

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**Proposta de Implementação em Hardware dedicado
de Redes Neurais Competitivas
com Técnicas de Circuitos Integrados Analógicos**

por

ROLF FREDI MOLZ

Dissertação submetida à avaliação, como
requisito parcial para a obtenção do grau de
Mestre em Ciência da Computação.

Prof. Dr. Paulo M. Engel
Orientador

Porto Alegre, abril de 1998.

UFRRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Molz, Rolf Fredi

Proposta de Implementação em Hardware dedicado de Redes Neurais Competitivas com Técnicas Analógicas / por Rolf Fredi Molz. - Porto Alegre : CPGCC da UFRGS, 1997.

103 f.:il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Sul. Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, BR - RS, 1997. Orientador: Engel, Paulo M.

1. Redes Neurais Artificiais. 2. Inteligência Artificial. 3. Microeletrônica. 4. Modelamento. 5. Simulação Elétrica. I. Engel, Paulo M. II. Título

Engenharia elétrica - SBU
Microeletrônica
Inteligência artificial
Redes neurais
Simulação elétrica

UFRGS INSTITUTO DE INFORMÁTICA BIBLIOTECA		
N.º CHAMADA 621.38-481.4(043) M731P	N.º REG.: 34698	DATA: 01/10/98
ORIGEM: D	DATA: 10/9/98	PREÇO: R\$ 30,00
FUNDO: II	FORN.: II	

CNPq 3.04.03.00-6

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profa. Wrana Panizzi

Pró-Reitor de Pós-Graduação: Prof. José Carlos Ferraz Hennemann

Diretor do Instituto de Informática: Prof. Dr. Roberto Tom Price

Coordenador do CPGCC: Prof. Dr. Flávio Rech Wagner

Bibliotecária-Chefe do Instituto de Informática: Zita Prates de Oliveira



SABi



05225391

Agradecimentos

Ao Prof. Dr. Paulo M. Engel, pela sua valiosa orientação e aos incentivos fornecidos para a realização deste curso e na elaboração da presente dissertação.

Agradeço também aos professores membros da banca, por terem aceitado a fazerem parte desta, e assim fazem-me sentir honrado por tais presenças.

Ao Prof. Dr. Luigi Carro, pela sua disposição em me ajudar durante o transcurso deste mestrado.

A todos os colegas do Instituto que com o passar do tempo se tornaram meus grandes amigos, dentre eles: Marcello Macarthy, Rosaldo Rossetti, Juliano Tonezer e João Baptista que através do espírito de companheirismo tornaram mais alegres os momentos dentro do Instituto de Informática. A todas as demais pessoas que conheci durante este período.

A todos os funcionários do Instituto de Informática e da Biblioteca pelo esmero e disposição que sempre colocaram em seus serviços.

A minha mãe Dulce A. Molz e mesmo em memória ao meu pai Arlindo Molz, pelo seu profundo amor e apoio, e por tudo que me ensinaram no transcorrer da minha vida. Ao meu irmão Kurt W. Molz, com o qual dividi vários dos momentos vividos no transcorrer deste curso. Agradeço também a minha esposa Simoni pelo amor e confiança depositada em mim e pela paciência em escutar as tantas palavras difíceis que eu tinha que desabafar.

Agradeço a Deus, sem o qual nada seria possível, pela vida, pela esperança e pelas bênçãos.

Por fim, agradeço a todas as pessoas que ajudaram, direta ou indiretamente, durante a realização deste curso de mestrado.

A todos vocês o meu muito obrigado!

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Sistema de Bibliotecas da UFRGS

34698

621.38-181.4(043)
M731P

INF
1998/98009-2
1998/10/01

Sumário

Lista de Figuras	6
Lista de Tabelas	8
Lista de Abreviaturas	9
Lista de Símbolos	10
Resumo	11
Abstract.....	12
1 Introdução	13
2 Fundamentos de Redes Neurais Artificiais	15
2.1 Introdução	15
2.2 Componentes das Redes Neurais Artificiais.....	16
2.2.1 Unidades de Processamento.....	16
2.2.2 Estado de Ativação.....	17
2.2.3 Função de Saída	17
2.2.4 Padrão de Interconexão	17
2.2.5 Regra de Propagação.....	18
2.2.6 Função de Ativação	19
2.2.7 Regra de Aprendizado	20
2.2.8 Ambiente.....	21
2.3 Técnicas de aprendizado em Redes Neurais.....	21
2.3.1 Formas de Aprendizado.....	21
2.3.2 Dinâmica : Convergência e Estabilidade.....	23
2.4 Modos de Implementação de Redes Neurais.....	24
3 Implementação de redes neurais em <i>hardware</i> analógico	25
3.1 Introdução	25
3.2 Conexões Sinápticas Analógicas	25
3.3 Pesos Controlados por Tensão	26
3.4 Circuitos Ativos para Implementação de Redes Neurais.....	27
3.4.1 Neurônio Baseado em Inversor.....	28
3.4.2 Amplificadores de Tensão Diferencial.....	29
3.4.3 Amplificadores de Transcondutância.....	31
4 Redes Competitivas - Modelos Contínuos	32
4.1 Introdução	32
4.2 O modelo de Hopfield.....	32
4.3 Equação da Membrana.....	36
4.4 Derivações de Redes Neurais Competitivas.....	38
4.4.1 Redes <i>Shunting Feedforward</i>	39
4.4.2 Redes <i>Shunting Feedback</i>	40
4.5 Simulações Contínuas	42

5 Proposta de Implementação em Hardware dedicado de Redes Neurais Competitivas com Técnicas Analógicas.....	48
5.1 Introdução	48
5.2 Condutâncias.....	48
5.3 Capacitância.....	50
5.4 Rede Neural <i>Shunting Feedforward</i>	53
5.5 Rede Neural <i>Shunting Feedback</i>	56
5.6 Rede Neural <i>Shunting Feedback</i> com pesos	59
6 Validação da proposta de hardware dedicado através de uma aplicação ...	63
6.1 Introdução	63
6.2 Descrição do sistema.....	63
6.3 Resultados da aplicação	65
7 Conclusão	69
Anexo 1 Listagem C para redes neurais <i>shunting feedforward</i>	71
Anexo 2 Listagem C para redes neurais <i>shunting feedback</i>	74
Anexo 3 Listagem C para redes neurais <i>shunting feedback</i> com pesos	78
Anexo 4 Descrição elétrica de um transistor.....	82
Anexo 5 Descrição elétrica de uma rede neural <i>shunting feedforward</i>.....	83
Anexo 6 Descrição elétrica de uma rede neural <i>shunting feedback</i>.....	84
Anexo 7 Descrição elétrica de uma rede neural <i>shunting feedback</i> com pesos	86
Anexo 8 Descrição elétrica do sistema proposto	89
Anexo 9 Descrição elétrica do sistema proposto (tecnologia de 0.8μ)	94
Bibliografia.....	101

Lista de Figuras

FIGURA 2.1 - Neurônio como unidade limiar.....	17
FIGURA 2.2 - Exemplos de funções de ativação.....	19
FIGURA 2.3 - Representação do mínimo global.....	23
FIGURA 3.1 - Modelo neural para computação por software.....	25
FIGURA 3.2 - Implementação de um neurônio utilizando resistores e amplificador diferencial.....	26
FIGURA 3.3 - Implementação de um neurônio utilizando pesos ajustáveis e amplificador diferencial.....	27
FIGURA 3.4 - Diagrama do circuito do neurônio baseado em inversor.....	28
FIGURA 3.5 - Função de ativação característica de um neurônio baseado em inversor.....	28
FIGURA 3.6 - Memória autoassociativa recorrente baseado em neurônio do tipo inversor.....	29
FIGURA 3.7 - Esquema básico de um amplificador de tensão diferencial.....	30
FIGURA 3.8 - Função de ativação característica de um neurônio baseado em amplificador de tensão diferencial.....	30
FIGURA 3.9 - Diagrama do circuito de um amplificador de transcondutância.....	31
FIGURA 4.1 - Modelo sugerido por Hopfield.....	33
FIGURA 4.2 - Circuito eletrônico para o modelo de Hopfield.....	34
FIGURA 4.3 - Relação entre a saída e a entrada de um neurônio.....	34
FIGURA 4.4 - Um neurônio natural.....	37
FIGURA 4.5 - Circuito equivalente da membrana nervosa.....	38
FIGURA 4.6 - Camada Competitiva da Rede Neural do tipo <i>Shunting Feedforward</i>	39
FIGURA 4.7 - Camada Competitiva da Rede Neural do tipo <i>Shunting Feedback</i>	41
FIGURA 4.8 - Modelo da rede neural do tipo <i>shunting forward</i>	43
FIGURA 4.9 - Resultados da simulação de uma rede neural do tipo <i>shunting forward</i>	43
FIGURA 4.10 - Modelo da rede neural do tipo <i>shunting feedback</i>	44
FIGURA 4.11 - Resultados da simulação de uma rede neural do tipo <i>shunting feedback</i>	45
FIGURA 4.12 - Modelo da rede neural do tipo <i>shunting feedback</i> com pesos.....	46
FIGURA 4.13 - Resultados da simulação de uma rede neural do tipo <i>shunting feedback</i> com pesos.....	47
FIGURA 5.1 - Curva do transistor para diversas tensões de porta.....	49
FIGURA 5.2 - Gráfico da condutância pela tensão de dreno.....	50

FIGURA 5.3 - Descrição elétrica para verificação da capacitância.	51
FIGURA 5.4 - Gráfico da capacitância de um transistor MOS.	52
FIGURA 5.5 - Ligação física para um transistor operar como um capacitor não linear.	52
FIGURA 5.6 - Circuito para simulações da Rede Neural <i>Shunting Feedforward</i>	53
FIGURA 5.7 – Modelo do amplificador diferencial utilizado para as simulações.	54
FIGURA 5.8 - Circuito equivalente utilizando resistores e capacitores.	54
FIGURA 5.9 - Simulação da Rede Neural <i>Shunting Feedforward</i>	56
FIGURA 5.10 - Circuito para simulações da Rede Neural <i>Shunting Feedback</i>	57
FIGURA 5.11 - Simulação da Rede <i>Shunting Feedback</i> para o padrão final (0 ; 3).	58
FIGURA 5.12 - Simulação da Rede <i>Shunting Feedback</i> para o padrão final (3 ; 0).	59
FIGURA 5.13 - Circuito para simulações da Rede <i>Shunting Feedback</i> com pesos.	60
FIGURA 5.14 - Simulação da Rede <i>Shunting Feedback</i> com pesos para o padrão final (3;0).	61
FIGURA 5.15 - Simulação da Rede <i>Shunting Feedback</i> com pesos para o padrão final (0;3).	62
FIGURA 5.16 - Simulação da Rede <i>Shunting Feedback</i> com pesos para o padrão final (3;3).	62
FIGURA 6.1 - Modelo Neural.	63
FIGURA 6.2 - Esquema dos padrões e saídas desejadas para a aplicação em questão... ..	64
FIGURA 6.3 - Esquema elétrico do circuito para a aplicação.	64
FIGURA 6.4 - Simulação da rede neural para a aplicação proposta.	66
FIGURA 6.5 - Simulação da rede neural para a aplicação proposta para uma tecnologia de 0.8 μ	67
FIGURA 6.6 - Esquema elétrico do circuito para verificar a substituição de resistores e capacitores por transistores MOS.	68
FIGURA 6.7 - Simulação da rede neural para a aplicação proposta para uma tecnologia de 0.8 μ utilizando transistores no lugar das resistências e capacitâncias.	68

Lista de Tabelas

TABELA 2.1 - Diferenças entre o computador e o cérebro humano.....16

Lista de Abreviaturas

CMOS	Complementary Metal Oxide Semiconductor
CNN	Competitive Neural Networks
MOS	Metal Oxide Semiconductor
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
NMOS	Transistor MOS de canal n
PDP	Processamento Paralelo Distribuído
PMOS	Transistor MOS de canal p
RNA	Redes Neurais Artificiais
RNC	Redes Neurais Competitivas
SPICE	Simulating Program with Integrated Circuit Emphasis
STM	Short Term Memory
VLSI	Very Large Scale of Integration

Lista de Símbolos

C_{ox}	capacitância por unidade de área do óxido.
$f(x_i)$	função de ativação da unidade x_i .
$f(x_k)$	função de ativação da unidade x_k .
I_i	padrão aplicado a entrada da unidade i .
I_k	padrão aplicado a entrada da unidade j .
L_{mod}	comprimento do canal modulado pelo parâmetro de modulação do comprimento do canal (λ);
net	soma da entrada líquida dos pesos.
net_i	somatório do produto entre os valores dos pesos e sinais de entrada para a unidade i .
net_k	somatório do produto entre os valores dos pesos e sinais de entrada para a unidade k .
R_i	combinação paralela de ρ_i .
T_{ij}	condutância que conecta uma saída do amplificador j a entrada do amplificador i .
u_j	é o estado da j -ésima unidade.
W	largura do canal;
w_{ij}	é o peso da conexão da j -ésima unidade para a i -ésima unidade.
θ_i	é o limiar da i -ésima unidade.
μ_0	mobilidade superficial.
μ_s	valor estimado pela degradação da mobilidade superficial μ_0 .
γ_s	parâmetro de limiar do substrato corrigido para pequenas geometrias.
ϕ_f	potencial da superfície em inversão forte.

Resumo

Neste trabalho apresenta-se uma proposta de uma técnica para implementação em *hardware*, das estruturas básicas de uma Rede Neural Competitiva, baseada em técnicas analógicas.

Através desta proposta, será abordada uma das classes mais interessantes de Redes Neurais Artificiais (RNA) que são as Redes Neurais Competitivas (RNC), que possuem forte inspiração biológica. As equações fundamentais que descrevem o comportamento da RNC foram derivadas de estudos interdisciplinares, a maioria envolvendo observações neurofisiológicas. O estudo do neurônio biológico, por exemplo, nos leva à clássica equação da membrana.

A técnica mostrada para a implementação das Redes Neurais Competitivas se baseia no uso das técnicas analógicas. Estas conduzem a um projeto mais compacto além de permitirem um processamento em tempo real, visto que o circuito computacional analógico altera simultaneamente e continuamente todos os estados dos neurônios que se encontram interligados em paralelo.

Para esta proposta de implementação, é mostrado que as equações fundamentais que governam as Redes Neurais Competitivas possuem uma relação com componentes eletrônicos básicos, podendo então, serem implementados através destes simples componentes com os quais as equações fundamentais se relacionam.

Para tanto, é mostrado por meio de simulações em software, o comportamento das equações fundamentais deste tipo de Redes Neurais, e então, é comparado este comportamento, com os obtidos através de simulações elétricas dos circuitos equivalentes oriundos destas equações fundamentais. Mostra-se também, em ambas as simulações, uma das características mais importantes existentes nos modelos de RNC, conhecida como Memória de Tempo Curto (STM).

Por fim, é apresentada uma aplicação típica na área de clusterização de padrões utilizando pesos sinápticos, a fim de demonstrar a implementação utilizando as técnicas descritas durante o trabalho. Esta aplicação é demonstrada através de simulações elétricas, sendo estas realizadas para tipos diferentes de tecnologia, mostrando assim, o correto desempenho da proposta deste trabalho.

Palavras-chave: Redes neurais artificiais, Implementação em *hardware*, Técnicas analógicas, Redes neurais competitivas.

TITLE: "Proposal for implementation in dedicate hardware of competitive neural networks with analog integrated circuits techniques".

Abstract

In this work we present a proposal of a technique to hardware implementation of the basic structures of a Competitive Neural Network, based on analog circuits techniques.

This proposal approaches one of the most interesting classes of Artificial Neural Networks (ANN) that are the Competitive Neural Networks (CNN), that possess strong biological inspiration. The fundamental equations that describe the behavior of CNN were derived from interdisciplinary studies, mostly involving neurophysiological observations. The study of the biological neuron, for example, leads to the classical membrane equation.

The presented technique for implementation of Competitive Neural Networks is based on the use of analog circuits techniques. This leads to a more compact project and allows real time processing, because computation in analog circuits modifies simultaneously and continuously all the states of the neurons that are connected in parallel.

In this proposal, it is shown that the fundamental equations that describe the behavior of Competitive Neural Networks possess a relationship with some basic electronic components. This fact allows the direct implementation of CNN with these electronic components.

Initially the behavior of the fundamental equations of this type of Neural Networks is studied by means of software simulations. This behavior is then compared, with the one obtained through electric simulations of the equivalent circuits originated from these fundamental equations. It is also shown, in both simulations, one of the more important characteristic in the models of CNN, known as Short Term Memory (STM).

Finally, a typical application is presented in the area of pattern clustering using synaptic weights, to demonstrate an implementation using the techniques described in this work. This application is demonstrated through electric simulations, for different IC technologies, comproving the correctness of the presented proposal.

Keywords: Artificial neural networks, Hardware implementation, Analog technique, Competitive neural networks.

1 Introdução

Com o desenvolvimento da área de redes neurais artificiais, acentuou-se a necessidade da realização de circuitos integrados neurais, e não apenas a simulação dessas redes por meio de um programa. As principais razões para tal necessidade são: o aumento da capacidade de processamento; o desenvolvimento de arquiteturas adequadas para o processamento neural; e uma maior facilidade para o desenvolvimento de equipamentos comerciais com a redução de suas dimensões, de seus custos e de um aumento de confiabilidade dos mesmos.

A implementação de redes neurais artificiais em VLSI tira vantagens do inerente paralelismo para se obter soluções rápidas. Nas implementações em VLSI de redes neurais, técnicas analógicas são preferíveis, por essas conduzirem a um projeto mais compacto e permitirem o processamento em tempo real [MEA89]. Esta velocidade, necessária para processamento em tempo real de informações, pode ser fornecida por um circuito computacional analógico porque todos os neurônios alteram simultaneamente e continuamente seus estados analógicos em paralelo. Quando comparado a modernos computadores digitais de propósito geral, construídos com circuitos convencionais, os circuitos neurais computacionais possuem diferenças marcantes quanto as características e à organização [MEA89]. Cada porta lógica, tipicamente, obterá entradas de dois ou três outros neurônios, e um grande número de decisões binárias independentes são feitas no curso de uma computação digital. Em contrapartida, cada processador (neurônio) neural não linear em uma rede computacional analógica adquire entradas de dez ou centenas de outros neurônios e uma solução coletiva é computada na base de interações simultâneas de centenas de dispositivos.

O objetivo deste trabalho é propor uma nova técnica para a implementação em *hardware*, das estruturas básicas de uma rede neural artificial, mais especificamente uma rede neural do tipo competitiva, baseado em técnicas de circuitos integrados analógicos. Para tanto, serão utilizados circuitos básicos para a construção de redes neurais artificiais.

Após esta introdução, o capítulo 2 apresenta os conceitos básicos de redes neurais artificiais, sem contudo ter a pretensão de exaurir os conceitos, sendo que tal objetivo não pertence ao escopo deste trabalho.

No capítulo 3, disserta-se sobre algumas das técnicas utilizadas para a implementação analógica de redes neurais artificiais, tentando com isso esclarecer alguns dos pontos mais básicos.

O capítulo 4, aborda os conceitos relativos a algumas das redes neurais que possuem seus modelos descritos de forma contínua. As redes competitivas e suas derivações propostas por Stephen Grossberg em seu trabalho [GRO88] também são descritas neste capítulo. Ao fim deste capítulo também são demonstrados os resultados obtidos através da simulação computacional das derivações das redes neurais competitivas.

No capítulo 5, é proposta uma implementação em *hardware* de redes neurais competitivas através de técnicas de circuitos integrados analógicos. É mostrado por simulações, que as características mais importantes da rede neural competitiva são obtidas com a implementação apresentada. Neste capítulo também são demonstradas diversas simulações através do uso do software de simulações elétricas HSPICE, visando demonstrar a possível implementação em *hardware* dos circuitos propostos.

Por fim, no capítulo 6, é apresentado um modelo para uma aplicação específica na área de clusterização de padrões. Esta aplicação é somente para validar o modelo proposto, sendo por isto, de pequeno porte.

2 Fundamentos de Redes Neurais Artificiais

2.1 Introdução

Neste capítulo é feita uma revisão sobre redes neurais artificiais. Inicialmente são descritos os componentes básicos destes modelos e como estes se interrelacionam. A seguir é introduzido o conceito de aprendizado em redes neurais e conceitos como convergência e estabilidade. Por fim, são abordados os padrões básicos de conexão entre os neurônios.

As redes neurais artificiais se fundamentam nos estudos sobre a estrutura do cérebro humano para tentar emular sua forma inteligente de processar informação. Alguns estudos da neurofisiologia consideram que a riqueza computacional do cérebro humano vem do grande número de neurônios que estão interconectados por uma rede complexa de sinapses [CAV88].

A velocidade de processamento destes componentes individuais é baixa se comparada com a velocidade dos componentes digitais dos computadores tradicionais. Esta deficiência, entretanto, é largamente superada pela imensa quantidade de neurônios existentes operando em paralelo [SIM90]. Estima-se que existam cerca de 10^{11} a 10^{14} neurônios operando em paralelo no cérebro humano. Cada um destes está conectado através de 10^3 a 10^4 sinapses em média [COT85].

Tais características permitem ao cérebro humano executar rapidamente certas funções (por exemplo, reconhecer fisionomias) que os computadores convencionais não conseguem realizar com o mesmo desempenho. Na tabela 2.1, baseada em Cottrell [COT85] e Simpson [SIM90], é apresentada uma comparação das diferenças existentes entre os computadores tradicionais e o cérebro humano. Esta comparação nos permite ter uma idéia mais clara sobre a capacidade adaptativa do cérebro humano, em contraste com a rigidez e a precisão dos computadores convencionais.

Uma possível definição, de redes neurais artificiais (RNAs), também chamadas de modelos conexionistas de computação, redes neuronais ou sistemas de processamento paralelo distribuído (PDP), é apresentada a seguir:

“Um modelo conexionista é uma estrutura de processamento de informação distribuída e paralela. Ela é formada por unidades de processamento, comumente chamadas de nós, neurônios ou células, interconectadas por arcos unidirecionais, também chamadas de ligações, conexões ou sinapses. Os nós possuem memória local e podem realizar operações de processamento de informação localizada. Cada célula possui uma única saída (axônio), a qual pode se ramificar em muitas ligações colaterais (cada ramificação possuindo o mesmo sinal de saída do neurônio). Todo o processamento que se realiza em cada unidade deve ser completamente local, isto é, deve depender apenas dos valores correntes dos sinais de entrada que chegam dos neurônios através

das conexões. Estes valores atuam sobre os valores armazenados na memória local da célula". [HEC89]

TABELA 2.1 - Diferenças entre o computador e o cérebro humano.

Fonte : [COT85] e [SIM90].

Características	Computador	Cérebro humano
Elementos computacionais	processadores	neurônio simples
Velocidade de processamento	10^{-9} segundos	10^{-3} segundos
Tipo de processamento	serial	paralelo
Confiabilidade dos elementos	confiável	não-confiável
Tolerância a falhas	quase nenhuma	grande
Tipo de sinal	precisos, simbólicos	imprecisos
Tipo de controle	centralizado	distribuído
Armazenamento de informações	substituível	adaptativo

Os principais elementos utilizados nas descrições de RNAs são a representação distribuída, as operações locais e o processamento não-linear [SIM90]. Estes atributos especificam duas aplicações básicas dos modelos conexionistas: situações onde poucas decisões tem que ser tomadas a partir de uma grande quantidade de dados e situações onde um complexo mapeamento não-linear deve ser aprendido. Geralmente, o ferramental matemático usado nesta tecnologia inclui: equações diferenciais, sistemas dinâmicos, álgebra linear, probabilidade e estatística.

2.2 Componentes das Redes Neurais Artificiais

De acordo com Rumelhart [RUM86], uma rede neural artificial pode ser descrita por oito elementos principais:

- um conjunto de unidades de processamento;
- um estado de ativação;
- uma função de saída;
- um padrão de interconexão;
- uma regra de propagação;
- uma regra de ativação;
- uma regra de aprendizado;
- um ambiente onde o sistema deve funcionar.

2.2.1 Unidades de Processamento

As células, nós ou neurônios constituem o meio de representação do conhecimento, por exemplo conceitos de um domínio, existente na rede neural. Os nós podem representar pontos (pixels), caracteres (letras, números), palavras ou outros conceitos, dependendo da aplicação. Na figura 2.1 temos uma ilustração de um neurônio como unidade de limiar. As entradas que chegam a ele representam os dendritos. Cada dendrito possui um

signal que é adicionado, (Σ). Depois da adição, o sinal é processado através da função limiar, $f()$, a qual produz um sinal de saída. Nesta figura, o neurônio de limiar pode ser considerado como uma representação simplificada dos neurônios biológicos.

Em geral, segundo o nível de conhecimento representado pela RNAs, existem dois tipos de representação do conhecimento possíveis: a localizada e a distribuída. A representação localizada corresponde a um único neurônio representando um conceito. Na representação distribuída, o padrão de ativação de um conjunto de unidades é que possui significado [RUM86]. Neste capítulo, as unidades de processamento da rede são designadas pela letra u , seguida de um índice i que indica a posição que o neurônio ocupa na rede.

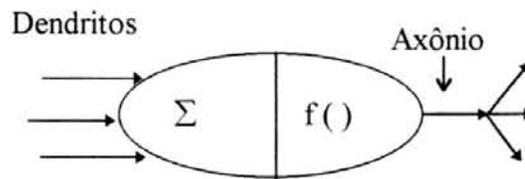


FIGURA 2.1 - Neurônio como unidade limiar.

2.2.2 Estado de Ativação

Cada célula u_i , da rede computa um estado de ativação, que é um valor numérico líquido de saída. O cálculo desta ativação é computado a partir das ativações das células conectadas diretamente a este nó, e dos correspondentes pesos destas conexões.

O estado de ativação de todas as unidades da rede, ou seja, o estado de ativação do sistema, especifica o que está sendo representado nas redes em um determinado instante t qualquer. Este estado de ativação do sistema pode ser representado por um vetor $\mathbf{a}(t)$. Os valores das ativações existentes na rede podem ser discretos, por exemplo assumindo os valores $\{0,1\}$ ou $\{-1,0,1\}$, como também podem ser contínuos, assumindo valores no intervalo $[0,1]$ ou $[-1,1]$, que são computados pela regra de ativação a ser vista posteriormente [LIP87] [GAL88].

2.2.3 Função de Saída

As unidades interagem entre si através de um valor que é transmitido pelas sinapses. Este valor é determinado pela ativação da unidade estimuladora. Formalmente, o valor de saída é dado por uma função do tipo $o_i(t) = g(a_i(t))$.

2.2.4 Padrão de Interconexão

Pode-se representar o padrão de interconexão da rede por uma matriz de pesos \mathbf{w} , onde o elemento w_{ij} corresponde à influência da célula u_j sobre a célula u_i . Conexões, também chamadas de sinapses, com pesos positivos, indicam o reforço na ativação de neurônio u_i . Estas conexões são chamadas de excitatórias. Sinapses com pesos negativos,

chamadas de inibitórias, indicam inibição na ativação da célula u_i . O conjunto das ligações excitatórias e inibitórias existentes na rede determina o comportamento da mesma.

Redes neurais artificiais podem ser vistas como grafos ponderados orientados, onde os neurônios artificiais são os nodos e os arcos orientados, com os seus respectivos pesos, são conexões entre os neurônios de saída e os neurônios de entrada.

Baseados no padrão de conexão, as redes neurais artificiais podem ser agrupadas em dois grandes grupos. O primeiro grupo é formado pelas redes neurais *feedforward*, onde os grafos orientados não possuem realimentação; o segundo grupo é formado pelas redes neurais recorrentes, também chamadas de redes neurais *feedback*, onde há conexões realimentando os neurônios.

Na maioria das redes neurais *feedforward*, os neurônios são organizados em camadas que possuem conexões unidirecionais entre elas. Além disto, estas redes são consideradas estáticas e desprovidas de traços de memória, com tempo de retenção finita, já que a resposta deste tipo de rede neural é independente do estado anterior da mesma.

Contudo, as redes neurais *feedback* são consideradas como sistemas dinâmicos e providas de traços de memória, com tempo de retenção diferente de zero, pois a ativação deste tipo de rede neural é dependente do estado anterior da mesma. Tal fato se deve à realimentação das conexões.

2.2.5 Regra de Propagação

Cada célula u_i computa sua nova ativação através de uma regra de propagação. Em geral, ela é definida como sendo uma função (**net**) da soma dos produtos das entradas pelos pesos das células u_j que estão diretamente conectadas a célula u_i conforme a equação (2.1):

$$net_i = F \left(\sum_{j=1}^n w_{ij} * u_j - \theta_i \right) \quad (2.1)$$

Onde: u_j é o estado da j-ésima unidade,

w_{ij} é o peso da conexão da j-ésima unidade para a i-ésima unidade, e

θ_i é o limiar da i-ésima unidade. Este limiar, que pode ser nulo inclusive, deve ser superado para que ocorra a ativação da célula.

Existem variações da regra de propagação que utilizam os conceitos de lógica nebulosa proposta por Zadeh [ZAD65].

2.2.6 Função de Ativação

O sinal de saída de um neurônio é calculado a partir da sua ativação, pela função de ativação.

Geralmente, esta função possui a forma $x_i = f(\text{net}_i)$, onde f é a função de ativação, também chamada de função limiar. Esta função mapeia os neurônios de entrada para um intervalo pré-especificado de saída. As quatro funções de ativação mais utilizadas são linear, rampa, salto e sigmóide [SIM90]. Uma ilustração das quatro é apresentada na Figura 2.2.

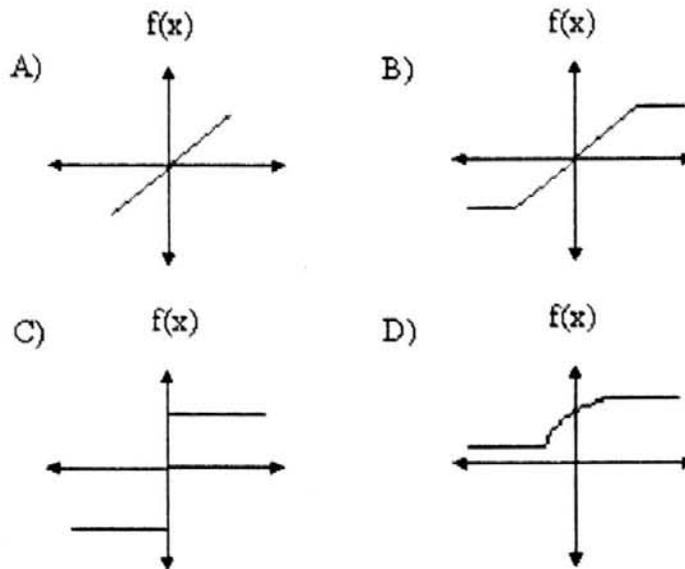


FIGURA 2.2 - Exemplos de funções de ativação.

A função linear, figura 2.2a, possui a seguinte equação: $f(x) = \alpha x$, onde α é uma constante real que regula a intensidade da atividade de x .

A figura 2.2b representa a função limiar rampa. Ela é uma função linear limitada pelo intervalo $[-y, +y]$, definida pelas seguintes equações (2.2):

$$f(x) = \begin{cases} +y & \text{se } x \geq y \\ x & \text{se } |x| < y \\ -y & \text{se } x \leq -y \end{cases} \quad (2.2)$$

Onde y indica os valores de saída máximo e mínimo, sendo chamados de pontos de saturação.

A função salto, figura 2.2c, responde ao sinal de entrada emitindo o valor $+1$ se o somatório for positivo e -1 nos demais casos.

A figura 2.2d representa a função sigmóide. Esta função limiar é monotônica, não-decrescente e sua resposta é gradual e não-linear. A função sigmóide mais conhecida é a função logística, cujos pontos de saturação são 0 e 1. É definida pela equação (2.3).

$$S(x) = (1 + e^{-x})^{-1} \quad (2.3)$$

2.2.7 Regra de Aprendizado

A modificação do processamento ou da estrutura de conhecimento de uma rede neural envolve modificar seu padrão de interconexão [RUM86]. Em princípio, isto pode ser feito de três maneiras:

- desenvolvimento de novas conexões;
- perda de conexões existentes na rede;
- modificação dos pesos das conexões já existentes.

Quando o padrão de interconexão for uma matriz de pesos \mathbf{w} , os dois primeiro itens podem ser simulados através do último. Tomando-se uma ligação com peso zero, e modificando-se para um valor positivo ou negativo, equivale a desenvolver esta sinapse. Da mesma forma, alterar o peso de uma conexão para zero significa desconectá-la.

As regras de aprendizado determinam como os pesos das sinapses das redes são alterados através da experiência.

Em geral, as regras de aprendizado podem ser consideradas como uma variante da regra de Hebb [HEB49]. Ele estabeleceu o princípio de que a alteração da eficiência sináptica é a base do aprendizado, segundo o postulado apresentado a seguir:

Quando o axônio da célula A se encontra próximo da célula B de forma a poder excitá-la, e o faz repetidas vezes, há um processo de crescimento de alguma troca metabólica em uma das células, ou em ambas, de tal modo que a eficiência de A, em desencadear o disparo de B, é incrementada..

Especificamente, se uma unidade \mathbf{u}_i recebe uma entrada de outra \mathbf{u}_j , e ambas estão fortemente ativas, o peso w_{ij} , de \mathbf{u}_j para \mathbf{u}_i , deve ser fortalecido. Uma implementação desta idéia é apresentada na equação 2.4 a seguir:

$$\delta w_{ij} = g(a_i(t), t_i(t)) \cdot h(o_j(t), w_{ij}) \quad (2.4)$$

Onde t_i é uma espécie de unidade de entrada instrutora (professor) da unidade \mathbf{u}_i . Esta equação estabelece que a mudança no peso da conexão \mathbf{u}_j para \mathbf{u}_i é resultado do produto da função $g()$ (da ativação de \mathbf{u}_i e de sua entrada instrutora t_i) pela função $h()$ (do valor de saída de \mathbf{u}_j e do peso w_{ij}).

Uma variação desta regra apresenta:

$$h(o_j(t), w_{ij}) = o_j(t) \quad e \quad g(a_i(t)) = \eta \cdot (t_i(t) - a_i(t)) \quad (2.5)$$

Onde η é uma constante de proporcionalidade que representa a taxa de aprendizado. Esta regra é chamada de regra delta, pois o aprendizado é proporcional a diferença, delta, entre a ativação realmente encontrada e a ativação fornecida pela instrutora. Esta regra é uma generalização da regra de convergência do *perceptron*, para o qual o teorema de convergência do *perceptron* foi provado.

2.2.8 Ambiente

O último componente de RNAs é o ambiente onde a rede deve funcionar. É necessário especificar a natureza do ambiente, estabelecendo os possíveis padrões de entrada e saída. Em alguns modelos, por exemplo, o PDP [RUM86], o ambiente é representado como uma função estocástica que varia ao longo do tempo sobre um espaço de padrões de entrada.

Geralmente, o ambiente é caracterizado como uma distribuição de probabilidade estável sobre um conjunto de padrões de entrada. Esta distribuição pode ser independente, ou não, de entradas ou de respostas passadas ao ambiente.

2.3 Técnicas de aprendizado em Redes Neurais

Nesta seção é feita uma análise sobre as técnicas de aprendizado utilizadas em redes neurais. A seguir são apresentados os conceitos de convergência e de estabilidade.

2.3.1 Formas de Aprendizado

Na sua essência o conceito de aprendizado envolve mudança associada ao aperfeiçoamento. Carbonell [CAR89] define o conceito de aprendizado, dentro do conceito de IA, como a habilidade de realizar tarefas novas que não podiam ser realizadas anteriormente, ou melhorar a realização de tarefas antigas, como resultado de mudanças produzidas pelo processo de aprendizado.

Uma classificação apresentada por Rumelhart e Zipser [RUM85], relacionou a função do aprendizado em modelos conexionistas. Eles distinguiram quatro paradigmas que são apresentados abaixo acompanhados de uma breve distinção:

- **Auto-associador** : Um conjunto de padrões é repetidamente apresentado e o sistema o armazena. Posteriormente, um padrão, ou parte de um padrão, semelhante aos originais é apresentado ao sistema e o mesmo retorna o padrão original. Este é um processo de auto-associação onde um padrão é associado consigo mesmo, de tal forma que uma versão modificada do original pode servir para o procedimento de recuperação;
- **Associador de padrões** : Inicialmente, um conjunto de pares de padrões é apresentado ao sistema. Posteriormente, quando um membro do par é apresentado, o sistema faz a associação, produzindo o outro elemento correspondente ao par;

- **Classificador** : Neste caso há um número fixo de categorias nas quais são classificados os padrões de entrada. O objetivo é ensinar a rede a classificar corretamente os padrões de entrada, de tal forma que, quando apresentado um padrão, mesmo parcialmente modificado, a rede saiba classificá-lo corretamente;
- **Detector de regularidades** : Existe uma população de padrões estímulo, sendo que cada membro possui uma probabilidade associada. O sistema desenvolve uma representação das características dos estímulos desta população, a qual captura as propriedades mais importantes dos padrões de entrada. Não existe, a priori, um conjunto de categorias no qual os padrões desta população possam ser classificados.

Posteriormente, Rumelhart [RUM86] apresentou uma outra classificação semelhante a esta apresentada acima. Neste caso existem dois tipos de aprendizado: o associativo e o detector de regularidades.

Nesta classificação, os três primeiros tipos apresentados acima foram agrupados sobre o rótulo de aprendizado associativo. Neste aprendizado, dois padrões a serem associados são apresentados, e a rede deve aprender a mapeá-los (no caso do auto-associador o objetivo é mapeá-lo em si mesmo). No caso do detector de regularidades, não existe, a priori, um conjunto de classes determinadas para separar os padrões de entrada.

A classificação conforme [RUM86] se assemelha a de [LIP87] que classifica as redes quanto ao tipo de controle realizado durante o aprendizado, supervisionado ou não-supervisionado. No supervisionado existem rótulos que classificam a classe correta para os padrões de entrada, instâncias, enquanto que no aprendizado não-supervisionado não existem tais rótulos. A classificação de [RUM85] e Zipser é mais específica do ponto de vista de como são processados, armazenados e recuperados, os padrões durante o treinamento das redes.

Uma descrição geral das diversas técnicas de aprendizado utilizadas nas RNAs, inspirada em [SIM90], é apresentada a seguir:

- **Correção de erros** : Aprendizado supervisionado que ajusta os pesos das conexões entre nós na proporção da diferença entre os valores desejados e computados de cada nó da camada de saída;
- **Reforço** : Aprendizado supervisionado onde os pesos são recompensados quando o sistema executa ações apropriadas e punidos caso ele não os execute;
- **Estocástico** : Aprendizado supervisionado que usa processos aleatórios, probabilidade e relações de energia para ajustar os pesos dos arcos;
- **Sistemas Hardwired** : As conexões e respectivos pesos são pré-determinados, semelhante a um autômato de estado finito;
- **Regra de Hebb** : Aprendizado onde os ajustes dos pesos das conexões é realizado em função da relação de valores dos dois nós que ele conecta. Pode ser aplicado tanto ao aprendizado supervisionado quanto ao aprendizado não-supervisionado.
- **Competitivo e Cooperativo** : Aprendizado não-supervisionado onde os processos competitivo e cooperativo são descritos em termos de redes com conexões recorrentes auto-excitáveis. Estes arcos podem ser inibitórios dos nós vizinhos, competitivo, e ou excitatórios dos vizinhos, cooperativo;

- **Sistemas conectados aleatoriamente (SCA)** : aprendizado não-supervisionado utilizado para suportar a teoria de que a mente é uma rede conectada aleatoriamente quando vista do nível macroscópico.

2.3.2 Dinâmica : Convergência e Estabilidade

Dois importantes conceitos regem a dinâmica das redes neurais: a convergência e a estabilidade global. Estes foram estudados recentemente por [JIM96]. Biologicamente, convergência é a formação de similaridades sucessivas entre os organismos ou associações distintas. Dentro do contexto dos modelos conexionistas, a convergência está relacionada com a minimização do erro, eventual, entre as saídas computadas e desejadas dos neurônios. Em geral, o conceito de convergência está associado ao aprendizado supervisionado.

Conceito de estabilidade está fortemente relacionado com a manutenção, ou retorno, a um estado de equilíbrio, sólido e estável. No contexto das RNAs, a estabilidade global é a estabilização de todas as ativações das unidades que compõem a rede, independente de qualquer entrada. Geralmente, o conceito de estabilidade está relacionado com a chamada recursiva existente nas redes recorrentes. Durante a execução de uma determinada tarefa, estas redes executam, eventualmente, vários ciclos de processamento até atingir um estado estável.

Quando se estuda a questão da convergência e da estabilidade em sistemas dinâmicos é comum se utilizar o conceito de atrator. Associado a cada RNA pode-se atribuir uma superfície de energia, de modo que, quando excitada com um padrão de atividade inicial, representada por um ponto sobre a referida superfície, a rede evoluirá dinamicamente, reduzindo a cada instante o valor de sua energia, tendendo para o ponto de menor energia mais próximo que possa atingir. O atrator seria justamente este ponto de mínimo da superfície. O ponto de mínimo mais baixo de toda a superfície é chamado de mínimo global. Os demais pontos de mínimos existentes na rede são chamados de mínimos locais. Uma representação de mínimo global é apresentado na figura 2.3.

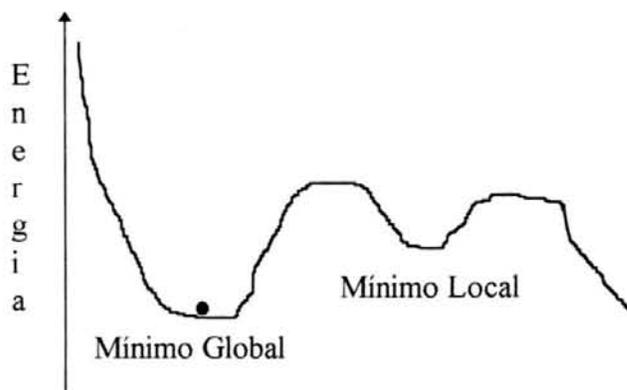


FIGURA 2.3 - Representação do mínimo global.

2.4 Modos de Implementação de Redes Neurais

Basicamente há dois modos de implementação de redes neurais artificiais: por *software* e por *hardware*.

O primeiro deles é obtido através da simulação, por meio de uma linguagem de programação, em um *hardware* sequencial, geralmente um computador de uso genérico. Através deste modo, consegue-se de uma forma mais ágil a implementação da rede neural para uma determinada aplicação. Contudo, devido à natureza serial da execução das instruções do programa, o desempenho da rede é mais lento que o segundo modo de implementação.

Como o tempo de desenvolvimento de um protótipo é mais curto, este modo é ideal para a realização de simulações de redes neurais discretas ou contínuas, esta última se dando através de métodos computacionais para a resolução de equações diferenciais que geralmente norteiam os modelos de redes neurais contínuas.

Tal modo de implementação garante ainda, uma precisão maior que o segundo modo de implementação, conforme [KOS92].

Na implementação por *hardware*, o tempo de desenvolvimento de um protótipo é maior, mas devido a sua alta taxa de processamento paralelo que pode ser conseguida, torna-se ideal para aplicações que envolvam o processamento de sinais em tempo real.

As redes neurais artificiais podem ser implementadas em *hardware* semi-dedicado ou dedicado. Para a implementação em *hardware* dedicado pode-se abordar a forma do projeto sob três focos distintos. O primeiro foco se utiliza de técnicas digitais de implementação, [HAS97], o segundo foco aborda as técnicas analógicas, como pode ser examinado em [MON97], [HSU96], [WAN95] e [KAN95]; e por fim, o terceiro foco se utiliza das técnicas híbridas, como em [HOL94] e em [MAS97], tendo a implementação composta em parte digital e em parte analógica.

O estilo de projeto utilizando técnicas analógicas é popular para a implementação de redes neurais, pois, através desta técnica se obtém circuitos compactos capazes de realizar um processamento assíncrono de alta velocidade.

3 Implementação de redes neurais em *hardware* analógico

3.1 Introdução

Neste capítulo examinaremos alguns conceitos básicos para a implementação analógica de redes neurais artificiais em *hardware*.

A unidade básica de processamento consiste de nodos (neurônios) com pesos associados (conexões sinápticas).

Na figura 3.1 pode-se observar um modelo de uma célula neural para computação baseada em software.

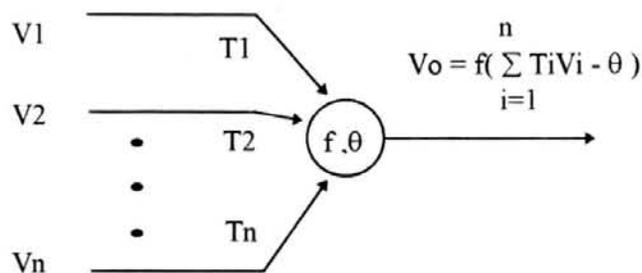


FIGURA 3.1 - Modelo neural para computação por software.

A computação analógica é o tipo encontrado em redes neurais biológicas, e também constitui a forma fundamental de processamento em redes neurais concebidas em *hardware*. Como se observa, o sistema biológico também opera em um modo analógico coletivo, com cada neurônio somando as entradas de centenas ou milhares de outros, produzindo assim uma determinada saída. Embora os circuitos analógicos sejam extremamente rápidos, a soma analógica é inevitavelmente menos precisa do que as somas digitais. Apesar deste compromisso não ser crítico, entretanto, tal fato terá que ser levado em conta quando se tratar de tarefas perceptuais [ZUR92a].

3.2 Conexões Sinápticas Analógicas

As redes neurais artificiais usualmente contêm um grande número de conexões sinápticas em comparação com os neurônios de processamento. No processamento neural analógico, o conhecimento é encapsulado na forma de um conjunto de pesos analógicos. A rede aprende via mudanças graduais nestes pesos quando a rede está no modo de aprendizagem. A recuperação de um dado é conseguida através de sucessivas computações analógicas em todos os nodos da matriz. Em uma rede neural simples, os pesos podem ser determinados por valores de resistência. A computação analógica do

produto escalar e posterior mapeamento não linear pode então ser realizado, por exemplo, por um amplificador somador com saturação ou por um amplificador diferencial, sendo que esse último foi utilizado nas simulações realizadas neste trabalho e será alvo de maiores explicações no decorrer do trabalho.

Na figura 3.2, apresenta-se o mesmo neurônio representado na figura 3.1, porém implementado em *hardware*. Um amplificador diferencial pode implementar a função de ativação dos nodos. As resistências implementam os pesos sinápticos.

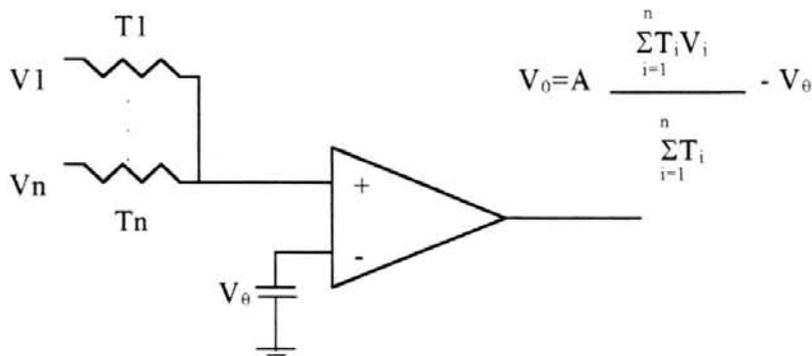


FIGURA 3.2 - Implementação de um neurônio utilizando resistores e amplificador diferencial.

É importante observar que a tensão de entrada do neurônio da figura 3.2 é escalada por um fator que significa o somatório das condutâncias conectadas ao neurônio em questão. Para manter a operação normal do modelo, deve-se fornecer um ganho do mesmo fator que está sendo diminuído no escalamento. A tensão de limiar é causada por diferenças dos componentes utilizados e é muito difícil de se controlar.

O circuito baseado em amplificador diferencial e resistores pode produzir tanto pesos positivos quanto negativos. Este circuito ilustra os conceitos de projetos básicos de redes neurais analógicas. Entretanto, esses circuitos possuem uma limitada praticidade por diversas razões. Entre estas limitações está a forma das suas funções características, excessivo consumo de potência, e a grande área ocupada pelos resistores discretos e amplificadores operacionais. Outra limitação está nos valores dos pesos estarem encapsulados em resistores fixos, o qual não é fácil de se controlar nem de se ajustar. Como consequência, eles não podem ser utilizados para tarefas distintas das que foram projetadas.

3.3 Pesos Controlados por Tensão

Como se pode notar, o problema central na implementação de redes neurais artificiais é a realização de pesos sinápticos que são continuamente ajustáveis, preferencialmente seguindo a estímulos dados por um controle de sinais analógicos. Estes pesos sinápticos não requerem muitos transistores para a implementação ou para o controle de seus valores. Além disto, os pesos sinápticos também deveriam ser capazes de se modificar conforme uma determinada regra de aprendizado.

Estas considerações nos levam a duas noções diferentes: pesos ajustáveis que podem ser modificados somente ao fim do processo de aprendizado; e a noção de aprendizado de pesos, que é entendido como sendo pesos que se modificam durante o processo de aprendizado, em resposta a um sinal fornecido por um circuito de ajuste de pesos [ZUR92a].

Conexões sinápticas ajustáveis através de controle digital ou analógico podem ser implementadas em diferentes versões de circuitos integrados, onde uma dessas formas pode ser analisada na figura 3.3. Tal circuito foi utilizado neste trabalho.

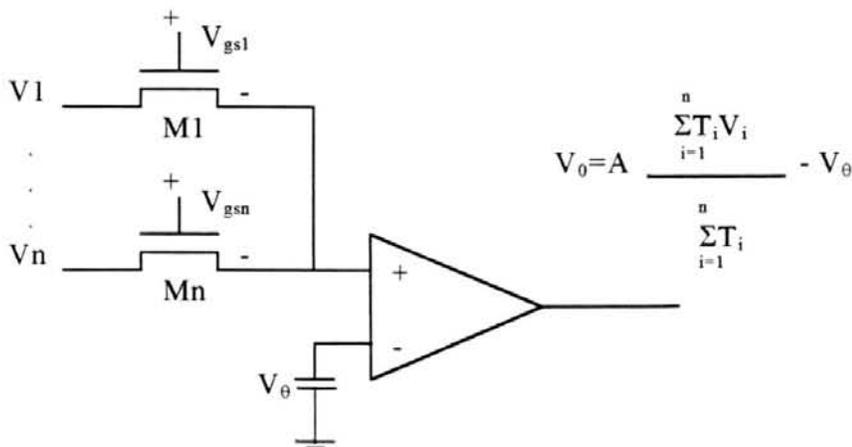


FIGURA 3.3 - Implementação de um neurônio utilizando pesos ajustáveis e amplificador diferencial.

Observa-se que os resistores controlados por tensão, podem ser obtidos de transistores MOS operando na região resistiva. Na figura 3.3 acima, é ilustrado um neurônio com “n” conexões sinápticas, cada uma delas consistindo de transistores MOS simples operando como uma chave analógica controlada por tensão. Cada resistência de canal do transistor, R_{ds} , entre o dreno e o fonte, é controlada pela tensão existente entre os terminais porta e fonte, V_{gs} .

Uma vantagem desta configuração simples é que o valor de R_{ds} pode ser escalado pela alteração da relação entre o comprimento e a largura do canal do transistor. Para grandes valores dessa relação, a resistência do canal é escalada para grandes valores e baixo consumo de potência.

3.4 Circuitos Ativos para Implementação de Redes Neurais

Além das conexões sinápticas acima descritas, um circuito de rede neural artificial é composto de um importante bloco para o processamento da informação. Este serve para realizar funções do tipo soma, subtração, produto escalar ou mapeamento não-linear. Um grande número de circuitos eletrônicos tais como espelhos de corrente, inversores, amplificadores diferenciais, amplificadores de transcondutância, multiplicadores analógicos e circuitos para produto escalar podem realizar estas operações.

Circuitos integrados capazes de implementar algumas destas funções e implementados utilizando tecnologia CMOS, serão abordados nesta seção.

3.4.1 Neurônio Baseado em Inversor

O circuito inversor CMOS é mostrado na figura 3.4, e consiste de transistores complementares do tipo NMOS e PMOS com seus terminais de dreno conectados juntos e também atuando como terminal de saída. Os terminais de porta, de ambos, são também conectados juntos, e os terminais de fonte são alimentados por duas fontes de tensão opostas em sinais. O circuito comporta-se como um amplificador de tensão inversor com considerável ganho de tensão.

Como se pode observar na figura 3.4, este circuito implementa um neurônio com somente uma saída do tipo inversora. Para conseguirmos, uma saída inversora e outra saída não-inversora, há a necessidade de conectarmos outro circuito idêntico, em cascata, de modo a obtermos ambas as saídas comentadas anteriormente.

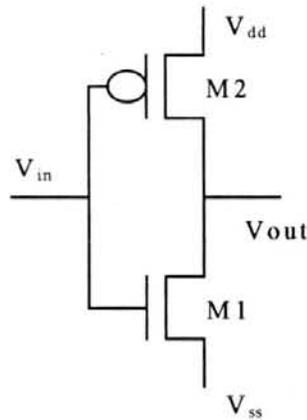


FIGURA 3.4 - Diagrama do circuito do neurônio baseado em inversor.

Para derivar a função de ativação para este tipo de neurônio, assume-se que ambos os transistores possuam as mesmas condições construtivas, e também requerem que ambas as tensões de limiar sejam da mesma magnitude. A dedução desta função se encontra em [ZUR92] e seu resultado final é mostrado na figura 3.5.

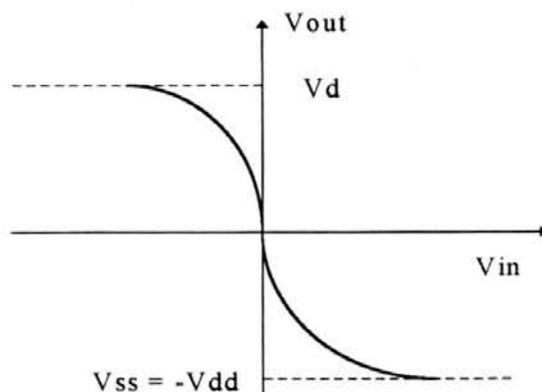


FIGURA 3.5 - Função de ativação característica de um neurônio baseado em inversor.

Esta configuração de neurônio é especialmente empregada na implementação de memórias associativas. A figura 3.6 ilustra um exemplo de um circuito de uma memória autoassociativa recorrente, de uma única camada e com “n” entradas, usando como neurônio dois inversores. Devido à disponibilidade de ambos os tipos de saída nos neurônios, saída invertida e não-invertida, pesos negativos e positivos podem ser implementados neste circuito. A matriz de pesos da memória contém pesos implementados através de transistores controlados por seus terminais porta. Os transistores deste circuito necessitam estar operando na região linear para manterem os valores dos pesos constantes.

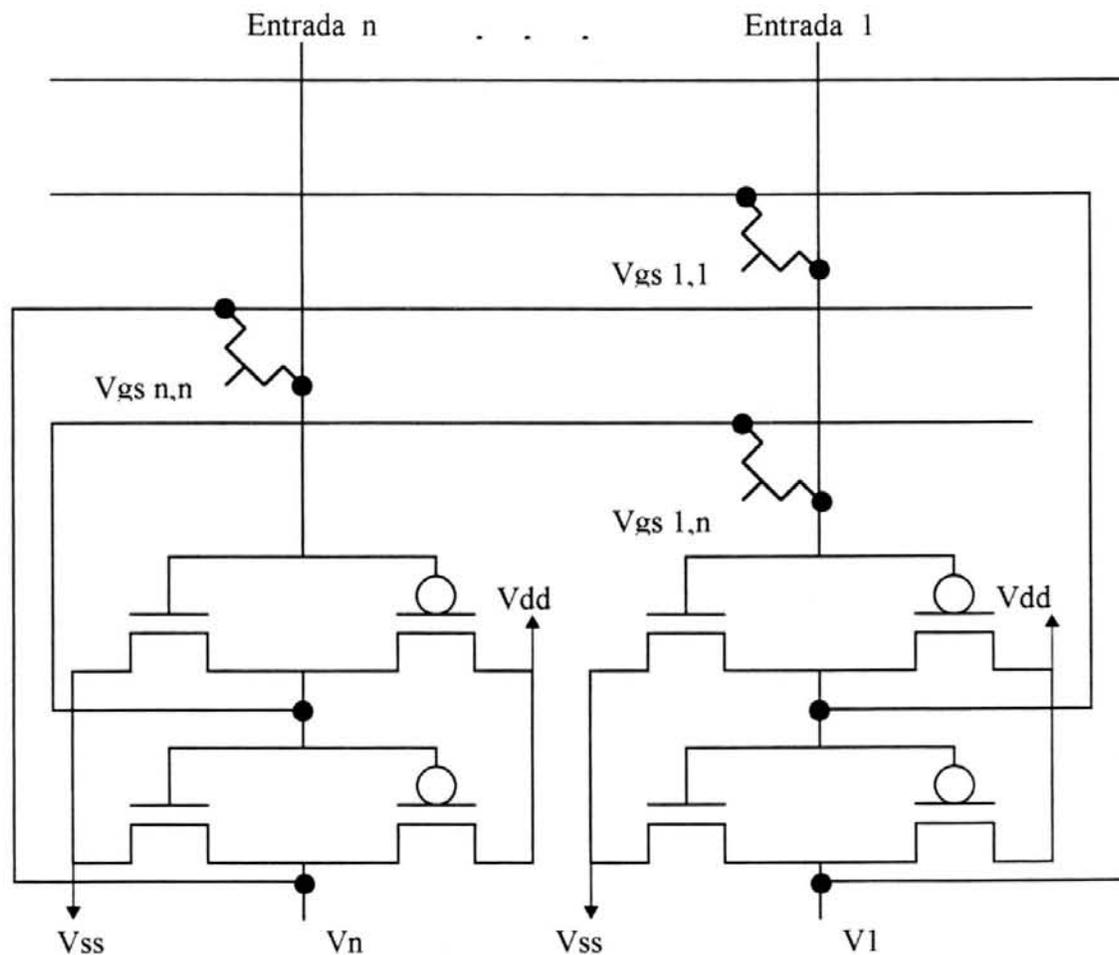


FIGURA 3.6 - Memória autoassociativa recorrente baseado em neurônio do tipo inversor.

3.4.2 Amplificadores de Tensão Diferencial

Um amplificador operacional é essencialmente um amplificador de tensão diferencial de ganho de tensão de saída muito alto. Contudo, o amplificador de tensão diferencial pode operar sem a necessidade de uma realimentação negativa, que teria que ser empregada caso fosse utilizado um amplificador operacional operando como neurônio. Seu ganho

de tensão é tipicamente uma centena ou mais vezes mais baixo que o ganho apresentado pelo amplificador operacional.

A figura 3.7 mostra um circuito esquemático de um amplificador de tensão diferencial com tecnologia NMOS, denominado resumidamente de par diferencial. O circuito consiste de um par de transistores NMOS, M1 e M2, duas cargas possivelmente simétricas e uma fonte de corrente para referência, I_{ss} .

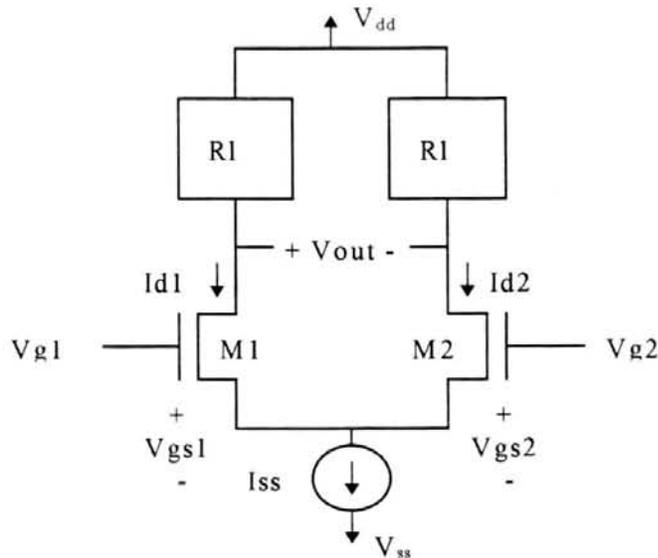


FIGURA 3.7 - Esquema básico de um amplificador de tensão diferencial.

Como definido em [ZUR92], a tensão diferencial, V_{ind} , é como segue:

$$V_{ind} = V_{gs1} - V_{gs2} \quad (3.5)$$

A característica da função de ativação de saída se comporta como uma função sigmoideal. Tal pode ser examinado na figura 3.8.

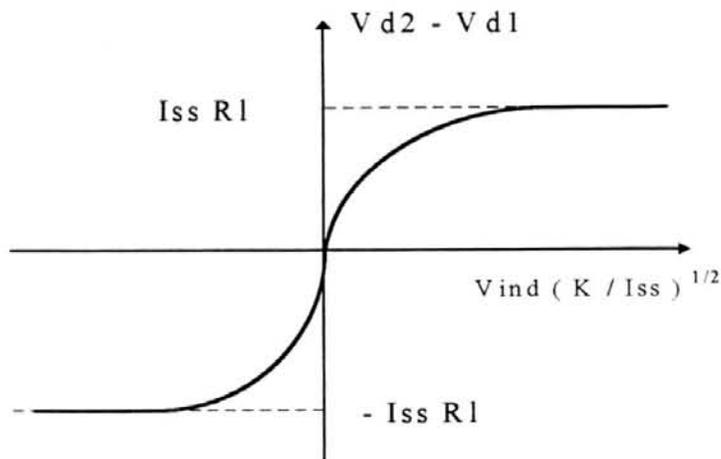


FIGURA 3.8 - Função de ativação característica de um neurônio baseado em amplificador de tensão diferencial.

É importante salientar que a faixa abrangida por esta configuração está limitada pelos valores $(-I_{ss} \cdot R_l)$ e $(I_{ss} \cdot R_l)$, onde R_l é o valor da carga conectada ao circuito.

3.4.3 Amplificadores de Transcondutância

Para se evitar a configuração de cargas apresentadas na seção anterior, um espelho de corrente pode ser utilizado para formar a carga necessária ao circuito. A vantagem de tal configuração é que o sinal de saída diferencial na forma de correntes de dreno no circuito da figura 3.7, é convertida a uma simples saída com a carga aterrada em um terminal.

A configuração do circuito implementando este conceito é mostrado na figura 3.9. Nesse circuito, a tensão de saída ou corrente é fornecida pela ligação entre os transistores M2 e M4. Esse circuito é também conhecido como amplificador de transcondutância.

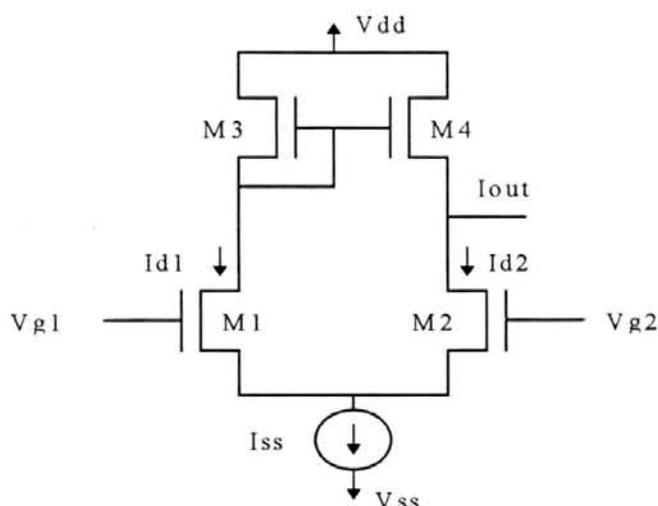


FIGURA 3.9 - Diagrama do circuito de um amplificador de transcondutância.

Um modo de implementação analógica de redes neurais artificiais utilizando os amplificadores de transcondutância citados acima, foi demonstrado por [REE89]. Esse trabalho mostra o potencial do uso de circuitos com amplificadores operacionais de transcondutância para a construção de neurônios artificiais e de pesos sinápticos.

4 Redes Competitivas - Modelos Contínuos

4.1 Introdução

Neste capítulo são abordados alguns conceitos sobre modelos contínuos de redes neurais competitivas.

Primeiramente é abordado o modelo de Hopfield. A seguir é descrita a chamada equação da membrana descrita por Hodgkin e Huxley. Após, são apresentados os modelos que derivam das redes competitivas, tais como as redes *shunting feedforward* e as redes *shunting feedback*. Por fim, são apresentados os resultados de simulações computacionais realizadas para a verificação das equações aqui apresentadas.

4.2 O modelo de Hopfield

Antes de examinarmos o modelo de rede neural sugerido por Hopfield, explanaremos sobre o conceito de memória associativa. Esse conceito é razoavelmente intuitivo. Veja o seguinte exemplo: uma pessoa consegue facilmente associar a face de um amigo com o seu respectivo nome, ou um nome com um número de telefone.

Muitos dispositivos possuem características de memória associativa. Por exemplo, o banco de memória de um computador é um tipo de memória associativa; este associa endereços com dados. Porém, esse tipo de memória associativa possui uma limitação; caso haja um erro no cálculo do endereço ou se houver um pequeno ruído no endereço, o dado a ser lido será completamente diferente do esperado. Como alternativa, há um tipo de memória que endereça o conteúdo armazenado pelo próprio conteúdo, isto implica que, o conteúdo da memória pode ser recuperado, lido, sempre por uma entrada parcial ou distorcida do próprio conteúdo.

Esta característica é extremamente desejada, e pode ser utilizada para reconhecimento de padrões e reconstituição de sinais, dentre muitas outras utilidades.

Dentre as topologias de redes neurais artificiais, o modelo proposto por Hopfield [HOP84] preenche as características de uma memória associativa e, além disto, esse modelo também é utilizado para resolver problemas de otimização.

Este modelo surgiu utilizando uma rede onde se dá destaque ao comportamento coletivo, estando os neurônios totalmente interconectados entre si, mas não estando conectado a si próprio. A figura 4.1 apresenta uma topologia de uma rede de quatro neurônios interligados conforme o modelo sugerido por Hopfield.

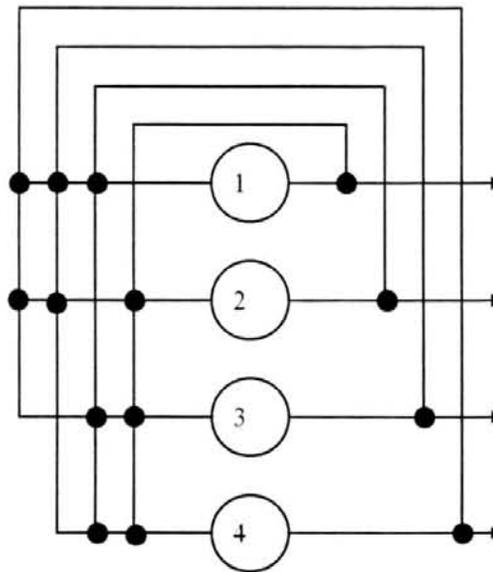


FIGURA 4.1 - Modelo sugerido por Hopfield.

Há duas versões para o modelo: a memória discreta e a memória contínua; a diferença entre elas está em se os valores de saída são uma função discreta ou contínua das entradas, respectivamente.

O modelo de memória contínua é uma extensão do modelo de memória discreto, pela incorporação de alguns resultados de neurobiologia que fez com que os neurônios do modelo representassem de melhor forma um neurônio real. Por exemplo, é conhecido que neurônios reais tem respostas contínuas como função de suas entradas, e não somente dois estados, como na saída binária. Por essa e outras modificações, Hopfield construiu o modelo de memória contínua, o qual possui as mesmas propriedades de uma memória associativa que o modelo discreto apresentou anteriormente. Além disso, há uma analogia com um circuito eletrônico utilizando amplificadores operacionais e resistores, o qual sugere a possibilidade de implementação desse modelo utilizando a tecnologia VLSI [MOL96].

Na estrutura da rede mostrada na figura 4.1 observa-se as três grandes formas de organizações paralelas encontradas em sistemas neurais: canais de entrada paralelos, canais de saída paralelos, e uma grande quantidade de interconexões entre os elementos de processamento neural. Os elementos de processamento, ou usualmente denominados de neurônios, podem ser modelados como amplificadores em conjunto com circuitos que realimentam os sinais, sendo estes formados por fios, resistores e capacitores organizados desta forma para modelar as características computacionais mais básicas dos neurônios, denominados axônios, arborização dendrítica, e sinapses, conectando os diferentes neurônios. Tal configuração pode ser vista na figura 4.2.

O amplificador possui uma relação sigmoideal entre a entrada e a saída, como mostrada na figura 4.3. A função $V_i = g(u_i)$ a qual caracteriza sua relação entre a entrada e a saída, descreve a tensão de saída do amplificador V_i devido a uma tensão de entrada u_i . As constantes de tempo dos amplificadores são desprezadas. Entretanto, assim como a

impedância de entrada associada à membrana celular em um neurônio biológico, cada amplificador i possui um resistor ρ_i na entrada e um capacitor C_i .

Estes componentes, parcialmente definem as constantes de tempo dos neurônios e realizam a operação de integração nas somas analógicas das correntes sinápticas de entradas dos outros neurônios na rede.

Para fornecer ambas as conexões sinápticas, excitatórias e inibitórias, entre os neurônios quando utilizando componentes elétricos convencionais, cada amplificador é dotado de duas saídas, uma normal e uma invertida. As saídas máxima e mínima do amplificador normal, não invertido, é tida como sendo compreendida pela faixa entre 0 (zero) e 1 (um), enquanto que a saída invertida tem a faixa de valores compreendidas pelos valores 0 (zero) e -1(menos um). Tal representação também pode ser observada na figura 4.2.

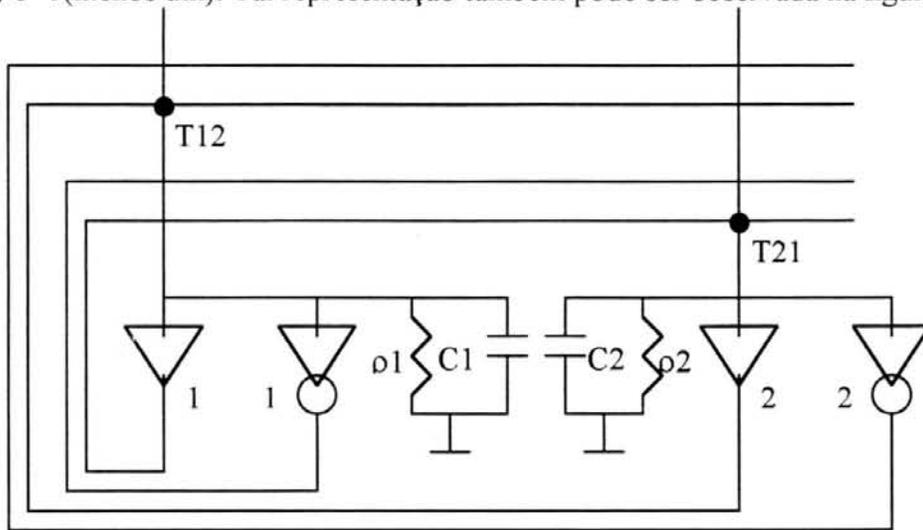


FIGURA 4.2 - Circuito eletrônico para o modelo de Hopfield.

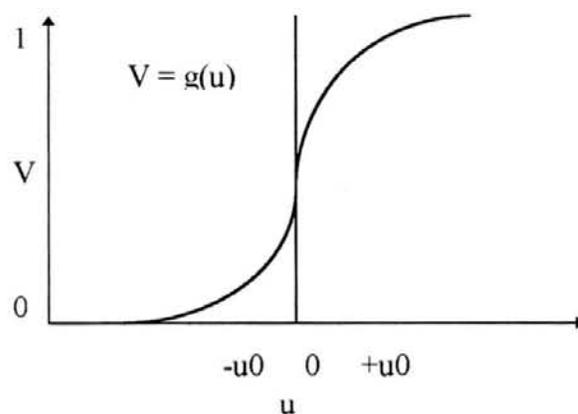


FIGURA 4.3 - Relação entre a saída e a entrada de um neurônio.

O peso sináptico entre dois neurônios é definido por uma condutância, T_{ij} , o qual conecta uma das duas saídas do amplificador j à entrada do amplificador i . Esta conexão é realizada por um resistor com um valor de aproximadamente:

$$R_{ij} = 1 / | T_{ij} | \quad (4.1)$$

Se a sinapse é excitatória, $T_{ij} > 0$, este resistor é conectado à saída normal do amplificador i . Para uma sinapse inibitória, $T_{ij} < 0$, este é conectado à saída invertida do amplificador i . A matriz, T_{ij} , define a conectividade entre os neurônios. A corrente de entrada para qualquer neurônio, e consequentemente a tensão de entrada u_i , é a soma das correntes fluindo através do conjunto de resistores conectando suas entradas às saídas dos outros neurônios. Então, a saída normal e a invertida para cada neurônio permitem a construção de ambas as conexões excitatória e inibitória usando resistores. Tal implementação corresponde àquela encontrada nos neurônios biológicos, que não requerem uma saída normal e outra invertida, pois as diferentes sinapses são realizadas pelo uso de diferentes combinações químicas [MEA89].

Como mostrado na figura 4.1, o circuito inclui uma entrada externa para cada neurônio, simbolizada pelas letras I_i . Estas entradas podem ser usadas para ajustar um nível geral de excitabilidade da rede através de constantes denominadas de *bias*, que efetivamente deslocam a relação entre a saída e a entrada ao longo do eixo u_i , ou para fornecer entradas paralelas para acionar neurônios específicos.

A equação extraída pela aplicação direta da lei das correntes de Kirchhoff, que descreve o comportamento do circuito através do tempo, é conforme [HOP85]:

$$C_i (du_i / dt) = \sum_{j=1}^n T_{ij} V_j - u_i / R_i + I_i \quad (4.2)$$

onde: $V_j = g_j (u_j)$,

R_i é a combinação paralela de ρ_i e as R_{ij} , e é calculada como sendo:

$$1 / R_i = 1 / \rho_i + \sum_{j=1}^n 1 / R_{ij} \quad (4.3)$$

Para um problema com valores iniciais definidos, no qual as tensões de entrada (u_i) são dadas no tempo zero, a equação (4.2) fornece uma completa descrição da evolução no tempo dos estados do circuito.

Neste tipo de rede neural, foi demonstrado por [HOP85], que a equação (4.2) para uma rede com conexões simétricas ($T_{ij} = T_{ji}$) sempre convergirá para estados estáveis, onde as saídas de todos os neurônios permanecem constantes. Também, quando a largura da curva de ganho do transistor, figura 4.2, é estreita, os estados estáveis da rede composta de N neurônios correspondem aos mínimos locais calculados através da seguinte equação de energia:

$$E = - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N T_{ij} V_i V_j - \sum_{i=1}^N V_i I_i \quad (4.4)$$

O espaço dos estados sobre o qual o circuito opera é o interior de um hipercubo de N dimensões definido por $V_i = 0$ ou 1 . Entretanto, em um limite com alto ganho, os mínimos ocorrem nos vértices deste espaço. Com isso, os estados estáveis da rede correspondem a estas localizações discretas no espaço, formado por 2^N vértices deste hipercubo, o qual minimiza a equação (4.4).

Foi mostrado no trabalho de [HOP84] que o circuito acima mencionado opera como uma memória endereçável por conteúdo (CAM - Content Addressable Memory). As saídas normais dos N amplificadores compõem o circuito de memória, no qual, para estas aplicações, foram permitidas a faixa de valores de -1 a $+1$, em vez de valores compreendidos entre 0 e $+1$.

Um dado armazenado nesta rede pode ser recuperado pelo ajuste das entradas dos amplificadores, e permitindo que a rede convirja para um estado estável. Este estado estável é interpretado como o dado recuperado.

Como foi apresentado por [HOP85], foi mostrado também que a energia será mínima quando houver todos $I_i = 0$ e T_{ij} definidos como sendo:

$$T_{ij} = \sum_{s=1}^m V_i^s V_j^s \quad (4.5)$$

onde: V^s é um conjunto particular de m estados de memória, e $s = \{1, 2, \dots, m\}$.

Através da equação (4.5), pode-se então calcular os valores das transcondutâncias e, posteriormente, os valores das resistências para que determinado conjunto de padrões possa ser armazenado adequadamente na rede neural.

4.3 Equação da Membrana

A unidade básica no sistema nervoso é uma célula especializada chamada de neurônio. Na figura abaixo temos a forma básica deste neurônio. Os neurônios possuem extensões longas e filamentosas que são denominadas de processos [MEA89]. Cada neurônio possui uma série de características, dentre as quais citamos:

1. O sistema metabólico da célula fornece uma fonte de tensão para funções de processamento de informações. Além disso, a célula reforça uma outra célula através de mecanismos bioquímicos.
2. Uma árvore de processos chamada de dendritos é coberta com estruturas especiais chamadas de sinapses, onde as junções são formadas com outros neurônios. Estes contatos sinápticos são os primeiros elementos do processamento da informação em sistemas neurais.
3. Os processos atuam como fios, conduzindo a informação sobre uma extensão espacial finita.
4. Integração temporal de sinais ocorre sobre um curto tempo, através do armazenamento de carga na capacitância da membrana celular, e sobre um longo tempo, pelo meio de complexos mecanismos bioquímicos.

5. Certos neurônios são equipados com um processo longo e especializado, denominado de axônio. O axônio é utilizado para a transmissão de dados sobre longas distâncias.

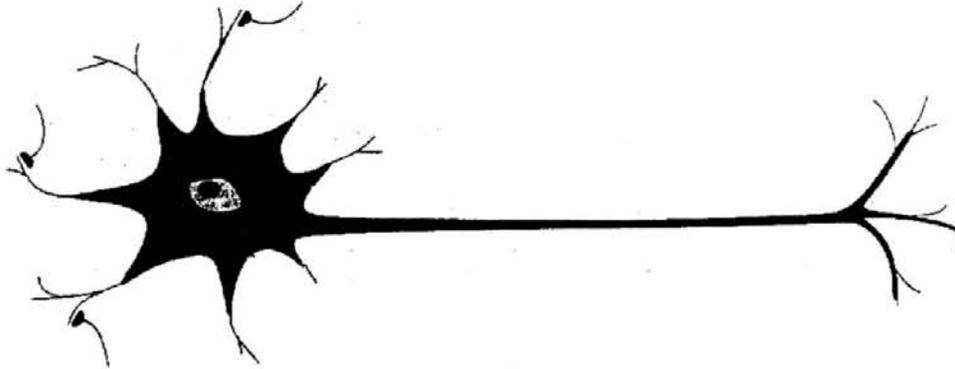


FIGURA 4.4 - Um neurônio natural

O neurônio clássico é equipado com uma árvore de dendritos filamentosos que agregam entradas sinápticas de outros neurônios. Tal aspecto pode ser observado na figura 4.4. As correntes de entrada são integradas pela capacitância da célula até que um potencial limiar seja alcançado. Neste ponto, uma saída é gerada na forma de um pulso nervoso, denominado de potencial de ação. Este pulso nervoso de saída é propagado pelo axônio, o qual finaliza em uma árvore de contatos sinápticos de outros neurônios.

Toda atividade elétrica em um neurônio está concentrada em uma fina membrana que separa eletricamente o interior do neurônio do fluido extracelular. Esta membrana pode ser considerada como um perfeito isolador elétrico. Mas, para haver o processamento da informação deve existir uma fonte de tensão. Essa fonte é obtida através de agentes de transferência básicos existentes em todas as membranas nervosas, que acionam metabolicamente o bombeamento que ativamente expelle íons de sódio do citoplasma e, concomitantemente, importam íons de potássio do fluido extracelular. Como resultado deste processo de bombeamento, o citoplasma é enriquecido em potássio e empobrecido de sódio, enquanto o reverso é verdadeiro para o fluido extracelular. A concentração de íons relevantes dentro e fora de uma célula nervosa é mostrada por [MEA89]. Os íons mais importantes a se considerar são os íons de potássio, de sódio e de cloro. Um gradiente de concentração de qualquer partícula carregada pode ser utilizada para a atividade elétrica da célula.

Um diagrama que resume as contribuições dos três gradientes iônicos para a corrente na membrana é mostrada na figura 4.5, onde as fontes de tensão representam os potenciais das partículas iônicas; as condutâncias representam a permeabilidade da membrana para o referido íon. A capacitância da membrana é ilustrada como um capacitor.

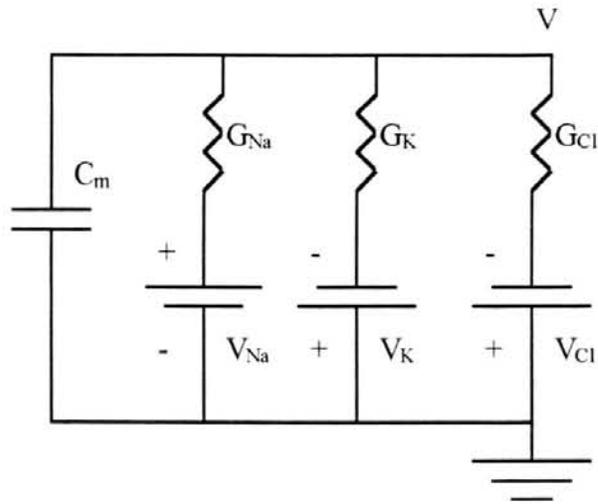


FIGURA 4.5 - Circuito equivalente da membrana nervosa.

A equação da membrana que descreve a voltagem $V(t)$ de uma célula é [HOD52]:

$$\frac{\partial V}{\partial t} * C_m = (V_K - V) * G_K + (V_{Na} - V) * G_{Na} + (V_{Cl} - V) * G_{Cl} \quad (4.6)$$

Qualquer corrente descarregará ou carregará a capacitância da membrana até a corrente ser reduzida a zero. Conforme descrito por [HOD52], a corrente referente ao íon de cloro pode ser desprezada. Rearranjando a equação (4.6) e resolvendo para V_0 onde a corrente é igual a zero, temos:

$$V_0 = (V_K * G_K + V_{Na} * G_{Na}) / (G_K + G_{Na}) \quad (4.7)$$

A variável V_0 é denominada de potencial residual do citoplasma. Em um neurônio típico, G_K é aproximadamente vinte vezes maior que G_{Na} . Usando estes valores e a tabela demonstrada por [MEA89], o valor de V_0 é de -85 milivolts. O potencial residual pode variar consideravelmente de um conjunto de condições experimentais para outro.

Um neurônio é normalmente polarizado a um residual de potencial negativo, pois sua membrana é seletivamente permeável ao potássio. Com isto, se um pulso nervoso realiza uma excursão do transiente do potencial citoplasmático, na direção positiva, teremos um exemplo de um sinal excitatório, pois despolariza a membrana. Porém, se a membrana é carregada mais negativamente que o potencial residual, então esta está hiperpolarizada; neste caso temos um exemplo de sinal inibitório.

4.4 Derivações de Redes Neurais Competitivas

Nesta seção abordaremos as redes do tipo *shunting*. Tais redes possuem habilidades vantajosas no tratamento dos problemas que tem confrontado pesquisadores que utilizam redes do tipo aditivo [GRO88].

Uma das características básicas dos modelos de redes neurais competitivas do tipo *shunting* é a propriedade da normalização da atividade total que é tida como sendo um controle global na dinâmica da rede, que previne atividades muito baixas ou muito altas [GRO88].

As redes neurais competitivas do tipo *shunting* possuem propriedades de controle de ganho automático capazes de gerar uma grande faixa dinâmica, através das quais os padrões de entrada podem ser efetivamente processados, com isso, solucionando o dilema ruído-saturação. Redes especializadas do tipo *shunting* têm sido classificadas em termos dos seus específicos padrões de processamento e propriedades de armazenamento como em memórias, fornecendo uma variedade de redes, as quais servem como um recurso para a solução de problemas computacionais particulares.

As redes do tipo *shunting feedforward* e *feedback* receberam tais nomes devido a técnica implementada para a realização da competição entre os neurônios pertencentes a camada competitiva. Esta técnica é chamada de reforço central e inibição lateral, do inglês, “*on-center off-surround*”.

4.4.1 Redes *Shunting Feedforward*

Como dito acima, a rede neural competitiva do tipo *shunting feedforward* possui uma técnica de competição entre os neurônios pertencentes à camada competitiva, denominada de reforço central e inibição lateral. Tal técnica, demonstrada na figura 4.6, é definida pelos sinais de entrada aplicados à camada competitiva, onde, estes sinais de entrada, definem as interações de reforço central e inibição lateral.

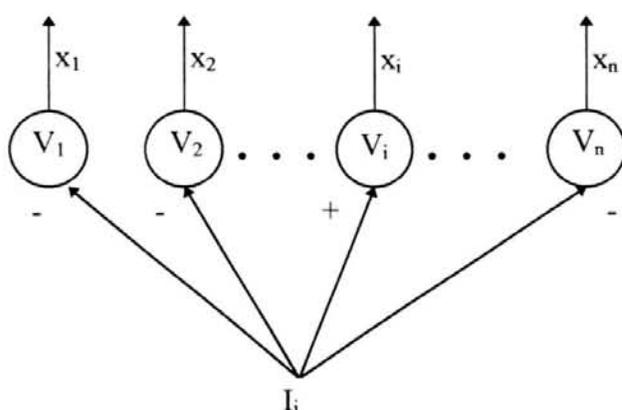


FIGURA 4.6 -Camada Competitiva da Rede Neural do tipo *Shunting Feedforward*

Esta camada competitiva possui n elementos de processamento, $\{V_1, V_2, \dots, V_n\}$. Cada valor de entrada, I_i , é conectado com uma conexão excitatória (seta com sinal positivo) a seu correspondente elemento de processamento, V_i . Cada I_i está conectado, também, a cada um dos outros elementos de processamento, V_k , com $k \neq i$, através de uma conexão inibitória (seta com sinal negativo).

Portanto, há um valor de entrada, I_i , para cada uma das n unidades na camada. A intensidade do padrão de entrada é dado por :

$$I = \sum_i I_i \quad (4.8)$$

A parte que corresponde a cada I_i , é definida como a seguinte quantidade :

$$\theta = I_i (\sum_i I_i)^{-1} \quad (4.9)$$

O vetor, $(\theta_1, \theta_2, \dots, \theta_n)^t$, é chamado de padrão de ponderação. Observa-se que este padrão é normalizado e que $\sum_i \theta_i = 1$.

O padrão de ponderação é independente da intensidade total do padrão correspondente. Para o processamento de informações visuais, por exemplo, o padrão de ponderação correspondente à imagem da face de uma pessoa é independente se a pessoa está sendo visualizada sob a luz solar ou sob a sombra. Este exemplo sugere que nossa memória armazena e recupera padrões de ponderação.

As saídas dos elementos de processamento na figura 4.6, são governados pela seguinte equação diferencial:

$$\frac{dx_i}{dt} = -A*x_i + (B - x_i)*I_i - x_i*\sum_{k \neq i} I_k \quad (4.10)$$

onde: $0 < x_i(0) < B$ e
 A e $B > 0$.

Cada elemento de processamento recebe um sinal de excitação que é igual a $(B - x_i)*I_i$ de seu valor de entrada, I_i , correspondente. A adição das conexões inibitórias, $-x_i*I_k$, provindo das outras unidades, é responsável pela prevenção do crescimento exagerado da intensidade do padrão I_i .

Após um padrão ser aplicado à rede, os elementos de processamento rapidamente alcançam um estado de equilíbrio [FRE93]. Quando o padrão de entrada for removido, as atividades das unidades não permanecem em seus valores de equilíbrio, nem retornam imediatamente a zero. As atividades dos neurônios persistem por algum tempo, enquanto o termo $-A*x_i$ reduz as atividades gradualmente ao valor zero.

A equação (4.10) é um caso especial da equação da membrana que foi descrita na seção 4.3 do presente capítulo. Tal demonstração está comentada em [GRO88].

4.4.2 Redes *Shunting Feedback*

Como foi comentado no início deste capítulo, a rede neural competitiva do tipo *shunting feedback* possui uma técnica de competição entre os neurônios pertencentes a camada competitiva denominada de reforço central e inibição lateral. Tal técnica é demonstrada na figura 4.7.

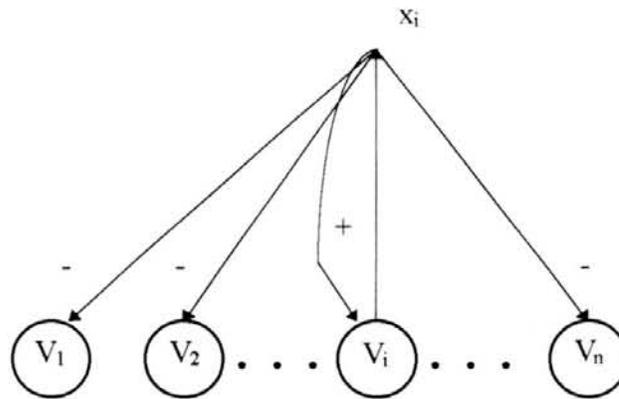


FIGURA 4.7 -Camada Competitiva da Rede Neural do tipo *Shunting Feedback*

Conforme pode ser observado na figura 4.7, cada unidade recebe um sinal de realimentação positiva, sinal excitatório, de si mesma, e envia um sinal inibitório para todas as outras unidades pertencentes a camada competitiva. A unidade cujo vetor de pesos que mais combina com o vetor de entrada apresentado, envia sinais inibitórios mais fortes às outras unidades e também recebe sinais excitatórios mais fortes de si mesma, realizando assim a competição entre as unidades pertencentes a camada competitiva.

As saídas dos elementos de processamento na figura 4.7 podem ser calculados pela seguinte equação diferencial:

$$\frac{dx_i}{dt} = -A*x_i + (B - x_i)*[f(x_i) + I_i] - x_i*\left[\sum_{k \neq i} f(x_k) + \sum_{k \neq i} I_k\right] \quad (4.11)$$

onde: A e $B > 0$;

I_i e I_k são os padrões aplicados a entrada das respectivas unidades;

$f(x_i)$ e $f(x_k)$ são as funções de ativação das respectivas unidades.

Observa-se que a equação (4.11) não apresenta pesos associados às entradas da rede neural. Se considerarmos estes pesos, a equação diferencial que rege o comportamento através do tempo é dada por:

$$\frac{dx_i}{dt} = -A*x_i + (B - x_i)*[f(x_i) + net_i] - x_i*\left[\sum_{k \neq i} f(x_k) + \sum_{k \neq i} net_k\right] \quad (4.12)$$

onde: A e $B > 0$;

net_i e net_k são os somatórios dos produtos entre os valores dos pesos e sinais de entrada para as respectivas unidades;

$f(x_i)$ e $f(x_k)$ são as funções de ativação das respectivas unidades.

A este tipo de rede neural com pesos associados, dá-se o nome de rede neural competitiva do tipo *shunting feedback* com pesos.

Esta rede neural possui um comportamento diferente do comportamento descrito no item 4.4.1. Na equação (4.11), dependendo da função de ativação utilizada, teremos uma característica que é denominada de termo de memória de tempo curto (STM), que é explicada pelo fato de mesmo após ser retirado o padrão de entrada, as unidades não retornarão ao valor zero, tendo com isto, armazenado o padrão.

A rede apresenta diferentes características dependendo das funções de ativação implementadas nas unidades da camada competitiva [GRO88].

A função de ativação sigmoideal, a qual será objeto de estudo utilizado neste trabalho, combina supressão de ruído em baixas atividades, armazenamento linear de padrões em valores intermediários, e uma saída limitada em grandes valores de atividades. Quando um vetor de entrada está presente, a unidade alcança um valor de equilíbrio; após removido o padrão, as atividades acima de um certo limiar são aumentadas enquanto que as atividades que estão abaixo deste limiar são suprimidas.

4.5 Simulações Contínuas

Nesta seção são demonstrados os comportamentos das redes neurais artificiais abordadas nos itens 4.4.1 e 4.4.2, através da simulação das equações diferenciais apresentadas nos respectivos itens mencionados anteriormente.

Estas simulações foram realizadas em um hardware sequencial do tipo computador de uso genérico e a linguagem de programação adotada foi a linguagem C. As listagens completas se encontram nos anexos 1, 2 e 3 sendo estas respectivamente para as simulações de redes do tipo *shunting feedforward*, *shunting feedback* e *shunting feedback* com pesos associados as entradas.

Para as três simulações apresentadas foi adotado como método de resolução das equações diferenciais o método das diferenças finitas.

A primeira rede simulada foi a rede do tipo *shunting feedforward*. A rede é composta de dois neurônios na camada competitiva e duas entradas, portanto pode distinguir dois padrões distintos. Apresenta-se à camada de entrada, dois padrões ortogonais entre si. Os padrões apresentados à entrada da rede neural são respectivamente: (0.3 ; 1.8) e (1.8 ; 0.3). Estes valores numéricos não precisam ser exatos, apenas devem ser bem diferentes, se possível ortogonais entre si, para não causar uma resposta errônea por parte da rede neural.

Na figura 4.8 está demonstrado o modelo utilizado para esta simulação.

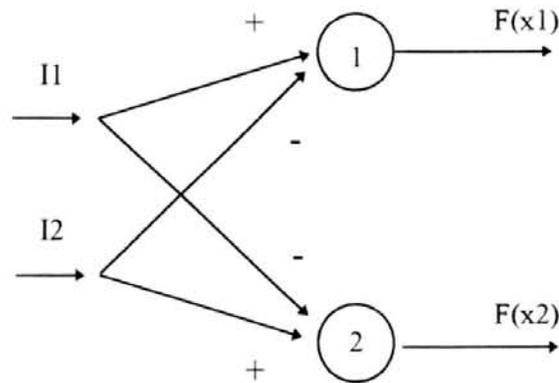


FIGURA 4.8 - Modelo da rede neural do tipo *shunting forward*

A resposta obtida na saída da rede neural pode ser observada na figura 4.9. Quando o primeiro padrão é apresentado, somente um dos dois neurônios, o neurônio número 1, responde como vencedor, ficando a saída do outro neurônio, o neurônio número 2, com um sinal bem menor. Contudo, quando o padrão é alterado, isto é, apresenta-se à rede o segundo padrão, a rede se adapta a esta nova situação, indicando assim, o neurônio número 2 como vencedor para este padrão.

Este gráfico mostra ainda a característica comentada no item 4.4.1, que diz que no momento em que os padrões são retirados da entrada o sinal decai gradualmente a zero.

Estes momentos acima explicados, são sinalizados na figura 4.9, como sendo troca do padrão e retirada do padrão. Nestes momentos, é que ocorrem respectivamente a troca do padrão 1 para o padrão 2 e por fim a retirada do padrão da entrada da rede neural.

O número junto a cada uma das linhas indica a saída do respectivo neurônio.

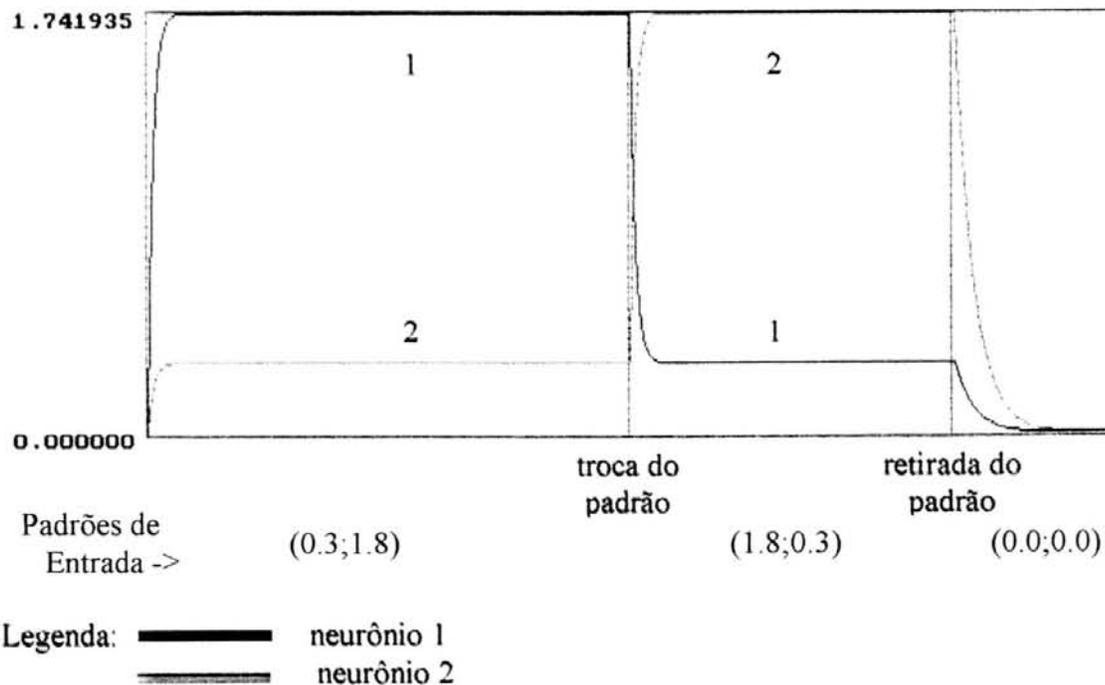


FIGURA 4.9 - Resultados da simulação de uma rede neural do tipo *shunting forward*

A próxima rede simulada foi a rede do tipo *shunting feedback*. A rede é composta de dois neurônios na camada competitiva e duas entradas. Na figura 4.10 pode-se observar o modelo utilizado.

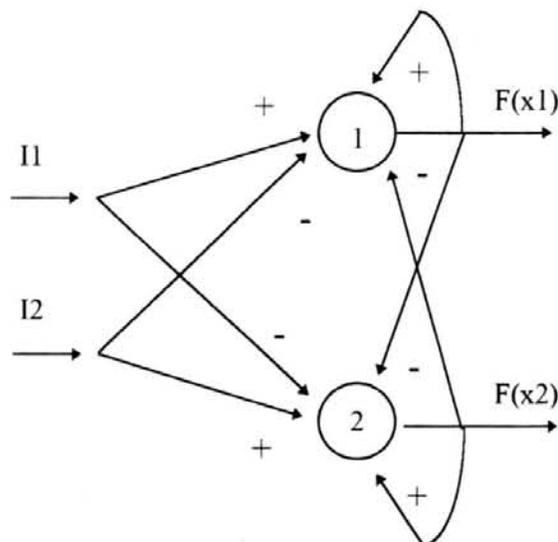


FIGURA 4.10 - Modelo da rede neural do tipo *shunting feedback*

Apresenta-se à rede dois padrões de entrada ortogonais entre si. Os padrões apresentados à rede neural são respectivamente: (1.0 ; 2.0) e (2.0 ; 1.0). Como foi também comentado acima, estes valores numéricos não precisam ser exatos, porém devem ser ortogonais.

A resposta desta simulação pode ser observada na figura 4.11. Foi seguido o mesmo procedimento adotado na simulação anterior, isto é apresenta-se à rede o primeiro padrão e após retira-se este padrão e apresenta-se o segundo padrão, por fim retira-se o segundo padrão e observa-se o comportamento da rede neural.

No momento em que o primeiro padrão foi apresentado à entrada da rede neural somente um dos dois neurônios, o neurônio número 1, responde como vencedor, ficando a saída do outro neurônio, o neurônio número 2, com um sinal bem menor. Contudo, quando o padrão é alterado para o segundo padrão, a rede se adapta a esta nova situação, indicando assim, o neurônio número 2 como vencedor para este padrão.

Este gráfico mostra ainda a característica comentada no item 4.4.2, que diz que mesmo com os padrões retirados, a rede permanece com o último neurônio vencedor ativo, caracterizando assim o termo de memória de tempo curto (STM) comentado no item 4.4.2.

Para seguirmos a mesma padronização adotada na simulação anterior, temos na figura 4.11 duas linhas verticais que indicam respectivamente o momento em que é realizado a troca do primeiro padrão para o segundo padrão e o momento em que é retirado o padrão que está sendo aplicado à entrada da rede neural.

Novamente o número junto a cada uma das linhas indica a saída do respectivo neurônio.

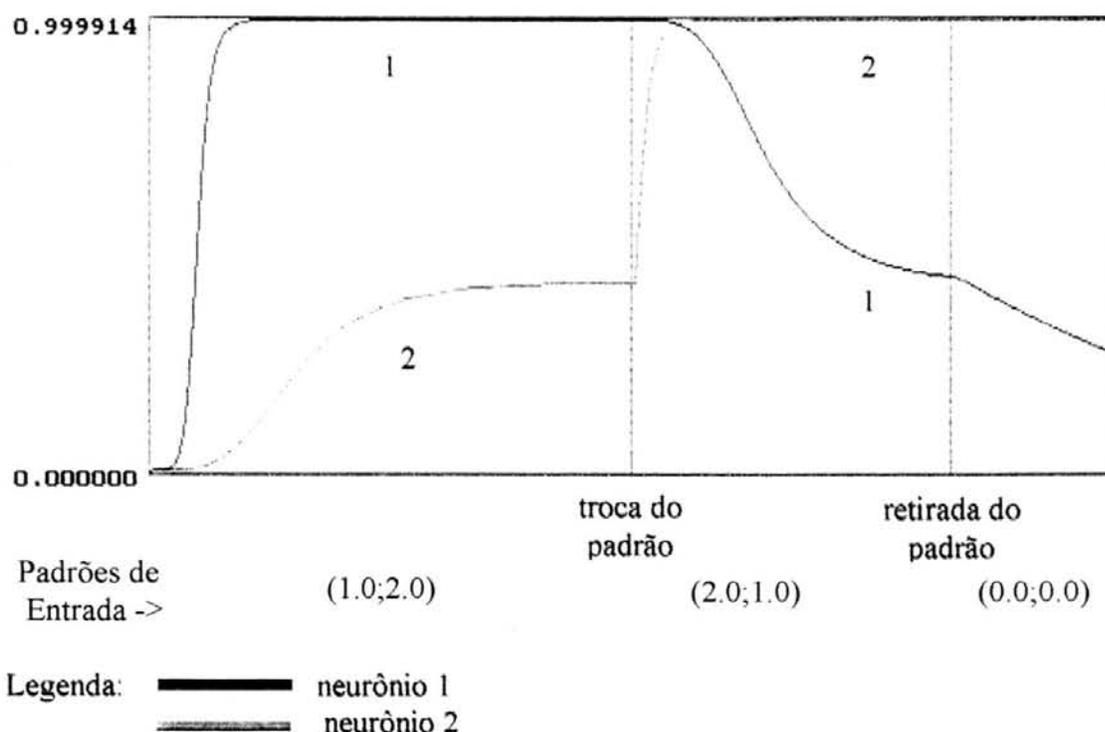


FIGURA 4.11 - Resultados da simulação de uma rede neural do tipo *shunting feedback*

Até o momento realizaram-se simulações com somente dois neurônios na camada competitiva, o que somente possibilitava o reconhecimento de dois padrões diferentes entre si. Relembrando os padrões apresentados na primeira simulação realizada anteriormente, temos: (0.3 ; 1.8) para o primeiro padrão e (1.8 ; 0.3) para o segundo padrão. A rede, com dois neurônios na sua camada competitiva, não pode reconhecer corretamente um terceiro padrão, por exemplo, o padrão (1.8 ; 1.8). Para conseguirmos reconhecer um terceiro padrão necessita-se de uma rede com no mínimo três neurônios na camada competitiva.

Observando que as respostas obtidas até o momento correspondem ao esperado, realizou-se a simulação da rede neural do tipo *shunting feedback* com pesos, utilizando uma camada competitiva com três neurônios. Na figura 4.12, se encontra o modelo utilizado para esta simulação.

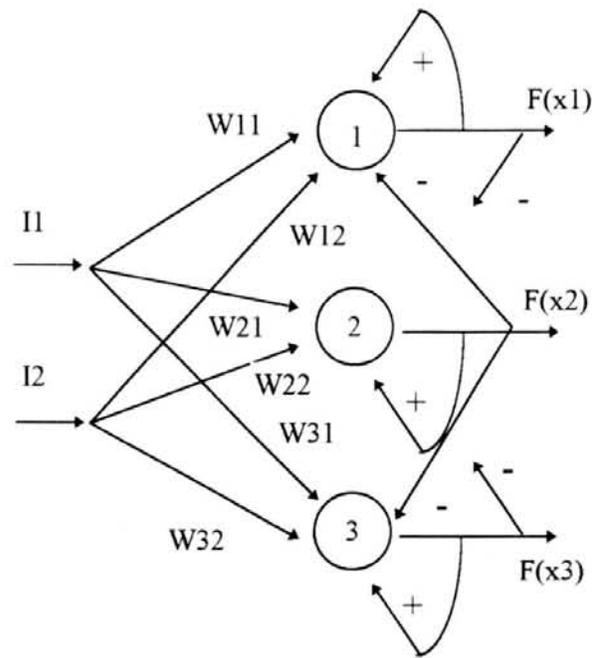


FIGURA 4.12 - Modelo da rede neural do tipo *shunting feedback* com pesos

Escolheu-se, para esta simulação, os seguintes padrões: (1.8 ; 0.3) , (0.3 ; 1.8) e (1.8 ; 1.8).

Na figura 4.13, pode-se observar que, com o primeiro padrão apresentado, somente um dos três neurônios, o neurônio número 3, responde como vencedor, ficando a saída dos outros dois neurônios, o neurônio número 1 e o neurônio número 2, com um sinal menor. Contudo, quando o segundo padrão é apresentado, a rede se adapta a esta nova situação, indicando assim, o neurônio número 2 como vencedor para este padrão. Por fim, quando se apresenta à rede o terceiro padrão, a mesma se adapta, indicando como vencedor o neurônio número 1.

Este gráfico mostra também, a característica citada no item 4.4.2, que comprova novamente a existência da característica do termo de memória de tempo curto (STM). Tal característica é observada quando se retira o padrão que está sendo aplicado à camada competitiva.

Observa-se na figura 4.13 três linhas verticais que indicam respectivamente o momento em que são realizadas as alterações entre os padrões apresentados à rede e o momento em que é retirado o padrão que está sendo aplicado à entrada da rede neural.

Como foi dito anteriormente, esta rede simulada é uma rede neural com pesos associados às entradas. Para esta simulação, adotamos os seguintes pesos: (0.15 ; 0.15), (0.0 ; 0.85) e (0.85 ; 0.0). Por ser uma rede sem a característica de aprendizagem, estes pesos foram ajustados de forma manual, correspondendo aos três padrões de entrada.

Novamente o número junto a cada uma das linhas indica a saída do respectivo neurônio.

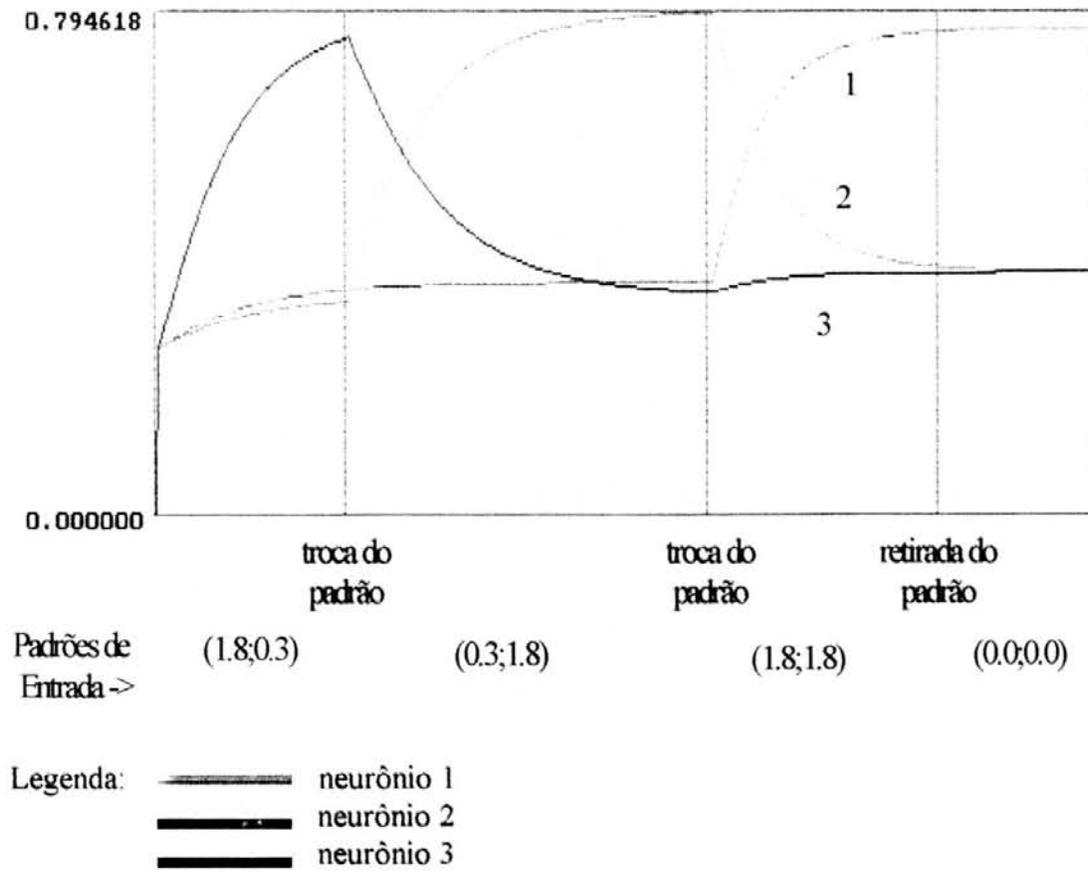


FIGURA 4.13 - Resultados da simulação de uma rede neural do tipo *shunting feedback* com pesos

5 Proposta de Implementação em Hardware dedicado de Redes Neurais Competitivas com Técnicas Analógicas

5.1 Introdução

Neste capítulo, propomos a implementação das equações descritas durante o capítulo 4, item 4.4, utilizando componentes eletrônicos. Os componentes utilizados foram interligados de modo que as equações de tensões ou correntes são equivalentes às expressões teóricas anteriormente descritas.

Além destas proposições iniciais, também foram realizadas simulações onde os resistores e capacitores pertencentes ao circuito em questão foram substituídos por transistores operando como resistores e como capacitores não lineares.

Antes de abordarmos a parte prática propriamente dita, convém ressaltarmos duas importantes características necessárias para alguns dos dispositivos que serão utilizados durante as simulações. Estas características são a condutância e a capacitância que os transistores possuem quando estes operam sob certas condições elétricas.

5.2 Condutâncias

Como será visto nas próximas seções, temos que alguns dos transistores que compõem as redes aqui em questão, devem operar como resistores, ou condutâncias, de modo a preencherem as características necessárias para que as redes aqui simuladas operem de forma adequada.

Apenas para relembrar; se aplicarmos uma tensão no terminal de porta de um transistor MOS irá ocorrer uma inversão na superfície do condutor [SZE81]. Se, além disto, for aplicada uma pequena tensão no terminal dreno, uma corrente fluirá do terminal dreno para o terminal fonte através do canal de condução criado pela tensão aplicada no terminal de porta. Logo, este canal atua como uma resistência, e a corrente de dreno é proporcional a tensão de dreno. Nesta situação diz-se que o transistor está na região linear.

Como demonstrado por [ALL87] e por [SZE81] a corrente de dreno pode ser calculada através da seguinte equação:

$$I_d = \frac{\mu_s * C_{ox} * W}{L_{mod}} \left\{ [V_{gs} - V_{bin} - \theta * V_{ds}] * V_{ds} - \frac{2 * \gamma_s}{3} [(2|\phi_f| + V_{ds} + V_{sb})^{3/2} - ((2|\phi_f| + V_{sb})^{3/2})] \right\} \quad (5.1)$$

onde:

- μ_s é estimado pela degradação da mobilidade superficial μ_0 ;
- γ_s é o parâmetro de limiar do substrato corrigido para pequenas geometrias;
- W é a largura do canal;
- L_{mod} é o comprimento do canal modulado pelo parâmetro de modulação do comprimento do canal (λ);
- C_{ox} é a capacitância por unidade de área do óxido;
- ϕ_f é o potencial da superfície em inversão forte e
- V_{bin} e θ são parâmetros calculados como apresentado em [ALL87].

Esta equação (5.1) prediz que, para uma tensão de porta constante, $V_{gs} = \text{constante}$, a corrente de dreno primeiro aumenta linearmente com a tensão de dreno, região conhecida como linear, e então gradualmente aproxima-se de um valor de saturação.

A equação (5.1) é utilizada para a realização das simulações elétricas no aplicativo HSPICE. Para tanto deve-se selecionar o nível dois para os modelos dos transistores em questão. Tal equação é apresentada no manual do usuário do referido aplicativo.

Na figura 5.1, a seguir, temos o levantamento da curva característica de um transistor com $L=5\mu$ e $W=10\mu$. Esta dimensão foi escolhida de forma arbitrária, mas condiz com a maioria dos transistores utilizados nas diversas simulações que a seguir serão discutidas.

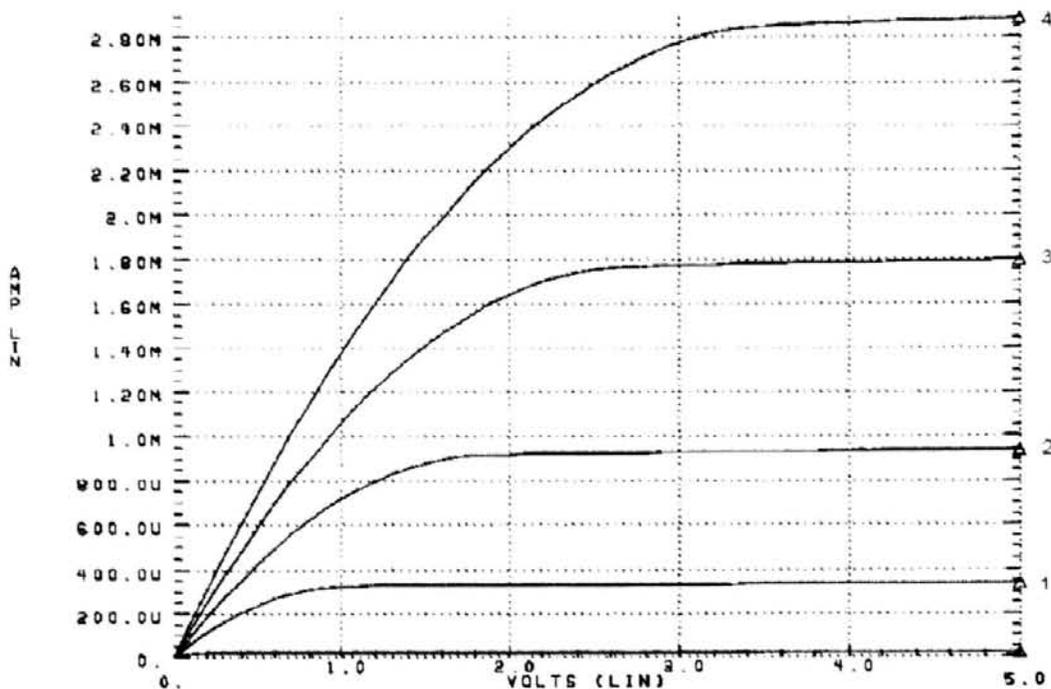


FIGURA 5.1 - Curva do transistor para diversas tensões de porta..

Observa-se que há diversas curvas no mesmo gráfico; cada curva apresentada é representativa de uma determinada tensão de porta aplicada. As tensões aplicadas no terminal de porta da figura acima são respectivamente de baixo para cima 1.0V, 1.5V, 2.0V, 2.5V.

A descrição HSPICE utilizada para a aquisição destas curvas é mostrada no anexo 4.

Se, agora, dividirmos os valores de corrente de dreno obtidas nas simulações, pelas respectivas tensões de dreno, adotando-se uma tensão de porta constante, teremos as condutâncias nos diversos pontos para este transistor em questão. Na figura 5.2, abaixo, mostra-se o gráfico da condutância para uma tensão de porta de 2,5 Volts. Sendo que, a unidade do eixo das abscissas está em Volts e a unidade do eixo das ordenadas está em Mhos.

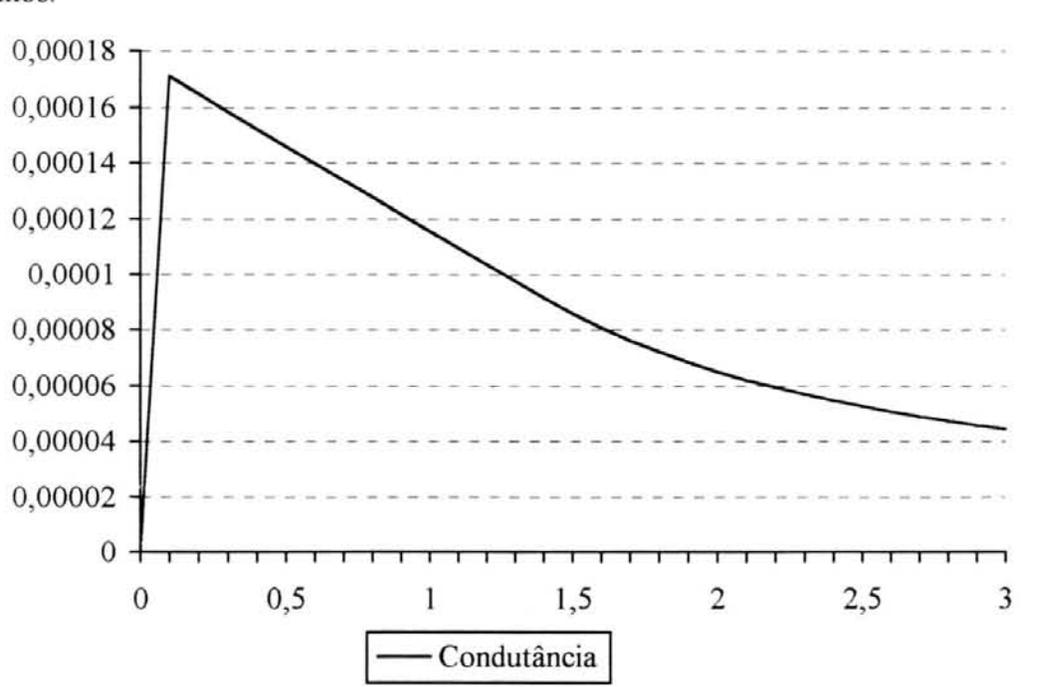


FIGURA 5.2 - Gráfico da condutância pela tensão de dreno.

Através da figura 5.2 pode-se observar que a condutância do transistor em questão, com uma tensão de porta constante, diminui de forma aproximadamente linear até um certo ponto, onde a partir deste, não possui mais um comportamento linear.

5.3 Capacitância

Assim como se descreveu o comportamento de um transistor MOS operando como uma condutância, iremos agora fornecer subsídios para o entendimento de um transistor MOS operando como um capacitor não linear.

Esta técnica será posteriormente testada em um modelo de rede neural, tendo como objetivo, verificar o comportamento do circuito a nível de simulação, visando com isto uma possível minimização do componente eletrônico, o capacitor, e por consequência uma menor dimensão para todo o circuito em questão.

Analisando-se o transistor MOS sob o ponto de vista de um modelo para grandes sinais, temos que considerar outras características, dentre elas cita-se as várias capacitâncias inerentes ao modelo, como descrito por [ALL87].

Há três tipos de capacitâncias pertinentes ao modelo. O primeiro tipo inclui as capacitâncias que possuem a porta em comum. O segundo tipo são as capacitâncias existentes entre o substrato e o dreno ou entre o substrato e o fonte, associadas com a região de depleção. O terceiro e último tipo diz respeito às capacitâncias parasitas.

Para este trabalho, iremos somente abordar as capacitâncias que possuem a porta em comum e que fazem parte do segundo tipo dentre os citados no parágrafo anterior. Estas são capacitâncias entre a porta e o dreno (C_{gd}), a porta e o fonte (C_{gs}) e entre a porta e o substrato (C_{gb}).

O capacitor C_{gd} e C_{gs} são capacitâncias de sobreposição e são devidas as sobreposições de duas superfícies condutoras separadas por um isolante dielétrico. Estas capacitâncias são dependentes da condição de operação do transistor, e são de valores desprezíveis.

As equações que regem o comportamento destas capacitâncias são mostrados em [ALL87].

Para efeito de um melhor entendimento desta técnica, realizou-se uma simulação a nível elétrico com a seguinte descrição:

```
*
* modelos dos transistores
*
.model nmos nmos level=2 ld=0.325u tox=250e-10 nsub=2e16 vto=0.7
+uo=510 uexp=0.22 ucrit=24.3k delta=0.4 xj=0.4u vmax=54k neff=4
+rsh=55 nfs=0 js=2u cj=130u cjsw=620p mj=0.53 mjsw=0.53 pb=0.68v
+cgdo=320p cgso=320p
*
.model pmos pmos level=2 ld=0.3u tox=250e-10 nsub=5e16 vto=-1.1
+uo=210 uexp=0.33 ucrit=51k delta=0.4 xj=0.5u vmax=47k neff=0.88
+rsh=75 nfs=0 js=10u cj=490u cjsw=590p mj=0.46 mjsw=0.46 pb=0.78v
+cgdo=320p cgso=320p
*
vgs 1 0 pulse(0 5 0 100n 100n 1n 200n)
m1 2 1 2 0 nmos w=100u l=100u ad=330p as=330p pd=106.6u ps=106.6u
vids 2 0 dc 0
*
.tran 1n 100n
.options post nomod nopage
.end
```

FIGURA 5.3 - Descrição elétrica para verificação da capacitância.

Pode-se observar que nesta descrição, os terminais fonte e dreno estão conectados juntos e ambos se encontram ligados ao nível de tensão mais baixo (zero volts). Através desta simulação obteve-se a curva da corrente entre o terminal de dreno e o terminal de fonte para o massa, zero volts. Com esta curva pode-se traçar o gráfico mostrado na figura 5.4, onde demonstra-se os diversos valores de capacitância que um transistor MOS, ligado deste modo, pode assumir. No eixo 'x' encontra-se os valores pertinentes a tensão entre os terminais de porta e de fonte e no eixo dos 'y' está colocado o valor das capacitâncias, em fentoFarad, obtidas.

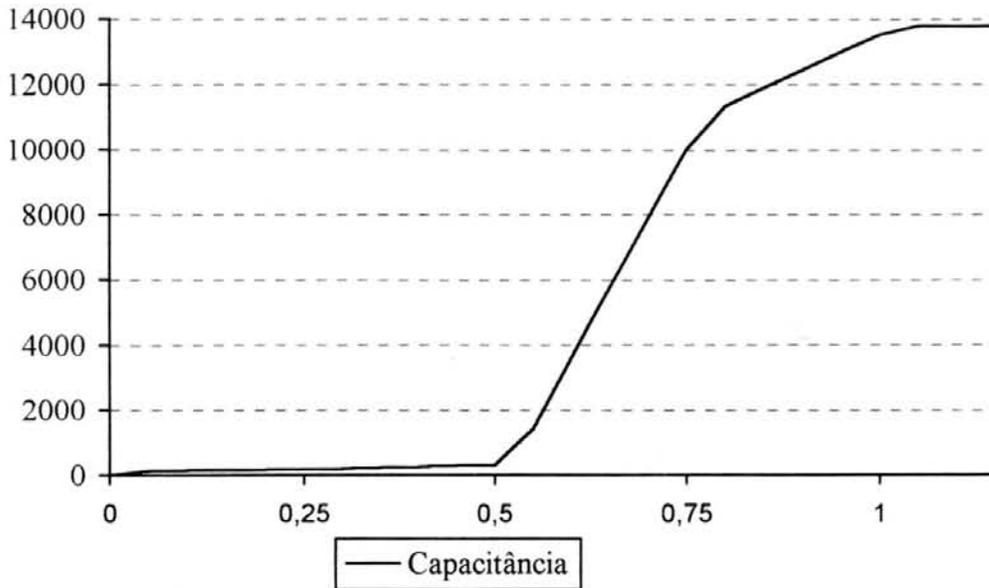


FIGURA 5.4 - Gráfico da capacitância de um transistor MOS.

O circuito descrito na figura 5.3 possui a forma física, como a demonstrada na figura 5.5.

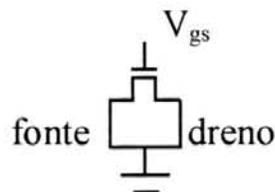


FIGURA 5.5 - Ligação física para um transistor operar como um capacitor não linear.

Esta topologia de ligação mostrada na figura 5.5, será implementada no modelo neural e por fim será realizada uma simulação a nível elétrico para verificar o possível funcionamento com este tipo de ligação ao invés do uso dos capacitores normais. Tal artifício visa uma diminuição do tamanho do componente eletrônico e também do circuito como um todo.

5.4 Rede Neural *Shunting Feedforward*

A primeira simulação realizada a nível elétrico, foi a rede neural do tipo *shunting feedforward*.

Para facilitar a compreensão, a equação 5.2 repete a equação já discutida no capítulo 4, seção 4.4.

$$\frac{dx_i}{dt} = -A*x_i + (B - x_i)*I_i - x_i*\sum_{k \neq i} I_k \quad (5.2)$$

Como já foi também abordado na seção 4.4 a técnica de competição de reforço central e inibição lateral é realizada através das ligações dos sinais de entrada à camada competitiva.

Abaixo, na figura 5.6, mostra-se o circuito elétrico que foi utilizado para a simulação do tipo de rede neural *shunting feedforward*. Este circuito foi implementado a partir da figura 4.8 apresentada no capítulo anterior.

A rede implementada para a simulação possui somente dois neurônios na sua camada competitiva e duas entradas, sendo representada por cada um dos dois amplificadores diferenciais. Os pesos sinápticos são representados pelos transistores T1, T2, T3 e T4.

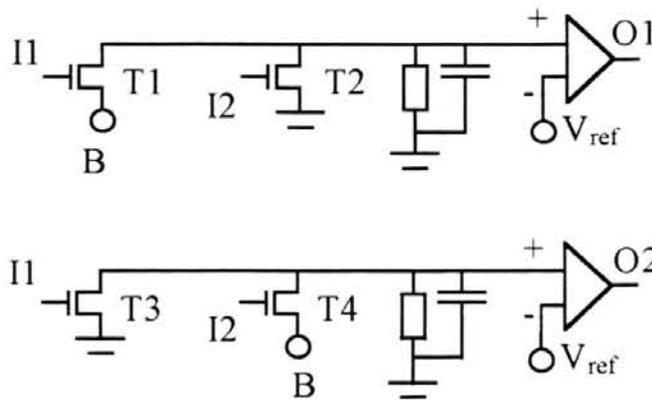


FIGURA 5.6 - Circuito para simulações da Rede Neural *Shunting Feedforward*.

A descrição HSPICE relacionada a figura 5.6 pode ser observada através do anexo 5. Nesta descrição, pode-se observar o valor utilizado para a fonte de tensão , B, e para a fonte de tensão utilizada na entrada negativa dos amplificadores diferenciais, Vref.

Como pode-se observar nesta descrição e nas demais, o amplificador diferencial utilizado para as simulações elétricas é o mesmo. Para tanto, na figura 5.7, é apresentado o desenho do modelo do amplificador diferencial utilizado neste trabalho.

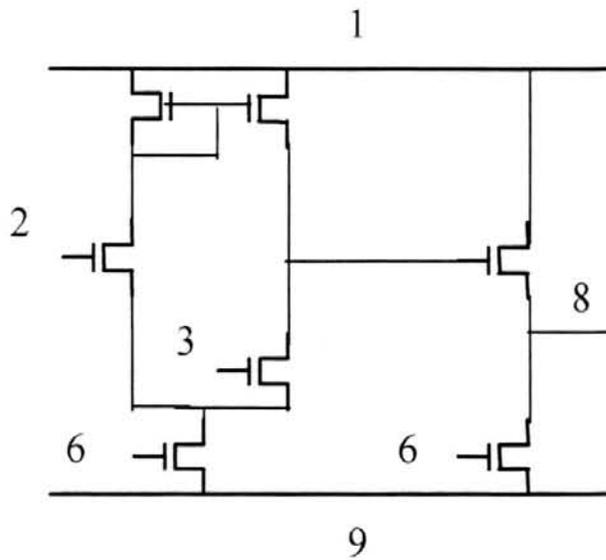


FIGURA 5.7 – Modelo do amplificador diferencial utilizado para as simulações.

É importante observar que os transistores T1, T2, T3 e T4 são os dispositivos fundamentais para que a rede opere de modo adequado. As tensões aplicadas em seus respectivos terminais de porta, I1 e I2, fazem com que estes transistores operem como condutâncias.

Tal fato pode ser mais bem demonstrado através da figura abaixo, onde coloca-se o circuito elétrico utilizando capacitâncias em vez de transistores, ressaltando-se o fato de que na figura abaixo está a representação de apenas um dos dois neurônios utilizados na figura 5.6.

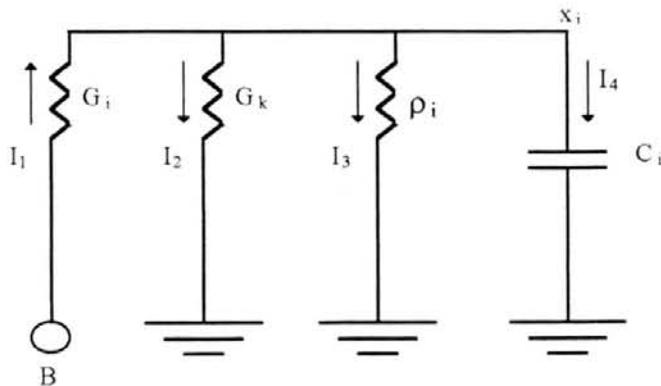


FIGURA 5.8 - Circuito equivalente utilizando resistores e capacitores.

Por meio da lei das correntes, tem-se que:

$$I_4 = I_1 - I_2 - I_3$$

$$C_i (dx_i/dt) = (B - x_i)G_i - x_i G_k - (x_i/\rho_i)$$

$$\frac{dx_i}{dt} = - \frac{1}{\rho_i C_i} x_i + (B - x_i) \frac{G_i}{C_i} - x_i \frac{G_k}{C_i} \quad (5.3)$$

onde B é a tensão de alimentação do circuito.

Nota-se, claramente, a correspondência desta equação (5.3) com a equação (5.2), onde:

$$A = \frac{1}{\rho_i C_i}$$

$$I_i = \frac{G_i}{C_i}$$

$$I_k = \frac{G_k}{C_i}$$

Simulando-se o circuito da figura 5.6 obteve-se resultados que estão apresentados na figura 5.9.

No gráfico de número 1, da figura 5.9, apresentam-se os sinais aplicados à entrada da camada competitiva. Observa-se que estes padrões de entrada são aplicados como tensões nos terminais de porta dos transistores T1, T2, T3 e T4 que por sua vez operam como condutâncias. Os padrões são formados por pulsos cuja amplitude máxima é de 3.0 Volts. Portanto, os padrões aplicados à camada competitiva para esta simulação foram, respectivamente: (3 ; 3), (3 ; 0) e (0 ; 3). Como a rede neural possui dois neurônios na sua camada competitiva, esta somente poderá reconhecer dois padrões, que serão (0 ; 3) e (3 ; 0). O padrão (3 ; 3) não será reconhecido corretamente pela rede neural, visto que para o seu correto reconhecimento a rede neural deveria possuir no mínimo três neurônios em sua camada competitiva. A inclusão deste padrão na simulação é necessária pelas características de geração de sinais do simulador HSPICE, não interferindo, entretanto, nos resultados esperados.

O gráfico de número 2 é representativo dos sinais obtidos nas saídas dos dois neurônios. Através deste gráfico, pode-se observar o neurônio vencedor para cada um dos padrões apresentados à camada competitiva. Pode-se observar que no instante em que o padrão (3 ; 3) é apresentado à rede, ambos os neurônios respondem como vencedores, mostrando a incapacidade da rede em reconhecer este padrão. A retirada dos padrões aplicados à entrada da rede, padrão (0 ; 0), é utilizada para verificar a existência ou não do termo de memória de tempo curto.

É importante observar nesta simulação, que após os padrões terem sido retirados da entrada da rede neural, as saídas dos neurônios pertencentes a camada competitiva diminuem gradualmente até zero. Com isto, confirma-se o correto comportamento deste circuito com a simulação computacional realizada na seção 4.5.

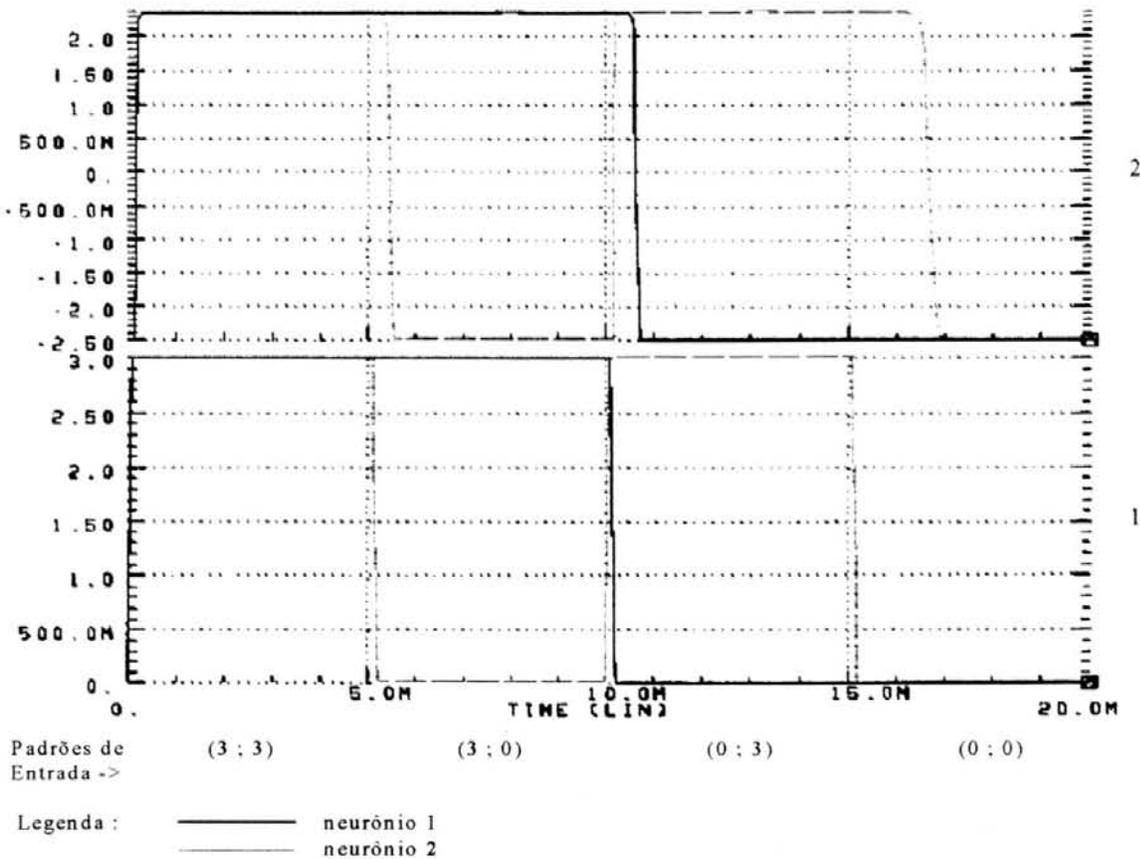


FIGURA 5.9 - Simulação da Rede Neural *Shunting Feedforward*.

5.5 Rede Neural *Shunting Feedback*

A próxima simulação realizada a nível elétrico, foi a rede neural do tipo *shunting feedback*.

Foi apresentado na seção 4.4, a equação diferencial que rege este tipo de rede neural, e aqui iremos novamente colocá-la para facilitar a compreensão do restante desta seção.

$$\frac{dx_i}{dt} = -A*x_i + (B - x_i)*[f(x_i) + I_i] - x_i*\left[\sum_{k \neq i} f(x_k) + \sum_{k \neq i} I_k\right] \quad (5.4)$$

Como já foi também abordado na seção 4.4 a técnica de competição de reforço central e inibição lateral é realizada através das interligações entre os neurônios que compõem a camada competitiva.

Abaixo, na figura 5.10, mostra-se o circuito elétrico que foi utilizado para a simulação do tipo de rede neural *shunting feedback*. Este circuito elétrico foi implementado através da figura 4.10 do capítulo anterior.

A rede implementada para esta simulação possui somente dois neurônios na sua camada competitiva e duas entradas. Cada um dos neurônios é representado pelos amplificadores

diferenciais apresentados na figura 5.10. Os pesos sinápticos são representados pelos transistores 1, 2, 3 e 4. Os transistores 6 e 5 realizam, respectivamente, a realimentação excitatória para o neurônio 2 e inibitória para o neurônio 1. Já os transistores 8 e 7 realizam, respectivamente, a realimentação excitatória para o neurônio 1 e inibitória para o neurônio 2. Com este método de ligação, implementou-se a técnica de competição de reforço central e inibição lateral.

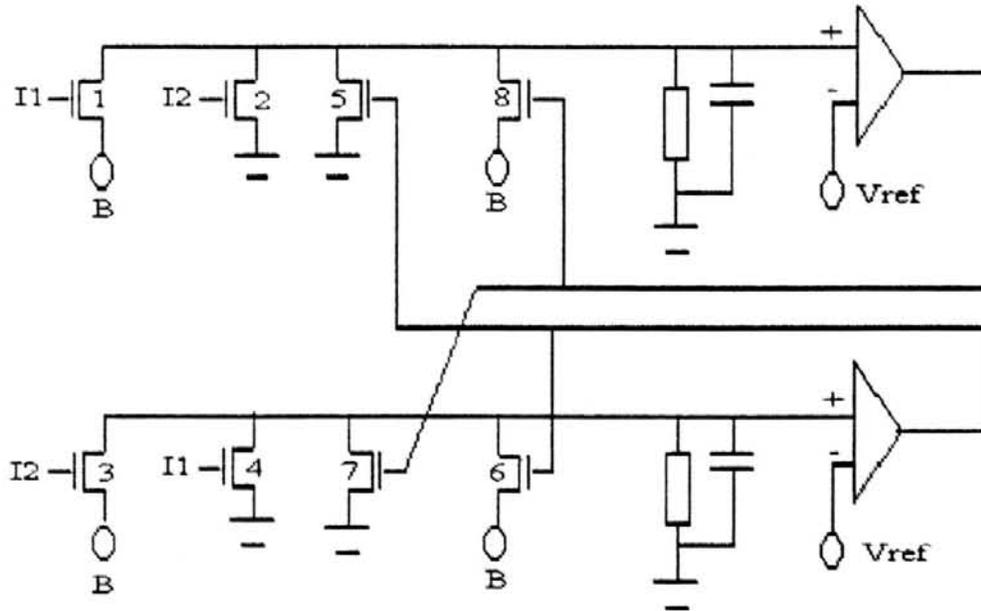


FIGURA 5.10 - Circuito para simulações da Rede Neural *Shunting Feedback*.

A descrição HSPICE relacionada à figura 5.10 pode ser observada através do anexo 6. Nesta descrição pode-se também observar os valores utilizados para a fonte de tensão utilizada, B, e a fonte de tensão utilizada na entrada negativa dos amplificadores diferenciais, Vref.

Nas figuras 5.11 e 5.12, são apresentados os resultados obtidos através das simulações realizadas tendo como modelo o circuito da figura 5.10, com diferentes sequências de padrões de entrada.

Em ambas as figuras 5.11 e 5.12, gráficos 1, mostram-se os padrões apresentados à rede no decorrer do tempo. Estes padrões que são aplicados às entradas I1 e I2 como tensões nos terminais de porta, para os respectivos transistores, fazem com que estes transistores operem como condutâncias e deste modo respeitem a equação citada no início desta seção. Pode-se observar, que os padrões apresentados são os mesmos que foram apresentados na simulação anteriormente discutida neste capítulo, são eles: (3 ; 3), (3 ; 0) e (0 ; 3).

O resultado visualizado através da figura 5.11, gráfico 2, difere do resultado obtido na figura 5.12, gráfico 2, apenas por qual dos neurônios, ao fim das apresentações dos padrões, apresenta a característica do termo de memória de tempo curto. Ressalta-se aqui, que o padrão (3 ; 3) não é tido como válido para as análises dos resultados obtidos, pois como já foi comentado, esta é uma rede com somente dois neurônios em sua

camada competitiva, portanto somente pode haver dois padrões corretamente classificados, são eles: (3 ; 0) e (0 ; 3). O padrão (0 ; 0), que significa a retirada dos demais padrões, nos possibilita verificar a existência ou não do termo de memória de tempo curto. Pode-se observar nas respectivas figuras, 5.11 e 5.12, gráficos 2, que esta característica ocorreu, comprovando assim, que o circuito implementado também responde corretamente ao esperado e demonstrado através de simulações computacionais realizadas na seção 4.4.

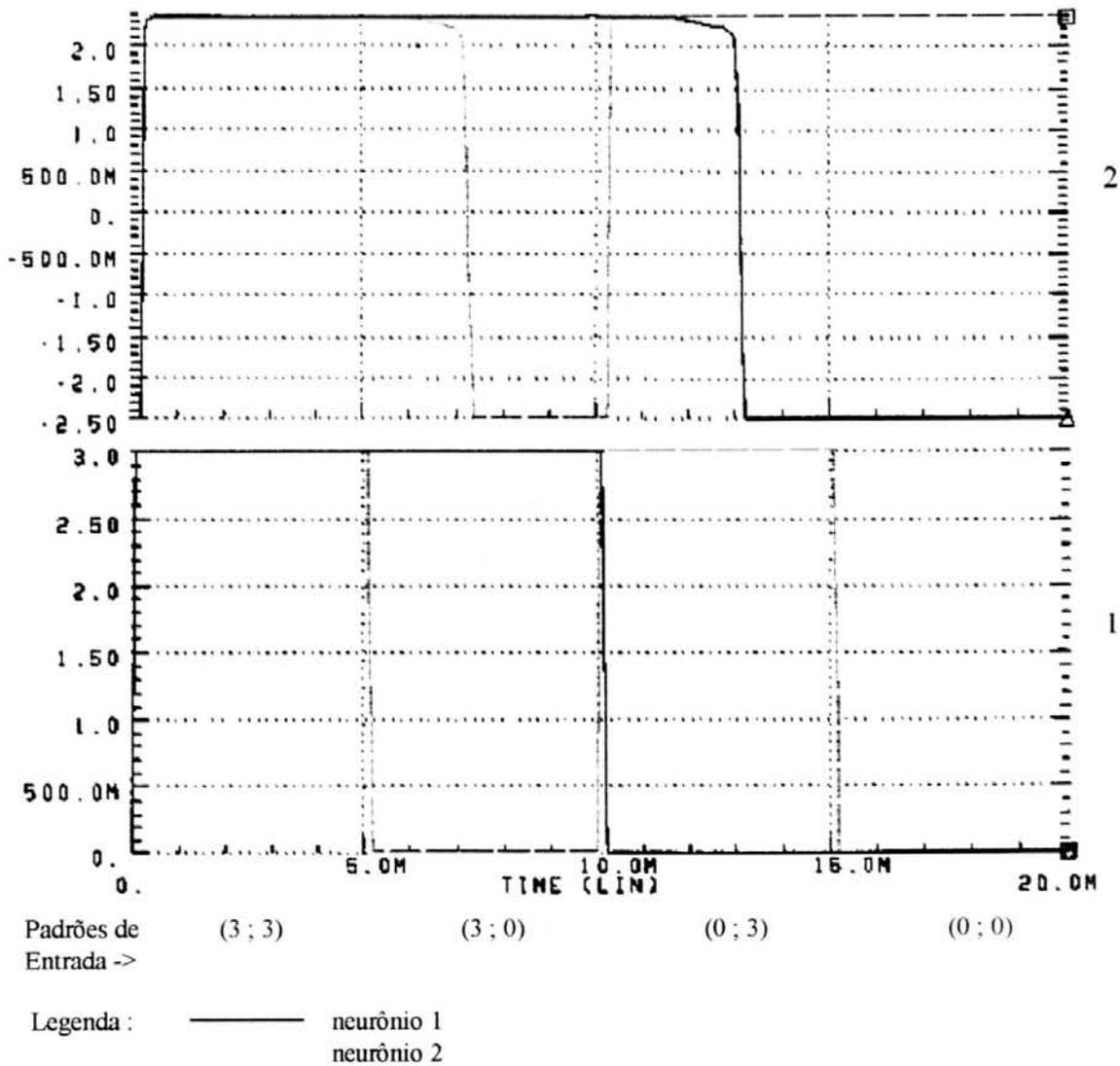


FIGURA 5.11 - Simulação da Rede *Shunting Feedback* para o padrão final (0 ; 3).

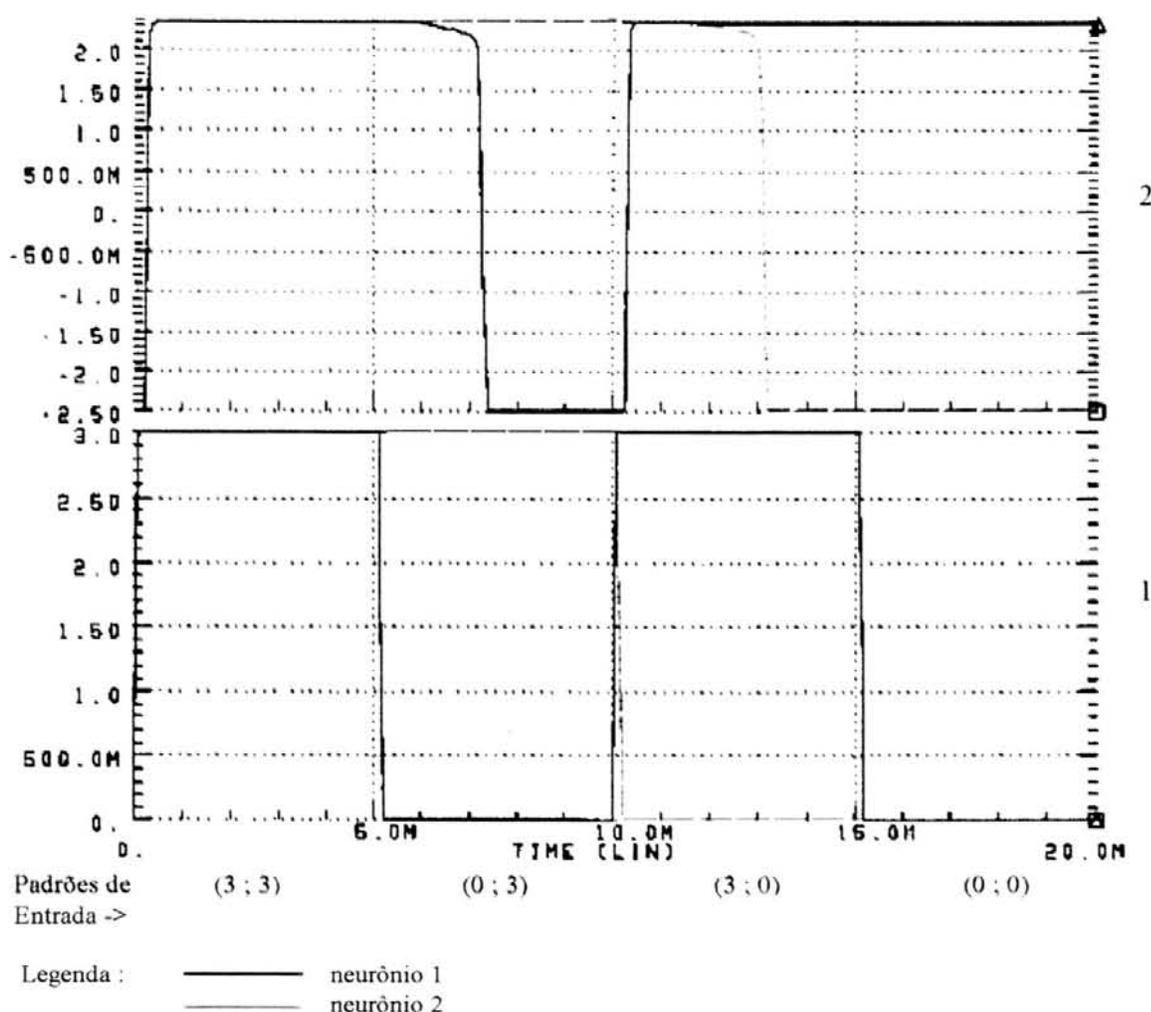


FIGURA 5.12 - Simulação da Rede *Shunting Feedback* para o padrão final (3 ; 0).

5.6 Rede Neural *Shunting Feedback* com pesos

A próxima simulação realizada a nível elétrico, foi a rede neural do tipo *shunting feedback* com pesos associados aos sinais aplicados às entradas.

Novamente, apresenta-se a equação diferencial que rege o comportamento desta rede neural para facilitar a compreensão do texto.

$$\frac{dx_i}{dt} = -A*x_i + (B - x_i)*[f(x_i) + net_i] - x_i*[\sum_{k \neq i} f(x_k) + \sum_{k \neq i} net_k] \quad (5.5)$$

Como já foi também abordado na seção 4.4 a técnica de competição de reforço central e inibição lateral é realizada através das interligações entre os neurônios que compõem a camada competitiva.

Abaixo, na figura 5.13, mostra-se o circuito elétrico que foi utilizado para a simulação do tipo de rede neural *shunting feedback* com pesos:

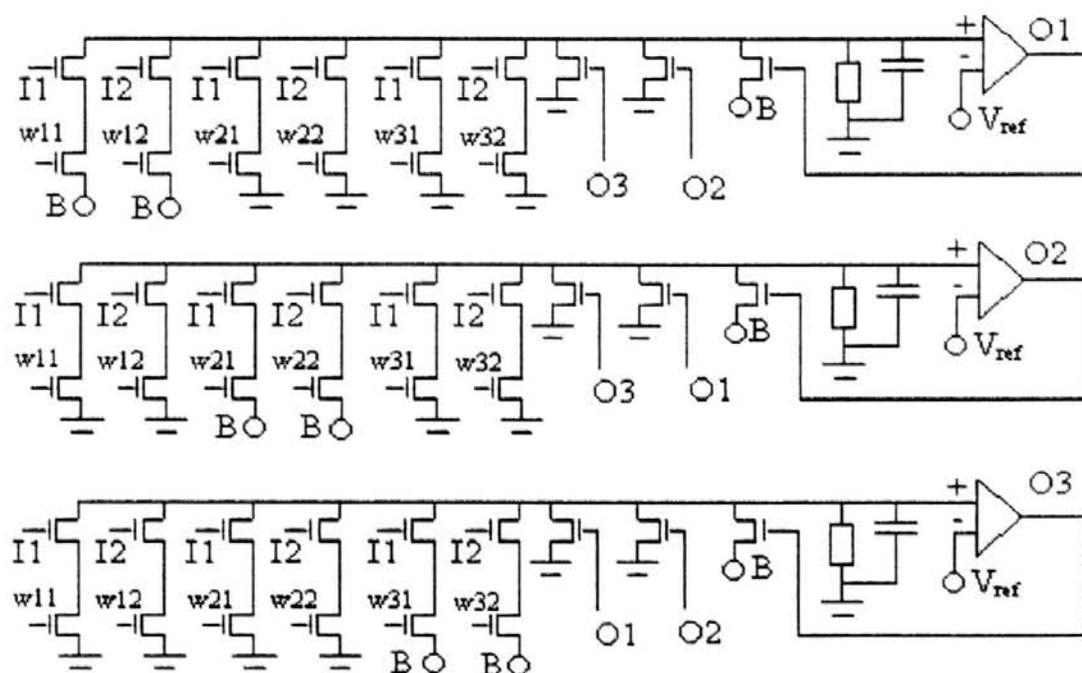


FIGURA 5.13 - Circuito para simulações da Rede *Shunting Feedback* com pesos.

Como pode-se verificar, neste circuito apresentado, há três neurônios que compõem a camada competitiva. Isto se deve ao fato de testarmos o reconhecimento de mais um padrão que será relevante, tal padrão é (3 ; 3). Nas simulações anteriores, as redes neurais utilizadas não reconheciam este padrão corretamente. Novamente o padrão (0 ; 0), que significa a retirada dos padrões aplicados às entradas da rede, é válido para a verificação ou não da característica do termo de memória de tempo curto.

Os transistores nomeados de w_{11} , w_{12} , w_{21} , w_{22} , w_{31} e w_{32} operam como os pesos, e são necessários para respeitar a equação (5.5). Pode-se observar que a técnica de competição de reforço central e inibição lateral é também implementada com três transistores para cada neurônio, sendo um deles para realizar o reforço central e os demais, para realizar a inibição lateral.

A descrição HSPICE relacionada a figura 5.13 pode ser observada através do anexo 7. Nesta descrição pode-se também observar os valores utilizados para a fonte de tensão utilizada, B, e a fonte de tensão utilizada na entrada negativa dos amplificadores diferenciais, V_{ref} .

Nas figuras 5.14, 5.15 e 5.16, são apresentados os resultados obtidos através das simulações realizadas tendo como modelo o circuito da figura 5.13. Estão aqui mostrados as três situações para demonstrar que a característica do termo de memória de tempo curto existe para todos os neurônios da camada competitiva ilustrada na figura 5.13. Nas figuras 5.14 e 5.15, são apresentados os três padrões em diferentes sequências. Na figura 5.16, é apresentado apenas o padrão (3 ; 3) à entrada da rede.

Nos gráficos 1, 2 e 3 das figuras 5.14, 5.15 e 5.16 estão apresentados as saídas dos respectivos neurônios da camada competitiva da rede neural simulada. Os resultados

diferem entre si apenas por qual dos neurônios, ao fim das apresentações dos padrões, apresenta a característica do termo de memória de tempo curto. Com isto pode-se verificar que para cada uma das figuras, há um neurônio apresentando a característica de STM quando as entradas voltam a zero, (0 ; 0).

Nas figuras 5.14, 5.15 e 5.16, também estão colocados os padrões apresentados a rede no decorrer do tempo, estes padrões são mostrados nos gráficos de número 4. Os padrões de entrada que são aplicados às entradas I1 e I2 são tensões entre o terminal porta e massa do circuito, fazendo com que estes transistores operem como condutâncias, e assim, respeitando a equação apresentada no início desta seção.

Os valores das tensões aplicadas nas entradas referidas como W_{xy} , sendo x qualquer valor entre 1 e 3 e y como qualquer valor entre 1 e 2, conforme figura 5.13, foram ajustados de forma manual, visto que, a rede aqui simulada é desprovida de qualquer técnica de aprendizado.

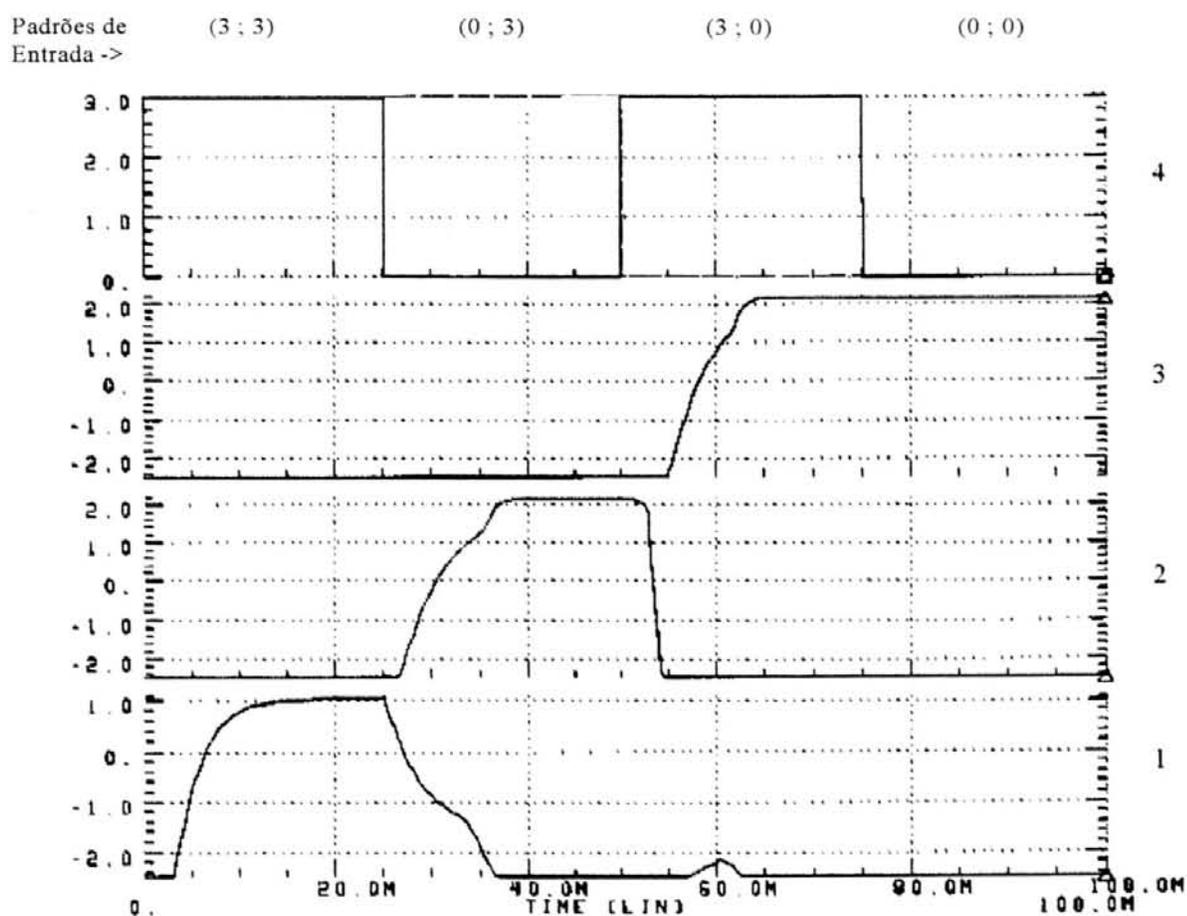


FIGURA 5.14 - Simulação da Rede *Shunting Feedback* com pesos para o padrão final (3;0).

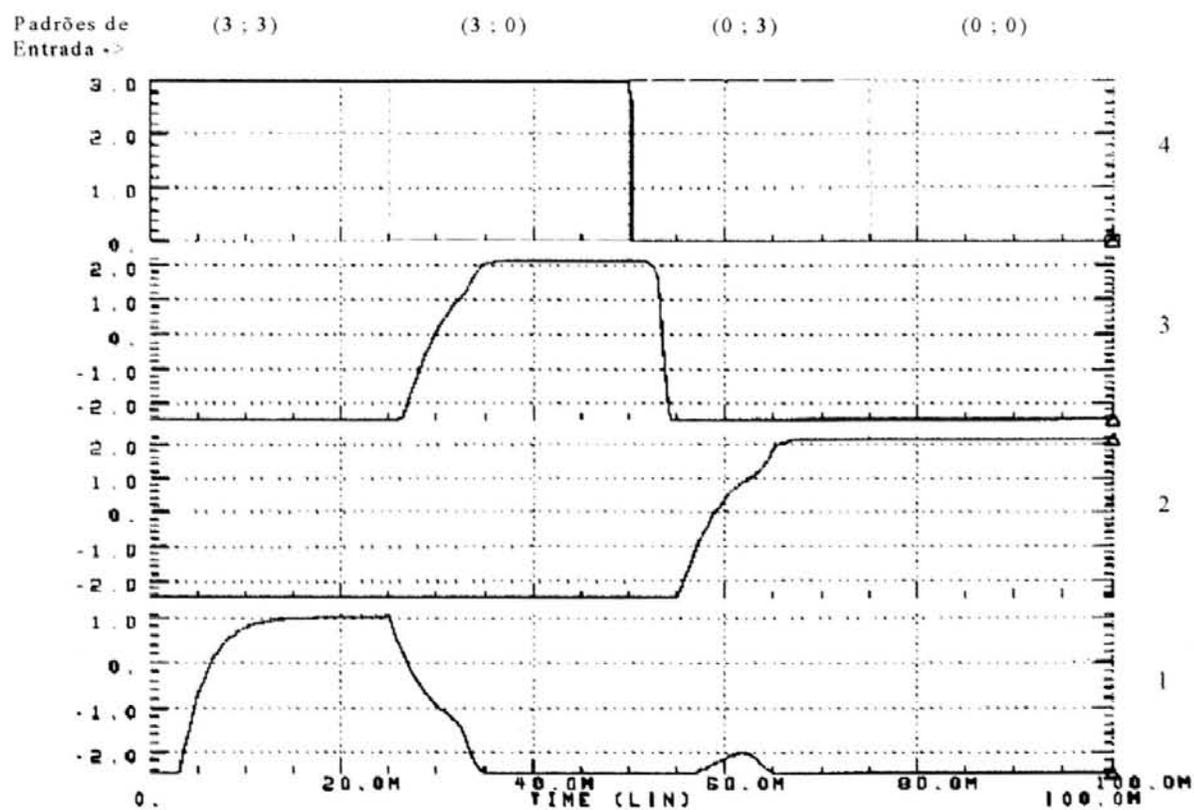


FIGURA 5.15 - Simulação da Rede *Shunting Feedback* com pesos para o padrão final (0,3).

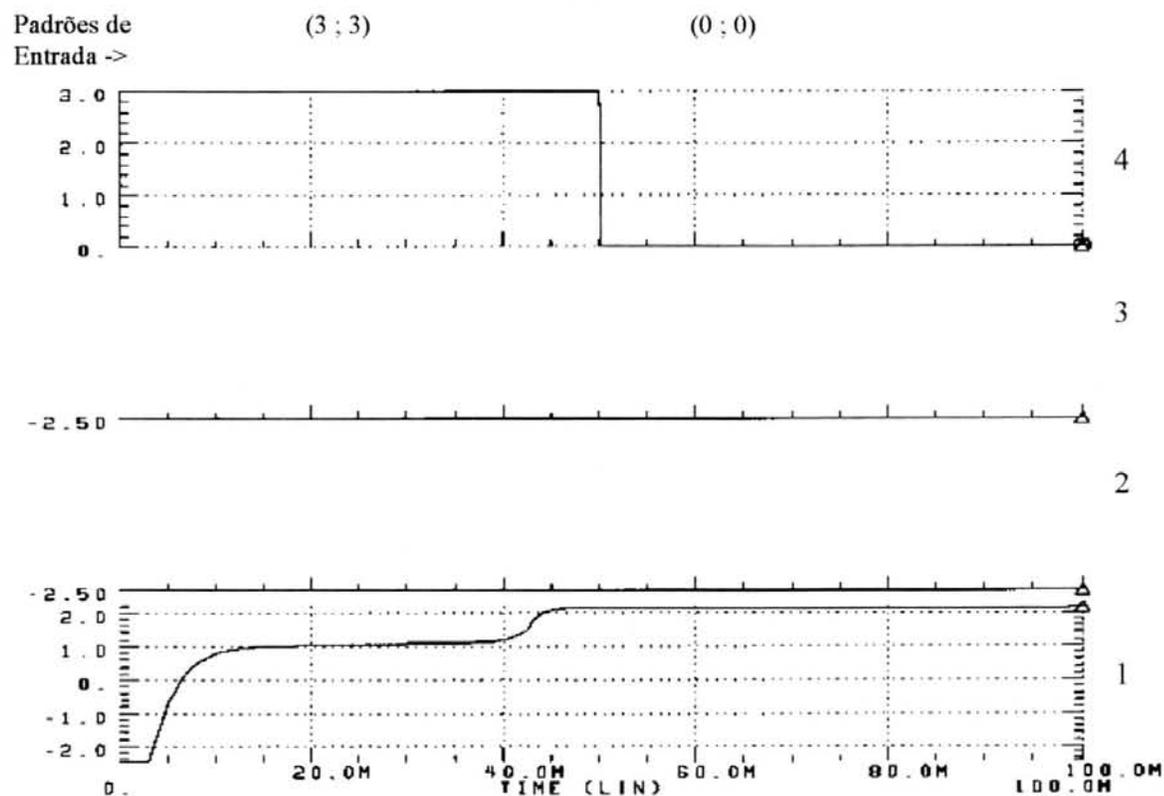


FIGURA 5.16 - Simulação da Rede *Shunting Feedback* com pesos para o padrão final (3,3).

6 Validação da proposta de hardware dedicado através de uma aplicação

6.1 Introdução

Neste capítulo apresentamos uma aplicação prática de um circuito *shunting feedback* para a clusterização de padrões, visando a validação da proposta.

Para tanto utilizar-se-á um circuito com as mesmas características básicas do circuito descrito no item referente a redes neurais do tipo *shunting feedback* com pesos associados as entradas da rede neural.

6.2 Descrição do sistema

A rede neural competitiva utilizada, como já foi dito anteriormente, é baseada na topologia da rede neural *shunting feedback* com pesos associados aos sinais aplicados as entradas da rede.

Na figura 6.1, observa-se o modelo que foi utilizado nas diversas simulações.

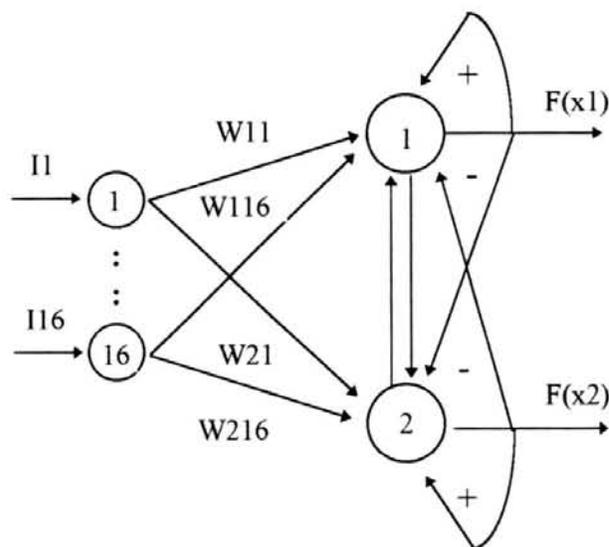


FIGURA 6.1 - Modelo Neural.

Através da figura 6.1 pode-se notar que o sistema empregado será capaz de somente diferenciar entre dois padrões distintos, visto que a camada competitiva do modelo proposto é composta por somente dois neurônios. Sendo este sistema tão somente para a validação da proposta dissertada até aqui, tal modelo neural se torna conveniente devido a mostrar os resultados necessários e possuir um tamanho pequeno, facilitando assim, um possível desenho a nível de *lay-out* de circuito integrado para uma futura implementação em *hardware*.

Conectado a esta camada competitiva, como pode-se observar na figura 6.1, há dezesseis entradas diferentes que podem, por exemplo, ser uma representação de uma imagem ou parte dela, formando uma matriz quadrada de 4x4 posições distintas. A cada uma destas dezesseis entradas está associado um valor de peso.

Para fins de simplificação, definiu-se dois padrões constantes durante as simulações realizadas, e convencionou-se, através do ajuste manual dos pesos, qual neurônio na camada competitiva seria o vencedor para um dos dois padrões pré-determinados.

Na figura 6.2, abaixo, pode-se observar mais claramente quais são os padrões pré-estabelecidos e que serão apresentados a rede neural competitiva, bem como a saída desejada perante cada um dos dois padrões apresentados.

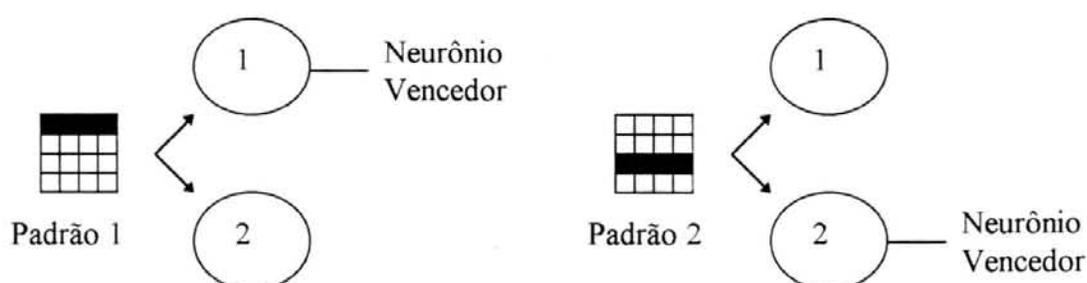


FIGURA 6.2 - Esquema dos padrões e saídas desejadas para a aplicação em questão.

Todas as simulações realizadas foram através do uso do simulador elétrico HSPICE.

A partir do modelo apresentado na figura 6.1, realizou-se a conversão do modelo para um esquema elétrico que correspondesse a este. Na figura 6.3, observa-se o modelo do sistema proposto, agora em termos de componentes eletrônicos. Pode-se observar através desta figura, que a técnica de reforço central e inibição lateral também está implementada.

Para conseguir uma melhor visualização deste modelo somente está apresentado as entradas 1 e 16 com os seus respectivos pesos.

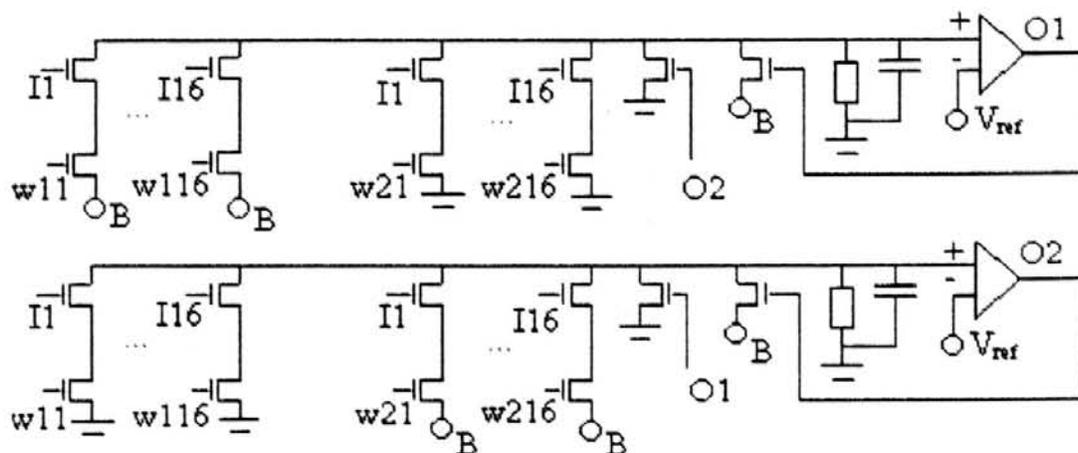


FIGURA 6.3 - Esquema elétrico do circuito para a aplicação.

6.3 Resultados da aplicação

Os resultados para esta aplicação, como já foi comentado, foram obtidos a partir de simulações a nível elétrico no simulador HSPICE. A listagem desta descrição se encontra no anexo 8. Nesta listagem se encontra o valor utilizado para a tensão de alimentação do circuito, B , e o valor utilizado para a tensão na entrada invertida dos amplificadores diferenciais, V_{ref} .

Os modelos que descrevem as características necessárias para as simulações dos transistores MOS foram adquiridas de *foundry* ES2 para a tecnologia de 1.2μ .

Na figura 6.4, pode-se observar a resposta da rede neural competitiva.

Há três gráficos colocados nesta figura 6.4. Nos gráficos 1 e 2, observa-se os padrões aplicados à rede neural. Esta representação é uma simplificação das combinações dos padrões 1 e 2 da figura 6.2. Os dois padrões podem ser apresentados individualmente à rede, padrão 1 ou padrão 2, ou simultaneamente à rede, padrão 1 + padrão 2.

Quando os dois padrões são apresentados simultaneamente à rede, padrão 1 + padrão 2, a resposta obtida não é correta, pois esta rede possui apenas dois neurônios, podendo reconhecer somente dois padrões diferentes, que são: padrão 1 ou padrão 2. A inclusão deste terceiro padrão, padrão 1 + padrão 2, na simulação, é necessária pelas características de geração de sinais do simulador HSPICE, não interferindo nos resultados esperados.

No entanto, se os padrões são apresentados em instantes diferentes, padrão 1 ou padrão 2, a rede competitiva os reconhece, clusterizando-os.

No gráfico 3, da figura 6.4, observa-se a saída da rede neural perante determinados padrões aplicados à entrada da rede. Neste gráfico nota-se que estão sobrepostas as saídas de ambos os neurônios pertencentes a camada competitiva. Observa-se, ainda, que após os padrões terem sido retirados, o último neurônio vencedor, permanece com sua saída ativa, demonstrando assim, o termo de memória de tempo curto.

Além destes padrões perfeitos, foram também apresentados algumas amostras de padrões incompletos ou acrescidos de ruído para verificar a resposta do sistema nestas situações. Verificou-se que a rede responde corretamente para padrões com pequenas distorções, correspondendo ao resultado esperado.

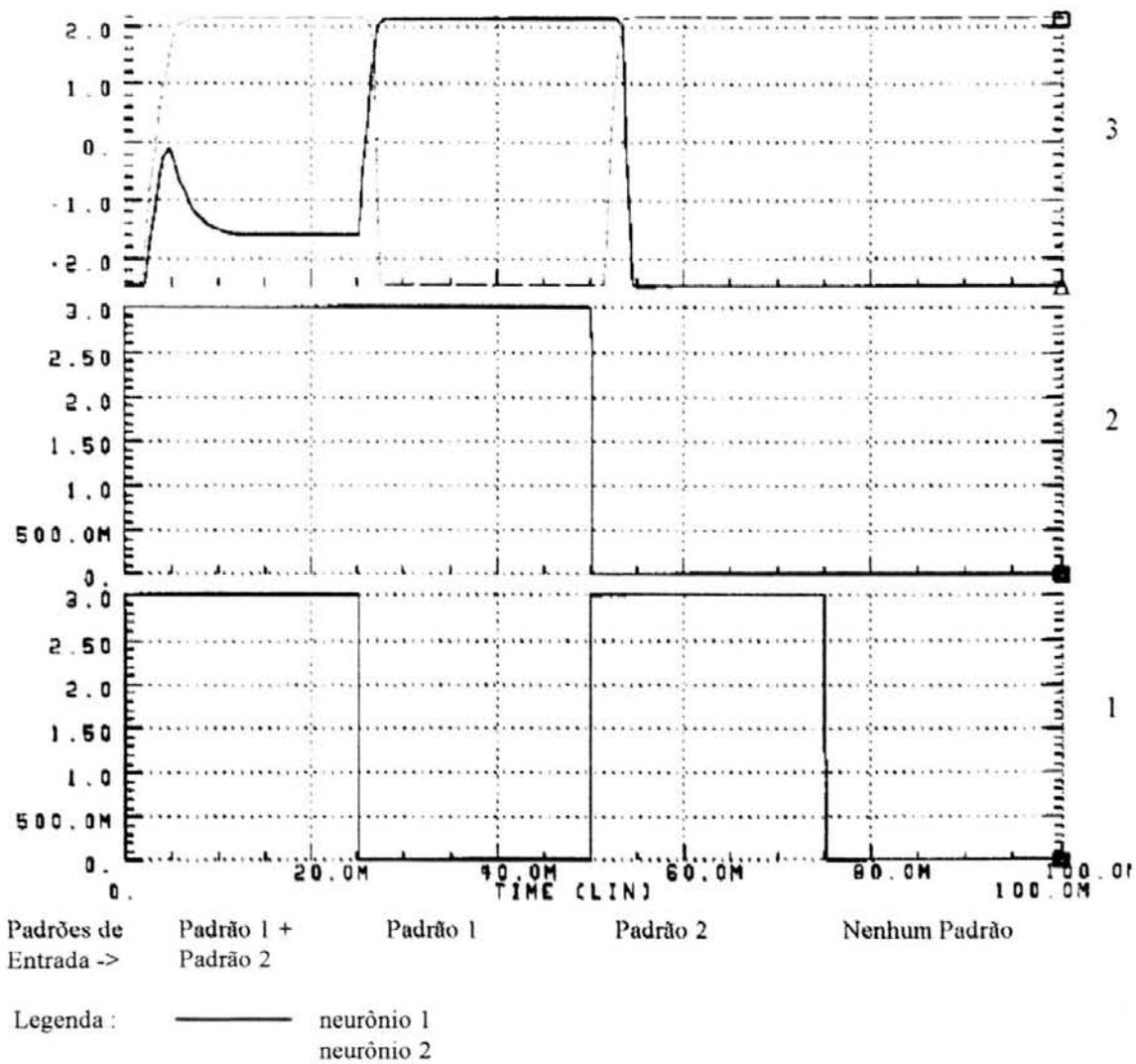


FIGURA 6.4 - Simulação da rede neural para a aplicação proposta.

Após estas simulações, simulou-se o mesmo circuito mostrado na figura 6.3 porém para uma tecnologia de 0.8μ .

O resultado destas simulações estão apresentados na figura 6.5. Nos gráficos 1 e 2 temos os padrões apresentados à rede neural, enquanto que, no gráfico 3, apresenta-se as saídas da rede neural.

A listagem contendo a descrição completa está contida no anexo 9.

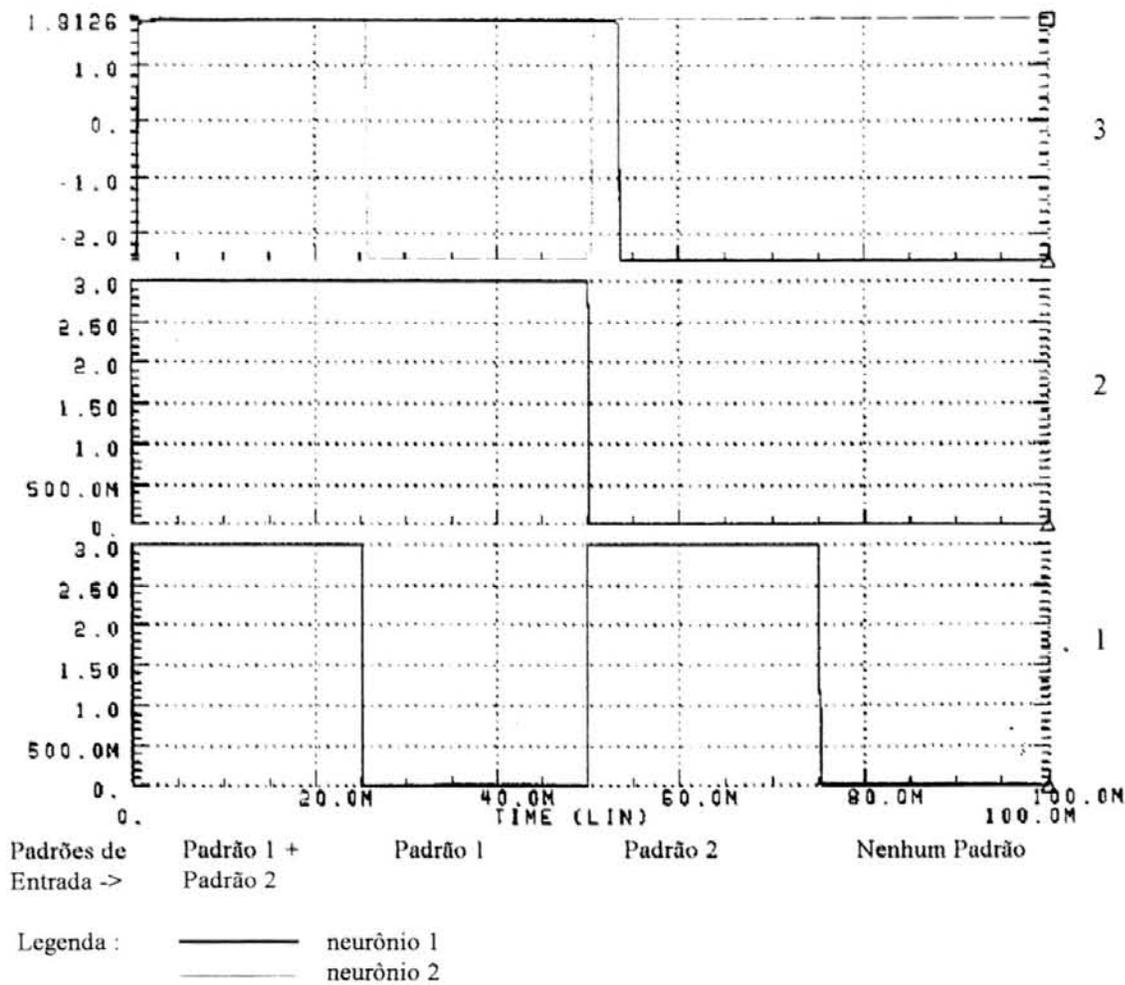


FIGURA 6.5 - Simulação da rede neural para a aplicação proposta para uma tecnologia de 0.8μ .

Como foi comentado no item 5.2 e no item 5.3 do capítulo 5, pode-se empregar transistores operando como resistores e como capacitores não lineares. Tal utilização foi testada empregando para isto o circuito da figura 6.6, onde pode-se observar que as resistências e capacitâncias responsáveis pelo decaimento passivo do sinal aplicado a entrada do neurônio, amplificador, foram trocadas por transistores MOS.

O resultado desta simulação pode ser vista na figura 6.7, gráfico 3. Nos gráficos 1 e 2 temos os padrões apresentados a rede neural.

Observa-se que com as alterações comentadas acima, o circuito proposto comportou-se de modo análogo ao circuito anterior que se utilizava de componentes do tipo resistor e capacitor.

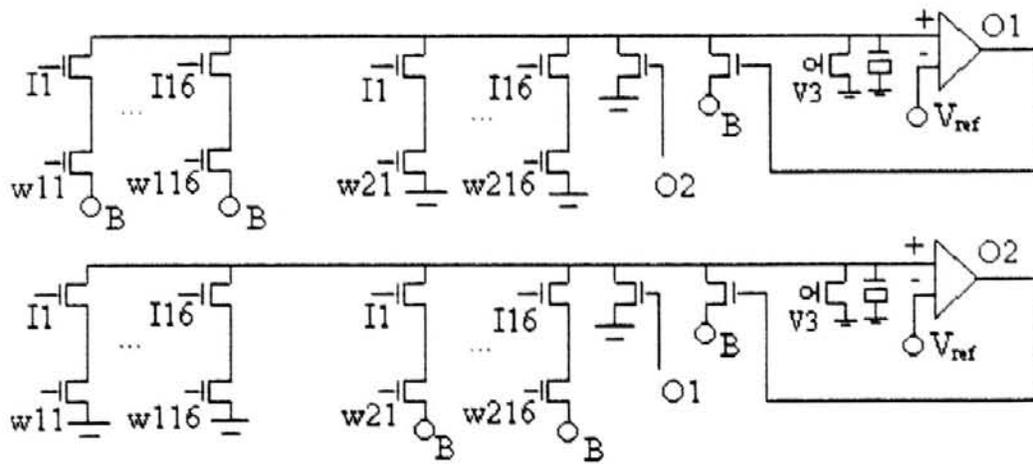


FIGURA 6.6 - Esquema elétrico do circuito para verificar a substituição de resistores e capacitores por transistores MOS.

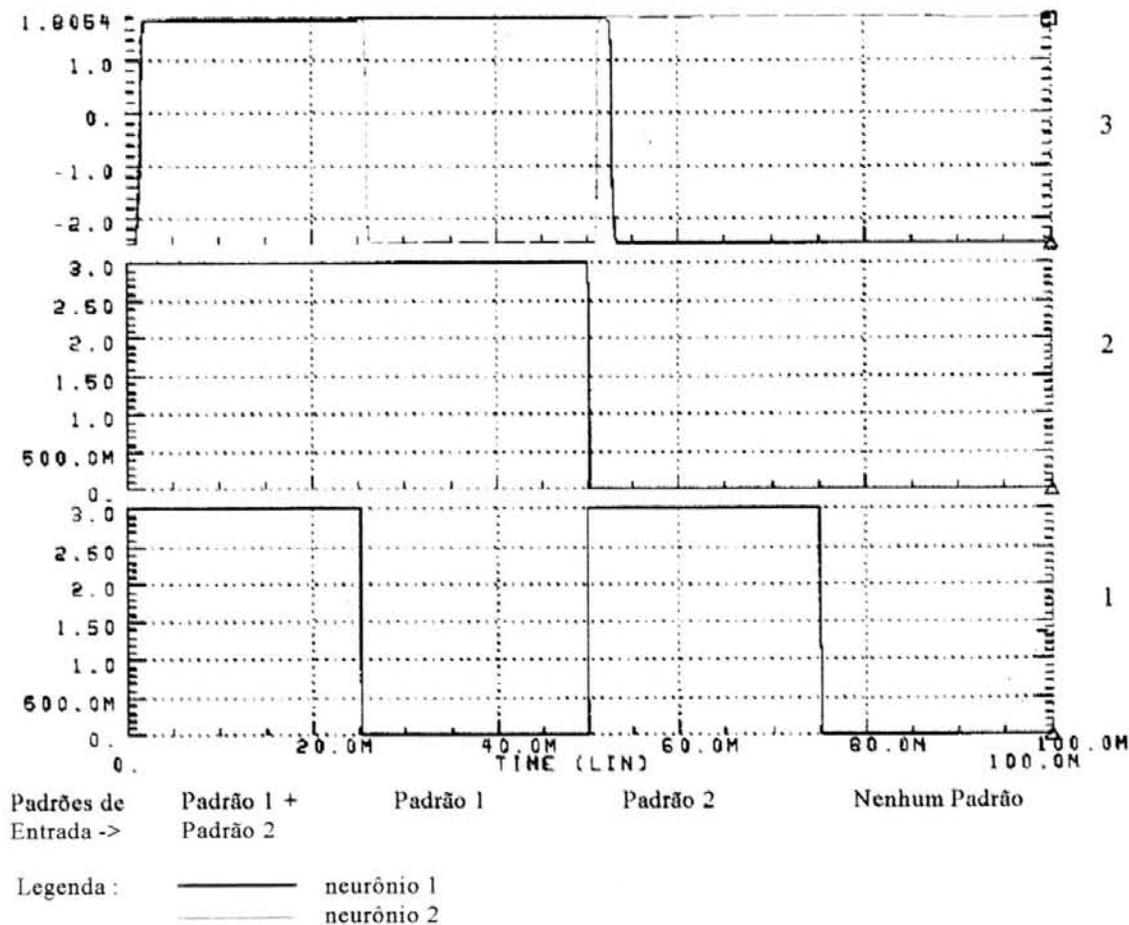


FIGURA 6.7 - Simulação da rede neural para a aplicação proposta para uma tecnologia de 0.8μ utilizando transistores no lugar das resistências e capacitâncias.

7 Conclusão

Neste trabalho apresentou-se uma nova proposta de implementação de redes neurais competitivas utilizando técnicas de circuitos integrados analógicos.

Realizaram-se diversas simulações elétricas para se comprovar esta proposta. Tais simulações foram realizadas através do simulador de circuitos elétricos HSPICE. Os modelos aqui mostrados, foram desenvolvidos utilizando-se componentes eletrônicos básicos, o que demonstra a simplicidade do modelo proposto. Juntamente com tais simulações, foram realizadas simulações a nível computacional, através do desenvolvimento de um aplicativo que solucionou as equações diferenciais. Tal aplicativo tem a sua importância na comparação dos resultados obtidos entre os modelos elétricos aqui propostos e as equações aqui mostradas durante o capítulo quatro.

A seguir apresentou-se um sistema prático utilizando as técnicas demonstradas nos capítulos anteriores. Tal sistema foi utilizado para clusterizar até dois padrões distintos. Foram realizadas simulações que comprovaram o funcionamento do sistema proposto através da correta clusterização dos padrões apresentados.

Os resultados das simulações, também demonstraram que a substituição dos resistores e capacitores por transistores operando como resistores e capacitores não lineares, é vantajosa pela diminuição de área do circuito sem acarretar numa degradação da resposta do sistema.

As simulações a nível elétrico, aqui comentadas no capítulo 5 e no capítulo 6, dos modelos competitivos, serviram mais uma vez para comprovar o comportamento dinâmico de tais redes bem como a descrição destes comportamentos dinâmicos através das equações propostas. Com os circuitos aqui apresentados, conseguiu-se de uma forma fácil propor uma nova técnica de implementação de redes neurais competitivas do tipo *shunting*. Contudo, as simulações a nível elétrico obtidas, podem não representar uma igualdade de resultados em caso de uma real implementação em *hardware* destes circuitos aqui propostos e devidamente simulados. Tal fato pode ser devido a exatidão de alguns dos valores de tensão exigidos nos diversos pontos do circuito, como por exemplo, a entrada inversora do amplificador diferencial.

Com isto, pode-se dizer que o ponto mais crítico nas implementações de tais modelos de redes neurais artificiais se encontra com relação a precisão das tensões que devem ser aplicadas em seus diversos pontos, portanto, deve-se realizar um estudo mais profundo em um futuro trabalho.

É importante ressaltar o fato de que durante as simulações elétricas houve o ajuste manual dos pesos. A característica de aprendizado destas redes neurais não foi investigado. Com isso dedicou-se um maior tempo para as diversas simulações realizadas. Esta característica de aprendizado deve futuramente, num trabalho próximo, ser analisada e devidamente implementada. Apenas como um ponto de partida, pode-se consultar [MON97] e [HOL94].

Como sequência deste trabalho, está sendo desenhado o *lay-out* de um *chip*, correspondente a aplicação mostrada no capítulo 6, para ser realizado em uma *foundry*.

Salienta-se, também, que estas redes são de fácil desenho a nível de *lay-out* para futura implementação em *hardware*, devido ao fato de se conseguir criar blocos que podem ser repetidos conforme o número de neurônios existentes na rede neural. Contudo, com o aumento do número de neurônios pertencentes a rede neural, observa-se um problema com relação a implementação dos pesos sinápticos, formada pelos transistores CMOS, e também do roteamento necessário para a completa interligação, tornando-se um fator limitante do tamanho da rede a ser implementada em *hardware*. Com isso, conclui-se que embora tenha se discutido sobre algumas das possíveis implementações de redes neurais artificiais em um circuito integrado, há um problema dentre outros, que é limitador para o tamanho das redes neurais, é ele: o número de conexões entre os neurônios.

Este problema é devido a grande quantidade de conexões entre os neurônios que dificultam o roteamento dos sinais entre os neurônios do circuito integrado, aumentando a sua área. O aumento de área do circuito tem reflexos nos custos e no desempenho do chip, com a diminuição da frequência de operação do mesmo. O aumento da área provoca além do aumento de custos, um aumento do consumo de potência do circuito integrado, trazendo problemas na dissipação de potência e aumento dos custos de encapsulamento do circuito. Além disso, um maior número de conexões entre os neurônios do chip, diminui a frequência de operação do circuito por aumentar as capacitâncias parasitas do circuito.

Anexo 1 Listagem C para redes neurais *shunting feedforward*

```

#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <time.h>
#include <bios.h>
#include <graphics.h>
#define VEZES 500 /* numero de vezes a propagar no tempo continuo */
#define CAM_IN 2 /* numero de neuronios na camada de entrada */
#define RO 1.e+4 /* valor da resistencia parasita */
#define CAPAC 1.e-7 /* valor da capacitancia parasita */
#define DELTA_T 1.e-4 /* valor do incremento do tempo */
#define B 3.0 /* valor da fonte de alimentaçãEo */
void titulo(void);
void propagacao(void);
void grafico(void);
void gera_view(int,int,int,int);

int cores[4] = {5,1,1,0};
float padrao[CAM_IN]; /* padrao de entrada para reconhecimento */
float on_center; /* realimentacoes positivas */
float off_surround; /* realimentacoes negativas */
float entrada[VEZES][CAM_IN]; /* valor na entrada do amplificador operac */
float somat[VEZES][CAM_IN]; /* variavel auxiliar */
float max=0.0,min=99.0;
int itera; /* numero corrente da iteracao */
int graphdriver = DETECT;
int graphmode;
/* */
/* PROGRAMA PRINCIPAL */
/* */
void main(void)
{
int x,y;
float aux;
clrscr();
titulo();
padrao[0]=0.3;
padrao[1]=1.8;
for(x=0;x<VEZES;x++)
for(y=0;y<CAM_IN;y++)
entrada[x][y] = 0.0;

```

```

for(itera=0;itera<VEZES;itera++) {
    propagacao();
    if(itera>=(VEZES/1.2)) {
        padrao[0] = padrao[1] = 0.0;
    }
    if(itera==(VEZES/2)) {
        aux = padrao[0];
        padrao[0] = padrao[1];
        padrao[1] = aux;
    }
}
grafico();
closegraph();
}
/*          */
/* ROTINAS PARA USO GERAL*/
/*          */
void titulo(void)
{
    gotoxy(15,1);
    printf("PROGRAMA PARA SIMULACAO DE UMA REDE SHUNTING FEED-
FORWARD");
}
/*          */
/* ROTINAS DE CALCULO */
/*          */
void propagacao(void)
{
    int x,y,z;
    for (x=0;x<CAM_IN;x++) {
        on_center = 0.0;
        off_surround = 0.0;
        for(y=0;y<CAM_IN;y++)
            if(y==x)
                on_center += (padrao[y])/(CAPAC*10000);
        for(z=0;z<CAM_IN;z++)
            if(z != x)
                off_surround += (padrao[z])/(CAPAC*10000);
        somat[itera][x] = - entrada[itera][x]/(RO*CAPAC) + (on_center*(B-
entrada[itera][x])) - (off_surround*entrada[itera][x]);
        entrada[itera+1][x] = entrada[itera][x] + DELTA_T * somat[itera][x];
        if(entrada[itera][x] < min)
            min = entrada[itera][x];
        if(entrada[itera][x] > max)
            max = entrada[itera][x];
    }
}
}
/* Rotina para inicializacao grafica */
void grafico(void)

```

```

{
int x,y,z,aux;
char str[6],buf[10],buf1[22];
detectgraph(&graphdriver,&graphmode);
initgraph(&graphdriver,&graphmode,"");
cleardevice();
setbkcolor(WHITE);
setcolor(BLUE);
outtextxy(150,5,"SIMULACAO DE UMA REDE SHUNTING FEED-FORWARD");
sprintf(buf,"%f",max);
outtextxy(30,100,buf);
sprintf(buf,"%f",min);
outtextxy(30,300,buf);
sprintf(buf1,"Troca em => %f",VEZES/2.0);
outtextxy(30,350,buf1);
sprintf(buf1,"STM em => %f",VEZES/1.2);
outtextxy(30,380,buf1);
gera_view(100,100,600,300);
for(z=0;z<CAM_IN;z++) {
    y = 0;
    setcolor(cores[z]);
    for(x=0;x<VEZES;x++) {
        aux = ((entrada[x][z]-min)*(300-100)/(max-min))+(100-300);
        if(y == 0){
            putpixel(y,fabs(aux),cores[z]);
            moveto(y,fabs(aux));
        }
        else {
            lineto(y,fabs(aux));
            putpixel(y,fabs(aux),cores[z]);
        }
        y += 1;
    }
    getch();
}
clearviewport();
}

void gera_view(int x1,int y1,int x2,int y2)
{
int x;
setcolor(YELLOW);
setcolor(RED);
rectangle(x1-1,y1-1,x2+1,y2+1);
setviewport(x1,y1,x2,y2,0);
setcolor(RED);
line(VEZES/2,0,VEZES/2,(300-100));
line(VEZES/1.2,0,VEZES/1.2,(300-100));
}

```

Anexo 2 Listagem C para redes neurais *shunting feedback*

```

#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <time.h>
#include <bios.h>
#include <graphics.h>
#define VEZES 500 /* numero de vezes a propagar no tempo continuo */
#define CAM_IN 2 /* numero de neuronios na camada de entrada */
#define CAM_COMP 2 /* numero de neuronios na camada competitiva */
#define RO 1.e+5 /* valor da resistencia parasita */
#define CAPAC 1.e-6 /* valor da capacitancia parasita */
#define GANHO 10. /* valor do ganho do amplificador */
#define DELTA_T 1.e-4 /* valor do incremento do tempo */
#define B 3.0 /* valor da fonte de alimentacao */
void titulo(void);
void propagacao(void);
void grafico(void);
void gera_view(int,int,int,int);

int cores[4] = {5,1,1,0};
float saida[VEZES][CAM_COMP]; /* valor na saida do amplif. operac.
                                prim.indice: numero de iteracoes
                                seg. indice: num. de neuronios */
float padrao[CAM_IN]; /* padrao de entrada para reconhecimento */
float on_center; /* realimentacoes positivas */
float off_surround; /* realimentacoes negativas */
float entrada[VEZES][CAM_COMP]; /* valor na entrada do amplificador operac */
float somat[VEZES][CAM_COMP]; /* variavel auxiliar */
float min=99.0,max=0.0;
int itera; /* numero corrente da iteracao */
int graphdriver = DETECT;
int graphmode;
/* */
/* PROGRAMA PRINCIPAL */
/* */
void main(void)
{
int x,y;
float aux;
clrscr();
titulo();

```

```

padrao[0]=1.0;
padrao[1]=2.0;
for(x=0;x<VEZES;x++)
  for(y=0;y<CAM_COMP;y++) {
    entrada[x][y] = 0.0;
    saida[x][y] = 0.0;
  }
for(itera=0;itera<VEZES;itera++) {
  propagacao();
  if(itera>=(VEZES/1.2)) {
    padrao[0]=0.0;
    padrao[1]=0.0;
  }
  if(itera==(VEZES/2)) {
    aux = padrao[0];
    padrao[0] = padrao[1];
    padrao[1] = aux;
  }
}
grafico();
closegraph();
}
/*          */
/* ROTINAS PARA USO GERAL*/
/*          */
void titulo(void)
{
gotoxy(15,1);
printf("PROGRAMA PARA SIMULACAO DE UMA REDE SHUNTING
FEEDBACK");
}
/*          */
/* ROTINAS DE CALCULO */
/*          */
void propagacao(void)
{
int x,y,z;
float on,off;
for (x=0;x<CAM_COMP;x++) {
  on_center = 0.0;
  off_surround = 0.0;
  on = 0.0;
  off = 0.0;
  for(y=0;y<CAM_IN;y++)
    if(y==x)
      on_center += padrao[y]/(CAPAC*10000);
  for(z=0;z<CAM_IN;z++)
    if(z != x)
      off_surround += padrao[z]/(CAPAC*10000);
}
}

```

```

for(y=0;y<CAM_COMP;y++)
    if(y != x)
        off += saida[itera][y];
on += saida[itera][x];
somat[itera][x] = - entrada[itera][x]/(RO*CAPAC) + ((on_center+on)*(B-
entrada[itera][x])) - ((off_surround+off)*entrada[itera][x]) ;
entrada[itera+1][x] = entrada[itera][x] + DELTA_T * somat[itera][x];
saida[itera+1][x] = 1/(1+exp(-GANHO*(entrada[itera][x]-1.0)));
if(saida[itera+1][x] < min)
    min = saida[itera][x];
if(saida[itera+1][x] > max)
    max = saida[itera][x];
}
}
/* Rotina para inicializacao grafica */
void grafico(void)
{
int x,y,z,aux;
char str[6],buf[10],buf1[22];
detectgraph(&graphdriver,&graphmode);
initgraph(&graphdriver,&graphmode,"");
cleardevice();
setbkcolor(WHITE);
setcolor(BLUE);
outtextxy(150,5,"SIMULACAO DE UMA REDE SHUNTING FEEDBACK");
sprintf(buf,"%f",max);
outtextxy(30,100,buf);
sprintf(buf,"%f",min);
outtextxy(30,300,buf);
sprintf(buf1,"Troca em => %f",VEZES/2.0);
outtextxy(30,350,buf1);
sprintf(buf1,"STM em => %f",VEZES/1.2);
outtextxy(30,380,buf1);
gera_view(100,100,600,300);
for(z=0;z<CAM_COMP;z++) {
    y = 0;
    setcolor(cores[z]);
    for(x=0;x<VEZES;x++) {
        aux = ((saida[x][z]-min)*(300-100)/(max-min))+(100-300);
        if(y == 0){
            putpixel(y,fabs(aux),cores[z]);
            moveto(y,fabs(aux));
        }
        else {
            lineto(y,fabs(aux));
            putpixel(y,fabs(aux),cores[z]);
        }
        y += 1;
    }
}
}

```

```
    getch();
}
clearviewport();
}

void gera_view(int x1,int y1,int x2,int y2)
{
int x;
setcolor(YELLOW);
setcolor(RED);
rectangle(x1-1,y1-1,x2+1,y2+1);
setviewport(x1,y1,x2,y2,0);
setcolor(RED);
line(VEZES/2,0,VEZES/2,(300-100));
line(VEZES/1.2,0,VEZES/1.2,(300-100));
}
```

Anexo 3 Listagem C para redes neurais *shunting feedback* com pesos

```

#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <time.h>
#include <bios.h>
#include <graphics.h>
#define VEZES 500 /* numero de vezes a propagar no tempo continuo */
#define CAM_IN 2 /* numero de neuronios na camada de entrada */
#define CAM_COMP 3 /* numero de neuronios na camada competitiva */
#define RO 1.e+5 /* valor da resistencia parasita */
#define CAPAC 1.e-6 /* valor da capacitancia parasita */
#define GANHO 3.5 /* valor do ganho do amplificador */
#define DELTA_T 1.e-4 /* valor do incremento do tempo */
#define B 1.0 /* valor da fonte de alimentacao Eo */
void titulo(void);
void propagacao(void);
void grafico(void);
void gera_view(int,int,int,int);
int cores[4] = {5,10,1,0};
float w[CAM_COMP][CAM_IN]={{0.15,0.15},
                           {0.0,0.85},
                           {0.85,0.0}};
float saida[VEZES][CAM_COMP]; /* valor na saida do amplif. operac.
                               prim.indice: numero de iteracoes
                               seg. indice: num. de neuronios */
float padrao[CAM_IN]; /* padrao de entrada para reconhecimento */
float on_center; /* realimentacoes positivas */
float off_surround; /* realimentacoes negativas */
float entrada[VEZES][CAM_COMP]; /* valor na entrada do amplificador operac */
float somat[VEZES][CAM_COMP]; /* variavel auxiliar */
float min=99.0,max=0.0;
int itera; /* numero corrente da iteracao */
int graphdriver = DETECT;
int graphmode;
/* */
/* PROGRAMA PRINCIPAL */
/* */
void main(void)
{
int x,y;
float aux;

```

```

clrscr();
titulo();
padrao[0]=1.8;
padrao[1]=0.3;
for(x=0;x<VEZES;x++)
  for(y=0;y<CAM_COMP;y++) {
    entrada[x][y] = 0.0;
    saida[x][y] = 0.0;
  }
for(itera=0;itera<VEZES;itera++) {
  propagacao();
  if(itera>=(VEZES/1.2)) {
    padrao[0]=0.0;
    padrao[1]=0.0;
  }
  if(itera==(VEZES/5)) {
    aux = padrao[0];
    padrao[0] = padrao[1];
    padrao[1] = aux;
  }
}
grafico();
closegraph();
}
/*          */
/* ROTINAS PARA USO GERAL */
/*          */
void titulo(void)
{
gotoxy(15,1);
printf("PROGRAMA PARA SIMULACAO DE UMA REDE CPN (SEM
APRENDIZADO)");
}
/*          */
/* ROTINAS DE CALCULO */
/*          */
void propagacao(void)
{
int x,y,z;
float on,off;
for (x=0;x<CAM_COMP;x++) {
  on_center = 0.0;
  off_surround = 0.0;
  on = 0.0;
  off = 0.0;
  for(y=0;y<CAM_IN;y++)
    on_center += (padrao[y]*w[x][y])/(CAPAC*10000);
  for(z=0;z<CAM_COMP;z++){
    for(y=0;y<CAM_IN;y++)

```

```

        if(z != x)
            off_surround += (padrao[y]*w[z][y])/(CAPAC*10000);
    }
    for(y=0;y<CAM_COMP;y++)
        if(y != x)
            off += saida[itera][y];
            on += saida[itera][x];
            somat[itera][x] = - entrada[itera][x]/(RO*CAPAC) + ((on_center+on)*(B-
            entrada[itera][x])) - ((off_surround+off)*entrada[itera][x]);
            entrada[itera+1][x] = entrada[itera][x] + DELTA_T * somat[itera][x];
            saida[itera+1][x] = tanh(GANHOS * entrada[itera][x]);
            if(saida[itera+1][x] < min)
                min = saida[itera][x];
            if(saida[itera+1][x] > max)
                max = saida[itera][x];
    }
}
/* Rotina para inicializacao grafica */
void grafico(void)
{
    int x,y,z,aux;
    char str[6],buf[10],buf1[22];
    detectgraph(&graphdriver,&graphmode);
    initgraph(&graphdriver,&graphmode,"");
    cleardevice();
    setbkcolor(WHITE);
    setcolor(BLUE);
    outtextxy(150,5,"SIMULACAO DE UMA REDE CPN (SEM APRENDIZADO)");
    sprintf(buf,"%f",max);
    outtextxy(30,100,buf);
    sprintf(buf,"%f",min);
    outtextxy(30,300,buf);
    sprintf(buf1,"Troca em => %f",VEZES/5.0);
    outtextxy(30,350,buf1);
    sprintf(buf1,"STM em => %f",VEZES/1.2);
    outtextxy(30,380,buf1);
    gera_view(100,100,600,300);
    for(z=0;z<CAM_COMP;z++) {
        y = 0;
        setcolor(cores[z]);
        for(x=0;x<VEZES;x++) {
            aux = ((saida[x][z]-min)*(300-100)/(max-min))+(100-300);
            if(y == 0){
                putpixel(y,fabs(aux),cores[z]);
                moveto(y,fabs(aux));
            }
            else {
                lineto(y,fabs(aux));
                putpixel(y,fabs(aux),cores[z]);
            }
        }
    }
}

```

```
    }  
    y += 1;  
  }  
  getch();  
}  
clearviewport();  
}
```

```
void gera_view(int x1,int y1,int x2,int y2)  
{  
  int x;  
  setcolor(YELLOW);  
  setcolor(RED);  
  rectangle(x1-1,y1-1,x2+1,y2+1);  
  setviewport(x1,y1,x2,y2,0);  
  setcolor(RED);  
  line(VEZES/5,0,VEZES/5,(300-100));  
  line(VEZES/1.2,0,VEZES/1.2,(300-100));  
}
```

Anexo 4 Descrição elétrica de um transistor

Simulação das características de um transistor operando como um resistor

*

* modelos dos transistores

*

```
.model nmos nmos level=2 ld=0.325u tox=250e-10 nsub=2e16 vto=0.7
+uo=510 uexp=0.22 ucrit=24.3k delta=0.4 xj=0.4u vmax=54k neff=4
+rsh=55 nfs=0 js=2u cj=130u cjsw=620p mj=0.53 mjsw=0.53 pb=0.68v
+cgdo=320p cgso=320p
```

*

```
.model pmos pmos level=2 ld=0.3u tox=250e-10 nsub=5e16 vto=-1.1
+uo=210 uexp=0.33 ucrit=51k delta=0.4 xj=0.5u vmax=47k neff=0.88
+rsh=75 nfs=0 js=10u cj=490u cjsw=590p mj=0.46 mjsw=0.46 pb=0.78v
+cgdo=320p cgso=320p
```

*

```
m1 2 1 3 3 nmos l=5u w=10u
```

```
vs1 3 0 dc 0
```

```
vgs1 1 0 0
```

```
vds1 2 0 0
```

```
.dc vds1 0 3.0 0.1 vgs1 1.0 2.5 0.5
```

```
.option limpts=1000 post
```

```
.print dc i(vds1)
```

```
.end
```

Anexo 5 Descrição elétrica de uma rede neural *shunting feedforward*

Simulacao de uma rede neural Shunting Feedforward com 2 neuronios

*

* modelos dos transistores

*

```
.model nmos nmos level=2 ld=0.325u tox=250e-10 nsub=2e16 vto=0.7
+uo=510 uexp=0.22 ucrit=24.3k delta=0.4 xj=0.4u vmax=54k neff=4
+rsh=55 nfs=0 js=2u cj=130u cjsw=620p mj=0.53 mjsw=0.53 pb=0.68v
+cgdo=320p cgso=320p
```

*

```
.model pmos pmos level=2 ld=0.3u tox=250e-10 nsub=5e16 vto=-1.1
+uo=210 uexp=0.33 ucrit=51k delta=0.4 xj=0.5u vmax=47k neff=0.88
+rsh=75 nfs=0 js=10u cj=490u cjsw=590p mj=0.46 mjsw=0.46 pb=0.78v
+cgdo=320p cgso=320p
```

*

```
.subckt opamp 3 8
```

```
mp1 1 4 4 1 pmos l=1.2u w=12u
```

```
mp2 1 4 7 1 pmos l=1.2u w=12u
```

```
mp3 1 7 8 1 pmos l=1.2u w=36u
```

```
mn1 4 2 5 5 nmos l=1.2u w=36u
```

```
mn2 7 3 5 5 nmos l=1.2u w=36u
```

```
mn3 8 6 9 9 nmos l=36u w=36u
```

```
mn4 5 6 9 9 nmos l=12u w=12u
```

```
ro 3 0 1e+4
```

```
ca 3 0 1e-7
```

```
vdd 1 0 dc 2.5
```

```
vss 0 9 dc 2.5
```

```
vi- 2 0 dc 0.2
```

```
vbi 6 0 dc 2.5
```

```
.ends opamp
```

*

```
mp1 2 3 1 1 nmos w=10u l=5u
```

```
mn1 1 6 0 0 nmos w=10u l=5u
```

```
mp2 2 6 5 5 nmos w=10u l=5u
```

```
mn2 5 3 0 0 nmos w=10u l=5u
```

```
xop1 1 4 opamp
```

```
xop2 5 7 opamp
```

```
vbi 2 0 dc 2.5
```

```
vi1 3 0 pulse (0 3 0n 0n 0n 10m)
```

```
vi2 6 0 pulse (0 3 0n 0n 0n 5m 10m)
```

```
.tran 0.1m 20m
```

```
.options post nomod nopage
```

```
.end
```

Anexo 6 Descrição elétrica de uma rede neural *shunting feedback*

Simulacao de uma rede neural Shunting Feedback com 2 neuronios

*

* modelos dos transistores

*

```
.model nmos nmos level=2 ld=0.325u tox=250e-10 nsub=2e16 vto=0.7
+uo=510 uexp=0.22 ucrit=24.3k delta=0.4 xj=0.4u vmax=54k neff=4
+rsh=55 nfs=0 js=2u cj=130u cjsw=620p mj=0.53 mjsw=0.53 pb=0.68v
+cgdo=320p cgso=320p
```

*

```
.model pmos pmos level=2 ld=0.3u tox=250e-10 nsub=5e16 vto=-1.1
+uo=210 uexp=0.33 ucrit=51k delta=0.4 xj=0.5u vmax=47k neff=0.88
+rsh=75 nfs=0 js=10u cj=490u cjsw=590p mj=0.46 mjsw=0.46 pb=0.78v
+cgdo=320p cgso=320p
```

*

.subckt opamp 3 8

```
mp1 1 4 4 1 pmos l=1.2u w=12u
mp2 1 4 7 1 pmos l=1.2u w=12u
mp3 1 7 8 1 pmos l=1.2u w=36u
mn1 4 2 5 5 nmos l=1.2u w=36u
mn2 7 3 5 5 nmos l=1.2u w=36u
mn3 8 6 9 9 nmos l=36u w=36u
mn4 5 6 9 9 nmos l=12u w=12u
ro 3 0 1e+4
ca 3 0 1e-6
vdd 1 0 dc 2.5
vss 0 9 dc 2.5
vi- 2 0 dc 0.2
vbi 6 0 dc 2.5
.ends opamp
```

*

```
mp1 2 3 1 1 nmos w=50u l=5u
mpf1 2 4 1 1 nmos w=50u l=5u
mn1 1 6 0 0 nmos w=50u l=5u
mnf1 1 7 0 0 nmos w=50u l=5u
mp2 2 6 5 5 nmos w=50u l=5u
mpf2 2 7 5 5 nmos w=50u l=5u
mn2 5 3 0 0 nmos w=50u l=5u
mnf2 5 4 0 0 nmos w=50u l=5u
xop1 1 4 opamp
xop2 5 7 opamp
vb1 2 0 dc 2.5
vi1 3 0 pulse (0 3 0n 0n 0n 5m 10m)
vi2 6 0 pulse (0 3 0n 0n 0n 10m)
```

```
*  
.tran 0.1m 20m  
.options post nomod nopage  
.measure tot_power avg power  
.print tran v(1) v(5) v(4) v(7)  
.end
```

Anexo 7 Descrição elétrica de uma rede neural *shunting feedback* com pesos

Simulacao de uma rede neural Shunting Feedback com pesos (com 3 neuronios)

*

* modelos dos transistores

*

```
.model nmos nmos level=2 ld=0.325u tox=250e-10 nsub=2e16 vto=0.7
+uo=510 uexp=0.22 ucrit=24.3k delta=0.4 xj=0.4u vmax=54k neff=4
+rsh=55 nfs=0 js=2u cj=130u cjsw=620p mj=0.53 mjsw=0.53 pb=0.68v
+cgdo=320p cgso=320p
```

*

```
.model pmos pmos level=2 ld=0.3u tox=250e-10 nsub=5e16 vto=-1.1
+uo=210 uexp=0.33 ucrit=51k delta=0.4 xj=0.5u vmax=47k neff=0.88
+rsh=75 nfs=0 js=10u cj=490u cjsw=590p mj=0.46 mjsw=0.46 pb=0.78v
+cgdo=320p cgso=320p
```

*

```
.subckt opamp 3 8
```

```
mp1 1 4 4 1 pmos l=1.2u w=12u
```

```
mp2 1 4 7 1 pmos l=1.2u w=12u
```

```
mp3 1 7 8 1 pmos l=1.2u w=36u
```

```
mn1 4 2 5 5 nmos l=1.2u w=36u
```

```
mn2 7 3 5 5 nmos l=1.2u w=36u
```

```
mn3 8 6 9 9 nmos l=36u w=36u
```

```
mn4 5 6 9 9 nmos l=12u w=12u
```

```
ro 3 0 1e+4
```

```
ca 3 0 1e-6
```

```
vdd 1 0 dc 2.5
```

```
vss 0 9 dc 2.5
```

```
vi- 2 0 dc 0.3
```

```
vbi 6 0 dc 5.0
```

```
.ends opamp
```

* interconecoos para o primeiro neuronio

```
M1p11 5 3 1 1 nmos w=10e-6 l=5e-6
```

```
M1pw11 15 11 5 5 nmos w=10e-6 l=5e-6
```

```
M1p12 6 4 1 1 nmos w=10e-6 l=5e-6
```

```
M1pw12 15 12 6 6 nmos w=10e-6 l=5e-6
```

```
M1n21 1 3 7 7 nmos w=10e-6 l=5e-6
```

```
M1nw21 7 21 0 0 nmos w=10e-6 l=5e-6
```

```
M1n22 1 4 8 8 nmos w=10e-6 l=5e-6
```

```
M1nw22 8 22 0 0 nmos w=10e-6 l=5e-6
```

```
M1n31 1 3 9 9 nmos w=10e-6 l=5e-6
```

```
M1nw31 9 31 0 0 nmos w=10e-6 l=5e-6
```

```
M1n32 1 4 10 10 nmos w=10e-6 l=5e-6
```

```
M1nw32 10 32 0 0 nmos w=10e-6 l=5e-6
```

```
M1pfl 15 2 1 1 nmos w=10e-6 l=5e-6
```

```

M1nf2 1 102 0 0 nmos w=10e-6 l=5e-6
M1nf3 1 202 0 0 nmos w=10e-6 l=5e-6
* interconecoos para o segundo neuronio
M2n11 101 3 150 150 nmos w=10e-6 l=5e-6
M2nw11 150 11 0 0 nmos w=10e-6 l=5e-6
M2n12 101 4 160 160 nmos w=10e-6 l=5e-6
M2nw12 160 12 0 0 nmos w=10e-6 l=5e-6
M2p21 170 3 101 101 nmos w=10e-6 l=5e-6
M2pw21 15 21 170 170 nmos w=10e-6 l=5e-6
M2p22 180 4 101 101 nmos w=10e-6 l=5e-6
M2pw22 15 22 180 180 nmos w=10e-6 l=5e-6
M2n31 101 3 190 190 nmos w=10e-6 l=5e-6
M2nw31 190 31 0 0 nmos w=10e-6 l=5e-6
M2n32 101 4 100 100 nmos w=10e-6 l=5e-6
M2nw32 100 32 0 0 nmos w=10e-6 l=5e-6
M2nf1 101 2 0 0 nmos w=10e-6 l=5e-6
M2pf2 15 102 101 101 nmos w=10e-6 l=5e-6
M2nf3 101 202 0 0 nmos w=10e-6 l=5e-6
* interconecoos para o terceiro neuronio
M3n11 201 3 250 250 nmos w=10e-6 l=5e-6
M3nw11 250 11 0 0 nmos w=10e-6 l=5e-6
M3n12 201 4 260 260 nmos w=10e-6 l=5e-6
M3nw12 260 12 0 0 nmos w=10e-6 l=5e-6
M3n21 201 3 270 270 nmos w=10e-6 l=5e-6
M3nw21 270 21 0 0 nmos w=10e-6 l=5e-6
M3n22 201 4 280 280 nmos w=10e-6 l=5e-6
M3nw22 280 22 0 0 nmos w=10e-6 l=5e-6
M3p31 290 3 201 201 nmos w=10e-6 l=5e-6
M3pw31 15 31 290 290 nmos w=10e-6 l=5e-6
M3p32 200 4 201 201 nmos w=10e-6 l=5e-6
M3pw32 15 32 200 200 nmos w=10e-6 l=5e-6
M3nf1 201 2 0 0 nmos w=10e-6 l=5e-6
M3nf2 201 102 0 0 nmos w=10e-6 l=5e-6
M3pf3 15 202 201 201 nmos w=10e-6 l=5e-6
xop1 1 2 opamp
xop2 101 102 opamp
xop3 201 202 opamp
Vb1 15 0 dc 2.5
Vw11 11 0 dc 2.6
Vw12 12 0 dc 1.0
Vw21 21 0 dc 0.93
Vw22 22 0 dc 2.6
Vw31 31 0 dc 2.26
Vw32 32 0 dc 2.3
*Vi1 3 0 pulse (0 3 0N 0N 0N 50e-3)
*Vi2 4 0 pulse (0 3 0N 0N 0N 25e-3 50e-3)
Vi1 3 0 pulse (0 3 0N 0N 0N 50e-3)
Vi2 4 0 pulse (0 3 0N 0N 0N 50e-3)
.tran 0.1e-3 100e-3

```

```
.options post nopage  
.END
```

Anexo 8 Descrição elétrica do sistema proposto

Simulacao de um aplicativo com 2 neuronios na camada competitiva

*

* modelos dos transistores

*

```
.model nmos nmos level=2 ld=0.325u tox=250e-10 nsub=2e16 vto=0.7
+uo=510 uexp=0.22 ucrit=24.3k delta=0.4 xj=0.4u vmax=54k neff=4
+rsh=55 nfs=0 js=2u cj=130u cjsw=620p mj=0.53 mjsw=0.53 pb=0.68v
+cgdo=320p cgso=320p
```

*

```
.model pmos pmos level=2 ld=0.3u tox=250e-10 nsub=5e16 vto=-1.1
+uo=210 uexp=0.33 ucrit=51k delta=0.4 xj=0.5u vmax=47k neff=0.88
+rsh=75 nfs=0 js=10u cj=490u cjsw=590p mj=0.46 mjsw=0.46 pb=0.78v
+cgdo=320p cgso=320p
```

*

.subckt opamp 3 8

mp1 1 4 4 1 pmos l=1.2u w=12u

mp2 1 4 7 1 pmos l=1.2u w=12u

mp3 1 7 8 1 pmos l=1.2u w=36u

mn1 4 2 5 5 nmos l=1.2u w=36u

mn2 7 3 5 5 nmos l=1.2u w=36u

mn3 8 6 9 9 nmos l=36u w=36u

mn4 5 6 9 9 nmos l=12u w=12u

ro 3 0 1e+4

ca 3 0 1e-6

vdd 1 0 dc 2.5

vss 0 9 dc 2.5

vi- 2 0 dc 0.3

vbi 6 0 dc 5.0

.ends opamp

* interconecoos para o primeiro neuronio

M1p11 51 1 201 201 nmos w=10e-6 l=5e-6

M1pw11 70 101 51 51 nmos w=10e-6 l=5e-6

M1p12 52 2 201 201 nmos w=10e-6 l=5e-6

M1pw12 70 102 52 52 nmos w=10e-6 l=5e-6

M1p13 53 3 201 201 nmos w=10e-6 l=5e-6

M1pw13 70 103 53 53 nmos w=10e-6 l=5e-6

M1p14 54 4 201 201 nmos w=10e-6 l=5e-6

M1pw14 70 104 54 54 nmos w=10e-6 l=5e-6

M1p15 55 5 201 201 nmos w=10e-6 l=5e-6

M1pw15 70 105 55 55 nmos w=10e-6 l=5e-6

M1p16 56 6 201 201 nmos w=10e-6 l=5e-6

M1pw16 70 106 56 56 nmos w=10e-6 l=5e-6

M1p17 57 7 201 201 nmos w=10e-6 l=5e-6

M1pw17 70 107 57 57 nmos w=10e-6 l=5e-6
 M1p18 58 8 201 201 nmos w=10e-6 l=5e-6
 M1pw18 70 108 58 58 nmos w=10e-6 l=5e-6
 M1p19 59 9 201 201 nmos w=10e-6 l=5e-6
 M1pw19 70 109 59 59 nmos w=10e-6 l=5e-6
 M1p110 60 10 201 201 nmos w=10e-6 l=5e-6
 M1pw110 70 110 60 60 nmos w=10e-6 l=5e-6
 M1p111 61 11 201 201 nmos w=10e-6 l=5e-6
 M1pw111 70 111 61 61 nmos w=10e-6 l=5e-6
 M1p112 62 12 201 201 nmos w=10e-6 l=5e-6
 M1pw112 70 112 62 62 nmos w=10e-6 l=5e-6
 M1p113 63 13 201 201 nmos w=10e-6 l=5e-6
 M1pw113 70 113 63 63 nmos w=10e-6 l=5e-6
 M1p114 64 14 201 201 nmos w=10e-6 l=5e-6
 M1pw114 70 114 64 64 nmos w=10e-6 l=5e-6
 M1p115 65 15 201 201 nmos w=10e-6 l=5e-6
 M1pw115 70 115 65 65 nmos w=10e-6 l=5e-6
 M1p116 66 16 201 201 nmos w=10e-6 l=5e-6
 M1pw116 70 116 66 66 nmos w=10e-6 l=5e-6
 M1n11 201 1 31 31 nmos w=10e-6 l=5e-6
 M1nw11 31 141 0 0 nmos w=10e-6 l=5e-6
 M1n12 201 2 32 32 nmos w=10e-6 l=5e-6
 M1nw12 32 142 0 0 nmos w=10e-6 l=5e-6
 M1n13 201 3 33 33 nmos w=10e-6 l=5e-6
 M1nw13 33 143 0 0 nmos w=10e-6 l=5e-6
 M1n14 201 4 34 34 nmos w=10e-6 l=5e-6
 M1nw14 34 144 0 0 nmos w=10e-6 l=5e-6
 M1n15 201 5 35 35 nmos w=10e-6 l=5e-6
 M1nw15 35 145 0 0 nmos w=10e-6 l=5e-6
 M1n16 201 6 36 36 nmos w=10e-6 l=5e-6
 M1nw16 36 146 0 0 nmos w=10e-6 l=5e-6
 M1n17 201 7 37 37 nmos w=10e-6 l=5e-6
 M1nw17 37 147 0 0 nmos w=10e-6 l=5e-6
 M1n18 201 8 38 38 nmos w=10e-6 l=5e-6
 M1nw18 38 148 0 0 nmos w=10e-6 l=5e-6
 M1n19 201 9 39 39 nmos w=10e-6 l=5e-6
 M1nw19 39 149 0 0 nmos w=10e-6 l=5e-6
 M1n110 201 10 40 40 nmos w=10e-6 l=5e-6
 M1nw110 40 150 0 0 nmos w=10e-6 l=5e-6
 M1n111 201 11 41 41 nmos w=10e-6 l=5e-6
 M1nw111 41 151 0 0 nmos w=10e-6 l=5e-6
 M1n112 201 12 42 42 nmos w=10e-6 l=5e-6
 M1nw112 42 152 0 0 nmos w=10e-6 l=5e-6

 M1n113 201 13 43 43 nmos w=10e-6 l=5e-6
 M1nw113 43 153 0 0 nmos w=10e-6 l=5e-6
 M1n114 201 14 44 44 nmos w=10e-6 l=5e-6
 M1nw114 44 154 0 0 nmos w=10e-6 l=5e-6
 M1n115 201 15 45 45 nmos w=10e-6 l=5e-6

M1nw115 45 155 0 0 nmos w=10e-6 l=5e-6
M1n116 201 16 46 46 nmos w=10e-6 l=5e-6
M1nw116 46 156 0 0 nmos w=10e-6 l=5e-6
M1pf1 70 211 201 201 nmos w=10e-6 l=5e-6
M1nf2 201 212 0 0 nmos w=10e-6 l=5e-6
* interconecoos para o segundo neuronio
M2p21 71 1 202 202 nmos w=10e-6 l=5e-6
M2pw21 70 141 71 71 nmos w=10e-6 l=5e-6
M2p22 72 2 202 202 nmos w=10e-6 l=5e-6
M2pw22 70 142 72 72 nmos w=10e-6 l=5e-6
M2p23 73 3 202 202 nmos w=10e-6 l=5e-6
M2pw23 70 143 73 73 nmos w=10e-6 l=5e-6
M2p24 74 4 202 202 nmos w=10e-6 l=5e-6
M2pw24 70 144 74 74 nmos w=10e-6 l=5e-6
M2p25 75 5 202 202 nmos w=10e-6 l=5e-6
M2pw25 70 145 75 75 nmos w=10e-6 l=5e-6
M2p26 76 6 202 202 nmos w=10e-6 l=5e-6
M2pw26 70 146 76 76 nmos w=10e-6 l=5e-6
M2p27 77 7 202 202 nmos w=10e-6 l=5e-6
M2pw27 70 147 77 77 nmos w=10e-6 l=5e-6
M2p28 78 8 202 202 nmos w=10e-6 l=5e-6
M2pw28 70 148 78 78 nmos w=10e-6 l=5e-6
M2p29 79 9 202 202 nmos w=10e-6 l=5e-6
M2pw29 70 149 79 79 nmos w=10e-6 l=5e-6
M2p210 80 10 202 202 nmos w=10e-6 l=5e-6
M2pw210 70 150 80 80 nmos w=10e-6 l=5e-6
M2p211 81 11 202 202 nmos w=10e-6 l=5e-6
M2pw211 70 151 81 81 nmos w=10e-6 l=5e-6
M2p212 82 12 202 202 nmos w=10e-6 l=5e-6
M2pw212 70 152 82 82 nmos w=10e-6 l=5e-6
M2p213 83 13 202 202 nmos w=10e-6 l=5e-6
M2pw213 70 153 83 83 nmos w=10e-6 l=5e-6
M2p214 84 14 202 202 nmos w=10e-6 l=5e-6
M2pw214 70 154 84 84 nmos w=10e-6 l=5e-6
M2p215 85 15 202 202 nmos w=10e-6 l=5e-6
M2pw215 70 155 85 85 nmos w=10e-6 l=5e-6
M2p216 86 16 202 202 nmos w=10e-6 l=5e-6
M2pw216 70 156 86 86 nmos w=10e-6 l=5e-6
M2n21 202 1 121 121 nmos w=10e-6 l=5e-6
M2nw21 121 101 0 0 nmos w=10e-6 l=5e-6
M2n22 202 2 122 122 nmos w=10e-6 l=5e-6
M2nw22 122 102 0 0 nmos w=10e-6 l=5e-6
M2n23 202 3 123 123 nmos w=10e-6 l=5e-6
M2nw23 123 103 0 0 nmos w=10e-6 l=5e-6
M2n24 202 4 124 124 nmos w=10e-6 l=5e-6
M2nw24 124 104 0 0 nmos w=10e-6 l=5e-6
M2n25 202 5 125 125 nmos w=10e-6 l=5e-6
M2nw25 125 105 0 0 nmos w=10e-6 l=5e-6
M2n26 202 6 126 126 nmos w=10e-6 l=5e-6

M2nw26 126 106 0 0 nmos w=10e-6 l=5e-6
 M2n27 202 7 127 127 nmos w=10e-6 l=5e-6
 M2nw27 127 107 0 0 nmos w=10e-6 l=5e-6
 M2n28 202 8 128 128 nmos w=10e-6 l=5e-6
 M2nw28 128 108 0 0 nmos w=10e-6 l=5e-6
 M2n29 202 9 129 129 nmos w=10e-6 l=5e-6
 M2nw29 129 109 0 0 nmos w=10e-6 l=5e-6
 M2n210 202 10 130 130 nmos w=10e-6 l=5e-6
 M2nw210 130 110 0 0 nmos w=10e-6 l=5e-6
 M2n211 202 11 131 131 nmos w=10e-6 l=5e-6
 M2nw211 131 111 0 0 nmos w=10e-6 l=5e-6
 M2n212 202 12 132 132 nmos w=10e-6 l=5e-6
 M2nw212 132 112 0 0 nmos w=10e-6 l=5e-6
 M2n213 202 13 133 133 nmos w=10e-6 l=5e-6
 M2nw213 133 113 0 0 nmos w=10e-6 l=5e-6
 M2n214 202 14 134 134 nmos w=10e-6 l=5e-6
 M2nw214 134 114 0 0 nmos w=10e-6 l=5e-6
 M2n215 202 15 135 135 nmos w=10e-6 l=5e-6
 M2nw215 135 115 0 0 nmos w=10e-6 l=5e-6
 M2n216 202 16 136 136 nmos w=10e-6 l=5e-6
 M2nw216 136 116 0 0 nmos w=10e-6 l=5e-6
 M2pf1 70 212 202 202 nmos w=10e-6 l=5e-6
 M2nf2 202 211 0 0 nmos w=10e-6 l=5e-6

xop1 201 211 opamp
 xop2 202 212 opamp
 Vb1 70 0 dc 2.5
 Vw11 101 0 dc 2.6
 Vw12 102 0 dc 1.0
 Vw13 103 0 dc 2.6
 Vw14 104 0 dc 1.0
 Vw15 105 0 dc 1.0
 Vw16 106 0 dc 1.0
 Vw17 107 0 dc 1.0
 Vw18 108 0 dc 1.0
 Vw19 109 0 dc 0.8
 Vw110 110 0 dc 0.8
 Vw111 111 0 dc 0.8
 Vw112 112 0 dc 0.8
 Vw113 113 0 dc 1.0
 Vw114 114 0 dc 1.0
 Vw115 115 0 dc 1.0
 Vw116 116 0 dc 1.0

Vw21 141 0 dc 0.9
 Vw22 142 0 dc 1.0
 Vw23 143 0 dc 0.9
 Vw24 144 0 dc 1.0
 Vw25 145 0 dc 1.0

```
Vw26 146 0 dc 1.0
Vw27 147 0 dc 1.0
Vw28 148 0 dc 1.0
Vw29 149 0 dc 2.6
Vw210 150 0 dc 1.0
Vw211 151 0 dc 2.6
Vw212 152 0 dc 1.0
Vw213 153 0 dc 1.0
Vw214 154 0 dc 1.0
Vw215 155 0 dc 1.0
Vw216 156 0 dc 1.0
* padroes de entrada
Vi1 1 0 pulse (0 0 0N 0N 0N 50e-3)
Vi2 2 0 pulse (0 0 0N 0N 0N 50e-3)
Vi3 3 0 pulse (0 0 0N 0N 0N 50e-3)
Vi4 4 0 pulse (0 0 0N 0N 0N 50e-3)
Vi5 5 0 pulse (0 0 0N 0N 0N 50e-3)
Vi6 6 0 pulse (0 0 0N 0N 0N 50e-3)
Vi7 7 0 pulse (0 0 0N 0N 0N 50e-3)
Vi8 8 0 pulse (0 0 0N 0N 0N 50e-3)
Vi9 9 0 pulse (0 3 0N 0N 0N 25e-3 50e-3)
Vi10 10 0 pulse (0 3 0N 0N 0N 25e-3 50e-3)
Vi11 11 0 pulse (0 3 0N 0N 0N 25e-3 50e-3)
Vi12 12 0 pulse (0 3 0N 0N 0N 25e-3 50e-3)
Vi13 13 0 pulse (0 0 0N 0N 0N 50e-3)
Vi14 14 0 pulse (0 0 0N 0N 0N 50e-3)
Vi15 15 0 pulse (0 0 0N 0N 0N 50e-3)
Vi16 16 0 pulse (0 0 0N 0N 0N 50e-3)
.tran 0.1e-3 100e-3
.options post nopage
.END
```

Anexo 9 Descrição elétrica do sistema proposto (tecnologia de 0.8μ)

Simulacao de um aplicativo com 2 neuronios na camada competitiva

```
.MODEL nmos NMOS LEVEL=47
+SUBTHMOD=3.000e+00 SATMOD =2.000e+00 BULKMOD=1.000e+00
*   *** Threshold voltage related model parameters ***
+
+K3   =7.546e+00 K3B   =-1.66e+00
+NPEAK =6.500e+16 VTH0  =8.300e-01
+VOFF  =-6.22e-02 DVT0  =1.000e+00 DVT1  =1.120e+00
+DVT2  =-6.98e-01 KETA  =-5.90e-02
+VBM   =-5.00e+00 PSCBE1=4.700e+08 PSCBE2 =1.000e-05
*   *** Mobility related model parameters ***
+UA    =0.000e+00 UB    =1.268e-18 UC    =-8.52e-03
+U0    =4.422e+02
*   *** Subthreshold related parameters ***
+DSUB  =5.000e-01 ETA0  =6.765e-02 ETAB  =-2.15e-02
+NFACTOR=1.000e+00 VGHIGH=1.500e-01 VGLOW =-1.10e-01
*   *** Saturation related parameters ***
+ETA   =3.000e-01 PCLM  =1.562e+00
+PDIBL1=1.142e-01 PDIBL2=2.373e-03 DROUT =5.000e-01
+A0    =6.157e-01 A1    =0.000e+00 A2    =1.000e+00
+PVAG  =0.000e+00 VSAT  =8.186e+06
*   *** Geometry modulation related parameters ***
+LDD   =2.000e-07 W0    =0.000e+00
*   *** Temperature effect parameters ***
+AT    =3.300e+04 UTE   =-1.80e+00
+KT1   =-4.20e-01 KT2   =2.200e-02 KT1L  =0.000e+00
+UA1   =0.000e+00 UB1   =0.000e+00 UC1   =0.000e+00
*   *** Overlap capacitance related and dynamic model parameters ***
+CGDO  =3.400e-10 CGSO  =3.400e-10 CGBO  =1.400e-10
+XPART =1.000e+00
*   *** Parasitic resistance and capacitance related model parameters ***
+RDS0  =1.000e+00 RDSW  =1.809e+03
+CDSC  =8.221e-04 CDSCB =0.000e+00 CIT   =5.117e-04
*   *** Process and parameters extraction related model parameters ***
+TOX   =1.700e-08 NSUB  =5.000e+16
+XT    =1.200e-07 NLX   =1.000e-09
+XL    =0.000e+00 XW    =0.000e+00
*   *** Noise effect related model parameters ***
+AF    =1.451e+00 KF    =2.330e-26
+NLEV  =0
*   *** Common extrinsic model parameters ***
+ACM   =2
+RD    =0.000e+00 RS    =0.000e+00 RSH   =2.418e+01
```

```

+RDC =0.000e+00 RSC =0.000e+00 LD =2.000e-07
+WD =-6.00e-08
+LDIF =0.000e+00 HDIF =1.250e-06 WMLT =1.000e+00
+LMLT =1.000e+00 XJ =3.000e-07
+JS =2.000e-05 JSW =0.000e+00 IS =0.000e+00
+N =1.000e+00 NDS =1.000e+12 VNDS =0.000e+00
+CBD =0.000e+00 CBS =0.000e+00 CJ =4.500e-04
+CJSW =6.000e-10 FC =0.000e+00
+MJ =4.800e-01 MJSW =3.200e-01 TT =0.000e+00
+PB =9.200e-01 PHP =9.200e-01

.MODEL pmos PMOS LEVEL=47
+SUBTHMOD=3.000e+00 SATMOD =2.000e+00 BULKMOD=1.000e+00
* *** Threshold voltage related model parameters ***
+
+K3 =9.689e+00 K3B =-4.56e-01
+NPEAK =2.818e+16 VTH0 =-8.20e-01
+VOFF =-6.03e-02 DVT0 =1.000e+00 DVT1 =8.613e-01
+DVT2 =-8.87e-02 KETA =-1.41e-02
+VBM =-5.00e+00 PSCBE1 =0.000e+00 PSCBE2 =0.000e+00
* *** Mobility related model parameters ***
+UA =9.666e-10 UB =8.931e-19 UC =-4.30e-02
+U0 =1.719e+02
* *** Subthreshold related parameters ***
+DSUB =5.000e-01 ETA0 =1.239e-01 ETAB =0.000e+00
+NFACTOR=1.000e+00 VGHIGH =1.500e-01 VGLOW =-1.10e-01
* *** Saturation related parameters ***
+ETA =3.000e-01 PCLM =3.565e+00
+PDIBL1 =2.130e-01 PDIBL2 =3.494e-03 DROUT =5.000e-01
+A0 =5.119e-01 A1 =0.000e+00 A2 =1.000e+00
+PVAG =0.000e+00 VSAT =5.994e+06
* *** Geometry modulation related parameters ***
+LDD =2.000e-07 W0 =0.000e+00
* *** Temperature effect parameters ***
+AT =3.300e+04 UTE =-1.50e+00
+KT1 =-5.40e-01 KT2 =2.200e-02 KT1L =0.000e+00
+UA1 =0.000e+00 UB1 =0.000e+00 UC1 =0.000e+00
* *** Overlap capacitance related and dynamic model parameters ***
+CGDO =3.400e-10 CGSO =3.400e-10 CGBO =1.400e-10
+XPART =1.000e+00
* *** Parasitic resistance and capacitance related model parameters ***
+RDS0 =1.000e+00 RDSW =4.626e+03
+CDSC =9.038e-04 CDSCB =3.610e-05 CIT =1.733e-03
* *** Process and parameters extraction related model parameters ***
+TOX =1.700e-08 NSUB =4.037e+16
+XT =1.200e-07 NLX =1.698e-07
+XL =0.000e+00 XW =0.000e+00
* *** Noise effect related model parameters ***
+AF =1.279e+00 KF =6.314e-29

```

```

+NLEV =0
*   *** Common extrinsic model parameters ***
+ACM =2
+RD  =0.000e+00 RS  =0.000e+00 RSH  =5.000e+01
+RDC  =0.000e+00 RSC  =0.000e+00 LD   =1.600e-07
+WD   =6.000e-08
+LDIF =0.000e+00 HDIF =1.250e-06 WMLT  =1.000e+00
+LMLT =1.000e+00 XJ   =3.000e-07
+JS   =2.000e-05 JSW  =0.000e+00 IS   =0.000e+00
+N    =1.000e+00 NDS  =1.000e+12 VNDS  =0.000e+00
+CBD  =0.000e+00 CBS  =0.000e+00 CJ   =6.000e-04
+CJSW =5.400e-10 FC   =0.000e+00
+MJ   =4.600e-01 MJSW =3.100e-01 TT   =0.000e+00
+PB   =8.600e-01 PHP  =8.600e-01

```

```

.subckt opamp 3 8

```

```

mp1 1 4 4 1 pmos l=1.2u w=12u
mp2 1 4 7 1 pmos l=1.2u w=12u
mp3 1 7 8 1 pmos l=1.2u w=36u
mn1 4 2 5 5 nmos l=1.2u w=36u
mn2 7 3 5 5 nmos l=1.2u w=36u
mn3 8 6 9 9 nmos l=36u w=36u
mn4 5 6 9 9 nmos l=12u w=12u
ro 3 0 1e+4
ca 3 0 1e-6
vdd 1 0 dc 2.5
vss 0 9 dc 2.5
vi- 2 0 dc 0.25
vbi 6 0 dc 5.0

```

```

.ends opamp

```

```

*

```

```

* interconecoos para o primeiro neuronio

```

```

M1p11 51 1 201 201 nmos w=10e-6 l=5e-6
M1pw11 70 101 51 51 nmos w=10e-6 l=5e-6
M1p12 52 2 201 201 nmos w=10e-6 l=5e-6
M1pw12 70 102 52 52 nmos w=10e-6 l=5e-6
M1p13 53 3 201 201 nmos w=10e-6 l=5e-6
M1pw13 70 103 53 53 nmos w=10e-6 l=5e-6
M1p14 54 4 201 201 nmos w=10e-6 l=5e-6
M1pw14 70 104 54 54 nmos w=10e-6 l=5e-6
M1p15 55 5 201 201 nmos w=10e-6 l=5e-6
M1pw15 70 105 55 55 nmos w=10e-6 l=5e-6
M1p16 56 6 201 201 nmos w=10e-6 l=5e-6
M1pw16 70 106 56 56 nmos w=10e-6 l=5e-6
M1p17 57 7 201 201 nmos w=10e-6 l=5e-6
M1pw17 70 107 57 57 nmos w=10e-6 l=5e-6
M1p18 58 8 201 201 nmos w=10e-6 l=5e-6
M1pw18 70 108 58 58 nmos w=10e-6 l=5e-6
M1p19 59 9 201 201 nmos w=10e-6 l=5e-6

```

M1pw19 70 109 59 59 nmos w=10e-6 l=5e-6
 M1p110 60 10 201 201 nmos w=10e-6 l=5e-6
 M1pw110 70 110 60 60 nmos w=10e-6 l=5e-6
 M1p111 61 11 201 201 nmos w=10e-6 l=5e-6
 M1pw111 70 111 61 61 nmos w=10e-6 l=5e-6
 M1p112 62 12 201 201 nmos w=10e-6 l=5e-6
 M1pw112 70 112 62 62 nmos w=10e-6 l=5e-6
 M1p113 63 13 201 201 nmos w=10e-6 l=5e-6
 M1pw113 70 113 63 63 nmos w=10e-6 l=5e-6
 M1p114 64 14 201 201 nmos w=10e-6 l=5e-6
 M1pw114 70 114 64 64 nmos w=10e-6 l=5e-6
 M1p115 65 15 201 201 nmos w=10e-6 l=5e-6
 M1pw115 70 115 65 65 nmos w=10e-6 l=5e-6
 M1p116 66 16 201 201 nmos w=10e-6 l=5e-6
 M1pw116 70 116 66 66 nmos w=10e-6 l=5e-6
 M1n11 201 1 31 31 nmos w=10e-6 l=5e-6
 M1nw11 31 141 0 0 nmos w=10e-6 l=5e-6
 M1n12 201 2 32 32 nmos w=10e-6 l=5e-6
 M1nw12 32 142 0 0 nmos w=10e-6 l=5e-6
 M1n13 201 3 33 33 nmos w=10e-6 l=5e-6
 M1nw13 33 143 0 0 nmos w=10e-6 l=5e-6
 M1n14 201 4 34 34 nmos w=10e-6 l=5e-6
 M1nw14 34 144 0 0 nmos w=10e-6 l=5e-6
 M1n15 201 5 35 35 nmos w=10e-6 l=5e-6
 M1nw15 35 145 0 0 nmos w=10e-6 l=5e-6
 M1n16 201 6 36 36 nmos w=10e-6 l=5e-6
 M1nw16 36 146 0 0 nmos w=10e-6 l=5e-6
 M1n17 201 7 37 37 nmos w=10e-6 l=5e-6
 M1nw17 37 147 0 0 nmos w=10e-6 l=5e-6
 M1n18 201 8 38 38 nmos w=10e-6 l=5e-6
 M1nw18 38 148 0 0 nmos w=10e-6 l=5e-6
 M1n19 201 9 39 39 nmos w=10e-6 l=5e-6
 M1nw19 39 149 0 0 nmos w=10e-6 l=5e-6
 M1n110 201 10 40 40 nmos w=10e-6 l=5e-6
 M1nw110 40 150 0 0 nmos w=10e-6 l=5e-6
 M1n111 201 11 41 41 nmos w=10e-6 l=5e-6
 M1nw111 41 151 0 0 nmos w=10e-6 l=5e-6
 M1n112 201 12 42 42 nmos w=10e-6 l=5e-6
 M1nw112 42 152 0 0 nmos w=10e-6 l=5e-6
 M1n113 201 13 43 43 nmos w=10e-6 l=5e-6
 M1nw113 43 153 0 0 nmos w=10e-6 l=5e-6
 M1n114 201 14 44 44 nmos w=10e-6 l=5e-6
 M1nw114 44 154 0 0 nmos w=10e-6 l=5e-6
 M1n115 201 15 45 45 nmos w=10e-6 l=5e-6
 M1nw115 45 155 0 0 nmos w=10e-6 l=5e-6
 M1n116 201 16 46 46 nmos w=10e-6 l=5e-6
 M1nw116 46 156 0 0 nmos w=10e-6 l=5e-6
 M1pf1 70 211 201 201 nmos w=10e-6 l=5e-6
 M1nf2 201 212 0 0 nmos w=10e-6 l=5e-6

* interconecoes para o segundo neuronio

M2p21 71 1 202 202 nmos w=10e-6 l=5e-6
M2pw21 70 141 71 71 nmos w=10e-6 l=5e-6
M2p22 72 2 202 202 nmos w=10e-6 l=5e-6
M2pw22 70 142 72 72 nmos w=10e-6 l=5e-6
M2p23 73 3 202 202 nmos w=10e-6 l=5e-6
M2pw23 70 143 73 73 nmos w=10e-6 l=5e-6
M2p24 74 4 202 202 nmos w=10e-6 l=5e-6
M2pw24 70 144 74 74 nmos w=10e-6 l=5e-6
M2p25 75 5 202 202 nmos w=10e-6 l=5e-6
M2pw25 70 145 75 75 nmos w=10e-6 l=5e-6
M2p26 76 6 202 202 nmos w=10e-6 l=5e-6
M2pw26 70 146 76 76 nmos w=10e-6 l=5e-6
M2p27 77 7 202 202 nmos w=10e-6 l=5e-6
M2pw27 70 147 77 77 nmos w=10e-6 l=5e-6
M2p28 78 8 202 202 nmos w=10e-6 l=5e-6
M2pw28 70 148 78 78 nmos w=10e-6 l=5e-6
M2p29 79 9 202 202 nmos w=10e-6 l=5e-6
M2pw29 70 149 79 79 nmos w=10e-6 l=5e-6
M2p210 80 10 202 202 nmos w=10e-6 l=5e-6
M2pw210 70 150 80 80 nmos w=10e-6 l=5e-6
M2p211 81 11 202 202 nmos w=10e-6 l=5e-6
M2pw211 70 151 81 81 nmos w=10e-6 l=5e-6
M2p212 82 12 202 202 nmos w=10e-6 l=5e-6
M2pw212 70 152 82 82 nmos w=10e-6 l=5e-6
M2p213 83 13 202 202 nmos w=10e-6 l=5e-6
M2pw213 70 153 83 83 nmos w=10e-6 l=5e-6
M2p214 84 14 202 202 nmos w=10e-6 l=5e-6
M2pw214 70 154 84 84 nmos w=10e-6 l=5e-6
M2p215 85 15 202 202 nmos w=10e-6 l=5e-6
M2pw215 70 155 85 85 nmos w=10e-6 l=5e-6
M2p216 86 16 202 202 nmos w=10e-6 l=5e-6
M2pw216 70 156 86 86 nmos w=10e-6 l=5e-6
M2n21 202 1 121 121 nmos w=10e-6 l=5e-6
M2nw21 121 101 0 0 nmos w=10e-6 l=5e-6
M2n22 202 2 122 122 nmos w=10e-6 l=5e-6
M2nw22 122 102 0 0 nmos w=10e-6 l=5e-6
M2n23 202 3 123 123 nmos w=10e-6 l=5e-6
M2nw23 123 103 0 0 nmos w=10e-6 l=5e-6
M2n24 202 4 124 124 nmos w=10e-6 l=5e-6
M2nw24 124 104 0 0 nmos w=10e-6 l=5e-6
M2n25 202 5 125 125 nmos w=10e-6 l=5e-6
M2nw25 125 105 0 0 nmos w=10e-6 l=5e-6
M2n26 202 6 126 126 nmos w=10e-6 l=5e-6
M2nw26 126 106 0 0 nmos w=10e-6 l=5e-6
M2n27 202 7 127 127 nmos w=10e-6 l=5e-6
M2nw27 127 107 0 0 nmos w=10e-6 l=5e-6
M2n28 202 8 128 128 nmos w=10e-6 l=5e-6
M2nw28 128 108 0 0 nmos w=10e-6 l=5e-6

M2n29 202 9 129 129 nmos w=10e-6 l=5e-6
M2nw29 129 109 0 0 nmos w=10e-6 l=5e-6
M2n210 202 10 130 130 nmos w=10e-6 l=5e-6
M2nw210 130 110 0 0 nmos w=10e-6 l=5e-6
M2n211 202 11 131 131 nmos w=10e-6 l=5e-6
M2nw211 131 111 0 0 nmos w=10e-6 l=5e-6
M2n212 202 12 132 132 nmos w=10e-6 l=5e-6
M2nw212 132 112 0 0 nmos w=10e-6 l=5e-6
M2n213 202 13 133 133 nmos w=10e-6 l=5e-6
M2nw213 133 113 0 0 nmos w=10e-6 l=5e-6
M2n214 202 14 134 134 nmos w=10e-6 l=5e-6
M2nw214 134 114 0 0 nmos w=10e-6 l=5e-6
M2n215 202 15 135 135 nmos w=10e-6 l=5e-6
M2nw215 135 115 0 0 nmos w=10e-6 l=5e-6
M2n216 202 16 136 136 nmos w=10e-6 l=5e-6
M2nw216 136 116 0 0 nmos w=10e-6 l=5e-6
M2pf1 70 212 202 202 nmos w=10e-6 l=5e-6
M2nf2 202 211 0 0 nmos w=10e-6 l=5e-6

xop1 201 211 opamp
xop2 202 212 opamp
Vb1 70 0 dc 2.5
Vw11 101 0 dc 2.6
Vw12 102 0 dc 1.0
Vw13 103 0 dc 2.6
Vw14 104 0 dc 1.0
Vw15 105 0 dc 1.0
Vw16 106 0 dc 1.0
Vw17 107 0 dc 1.0
Vw18 108 0 dc 1.0
Vw19 109 0 dc 0.8
Vw110 110 0 dc 0.8
Vw111 111 0 dc 0.8
Vw112 112 0 dc 0.8
Vw113 113 0 dc 1.0
Vw114 114 0 dc 1.0
Vw115 115 0 dc 1.0
Vw116 116 0 dc 1.0
Vw21 141 0 dc 1.0
Vw22 142 0 dc 1.0
Vw23 143 0 dc 1.0
Vw24 144 0 dc 1.0
Vw25 145 0 dc 1.0
Vw26 146 0 dc 1.0
Vw27 147 0 dc 1.0
Vw28 148 0 dc 1.0
Vw29 149 0 dc 2.6
Vw210 150 0 dc 1.0
Vw211 151 0 dc 2.6

```
Vw212 152 0 dc 1.0
Vw213 153 0 dc 1.0
Vw214 154 0 dc 1.0
Vw215 155 0 dc 1.0
Vw216 156 0 dc 1.0
* padroes de entrada
Vi1 1 0 pulse (0 0 0N 0N 0N 25e-3 50e-3)
Vi2 2 0 pulse (0 0 0N 0N 0N 25e-3 50e-3)
Vi3 3 0 pulse (0 0 0N 0N 0N 25e-3 50e-3)
Vi4 4 0 pulse (0 0 0N 0N 0N 25e-3 50e-3)
Vi5 5 0 pulse (0 0 0N 0N 0N 50e-3)
Vi6 6 0 pulse (0 0 0N 0N 0N 50e-3)
Vi7 7 0 pulse (0 0 0N 0N 0N 50e-3)
Vi8 8 0 pulse (0 0 0N 0N 0N 50e-3)
Vi9 9 0 pulse (0 3 0N 0N 0N 25e-3 50e-3)
Vi10 10 0 pulse (0 3 0N 0N 0N 25e-3 50e-3)
Vi11 11 0 pulse (0 3 0N 0N 0N 25e-3 50e-3)
Vi12 12 0 pulse (0 3 0N 0N 0N 25e-3 50e-3)
Vi13 13 0 pulse (0 0 0N 0N 0N 50e-3)
Vi14 14 0 pulse (0 0 0N 0N 0N 50e-3)
Vi15 15 0 pulse (0 0 0N 0N 0N 50e-3)
Vi16 16 0 pulse (0 0 0N 0N 0N 50e-3)
.tran 0.1e-3 100e-3
.options post nopage
.END
```

Bibliografia

- [ALL87]ALLEN, Phillip E.; HOLBERG, Douglas R. **CMOS Analog Circuit Design**. [S.l]: Holt,Rinehart and Winston,Inc, 1987.
- [CAR89]CARBONELL, J.G. Introduction: Paradigms for Machine Learning. **Artificial Intelligence**,Amsterdam,v.40,p.1-9,Apr.1989.
- [CAV88]CARVALHO, L.A.V. **Redes Neurais e a Tradição Conexionista da Inteligência Artificial**. Rio de Janeiro : COPPE/ UFRJ, 1988. (Relatório Interno).
- [COT85]COTTREL, G.W. **A Connectionist Approach to Word Sense Disambiguation**. Rochester : University of Rochester, 1985. Tese de Doutorado.
- [FRE93]FREEMAN, James A.; SKAPURA, David M. **Redes Neurolas algoritmos, aplicaciones y técnicas de programación**. [S.l]: Addison Wesley, 1993.
- [GAL88]GALLANT, S.I. Connectionist Expert Systems. **Communications of the ACM**,New York,v.31,n.2,p.152-168,Feb.1988.
- [GRO88]GROSSBERG, Stephen. Nonlinear Neural Networks: Principles, Mechanisms, and Architectures. **Neural Networks**, USA, v.1, p.17-61, 1988.
- [HAS97]HASAN, S.M. Rezaul; SIONG, Ng Kang. A Parallel Processing VLSI BAM Engine. **IEEE Trans. on Neural Network**, New York, v.8, n.2, p.424-436, Mar. 1997.
- [HEB49]HEBB, Donald O. **The Organization of Behavior**. New York: John Wiley & Sons, 1949.
- [HEC89]HECHT-NIELSEN, R. Applications of Counterpropagation Networks. **Neural Networks**, Elmsford, New York, v.1, n.1, p.131-140, May 1989.
- [HOD52]HODGKIN,A.L.; HUXLEY, A.F. Current carried by sodium and potassium ions through the membrane of the giant axon of Loligo. **Journal of Physiology**, [S.l], p.116-449, 1952.
- [HOL94]HOLLIS, Paul W.; PAULOS, John J. A Neural Network Learning Algorithm Tailored for VLSI Implementation. **IEEE Trans. on Neural Network**, New York, v.5, n.5, p.784-791, Sept. 1994.
- [HOP84]HOPFIELD,J.J. Neurons with graded response have collective computational properties like those of two-state neurons. In: NATL. ACAD. SCIENCE, 1984, USA. **Proceedings ... USA**: [s.n],1984. p. 3088-3092.

- [HOP85]HOPFIELD,J.J; TANK, D.W. "Neural" Computation of Decisions in Optimization Problems. **Biological Cybernetics**, [S.I], n.52, p.141-152, 1985.
- [HSU96]HSU, Charles C.; GOBOVIC, Desa; ZAGHLOUL, Mona E.; et al. Chaotic Neuron Models and their VLSI Circuit Implementations. **IEEE Trans. on Neural Network**, New York, v.7, n.6, p.1339-1350, Nov. 1996.
- [JIM96]JIM, Kam-Chuen; GILES, Lee; HONE, Bill G. An Analysis of Noise in Recurrent Neural Networks: Convergence and Generalization. **IEEE Trans. on Neural Network**, New York, v.7, n.6, p.1424-1438, Nov. 1996.
- [KAN95]KANE, Jonathan S.; KINCAID, Thomas G. Optoelectronic Winner-Take-All VLSI Shunting Neural Network. **IEEE Trans. on Neural Network**, New York, v.6, n.5, p.1275-1279, Sept. 1995.
- [KOS92]KOSKO, Bart. **Neural Networks for Signal Processing**. [S.I]: Prentice Hall,1992 .
- [LIP87]LIPPMAN, R.P. An Introduction to Computing with Neural Nets. **IEEE Acoustics, Speech and Signal Processing Magazine**, New York, v.3, n.4, p.4-22, Apr.1987.
- [MAS97]MASRY, Ezz I.; YANG, Hong-Kui; YAKOUT, Mohamed A. Implementations of Artificial Neural Networks using Current- Mode Pulse width Modulation Technique. **IEEE Trans. on Neural Network**, New York, v.8, n.3, p.532-548, May 1997.
- [MEA89]MEAD, Carver. **Analog VLSI and Neural Systems**. [S.I]: Addison Wesley, 1989.
- [MON97]MONTALVO, Antonio J.; GYURCSIK, Ronald S.; PAULOS John J. Toward a General-Purpose Analog VLSI Neural Network with on-chip Learning. **IEEE Trans. on Neural Network**, New York, v.8, n.2, p.413-423, Mar. 1997.
- [REE89]REED, R.D.; GEIGER, R.L. A multiple-input OTA circuit for neural networks. **IEEE Trans. Circuits and Syst**, New York, v.36, n.5, p.767-769, May 1989.
- [MOL96]MOLZ, Rolf F. **Estudo de Técnicas de Implementação de Redes Neurais Competitivas utilizando tecnologias de Circuito VLSI Analógicos: Trabalho Inividual**. Porto Alegre: CPGCC da UFRGS,1996. (TI-582).
- [RUM85]RUMELHART, D.E.; ZIPSER, D. **Feature Discovery by Competitive Learning**. La Jolla: University of California at San Diego, Institute of Cognitive Science,1985. 151p.

- [RUM86]RUMELHART, D.E.; HINTON, G.E.; WILLIAMS, R.J. **Learning Internal Representations by Error Propagation**. La Jolla: University of California at San Diego, Institute of Cognitive Science,1986. (ICS Report, 8506).
- [SIM90]SIMPSON, P. **Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations**. New York : Pergamon Press Inc., 1990.
- [SZE81]SZE, S.M. **Physics of Semiconductor Devices**. New York: John Wiley & Sons, 1981.
- [VER89]VERLEYSSEN, Michel; JESPERS, Paul G.A. An Analog VLSI Implementation of Hopfield's Neural Network. **IEEE Micro**, New York, p.46-55, Dec. 1989.
- [WAN95]WANG, Jun. Analysis and Design of an Analog Sorting Network. **IEEE Trans. on Neural Network**, New York, v.6, n.4, p.962-971, July 1995.
- [ZAD65]ZADEH, L.A. Fuzzy Sets. **Information and Control**, [S.l.], v.8,p.338-353,1965.
- [ZUR92]ZURADA, Jacek M. **Introduction to Artificial Neural Systems**. [S.l]: West Publishing Company, 1992.
- [ZUR92a]ZURADA, Jacek M. Analog Implementation of Neural Networks. **IEEE Circuits & Devices**, [S.l], p.36-41,Sept. 1992.

Informática



UFRGS

CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

*"Proposta de Implementação em Hardware Dedicado de Redes
Neurais Competitivas com Técnicas de Circuitos Integrados Analógicos"*

por

Rolf Fredi Molz

Dissertação apresentada aos Senhores:

Prof. Dr. Dante Augusto Couto Barone

Prof. Dr. Philippe Olivier Alexandre Navaux

Prof. Dr. Fernando Gehm Moraes (PUCRS)

Vista e permitida a impressão.

Porto Alegre, / / .

Prof. Dr. Paulo Martins Engel,
Orientador.