



AUTARQUIA ASSOCIADA À UNIVERSIDADE  
DE SÃO PAULO

**UTILIZAÇÃO DE REDES NEURAIS NA MONITORAÇÃO  
DA POTÊNCIA DO REATOR IEA-R1**

**ROSA HELENA PECCININI SILVA ROSSI**

**Dissertação apresentada como parte  
dos requisitos para obtenção do Grau de  
Mestre em Ciências na Área de  
Tecnologia Nuclear - Reatores.**

**Orientador:  
Dr. Adalberto José Soares**

**São Paulo  
2001**

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES  
Autarquia associada à Universidade de São Paulo

UTILIZAÇÃO DE REDES NEURAS NA MONITORAÇÃO DA POTÊNCIA DO  
REATOR IEA-R1

ROSA HELENA PECCININI SILVA ROSSI



Dissertação apresentada como parte dos requisitos para  
obtenção do Grau de Mestre na Área de Tecnologia  
Nuclear - Reatores

Orientador:  
Dr. Adalberto José Soares

São Paulo  
2001

Dedico este trabalho aos meus avós, exemplos de vida meus eternos incentivadores.

Aos meus pais, que me ensinaram princípios de perseverança, honestidade e trabalho.

Às minhas filhas Mariana e Juliana, razão da minha existência, ao meu esposo Raimundo pela constante dedicação, apoio e incentivo e à Estrelinha.

Aos meus tios, familiares e amigos que sempre me apoiaram e me incentivaram em todos momentos da minha vida.

E principalmente a Deus, pois sem Ele nada seria possível.

## *AGRADECIMENTOS*

Ao meu Orientador e amigo, Prof. Dr. Adalberto José Soares, pelas diretrizes sempre objetivas, transmitindo e compartilhando seus conhecimentos e pelo paciente acompanhamento deste trabalho, o meu muito obrigada.

Ao CNPq cujos recursos possibilitaram o desenvolvimento deste trabalho.

À toda equipe de operação do reator IEA-R1 pela capacidade e disposição na aquisição de dados operacionais.

Aos colegas Roberto Frajndlich, Carlos Loyola, Patrício Cárdenas, Icimone Braga, Sergio Perillo, Marcos Carvalho, Franco Brancaccio, Benedito Baptista, Algeny Leite e Hélio Massaki pelo empréstimo de material didático e pelas palavras de incentivo.

À todos meus amigos e colegas do IPEN, que colaboraram direta ou indiretamente para este trabalho.

Ao meu irmão Alejandro pela sua dedicação e amizade nesta jornada.

Aos amigos da CPG pelo apoio recebido durante todo o Mestrado.

Aos meus amigos da Escola Politécnica da USP Mariana Aguiar, Rosemeire Jerônimo, Pedro Timoszczuk e a todos da Poli que também colaboraram direta ou indiretamente para este trabalho.

# UTILIZAÇÃO DE REDES NEURAIIS NA MONITORAÇÃO DA POTÊNCIA DO REATOR IEA-R1

**Rosa Helena Peccinini Silva Rossi**

## **Resumo**

Após o acidente ocorrido na central nuclear de Three Mile Island, engenheiros e pesquisadores da área de instrumentação e controle começaram a analisar a praticidade e funcionamento das salas de controle convencionais. Observaram que as mesmas, apresentavam uma grande quantidade de informações, porém não seletivas. Do ponto de vista operacional, todas as condições anormais tinham a mesma prioridade, ficando a cargo do supervisor a responsabilidade de coordenar as ações conforme a sua experiência, o que, como ficou demonstrado no caso de Three Mile, nem sempre era suficiente para evitar consequências indesejadas para a instalação.

Em vista do resultado dos estudos realizados, várias ferramentas e técnicas de processamento de informação foram desenvolvidas, com o objetivo principal de priorizar informações a serem disponíveis ao operador, no sentido de facilitar a identificação de falhas, e ações a serem tomadas para mitigar os efeitos das mesmas.

A aplicação de Redes Neurais é uma das técnicas, com um grande potencial de exploração, que vem sendo constantemente pesquisada na atualidade, com objetivo de auxiliar o operador.

Neste trabalho é avaliado o potencial de dois tipos de redes neurais, uma do tipo Multilayer Perceptron (MLP) e outra do tipo Radial Basis Function (RBF), para monitorar a potência do reator de pesquisas IEA-R1, situado no IPEN, em São Paulo. O objetivo desta técnica é criar uma diferente forma de apresentação da informação da potência do reator, no intuito de identificar possíveis falhas na instrumentação e sistema de controle do reator. Será apresentado o desempenho de cada rede, e uma comparação entre ambas, onde será concluído que as redes do tipo RBF apresentam uma melhor performance.

# **DEVELOPMENT OF A NEURAL NETWORK TO MONITOR THE NUCLEAR POWER OF IEA-R1 RESEACH REACTOR**

**Rosa Helena Peccinini Silva Rossi**

## **Abstract**

After the Three Mile Island accident, engineers and researches from the control and instruments area, started to analyze the practicability and functionality of conventional control rooms. The conclusion was that a great amount of information was available, but without any priority, so the responsibility of coordination and selection of actions was based on the experience or the reactor supervisor, and as shown in the case or Three Mile , sometimes it was not enough to avoid undesired consequences to the installation.

According to these results, several tools and techniques of data processing were developed to prioritize information in order to give the operators an easier way to identify failures and consequent actions to minimize its effects

The use of Neural Networks is one of these techniques, with a great potential of development to help the operator to identify plant situations.

This work evaluates two proposed types of Neural Networks to monitor the nuclear power of IEAR1 research reactor. One network is the so called Multi Layer Perceptron (MLP), and the other is the Radial Basis Function (RBF). The objective is to create a diversional form to present the information about the reactor power to the operator, in order to help him to identify possible failures in the instrumentation and control system of the reactor. It will be shown the performance of each network, and a comparison between them, with the conclusion that the RBF Neural Network presents a better performance for the proposed application.

## *SUMÁRIO*

	<b>Página</b>
<b>1. Introdução</b>	
1.1 Motivação para o trabalho .....	<b>1</b>
1.2 Objetivo.....	<b>2</b>
1.3 Organização da Dissertação .....	<b>3</b>
<b>2. Histórico das tecnologias utilizadas nas salas de controle para os Sistemas de Controle e Segurança</b>	<b>4</b>
<b>3. Principais Aspectos do Reator IEA-R1</b>	
3.1 Descrição do Reator IEA-R1 .....	<b>22</b>
3.2 Métodos de Medida de Potência do Reator Nuclear .....	<b>33</b>
3.2.1 Método utilizando Detectores de Radiação .....	<b>33</b>
3.2.1.1 Instrumentos de Controle.....	<b>37</b>
3.2.1.2 Instrumentos de Segurança .....	<b>38</b>
3.2.2 Balanço Térmico .....	<b>42</b>
3.2.3 Monitoração da Atividade do Nitrogênio 16 .....	<b>44</b>
3.2.4 Vantagens e Desvantagens de cada Método utilizado para Monitorar a Potência do Reator IEA-R1 .....	<b>46</b>
<b>4. Aspectos Históricos e Teóricos das Redes Neurais</b>	
4.1 Introdução .....	<b>47</b>
4.2 Aspectos Históricos .....	<b>50</b>
4.2.1 Neurônio de McCulloch .....	<b>50</b>
4.2.2 Perceptrons .....	<b>51</b>
4.2.3 Adaline e Madaline.....	<b>53</b>
4.2.4 As Dificuldades do Perceptron.....	<b>55</b>
4.2.5 Aprendizado Competitivo e Algoritmo da Retropropagação .....	<b>56</b>
4.3 Arquiteturas de Redes Neurais Artificiais.....	<b>57</b>
4.4 Principais Algoritmos de Aprendizado .....	<b>61</b>

4.4.1	Aprendizagem Hebbiana .....	62
4.4.2	Aprendizagem por Correção de Erro.....	63
4.4.3	Regra Delta ou Método do Gradiente Descendente .....	64
4.4.4	Algoritmo da Retropropagação .....	65
4.5	Redes Neurais do Tipo Multi Layer Perceptron (MLP) .....	69
4.6	Redes Neurais do Tipo Radial Basis Function (RBF) .....	73
<b>5.</b>	<b>Desenvolvimento do Trabalho</b>	
5.1	Dados para Treinamento das Redes Neurais .....	79
5.2	Desenvolvimento das Redes Neurais .....	82
5.3	Desenvolvimento da Tela de informação .....	93
<b>6.</b>	<b>Resultados e Comparações</b>	
6.1	Resultados para Redes do tipo MLP .....	95
6.2	Resultados para Redes do tipo RBF .....	100
6.3	Comparação entre os resultados das redes do tipo MLP e RBF .....	105
6.4	Generalização da Rede.....	105
<b>7.</b>	<b>Conclusões e Recomendações</b>	<b>113</b>
<b>8.</b>	<b>Referências Bibliográficas</b>	<b>115</b>
	<b>Apêndice I</b>	<b>119</b>
	<b>Apêndice I I</b>	<b>149</b>

## LISTA DE FIGURAS

	<b>Página</b>
Figura 1 - Sala de controle convencional de uma central nuclear com reator do tipo PWR. ....	5
Figura 2 - Simulador de uma sala de controle de uma usina nuclear com reator do tipo PWR. ....	6
Figura 3 - Sala de Controle de segunda geração para reatores do tipo PWR.....	10
Figura 4 - Sala de controle do tipo PODIA, desenvolvida pela Toshiba para reatores do tipo BWR do Japão. ....	12
Figura 5 - Sala de Controle de última geração para usinas nucleares, que utilizam reatores do tipo CANDU.....	21
Figura 6 - Vista em corte do prédio do reator IEA-R1. ....	25
Figura 7 - Vista da piscina. ....	26
Figura 8 - Vista da placa matriz. ....	27
Figura 9 – Sala de Controle do reator de pesquisa IEA-R1. ....	31
Figura 10 - Intervalo operacional dos detectores de nêutrons (canais nucleares).....	35
Figura 11 - Vista do Topo da Placa Matriz do reator IEA-R1. ....	36
Figura 12 - Estrutura interna do cérebro humano. ....	48
Figura 13 - Neurônio de McCulloch. ....	50
Figura 14 - Perceptron de Rosenblatt.....	52
Figura 15 - Modelo de um neurônio Adaline.....	54
Figura 16 - Funções Booleanas de duas variáveis representadas no plano binário. ....	55
Figura 17 - Modelo Matemático de um Neurônio Artificial. ....	58
Figura 18 - Exemplos de Funções de Ativação.....	59
Figura 19 - Rede Neural com topologia em camadas.....	60

Figura 19 - Rede Neural com topologia em camadas.....	60
Figura 20 - Fluxo de Processamento do Algoritmo Backpropagation. ....	66
Figura 21 - Rede Neural do Tipo MLP. ....	69
Figura 22 - Rede Neural do Tipo RBF. ....	74
Figura 23 - Funções de Base Radial considerando $c=0$ e $\tau=1$ . ....	76
Figura 24 - Arquitetura da rede desenvolvida neste trabalho tanto para rede do tipo MLP como para rede do tipo RBF.....	82
Figura 25 - Erro Médio por Época em Função do nº de Épocas, para as redes do tipo MLP com 2, 8, 15 e 37 neurônios na Camada Oculta. ....	89
Figura 26 - Erro Médio por Época em Função do nº de Épocas, para as redes do tipo RBF com 2, 8, 22 e 37 neurônios na Camada Oculta. ....	91
Figura 27 - Tela para Monitoração da potência do reator IEA-R1.....	94
Figura 28 - Erro médio relativo em função do número de neurônios na camada oculta das redes do tipo MLP.....	96
Figura 29 - O valor da Potência fornecida pelo Canal Linear e o valor da Potência fornecida pela rede do tipo MLP.....	99
Figura 30 - Erro médio relativo em função do número de neurônios na camada oculta das redes do tipo RBF. ....	101
Figura 31 - O valor da Potência fornecida pelo Canal Linear e o valor da Potência fornecida pela rede do tipo RBF. ....	104
Figura 32 - O valor da Potência fornecida pelo Canal Linear e o valor da Potência fornecida pela rede MLP, com 15 neurônios na camada oculta, utilizando novo conjunto de dados, antes do novo Treinamento.....	108

Figura 33 - O valor da Potência fornecida pelo Canal Linear e o valor da Potência fornecida pela rede RBF, com 22 neurônios na camada oculta, utilizando novo conjunto de dados, antes do novo Treinamento.....	109
Figura 34 - O valor da Potência fornecida pelo Canal Linear e o valor da Potência fornecida pela rede MLP, com 15 neurônios na camada oculta, utilizando novo conjunto de dados, após o novo Treinamento.....	110
Figura 35 - O valor da Potência fornecida pelo Canal Linear e o valor da Potência fornecida pela rede RBF, com 22 neurônios na camada oculta, utilizando novo conjunto de dados, após o novo Treinamento.....	111

## LISTA DE TABELAS

	<b>Página</b>
Tabela 3.1 - Vantagem e Desvantagem dos Métodos utilizados para monitorar potência no reator IEA-R1 .....	46
Tabela 4.1 - Expressão de algumas funções de Base Radial.....	76
Tabela 5.1 - Valores utilizados nas fases de treinamento e teste das redes .....	80
Tabela 5.2 - Tempo de Processamento para 10.000 épocas na fase de aprendizado das redes do tipo MLP.....	88
Tabela 5.3 - Tempo de Processamento para 10.000 épocas na fase de aprendizado das redes do tipo RBF.....	90
Tabela 6.1 – Resultados obtidos para as diferentes Redes do tipo MLP, em função do número de neurônios na Camada Oculta.....	95
Tabela 6.2 – Respostas Típicas fornecidas pela rede do tipo MLP na fase de Teste, em função do nº de neurônios na Camada Oculta.....	97
Tabela 6.3 – Resultados obtidos para as diferentes Redes do tipo RBF, em função do número de neurônios na Camada Oculta.....	100
Tabela 6.4 – Respostas Típicas fornecidas pela rede do tipo RBF na fase de Teste, em função do nº de neurônios na Camada Oculta.....	102
Tabela 6.5 – Conjunto adicional de dados para teste de generalização da rede.....	106
Tabela 6.6 – Resultados obtidos para as diferentes redes com o conjunto adicional de dados, antes e depois do novo treinamento.....	112
Tabela A II.1 – Centróides (normalizados) para a Rede do tipo RBF com 2 neurônios na camada oculta.....	149
Tabela A II.2 – Centróides (normalizados) para a Rede do tipo RBF com 3 neurônios na camada oculta.....	149

Tabela A II.3 – Centróides (normalizados) para a Rede do tipo RBF com 4 neurônios na camada oculta.....	149
Tabela A II.4 – Centróides (normalizados) para a Rede do tipo RBF com 5 neurônios na camada oculta.....	149
Tabela A II.5 – Centróides (normalizados) para a Rede do tipo RBF com 6 neurônios na camada oculta.....	149
Tabela A II.6 – Centróides (normalizados) para a Rede do tipo RBF com 7 neurônios na camada oculta.....	149
Tabela A II.7 – Centróides (normalizados) para a Rede do tipo RBF com 8 neurônios na camada oculta.....	149
Tabela A II.8 – Centróides (normalizados) para a Rede do tipo RBF com 9 neurônios na camada oculta.....	149
Tabela A II.9 – Centróides (normalizados) para a Rede do tipo RBF com 10 neurônios na camada oculta.....	149
Tabela A II.10 – Centróides (normalizados) para a Rede do tipo RBF com 11 neurônios na camada oculta.....	150
Tabela A II.11 – Centróides (normalizados) para a Rede do tipo RBF com 12 neurônios na camada oculta.....	150
Tabela A II.12 – Centróides (normalizados) para a Rede do tipo RBF com 15 neurônios na camada oculta.....	150
Tabela A II.13 – Centróides (normalizados) para a Rede do tipo RBF com 20 neurônios na camada oculta.....	150
Tabela A II.14 – Centróides (normalizados) para a Rede do tipo RBF com 22 neurônios na camada oculta.....	150

Tabela A II.15 – Centróides (normalizados) para a Rede do tipo RBF com 37 neurônios  
na camada oculta..... 151

## **1. Introdução**

### **1.1 Motivação para o Trabalho**

Até cerca de 20 anos atrás a monitoração de qualquer instalação industrial era realizada por meio de operadores que observavam as informações disponíveis na sala de controle e que, por meio de uma lógica associativa, concluíam se a instalação estava na condição desejada. Em determinados casos, em instalações mais complexas, uma lógica fixa, montada com utilização de relés eletromecânicos, era utilizada para identificar situações críticas que pudessem comprometer a segurança da instalação.

Após o acidente ocorrido na central nuclear de Three Mile Island em 1979 [1], engenheiros e pesquisadores da área de instrumentação e controle começaram a analisar a praticidade e funcionamento das salas de controle convencionais. Observaram que, as mesmas, apresentavam uma grande quantidade de informações, porém não seletivas. Do ponto de vista operacional, todas as condições anormais tinham a mesma prioridade, ficando a cargo do supervisor a responsabilidade de coordenar as ações conforme a sua experiência, o que, como ficou demonstrado no caso de Three Mile, nem sempre era suficiente para evitar consequências indesejadas para a instalação.

Diversos métodos e sistemas foram desenvolvidos com a finalidade de otimizar os procedimentos para controle e operação de reatores nucleares. Em vista dos resultados dos estudos realizados várias ferramentas e técnicas de processamento de informação foram desenvolvidas, com a função principal de priorizar informações a serem disponíveis ao operador, no sentido de facilitar a identificação de falhas e ações a serem tomadas, para mitigar efeitos das mesmas.

A aplicação de Redes Neurais é uma destas técnicas e vem sendo bastante pesquisada na atualidade, com um grande potencial de exploração. É um sistema computacional que utiliza uma estrutura semelhante à rede de neurônios de um cérebro humano, e cuja aplicação tem mostrado excelentes resultados no tratamento de problemas complexos.

O principal motivo da utilização desta técnica deve-se ao fato de que a rede neural apresenta uma estrutura paralela e distribuída, composta por unidades de processamento (neurônios) com uma habilidade de aprender, e, portanto de generalizar. A generalização permite que a rede neural possa produzir saídas adequadas, para entradas que não estavam presentes durante a fase de treinamento, ou fase de aprendizagem. Esta capacidade de processamento de informação torna as redes neurais uma excelente ferramenta para a solução dos mais diversos tipos de problemas. [29]

## **1.2 Objetivo**

Este trabalho tem como objetivo o estudo de uma forma alternativa para a função da Medida da Potência Nuclear do reator de pesquisas IEA-R1, visando o desenvolvimento de uma interface homem máquina inteligente, na qual a informação seja devidamente priorizada. Para tanto foi desenvolvido um método de monitoração da Potência do Reator utilizando a técnica de Redes Neurais, cujo resultado é mostrado em uma tela de informações, junto com o resultado obtido através de outras formas de medida potência, auxiliando aos operadores da instalação na verificação de uma eventual necessidade de recalibração dos instrumentos medidores da Potência Nuclear.

O trabalho descreve o resultado da pesquisa realizada com dois tipos de redes neurais, uma do tipo Multilayer Perceptron (MLP) e outra do tipo Radial Basis Function (RBF). Serão apresentados o desempenho de cada rede, e uma comparação entre ambas.

### **1.3 Organização da Dissertação**

- **Capítulo 1:** Introdução, já descrita, que situa o tema da dissertação e define os objetivos do trabalho.
- **Capítulo 2:** São abordados os principais aspectos históricos das tecnologias utilizadas nas salas de controle para os sistemas de controle e segurança.
- **Capítulo 3:** Principais aspectos do Reator IEA-R1, descrevendo o Reator IEA-R1 e os métodos de medida de potência.
- **Capítulo 4:** Redes Neurais, aspectos históricos e principais arquiteturas.
- **Capítulo 5:** Desenvolvimento do Trabalho, onde são descritos os detalhes das redes utilizadas no trabalho e os dados experimentais obtidos no Reator IEA-R1.
- **Capítulo 6:** Resultados e Comparações, onde são apresentadas diversas avaliações referentes aos resultados obtidos pelas redes neurais pesquisadas.
- **Capítulo 7:** Apresenta as principais conclusões e as recomendações para prosseguimento da pesquisa com novos trabalhos.
- **Capítulo 8:** Encontramos a Bibliografia do trabalho.

## **2. Histórico das tecnologias utilizadas nas salas de controle para os Sistemas de Controle e Segurança.**

Várias centrais nucleares foram projetadas e construídas nas décadas de 50 e 60. Devido à tecnologia existente na época, elas eram dotadas de instrumentos analógicos e com enormes sistemas computacionais que, por motivos diversos, não apresentavam a confiabilidade necessária para aplicação no controle da instalação nuclear. Os computadores eram instalados com o objetivo de aquisição de dados, registro de eventos, alarmes e cálculos para análise de desempenho, medidas de reatividade, elaboração de cálculos, etc. As principais funções da sala de controle, sistema de controle e sistema de segurança, eram realizadas por dois sistemas puramente analógicos responsáveis pelo controle e acionamento dos dispositivos de proteção e segurança.[3]

Os instrumentos analógicos são simples, robustos e confiáveis, porém quando integrados na sala de controle apresentam algumas desvantagens, pois devem ser independentes, portanto devem ser constituídos por vários módulos, numa configuração específica para cada aplicação. Para atenderem requisitos de segurança, que exigem redundância e separação física, e para serem suficientemente rápidos no processamento das informações, cada canal de medida apresenta sua própria fonte de alimentação, seu processador de informação, seu mostrador (display), sua lâmpada de alarme etc. Pode-se usar uma chave seletora de variáveis, porém não é muito seguro, pois ao selecionar uma das variáveis deixamos de ver a resposta das outras.[3]

Devido a modularidade adotada e aos requisitos das normas aplicáveis à área nuclear, que exigem alto grau de confiabilidade para os sistemas de controle e proteção, as salas de controle das Centrais Nucleares tornaram-se enormes. [3] As figuras 1 e 2 mostram salas de controle

típicas dos projetos desenvolvidos nas décadas de 1950 e 1960, onde as informações não são priorizadas.



Figura 1 - Sala de controle convencional de uma central nuclear com reator do tipo PWR. [12]

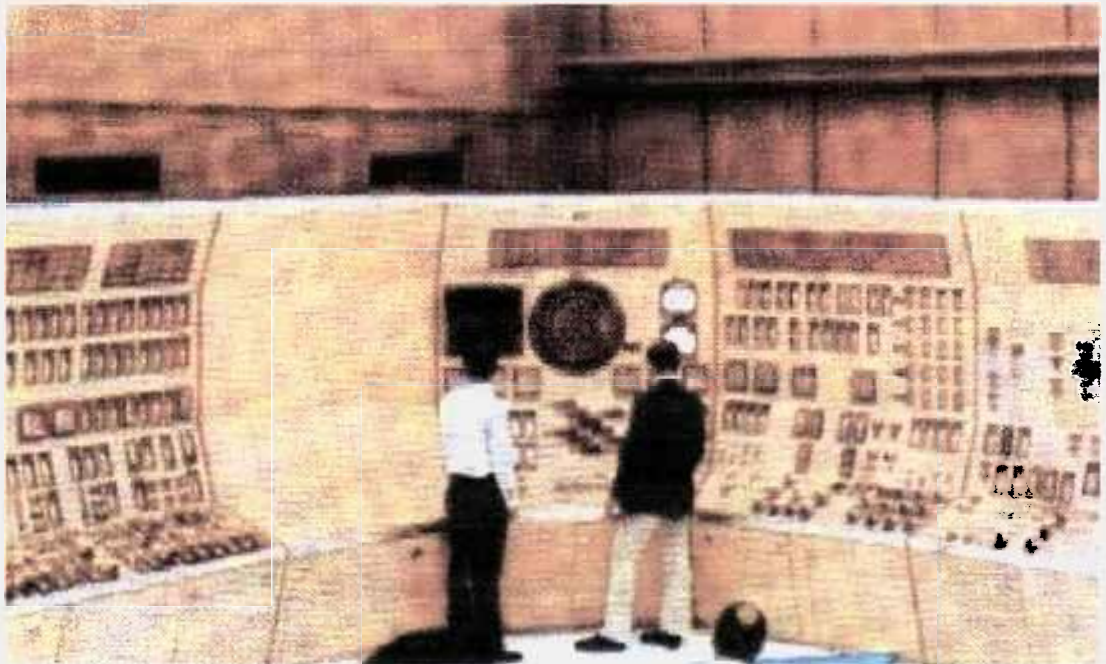


Figura 2 - Simulador de uma sala de controle de uma usina nuclear com reator do tipo PWR. [31]

Do ponto de vista operacional, numa sala de controle convencional todas as condições inseguras são alarmadas com a mesma prioridade, ficando claro a quantidade de informação que deve ser analisada pelo operador da instalação, requerendo do mesmo uma concentração muito grande, bem como uma rapidez de raciocínio para tomada de decisões em condição de alarme. [3]

No final da década de 60, a função do operador, e a possibilidade do mesmo falhar ao tomar decisões que comprometessem a segurança da instalação, passaram a ser uma preocupação muito grande para as empresas projetistas de centrais nucleares. Algumas começaram a implantar malhas de controle via computadores digitais, outras deram ênfase na utilização de computadores para priorizar a informação disponível ao operador. [3]

Em 1966 houve a primeira introdução de computadores digitais no controle de uma central nuclear. Na central nuclear de Douglas Point, num projeto da Ontario Hydro com um reator do tipo CANDU [65], foi instalada a primeira malha de controle operada por um computador digital, onde este era utilizado para obter o valor médio da potência demandada (medida através de 3 canais analógicos) e ajustar o nível de referência para controle do reator. [3]

A utilização de computadores para o controle de uma central nuclear ainda era uma decisão muito difícil, pois os computadores não apresentavam a confiabilidade desejada. Portanto, a maior parte das empresas projetistas optaram por investir em sistemas computacionais que priorizassem as informações a serem transmitidas aos operadores. [3]

No início da década de 70, foram propostas novas salas de controle de centrais nucleares, onde eram substituídos os mostradores convencionais por terminais de vídeos coloridos controlados por computadores. As variáveis da central nuclear seriam monitoradas e processadas por instrumentos convencionais (analógicos), que se comunicariam com o computador através de unidades remotas de conversão analógica-digital. Uma vez digitalizada, a informação poderia ser manipulada de acordo com as necessidades (convertida para unidades de engenharia, comparada com níveis de ajuste para indicação de alarme, analisada temporalmente, etc).

Com base na experiência obtida na central nuclear de Douglas Point, a Ontario Hydro projetou um avançado sistema de controle para operar a central nuclear de Pickering, num reator do tipo CANDU. O sistema é definido como um canal simples e dois computadores (single channel dual computer reactor control system). Segundo a filosofia adotada o sistema apresenta dois computadores independentes, onde ambos recebem as mesmas informações. Um dos computadores é o principal e o outro reserva. Cada função do computador tem um “cão de

guarda” (watchdog), uma vez que é detectada uma falha em uma das funções do computador principal, a mesma passa a ser exercida pelo computador reserva. Este sistema com pequenas modificações foi também implantado nas centrais nucleares de Gentilly e Kanupp, todas com reatores do tipo CANDU. [3]

A escolha de um sistema trabalhando com dois computadores foi adotada por dois motivos: O custo de um sistema composto por três computadores, para substituir os sistemas convencionais era proibitivo e um sistema com um computador apenas não teria a confiabilidade e disponibilidade desejada. No período de 1973 a 1976 o sistema só deixou de operar nove (9) vezes, resultando num índice de disponibilidade de 99.89%, bem próximo do índice esperado pelo projeto, que era de 99.95%. [3]

Os projetos das salas de controle das centrais nucleares de Pickering, Gentilly e Kanupp, introduziram definitivamente o computador digital no controle das centrais nucleares. As demais empresas projetistas de salas de controle também iniciaram atividades com o objetivo de integrar funções de controle por computador nos novos sistemas. Nos novos projetos a instrumentação de processo permaneceu analógica, porém a transferência de informações para o operador passou a ser de forma mais eficiente, através de computadores digitais. Dessa forma a informação passou a ser processada e priorizada, fornecendo ao operador dados solicitados pelo mesmo ou informações previamente definidas conforme uma programação que visava priorizar situações que pudessem exigir uma pronta ação do operador. [3]

Devido ao uso dos computadores o painel de controle da central ficou menor e mais funcional. Com o objetivo de manter a instalação numa condição segura, e atender às exigências dos órgãos licenciadores, que assumiam a possibilidade de falha do sistema computacional, foram

instalados painéis adicionais com instrumentos convencionais, em quantidade menor do que numa sala de controle convencional. [3] Várias foram as empresas que passaram a utilizar a tecnologia digital para melhorar o desempenho das salas de controle de centrais nucleares. Entre elas, a General Electric, que desenvolveu um sistema para reatores do tipo BWR, chamado “complexo de controle NUCLINET 1000”. O sistema era composto de um console cerca de 75% menor que o ocupado pelo sistema convencional, apresentando oito (8) terminais de vídeo coloridos. Neste sistema era necessário apenas um operador para controlar a central. Sua confiabilidade era assegurada através da adoção de redundância, e circuitos de auto-teste. [3]

Outro sistema também desenvolvido para reatores do tipo BWR [4] foi o PODIA (Plant Operation Displayed Information and Automation), projetado e construído pela Toshiba. A filosofia adotada foi à mesma que a do NUCLINET da GE, apresentando um único console onde são instalados sete (7) terminais de vídeo, do qual o operador tem acesso a toda informação da instalação e aos comandos dos dispositivos de controle, além de ter, embutido em si, malhas de controle automático. Como nos demais sistemas, a instrumentação de processo era convencional (analógica), e um outro sistema, o de proteção, totalmente independente, era responsável por detectar condições inseguras (fora de controle), e acionar dispositivos de segurança (incluindo a ação de desligamento do reator). [3]

Outras firmas que investiram no desenvolvimento de salas de controle com computadores foram a Westinghouse (com sua sala de controle integrada para reatores do tipo PWR) [5], KWU com o sistema KONVOI [5], Hitachi, com o sistema NURECS-300 [6], e ASEA-ATOM, com o sistema PROMASTER [5].

As figuras 3 e 4 mostram salas de controle nas quais a tecnologia digital começou a ser utilizada para a priorização da informação para auxiliar os operadores das centrais nucleares. A figura 3, mostra a sala de controle remodelada de um reator do tipo PWR , construída pela Westinghouse no sul do Texas. Nesta sala já se aplica a priorização das informações. [13] A figura 4 mostra uma sala de controle do tipo PODIA, desenvolvida pela Toshiba para reatores do tipo BWR do Japão. [32]

Até 1978, os projetos dos sistemas de controle não tinham nenhuma regulamentação oficial, e seguiram por conta das empresas projetistas de centrais nucleares. A única exigência era com relação ao sistema de proteção, que sempre teve requisitos rigorosos, e de certa forma, sempre foi independente do sistema de controle. Portanto, ficava a cargo da empresa projetista definir, com base em sua experiência, como e qual a informação que deveria ser mostrada ao operador. [3]



Figura 3 - Sala de controle de segunda geração para reatores do tipo PWR. [13]

Em 1979, após o acidente de Three Mile Island, por solicitação de vários órgãos oficiais, foram criadas várias comissões especiais para analisar a praticidade e eficiência das salas de controle existentes. As conclusões foram as mais críticas possíveis, principalmente com relação as salas de controle convencionais, nas quais ficou comprovado que, dependendo da situação, a forma de apresentação da informação disponível, ao invés de auxiliar o operador, poderia levá-lo a conclusões erradas, induzindo-o a tomar ações indevidas. Outra conclusão obtida foi a de que a maior parte dos computadores de processo, instalados para registros de eventos, não possuíam a confiabilidade desejada, pois dos dois computadores instalados na Central de TMI, um estava desligado para reparos e o outro operacional, apresentou problemas na impressora de alta velocidade, que ficou congestionada pelo excesso de informações. [3]



Figura 4 - Sala de controle do tipo PODIA, desenvolvida pela Toshiba para reatores do tipo BWR do Japão. [32]

Com base no resultado dos diversos relatórios divulgados, a Comissão de Regulamentação Nuclear Americana (US Nuclear Regulatory Commission), e outros órgãos licenciadores de instalações nucleares, passaram a exigir das usinas nucleares que fossem operacionais, ou que estivessem em fase de obtenção de licença de operação, as seguintes providências: [7]

1. Uma revisão do projeto das salas de controle, na qual deveriam ser considerados os seguintes itens:
  - Possibilidade de falha humana;
  - Relevância e priorização da informação;
  - Melhor e mais eficiente forma de visualização da informação;
  - Rearranjo da sala de controle;
  - Facilidade e objetividade na identificação dos itens de controle.
  
2. Instalação de um sistema de monitoração de parâmetros de segurança; um sistema com alto grau de confiabilidade e com a capacidade de mostrar o intervalo completo de operação de um conjunto mínimo de parâmetros importantes para a segurança da instalação. Um sistema que ficou conhecido com SPDS “Safety Parameter Display System”.

O SPDS é considerado um sistema de informação independente do sistema de controle. Sua finalidade é priorizar a informação referente à segurança da instalação, mostrando-a ao operador de forma simples e eficiente. Para isto, o SPDS utiliza informações provenientes do sistema de proteção da instalação, podendo também receber informações do sistema de controle.

O sistema SPDS é dividido em dois sub-sistemas: sub-sistema de monitoração de parâmetros de segurança (SMPS) e sub-sistema de monitoração de funções críticas de segurança (SMFS). O SMPS tem como objetivo agrupar um conjunto mínimo de parâmetros, mostrando-os aos operadores, com a finalidade de auxiliá-los a identificar, de forma rápida e eficiente, condições que ameacem a segurança da instalação. O SMFS tem como objetivo monitorar um conjunto de funções críticas, essenciais para manter a segurança da instalação, antecipando para

os operadores, segundo uma prioridade pré-definida, a necessidade de ações apropriadas para se minimizar os riscos da instalação. Isto inclui a função de mostrar procedimentos identificando as ações a serem tomadas pelos operadores. [3]

Numa instalação típica com reatores tipo PWR, os parâmetros mostrados pelo SMPS são: potência do reator, pressão, nível de refrigerante no pressurizador, pressão na contenção e o grau subresfriamento (um parâmetro calculado que indica se o fluido no sistema de resfriamento do reator está ou não próximo da condição de saturação). Dependendo da instalação, são monitorados, ainda, o nível e a pressão do vapor no gerador de vapor, e o nível de radiação na contenção [3].

Além de apresentar os valores atualizados dos parâmetros de segurança, o SMPS pode mostrar ainda, a taxa de variação dos parâmetros, e, segundo solicitação do operador, gráficos mostrando o histórico dos parâmetros. A forma de apresentação dos parâmetros de segurança podem ser analógico, digital ou uma combinação análogo-digital. [3]

O subsistema de monitoração das funções críticas de segurança (SMFS) monitora um conjunto de funções pré-selecionadas com o objetivo principal de evitar a fusão do núcleo do reator e a liberação de material radioativo para o meio ambiente. Num reator do tipo PWR as funções críticas mais comuns são: controle de reatividade, resfriamento do núcleo, resfriamento do refrigerante, integridade do sistema de resfriamento, integridade da contenção, inventário de refrigerante e sistemas vitais. [3]

Durante a operação normal do reator, o SPDS mantém ativo o SMPS, e o operador, se desejar, pode solicitar informações de histórico e/ou tendências, ou outras informações

disponíveis no sistema. Quando ocorrer um desligamento (“tríp”), o SPDS aciona, automaticamente, o SMFS, que passa a mostrar ao operador a condição de cada função crítica, segundo um símbolo e um código de cores pré-definidos. O símbolo pode ser um círculo, ou uma coroa dividida em setores coloridos, e é mostrado junto com os parâmetros de segurança num mesmo terminal de vídeo. O código de cores tem como objetivo permitir uma identificação rápida e segura da violação de qualquer uma das funções de segurança. Em geral as cores utilizadas são: verde, significa que a função está satisfeita, e não requer atenção do operador; amarelo, que significa que a função não está plenamente satisfeita, e requer no mínimo, a atenção do operador; laranja, que é uma indicação de que a função está na iminência de ser violada, e requer uma ação do operador; e finalmente vermelha, que significa que a função crítica foi violada, e requer imediata ação do operador. Procedimentos específicos de cada instalação estabelecem a prioridade de ação do operador no caso de aparecerem, simultaneamente duas cores diferentes da verde, para funções distintas. [3]

O SPDS foi uma solução paliativa, adotada para superar a deficiência na interface homem máquina, sem a necessidade de uma reforma completa das salas de controle existentes.

As principais dificuldades de implementação de um SPDS foram a necessidade do sistema ser classificação tipo 1E para usinas nucleares e apresentar disponibilidade de 99%. Essas dificuldades, associadas ao fato de muitas salas de controle serem consideradas obsoletas, e encontrarem muita dificuldade na obtenção de itens para manutenção, fez com que as empresas projetistas de centrais nucleares investissem cada vez mais no desenvolvimento de sistemas digitais que tivessem embutidos em si um SPDS a partir da segunda metade da década de 80. Entre elas destacaram-se a AECL (Canadá), Merlin Gerin (França) e Westinghouse (Estados Unidos), que desenvolveram sistemas integrados de proteção e controle. A General Electric na

época, optou pelo desenvolvimento de módulos digitais para substituição de canais analógicos, ou seja, eram instrumentos cuja finalidade era substituir equipamentos obsoletos. [3] O projeto dos sistemas de controle e proteção digital foi possível devido ao enorme progresso na área tecnológica dos circuitos integrados e microprocessadores, possibilitando velocidade e versatilidade no processamento das informações, menor espaço físico, e possibilidade de utilização de processamento inteligente, a partir do qual são selecionadas as informações relevantes ao operador etc.

A primeira central nuclear a ter um projeto de sistema de controle e proteção totalmente digital e distribuído foi a unidade B1 da Central Nuclear de Chooz, cujo sistema foi projetado pela Merlin Gerin baseado no uso de redes locais dedicadas se comunicando entre si, e incorporando modernas técnicas de programação. A central nuclear de Chooz, construída próximo à divisa com a Bélgica pela Framatome (França), tem um reator do tipo PWR para geração de 1400 MWe.

O sistema proposto pela Merlin Gerin era um aperfeiçoamento do sistema SPIN (integrated Digital Protection System), que foi desenvolvido por ela desde 1979 para servir como sistema de proteção e atuação de sistemas de segurança, e que era considerado operacional nas centrais nucleares da França desde 1984.

A instrumentação proposta para ser utilizada em Chooz era o resultado da combinação do SPIN (que já tinha como extensão o sub sistema de instrumentação nuclear), com o sistema de controle digital desenvolvido pela CGEE Alstom. [8] Devido às dificuldades de qualificação do software e incerteza da aprovação do órgão licenciador, a parte de proteção acabou sendo convencional. [37]

O sistema proposto pela Westinghouse utilizava uma filosofia semelhante ao da Merlin Gerin, sendo dividido em três grandes sub sistemas: lógica, controle e proteção. Segundo um contrato assinado com CEGB (Inglaterra), uma versão ligeiramente modificada do sub sistema de proteção foi utilizada na Central Nuclear de Sizewell B no reator do tipo PWR.

Com o passar dos anos o tamanho e a complexidade das centrais nucleares cresceu significativamente, acrescentando requisitos nas atividades de operações, manutenção, engenharia e administração de pessoal para melhorar o uso, a confiabilidade e o grau de segurança da central. Desde o acidente de Three Mile Island, os sistemas de suporte ao operador mostraram sua habilidade para atender os operadores nas suas decisões. Em algumas centrais, modernos computadores têm provido acesso a grandes números de medidas e cálculos de dados.

Seguindo uma tendência natural, o desenvolvimento de componentes digitais cada vez mais rápidos e de grande capacidade, incentivaram o desenvolvimento de sistemas de monitoração e diagnóstico cada vez mais complexos, de forma que, na década de 90 os trabalhos e pesquisas realizados foram focados no aperfeiçoamento do sistema computadorizado para suporte ao operador de centrais nucleares, (Operator Support System - OSS), adotando cada vez mais o conceito de que as ações em uma central nuclear é uma combinação de ações humanas com automação, ou seja, computador e relação homem-máquina, onde sistemas de computadores são usados para operações de suporte e manutenção de tarefas complexas, procurando facilitar as ações do operador, cuja função principal passou a ser de supervisor e antecipador de situações indesejáveis. Dessa forma ficaram evidentes os benefícios advindos do uso de computadores, desde que fosse assegurado que o projeto e implementação do sistema desenvolvido fossem realizados por pessoas devidamente treinadas e levando-se em conta fatores como integridade do sistema, software de validação e atividades de verificação e análise de consequência de erros. [9]

O sistema de suporte ao Operador (OSS) tem sido baseado numa metodologia de monitoração em tempo real associando programas de simulação do processo com medida de parâmetros específicos, para identificar situações consideradas anormais. A monitoração implica essencialmente na operação em tempo real, portanto o sistema de operação é baseado em modelos dinâmicos para análise e decisões. Diferentes técnicas de modelos dinâmicos foram desenvolvidas para o sistema de suporte. [9]

Entre os sistemas do tipo OSS desenvolvidos destacamos o sistema da Westinghouse, que desenvolveu um sistema modular para supervisão de centrais nucleares com três subsistemas, ou módulos distintos, a saber, Advanced Alarm Management System (AWARE), Revolutionary Core monitoring Program (BEACON) e Computerized Procedures Package (COMPRO).

O AWARE é um sub-sistema inovativo com o objetivo de evitar que o operador seja surpreendido por um grande número de alarmes durante um grande distúrbio, para tanto, ele alerta o operador para pequenos distúrbios, melhorando a resolução num estado inicial do problema e realizando decisões durante algum desligamento. [9]

BEACON é um sub sistema projetado para melhorar a operação de suporte específico para reatores do tipo PWR. Utilizando programas de simulação e medidas específicas de detectores de fluxo neutrônico e alguns de temperatura. Apresenta uma avançada monitoração do núcleo em tempo real, permitindo uma visão tridimensional da distribuição da potência no interior do mesmo. Devido à monitoração do BEACON é possível detectar imediatamente condições de distribuição de potência com anomalia no núcleo, posições indevidas das barras de controle/segurança, e condições de resfriamento (dos elementos combustíveis) consideradas anormais. O

sub sistema Beacon, de uma forma geral, é considerado uma ferramenta que simplifica muito o processo de supervisão da central nuclear e melhora as funções de suporte principais. [9]

Para auxiliar os operadores das centrais nucleares, nas tomadas de decisão, em ambos os casos, de operação normal e acidente foi desenvolvido o sistema COMPRO (Computerized Procedures System), que é basicamente um sistema de processamento de texto baseado em CRT. Este sistema é utilizado para orientar os operadores, na medida em que apresenta os procedimentos a serem seguidos podendo, dependendo da situação, apresentar algumas informações específicas relativas à uma tomada de decisão. Desta forma minimiza-se o risco de erros durante o uso dos procedimentos na central nuclear. [9]

Embora as técnicas de redes neurais ainda não tenham sido implantadas em sistemas de controle/proteção de centrais nucleares, muitos são os trabalhos realizados mostrando a adequação dos mesmos para identificar condições anômalas. Um caso típico é o trabalho realizado por Bartlett e Uhrig, que desenvolveram uma rede neural para classificar algumas das condições de acidente de uma central nuclear, tais como rompimento da “perna fria”, rompimento da linha principal de vapor, falha nos tubos do gerador de vapor, etc. Após utilizar dados de um simulador de acidentes para treinar a mesma, eles concluíram que a rede obteve uma ótima identificação (e classificação) dos acidentes simulados. [36]

Devemos observar que muitos dos trabalhos são desenvolvidos utilizando-se dados de simulações ou dados reais de centrais nucleares, como é também no caso do trabalho desenvolvido por Embrechts e Beneck, que desenvolveram um sistema de redes neurais para identificação de transientes da potência de um reator nuclear. A rápida identificação de um mal funcionamento é fundamental para a segurança operacional de uma central nuclear, e um mal

funcionamento deve ser identificado em intervalo, inferior a 60 segundos. Nesse sentido uma rede neural do tipo feedforward com algoritmo de backpropagation, foi desenvolvida e treinada para modelar a simulação de um mal funcionamento do reator nuclear do tipo PWR, (Pressurized Water Reactor) da Hungarian Parks Nuclear Power Plant, e esta sendo utilizado, com sucesso, para identificar falhas no simulador da instalação. [10]

Além de serem utilizadas para monitoração e diagnóstico em centrais nucleares, as técnicas de inteligência artificial também encontram aplicação no gerenciamento do combustível nuclear. É o caso dos algoritmos evolucionários, do tipo PBIL "*Population-Based Incremental Learning* (PBIL)", que são sistemas adaptativos motivados biologicamente que têm sido usados, com bons resultados, para a otimização de funções. O trabalho apresenta uma forma de paralelização a ser aplicada ao algoritmo, que é um método de combinação dos mecanismos do algoritmo genético padrão com o aprendizado competitivo simples, visando a elaboração de um algoritmo evolucionário a ser aplicado em processos de otimização de problemas combinatórios. [11]

O PBIL foi desenvolvido especialmente para aplicação no problema de otimização da recarga em reatores nucleares do tipo PWR. Para a realização dos testes com o algoritmo PBIL foram utilizados problemas combinatórios similares ao problema da recarga. Ganhos significativos mostraram as vantagens e a viabilidade do algoritmo, demonstrando o potencial da ferramenta desenvolvida na solução do problema da recarga.

Atualmente outras técnicas de inteligência artificial estão sendo estudadas e pesquisadas, como Lógica Fuzzy, conhecida também como Lógica Nebulosa e Sistemas Especialistas, sempre visando a otimização da interface homem-máquina.

Como exemplos, podemos citar os seguintes trabalhos apresentados no segundo semestre de 1999, utilizando técnicas de inteligência artificial, rede neural e lógica nebulosa: “Sistema de Monitoração e Diagnóstico de Válvula Moto-Operada, Utilizando Lógica Nebulosa” [33], “Modelos de Validação de Sinal Utilizando Técnicas de Inteligência Artificial Aplicados a um Reator Nuclear” [34], “Um Algoritmo Genético com Clusterização adaptativa Nebulosa para a Otimização de Problemas de reatores Nucleares”. [35]

De uma forma geral podemos dizer que no futuro, as salas de controle das centrais nucleares devem fazer uso cada vez mais da tecnologia digital e das técnicas de inteligência artificial para auxiliar o trabalho dos operadores e supervisores nas tarefas de manter a central operando dentro de condições seguras. A figura 5 mostra a concepção de uma das salas mais modernas que deve ser o centro para controle dos reatores do tipo CANDU de 6ª geração. [14]



Figura 5 - Sala de Controle de última geração para usinas nucleares, que utilizam reatores do tipo CANDU. [14]

### **3. Principais Aspectos do Reator IEA-R1**

#### **3.1 Descrição do Reator IEA-R1**

O reator nuclear de pesquisas IEA-R1 pertence ao Instituto de Pesquisas Energéticas e Nucleares (IPEN), localizado no Campus da Universidade de São Paulo na Cidade Universitária Armando de Salles Oliveira, em São Paulo, SP.

O reator foi projetado pela “Babcock & Wilcox Company”, e começou a ser construído em 1956. O IEA-R1 é um reator de pesquisa do tipo piscina, moderado e refrigerado a água leve, utilizando grafite como refletor, tendo atingido sua primeira criticalidade em 16 de setembro de 1957 e, após a fase inicial de testes e comissionamento, passou a operar na potência de 2MW, conforme um ciclo de 8 horas por dia, 5 dias da semana. [15]

É um reator de pesquisa com as seguintes finalidades: [15]

- Produção de radioisótopos para aplicações na medicina, indústria, agricultura e para pesquisas;
- Servir como uma fonte intensa de nêutrons em experimentos científicos de física nuclear, química, engenharia e biologia;
- Treinamento do corpo científico do IPEN-CNEN/SP em física de reatores, projetos, desenvolvimento de instrumentação nuclear e segurança de reatores;
- Treinamento e formação de operadores de reator; e
- Análise radioquímica de amostras através do método de análise por ativação.

Em 1995, devido à um aumento significativo das solicitações de radioisótopos para medicina, indústria e agricultura e à necessidade de uma readequação aos requisitos de segurança, a Comissão Nacional de Energia Nuclear e o IPEN iniciaram o projeto de modernização do reator, incluindo o aumento da sua potência para 5 MW. Em 16 de setembro de 1997, foi alcançada a potência planejada, ampliando-se o ciclo de operação para 64 horas contínuas semanais. A partir de 2001, visando a produção de novos radioisótopos, o ciclo de operação deverá ser alterado para 120 horas contínuas semanais. [16]

O prédio do reator IEA-R1 é composto por cinco pavimentos, mostrados na figura 6 e descritos a seguir: [16]

- Subsolo: acomoda a casa de máquinas dos sistemas de resfriamento, tratamento e retratamento de água da piscina;
- 1º pavimento: no saguão encontram-se os “beam hole”, ou seja, canais que disponibilizam os feixes de nêutrons para os experimentos de física nuclear, física da matéria condensada e aplicações comerciais (neutronografia);
- 2º pavimento: acomoda a casa de máquinas do sistema de ventilação e ar condicionado e sala de apoio da proteção radiológica;

- 3º pavimento: neste andar encontram-se a sala de controle, o saguão da piscina, a oficina mecânica e uma sala de exposição. No interior da piscina localiza-se a placa matriz, que acomoda os elementos combustíveis, elementos combustíveis de controle, refletores, dispositivos para irradiação de amostras e os canais para medição da potência do reator;
- 4º pavimento: onde está localizado o Sistema de Exaustão do Reator.

O reator de pesquisas IEA-R1 consiste num arranjo nuclear imerso em uma piscina, conforme figura 7, contendo 273 m<sup>3</sup> de água desmineralizada. A piscina possui cerca de 9 metros de profundidade por 3,05 metros de largura e 10,65 metros de comprimento. A piscina está dividida em dois compartimentos, o da esquerda, utilizado para operação do reator e o da direita utilizado para armazenamento do combustível queimado e manuseio de amostras. A piscina é revestida internamente por chapas de aço inoxidável, sendo que a blindagem radiológica que envolve o núcleo é feita por uma camada de água de 1,5 metros de espessura na região lateral, e mais uma camada de concreto com barita que pode atingir 2,4 metros nas paredes da piscina. [16]

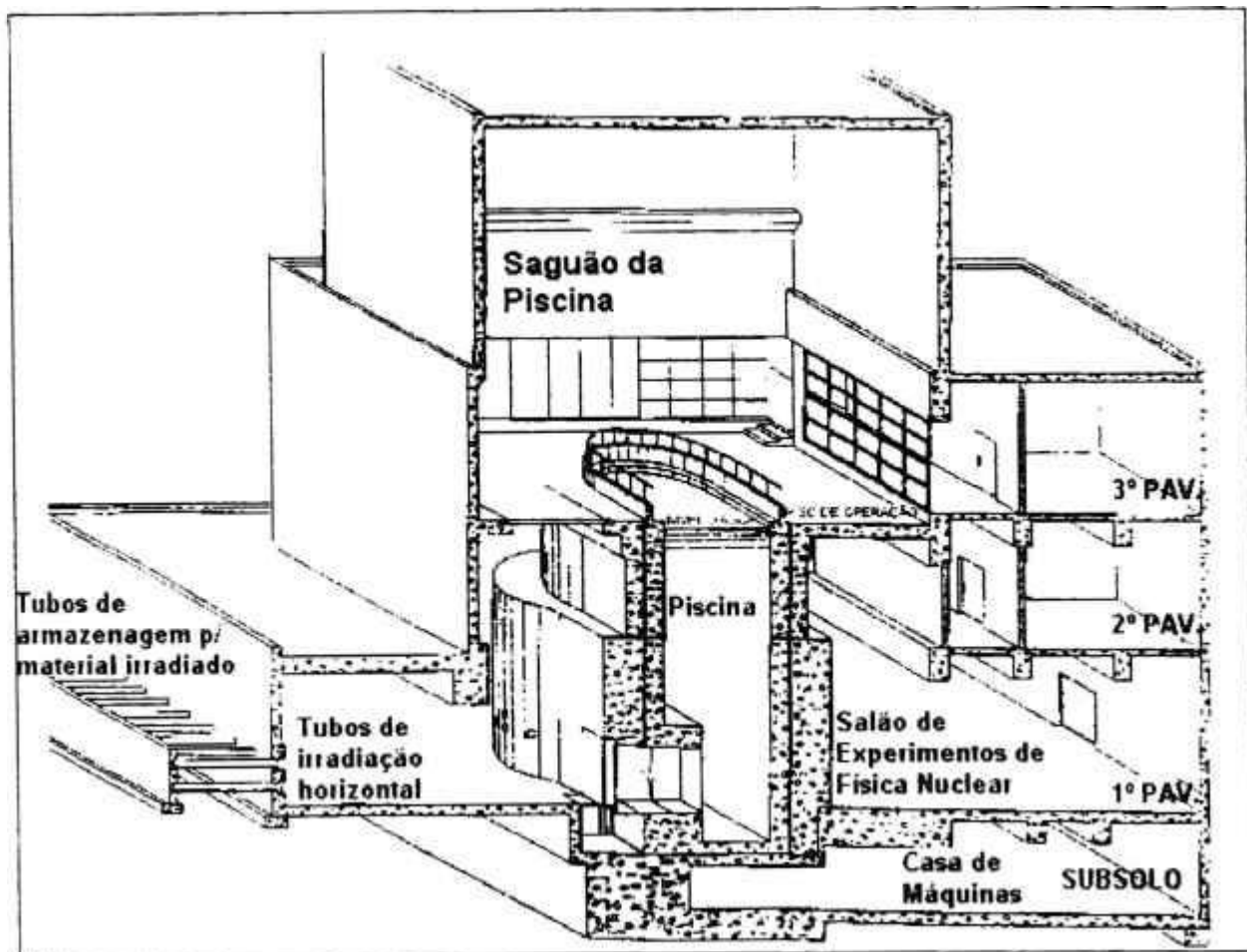


Figura 6 - Vista em corte do prédio do reator IEA-R1. [15]

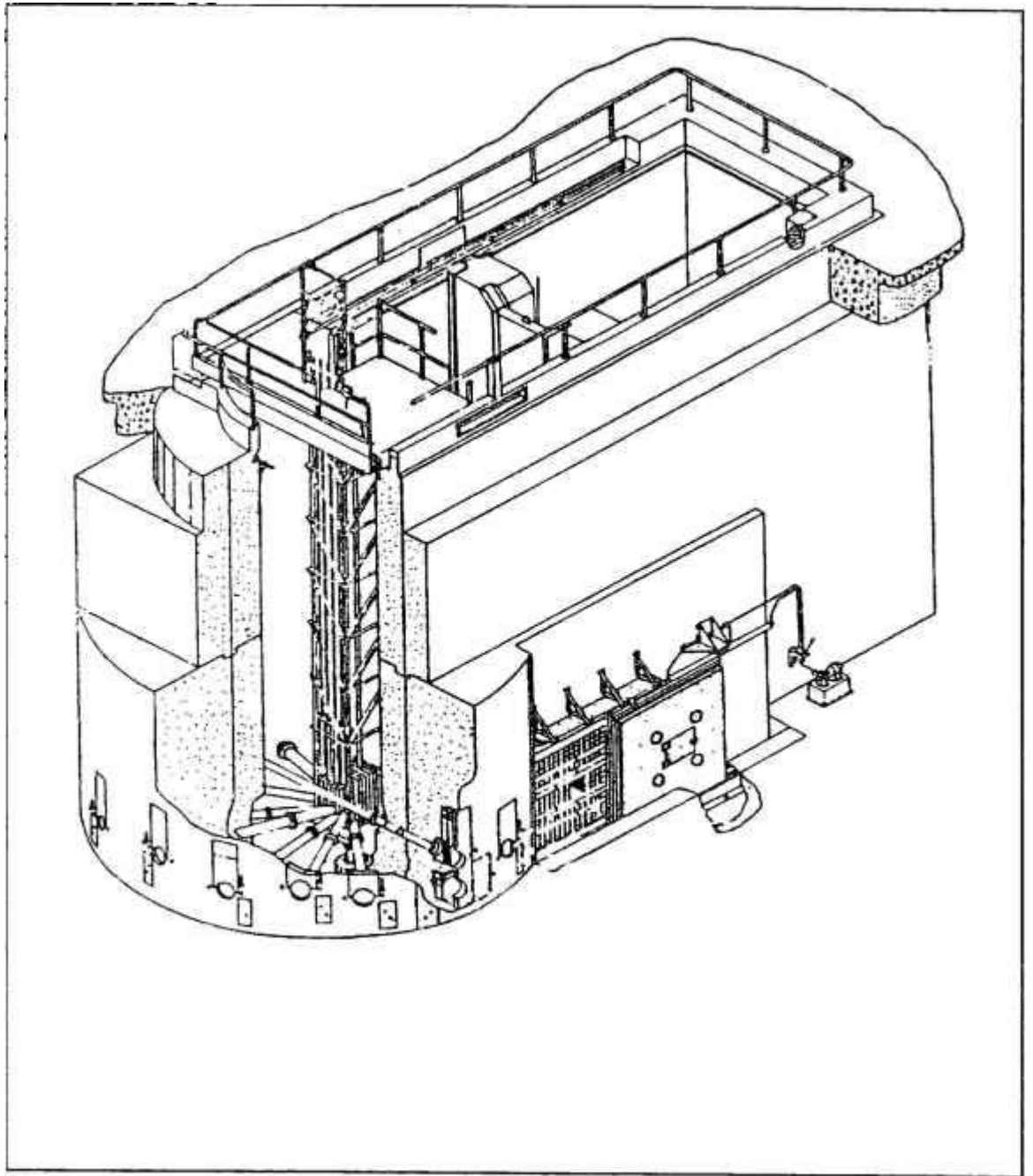


Figura 7 - Vista da piscina. [15]

O núcleo do reator encontra-se dentro da piscina a uma profundidade de 8 metros da superfície da piscina, tem a forma de um paralelepípedo composto por 20 elementos combustíveis padrões, 4 elementos combustíveis de controle, cerca de 25 elementos refletores, 7 posições para irradiação de amostras e tampões encaixados verticalmente em uma placa matriz de alumínio com 80 orifícios, conforme ilustrado na figura 8. Esta placa é sustentada por uma treliça conectada a uma plataforma móvel que permite o deslocamento do reator ao longo da piscina. [16]

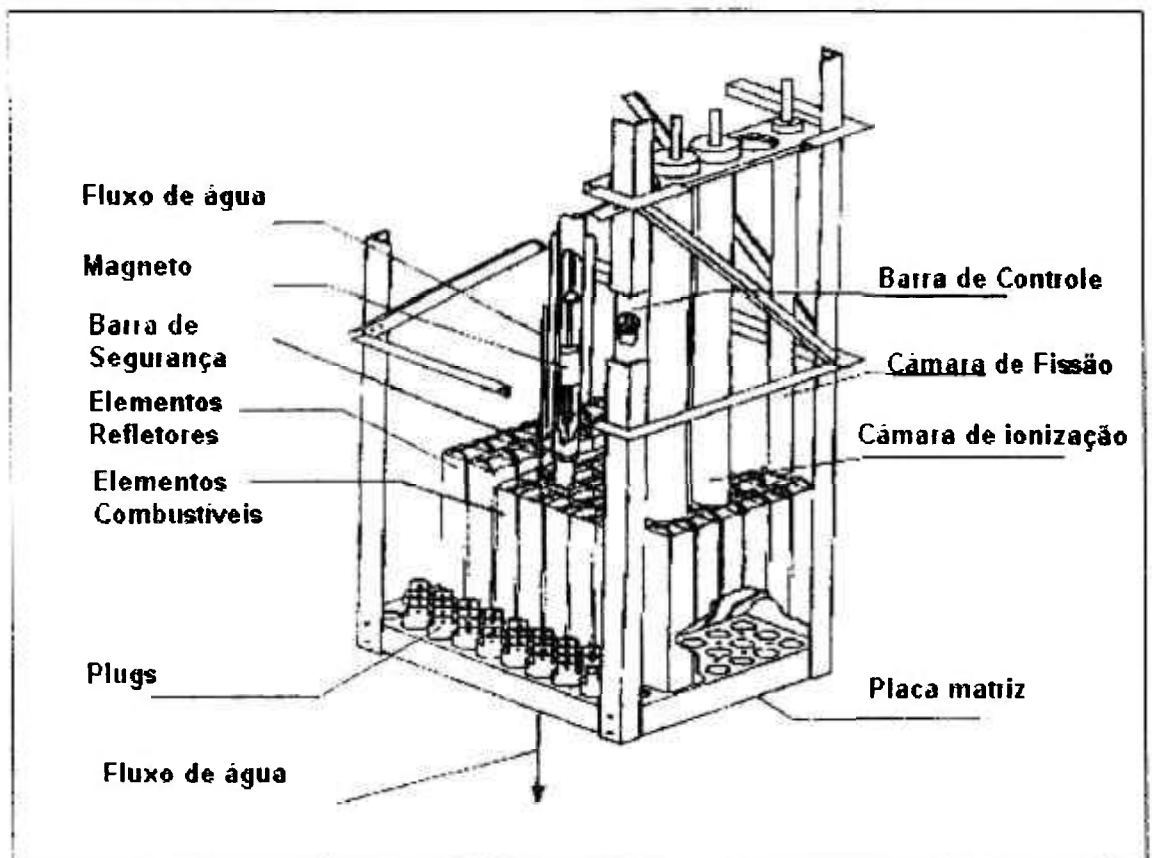


Figura 8 - Vista da placa matriz [15]

Para as atividades de irradiação, o reator possui dispositivos específicos na placa matriz, um sistema pneumático e diversos tubos de irradiação horizontais, chamado “Beam holes”.

- Dispositivos de irradiação na placa matriz: Para irradiação em altos campos de fluxo neutrônico, o reator possui uma série de dispositivos onde são colocadas cápsulas de alumínio contendo as amostras de material a ser irradiado. Esses dispositivos são posicionados na placa matriz para irradiações com duração mínima de 1 hora e fluxo de neutrons superior a  $10^{12}$  n/cm<sup>2</sup>.seg.
- Sistema Pneumático de Irradiação: É utilizado para irradiações rápidas, em geral, inferior a 30 minutos, e com amostras pequenas, em geral inferior a 1 grama. As amostras são introduzidas em cápsulas de polietileno e enviadas até as proximidades do núcleo através de um sistema pneumático. As estações de envio e recebimento destas cápsulas estão situadas em dois laboratórios localizados na parte externa do prédio do reator.
- Tubos de Irradiação Horizontais: O reator possui nove tubos com diâmetros entre 6” e 8” que se prolongam até a face externa da parede do reator. Oito tubos são radiais e um tubo é tangencial. Todos eles são utilizados para realização de experimentos de física nuclear, física do estado sólido e neutrongrafia.

O circuito primário de resfriamento do reator é constituído por duas bombas, dois trocadores de calor e diversas válvulas que permitem fazer o alinhamento de qualquer uma das bombas com qualquer um dos trocadores de calor. Durante a operação do reator basta uma bomba e um dos trocadores para dissipar a potência produzida no reator, cujo valor nominal é 5 MW ( térmicos).

Um conjunto de duas bombas e duas torres de resfriamento, mais diversas válvulas que permitem alinhar o trocador de calor selecionado e qualquer uma das torres forma o circuito secundário de resfriamento.

O calor produzido no núcleo do reator, pelas fissões nucleares, é removido pela passagem de água do circuito primário através dos elementos combustíveis. Este calor é transferido para a água do circuito secundário no trocador de calor e dissipado na atmosfera através da torre de resfriamento. A tubulação do circuito primário é dotada de quatro válvulas de isolamento, sendo duas na entrada e duas na saída da piscina. A função destas válvulas é o de evitar um esvaziamento da piscina em caso de ruptura acidental de uma tubulação. [16]

Um tanque de decaimento localizado no circuito primário de resfriamento funciona como um retardo para o retorno da água para a piscina, possibilitando o decaimento do  $^{16}\text{N}$  formado na água através da reação  $^{16}\text{O} (n,p) ^{16}\text{N}$ .

O reator possui dois tipos de elementos combustíveis, um chamado padrão e o outro denominado de “controle”, ou parcial. O elemento combustível padrão é do tipo MTR (Material Testing Reactor) com 18 placas planas contendo urânio enriquecido a 20%. O elemento combustível de controle é semelhante ao padrão, só que com 6 placas a menos, ou seja, possui apenas 12 placas combustíveis.

O controle de reatividade do reator é feito através de três barras de segurança e uma barra de controle, do tipo placa, que são movimentadas no núcleo por um mecanismo de acionamento de motores síncronos. As barras são sustentadas por magnetos que são desenergizados no caso

de algum evento não usual, fazendo-as cair por ação da gravidade, desligando o reator. Cada barra é formada por duas lâminas constituídas de uma liga de prata (80%), índio (15%) e cádmio (5%) revestida com uma película de Níquel. [16, 15]

Duas câmaras de ionização não compensadas, uma câmara de ionização compensada e uma câmara de fissão fazem parte da instrumentação nuclear, que enviam sinais para o console principal localizado na sala de controle do reator. Também na sala de controle, mostrada na figura 9, estão instalados os instrumentos de apoio à operação, abaixo relacionados: [16]

- Monitores de radiação de área, dutos e contaminação de ar;
- Sistema de alarmes de radiação;
- Comandos das bombas dos circuitos primário, secundário e ventiladores das torres de resfriamento;
- Comandos do sistema de ventilação e exaustão de ar-condicionado;
- Alarme de incêndio;
- Comando do sistema de resfriamento de emergência;
- Comando das válvulas de isolamento do circuito primário;
- Indicação de condutividade da água da piscina e da água de reposição;
- Sistema de análise de vibração dos mancais de equipamentos rotativos.



Figura 9 - Sala de Controle do reator de pesquisa IEA-R1.

A água da piscina do reator é mantida em boa qualidade por sistemas de resinas trocadoras de íons, consistindo nos sistemas de tratamento e retratamento da água. O sistema de tratamento é utilizado para completar o nível de água da piscina com a pureza necessária aos níveis operacionais. O sistema de retratamento opera continuamente e tem como finalidade a manutenção da qualidade da água da piscina de maneira a minimizar os efeitos de corrosão e os níveis de dose de radiação na superfície da piscina.

O fornecimento de energia elétrica é feito por meio da rede elétrica da concessionária local. No caso de interrupção no fornecimento, um conjunto de quatro moto-geradores fornecem energia elétrica aos sistemas vitais e essenciais para manter o reator em operação.

## **3.2 Métodos de Medida de Potência do Reator Nuclear**

A potência dissipada pelo reator nuclear IEA-R1, cuja potência nominal é de 5 MW, é geralmente determinada por uma combinação dos seguintes métodos:

### **3.2.1 Método Utilizando Detectores de Radiação**

A potência do reator nuclear é diretamente proporcional à população de nêutrons em seu núcleo. Havendo uma variação na população neutrônica, conseqüentemente a potência do reator irá sofrer a mesma taxa de variação. Para monitorar esta variação, são utilizados detectores de nêutrons, cujo sinal, uma vez processados permitem a monitoração e os registros (manual e/ou automático) dos níveis dos fluxos de nêutrons, que são linearmente proporcionais à potência do reator.

Um processamento adicional permite monitorar o período (**T**) do reator, que representa a taxa de variação do fluxo neutrônico (o período de um reator é definido como o tempo necessário para que o fluxo neutrônico seja multiplicado pelo fator “e”, onde “e” = 2,7182), ou seja, o tempo em segundos, necessário para que a potência do reator aumente de um fator “e”.

Existem duas classes de instrumentos que permitem monitorar o fluxo neutrônico, e conseqüentemente a potência nuclear do reator, são instrumentos de controle e instrumentos de segurança, formando um subsistema de instrumentação nuclear [17]

O subsistema de instrumentação nuclear é constituído por um canal de faixa ampla, três canais de segurança, e um canal linear com mudança de escala. Ao todo são utilizados 4

detectores de nêutrons, sendo uma câmara de fissão, duas câmaras de ionização não compensadas e um câmara de ionização compensada. Um canal adicional, utilizado para monitorar a atividade do  $^{16}\text{N}$  presente na água que passa pelo núcleo do reator, completa o subsistema. O canal de faixa ampla e os canais de segurança estão diretamente ligados ao sistema de proteção, e os demais são utilizados apenas para controle ou monitoração.

A figura 10 mostra o intervalo operacional dos vários canais nucleares, e a figura 11 mostra a localização dos detectores de nêutrons, em relação ao núcleo do reator. Todos os detectores de nêutrons estão instalados na piscina do reator, dentro de tubulões estanques de alumínio, que por sua vez são fixados na estrutura que suporta o núcleo do reator. A posição axial dos detectores é variável, e ajustada para que os mesmos forneçam, um sinal de 100% quando o reator estiver a 5 MW. Uma película isolante, colocada ao redor dos detectores, garante a isolamento entre a carcaça do detector e o tubulão de alumínio.

### Potência Nominal em %

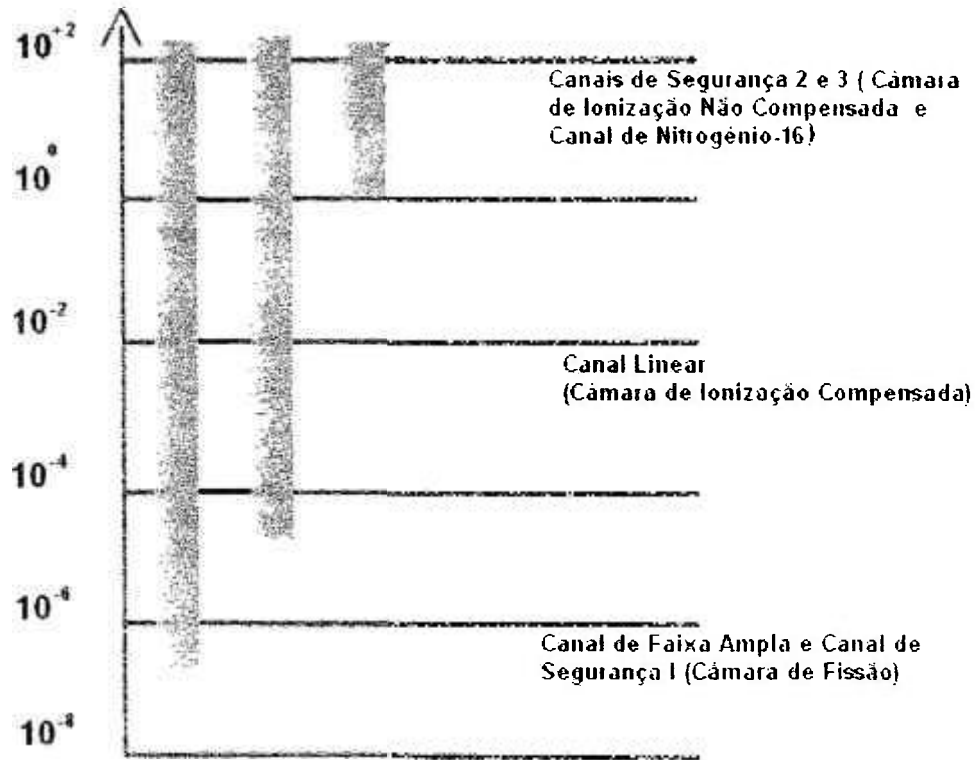
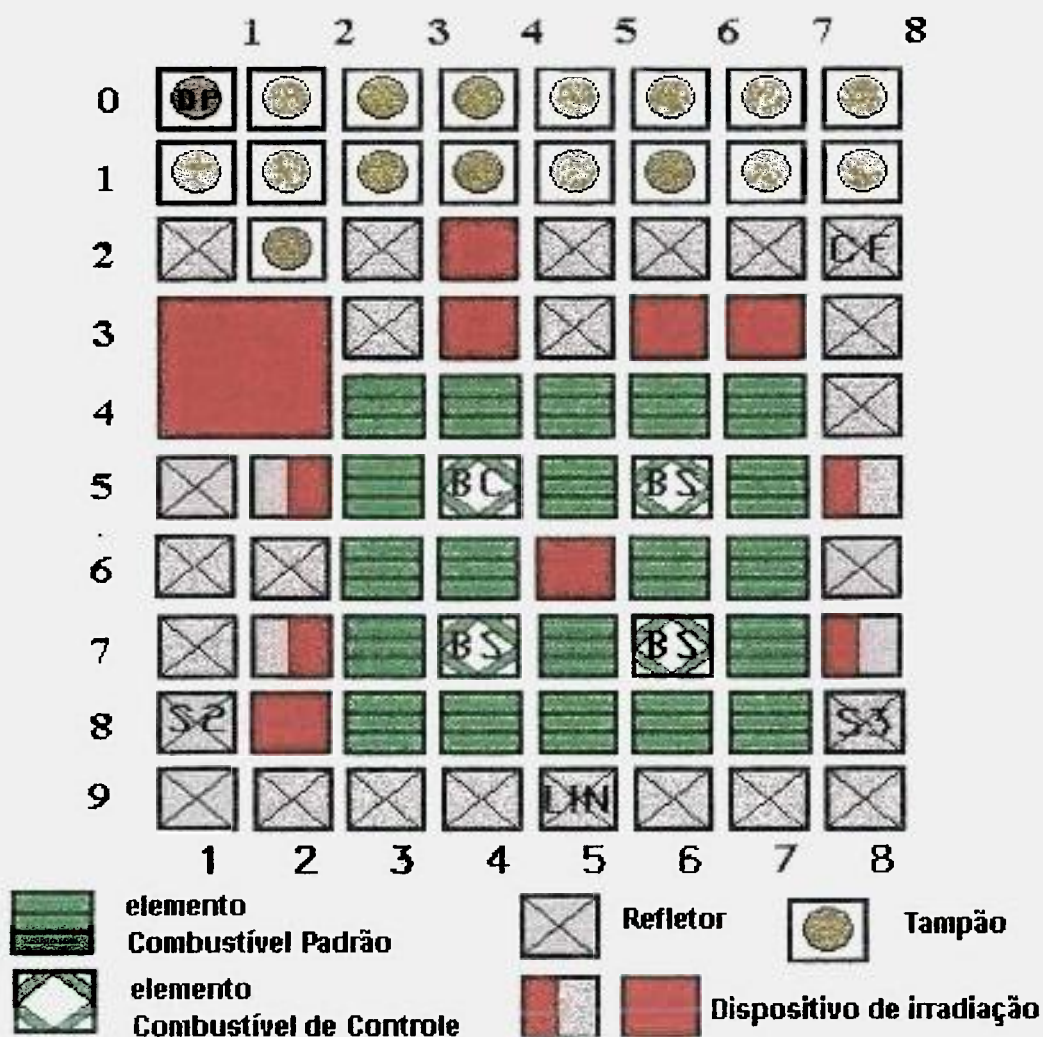


Figura 10 - Intervalo operacional dos detectores de nêutrons (canais nucleares). [15]



**CF – CÂMARA DE FISSÃO (Canal de faixa ampla)**

**S2 – CÂMARA DE IONIZAÇÃO NÃO COMPENSADA (Safety 2)**

**S3 – CÂMARA DE IONIZAÇÃO NÃO COMPENSADA (Safety 3)**

**LIN – CÂMARA DE IONIZAÇÃO COMPENSADA (Canal Linear )**

**BS – BARRAS DE SEGURANÇA**

**BC – BARRA DE CONTROLE**

**DP – MEDIDOR DE DP (variação da pressão) no núcleo**

Figura 11 - Vista do Topo da Placa Matriz do reator IEA-R1

Os atuais Sistemas de Controle e Proteção do Reator operam de forma integrada desde 1976, demonstrando eficiência e confiabilidade na função de proteção do reator. Deve-se destacar que os diversos instrumentos que compõem os Sistemas de Controle e Proteção operam num ambiente de laboratório, no qual as condições ambientais são devidamente controladas, o que ajuda a minimizar a sua degradação.

### **3.2.1.1 Instrumentos de Controle**

O Sistema de Controle da Reatividade do reator é composto por um canal nuclear (denominado canal linear com mudança de escala), pelos mecanismos de movimentação de barras de controle/segurança, pelos dispositivos de acionamento manual e indicação de posição das barras, por uma chave de seleção automático/manual, por um potenciômetro de ajuste de demanda, que fornece o valor de referência para o controlador automático, e por uma unidade de controle automático. Um canal adicional, de  $^{16}\text{N}$  que será descrito posteriormente, permite uma monitoração adicional do nível de potência térmica do reator, e sua medida é utilizada como referência para calibração dos demais canais nucleares. [15]

O canal linear com mudança de escala é composto por uma câmara de ionização compensada (CIC), revestida com  $^{10}\text{B}$ , fonte de alimentação e um picoamperímetro de multi-faixa linear. Este canal é capaz de monitorar a potência do reator de 1miliwatt até 10 MW (com precisão que vai de 10% da medida em baixa potência até 1% na região de alta potência). Projetado especificamente para ser utilizado em reatores nucleares, sua eletrônica consiste de um amplificador linear de corrente com impedância de entrada zero e uma chave para mudança de escalas com acionamento manual, que é utilizada para selecionar a sensibilidade da corrente de

entrada para o amplificador. O picoamperímetro gera dois sinais, um para a unidade de controle automático do reator e outro para registro (aquisição de dados). O canal linear não gera sinais para desligamento do reator, ele apenas informa o percentual (%) de potência. [15]

A incerteza da medição do Canal Linear é por volta de 5%, onde temos, 4% devido ao processo de calibração (como veremos mais adiante, no item 3.2.2, onde o balanço térmico apresenta uma incerteza de 4%). A incerteza da leitura do registrador, que é dada pela mínima divisão do mesmo, adiciona mais 1%. [18]

### **3.2.1.2 Instrumentos de Segurança**

O sistema de proteção e segurança do reator atua basicamente no circuito de desligamento (SCRAM) do reator, interrompendo, se necessário, a corrente que alimenta os magnetos responsáveis pelo acoplamento das barras de controle/segurança aos respectivos mecanismos de movimentação. Desta forma, se as barras de controle/segurança ainda estiverem inseridas no núcleo do reator, as mesmas são impedidas de serem removidas, e no caso de estarem fora do núcleo, são imediatamente inseridas pela ação da gravidade. O sistema atua automaticamente, iniciando o processo de desligamento do reator sempre que uma das variáveis do sistema atinge um valor pré-definido, e, uma vez iniciado o processo, é impossível evitar-se o desligamento do reator. Para assegurar o desligamento do reator são utilizadas três barras de segurança e uma de controle. Uma vez iniciado o processo de desligamento as quatro barras são liberadas para inserção no interior do núcleo do reator. A inserção ocorre num intervalo inferior a 1 (um) segundo, após as barras serem liberadas dos respectivos mecanismos de acionamento [15].

Fazem parte do sistema de proteção do reator todos os canais que monitoram os parâmetros relacionados com a segurança da instalação, e que conforme estabelecido na análise de segurança permitem identificar situações em que deve ser providenciado o desligamento do reator. Fazem parte também do circuito de desligamento do reator, os dispositivos de sinalização e alarme, e os dispositivos para acionamento manual do sistema. [15]

O sistema é composto por quatro canais nucleares, que fazem parte do subsistema de instrumentação nuclear e permitem monitorar a potência e o período do reator; três monitores de radiação que permitem monitorar o nível de radiação na superfície da piscina do reator e no salão de experimentos de física nuclear; um monitor de vazão no circuito primário do sistema de resfriamento do reator; um transmissor de pressão diferencial utilizado para medir a diferença de pressão no núcleo do reator; um termômetro utilizado para monitorar a temperatura da água na saída da piscina; um transmissor de pressão diferencial e uma chave de nível utilizados para monitorar o nível de água na piscina do reator; três botões de desligamento manual; e vários contatos, do tipo on/off, para monitorar que alguns equipamentos estão na condição correta para habilitar a operação do reator. [15]

Um dos canais nucleares é o canal de faixa ampla, ou Campbell, que tem como detector uma câmara de fissão (com urânio enriquecido no isótopo -  $^{235}\text{U}$ ). É um detector especial, conectado a uma eletrônica que permite monitorar o nível de fluxo neutrônico num intervalo de 10 décadas. Usando o sinal proveniente da câmara de fissão, o canal combina um sinal proporcional ao logaritmo da taxa de contagem nas seis primeiras décadas, com um sinal proporcional à flutuação da corrente (técnica Campbell) nas quatro últimas décadas, para permitir a monitoração do fluxo neutrônico no intervalo de dez décadas. Ou seja, o Canal de faixa ampla do reator de pesquisa IEA-R1 é responsável pela monitoração do fluxo de nêutrons nas faixas

operacionais de partida e intermediária. É usado para monitorar o fluxo de nêutrons desde o início da partida ( $10^{-7}$  % da potência, 0,5 W, ou “background”) até a região intermediária (10 % da potência, 500KW). [15]

O canal Campbell também permite monitorar o período do reator, gerando um sinal eletrônico que é devidamente processado e utilizado para mostrar o período num mostrador calibrado em segundos, e que vai de -30 a +3 segundos, passando pelo valor  $-/+$  infinito (menos e mais infinito). Quando o reator está na condição de regime estacionário, o mostrador indica o valor infinito. [15]

Durante a partida do reator, e até que a potência do mesmo alcance 500 KW (no caso 10% da potência nominal), o processo de monitoração do fluxo neutrônico é feito através do canal de faixa ampla, que neste caso gera sinais para desligamento do reator nas seguintes condições: [15]

- Perda da alta tensão que alimenta o detector;
- Indicação de taxa de contagem abaixo de 5 cps;
- Período do reator abaixo de 12 segundos; e
- Canal fora de operação (para calibração, manutenção ou testes).

Para potências acima de 200 KW, o canal de faixa ampla provoca o desligamento do reator se a vazão no circuito primário estiver abaixo de 90% da vazão nominal, ou se houver falha de energia no barramento que alimenta as bombas do circuito primário.

Os canais de segurança, num total de 3 permitem, através de um sinal linear, monitorar a potência do reator na faixa de 1 a 110% da potência nominal. O canal de segurança 1 (Safety 1) é

composto por um circuito eletrônico acoplado à câmara de fissão, em paralelo com o canal de faixa ampla, onde a partir do nível de 1% da potência nominal (50KW), este detector se comporta como uma câmara de ionização, fornecendo corrente no sinal de saída.

Já os canais de segurança 2 (Safety 2) e 3 (Safety 3) são dois canais totalmente independentes, ligados a detectores do tipo câmara de ionização não compensada com depósito de  $^{10}\text{B}$  em seu interior. Os três canais de segurança são considerados amplificadores de corrente de alta confiabilidade, construídos totalmente com componentes de estado sólido. Utilizando uma lógica 2 em 3, os canais de segurança geram sinais para desligamento do reator (SCRAM) nas seguintes condições:

- Perda da alta tensão que alimenta o detector;
- Potência do reator acima de 105% do valor nominal;
- Canal fora de operação (para calibração, manutenção ou testes).

A geração do sinal para desligamento do reator é feita através da interrupção da corrente elétrica que energiza os magnetos que acoplam as barras de controle/segurança aos respectivos mecanismos de movimentação.

Os sinais de desligamento correspondentes ao baixo período e baixa taxa de contagem do canal de faixa ampla são bloqueados quando 2 dos 3 canais de segurança estiverem operacionais, isto é, não estão desativados para atividades de calibração, manutenção ou testes, e indicarem que a potência do reator é superior a 10% da potência nominal, e que a vazão no circuito primário é superior a 90% do valor nominal.

Da mesma forma que no caso do canal linear, a incerteza da medição dos canais nucleares é da ordem de 5%, sendo 4% devido ao processo de calibração, mais 1% devido à incerteza da leitura do mostrador.[18]

### **3.2.2 Balanço Térmico**

Balanço Térmico é a técnica utilizada para calibração dos detectores de potência do reator. Esta técnica baseia-se no acréscimo de temperatura sofrido pelo fluido de refrigeração ao passar pelo núcleo do reator, sendo a Potência determinada pela equação (1):

$$P = \dot{M} \cdot C_p \cdot \Delta t \quad (1)$$

**Onde:**

$\dot{M}$  = Vazão do fluido refrigerante através do núcleo do reator

$C_p$  = Calor específico do fluido refrigerante

$\Delta t$  = Variação da temperatura do fluido refrigerante ao passar pelo núcleo do reator  
(temperatura de saída - temperatura de entrada)

Os valores de temperatura são obtidos através do subsistema de medição de temperatura que é formado por um conjunto de 24 termopares distribuídos em vários pontos, dentro e fora do prédio do reator. Estes termopares estão ligados a um registrador, localizado no painel existente na sala de controle, onde são registrados continuamente a cada ciclo de 2,5 minutos, a temperatura dos 24 pontos. Uma das funções do subsistema de medição de temperatura é registrar o incremento de temperatura da água ao passar pelo núcleo do reator, que é indicado pela diferença entre os valores de T3 (localizado a 30 centímetros acima do núcleo) e T4 (localizado na tubulação de saída do fluido refrigerante), medidos por termopares do tipo K

(cromel-alumel), que são os mais resistentes à radiação. Todos os demais termopares são do tipo T (Cobre-Constantan).

A vazão do fluido refrigerante é determinada pelo subsistema de medição de vazão e nível, cuja principal função é permitir ao operador monitorar o nível de água na piscina do reator e nas torres de resfriamento, e a vazão nos sistemas de resfriamento do reator, primário e secundário. Os sinais do subsistema de medição de vazão e nível são utilizados pelo sistema de proteção para evitar que a potência do reator seja elevada acima de 200 KW se a vazão no circuito primário for inferior a 90% do valor nominal. [15]

Conforme mostrado na referência [18], a incerteza na medida da potência térmica é da ordem de 4% para o reator a 5MW, quando a diferença entre a temperatura da água na saída do núcleo e a temperatura da água na entrada do mesmo é da ordem de 6°C. Para potências menores a diferença pode ser maior, já que o critério de segurança estabelecido é que a potência do reator, não pode ultrapassar 5,25 MW. Isto significa que quando o reator estiver a 5 MW a diferença entre o valor de potência fornecido pelos canais nucleares e o valor obtido com o balanço térmico deve ser inferior a 1%. Para potências menores a diferença pode ser maior, já que, como mencionado acima, o critério de segurança estabelecido é que a potência do reator, não pode ultrapassar 5,25 MW. A experiência tem mostrado uma diferença entre 8 e 10 %, entre a potência térmica e a potência indicada pelos canais nucleares, quando os mesmos indicam 2MW [18].

### 3.2.3 Monitoração da Atividade do Nitrogênio 16

A formação do  $^{16}\text{N}$  é devida à reação  $^{16}\text{O} (n,p) ^{16}\text{N}$ , quando a água passa no núcleo do reator. Devido ao alto limiar de energia da reação (11 MeV), a formação do  $^{16}\text{N}$  depende basicamente do fluxo de nêutrons de fissão, (com energia acima de 11 MeV) e portanto da potência do reator. O  $^{16}\text{N}$  tem meia vida de 7,2 segundos e se desintegra emitindo betas e gamas, sendo 68% dos gamas com energia igual a 6,14 MeV. A atividade em  $^{16}\text{N}$  representa 99% da atividade total da água que passa pelo núcleo do reator (a atividade restante é devida à ativação de impurezas, produtos de corrosão, materiais estruturais e produtos de fissão de reações menos importantes que formam  $^{17}\text{N}$ ,  $^{19}\text{O}$ ,  $^{41}\text{A}$ ,  $^{27}\text{Mg}$ ,  $^{28}\text{Al}$  e  $^{24}\text{Na}$ ) etc., e a sua monitoração resulta numa medida indireta da potência do reator.

Um monitor de potência através da atividade do  $\text{N}^{16}$ , consiste em um detector de radiação gama que monitora a água de refrigeração, após a saída do núcleo, em uma posição determinada cujo tempo para atingi-lo seja suficientemente curto com relação ao tempo de decaimento da atividade do  $^{16}\text{N}$ . No caso do reator IEA-R1 o canal do  $^{16}\text{N}$  utiliza como detector uma câmara de ionização não compensada (CINC), sensível à radiação gama localizada na casa de máquinas entre a saída da piscina do reator e o tanque de decaimento de  $^{16}\text{N}$ . Acoplado ao detector existe um picoamperímetro semelhante ao do canal linear. [15]

Sendo o detector uma câmara de ionização, a leitura do monitor é expressa matematicamente pela equação (2): [17]

$$A = K \cdot P (1 - e^{-\lambda t_1}) e^{-\lambda t_2} \quad (2)$$

Onde:

A = corrente da câmara de ionização

$K$ = constante de proporcionalidade de potência, que combina a secção de choque de ativação com a sensibilidade e geometria do detector.

$P$ = potência do reator (megawatts)

$\lambda$ = constante de desintegração do  $N^{16}$  ( $s^{-1}$ )

$t_1$ = tempo para um segmento de água atravessar o núcleo

$t_2$  = tempo para um segmento de água se deslocar do núcleo ao detector.

Como os fatores  $t_1$  e  $t_2$ , dependem da vazão do fluido refrigerante, devemos realizar a medida quando a vazão for constante, condição na qual é possível obter uma proporcionalidade direta entre a leitura do  $N^{16}$  e a potência do reator.

Como a ativação de outros materiais ( $Al^{28}$ ,  $Mg^{27}$ ,  $Na^{24}$  e  $Ar^{41}$ ), influem na monitoração da potência da atividade do  $N^{16}$ , já que esses isótopos possuem meia vida grande em relação à do  $N^{16}$  e podem ser acumulados em diversos ciclos de operação, para que a medida não represente a atividade de outros nuclídeos, que não o  $N^{16}$ , a câmara de ionização é calibrada para a energia emitida pelo  $N^{16}$ . [17]

Pelos mesmos motivos já explicados para o caso dos Canais Nucleares, a incerteza da medição da monitoração do  $N^{16}$  é da ordem de 5%, e assim como o canal linear, o canal de  $^{16}N$  também não gera sinais para desligamento do reator.

### 3.2.4 Vantagens e Desvantagens de cada Método Utilizado para Monitorar a Potência do Reator IEA-R1

A tabela 3.1 apresenta um resumo sobre as vantagens e desvantagens de cada método utilizado para monitorar a potência do reator IEA-R1. No caso da determinação de potência pela técnica de balanço térmico a única desvantagem é que o método requer um tempo muito grande, uma vez que o sistema apresenta uma inércia térmica muito grande, já que é composto pela estrutura da piscina, pelas tubulações e válvulas do circuito primário de refrigeração, pelo tanque de decaimento e pelo trocador de calor. No caso dos canais nucleares, a desvantagem é que com o tempo o material sensível a neutrons existente dentro dos detectores vai sendo consumido, diminuindo a sensibilidade dos mesmos e, conseqüentemente, perdendo a calibração.

<i>Método</i>	<i>Vantagem</i>	<i>Desvantagem</i>
<u>Balanço Térmico</u>	Melhor precisão e estabilidade.	Tempo de resposta muito lento, devido à inércia térmica do sistema.
<u>Canal Nuclear</u>	Resposta rápida ( considerada instantânea em alguns casos).	Menor precisão e perde a calibração na medida em que é utilizado.
<u>Nitrogênio -16</u>	Resposta rápida ( apesar de ser mais lenta do que a dos canais nucleares)	Menor precisão, porém não perde a calibração.

Tabela 3.1 – Vantagens e desvantagens dos métodos utilizados para monitorar a potência do reator IEA-R1.

## **4. Aspectos Históricos e Teóricos das Redes Neurais**

### **4.1 Introdução**

Redes Neurais é o nome que se dá a um sistema de processamento computacional que utiliza uma estrutura semelhante à rede de neurônios de um cérebro humano. A origem da teoria remonta aos modelos matemáticos e aos modelos de engenharia de neurônios biológicos. É também identificada como subespecialidade da inteligência artificial sempre visando o estudo de fenômenos complexos.[19]

O neurônio, é o principal componente do sistema nervoso. Considerada sua unidade anatomo-fisiológica, estima-se que no cérebro humano existam aproximadamente 15 bilhões destas células, responsável por todas as funções do sistema.

Existem diversos tipos de neurônios, com diferentes funções dependendo da sua localização e estrutura morfológica, mas em geral os neurônios são divididos em três seções: o corpo celular (soma), os dendritos e o axônio, como mostra a figura 12 a seguir:

- O corpo celular (soma) é a região onde as informações são processadas e novos impulsos são gerados.
- O axônio é a parte do neurônio responsável pela condução do impulso nervoso para o próximo neurônio.
- Os dendritos, são pequenos prolongamentos em forma de ramificações (arborizações terminais), e têm por função receber as informações, ou impulsos nervosos, de outros

neurônios conduzindo os mesmos até o corpo celular. O ponto de contato entre a terminação axônica de um neurônio e o dendrito de outro neurônio é denominado sinapse.

[20]

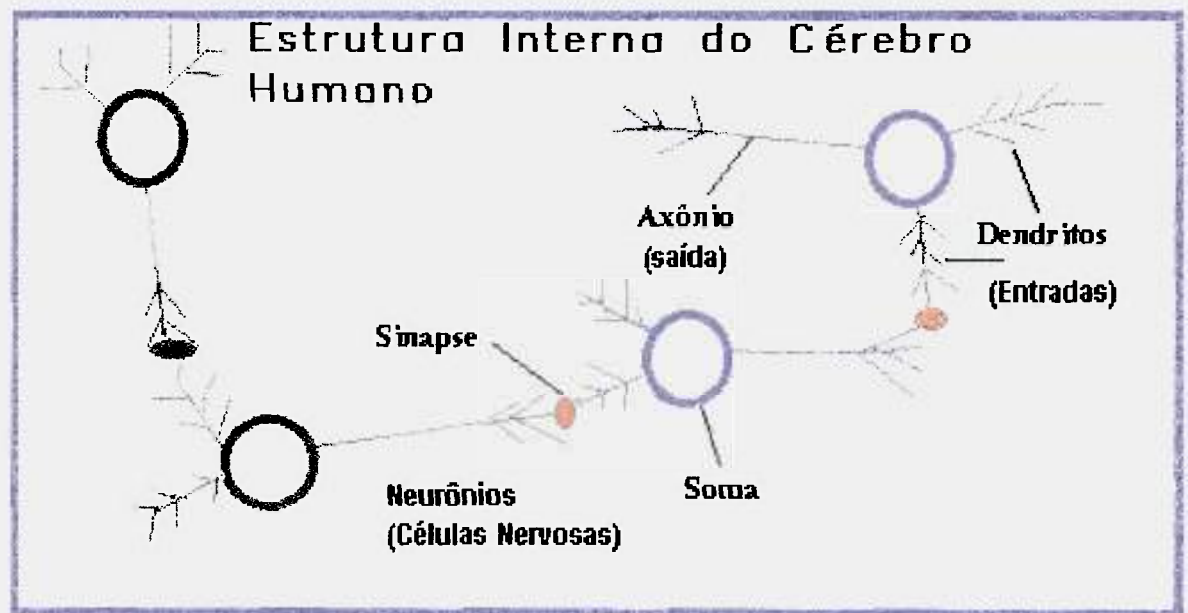


Figura 12 - Estrutura interna do cérebro humano.

A sinapse é o ponto da estrutura de uma rede de neurônios através da qual ocorrem os processos de comunicação entre os mesmos, ou seja, onde ocorre a passagem do sinal neural (transmissão sináptica) por meio de processos eletroquímicos específicos, isso graças a certas características particulares da sua constituição.

As entradas dos dados, sinais elétricos gerados nos neuro-sensores (retina ocular, papilas gustativas etc.) ocorrem através das conexões sinápticas, que conectam a árvore dendrital aos axônios de outras células nervosas através de um espaço sináptico.

As sinapses são basicamente regiões eletroquimicamente ativas, compreendidas entre duas membranas celulares: a membrana pré-sináptica, axônio por onde chega o estímulo proveniente de uma outra célula, e a membrana pós-sináptica, que é a do dendrito. [19] Nesta região intersináptica, o estímulo nervoso que chega a sinapse é transferido à membrana dendrital através de substâncias conhecidas como neurotransmissores ocasionando ou não uma alteração no potencial elétrico da membrana pós-sináptica. Conforme o neurotransmissor a conexão sináptica poderá ser excitatória ou inibitória. Se for excitatória, haverá uma alteração no potencial da membrana formando um impulso nervoso no axônio, caso contrário nada ocorrerá. [19]

Cada região do cérebro é especializado em uma dada função, como processamento de sinais auditivos, sonoros, elaboração de pensamentos, desejos etc. Esse processamento se dá através de redes particulares, onde cada região do cérebro possui uma arquitetura de rede diferente, variando o número de neurônios, de sinapses por neurônio, etc. [21]

## 4.2. Aspectos Históricos

### 4.2.1 Neurônio de McCulloch

Em 1943 o neurologista Warren McCulloch e o matemático Walter Pitts, propuseram o primeiro modelo artificial de um neurônio biológico denominado neurônio de McCulloch. Era um dispositivo binário, no qual a sua saída  $Y$  poderia ser pulso ou não pulso, e as suas entradas  $X$  tinham ganho arbitrário  $w$  e poderiam ser excitatórias ou inibitórias. Para determinar a saída do neurônio, calculava-se a soma ponderada das entradas  $X$  com os seus respectivos ganhos como fatores de ponderação  $w$ , positivos nos casos excitatórios e negativos nos casos inibitórios. Caso o resultado fosse maior ou igual a um certo limiar então a saída do neurônio gerava um pulso, e caso contrário era não pulso. [19]

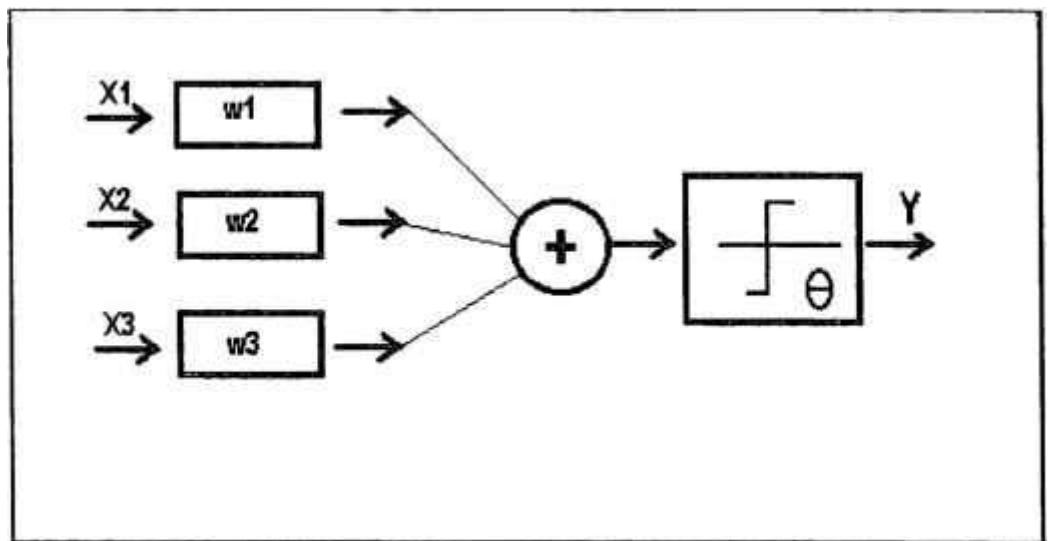


Figura 13 - Neurônio de McCulloch

O neurônio deste modelo atua como uma unidade de decisão binária, que é ativado quando a soma ponderada de suas entradas ( $X$ ) pelos ganhos excede a um certo limiar  $\theta$  (threshold), produzindo saídas correspondentes a 0 ou 1. O valor  $\theta$ , pode ser interpretado como uma polarização, correspondendo a uma entrada com valor fixo, com sua conexão de peso correspondente igual a 1. A equação (3) mostra a formulação matemática utilizada para representar o neurônio de McCulloch:

$$Y = F(\sum w_i X_i - \theta) \quad (3)$$

Onde:

$$F(x) = 1 \text{ se } x \geq 0 \text{ e } F(x) = 0 \text{ se } x < 0$$

$w$  representam os ganhos associados as entradas

$X$  são as entradas

$\theta$  limiar

$Y$  é a saída

#### **4.2.2 Perceptrons**

Em 1958 Frank Rosenblatt deu prosseguimento às idéias de McCulloch. Ele criou uma rede de múltiplos neurônios do tipo discriminadores lineares e chamou de Perceptron, com uma topologia, representada conforme figura 14, batizado de Mark 1, de 400 fotocélulas arranjadas numa matriz de 20 X 20 pixels, que representavam o vetor de entrada e que servia para o reconhecimento de caracteres grafados nesta matriz. Sendo assim, se a função do Perceptron fosse reconhecer o caractere "A", a resposta de saída da rede deveria ser "1", caso contrário "0".

[19]

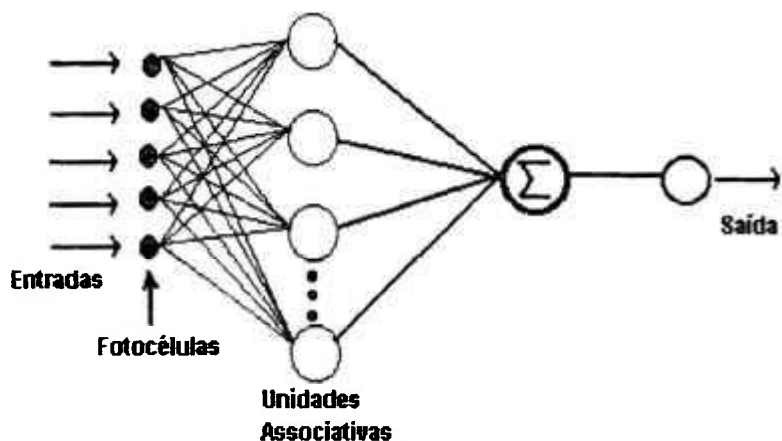


Figura 14 – Perceptron de Rosenblatt

A topologia original descrita por Rosenblatt era composta por unidades de entradas (fotocélulas), por um nível intermediário formado pelas unidades associativas e por um nível de saída (unidade de resposta). Embora a rede apresentasse três níveis, ela foi identificada como Perceptron de apenas uma única camada, pois apenas o nível de saída apresenta propriedades adaptativas. As fotocélulas são unidades sensoras, e as unidades intermediárias associativas, embora sejam neurônios, possuem pesos fixos, pré-estabelecidos antes do treinamento. [20]

Rosenblatt também propôs um algoritmo de treinamento da rede, baseado no estudo do biólogo Donald Hebb (1949), pelo ajuste gradual dos pesos de um discriminador linear. O Perceptron é treinado para fornecer saídas de acordo com os dados do conjunto de treinamento, como mostra as equações (4), (5) e (6). [19]

$$W_{\text{novo}} = W_{\text{velho}} + \Delta W \quad (4)$$

$$W_{\text{novo}} = W_{\text{velho}} + \eta \cdot e \cdot X \quad (5)$$

$$e = Y_d - Y_r \quad (6)$$

Onde:

$\eta$  é taxa de aprendizado da rede

$W_{\text{velho}}$  é o peso anterior

$W_{\text{novo}}$  é o peso novo

$\Delta W$  correção do peso

$X$  entrada do neurônio

$e$  erro da resposta da rede

$Y_d$  é a saída desejada

$Y_r$  é a saída da rede

### **4.2.3 Adaline e Madaline**

Contemporâneos de Rosenblatt, Widrow e Hoff desenvolveram em 1960, um modelo neural linear denominado Adaline (Adaptive Linear Element) ou elemento linear adaptivo. Com ele definiram, em 1962, uma das primeiras redes construídas com elementos adaptativos, o Madaline (Multiple-Adaline).

O modelo Adaline apresenta saídas binárias bipolares, ou seja, saídas ( $Y$ ) que pertencem ao domínio  $[-1, 1]$ . Seus pesos são ajustados em função do erro da diferença entre sua saída linear  $Y_l$  e sua saída desejada  $Y_d$ , antes da aplicação da função de ativação, como mostrado na figura 15.

A saída linear é mostrada nas equações (7) e (8). [19,20]

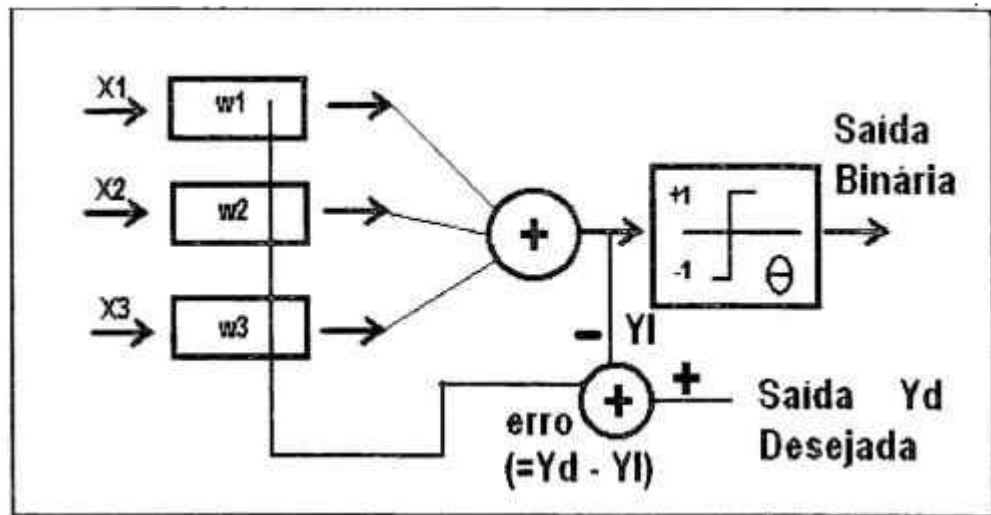


Figura 15 – Modelo de um neurônio Adaline

$$Yl = \sum w \cdot X \quad (7)$$

$$e = Yd - Yl \quad (8)$$

Onde:

$Yl$  é a saída linear

$w$  pesos

$X$  entradas

$Yd$  é a saída desejada

$e$  é o erro obtido com relação à saída linear

Adicionalmente, Widrow e Hoff desenvolveram um algoritmo de aprendizado de grande importância na área das redes neurais, utilizando o conceito de minimização do desvio quadrático

médio (Least Mean Square algorithm - LMS). A técnica, que ficou conhecida como regra delta ou método do gradiente decrescente para a minimização do erro, será comentada no item 4.4.3.

#### 4.2.4 As Dificuldades do Perceptron

Em 1969, Minsky e Papert lançaram o polêmico livro *Perceptrons*, onde foi demonstrado matematicamente que o teorema desenvolvido por Rosenblatt só era aplicável à classe de problemas linearmente separáveis, não sendo capaz de resolver problemas elementares como o OU – Exclusivo e o seu complemento ilustrado na figura 16. Ainda neste mesmo trabalho, Minsky e Papert afirmaram que não havia motivo aparente para acreditar que redes formadas por múltiplas camadas de Perceptrons, pudessem herdar as boas qualidades do Perceptron simples. Estas colocações desestimularam os pesquisadores que atuavam na área, fazendo com que as pesquisas sobre Redes Neurais sofressem um grande desinteresse durante a década de 70 e início dos anos 80.

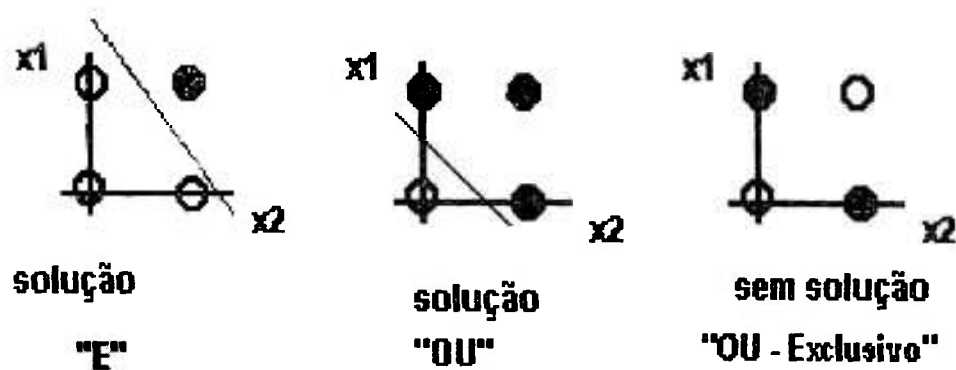


Figura 16 -Funções Booleanas de duas variáveis representadas no plano binário

#### **4.2.5 Aprendizado Competitivo e Algoritmo da Retropropagação**

Apesar das poucas atividades de pesquisa a partir deste período, alguns trabalhos tiveram destaque, especialmente na pesquisa das redes auto-organizáveis utilizando algoritmos competitivos, como foi o caso do trabalho de Kohonen (1972) [23] que desenvolveu modelos de memórias associativas, utilizando o conceito de aprendizado competitivo, nos quais as unidades competem para responder a determinada entrada, e o elemento vencedor tem os pesos de sua entrada modificados, convergindo para responder com mais força a valores próximos do desejado. [22]

Entre 1967 a 1988 Grossberg apresentou contribuições importantes, baseado em seus trabalhos prévios sobre o aprendizado competitivo, em 1987 estabeleceu os princípios para uma nova classe de redes neurais junto com Carpenter denominadas de ART ou Adaptive Resonance Theory. [23]

Em 1982 Hopfield utilizou a idéia de uma função de energia, para um novo modo de funcionamento das redes recorrentes com conexões sinápticas simétricas, onde os elementos são ligados buscando o aprendizado com “um mínimo de energia”, tendo dado origem às Redes de Hopfield.

Somente em 1986 tiveram reinício, em grande escala, as atividades de desenvolvimento das redes neurais artificiais, com o desenvolvimento do algoritmo de retropropagação (backpropagation) por Rumelhart, Hinton e Williams, embora este algoritmo já tivesse sido proposto anteriormente em 1974 por Werbos em sua tese de doutorado, por Parker e LeCun em 1985. [29] Com o lançamento do livro denominado *Parallel Distributed Processing: Explorations*

in the Microstructures of Cognition, editado por Rumelhart e McClelland, o qual apresentava de uma forma organizada todo o progresso das redes neurais, ressurgiu o grande interesse pela técnica. [22]

### **4.3. Arquiteturas das Redes Neurais Artificiais**

As redes neurais artificiais procuram explorar os princípios adotados pelo cérebro humano, apresentando um processamento altamente paralelo que ocorre em sua estrutura, além de uma capacidade de generalizar, ou seja, uma capacidade de gerar respostas adequadas, para entradas que nunca foram apresentadas durante o treinamento. Estes dois aspectos fazem com que as redes neurais sejam capazes de solucionar problemas altamente complexos. [29]

Atualmente as principais aplicações de modelos utilizando redes neurais são: [27]

- Classificação, onde a entrada representa um elemento de uma classe desconhecida, dentre várias existentes, onde a rede produzirá uma saída determinando a classe correta.
- Complementação de padrões, cuja entrada representa um certo padrão incompleto, como por exemplo, uma letra A sem o traço do meio, a rede fornecerá uma saída com o padrão completo corrigido.
- Controle, onde a entrada representa as características do estado atual e a resposta desejada de um controlador, a rede irá fornecer uma saída com uma sequência de controle.
- Aproximação de funções, onde uma rede neural fornece um resultado dentro de um intervalo contínuo, com base nas informações disponíveis na entrada.

No modelo do neurônio artificial da figura 17, as entradas formadas por  $X_1, X_2 \dots X_n$ , são ponderadas pelos pesos  $W_1, W_2 \dots W_n$ , para produzir um estado de ativação no neurônio, através da função  $f()$ , chamada também de, “função de ativação do neurônio”. Esta função é responsável pela geração do sinal de saída do neurônio. Um valor auxiliar  $\theta$  é geralmente utilizado para representar a existência de um limiar de excitação do neurônio. Os pesos representam o padrão de conectividade da rede, ou seja, define como os neurônios estão conectados entre si. As funções de ativação, mais utilizadas nos neurônios são do tipo linear, degrau, sigmóide e tangente hiperbólica, conforme ilustrado figura 18. [24]

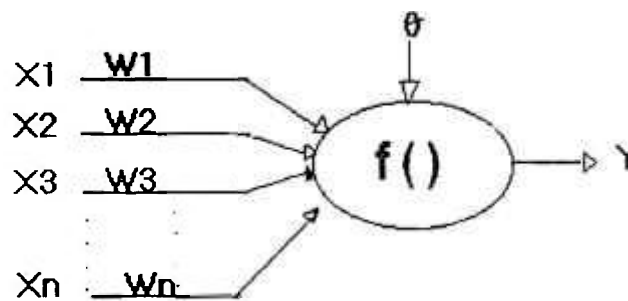
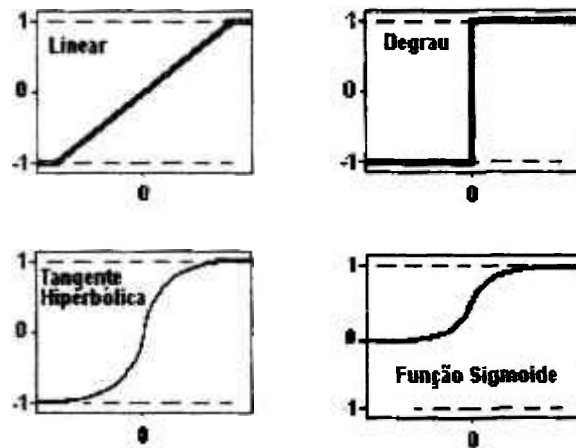


Figura 17 - Modelo Matemático de um Neurônio Artificial

O sinal de saída do neurônio é dado pela equação (9):

$$Y = f(\sum X_n \cdot W_n - \theta) \quad (9)$$



**Figura 18 – Exemplos de Funções de Ativação**

Uma Rede Neural é representada como um arranjo de neurônios, conforme exemplo apresentado na figura 19. Neste caso o modelo é baseado em camadas, embora existam redes que não apresentem este tipo de estrutura. A forma da topologia da rede utilizada neste trabalho é organizada em camadas. Existem basicamente três tipos de camadas, a saber, entrada, intermediária e saída:

- A camada de entrada, é constituída por neurônios de entrada que recebem excitação do mundo exterior. Geralmente uma rede neural apresenta apenas uma camada de entrada, com uma quantidade de neurônios igual ao número de sinais que “entram” na rede. Neste caso, cada neurônio apresenta apenas um sinal de entrada.
- A camada intermediária, também conhecida como “Camada Oculta” ou “Hidden”, representa o conjunto de neurônios onde os vários sinais de entrada são processados.

Numa rede genérica podemos ter mais de uma camada intermediária, porém na maior parte dos casos práticos se utiliza apenas uma.

- A camada de saída é constituída por neurônios que geram os sinais de resposta da rede à excitação das entradas. Estas respostas são usadas para gerar os sinais de resposta do sistema.

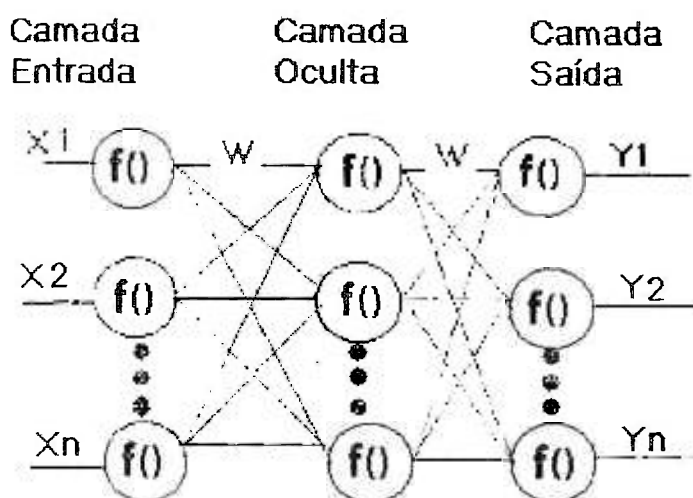


Figura 19 – Rede Neural com topologia em camadas

A figura 19 mostra uma estrutura clássica das redes neurais do tipo Multi Layer Perceptron (MLP), onde X é entrada, W são os pesos,  $f()$  função de transferência e Y a saída.

Para se construir uma rede neural deve-se determinar as propriedades da rede, a topologia (ou conectividade), o tipo de conexões, faixa dinâmica dos pesos e inicialização dos mesmos; determinar as propriedades dos neurônios, função do neurônio (ou função de ativação) e a regra de aprendizado. [22]

O padrão de conectividade de uma rede neural define como os neurônios estão conectados entre si, e em geral é representado pela matriz de conectividade  $W$  ou matriz de pesos. A matriz de pesos é ajustada durante a fase de aprendizado da rede. Os pesos podem ser números reais ou inteiros, e podem ou não ser confinados em uma determinada faixa. A forma de inicialização dos pesos, depende do tipo de rede utilizada, porém geralmente esta inicialização dos pesos é feita com números pequenos e aleatórios. Após o treinamento, em geral, os valores dos pesos não são mais alterados.

De uma forma geral, na literatura, é citado que os pesos são os responsáveis pelo armazenamento do conhecimento da rede. Esta afirmação não é errônea, porém existem arquiteturas nas quais os neurônios, por meio de sua função de ativação também armazenam parte do conhecimento da rede, como por exemplo, a rede do tipo RBF, partilhando esta responsabilidade com os pesos das conexões. [22]

A seguir serão apresentados os principais algoritmos de aprendizado.

#### **4.4 Principais Algoritmos de Aprendizado**

A regra de aprendizado é uma das características mais importantes de uma rede neural artificial. Pois através do aprendizado, será determinada a forma de atualização dos pesos das conexões entre os neurônios durante o treinamento, de forma a otimizar o desempenho da rede.

Do ponto de vista de aprendizado, as redes podem ser apresentadas como sendo de treinamento supervisionado ou não supervisionado. O aprendizado supervisionado é quando a

entrada e a saída desejada pela rede é fornecida por um supervisor (professor) externo. No treinamento não supervisionado, não há um professor, por sua vez, consiste na apresentação apenas dos dados de entrada para a rede, ficando a cargo desta a determinação das saídas existentes. [20] Em ambos os casos o processo de aprendizado consiste em modificar o valor dos pesos, de forma que, o sinal de saída da rede convirja para o valor desejado.

Existem várias regras de aprendizagem que podem ser vistas na referência [29]. As principais regras de aprendizagem descritas neste trabalho são: aprendizagem Hebbiana, aprendizagem por Correção de Erro e aprendizagem pelo Método do Gradiente Descendente.

#### **4.4.1 Aprendizagem Hebbiana**

A técnica de aprendizagem Hebbiana é considerada a mais antiga de todas as regras. [29] Segundo essa técnica, as ligações sinápticas são influenciadas de forma a aumentar a eficiência de ligação, ou seja, Hebb propõe que o peso de uma conexão sináptica deve ser ajustado se houver sincronismo entre os níveis de atividade das entradas e saídas. [20] Se dois neurônios em ambos os lados de uma sinapse são ativados simultaneamente teremos um fortalecimento da sinapse. Caso contrário, se os dois neurônios são ativados assincronamente teremos um enfraquecimento ou eliminação da sinapse. Matematicamente, no aprendizado Hebbiano os pesos das ligações são corrigidos a partir da equação (10):

$$\Delta W_{ij} = \eta \cdot Y_i(t) X_j(t) \quad (10)$$

Onde  $\eta$  é uma constante que expressa a taxa de aprendizado,  $X_i$  as entradas e  $Y_i$  as saídas da rede.

#### **4.4.2 Aprendizagem por Correção de Erro**

O processo de aprendizagem por correção de erro tem como objetivo minimizar a diferença entre a saída calculada pela rede e a saída desejada. Esta diferença  $e(t)$ , é o erro da resposta gerada pela rede, como mostra a equação genérica (11). [20]

$$e(t) = Y_d(t) - Y(t) \quad (11)$$

Onde:

$Y_d(t)$  é a saída desejada

$Y(t)$  é a resposta da saída atual calculada pela rede no instante  $t$

A equação (12) mostra a forma genérica de ajuste de pesos da rede por Correção de Erros e onde o ajuste de pesos deve ser proporcional ao produto do erro pelo valor de entrada da sinapse naquele instante de tempo. [20]

$$W_i(t+1) = W_i(t) + \eta \cdot e(t) \cdot X_i(t) \quad (12)$$

Onde:

$\eta$  é taxa de aprendizado da rede

$W_i(t)$  peso antigo

$W_i(t+1)$  peso novo

$X_i$  entrada do neurônio

$e(t)$  erro da resposta atual da rede

#### 4.4.3 Regra Delta ou Método do Gradiente Decrescente

O método do gradiente decrescente, conhecido também como regra delta ou regra de Widrow e Hoff é uma generalização da técnica de correção de erros, onde a função de custo a ser minimizada é uma função de erro ou energia, definida pela soma dos erros quadráticos definida na equação (13).

$$E(w) = \frac{1}{2} \sum_p \sum_{k=1}^k (Y_d(k) - Y(k))^2 \quad (13)$$

Onde E, representa a medida de erro, p é o número de padrões, k a número de unidades de saída, Y(k) saídas geradas pela rede e Y<sub>d</sub>(k) saídas desejadas. A equação (13) define o erro total apresentado pela rede, ou a quantidade que, para todos padrões da rede, as saídas geradas pela rede Y diferem das saídas desejadas Y<sub>d</sub>. [20]

A regra delta utiliza a variação dos pesos com relação ao gradiente decrescente do erro com relação ao peso, pois o gradiente aponta na direção de maior crescimento da mesma, conseqüentemente “o negativo” do gradiente, aponta na direção contrária, resultando na taxa de diminuição. A regra delta é definida conforme a equação (14):

$$W(\text{novo}) = W(\text{velho}) - \eta \cdot \frac{\partial E}{\partial W} \quad (14)$$

Onde, W são os pesos, η determina o passo que se dará naquela direção (taxa de aprendizagem) e  $\frac{\partial E}{\partial W}$  o gradiente. [19]

Esta Regra será posteriormente generalizada para o algoritmo de Retropropagação item 4.4.4.

#### **4.4.4 Algoritmo da Retropropagação**

O algoritmo de aprendizado mais conhecido para redes neurais do tipo Multi Layer Perceptron (MLP), a serem descritas no item 4.5, é o backpropagation ou da retropropagação, ou propagação-para-trás. Este algoritmo foi um dos principais responsáveis pelo ressurgimento das redes neurais artificiais, conforme comentado no item 4.2.5.

O algoritmo backpropagation é um algoritmo supervisionado que utiliza pares (entrada, saída desejada) para, por meio de um mecanismo de correção, ajustar os pesos da rede. O treinamento ocorre em duas fases distintas conhecidas como forward e backward. Na fase forward onde a rede é utilizada para definir a saída da rede para um dado padrão de entrada, e na fase backward a diferença entre a saída desejada e a saída fornecida é utilizada pela rede para atualizar os pesos de suas conexões. A figura 20 ilustra a dinâmica do método. [20]

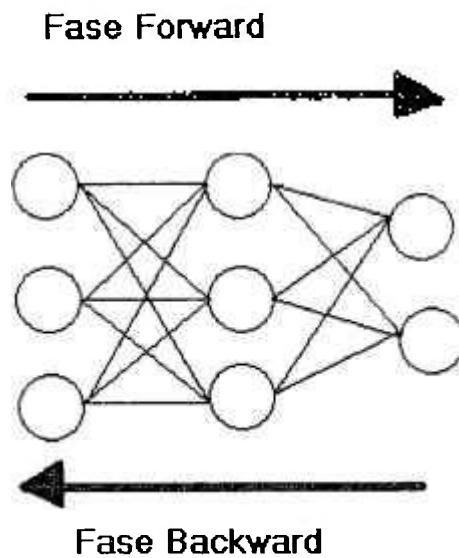


Figura 20 - Fluxo de Processamento do Algoritmo Backpropagation

A implantação computacional do algoritmo de retropropagação apresenta os seguintes passos: [20]

Passo 1: Inicialização de todos os pesos e parâmetros.

Passo 2: Fase Forward, consiste em:

- Calcular todas as entradas e saídas, para todas as camadas da rede
- Calcular o erro de saída da rede

Passo 3: Fase Backward, consiste em:

- Efetuar o cálculo das atualizações dos pesos entre as camadas da rede. Iniciar a partir da última camada até chegar a camada de entrada.

O algoritmo de backpropagation consiste na utilização dos pesos, considerando a propagação do erro da saída da rede para a sua entrada. Este algoritmo, baseado na regra delta proposta por Widrow e Hoff, é também conhecido como regra delta generalizada. Este algoritmo propõe uma forma de definir os erros dos neurônios das camadas intermediárias, possibilitando o ajuste de seus pesos.

Embora o erro total  $E$  seja definido pela soma dos erros dos neurônios de saída para todos os padrões, conforme citado na regra delta no item 4.4.3, será considerado sem perda de generalidade, que a minimização do erro para cada padrão individualmente levará à minimização do erro total para todos os padrões. Portanto, o erro passa a ser definido conforme a equação (15). [20]

$$E(\mathbf{w}) = \frac{1}{2} \cdot \sum_{k=1}^k (Y_d(k) - Y(k))^2 \quad (15)$$

Onde  $E$ , representa a medida de erro cometido pela rede para cada padrão individualmente, ou seja, o quanto as saídas geradas pela rede  $Y(k)$  diferem das saídas desejadas  $Y_d(k)$ , sendo “ $k$ ” o número de saídas da rede.

Como na regra delta a variação dos pesos é definida de acordo com o gradiente decrescente, conforme comentado no item 4.4.3, partindo da equação (16) é calculada matematicamente a equação (17) para a regra delta generalizada:

$$W(\text{novo}) = W(\text{velho}) - \eta \cdot \frac{\partial E}{\partial W} \quad (16)$$

$$\mathbf{W}(\text{novo}) = \mathbf{W}(\text{velho}) - \eta \cdot \delta_j \cdot \mathbf{X}_i \quad (17)$$

Onde, para um neurônio de saída, a variável  $\delta_j$  (erro) é calculada conforme a equação (18):

$$\delta_j = (\mathbf{Y}_{dj} - \mathbf{Y}_j) \cdot f'(\mathbf{u}_j) \quad (18)$$

$$\text{Onde } \mathbf{u}_j = \sum_{i=1}^m \mathbf{x}_i \cdot \mathbf{w}_{ij}$$

Para os demais neurônios, diferentes dos de saída, a variável  $\delta_j$  (erro) é calculada conforme a equação (19):

$$\delta_j = f'(\mathbf{u}_j) \sum_l \delta_l \mathbf{W}_{jl} \quad (19)$$

$\mathbf{W}$  valor dos pesos

$f'()$  é a derivada da função de ativação do neurônio

O termo  $\delta_j$ , corresponde a derivada da função de erro com relação aos pesos, sendo responsável pela propagação do erro da saída da rede para as suas camadas intermediárias, através de um cálculo recursivo.

$\mathbf{Y}_{dj}$  é a saída desejada pela rede

$\mathbf{Y}_j$  é a saída fornecida pela rede

#### 4.5 Redes Neurais do Tipo Multi Layer Perceptron (MLP)

As redes do tipo MLP enquadram-se na classe das redes de multicamadas, sendo formada geralmente por uma camada de entrada, uma ou várias camadas ocultas (intermediárias) e uma camada de saída. Devido ao uso de uma ou mais camadas intermediárias, pode-se trabalhar com dados que não são linearmente separáveis. [20] Portanto este tipo de rede tem sido bastante utilizado em modelos e sistemas de identificação não lineares. O algoritmo das redes do tipo MLP foi desenvolvido originalmente para redes de neurônios com funções de ativação do tipo semilineares e portanto diferenciáveis. [19] As funções precisam ser diferenciáveis para que o gradiente possa ser calculado, para ajuste dos pesos. A função de ativação mais utilizada pela rede MLP é a do tipo função sigmoideal, mostrada na figura 18.

A figura 21, mostra a arquitetura de uma rede neural do tipo MLP, como a utilizada neste trabalho, onde  $X$  representa os sinais entrada,  $W$  os respectivos pesos,  $f()$  a função de ativação e  $Y$  o sinal de saída da rede.

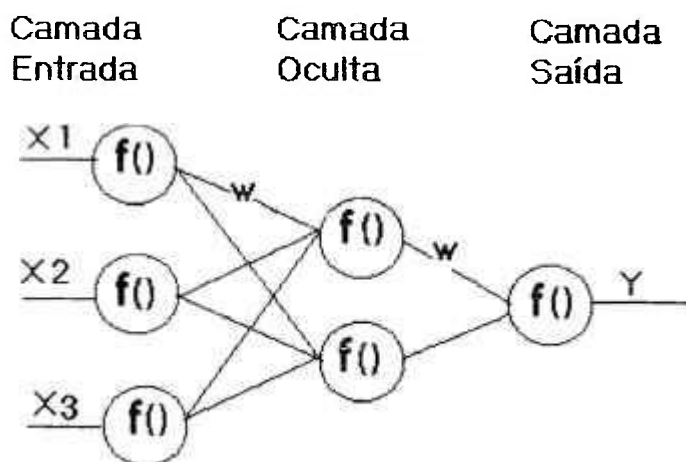


Figura 21 - Rede Neural do Tipo MLP

■ As redes do tipo MLP, na sua forma clássica, utilizam uma estrutura composta por três camadas de neurônios, a saber, camada de entrada, camada oculta e camada de saída. A função de ativação dos neurônios da camada de entrada, em geral é uma função linear (ou identidade), e nas camadas ocultas (intermediárias) e de saída, em geral são utilizadas funções do tipo sigmóide.

A construção deste tipo de uma rede do tipo MLP consiste em estipular o número de neurônios em cada camada e as funções de ativação, que serão utilizadas para cada uma das camadas. O número de neurônios da camada de entrada  $N_f$  deve ser adequado ao vetor de entrada que pretende-se utilizar, sendo este dado dependente do problema e da forma de representação dos dados de entrada utilizados. Da mesma forma, o número de neurônios na camada de saída  $N_l$  deve ser adequado à resposta desejada para o problema, caso esteja-se lidando com um problema de classificação o número de neurônios da camada de saída poderia ser igual ao número de classes existentes no problema. [22]

O número de neurônios ideal da camada oculta  $N_h$  nas redes do tipo MLP é determinado geralmente de forma empírica, através da observação da taxa de erro da rede quando é aplicado o conjunto de dados de treinamento. Uma forma prática consiste em iniciar os testes com um valor de  $N_h$  pequeno, aumentando-o a cada teste até que a taxa de erro na fase de aprendizado da rede não apresente mais aumento significativo. Quando for obtida esta condição assume-se que  $N_h$  atingiu o número ideal de neurônios. [22]

No processo de aprendizado de uma rede MLP, em geral utiliza-se a técnica do gradiente decrescente, através do algoritmo de retropropagação, com um coeficiente de aprendizado, ou taxa de aprendizado  $\eta$  ( learning rate) inicialmente próximo de zero. Ao adotarmos um valor muito pequeno para o coeficiente de aprendizado, a trajetória calculada para a função erro será

suave, porém o tempo gasto para atingir o erro mínimo será maior. Se isto ocorrer deve-se aumentar o valor de  $\eta$ , para acelerar o processo de aprendizado da rede, tomando-se o cuidado de acompanhar a variação da função erro, uma vez que valores de  $\eta$  muito grande (próximo de 1), podem ocasionar uma oscilação na rede, onde os valores dos pesos variariam abruptamente, fazendo com que estes oscilassem em torno de um ponto de mínimo erro da rede (denominado mínimo local). [22]

O problema do mínimo local pode ocorrer durante o treinamento da rede, quando há uma convergência da rede para um mínimo local da superfície de erro. Existem várias técnicas que podem ser utilizadas para evitar este problema, como por exemplo a utilização do termo de momento, possibilitando alterações adequadas nos pesos.

O momento (**M**) é um termo adicional para o cálculo da atualização dos pesos, e corresponde a um fator multiplicador, que determina um percentual da correção dos pesos da interação anterior que deve ser aplicado à correção atual, fazendo com que as oscilações sejam amortecidas pelo fator de inércia dos pesos assim criados. O processo de correção dos pesos utilizando o momento é feito utilizando o conjunto das equações 20 a 23 abaixo: [20, 22]

$$\Delta w_{ij}(t) = \eta \cdot \delta_i \cdot X_j + M \quad (20)$$

$$M = \alpha \cdot \Delta w_{ij}(t-1) \quad (21)$$

$$\Delta w_{ij}(t-1) = w_{ij} - w_{ij}(t-1) \quad (22)$$

$$\Delta w_{ij}(t) = w_{ij}(t+1) - w_{ij}(t) \quad (23)$$

Onde:

$w_{ij}(t+1)$  peso no instante  $t+1$

$w_{ij}(t)$  peso no  $t$

$w_{ij}(t-1)$  peso no instante  $t-1$

$\alpha$  fator multiplicador

$X_i$  entradas da rede

$\eta$  taxa de aprendizado

$\delta$  derivação da função de erro com relação aos pesos

Devemos observar que a utilização do momento deve ser feita com critério, pois pode provocar o mesmo efeito de um coeficiente de aprendizado com valor muito alto gerando uma oscilação na rede. [22]

O processo de otimização do MLP envolve os seguintes parâmetros:

- a) O tamanho ótimo da rede: Uma maneira de determinar o tamanho adequado da rede é considerar o número de exemplos de treinamento ou a dimensionalidade dos dados de entrada, construindo um modelo adequado ao problema em questão. [22]
- b) Tempo de treinamento da rede: o tempo de treinamento pode ser feito através da escolha de coeficientes de aprendizado e técnicas para evitar a permanência da rede em mínimos locais. [22]
- c) Capacidade de generalização da rede: habilidade da rede produzir saídas significativas quando apresentada à padrões de entrada novos. A solução para este problema está intimamente ligada aos dois aspectos anteriores, sendo atingida através da utilização de um número suficiente de padrões de treinamento e da adequação da arquitetura da rede de forma a minimizar a representação de redundâncias e aspectos pouco significativos dos padrões representados. [22]

#### **4.6. Redes Neurais do Tipo Radial Basis Function (RBF)**

A rede do tipo RBF enquadra-se na classe das redes de multicamadas, sendo formada geralmente por uma camada de entrada, uma camada oculta (intermediária) e uma camada de saída. A diferença com relação às outras redes do tipo MLP, deve-se ao fato dos neurônios da camada oculta utilizarem funções de ativação de base radial, que apresentam uma resposta localizada que é radialmente simétrica em relação a um centro. Desta forma, são formadas regiões denominadas campos receptivos e a camada de saída efetua uma combinação linear da resposta destas regiões. Na maioria das redes de multicamadas a função de ativação aplicada a cada neurônio da camada oculta recebe como argumento o produto escalar do vetor de entrada e do vetor de peso deste neurônio. No caso da RBF, a função de ativação dos neurônios da camada oculta são em função da distância entre os vetores de entrada e os vetores dos centros (centróides). [22]

A utilização de funções de base radial na camada oculta desta rede provoca um mapeamento das amostras de entrada para um espaço não linear, buscando tornar estas linearmente separáveis pela camada de saída. Esta propriedade é descrita pelo Teorema de Cover, para a separação de padrões, onde é determinado que: “um problema de classificação de padrões complexo, tem maior chance de ser linearmente separável, se for mapeado não linearmente para um outro espaço diferente daquele utilizado originalmente pelo problema”, ou seja, levar um problema de classificação para um espaço, eventualmente, de maior dimensão pode fazer com que sejam observadas propriedades que não são observáveis no espaço original do problema. [29]

a proximidade de um padrão apresentado em sua entrada com relação aos padrões representados pelos campos receptivos. [27]

A figura 22 mostra a estrutura da rede neural do tipo RBF usada neste trabalho. Apresenta uma camada oculta, onde os neurônios usam funções radiais, X é a entrada da rede, W são os pesos,  $f()$  função de ativação e Y a saída.

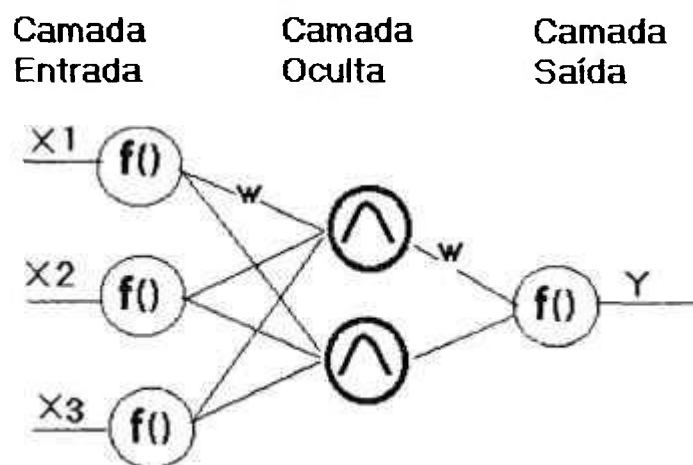


Figura 22 - Rede Neural do Tipo RBF

As redes do tipo RBF, na sua forma clássica, utilizam uma estrutura composta por três camadas de neurônios, camada de entrada, camada oculta (intermediária) e camada de saída. A função de ativação dos neurônios da camada de entrada é a identidade, e a camada oculta utiliza funções de base radial como funções de ativação. As funções de base radial (RBF) são uma classe especial de funções, cuja característica é a sua resposta crescer (ou decrescer) de forma monotônica com a distância de um ponto central. A expressão geral para qualquer função de base radial tem a seguinte equação (24):

$$F(X) = \phi ((X-C)^t R^{-1}(X-C)) \quad (24)$$

onde  $\phi$  é a função aplicada,  $C$  é o centro da função,  $R$  é a métrica utilizada e  $X$  são as entradas.

[27]

Freqüentemente utiliza-se a métrica Euclidiana e neste caso para um dado raio  $r$  (largura da função radial), a métrica fica sendo  $R = \sigma^2 I$  onde  $I$  é a matriz identidade. Caso o espaço seja unidimensional temos a equação acima simplificada conforme equação (25). [27] A equação (25) foi adotada neste trabalho:

$$F(X) = \phi \left( \frac{(X-C)^2}{\sigma^2} \right) \quad (25)$$

Onde:

$X$  são as entradas

$C$  os centros ( centróides)

$\sigma$  é o raio ou largura da Função de Base Radial

$\phi$  é a Função de Base Radial

Diferentes funções de base radiais são utilizadas em redes do tipo RBF. As mais comuns são apresentadas na tabela 4.1 e na figura 23, onde  $d = \| X-C \|$ , que é dado pela distância euclidiana,  $X$  é o vetor de entrada,  $C$  o centro ( centróide) e  $\sigma$  a largura da função radial (raio). Neste trabalho é usada a função Gaussiana, conforme equação (26)

$$\phi(z) = \exp(- d^2 / \sigma^2). \quad (26)$$

Tabela 4.1 – Expressão de algumas funções de Base Radial

Função	Expressão
Gaussiana	$\phi(z) = e^{-z^2}$
Multiquadratic	$\phi(z) = (1+z^2)^{1/2}$
Inverse Multiquadratic	$\phi(z) = (1+z^2)^{-1/2}$
Cauchy	$\phi(z) = (1+z^2)^{-1}$

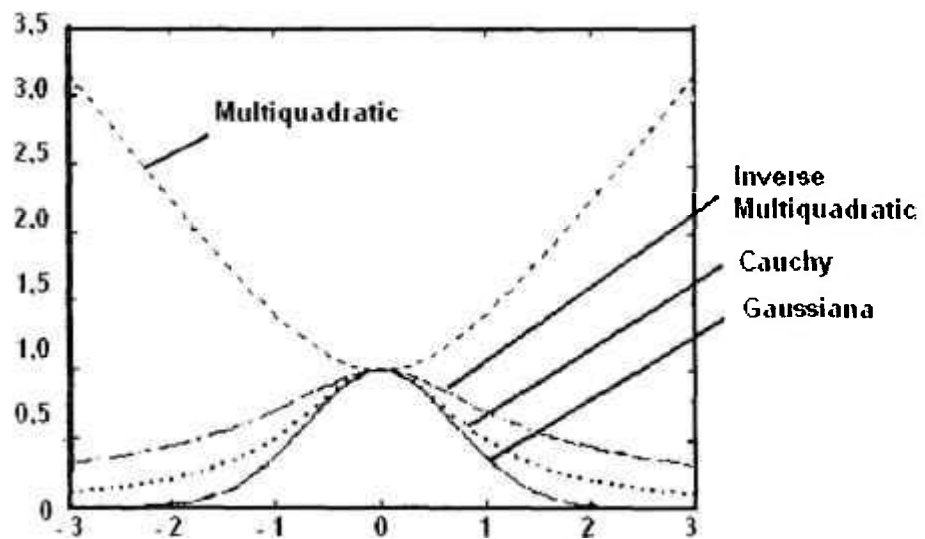


Figura 23 - Funções de Base Radial considerando  $c=0$  e  $\sigma=1$ . [22]

Assim como ocorre nas redes dos tipos MLP, em geral na camada de saída utiliza-se neurônios com funções de ativação do tipo sigmoidais ou lineares.

A construção de uma rede do tipo RBF é semelhante ao processo da rede MLP. Consiste em estipular qual será o número de neurônios em cada camada da rede e as funções que serão utilizadas para cada uma das camadas. O número de neurônios da camada de entrada  $N_f$ , da camada de saída  $N_I$ , dependem do problema em questão e da forma de representação dos dados de entrada utilizados.

O grande problema não só no caso das redes do tipo RBF como em qualquer outro tipo de rede neural, é determinar de forma eficiente o número de neurônios da camada oculta  $N_h$ . Existem duas recomendações durante a otimização do número de neurônios da camada oculta (ou número de funções de base radial) que devem ser observadas: a primeira consiste em garantir um número de neurônios na camada oculta suficiente para aumentar a capacidade de separação linear dos padrões de cada uma das classes do problema; a segunda recomendação por sua vez consiste em que o número de neurônios na camada oculta não deve aumentar demasiadamente para que a rede não aprenda o ruído associado às amostras. O número de neurônios ideal geralmente é determinado de forma empírica através da observação da taxa de erro da rede quando é aplicado o conjunto de dados de teste.[22]

O aprendizado da rede do tipo RBF é feito em duas fases:

A primeira, onde são posicionados os centros da RBF, consiste em determinar os centros e o raio das funções gaussianas. A forma mais simples de posicionar os centros das RBFs, é escolher os mesmos de forma aleatória, a partir do conjunto de dados de treinamento. Porém pode levar à polarização da rede sobre uma determinada parte do espaço de entrada. A solução para este tipo de problema consiste em construir um conjunto de vetores (centróides) representativos das classes do problema.

O método mais utilizado para determinar os centróides é o algoritmo de K-means-clustering ou K-médias. Estes algoritmos permitem a determinação dos centros (centróides) posicionados em regiões do espaço onde estão situados os vetores de entrada mais representativos. [20]

O raio  $\sigma$  é a largura da Função Radial. É ele que determina a extensão do campo receptivo de cada neurônio da camada oculta. Várias teorias tem sido propostas para determinar o raio  $\sigma$ . Uma delas define o raio como sendo a média sobre todas distâncias euclidianas entre o centro de cada unidade  $n_i$  e o centro da unidade  $n_j$ . O raio a ser escolhido pode ser igual ou diferente para todas as funções da camada oculta.

Não existe um valor padrão que possa ser adotado para o raio  $\sigma$ , devendo o mesmo ser determinado de forma experimental. Em geral o valor final do raio  $\sigma$  depende do tipo de problema. [27]

É importante observar que a determinação criteriosa dos parâmetros (centro e raio), evita que a rede RBF fique presa em mínimos locais, já comentado no item 4.5.

A segunda fase de aprendizado consiste em ajustar os pesos entre a camada oculta e a camada de saída. No tipo de arquitetura onde os centros são fixos, como é o caso da rede RBF desenvolvida neste trabalho, o ajuste do peso ocorre apenas entre a camada oculta e a camada de saída, portanto o método de treinamento é mais simples, o método da Regra Delta, apresentado no item 4.4.3, evitando-se métodos mais complexos, como por exemplo, o Algoritmo de Retropropagação apresentado no item 4.4.4.

Existem alguns algoritmos onde os ajustes dos parâmetros das funções de base radial ocorrem durante o treinamento da rede, o que não ocorre neste trabalho.

## **5. Desenvolvimento do Trabalho**

### **5.1 Dados para Treinamento das Redes Neurais**

Conforme já mencionado, este trabalho, tem como objetivo o desenvolvimento de uma rede neural para monitorar, de forma diversional, a potência do reator nuclear de pesquisas IEA-R1. Para tanto foram colhidos dados do reator nuclear IEA-R1, para alimentação da rede. Foram considerados três parâmetros de entrada, a saber: a temperatura da água antes de entrar no núcleo do reator (T3), a temperatura da água após passar pelo núcleo do reator (T4), e a atividade do nitrogênio 16 ( $N^{16}$ ). Como parâmetro de saída, foi estabelecido que a rede deve fornecer a potência do reator, baseado na informação dos 3 parâmetros selecionados, que, como sabemos, mudam de acordo com a potência do reator. Os valores do % de potência escalar foram coletados do Canal Linear para servirem de padrão durante a fase de treinamento da rede.

Foram realizadas 74 medidas de potência, onde foram coletados os parâmetros acima definidos (Canal linear, T3, T4 e  $N^{16}$ ). Para a obtenção destes dados a potência do reator foi alterada em intervalos de 100 KW, aguardando-se cerca de 1 a 2 minutos para realizar a leitura dos valores, e foram efetuadas, duas leituras para cada nível de potência, inicialmente num intervalo descendente de 3.800 KW até 200 KW e depois num intervalo ascendente de 200 KW até 3.800 KW. Todas as medidas foram realizadas num mesmo dia, mantendo-se a vazão no circuito primário constante. Estas medidas foram distribuídas em dois grupos com 37 medidas cada, onde ambos os grupos contém dados dos dois intervalos (ascendente e descendente). O primeiro grupo foi utilizado para a fase de treinamento das redes neurais e o outro grupo de dados para a fase de teste das mesmas. Em todos os casos as entradas foram normalizadas entre os valores 0 e 1. A tabela 5.1 mostra o conjunto dos 74 valores utilizados para treinamento das redes.

**Tabela 5.1 – Valores utilizados nas fases de treinamento e teste das redes.**

Valores Absolutos						Valores Normalizados			
	T3 °C	T4 °C	N16 %	POT (KW)	Canal Linear %	T3	T4	N16	Canal Linear
1	32,2	36,5	42,2	3800	66,000	0,2459	0,9508	1,0000	0,660
2	32,5	36,7	40,8	3700	65,000	0,2951	0,9836	0,9623	0,650
3	32,6	36,8	39,6	3600	63,500	0,3115	1,0000	0,9299	0,635
4	32,9	36,8	38,8	3500	62,000	0,3607	1,0000	0,9084	0,620
5	32,9	36,8	37,8	3400	60,000	0,3607	1,0000	0,8814	0,600
6	32,9	36,7	36,8	3300	58,500	0,3607	0,9836	0,8544	0,585
7	33	36,7	35,7	3200	56,500	0,3770	0,9836	0,8248	0,565
8	33	36,6	34,6	3100	55,000	0,3770	0,9672	0,7951	0,550
9	33	36,5	33,6	3000	53,000	0,3770	0,9508	0,7682	0,530
10	33,1	36,5	32,5	2900	51,500	0,3934	0,9508	0,7385	0,515
11	33,1	36,4	31,6	2800	49,500	0,3934	0,9344	0,7143	0,495
12	33	36,3	30	2700	48,000	0,3770	0,9180	0,6712	0,480
13	33	36,2	29,5	2600	47,000	0,3770	0,9016	0,6577	0,470
14	33	36,1	28,5	2500	45,000	0,3770	0,8852	0,6307	0,450
15	33,1	36	27,6	2400	43,000	0,3934	0,8689	0,6065	0,430
16	33,1	35,9	26,5	2300	41,500	0,3934	0,8525	0,5768	0,415
17	33	35,8	25,4	2200	40,000	0,3770	0,8361	0,5472	0,400
18	33	35,6	24,5	2100	38,500	0,3770	0,8033	0,5229	0,385
19	33	35,5	23,5	2000	37,000	0,3770	0,7869	0,4960	0,370
20	33	35,4	22,5	1900	35,500	0,3770	0,7705	0,4690	0,355
21	33	35,2	21,5	1800	33,500	0,3770	0,7377	0,4420	0,335
22	33	35,1	20,5	1700	32,000	0,3770	0,7213	0,4151	0,320
23	32,9	34,9	19,5	1600	30,000	0,3607	0,6885	0,3881	0,300
24	32,9	34,8	18,3	1500	28,500	0,3607	0,6721	0,3558	0,285
25	32,8	34,7	17,3	1400	27,000	0,3443	0,6557	0,3288	0,270
26	32,8	34,6	16,3	1300	25,000	0,3443	0,6393	0,3019	0,250
27	32,7	34,4	15,4	1200	23,500	0,3279	0,6066	0,2776	0,235
28	32,7	34,3	14,4	1100	22,000	0,3279	0,5902	0,2507	0,220
29	32,6	34,1	13,3	1000	20,500	0,3115	0,5574	0,2210	0,205
30	32,6	34	12,2	900	18,500	0,3115	0,5410	0,1914	0,185
31	32,5	33,7	11,2	800	17,000	0,2951	0,4918	0,1644	0,170
32	32,4	33,6	10,3	700	15,500	0,2787	0,4754	0,1402	0,155
33	32,3	33,4	8,7	600	14,000	0,2623	0,4426	0,0970	0,140
34	32,2	33,1	8,1	500	13,500	0,2459	0,3934	0,0809	0,135

	T3 °C	T4 °C	N16 %	POT (KW)	Canal Linear %	T3	T4	N16	Canal Linear
35	32,1	32,9	7,1	400	11,000	0,2295	0,3607	0,0539	0,110
36	31,9	32,7	6,1	300	9,500	0,1967	0,3279	0,0270	0,095
37	31,7	32,4	5,1	200	7,500	0,1639	0,2787	0,0000	0,075
38	31,7	32,4	5,1	200	7,500	0,1639	0,2787	0,0000	0,075
39	31,6	32,3	5,9	300	8,000	0,1475	0,2623	0,0216	0,080
40	31,5	32,2	6,9	400	9,500	0,1311	0,2459	0,0485	0,095
41	31,2	32,1	7,8	500	11,000	0,0820	0,2295	0,0728	0,110
42	31,1	32,1	8,7	600	12,500	0,0656	0,2295	0,0970	0,125
43	31	32,1	9,6	700	14,000	0,0492	0,2295	0,1213	0,140
44	30,9	32,2	10,8	800	15,500	0,0328	0,2459	0,1536	0,155
45	30,9	32,2	11,8	900	17,500	0,0328	0,2459	0,1806	0,175
46	30,9	32,3	13	1000	19,000	0,0328	0,2623	0,2129	0,190
47	31	32,3	13,8	1100	20,500	0,0492	0,2623	0,2345	0,205
48	30,9	32,4	14,8	1200	22,000	0,0328	0,2787	0,2615	0,220
49	30,8	32,5	15,7	1300	23,500	0,0164	0,2951	0,2857	0,235
50	30,7	32,6	17,1	1400	25,000	0,0000	0,3115	0,3235	0,250
51	30,8	32,7	18,1	1500	27,000	0,0164	0,3279	0,3504	0,270
52	30,8	32,8	19,1	1600	28,500	0,0164	0,3443	0,3774	0,285
53	30,9	33	20,3	1700	31,000	0,0328	0,3770	0,4097	0,310
54	30,9	33,1	21,1	1800	32,000	0,0328	0,3934	0,4313	0,320
55	30,9	33,3	22,4	1900	34,000	0,0328	0,4262	0,4663	0,340
56	31	33,4	23,5	2000	35,500	0,0492	0,4426	0,4960	0,355
57	31	33,6	24,4	2100	37,500	0,0492	0,4754	0,5202	0,375
58	31,1	33,7	25,5	2200	39,000	0,0656	0,4918	0,5499	0,390
59	31,2	33,9	26,5	2300	40,000	0,0820	0,5246	0,5768	0,400
60	31,2	34	27,3	2400	41,500	0,0820	0,5410	0,5984	0,415
61	31,2	34,1	28,2	2500	43,000	0,0820	0,5574	0,6226	0,430
62	31,4	34,4	29,1	2600	45,000	0,1148	0,6066	0,6469	0,450
63	31,5	34,5	30,2	2700	46,500	0,1311	0,6230	0,6765	0,465
64	31,6	34,7	31,2	2800	48,500	0,1475	0,6557	0,7035	0,485
65	31,6	34,9	32,5	2900	50,000	0,1475	0,6885	0,7385	0,500
66	31,7	35,1	33,6	3000	52,000	0,1639	0,7213	0,7682	0,520
67	31,7	35,2	34,5	3100	54,000	0,1639	0,7377	0,7925	0,540
68	31,7	35,4	35,6	3200	55,500	0,1639	0,7705	0,8221	0,555
69	31,9	35,7	36,7	3300	57,500	0,1967	0,8197	0,8518	0,575
70	32	35,8	37,8	3400	59,000	0,2131	0,8361	0,8814	0,590

	T3 °C	T4 °C	N16 %	POT (KW)	Canal Linear %	T3	T4	N16	Canal Linear
71	32	36	38,8	3500	60,500	0,2131	0,8689	0,9084	0,605
72	32,2	36,3	39,8	3600	62,500	0,2459	0,9180	0,9353	0,625
73	32,3	36,5	40,8	3700	64,000	0,2623	0,9508	0,9623	0,640
74	32,3	36,6	41,9	3800	66,000	0,2623	0,9672	0,9919	0,660

## 5.2 Desenvolvimento das Redes Neurais

Foram desenvolvidas duas redes em linguagem C, apresentadas no Apêndice I, uma do tipo MLP e outra do tipo RBF, ambas com 3 camadas, conforme ilustrado na figura 24, com o objetivo de determinar o percentual da potência do reator a partir dos valores fornecidos pelos instrumentos responsáveis pelas medidas de T3, T4 e N<sup>16</sup>.

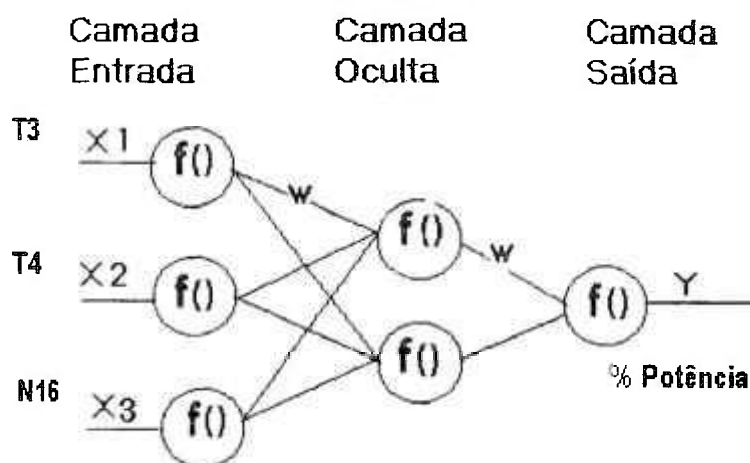


Figura 24 – Arquitetura da rede desenvolvida neste trabalho, tanto para rede do tipo MLP como para rede do tipo RBF.

### **Arquitetura das redes MLP e RBF desenvolvidas neste trabalho:**

Foram desenvolvidas as redes MLP e RBF com três camadas cada uma, sendo uma de entrada, uma intermediária (oculta) e uma de saída:

- **Camada de entrada**

O número de neurônios da camada de entrada das redes do tipo MLP e RBF é três (3). A camada de entrada será responsável pela entrada de dados nas redes, com os valores de T3, T4 e N<sup>16</sup>.

A função utilizada nos neurônios da camada de entrada é a função linear (identidade) representada pela equação (27):

$$Y = \alpha \cdot X \quad (27)$$

Onde :

$$\alpha = 1$$

X é entrada no neurônio da camada de entrada

Y é a saída do neurônio da camada de entrada

- **Camada Oculta ( intermediária)**

Para a camada oculta das redes do tipo MLP e RBF foram estudados e testados 15 casos com diferentes números de neurônios. O objetivo deste estudo é encontrar um número ideal de

neurônios para a camada oculta, de forma a obter a saída da rede com valor o mais aproximado possível da saída desejada. Foram geradas redes com número de neurônios da camada oculta variando entre 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 20, 22 e 37.

Para função de ativação dos neurônios da camada oculta da rede tipo MLP foi escolhida a função sigmóide representada na equação (28).

$$Y=1/(1+\exp (-X)) \quad (28)$$

Onde:

X são as entradas

Y é a saída (resposta da função de ativação)

Para função de ativação dos neurônios da camada oculta da rede tipo RBF foi escolhida a função de Base Radial Gaussiana representada na equação (29).

$$\phi(z) = \exp(- (X-C)^2 / \sigma^2) \quad (29)$$

Onde:

X são as entradas

C os centros ( centróides)

$\sigma$  é o raio ou largura da Função de Base Radial

$\phi$  é a Função de Base Radial

Para cada um dos quinze (15) casos estudados na rede do tipo RBF, foram calculados os seus respectivos centróides conforme o número de neurônios da camada oculta. Portanto, o número de centróides variou entre 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 20, 22 e 37.

Neste trabalho foi utilizado um tipo de arquitetura onde, uma vez determinados os centros e o raio, eles ficam com valores fixos na camada oculta. Os centróides das RBFs foram calculados como sendo o centro mais representativo de cada classe. O algoritmo para determinação dos centróides foi o algoritmo de K-means-clustering. [38]

Para a determinação do raio  $\sigma$  inicial foi considerado um valor calculado através da média da distância euclidiana entre os centróides da rede, conforme comentado no item 4.6. O valor encontrado foi sendo ajustado experimentalmente conforme a convergência da rede, procurando uma taxa de erro de convergência menor.

O Apêndice II apresenta as tabelas com os valores dos centróides calculados conforme o número de neurônios na camada oculta, e o valor do raio  $\sigma$  ajustado.

- **Camada de Saída**

A camada de saída das redes do tipo MLP e RBF, apresentam neste trabalho apenas um (1) neurônio, que representa a resposta de saída da rede, no caso o percentual (%) da potência.

A função de ativação utilizada no neurônio da camada de saída das redes tipo MLP e RBF, é a mesma, no caso a função sigmóide representada na equação (30).

$$Y=1/(1+\exp (-X)) \quad (30)$$

Onde:

X são as entradas

Y é a saída (resposta da função de ativação)

### **Treinamento das Redes**

Na fase de treinamento a rede é alimentada com valores pré-selecionados, no caso um conjunto de treinamento composto por 37 dados, que resultavam em respostas conhecidas (resposta desejada).

Os pesos foram inicializados com valores gerados aleatoriamente, e a regra de aprendizado foi utilizada conforme a característica de cada rede, onde a matriz de conectividade (pesos) é ajustada durante o treinamento de forma que a resposta da rede seja igual à resposta desejada.

Em geral, para o treinamento destas redes são necessárias muitas épocas, principalmente no caso da MLP. Como será explicado adiante, neste trabalho foi adotado 10.000 épocas, para ambas as redes, sendo que cada época corresponde a um ciclo completo de leitura de todas as entradas selecionadas para a fase de treinamento.

No caso da MLP, o algoritmo de treinamento da rede para efetuar a correção dos pesos é o da retropropagação, citado no item 4.4.4 .

Assim como no caso da rede do tipo MLP, a rede RBF é treinada até obter saídas aproximadas das desejadas. Neste trabalho foi utilizado um tipo de arquitetura, onde os centros e raio são fixos, portanto o ajuste dos pesos ocorre apenas entre a camada oculta e a camada de saída, não sendo utilizado o método de “backpropagation” para se ajustar os pesos da informação que entra nos neurônios da camada intermediária. Neste caso a correção dos pesos é feita através da regra delta.

#### ■ **Treinamento da MLP**

Num processo de treinamento em geral são consideradas duas condições para se parar o processo. A primeira condição é o critério de convergência, e a segunda condição é o número de épocas permitido, (de modo a não ter um processo que dure indefinidamente). O critério de convergência das redes, que é o erro médio de uma época, consiste na média da diferença em módulo, entre a resposta da rede e a resposta desejada para todos os padrões de entrada, conforme equação (31). Caso o erro médio por época seja inferior a um valor de erro previamente especificado, significa que a rede convergiu, apresentando um erro médio por época inferior ao valor do erro, portanto o treinamento é parado e considerado concluído. No início do trabalho o valor estipulado para o erro médio por época, para critério de parada, deveria ser inferior à 0.01, como a grande maioria das redes estava atendendo este critério e no intuito de reduzir ainda mais o valor do erro médio por época, foi adotado neste trabalho como critério de parada do treinamento o número de épocas, no caso 10.000 épocas.

$$\Sigma || Y_d - Y || / N_p \quad (31)$$

Onde:

$Y_d$  é saída desejada

$Y$  é a saída da rede

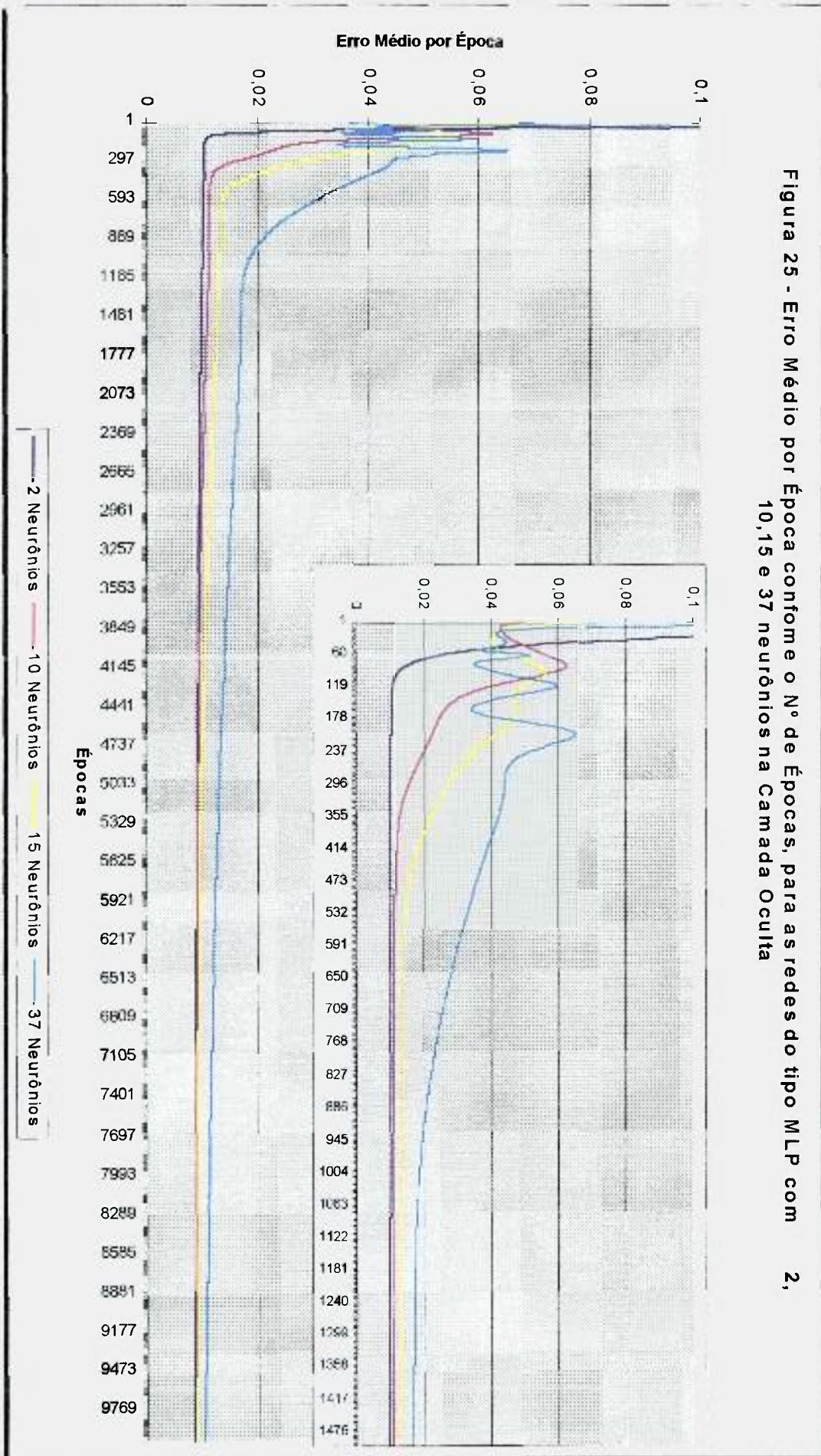
$N_p$  é o número de padrões de dados

Como podemos observar na figura 25, após cerca de 10.000 épocas o erro médio por época permanece constante e, tendo em vista que, o tempo de processamento envolvido na fase de aprendizagem ser relativamente pequeno, como mostra alguns exemplos da tabela 5.2, resolvemos considerar a limitação de 10.000 épocas como critério para concluir a fase de aprendizado.

Tabela 5.2 – Tempo de Processamento para 10.000 épocas na fase de aprendizado das redes do tipo MLP.

<i>Número de neurônios na Camada Oculta</i>	<i>Tempo de processamento</i>
37	3'(minutos) e 55,53"(segundos)
10	2'(minutos) e 0,85"(segundos)
2	1'(minuto) e 25,03"(segundos)

Figura 25 - Erro Médio por Época conforme o N° de Épocas, para as redes do tipo MLP com 2, 10, 15 e 37 neurônios na Camada Oculta



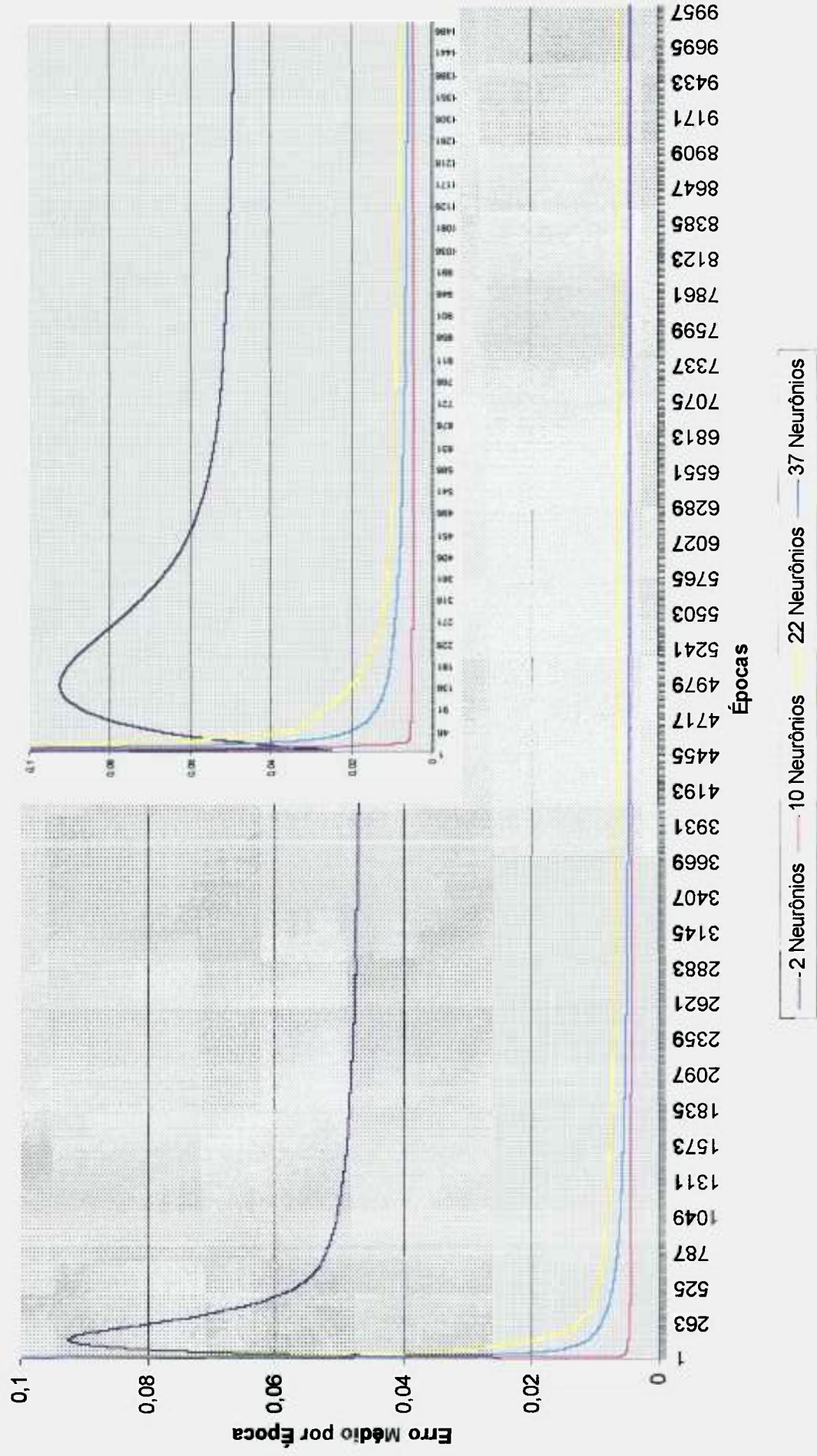
### ■ Treinamento da RBF

Assim como no caso das redes do tipo MLP, para as redes do tipo RBF o processo de treinamento foi considerado concluído quando o número de épocas atingisse o valor de 10.000. A justificativa é a mesma do caso das redes do tipo MLP, ou seja, como pode ser visto na figura 26, foi considerado que após 10.000 épocas o erro médio relativo permanece constante, e tendo em vista que o tempo de processamento envolvido na fase de treinamento é relativamente pequeno, conforme alguns exemplos da tabela 5.3, resolvemos considerar a limitação de 10.000 épocas como critério para concluir a fase de aprendizado.

**Tabela 5.3 – Tempo de Processamento para 10.000 épocas na fase de aprendizado das redes do tipo RBF.**

<i>Número de neurônios na Camada Oculta</i>	<i>Tempo de processamento</i>
37	2'(minutos) e 30,08'' (segundos)
10	1'(minuto) e 49,79''(segundos)
2	1'(minuto) e 39,11''(segundos)

**Figura 26 - Erro Médio por Época conforme o número de Épocas, para as Redes do tipo RBF com 2, 10, 22 e 37 neurônios na Camada Oculta**



### Teste das Redes

Na fase de teste a rede inicialmente é alimentada com a outra metade dos valores que não participaram da fase de treinamento, no caso o conjunto de teste é composto por 37 dados, e posteriormente pelo conjunto total de 74 dados. Nesta fase não ocorre mais aprendizado da rede e os pesos das camadas são fixos. Como resultado a rede fornece, na sua saída o valor da potência do reator.

Como parâmetro para comparação do desempenho de cada rede foi utilizado o erro médio relativo da resposta de cada rede em relação ao grupo de teste e ao grupo do conjunto total de dados. A rede que apresentar o menor erro médio relativo é a que será escolhida para o aplicativo. O erro médio relativo é calculado conforme a equação (32):

$$\Sigma (|| Y_d - Y || / Y_d) / N_p \quad (32)$$

Onde:

$Y_d$  é saída desejada

$Y$  é a saída da rede

$N_p$  número de padrões

### **5.3 Desenvolvimento da Tela de informação**

Tendo em vista que o trabalho tem como objetivo o desenvolvimento de uma forma diversional de se monitorar a potência do reator IEAR-1, foi desenvolvida uma tela, através do software Visual C++, para mostrar ao operador, o valor da potência calculada pela rede e permitir ao mesmo visualizar outros parâmetros de potência. Na Tela, conforme ilustrado na figura 27, o operador pode visualizar em nível de percentagem (%), a potência indicada pelo Canal do Nitrogênio 16, Canal Linear, Canais de Segurança 1, 2, 3 e o valor calculado pela Rede Neural, além dos parâmetros de Temperatura T3, Temperatura T4 e a diferença (DP) entre a Potência do Canal Linear e a Potência calculada pela Rede Neural. Devemos destacar que o valor de DP é um referencial de avaliação para o operador do reator poder verificar a precisão dos seus instrumentos, pois se houver uma grande variação do DP, será devido a uma falha em um dos instrumentos utilizados na monitoração da Potência, ou seja, deve ser estabelecido um valor máximo para DP (em módulo) que, uma vez ultrapassado, deve gerar um alarme para reconhecimento do operador.

Para desenvolver a tela e os programas das redes foi utilizado um computador do tipo PC com 32 MB de memória RAM e um HD com capacidade de 4G.

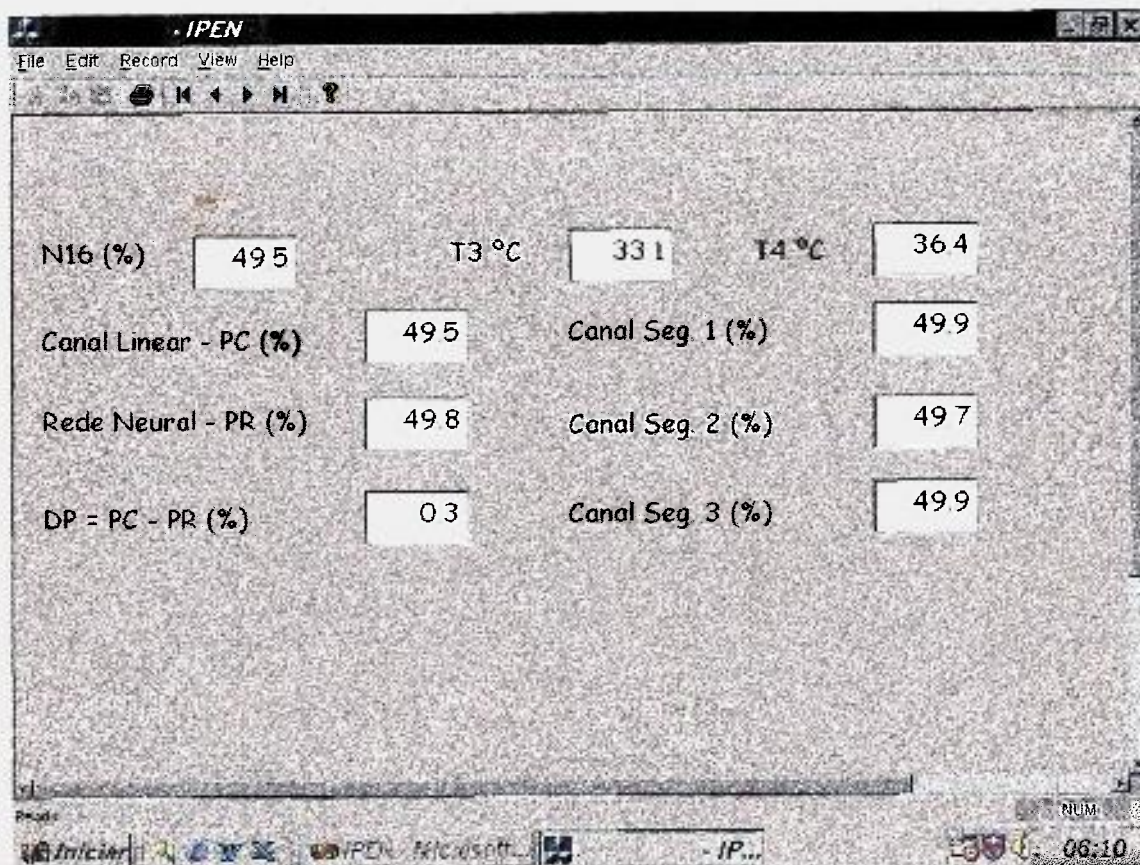


Figura 27 - Tela para Monitoração da potência do reator IEA-R1.

## 6. Resultados e Comparações

### 6.1 Resultados para Redes do tipo MLP

Os resultados obtidos para as diferentes redes do tipo MLP é mostrado na tabela 6.1, onde o erro médio relativo ( em %) é definido como sendo a média da diferença em módulo do valor obtido pela rede menos o valor desejado, dividido pelo valor desejado e multiplicado por 100. A figura 28 mostra o gráfico do erro médio relativo ( em %) em função do número de neurônios da camada oculta para o caso das redes MLP.

Tabela 6.1 - Resultados obtidos para as diferentes Redes do tipo MLP, em função do número de neurônios na Camada Oculta:

Nº de neurônios na camada Oculta	Erro médio relativo (%) entre a resposta da rede e a resposta desejada utilizando os 37 dados da fase de teste):	Erro médio relativo (%) entre a resposta da rede e a resposta desejada (utilizando todos os 74 dados disponíveis)
2	3,72	4,01
3	3,67	3,95
4	3,72	4,00
5	3,57	3,83
6	3,46	3,71
7	3,09	3,36
8	2,97	3,25
9	2,93	3,20
10	2,90	3,17
11	2,87	3,15
12	2,86	3,13
15	2,85	3,10
20	2,91	3,12
22	2,95	3,14
37	3,39	3,47

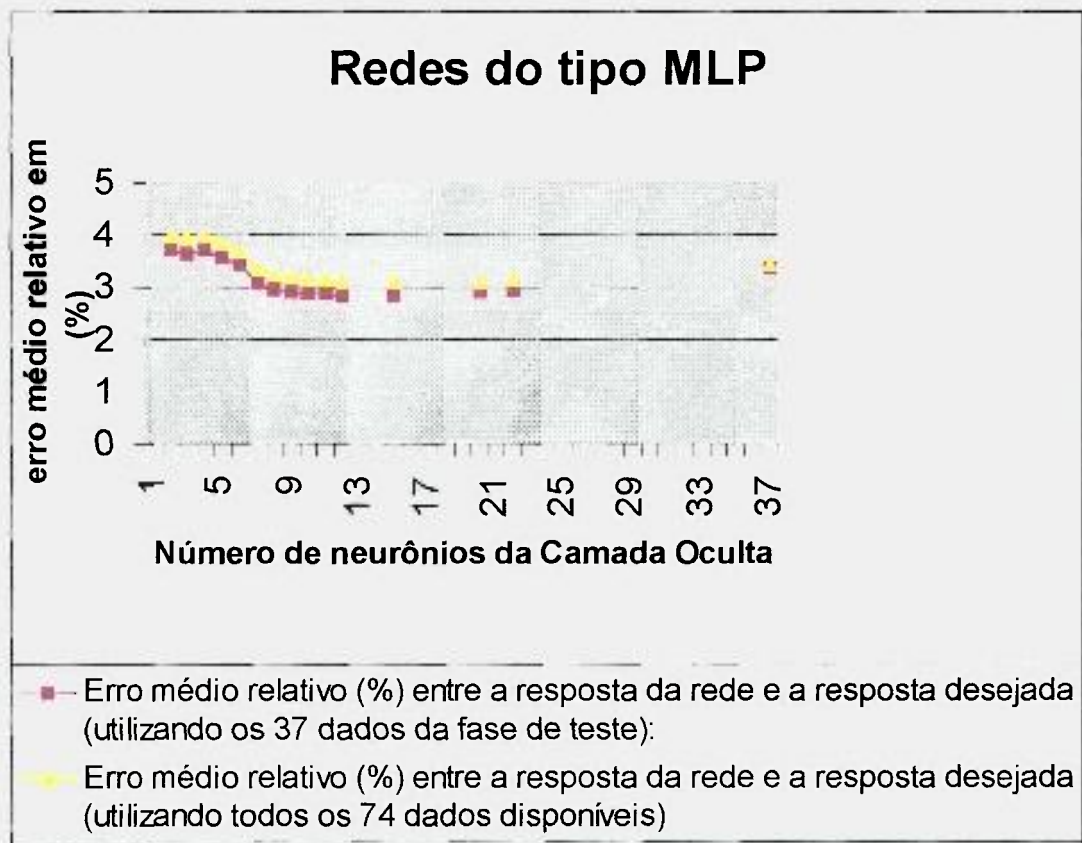


Figura 28 – Erro médio relativo em função do número de neurônios na camada oculta das redes do tipo MLP

A tabela 6.2 mostra as respostas típicas fornecidas pela rede na fase de Teste, utilizando-se as diferentes redes do tipo MLP analisadas.

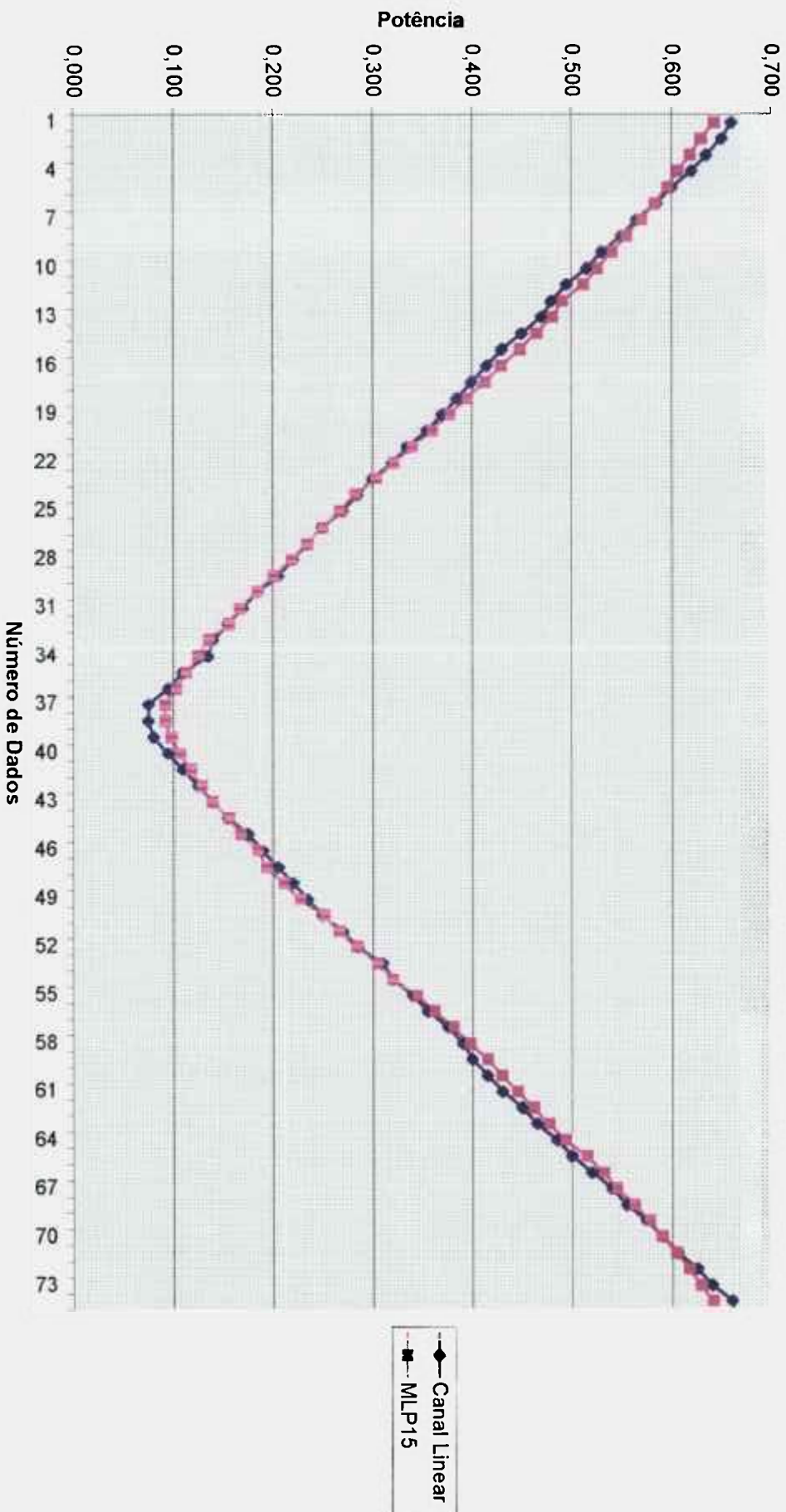
Tabela 6.2 - Respostas Típicas fornecidas pela rede do tipo MLP, na fase de Teste, em função do nº de neurônios da camada oculta

Canal Linear	Número de neurônios na Camada Oculta																					
	2	3	4	5	6	7	8	9	10	11	12	15	20	22	37							
0,660	0,645	0,646	0,645	0,638	0,634	0,643	0,644	0,644	0,644	0,644	0,644	0,643	0,641	0,641	0,637							
0,650	0,631	0,631	0,631	0,626	0,622	0,629	0,631	0,631	0,631	0,630	0,630	0,629	0,628	0,628	0,624							
0,635	0,619	0,620	0,619	0,616	0,613	0,618	0,620	0,619	0,619	0,619	0,619	0,618	0,617	0,617	0,615							
0,620	0,606	0,607	0,606	0,604	0,602	0,604	0,609	0,608	0,608	0,607	0,607	0,606	0,605	0,604	0,602							
0,600	0,595	0,596	0,595	0,594	0,592	0,594	0,598	0,598	0,597	0,597	0,597	0,596	0,595	0,595	0,593							
0,585	0,582	0,582	0,582	0,582	0,580	0,581	0,585	0,585	0,584	0,584	0,584	0,583	0,582	0,582	0,581							
0,565	0,568	0,568	0,567	0,568	0,568	0,566	0,572	0,571	0,571	0,570	0,570	0,569	0,569	0,569	0,569							
0,550	0,552	0,552	0,551	0,554	0,554	0,551	0,557	0,556	0,556	0,555	0,555	0,555	0,554	0,554	0,555							
0,530	0,537	0,537	0,536	0,539	0,540	0,536	0,543	0,542	0,542	0,541	0,541	0,541	0,541	0,541	0,542							
0,515	0,521	0,521	0,520	0,524	0,526	0,521	0,528	0,527	0,526	0,526	0,526	0,525	0,525	0,526	0,527							
0,495	0,507	0,506	0,506	0,511	0,513	0,506	0,514	0,513	0,513	0,512	0,512	0,512	0,512	0,512	0,514							
0,480	0,484	0,484	0,483	0,489	0,492	0,485	0,491	0,491	0,491	0,490	0,490	0,490	0,491	0,492	0,495							
0,470	0,474	0,474	0,473	0,480	0,483	0,475	0,483	0,482	0,482	0,481	0,481	0,481	0,482	0,483	0,486							
0,450	0,457	0,457	0,456	0,463	0,467	0,459	0,466	0,466	0,465	0,465	0,465	0,465	0,466	0,467	0,470							
0,430	0,440	0,439	0,438	0,446	0,451	0,441	0,450	0,449	0,449	0,448	0,448	0,448	0,449	0,449	0,453							
0,415	0,420	0,420	0,419	0,427	0,433	0,422	0,432	0,431	0,430	0,430	0,430	0,430	0,431	0,431	0,436							
0,400	0,403	0,403	0,402	0,411	0,416	0,406	0,414	0,414	0,413	0,413	0,413	0,413	0,415	0,415	0,420							
0,385	0,384	0,384	0,383	0,392	0,397	0,387	0,397	0,396	0,395	0,395	0,395	0,395	0,396	0,396	0,401							
0,370	0,366	0,366	0,365	0,374	0,380	0,369	0,379	0,378	0,378	0,377	0,377	0,377	0,378	0,379	0,384							
0,355	0,348	0,348	0,347	0,356	0,362	0,351	0,361	0,361	0,360	0,360	0,360	0,360	0,361	0,361	0,366							
0,335	0,328	0,328	0,327	0,335	0,341	0,331	0,342	0,341	0,340	0,340	0,339	0,339	0,340	0,340	0,345							

Baseado na figura 28, resolvemos selecionar como rede de melhor desempenho do tipo MLP, a rede que possui 15 neurônios na camada oculta por considerarmos que foi a rede que apresentou o menor erro médio relativo, e situa-se numa região na qual o aumento, ou diminuição, do número de neurônios da Camada Oculta praticamente não altera o resultado.

A figura 29 mostra, o valor da potência fornecida pelo Canal Linear e a potência fornecida pela rede do tipo MLP, com 15 neurônios na camada oculta, para cada padrão pertencente ao conjunto total de dados, composto de valores que foram utilizados para teste e treinamento da rede.

Figura 29 - O valor da Potência fornecida pelo Canal Linear e o valor da Potência fornecida pela rede MLP

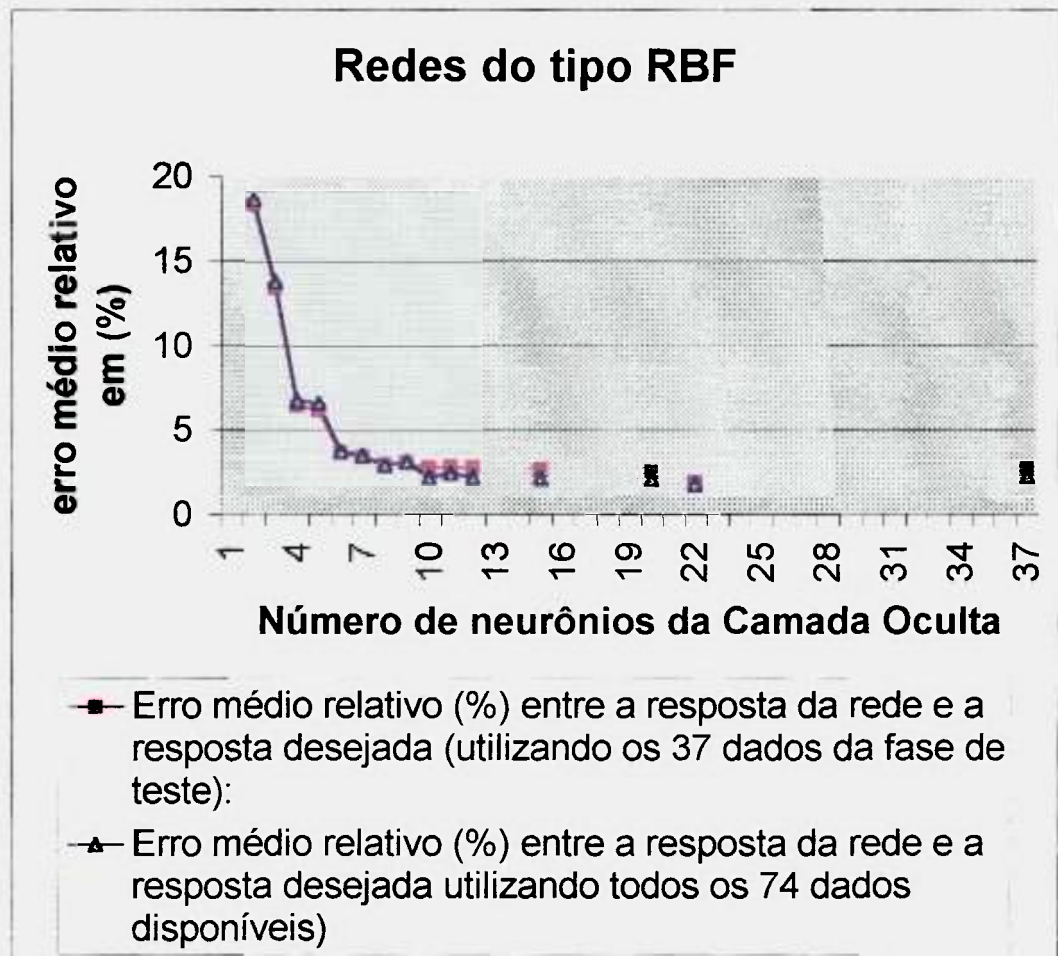


## 6.2 Resultados para redes do tipo RBF

Os resultados obtidos para as diferentes redes do tipo RBF na fase de teste é mostrado na tabela 6.3, onde é apresentado o erro médio relativo ( em %). A figura 30 mostra o gráfico do erro médio relativo ( em %) em função do número de neurônios da camada oculta para o caso das redes RBF.

**Tabela 6.3 - Resultados obtidos para as diferentes Redes do tipo RBF, em função do número de neurônios na Camada Oculta:**

<b>Nº de neurônios na camada Oculta</b>	<b>Erro médio relativo (%) entre a resposta da rede e a resposta desejada (utilizando os 37 dados da fase de teste):</b>	<b>Erro médio relativo (%) entre a resposta da rede e a resposta desejada (utilizando todos os 74 dados disponíveis)</b>
2	18,43	18,71
3	13,50	13,84
4	6,44	6,81
5	6,27	6,66
6	3,68	3,79
7	3,44	3,54
8	3,01	2,95
9	3,10	3,15
10	2,80	2,24
11	2,90	2,52
12	2,81	2,23
15	2,68	2,21
20	2,60	2,13
22	1,94	1,85
37	2,75	2,27



A Figura 30 – Erro médio relativo em função do número de neurônios da camada oculta das redes do tipo RBF.

A tabela 6.4 mostra alguns resultados típicos gerados pelas redes do tipo RBF na fase de Teste.

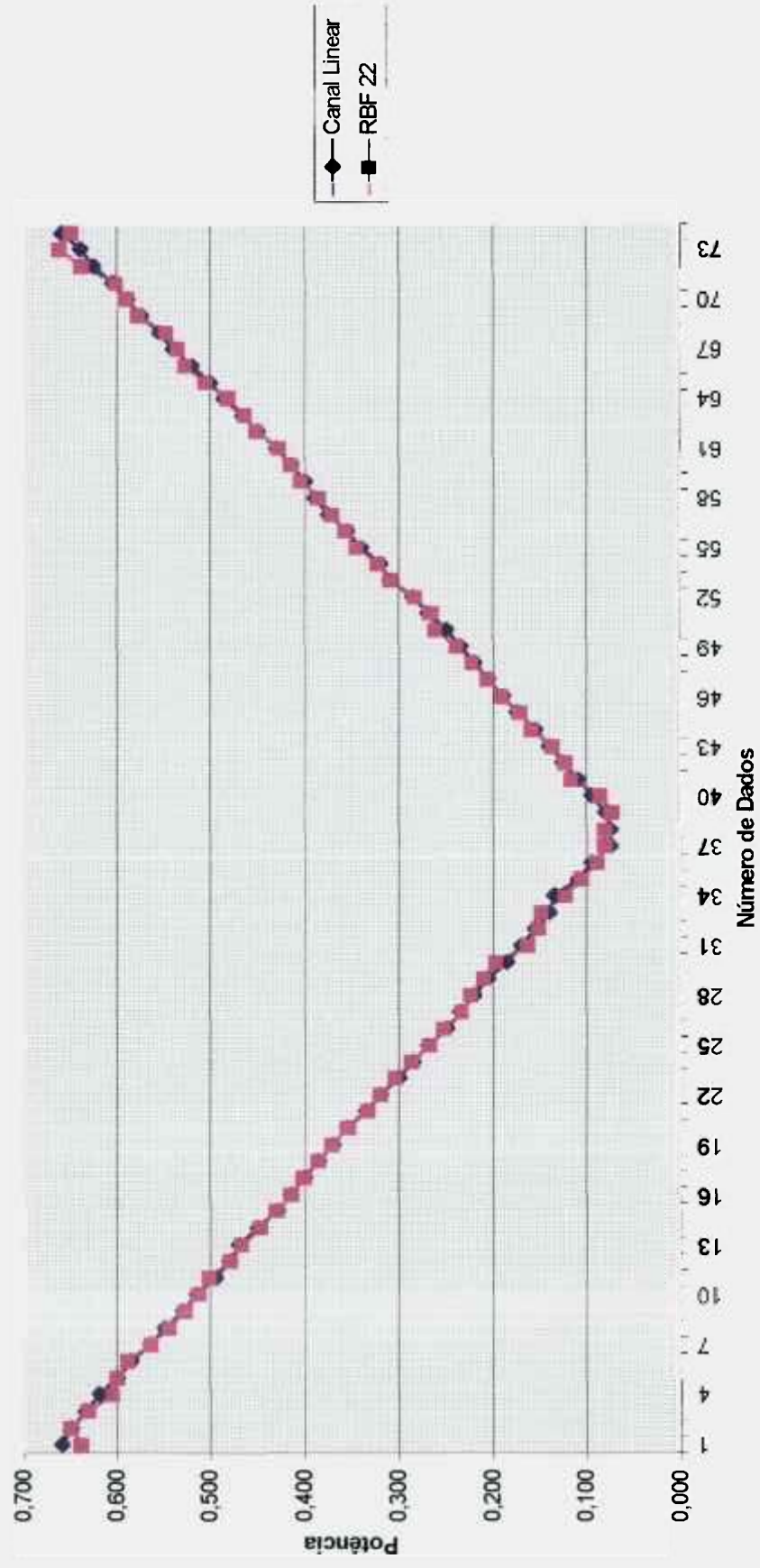
Tabela 6.4 - Respostas Típicas fornecidas pela rede do tipo RBF, na fase de Teste, em função do n° de neurônios da camada oculta

Canal Linear	Número de neurônios na Camada Oculta															
	2	3	4	5	6	7	8	9	10	11	12	15	20	22	37	
0,660	0,636	0,676	0,657	0,639	0,650	0,656	0,648	0,651	0,640	0,641	0,643	0,643	0,654	0,639	0,643	
0,650	0,638	0,668	0,638	0,622	0,643	0,643	0,637	0,640	0,644	0,643	0,643	0,645	0,648	0,650	0,648	
0,635	0,634	0,657	0,625	0,611	0,633	0,631	0,626	0,629	0,635	0,633	0,632	0,634	0,632	0,632	0,626	
0,620	0,632	0,647	0,609	0,596	0,619	0,614	0,610	0,613	0,611	0,609	0,607	0,610	0,604	0,605	0,598	
0,600	0,624	0,634	0,598	0,588	0,608	0,602	0,600	0,602	0,601	0,600	0,598	0,600	0,595	0,600	0,601	
0,585	0,611	0,617	0,585	0,578	0,594	0,588	0,587	0,589	0,591	0,590	0,588	0,588	0,586	0,589	0,596	
0,565	0,602	0,602	0,569	0,564	0,577	0,570	0,570	0,571	0,567	0,567	0,566	0,564	0,567	0,565	0,569	
0,550	0,587	0,582	0,553	0,551	0,558	0,552	0,553	0,553	0,550	0,549	0,549	0,547	0,552	0,544	0,542	
0,530	0,573	0,563	0,538	0,538	0,539	0,534	0,536	0,535	0,532	0,531	0,531	0,529	0,535	0,527	0,521	
0,515	0,564	0,547	0,520	0,522	0,520	0,515	0,516	0,515	0,510	0,509	0,510	0,509	0,515	0,513	0,516	
0,495	0,550	0,529	0,505	0,509	0,502	0,498	0,500	0,498	0,494	0,493	0,495	0,495	0,497	0,501	0,504	
0,480	0,526	0,501	0,482	0,488	0,475	0,474	0,475	0,472	0,470	0,470	0,473	0,475	0,469	0,478	0,473	
0,470	0,516	0,488	0,472	0,479	0,464	0,463	0,464	0,462	0,460	0,460	0,463	0,466	0,459	0,467	0,463	
0,450	0,500	0,468	0,454	0,462	0,445	0,446	0,446	0,443	0,444	0,445	0,448	0,451	0,444	0,447	0,449	
0,430	0,487	0,450	0,435	0,443	0,428	0,429	0,428	0,426	0,431	0,432	0,434	0,436	0,433	0,430	0,436	
0,415	0,470	0,429	0,415	0,424	0,410	0,413	0,410	0,409	0,417	0,419	0,419	0,419	0,420	0,414	0,415	
0,400	0,451	0,409	0,398	0,406	0,394	0,397	0,394	0,394	0,403	0,404	0,403	0,401	0,407	0,401	0,396	
0,385	0,430	0,387	0,377	0,386	0,377	0,381	0,377	0,378	0,387	0,387	0,384	0,380	0,387	0,385	0,382	
0,370	0,414	0,368	0,359	0,367	0,363	0,367	0,363	0,364	0,373	0,373	0,370	0,365	0,369	0,370	0,372	
0,355	0,398	0,351	0,341	0,348	0,350	0,354	0,350	0,352	0,359	0,358	0,355	0,351	0,352	0,354	0,359	
0,335	0,377	0,330	0,319	0,326	0,335	0,337	0,334	0,337	0,338	0,338	0,335	0,335	0,329	0,332	0,334	

Com base na análise da figura 30 resolvemos selecionar a rede do tipo RBF que apresentou um melhor desempenho, no caso a rede RBF com 22 neurônios na camada oculta, pois apresentou o menor erro médio relativo.

A figura 31 mostra, para cada ponto obtido, que inclui o conjunto de dados utilizados para treinamento mais o conjunto de dados utilizados para teste, o valor da potência fornecida pelo Canal Linear e o valor da potência calculada pela Rede.

Figura 31 - O valor da Potência fornecida pelo Canal Linear e o valor da Potência fornecida pela rede RBF



### **6.3 Comparação entre os resultados das redes do tipo MLP e RBF**

Comparando os dois tipos de rede analisadas com base nas figuras 28 e 30, podemos dizer que a rede do tipo RBF é a mais indicada para o aplicativo proposto, o que é confirmado pelas tabelas 6.1 e 6.4, que mostram que a rede do tipo RBF com 22 neurônios é a que apresenta menor erro médio relativo, sendo de 1,94 % para o conjunto de dados de teste e 1,85 % para o conjunto de todos os dados (dados de teste mais treinamento).

### **6.4 Generalização da Rede**

Para analisar a capacidade da rede quanto à generalização em outros dias distintos, quando as temperaturas envolvidas forem diferentes daquelas analisadas anteriormente, elaboramos uma nova base de dados acrescentando aos dados anteriormente treinados dados de 3 dias distintos, com um espectro de temperaturas diferentes. Os parâmetros de entrada considerados foram os mesmos, ou seja, a temperatura da água antes de entrar no núcleo do reator (T3), a temperatura da água após passar pelo núcleo do reator (T4), e a atividade do nitrogênio 16 ( $N^{16}$ ). Como parâmetro de saída, a rede deve fornecer a potência do reator, baseado na informação dos 3 parâmetros entrada. Os valores de porcentagem (%) de potência escalar foram coletados do Canal Linear para servir como novo padrão durante a nova fase de treinamento da rede.

A nova base de dados passou a ser composta por 92 medidas de potência, sendo 74 medidas dos dados anteriores e 18 medidas provenientes dos 3 dias distintos A, B e C, conforme indicado na tabela 6.5. A base de dados adicional foi subdividida em 2 subconjuntos, cada um

com 9 dados. O primeiro subconjunto foi utilizado para a nova fase de treinamento das redes neurais e o outro subconjunto foi utilizado para a nova fase de teste das mesmas. Em todos os casos as entradas foram normalizadas entre os valores 0 e 1.

Tabela 6.5 - Conjunto adicional de dados para teste de generalização da rede.

Dados	Valores Absolutos					Valores Normalizados			
	T3 °C	T4 °C	N16 %	POT (KW)	Canal Linear %	T3	T4	N16	Canal Linear
<b>A</b>									
1	29	31,5	24	2000	37	0,29	0,315	0,24	0,37
2	28,9	31,4	22	1900	35,5	0,289	0,314	0,22	0,355
3	28,9	31,2	21,5	1800	33,5	0,289	0,312	0,215	0,335
4	28,8	30,5	16	1300	25	0,288	0,305	0,16	0,25
5	28,6	30,1	14	1100	22	0,286	0,301	0,14	0,22
6	28,3	29,5	10	700	15,5	0,283	0,295	0,1	0,155
<b>B</b>									
1	25,1	25,5	22,5	1900	35,5	0,251	0,255	0,225	0,355
2	25,1	27,2	19,5	1600	30	0,251	0,272	0,195	0,300
3	25	26,9	16,5	1300	25	0,250	0,269	0,165	0,250
4	25	26,5	14	1100	22	0,250	0,265	0,140	0,220
5	24,8	26,1	10,5	700	15,5	0,248	0,261	0,105	0,155
6	24,4	25,1	5	200	7,5	0,244	0,251	0,050	0,075
<b>C</b>									
1	27,5	30	23,5	2000	37	0,275	0,300	0,235	0,370
2	27,4	29,9	24	2000	37	0,274	0,299	0,240	0,370
3	27,6	30	22,5	1900	35,5	0,276	0,300	0,225	0,355
4	27,6	29,9	22,5	1900	35,5	0,276	0,299	0,225	0,355
5	27,5	29,8	21,5	1800	33,5	0,275	0,298	0,215	0,335
6	27,5	29,8	21,5	1800	33,5	0,275	0,298	0,215	0,335

No teste de generalização, inicialmente utilizou-se o novo conjunto de dados mantendo as duas redes, MLP-15 e RBF-22, com os mesmos parâmetros (pesos) calculados anteriormente. Os resultados obtidos para ambas as redes é mostrado nas figuras 32 e 33.

Numa segunda fase, a base de dados adicional foi subdividida em 2 subconjuntos, cada um com 9 dados, sendo 3 de cada dia. O primeiro subconjunto foi utilizado para uma nova fase de treinamento das redes neurais, partindo dos parametros (pesos) calculados anteriormente, e o outro subconjunto foi utilizado para a nova fase de teste das mesmas.

As figuras 34 e 35 mostram, para cada tipo de rede selecionada o resultado final do teste de generalização, considerando-se o conjunto total de 92 dados.

Figura 32 - O valor da Potência fornecida pelo Canal Linear e o valor da Potência fornecida pela rede MLP, com 15 neurônios na camada oculta, utilizando novo conjunto de dados, antes do novo Treinamento

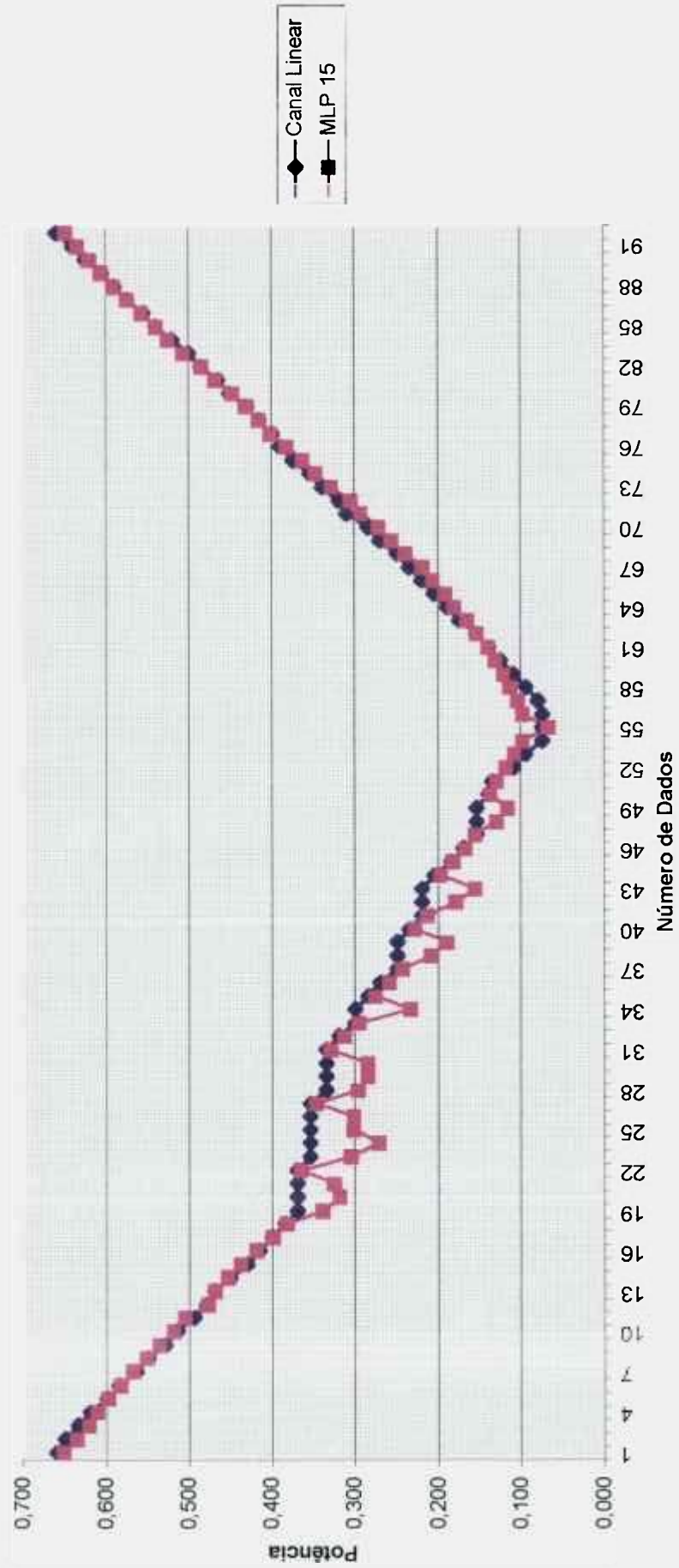


Figura 33 - O valor da Potência fornecida pelo Canal Linear e o valor da Potência fornecida pela rede RBF, com 22 neurônios na camada oculta, utilizando novo conjunto de dados, antes do Treinamento

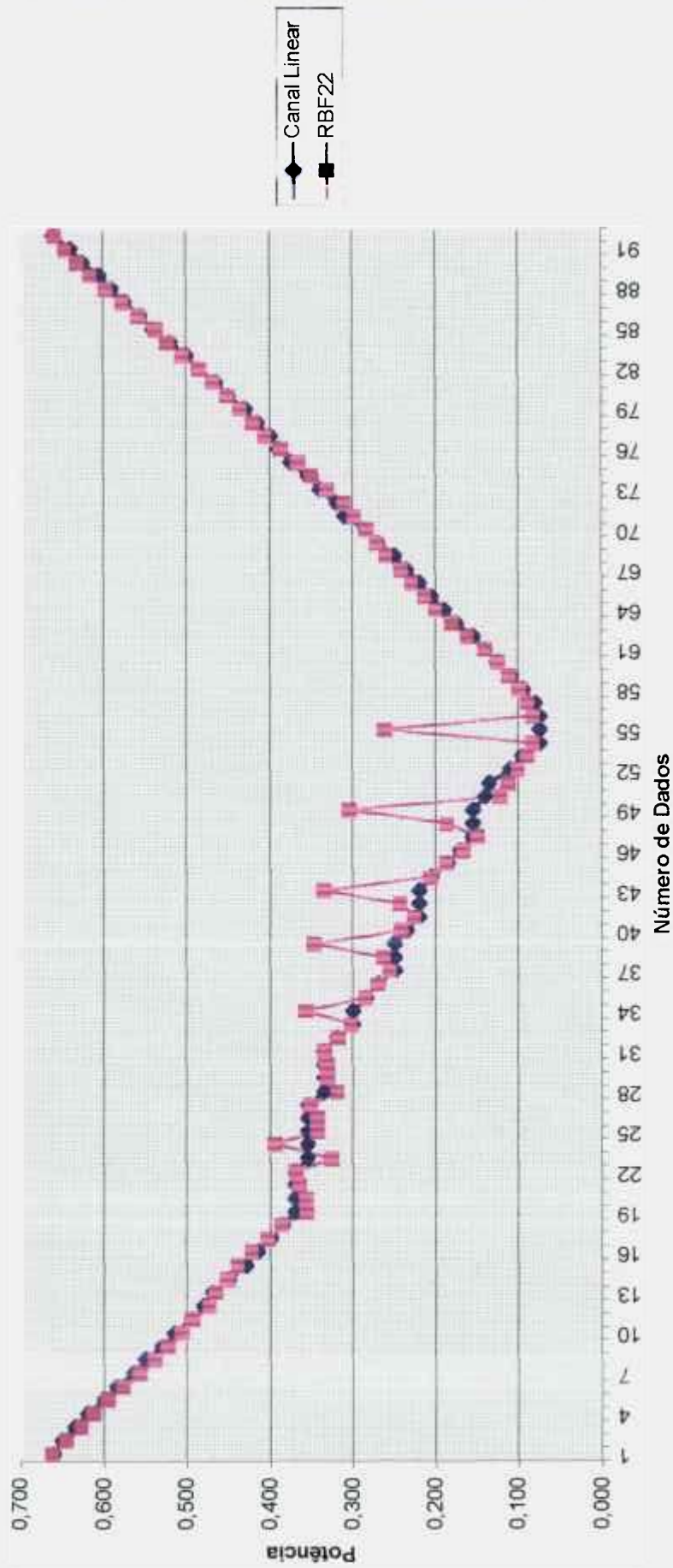


Figura 34 - O valor da Potência fornecida pelo Canal Linear e o valor da Potência fornecida pela rede MLP, com 15 neurônios na camada oculta, utilizando novo conjunto de dados, após o novo Treinamento

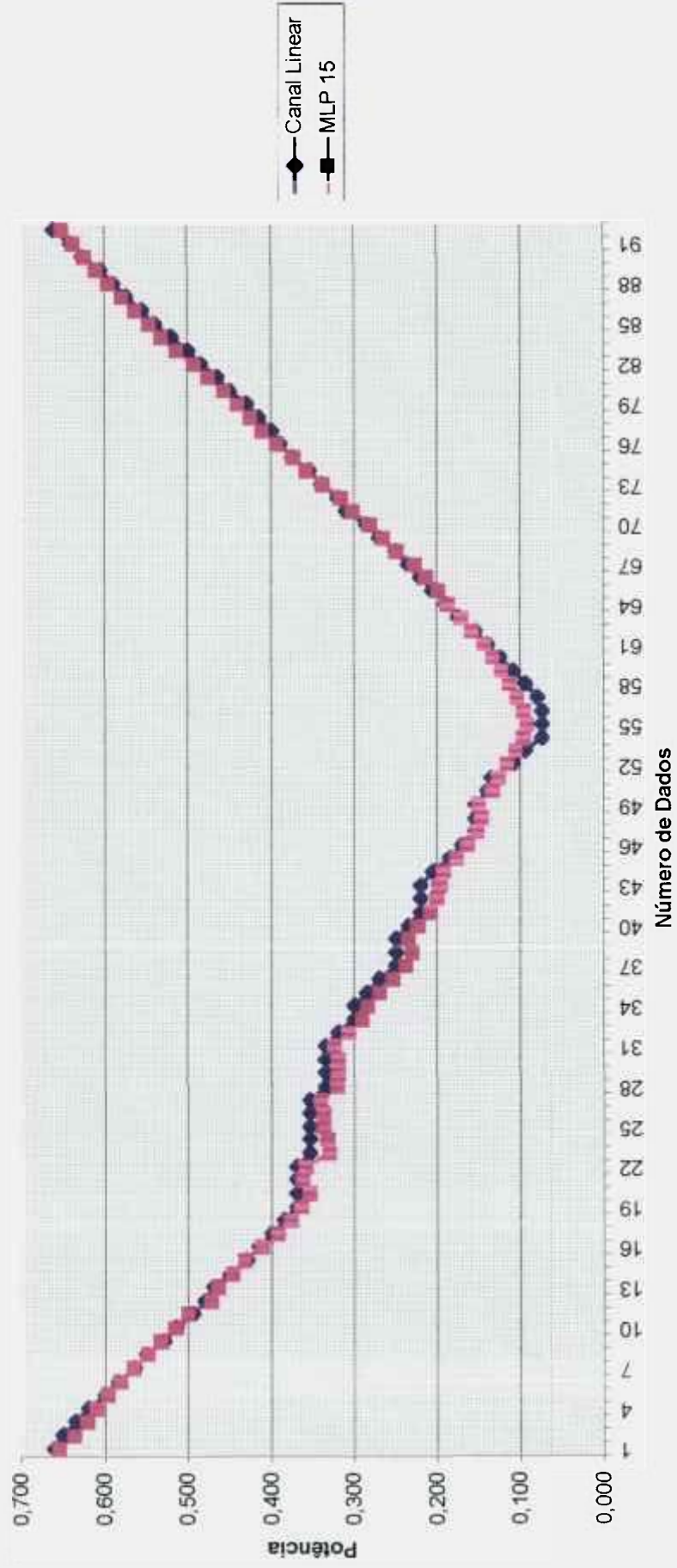
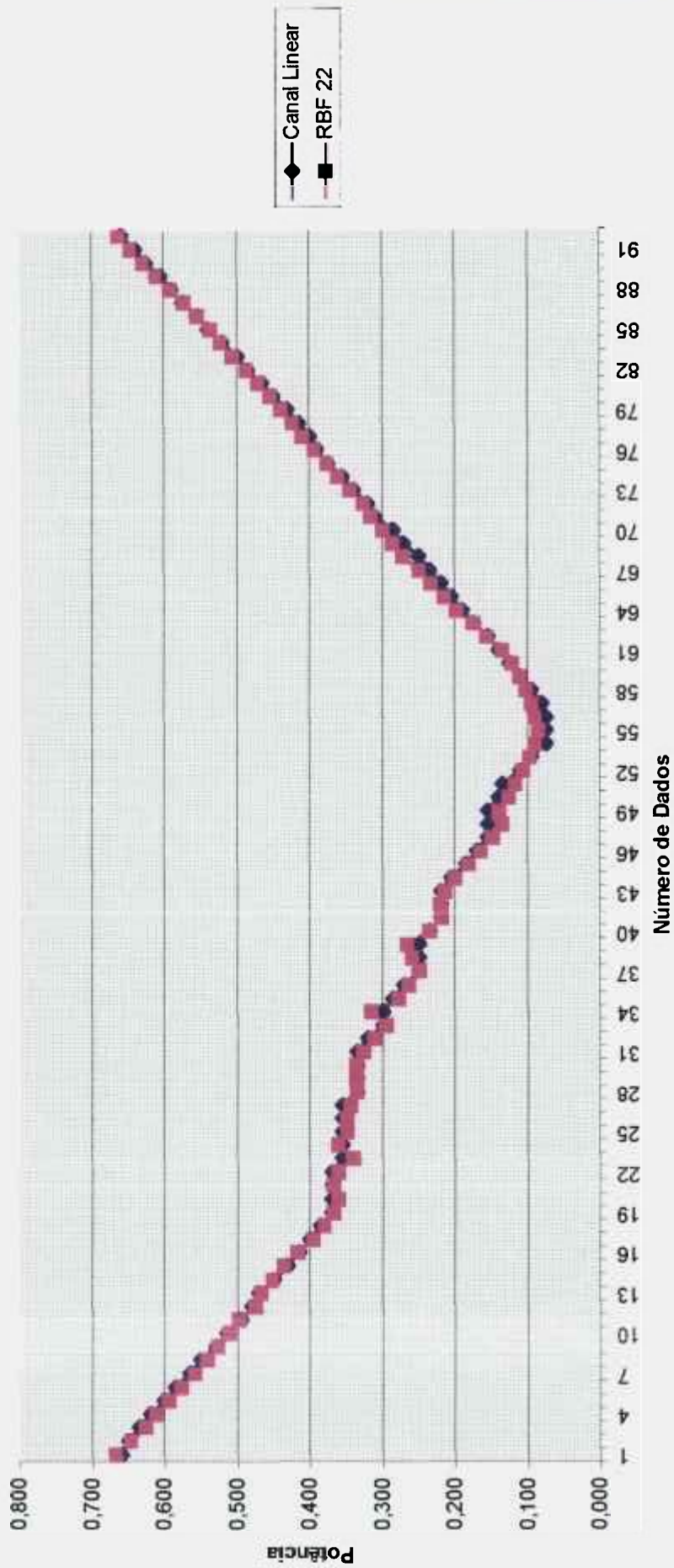


Figura 35 - O valor da Potência fornecida pelo Canal Linear e o valor da Potência fornecida pela rede RBF, com 22 neurônios na camada oculta, utilizando novo conjunto de dados, após o novo Treinamento



Analisando as figuras 32 a 35 podemos concluir que, no caso do aplicativo proposto, o poder de generalização das redes depende muito do universo de dados utilizado na fase de treinamento. A tabela 6.6 apresenta o erro médio relativo em (%) para os dados do conjunto adicional, antes e depois do re-treinamento da rede.

**Tabela 6.6 - Resultados obtidos para as diferentes redes com o conjunto adicional de dados, antes e depois do novo treinamento.**

Tipo de Rede		Erro médio relativo (%) entre a resposta da rede e a resposta desejada (utilizando os 46 dados da fase de teste):	Erro médio relativo (%) entre a resposta da rede e a resposta desejada (utilizando todos os 92 dados disponíveis)
MLP -	antes do novo treinamento	6,56	6,26
MLP -	após o novo treinamento	4,45	4,30
RBF -	antes do novo treinamento	3,68	7,62
RBF -	após o novo treinamento	3,15	2,96

## **7. Conclusões e Recomendações**

Tendo em vista os resultados apresentados na seção anterior, podemos concluir que é plenamente viável a aplicação de uma rede neural para monitorar, de forma diversional, a potência do reator IEAR1. Assim sendo, a sua utilização, associada à tela desenvolvida e ilustrada na figura 27, permite alertar o operador do reator sobre a existência de desvios entre os diversos canais utilizados para monitorar a potência do reator, de forma que o mesmo possa tomar as ações apropriadas para sanar o problema.

Comparando os resultados mostrados nas tabelas 6.1 e 6.4, e nas figuras 28 e 30, podemos concluir que para o caso analisado, de monitoração da potência do reator, a rede mais apropriada é a rede do tipo RBF, com 22 neurônios na camada oculta (intermediária). Apesar da rede do tipo MLP com 15 neurônios na camada intermediária ter apresentado um erro considerado pequeno, da ordem de 3,10%, o erro obtido com a rede do tipo RBF foi muito inferior, atingindo 1,85% no universo dos dados utilizados.

A fim de tornar o aplicativo mais completo, recomenda-se a continuidade deste trabalho, de forma a aumentar a arquitetura da rede para que a mesma tenha condições de identificar e quantificar além da potência nuclear, outras condições operacionais tais como reator operando em condição normal (regime estacionário, regime de transiente com potência ascendente, ou regime de transiente com potência descendente) ou reator operando em condição degradada, identificando, através de uma segunda rede, a condição de degradação.

Outra recomendação é a implantação de um sistema de aquisição de dados que permita a coleta e o processamento dos diversos parâmetros da instalação “on-line” e em tempo real. Para tanto deve ser adquirida uma interface que permita a aquisição dos dados da instalação para que os mesmos possam ser devidamente processados.

## **8. Referências Bibliográficas**

- [1] Kessie, Marshall K, **The ‘aftermath’ nuclear control room**, Instrumentation and Control, Power, abtil 1981
- [2] Filho Baptista, B. D., **Redes Neurais para Controle de Sistemas de Reatores Nucleares**, Tese de Doutorado, S. Paulo 1998.
- [3] Soares, Adalberto J., **Aplicação de Sistemas Digitais para Controle e Proteção de Centrais nucleares**, Apostila de Aula.
- [4] Monta, K., et al ii, “Computerized operator support system with new man-machine interface for BWR power plants”, Proc. of a Symposium on Operational Safety of Nuclear Power Plants, IAEA-1983.
- [5] IAEA, **Nuclear power plant instrumentation and Control, a guidebook**, IAEA Technical Reports Series n°239
- [6] Asani, K., et alii, **Development of high-reability control system for nuclear power plants**, Proc. Of a Symposium on Nuclear Power Plant control and Instrumentation, IAEA-1982.
- [7] Moore, V. A.,**Experience in the review of utility control room design review and safety parameter display system programs**, IEEE TRANSACTIONS ON Nuclear Science, vol NS-32, n° 1, Fev. 1985.

- [8] Tetreau, F. & Parry, A ., **“Framatone proposes a package approach to revamping”**, Nuclear Eng. Intl. Vol 33, nº406, pp.25-29, maio-1988.
- [9] Richelle, G. & Boucau, J. & Carrera, J., **Westinghouse Computer – Based Operator Support Systems**, Operator support systems in nuclear power plants, TECDOC-762, IAEA, setembro de 1994.
- [10] Embrechts, M. J. & Beneck, S., **Identification of nuclear power plant transients with neural networks**, IEEE, 1997 5 vol. 4535 pp., New York, USA.
- [11] Lima, Alan M. M. de & Schirru, Roberto, **Modelo de Ilhas para Implementação Paralela do Algoritmo Evolucionário de Otimização PBIL**, XII ENFIR 1999, Rio de Janeiro – RJ.
- [12] Theibert, Phil, **Setting records- always room to improve**, Nuclear Engineering International, V.43 , N526-1998.
- [13] Staff Report, **Taking to open road in South Texas and Vandellos**, Nuclear Engineering International, V.43 , N522-1998.
- [14] AECL Atomic Energy of Canada Limited, **Energy for the next millennium** , Nuclear Engineering International, V.43 , N531-1998.
- [15] **RAS IEA-R1**, Rev.3, 6/97

- [16] Informativo do IPEN Reator IEAR-1
- [17] Sousa, Joel Alvarenga de, **Sobre a Monitoração da Potência do Reator nuclear IEAR-1 pela Atividade do N-16**, Dissertação e Tese IEA 011.
- [18] Cárdenas, José Patricio N., **Monitoração Contínua do Balanço Térmico para Determinação da potência do Reator IEA-R1**, XII ENFIR, 2000.
- [19] Kovács, Zsolt L., **Redes Neurais Artificiais: Fundamentos e Aplicações**, 2. Ed. – Edição Acadêmica, São Paulo, 1996.
- [20] Braga, A . de P. & Ludemir, T. B. & Carvalho, A . C. P. de L. F. , **Redes Neurais e Artificiais Teoria e Aplicações**, Livros Técnicos e Científicos Editora, Rio de Janeiro- RJ, 2000.
- [21] Tafner, M. A. , **Redes Neurais Artificiais: Aprendizado e Plasticidade**, Revista "Cérebro & Mente" 2(5), Março/Maio 1998.
- [22] Timoszczuk, P.A. & Cabral Jr., E. F. **RBF Neural Networks and MTI for text Independent Speaker Identification.**
- [23] Fausett, Laurene, **Fundamentals of Neural Networks Architectures, Algorithms, and Applications**, New Jersey 1994.

- [24] Barreto, Jorge Muniz, **Redes Neurais Fundamentos e Aplicações**, II SBAI-Segundo Simpósio Brasileiro de Automação Inteligente, Curitiba, Setembro 1995.
- [25] Tsoi, A.C.. **Multilayer perceptron trained using radial basis functions**. *Electronic Letters* 25(9) 1989.
- [26] Hush, D. R. & Horne, B. G., **Progress in supervised neural networks: What's new since Lippmann?**, IEEE Signal Processing Magazine, v.10, n.1, p.8-39, jan.1993.
- [27] Cabral Jr., E.F., **Redes Neurais Artificiais**. Ed. Autor. 324p., 1999.
- [28] Pandya, S. A. & Macy, Robert B. Macy, **Pattern Recognition with Neural Networks in C++**, Florida Atlantic University Boca Raton, Florida 1995.
- [29] Haykin, S., **Neural networks: a comprehensive foundation**, New York , MacMillan, 1994.
- [30] Kee, F. J., Smith, **“Development of a reliable single-Channel direct digital reactor control system”**, Proc. Of a Symposium on Nuclear Power Plant Control and Instrumentation, IAEA – 1978.
- [31] **Nuclear Engineering**, maio de 1973
- [32] Committed of Energy – **Toshiba Nuclear Energy**

## APÊNDICE I

Os programas das redes MLP e RBF desenvolvidos foram baseados nos algoritmos encontrados na referência [27].

```
-----
*
      Helena Rossi
      Multi Layer Perceptron (MLP), aplicado no Reator de
      Pesquisa IEA_R1 para predição de Potencia
-----
*/

#include <time.h>
#include <math.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
#include <process.h>

/***** CONFIGURACAO DO MLP
*****/

#define ni          3 /*No. de ptos de entrada da rede
*/
#define nf          3 /*No. de neurons da First layer, = n_i
*/
#define nh          15 /*No. de neurons da Hidden layer
*/
#define nl          1 /*No. de neurons da Last layer, No. target
*/

#define a_n         2.0/*0.9 /*0.01*/ /*0.9*/
#define b_n         0.0 /*04 /*0.0 /*n = learning rate = an * exp(-bn.t)
* t^-cn */
#define c_n         0.0 /*4 /*0.01 /*0.01*/

#define a_ph         2.0 /* 0.9*/
#define b_ph         0.0 /*0.0 /*alpha = coef. momento (idem learning
rate) */
#define c_ph         0.0 /*0.0*/

#define a_nf         1.0 /*          a_nf + b_nf
*/
#define b_nf         0.0 /*NF=----- - b_nf, N. Function
*/
#define beta        1.0 /*          1 + exp(-beta.i)
*/

#define Eps         0.001 /*Erro puxado a zero, se erro < Eps => ok
*/

#define MAXITERACAO 10000 /*Maximo tempo de treinamento
*/

#define peso        0.0 /* variavel de inicialização do peso */
```

```

#define TREINAMENTO "TR"
#define TESTE "TE"
/*****
*/

/***** DEFINICAO DAS VARIAVEIS GLOBAIS
*****/

int usa_BIAS = 1; /*Defini se usaremos ou nao o bias
*/
int usa_BIAS_f = 0; /*Defini se usaremos ou nao o bias na primeira
camada*/

float Wfh[nf][nh]; /*Matriz de pesos entre as camadas F e H
*/
float Whl[nh][nl]; /*Matriz de pesos entre as camadas H e L
*/

float D_Wfh[nf][nh]; /*Matriz Delta de pesos entre as camadas F e H
*/
float D_Whl[nh][nl]; /*Matriz Delta de pesos entre as camadas H e L
*/

float Bf[nf]; /*Vetor de bias da camada F
*/
float Bh[nh]; /*Vetor de bias da camada H
*/
float Bl[nl]; /*Vetor de bias da camada L
*/

float D_Bh[nh]; /*Vetor Delta de bias da camada H
*/
float D_Bl[nl]; /*Vetor Delta de bias da camada L
*/

float I[ni]; /*Vetor de entrada da rede
*/
float Fi[nf]; /*Vetor de entrada da camada F
*/
float Fo[nf]; /*Vetor de saida da camada F
*/
float Hi[nh]; /*Vetor de entrada da camada H
*/
float Ho[nh]; /*Vetor de saida da camada H
*/
float Li[nl]; /*Vetor de entrada da camada L
*/
float Lo[nl]; /*Vetor de saida da camada L
*/
float T[nl]; /*Vetor de target da rede
*/
float E[nl]; /*Vetor de erro da rede
*/
float Eac[nl]; /*Vetor de erro acumulado da rede
*/

float sai=0.0; /*Saida */
float saides=0.0; /*Saida Desejada*/

```

```

float n_lr;          /*learning rate
*/
float alpha;        /*coeficiente de momento
*/
float erro = 0.0;   /*erro da pattern atual
*/
float erro_EP = 0.0; /*err de toda a epoca
*/
int t_epc = 1;      /*tempo, epoca
*/
int n_batch_atual = 0; /*no. que conta em que iteracao estamos, batch
*/
int n_batch_total = 1; /*no. para atualizacao batch dos pesos
*/

char nomearqIN[20]; /*nome do arquivo de entrada
*/
FILE *fileIN;      /*arquivo de entrada
*/
char nomearqWB[20]; /*nome do arquivo dos pesos e dos bias da rede
*/
FILE *fileWB;      /*arquivo de pesos e bias
*/
char nomearqOUT[20]; /*nome do arquivo de saida ou target
*/
FILE *fileOUT;     /*arquivo de saida ou target
*/
int narqIN;        /*numero de arquivos do arquivo de entrada
*/
int n_IN_atual = 1; /*numero do arquivo de entrada atual
*/
FILE *fout;        /*arquivo de saida ou target */

char nomearqWERR[20]; /* Nome do arquivo de erro */
FILE *fileWERR;

/*****
*/

/***** ROTINAS DE INPUT/OUTPUT DE DADOS
*****/

void Input(void)
/*Funcao que tem por finalidade abrir o arquivo de entrada e atualizar o
numero
de patterns de entrada da rede.*/
{
    if ( (fileIN = fopen(nomearqIN,"rt")) == NULL )
    {
        printf ("ERRO1 - arquivo ENTRADA\n\n");
        exit(0);
    }
    fscanf(fileIN,"%d",&narqIN);

    if ( (fileOUT = fopen(nomearqOUT,"rt")) == NULL )
    {
        printf ("ERRO2 - arquivo SAIDA\n\n");
        exit(0);
    }
    fscanf(fileOUT,"%d",&narqIN);
    return;
}

```

```

void Carregar_I_T(void)
/*Funcao que tem por finalidade carregar os vetores I e T. Nos
arquivos
nomearqIN e nomearqOUT, respectivamente estao os nomes dos arquivos a
serem
lidos. Deve ser feita a verificacao de voltar ou nao para o primeiro
arquivo
de I e T.*/

char auxin[15];
char auxt[15];
int cont;

float ent,pad;

if ( n_IN_atual > narqIN )/*Devemos voltar ao primeiro*/
{
fclose(fileIN);
fclose(fileOUT);
if ( (fileIN = fopen(nomearqIN,"rt")) == NULL )
{
printf ("*ERRO3 - arquivo ENTRADA\n\n");
exit(0);
}
fscanf(fileIN,"%d",&narqIN);

if ( (fileOUT = fopen(nomearqOUT,"rt")) == NULL )
{
printf ("*ERRO4 - arquivo SAIDA\n\n");
exit(0);
}
fscanf(fileOUT,"%d",&narqIN);
n_IN_atual = 1;

n_IN_atual++;

for ( cont = 0 ; cont < nf ; cont++ )
{

fscanf(fileIN,"%f",&ent);
I[cont] = ent;

for ( cont = 0 ; cont < nl ; cont++ )
{
fscanf(fileOUT,"%f",&pad);
T[cont] = pad;

/* if (cont>0)
{
printf("EPOCA : %d \t\tERRO : %f\t\t ENTR: %d %f %f \n",
t_epc, erro_EP, last, ent, pad);

exit(1);
}
}
}

```

```

    }

    return;
}

void Fecha_I_T(void)
/*Funcao que tem por finalidade fechar os arquivos nomearqIN e nomearqOUT.*/
{
    fclose(fileIN);
    fclose(fileOUT);
    return;
}

void CarregaIN_Teste(void)
/*Funcao que tem por finalidade carregar o vetor de entrada da rede.*/
{
    int cont;
    float aux;

    /*NAO E NECESSARIO FAZER A VERIFICACAO SE HA AINDA ARQUIVOS A SEREM
LIDOS,*/
    /*ISTO JA E FEITO NA ROTINA TESTE.
*/

    for ( cont = 0 ; cont < nf ; cont++ )
    {
        fscanf(fileIN , "%f", &aux);
        I[cont] = aux;
    }

    return;
}

void SalvaOUT_Teste(void)
/*Funcao que tem por finalidade salvar a saida da rede.*/
{
    char auxout[15];
    int cont;
    float aux;

    fclose(fileOUT);

    if ( (fileOUT = fopen(nomearqOUT, "wt")) == NULL )
    {
        printf ("ERRO 8 - arquivo saida\n\n");
        exit(0);
    }

    fscanf(fileOUT, "%d", &narqIN);
    /*
        fprintf(fauxout, "%d\n", nl);
    for ( cont = 0 ; cont < nl ; cont++ )
    {
        aux = Lo[cont];
        fprintf(fauxout, "%f\n", aux);
    }
*/
}

```

```

        fclose(fauxout);*/

    return;
}

/* testando salvar saida*/

void Saida(void)
/*Funcao que tem por finalidade salvar a saida da rede.*/
{
    char auxout[15];
    int cont;
    float aux;

    /*    fprintf(fileOUT,"%d\n",nl);*/
    for ( cont = 0 ; cont < nl ; cont++ )
    {
        aux = Lo[cont];

        fprintf(fileOUT,"%f\n",aux);
        ;
    }
    return;
}

void Fecha(void)
/*Funcao que tem por finalidade fechar os arquivos nomearqIN e nomearqOUT.*/
{
    fclose(fout);
    return;
}

void Fecha_I_OUT(void)
/*Funcao que tem por finalidade fechar os arquivos nomearqIN e nomearqOUT.*/
{
    fclose(fileIN);
    fclose(fileOUT);
    return;
}

void Salva_W_B(void){
/*Funcao que tem por finalidade salvar nas devidas variaveis os pesos e
bias
correspondentes.*/
    int f, h, l;
    float aux;

    if ( (fileWB = fopen(nomearqWB, "wt")) == NULL ){
        printf ("ERRO55 - arquivo WB\n\n");
        exit(0);
    }

    for ( f = 0 ; f < nf ; f++){
        for ( h = 0 ; h < nh ; h++){
            aux = Wfh[f][h];
            fprintf(fileWB,"%f\n",aux);
        }

        for ( h = 0 ; h < nh ; h++)

```

```

        for ( l = 0 ; l < nl ; l++){
            aux = Whl[h][l];
            fprintf(fileWB,"%f\n",aux);
        }

    if ( usa_BIAS ){
        if( usa_BIAS_f ){
            for ( f = 0 ; f < nf ; f++){
                aux = Bf[f];
                fprintf(fileWB,"%f\n",aux);
            }
            for ( h = 0 ; h < nh ; h++){
                aux = Bh[h];
                fprintf(fileWB,"%f\n",aux);
            }

            for ( l = 0 ; l < nl ; l++){
                aux = Bl[l];
                fprintf(fileWB,"%f\n",aux);
            }
        }

        fclose(fileWB);
        return ;
    }

void Carregar_W_B(void){
/*Funcao que tem por finalidade carregar nas devidas variaveis os pesos e
bias
correspondentes.*/
    int f, h, l;
    float aux;

    if ( (fileWB = fopen(nomearqWB, "rt")) == NULL ){
        printf ("ERRO66 - arquivo WB\n\n");
        exit(0);
    }

    for ( f = 0 ; f < nf ; f++){
        for ( h = 0 ; h < nh ; h++){
            fscanf(fileWB, "%f",&aux);
            Wfh[f][h] = aux;
        }

        for ( h = 0 ; h < nh ; h++){
            for ( l = 0 ; l < nl ; l++){
                fscanf(fileWB, "%f",&aux);
                Whl[h][l] = aux;
            }
        }

        if ( usa_BIAS ){
            if( usa_BIAS_f ){
                for ( f = 0 ; f < nf ; f++){
                    fscanf(fileWB, "%f",&aux);
                    Bf[f] = aux;
                }
            }

            for ( h = 0 ; h < nh ; h++){
                fscanf(fileWB, "%f",&aux);
                Bh[h] = aux;
            }
        }
    }
}

```

```

        for ( l = 0 ; l < nl ; l++){
            fscanf(fileWB, "%f",&aux);
            Bl[l] = aux;
        }
    }
    fclose(fileWB);
    return ;
}
/*****
*/
/*****
INICIALIZACOES*****/

void calc_lr(void){
/*Funcao que tem por finalidade calcular a lerning rate.*/
    n_lr = a_n * exp( -b_n*t_epc) * exp(-c_n * log(t_epc)) ;
    return;
}

void calc_alpha(void){
/*Funcao que tem por finalidade calcular a alpha, coeficiente de momento.*/
    alpha = a_pha * exp( -b_pha*t_epc) * exp(-c_pha * log(t_epc)) ;
    return;
}

void Ini_pesos_e_bias(void){
/*Funcao que tem por finalidade inicializar os pesos e bias.*/
    int f, h, l;
    float aux;
    time_t t;

    srand((unsigned) time(&t));

    for ( f = 0 ; f < nf ; f++)                /* camada primeira*/
        for ( h = 0 ; h < nh ; h++){
            aux=peso;
            Wfh[f][h] = aux;
            D_Wfh[f][h] = 0.0; /*matriz delta entre f e h*/
        }

    for ( h = 0 ; h < nh ; h++)                /*camada oculta*/
        for ( l = 0 ; l < nl ; l++){
            aux=peso;
            Whl[h][l] = aux;
            D_Whl[h][l] = 0.0; /*matriz delta entre h e l*/
        }

    if ( usa_BIAS ){                            /* bias da primeira camada*/
        if( usa_BIAS_f ){
            for ( f = 0 ; f < nf ; f++){
                aux=peso;
                Bf[f] = aux;
            }
        }
        for ( h = 0 ; h < nh ; h++){ /* bias da camada h*/
            aux=peso;
            Bh[h] = aux;
            D_Bh[h] = 0.0; /* matriz delta da camada h*/
        }

        for ( l = 0 ; l < nl ; l++){ /* bias da camada l*/
            aux=peso;
            Bl[l] = aux;
        }
    }
}

```

```

        D_Bl[1] = 0.0; /* matriz delta da camada h*/
    }
}

t_epc = 1;
calc_lr();
calc_alpha();

return;
}
/*****
*/

/***** NEURAL FUNCTION
*****/

float NF_linear(float a, float in){
/*Funcao que tem por finalidade ser a NEURAL FUNCTION da camada com
caracteris-
ticas lineares.*/
    return (a*in);
}

float NF_sig(float in){
/*Funcao que tem por finalidade ser a NEURAL FUNCTION da camada com
caracteris-
ticas sigmoides, geral.*/
    float aux;

/* Neste caso tangente hiperbólica*/

/*    if (in<-100)
    {
        aux= -(1-exp(2*in))/(1+exp(2*in));
    }
    else
        aux= (1-exp(-2*in))/(1+exp(-2*in));
    return(aux);

*/

/* Neste caso sigmoide*/

    aux=1.0+exp(-(beta*in));
    return(((a_nf+b_nf)/aux)-b_nf);

}

float d_NF_linear(float a){
/*Funcao que tem por finalidade derivada da NEURAL FUNCTION da camada com
carac-
teristicas lineares, geral.*/
    return (a);
}

float d_NF_sig(float in){
/*Funcao que tem por finalidade derivada da NEURAL FUNCTION da camada com
carac-
teristicas sigmoides, geral.*/

```

```

float auxn, auxd;

/*Neste caso derivada tangente hiperbólica */
/*
auxn=(4*exp(-fabs(2*in)))/(pow((1+exp(-fabs(2*in))),2));
return(auxn);*/

/* sigmoide*/
auxn=beta*(a_nf+b_nf)*exp(-(beta*in));
auxd=(1.0+exp(-(beta*in)))*(1.0+exp(-(beta*in)));
return(auxn/auxd);
}

float NF_f(float in)
/*Funcao que tem por finalidade ser a NEURAL FUNCTION da camada e.*/
{
return (NF_linear(1.0,in));

/*return (NF_sig(in));*/
}

float d_NF_f(float in)
/*Funcao que tem por finalidade ser a DERIVADA da NEURAL FUNCTION da
camada
First.*/
{
return (d_NF_linear(1.0,in));

/*return (d_NF_sig(in));*/
}

float NF_h(float in)
/*Funcao que tem por finalidade ser a NEURAL FUNCTION da camada Hidden.*/
{
return (NF_sig(in));
}

float d_NF_h(float in)
/*Funcao que tem por finalidade ser a DERIVADA da NEURAL FUNCTION da
camada
Hidden.*/
{
return (d_NF_sig(in));
}

float NF_l(float in)
/*Funcao que tem por finalidade ser a NEURAL FUNCTION da camada Last.*/
{
return (NF_sig(in));
}

float d_NF_l(float in)
/*Funcao que tem por finalidade ser a DERIVADA da NEURAL FUNCTION da
camada
Last.*/

```

```

{
    return (d_NF_sig(in));
}
/*****
*/
/***** FUNCOES DAS
LAYERS*****/

void First_layer(void)
/*Funcao que tem por finalidade fazer os calculos dos vetores de entrada
e
saida da FIRST LAYER.*/
{
    int f;

    for ( f = 0 ; f < nf ; f++)
    {
        Fi[f] = I[f] + (usa_BIAS_f*Bf[f]);
        Fo[f] = NF_f(Fi[f]);
    }
    return;
}

void Hidden_layer(void)
/*Funcao que tem por finalidade fazer os calculos dos vetores de entrada
e
saida da HIDDEN LAYER.*/
{
    int h, f;
    float soma;

    for ( h = 0 ; h < nh ; h++)
    {
        soma = 0.0;
        for ( f = 0 ; f < nf ; f++ )
            soma += ( Wfh[f][h] * Fo[f] );

        Hi[h] = soma + (usa_BIAS*Bh[h]);
        Ho[h] = NF_h( Hi[h] );
    }
    return;
}

void Last_layer(void)
/*Funcao que tem por finalidade fazer os calculos dos vetores de entrada
e
saida da LAST LAYER.*/
{
    int h, l;
    float soma;

    for ( l = 0 ; l < nl ; l++)
    {
        soma = 0.0;
        for ( h = 0 ; h < nh ; h++ )
            soma += ( Whl[h][l] * Ho[h] );

        Li[l] = soma + (usa_BIAS*Bl[l]);
        Lo[l] = NF_l( Li[l] );
    }
    return;
}

```

```

/*****
*/

/***** FUNCOES DE
ERRO*****/

void Calculo_Erro(void)
/*Funcao que tem por finalidade calcular o E = T -Lo, erro e erroac*/
{
    int l;
    float erro;

    for ( l = 0; l < nl ; l++)
    {
        E[l] = T[l] - Lo[l];
        sai=Lo[l];
        saides=T[l];
        erro= fabs(E[l]);

/*          printf("EPOCA : %d \t\tERRO : %f\t\t SAI:%f SAIDES:%f  \n",
t_epc, erro, sai, saides);*/

    }

    n_batch_atual++;

}

int Atualiza_Ind_Lote(void)
/*Funcao que tem por finalidade atualizar o vetor erro e verificar se
devemos
atualizar os pesos agora ou nao, para processamento em lote.*/
{
    int l;
    int atualiza = 0;
    float soma=0.0;

    if ( n_batch_atual == 1 )
        for ( l = 0 ; l < nl ; l++ )
        {
            Eac[l] = E[l];
            soma += fabs(E[l]);

/*          adicao[l]=E[l];*/
        }
    else
        for ( l = 0 ; l < nl ; l++)
        {
            Eac[l] += E[l];
            soma += fabs(E[l]);
        }

    soma /= (float)nl;
    if ( n_IN_atual == 2 )/* primeira vez*/
        erro_EP = soma;
    else

```

```

        erro_EP += soma;

/*if ( n_batch_atual == 1 )
    erro_EP = soma;
else
    erro_EP += soma;*/

if ( n_batch_atual == n_batch_total )
    atualiza = 1;

return (atualiza);
}
/*****
*/

/***** ATUALIZACOES
*****/

void Atualiza_Param(void)
/*Funcao que tem por finalidade atualizar os paramentros learning_rate
e
alpha.*/
{
    _epc++;
    calc_lr();
    calc_alpha();
    return;
}

void Atualiza_W_B(void)
/*Funcao que tem por finalidade atualizar os pesos e bias.*/
{
    int f, h, l;
    float d_Whl_Bl[nl];
    float d_aux, D_aux, soma_aux;

/*Atualizacao dos PESOS - Whl ___ BIAS - Bl
*/
    for ( l = 0 ; l < nl ; l++ )
    {
        d_Whl_Bl[l] = Eac[l] * d_NF_l(Li[l]);
        /*Whl*/
        for ( h = 0 ; h < nh ; h++ )
        {
            D_aux = n_lr * d_Whl_Bl[l] * Ho[h];
            Whl[h][l] = Whl[h][l] + D_aux + alpha*D_Whl[h][l];
            D_Whl[h][l] = D_aux;
        }
        /*Bl*/
        if ( usa_BIAS )
        {
            D_aux = n_lr * d_Whl_Bl[l];
            Bl[l] = Bl[l] + D_aux + alpha*D_Bl[l];
            D_Bl[l] = D_aux;
        }
    }

/*Atualizacao dos PESOS - Wfh ___ BIAS - Bh
*/
    for ( h = 0 ; h < nh ; h++ )
    {
        soma_aux = 0.0;

```

```

for ( l = 0 ; l < nl ; l++ )
    soma_aux += ( d_Whl_Bl[l] * (Whl[h][l] + usa_BIAS*Bl[l]) );

d_aux = soma_aux * d_NF_h(Hi[h]);

/*Wfh*/
for ( f = 0 ; f < nf ; f++ )
{
    D_aux = n_lr * d_aux * Fo[f];
    Wfh[f][h] = Wfh[f][h] + D_aux + alpha*D_Wfh[f][h];
    D_Wfh[f][h] = D_aux;
}
/*Bh*/
if ( usa_BIAS )
{
    D_aux = n_lr * d_aux;
    Bh[h] = Bh[h] + D_aux + alpha*D_Bh[h];
    D_Bh[h] = D_aux;
}
}

n_batch_atual = 0;

return;
}
/*****
*/

/***** TREINAMENTO DA REDE
*****/

int Treina(void)
/*Funcao que tem por finalidade treinar a rede MLP.*/
{
    int fim = 0;
    int atua;

    Ini_pesos_e_bias(); /*Inicia pesos e bias*/

    Input();

    strcpy (nomearqWERR, "errotre"); /* Salva erro durante o treino */
    if ( (fileWERR = fopen (nomearqWERR, "wt")) == NULL )
    {
        printf ("Erro no arquivo %s\n\n", nomearqWERR);
        exit(0);
    }

    while (!fim)
    {
        Carregar_I_T(); /*Atualiza I e T*/

        First_layer(); /*Atualiza Fi e Fo*/
        Hidden_layer(); /*Atualiza Hi e Ho*/
        Last_layer(); /*Atualiza Li e Lo*/

        Calculo_Erro(); /*Atualiza E = T -Lo, erro*/
        /*Atualizacao dos pesos*/
        atua = Atualiza_Ind_Lote(); /*Verifica se a atualiz deve ser feita*/
        if (atua)
            Atualiza_W_B();
    }
}

```

```

if ( n_IN_atual > narqIN ) /*E uma EPOCA*/
{
    erro_EP /= (float)narqIN;
    printf("EPOCA : %d \t\tERRO : %f \n", t_epc, erro_EP);

    fprintf (fileWERR, " %f\n", erro_EP); /* Salva erro durante
o treino */

    Atualiza_Param();
    fim = t_epc / MAXITERACAO;
    /*por erro*/
    if ( erro_EP < Eps )
        fim = 1;
}
Salva_W_B();
Fecha_I_T();

return (1);
}
/*****
*/

/***** TESTE DA
REDE*****/

int Teste(void)
/*Funcao que tem por finalidade testar a rede MLP.*/
{
    int cont;

    Carregar_W_B();/*Atualiza W e B e*/

    Input();
    SalvaOUT_Testes();
    for ( cont = 0 ; cont < narqIN ; cont++)
    {
        CarregaIN_Testes();/*Atualiza I*/

        First_layer(); /*Atualiza Fi e Fo*/

        Hidden_layer(); /*Atualiza Hi e Ho*/
        Last_layer(); /*Atualiza Li e Lo*/

        Saida();

        /* SalvaOUT_Testes();*/
    }
    /* Fecha();*/
    Fecha_I_OUT();

    return (1);
}

/*****
*/

/***** PROGRAMA PRINCIPAL
*****/

```

```

int main(int argc, char *argv[])
{
    int ok;

    if ( argc != 5 )
    {
        printf("Uso Correto : rede_mlp2 TR||TE input.txt||teste.txt arq_WB2
padrao.txt||output.txt \n\n");
        printf("      TR||TE : TREINAMENTO ou TESTE da rede.\n");
        printf("      input||teste : arquivo de ENTRADA ou TESTE da rede.\n");
        printf("      arq_WB2 : arquivo de PESOS e BIAS da rede.\n");
        printf(" padrao||output: arquivo de TARGET ou SAIDA da rede.\n");
        return (0);
    }

    strcpy (nomearqIN , argv[2]);
    strcpy (nomearqWB , argv[3]);
    strcpy (nomearqOUT, argv[4]);

    if (!(strcmp(argv[1], TREINAMENTO)))
        ok = Treina ();
    else
        if (!(strcmp(argv[1], TESTE)))
            ok = Teste ();
        else
            return (0);

    if (!ok)
    {
        printf("ERRO NA OPERACAO\n\n");
        return (0);
    }
    return(1);
}
/*****
*/

```

```

/*****
*/

```

```

*   Autor   : Helena Rossi
*           Software para prediçao de potencia usando
*           Radial Basis Function Network
*
#include <conio.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <process.h>

/***** Definicao da configuracao da rede *****/

#define Nf 3      /* Numero de neurons do first layer(number of inputs) */
#define Nh 22    /* Numero de neurons do hidden layer(basis functions, n
centros) */
#define Nl 1     /* Numero de neurons do last layer(number of outputs) */

#define sigma 0.1 /* Raio das basis functions do hidden layer */

#define a_lr 1.8 /* Learning rate do output layer ajustavel com o tempo */
#define b_lr 0.0004 /*001  /* learning rate = lr = alr * exp(-blr * t) *
exp(-clr*log(t)) */
#define c_lr 0.4   /**/

#define usa_MOMENTO 1 /* Determina uso do momento: 0=nao usa, 1=usa */
#define a_mm 0.3 /*5      /* Momento do output layer ajustavel com o tempo
*/
#define b_mm 0.0      /* Momento = mm = amm * exp(-bmm * t) * exp(-
cmm*log(t)) */
#define c_mm 0.3     /**/

#define a_nf 1.0      /* Neural function sigmoidal generica - facil de
derivar */
#define b_nf 0.0      /*          anf
*/
#define c_nf 1.0      /*5.0/* NF = ----- - bnf      */
#define d_nf 0.0      /*          1 + exp[( -cnf * i)+d]      */

#define F_escala 5.0  /* Fator de escala para o last layer */
#define usa_BIAS 0    /* Define uso do BIAS: 0=nao usa, 1=usa */
#define Erro_final 0.001 /* Define erro final desejado */
#define N_epocas 10000 /* Numero de epocas de treinamento maximo */

#define Nparam 6      /* Numero de parametros recebidos por main */
#define TREINAMENTO "TR"
#define TESTE "TE"
#define cria_pesos 0  /*Cria pesos aleatoriamente ou inicia com alguns

/***** Definicao das variaveis globais *****/

float Ch[Nh][Nf];    /* Matriz de Centros das basis functions */
/* Matriz de pesos entre first e hidden layer */
float Whl[Nh][Nl];  /* Matriz de pesos entre hidden e last layer */
/* Matriz Delta de pesos entre first e hidden layer */
float D_Whl[Nh][Nl]; /* Matriz Delta de pesos entre hidden e last layer */

float Bl[Nl];        /* Vetor de BIAS do last layer */
float D_Bl[Nl];      /* Vetor Delta de BIAS do last layer */

float I[Nf];         /* Vetor de entrada da rede */
float Fi[Nf];        /* Vetor de entrada do hidden layer */
float Fo[Nf];        /* Vetor de saida do first layer */

```

```

float Hi[Nh];      /* Vetor de entrada do hidden layer */
float Ho[Nh];      /* Vetor de saida do hidden layer */
float Li[Nl];      /* Vetor de entrada do last layer */
float Lo[Nl];      /* Vetor de saida do last layer */
float T[Nl];       /* Vetor Target da rede */
float E[Nl];       /* Vetor Erro da rede */

int t_epoca = 1;    /* Indicador de numero de epocas efetuadas */
float n_lr;        /* Learning rate */
float n_mm;        /* Coeficiente de momento */

/*float erro_EP = 0.0; /* Erro de toda época*/

char modo[2];      /* Modo de operacao: TR || TE */
char nomearqIN[20]; /* Nome do arquivo de entrada */
FILE *fileIN;
char nomearqWERR[20]; /* Nome do arquivo de erro */
FILE *fileWERR;
char nomearqWB[20]; /* Nome do arquivo de pesos e bias */
FILE *fileWB;
char nomearqC[20]; /* Nome do arquivo de centros das basis functions */
FILE *fileC;
char nomearqOUT[20]; /* Nome do arquivo de saida ou target */
FILE *fileOUT;
int narqIN;        /* Numero arquivos do arquivo de entrada */
int n_IN_atual = 1; /* Numero do arquivo de entrada atual */

/****** Declaracao Prototipo das funcoes utilizadas *****/

void Inicializa_W_B(void);
int Treina(void);
int Teste(void);
void Carrega_C(void);
void Carrega_I_T(void);
void First_layer(void);
void Hidden_layer(void);
void Last_layer(void);
float Calcula_erro(void);
void Atualiza_W_B(void);
void Atualiza_lr_mm(void);
void Salva_W_B(void);
void Fecha_I_T(void);
void Carrega_W_B(void);
void Le_n_Input(void);
void Carrega_I_Testes(void);
void Salva_Out_Testes(void);
void Fecha_I_Out(void);
void calc_lr(void);
void calc_mm(void);
    Erro(int codigo, char nome);
float NF_first(float in);
float NF_linear(float a, float in);
float NF_hidden(float in);
float NF_gaussiana(float in);
float NF_last(float in);
float NF_sigmoide(float in);

/****** Programa Principal *****/

int main(int argc, char *argv[])
{
    int ok;

```

```

    if (argc != Nparam)
    {
        printf("Uso Correto : RBF2 TR||TE  INPUT.TXT||TESTE.TXT
arq_WB2 arq_c2 padrao.txt||output.txt \n\n");
        printf("      TR||TE                : TREINAMENTO ou TEste da
rede\n");
        printf("      INPUT.TXT || TESTE.TXT : arquivo de ENTRADA ou
TESTE da rede\n");
        printf("      arq_WB2                : arquivo de PESOS e BIAS
da rede\n");
        printf("      arq_C2                : arquivo de CENTROS das
basis functions\n");
        printf("      PADRAO.TXT||OUTPUT.TXT : arquivo de TARGET ou
SAIDA da rede\n\n");
        return (0);
    }
    strcpy (modo, argv[1]);
    strcpy (nomearqIN , argv[2]);
    strcpy (nomearqWB , argv[3]);
    strcpy (nomearqC , argv[4]);
    strcpy (nomearqOUT , argv[5]);

    if (!(strcmp(argv[1], TREINAMENTO))) /* Verifica se vai treinar */
        ok = Treina();
    else
        if (!(strcmp(argv[1], TESTE))) /* Verifica se vai testar */
            ok = Teste();
        else
            return (0);

    if (!ok)
    {
        printf("ERRO NO USO DO PROGRAMA\n\n");
        return (0);
    }
    return (1);
}

```

/\*----- TREINAMENTO DA REDE -----\*/

```

int Treina()
{
    int cont = 0; /* Contador do numero de interacoes em uma epoca */
    int fim = 0; /* Flag de final do treinamento: por erro ou tempo */
    int atua;
    float error, erro_EP, soma_t = 0.0;

    Inicializa_W_B(); /*Inicializa os valores dos pesos e bias */
    Carrega_C(); /*Carrega os vetores dos centros das basis functions */

    Le_n_Input(); /* Verifica qtos vetores de treino - retorna por
narqIN */

    strcpy (nomearqWERR, "errotre"); /* Salva erro durante o treino */
    if ( (fileWERR = fopen (nomearqWERR, "wt")) == NULL )
    {
        printf ("Erro no arquivo %s\n\n", nomearqWERR);
        exit(0);
    }
    while (!fim)
    {
        Carrega_I_T(); /* Carrega os vetores I e T usados no step
atual */
    }
}

```

```

First_layer(); /* Atualiza entradas e saidas do first layer
*/
Hidden_layer(); /* Atualiza entradas e saidas do hidden layer
*/
Last_layer(); /* Atualiza entradas e saidas do output layer
*/

error = Calcula_erro(); /* Calcula o erro  $E = T - Lo$  */

/*          fprintf (fileWERR, "%f\n", error); /* Salva erro durante o
treino */

if ( !fim ) /* Se nao fim, atualiza os pesos e bias */
{
    Atualiza_W_B(); /* Atualiza pesos e bias */
    cont++; /* Auxiliar para o calculo da epoca */

/*          printf("t_epoca=%d      inter.=%d      erro=%f
Lrate=%f      Mom=%f\n", t_epoca, cont, error, n_lr, n_mm); */

    soma_t += error; /* Soma parcial para o
calculo do erro medio de uma epoca*/

    if ( cont == narqIN )
    {
        erro_EP=soma_t/(float) narqIN;
        cont = 0;

        fprintf (fileWERR, " %f\n", erro_EP); /*
Salva erro durante o treino */

        printf("t_epoca=%d
inter.=%d      erro_EP=%f Lrate=%f      Mom=%f\n", t_epoca, cont, erro_EP,
n_lr, n_mm);

        t_epoca++;
        Atualiza_lr_mm(); /* Ao fim de uma epoca
atualiza lr e mm */
        fim = t_epoca / (N_epocas+1); /* Fim do
treinamento por epocas */
        if ( erro_EP < Erro_final ) /* Fim do
treinamento por erro alcancado */
            fim = 1; /*helena habilitou*/

        soma_t = 0.0; /* Zera o soma p/calcular erro
da proxima epoca */

    }
}
fclose (fileWERR);
Salva_W_B();

```

```

    Fecha_I_T();
    return (1);
}

/***** TESTE DA REDE *****/

int Teste()
{
    int cont_tst;

    Carrega_W_B(); /* Carrega os pesos e bias */
    Carrega_C(); /*Carrega os vetores dos centros das basis functions */

    Le_n_Input(); /* Verifica qtos vetores de teste - retorna por
narqIN */

    if ( (fileOUT = fopen (nomearqOUT, "wt")) == NULL) /* Abre o arquivo
de saida */
    {
        printf ("Erro no arquivo %s\n\n", nomearqOUT);
        exit(0);
    }

    for ( cont_tst = 0; cont_tst < narqIN; cont_tst++ )
    {
        Carrega_I_Teste(); /* Carrega o vetor de teste */

        First_layer(); /* Atualiza entradas e saidas do first layer
*/
        Hidden_layer(); /* Atualiza entradas e saidas do hidden layer
*/
        Last_layer(); /* Atualiza entradas e saidas do output layer
*/

        Salva_Out_Teste(); /* Salva o resultado do teste */
    }
    Fecha_I_Out();
    return(1);
}

/***** CALCULOS E ATUALIZACOES *****/

/*****INICIALIZACAO DOS PESOS E BIAS *****/

void Inicializa_W_B(void)
/* Funcao que efetua a inicializacao dos pesos e bias
So e utilizada no inicio do treinamento da rede
Esta versao so inicializa os pesos da hidden para a last layer
O Bias so e considerado para o last layer */

{
    int h, l;
    float aux;
    time_t t;

    srand((unsigned) time(&t));

    /* Aqui pode ser encaixada a inicializacao dos pesos da first p/
hidden layer
* E inicializada tambem a matriz delta dos pesos (com 0.0) */
    if (cria_pesos)
    {
        for ( h = 0; h < Nh; h++ )

```

```

        for ( l = 0; l < Nl; l++ )
        {
            aux = (( ((float)(rand() % 20001)) -
10000.0)/10000.0);
            Whl[h][l] = aux; /* Armazena o peso inicial */
            D_Whl[h][l] = 0.0; /* Inicializa o elemento delta-
peso corresp. */
        }
        if ( usa_BIAS )
        {
            /* Aqui entra o bias do input layer */
            /* Aqui entra o bias do hidden layer */

            for ( l = 0; l < Nl; l++ )
            {
                aux = (( ((float)(rand() % 20001)) -
10000.0)/10000.0);
                Bl[l] = aux;
                D_Bl[l] = 0.0; /* tambem e inicializado o vetor delta
do BIAS */
            }
        }
        else
            /*Usa os pesos ja criados*/

            if ( (fileWB = fopen(nomearqWB, "rt")) == NULL )
            {
                printf("Erro no arquivo %s\n\n", nomearqWB);
                exit(0);
            }

            /* Se existir pesos entre o first e hidden layer, colocar aqui */
            for ( h = 0; h < Nh; h++)
                for ( l = 0; l < Nl; l++)
                {
                    fscanf (fileWB, "%f", &aux);
                    Whl[h][l] = aux;
                    D_Whl[h][l] = 0.0;
                }

            if (usa_BIAS)
            {
                /* Se existir bias para o first layer, colocar aqui */
                /* Se existir bias para o last layer, colocar aqui */

                for ( l = 0; l < Nl; l++)
                {
                    fscanf (fileWB, "%f", &aux);
                    Bl[l] = aux;
                    D_Bl[l] = 0.0;
                }
            }
            fclose (fileWB);

            }

            calc_lr(); /* Calcula o learning rate inicial */
            calc_mm(); /* Calcula o momento inicialmente utilizado */

```

```

        return;
    }

/***** CALCULO DO LEARNING RATE P/ OUTPUT LAYER *****/

void calc_lr(void)
/* Funcao que calcula a learning rate (lr) */
{
    n_lr = a_lr * exp(-b_lr*t_epoca) * exp(-c_lr*log(t_epoca));
    return;
}

/***** CALCULO DO MOMENTO P/ OUTPUT LAYER *****/

void calc_mm(void)
/* Funcao que calcula o fator de momento (mm) */
{
    n_mm = a_mm * exp(-b_mm*t_epoca) * exp(-c_mm*log(t_epoca));
    return;
}

/***** Rotina para efetuar o calculo do first layer da rede */

void First_layer(void)
{
    int f;
    for ( f = 0; f < Nf; f++ )
    {
        Fi[f] = I[f]; /* Aqui pode ser incluido o BIAS
Fi[f]=I[f]+usa_BIAS*Bf[f] */
        Fo[f] = NF_first( Fi[f]);
    }
    return;
}

/***** Rotina para efetuar o calculo da hidden layer da rede */

void Hidden_layer(void)
{
    int f,h;
    float d, dist;
    for ( h = 0; h < Nh; h++ )
    {
        d = 0.0;
        for ( f = 0; f < Nf; f++ )
            d += (Fo[f] - Ch[h][f]) * (Fo[f] - Ch[h][f]);
        dist = sqrt(d); /* Calcula a distancia Euclideana */
        Hi[h] = dist; /* Aqui pode ser incluido o BIAS
Hi[h]=dist+usa_BIAS*Bf[f] */
        Ho[h] = NF_hidden( Hi[h]);

        //printf("Hi=%f Ho=%f\n", Hi[h], Ho[h]); /* usado para
verificar range de Hi[] e Ho[] */
    }

    return;
}

/***** Rotina para efetuar o calculo da last layer da rede */

void Last_layer(void)

```

```

int h,l;
float soma;
for ( l = 0; l < Nl; l++ )
{
    soma = 0.0;
    for ( h = 0; h < Nh; h++ )
        soma += ( Whl[h][l] * Ho[h] );
    Li[l] = (F_escala * soma) + (usa_BIAS * Bl[l]);

    //printf("Li=%f\n", Li[l]);/**/

    Lo[l] = NF_last( Li[l] );

    //printf("Lo=%f\n", Lo[l]);/**/
    //printf("Li=%f Lo=%f\n", Li[l], Lo[l]); /* usado para
verificar range de Li[] e L[o] */
}
return;
}

/*****ROTINAS PARA CALCULO DAS NEURAL FUNCTIONS *****/

/***** Neural function do first layer */

float NF_first(float in)
{
    return (NF_linear(1.0, in));
}

/***** Neural function do hidden layer */

float NF_hidden(float in)
{
    return (NF_gaussiana(in));
}

/***** Neural function do last layer */

float NF_last(float in)
{
    return (NF_sigmoide(in));

    /* return (NF_linear(0.5, in));*/
}

/***** Neural function linear */

float NF_linear(float a, float in)
{
    return (a * in);
}

/***** Neural function Gaussiana */

float NF_gaussiana(float in)
{
    return ( exp(-(in*in) / (sigma*sigma)));
}

/***** Neural function sigmoide */
/* esta implementacao e mais facil de derivar */

float NF_sigmoide(float in)

```

```

|
double aux;
aux = 1.0 + exp( -(c_nf * in)+d_nf );
return ( (a_nf / aux) - b_nf);
}

/***** Rotina para o calculo da funcao de erro */

float Calcula_erro(void)
{
    int l;
    float soma, erro;

    soma = 0.0;
    for ( l = 0; l < Nl; l++ )
    {
        E[l] = T[l] - Lo[l]; /* Erro pela regra do Adaline */
        soma += E[l] * E[l];
    }
    erro = (sqrt(soma))/Nl;
    return(erro);
}

/***** Rotina que atualiza os pesos(W) e bias(B) da rede */

void Atualiza_W_B(void)
{
    int h, l;
    float D_Whl_aux;

    for ( l = 0; l < Nl; l++ )
    {
        for ( h = 0; h < Nh; h++ )
        {
            D_Whl_aux = ( Ho[h] * E[l] * n_lr ) + ( D_Whl[l][h] *
n_mm * usa_MOMENTO );
            D_Whl[l][h] = D_Whl_aux;
            Whl[l][h] += D_Whl[l][h];
        }
    }
    return;
}

/***** Rotina que atualiza a learning rate e o momento da rede */

void Atualiza_lr_mm(void)
{
    calc_lr();
    calc_mm();
}

/***** ROTINAS DE ENTRADA E SAIDA DE DADOS *****/

/***** Rotina que carrega os vetores de Centro das basis functions */

void Carrega_C(void)
/* Funcao que carrega a matriz que contem os vetores de Centro das basis
functions com os valores dos arquivos de Centro */

{
    int f;
    int narqC; /* Numero de arquivos de Centro a serem lidos */
    int neleV; /* Numero de elementos do Vetor Centro */
    int n = 0; /* Numero do centro lido */
}

```

```

FILE *fauxc; /* Arquivo auxiliar para leitura do vetor Centro */
char auxc[15];
float aux;

if ( (fileC = fopen(nomearqC, "rt")) == NULL) /* Verifica arq. de
centros */
{
    printf ("ERRO1 no arquivo %s\n\n", nomearqC);
    exit(0);
}
fscanf(fileC, "%d", &narqC);
if ( Nh > narqC) /* Verifica se a quantidade de Centros esta correta
*/
{
    printf ("ERRO2 no arquivo %s - numero de elem. no vetor de
Centros errado\n\n", nomearqC);
    exit(0);
}
for ( n = 0; n < Nh; n++)
{
    fscanf (fileC, "%s", auxc);
    if ( (fauxc = fopen(auxc, "rt")) == NULL) /* Verifica arq. de
Centros */
    {
        printf ("ERRO3 no arquivo %s\n\n", auxc);
        exit(0);
    }
    fscanf(fauxc, "%d", &neleV);
    if ( Nf > neleV) /* Verifica se o numero de elem. do vetor
esta correto */
    {
        printf ("ERRO4 no arquivo %s - numero de Centros
errado\n\n", auxc);
        exit(0);
    }
    for ( f = 0; f < Nf; f++ ) /* Carrega vetor de Centro do arq.
correspondente */
    {
        fscanf (fauxc, "%f", &aux);
        Ch[n][f] = aux;
    }
    fclose(fauxc);
}
fclose(fileC);
}

/***** Rotina para carga dos vetores de entrada e targets para o treinamento
*/

void Carrega_I_T(void)
/* Funcao que tem por finalidade carregar os vetores I e T. Nos arquivos
nomearqIN e nomearqOUT, estao os nomes dos arquivos a serem lidos.
Tambem faz a verificacao se precisa voltar ou nao para o primeiro
arquivo */

{

    int cont, auxi;
    float aux;

    if ( n_IN_atual > narqIN ) /* Devemos voltar ao primeiro */
    {
        fclose(fileIN);
        fclose(fileOUT);
    }
}

```

```

if ( ( fileIN = fopen(nomearqIN, "rt") ) == NULL )
{
    printf ("Erro5 no arquivo %s\n\n", nomearqIN);
    exit(0);
}
fscanf(fileIN, "%d", &narqIN);

if ( ( fileOUT = fopen(nomearqOUT, "rt") ) == NULL )
{
    printf ("Erro6 no arquivo %s\n\n", nomearqOUT);
    exit(0);
}
fscanf(fileOUT, "%d", &narqIN);
n_IN_atual = 1;
printf("\n");
/* getch(); para apos cada iteracao esperando teclar algo */
/* Continua lendo de onde havia parado */
}

```

```

n_IN_atual++;

```

```

for ( cont = 0; cont < Nf; cont++)
{
    fscanf(fileIN, "%f", &aux);
    I[cont] = aux;
}
for ( cont = 0; cont < Nl; cont++)
{
    fscanf(fileOUT, "%f", &aux);
    T[cont] = aux;
}

```

```

return;

```

```

}

```

```

/***** Rotina que fecha os arquivos de entrada e targets apos o treinamento */

```

```

void Fecha_I_T(void)
{
    fclose (fileIN);
    fclose (fileOUT);
    return;
}

```

```

/***** Rotina que salva os pesos(W) e bias(B) apos o treinamento */

```

```

void Salva_W_B(void)
{
    int h, l;
    float aux;

    printf("Salvando pesos e bias...\n");
    if ( (fileWB = fopen (nomearqWB, "wt") ) == NULL )
    {
        printf ("Erro no arquivo %s\n\n", nomearqWB);
        exit(0);
    }
    for ( h = 0; h < Nh; h++ )

```

```

        for ( l = 0; l < Nl; l++ )
        {
            aux = Whl[h][l];
            fprintf (fileWB, "%f\n", aux);
        }
    if ( usa_BIAS )
    {
        for ( l = 0; l < Nl; l++ )
        {
            aux = Bl[l];
            fprintf (fileWB, "%f\n", aux);
        }
    }
    fclose (fileWB);
    return;
}

/***** Rotina que carrega os pesos(W) e bias(B) para o teste da rede */
void Carrega_W_B(void)
{
    int h, l;
    float aux;

    if ( (fileWB = fopen(nomearqWB, "rt")) == NULL )
    {
        printf("Erro no arquivo %s\n\n", nomearqWB);
        exit(0);
    }

    /* Se existir pesos entre o first e hidden layer, colocar aqui */
    for ( h = 0; h < Nh; h++)
        for ( l = 0; l < Nl; l++)
        {
            fscanf (fileWB, "%f", &aux);
            Whl[h][l] = aux;
        }
    if (usa_BIAS)
    {
        /* Se existir bias para o first layer, colocar aqui */
        /* Se existir bias para o last layer, colocar aqui */

        for ( l = 0; l < Nl; l++)
        {
            fscanf (fileWB, "%f", &aux);
            Bl[l] = aux;
        }
    }
    fclose (fileWB);
    return;
}

/***** Rotina que determina o numero de vetores de entrada que sera utilizado
*/
void Le_n_Input(void)
{
    int aux;
    if ((fileIN = fopen(nomearqIN, "rt")) == NULL)
    {
        printf ("ERRO no arquivo %s\n\n", nomearqIN);
        exit(0);
    }
}

```

```

    }
    fscanf(fileIN, "%d", &narqIN);
    aux = narqIN;

    if (!(strcmp(modo, TREINAMENTO))) /* So executa este trecho se vai
treinar */
    {
        if ((fileOUT = fopen(nomearqOUT, "rt")) == NULL)
        {
            printf ("ERRO no arquivo %s\n\n", nomearqOUT);
            exit(0);
        }
        fscanf(fileOUT, "%d", &narqIN);
        if (aux - narqIN)
        {
            printf ("ERRO n.vet.IN diferente n.vet.OUT\n\n");
            exit(0);
        }
    }
    return;
}

/***** Rotina que carrega os vetores de entrada usados no teste da rede */
void Carrega_I_Teste(void)
{
    int cont, auxi;
    float aux;

    /* Nao e necessario verificar se ainda ha arquivos para ler, pois
isto
    ja e feito da rotina Teste() */
    for ( cont = 0; cont < Nf; cont++ )
    {
        fscanf (fileIN, "%f", &aux);
        I[cont] = aux;
    }

    return;
}

/***** Rotina que salva a saida da rede apos o teste */
void Salva_Out_Teste(void)
{
    int cont;
    float aux;

    /* fprintf (fileOUT, "%d\n", N1);*/
    for ( cont = 0; cont < N1; cont++ )
    {
        aux = Lo[cont];
        fprintf (fileOUT, "%f\n", aux);
    }

    return;
}

/***** Rotina que fecha os arquivos de entrada e saida apos o teste da rede
*/

```

```
void Fecha_I_Out(void)
{
    fclose (fileIN);
    fclose (fileOUT);
    return;
}
```

```
/***** Rotina para tratamento de erros *****/
```

```
Erro(int codigo, char nome)
{
}
```

**APÊNDICE I I**

**Tabela AII.1 – Centróides (normalizados) para a Rede do tipo RBF com 2 neurônios na camada oculta**

sigma =2.0	T3	T4	N16
Neurônio 1	0.1386	0.3813	0.2298
Neurônio 2	0.2941	0.8523	0.7468

**Tabela AII.6 – Centróides (normalizados) para a Rede do tipo RBF com 7 neurônios na camada oculta**

sigma=0.5	T3	T4	N16
Neurônio 1	0.1359	0.2754	0.0568
Neurônio 2	0.3731	0.9251	0.7223
Neurônio 3	0.0985	0.5685	0.6192
Neurônio 4	0.2714	0.9401	0.9311
Neurônio 5	0.3711	0.7637	0.4750
Neurônio 6	0.0259	0.3197	0.3201
Neurônio 7	0.3112	0.5581	0.2263

**Tabela AII.2 – Centróides (normalizados) para a Rede do tipo RBF com 3 neurônios na camada oculta**

sigma =1.8	T3	T4	N16
Neurônio 1	0.2978	0.908	0.8318
Neurônio 2	0.1031	0.3113	0.1737
Neurônio 3	0.2800	0.6522	0.4444

**Tabela AII.7 – Centróides (normalizados) para a Rede do tipo RBF com 8 neurônios na camada oculta**

sigma=0.5	T3	T4	N16
Neurônio 1	0.3073	0.5443	0.2099
Neurônio 2	0.1501	0.2808	0.0419
Neurônio 3	0.3705	0.7391	0.4413
Neurônio 4	0.2900	0.9610	0.9337
Neurônio 5	0.0623	0.4845	0.5334
Neurônio 6	0.0299	0.2860	0.2607
Neurônio 7	0.1598	0.7149	0.7610
Neurônio 8	0.3807	0.9154	0.6889

**Tabela AII.3 – Centróides (normalizados) para a Rede do tipo RBF com 4 neurônios na camada oculta**

sigma=1.4	T3	T4	N16
Neurônio 1	0.3540	0.7123	0.4184
Neurônio 2	0.1164	0.2943	0.1226
Neurônio 3	0.3046	0.9289	0.8549
Neurônio 4	0.0778	0.5007	0.5356

**Tabela AII.8 – Centróides (normalizados) para a Rede do tipo RBF com 9 neurônios na camada oculta**

sigma=0.5	T3	T4	N16
Neurônio 1	0.3737	0.7558	0.4603
Neurônio 2	0.0840	0.5385	0.5915
Neurônio 3	0.1676	0.7404	0.7860
Neurônio 4	0.0420	0.2562	0.1881
Neurônio 5	0.2902	0.9638	0.9377
Neurônio 6	0.1716	0.2973	0.0320
Neurônio 7	0.3159	0.5653	0.2299
Neurônio 8	0.3813	0.9222	0.7001
Neurônio 9	0.0273	0.3699	0.3989

**Tabela AII.4 – Centróides (normalizados) para a Rede do tipo RBF com 5 neurônios na camada oculta**

sigma= 1.2	T3	T4	N16
Neurônio 1	0.0708	0.4960	0.5398
Neurônio 2	0.2883	0.9372	0.8987
Neurônio 3	0.3178	0.5853	0.2634
Neurônio 4	0.0952	0.2733	0.1241
Neurônio 5	0.3679	0.8458	0.6000

**Tabela AII.9 – Centróides (normalizados) para a Rede do tipo RBF com 10 neurônios na camada oculta**

sigma= 0.2	T3	T4	N16
Neurônio 1	0.2909	0.4993	0.1645
Neurônio 2	0.0822	0.5357	0.5896
Neurônio 3	0.3767	0.9572	0.7780
Neurônio 4	0.0268	0.3703	0.4008
Neurônio 5	0.1594	0.2825	0.0284
Neurônio 6	0.2730	0.9556	0.9526
Neurônio 7	0.0394	0.2571	0.1951
Neurônio 8	0.3813	0.8356	0.5601
Neurônio 9	0.1642	0.7322	0.7795
Neurônio 10	0.3530	0.6676	0.3503

**Tabela AII.5 – Centróides (normalizados) para a Rede do tipo RBF com 6 neurônios na camada oculta**

sigma= 0.5	T3	T4	N16
Neurônio 1	0.3758	0.8687	0.6211
Neurônio 2	0.2909	0.9471	0.9103
Neurônio 3	0.1079	0.5855	0.6329
Neurônio 4	0.1466	0.2878	0.0591
Neurônio 5	0.3338	0.6207	0.2974
Neurônio 6	0.0287	0.3239	0.3223

**APÊNDICE II**

**Tabela AII.10 – Centróides (normalizados) para a Rede do tipo RBF com 11 neurônios na camada oculta**

<b>sigma =0.2</b>	<b>T3</b>	<b>T4</b>	<b>N16</b>
Neurônio 1	0.1658	0.7387	0.7863
Neurônio 2	0.3828	0.8676	0.6050
Neurônio 3	0.0920	0.5584	0.6123
Neurônio 4	0.0204	0.3024	0.2998
Neurônio 5	0.0532	0.2405	0.1329
Neurônio 6	0.3075	0.5403	0.2031
Neurônio 7	0.1770	0.3005	0.0261
Neurônio 8	0.2676	0.9531	0.9576
Neurônio 9	0.3721	0.9689	0.8087
Neurônio 10	0.0401	0.4231	0.4641
Neurônio 11	0.3656	0.7056	0.3992

**Tabela AII.11 – Centróides (normalizados) para a Rede do tipo RBF com 12 neurônios na camada oculta**

<b>sigma = 0.2</b>	<b>T3</b>	<b>T4</b>	<b>N16</b>
Neurônio 1	0.1646	0.2849	0.0220
Neurônio 2	0.1652	0.7385	0.7869
Neurônio 3	0.2835	0.4811	0.1469
Neurônio 4	0.0419	0.4316	0.4749
Neurônio 5	0.3672	0.9775	0.8330
Neurônio 6	0.0166	0.3116	0.3177
Neurônio 7	0.3415	0.6368	0.3090
Neurônio 8	0.0928	0.5611	0.6155
Neurônio 9	0.0468	0.2429	0.1488
Neurônio 10	0.3815	0.8947	0.6466
Neurônio 11	0.2639	0.9508	0.9604
Neurônio 12	0.3762	0.7722	0.4792

**Tabela AII.12 – Centróides (normalizados) para a Rede do tipo RBF com 15 neurônios na camada oculta**

<b>sigma = 0.2</b>	<b>T3</b>	<b>T4</b>	<b>N16</b>
Neurônio 1	0.3842	0.8746	0.6141
Neurônio 2	0.3844	0.9476	0.7481
Neurônio 3	0.3499	0.6582	0.3358
Neurônio 4	0.1351	0.6385	0.6869
Neurônio 5	0.0510	0.2371	0.1288
Neurônio 6	0.3765	0.7704	0.4761
Neurônio 7	0.1730	0.7675	0.8151
Neurônio 8	0.0740	0.5170	0.5720
Neurônio 9	0.3112	0.5482	0.2089
Neurônio 10	0.2507	0.4111	0.0877
Neurônio 11	0.1578	0.2752	0.0171
Neurônio 12	0.0255	0.2874	0.2738
Neurônio 13	0.0316	0.3899	0.4261
Neurônio 14	0.3568	0.9916	0.8777
Neurônio 15	0.2606	0.9510	0.9674

**Tabela AII.13 – Centróides (normalizados) para a Rede do tipo RBF com 20 neurônios na camada oculta**

<b>sigma = 0.2</b>	<b>T3</b>	<b>T4</b>	<b>N16</b>
Neurônio 1	0.3181	0.5696	0.2309
Neurônio 2	0.1627	0.7330	0.7837
Neurônio 3	0.2839	0.4797	0.1463
Neurônio 4	0.0622	0.2316	0.1043
Neurônio 5	0.0104	0.3189	0.3340
Neurônio 6	0.2645	0.9588	0.9736
Neurônio 7	0.0322	0.3869	0.4223
Neurônio 8	0.3804	0.8922	0.6406
Neurônio 9	0.0506	0.4629	0.5121
Neurônio 10	0.2349	0.3753	0.0652
Neurônio 11	0.2091	0.8414	0.8807
Neurônio 12	0.3754	0.7312	0.4305
Neurônio 13	0.0376	0.2628	0.2217
Neurônio 14	0.3584	0.9936	0.8791
Neurônio 15	0.3782	0.8120	0.5270
Neurônio 16	0.3482	0.6545	0.3285
Neurônio 17	0.1569	0.2728	0.0133
Neurônio 18	0.1337	0.6327	0.6816
Neurônio 19	0.3841	0.9519	0.7573
Neurônio 20	0.0821	0.5402	0.5979

**Tabela AII.14 – Centróides (normalizados) para a Rede do tipo RBF com 22 neurônios na camada oculta**

<b>sigma = 0.1</b>	<b>T3</b>	<b>T4</b>	<b>N16</b>
Neurônio 1	0.2334	0.3724	0.0632
Neurônio 2	0.3111	0.5478	0.2048
Neurônio 3	0.3900	0.8565	0.5842
Neurônio 4	0.0818	0.5393	0.5970
Neurônio 5	0.0498	0.4605	0.5096
Neurônio 6	0.2088	0.8409	0.8804
Neurônio 7	0.3808	0.9570	0.7718
Neurônio 8	0.1568	0.2726	0.0131
Neurônio 9	0.3782	0.9082	0.6639
Neurônio 10	0.0376	0.2621	0.2200
Neurônio 11	0.3516	0.6624	0.3404
Neurônio 12	0.1626	0.7322	0.7830
Neurônio 13	0.3755	0.7263	0.4250
Neurônio 14	0.0626	0.2314	0.1036
Neurônio 15	0.0101	0.3176	0.3316
Neurônio 16	0.3568	0.9950	0.8848
Neurônio 17	0.2808	0.4750	0.1409
Neurônio 18	0.3769	0.7904	0.5011
Neurônio 19	0.2637	0.9583	0.9743
Neurônio 20	0.0319	0.3854	0.4207
Neurônio 21	0.3289	0.6021	0.2685
Neurônio 22	0.1329	0.6311	0.6801

**APÊNDICE II**

**Tabela AII.15 – Centróides (normalizados)  
para a Rede do tipo RBF com 37 neurônios na  
camada oculta**

<b>sigma =0.1</b>	<b>T3</b>	<b>T4</b>	<b>N16</b>
Neurônio 1	0.1659	0.7729	0.8234
Neurônio 2	0.2557	0.9547	0.9841
Neurônio 3	0.3770	0.7814	0.4870
Neurônio 4	0.3445	0.6482	0.3167
Neurônio 5	0.2005	0.3317	0.0307
Neurônio 6	0.0226	0.2890	0.2766
Neurônio 7	0.3280	0.5999	0.2665
Neurônio 8	0.0820	0.5432	0.6020
Neurônio 9	0.2980	0.9861	0.9521
Neurônio 10	0.3794	0.8828	0.6273
Neurônio 11	0.3595	0.9996	0.8976
Neurônio 12	0.0415	0.2626	0.2251
Neurônio 13	0.3929	0.9434	0.7284
Neurônio 14	0.0568	0.2298	0.1106
Neurônio 15	0.3770	0.9104	0.6651
Neurônio 16	0.3110	0.5398	0.1907
Neurônio 17	0.3932	0.8542	0.5802
Neurônio 18	0.0016	0.3129	0.3256
Neurônio 19	0.2460	0.3957	0.0799
Neurônio 20	0.0876	0.2316	0.0705
Neurônio 21	0.0165	0.3407	0.3713
Neurônio 22	0.3607	0.6817	0.3747
Neurônio 23	0.3768	0.9620	0.7870
Neurônio 24	0.0330	0.2460	0.1669
Neurônio 25	0.2829	0.4788	0.1456
Neurônio 26	0.3770	0.8292	0.5422
Neurônio 27	0.1587	0.2736	0.0072
Neurônio 28	0.0328	0.3867	0.4223
Neurônio 29	0.1473	0.6632	0.7116
Neurônio 30	0.1635	0.7271	0.7772
Neurônio 31	0.3655	0.9836	0.8455
Neurônio 32	0.0442	0.4390	0.4879
Neurônio 33	0.1219	0.6137	0.6599
Neurônio 34	0.2116	0.8458	0.8872
Neurônio 35	0.0621	0.4889	0.5434
Neurônio 36	0.3769	0.7300	0.4292
Neurônio 37	0.3116	0.5575	0.2211