

Unioeste - Universidade Estadual do Oeste do Paraná
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
Colegiado de Ciência da Computação
Curso de Bacharelado em Ciência da Computação

**Estudo Comparativo entre redes neurais artificiais e redes neurais pulsadas
usando MATLAB[®]**

Diego Henrique Pagani

CASCADEL
2012

DIEGO HENRIQUE PAGANI

**Estudo Comparativo entre redes neurais artificiais e redes neurais pulsadas
usando MATLAB[®]**

Monografia apresentada como requisito parcial
para obtenção do grau de Bacharel em Ciência da
Computação, do Centro de Ciências Exatas e Tec-
nológicas da Universidade Estadual do Oeste do
Paraná - Campus de Cascavel

Orientador: Prof. Josué Pereira de Castro

CASCADEL
2012

DIEGO HENRIQUE PAGANI

**Estudo Comparativo entre redes neurais artificiais e redes neurais pulsadas
usando MATLAB[®]**

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em
Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel,
aprovada pela Comissão formada pelos professores:

Prof. M. Eng. Josué Pereira de Castro
(Orientador)
Colegiado de Ciência da Computação,
UNIOESTE

Prof. M. Eng. Adriana Postal
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Msc. Suzan Kelly Borges Piovesan
Colegiado de Engenharia com Ênfase em
Controle e Automação, FAG
Colegiado de Sistema de Informação, UNIPAR

Cascavel, 29 de novembro de 2012

DEDICATÓRIA

Aos meus pais e a meu irmão por ser meu fiel companheiro.

“A curiosidade é sempre o primeiro passo em um problema que vai ser resolvido”. Galileu Galilei.

AGRADECIMENTOS

Primeiramente, ao criador de todas as coisas por permitir que exista a vida, inteligência e por conceder-me o anseio do entendimento.

Aos meus pais, por servirem de exemplo e por me apoiarem durante toda minha vida.

Ao meu irmão que sempre foi meu companheiro fiel.

À minha namorada por ter suportado e dado apoio nos momentos difíceis.

À toda minha família, por sempre darem seu apoio.

Aos meus antigos e novos amigos por me propiciarem momentos de alegria e descontração.

Aos meus professores, por terem dado todo suporte teórico e prático necessário para minha formação.

Ao meu orientador professor Josué, à professora Adriana pela grande contribuição e dedicação que tiveram na realização deste e de outros trabalhos. À professora Suzan pelas sugestões para que este trabalho se torne melhor e se conclua com qualidade.

Ao professor Clodis, aos colegas do grupo PETComp, pela amizade, companheirismo e trabalho em grupo.

À todos que, de alguma maneira, foram responsáveis pela realização deste trabalho.

Lista de Figuras

2.1	Modelo de neurônio biológico.	6
2.2	Períodos Refratários Absoluto e Relativo.	7
3.1	Modelo de neurônio <i>perceptron</i>	11
3.2	Exemplos de funções de ativação.	12
3.3	Modelo de uma RNA multicamadas do tipo <i>feedforward</i>	14
3.4	Modelo de uma RNA do tipo <i>recorrente</i>	15
3.5	Uma rede <i>Neural Backpropagation</i> com uma camada oculta.	17
4.1	Codificação em fase	22
4.2	Codificação através de Sincronismo	22
4.3	Modelo esquemático do circuito do modelo de integração e disparo.	25
4.4	Gráfico do neurônio do tipo SRM, durante o disparo.	26
4.5	a) uma rede <i>FeedForward</i> de neurônios pulsados. b) Uma ligação dividida em sub-conexões contendo unidades de atraso diferentes.	28
5.1	Codificação de um valor contínuo (130) em pulsos temporais em 8 neurônios de entrada. O intervalo de entrada é de [0:255], coberto por $g = 8$ campos receptivos gaussianos.	38

Lista de Tabelas

5.1	Dimensões das amostras	34
6.1	Resultados obtidos. A coluna de Iterações do treinamento é calculada como o total de iterações necessária para realizar o treinamento. Cada amostra da base de treinamento é considerada 1 iteração.	40

Lista de Abreviaturas e Siglas

RNA	Rede Neural Artificial
RNP	Rede Neural Pulsada
MLP	<i>Multi-layer Perceptron</i>
IA	Inteligência Artificial
ERA	Estado Refratário Absoluto
ERR	Estado Refratário Relativo
RBF	<i>Radial Base Function</i> , função de base radial, em inglês
SRM	<i>Simple Response Model</i> , Modelo de resposta simples, em inglês

Lista de Símbolos

n	Total de neurônios da camada de entrada
w	Peso sináptico
X_i	Representa a entrada do neurônio
Z_p	Saída do neurônio p da camada intermediária
Y_k	Saída do neurônio sem aplicação da função de ativação
y_k	Saída final do neurônio
p	Total de neurônios da camada de intermediária
m	Total de neurônio da camada de saída
δ	Taxa de correção de erro de w
Δ_t	Intervalo de tempo
n_p	Neurônio pulsado
t	Tempo
u	Potencial da membrana
I	Índice da camada de entrada
J	Índice da camada oculta
K	Índice da camada de saída
\bar{t}_k	Tempo de Saída desejado
t_k	Tempo de saída do neurônio
w_{jk}^d	Peso da subconexão d entre os neurônios j e k
η	Taxa de aprendizado
τ	Constante de decaimento do potencial pós-sináptico
x_{min}	Valor mínimo da entrada x
x_{max}	Valor máximo da entrada x
c_k	Centro da gaussiana
h	Tempo de disparo
g	Total de gaussianas

Sumário

Lista de Figuras	vii
Lista de Tabelas	viii
Lista de Abreviaturas e Siglas	ix
Lista de Símbolos	x
Sumário	xi
Resumo	xiv
1 Introdução	1
1.1 Objetivos	3
1.2 Justificativas	3
1.3 Organização	4
2 Redes Neurais Biológicas	5
2.1 Neurônio Biológico	5
2.2 Sinapses	7
2.3 Codificação da informação	7
3 Redes Neurais Artificiais	9
3.1 Histórico	9
3.2 Neurônio Artificial Perceptron	10
3.3 Função de ativação	11

3.4	Aprendizado	12
3.5	Arquitetura	13
3.5.1	Redes do tipo <i>FeedFoward</i>	13
3.5.2	Redes Recorrentes	14
3.6	Algoritmos de aprendizado	15
3.6.1	Algoritmo <i>Backpropagation</i>	16
4	Redes Neurais Pulsadas	18
4.1	Hipóteses de Codificação de Informação	19
4.1.1	Taxa de Pulsos	19
4.1.2	Codificação através de pulsos	20
4.2	Modelos de neurônios pulsados	22
4.2.1	Condutância e Compartimentos	24
4.2.2	Limiar de disparo	24
4.3	Arquiteturas Pulsadas	27
4.4	Algoritmos de Treinamento	28
5	Materiais e Métodos	33
5.1	Problemas encontrados	33
5.2	Metodologia dos testes	33
5.3	Bases de dados Utilizadas	34
5.3.1	Iris	34
5.3.2	<i>Breast Cancer Wisconsin</i>	35
5.4	Codificação das informações de entrada	37
6	Resultados obtidos	39
7	Considerações Finais	41
7.1	Sugestão para trabalhos futuros	42

Resumo

O objetivo deste trabalho é realizar um estudo comparativo entre dois modelos de redes neurais artificiais: o modelo *Multilayer Perceptron* (MP) e o modelo chamado de Spiking ou Pulsado. Como o modelo pulsado é relativamente recente comparado às Redes Neurais Artificiais, suas características ainda não são totalmente conhecidas, mesmo considerando modelos mais simples de rede, como o Perceptron. Este estudo tem por objetivo comparar os dois modelos de Perceptron e estabelecer os principais pontos positivos e negativos de seu funcionamento, utilizando para isto tarefas de classificação. Estabelecemos que as Redes Neurais Pulsadas, com neurônios SRM e *SpikeProp* como algoritmo de treinamento, não apresentam resultado significativo no quesito qualidade de solução, perante as redes neurais artificiais tradicionais.

Palavras-chave: Redes Neurais artificiais, Redes neurais pulsadas, *Backpropagation*, Campos Receptivos Gaussianos, *SpikeProp*.

Capítulo 1

Introdução

O cérebro é o principal órgão do sistema nervoso em várias espécies de animais. Este órgão nos seres humanos contém aproximadamente 100 bilhões de neurônios, ligados por mais de sessenta trilhões de sinapses. Estes neurônios interligados formam o que se chama de “Rede Neural” [1].

Existem modelos teóricos que visam simular esta estrutura neuronal de forma que seja possível emular o processo de raciocínio. Atualmente dois modelos em destaque são as conhecidas Redes Neurais Artificiais (RNA), bem difundidas na literatura e as Redes Neurais Pulsadas (RNP) que é uma abordagem mais plausível biologicamente que as RNAs.

A RNA é uma arquitetura responsável por codificar informações para que se obtenha a solução de algum problema, seja ele de classificação, monitoramento ou predição. Elas podem ser desenvolvidas a partir de modelos matemáticos assumindo que [2]:

- A informação é processada por uma estrutura simples chamadas de neurônios;
- Sinais são passados entre neurônios por meio de conexões;
- Cada conexão está associada a um peso, que é responsável pela codificação da informação;
- Cada neurônio avalia a entrada e determina se deve fazer seu disparo, por meio de sua função de ativação.

Um dos modelos mais conhecidos e simples é o *perceptron* que é constituído de um único neurônio e uma única camada que recebe uma ou várias entradas e retorna uma saída. A característica principal deste modelo é reconhecer padrões ditos como *linearmente separáveis*, isto é, padrões que se caracterizam por apresentar diferenças possíveis de se separar em duas regiões distintas. Com avanços nas pesquisas nesta área da Inteligência Artificial (IA), passou-se de um único neurônio para um modelo de múltiplos neurônios com múltiplas camadas conectadas por ligações ponderadas. Este modelo é chamado de *Multilayer Perceptron* (MLP) e contém ao menos três camadas [3]:

- *Input* (Entrada): responsável pelo envio dos sinais para a camada oculta;
- *Hidden* (Oculta): processa os sinais oriundos da Entrada, realizando a ponderação dos sinais;
- *Output* (Saída): recebe os sinais provenientes da camada oculta, realiza a ponderação dos sinais, e a saída é o resultado final da operação.

Com o avanço da neurociência, descobriu-se que com este modelo de cálculo seria impossível para o cérebro realizar em tempo hábil o processamento e então os pesquisadores elaboraram outra teoria: o conjunto dos neurônios funciona com a taxa de envio de sinais dependente do tempo e, alguns modelos, da sua ordem e não com ligações com custos e disparo sincronizado. A partir desta ideia, desenvolveu-se um novo modelo de RNA, as chamadas “Redes Neurais Pulsadas” (*Spiking Neural Network*).

A realização de um estudo comparativo entre estas duas abordagens de Rede Neuronal (RNA e RNP) é importante porque poderemos conhecer se o modelo Pulsado é ao menos equivalente ao modelo tradicional. O problema a ser resolvido é realizar a classificação de uma base de dados. Testaremos estas redes, analisando o comportamento por meio de testes de desempenho e qualidade de solução. Implementando os algoritmos em uma mesma linguagem e utilizando um mesmo ambiente de desenvolvimento, MATLAB[®], poderemos avaliar ambos os modelos quanto a tarefa de classificação.

1.1 Objetivos

O objetivo geral deste trabalho é realizar um estudo comparativo entre as arquiteturas RNAs e as RNPs na tarefa de classificação de base de dados, utilizando a qualidade de solução e tempo de resposta como fatores principais.

Para alcançar este objetivo geral será necessária a realização dos seguintes objetivos específicos:

- Implementar a “Rede Neural Artificial” *Multilayer perceptron*;
- Avaliar o desempenho da tarefa de classificação de base de dados;
- Implementar a “Rede Neural Pulsada” de múltiplas camadas;
- Avaliar o desempenho da “Rede Neural Pulsada” da tarefa de classificação de base de dados;
- Realizar testes que permitam a análise do comportamento das redes;
- Comparar a qualidade de solução e tempo de resposta.

1.2 Justificativas

Realizar um estudo comparativo entre estes dois modelos de redes neurais é interessante porque conhecendo as características específicas destes modelos, poderemos saber qual a melhor arquitetura para o problema que tentamos resolver.

Analisando e entendendo os limites de cada modelo, os pontos positivos e negativos e quais classes de problemas apresentam características que possam determinar qual das redes obtém o melhor desempenho para a tarefa de classificação, podemos utilizar novos métodos mais eficientes para resolver estas classes de problemas.

1.3 Organização

O trabalho está organizado da seguinte maneira: o capítulo 2 disserta sobre as redes neurais biológicas e sobre hipóteses sobre o seu funcionamento; no capítulo 3, falaremos sobre as Redes Neurais Artificiais tradicionais, explicando sua ideia e funcionamento; o capítulo 4 explicará como funciona uma rede neural pulsada e suas diferenças teóricas perante a Rede Neural Artificial Tradicional; no capítulo 5 será discutida a metodologia, as bases de dados utilizadas e detalhes sobre a implementação; no capítulo 6 serão discutidos os resultados obtidos com testes de reconhecimento de base de dados envolvendo as duas abordagens neurais e no capítulo 7 abordaremos as considerações finais e os trabalhos futuros.

Capítulo 2

Redes Neurais Biológicas

O cérebro humano é o principal órgão do sistema nervoso central. Ele é responsável por realizar o processamento dos sinais enviados pelos órgãos dos sentidos e pelo processo decisório que leva a realização de ações que serão efetuadas pelos músculos. É responsável também pelo raciocínio abstrato que leva a aquisição de conhecimento e melhora a compreensão do mundo. Estima-se que ele é formado por cerca de cem bilhões de neurônios, ligados por mais de sessenta trilhões de sinapses, que são os estímulos entre os neurônios [1].

Sendo as Redes Neurais Artificiais e as Redes Neurais Pulsadas baseadas no modelo biológico, é importante conhecer como funciona o sistema biológico para compreender melhor os sistemas artificiais.

2.1 Neurônio Biológico

O neurônio é a unidade básica de processamento neural. Os neurônios podem ser encontrados nas mais variadas formas e tamanhos, o que faz com que no cérebro humano encontremos vários modelos de neurônios. Apesar de serem diferentes entre si, os neurônios funcionam de maneira semelhante, recebendo estímulos e avaliando se devem, ou não, responder a este estímulo. Eles são divididos basicamente em três partes, conforme mostra a figura 2.1 adaptada de Tatibana e Kaetsu [4].

Dendritos: recebem a conexão sináptica.

Soma: corpo celular onde se faz o processamento do sinal e decide se deve produzir ou não um sinal de saída.

Axônio: responsável por propagar a saída de sinal para outras células nervosas.

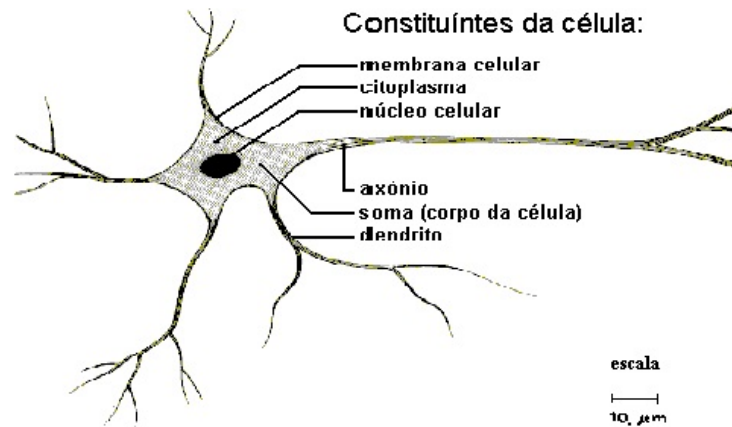


Figura 2.1: Modelo de neurônio biológico.

É necessário que a *membrana celular* receba uma quantidade de estímulos específica para que o *potencial da membrana* atinja um nível mínimo e realize um disparo. Com os estímulos pré-sinápticos oriundos de outros neurônios, este potencial é alterado e se não é recebido, tende a ser constante. Caso este potencial seja superior ao chamado *limiar de disparo*, o neurônio realiza o disparo e o sinal é propagado. Entretanto, não é somente necessário que estes estímulos ocorram, mas estes devem ocorrer em tempos próximos e com intensidade suficiente para provocar o disparo. Pode acontecer de um único estímulo ser responsável por alterar o potencial de forma significativa e faça o neurônio disparar, assim como é possível que mesmo com múltiplos estímulos, em tempos próximos, o neurônio não dispare [5].

A propagação de um sinal por um neurônio tem uma frequência limitada. Após ocorrer um disparo, o neurônio entra em um estado em que, mesmo recebendo estímulos, ele não realiza disparos. Este estado é chamado de Estado Refratário Absoluto (ERA). Após o neurônio sair deste estado, ele entra em outro estado chamado de Estado Refratário Relativo (ERR) em que o seu limiar é mais alto, tornando necessário mais estímulos para que um disparo ocorra [6]. A figura 2.2, por Kolb e Whishaw [7] adaptada por Jaskowiak [5], mostra os estados refratários.

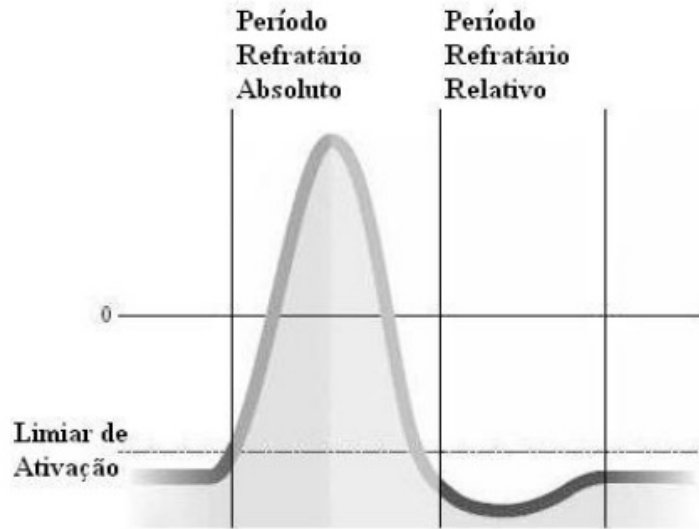


Figura 2.2: Períodos Refratários Absoluto e Relativo.

2.2 Sinapses

As sinapses são as unidades básicas que realizam as interações entre os neurônios. Entre os diversos tipos de sinapses, a mais comum é a *sinapse química*, que libera uma substância transmissora que se espalha pelo espaço entre os neurônios, chamado junção sináptica. Assim um sinal elétrico pré-sináptico é transformado em um sinal químico e então retorna um sinal elétrico pós-sináptico [1].

Uma característica das sinapses é que estas podem mudar a força do estímulo e este efeito é chamado de plasticidade sináptica e permite à rede aprender com experiências passadas [5].

2.3 Codificação da informação

Sendo a força do estímulo a unidade básica de transmissão de informação entre os neurônios e sabendo-se que este estímulo não transmite informação em seu formato, o cérebro certamente possui outras maneiras para codificar e decodificar as informações em seu interior. Mesmo com os avanços nas pesquisas, ainda não chegou-se a um consenso de como o cérebro faz esta codificação/decodificação biologicamente [8].

Existem duas hipóteses que tentam explicar como a informação é codificada e decodificada

nos pulsos trocados entre os neurônios, que são a codificação através de pulsos (*Pulse Coding*) e a hipótese da taxa de pulsos (*Rate Coding*) que serão explicadas no capítulo 4.

Capítulo 3

Redes Neurais Artificiais

Segundo Fausset [3], uma Rede Neural Artificial é um sistema de processamento de informação que apresenta certas características de redes neurais biológicas. Os modelos de RNAs foram desenvolvidos a partir de generalizações de modelos matemáticos sobre a biologia neural, baseado em alguns princípios:

- O processamento de sinal ocorre em elementos simples chamados de neurônios;
- Sinais são transmitidos entre neurônios por meio de conexões;
- Cada conexão tem um peso que multiplica o sinal transmitido;
- Cada neurônio aplica uma função de ativação, que tem como objetivo verificar se o neurônio deve disparar ou não, como será visto na seção 3.3.

Existem modelos ou paradigmas diferentes que visam separar em categorias as RNAs. Duas abordagens que contribuem para caracterizar sua habilidade funcional, além de diferenciá-las, é referente ao *aprendizado* e à *arquitetura*.

3.1 Histórico

As RNAs surgiram em 1943, quando Warren McCulloch e Walter Pitts propuseram um primeiro modelo matemático para uma rede neural (Neurônio McCulloch-Pitts) [1]. Estes pesquisadores reconheceram que na combinação de simples neurônios em sistemas neurais estava a fonte do aumento de poder computacional [3].

O próximo desenvolvimento significativo das redes neurais apareceu em 1949, com a publicação do livro *The Organization of Behavior*, escrita por Donald O. Hebb, que descreveu um modelo de aprendizado por correlação dos neurônios, que acabou dando origem a *Regra de Aprendizagem de Hebb* e essa teoria é comumente lembrada para alguns tipos de aprendizagem associativas, onde o peso da ligação entre dois neurônios é aumentado, no caso de ativação simultânea [1].

Na década de 1950 e 1960, chamada de “*The first Golden Age*” (A primeira era de Ouro) por Fausset [3], ocorreu quando vários pesquisadores desenvolveram uma grande classe de redes neurais artificiais chamadas de *perceptrons*. Explicaremos mais sobre estes neurônios na seção 3.2. Nesta época foi desenvolvida também a chamada *Regra Delta*, um método para ajuste dos pesos do *perceptron*. A Regra Delta faz o ajuste dos pesos para reduzir a diferença entre o resultado da rede com o valor que deveria sair.

A década de 1970 não teve grandes avanços na área das redes neurais, apenas 3 trabalhos principais [3]:

- Teuvo Kohonen, memórias associativas e auto-organizadas utilizando *clusters*;
- James Anderson, iniciou pesquisa em redes neurais com redes de memória associativa;
- Gail Carpenter, junto com Stephen Grossberg desenvolveram a *teoria de ressonância adaptativa*.

Na década de 1980, chamada de “ressurgimento das redes”, o problema das redes *perceptron* de uma única camada reconhecerem apenas problemas linearmente separáveis teve uma solução, com o surgimento do algoritmo *backpropagation*, que é um algoritmo de treinamento para redes de múltiplas camadas. Além disso, surgiram as chamadas *Redes de Hopfield* que trabalham com pesos fixos e com ativações adaptativas, usadas para criar redes de memória associativa [3].

3.2 Neurônio Artificial Perceptron

Segundo Haykin [1], “Um neurônio é uma unidade de processamento de informações que é fundamental para a operação de uma rede neural”.

Cada neurônio pode receber diversas entradas e cada entrada dessas passa por uma conexão que possui um peso. Um sinal x_n na entrada n do neurônio k é multiplicado pelo peso sináptico w_{kn} . Após o cálculo de todas as conexões, todos os sinais são somados e este valor é passado para função de ativação. Na seção 3.3 abordaremos com mais detalhes as funções de ativação. A figura 3.1 mostra o modelo de um neurônio do tipo *perceptron*, com suas entradas x_n , pesos w_{kn} e a saída y_k .

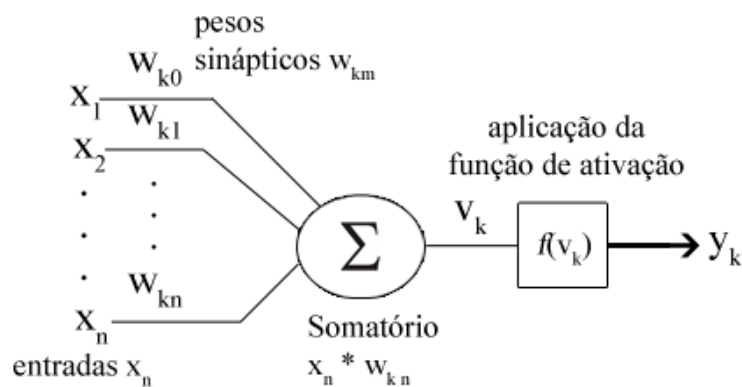


Figura 3.1: Modelo de neurônio *perceptron*

3.3 Função de ativação

A operação básica de um neurônio envolve a soma dos sinais ponderados e aplicação da função da ativação. Normalmente, usa-se a mesma função de ativação para os neurônios de uma mesma camada, mas isto não é regra. Na maioria dos casos, utiliza-se uma função que não seja a identidade ($f(x) = x$). Quando usa-se redes neurais de múltiplas camadas, é obrigatório que a função de ativação não seja linear [3].

Existem vários tipos de funções que podem ser usadas, como exemplo as da figura 3.2, adaptada por Postal [9]:

1. Identidade
2. Binária
3. Sigmoidal

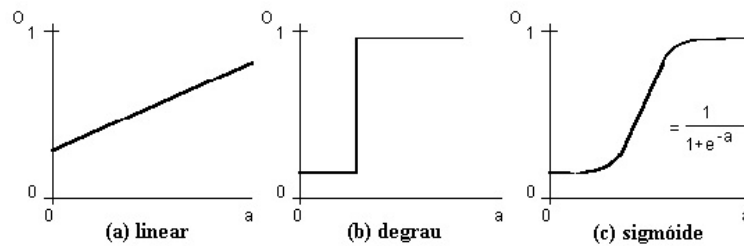


Figura 3.2: Exemplos de funções de ativação.

3.4 Aprendizado

Segundo Haykin [1], a propriedade primordial de uma RNA é a sua habilidade de aprender a partir do seu ambiente e de melhorar o seu desempenho através da *aprendizagem*. Este processo consiste na adequação dos parâmetros da rede, que são adaptados através de um algoritmo específico. O tipo de aprendizado é determinado pela maneira em que modificação dos parâmetros ocorre. Para que o treinamento seja finalizado, é necessário que a rede atinja uma solução generalizada para uma classe de problemas. Em uma RNA em fase de treinamento, ocorrem três eventos:

1. Estimulação pelo ambiente;
2. Ocorrem modificações nos seus parâmetros;
3. Mudança na resposta as ações do ambiente.

Existem dois tipos de paradigmas de aprendizado de uma rede, que são [1]:

Supervisionado: A rede aprende através de exemplos propostos por um supervisor. Com isso, um conjunto de exemplos de um domínio é apresentado à rede com seu valor de saída correto.

Não supervisionado: Este modo é usado quando não existem informações claras sobre os possíveis mapeamentos com base nos dados disponíveis. Então não existe um supervisor ensinando-a, espera-se que ela possa agrupar as entradas por alguma similaridade.

3.5 Arquitetura

A arquitetura de uma RNA representa a forma como os neurônios estão organizados e conectados entre si. Esta forma de organização está relacionada com os algoritmos de aprendizagem usados para treinar a rede [3].

Em geral, podem ser identificadas duas classes de redes neurais com duas ou mais camadas, chamadas de *feedforward* e as redes *recorrentes*.

A escolha de uma arquitetura de uma Rede Neural depende de vários fatores, principalmente:

- Complexidade do problema;
- Dimensão das entradas;
- Conhecimento *a priori* sobre o problema;
- Representatividade dos dados;
- Tipo de algoritmo de treinamento escolhido.

3.5.1 Redes do tipo *FeedFoward*

Normalmente, as redes neurais são tipicamente organizadas em camadas, que podem estar conectadas às unidades da camada seguinte. Basicamente, são definidas três tipos de camadas:

Entrada: este tipo de camada é único na rede, onde os padrões são apresentados e não é feito nenhum tipo de cálculo.

Escondidas ou Intermediárias: este tipo de camada é onde realiza-se a maior parte do processamento, através de conexões ponderadas com os neurônios da camada anterior. Dependendo do domínio do problema, é necessário que se tenha mais de uma camada deste tipo, para que possa extrair corretamente as características do problema.

Saída: este tipo é único na rede, em que o resultado final é apresentado.

A figura 3.3 representa um modelo esquemático de uma rede neural do tipo *feedforward* com as três camadas principais. Estas redes são caracterizadas por sempre serem propagadas à “frente”, em que o fluxo de dados sempre ocorre no mesmo sentido.

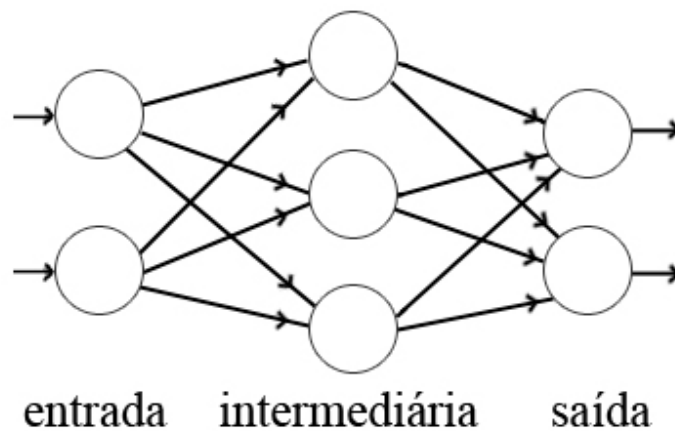


Figura 3.3: Modelo de uma RNA multicamadas do tipo *feedforward*

Cada círculo representa um neurônio e cada seta representa a ligação ponderada de cada neurônio. Os neurônios à esquerda representam a entrada. Os neurônios centrais representam a camada escondida e os à direita, a camada de saída. As setas à esquerda indicam as entradas chegando nos neurônios de entrada e as que saem dos neurônios de saída, são a saída da rede. As setas entre os neurônios indicam as ligações sinápticas ponderadas e mostram que a rede está interconectada.

3.5.2 Redes Recorrentes

As redes recorrentes são aquelas em que a estrutura dos neurônios, além da alimentação para “frente”, possui conexões para realimentação (*feedback*), portanto, não existe direção privilegiada para propagação da informação. A figura 3.4 mostra um exemplo de uma rede recorrente com três neurônios.

Esta arquitetura é a mais utilizada para memória associativa. Pode-se ter mais de uma camada, com realimentação a partir de qualquer uma destas, tornando-o a organização da rede as mais variadas possíveis. O treinamento deste modelo de rede ocorre desta maneira: é dado a

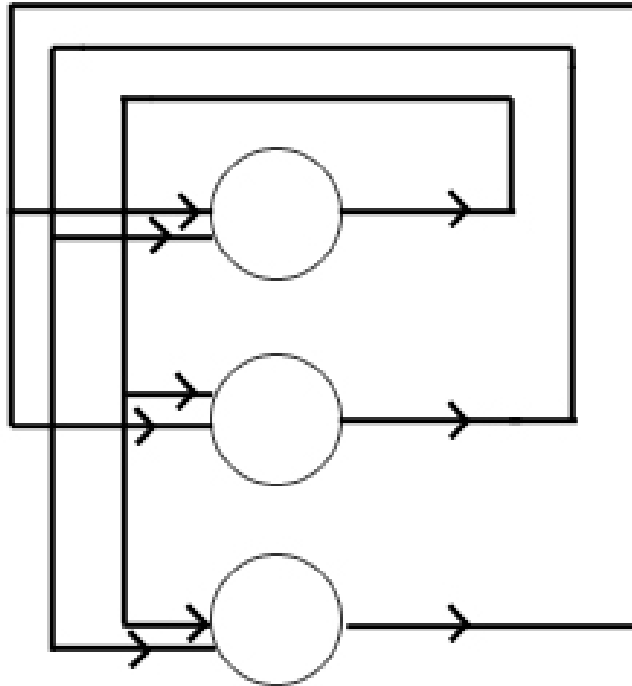


Figura 3.4: Modelo de uma RNA do tipo *recorrente*

rede os dados iniciais, é calculada a saída da mesma e esta saída é usada como entrada na próxima iteração. Este tipo de treinamento só é finalizado quando a saída durante algumas iterações seja constante.

3.6 Algoritmos de aprendizado

Define-se algoritmo de aprendizado como um conjunto de regras que define a solução de um problema de aprendizado [1]. Existem muitos tipos de algoritmos específicos para determinados modelos de RNA, e estes se diferem principalmente como os pesos são ajustados.

Existem vários tipos de algoritmos de aprendizado e cada um apresenta características específicas para cada modelo de RNA, como por exemplo [1]:

- Correção de erro;
- Baseada em memória;
- Hebbiana;

- Competitiva;
- Boltzmann;
- Por reforço.

Neste trabalho abordaremos um algoritmo por Correção de Erro, que é o *backpropagation*.

3.6.1 Algoritmo *Backpropagation*

A demonstração das limitações de redes neurais de uma única camada foi um fator na queda do interesse por redes neurais nos anos 70. A descoberta por vários cientistas independentes e a ampla divulgação de um método eficaz geral de treinamento de uma rede neural multicamadas desempenhou um papel importante no ressurgimento de redes neurais (1980) como uma ferramenta para resolver uma grande variedade de problemas [3].

O treinamento de uma rede usando o *backpropagation* envolve 3 estágios:

1. A propagação à frente das amostras de treinamento;
2. O cálculo e a retropropagação do erro associado;
3. Ajuste dos pesos.

Após o treinamento, o uso da rede envolve apenas a propagação à frente, que é a operação das entradas com os pesos, que é, nos computadores atuais, executado de maneira bem rápida. Como o treinamento não tem um tempo fixo de execução, pode demandar bastante paciência, entretanto, existem várias maneiras de melhorar o desempenho do processo de treinamento [3]. Um exemplo, é ao invés de usar o *backpropagation* tradicional que usa o *gradiente descendente*, utilizar o *Levenberg-Marquardt backpropagation* que apresenta resultados mais rápidos (este método é o padrão definido na *toolbox* de redes neurais do MATLAB[®]).

Como uma rede de uma única camada é limitada nos problemas que pode aprender, uma rede multicamadas (com uma ou mais camadas ocultas) pode aprender qualquer problema contínuo com uma alta precisão. Mais de uma camada oculta pode ser benéfica, para algumas aplicações, mas uma camada oculta é suficiente. A figura 3.5, adaptada de Fausset [3], mostra um esquema de uma rede neural artificial com uma camada oculta.

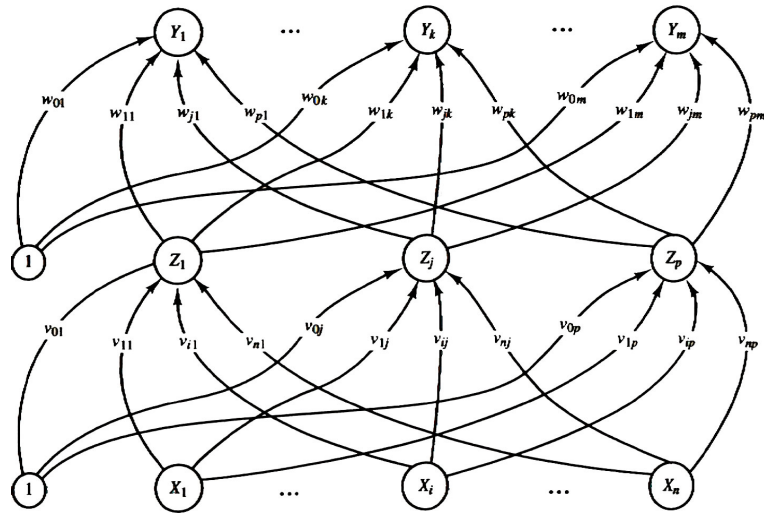


Figura 3.5: Uma rede *Neural Backpropagation* com uma camada oculta.

Algoritmo

Esta subseção é baseada em Fausset [3].

Durante à propagação a frente, cada unidade de entrada (x_i) recebe o sinal e distribui para todos os neurônios da camada oculta Z_p . Cada neurônio Z_p faz soma da ponderação de todas as entradas que recebeu e aplica a função de ativação $f(x)$. Com a saída da função, espalha-se esta saída como entrada a todos os neurônios da próxima camada, sendo ela a última ou não. Na última camada, após todos os cálculos idênticos aos outros neurônios, a saída da rede será a saída dos neurônios Y_k aplicadas a uma função de ativação, podendo esta ser a mesma dos outros.

Após isso é calculada, a partir da saída obtida e a saída correta, o erro referente a esta entrada. Com este erro, é calculado um fator δ_k do neurônio da camada de saída e será repassado para os neurônios da camada anterior, para que estes possam calcular qual a sua parcela de erro e o fator de correção do pesos.

Com os valores δ calculados, o peso da ligação w_{ij} é atualizado utilizando como uma parcela da correção este valor δ . Com isto sendo feito para todas as ligações entre os neurônios, os pesos são ajustados para aperfeiçoar a rede.

Capítulo 4

Redes Neurais Pulsadas

Chamada por alguns autores como a terceira evolução das Redes Neurais Artificiais, as Redes Neurais Pulsadas utilizam a hipótese de que a codificação de informação ocorre na forma de taxa de pulsos (*rate code*). Os experimentos e estudos de Gerstner [10] assumem que a codificação de informação no cérebro ocorre na forma de taxa de pulsos e conseguem justificar os resultados obtidos por meio de observações. Vários modelos matemáticos foram inspirados pelas hipóteses de taxa de pulsos, além de experimentos e estudos no campo da neurociência [5]. Todos estes dados fizeram com que esta hipótese de codificação tivesse sucesso nos últimos tempos [10].

Vários modelos RNAs tradicionais utilizam a hipótese de codificação de informação por meio da taxa de pulsos produzida pelos neurônios, entretanto não há parâmetros que incrementem ao modelo alguma informação temporal. A noção de atraso que existe nas redes de Hopfield não possui influência direta no processamento de informação deste modelo de rede, ou seja, não há influência do tempo na informação [5].

Estudos realizados revelam que a codificação baseada em taxas de pulsos não seriam rápidas o suficiente para proporcionar a realização de certos tipos de operações e um dos principais argumentos foi dado por Thorpe e Imbert [11], em 1989: Primatas podem realizar processamento e classificação de padrões visuais (reconhecimento de objetos e classificação) em tempos próximos a cem milissegundos. Este intervalo de tempo não seria o suficiente para que os neurônios pudessem realizar uma estimativa da taxa de disparos, pois muitos neurônios não conseguiriam gerar mais de um disparo. Para dizer isto, os autores usam a lógica de que os estímulos na retina passam por volta de dez camadas de neurônios, o que deixaria cerca de dez milissegundos de processamento para cada camada de neurônios envolvida no processamento [5].

Com base nestes argumentos, de Thorpe e Imbert, surgiu interesse por parte de pesquisadores em encontrar alguma forma de processamento, como RNAs, mas que fossem inspirados em neurônios biológicos de maneira mais realística. Com a definição de tais modelos e através de simulações, espera-se ter modelos capazes de realizar computações de maneira tão eficiente quanto as realizadas no cérebro [5].

4.1 Hipóteses de Codificação de Informação

Existem diversas hipóteses que tentam justificar a maneira com que a informação é transmitida internamente no cérebro. A seguir são abordados alguns conceitos que são usados tipicamente em experimentos na área da neurociência e da computação.

4.1.1 Taxa de Pulsos

Esta hipótese de codificação vem sofrendo grandes críticas por diversos autores, como Gerstner [10] e Thorpe e Imbert [12], pois seria muito simples e não permitiria que os neurônios respondessem a estímulos em tempos curtos suficientemente para certas computações. Existem três formas de como codificar pela taxa de pulsos, descritas com mais detalhes em Jaskowiak [5]:

Contagem de Pulsos: Esta é a hipótese mais usada em experimentos. A representação consiste na determinação de um intervalo Δt no qual os pulsos recebidos pelo neurônio n_p são contados, a partir de um instante t , até outro instante $t + \Delta t$ que delimita o intervalo da contagem.

Densidade de Pulsos: Esta forma de codificação consiste na repetição de um determinado padrão em um determinado intervalo de tempo. O resultado pode ser interpretado de maneira sucinta criando um histograma que represente a densidade de pulsos ao longo do tempo. É interessante ressaltar que esta codificação pode não ser usada biologicamente, pois é inviável a realização de várias repetições por parte de um neurônio para que possa decidir se dispara ou não.

Atividade populacional: Para utilização desta forma de codificação, os neurônios devem possuir propriedades idênticas, além de um mesmo padrão de conexões. Com isso, podemos dividir a população de neurônios em sub-populações, que trocam informações entre si. O cálculo se dá pelo total de disparos de uma sub-população em um determinado intervalo de tempo.

4.1.2 Codificação através de pulsos

Segundo Gerstner [10], a escolha de um modelo de codificação para representar a informação não deve se basear apenas na simples escolha entre um modelo de taxa de disparos, pois o modelo deve ir além disso, sendo importante que o esquema ou hipótese de codificação permita que os neurônios respondam de maneira rápida a estímulos, sendo assim biologicamente plausíveis. Por este motivo, os modelos baseados na hipótese de que a informação é codificada por qualquer meio de taxa de pulsos não é interessante, pois negligencia informações de pulsos produzidos pelos neurônios. As codificações através das taxas de pulsos permitem que o neurônio responda de maneira rápida a estímulos, pois levam em conta tempos precisos em que são gerados os pulsos. Nas próximas subseções serão abordadas as teorias, segundo este método.

Instante para Primeiro Pulso

Neste esquema, o primeiro pulso gerado contém grande parte da informação (se não toda) a ser transmitida de acordo com os estímulos recebidos. Após estímulos em um neurônio i , um pulso é gerado dentro de um intervalo de tempo Δt . Desta forma, quanto mais cedo um neurônio dispara, mais forte é o disparo [5]. Thorpe argumenta que certas atividades do cérebro não possuem tempo suficiente para analisar mais de um pulso em cada neurônio em cada etapa do processamento [13].

Mesmo considerando que cada neurônio gera apenas um pulso, existem algumas maneiras de se interpretar estes pulsos, além da latência de geração do pulso, por exemplo:

Ordem de chegada: Nesta hipótese o tempo deixa de ser importante para que a ordem de chegada do pulso seja relevante. Supondo um intervalo t_n com $t_0 = 0$ e $t_{10} = 10$, se

houver disparos ocorridos neste intervalo, existem $10!$ possíveis ordens diferentes, que demonstra uma grande quantidade de informação que pode ser transmitida [5].

Código binário: A informação neste método é representado como um código binário. Para n neurônios, existem 2^n variações possíveis, o que pode ser uma grande quantidade de informação.

Tempos Precisos dos Pulsos: Esta hipótese codifica e decodifica informações dependendo dos tempos de chegada dos pulsos. O número de variações depende apenas do número de neurônios e o grau de precisão com que estas diferenças são abordadas.

Estes três métodos são formas que podem ser utilizadas para trocar informações baseando-se apenas no primeiro pulso de um neurônio. Em qualquer um dos métodos, uma grande quantidade de informação pode ser trocada, mesmo usando poucos neurônios.

Codificação em Fase

Esta forma de codificação é semelhante a da seção 4.1.2, mas ao invés de utilizar apenas um pulso é usada como referência um sinal periódico, ou seja, um ciclo. A cada ciclo da entrada é feito o disparo dos neurônios. Se o ciclo de entrada não se altera durante o tempo, o padrão de saída em fases é produzido. A figura 4.1, adaptada por Jaskowiak [5] retirado de Gerstner e Kistler [14], mostra seu funcionamento.

Sincronismo e Correlação

A codificação pela forma de sincronismo estabelece que disparos em tempos próximos podem ser considerados sendo o mesmo, como se houvessem disparado ao mesmo tempo. A figura 4.2, de Gerstner e Kistler [14], mostra o diagrama de pulsos. As regiões em destaque representam os pulsos que serão considerados ao mesmo tempo.

A codificação por correlação leva em conta os tempos e a ordem de produção de sinais, ou seja, caso a mesma ordem de produção de sinais ocorra, mas com tempos diferentes, a codificação das entradas também seria diferente, conseqüentemente a saída do neurônio também.

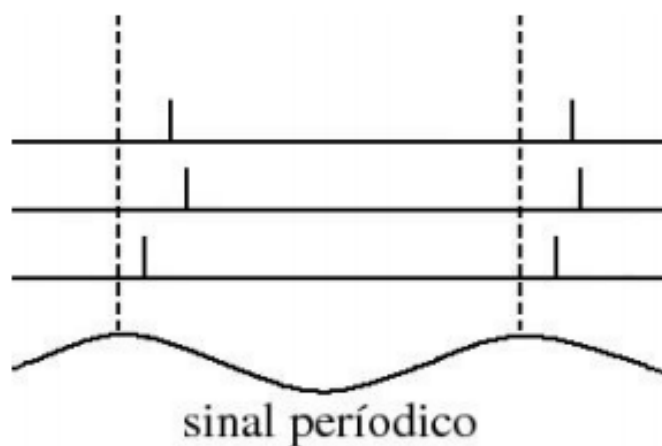


Figura 4.1: Codificação em fase

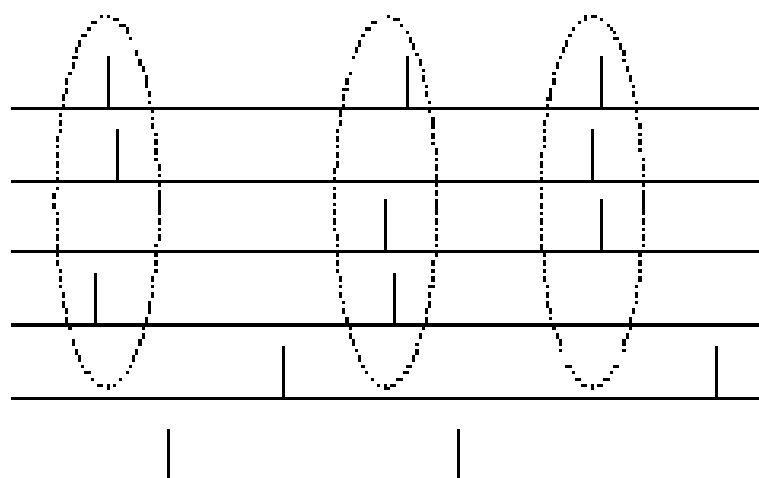


Figura 4.2: Codificação através de Sincronismo

4.2 Modelos de neurônios pulsados

Mass [15] separa as redes neurais em três gerações, segundo os modelos de neurônios:

Primeira geração: são os modelos que possuem as unidades funcionais baseadas no neurônio criado por McCulloch e Pitts, que tem como característica as entradas e saídas digitais, tornando o modelo universal para realização para computações deste tipo;

Segunda geração: a principal característica da segunda geração é a utilização de funções de ativação contínuas, do tipo sigmoidal ou de base radial. Com estes tipos de funções é

possível a computação de valores analógicos e digitais, se na saída da rede houver algum tipo de normalização. Assim como nos neurônios de primeira geração, não há qualquer informação temporal na codificação das informações pelo neurônio;

Terceira geração: os modelos classificados como pertencentes à terceira geração são aqueles que codificam as informações utilizando o tempo como um fator decisivo. Estes modelos são chamados de neurônios pulsados (*spiking* ou *pulsed*, em inglês).

É importante dizer que nas Redes Neurais Artificiais, ditas tradicionais, a decisão da escolha do modelo do neurônio implica apenas na escolha da função de ativação que melhor se adeque ao problema. Já nas Redes Neurais Pulsadas a escolha, talvez por enquanto, não é tão simples. Existem diversos modelos de neurônios pulsados na literatura e por enquanto não se encontram muitos estudos práticos comparando estes modelos para determinar qual obtêm um melhor resultado para algum tipo de tarefa.

Existem várias linhas de pesquisa envolvendo neurônios pulsados e existem diversos modelos destas unidades funcionais, que podem ser divididos em duas categorias:

- Condutância e Compartimentos;
- Limiar de disparo.

É interessante saber que os neurônios pulsados podem ser usados em implementações de unidades em redes neurais de base radial. Em uma abordagem convencional, estes neurônios funcionam como uma unidade que integra os sinais de entrada e dispara caso estes alcancem um limiar. Através de uma interpretação dos tempos de disparo, pode-se fazer com que este se assemelhe a um neurônio RBF (*Radial Base Function*). O tempo de disparo de um neurônio representa a distância do sinal de entrada recebido do centro de uma função de base radial. Para que isso seja possível, as conexões sinápticas devem possuir unidades de atraso associadas. Quando todas as unidades de atraso que chegam a um neurônio estão sintonizadas de forma que os sinais cheguem em tempos próximos, isto faz com que o neurônio dispare, sendo que quanto mais próximos os sinais chegam, mais próxima do centro da função RBF se encontra a entrada associada a estes sinais [5] [16].

Nas sub-seções seguintes serão abordados alguns modelos de neurônios que mais são citados na literatura.

4.2.1 Condutância e Compartimentos

Os tipos de neurônios do modelo de condutância descrevem o funcionamento dos neurônios de forma bem detalhada. Os neurônios deste tipo são usados para simulações de pequeno porte em que o tempo de execução não é um fator crítico [17]. Por serem bem detalhados, apresentam características fortemente ligadas aos neurônios biológicos mas possuem um gasto computacional muito alto. O modelo de Hodhkin-Huxley, por exemplo, precisa de 1200 operações de ponto flutuante para cada milissegundo de simulação [17]. Estes modelos não apresentam descrição da estrutura espacial do neurônio ou das sinapses envolvidas na modelagem, apenas são considerados como estruturas pontuais. O foco é a maneira em que ocorre o processo de geração do potencial de ação. Para mais detalhes, ver Gerstner [10].

Os modelos de compartimentos se baseiam na descrição do funcionamento do neurônio globalmente e suas interligações. São baseados nos modelos de condutância e acrescentam informações espaciais nas suas definições. Diferente dos modelos de condutância, este modelo leva em conta como as dimensões espaciais dos neurônios e modelam a transmissão sináptica de maneira mais detalhada [10].

4.2.2 Limiar de disparo

Os modelos de limiar de disparo (*threshold-fire models*) possuem uma maior plausibilidade biológica, mas não descrevem de maneira fiel o funcionamento das suas estruturas. Estes modelos utilizam várias abstrações em função de desempenho, mas mantém a característica dos neurônios pulsados, que é utilizar o tempo como fator de codificação. Sendo assim, são melhores para aplicações práticas, podendo ter redes de tamanhos maiores e melhor adequados para aplicações práticas [5]. Os parágrafos seguintes dissertam sobre alguns tipos de neurônios deste modelo.

Modelo de Integração e Disparo

O modelo de integração e disparo é um dos modelos mais simples do limiar de disparo [14]. Ele necessita de 5 operações de ponto flutuante para computar um milissegundo de simulação. Comparando este modelo ao de Hodhkin-Huxley, há uma drástica redução de operações, conseqüentemente há reduções também na lista de fenômenos biológicos observados [5]. A compreensão deste modelo é mais simples se interpretarmos a rede como um circuito elétrico com um capacitor de capacitância C e um resistor de resistência R ligados em paralelo, ligados com uma corrente I que varia de acordo com o tempo, como mostra a figura 4.3, retirada de Jaskowiak [5].

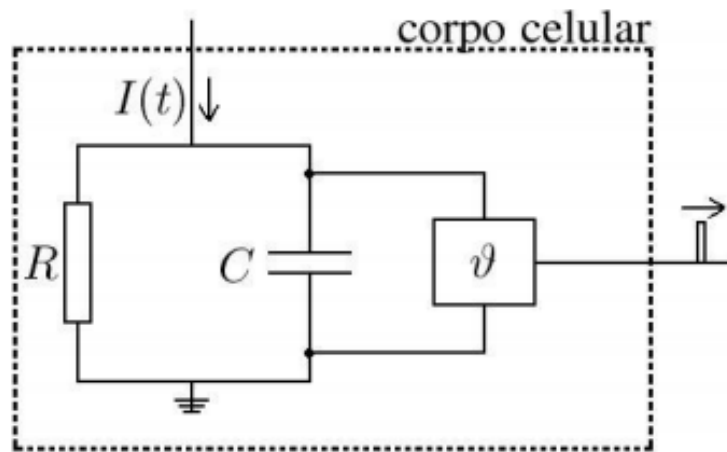


Figura 4.3: Modelo esquemático do circuito do modelo de integração e disparo.

Utilizando a Equação 4.1, demonstrada em Jaskowiak [5], podemos calcular o potencial da membrana em um determinado instante de tempo. Para que o neurônio possa disparar, este potencial deve atingir ao menos um limiar ϑ já previamente estabelecido.

$$\Gamma_m \frac{du}{dt} = -u(t) + RI(t) \quad (4.1)$$

É possível incorporar ao modelo o período refratário absoluto modelando uma interrupção do cálculo da Equação 4.1 por um intervalo Δt .

Vale ressaltar que neste modelo, a forma do potencial de ação, dos potenciais pós-sinápticos e os períodos refratários não estão inseridos, apenas o potencial da membrana é rigorosamente

definido.

Os valores para os parâmetros Γ_m e R da Equação 4.1 não são pré-estabelecidos e devem ser ajustados de maneira empírica, para obter o padrão de pulsos e comportamento desejados do respectivo neurônio.

Modelo de Resposta Impulsiva

Este modelo, *Simple Response Model* (SRM), criado por Gerstner [18], é constituído de uma abordagem diferente da utilizada pelo modelo de integração e disparo. O autor diz que os modelos de condutância e compartimentos levam em conta aspectos irrelevantes, quando se está interessado apenas em informações carregadas nos trens de pulso (sequência de pulsos) gerados pelos neurônios.

Comparando este modelo aos de condutância e compartimentos, temos um gasto computacional menor, pois apenas os trens de pulsos são gerados, semelhante aos neurônios biológicos. A figura 4.4, retirada de Jaskowiak [5], mostra um gráfico de disparo de um neurônio deste modelo, em que antes de disparar o neurônio tem uma variação de crescimento maior do que após o disparo, em que o decaimento da função é mais suave.

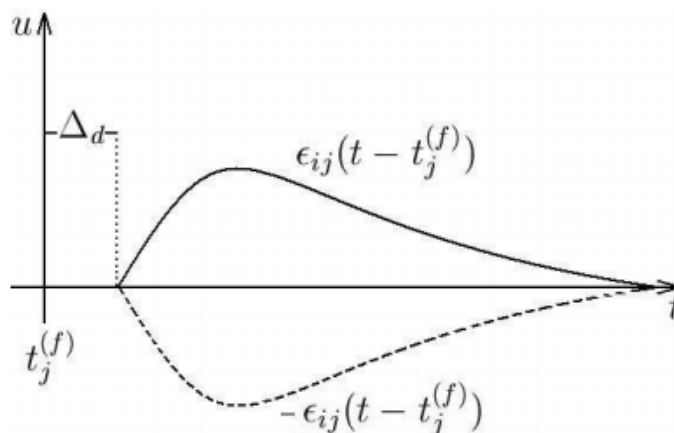


Figura 4.4: Gráfico do neurônio do tipo SRM, durante o disparo.

Este modelo é constituído de uma variável, u , que representa o potencial da membrana de um neurônio i em função do tempo, ou seja $u_i(t)$. Quando o potencial u_i se iguala ou ultrapassa o limiar ϑ a geração de um potencial de ação ocorre. Bohte [19] utiliza a fórmula 4.2 para

a modelagem dos potenciais pós-sinápticos do SRM, sendo τ uma constante que determina o decaimento do potencial da membrana, definido empiricamente.

$$\varepsilon_{ij}(t - t_j^{(f)}) = \frac{t}{\tau} \exp\left(1 - \frac{t}{\tau}\right) \quad (4.2)$$

O potencial da membrana $u_i(t)$, definido pela equação 4.3 é alterado em apenas dois eventos. O primeiro é o recebimento de um sinal proveniente de um neurônio pré-sináptico j e o segundo é o estabelecimento do potencial de equilíbrio após a ocorrência de um disparo e a passagem dos períodos refratários (absoluto e relativo) [5].

$$u_j(t) = \sum_{i \in \Gamma_j} \sum_{k=1}^m w_{ij}^d \varepsilon(t - t_- d^k) \quad (4.3)$$

4.3 Arquiteturas Pulsadas

Assim como as Redes Neurais Artificiais tradicionais, a distribuição dos neurônios de forma que cada unidade seja conectada a outro neurônio, formando uma rede neural, é necessária.

A arquitetura que se deve usar em redes com neurônios pulsados deve ser adequada ao problema que se deseja resolver. [6]. Como as RNAs são bem difundidas na literatura, uma variedade de problemas já possuem solução e a maneira de configuração da rede explicada. Tais conhecimentos sobre a solução destes problemas têm sido aplicados na escolha ou montagem da arquitetura das redes neurais pulsadas. Ou seja, a ideia de redes de única camada, múltiplas camadas e competitivas ainda se mantém possível nas redes pulsadas. Entretanto, não há como afirmar que o mesmo modelo de RNAs apresente um resultado semelhante nas RNPs.

Sobre as conexões sinápticas entre os neurônios, os pesos das ligações multiplicam a intensidade do sinal de entrada, alterando a sua intensidade ou valor, analogamente as RNAs. Como as RNPs utilizam tempo como codificação, um sinal em um determinado tempo t , que pode sofrer alguma alteração devido aos pesos, poderá causar influência no neurônio de destino quando este determinará se o neurônio deve disparar ou não, causando assim diferença na saída do sinal.

Em alguns tipos de arquitetura de RNPs existem as chamadas unidades de atraso. Estas unidades, que podem ser diferentes em cada sinapse (ou sub-conexão), causam a propagação dos

pulsos de forma atrasada por um determinado período pré-estabelecido. Por alterar o tempo de propagação, estas unidades tem ligação direta na codificação da informação, podendo influenciar no aprendizado.

Outro fator diferente das arquiteturas das RNAs, é o conceito de sub-conexões, sub-sinapses ou desdobramento sináptico, em que uma ligação entre um neurônio e outro contém mais de um canal, tendo cada sub-canal um peso e talvez unidades de atraso específicas para aquela sub-conexão [5]. A figura 4.5, retirada de Bohte [19], exemplifica as sub-conexões. Para exemplos de ferramentas que utilizam diferentes arquiteturas, ver Jaskowiak [5].

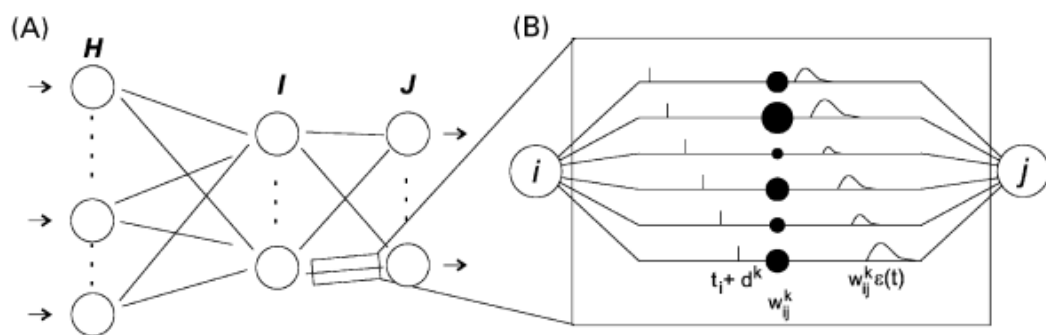


Figura 4.5: a) uma rede *FeedForward* de neurônios pulsados. b) Uma ligação dividida em sub-conexões contendo unidades de atraso diferentes.

4.4 Algoritmos de Treinamento

Assim como as RNAs, as RNP's precisam ser treinadas para que a rede possa aprender e realizar as tarefas específicas. Para as RNAs existem diversos algoritmos espalhados na literatura (ver Fausset [3] e Haykin [1]). Em se tratando das RNP's, faltam algoritmos de treinamento amplamente formalizados e aceitos. Diversos algoritmos para treinamento supervisionado, ou não-supervisionado foram propostos mas nenhum deles surge como uma resposta definitiva para uma determinada arquitetura, deixando assim como um problema ainda em aberto a determinação de métodos de treinamento eficientes [5].

Um fato que indica o não-estabelecimento dos algoritmos de treinamento é o que informações importantes sobre o seu funcionamento e parâmetros são omitidas por seus autores. Por

exemplo, o algoritmo *SpikeProp* [19] não tem convergência garantida e não apresenta informações sobre os valores que devem ser usados em seus pesos sinápticos [20]. Existem casos que não há um algoritmo de treinamento genérico, mas um algoritmo específico para aquele problema. Existem outros casos em que o treinamento ocorre dentro do modelo neurônio [5].

Existem alguns algoritmos para treinamento de RNPs, como *SpikeProp* (que veremos a seguir), de aprendizagem competitiva, baseado na regra de Hebb e para redes do tipo RBF [5].

SpikeProp

Este algoritmo [19] [21] é um método de treinamento supervisionado semelhante ao algoritmo *backpropagation* aplicado às RNAs e é considerado como o primeiro deste tipo para Redes Neurais Pulsadas [16]. Para que ele possa ser utilizado, a rede neural deve utilizar neurônios do tipo e Resposta Impulsiva (seção 4.2.2) e codificar a informação com apenas um pulso produzido por neurônio para cada sinal de entrada apresentado. Além disso, é eliminada a necessidade dos períodos refratários no modelo do neurônio. Ele é destinado para redes de múltiplas camadas, com um ou mais ocultas, tendo seus neurônios conectados através de sinapses múltiplas. A discussão a seguir baseia-se em Jaskowiak [5], Simões [16], Sporea [22] e Bohte [19].

A definição do algoritmo é feita a partir de uma rede neural de três camadas I , J e K . O objetivo do algoritmo é o de ensinar os neurônios da camada de saída K um conjunto de tempos de disparos para cada uma das entradas que é apresentada à rede, a partir da camada de entrada I .

Definindo \bar{t}_k como o valor do tempo de disparo de um neurônio da camada de saída e t_k como o valor de saída efetiva do neurônio, a equação 4.4 calcula o erro E apresentado na camada de saída da rede, para uma entrada.

$$E = \frac{1}{2} \sum_{k \in K} (t_k - \bar{t}_k)^2 \quad (4.4)$$

De maneira diferente do algoritmo *backpropagation* tradicional, existem duas regras de adaptação distintas: uma para os pesos da camada de saída e outra para os pesos da camada escondida da rede.

Como podem existir mais de uma conexão (ou sub-conexões) entre dois neurônios distintos, um da camada oculta (J) e outro da camada de saída (K), existirá um peso w_{jk}^d , com j e k representando os índices dos neurônios da camada oculta e de saída respectivamente e d indicando o índice da subconexão entre os neurônios.

Para o ajuste dos pesos sinápticos da camada de saída, os autores propõe a equação 4.5, onde:

Δw_{jk}^d é a variação do peso da ligação entre os neurônios j e k na conexão d ;

η é a taxa de aprendizado;

$\epsilon_{jk}^d(t_k)$ é a função do potencial pós-sináptico do neurônio k em um instante t_k , em resposta ao sinal transmitido a esta ligação na fase de propagação;

δ_k é definida pela Equação 4.6.

$$\Delta w_{jk}^d = -\eta \cdot \epsilon_{jk}^d(t_k) \cdot \delta_k \quad (4.5)$$

Na equação 4.6, \bar{t}_k e t_k são o tempo de disparo desejado e o tempo de disparo real que ocorreu no neurônio k . Os somatórios na equação fazem que seja levado em conta cada uma das subconexões de todos os neurônios ligados à k .

$$\delta_k = \frac{(\bar{t}_k - t_k)}{\sum_{j \in J} \sum_{d=1}^n w_{jk}^d \cdot \left(\frac{\partial \epsilon_{jk}^d(t)}{\partial t} \cdot (t_j) \right)} \quad (4.6)$$

Para as camadas ocultas, os pesos sinápticos são calculados conforme mostra a equação 4.7.

$$\Delta w_{ij}^d = -\eta \cdot \epsilon_{ij}^d(t_j) \cdot \delta_j \quad (4.7)$$

Pode ser utilizada na equação 4.8, as mesmas definições das equações 4.5 à 4.7.

$$\delta_j = \frac{\sum_k \{ \delta_k \cdot \sum_d w_{jk}^d \frac{\partial \epsilon_{jk}^d(t)}{\partial t_j} \}}{\sum_{i \in I} \sum_{d=1}^n w_{ij}^d \cdot \left(\frac{\partial \epsilon_{ij}^d(t_j)}{\partial t_j} \right)} \quad (4.8)$$

As equações acima definem as regras de adaptação dos pesos de uma RNP que utiliza neurônios do tipo SRM e o algoritmo *Spikeprop*.

Observando as equações acima, podemos notar que se o neurônio não realiza o disparo, sua contribuição é nula, pois seu tempo de disparo é indefinido e conseqüentemente a variação dos pesos também. Este fator pode causar problemas, pois caso vários neurônios não disparem para um padrão de entrada, não é possível realizar o ajuste dos pesos. Com isso, deve-se alterar os parâmetros do SRM para que uma maior quantidade de neurônios dispare, ou então modificar os pesos sinápticos de forma que causem o disparo. Os autores não fornecem informações de como se deve inicializar os pesos sinápticos ou distribuí-los.

Um fator importante a respeito do algoritmo é que sua convergência não é garantida e após a criação, adaptações foram sugeridas por Booij [6] e McKennoch [23] a fim de melhorar o desempenho. O algoritmo 1 mostra o pseudo-código do *SpikeProp*.

Algoritmo 1 Algoritmo *SpikeProp*, adaptado de Moore [24]

- o tempo de disparo de um neurônio é encontrado quando $u_j(t) \geq \vartheta$ usando a equação 4.3. Isso é repassado para os neurônios da camada oculta e logo em seguida para saída;
- calcular δ_k usando a equação 4.6, usando o tempo de disparo atual e o desejado nos neurônios de saída;
- calcular δ_j usando a equação 4.8, usando o tempo de disparo atual dos neurônios da camada oculta e os valores δ_k ;
- para a camada de saída, calcular w_{jk} usando a equação 4.5;
- para as camadas intermediárias, calcular w_{ij} usando a equação 4.7;
- adaptar os pesos utilizando $NovoW = AntigoW + w_{ij}$ para camada intermediária e $NovoW = AntigoW + w_{jk}$ para camada de saída.

Comparando os tempos de pesquisa das RNAs tradicionais com as das RNPs estas podem ser consideradas relativamente recentes e existem muitos fatores importantes que são descritos pelos criadores dos algoritmos, o que dificulta a sua determinação [5]. Simões [16] argumenta que o desenvolvimento do *SpikeProp* é recente e mesmo com alguns testes, maiores análises precisam ser realizadas para que informações concretas sobre suas implementações se tornem possíveis.

Existem diversas fontes na literatura que mostram, ou tentam, explicar o funcionamento do

SpikeProp. Algumas delas mostram resultados de uso deste algoritmo utilizando certas metodologias e bases de dados. Entretanto, não dizem claramente como foram feitas as implementações, a codificação/decodificação dos dados ou até mesmo o porquê da escolha de determinados valores.

Capítulo 5

Materiais e Métodos

5.1 Problemas encontrados

Devido há vários problemas não foi possível realizar a implementação da Rede Neural Pulsada. Entre os problemas encontrados, podemos citar:

- Omissão de parâmetros e detalhes da implementação da RNP;
- Falta de material definitivo ou poucos estudos definindo padrões;
- Poucos estudos em português;
- Vários trabalhos teóricos e poucos práticos.

Apesar deste inconveniente, Bothe[21] disse em sua tese que realizou a implementação e, além disso, realizou uma comparação entre as RNPs e RNAs, que veremos com detalhes nas próximas seções.

5.2 Metodologia dos testes

A metodologia de análise das RNAs usada para estabelecer a comparação foi:

1. 300 baterias de teste;
2. 10.000 épocas de treinamento;

3. taxa de aprendizado de 0.01;
4. uso de 30% da amostra, escolhida aleatoriamente, total para treinamento.

A escolha de 30% da amostra foi definida empiricamente e as amostras são selecionadas de acordo com o resultado da função *dividerand()* presente no MATLAB[®], que recebe como parâmetros o total de amostras e as porcentagens referentes a 3 categorias quaisquer. Definindo apenas uma categoria como 30%, o retorno é um vetor de índices de amostras.

5.3 Bases de dados Utilizadas

5.3.1 Iris

A base de dados Iris foi criada por R.A. Fisher em 1936 e em 1988 se tornou pública. Ela contém 150 amostras de dimensões das sépalas e pétalas da planta Iris, que as divide em três classes: *Iris-Setosa*, *Iris-Versicolour* e *Iris-Virginica* [25]. Os limites dos dados das amostras é dada pela tabela 5.1.

Tabela 5.1: Dimensões das amostras

		Mínimo	Máximo	Média	Desvio padrão
Sépala	largura	4.3	7.9	5.84	0.83
	altura	2.0	4.4	3.05	0.43
Pétala	largura	1.0	6.9	3.76	1.76
	altura	0.1	2.5	1.20	0.76

A configuração dos neurônios da RNA se deu da seguinte maneira para esta base:

Camada de Entrada: como a base Iris contém 4 valores de ponto flutuante, altura e largura da pétala e sépala, existem 4 neurônios na camada de entrada que repassam para os neurônios da camada de saída, acrescido um de bias, totalizando 5;

Camada oculta: como não há um modo exato de se calcular a quantidade ideal de neurônios na camada oculta, o valor de 11 neurônios foi determinado empiricamente;

Camada saída: como existem três categorias de flores, foram utilizadas saídas binárias em que apenas um neurônio é ativado para cada classe.

Bohte [19] usou a seguinte configuração nos seus testes envolvendo RNAs e RNPs para fazer a classificação desta base de dados.

Camada de Entrada: devido as redes pulsadas possuírem um conceito diferente das RNAs, o padrão de entrada é diferente também. Usando campos receptivos gaussianos (explicado na seção 5.4) para realizar a codificação da entrada, ele usa 50 neurônios nesta camada. Bothe [19] cita também que utilizou 12 neurônios para cada padrão de entrada, o que totalizaria, a partir das 4 entradas da Iris, 48 neurônios. Os dois neurônios extras não são citados se são usados como *bias*;

Camada oculta: 10 neurônios;

Camada saída: como existem três categorias de flores, utilizou saídas binárias, totalizando três neurônios.

Afirma também que o intervalo de codificação usado é de quatro milissegundos e a constante de tempo $\tau = 7ms$. A taxa de aprendizado foi definida como $\eta = 0.0075$.

5.3.2 *Breast Cancer Wisconsin*

Esta base de dados, criada por Dr. W. H. Wolberg, contém análises de amostras de mamas com câncer [26]. A amostra contém 699 amostras. Existem nove atributos que descrevem esta classe:

- *Clump Thickness*;
- *Uniformity of Cell Size*;
- *Uniformity of Cell Shape*;
- *Marginal Adhesion*;
- *Single Epithelial Cell Size*;
- *Bare Nuclei*;

- *Bland Chromatin*;
- *Normal Nucleoli*;
- *Mitoses*.

A configuração dos neurônios da RNA se deu da seguinte maneira para esta base:

Camada de Entrada: como esta base contém nove entradas, foram utilizados nove neurônios de entrada, acrescido 1 de bias, totalizando 10;

Camada oculta: como dito anteriormente, empiricamente foi determinado o uso de dezessete neurônios na camada oculta;

Camada saída: como a saída é benigno ou maligno, foi determinado dois neurônios utilizando saída binária.

Foi usada em Bohte [19] a seguinte configuração para as Redes Neurais, tanto Pulsada quanto Artificial, para realizar os testes com esta base de dados.

Camada de Entrada: pelo mesmo motivo abordado em respeito as entradas na base Iris, ele usa 64 neurônios nesta camada. Este afirma que utilizou 7 gaussianas para codificar a entrada, sendo 9 entradas da base, o que totaliza 63 entradas. Este neurônio faltante é o bias;

Camada oculta: 15 neurônios;

Camada saída: como existem duas categorias, benigno ou maligno, a saída contém 2 neurônios binários.

O intervalo de codificação usado é de quatro milissegundos, a constante de tempo $\tau = 7ms$ e a taxa de aprendizado foi definida como $\eta = 0.0075$.

5.4 Codificação das informações de entrada

Como as RNPs levam o tempo em consideração é necessário que a entrada também seja dada por tempos. Como os dados disponibilizados para as bases utilizadas nesta monografia são valores estáticos e não variam com o tempo, é necessário que estes dados sejam transformados em sinais de pulsos.

Bohte [21] introduz um método para codificar entradas digitais em sequências de pulsos e utiliza campos receptivos gaussianos para isto. Como um meio de codificação de uma população de dados, usa-se múltiplos campos locais receptivos para distribuir a variável de entrada em múltiplos neurônios de entrada. Este método origina um grande número de disparos, para cada entrada digital. A seguir segue o funcionamento do método, retirado de Bohte [21] e Simões [16].

Primeiramente precisaremos definir que:

1. n é a dimensão de uma componente do vetor de entrada;
2. g é o conjunto de gaussianas;
3. k é o índice que percorreremos as gaussianas, que será usada para percorrer o número de neurônios da camada de entrada;
4. c_k é o centro da gaussiana;
5. o intervalo dos dados, no exemplo mostrado na figura 5.1, retirada de Simões [16], de entrada é de $[0:255]$;
6. h é o tempo de disparo (valor de retorno da aplicação do dado de entrada na função gaussiana);
7. o intervalo de tempo, usado no exemplo, é de 10 iterações;
8. $x : [x_{min} \ x_{max}]$ é a entrada.

A camada de entrada da rede é composta com $n.g$ neurônios e as gaussianas determinam o atraso no disparo de cada um dos $n.g$ neurônios da camada de entrada. Se alguma gaussiana k

for estimulada no seu valor máximo local, o neurônio i_k gerará um pulso da primeira iteração. Se isto não ocorrer, o neurônio i_k disparará após h iterações. Ou seja, o k -ésimo neurônio da camada de entrada irá disparar de acordo com o estímulo da gaussiana k . A equação 5.1 demonstra como é realizado o cálculo do centro de cada uma das k gaussianas.

$$c_k = x_{min} + \frac{2k - 3}{2} \cdot \frac{x_{max} - x_{min}}{(n.g) - 2} \quad \forall n.g > 2 \quad (5.1)$$

A partir de descoberto o centro da gaussiana, ela é segmentada no total de iterações (atraso de disparo), como mostra a figura 5.1. O valor utilizado na figura, 130, é calculado em todas as gaussianas e a partir do resultado obtido, verifica-se em qual iteração o valor resultante se aproxima, este sendo definido o tempo no qual o neurônio (ou gaussiana) disparará.

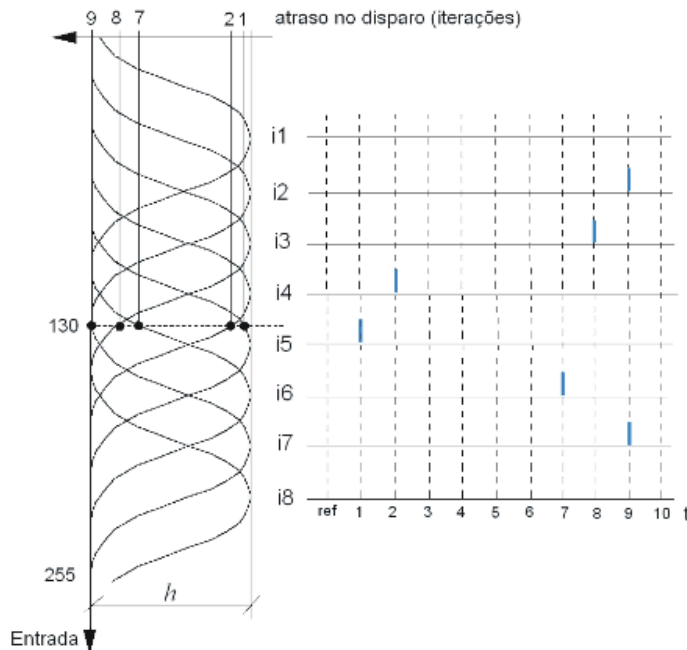


Figura 5.1: Codificação de um valor contínuo (130) em pulsos temporais em 8 neurônios de entrada. O intervalo de entrada é de $[0:255]$, coberto por $g = 8$ campos receptivos gaussianos.

É interessante dizer que dentre as pesquisas realizadas, este foi o único método encontrado de codificar sinais de entrada em pulsos. Em sua tese, Bohte [21] afirma que não tinha conhecimento sobre algum método de codificação de entradas contínuas/digitais em pulsos. Ele também não afirma se, em uma rede neural, é necessário fazer a decodificação deste método

Capítulo 6

Resultados obtidos

A tabela 6.1 mostra os resultados obtidos por Bohte [19] e os resultados seguindo a metodologia proposta. As implementações utilizando *MatlabBP* e *MatlabLM* utilizam a *toolbox* de redes neurais disponível no MATLAB[®]. *MatlabBP* usa a função de gradiente descendente tradicional e *MatlabLM* usa o treinamento *Levenberg-Marquardt* para o ajuste dos pesos. *SpikeProp* se refere a implementação de Bohte [19] utilizando neurônios SRM e *SpikeProp* como algoritmo de treinamento.

Analisando a tabela 6.1, podemos ver que sempre obteve-se uma taxa de acerto semelhante, com exceção do *MatlabLM*, com a base Iris, que teve uma taxa de 98% que pode ser considerada alta. Se compararmos o uso do *SpikeProp* com as RNAs, utilizando os campos receptivos gaussianos como método de codificação das entradas, podemos notar que o *SpikeProp* apresentou um treinamento bem mais rápido do que os outros e tem que se levar em conta, que os dados de entrada são preparados para as RNPs.

Se compararmos as RNAs com e sem codificação das entradas, apresentaram uma taxa de iterações semelhante (Bohte[21] utilizou 70% da base, enquanto nesta dissertação, 30%), mas os métodos sem codificação apresentam menor custo computacional devido a utilizar menos neurônios na camada de entrada.

Tabela 6.1: Resultados obtidos. A coluna de Iterações do treinamento é calculada como o total de iterações necessária para realizar o treinamento. Cada amostra da base de treinamento é considerada 1 iteração.

Algoritmos	Entrada	Intermediária	Saída	Iterações do treinamento	Taxa de acerto
Resultados obtidos por Bohte [19]					
<i>Iris dataset</i>					
SpikeProp	50	10	3	1000	96.1%
MatlabBP	50	10	3	2600000	95.5%
MatlabLM	50	10	3	3750	95.7%
<i>Wisconsin breast cancer dataset</i>					
SpikeProp	64	15	2	1500	97.0%
MatlabBP	64	15	2	9200000	96.3%
MatlabLM	64	15	2	3500	96.7%
Resultados obtidos a partir da metodologia usada					
<i>Iris dataset</i>					
RNA MP	5	11	3	450000	96.7%
MatlabBP	5	11	3	450000	94.7%
MatlabLM	5	11	3	540	98%
<i>Wisconsin breast cancer dataset</i>					
RNA MP	10	17	2	126315	97.4%
MatlabBP	10	17	2	2100000	94.9%
MatlabLM	10	17	2	2310	96.9%

Capítulo 7

Considerações Finais

Analisando a literatura sobre as Redes Neurais Pulsadas, pudemos notar que ainda são feitos muitos estudos teóricos sobre os modelos de neurônios e algoritmos que os descrevem. Por ser um abordagem relativamente recente, comparado as RNAs, as RNPs não tem uma definição sólida sobre suas unidades básicas ou sobre algoritmos de treinamento. O que existe são adaptações dos algoritmos das RNAs para que possam ser aplicados às RNPs.

Determinar qual das duas arquiteturas tem o melhor funcionamento na resolução de um determinado problema ainda é carente de experimentos. Pela falta de clareza dos autores dos algoritmos, a implementação se torna um desafio de força-bruta para ajustar alguns parâmetros que, por enquanto, são definidos empiricamente. Com novos trabalhos teóricos e práticos surgindo sobre as RNPs, as definições vão se tornando mais sólidas e os experimentos mais realistas, para que possam a partir daí determinar as aplicações que apresentam resultados melhores com esta abordagem.

Como visto no capítulo 6, analisando os resultados obtidos por Bohte [19], comparando com os abordados segundo a metodologia proposta, nota-se que as redes Neurais Pulsadas apresentam um resultado semelhante as Redes Neurais Artificiais, se analisados a taxa de acertos obtida. Ao se analisar do ponto de vista computacional, as RNPs possuem um custo maior que as RNAs tradicionais, devido necessitar de codificar informações de maneira temporal e exigir uma quantidade maior de neurônios na camada de entrada.

Para o problema de classificação destas bases, onde os dados são estáticos e não variam com o tempo, nenhuma das arquiteturas se destaca quanto a qualidade de solução, exceto se

não for utilizada a codificação pro campos receptivos e utilizar MatlabLM como método de treinamento. É possível que as RNPs tenham um resultado melhor que as RNAs em outras tarefas ou com outras bases de dados. Ao se analisar as iterações do treinamento, percebe-se que as RNPs apresentam um treinamento com menos iterações, que pode ser mais rápido.

7.1 Sugestão para trabalhos futuros

1. Estudar de maneira detalhada os modelos de neurônios do tipo de limiar de disparo, implementá-los e avaliar o seu comportamento;
2. Analisar o modelo de Resposta Impulsiva (SRM), analisar o algoritmo SpikeProp, escolher uma tarefa, implementá-los e avaliar o comportamento;
3. Estudar mais profundamente os métodos de codificação de entrada e saída;
4. Formalizar um algoritmo que possa explicar de maneira precisa o treinamento para RNPs;
5. Estudar a aplicação de RNPs à dados de natureza temporal (ECG ou EEG).

Referências Bibliográficas

- [1] HAYKIN, S. *Redes Neurais - Princípios e prática*. 2. ed. Porto Alegre: Bookman, 2001.
- [2] BRAGA, A. de P. *Redes Neurais Artificiais: Teoria e Aplicações*. 1. ed. Rio de Janeiro - RJ: LTC - Livros Técnicos e Científicos Editora S.A., 2000.
- [3] FAUSETT, L. *Fundamentals of Neural Networks*. 1. ed. New Jersey: Printice Hall, 1994.
- [4] TATIBANA, C. Y.; KAETSU, D. Y. *Uma Introdução às Redes Neurais*. Consultado na INTERNET: <http://www.din.uem.br/ia/neurais>, 28/06/2012.
- [5] JASKOWIAK, P. A. *Um Estudo Sobre as Redes Neurais Pulsadas*. Dissertação (Trabalho de Conclusão de Curso) — UNIOESTE – Universidade Estadual do Oeste do Paraná, Cascavel - PR, Dezembro 2008.
- [6] BOOIJ, O. *Temporal Pattern Classification using Spiking neural Networks*. Dissertação (Mestrado) — University of Amsterdam, Amasterdam, Agosto 2004.
- [7] KOLB, B.; WHISHAW, I. *An Introduction to Brain and Behavior*. 2. ed. New York: Worth Publishers, 2001.
- [8] TIMOSZCZUK, A. P. *Reconhecimento automático de locutor usando Redes Neurais Pulsadas*. Tese (Doutorado) — Universidade de São Paulo, São Paulo, SP, Dezembro 2004.
- [9] POSTAL, A. *Reconhecimento de Imagens de Impressões Digitais Utilizando Redes Neurais Artificiais*. Dissertação (Trabalho de Conclusão de Curso) — UNIOESTE – Universidade Estadual do Oeste do Paraná, Cascavel - PR, Dezembro 2001.
- [10] GERSTNER, W. *Pulsed Neural Networks*. 1. ed. Cambridgem, USA: MIT Press, 1999.

- [11] THORPE, S. J.; IMBERT, M. *Biological constraints on connectionist modelling*. [S.l.]: Elsevier, 1989. 63–92 p.
- [12] THORPE, S. J.; DELORME, A.; VANRULLEN, R. *Spike-based strategies for rapid processing*. Toulouse, France: Nature, 2001.
- [13] VANRULLEN, R.; GUYONNEAU, R.; THORPE, S. J. Spike times make sense. *Trends in Neurosciences*, Centre de Recherche Cerveau et Cognition, 133 Route de Narbonne, 31062 Toulouse Cedex, France. rufin.vanrullen@cerco.ups-tlse.fr, v. 28, n. 1, p. 1–4, jan. 2005. ISSN 01662236. Disponível em: <<http://dx.doi.org/10.1016/j.tins.2004.10.010>>.
- [14] GERSTNER, W.; KISTLER, W. M. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. 1. ed. New York: New York, 2002.
- [15] MAASS, W. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, v. 10, p. 1659–1671, 1996.
- [16] SIMÕES, A. da S. *Aprendizado não supervisionado em redes neurais pulsadas de base radial*. Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo, 2006.
- [17] IZHIKEVICH, E. M. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, The Neurosciences Institute, San Diego, CA 92121, USA. Eugene.Izhikevich@nsi.edu, v. 15, n. 5, p. 1063–1070, set. 2004. Disponível em: <<http://dx.doi.org/10.1109/TNN.2004.832719>>.
- [18] GERSTNER, W. Associative memory in a network of biological neurons. In: LIPPMANN, R.; MOODY, J. E.; TOURETZKY, D. S. (Ed.). *Advances in Neural Information Processing Systems 3, [NIPS Conference, Denver, Colorado, USA, November 26-29, 1990]*. [S.l.]: Morgan Kaufmann, 1990. p. 84–90. ISBN 1-55860-184-8.
- [19] BOHTE, S. M.; KOK, J. N.; POUTRÉ, J. A. L. Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, v. 48, n. 1-4, p. 17–37, 2002.
- [20] BELATRECHE, A.; MAGUIRE, L. P.; MCGINNITY, M. Advances in design and application of spiking neural networks. *Soft Comput.*, Springer-Verlag, Berlin, Hei-

- delberg, v. 11, n. 3, p. 239–248, out. 2006. ISSN 1432-7643. Disponível em: <<http://dx.doi.org/10.1007/s00500-006-0065-7>>.
- [21] BOHTE, S. M. *Spiking Neural Networks*. Tese (Doutorado) — Universiteit Leiden, 2003.
- [22] SPOREA, I.; GRÜNING, A. Reference time in spikeprop. *Proceeding of the International Joint Conference on Neural Networks (IJCNN 2011)*, San Jose, USA, 2011.
- [23] MCKENNOCH, S.; LIU, D.; BUSHNELL, L. G. Fast modifications of the spikeprop algorithm. In: *IJCNN*. IEEE, 2006. p. 3970–3977. Disponível em: <<http://dblp.uni-trier.de/db/conf/ijcnn/ijcnn2006.html>>.
- [24] MOORE, S. C. *Back-propagation in Spiking Neural Networks*. Tese (Doutorado) — University of Bath, 2002.
- [25] FISHER, R. *Iris Data Set*. 1936. Consultado na INTERNET: <http://archive.ics.uci.edu/ml/datasets/Iris>, 28/06/2012.
- [26] WOLBERG, W. H.; STREET, W. N.; MANGASARIAN, O. L. *Wisconsin Diagnostic Breast Cancer (WDBC)*. 1995. Consultado na INTERNET: [http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Original\)](http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)), 05/08/2012.