



A UTILIZAÇÃO DE REDES NEURAIS PARA OBTENÇÃO DA CONCENTRAÇÃO
DE BORO CRÍTICA DE UM REATOR PWR

Cicero Rogério da Silva Barros

Dissertação de Mestrado apresentada ao programa de Pós-graduação em Engenharia Nuclear, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Nuclear.

Orientadores: Roberto Schirru

Fernando Carvalho da Silva

Rio de Janeiro

Maio de 2011

A UTILIZAÇÃO DE REDES NEURAIIS PARA OBTENÇÃO DA CONCENTRAÇÃO DE
BORO CRÍTICA DE UM REATOR PWR

Cicero Rogério da Silva Barros

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA
(COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE
EM CIÊNCIAS EM ENGENHARIA NUCLEAR.

Examinada por:

Prof. Roberto Schirru, D.Sc.

Prof. Fernando Carvalho da Silva, D.Sc.

Prof. Jose Antonio Carlos Canedo Medeiros, D.Sc.

Prof. Cláudio Marcio Nascimento Abreu Pereira, D.Sc.

RIO DE JANEIRO, RJ-BRASIL

MAIO DE 2011

Barros, Cicero Rogério da Silva

A Utilização de Redes Neurais para Obtenção da Concentração de Boro Crítica de um Reator PWR / Cicero Rogério da Silva Barros. — Rio de Janeiro: UFRJ/COPPE, 2011.

XI, 63p.: il.; 29,7 cm.

Orientadores: Roberto Schirru

Fernando Carvalho da Silva

Dissertação (mestrado) – UFRJ / COPPE / Programa de Engenharia Nuclear, 2011

Referências Bibliográficas: p. 52-53

1. Redes Neurais. 2. Método de Expansão Nodal. 3. Criticalidade. I. Schirru, Roberto *et. al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Nuclear. III. Título.

Dedicatória

Dedico este trabalho ao meu pai e a minha madrasta.

Em memória de minha mãe e de minha madrinha

Agradecimentos

Agradeço a meu pai e a minha madrasta pelo apoio e a pela confiança em mim e por todas as condições que me deram para que eu vivesse este momento;

Agradeço aos meus professores do Programa de Engenharia Nuclear da COPPE da Universidade Federal do Rio de Janeiro, por me dar a base teórica necessária para a realização deste trabalho;

Agradeço especialmente aos meus professores orientadores Roberto Schirru e Fernando Carvalho da Silva pela paciência com que muito me ajudaram na realização deste trabalho;

Agradeço a todos os funcionários da secretaria do PEN/COPPE/UFRJ, por serem solícitos e atenciosos nos momentos de resolver tramites administrativos referentes ao curso;

Agradeço a todos os funcionários do Laboratório de Monitoração pelo apoio e paciência;

Agradeço ao amigo Alan pelos dados para construção do gráfico da concentração crítica de boro em função da queima;

Agradeço ao Professor Canedo pelo ensino do MATLAB e as explicações da utilização de redes neurais em MATLAB;

Agradeço aos Diretores, Professores e Colegas do Colégio Estadual Raul Vidal pelo apoio nos momentos difíceis da minha vida;

Agradeço pelo apoio dos meus amigos Átila, Paulinho e Luizinho nos momentos difíceis de minha vida.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc)

A UTILIZAÇÃO DE REDES NEURAIIS PARA OBTENÇÃO DA CONCENTRAÇÃO DE BORO CRÍTICA DE UM REATOR PWR

Cicero Rogério da Silva Barros

Maior/2011

Orientadores: Roberto Schirru

Fernando Carvalho da Silva

Programa: Engenharia Nuclear

Para a manutenção de um reator nuclear em estado crítico é necessário encontrar uma combinação entre a composição e a geometria do núcleo do reator, isto pode ser alcançado inserindo materiais absorvedores de nêutrons, como ácido bórico. A concentração crítica do ácido bórico é determinada atualmente por um processo iterativo, por exemplo, através do código NEM que faz uso do método nodal de malha grossa NEM, do inglês Nodal Expansion Method (Finnenanm et al., 1977). Nesta dissertação é feito um estudo da viabilidade do uso de uma rede neural artificial para obtenção da concentração crítica de boro, utilizando-se para este fim redes multicamadas com treinamento baseado no algoritmo de retropropagação e uma rede de base radial. Os resultados foram comparados com os cálculos feitos pelo código NEM e demonstraram a viabilidade do uso de redes neurais na substituição desse processo iterativo.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

THE USAGE OF NEURAL NETWORKS TO OBTAIN CRITICAL BORON ACID
CONCENTRATION IN PWR REACTORS

Cícero Rogério da Silva Barros

May/2011

Advisors: Roberto Schirru
Fernando Carvalho da Silva

Department: Nuclear Engineering

The maintenance of any nuclear reactor criticality requires the core composition and geometry arrangement by the neutron seizing materials insertion, in this case boron acid. The critical boron acid concentration is currently determined by an iterative process. As example of it we have the use of NEM (Nodal Expansion Method) (Finnerman et al.,1977). In this work we study the feasibility of using neural networks to obtain the critical boron acid concentration. For this purpose we use a radial basis function network and multilayer networks trained by back-propagation algorithm. The networks results were compared with the NEM code measurements and demonstrated the feasibility of using neural networks to replace this iterative process.

ÍNDICE

1 INTRODUÇÃO	01
1.1– A Condição de Criticalidade	02
1.1.1 – As Barras de Controle	03
1.1.2 – Veneno queimavel	03
1.1.3 – O Ácido Bórico	03
1.2 – A Obtenção da Concentração Crítica de Boro Atualmente	04
1.3 - Os Objetivos da Dissertação e a Metodologia	06
2 REDE NEURAL ARTIFICIAL	08
2.1 - Modelo de Funcionamento de uma Rede Neural Biológica	08
2.2 – Modelo Básico de uma Rede Neural Artificial	09
2.2.1 – Função de Ativação	09
2.2.2 – Neurônio Artificial	11
2.2.3 – Redes Neurais em Multicamadas	12
2.2.4 – Processo de Aprendizagem de uma Rede Neural	13
2.2.5 – O Algoritmo de Retropropagação	14
2.3 – Redes de Bases Radiais – RBR	17
2.3.1 – Redes de Bases Radiais e um Problema Complexo	17
2.3.2 – Aprendizagem das Redes de Bases Radiais	21
2.3.2.1 – Obtenção dos Centros e das Larguras	21
2.3.2.2 – O Algoritmo de Treinamento das redes RBR	22
3 O MÉTODO DE EXPANSÃO NODAL	23
3.1 – Coeficientes Primários	26
3.2 – Coeficientes Secundários	26
3.3 – Coeficientes da Fuga Transversal	28
3.4 – Correntes Parciais de Saída	30

3.5 – Equação de Balanço Nodal.....	31
4 O PROBLEMA DA CONCENTRAÇÃO CRÍTICA DE BORO	32
4.1 – As Informações Necessárias para a Criação das Redes	32
4.2 – Codificação da Entrada e da Saída das Redes Neurais	33
4.2.1 - A Entrada e a Saída das Redes Neurais	39
4.3 - As Fases da Obtenção das Redes	40
4.5 – A Utilização das Redes Treinadas	41
5 RESULTADOS OBTIDOS	43
5.1 – Resultados da Rede de Base Radial	43
5.2 – Resultados das Redes Multicamadas	44
5.3 - Concentração Crítica de Boro	48
6 CONCLUSÕES	50
7 REFERÊNCIAS BIBLIOGRÁFICAS	51
8 ANEXOS	54
8.1 - A Interpolação nas Redes de Funções de Bases Radiais	54
8.2 - As Funções de Base	61
8.3 - A Subrotina que Faz a Montagem da Matriz de Treinamento	62
8.4 - A Subrotina que Faz o Ajuste da Seção de Choque de Absorção	63

LISTA DE FIGURAS

FIGURA 1.1 – Curva da Concentração Crítica de Boro em Função da Queima – Angra I – Ciclo 16	04
FIGURA 1.2 – Modelo Computacional Simplificado para o Cálculo da Concentração Crítica de Boro	05
FIGURA 2.1 - Um Neurônio e Suas Partes	08
FIGURA 2.2 – Função Limiar	09
FIGURA 2.3 – Função Limiar por Partes	10
FIGURA 2.4 – Função Sigmóide	10
FIGURA 2.5 - Modelo Não Linear de um Neurônio	11
FIGURA 2.6 - Uma Rede Multicamada com uma Camada Oculta de Neurônio	13
FIGURA 3.1 - Representação de um nodo	23
FIGURA 4.1 - Uma Vista de Cima de um Núcleo e os Eixos de Simetria	33
FIGURA 4.2 – K EFETIVO em função da Concentração de Boro	39
FIGURA 4.3 - Camada de Entrada e Saída das Redes Junto com o Esquema das Posições dos ECs	39
FIGURA 4.4 - Fluxograma do Programa MATLAB que Calcula a Concentração Crítica de Boro	42

LISTA DE TABELAS

TABELA 4.1 - A Configuração do reator IAEA com Simetria de $1/8$	34
TABELA 4.2 - Configuração do Reator PWR com simetria de $1/4$	35
TABELA 4.3 - Parâmetros Nucleares do Reator IAEA – Dados Originais	36
TABELA 4.4 - Parâmetros Nucleares do Reator IAEA – Gerados – Lado Direito	37
TABELA 4.5 - Parâmetros Nucleares do Reator IAEA – Gerados – Lado Esquerdo ..	37
TABELA 4.6 – Concentração de Boro e $K_{EFETIVO}$	38
TABELA 5.1 - Rede de Base Radial	44
TABELA 5.2 - - Rede Multicamada com 1 Camada Oculta de 5 Neurônios	45
TABELA 5.3 - Rede Multicamada com 2 Camadas Ocultas de 5 Neurônios	46
TABELA 5.4 - Rede Multicamada com 3 Camadas Ocultas de 5 Neurônios	46
TABELA 5.5 - Rede Multicamada com 2 Camadas Ocultas de 10 Neurônios	47
TABELA 5.6 - - Rede Multicamada com 2 Camadas Ocultas de 15 Neurônios	47
TABELA 5.7 - $K_{EFETIVO}$ Obtido com CB critico-rede	48

1 INTRODUÇÃO

Um reator nuclear possui mecanismos e processos de controle, quando em operação, que servem para manter a segurança, a eficiência e a confiabilidade do mesmo durante toda a sua operação. Tais mecanismos e processos de controle são utilizados para garantir a integridade do reator e, por consequência, a segurança dos funcionários que trabalham na usina e da população ao redor dela.

Os mecanismos e processos de controle são muitas vezes baseados nos limites operacionais das variáveis de estado do processo de obtenção da energia nuclear. Estes são determinados pelas especificações técnicas do projeto do reator, tais como: população neutrônica, temperatura, pressão, vazão, nível de água de refrigeração, etc.

Das variáveis de estado acima, utilizaremos a população neutrônica, através do uso do fator de multiplicação do nêutron, que é também chamado $K_{EFETIVO}$, sendo definido como:

$$K_{EFETIVO} = \frac{\text{Número de Nêutrons na geração atual}}{\text{Número de Nêutrons na geração anterior}} \quad (1.1)$$

Considerando o $K_{EFETIVO}$, de um reator nuclear, podemos ter três estados possíveis, que são: subcrítico, crítico e supercrítico.

O reator subcrítico ($K_{EFETIVO} < 1$) significa que o número de nêutrons na geração atual é menor que o número da geração anterior. Como a reação em cadeia em um reator depende do número de nêutrons, isto indica que o reator tende a desligar persistindo a diminuição do número de nêutrons a cada geração.

O reator crítico ($K_{EFETIVO} = 1$) significa que o número de nêutrons na geração atual é igual ao número na geração anterior, que é o objetivo a ser atingido em um reator nuclear durante a sua operação. O número de nêutrons permanece constante a cada geração, ou seja, temos uma reação em cadeia de fissão controlada.

O reator supercrítico ($K_{EFETIVO} > 1$) significa que o número de nêutrons na geração atual é maior que o número da geração anterior, mantendo esta situação, temos uma reação em cadeia descontrolada.

Apesar do $K_{EFETIVO}$, definido na equação (1.1), ser conceitualmente de fácil entendimento, ele não é operacional, devido a não ser possível determinar a população de nêutrons a cada geração, portanto devemos encontrar um meio operacional de calcular o $K_{EFETIVO}$. Isto é feito através da solução da equação de difusão de nêutrons independente do tempo, onde o $K_{EFETIVO}$ é o maior autovalor associado aos autovetores da solução da equação de difusão.

1.1 – A Condição de Criticalidade

Quando o fluxo de nêutrons no reator for independente do tempo, ou seja, a reação de fissão em cadeia passa a ser estacionária dizemos que a condição de criticalidade foi atingida. Com isso, o fluxo de nêutrons pode ser mantido no reator na ausência de fontes externas de nêutrons, a partir daí a fonte de nêutrons será somente os gerados pela própria fissão.

A verificação da criticalidade de um reator é obtida através da solução do sistema de equações de difusão de nêutrons multigrupo, no caso, em que encontramos uma combinação entre a geometria e a composição do núcleo do reator. A geometria do núcleo foi fixada durante o projeto e a construção do mesmo, por isso, para se atingir a criticalidade do reator durante a sua operação só temos sua composição como o elemento a ser alterado. E esta pode ser alterada de três maneiras que são: o uso de barras de controle; de veneno queimável; e de ácido bórico. Em todos os casos utilizam-se substâncias que são absorvedoras de nêutrons.

Estas substâncias absorvedoras de nêutrons podem ser inseridas ou retiradas do núcleo do reator, de acordo com a reatividade, que é definida como:

$$\rho = \frac{K_{EFETIVO} - 1}{K_{EFETIVO}} \quad (1.2)$$

no caso de $K_{EFETIVO} < 1$ temos reatividade negativa e no caso de $K_{EFETIVO} > 1$ temos uma reatividade positiva. Quando a reatividade for negativa é necessário retirar materiais absorvedores de nêutrons para o reator tornar-se crítico, contudo quando a reatividade for positiva devemos acrescentar materiais absorvedores de nêutrons para o reator atingir o estado crítico.

1.1.1 – As Barras de Controle

As barras de controle que são materiais inseridos ou retirados do núcleo do reator durante a sua operação para diminuir ou aumentar a reatividade respectivamente, pois são feitos de materiais absorvedores de nêutrons como prata, índio e cádmio. As barras de controle podem ser usadas para alterar a potência do reator de acordo com a demanda de energia exigida, pois a potência térmica depende do fluxo de nêutrons.

1.1.2 – Veneno queimável

Quando fazemos uma recarga de combustível do reator, esta apresenta geralmente uma alta reatividade para permitir a partida do reator e garantir a plena potência do reator durante toda a sua operação. Devido a esta alta reatividade do combustível utiliza-se o veneno queimável, que são materiais absorvedores de nêutrons consumidos durante a operação, ou seja, diminuindo a sua quantidade, daí o nome de veneno queimável. Como o combustível esta sendo consumido e o veneno queimável também, há certa compensação, fazendo com que a reatividade se mantenha constante.

1.1.3 – O Ácido Bórico

Em reatores com refrigeração e moderação do núcleo com água leve (H_2O) há adição de ácido bórico (H_3BO_3), pois ao iniciar a operação da usina o combustível novo (recarga) apresenta uma alta reatividade positiva. Ao longo do tempo, a concentração de boro vai sendo diminuída (FIGURA 1.1) de forma a compensar a reatividade, ou seja, enquanto a reatividade positiva do combustível vai diminuindo devido à queima do combustível, a reatividade negativa do ácido bórico também é diminuída através de um sistema de controle químico e volumétrico da água do circuito primário do reator, de tal forma a manter a reatividade constante é igual a zero.

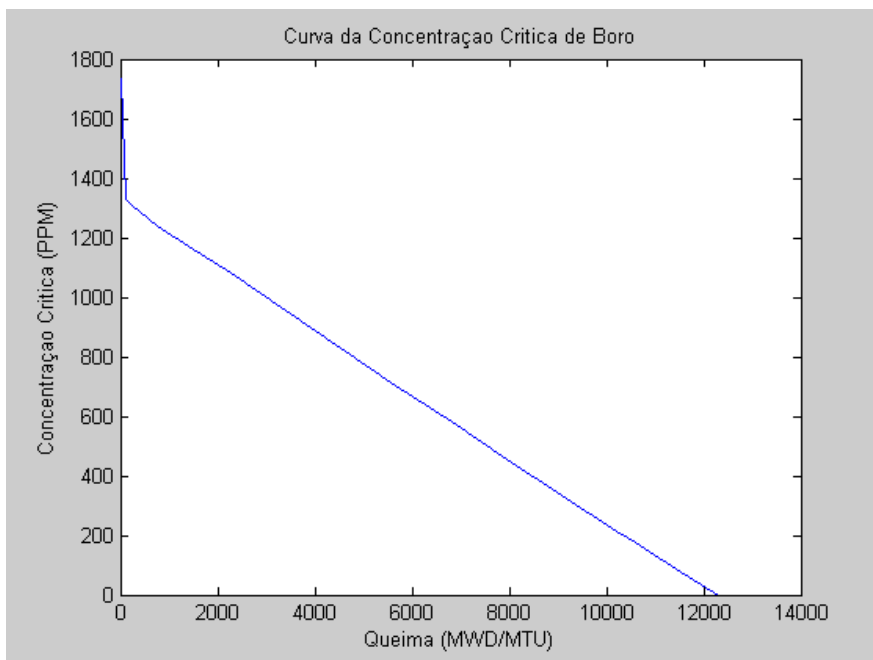


FIGURA 1.1 – Curva da Concentração Crítica de Boro em Função Queima – Angra I – Ciclo 16

1.2 – A Obtenção da Concentração Crítica de Boro Atualmente

Das maneiras citadas acima para o controle do $K_{EFETIVO}$, esta dissertação abordará, qual é Concentração de Boro (CB) que torna um reator crítico. Hoje esta CB é obtida utilizando processos iterativos (Pereira Neto, W. F., 2011), estes ocorrem em três etapas que são: o cálculo do $K_{EFETIVO}$, o cálculo do fluxo de nêutrons e o cálculo da concentração crítica de boro.

Conforme mostra a figura 1.2 do modelo computacional simplificado para o cálculo da concentração crítica de boro, temos o cálculo do fluxo e $K_{EFETIVO}$ e em seguida o teste do critério de convergência do $K_{EFETIVO}$

$$\left| \frac{K_{EFETIVO}^{atual} - K_{EFETIVO}^{anterior}}{K_{EFETIVO}^{atual}} \right| \leq \varepsilon_k \quad (1.3)$$

onde ε_k é a tolerância desejada para o erro no $K_{EFETIVO}$. Quando não ocorrer a convergência do $K_{EFETIVO}$, teremos novos cálculos do fluxo e $K_{EFETIVO}$

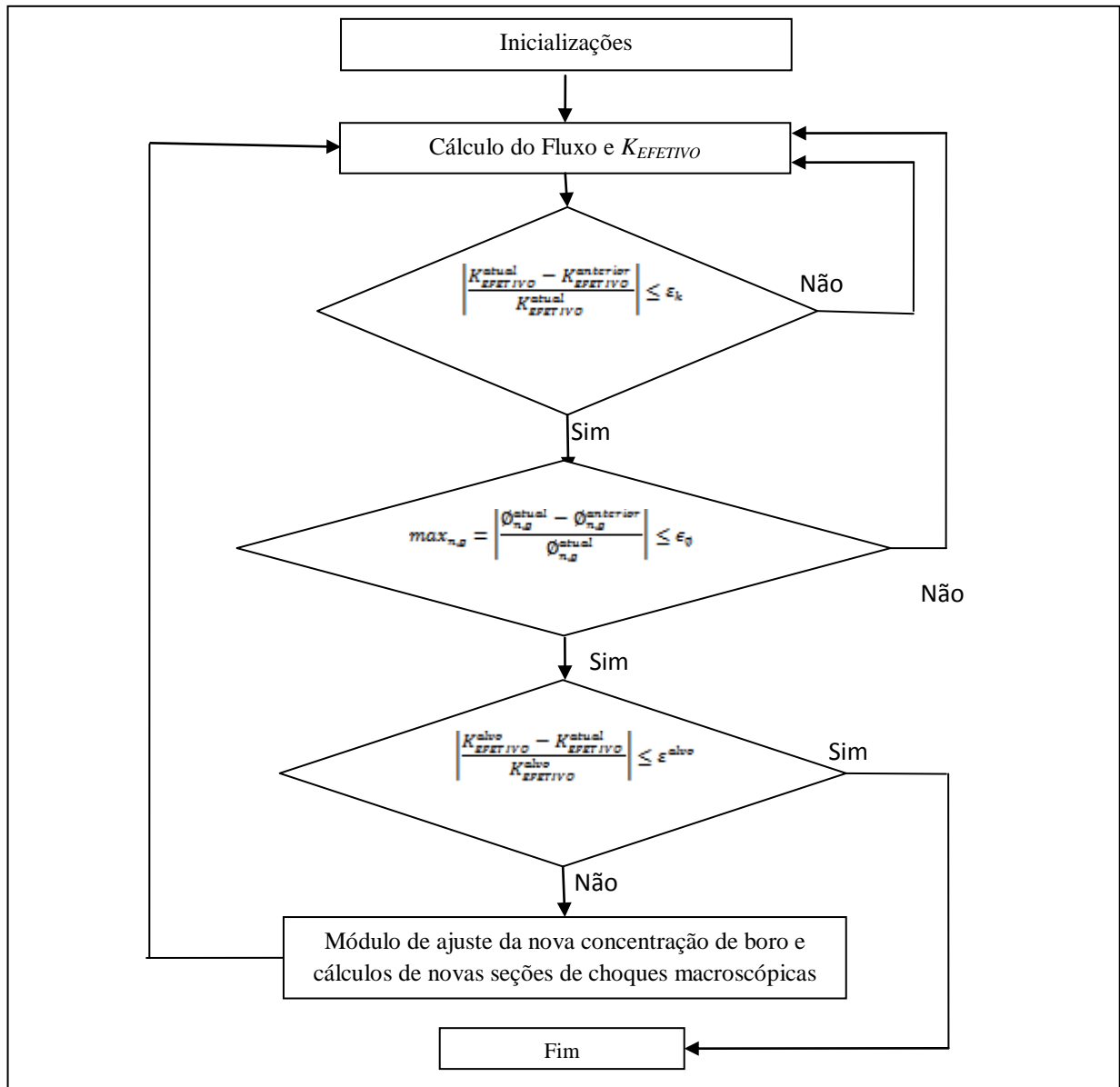


Figura 1.2 - - Modelo computacional simplificado para o cálculo da concentração crítica de boro – Fonte:: Pereira Neto, W. F., 2011.

Em seguida passamos para o teste do fluxo de nêutrons, que tem o seguinte critério de convergência

$$max_{n,g} = \left| \frac{\phi_{n,g}^{atual} - \phi_{n,g}^{anterior}}{\phi_{n,g}^{atual}} \right| \leq \epsilon_{\phi} \quad (1.4)$$

onde ϕ é o fluxo de nêutrons e ϵ_{ϕ} é a tolerância desejada para o erro no fluxo. Não ocorrendo a convergência do fluxo, teremos novos cálculos do fluxo, do $K_{EFETIVO}$ e novo teste de convergência para o $K_{EFETIVO}$.

E finalmente o cálculo da concentração crítica de boro, que é obtida fixando um $K_{EFETIVO}^{alvo} = 1$, e com a concentração de boro inicial, calculamos se o $K_{EFETIVO}^{alvo}$ foi atingido através do seguinte critério de convergência

$$\left| \frac{K_{EFETIVO}^{alvo} - K_{EFETIVO}^{atual}}{K_{EFETIVO}^{alvo}} \right| \leq \varepsilon^{alvo} \quad (1.5)$$

onde ε^{alvo} é a tolerância desejada para o $K_{EFETIVO}^{atual}$.

Em caso de não convergência ocorre o ajuste da nova concentração de boro e cálculo das novas seções de choque macroscópicas, bem como o cálculo do fluxo e $K_{EFETIVO}$ seguidos de novos testes para convergência do $K_{EFETIVO}$ e do fluxo de nêutrons. Haverá também um novo teste para verificar a obtenção do $K_{EFETIVO}^{alvo}$.

No caso do reator IAEA (Argonne National Laboratory, 1977 e Guimarães, C. de S., da Silva, F. C., Martinez, A. S., 2007) foi utilizado no processo descrito acima (através do código NEM), obtendo a concentração crítica de boro junto com o $K_{EFETIVO}$ e o fluxo de nêutron em cerca 5 segundos, enquanto a obtenção somente do $K_{EFETIVO}$ e do fluxo de nêutron levou 2,5 segundos. Portanto, a obtenção da concentração crítica de boro eleva significativamente o tempo computacional.

1.3 - Os Objetivos da Dissertação e a Metodologia

O objetivo principal desta dissertação é a pesquisa do uso de redes neurais para a obtenção da concentração crítica de boro. Para isso utilizou-se de um conjunto de dados contendo a posição dos vários tipos de Elementos Combustíveis (EC), a CB, os parâmetros nucleares (seção de choque de absorção, seção de choque de fissão vezes número médio de nêutrons emitidos na fissão, o coeficiente de difusão, a seção de choque de espalhamento, a seção de choque de fissão vezes a energia liberada por fissão) dos vários tipos de EC para dois grupos de energia e o $K_{EFETIVO}$.

Como objetivo secundário desta dissertação tem o estudo da viabilidade de se substituir processos iterativos de alto custo computacional em códigos de física de reatores por redes neurais. A motivação para tal é que os processos que utilizam redes neurais apresentam uma grande vantagem quando se considera o tempo de computação.

Para alcançar os objetivos mencionados anteriormente, foram treinadas Redes de Bases Radiais e Redes Multicamadas (Algoritmo de Retropropagação), com vários dados nucleares, CB e $K_{EFETIVO}$ conhecidos. Estas Redes treinadas foram utilizadas para calcular a CB que torna o reator crítico ($K_{EFETIVO} = 1$). No caso das redes multicamadas foram testadas varias arquiteturas para facilitar a analise dos resultados. Utilizou-se as mesmas quantidades de camadas ocultas com diferentes números de neurônios nas camadas ocultas, e diferentes quantidades de camadas ocultas com o mesmo número de neurônios.

2 REDE NEURAL ARTIFICIAL

2.1 - Modelo de Funcionamento de uma Rede Neural Biológica

A rede neural artificial é baseada em uma analogia de como funciona o cérebro humano. O cérebro é constituído de neurônios, que são as células nervosas. Estas são compostas basicamente por (KOVACS, Z. L., 1996):

Soma → corpo celular – onde ocorrem os processos metabólicos da célula;

Dendritos → por onde chegam os impulsos nervosos advindos de outras células;

Axônio → por onde o impulso nervoso da célula é transmitido para outras células.

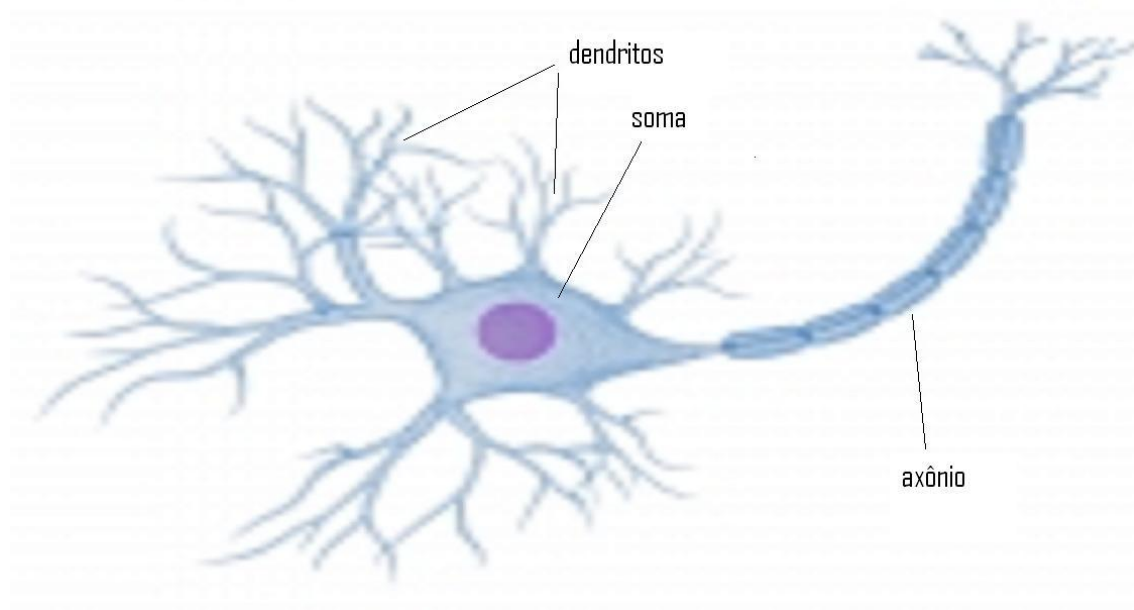


FIGURA 2.1 - Um Neurônio e Suas Partes

O axônio de um neurônio se interliga ao dendrito de outro, e o axônio deste a outros dendritos de outros neurônios e assim por diante. No *soma* de cada neurônio, o impulso nervoso é alterado através dos processos metabólicos e transmitido pelo axônio. O impulso nervoso não é nada mais que um impulso elétrico produzido no *soma* do neurônio. Este impulso pode ser aumentado ou diminuído no *soma*.

Qualquer informação ou dado recebido de meio externo é transformado em impulsos elétricos, e estes processados nos neurônios. Os impulsos elétricos processados são transformados na saída desejada.

2.2 - Modelo Básico de uma Rede Neural Artificial

2.2.1 – Função de Ativação

O processo metabólico que ocorre no soma pode ser representado por funções, chamadas de funções de ativação, geralmente limitadas. Descrevemos aqui três tipos básicos de funções de ativação, que são (HAYKIN, S., 2001):

a) Função Limiar → O neurônio que se utiliza da função limiar é conhecido como modelo de McCulloch-Pitts devido ao trabalho pioneiro de McCulloch e Pitts em 1943. Esta função é definida como:

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq 0 \\ 0, & \text{se } v < 0 \end{cases} \quad (2.1)$$

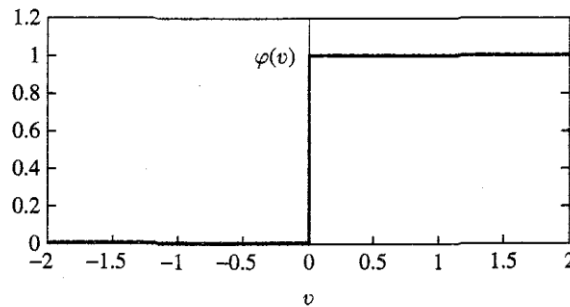


FIGURA 2.2 - Função Limiar – Fonte: Haykin, S., 2001

Que é a função degrau, que descreve a propriedade tudo-ou-nada do modelo de McCulloch-Pitts

b) Função Limiar por Partes → Pode ser considerada uma aproximação de um amplificador não linear dentro da região linear de operação. Um exemplo desta função pode ser:

$$\varphi(v) = \begin{cases} 0 & \text{se } v \leq -\frac{1}{2} \\ v + \frac{1}{2} & \text{se } -\frac{1}{2} < v < \frac{1}{2} \\ 1 & \text{se } v \geq \frac{1}{2} \end{cases} \quad (2.2)$$

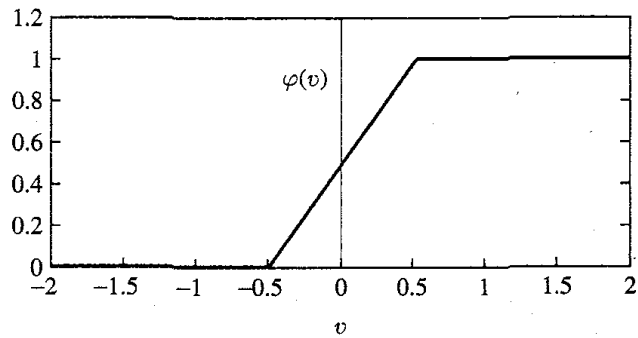


FIGURA 2.3 - Função Limiar por Partes - Fonte: Haykin, S., 2001

Na função limiar por partes se a região linear de operação é mantida entre os valores limites inferior e superior, surge um combinador linear, no caso contrario, ela se reduz a função limiar.

c) Função Sigmóide → É a função de ativação mais comum devido a sua propriedade de ser uma função estritamente crescente e diferenciável. Um exemplo de função sigmóide é a função logística, que é definida como:

$$\varphi(v) = \frac{1}{1 + e^{-av}} \quad (2.3)$$

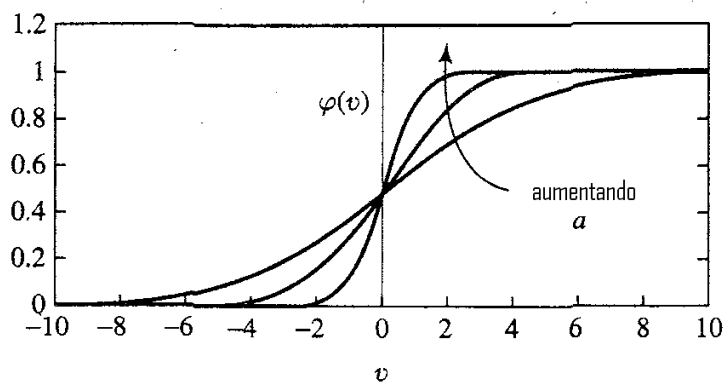


FIGURA 2.4 - Função Sigmóide - Fonte: Haykin, S., 2001

onde a é o parâmetro de inclinação da função sigmóide. No limite, quando o parâmetro se aproxima do infinito, a função sigmóide tende para função limiar. A função limiar só pode assumir os valores 0 ou 1, já a função sigmóide pode assumir qualquer valor entre 0 e 1, ou seja, um intervalo contínuo. Sendo esta contínua e diferenciável.

As funções de ativação descritas acima só tem como respostas valores entre 0 e 1, e algumas vezes é desejável que a função de ativação apresente valores positivos e negativos, como por exemplo, um valor entre -1 e 1, que pode ser alcançado pela função tangente hiperbólica:

$$\varphi(v) = \tanh(av) \quad (2.4)$$

onde a é um parâmetro de ajuste da função tangente hiperbólica, que é uma função contínua e diferenciável. A importância de utilizar funções contínuas e diferenciáveis como funções de ativação ficará clara quando se obtêm a equação de correção dos pesos sinápticos na seção 2.2.5.

2.2.2 – Neurônio Artificial

Um neurônio artificial que é a unidade fundamental de uma rede neural artificial, onde são processadas as informações que chegam a rede, tem o seguinte esquema, mostrada na figura 2.5 – modelo não linear de um neurônio.

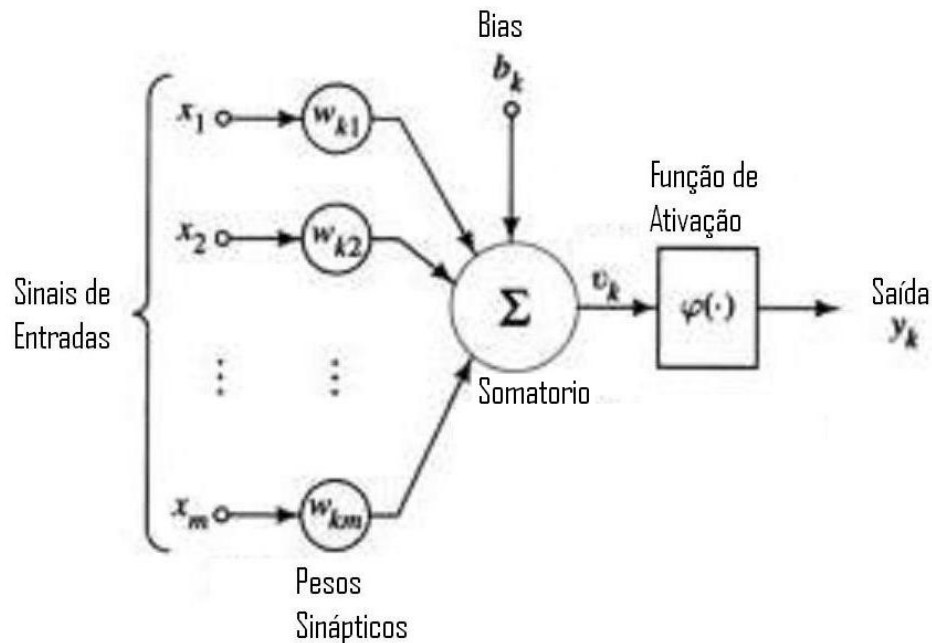


FIGURA 2.5 - Modelo Não Linear de um Neurônio - Fonte: Haykin, S., 2001

No modelo de neurônio podem ser identificados os três elementos básicos, que são (HAYKIN, S., 2001):

1 – Um conjunto de sinapses que é a conexão entre dois neurônios, esta conexão é representada por pesos e estes são chamados de Pesos Sinápticos e representados por w_{ki} . O primeiro índice k se refere ao neurônio e o segundo i se refere a sinapse de entrada.

2 – Um somatório que é soma das entradas x_i vezes o peso sináptico w_{ki} do respectivo neurônio k , esta operação descreve um combinador linear.

3 – Uma função de ativação que limita ou restringe a amplitude da saída, já descritos em alguns exemplos acima.

O modelo de neurônio da figura 2.5 também inclui bias de entradas externas e representado por b_k que pode produzir um aumento ou diminuição da entrada da função de ativação, dependendo se o valor das bias é positivo ou negativo, respectivamente.

Podemos descrever matematicamente um neurônio k através do seguinte par de equações:

$$v_k = \sum_{i=1}^m w_{ki} \cdot x_i + b_k \quad (2.5)$$

$$y_k = \varphi(v_k) \quad (2.6)$$

onde a equação (2.5) é chamada de campo induzido local e a equação (2.6) é função de ativação aplicada no campo induzido local.

2.2.3 – Redes Neurais em Multicamadas

Uma rede neural artificial em multicamadas consiste num conjunto de neurônios artificiais ligados em camadas, ou seja, uma camada de neurônios se liga a camada seguinte e esta se liga a próxima e assim sucessivamente até a última camada. A figura 2.6 mostra uma rede multicamada com uma camada de entrada, onde os sinais de entrada estão na primeira camada, e depois temos uma camada oculta e uma camada de saída, por onde obtemos a resposta da rede. O número de camadas ocultas pode variar, podendo existir redes com mais de uma camada oculta.

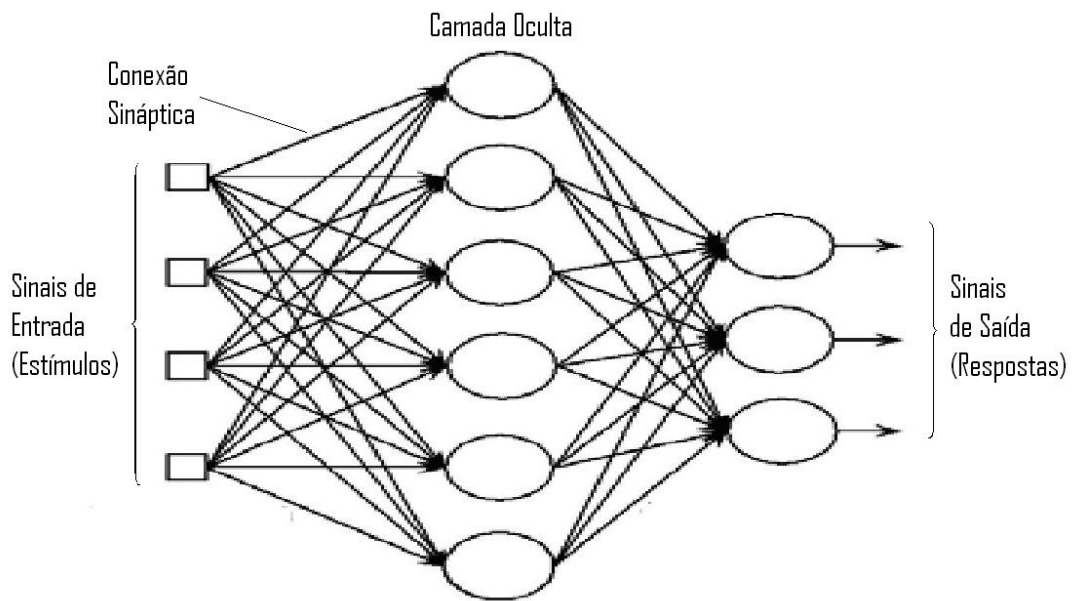


FIGURA 2.6 -Uma Rede Multicamada com uma Camada Oculta de Neurônio - Fonte: Haykin, S., 2001

O conhecimento de uma rede se encontra nos pesos sinápticos que são atribuídos às conexões sinápticas. Estes pesos sinápticos são inicialmente arbitrários, ou seja, uma rede projetada inicialmente não apresenta nenhum conhecimento.

2.2.4 – Processo de Aprendizagem de uma Rede Neural

Antes de utilizar uma rede devemos primeiramente fazer com que ela passe por um processo de aprendizagem, que nada mais é, do que o ajuste dos pesos sinápticos da rede para que ela adquira um conhecimento. Os processos de aprendizagem podem ser supervisionados ou não supervisionados.

O processo supervisionado consiste em apresentar à rede um conjunto de padrões dos quais conhecemos previamente a resposta desejada, em compararmos a resposta da rede com a resposta desejada e em fazer ajustes nos pesos w_{ki} de acordo com o erro encontrado. Este procedimento de ajustes dos w_{ki} é feito até que os erros sejam reduzidos até uma tolerância previamente definida.

O processo não supervisionado é baseado no fato de não se conhecer previamente a resposta desejada e pela competição existente nos neurônios da camada de saída. Nesta competição o neurônio vencedor é o que apresenta a saída da rede. As redes que aprendem por este processo são utilizadas para classificar um certo número de

padrões em um conjunto de categorias, e cada neurônio vencedor representa uma categoria (DA SILVA S. F., 2000).

Neste trabalho só será utilizado o processo supervisionado de aprendizagem e portanto será abordado este tipo de aprendizado através do algoritmo de retropropagação e das redes de bases radiais.

2.2.5 O Algoritmo de Retropropagação

Antes de apresentar o algoritmo de retropropagação, devemos determinar como é feita a correção dos pesos sinápticos (HAYKIN, S., 2001). O erro no neurônio k , na n -ésima iteração dos exemplos de treinamento, em um nó de saída, é dado por:

$$e_k(n) = d_k(n) - y_k(n) \quad (2.7)$$

sendo $d_k(n)$ a saída desejada e $y_k(n)$ o resultado obtido pelo neurônio k .

O erro quadrático instantâneo do neurônio k é definido como $\frac{1}{2}e_k^2(n)$, somando todos $\frac{1}{2}e_k^2(n)$ da camada de saída, obtemos:

$$\mathcal{E}(n) = \frac{1}{2} \sum_{k \in C} e_k^2 \quad (2.8)$$

onde $k \in C$ significa uma soma sobre o conjunto C que contem todos os neurônios da camada saída da rede. Considerando N o número total exemplos apresentados à rede para treinamento, podemos definir o erro quadrático médio, como:

$$\mathcal{E}_{med} = \frac{1}{N} \sum_{n=1}^N \mathcal{E}(n) \quad (2.9)$$

O erro quadrático médio é uma função de todos os parâmetros livres, isto é, os pesos sinápticos e as *bias* da rede. Esta função representa o desempenho da aprendizagem, sendo o objetivo da aprendizagem de uma rede neural minimizar a função \mathcal{E}_{med} , que é feito através do ajuste dos parâmetros livres.

Considere o neurônio k alimentado por outros neurônios da camada anterior na n -ésima iteração dos exemplos de treinamento, podemos então escrever as equações do neurônio k como:

$$v_k(n) = \sum_{i=1}^m w_{ki}(n) \cdot x_i(n) + b_k(n) \quad (2.10)$$

$$y_k(n) = \varphi_k(v_k(n)) \quad (2.11)$$

O método utilizado para determinar a correção dos pesos é baseado no gradiente de uma função, lembrando que o gradiente de uma função aponta no sentido em que a função cresce mais, portanto menos o gradiente aponta no sentido de menor valor. Então, calculando o gradiente de $\mathcal{E}(n)$ e usando a regra da cadeia, temos:

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ki}} = \frac{\partial \mathcal{E}(n)}{\partial e_k(n)} \frac{\partial e_k(n)}{\partial y_k(n)} \frac{\partial y_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial w_{ki}} \quad (2.12)$$

Diferenciando a (2.8) em relação a $e_k(n)$, obtemos:

$$\frac{\partial \mathcal{E}(n)}{\partial e_k(n)} = e_k(n) \quad (2.13)$$

Diferenciando a (2.7) em relação a $y_k(n)$, obtemos:

$$\frac{\partial e_k(n)}{\partial y_k(n)} = -1 \quad (2.14)$$

Agora, diferenciando a (2.11) em relação a $v_k(n)$, obtemos:

$$\frac{\partial y_k(n)}{\partial v_k(n)} = \varphi'_k(v_k(n)) \quad (2.15)$$

Finalmente, diferenciando a (2.10) em relação a $w_{ki}(n)$, obtemos:

$$\frac{\partial v_k(n)}{\partial w_{ki}(n)} = x_i(n) \quad (2.16)$$

Substituindo as equações (2.13) a (2.16) na equação (2.12), temos então:

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ki}} = -e_k(n) \varphi'_k(v_k(n)) x_i(n) \quad (2.17)$$

A correção do peso entre os neurônios i e k é $\Delta w_{ki}(n)$ e pode ser calculada pela regra delta (HAYKIN, S., 2001), que tem a seguinte definição:

$$\Delta w_{ki}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ki}} \quad (2.18)$$

onde η é o parâmetro da taxa de aprendizagem do algoritmo retropropagação. O sinal negativo indica a busca do mínimo da função \mathcal{E}_{med} .

Substituindo a (2.17) na (2.18) obtemos

$$\Delta w_{ki}(n) = \eta e_k(n) \varphi'_k(v_k(n)) x_i(n) \quad (2.19)$$

Definindo o gradiente local $\delta_k(n)$, como:

$$\delta_k(n) = -\frac{\partial \mathcal{E}(n)}{\partial v_k(n)} = -\frac{\partial \mathcal{E}(n)}{\partial e_k(n)} \frac{\partial e_k(n)}{\partial y_k(n)} \frac{\partial y_k(n)}{\partial v_k(n)} = e_k(n) \varphi'_k(v_k(n)) \quad (2.20)$$

Finalmente, temos para $\Delta w_{ki}(n)$

$$\Delta w_{ki}(n) = \eta \delta_k(n) x_i(n) \quad (2.21)$$

A taxa de aprendizagem η apresenta o seguinte problema, quanto menor for a taxa menor serão as variações nos pesos sinápticos, mas em contrapartida é mais lenta a aprendizagem, no caso contrario em que a taxa de aprendizagem é maior para acelerar a aprendizagem, as grandes variações nos pesos sinápticos podem tornar a rede instável. Uma solução para aprendizagem lenta e a instabilidade é acrescentar um novo termo na equação (2.21) que depende da correção anterior do peso sináptico e um novo parâmetro que chamamos de momento (HAYKIN, S., 2001), com isso, a equação de correção do peso sináptico, fica:

$$\Delta w_{ki}(n) = \alpha \Delta w_{ki}(n-1) + \eta \delta_k(n) x_i(n) \quad (2.22)$$

onde α é a constante de momento

Algoritmo de Retropropagação

- 1 – Iniciamos os pesos entre os neurônios.
- 2 - Enquanto critério de parada não for satisfeito faça

2.1 – Apresentação dos Exemplos de Treinamento

2.2 – Computação para Frente (Propagação)

Calcula a saída da rede com os pesos atuais

$$v_k^{(l)}(n) = \sum_{i=0}^m w_{ki}^{(l)}(n) x_i^{l-1}(n)$$

$$x_k^{(l)} = \varphi_k(v_k(n))$$

2.3 – Computação para Trás (Retropropagação)

Corrige os pesos da rede através de:

$$w_{ki}^{(l)}(n+1) = w_{ki}^{(l)}(n) + \alpha \{w_{ki}^{(l)}(n-1)\} + \eta \delta_k^{(l)}(n) x_i^{l-1}(n)$$

$\alpha \rightarrow$ constante de momento

$\eta \rightarrow$ taxa de aprendizagem

Fim Enquanto

- 3 – Fim do Algoritmo Retropropagação

2.3 - Redes de Bases Radiais - RBR

2.3.1 - Redes de Bases Radiais e um Problema Complexo

A construção de uma rede de base radial pode ser justificada pelo fato de que um problema complexo, poderá ser solucionado pela transformação de um espaço de baixa dimensionalidade em um de alta dimensionalidade através de funções não lineares. Uma vez que os problemas complexos, geralmente não permitem a separabilidade de padrões em espaços de baixa dimensionalidade, o que já não ocorre em espaços de alta

dimensionalidade. Isto está de acordo com o teorema de Cover sobre a separabilidade de padrões, que diz:

“Um problema complexo de classificação de padrões disposto não linearmente em um espaço de alta dimensão tem maior probabilidade de ser linearmente separável do que em um espaço de baixa dimensionalidade.” (Cover, T. M., 1965 e HAYKIN, S. 2001)

O problema de separabilidade dos padrões é que há um ganho no mapeamento de um espaço de entrada de baixa dimensão para um novo espaço de alta dimensão de acordo com o teorema de Cover, o que pode transformar um problema não linearmente separável em um problema linearmente separável.

Vamos considerar uma rede multicamada com uma camada de entrada, uma única camada oculta e uma camada de saída com um único neurônio. O uso de um só neurônio na camada de saída foi intencional para simplificar a exposição abaixo, sem nenhuma perda de generalidade. O projeto da rede faz um mapeamento não linear entre o espaço de entrada e o espaço oculto, e depois um mapeamento linear entre o espaço oculto e o espaço de saída. Considerando m_0 a dimensão do espaço de entrada, a rede como um todo faz um mapeamento do espaço de entrada para um espaço unidimensional, que pode ser escrito como:

$$s: \mathcal{R}^{m_0} \rightarrow \mathcal{R}^1 \quad (2.23)$$

o mapa s pode ser considerado uma hipersuperfície Γ contido \mathcal{R}^{m_0+1} . De modo geral Γ é desconhecido.

A aprendizagem apresenta duas fases que são a do treinamento e a de generalização. O treinamento corresponde a otimização dos procedimentos de ajuste da hipersuperfície Γ , baseados nos dados conhecidos apresentados à rede na forma de exemplos de treinamento, que são padrões de entrada-saída. A generalização pode ser vista como a interpolação entre os pontos de dados, com a interpolação sendo feita ao longo da superfície obtida pelo procedimento de ajuste, como uma aproximação ótima à superfície verdadeira Γ .

Com isso, sendo necessária uma teoria de interpolação multivariada em espaços de altas dimensões, o problema de interpolação pode ser resolvido usando a interpolação no sentido mais restrito, que pode ser formulado como:

“Dado um conjunto de N pontos diferentes $\{x_i \in \mathcal{R}^{m_0} | i = 1, 2, \dots, N\}$ e um conjunto correspondente de N números reais $\{d_i \in \mathcal{R}^1 | i = 1, 2, \dots, N\}$, encontre uma função $\{F: \mathcal{R}^N \rightarrow \mathcal{R}^1\}$ que satisfaça a condição de interpolação” (HAYKIN, S. 2001):

$$F(x_i) = d_i \quad i = 1, 2, \dots, N \quad (2.24)$$

Na interpolação estrita, a superfície de interpolação é obrigada a passar por todos os pontos de treinamento. As funções de bases radiais implicadas F (ver anexo 8.1 A Interpolação nas Redes de Funções de Bases Radiais) devem ser escolhidas, tendo a seguinte forma:

$$F(x_i) = \sum_{i=1}^N w_i \varphi(\|x_j - x_i\|) \quad (2.25)$$

onde $\{\varphi(\|x_j - x_i\|), i = 1, 2, \dots, N\}$ é um conjunto de N funções geralmente não lineares arbitrárias, conhecidas como funções de base radial e $\|x_j - x_i\|$ representa a norma, que geralmente é euclidiana. Os dados conhecidos $\{x_i \in \mathcal{R}^{m_0} | i = 1, 2, \dots, N\}$ são utilizados como centro das funções de base radial. Substituindo a condição de interpolação (2.24) na equação (2.25), obtemos o conjunto de equações lineares simultâneas para os coeficientes desconhecidos da expansão $\{w_i\}$:

$$\begin{bmatrix} \varphi_{11} & \cdots & \varphi_{1N} \\ \vdots & \ddots & \vdots \\ \varphi_{N1} & \cdots & \varphi_{NN} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_N \end{bmatrix} \quad (2.26)$$

onde

$$\varphi_{ij} = \varphi(\|x_j - x_i\|) \quad (i, j) = 1, 2, \dots, N \quad (2.27)$$

Definindo :

$$\mathbf{d} = [d_1 \quad \cdots \quad d_N]^T \quad (2.28)$$

$$\mathbf{w} = [w_1 \quad \cdots \quad w_N]^T \quad (2.29)$$

$$\Phi = \{\varphi_{ij} | (i, j) = 1, 2, \dots, N\} \quad (2.30)$$

os vetores \mathbf{d} e \mathbf{w} de dimensão N por 1 são a resposta desejada e o peso linear, respectivamente, já Φ representa uma matriz N por N com os elementos de φ_{ij} , que chamamos de matriz de interpolação, sendo N o tamanho da amostra de treinamento. Podemos escrever a equação (2.26) de forma compacta usando as equações (2.28) a (2.30), o que nos dá:

$$\Phi \mathbf{w} = \mathbf{d} \quad (2.31)$$

Assumindo que Φ seja não singular, portanto existe uma matriz inversa Φ^{-1} , podemos encontrar a solução de (2.31) para \mathbf{w} , obtendo:

$$\mathbf{w} = \Phi^{-1} \mathbf{d} \quad (2.32)$$

Como podemos ter certeza que a matriz de interpolação Φ é não singular, a resposta desta questão é dada pelo teorema de Micchelli, que diz:

“Considere $\{\mathbf{x}_i\}_{i=1}^N$ seja um conjunto de pontos distintos em \mathcal{R}^{m_0} . Então, a matriz de interpolação Φ , N por N , cujo elemento ij é $\varphi_{ij} = \varphi(\|\mathbf{x}_j - \mathbf{x}_i\|)$, é não singular” (Micchelli, C. A., 1986 e HAYKIN, S. 2001).

Existe uma grande classe de funções de base radial que é atendida pelo teorema Micchelli, entre elas temos as funções gaussianas, que tem a seguinte forma:

$$\varphi(x_i) = \exp\left(-\frac{1}{2\sigma_j^2} \|\mathbf{x}_j - \mathbf{x}_i\|\right) \quad (2.33)$$

onde \mathbf{x}_j representa o centro da função e σ_j representa a sua largura. Podemos substituir a equação (2.33) na (2.25), obtemos então:

$$F(x_i) = \sum_{i=1}^N w_i \exp\left(-\frac{1}{2\sigma_j^2} \|x_j - x_i\|\right) \quad (2.34)$$

onde w_i são os pesos.

As Redes de Bases Radiais (RBR) são uma variação das redes multicamadas que apresentam uma camada de entrada, uma única camada oculta e a camada de saída da rede, além disso, as funções de ativação que existem nas camadas não são funções sigmóides. As funções de ativação na camada oculta são funções gaussianas normalizadas, já na camada de saída são funções lineares (HAYKIN, S. 2001).

2.3.2 - Aprendizagem das Redes de Bases Radiais

O aumento da dimensionalidade do problema a ser resolvido está associado ao número de neurônios da camada oculta, ou seja, quanto maior o número de neurônios da camada oculta maior é o aumento da dimensão. Com isso durante o treinamento, o número de neurônios da camada oculta vai aumentar até que encontre a dimensão necessária para que ocorra a separabilidade do problema.

Os parâmetros ajustáveis σ_j , x_j e w_i , que são as larguras das funções gaussianas, os centros das funções gaussianas e os pesos das funções lineares da camada de saída respectivamente, foram estabelecidos de forma minimizar o erro médio entre a saída da rede e a saída desejada.

O treinamento de uma RBR pode ser feito em dois estágios, que são: Primeiramente obtemos os centros das funções gaussianas e as larguras das funções gaussianas de cada neurônio da camada oculta. Por último treinamos os pesos da camada de saída.

2.3.2.1 - Obtenção dos Centros e das Larguras das Gaussianas

a) Os Centros das Gaussianas

Os centros das funções gaussianas podem ser obtidos por um método simples, que corresponde a um centro para cada vetor de entrada do conjunto de treinamento, mas como os vetores de treinamento tendem a possuir grupos de centros muito próximos uns dos outros, ou seja, vários vetores com centros numa vizinhança comum. Por isso, esse método resulta em mais funções gaussianas do que o necessário. Há

outros métodos que podem ser utilizados para determinar os centros desses grupos sendo estes métodos mais elaborados.

b) As Larguras das Gaussianas

As larguras das funções gaussianas podem ser obtidas como sendo a distância quadrática média entre o seu centro e os vizinhos destes centros.

2.3.2.2 - O Algoritmo de Treinamento das redes RBR

O Algoritmo de Treinamento

- 1 – Aplicar um vetor de entrada x_i do conjunto de treinamento
- 2 – Calcular as saídas da camada oculta (usando os centros e as larguras obtidos em 2.3.2.1)
- 3 – Computar o vetor saída da camada linear e erro desta
- 4 – Ajustar os pesos da camada linear
- 5 – Repetir os passos 1 a 4 até completar o conjunto de treinamento
- 6 – Repetir os passos 1 a 5 até que erro seja pequeno ou número máximo de iterações seja atingido.

2 O MÉTODO DE EXPANSÃO NODAL

Os métodos nodais modernos de malha grossa obtêm, de maneira rápida e com bastante precisão, os fluxos médios de nêutrons nos nodos, as correntes líquidas e os fluxos médios de nêutrons nas áreas das faces destes nodos, mesmo que o nodo, retratado na figura 3.1, tenha a dimensão da área transversal de um elemento combustível. O Método de Expansão Nodal, de agora em diante chamado NEM (do inglês Nodal Expansion Method), é um dos métodos nodais mais usados atualmente, e este será usado nesta dissertação.

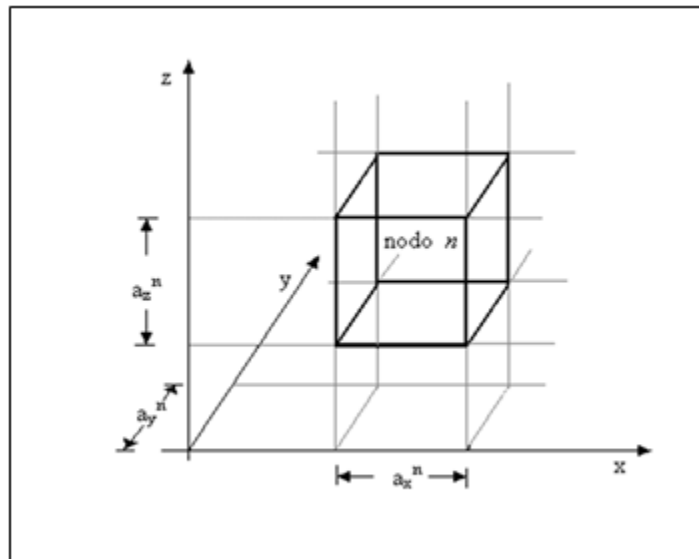


Figura .3.1 - Representação de um nodo - Fonte:: Pereira Neto, W. F., 2011.

A obtenção do NEM é feita a partir da equação da continuidade de nêutrons e da lei de Fick, que são respectivamente:

$$\begin{aligned}
 \vec{\nabla} \cdot \vec{j}_g(x, y, z) + \Sigma_{Rg}(x, y, z)\phi_g(x, y, z) \\
 &= \frac{1}{K_{EFETIVO}} \chi_g \sum_{g'=1}^G \nu \Sigma_{fg'}(x, y, z)\phi_{g'}(x, y, z) \\
 &+ \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{gg'}(x, y, z)\phi_{g'}(x, y, z)
 \end{aligned} \tag{3.1}$$

e

$$\vec{j}_g(x, y, z) = -D_g(x, y, z) \sum_{u=x,y,z} \frac{\partial \Phi_g(x, y, z)}{\partial u} \hat{e}_u \tag{3.2}$$

A equação (3.1) apresenta-se na sua formulação mais geral, no que diz respeito aos grupos de energia, mas é amplamente utilizada a discretização para dois grupos de energia e portanto foi adotado daqui em diante $G = 2$ (Silva, F. C. e Martinez, A. S., 2003).

Devido o núcleo do reator ser dividido em nodos de volume definidos V_n , onde os parâmetros nucleares são homogeneizados, pode-se integrar a equação (3.1) no volume V_n , resultando na equação de balanço nodal, ou seja,

$$\begin{aligned} \sum_{u=x,y,z} (J_{gur}^n - J_{gul}^n) / \alpha_u^n + \Sigma_{Rg}^n \phi_g^n \\ = \frac{1}{K_{EFETIVO}} \chi_g \sum_{g'=1}^2 \nu \Sigma_{fg'} \phi_{g'} \\ + \sum_{\substack{g'=1 \\ g' \neq g}}^2 \Sigma_{gg'} \phi_{g'} \end{aligned} \quad (3.3)$$

E integrando a equação (3.2) em uma área transversal à direção u do nodo, temos:

$$J_{gus}^n = J_{gus}^{+n} - J_{gus}^{-n} = -D_g^n \frac{d\Psi_{gu}^n(u)}{du} \Big|_{u=u'} \quad (3.4)$$

Os fluxos médios de nêutrons nodais (ϕ_g^n), as correntes parciais médias de nêutrons nas fases ($J_{gus}^{\pm n}$) e os fluxos médios de nêutrons nas faces ($\Psi_{gu}^n(u)$) são definidos como:

$$\phi_g^n \equiv \frac{1}{V_n} \int_0^{\alpha_x^n} \int_0^{\alpha_y^n} \int_0^{\alpha_z^n} \phi_g(x, y, z) dx dy dz \quad (3.5)$$

$$J_{gus}^{\pm n} \equiv \frac{1}{\alpha_v^n \alpha_w^n} \int_0^{\alpha_v^n} \int_0^{\alpha_w^n} J_{gu}^{\pm}(u_s^n, v, w) dv dw \quad (3.6)$$

$$\Psi_{gu}^n(u) \equiv \frac{1}{\alpha_v^n \alpha_w^n} \int_0^{\alpha_v^n} \int_0^{\alpha_w^n} \Phi_g(u, v, w) dv dw \quad (3.7)$$

onde, u indica a direção x , y ou z e s indica em qual face do nodo está sendo realizada a média. Sendo l para a face inferior, da frente ou esquerda e r para a face superior, detrás ou direita do nodo.

Por outro lado, tem-se que:

$$u_s^n = \begin{cases} 0 & \text{se } s = l \\ a_u^n & \text{se } s = r \end{cases} \quad (3.8)$$

Substituindo a lei de Fick, equação (3.2), na equação da continuidade, equação (3.1), e integrando na área transversal a uma direção u qualquer num nodo n , resulta:

$$\begin{aligned} -D_g^n \frac{d^2 \psi_{gu}^n(u)}{du^2} + \Sigma_{Rg}^n \Psi_{gu}^n(u) \\ = \frac{1}{K_{EFETIVO}} \chi_g \sum_{g'=1}^2 v \Sigma_{fg'} \Psi_{g'u}^n(u) \\ + \sum_{\substack{g'=1 \\ g'=g}}^2 \Sigma_{gg'} \Psi_{g'u}^n(u) - L_{gu}^n(u) \end{aligned} \quad (3.9)$$

onde $L_{gu}^n(u)$ é o termo que contabiliza a fuga transversal de nêutrons à direção u , a qual é definida pela seguinte equação:

$$L_{gu}^n(u) \equiv \int_0^{a_v^u} \int_0^{a_w^u} \left(\frac{\partial^2 \phi_g(u, v, w)}{\partial v^2} + \frac{\partial^2 \phi_g(u, v, w)}{\partial w^2} \right) dv dw \quad (3.10)$$

Uma das particularidades do NEM é que $\Psi_{gu}^n(u)$ é expandido em polinômios de quarto grau e $L_{gu}^n(u)$ é expandido em polinômios de segundo grau, da seguinte forma:

$$\Psi_{gu}^n(u) = \sum_{i=0}^4 c_{igu}^n h_i \left(\frac{u}{a_u^n} \right) \quad (3.11)$$

$$L_{gu}^n(u) = \sum_{i=0}^2 \alpha_{igu}^n h_i \left(\frac{u}{a_u^n} \right) \quad (3.12)$$

onde c_{0gu}^n , c_{1gu}^n e c_{2gu}^n são chamados de coeficientes primários, c_{3gu}^n e c_{4gu}^n são chamados de coeficientes secundários e $h_i \left(\frac{u}{a_u^n} \right)$ são as funções de base do NEM (ver Anexo 8.2 -As Funções de Base).

3.1 - Coeficientes Primários

Os coeficientes primários (c_{0gu}^n , c_{1gu}^n e c_{2gu}^n) são obtidos aplicando a condição de consistência, qual seja:

$$\Phi_g^n = \frac{1}{a_u^n} \int_0^{a_u^n} \Psi_{gu}^n(u) du \quad (3.13)$$

E a aproximação da difusão, qual seja:

$$\Psi_{gus}^n \equiv \Psi_{gu}^n(u_s^n) = 2(J_{gus}^{+n} + J_{gus}^{-n}) \quad (3.14)$$

Com isso os coeficientes primários são assim obtidos:

$$c_{0gu}^n = \phi_g^n; \forall u = x, y, z \quad (3.15)$$

$$c_{1gu}^n = (J_{gur}^{+n} + J_{gur}^{-n}) - (J_{gul}^{+n} + J_{gul}^{-n}) \quad (3.16)$$

e

$$c_{2gu}^n = \phi_g^n - \left((J_{gur}^{+n} + J_{gur}^{-n}) + (J_{gul}^{+n} + J_{gul}^{-n}) \right) \quad (3.17)$$

3.2 - Coeficientes Secundários

Os coeficientes secundários (c_{3gu}^n e c_{4gu}^n) são obtidos através da técnica de resíduos ponderados aplicada à equação da difusão unidimensional, equação (3.9), e utilizando a expansão polinomial para $L_{gu}^n(u)$, equações (3.12).

Com isso, os coeficientes c_{3gu}^n e c_{4gu}^n podem ser calculados a partir da seguinte equação de resíduos ponderados:

$$\int_0^{a_u^n} w_i \left(\frac{u}{a_u^n} \right) \left\{ -\bar{D}_g^n \frac{d^2 \bar{\Psi}_{gu}^n(n)}{du^2} + \left[\Sigma_{ag}^n + \sum_{g'=1}^2 \bar{\Sigma}_{g'g}^n \right] \Psi_{gu}^n(u) - \frac{1}{K_{EFETIVO}} \chi_g \sum_{g'=1}^2 v \Sigma_{fg'}^n \Psi_{g'u}^n(n) - \sum_{g'=1}^2 \bar{\Sigma}_{gg'}^n \Psi_{g'u}^n(n) + L_{gu}^n(u) \right\} du = 0 \quad (3.18)$$

onde, por uma questão de precisão e eficiência, as funções peso $w_i \left(\frac{u}{a_u^n} \right)$ são escolhidas

como sendo $h_1 \left(\frac{u}{a_u^n} \right)$ e $h_2 \left(\frac{u}{a_u^n} \right)$ para calcular c_{3gu}^n e c_{4gu}^n , respectivamente.

Então, substituindo as equações (3.11) e (3.12) na equação (3.18) e observando que

$$\int_0^1 h_1(\xi) \frac{d^2 h_k(\xi)}{d\xi^2} d\xi = \begin{cases} -12 & \text{se } k = 3 \\ 0 & \text{se } k \neq 3 \end{cases} \quad (3.19)$$

$$\int_0^1 h_2(\xi) \frac{d^2 h_k(\xi)}{d\xi^2} d\xi = \begin{cases} 12 & \text{se } k = 4 \\ 0 & \text{se } k \neq 4 \end{cases} \quad (3.20)$$

$$\int_0^1 h_1(\xi) h_k(\xi) d\xi = \begin{cases} \frac{1}{3} & \text{se } k = 1 \\ 0 & \text{se } k = 2 \\ \frac{1}{5} & \text{se } k = 3 \\ 0 & \text{se } k = 4 \end{cases} \quad (3.21)$$

e

$$\int_0^1 h_2(\xi) h_k(\xi) d\xi = \begin{cases} 0 & \text{se } k = 1 \\ 1/5 & \text{se } k = 2 \\ 0 & \text{se } k = 3 \\ -3/35 & \text{se } k = 4 \end{cases} \quad (3.22)$$

obtem-se os seguintes sistemas de equações a partir dos quais os coeficientes secundários são, respectivamente, calculados:

$$\begin{aligned}
& \left\{ \frac{12\bar{D}_g^n}{(a_u^n)^2} + \frac{1}{5} \left[\Sigma_{ag}^n + \sum_{g'=1}^2 \bar{\Sigma}_{g'g}^n \right] \right\} c_{3gu}^n - \frac{1}{5} \sum_{g'=1}^2 \left\{ \frac{1}{K_{EFETIVO}} \chi_g \nu \Sigma_{fg'}^n + \bar{\Sigma}_{gg'}^n \right\} c_{3g'u}^n \\
& = -\frac{1}{3} \alpha_{1gu}^u \\
& - \frac{1}{3} \left\{ \left[\Sigma_{ag}^n + \sum_{g'=1}^2 \bar{\Sigma}_{g'g}^n \right] c_{1gu}^n \right. \\
& \left. - \sum_{g'=1}^2 \left[\frac{1}{K_{EFETIVO}} \chi_g \nu \Sigma_{fg'}^n + \bar{\Sigma}_{gg'}^n \right] c_{2g'u}^n \right\} \quad (3.23)
\end{aligned}$$

e

$$\begin{aligned}
& \left\{ \frac{12\bar{D}_g^n}{(a_u^n)^2} + \frac{3}{35} \left[\Sigma_{ag}^n + \sum_{g'=1}^2 \bar{\Sigma}_{g'g}^n \right] \right\} c_{4gu}^n - \frac{3}{35} \sum_{g'=1}^2 \left\{ \frac{1}{K_{EFETIVO}} \chi_g \nu \Sigma_{fg'}^n + \bar{\Sigma}_{gg'}^n \right\} c_{4g'u}^n \\
& = \frac{1}{5} \alpha_{2gu}^u \\
& - \frac{1}{5} \left\{ \left[\Sigma_{ag}^n + \sum_{g'=1}^2 \bar{\Sigma}_{g'g}^n \right] c_{2gu}^n \right. \\
& \left. - \sum_{g'=1}^2 \left[\frac{1}{K_{EFETIVO}} \chi_g \nu \Sigma_{fg'}^n + \bar{\Sigma}_{gg'}^n \right] c_{2g'u}^n \right\} \quad (3.24)
\end{aligned}$$

3.3 – Coeficientes da Fuga Transversal

Os coeficientes da expansão que representa a fuga transversal, equação (3.12), são obtidos do mesmo modo que os coeficientes primários da expansão de $\bar{\Psi}_{gu}^n(u)$, ou seja, com uma condição de consistência onde

$$\frac{1}{a_u^n} \int_0^{a_u^n} L_{gu}^n(u) du \equiv \bar{L}_g^u \quad (3.25)$$

sendo \bar{L}_{gu}^n a fuga transversal média a direção u , e com condições nas superfícies do nodo, quais sejam,

$$L_{gu}^n(u_s^n) = \sum_{k=0}^2 \alpha_{kgu}^n h_k\left(\frac{u_s^u}{a_u^n}\right) \equiv L_{gus}^n; \text{ para } s = r, 1 \quad (3.26)$$

Então, substituindo a equação (3.12) na equação (3.25) e fazendo uso da seguinte propriedade das funções de base do NEM:

$$\int_0^1 h_\xi(\xi) d\xi = 0; \quad \forall k \geq 1 \quad (3.27)$$

obtem-se que

$$\alpha_{0gu}^n = \bar{L}_{gu}^n \quad (3.28)$$

Substituindo as funções de base, para os dois valores de u_s^n , na equação (3.26) obtém-se um sistema de equações cuja solução resulta em

$$\alpha_{1gu}^n = \frac{1}{2} (L_{gur}^n - L_{gul}^n) \quad (3.29)$$

e

$$\alpha_{2gu}^n = \bar{L}_{gu}^n - \frac{1}{2} (L_{gur}^n - L_{gul}^n) \quad (3.30)$$

Para o cálculo dos termos L_{gus}^n as seguintes condições de continuidade são impostas à função $L_{gu}^n(n)$ e sua derivada, na interface entre dois nodos adjacentes (indicados por m e n):

$$L_{gul}^n = L_{gur}^m \quad (3.31)$$

e

$$\left. \frac{dL_{gu}^n(n)}{du} \right|_{u=u_l^n} = \left. \frac{dL_{gu}^m(u)}{du} \right|_{u=u_r^m} \quad (3.32)$$

Segundo o método NEM, as derivadas na equação (2.32) são aproximadas por diferenças finitas, resultando em

$$\frac{\bar{L}_{gu}^n - L_{gul}^n}{a_u^n/2} = \frac{L_{gur}^m - \bar{L}_{gu}^m}{a_u^m/2} \quad (3.33)$$

Então, com uso da equação (3.31) na equação (3.33), obtém-se que

$$L_{gur}^m = \frac{a_u^n \bar{L}_{gu}^m + a_u^m \bar{L}_{gu}^n}{a_u^n + a_u^m} \quad (3.34)$$

Para o cálculo das fugas transversais médias, \bar{L}_{gu}^n , a equação (3.10) é substituída na equação (3.25), são usadas as equações (3.2) e (3.4), é executada a integração e usando a definição de $\bar{J}_{gus}^{\pm n}$, dada pela equação (3.6), resulta em

$$\bar{L}_{gu}^n = \sum_{v=v,w} \frac{1}{a_v^n} \{ [\bar{J}_{gur}^{+n} - \bar{J}_{gur}^{-n}] - [\bar{J}_{gul}^{+n} - \bar{J}_{gul}^{-n}] \} \quad (3.35)$$

3.4 – Correntes Parciais de Saída

Neste momento, pode-se calcular todos os coeficientes da expansão de $\Psi_{gu}^n(u)$, e com isso, obter as correntes parciais de saída do nodo. Para isso, as equações (3.4) e (3.11) são substituídas na equação (3.2) e a equação resultante é escrita para os dois valores de u_s^n , o que resulta no seguinte sistema de equações:

$$\bar{J}_{gul}^{+n} - \bar{J}_{gul}^{-n} = -\bar{D}_g^n \{ 2c_{1gu}^n + 6c_{2gu}^n - 6c_{3gu}^n + 6c_{4gu}^n \} \quad (3.36)$$

e

$$\bar{J}_{gur}^{+n} - \bar{J}_{gur}^{-n} = -\bar{D}_g^n \{ 2c_{1gu}^n - 6c_{2gu}^n - 6c_{3gu}^n - 6c_{4gu}^n \} \quad (3.37)$$

Usando a equação (3.14) nas equações (3.16) e (3.17) e substituindo as equações resultantes nas equações (3.36) e (3.37), obtém-se um sistema de equações para as correntes parciais, do qual resultam as correntes parciais de saída do nodo, na seguinte forma:

$$\bar{J}_{gul}^{-n} = A_{0gul}^n \{ \bar{\Phi}_g^n + c_{4gu}^n \} + A_{1gul}^n \bar{J}_{gul}^{+n} + A_{2gu}^n \bar{J}_{gur}^{-n} - A_{3gul}^n c_{3gu}^n \quad (3.38)$$

e

$$\bar{J}_{gur}^{+n} = A_{0gur}^n \{ \bar{\Phi}_g^n + c_{4gu}^n \} + A_{2gur}^n \bar{J}_{gul}^{+n} + A_{1gur}^n \bar{J}_{gur}^{-n} + A_{3gur}^n c_{3gu}^n \quad (3.39)$$

onde:

$$A_{0gu}^n = \frac{6(D_g^n/a_u^n)}{(1 + 12(D_g^n/a_u^n))} \quad (3.40)$$

$$A_{1gu}^n = \frac{(1 - 48(D_g^n/a_u^n)^2)}{\left(\left(1 + 12(D_g^n/a_u^n)\right)\left(1 + 4(D_g^n/a_u^n)\right)\right)} \quad (3.41)$$

$$A_{2gu}^n = \frac{-8(D_g^n/a_u^n)}{\left(\left(1 + 12(D_g^n/a_u^n)\right)\left(1 + 4(D_g^n/a_u^n)\right)\right)} \quad (3.42)$$

e

$$A_{3gu}^n = \frac{6(D_g^n/a_u^n)}{(1 + 4(D_g^n/a_u^n))} \quad (3.43)$$

3.5 - Equação de Balanço Nodal

Substituindo as equações (3.38) e (3.39) na equação de balanço nodal, equação (3.3), obtém-se a equação da qual o fluxo médio nodal é obtido.

$$\begin{aligned} & \left(\sum_{u=x,y,z} \frac{2A_{0gu}^n}{a_u^n} + \Sigma_{Rg}^n \right) \phi_g^n \\ &= \frac{1}{K_{EFETIVO}} \chi_g \sum_{g'=1}^2 \nu \Sigma_{fg'}^n \phi_{g'}^n \\ &+ \sum_{g'=1}^2 \Sigma_{gg'}^n \phi_{g'}^n + \sum_{u=x,y,z} 2A_{0gu}^n (2(J_{gur}^{-n} + J_{gui}^{+n}) - c_{4gu}^n) / a_u^n \quad (3.44) \end{aligned}$$

Com as condições de contorno e com as equações descritas acima pode-se calcular, em um processo iterativo de cálculo (Alvin, A. C. M., 2010) os fluxos médios no nodo e o fator de multiplicação para o núcleo de um reator nuclear utilizando a equação (3.44).

4 O PROBLEMA DA CONCENTRAÇÃO CRÍTICA DE BORO

O problema de se encontrar a concentração crítica de boro está no fato de que atualmente é utilizado para este fim um processo iterativo, que geralmente tem alto custo computacional. Tais problemas são aumentados em complexidade, pois durante este processo ocorre o cálculo do fluxo de nêutrons e do $K_{EFETIVO}$ que também são processos iterativos, como descrito no capítulo 1 (Pereira Neto, W. F., 2011), o que aumenta ainda mais o tempo computacional.

Agora descreveremos um novo método de como calcular a concentração crítica de boro baseado em redes neurais artificiais, e com isso evitar um dos processos iterativos, portanto diminuindo o tempo de cálculo.

A questão de determinar a concentração crítica de boro usando uma rede neural é saber quais informações são relevantes e como passar estas para a rede. Para isso, primeiramente vamos analisar quais dos dados passados para o código de física de reatores, utilizado neste trabalho, são relevantes para uma rede. O programa usado é o código NEM (Silva, F.C. e Martinez, A. S., 2003) que é baseado no Método de Expansão Nodal (NEM), do inglês Nodal Expansion Method (Finnean et al., 1977).

4.1 – As Informações Necessárias para a Criação das Redes

A criação das redes passa pelos dados da configuração do reator, o número total de regiões na direção X, o número total de regiões na direção Y e os parâmetros nucleares. A configuração do reator pode ser expressa na forma de uma matriz que contem os diversos tipos de Elementos Combustíveis (ECs) e estes diversos tipos de ECs são identificados por números de 1 até o número total de tipos existente no reator. Cada tipo de EC tem os seus parâmetros nucleares associados ao número do tipo, que são compostos pela seção de choque de absorção Σ_a , pela seção de choque de fissão vezes o número médio de nêutrons emitidos na fissão $\nu\Sigma_f$, pelo coeficiente de difusão D, pela seção de choque de espalhamento Σ_s e pela seção de choque de fissão vezes a energia liberada por fissão $\omega\Sigma_f$. Todos esses parâmetros são para dois grupos de energia.

Os ECs são distribuídos no núcleo do reator de maneira a obedecer a simetrias de $\frac{1}{2}$, $\frac{1}{4}$ e $\frac{1}{8}$ do núcleo do reator. Esta simetria é usada para facilitar o cálculo do fluxo de nêutrons, pois com isso, é necessário somente o uso da metade dos ECs no cálculo do fluxo, no caso de simetria de $\frac{1}{2}$, um quarto dos ECs no caso de simetria de $\frac{1}{4}$ e um oitavo no caso da simetria de $\frac{1}{8}$. A figura 4.1 é uma visão de cima do núcleo do reator onde colocamos os eixos de simetria de $\frac{1}{2}$ (eixo x ou eixo y), simetria de $\frac{1}{4}$ (eixo x e eixo y) e de simetria de $\frac{1}{8}$ (eixo x e eixo y e eixo B13 ou eixo x eixo y e eixo B24).

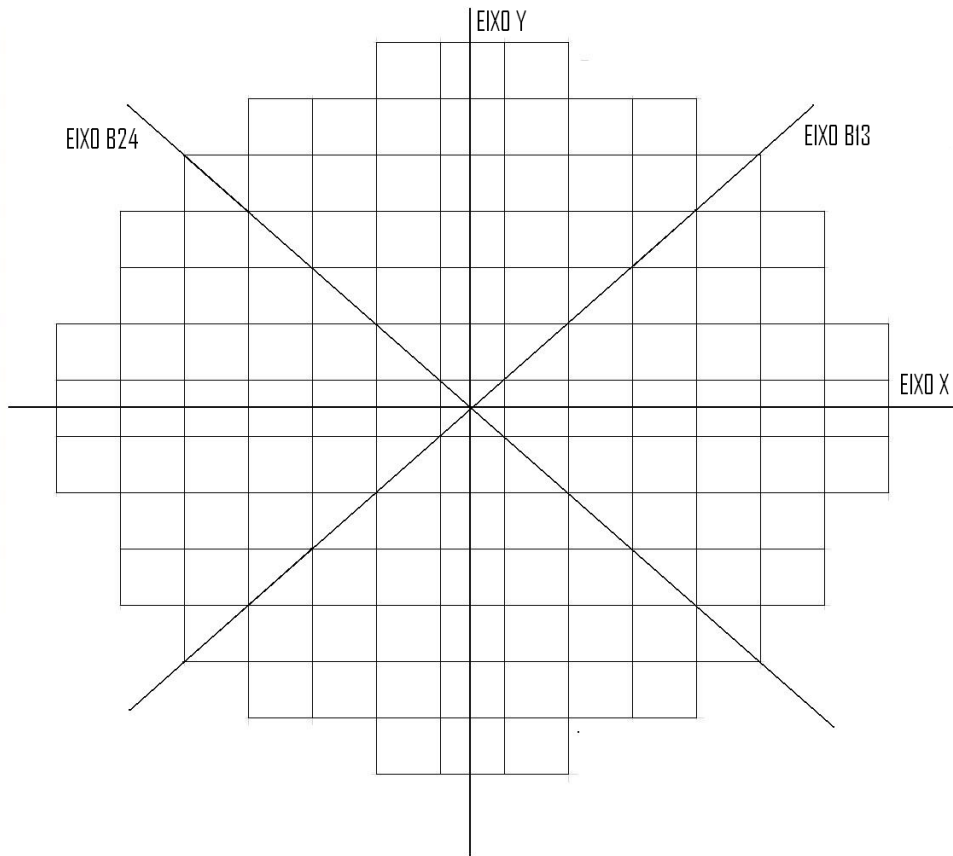


FIGURA 4.1 - Uma Vista de Cima de um Núcleo e os Eixos de Simetria

Os ECs que apresentam as mesmas características são numerados com o mesmo número, ou seja, apresentam o mesmo enriquecimento, o mesmo ciclo e outros fatores comuns. Estes ECs com mesmo número são chamados de mesmo tipo de ECs. Estes ECs são colocados na sua posição obedecendo às simetrias desejadas, as posições possíveis dos ECs são as mostradas na figura 4.1, ou seja, cada quadrado representa uma posição.

O perfil do fluxo de nêutrons depende da posição dos diversos tipos de ECs, para o fluxo no núcleo do reator, o perfil deve ser uniforme, de modo a evitar fatores de

picos. Em geral, para se alcançar este perfil uniforme há necessidade de uma distribuição também uniforme dos ECs de maneira a não ocorrer acúmulo de ECs com alto ou baixo enriquecimento.

4.2 – Codificação da Entrada e da Saída das Redes Neurais

Agora discutiremos como foram passadas as informações para rede levando em conta que as posições dos ECs são relevantes para os cálculos. A configuração do núcleo é passada para o código em forma de matriz que é a CONFIG mostrada na tabela 4.1

TABELA 4.1 - - A configuração do Reator IAEA com simetria de 1/8

3	2	2	2	3	2	2	1	4
0	2	2	2	2	2	2	1	4
0	0	2	2	2	2	1	1	4
0	0	0	2	2	2	1	4	4
0	0	0	0	3	1	1	4	0
0	0	0	0	0	1	4	4	0
0	0	0	0	0	0	4	0	0

As posições numeradas com zeros correspondem àquelas nas quais não existe EC e as outras posições com números diferentes de zeros representam os diversos tipos de ECs da configuração. A idéia central foi fazer uma varredura da matriz CONFIG, e através da varredura coletar os parâmetros nucleares associados ao tipo de EC correspondente e armazená-los em forma de um vetor. Os vetores são armazenados em forma de colunas em matriz. O número total de elementos dos vetores colunas corresponde o número total de neurônios da camada de entrada das redes utilizadas e pode ser calculado como

$$M = NTRX \cdot NTRY \cdot NUMTIP \cdot 2 + 1 \quad (4.1)$$

onde NTRX é o número de regiões na direção x, e NTRY é o número de regiões na direção y, e NUMTIP é o numero total de tipos existente dos ECs na configuração, já o fator 2 é devido a existência de dois grupos de energia. Portanto o número de colunas da matriz corresponde ao número total do conjunto de treinamento.

Podemos perceber, comparando a configuração do reator IAEA (Argonne National Laboratory, 1977 e Guimarães, C. de S., da Silva, F. C., Martinez, A. S., 2007) na tabela 4.1 com a configuração do reator PWR (Pereira Neto, W. F., 2011) na tabela 4.2, que ambos apresentavam quantidades diferentes de regiões nas direções X e Y, e, além disso, o número total de tipos existentes são também diferentes.

TABELA 4.2 - Configuração do Reator PWR com Simetria de 1/4

1	7	2	5	1	6	1	3	8
7	1	7	2	7	1	1	3	8
2	7	1	7	2	6	1	3	8
5	2	7	2	7	1	7	3	8
1	7	2	7	2	4	3	8	8
6	1	6	1	4	3	3	8	0
1	1	1	7	3	3	8	8	0
3	3	3	8	8	8	0	0	0
8	8	8	8	8	0	0	0	0

No caso do reator IAEA temos 9 regiões na direção X e 7 regiões na direção Y e um total de 5 tipos de ECs, já no caso do reator PWR temos 9 regiões na direção X e 9 regiões na direção Y com 8 tipos de ECs, o que faz o número de elementos dos vetores da configuração do reator IAEA e do reator PWR serem diferentes de acordo com a equação (4.1), isto é, para o reator IAEA temos $M = 9 \times 7 \times 5 \times 2 + 1 = 631$ neurônios na camada de entrada, já o reator PWR temos $M = 9 \times 9 \times 8 \times 2 + 1 = 1297$ neurônios na camada de entrada, com isso, não é possível utilizar essas duas configurações em um mesmo conjunto de treinamento.

Para se resolver a questão de ter dados para treinar uma rede, utilizamos um determinado reator e sua configuração, e desta forma garantimos um número fixo de regiões nas direções X e Y e o mesmo número total de tipos existentes de ECs, e para esta configuração calculamos vários $K_{EFETTIVO}$ para diversas concentrações de boro. Mas de qualquer maneira não haveria variedade de informações suficientes para garantir que a rede generalizaria o conhecimento dos dados. Para garantir uma maior variedade de informações à rede, pensamos inicialmente em variar as posições dos diversos tipos de ECs existentes no reator, contudo esta solução ainda apresentava algumas limitações, como ter que manter o número de tipos existentes dos ECs fixo e garantir um fluxo de nêutrons uniforme de forma a evitar fatores de pico, pois o acúmulo de ECs mais

enriquecido na periferia do núcleo e uma maior quantidade de ECs menos enriquecidos no centro do núcleo ou vice-versa pode levar a uma distribuição bastante irregular do fluxo de nêutrons.

A solução do problema de obter os dados para o treinamento e garantir a variedade das informações para as redes, foi a utilização de diversos arquivos dos parâmetros nucleares associados à mesma configuração do núcleo, estes diversos arquivos dos parâmetros nucleares apresentam pequenas variações dos parâmetros nucleares originais.

Com base nos parâmetros nucleares da tabela 4.3 foram feitas pequenas variações em seus valores, como consequência foram gerados novos arquivos dos parâmetros nucleares, que por sua vez foram utilizados para o treinamento da rede. Essas variações ocorreram no sentido de aumentar e diminuir os valores dos parâmetros nucleares. Essas variações garantiram a diversidade nas informações para os dados de treinamento das redes.

TABELA 4.3 - Parâmetros Nucleares do Reator IAEA – Dados Originais

Tipo	Grupo	Σ_a	$\nu\Sigma_f$	D	Σ_s	$\omega\Sigma_f$
1	1	0,0100	0,0000	1,5000	0,0200	0,0000
	2	0,0800	0,1350	0,4000	0,0000	0,1350
2	1	0,0100	0,0000	1,5000	0,0200	0,0000
	2	0,0850	0,1350	0,4000	0,0000	0,1350
3	1	0,0100	0,0000	1,5000	0,0200	0,0000
	2	0,1300	0,1350	0,4000	0,0000	0,1350
4	1	0,0000	0,0000	2,0000	0,0400	0,0000
	2	0,0100	0,0000	0,3000	0,0000	0,0000
5	1	0,0000	0,0000	2,0000	0,0400	0,0000
	2	0,0550	0,0000	0,3000	0,0000	0,0000

As tabelas 4.4 e 4.5 apresentam os parâmetros nucleares que foram obtidos dos parâmetros nucleares originais do reator IAEA mostrados na tabela 4.3, utilizando-se para isso de um conjunto de equações empíricas que produziu essas variações nestes valores de acordo com um fator de divisão. As equações empíricas são:

$$\Sigma_a = \Sigma_a - \frac{\Sigma_a}{V} \quad \text{ou} \quad \Sigma_a = \Sigma_a + \frac{\Sigma_a}{V} \quad (4.2)$$

$$v\Sigma_f = v\Sigma_f + \frac{\Sigma_a}{V} \text{ ou } v\Sigma_f = v\Sigma_f - \frac{v\Sigma_f}{V} \quad (4.3)$$

$$D = D + \frac{D}{V} \text{ ou } D = D - \frac{D}{V} \quad (4.4)$$

$$\Sigma_s = \Sigma_s - \frac{\Sigma_s}{V} \text{ ou } \Sigma_s = \Sigma_s + \frac{\Sigma_s}{V} \quad (4.5)$$

$$\omega\Sigma_f = \omega\Sigma_f - \frac{\Sigma_a}{V} \text{ ou } \omega\Sigma_f = \omega\Sigma_f - \frac{v\Sigma_f}{V} \quad (4.6)$$

onde V é o fator de divisão

TABELA 4.4 - Parâmetros Nucleares do Reator IAEA – Gerados – Lado Direito – V =2

Tipo	Grupo	Σ_a	$v\Sigma_f$	D	Σ_s	$\omega\Sigma_f$
1	1	0,0050	0,0050	2,2500	0,0100	0,0050
	2	0,0400	0,1750	0,6000	0,0000	0,1750
2	1	0,0050	0,0050	2,2500	0,0100	0,0050
	2	0,0425	0,1775	0,6000	0,0000	0,1775
3	1	0,0050	0,0050	2,2500	0,0100	0,0050
	2	0,0650	0,2000	0,6000	0,0000	0,2000
4	1	0,0000	0,0000	3,0000	0,0200	0,0000
	2	0,0050	0,0050	0,4500	0,0000	0,0050
5	1	0,0000	0,0000	3,0000	0,0200	0,0000
	2	0,0275	0,0275	0,4500	0,0000	0,0275

TABELA 4.5 - Parâmetros Nucleares do Reator IAEA – Gerados – Lado Esquerdo – V =2

Tipo	Grupo	Σ_a	$v\Sigma_f$	D	Σ_s	$\omega\Sigma_f$
1	1	0,0150	0,0000	0,7500	0,0300	0,0000
	2	0,1200	0,0675	0,2000	0,0000	0,0675
2	1	0,0150	0,0000	0,7500	0,0300	0,0000
	2	0,1275	0,0675	0,2000	0,0000	0,0675
3	1	0,0150	0,0000	0,7500	0,0300	0,0000
	2	0,1950	0,0675	0,2000	0,0000	0,0675
4	1	0,0000	0,0000	1,0000	0,0600	0,0000
	2	0,0150	0,0000	0,1500	0,0000	0,0000
5	1	0,0000	0,0000	1,0000	0,0600	0,0000
	2	0,0825	0,0000	0,1500	0,0000	0,0000

Na construção da tabela 4.4 foram utilizadas as equações (4.2) a (4.6) do lado direito com fator de divisão igual 2, já tabela 4.5 utilizou as equações do lado esquerdo com o mesmo fator. Além dos dados das tabelas 4.4 e 4.5 foram geradas mais 14 tabelas de dados nucleares, sendo 7 para as equações do lado direito e 7 para as do lado esquerdo, com os fatores 4, 8, 16, 32, 64, 128 e 256.

Obtido o conjunto dos parâmetros nucleares como descrito acima, iniciamos o procedimento para o cálculo do $K_{EFETTIVO}$, utilizando o código NEM, da seguinte maneira: Para cada um dos arquivos dos parâmetros nucleares calculamos o $K_{EFETTIVO}$ para diferentes concentrações de boro desde QIB (Quantidade Inicial de Boro) até 2000 ppm, de 50 ppm em 50 ppm, usando a configuração da tabela 4.1 para todos os arquivos dos parâmetros nucleares.

Para cada um dos arquivos dos parâmetros nucleares tivemos uma QIB diferente, deste modo gerando uma maior quantidade de dados para a montagem dos dados de treinamento das redes. No caso dos parâmetros nucleares originais podemos ver através da tabela 4.6 que QIB = 0.

TABELA 4.6 - Concentração de Boro e K EFETIVO no Caso dos Parâmetros Nucleares Originais

CB	K EFETIVO	CB	K EFETIVO	CB	K EFETIVO	CB	K EFETIVO
0	1,02897650	550	0,92502869	1100	0,84020837	1650	0,76908680
50	1,01847270	600	0,91663770	1150	0,83323401	1700	0,76318155
100	1,00821990	650	0,90839509	1200	0,82636886	1750	0,75736076
150	0,99819973	700	0,90029624	1250	0,81961027	1800	0,75162262
200	0,98839729	750	0,89233682	1300	0,81295565	1850	0,74596533
250	0,97880023	800	0,88451275	1350	0,80640252	1900	0,74038718
300	0,96939807	850	0,87682020	1400	0,79994849	1950	0,73488650
350	0,96018179	900	0,86925552	1450	0,79359126	2000	0,72946166
400	0,95114346	950	0,86181523	1500	0,78732860		
450	0,94227598	1000	0,85449602	1550	0,78115834		
500	0,93357300	1050	0,84729475	1600	0,77507842		

A Figura 4.2 mostra como varia o $K_{EFETTIVO}$ em função da concentração de boro. O gráfico foi construído com os dados da tabela 4.6. Com os arquivos dos parâmetros nucleares gerados e seus respectivos dados das concentrações de boro e os $K_{EFETTIVOS}$ associados, podemos montar os dados de treinamento para a rede. Isto foi

feito utilizando a configuração do núcleo que contém os diversos tipos de ECs conforme a tabela 4.1.

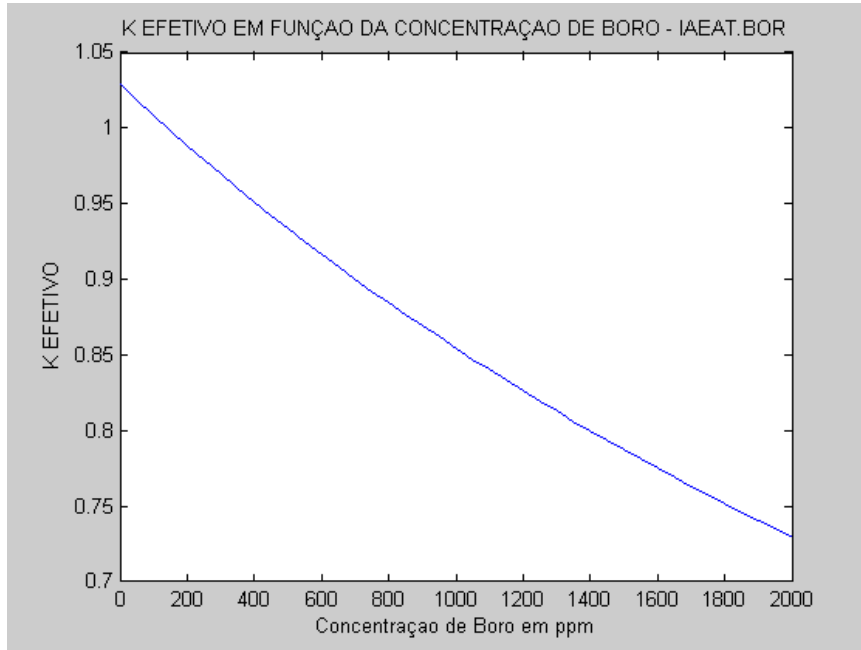


FIGURA 4.2 - K EFETIVO em Função da Concentração de Boro

4.2.1 - A Entrada e a Saída das Redes Neurais

A camada de entrada e a camada de saída de todas as redes utilizadas nesta dissertação são apresentadas na figura 4.3.

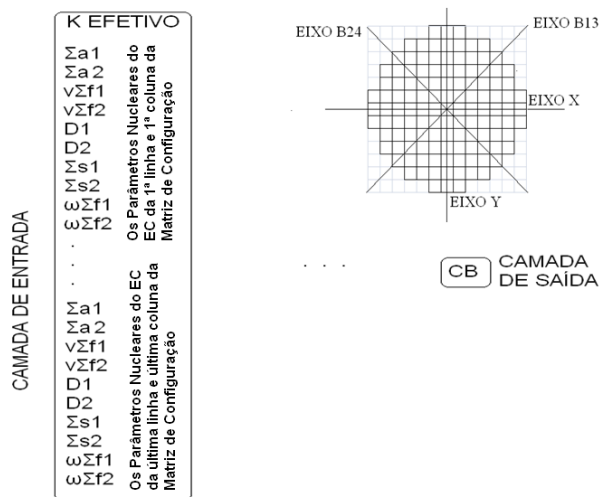


FIGURA 4.3 – Camada de Entrada e Saída das Redes junto com o Esquema das posições dos ECs

A camada de entrada é formada pelo $K_{EFETIVO}$, os parâmetros nucleares do EC da 1ª linha e 1ª coluna da matriz de configuração, os parâmetros nucleares do EC da 1ª linha e 2ª coluna da matriz de configuração até os parâmetros nucleares do EC da 1ª linha e última coluna da matriz de configuração, portanto colocamos os parâmetros nucleares dos ECs da matriz de configuração linha a linha até os parâmetros nucleares do EC da última linha e última coluna da matriz de configuração; já a camada de saída é formada somente pela concentração de boro, isto é, um único neurônio na camada de saída. O número de neurônios da camada de entrada depende do número de regiões na direção X (NTRX), do número de regiões na direção Y (NTRY) e do número total de tipos existentes (NUMTIP), que no caso da configuração do reator IAEA (Argonne National Laboratory, 1977 e Guimarães, C. de S., da Silva, F. C., Martinez, A. S., 2007) são NTRX = 9, NTRY = 7 e NUMTIP = 5, o que nos dá $M = 631$ neurônios de acordo com a equação (4.1).

A sub-rotina MONTATREINA4 em MATLAB que faz a montagem da matriz dos dados utilizados para o treinamento das redes em multicamadas e da rede de base radial se encontra no anexo na seção 8.3. A montagem dos dados de treinamento é baseada na posição dos diversos tipos de ECs e nos parâmetros nucleares associados a estes ECs. A sub-rotina que faz alterações na seção de choque de absorção de acordo com a concentração de boro está no anexo na seção 8.4

Na rede de base radial o número de neurônios da camada oculta inicia em 25 neurônios e sofre acréscimo de 25 neurônios em 25 neurônios até que a rede esteja treinada. Nas redes multicamadas o número de camadas ocultas e o número de neurônios destas foram alterados, manualmente, em diversas combinações para o treinamento, ou seja, foram construídas varias arquiteturas

4.3 - As Fases da Obtenção das Redes

A primeira fase é a criação dos novos arquivos dos parâmetros nucleares da configuração do reator do qual se deseja obter a concentração crítica de boro. Estes novos arquivos dos parâmetros nucleares são criados a partir do arquivo dos parâmetros nucleares originais utilizando para isto às equações empíricas (4.2) a (4.6).

A segunda fase é utilizar o código NEM para calcular o K_{EFETIVO} para todos os novos os arquivos dos parâmetros nucleares obtidos na primeira fase. Cada um dos novos arquivos dos parâmetros nucleares foi utilizado para calcular o K_{EFETIVO} em diferentes concentrações de boro, gerando, com isso, variedades de informações suficientes para o treinamento das redes. O código NEM é utilizado para calcular os K_{EFETIVOS} do arquivo dos parâmetros nucleares originais em diferentes concentrações de boro, estes dados obtidos do código NEM não serão utilizados no treinamento das redes, mas sim utilizados para verificação da capacidade da rede generalizar as informações contidas nos dados de treinamento.

A terceira fase é a utilização dos novos parâmetros nucleares, dos K_{EFETIVOS} e das correspondentes concentrações de boro no treinamento das redes. Com as redes já treinadas utilizam-se os parâmetros nucleares originais, os K_{EFETIVOS} e as respectivas concentrações de boro para verificar se o treinamento foi bem sucedido.

A quarta fase é a utilização da rede, já treinada, na obtenção da concentração crítica de boro, isto é feito com o $K_{\text{EFETIVO}} = 1$ e os parâmetros nucleares originais como entrada da rede. Nesta fase é gerado um arquivo com os parâmetros nucleares originais e com a concentração crítica de boro, para ser utilizado no código NEM.

4.4 - A Utilização das Redes Treinadas

A figura 4.4 abaixo apresenta o fluxograma do funcionamento de uma rede treinada na obtenção da concentração crítica de boro, como podemos observar não ocorre processo iterativo, e deste modo há uma diminuição no tempo de cálculo da concentração crítica de boro. Este fluxograma representa um programa em MATLAB que gera um arquivo com os parâmetros nucleares e com a concentração crítica de boro obtida. Estes parâmetros nucleares com a concentração crítica de boro gerada pelo programa descrito anteriormente são utilizados no Código NEM para obter o fluxo de nêutrons e o K_{EFETIVO} .

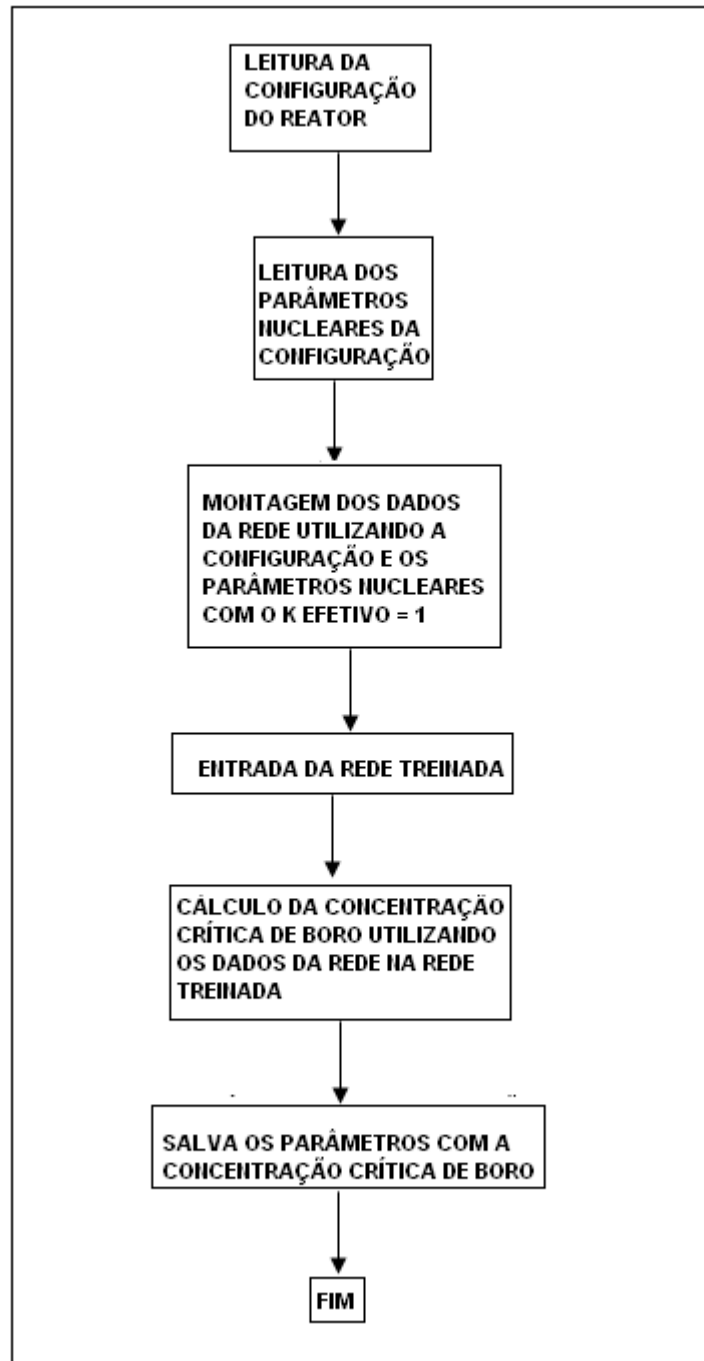


FIGURA 4.4 - Fluxograma do Programa MATLAB que Calcula a Concentração Crítica de Boro

5 RESULTADOS OBTIDOS

Os dados treinamentos são geralmente divididos em três grupos que são: dados de treinamento propriamente ditos (que são usados para corrigir os pesos dos neurônios durante o treinamento até que a rede aprenda), os dados de teste (que são utilizados para testar a rede a cada passagem completa do conjunto de dados de treinamento, em caso dos dados de teste apresentarem erro grande em comparação com os erros dos dados de treinamento ocorrem alguns ajustes nos pesos dos neurônios) e os dados de produção (ou validação, conjunto de dados nunca visto, nem nos treinamentos e nem nos testes, utilizado para verificar os resultados da rede após o treinamento terminado).

Os resultados obtidos com os parâmetros nucleares gerados foram utilizados para o treinamento de todas as redes apresentadas neste trabalho, enquanto que os resultados obtidos com os parâmetros nucleares originais serviram como um conjunto de dados para a validação do sucesso ou fracasso das redes treinadas, já que estes resultados não foram utilizados no treinamento das redes em nenhum momento e, portanto não influenciando no treinamento das mesmas. Isto garante no caso de sucesso das redes treinadas com os dados de validação, que a rede generalizou as informações contidas nos dados de treinamento. Os dados de validação são compostos pelos parâmetros nucleares originais em diferentes CBs e os respectivos $K_{EFETIVOS}$ a elas associadas. Todas as tabelas apresentadas neste capítulo foram obtidas com os dados de produção.

5.1 – Resultados da Rede de Base Radial

No caso da rede de base radial (Função NEWRB do Neural Network Toolbox versão 5.1 MATLAB R2007b) foram necessários 675 neurônios na camada oculta para a rede convergir. Os parâmetros da rede de base radial são:

Camada de Entrada: 631 neurônios (conforme descrito em 4.2.1)

Camada de Saída: 1 neurônio (Concentração de Boro)

Camada Oculta: 675 Neurônios (Quando terminou o treinamento)

Função de Ativação da Camada Oculta: Função Gaussiana

Função de Ativação da Camada de Saída: Função Linear

Critério de Parada: Soma do Erro Quadrático

Tolerância (Goal) = 0

A rede de base radial terminou o treinamento quando a soma do erro quadrático ficou praticamente constante. Para os dados de produção, a rede apresentou os resultados contidos na tabela 5.1.

TABELA 5.1 - Rede de Base Radial – Erro médio = 0,063 ppm

CB real	CB rede	ERRO	CB real	CB rede	ERRO	CB real	CB rede	ERRO
0	-0,10702	0,10702	700	700,10700	0,10660	1400	1399,97000	0,03404
50	49,97680	0,02317	750	750,11600	0,11580	1450	1449,95000	0,05263
100	100,01500	0,01538	800	800,11800	0,11770	1500	1499,94000	0,05627
150	150,03000	0,03022	850	850,10700	0,10685	1550	1549,95000	0,04655
200	200,03200	0,03200	900	900,08700	0,08738	1600	1599,95000	0,05022
250	250,03000	0,02993	950	950,07100	0,07102	1650	1649,93000	0,06632
300	300,03200	0,03249	1000	1000,07000	0,06629	1700	1699,94000	0,06203
350	350,04400	0,04362	1050	1050,07000	0,06874	1750	1749,96000	0,04302
400	400,06100	0,06097	1100	1100,06000	0,06299	1800	1799,96000	0,04221
450	450,07800	0,07849	1150	1150,04000	0,03790	1850	1849,96000	0,04215
500	500,09000	0,08994	1200	1200,00000	0,00229	1900	1899,95000	0,05025
550	550,09400	0,09406	1250	1249,98000	0,02221	1950	1950,04000	0,04156
600	600,09500	0,09483	1300	1299,97000	0,02568	2000	1999,74000	0,25630
650	650,09800	0,09822	1350	1349,98000	0,02302			

Conforme podemos observar na tabela 5.1 dos resultados obtidos pela rede de base radial com os dados de produção, os erros existentes entre CB real e os valores obtidos pela rede são pequenos conforme mostra o erro médio de 0,063 ppm.

5.2 – Resultados das Redes Multicamadas

Para as redes multicamadas (Função NEWFF do Neural Network Toolbox versão 5.1 MATLAB R2007b) foram utilizadas redes desde 1 até 10 camadas ocultas com o número de neurônios fixos (em 1, 3, 5, 7, 9,10 ou 15 neurônios) nas camadas ocultas. Isto se deve ao fato de que o número de neurônios da camada de entrada é definido pelo número dos dados de entrada do problema a ser resolvido e o número de

neurônios da camada de saída é definido pelo número de dados da camada de saída. Os parâmetros de treinamento das redes multicamadas são:

Camada de Entrada: 631 neurônios (conforme descrito em 4.2.1)

Camada de Saída: 1 neurônio (Concentração de Boro)

Função de Ativação das Camadas Ocultas: Tangente Hiperbólica

Função de Ativação da Camada de Saída: Função Linear

Critério de Parada: Erro Quadrático Médio

Tolerância (Goal) = 10^{-6}

As redes multicamadas treinadas apresentaram o seguinte resultado para os dados de produção quando aumentamos o número de camadas ocultas e mantivemos fixo o número de neurônios da camada oculta. À medida que aumentamos o número de camadas há uma melhora dos resultados, porém aumentando ainda mais o número de camadas ocultas o resultado tende a piorar, como mostram as tabelas 5.2, 5.3 e 5.4.

TABELA 5.2 - Rede Multicamada com 1 Camada Oculta de 5 Neurônios - Erro médio = 1,596 ppm

CB real	CB rede	ERRO	CB real	CB rede	ERRO	CB real	CB rede	ERRO
0	5,492860	5,492860	700	701,11400	1,113510	1400	1400,26000	0,255784
50	49,866300	0,133737	750	750,61700	0,617221	1450	1450,55000	0,548412
100	96,411500	3,588520	800	800,10300	0,102814	1500	1500,82000	0,819301
150	144,779000	5,220600	850	849,62900	0,370542	1550	1551,06000	1,060800
200	194,551000	5,448770	900	899,23800	0,761803	1600	1601,27000	1,267470
250	245,287000	4,713410	950	948,95300	1,047110	1650	1651,44000	1,436490
300	296,569000	3,431380	1000	998,78300	1,217260	1700	1701,57000	1,566990
350	348,040000	1,960000	1050	1048,73000	1,274480	1750	1751,66000	1,660330
400	399,427000	0,573380	1100	1098,77000	1,228800	1800	1801,72000	1,719510
450	450,547000	0,547031	1150	1148,90000	1,095550	1850	1851,75000	1,749450
500	501,308000	1,308300	1200	1199,11000	0,892476	1900	1901,76000	1,756240
550	551,693000	1,692720	1250	1249,36000	0,638383	1950	1951,75000	1,747270
600	601,738000	1,737960	1300	1299,65000	0,351405	2000	2001,73000	1,730940
650	651,516000	1,515930	1350	1349,95000	0,048381			

TABELA 5.3 - Rede Multicamada com 2 Camadas Ocultas de 5 Neurônios - Erro médio = 0,517 ppm

CB real	CB rede	ERRO	CB real	CB rede	ERRO	CB real	CB rede	ERRO
0	1,864000	1,864000	700	699,59000	0,410071	1400	1399,89000	0,109183
50	45,984000	4,016000	750	749,70200	0,297750	1450	1449,86000	0,141105
100	98,257700	1,742280	800	799,78600	0,213518	1500	1499,82000	0,180982
150	148,136000	1,864390	850	849,81500	0,184648	1550	1549,78000	0,217762
200	198,659000	1,340890	900	899,79800	0,201798	1600	1599,76000	0,240487
250	249,258000	0,741810	950	949,76200	0,238262	1650	1649,76000	0,240750
300	299,277000	0,722886	1000	999,73200	0,267987	1700	1699,78000	0,215063
350	349,170000	0,829818	1050	1049,72000	0,275185	1750	1749,83000	0,166950
400	399,310000	0,690088	1100	1099,74000	0,256308	1800	1799,89000	0,108644
450	449,570000	0,430464	1150	1149,78000	0,217530	1850	1849,94000	0,062293
500	499,695000	0,304919	1200	1199,83000	0,170485	1900	1899,94000	0,060665
550	549,639000	0,361012	1250	1249,87000	0,127968	1950	1949,85000	0,147338
600	599,536000	0,463782	1300	1299,90000	0,100495	2000	1999,62000	0,376424
650	649,514000	0,485789	1350	1349,91000	0,094061			

TABELA 5.4 - Rede Multicamada com 3 Camadas Ocultas de 5 Neurônios - Erro médio = 8,891 ppm

CB real	CB rede	ERRO	CB real	CB rede	ERRO	CB real	CB rede	ERRO
0	55,075400	55,075400	700	701,50200	1,502450	1400	1405,19000	5,190810
50	68,705400	18,705400	750	746,54200	3,457620	1450	1450,81000	0,809570
100	94,565700	5,434320	800	792,31600	7,684020	1500	1496,25000	3,747620
150	131,365000	18,635000	850	839,46600	10,533500	1550	1542,34000	7,656060
200	177,214000	22,786300	900	888,39100	11,609200	1600	1594,93000	5,070720
250	229,754000	20,245600	950	939,20200	10,798400	1650	1660,40000	10,396300
300	286,409000	13,590900	1000	991,72300	8,276750	1700	1707,34000	7,343310
350	344,702000	5,298100	1050	1045,52000	4,477480	1750	1748,74000	1,258040
400	402,565000	2,565100	1100	1099,97000	0,028334	1800	1797,05000	2,949170
450	458,541000	8,541010	1150	1154,34000	4,337500	1850	1850,16000	0,163840
500	511,845000	11,844600	1200	1207,89000	7,886590	1900	1903,08000	3,081720
550	562,310000	12,309500	1250	1260,00000	9,995000	1950	1950,87000	0,871600
600	610,264000	10,263600	1300	1310,25000	10,253200	2000	2004,93000	4,930910
650	656,376000	6,375960	1350	1358,56000	8,559220			

No caso de fixarmos o número de camadas e variarmos o número de neurônios podemos observar que, em alguns casos, à medida que aumentamos o número de neurônios das camadas ocultas, não ocorre uma melhora dos resultados, mas sim uma tendência a piorar, como mostram as tabelas 5.3, 5.5 e 5.6.

TABELA 5.5 - Rede Multicamada com 2 Camadas de 10 Neurônios - Erro médio = 2,030 ppm

CB real	CB rede	ERRO	CB real	CB rede	ERRO	CB real	CB rede	ERRO
0	7,244070	7,244070	700	699,35100	0,649394	1400	1399,26000	0,73934
50	51,763200	1,763180	750	749,83400	0,165835	1450	1448,71000	1,28796
100	98,056400	1,943560	800	800,35300	0,353044	1500	1498,32000	1,67910
150	145,807000	4,193440	850	850,89400	0,893524	1550	1548,13000	1,87444
200	194,692000	5,307730	900	901,42100	1,420750	1600	1598,14000	1,85807
250	244,406000	5,593770	950	951,88600	1,885700	1650	1648,36000	1,64056
300	294,671000	5,328830	1000	1002,24000	2,235100	1700	1698,74000	1,26217
350	345,253000	4,747220	1050	1052,42000	2,422070	1750	1749,21000	0,79466
400	395,969000	4,031160	1100	1102,42000	2,415300	1800	1799,66000	0,34225
450	446,693000	3,306550	1150	1152,20000	2,204880	1850	1849,96000	0,04121
500	497,355000	2,644780	1200	1201,80000	1,804750	1900	1899,94000	0,05836
550	547,931000	2,069330	1250	1251,25000	1,251100	1950	1949,41000	0,58795
600	598,432000	1,567820	1300	1300,60000	0,598089	2000	1998,15000	1,84674
650	648,893000	1,107170	1350	1349,91000	0,088622			

TABELA 5.6 - Rede Multicamada com 2 Camadas de 15 Neurônios - Erro médio = 48,369 ppm

CB real	CB rede	ERRO	CB real	CB rede	ERRO	CB real	CB rede	ERRO
0	22,139400	22,139400	700	672,47300	27,527100	1400	1405,92000	5,915150
50	36,429800	13,570200	750	734,10100	15,899400	1450	1405,93000	44,065200
100	75,914900	24,085100	800	802,61500	2,615080	1500	1405,91000	94,092900
150	119,331000	30,669400	850	878,12800	28,128300	1550	1332,20000	217,798000
200	166,904000	33,095500	900	956,58700	56,586900	1600	1678,20000	78,200500
250	218,268000	31,732300	950	1019,02000	69,021500	1650	1674,19000	24,194200
300	272,577000	27,422900	1000	1045,74000	45,744100	1700	1719,70000	19,703200
350	327,838000	22,162500	1050	1043,11000	6,891480	1750	1802,05000	52,051400
400	381,297000	18,703000	1100	1219,16000	119,159000	1800	1839,36000	39,357500
450	431,000000	19,000200	1150	1238,77000	88,773900	1850	1856,42000	6,424400
500	477,092000	22,907500	1200	1233,62000	33,615600	1900	1761,53000	138,475000
550	521,662000	28,337600	1250	1226,49000	23,512600	1950	1955,67000	5,671790
600	567,548000	32,452200	1300	1594,96000	294,958000	2000	1970,42000	29,578300
650	617,255000	32,745000	1350	1406,16000	56,163000			

As redes multicamadas apresentaram uma grande variação nos erros encontrados como mostram as tabelas anteriores. Isto ocorreu até nas melhores redes multicamadas. Comparando as melhores redes multicamadas com a Rede de Base Radial (RBR),

podemos observar que o erro encontrado na rede RBR, foi significativamente menor que nas redes multicamadas uma vez que os erros médios das redes multicamadas apresentadas foi 1,596 ppm (tabela 5.2); 0,517 ppm (tabela 5.3); 8,891 ppm (tabela 5.4); 2,030 ppm (tabela 5.5); 48,369 ppm (tabela 5.6) respectivamente, enquanto o erro médio da Rede RBR foi de 0,063 ppm (tabela 5.1).

Podemos observar nas tabelas que entre os valores iniciais da concentração de boro os erros são maiores do que entre os outros valores da tabela, conforme podemos ver nas tabelas 5.2, 5.3 e 5.4. Esta observação pode ser explicada pelo efeito de borda que ocorre devido ao fato de que o $K_{EFETIVO}$ ser próximo de 1 quando concentração de boro é igual a zero.

5.3 – Concentração Crítica de Boro

Feito o uso das redes acima mencionadas para determinar a concentração crítica de boro e posteriormente testadas para verificar se o reator IAEA apresentava $K_{EFETIVO} = 1$, o resultado obtido se encontra na tabela 5.7.

TABELA 5.7 - $K_{EFETIVO}$ obtido com CB crítico-rede

K EFETIVO obtido com o NEM	CB crítico-rede	TIPO DE REDE
1,000230302	139,777	Rede de base radial
0,999030278	145.813	Rede multicamada com 1 camada oculta de 10 neurônios
1,0013062017	134.379	Rede multicamada com 1 camada oculta de 15 neurônios
1,0020511539	130.649	Rede multicamada com 3 camadas ocultas de 10 neurônios
1,0019312389	131.249	Rede multicamada com 4 camadas ocultas de 15 neurônios
1,002178099	130,014	Rede multicamada com 1 camada oculta de 9 neurônios

As CBs crítico-rede foram encontradas fazendo $K_{EFETIVO} = 1$ nas respectivas redes neurais, ou seja, correspondem à concentração crítica de boro obtida pelas redes neurais, com os parâmetros nucleares originais. Ambos foram salvos em um novo arquivo de parâmetros nucleares e utilizado no código NEM para verificação da criticalidade do reator de acordo com a concentração de boro obtida pela rede. Os

resultados mostram que a rede de base radial apresentou o melhor desempenho. O cálculo da concentração crítica de boro na rede de base radial levou cerca de 0,31 segundos, utilizando esta concentração crítica obtida no código NEM para verificação do resultado vemos que o $K_{\text{EFETIVO}} = 1,000230302$ e levou cerca de 2,5 segundos, somando os tempos de cálculo da concentração crítica pela rede de base radial e o tempo de utilizado pelo código NEM para verificação se o reator se encontra crítico é ainda menor que o cálculo efetuado pela pesquisa de criticalidade do código NEM.

6 CONCLUSÕES

A grande dificuldade neste trabalho foi garantir informações para o treinamento, pois inicialmente não tínhamos informações suficientes para montar os dados de treinamento. Foi necessário construir uma estratégia para obter os dados de treinamento da rede. Além disso as posições dos diversos tipos de ECs influenciam no cálculo do K_{EFETIVO} e com isso foi necessário levar em consideração as posições dos diversos ECs na montagem dos dados de treinamento para as diversas redes. Por outro lado a obtenção dos parâmetros nucleares para o treinamento foi feito de maneira empírica, sendo necessário aprimorar esta obtenção dos parâmetros nucleares.

Os resultados mostram que uma rede neural artificial pode ser usada para determinar a concentração crítica de boro, sendo a rede que obteve melhor resultado foi a rede de base radial. Todas as redes que foram utilizadas neste trabalho têm uma limitação muito importante, elas dependem da configuração do núcleo do reator, ou seja, dependem do número de regiões nas direções X e Y e do número total de tipos existentes dos ECs. Esta limitação traz a questão da geometria do reator e a sua composição (Concentração de Boro) para obtenção de sua criticalidade.

Devido a esta limitação não há uma rede neural artificial que obtenha a concentração crítica de boro para todas as configurações possíveis de um reator, isto é, torna-se necessária uma rede para cada configuração do reator. As redes são obtidas em quatro fases que são: a primeira fase é a geração de novos arquivos dos parâmetros nucleares a partir do arquivo dos parâmetros nucleares originais; a segunda fase é a utilização dos novos arquivos dos parâmetros nucleares no código NEM em diferentes concentrações de boro para calcular os K_{EFETIVOS} ; a terceira fase é o treinamento das redes utilizando os arquivos dos parâmetros nucleares gerados em diferentes concentrações de boro e os respectivos K_{EFETIVOS} e a quarta fase é a obtenção da concentração crítica de boro utilizando a rede treinada.

A questão da geometria do reator e a sua composição na obtenção da concentração crítica de boro passam por uma simplificação dos parâmetros nucleares, apresentando aqueles de maior importância para o cálculo desta concentração e aqueles que podem ser desprezados, visando, desse modo, facilitar a construção e o treinamento das redes que obtêm a concentração crítica de boro.

O uso de redes neurais como substituto de processos iterativos de alto custo computacional em código de física de reatores se mostrou viável, pois o cálculo da concentração crítica de boro levou cerca 2,81 segundos considerando o tempo de cálculo da obtenção pela rede de base radial que foi de 0,31 segundos e somado com a verificação pelo código NEM que foi de 2,5 segundos, sendo ainda inferior aos 5 segundos pelo método descrito na introdução

7 REFERÊNCIAS BIBLIOGRÁFICAS

Alvim, A. C. M., 2007, “Métodos Numéricos em Engenharia Nuclear, 1 Ed. São Paulo, Centauro.

Argonne National Laboratory, 1977. Argonne Code Center: Benchmark Problem Book ANL – 7416 Supplement 2.

Cover, 1965., “Geometrical and Statistical Properties of Systems of linear Inequalities with Applications in Pattern Recognition”, IEEE Transactions on Electronic Computers, vol. EC-14, PP. 326-334.

Da Silva S. F., 2000. “Aplicações de Redes Neurais Supervisionadas e Não Supervisionadas para Identificação de Eventos Pós Desarme de um Reator PWR.” Tese de Mestrado, PEN/COPPE/UFRJ, Rio de Janeiro

Duderstadt, J. J. and Hamilton, L. J., 1976, “Nuclear reactor Analysis” 1ª Edição, John Wiley & Sons, Inc, New York.

Finnemann, H., Bennewitz, F. and Wagner, M. R., 1977, “Interface current techniques for multidimensional reactor calculations”, Atomkernenergie, vol. 300, pp. 123-127.

Glasstone, S., Sesonske A., 1981. “Nuclear Reactor Engineering” 3ª Edição, Van Nostrand Reinhold Company, New York.

Guimarães, C. de S., da Silva, F. C., Martinez, A. S., 2007., “Using the ACMFD for the Numerical Solution of the Three-Dimensional Neutron Diffusion Equation.” X Encontro de Modelagem Computacional, 21 a 23 de novembro de 2007, Nova Friburgo, RJ

Haykin, S., 2001. “Redes Neurais. Princípios e Práticas.” 2ª Edição, Bookman, Porto Alegre 2001

Kovács, Z. L., 1996. “Redes Neurais Artificiais: Fundamentos e Aplicações.”

2ª Edição, collegium cónitio, São Paulo.

Micchelli, C. A., 1986. "Interpolation of Scattered Data: Distance Matrices and Conditionally positive Definite Functions", *Constructive Approximation*, vol 2, pp. 11-22.

Pereira Neto, W. F., 2011. "Novo Procedimento para a Pesquisa de Criticalidade em Métodos Nodais de Malha Grossa." Tese de Mestrado, PEN/COPPE/UFRJ, Rio de Janeiro.

Silva, F. C. e Martinez, A. S., 2003. "Aceleração do método nodal NEM usando diferenças finitas de malha grossa." VI Encontro de Modelagem Computacional, 1 a 3 de dezembro de 2003, Nova Friburgo, RJ.

8 ANEXOS

8.1 - A Interpolação nas Redes de Funções de Bases Radiais

As funções de bases radiais são aplicadas em problemas multidimensionais que necessitam de interpolação de dados que advêm de amostras não uniformes. Um ponto crucial de uma aproximação por bases radiais é não necessidade da construção de uma malha, pois uma função radial depende somente da distância a um dado centro. Desta forma a propriedade geométrica da distância entre dois pontos é a única a ser utilizada, sendo esta propriedade de fácil utilização em qualquer dimensão.

As funções de bases radiais apresentam a seguinte forma:

$$s(\mathbf{x}) = \sum_{j=1}^N \beta_j \varphi(\|\mathbf{x} - \mathbf{x}_j\|) + p(\mathbf{x}), \mathbf{x} \in \mathfrak{R}^d \quad (1)$$

onde $\varphi: \mathbb{R}^+ \rightarrow \mathbb{R}$, $r = \|\mathbf{x} - \mathbf{x}_j\|$ é a norma euclidiana e $p \in \Pi_m^d = \text{span}(p_1, \dots, p_q)$ é um polinômio de grau m , com à restrição $\sum_{j=1}^N \beta_j p(\mathbf{x}_k) = 0$ para todo $p \in \Pi_m^d$. Relacionamos abaixo as funções de bases radiais mais comuns com os seus respectivos graus dos polinômios p .

Thin Plate Splines (TPS)	$\rightarrow \varphi = r^2 \log(r)$	$\rightarrow m = 2$
Multiquádricas	$\rightarrow \varphi = \sqrt{r^2 + c^2}$	$\rightarrow m = 1$
Gaussianas	$\rightarrow \varphi = e^{-r^2}$	$\rightarrow m = 0$
Multiquádricas Inversa	$\rightarrow \varphi = (r^2 + c^2)^{-1/2}$	$\rightarrow m = 0$
Base de suporte compacto	$\rightarrow \varphi = (1 - r)^4 + (4r + 1)$	$\rightarrow m = 0$

A questão da interpolação no treinamento de uma rede de base radial pode não ser uma boa estratégia para certas classes de problemas devido à pobreza da generalização para novos dados por causa das seguintes razões: Quando o número de pontos de dados na amostra de treinamento é muito maior que o grau de liberdade do processo físico relacionado, são necessárias tantas funções de base radial quanto forem os pontos da amostra, o problema é indeterminado. Em consequência, a rede pode

acabar ajustando variações enganosas devido ao ruído nos dados de entrada, desse modo resulta uma degradação do desempenho da generalização da rede (Broomhead e Lowe, 1988), ou seja, ajuste excessivo

O projeto de uma rede neural treinada consiste em recuperar um padrão de saída quando se apresenta um padrão de entrada, isto é, equivalente a aprender um mapeamento multidimensional entre o espaço de entrada e o espaço de saída da rede. O mapeamento entre os espaços é um problema direto, e como um problema direto pode ser bem formulado (bem posto) ou mal formulado (mal posto).

Para compreender o conceito de bem formulado, suponha que tenhamos um domínio X e um intervalo Y dotados de espaço métrico e relacionados por um mapeamento fixo e desconhecido f . Dizemos que o problema da determinação do mapeamento f é bem formulados se as três condições abaixo forem satisfeitas (HAYKIN, S., 2001):

1) Existência. Para cada vetor de entrada $\mathbf{x} \in X$, existe uma saída $\mathbf{y} = f(\mathbf{x})$, onde $\mathbf{y} \in Y$.

2) Unicidade. Para qualquer par de vetores de entrada $\mathbf{x}, \mathbf{t} \in X$, temos $f(\mathbf{x}) = f(\mathbf{t})$ se e somente se $\mathbf{x} = \mathbf{t}$.

3) Continuidade. O mapeamento é contínuo. Isto é, para qualquer $\varepsilon > 0$ existe $\delta = \delta(\varepsilon)$ tal que a condição $\|\mathbf{x} - \mathbf{t}\| < \delta$ implica que $\|f(\mathbf{x}) - f(\mathbf{t})\| < \varepsilon$, onde $\|\cdot\|$ é a distância entre os pontos nos respectivos espaços. A propriedade da continuidade é também conhecida como estabilidade.

Com relação aos problemas utilizados nas redes são geralmente diretos e bem formulados, mas existem situações que pode torna um problema mal formulado, que são (HAYKIN, S., 2001):

1) O critério de existência pode ser violado na medida que para toda a entrada pode não existir uma saída distinta.

2) Pode não haver tanta informação na amostra de treinamento quanto realmente necessitamos para reconstruir unicamente o mapeamento de entrada-saída, provavelmente o critério de unicidade seja violado.

3) A inevitável presença de ruído ou imprecisão nos dados de treinamento reais adiciona incertezas ao mapeamento de entrada-saída reconstruído. Em particular, se o nível de ruído na entrada for muito grande, é possível que a rede neural produza uma saída fora do intervalo \mathbf{Y} para uma entrada específica \mathbf{x} do domínio \mathbf{X} ; em outras palavras, há uma probabilidade de que o critério de continuidade seja violado.

Um problema mal formulado pode ser transformado em problema bem formulado, para isto, fazemos o uso da teoria de regularização, que foi proposta por Tikhonov em 1963, que consiste em estabilizar a solução por meio de algum funcional não negativo auxiliar que incorpore informação prévia sobre a solução. A regularização envolve dois termos que são:

1) O Termo do Erro Padrão \rightarrow Que é representado por $\mathcal{G}_S(F)$, e mede o erro padrão entre a resposta desejada d_i e a resposta real y_i para o exemplo de treinamento $i = 1, 2, \dots, N$. Que é definido como:

$$\mathcal{G}_S(F) = \frac{1}{2} \sum_{i=1}^N (d_i - y_i)^2 = \frac{1}{2} \sum_{i=1}^N [d_i - F(x_i)]^2 \quad (2)$$

2) Termo de Regularização \rightarrow Que é representado por $\mathcal{G}_C(F)$, depende das propriedades “geométricas” da função aproximativa $F(\mathbf{x})$. Que é definido como:

$$\mathcal{G}_C(F) = \frac{1}{2} \|\mathbf{D}F\|^2 \quad (3)$$

onde \mathbf{D} é um operador diferencial linear. A informação prévia sobre a forma da solução é incorporada ao operador \mathbf{D} , o que faz \mathbf{D} depender do problema. O operador \mathbf{D} é também conhecido como estabilizador, pois este estabiliza a solução para o problema de regularização através da suavização, e com isso satisfazendo a propriedade da continuidade. Todavia, suavidade implica continuidade, mas o inverso não é necessariamente verdade.

A regularização pode ser feita pelo seguinte princípio (HAYKIN, S., 2001):

Encontre a função $F_\lambda(\mathbf{x})$ que minimiza o funcional de Tikhonov $\mathcal{G}(F)$, definido por:

$$\mathcal{G}(F) = \mathcal{G}_S(F) + \lambda \mathcal{G}_C(F) \quad (4)$$

onde λ é o parâmetro de regularização.

O parâmetro λ é um indicador da suficiência do conjunto de dados fornecidos como exemplos que especificam a solução $F_\lambda(\mathbf{x})$. No caso limite $\lambda \rightarrow 0$ implica que o problema é irrestrito, isto é, a solução $F_\lambda(\mathbf{x})$ é totalmente determinada pelos exemplos, já o caso limite $\lambda \rightarrow \infty$ implica que os dados não são confiáveis, só sendo garantida a solução $F_\lambda(\mathbf{x})$ pela restrição prévia de suavidade imposta pelo operador \mathbf{D} . Em aplicações práticas o valor do parâmetro λ se encontra entre os casos limites, de forma que tanto os dados da amostra quanto a informação prévia contribuem para a solução $F_\lambda(\mathbf{x})$.

A minimização do funcional $\mathcal{G}(F)$ nos leva a seguinte solução:

$$F_\lambda(\mathbf{x}) = \frac{1}{\lambda} \sum_{i=1}^N [d_i - F(\mathbf{x}_i)] G(\mathbf{x}, \mathbf{x}_i) \quad (5)$$

Sendo a minimização do funcional $\mathcal{G}(F)$ do problema de regularização a superposição linear de N funções de Green $G(\mathbf{x}, \mathbf{x}_i)$. Os \mathbf{x}_i representam os centros da expansão e os pesos $[d_i - F(\mathbf{x}_i)]/\lambda$ representam os coeficientes de expansão.

A questão da determinação dos coeficientes da expansão da equação (5), pode ser resolvida da seguinte maneira, primeiro definindo

$$w_i = \frac{[d_i - F(\mathbf{x}_i)]}{\lambda} \quad (6)$$

o que permite escrever em $\mathbf{x}_j, j = 1, 2, \dots, N$

$$F_\lambda(\mathbf{x}_j) = \sum_{i=1}^N w_i G(\mathbf{x}_j, \mathbf{x}_i), \quad j = 1, 2, \dots, N \quad (7)$$

e fazendo as seguintes definições:

$$\mathbf{F}_\lambda = [F_\lambda(\mathbf{x}_1) \quad F_\lambda(\mathbf{x}_2) \quad \dots \quad F_\lambda(\mathbf{x}_N)]^T \quad (8)$$

$$\mathbf{d} = [d_1 \quad d_2 \quad \dots \quad d_N]^T \quad (9)$$

$$\mathbf{w} = [w_1 \quad w_2 \quad \dots \quad w_N]^T \quad (10)$$

$$\mathbf{G} = \begin{bmatrix} G(\mathbf{x}_1, \mathbf{x}_1) & G(\mathbf{x}_1, \mathbf{x}_2) & \dots & G(\mathbf{x}_1, \mathbf{x}_N) \\ G(\mathbf{x}_2, \mathbf{x}_1) & G(\mathbf{x}_2, \mathbf{x}_2) & \dots & G(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ G(\mathbf{x}_N, \mathbf{x}_1) & G(\mathbf{x}_N, \mathbf{x}_2) & \dots & G(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \quad (11)$$

Permite-nos escrever as equações (6) e (7) na forma matricial, como:

$$\mathbf{w} = \frac{1}{\lambda}(\mathbf{d} - \mathbf{F}_\lambda) \quad (12)$$

e

$$\mathbf{F}_\lambda = \mathbf{G}\mathbf{w} \quad (13)$$

Tirando \mathbf{F}_λ da equação (12) e substituindo na equação (13) obtemos

$$(\mathbf{G} + \lambda\mathbf{I})\mathbf{w} = \mathbf{d} \quad (14)$$

onde \mathbf{I} é a matriz Identidade N por N e \mathbf{G} é a matriz de Green.

Como a função de Green obtida na minimização do funcional de Tikhonov $\mathcal{G}(\mathbf{F})$ é simétrica (HAYKIN, S., 2001), ou seja,

$$G(\mathbf{x}_i, \mathbf{x}_j) = G(\mathbf{x}_j, \mathbf{x}_i) \quad \text{para todo } i \text{ e } j \quad (15)$$

Portanto, a matriz de Green definida na equação (11) é uma matriz simétrica, isto é,

$$\mathbf{G} = \mathbf{G}^T \quad (16)$$

A questão da determinação dos coeficientes da expansão nos leva ao problema da interpolação, ou seja, $\mathbf{G} + \lambda\mathbf{I}$ deve ser não é singular, para garantir a solução da equação (14). Devemos ter \mathbf{G} positivamente definida, isto é obtido com pontos de dados $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ distintos, e na prática, podemos escolher λ suficientemente grande, de modo a garantir que $\mathbf{G} + \lambda\mathbf{I}$ seja positivamente definida e com isso garantir a sua inversa. Portanto, a solução do sistema de equações (14) é obtida por

$$\mathbf{w} = (\mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{d} \quad (17)$$

Das classes de funções Green coberta pelo teorema de Micchelli sobre interpolação, temos as multiquádricas inversas e as funções gaussianas, que satisfazem a solução acima.

A solução do problema de regularização dada pela equação (7) é incompleta, pois esta representa um termo de solução de módulo $\mathbf{g}(\mathbf{x})$ que se encontra no espaço nulo do operador \mathbf{D} (Poggio e Girosi, 1990). Isto porque todas as funções que estão no espaço nulo de \mathbf{D} são “invisíveis” para o termo $\|\mathbf{D}\mathbf{F}\|^2$ no funcional de Tikhonov $\mathcal{J}(\mathbf{F})$. O **significado** de espaço nulo para \mathbf{D} , e quando todo um conjunto de funções $\mathbf{g}(\mathbf{x})$ satisfaz a condição $\mathbf{D}\mathbf{g}(\mathbf{x}) = \mathbf{0}$. O termo $\mathbf{g}(\mathbf{x})$ depende do problema, ou seja, depende da escolha do estabilizador e das condições de contorno do problema específico. Este termo não é necessário no caso de se utilizar um estabilizador \mathbf{D} que corresponde funções de Green com forma de sino, como por exemplo, gaussianas ou multiquádricas inversas. Devido a este fato, e como as principais conclusões não é afetada pela sua inclusão ou não, portanto não o consideraremos.

A característica da função Green $G(\mathbf{x}, \mathbf{x}_i)$ para um determinado centro \mathbf{x}_i é que ela depende da forma do estabilizador \mathbf{D} , ou seja, através da suposição feita a priori do mapeamento entrada-saída. No caso do estabilizador ser invariante em relação à translação, a função de Green $G(\mathbf{x}, \mathbf{x}_i)$ centrada em \mathbf{x}_i depende apenas da diferença entre os argumentos \mathbf{x} e \mathbf{x}_i , temos então:

$$G(\mathbf{x}, \mathbf{x}_i) = G(\mathbf{x} - \mathbf{x}_i) \quad (18)$$

Já no caso do estabilizador \mathbf{D} ser invariante em relação à translação e à rotação, a função de Green $G(\mathbf{x}, \mathbf{x}_i)$ depende apenas da norma do vetor diferença $\mathbf{x} - \mathbf{x}_i$, isto é,

$$G(\mathbf{x}, \mathbf{x}_i) = G(\|\mathbf{x} - \mathbf{x}_i\|) \quad (19)$$

Com estas condições a função de Green deve ser uma função de base radial, portanto a solução regularizada pode ser escrita como:

$$F_\lambda(\mathbf{x}) = \sum_{i=1}^N w_i G(\|\mathbf{x} - \mathbf{x}_i\|), \quad (20)$$

Esta solução corresponde a construção de um espaço linear de funções que depende dos pontos de dados conhecidos de acordo com a distância euclidiana. A solução dada pela equação (20) é denominada interpolação estrita, porque todos N pontos disponíveis para o treinamento são utilizados para construção da função de interpolação $F(\mathbf{x})$. A função de Green $G(\mathbf{x}, \mathbf{x}_i)$ em que operador diferencial linear \mathbf{D} é invariante em relação à translação e à rotação e satisfaz a condição da equação (20), é a função gaussiana multivariada, que é definida por

$$G(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{1}{2\sigma_i^2} \|\mathbf{x} - \mathbf{x}_i\|^2\right) \quad (21)$$

onde \mathbf{x}_i é o centro da gaussiana e σ_i é a largura da gaussiana.

8.2 -As Funções de Base

As funções de base do NEM são definidas como:

$$h_0(\xi) = 1$$

$$h_1(\xi) = 2\xi - 1$$

$$h_2(\xi) = 6\xi(1 - \xi) - 1$$

$$h_3(\xi) = 6\xi(1 - \xi)(2\xi - 1)$$

$$h_4(\xi) = 6\xi(1 - \xi)(5\xi^2 - 5\xi + 1)$$

onde $\xi = \frac{u}{a_u^n}$

e possuem as seguintes propriedades:

$$\int_0^1 h_k(\xi) d\xi = 0, \quad \forall k \geq 1$$

$$h_k(1) = h_k(0) = 0, \quad \forall k \leq 3$$

$$\int_0^1 h_1(\xi) \frac{d^2 h_k}{d\xi^2} d\xi = \begin{cases} -12 & \text{se } k = 3 \\ 0 & \text{se } k \neq 3 \end{cases}$$

$$\int_0^1 h_2(\xi) \frac{d^2 h_k}{d\xi^2} d\xi = \begin{cases} 12 & \text{se } k = 4 \\ 0 & \text{se } k \neq 4 \end{cases}$$

$$\int_0^1 h_1(\xi) h_k(\xi) d\xi = \begin{cases} \frac{1}{3} & \text{se } k = 1 \\ 0 & \text{se } k = 2 \\ \frac{1}{5} & \text{se } k = 3 \\ 0 & \text{se } k = 4 \end{cases}$$

e

$$\int_0^1 h_2(\xi) h_k(\xi) d\xi = \begin{cases} 0 & \text{se } k = 1 \\ \frac{1}{5} & \text{se } k = 2 \\ 0 & \text{se } k = 3 \\ \frac{-3}{35} & \text{se } k = 4 \end{cases}$$

8.3 – A Subrotina que Faz a Montagem da Matriz de Treinamento

```

function [TR] = MONTATREINA4(OPCAO,TRATA,DN,CBKEFF)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               MONTAGEM DE TR.P E TR.T                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
j = 1;
for i = 1 : CBKEFF.nb
    if OPCA0 == 'KE'
        TR.P(1,i) = CBKEFF.CB(i);
        TR.T(i) = CBKEFF.KEFF(i);
    else if OPCA0 == 'CB'
        TR.P(1,i) = CBKEFF.KEFF(i);
        TR.T(i) = CBKEFF.CB(i);
    end
end
DN = CBSIGA(DN,CBKEFF.CB(i));
for iy = 1 : TRATA.NTRY
    for ix = 1 : TRATA.NTRX
        if TRATA.CONFIG(iy,ix) > 0
            TR.P(j+01,i) = DN.SIGA (TRATA.CONFIG(iy,ix),1);
            TR.P(j+02,i) = DN.SIGA (TRATA.CONFIG(iy,ix),2);
            TR.P(j+03,i) = DN.NSIGF(TRATA.CONFIG(iy,ix),1);
            TR.P(j+04,i) = DN.NSIGF(TRATA.CONFIG(iy,ix),2);
            TR.P(j+05,i) = DN.CDIF (TRATA.CONFIG(iy,ix),1);
            TR.P(j+06,i) = DN.CDIF (TRATA.CONFIG(iy,ix),2);
            TR.P(j+07,i) = DN.SIGE (TRATA.CONFIG(iy,ix),1);
            TR.P(j+08,i) = DN.SIGE (TRATA.CONFIG(iy,ix),2);
            TR.P(j+09,i) = DN.GSIGF(TRATA.CONFIG(iy,ix),1);
            TR.P(j+10,i) = DN.GSIGF(TRATA.CONFIG(iy,ix),2);
        else
            TR.P(j+01,i) = 0.0;
            TR.P(j+02,i) = 0.0;
            TR.P(j+03,i) = 0.0;
            TR.P(j+04,i) = 0.0;
            TR.P(j+05,i) = 0.0;
            TR.P(j+06,i) = 0.0;
            TR.P(j+07,i) = 0.0;
            TR.P(j+08,i) = 0.0;
            TR.P(j+09,i) = 0.0;
            TR.P(j+10,i) = 0.0;
        end
        j = j + DN.NUMTIP * 2;
    end
end
j = 1;
end
return

```

8.4 - A Subrotina que Faz o Ajuste da Seção de Choque de Absorção

```
function [DN] = CBSIGA(DN,CB)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% AJUSTA A SECAO DE CHOQUE DE ABSORCAO %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for TIPO = 1 : DN.NUMTIP;
    for IG = 1 : 2
        AUX = 7.6891E-9 * CB * DN.SIGAB (IG);
        DN.SIGA (TIPO,IG) = DN.SIGA (TIPO,IG) + AUX;
    end
end
return
```