximo de ciclos passado como parâmetro. Se a BH utilizar 1000 ciclos, por exemplo, acarretaria em prováveis 1000 novas simulações com a RNA, totalizando, aproximadamente 138 dias de processamento.

Com isso, para que a aplicação da BH se torne viável, o número de ciclos máximos da RNA geradora do *fitness* é reduzido bruscamente, procurando ser suficiente apenas para diferenciar o desempenho entre as harmonias geradas pelo sistema.

O real desempenho da escolha automatizada é testada após a execução da BH. A partir da escolha do padrão de entrada por parte da BH, é realizado o mesmo conjunto de simulações feito com o padrão escolhido pelo especialista (Seção 3.3). A partir daí o desempenho é comparado para as bases de dados de *Foz do Areia* e *Tucuruí*.

Vale observar que o conteúdo abordado nesta seção não está dentro do escopo inicial do projeto. A aplicação é feita vislumbrando a comparação com uma nova abordagem de seleção de variáveis de entrada. O novo algoritmo é diretamente voltado para problemas envolvendo séries temporais e ainda está em fase de desenvolvimento.

4 *Resultados*

Este capítulo mostra os resultados obtidos a partir da aplicação da metodologia explicada no Capítulo 3. Os valores escolhidos para simulação foram baseados em algumas simulações prévias, que determinaram a faixa de valores que permitiram as técnicas convergirem. No caso de experimento com RNA, a experiência em projetos anteriores também ajudou.

As avaliações das técnicas aplicadas ao problema se basearam no chamado Erro Probabilístico Médio Absoluto (EPMA), dado pela Equação 4.1, onde N = número de previsões realizadas, d_i = saída desejada e y_i = saída prevista para a i -ésima previsão.

$$EMPA = \frac{\sum_{i=1}^N \left| \frac{d_i - y_i}{d_i} \right|}{N} \quad (4.1)$$

A Seção 4.1 mostra os resultados obtidos aplicando RNAs junto com os conseguidos pelos métodos lineares vigentes na maioria das usinas hidroelétricas brasileiras. Já a Seção 4.2, exhibe os resultados trazidos pelo SNH. Em seguida, a Seção 4.3 discorre sobre o que foi obtido a partir dos padrões de entrada misto. Posteriormente, na Seção 4.4, pode ser vista a precisão atingida pela MC desenvolvida e, por fim, a Seção 4.5 analisa os resultados advindos da técnica de BH na escolha dos padrões de entrada da rede.

4.1 Resultados da RNA MLP

Os parâmetros variados durante as simulações foram: taxa de aprendizado, *momentum*, número de neurônios escondidos e o número máximo de ciclos. Foram escolhidas três taxas de aprendizado e duas de *momentum*, que, combinados, somam seis simulações distintas. Para cada um desses conjuntos, foram testados cinco diferentes quantidades de neurônios escondidos, dando um total de 30 simulações. O número máximo de ciclos foi escolhido de acordo com a taxa de aprendizado; quanto menor ela, maior a quantidade de ciclos (Tabela 6). A Tabela 7 mostra os valores dos parâmetros descritos.

Tabela 6: Associação entre os parâmetros taxa de aprendizado e número máximo de ciclos para o experimento com RNA MLP.

Taxa de aprendizado	Número máximo de ciclos
0,3	10000
0,1	30000
0,01	100000

Tabela 7: Valores dos parâmetros variados no treinamento da RNA.

Parâmetro	Valores
Taxa de aprendizado	0,3; 0,1 e 0,01
<i>Momentum</i>	0,3 e 0,1
Neurônios escondidos	6, 8, 10, 14 e 16

As configurações que trouxeram melhores resultados para as bases de *Foz do Areia* de *Tucuruí*, são mostradas na Tabela 8. Os menores EPMA obtidos pelas RNAs para as duas bases de dados, bem como os obtidos pelos métodos utilizados pela ONS podem ser vistos na Tabela 9. O EPMA obtido pelos métodos utilizados pela ONS para *Foz do Areia* foram baseados no ano de 2001, ao passo que *Tucuruí* considerou os dados entre 1997 e 2004 [31]. Já os erros advindos da RNA considerou os anos entre 1969 e 2001 para *Foz do Areia* e entre 1968 e 2001 para *Tucuruí*.

Tabela 8: Valores dos parâmetros de treinamento da RNA MLP que trouxeram melhores resultados para as bases de dados de *Foz do Areia* e *Tucuruí*.

	Neurônios escondidos	Aprendizado	<i>Momentum</i>	Ciclos
<i>Foz do Areia</i>	14	0,1	0,1	10000
<i>Tucuruí</i>	16	0,01	0,1	100000

Tabela 9: EPMA para primeira semana de predição por RNAs e pelos padrões vigentes na ONS.

	ONS	RNAs
<i>Foz do Areia</i>	31,5%	18,67%
<i>Tucuruí</i>	12,13%	4,58%

4.2 Resultados do SNH

A variação dos parâmetros de simulação do SNH pode ser vista na Tabela 10. Nota-se uma grande variação no número de neurônios escondidos, que foi decorrente de uma busca em amplitude pela estabilização do sistema, o que não ocorreu. Ainda foram feitos testes

com várias configurações de função de pertinência, mas o resultado não mudou. O EPMA do sistema girou em torno de 200%.

Tabela 10: Parametrização da RNA MLP utilizada dentro do SNH proposto

Parâmetro	Valores
Número de entradas	52
Número de saídas	28
Número de neurônios escondidos	Entre 6 e 60
Função de ativação	Sigmoidal Logística
Inicialização dos pesos	Aleatórios entre -1 e 1
Taxa de aprendizado	0,3; 0,1 e 0,01
<i>Momentum</i>	0,3 e 0,1
Número máximo de ciclos	Entre 10000 e 100000

4.3 Resultado da RNA MLP com padrões de entrada misto

A formatação do atributo *Fuzzy*, segundo a Figura 24, foi obtida a partir de 10 testes para cada base de dados, com os parâmetros da RNA fixados segundo a Tabela 11, como explicado na Seção 3.5.1. A melhor configuração para ambas as bases está presente na Tabela 12.

Tabela 11: Parâmetros para simulação da RNA.

Parâmetro	Valor
Percentuais dos conjuntos de treinamento/validação cruzada/teste	50% / 25% / 25%
Número de entradas	14
Número de saídas	7
Número de neurônios escondidos	10
Taxa de aprendizado	0,3
<i>Momentum</i>	0,2
Número máximo de ciclos	10000

Tabela 12: Valores dos parâmetros que compõem as classes *Fuzzy Crescimento* e *Decrescimento* de acordo com a Figura 24, para as bases *Foz do Areia* e *Tucuruí*.

Base	a_1	a_2	b_1	b_2
<i>Foz do Areia</i>	-739	2051	$0,2 \times a_1$	$0,8 \times a_2$
<i>Tucuruí</i>	-3165	3896	$0,2 \times a_1$	$0,8 \times a_2$

Depois de atribuídos os parâmetros que formam o atributo *Fuzzy* e de gerar os padrões de treinamento com entrada mista, segundo visto na Seção 3.5, iniciaram-se as simulações com a RNA MLP. Foram feitas 30 simulações para cada uma das bases, totalizando 60. Os valores dos parâmetros testados estão resumidos na Tabela 13.

Tabela 13: Parametrização da RNA MLP utilizada dentro do SNH proposto

Parâmetro	Valores
Número de entradas	14
Número de saídas	7
Número de neurônios escondidos	8, 10, 12, 14 e 16
Função de ativação	Sigmoidal Logística
Inicialização dos pesos	Aleatórios entre -1 e 1
Taxa de aprendizado	0,3; 0,1 e 0,01
<i>Momentum</i>	0,3 e 0,1
Número máximo de ciclos	Entre 10000 e 100000

Os melhores resultados obtidos, bem como a correspondente parametrização estão apresentadas na Tabela 14, tanto para base do *Foz do Areia*, quando para *Tucuruí*.

Tabela 14: Melhores resultados obtidos e correspondente parametrização da MLP com padrão misto para as bases do *Foz do Areia* e de *Tucuruí*

	Escondidos	Taxa de aprendizado	<i>Momentum</i>	Número de Ciclos	EPMA
<i>Foz do Areia</i>	16	0,3	0,1	3668	18,15
<i>Tucuruí</i>	16	0,1	0,1	100000	5,17

4.4 Resultado da MC

A configuração do módulo *Fuzzy*, segundo a Figura 27, foi obtida a partir de 11 simulações para cada base de dados, tendo sido os parâmetros da RNA MLP fixos segundo a Tabela 15, de acordo com a Seção 3.6.2. A parametrização que trouxe os melhores resultados pode ser vista na Tabela 16, onde $M_{FozdoAreia}$ e $M_{Tucuruí}$ correspondem, respectivamente, ao valor médio de vazão nas usinas de *Foz do Areia* e *Tucuruí*.

Tabela 15: Parâmetros para simulação da RNA.

Parâmetro	Valor
Percentuais dos conjuntos de treinamento/validação cruzada/teste	50% / 25% / 25%
Número de entradas	14
Número de saídas	7
Número de neurônios escondidos	10
Taxa de aprendizado	0,3
<i>Momentum</i>	0,2
Número máximo de ciclos	10000

Tabela 16: Valores dos parâmetros que compõem o atributo *Fuzzy* definido na Figura 27.

Base	a_1	a_2	b_1	b_2
<i>Foz do Areia</i>	68	7830	$1,1 \times M_{FozdoAreia}$	$0,9 \times M_{FozdoAreia}$
<i>Tucuruí</i>	1374	64931	$1,3 \times M_{Tucuruí}$	$0,7 \times M_{Tucuruí}$

Com os módulos *Fuzzy* definidos para cada base de dados, foi dado início às simulações com a MC. Os parâmetros utilizados nas duas RNA foram os mesmos, e seus valores estão definidos na Tabela 17. Foram realizadas 42 simulações com cada uma das bases de dados. As predições mais precisas, assim como a respectiva parametrização, estão apresentadas na Tabela 18, onde o **EPMA Médio** indica que o cálculo da pertinência no módulo *Fuzzy* foi feito usando a abordagem *Direta* (Seção 3.6.2).

Tabela 17: Parametrização da RNA com padrões de treinamento com redundância *Fuzzy*.

Parâmetro	Valor
Percentuais conjuntos: treinamento/validação cruzada/teste	50%/25%/25%
Número de entradas	14
Número de saídas	7
Número de neurônios escondidos	6, 7, 8, 9, 10, 12 e 14
Taxa de aprendizado	0,3; 0,1 e 0,01
<i>Momentum</i>	0,3 e 0,1
Número máximo de ciclos	Entre 30000 e 100000

Tabela 18: Predições mais precisas e respectiva parametrização da MC para as bases de dados de *Foz do Areia* e *Tucuruí*.

	Escondidos	Taxa de aprendizado	<i>Momentum</i>	Número de Ciclos	EPMA Direto
<i>Foz do Areia</i>	9	0,3	0,1	30000	17,48
<i>Tucuruí</i>	10	0,1	0,3	50000	4,69

4.5 Resultado da BH

O algoritmo de BH aplicado na escolha das variáveis de entrada da RNA, segundo a Seção 3.7, foi executado apenas 9 vezes, devido ao seu alto custo computacional, em torno de 8 horas por simulação. A parametrização considerou algumas combinações entre os valores descritos na Tabela 19.

Já a parametrização fixa da RNA, responsável por gerar o *fitness*, pode ser vista na Tabela 20. Vale observar que o número de entradas varia, pois é peculiar a harmonia. Outra observação é o valor dado ao número máximo de simulações, que foi escolhido de forma a viabilizar os experimentos.

Tabela 19: Parametrização da BH.

Parâmetro	Valor
Memória harmônica	10, 20 e 40
Número de ciclos	1000 e 2000
Tamanho harmonia	30
Probabilidade de escolha	50%

Tabela 20: Parametrização da RNA utilizada como geradora de *fitness* para a BH.

Parâmetro	Valor
Percentuais conjuntos: treinamento/validação cruzada/teste	50%/25%/25%
Número de entradas	Varia com a harmonia
Número de saídas	7
Número de neurônios escondidos	10
Taxa de aprendizado	0,3;
<i>Momentum</i>	0,2
Número máximo de ciclos	200

As entradas escolhidas pela BH para as bases de dados de *Foz do Areia* e *Tucuruí* estão exibidas na Tabela 21.

Tabela 21: Padrões de entrada encolhidos pela BH.

Base	Índices escolhidos como entrada (1 a 30)
<i>Foz do Areia</i>	1,2,4,5,7,8,13,14,15,16,17,18,19,22,23,24,25,29 e 30
<i>Tucuruí</i>	2,7,14,15,22,27,28,29 e 30

Depois de coletar as entradas pela BH, foi executado o mesmo conjunto de simulações usado na Seção 4.1, correspondente à escolha do padrão de entrada por parte do especialista. A Tabela 22 mostra os melhores resultados obtidos para a base de *Foz do Areia* e *Tucuruí*.

Tabela 22: Melhores resultados obtidos e correspondente parametrização da MLP com padrão de entrada escolhido pelo algoritmo de BH, para as bases do *Foz do Areia* e de *Tucuruí*

	Escondidos	Taxa de aprendizado	<i>Momentum</i>	Número de Ciclos	EPMA
<i>Foz do Areia</i>	6	0,3	0,3	10000	20,00
<i>Tucuruí</i>	6	0,01	0,3	60000	4,32

As Figuras 28 e 29 mostram o desempenho da escolha do especialista junto com a da abordagem corrente, para as bases de dados do *Foz do Areia* e de *Tucuruí*, respectivamente. No eixo *x* encontram-se 30 diferentes simulações e no eixo *y* suas respectivas taxas de EPMA.

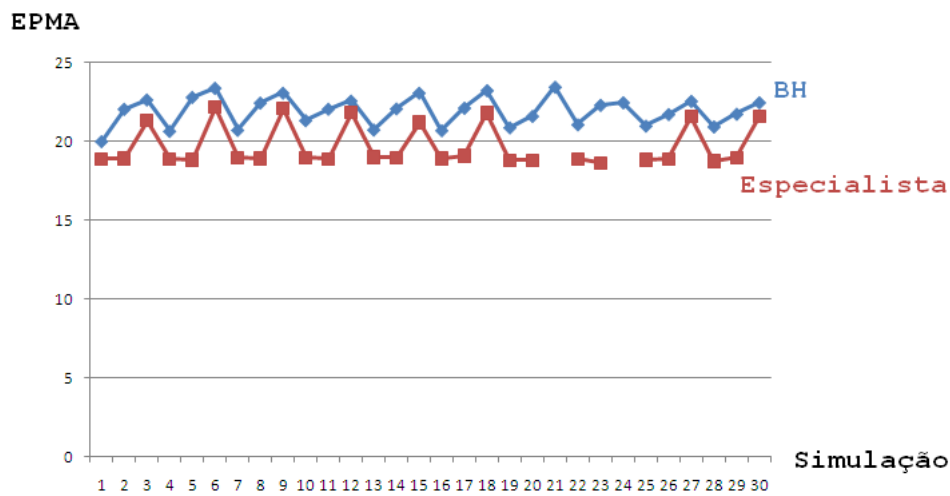


Figura 28: Resultados do EPMA para 30 simulações distintas, com a escolha do padrão de entrada por parte do Especialista (em vermelho), e pela BH (em azul), para a base de dados do *Foz do Areia*.

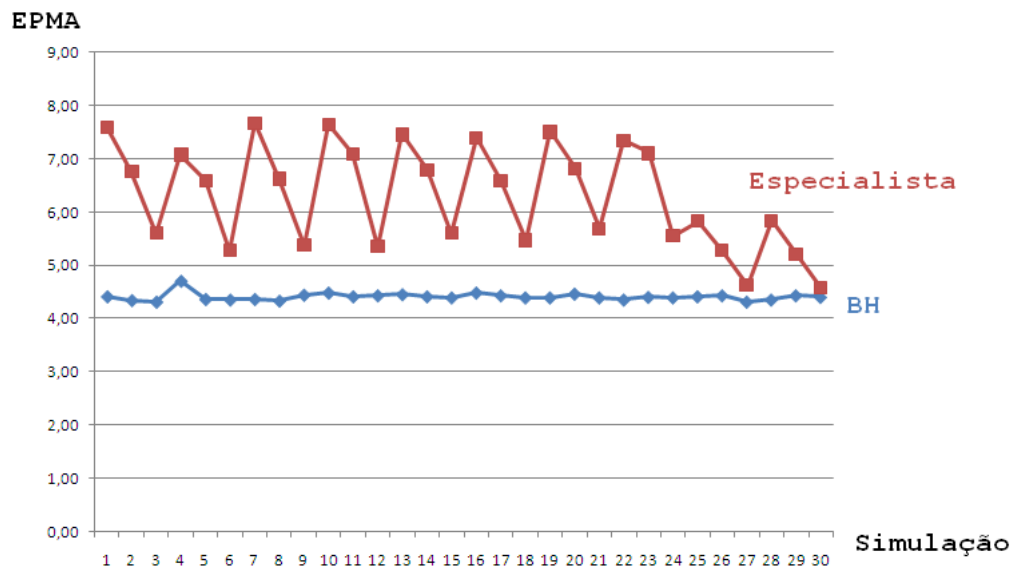


Figura 29: Resultados do EPMA para 30 simulações distintas, com a escolha do padrão de entrada por parte do Especialista (em vermelho), e pela BH (em azul), para a base de dados de *Tucuruí*.

5 Conclusão

O principal objetivo desta monografia foi melhorar a precisão das previsões de vazões diárias para sete dias, fortemente utilizada no PMO. Foi mostrado que a ONS utiliza métodos lineares e estatísticos para tal, e que o uso de RANs tem conseguido melhores resultados.

Dessa forma, foi implementada uma RNA MLP, que se utilizou do algoritmo de treinamento *Backpropagation* e aplicou-a ao problema de previsão de vazões diárias. Foram utilizadas duas bases de dados, uma com sazonalidade (usina de *Tucuruí*) e outra sem (usina do *Foz do Areia*). Os resultados obtidos nessa etapa foram uma referência para as etapas seguintes, que procuraram melhorá-los.

Então, a partir de técnicas de CI, foram desenvolvidas quatro novas formas de realizar predições diárias. A primeira delas se deu pela construção de um SNH, que introduziu tendências de curva fuzzificadas na RAN. Já a segunda, formou um padrão de treinamento que considerava vazões anteriores e tendências de curvas, denominado de padrão misto. A terceira foi pela implantação de uma MC com dois especialistas. Por fim, na quarta aplicaram-se BH na seleção das variáveis de entrada da RNA.

Dentre as técnicas aplicadas, o SNH foi a única que não convergiu, obtendo taxas de EPMA em torno de 200%. Todos os outros métodos de predição, desde os utilizados pela ONS, até os recentemente desenvolvidos nesse projeto, têm seus melhores resultados, para cada base de dados, resumidos na Figura 30, onde no eixo x se encontram as técnicas, em ordem de desenvolvimento, e no eixo y o respectivo EPMA.

Como pode ser visto, o grande declínio do EPMA se dá entre os padrões utilizados pela ONS e a técnica de RNA. A partir de então, o erro, de uma forma geral, se estabiliza, tendo apenas algumas poucas melhoras em relação à RNA, como no caso de RNA Mista e MC, para a base de dados do *Foz do Areia*, e na BH, para a base de *Tucuruí*.

No entanto, as melhoras obtidas com as novas abordagens não justificam seu uso nesse contexto. A primeira razão para tal conclusão é o aumento do custo computacional em

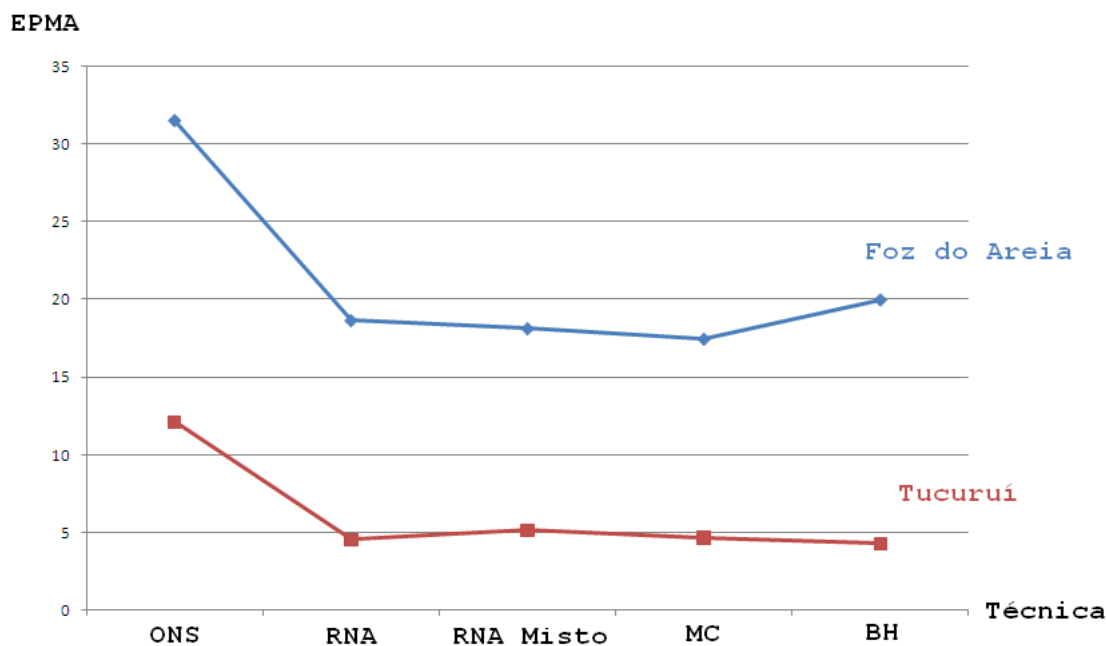


Figura 30: Comparativo dos resultados entre os métodos de predição: ONS, RNA, RNA Misto, MC e BH.

relação à RNA. A segunda é que, exceto o caso da BH, as melhoras foram obtidas em casos pontuais. Isso leva a crer que, ao fazer novas simulações, esse quadro pode se inverter. Diferente da BH, pois em todos os casos ela foi mais precisa que a RNA escolhida pelo especialista, para a base de *Tucuruí* e menos precisa para a base do *Foz do Areia*.

Entretanto, os métodos aplicados ainda não podem ser descartados. A não melhora dos resultados pode estar ligada às características do problema. Apenas quando testado em outros problemas será possível ter uma avaliação mais precisa. Vale lembrar que a maioria deles obteve alguma melhora.

Quanto ao confronto entre o padrão de entrada escolhido pelo especialista e o obtido por BH, as Figuras 29 e 28 deixam claro que não existe um vencedor, já que o primeiro obteve melhores resultados para base de *Foz do Areia* e o segundo para *Tucuruí*. No entanto, deixa claro que o uso de métodos de seleção de variáveis automáticos, no contexto de RNA, merece ser mais investigado.

A maior dificuldade encontrada durante as atividades de pesquisa foi o tempo de simulação. Como a base de dados usada e o número de ciclos exigidos para convergência dos algoritmos nesse contexto eram muito grandes, uma simulação corriqueira durava horas. Analisando alguns dos tempos de simulação colhidos, estima-se que foram gastos pelo menos 25 dias de simulação, caso tivessem sido feitas em apenas um computador

com processador *dual core* e razoável memória principal.

Outra dificuldade que vale a pena registrar foi a de implementação, pois a criação de alguns dos modelos exigiam algumas modificações dentro das técnicas, impedindo o uso de ferramentas. Por outro lado, do ponto de vista de aprendizado das técnicas, foi bastante proveitoso.

Como trabalhos futuros, pode-se testar os modelos RNA Mista, MC e BH em outros tipos de problemas que já utilizaram RNA, para saber se eles se adequariam. Outra possibilidade é investigar mais a fundo algumas técnicas de seleção de variáveis de entrada, tais como BH, dentro do contexto de séries temporais. Por fim, as possibilidades de desenvolvimento de técnicas de SNH e MC são as mais diversas possíveis e também merecem ser mais investigadas, pois muitas delas já trouxeram bons resultados.

Referências

- [1] COSTA, F. da S.; PIÑEIRO, M. E.; DAMÁZIO, J. M. Modelos de previsão hidrológica aplicados ao planejamento da operação do sistema elétrico brasileiro. *Revista Brasileira de Recursos Hídricos*, v. 12, p. 21–30, 2007.
- [2] ENERGÉTICO, S. de Planejamento e D. *Matriz Energética Nacional 2030*. 2007. [Http://www.mme.gov.br](http://www.mme.gov.br). Último Acesso: 20/10/2009.
- [3] MACEIRA, M. E. P. Chain of optimization models for setting the energy dispatch and spot price in the brazilian system. *XIV Power Systems Computation Conference*, Sevilla, Espanha, 2002.
- [4] BRAGA, A. de P.; CARVALHO, A. P. de Leon Filho de; LUDERMIR, T. B. *Redes Neurais Artificiais: Teoria e Aplicações*. Rio de Janeiro: LTC, 2000.
- [5] EIBEN, A. E.; SMIT, J. E. *Introduction to Evolutionary Computing*. New York: Springer, 2003.
- [6] VALENÇA, M. J. S.; LUDERMIR, T. B. The new model to forecast average monthly inflow. *VII Symposium on Neural Networks (SBRN)*, Porto de Galinhas, Brasil, v. 1, p. 74–79, 2002.
- [7] GALVÃO, C. de O. et al. *Sistemas Inteligentes: Aplicações a Recursos Hídricos e Ciências Ambientais*. Porto Alegre: Universidade, 1999.
- [8] VALENÇA, M. J. S. Uma nova metodologia para previsão de vazões médias mensais. *VII Simpósio de Recursos Hídricos do Nordeste*, São Luiz, Brasil, 2004.
- [9] ALBANO, G. D.; VALENÇA, M. J. S.; PORTO, R. L. L. Previsão de vazões médias mensais, afluentes com o uso de redes neurais construtivas, alternativa de planejamento operacional de represas. *XVII Simpósio Brasileiro de Recursos Hídricos*, São Paulo, Brasil, 2007.
- [10] VALENÇA, M. J. S.; VALENÇA, I. C. B. Uma rede gmdh com algoritmo multi-placativo aditivo e neurônio ativo para previsão de vazões. *VIII Simpósio de Recursos Hídricos do Nordeste*, 2006.
- [11] VALENÇA, M. J. S.; LUDERMIR, T. B. Neurodiário um modelo multivariado para previsão de vazões médias diárias. *Simpósio Brasileiro de Recursos Hídricos*, João Pessoa, Brasil, 2005.
- [12] VALENÇA, M. J. S. Sistema híbrido para previsão de vazões incrementais e afluências ao reservatório de sobradinho. *I Simpósio de Recursos Hídricos do Sul-Sudeste*, 2006.

- [13] KASABOV, N. K. *Foundation of Neural Networks, Furry System, and Knowledge Engineering*. Londres: A Bradford Book, 1996.
- [14] MACEIRA, M. E. P.; FALCÃO, D. M.; PENNA, D. D. J. Chain of optimization models for setting the energy dispatch and spot price in the brazilian system. *IEEE Porto Power Tech Proceedings*, Porto, Portugal, v. 3, n. 6, 2001.
- [15] COSTA, F. da S. et al. Modelagem estocástica de previsão de vazões mensais: Previvazm. *XV Simpósio Brasileiro de Recursos Hídricos*, Curitiba, Brasil, 2003.
- [16] COSTA, F. da S.; DAMÁZIO, J. M.; CARVALHO, A. L. de. Modelo estocástico de previsão de vazões diárias: Previvazh. *Seminário Internacional Sobre Hidrologia Operativa*, Concordia, Argentina, 2000.
- [17] BERNE; LEVY. *Fisiologia*. Porto Alegre: Artmed, 2003.
- [18] MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin os Mathematical Biophysics*, p. 115–133, 1943.
- [19] VALENÇA, M. J. S. *Fundamentos das Redes Neurais: Exemplos em Java*. Recife: Livro Rápido, 2007.
- [20] HAYKIN, S. *Redes Neurais: Princípios e Práticas*. Recife: Bookman, 2007.
- [21] VALENÇA, M. J. S. *Aplicando Redes Neurais: Um Guia Completo*. Recife: Livro Rápido, 2005.
- [22] WIDROW, B.; HOFF, M. A logical calculus of the ideas immanent in nervous activity. *Bulletin os Mathematical Biophysics*, p. 115–133, 1943.
- [23] ZADEH, L. Fuzzy algorithms. *Elsevier Inc*, p. 94–102, 1968.
- [24] ZIMMERMANN, H. J. *Fuzzy set theory and its applications*. Norwell: Kluwer Academic Publishers, 1996.
- [25] GREEM, Z. W.; KIM, J. H.; LOGANATHAN, G. V. A new heuristic optimization algorithm: Harmony search. *Society for Modeling and Limulation International (SCS)*, p. 60–68, 2001.
- [26] OBITKO, M. *Introdução aos Algoritmos Genéticos*. [Http://www.professor.webizu.org/ga/](http://www.professor.webizu.org/ga/). Último Acesso: 05/11/2009.
- [27] OVASKA, S. J. *Computationally Intelligent Hybrid Systems: The Fusion of Soft Computing and Hard Computing*. [S.l.]: Wiley-IEEE Press, 2004. 410 p.
- [28] GUO, J.-J.; LUH, P. B. Improving market clearing price prediction by using a committee machine of neural networks. *IEEE Transactions on Power Sytems*, v. 19, p. 1867– 1876, 2005.
- [29] SHI, M.; BERMAK, A. Committee machine with over 95% classification accuracy for combustible gas identifications. *Electronics, Circuits and Systems, 2006. ICECS '06. 13th IEEE International Conference on*, p. 862–865, 2006.

-
- [30] PACHECO, D. F. *Suporte à Decisão em Sistemas Inteligentes de Colheitas Agrícolas*. Dissertação (Monografia) — Escola Politécnica de Pernambuco, 2006.
- [31] LEITE, K. T. F.; MENDES, M. H. Modelo de previsão de vazão com informação de precipitação utilizando redes neurais. *Workshop Previsão de Vazões*, Rio de Janeiro, Brasil, 2007.