

MODELAGEM COMPUTACIONAL DA DOENÇA DE ALZHEIMER USANDO REDES DE VENN

Trabalho de Conclusão de Curso

Engenharia da Computação

Anderson Tenório Sergio

Orientador: Professor Fernando Buarque de Lima Neto

Recife, Dezembro de 2008

MODELAGEM COMPUTACIONAL DA DOENÇA DE ALZHEIMER USANDO REDES DE VENN

Trabalho de Conclusão de Curso

Engenharia da Computação

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

Anderson Tenório Sergio

Orientador: Professor Fernando Buarque de Lima Neto

Recife, Dezembro de 2008

ANDERSON TENÓRIO SERGIO

**MODELAGEM COMPUTACIONAL DA
DOENÇA DE ALZHEIMER USANDO
REDES DE VENN**

Agradecimentos

A Deus, por me dar discernimento, forças e fé para realizar meus sonhos.

Aos meus pais, Edite e Edson, pelo incentivo irrestrito, por serem meu porto seguro, por toda a educação e por todo o amor.

A toda a minha família, pelos bons momentos e pelos bons sentimentos compartilhados.

A todos os meus amigos, pelo apoio e pelo companheirismo.

Ao Professor Fernando Buarque, pela atuação e pela importância na minha vida acadêmica.

A Aunia, por ser a inspiração de tudo o que eu faço.

Obrigado a todos! Minhas conquistas também são de todos vocês!

Resumo

A Doença de Alzheimer é uma disfunção degenerativa do cérebro ainda sem cura, que atinge milhões de pessoas ao redor do mundo. Ao mesmo tempo em que modelos artificiais do cérebro podem representar e explicar síndromes neurológicas, redes de Venn têm a potencialidade de simular o comportamento desse tipo de patologia. Desse modo, pretendemos modelar computacionalmente a doença de Alzheimer e, através de simulações, realizar inferências acerca da prognose da patologia em modelos artificiais, para que no futuro, eventualmente sejam diminuídas as incertezas atuais de médicos e pacientes.

Na modelagem, foi criada uma rede neural artificial com estrutura baseada nas redes de Venn e algoritmo de treinamento do modelo de Hopfield. Essa rede neural foi treinada para reconhecer determinados padrões de treinamento, nesse caso, imagens binárias. Em seguida, a Doença de Alzheimer foi modelada computacionalmente levando-se em consideração alguns dos seus aspectos neuropatológicos. Através de diversas simulações, como previsto em hipótese, verificou-se que a aplicação da modelagem computacional da Doença de Alzheimer prejudicou o desempenho da rede neural treinada, simulando assim a atuação dessa patologia no cérebro humano.

Abstract

Alzheimer's disease is a degenerative disorder of the brain that is still without cure. It affects millions of people around the world. At the same time that artificial models of the brain may represent and explain neurological syndromes, Venn networks have the capability of simulating the behavior of this type of pathology. Thus, we intend to model, computationally, Alzheimer's disease and, through simulations, perform inferences about the prognosis of the pathology. This effort may, in the future, reduce uncertainties of doctors and patients.

During the modeling phase, we have developed an artificial neural network structure based on Venn networks and the training algorithm of standart Hopfield model. The neural network was trained to recognize certain patterns of training, in this case, binary images. On top of that the Alzheimer's disease was modeled computationally taking into consideration some of its neuropathological aspects. Through various simulations, we have found that the Alzheimer's disease model disturbed the performance of a regular trained neural network, thus mimicking the pathological effects in the human brain.

Sumário

Índice de Figuras	iv
Índice de Tabelas	v
Tabela de Símbolos e Siglas	vi
1 Introdução	7
1.1 Motivação	7
1.2 Objetivos e resultados esperados	8
1.3 Metodologia	9
1.4 Estrutura do trabalho	10
2 Fundamentação Teórica	11
2.1 A Doença de Alzheimer	11
2.1.1 Epidemiologia e etiologia	12
2.1.2 Aspectos neuropatológicos	12
2.2 Redes Neurais Artificiais	15
2.2.1 O neurônio artificial	15
2.2.2 O modelo de Hopfield	17
2.3 Redes de Venn	18
2.3.1 Estrutura das redes de Venn	19
3 Modelagem Computacional da Doença de Alzheimer	26
3.1 Modelos computacionais	26
3.2 Modelagem da Doença de Alzheimer	27
3.3 GVNS (Generalised Venn-Networks Simulator)	29
3.4 GVNS2	32
3.5 Módulo para modelagem computacional da Doença de Alzheimer	34
4 Resultados	39
4.1 Perda de neurônios	41
4.2 Atrofia neuronal	42
4.3 Perda sináptica	44
4.4 Experimentos mesclados	45
5 Conclusão	47
5.1 Discussão	47
5.2 Trabalho futuros	48

Índice de Figuras

Figura 1.	<i>'Neurofibrillary tangles'</i>	14
Figura 2.	Placas neuríticas.....	14
Figura 3.	Neurônio biológico (a) e neurônio artificial (b)	16
Figura 4.	Principais funções de ativação.....	16
Figura 5.	Esquema de uma rede de Hopfield	17
Figura 6.	Redes de Venn – Unidades de Processamento	20
Figura 7.	Redes de Venn – Regiões.....	21
Figura 8.	Redes de Venn – Fibras <i>Afferents</i>	22
Figura 9.	Redes de Venn – <i>U-fibras</i> tipo 1.....	23
Figura 10.	Redes de Venn – Fibras <i>Efferents</i>	23
Figura 11.	Redes de Venn – Fibras <i>Efferent-feedback</i>	24
Figura 12.	Redes de Venn – Fibras <i>U-fibras</i> tipo 2.....	24
Figura 13.	Modelo básico de uma rede de Venn.....	25
Figura 14.	Visão esquemática da modelagem computacional da Doença de Alzheimer usando a topologia das redes de Venn.	29
Figura 15.	Casos de uso iniciais do GVNS	30
Figura 16.	Diagrama de classes do GVNS.....	31
Figura 17.	Diagrama de seqüência do GVNS	31
Figura 18.	Herança na implementação de unidades de processamento. “Neuron” representa a classe-mãe.	33
Figura 19.	Herança na implementação das fibras. “Fibras” representa a classe-mãe.	34
Figura 20.	Classe Neuron.....	34
Figura 21.	Método <i>act</i> da classe <i>Neuron</i>	35
Figura 22.	Classe <i>Hopfield</i>	35
Figura 23.	Método <i>generateWeightsMatrix()</i>	36
Figura 24.	Classe <i>Alzheimer</i>	36
Figura 25.	Método que modela a perda de neurônios.....	37
Figura 26.	Método que modela a atrofia neuronal.	38
Figura 27.	Método que modela a perda de sinapses.	38
Figura 28.	Padrões de treinamento do experimento computacional.....	39
Figura 29.	Exp. 1 – Padrão de treinamento (a) com 20% de ruído (b) e resposta da rede (c).	41
Figura 30.	Exp. 2 – Padrão de treinamento (a) com 30% de ruído (b) e resposta da rede (c).	42
Figura 31.	Exp. 3 – Padrão de treinamento (a) com 20% de ruído (b) e resposta da rede (c).	43
Figura 32.	Exp. 4 – Padrão de treinamento (a) com 30% de ruído (b) e resposta da rede (c).	43
Figura 33.	Exp. 5 – Padrão de treinamento (a) com 20% de ruído (b) e resposta da rede (c).	44
Figura 34.	Exp. 6 – Padrão de treinamento (a) com 30% de ruído (b) e resposta da rede (c).	45
Figura 35.	Exp. 7 – Padrão de treinamento (a) com 20% de ruído (b) e resposta da rede (c).	46
Figura 36.	Exp. 8 – Padrão de treinamento (a) com 20% de ruído (b) e resposta da rede (c).	46

Índice de Tabelas

Tabela 1.	Comportamento da rede neural treinada para recuperação de padrões.....	40
Tabela 2.	Exemplo da topologia utilizada nas simulações.	40
Tabela 3.	Experimento 1 – Perda de Neurônios (ruído de 20%)	41
Tabela 4.	Experimento 2 – Perda de Neurônios (ruído de 30%)	42
Tabela 5.	Experimento 3 – Atrofia neuronal (ruído de 20%)	42
Tabela 6.	Experimento 4 – Atrofia neuronal (ruído de 30%)	43
Tabela 7.	Experimento 5 – Perda sináptica (ruído de 20%)	44
Tabela 8.	Experimento 6 – Perda sináptica (ruído de 30%)	44
Tabela 9.	Experimento 7 – Experimento mesclado (ruído de 20%)	45
Tabela 10.	Experimento 8 – Experimento mesclado (ruído de 20%)	46

Tabela de Símbolos e Siglas

(Dispostos por ordem de aparição no texto)

RNA:	Rede Neural Artificial
DA:	Doença de Alzheimer
GVNS:	Generalised Venn-Networks Simulator
OO:	Orientação a Objeto

Capítulo 1

Introdução

1.1 Motivação

A neurociência computacional destaca-se na computação inteligente por buscar modelar aspectos anatômicos e fisiológicos (forma e comportamento, respectivamente) do cérebro e demais componentes do sistema nervoso. Através de modelos matemáticos e computacionais, é possível tentar compreender a função e os mecanismos dessa complexa composição. É uma ciência intrinsecamente interdisciplinar, abrangendo áreas que vão desde a psicologia até a física e a matemática pura.

Mais especificamente, a modelagem do cérebro é atualmente um dos grandes desafios para os pesquisadores da área. Ao passo que o conhecimento acerca do funcionamento desse órgão vital animal é ainda um tanto restrito, a sua representação em um ambiente não natural parece ser um desafio no mínimo interessante. Entretanto, a dificuldade dessa tarefa não impede que diversas pesquisas ocorram com sucessos de variados níveis [1], sob diversas frentes e tendo alicerce em diferentes áreas do conhecimento.

Este trabalho é, *a priori*, uma modelagem computacional do cérebro humano sob um alvo específico, utilizando para isso redes neurais artificiais. Trata-se do estudo da atuação de uma disfunção neurológica, a Doença de Alzheimer, na representação de um cérebro natural sadio. Tal tarefa possui características muito peculiares. Como fator encorajador, as funções do cérebro podem ser consideradas, em micro-escala, processos computacionais como outro qualquer. Antes que emoções complexas sejam produzidas e finalmente a mente emerja de um sistema natural, cada um dos estímulos é processado por essas unidades mínimas de processamento – os neurônios e seus agrupamentos funcionais. Eles funcionam justamente como um sistema

computacional, ou seja, uma (ou mais) entrada é processada para que uma (ou mais) saída seja produzida.

Por outro lado, o cérebro definitivamente não é um sistema monolítico e desprovido de organização. Nele, diversos níveis são interligados para que o comportamento intrinsecamente humano possa surgir a partir de pequenos e numerosos agrupamentos funcionais de processamento.

Essa multitudine é que justamente dificulta sobremaneira a modelagem computacional do cérebro. Portanto a escolha de qual nível o modelo irá representar é crucial para o investigador. Obviamente, essa decisão pode terminar por simplificar demais o modelo, diminuindo assim a sua capacidade de generalização e extensão.

É importante notar que simulações mesmo mais elaboradas podem estar longe da real capacidade computacional de um sistema biológico. Ao longo deste trabalho, incluímos reflexões sobre essas questões, *i.e.*, vantagens e desvantagens de modelos computacionais do cérebro.

1.2 Objetivos e resultados esperados

O trabalho de conclusão de curso apresentado nesta monografia tem como objetivo principal a modelagem computacional da Doença de Alzheimer (DA).

DA é uma doença ainda sem cura que atinge geralmente pessoas a partir dos 50 anos de idade. Ela deteriora algumas regiões do cérebro, alterando o comportamento físico e mental, a linguagem e outras funções cognitivas [2].

Este trabalho torna-se ainda mais interessante pelo fato de que a relação causa-efeito dos mecanismos de atuação da neuropatologia é ainda pouco conhecida. Além disso, as simulações de modelos de disfunções neurológicas como as propostas aqui podem capturar relações entre a atividade das estruturas modeladas do cérebro e as suas características neuroanatômicas, em específico a influência de lesões e alterações neuropatológicas em relação ao comportamento e performance cognitiva [3]. Após a modelagem, todas as informações julgadas necessárias durante o estudo dessa disfunção serão aplicadas e simuladas num simulador de Redes de Venn.

Redes de Venn são redes neurais artificiais com ainda mais inspiração biológica se comparadas com outras arquiteturas de Redes Neurais Artificiais (RNA's). Essa arquitetura foi proposta por Buarque em sua tese de doutorado [4]. As redes de Venn são formadas por unidades de processamento que se organizam em camadas e se interligam umas às outras, simulando assim

o neurônio e suas diversas conexões. Buarque também produziu em seu trabalho de doutorado um simulador para esse tipo de rede neural artificial, o GVNS (*Generalised Venn-Networks Simulator*). Adicionalmente, a aplicação desse sistema na simulação de doenças degenerativas do cérebro é aceitável, visto que nele já foi simulada a esclerose múltipla, por exemplo. A esclerose múltipla possui algumas características que se assemelham à Doença de Alzheimer [5].

Como a modelagem computacional da DA é o resultado principal deste trabalho de conclusão de curso, houve a necessidade de modificação no simulador. O GVNS, apesar de oferecer a possibilidade de topologias complexas (multiplicidade das unidades de processamento, regiões de processamento, tipos de fibra e fontes de entrada e saída), não possui características de programação orientada a objetos. Tais características, como reuso de código, manutenção de aplicativo e modularidade, facilitariam em muito sua utilização nessa e em outras aplicações. Dessa forma, também é um objetivo a co-construção de outra versão desse simulador que incorpore as características supracitadas.

O trabalho pretende gerar como impacto principal a possibilidade de validação da simulação de mais uma doença degenerativa do cérebro utilizando o GVNS e conseqüentemente as redes de Venn. Através dessa aplicação, será possível realizar algumas inferências acerca da prognose dessa disfunção cerebral, diminuindo as incertezas atuais.

1.3 Metodologia

Primeiramente, é necessário estudar os mecanismos de atuação da Doença de Alzheimer. Mais especificamente, de que forma a patologia atua na anatomia e fisiologia da rede neural (biológica). A partir daí, deve-se definir como essas características devem ser modeladas computacionalmente e aplicadas no simulador de redes de Venn. Todas as características implementadas terão seus resultados analisados e validados. Assim, as atividades a serem realizadas podem ser enumeradas como se segue:

1. Estudo da literatura:
 - a. Modelagem computacional de doenças neurológicas;
 - b. Mecanismos de atuação da doença de Alzheimer;
 - c. Arquitetura das Redes de Venn;
 - d. Redes Neurais de Hopfield.
2. Definição de quais características da DA serão modeladas.

3. Implementação de um módulo no simulador de redes de Venn que treine uma rede neural artificial do tipo memória-associativa. Será utilizado o algoritmo de treinamento das redes de Hopfield.
4. Realização de experimentos da atuação da simulação da DA na rede neural treinada.
5. Realização de inferências a partir da aplicação da modelagem da DA em uma rede neural treinada;

1.4 Estrutura do trabalho

Esta monografia está organizada como especificado a seguir. Todo o referencial teórico é apresentado antes que a contribuição seja fundamentada e detalhadamente descrita. Finalmente, seguem-se simulações realizadas e o trabalho é finalizado com as conclusões e as perspectivas para trabalhos futuros.

Capítulo 1: Contém esta introdução que apresenta o contexto no qual o trabalho está inserido, apresenta sua motivação, o problema e os resultados esperados.

Capítulo 2: Aqui, toda a fundamentação teórica será mostrada. Primeiramente, a Doença de Alzheimer será apresentada. As características gerais da patologia serão mostradas, destacando-se a especificação da forma com que a doença atua na modificação da anatomia e nas características fisiológicas do cérebro humano. Após uma breve revisão acerca de redes neurais artificiais e a apresentação do modelo de Hopfield, as redes de Venn serão apresentadas em detalhe; principalmente, seus conceitos e arquitetura.

Capítulo 3: Nesse capítulo, a solução para o problema (*i.e.*, a modelagem computacional da Doença de Alzheimer) será descrita. O foco será na forma como as características da patologia foram transpostas para o ambiente computacional considerado.

Capítulo 4: Aqui, as simulações para a modelagem computacional da DA serão descritas e apresentadas.

Capítulo 5: Este capítulo apresenta as conclusões acerca deste trabalho de conclusão de curso. Além disso, serão discutidos os impactos que uma modelagem de uma disfunção neurológica podem gerar. Ao final, algumas motivações para trabalhos futuros serão explicitadas.

Capítulo 2

Fundamentação Teórica

Neste capítulo será apresentado todo o embasamento para a modelagem computacional da Doença de Alzheimer que foi desenvolvido. Primeiramente, a DA propriamente dita é apresentada. O foco serão seus aspectos neuropatológicos, ou seja, de que forma a disfunção atinge o funcionamento do cérebro e todo o sistema neural.

Em seguida, será feita uma breve discussão acerca das redes neurais artificiais em sua forma geral, bem como o modelo de treinamento utilizado na modelagem (modelo de Hopfield). Finalmente, as redes de Venn serão abordadas, centrando-se em seus conceitos e estrutura.

2.1 A Doença de Alzheimer

A Doença de Alzheimer é essencialmente uma disfunção neurológica. Foi primeiramente descrita por Alois Alzheimer em 1907, numa mulher de 51 anos de idade que apresentava um rápido progresso de demência [6]. Os efeitos da doença são muitos e variados. A depender do paciente e da forma como a doença atua no cérebro, a DA pode caracterizar no enfermo perda de funções intelectuais (*e.g.*, memória, linguagem, habilidades espaciais, visão, capacidade de resolver problemas, etc.), anormalidades comportamentais e distúrbios em funções executivas e motoras.

A disfunção, sempre progressiva, causa a perda da capacidade do doente de se socializar, e inclusive, em certos estágios da doença, torna-o incapaz de comunicar-se satisfatoriamente com o mundo exterior. A Doença de Alzheimer é de grande sofrimento para o doente e também para sua família.

2.1.1 Epidemiologia e etiologia

À medida que a idade avança, as chances de alguém ser acometido pela DA aumentam. A doença ou mesmo indícios dela podem estar presentes em 3% das pessoas com idade entre 65 e 74 anos, 19% para aqueles com idade entre 75 e 84 anos e 47% para indivíduos com mais de 85 anos [7]. Mulheres têm um risco maior de desenvolverem a disfunção [8], enquanto que a incidência é maior em homens brancos do que em negros [9]. A expectativa de vida para indivíduos doentes varia de três a vinte anos.

A etiologia da doença de Alzheimer é ainda hoje incerta. Entretanto, alguns fatores de risco podem ser listados: etnia, idade avançada, traumas no crânio, limitações educacionais, hipertensão arterial, diabetes *melitus*, infarto do miocárdio, dentre outros. A incidência também pode estar relacionada com o histórico familiar [10]. Alguns pesquisadores têm buscado uma origem genética para a doença de Alzheimer. Mutações no gene da proteína precursora do amilóide, localizado no cromossomo 21, aumentam a formação de beta-amilóide [11]. O beta-amilóide é uma substância tóxica acumulada em cérebros com DA que pode causar disfunções e mortes neuronais. Outro gene, localizado no cromossomo 19, pode também estar ligado à doença de Alzheimer.

2.1.2 Aspectos neuropatológicos

Os aspectos neuropatológicos da DA são bastante diversos. Há principalmente uma atrofia progressiva e morte de determinadas populações neurais [2]. Adicionalmente, também são observadas mudanças no citoesqueleto e formação de placas neuríticas relacionadas a uma deposição anormal de amilóide no cérebro. O avanço da doença também pode resultar na diminuição da velocidade da condução dos sinais e também na perda de conexões entre os neurônios. Sintomas esses similares a esclerose múltipla [4]. A seguir, alguns dos principais mecanismos de atuação da doença de Alzheimer.

Perda de Neurônios

A perda de neurônios é um dos efeitos mais presentes na doença de Alzheimer. Entretanto, é uma característica difícil de ser estudada. Isso porque a diminuição do número de neurônios é um fenômeno natural em indivíduos mais idosos. Porém, nem todos os tipos dessa célula são afetados, sendo esse efeito mais comum em neurônios piramidais das camadas III e V do córtex cerebral. Dependendo da idade, a perda varia entre 38% e 67% [12], [13]. Essa característica

também está presente nas regiões eferentes e aferentes do hipocampo. A perda de neurônios também reflete fortemente na perda de memória de longo e curto prazo do enfermo [14].

Atrofia neuronal

Assim como a perda de neurônios, a atrofia neuronal está intimamente ligada com a severidade da demência. Em indivíduos doentes, é possível observar uma diminuição progressiva no tamanho do núcleo celular [15] e na quantidade de ácidos ribonucléicos no citoplasma [15], [16]. A atrofia neuronal é principalmente causada pela diminuição da capacidade de o cérebro afetado pela DA de sintetizar proteínas [2]. Por esse motivo, também é possível observar uma determinada perda dendrítica.

‘Neurofibrillary tangles’ e placas neuríticas

‘*Neurofibrillary tangles*’ também podem estar presentes em outras disfunções neurológicas [17]. Diferentemente da perda de neurônios e atrofia neuronal, pode não estar relacionada com o aumento da demência em indivíduos afetados. Sua presença não é comum em todos os tipos de neurônios. Por exemplo, as chamadas células de Purkinge nunca são afetadas [18]. A figura 1 mostra a presença de ‘*neurofibrillary tangles*’. A formação dos ‘*neurofibrillary tangles*’ está relacionada com o depósito da proteína *tau* no citoplasma das células [19]. Dessa forma, pelo fato de o *tau* está concentrado no axônio do neurônio [20], há mudanças significativas nesse componente.

Placas neuríticas, facilmente encontradas em cérebros com DA e também chamadas de placas de amilóide ou ainda placas senis, são substâncias esféricas com diâmetro próximo dos 200 micrômetros. São geralmente constituídas de um núcleo central de amilóide. A deposição extracelular de placas neuríticas tem considerável importância na degeneração sináptica. Sua existência pode resultar de disfunções de diversas proteínas sinápticas, incluindo a proteína precursora do amilóide, a APP. A presença de placas neuríticas é mais comum no chamado córtex associativo, mais especificamente nas camadas II e III. Interessante notar que o número de placas neuríticas pode não estar ligado à perda de funções intelectuais. Inclusive, um grande número de placas pode ser observado em indivíduos sadios com idade avançada [21]. A figura 2 mostra a presença de placas neuríticas no neocórtex.

Importante saber que placas e ‘*neurofibrillary tangles*’ podem causar patologias sinápticas. Em contrapartida, como num círculo vicioso, alterações sinápticas podem provocar a formação dessas substâncias.

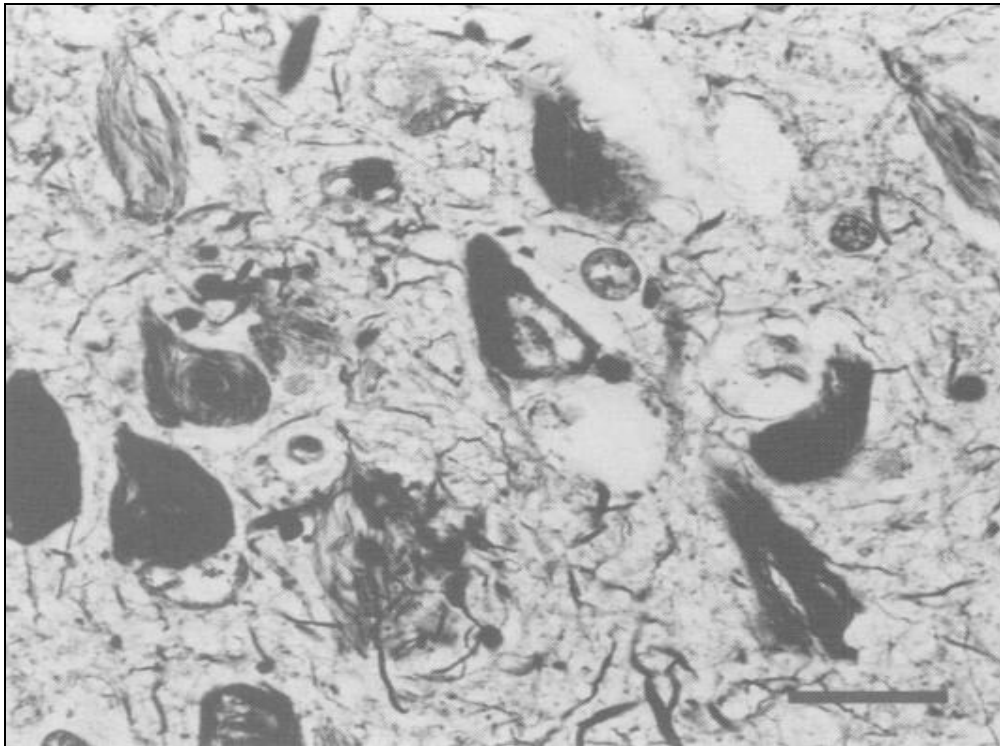


Figura 1. *'Neurofibrillary tangles'*

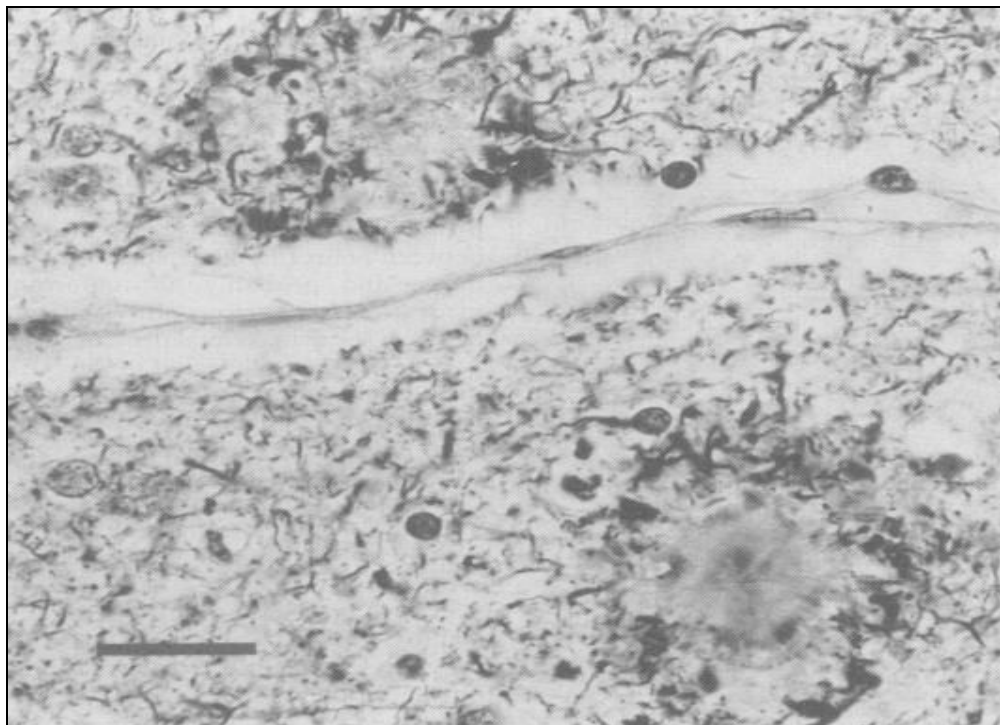


Figura 2. Placas neuríticas

2.2 Redes Neurais Artificiais

2.2.1 O neurônio artificial

Redes Neurais Artificiais são sistemas paralelos distribuídos compostos por unidades de processamento simples que calculam funções matemáticas. As unidades, por vezes chamadas nodos, ou simplesmente neurônios, são dispostas em camadas e interligadas por conexões. Por sua vez, as conexões são associadas a determinados pesos. RNA's têm inspiração biológica. São baseadas na forma como o cérebro animal processa informações e gera *outputs* a partir de certos estímulos.

O primeiro progresso nesse sentido foi fruto do trabalho de Warren McCulloch e Walter Pitts em 1943 [22]. O modelo hoje conhecido como McCulloch-Pitts (MCP) é uma simplificação do funcionamento de um neurônio natural. A figura 3 mostra o esquema de um neurônio biológico seguida de sua representação matemática proposta por estes pesquisadores.

O neurônio biológico, de forma simplificada, recebe estímulos de outras unidades conectadas a ele através dos dendritos. O sinal é então processado no corpo celular e transmitido através do axônio. No modelo artificial, existem n terminais de entrada x_1, x_2, \dots, x_n , representando os dendritos. Para representar o axônio, tem-se o terminal de saída y . Quanto as sinapses (conexões entre diferentes neurônios), para cada entrada existe um peso associado w_1, w_2, \dots, w_n . Assim, a emulação do corpo do neurônio recebe a soma das entradas ponderadas por seus respectivos pesos. Para simular o chamado limiar de excitação do neurônio biológico, a saída do modelo artificial só é ativada quando um determinado *threshold* é atingido. Dessa forma, a representação matemática do modelo MCP é dada pela equação 1:

$$\sum_{i=1}^n x_i w_i \geq \theta$$

(Eq. 1)

O modelo MCP original tem saída binária. O neurônio artificial está ativo ou não. Ao longo dos anos, outros modelos foram propostos para que a unidade de processamento produzisse uma saída qualquer, através da adoção de funções não lineares chamadas funções de ativação. Existem diversas delas, a figura 4 ilustra alguns modelos.

É interessante notar que o modelo MCP e seu algoritmo de aprendizado representaram apenas os passos iniciais da pesquisa e utilização de redes neurais artificiais. Hoje, a literatura possui diversas arquiteturas e várias técnicas de aprendizado da rede [23], [24].

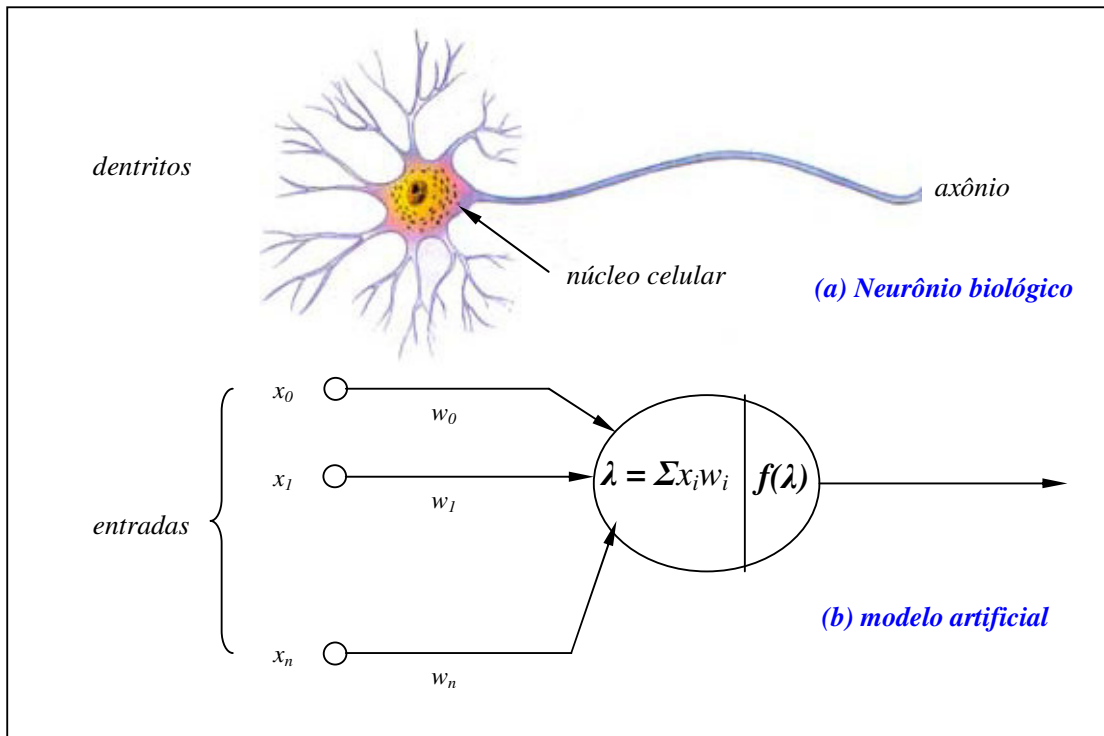


Figura 3. Neurônio biológico (a) e neurônio artificial (b)

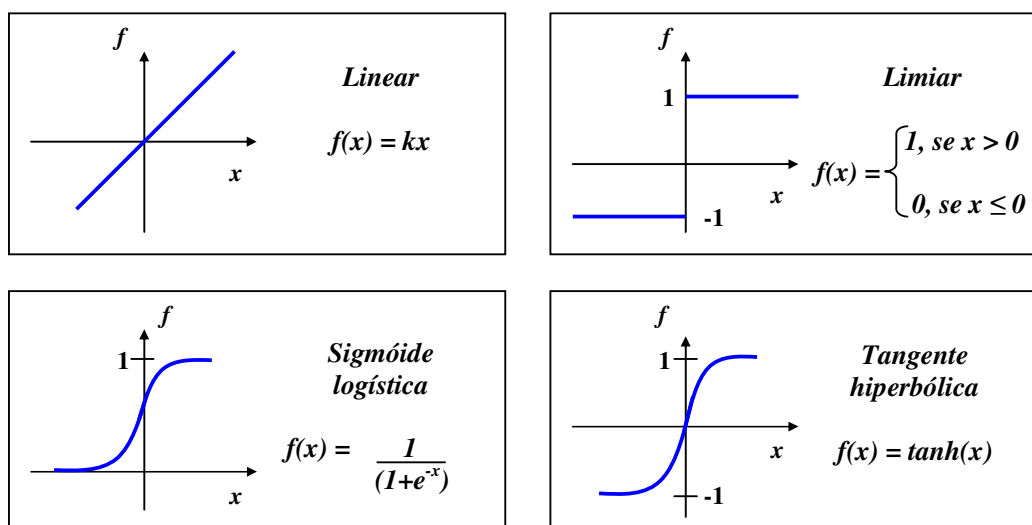


Figura 4. Principais funções de ativação

2.2.2 O modelo de Hopfield

As redes de Hopfield foram descritas em 1982 [25]. Nesse tipo de rede neural artificial, as saídas estão ligadas às entradas por um atraso de tempo, sendo assim um modelo matricial não-linear recorrente. Desse modo, a resposta da rede sempre depende do seu estado no intervalo de tempo anterior. As redes de Hopfield armazenam padrões para que estes sejam recuperados a partir de certos estímulos. A figura 5 representa o esquema de uma rede de Hopfield.

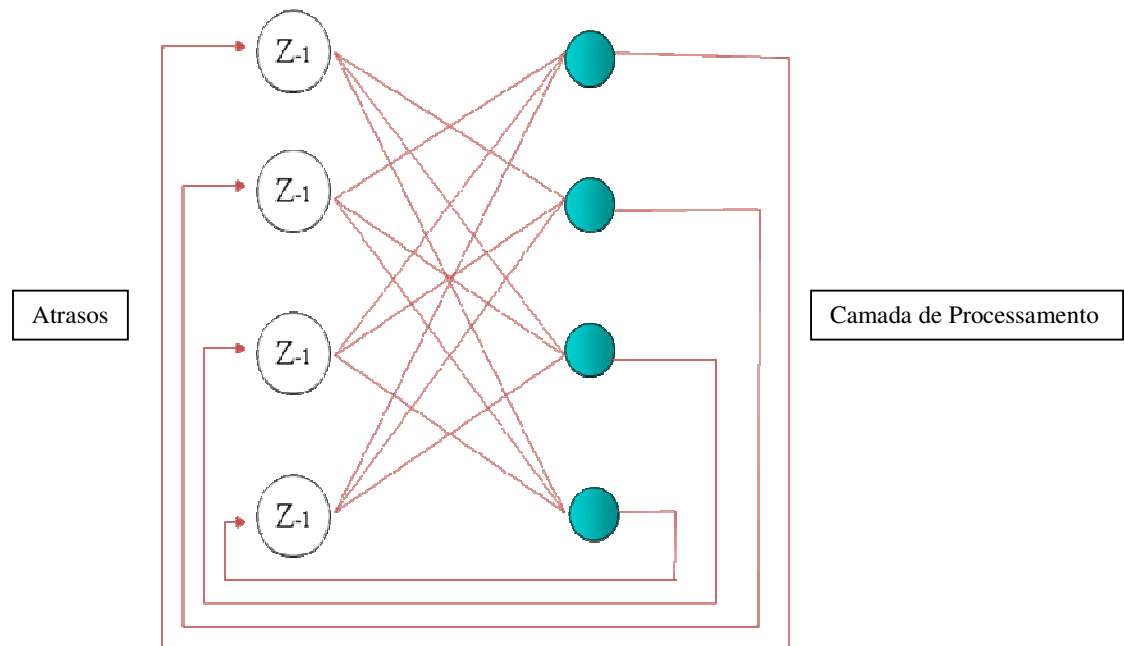


Figura 5. Esquema de uma rede de Hopfield

De acordo com a equação 2, a saída de um nó i num instante de tempo $t + 1$ é obtida através da soma ponderada das outras entradas menos o limiar. A única restrição nesse sentido, é que no modelo original, não existe conexão de um nó para si mesmo. Isso quer dizer que os elementos da matriz de pesos w_{ii} são nulos. Essa característica mantém a estabilidade da rede, ao mesmo tempo em que evita realimentações positivas. No trabalho original, os nós são do tipo MCP, isto é, somente podem assumir valores -1 ou 1 . Entretanto, algum tempo depois Hopfield mostrou que a utilização de saídas contínuas preserva as características do modelo original [26]. O modelo de Hopfield funciona de forma assíncrona, ou seja, as saídas são atualizadas em tempos diferentes. A cada iteração, um nó é selecionado aleatoriamente e sua saída é calculada.

$$x_i(t+1) = \operatorname{sgn} \left(\sum_{j=1}^p w_{ij} x_j(t) - \theta_i \right) \quad (\text{Eq. 2})$$
$$\operatorname{sgn}(x) = \begin{cases} +1, & \text{se } x \geq 0 \\ -1, & \text{se } x < 0 \end{cases}$$

Uma matriz de pesos torna-se estável quando, dado como entrada um vetor x , tem-se que $\operatorname{sgn} = x_i$ para todo i . Essa condição pode ser alcançada se treinarmos a rede com a regra de Hebb [23]. Assim, deve-se fazer $w_{ij} = x_i x_j$. Uma constante de proporcionalidade pode ser adicionada. Sendo k o número de nodos da rede, a condição de estabilidade pode ser considerada como $w_{ij} = i/k x_i x_j$. Essa divisão do vetor de pesos por k é equivalente a normalizar os vetores de entrada x . Essa operação elimina parte do ruído na recuperação de informação em memórias matriciais com treinamento por regra de Hebb [24]. Apesar de essa regra, como brevemente apresentada, armazenar informações de apenas um vetor de entrada, ela pode ser facilmente extrapolada para o armazenamento de n vetores.

Uma das características mais importantes do modelo de Hopfield é o conceito de energia. Hopfield mostrou em seu trabalho que um valor de energia pode estar associado a cada estado da rede e que esta energia decresce de maneira monotônica à medida que uma trajetória é descrita no espaço de estados em direção a um ponto fixo (*atrator*). A estabilização da rede ocorre quando os pontos estáveis de energia mínima criados através do processo de treinamento são alcançados. O modelo de Hopfield está então relacionado com sistemas físicos.

É importante notar a relação entre o modelo de Hopfield e memórias auto-associativas. É possível treinar uma rede desse tipo com diversos padrões de entrada. É esperado que essa rede, dado um padrão qualquer, apresente como resposta a entrada que mais se assemelha a esse padrão. Por exemplo, a rede pode recuperar informações de uma entrada com ruídos.

2.3 Redes de Venn

Não obstante a constituição do cérebro ser complexa e diversa [27], a maioria dos sistemas conexionistas não explora essa conhecida diversidade. Na maioria dos casos, os sistemas neurais artificiais possuem apenas um tipo de neurônio para todo o processamento. Essa característica é

ainda mais falha ao perceber que a estrutura cerebral é um importante aspecto da computação emergente. Nesse sentido, Buarque [4] propôs em sua tese de doutorado as redes de Venn. Redes de Venn são uma arquitetura de rede neural artificial que admite a definição de vários tipos de unidades de processamento, além de permitir que essas unidades agrupem-se entre si em regiões, cada qual com suas propriedades. A denominação “Venn” tem origem na semelhança entre os mapas de duas dimensões contendo diversas regiões utilizadas no processamento da rede e os diagramas de Venn.

Sistemas inteligentes podem ser analisados a partir de características como flexibilidade, adaptabilidade e escalabilidade. Especificamente em sistemas conexionistas, tem-se que a flexibilidade aborda o quanto a estrutura do sistema é mutável. A adaptabilidade aponta o quanto a eficiência é prejudicada frente a mudanças nas condições ambientais. Já a escalabilidade diz respeito a quanto o consumo computacional cresce com a complexidade da tarefa executada. É nesse aspecto que as redes de Venn são consideradas flexíveis, por aceitarem diversos tipos de unidades de processamento, definição de sub-regiões e conectividade não-trivial. Por outro lado, a possibilidade de se aplicar diversos algoritmos de aprendizado, as redes de Venn também são altamente adaptáveis e escaláveis.

2.3.1 Estrutura das redes de Venn

Unidades de Processamento

Kandel [27] e Levitan [28] mostram que nos sistemas neurais naturais, as unidades de processamento apresentam comportamentos diferentes entre si principalmente pela diversidade na codificação de seus sinais e nas propriedades químicas e elétricas. Nas redes de Venn, o conjunto dos atributos que definem as características funcionais é denominado *Unit Type* (UT), resultando numa unidade de processamento (PU – *Processing Unit*, no original em inglês). A equação 3 mostra a especificação formal desse conceito:

$$\begin{array}{l}
 UT_m \in \{unit_types\} \mid n = 0,1,\dots \\
 PU_n \in \{processing_units\} \mid n = 0,1,\dots \\
 PU_n \longleftarrow UT_m
 \end{array}
 \tag{Eq. 3}$$

Nas redes de Venn, uma unidade de processamento é assumida como uma coluna cortical, um conjunto de neurônios com propriedades e respostas a estímulos semelhantes. A figura 6 mostra diversas unidades de processamento num mapa de duas dimensões.

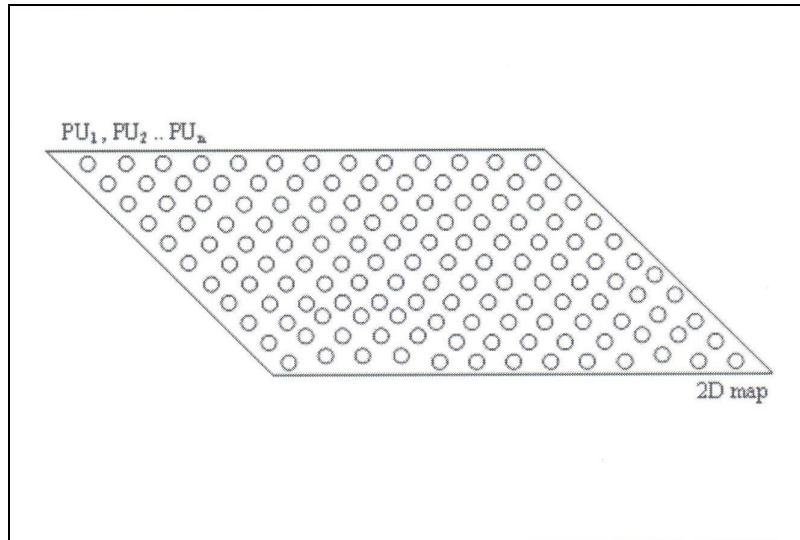


Figura 6. Redes de Venn – Unidades de Processamento

Regiões de Processamento

Nas redes de Venn, unidades de processamento podem se agrupar em regiões. Nas regiões, as unidades de processamento são do mesmo tipo, topologicamente vizinhos, tendo padrões de conectividade semelhantes. A equação 4 mostra esse conceito, na medida em que a figura 7 ilustra diversas regiões em um mapa de duas dimensões.

$$\begin{array}{l}
 R_o \in \{cortical_regions\} \mid o = 0, 1, \dots \\
 R_o \longleftarrow PU_n \\
 R_o \ni PU_n, \text{ iff}(UT(PU_n)) = m \\
 R_o \longleftarrow UT_m
 \end{array}
 \tag{Eq. 4}$$

Stimuli Sources e Effectors

As diversas regiões dentro do cérebro podem emitir ou receber sinais através do sistema nervoso. É nesse sentido que as redes de Venn incorporam o conceito de *Stimuli Sources* e *Effectors*, respectivamente os originadores e alvo da comunicação neural. Adicionalmente, há o conceito de que esses componentes podem ter diferentes cardinalidades para refletir diferenças no tamanho e na importância. A equação 5 formaliza essa dinâmica.

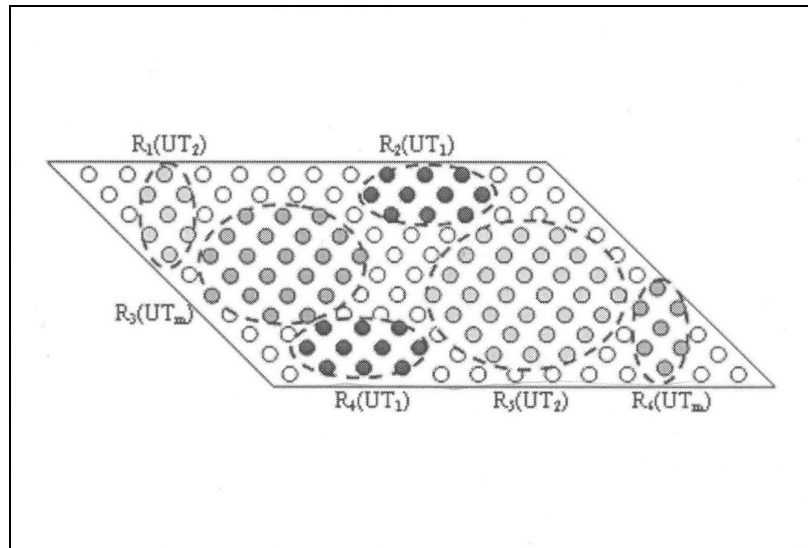


Figura 7. Redes de Venn – Regiões

$S_p^{fs} \in \{stimuli_sources\} \mid p = 1, 2, \dots$ $E_q^{te} \in \{effectors\} \mid q = 0, 1, \dots$ $cS = cardinality_stimuli \mid cS > 0$ $cE = cardinality_effectors \mid cE > 0$	(Eq. 5)
--	---------

Conexões

As redes de Venn oferecem a possibilidade de uma conectividade paralela e não-trivial. Para tanto, há o conceito de fibras (i.e. axônios). As fibras fazem a conexão entre os diversos componentes das redes de Venn: *stimuli sources*-região, região-região, região-*effector* e *effector*-região. Assim, de acordo com os componentes que as fibras conectam, elas podem ser classificadas em quatro tipos:

- Fibras que conectam *stimulus* S_p e região R_o ;
- Fibras que conectam uma região R_{o1} e outra região R_{o2} ;
- Fibras que conectam uma região R_o e *effector* E_q ;
- Fibras que conectam um *effector* E_q e uma região R_o .

Essa definição é formalizada pela equação 6.

$$F_r^{cO,cT} \in \{\text{fibras}\} \mid r = 0, 1, \dots$$

$$O = \text{signal_origin}$$

$$T = \text{signal_target}$$

$$F_r^{cO,cT} = [O, T] \mid [O, T] = \{(S_p, R_o), (R_{o1}, R_{o2}), (R_o, E_q), (E_q, R_o)\}$$

$$cO = \text{cardinality_origin} \mid cO \geq 0$$

$$cT = \text{cardinality_target} \mid cT \geq 0$$

(Eq. 6)

Outra classificação categoriza as fibras quanto ao seu papel num ciclo de processamento das redes de Venn. Nesse caso, as fibras podem ser:

- *Afferents*: sinais vindos a partir de *Stimulus* (figura 8);
- *U-fibres* tipo 1: sinais vindos após as fibras aferentes (figura 9);
- *Efferents*: enviam sinais aos *effectors* (figura 10).
- *Efferent-feedback*: sinais vindos a partir de *effectors* (figura 11);
- *U-fibres* tipo 2: sinais vindos após as fibras de *feedback* (figura 12).

A figura 13 ilustra uma visão esquemática de todos os componentes das redes de Venn.

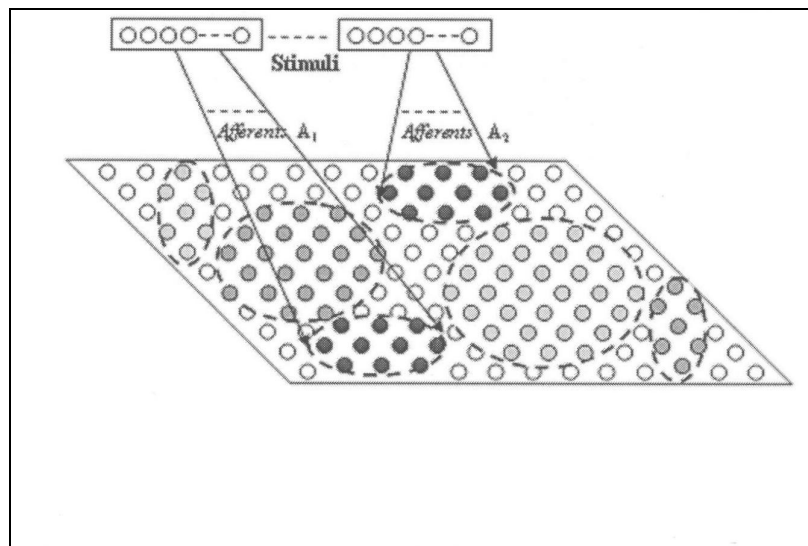


Figura 8. Redes de Venn – Fibras *Afferents*

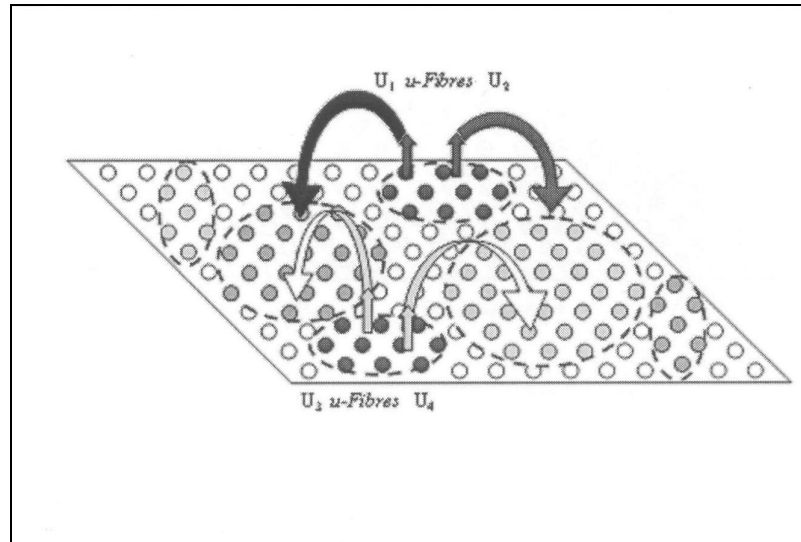


Figura 9. Redes de Venn – *U-fibres* tipo 1

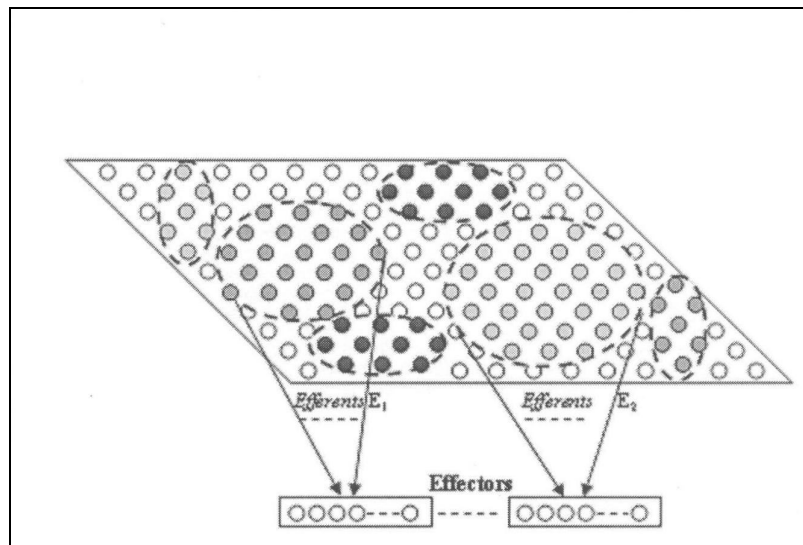


Figura 10. Redes de Venn – Fibras *Efferents*

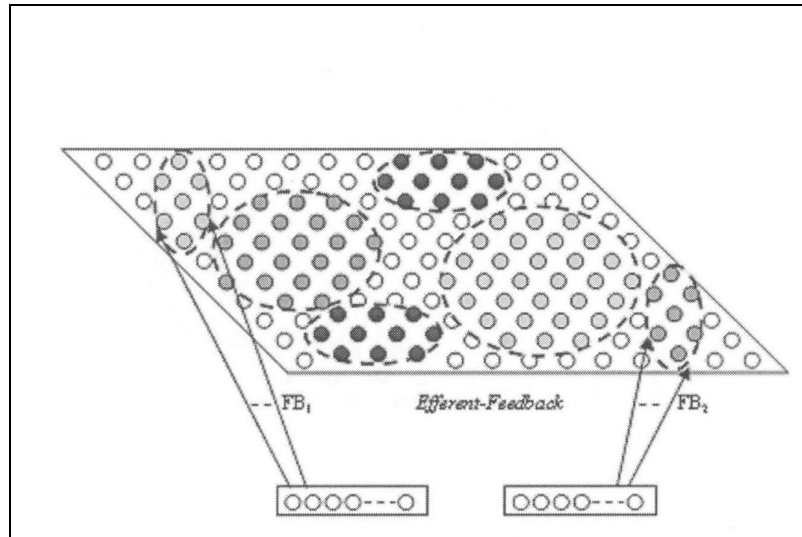


Figura 11. Redes de Venn – Fibras *Efferent-feedback*

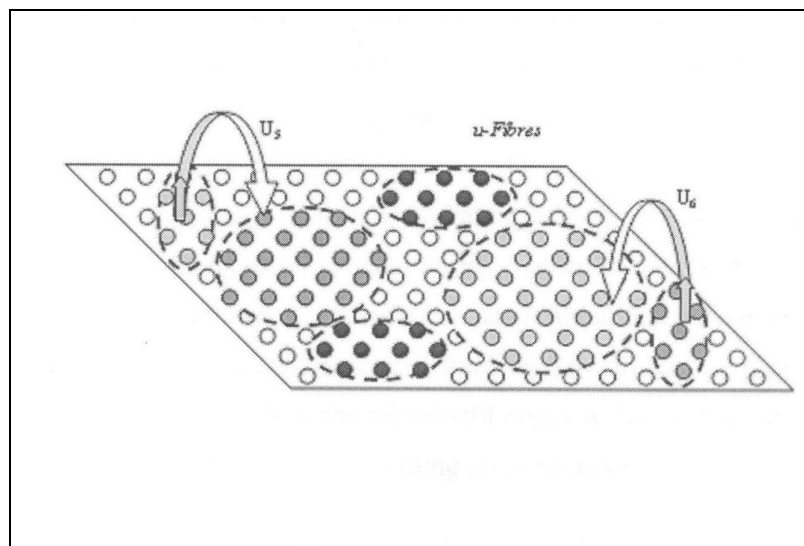


Figura 12. Redes de Venn – Fibras *U-fibres* tipo 2

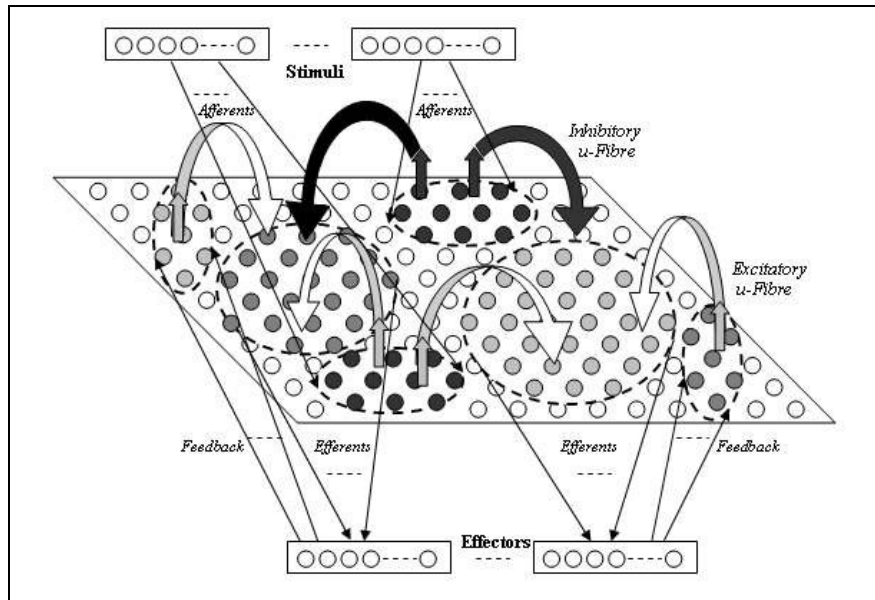


Figura 13. Modelo básico de uma rede de Venn

Capítulo 3

Modelagem Computacional da Doença de Alzheimer

Este capítulo descreverá de que forma os conceitos estudados (contidos na fundamentação teórica) foram utilizados para a modelagem da doença de Alzheimer. A idéia será apresentada após uma breve discussão acerca de modelos computacionais, seguida de detalhes da implementação.

3.1 Modelos computacionais

O trabalho aqui descrito trata principalmente da criação de um modelo computacional que represente a atuação de uma disfunção neurológica num sistema neural sadio. Através da utilização de redes neurais artificiais e técnicas para seu aprendizado, foi desenvolvida uma modelagem artificial da doença de Alzheimer, levando em consideração as características previamente apresentadas das redes de Venn.

É importante destacar a importância de se utilizar modelos para representar o “mundo real”. De forma mais específica, o problema de entender funções cerebrais através de métodos experimentais pode tornar-se mais fácil quando da utilização de modelos computacionais de mais baixo nível. Buarque [4] listou em sua tese de doutorado algumas das vantagens da utilização de modelos para obter sucesso na tarefa de representar dados e funções reais. Nesse aspecto, modelos:

- Têm a habilidade de adquirir propriedades do sistema modelado;
- Ajuda a abstrair detalhes do problema;

- Pode ser utilizado para checar a corretude de uma teoria.

Como a modelagem apresentada neste trabalho é uma modelagem computacional, algumas outras vantagens da utilização de modelos nesse sentido podem ser mostradas:

- O experimentador tem maior controle sobre o experimento, especificamente na simulação e na granularidade. A granularidade diz respeito ao nível de abstração na definição de um modelo;
- Há a possibilidade de utilizar dados artificiais e dados reais;
- Há a habilidade de escalonar sistemas complexos sem as dificuldades dos modelos matemáticos.

Vale ressaltar que modelos, computacionais ou não, possuem a inerente desvantagem de ter que utilizar abstrações. Essas abstrações, em última instância, podem resultar em imprecisões do sistema.

O trabalho de conclusão de curso apresentado nessa monografia utiliza um modelo computacional e sua simulação para demonstrar uma teoria. Assume-se a hipótese de que a atuação da doença de Alzheimer no cérebro humano pode ser refletida num modelo computacional que utilize redes neurais artificiais. Levando-se em consideração certas limitações e abstrações em relação ao modelo real, vários experimentos foram realizados, alguns mostrados mais tarde. Os dados utilizados foram artificiais, mas é plausível a possibilidade de extrapolação para o uso de dados reais. Além do mais, o trabalho foi conduzido de forma a favorecer o escalonamento do sistema em direção a sua calibração com processo biológicos.

3.2 Modelagem da Doença de Alzheimer

De forma geral, a modelagem computacional da Doença de Alzheimer desenvolvida nesse trabalho utiliza Redes Neurais Artificiais. Mais especificamente, foi construído um sistema computacional que, dada uma estrutura baseada nas redes de Venn, treina essa mesma rede com o algoritmo de treinamento do modelo de Hopfield. Posteriormente, foi modelada a degradação dessa rede, simulando assim a atuação da DA em um sistema biológico sadio.

A motivação para a utilização das redes de Venn advém do fato de que essa arquitetura é principalmente baseada em modelos naturais modulares do cérebro. Numa modelagem computacional de uma disfunção neurológica, essa característica é bastante importante. Na modelagem em questão, o conceito principal utilizado foi a idéia de regiões de processamento,

juntamente com a heterogeneidade das unidades de processamento. Dessa forma, foi possível simular um dos comportamentos da Doença de Alzheimer: a patologia pode não atuar no sistema natural como um todo, mas sim de forma específica em uma região do cérebro ou mesmo em um tipo de neurônio.

Por outro lado, o modelo de Hopfield possui uma característica que está intimamente relacionada com a atuação da Doença de Alzheimer. As redes de Hopfield podem agir como uma rede memória-associativa. É sabido que em certos estágios da DA, o paciente apresenta sinais claros de perda de memória [7]. Considerando que seu cérebro sadio seja um sistema memória-associativo que apresente bons resultados, a atuação da DA prejudica substancialmente essa dinâmica. Dessa forma, a representação do cérebro levando em consideração a topologia das redes de Venn, inspirada em sistemas naturais, juntamente com a inserção da capacidade de associar uma entrada (de maneira genérica) a uma saída compatível com as características daquela, pode simular um paciente sadio. Adicionalmente, a inclusão nesse sistema de um módulo que represente a atuação da Doença de Alzheimer baseando-se na forma como ela prejudica o sistema natural modela computacionalmente essa disfunção neurológica.

Para simular a Doença de Alzheimer, os aspectos neurológicos apresentados no capítulo anterior foram adaptados ao domínio das redes de Venn com algoritmo de treinamento do modelo de Hopfield. São eles:

- Perda de neurônios;
- Atrofia neuronal;
- Perda sináptica.

Essa última característica, a perda de sinapses (ou conexões), é baseada na atuação dos '*neurofibrillary tangles*' e placas neuríticas. É observado experimentalmente que, pela presença das placas e '*tangles*', a densidade sináptica pode diminuir com o progresso da doença [29]. A figura 14 mostra uma visão esquemática da modelagem computacional da Doença de Alzheimer desenvolvida neste trabalho.

Na literatura, existem diversas modelagens computacionais da Doença de Alzheimer [3, 30], inclusive utilizando Redes Neurais Artificiais [31], [32]. Uma das principais contribuições do trabalho apresentado nesta monografia é que a modelagem da DA foi desenvolvida utilizando-se as redes de Venn, nunca antes utilizada para simular essa disfunção neurológica. Outro ponto que merece destaque é na forma como as redes de Venn foram inicialmente propostas, seu treinamento era realizado por um algoritmo específico. A utilização do algoritmo de treinamento do modelo de Hopfield numa topologia das redes de Venn é original. Também vale lembrar que

esse trabalho pode significar a validação da utilização das redes de Venn em cenários patológicos diferentes do originalmente proposto.

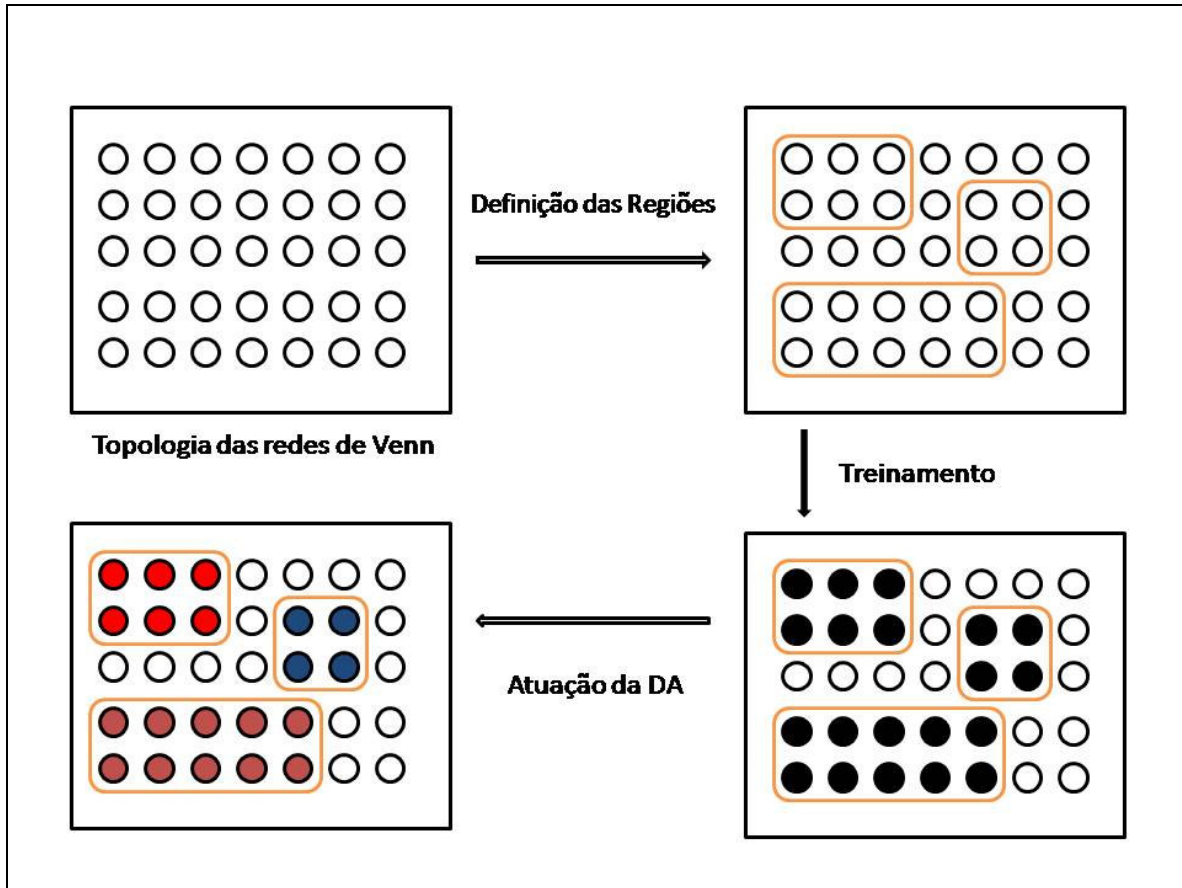


Figura 14. Visão esquemática da modelagem computacional da Doença de Alzheimer usando a topologia das redes de Venn.

As próximas seções apresentam detalhes de como a modelagem computacional da Doença de Alzheimer no domínio apresentado acima foi implementada, bem como a descrição das ferramentas e técnicas utilizadas.

3.3 GVNS (Generalised Venn-Networks Simulator)

Além de desenvolver uma nova arquitetura de redes neurais artificiais, as redes de Venn, Buarque desenvolveu um simulador computacional que implementa essa nova arquitetura em cenários fisiológicos e patológicos, o GVNS (Generalised Venn-Networks Simulator) [4]. O GVNS oferece, dentre outras vantagens, a possibilidade de topologias complexas devido à multiplicidade das unidades de processamento, regiões de processamento, tipos de fibras e fontes de entrada e saída. Também faz o cálculo interno de estatísticas de saída (simplificando a análise

dos dados) e tem a habilidade de produzir “*snapshots*” que se assemelham a imagens funcionais do cérebro. A figura 15 mostra os casos de uso originais do GVNS.

Por outro lado, o conceito de programação orientada a objeto mostra que o desenvolvimento de um aplicativo que segue proposições de uma engenharia de software bem trabalhada favorece vantagens, algumas das quais:

- Reuso de código;
- Manutenção do aplicativo;
- Modularidade.

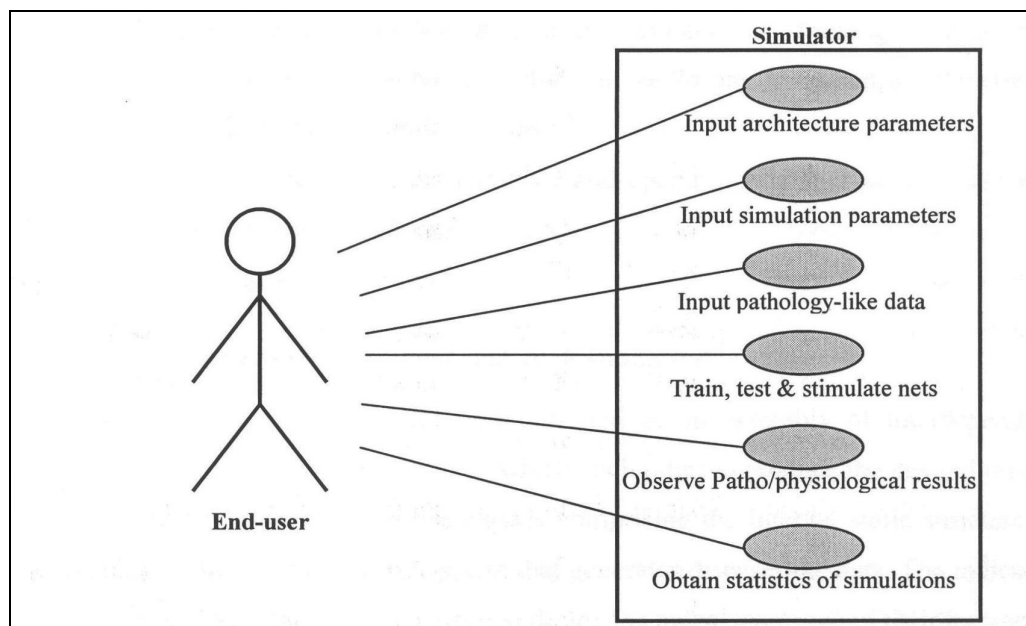


Figura 15. Casos de uso iniciais do GVNS

Nesse sentido, uma nova versão do simulador de redes de Venn deveria ser construída para melhor representar a arquitetura das redes de Venn. Analisando-se as diversas etapas UML (Unified Modelling Language) [33], como o diagrama de casos de uso (Figura 15), o diagrama de classes (Figura 16) e o diagrama de seqüência (Figura 17) da versão inicial do simulador, ficou claro que a utilização mais eficaz de programação orientada a objetos poderia tornar o simulador mais reutilizável e adaptável, resultante de uma boa modularização.

O GVNS foi originalmente utilizado para simular o comportamento da esclerose múltipla e outras doenças tipicamente senis. Daí vem a inspiração para a realização de uma modelagem computacional da doença de Alzheimer utilizando as redes de Venn juntamente com uma nova versão do simulador.

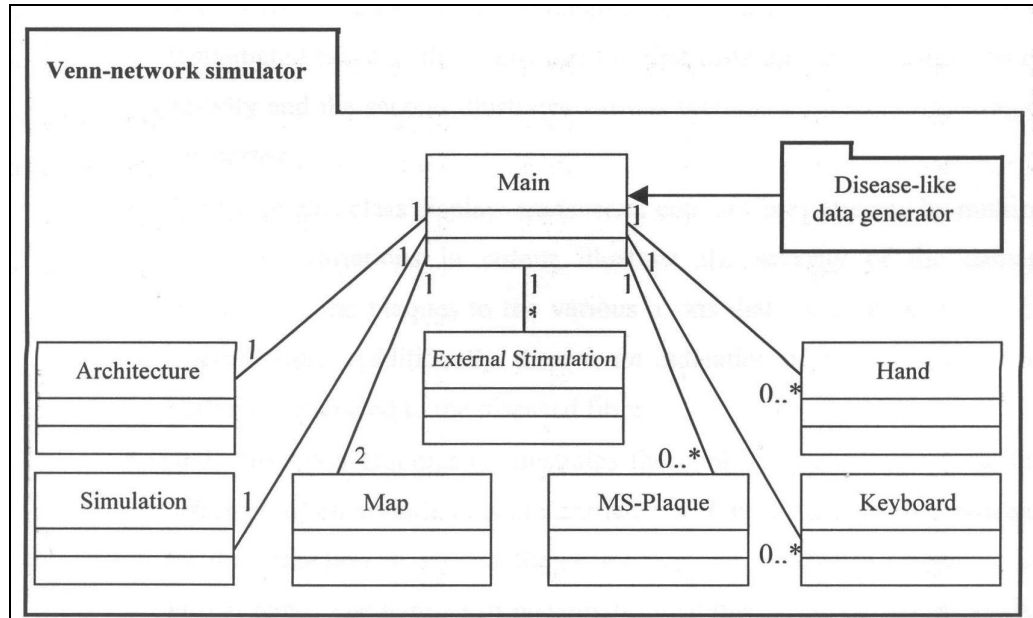


Figura 16. Diagrama de classes do GVNS

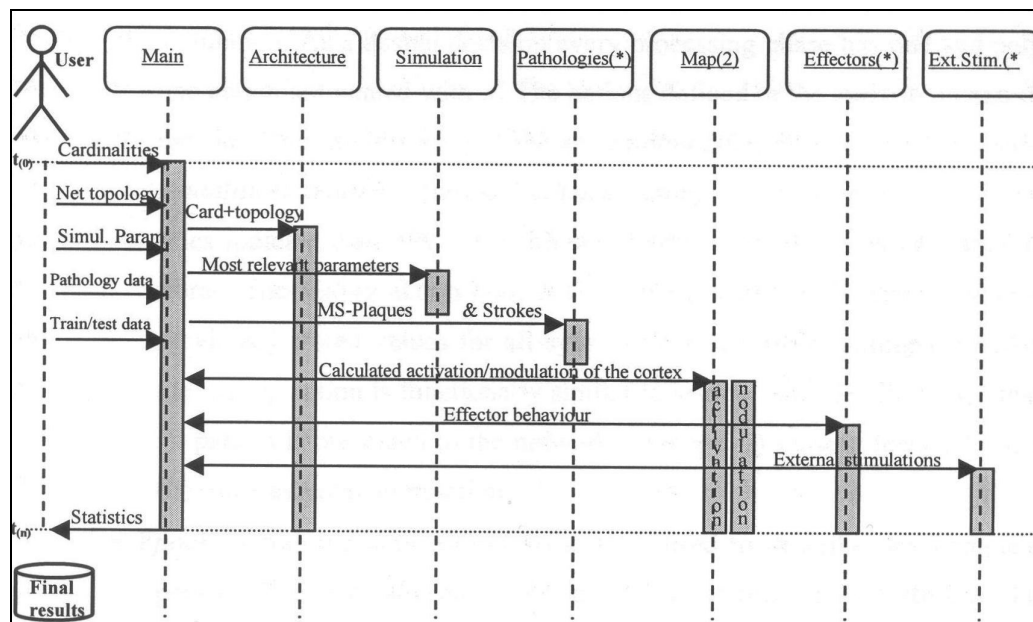


Figura 17. Diagrama de seqüência do GVNS

3.4 GVNS2

Levando-se em consideração as características do GVNS supracitadas, foi proposto um protótipo de uma nova arquitetura para que o simulador incorpore satisfatoriamente os conceitos de orientação a objetos. Foi utilizada para programação a ferramenta Eclipse [34], em um computador eletrônico digital com sistema operacional Windows XP®.

A primeira versão do simulador foi escrito na linguagem de programação Java, possuindo oito principais classes, a saber:

- `Main.java`: controla a maioria das funções do simulador, tais como, algoritmos de treinamento, processamento de patologias, etc., além de ser responsável pelos cálculos dos erros e do relacionamento com o usuário.
- `Architecture.java`: mostra graficamente a topologia das redes simuladas.
- `Simulation.java`: mostra a lista de parâmetros selecionados para as simulações.
- `Map.java`: mostra snapshots através da variação de cor as unidades que compõem o mapa cortical.
- `MS-plaque.java`: mostra cortes transversais das fibras afetadas pela esclerose múltipla.
- `Hand.java`: ilustra graficamente o comportamento de uma mão nas simulações.
- `Keyboard.java`: também ilustra graficamente o comportamento das simulações. São indicados os movimentos laterais das mãos.
- `Stimulation.java`: executa diversos estímulos externos nas redes simuladas.

Dada a relativa complexidade de um simulador de redes neurais artificiais, percebe-se que esse número reduzido de classes impossibilita uma aplicação eficaz de programação orientada a objetos. Soma-se a isso o fato de o armazenamento ser feito através de alocação estática em vetores, diminuindo a capacidade de manipulação e flexibilidade dos dados.

Desse modo, foi proposta uma nova arquitetura baseada em orientação a objetos que deve servir como base para o desenvolvimento completo da segunda versão do simulador de redes de Venn.

Foram propostos inicialmente os pacotes que seguem. A utilização de pacotes facilita a modularidade do simulador e aumenta a segurança dos dados.

- `activationsFunctions`: destinado às classes que tratam das funções de ativação de uma unidade de processamento. Aqui há uma classe abstrata definindo as diretrizes para implementação de qualquer função de ativação. Até o momento, foram implementadas a tangente hiperbólica e a função logística.

- `unitTypes`: destinado às classes que implementam as unidades de processamento. A classe principal desse pacote implementa o modelo mais simples de uma unidade de processamento. Outros tipos dos assim chamados neurônios poderão ser adicionados a partir da extensão dessa classe.
- `fibres`: destinado às classes que implementam as diversas fibras do modelo de redes de Venn. Há também aqui a figura de uma classe abstrata, `Fibre.java`. Os demais tipos de fibras, aferentes, eferentes, eferentes *feedback* e *u-fibres*, herdam dessa classe.
- `regions`: destinado às classes que tratam das regiões do modelo de Venn, bem como os mapas (compostos por regiões) e a origem e o alvo do processamento.

Foram programados os conceitos de herança em OO na construção das classes que tratam das fibras e das unidades de processamento. Essa arquitetura é interessante porque há inicialmente diversas fibras que compartilham características. Além disso, também poderão ser adicionados diversos tipos de unidades de processamento, ao gosto do domínio do problema, sem comprometer a arquitetura proposta e desse modo facilitando o reuso de código e a adaptabilidade. A implementação do conceito de herança nos tipos de neurônios e nas fibras está explicitada nas figuras 18 e 19, respectivamente.

Um ponto a destacar é o fato de que toda a interface gráfica ter sido construída independentemente do “interior” do sistema, provendo assim modularidade e facilidade na alteração das camadas de interação com o usuário.

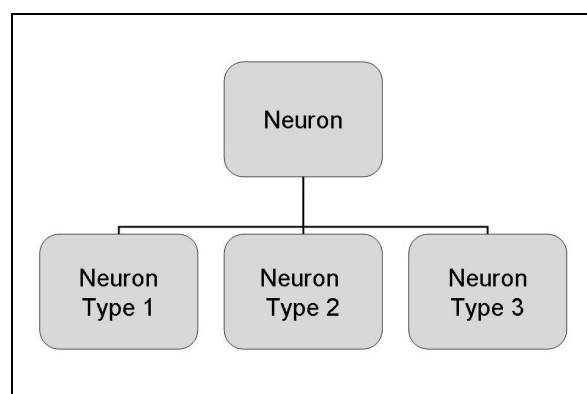


Figura 18. Herança na implementação de unidades de processamento. “Neuron” representa a classe-mãe.

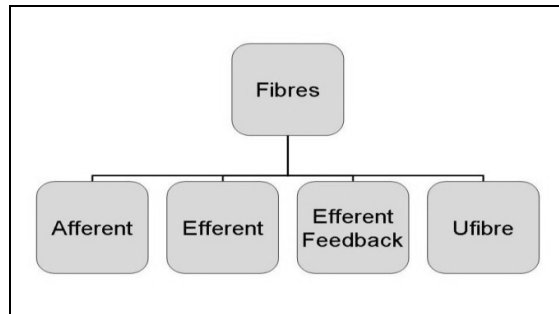


Figura 19. Herança na implementação das fibras. “*Fibres*” representa a classe-mãe.

Há também a utilização do conceito de interfaces para, por exemplo, abstrair a forma como os dados são armazenados. Nesse caso, as camadas “superiores” do simulador não deverão tratar dos dados que estão sendo armazenados (internamente). Adicionalmente, o uso de vetores é feito através de alocação dinâmica de dados.

3.5 Módulo para modelagem computacional da Doença de Alzheimer

Tendo como ferramenta-base o GVNS2, foi desenvolvido um módulo para a modelagem computacional da Doença de Alzheimer e a realização das simulações. Como explicitado anteriormente, a principal característica da arquitetura desenvolvida é a junção dos conceitos das redes de Venn juntamente com o modelo de Hopfield. De forma geral, um algoritmo de treinamento baseado no modelo de Hopfield foi aplicado numa estrutura de redes de Venn.

Primeiramente, as unidades de processamento (*i.e.* os neurônios artificiais, ou simplesmente neurônios) são selecionadas e devidamente separadas em regiões. Os neurônios selecionados são postos em uma estrutura de dados, cada um com seu rótulo que o identifica a uma determinada região. A figura 20 apresenta a estrutura da classe Java representativa de um neurônio artificial.

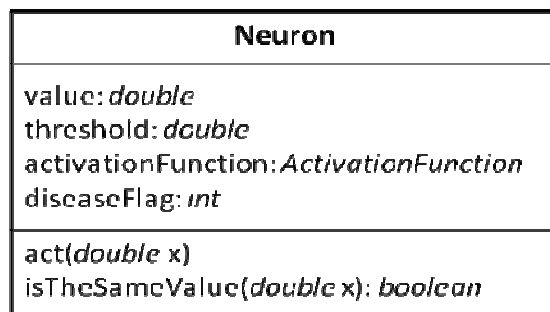


Figura 20. Classe Neuron.

Como visto, a classe *Neuron* possui atributos que definem suas características, como o seu valor de saída (*value*) e o limiar de excitação (*threshold*). No atributo *activationFunction* é definido a função de ativação da unidade de processamento. No sistema desenvolvido, foram implementadas duas funções de ativação com saídas contínuas, a função logística (equação 7) e função tangente hiperbólica (equação 8). Os métodos *act* e *isTheSameValue* calculam a saída do neurônio e um passo intermediário do algoritmo de treinamento, respectivamente.

$$F(t) = \frac{1}{1 + e^{-t}} \quad (\text{Eq. 7})$$

$$F(t) = \frac{e^t + e^{-t}}{e^t - e^{-t}} \quad (\text{Eq. 8})$$

A unidade de processamento é do tipo MCP, isto é, admite valores discretos (-1 ou 1). O cálculo da saída do neurônio implementado pelo método *act* é mostrado na figura 21, na linguagem de programação Java:

```
public void act(double x) {
    double y = 0;
    y = this.activationFunction.active(x) - this.threshold;
    if (y >= 0)
        this.value = 1;
    else
        this.value = -1;
}
```

Figura 21. Método *act* da classe *Neuron*.

Hopfield é a classe que implementa o algoritmo de treinamento do modelo de Hopfield. A figura 22 mostra seus atributos e métodos.

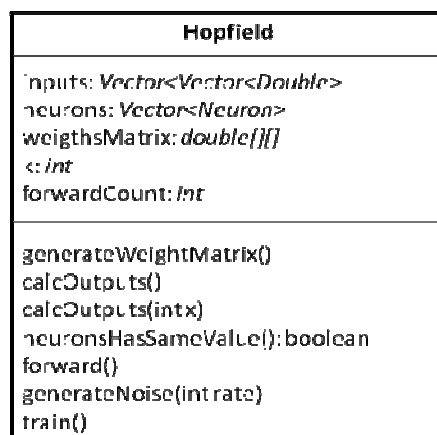


Figura 22. Classe *Hopfield*.

O atributo *inputs* armazena os padrões de treinamento, enquanto que o atributo *neuron* contém os objetos das unidades de processamento. O atributo *weightsMatrix* contém a matriz de pesos do algoritmo de treinamento. A matriz de pesos é calculada pelo método *generateWeightsMatrix()*, na medida em que a saída dos neurônios é calculada pelo método *calcOutputs()*. Há uma sobrecarga nesse caso. Se esse mesmo método receber um inteiro como parâmetro, é calculada apenas a saída para a unidade de processamento especificada. O método *train()* treina a rede, através da geração da matriz de pesos de acordo com os padrões de treinamento, e *forward()* calcula a resposta da rede levando-se em consideração o treinamento e um padrão apresentado. Um dos métodos mais importantes, para geração da matriz de pesos, é mostrado na figura 23.

```

public void generateWeightsMatrix(){
    for (int i = 0; i < this.inputs.elementAt(0).size(); i++){
        for (int j = 0; j < this.inputs.elementAt(0).size(); j++){
            this.weightsMatrix[i][j] = 0;
            for (int k = 0; k < this.inputs.size(); k++){
                this.weightsMatrix[i][j] +=
                    this.inputs.elementAt(k).elementAt(i).doubleValue() *
                    this.inputs.elementAt(k).elementAt(j).doubleValue();
            }
            this.weightsMatrix[i][j] = this.weightsMatrix[i][j] / this.k;
        }
    }
    for (int i = 0; i < this.inputs.elementAt(0).size(); i++){
        this.weightsMatrix[i][i] = 0;
    }
}

```

Figura 23. Método *generateWeightsMatrix()*.

Para a simulação da Doença de Alzheimer, a classe *Alzheimer* foi desenvolvida (figura 24). O atributo *hopfield* recebe um objeto da classe *Hopfield*, contendo uma rede treinada com determinados padrões de treinamento.

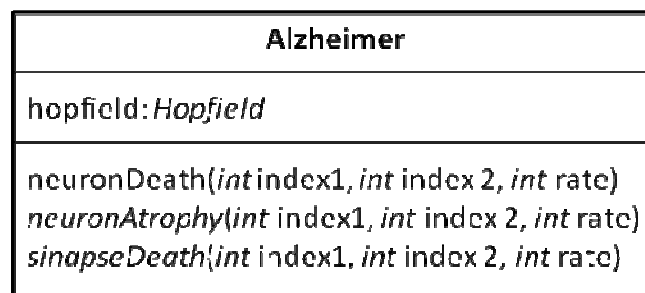


Figura 24. Classe *Alzheimer*.

Os métodos da classe Alzheimer representam cada um dos aspectos neuropatológicos da Doença de Alzheimer que foram selecionados para a modelagem. Os parâmetros para cada um desses métodos são os mesmos. Os inteiros *index1* e *index2* indicam qual região a simulação da doença vai atuar. Esse é o ponto-chave entre a utilização do algoritmo de treinamento do modelo de *Hopfield* na arquitetura das redes de Venn. Tendo-se num objeto da classe *Hopfield* uma estrutura de dados do tipo matriz de uma dimensão (*neurons*), as regiões são definidas de acordo com o índice dessa estrutura de dados.

Por exemplo, pode-se ter o caso de haver três regiões numa matriz de unidades de processamento de uma dimensão com tamanho 10. A primeira região pode englobar os neurônios de índice 0 a 4, a segunda região pode conter os neurônios de índice 5 a 7, e a terceira região pode englobar os neurônios restantes. Dessa forma, selecionando-se os índices nos métodos da simulação dos aspectos neuropatológicos da DA é possível indicar em qual região haverá aquela atuação. Por sua vez, o atributo *rate* indica a possibilidade daquela característica realmente atuar. É uma probabilidade, seus valores vão de 0 a 100 representando uma porcentagem.

As figuras 25, 26 e 27 mostram os métodos que simulam a perda de neurônios, a atrofia neuronal e a perda de sinapses, respectivamente.

O método *neuronDeath* altera o atributo *diseaseFlag* de cada unidade de processamento afetada. Através da modificação desse atributo, sua saída é anulada no método *forward()* da classe *Hopfield*. Para modelar a atrofia neuronal, o método *neuronAtrophy* aumenta o limiar de excitação do neurônios, diminuindo assim sua sensibilidade a determinados padrões de entrada. Já o método *synapseDeath* anula as conexões contidas na matriz de pesos.

```
public void neuronDeath(int index1, int index2, int rate){
    int min = 0;
    int max = 100;
    int random;
    for (int i = index1; i <= index2; i++){
        random = min + (int)((max - min) * Math.random());
        if (random <= rate){
            this.hopfield.getNeurons().elementAt(i).setDiseaseFlag(1);
        }
    }
}
```

Figura 25. Método que modela a perda de neurônios.

```
public void neuronAtrophy(int index1, int index2, int rate){
    int min = 0;
    int max = 100;
    int random;
    double threshold = 0;
    for (int i = index1; i <= index2; i++){
        random = min + (int)((max - min) * Math.random());
        if (random <= rate){
            threshold = this.hopfield.getNeurons().elementAt(i).getThreshold();
            this.hopfield.getNeurons().elementAt(i).setThreshold(threshold+1);
        }
    }
}
```

Figura 26. Método que modela a atrofia neuronal.

```
public void sinapseDeath(int index1, int index2, int rate){
    int min = 0;
    int max = 100;
    int random;
    for (int i = index1; i <= index2; i++){
        for (int j = 0; j < this.hopfield.getNeurons().size(); j++){
            random = min + (int)((max - min) * Math.random());
            if (random <= rate){
                this.hopfield.getWeightsMatrix()[i][j] = 0;
            }
        }
    }
}
```

Figura 27. Método que modela a perda de sinapses.

Capítulo 4

Resultados

Neste capítulo, serão apresentadas as simulações realizadas para a modelagem computacional da Doença de Alzheimer. O experimento computacional realizado ilustra o comportamento de uma rede de Venn que utiliza o modelo de Hopfield como algoritmo de treinamento. A rede é composta por cem neurônios, dispostos numa matriz 10x10. A idéia é que a rede neural seja treinada para reconhecer essas matrizes, tidas como imagens. Nessa matriz, valores +1 representam um pixel preto, enquanto que valores -1 representam um pixel branco. Assim, tem-se a hipótese de que a atuação da modelagem computacional da DA prejudique os resultados alcançados pela rede. A figura 28 mostra as imagens, representadas pelas matrizes 10x10, que foram utilizadas como padrões de treinamento.

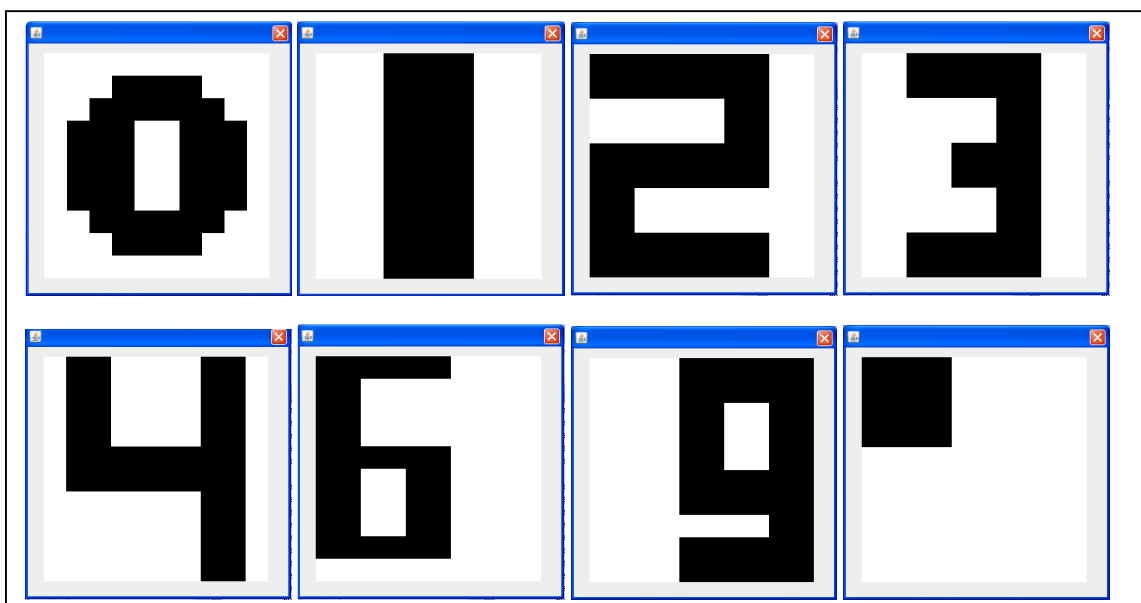




Figura 28. Padrões de treinamento do experimento computacional.

Com um ruído de 20%, isto é, uma probabilidade de 20% para que cada pixel altere o seu valor, a rede neural apresenta bons resultados. A tabela 1 mostra o número médio de iterações necessárias para recuperar os padrões apresentados à rede treinada.

Tabela 1. Comportamento da rede neural treinada para recuperação de padrões.

Padrão	Ruído (em %)	Número médio de iterações necessárias para recuperar padrão
0	20	445
1	20	288
2	20	343
3	20	364
4	20	350
6	10	382
9	10	353
 “  ”	5	290

A seguir, alguns experimentos com a modelagem computacional da Doença de Alzheimer serão apresentados. Primeiramente, as simulações serão mostradas para cada um dos aspectos neuropatológicos, e, em seguida, estão simulações que mesclam todas as características da disfunção neurológica. Para todas elas, a rede foi definida em um determinado número de regiões. Cada uma dessas regiões varia de acordo com o tipo de unidade de processamento. A tabela 2 mostra um exemplo da topologia utilizada.

Tabela 2. Exemplo da topologia utilizada nas simulações.

Região	Quantidade de Neurônios	Função de Ativação	Threshold
Região 1	30	Logística	0.1
Região 2	35	Tangente Hiperbólica	0.2
Região 3	35	Logística	0.25

No caso dessas simulações, as unidades de processamento diferem entre si pela função de ativação utilizada e pelo limiar de excitação (*threshold*). Para cada experimento, à rede neural é

apresentado um dos padrões de treinamento com determinado ruído. Interessante notar que esse ruído é também variável, sendo mais um parâmetro de configuração para as simulações. Com ruídos maiores, a rede apresenta uma maior dificuldade para reconhecer os padrões de treinamento.

4.1 Perda de neurônios

Nesse caso, a perda neuronal foi simulada. A tabela 2 mostra os parâmetros do experimento 1 e a tabela 3 faz o mesmo para o experimento 2. As figuras 29 e 30 mostram os resultados obtidos para esses dois experimentos computacionais.

Tabela 3. Experimento 1 – Perda de Neurônios (ruído de 20%)

Região	Quantidade de Neurônios	Função de Ativação	Threshold	Atuação da DA	Probabilidade (%)
Região 1	30	Logística	0.1	Perda de Neurônios	75
Região 2	35	Tangente Hiperbólica	0.2	-	-
Região 3	35	Logística	0.25	-	-

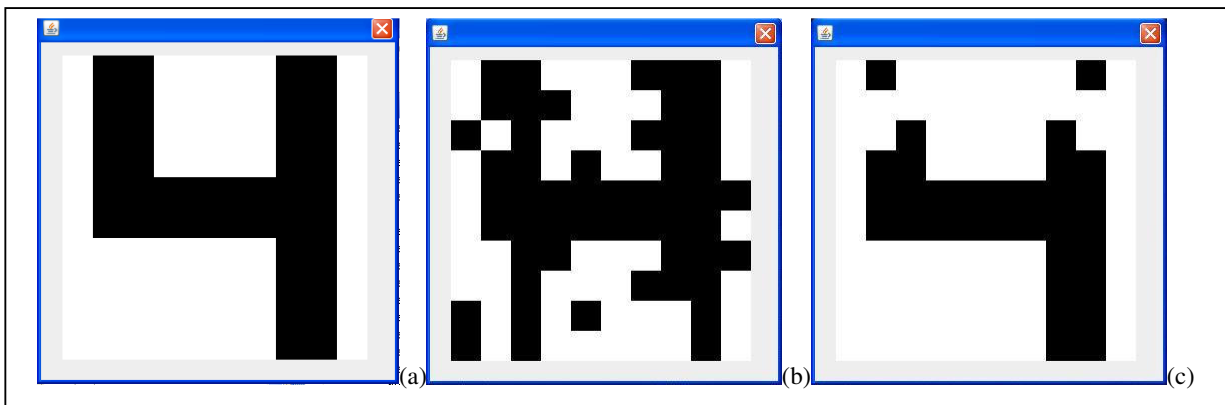


Figura 29. Exp. 1 – Padrão de treinamento (a) com 20% de ruído (b) e resposta da rede (c).

Tabela 4. Experimento 2 – Perda de Neurônios (ruído de 30%)

Região	Quantidade de Neurônios	Função de Ativação	Threshold	Atuação da DA	Probabilidade (%)
Região 1	30	Logística	0.1	Perda de Neurônios	50
Região 2	20	Logística	0.2	-	-
Região 3	20	Tangente Hiperbólica	0.2	Perda de Neurônios	50
Região 4	30	Tangente Hiperbólica	0.1	-	-

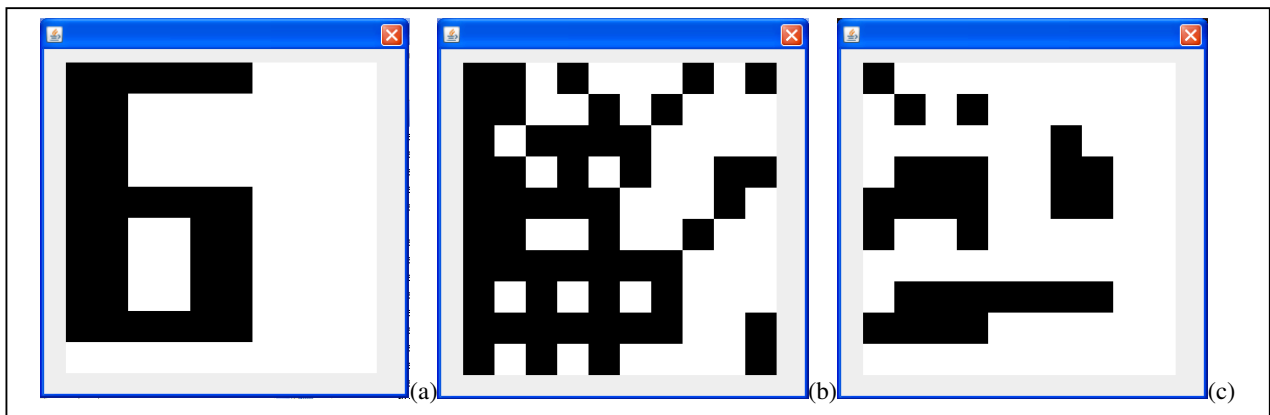


Figura 30. Exp. 2 – Padrão de treinamento (a) com 30% de ruído (b) e resposta da rede (c).

4.2 Atrofia neuronal

Como visto anteriormente, o aumento do limiar de excitação das unidades de processamento afetadas simulam o comportamento da atrofia neuronal. As tabelas 5 e 6 mostram os parâmetros dos experimentos computacionais envolvidos com esse aspecto, enquanto que as figura 31 e 32 apresentam os resultados das simulações.

Tabela 5. Experimento 3 – Atrofia neuronal (ruído de 20%)

Região	Quantidade de Neurônios	Função de Ativação	Threshold	Atuação da DA	Probabilidade (%)
Região 1	30	Logística	0.1	-	-
Região 2	35	Tangente Hiperbólica	0.2	Atrofia Neuronal	75
Região 3	35	Logística	0.25	-	-

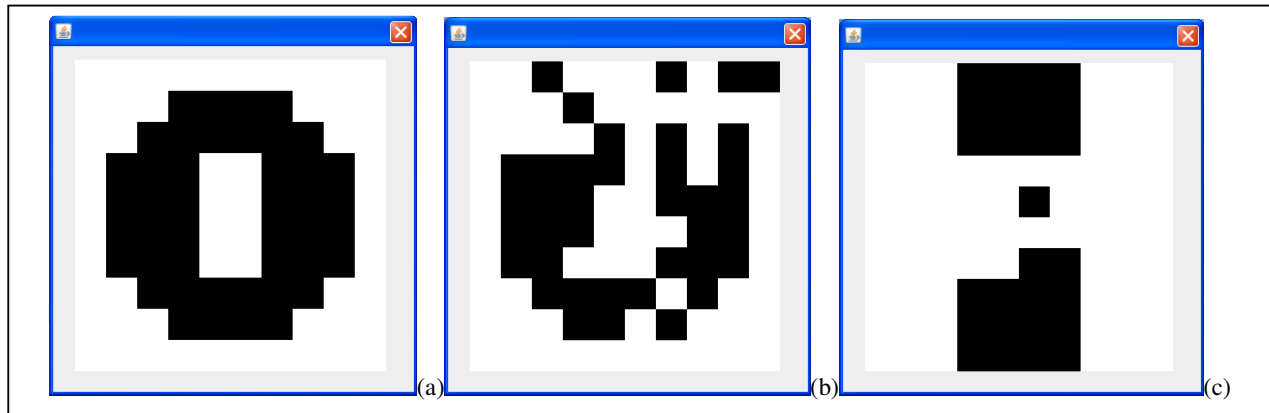


Figura 31. Exp. 3 – Padrão de treinamento (a) com 20% de ruído (b) e resposta da rede (c).

Tabela 6. Experimento 4 – Atrofia neuronal (ruído de 30%)

Região	Quantidade de Neurônios	Função de Ativação	Threshold	Atuação da DA	Probabilidade (%)
Região 1	20	Logística	0.1	-	-
Região 2	30	Tangente Hiperbólica	0.2	Atrofia Neuronal	75
Região 3	20	Logística	0.25	-	-
Região 4	30	Tangente Hiperbólica	0.1	Atrofia Neuronal	75

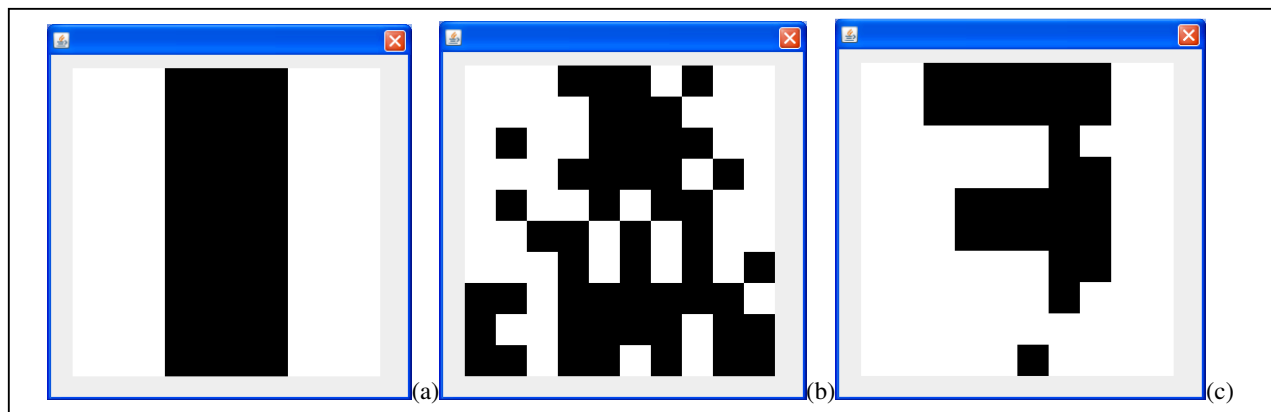


Figura 32. Exp. 4 – Padrão de treinamento (a) com 30% de ruído (b) e resposta da rede (c).

4.3 Perda sináptica

Os experimentos dessa seção simulam o comportamento da perda sináptica, ou seja, a anulação de algumas conexões entre os neurônios. As tabelas 6 e 7 apresentam esses experimentos computacionais, na medida em que as figuras 33 e 34 mostram as simulações.

Tabela 7. Experimento 5 – Perda sináptica (ruído de 20%)

Região	Quantidade de Neurônios	Função de Ativação	Threshold	Atuação da DA	Probabilidade (%)
Região 1	30	Logística	0.1	-	-
Região 2	35	Tangente Hiperbólica	0.2	Perda Sináptica	75
Região 3	35	Logística	0.25	-	-

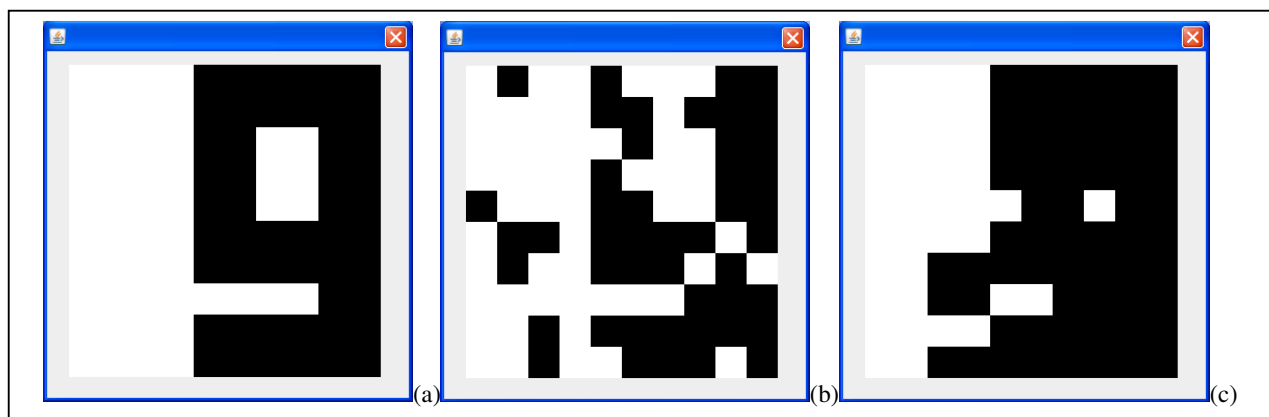


Figura 33. Exp. 5 – Padrão de treinamento (a) com 20% de ruído (b) e resposta da rede (c).

Tabela 8. Experimento 6 – Perda sináptica (ruído de 30%)

Região	Quantidade de Neurônios	Função de Ativação	Threshold	Atuação da DA	Probabilidade (%)
Região 1	20	Logística	0.1	Perda Sináptica	50
Região 2	30	Tangente Hiperbólica	0.2	-	-
Região 3	20	Logística	0.25	-	-
Região 4	30	Tangente Hiperbólica	0.1	Perda Sináptica	50

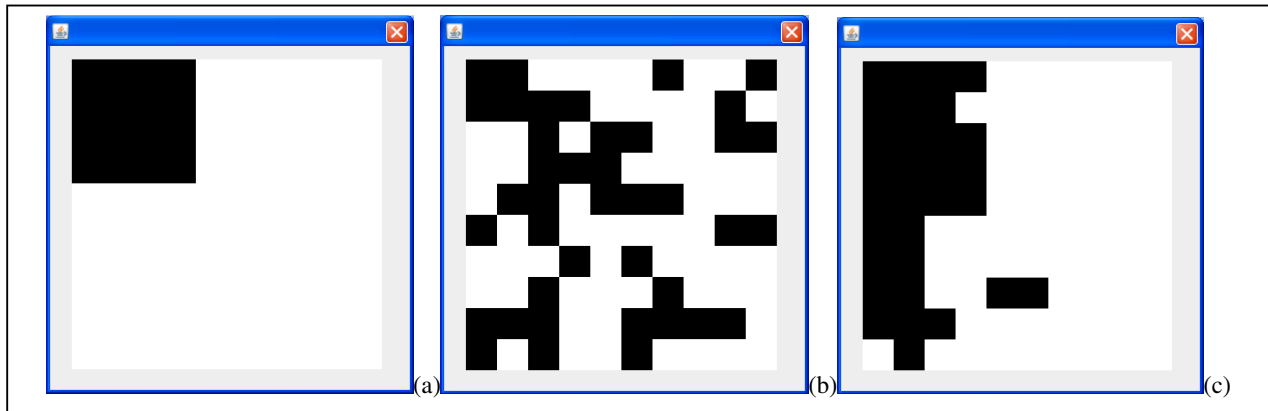


Figura 34. Exp. 6 – Padrão de treinamento (a) com 30% de ruído (b) e resposta da rede (c).

4.4 Experimentos mesclados

Na Doença de Alzheimer, os aspectos neuropatológicos podem atuar no paciente não de maneira pontual, mas sim agindo sob diversas frentes, prejudicando o comportamento do cérebro de diferentes formas. Nesse sentido, os experimentos computacionais nessa seção refletem esse caso. Em cada simulação, cada um dos aspectos modelados foram aplicados juntamente com os outros. As tabelas 8 e 9 mostram os parâmetros desses experimentos e as figuras 35 e 36 apresentam os resultados das simulações.

Tabela 9. Experimento 7 – Experimento mesclado (ruído de 20%)

Região	Quantidade de Neurônios	Função de Ativação	Threshold	Atuação da DA	Probabilidade (%)
Região 1	15	Logística	0.1	Perda de Neurônios	50
Região 2	15	Tangente Hiperbólica	0.2	-	-
Região 3	20	Logística	0.1	Atrofia Neuronal	50
Região 4	20	Tangente Hiperbólica	0.2	-	-
Região 5	15	Logística	0.1	Perda Sináptica	50
Região 6	15	Tangente Hiperbólica	0.2	-	-

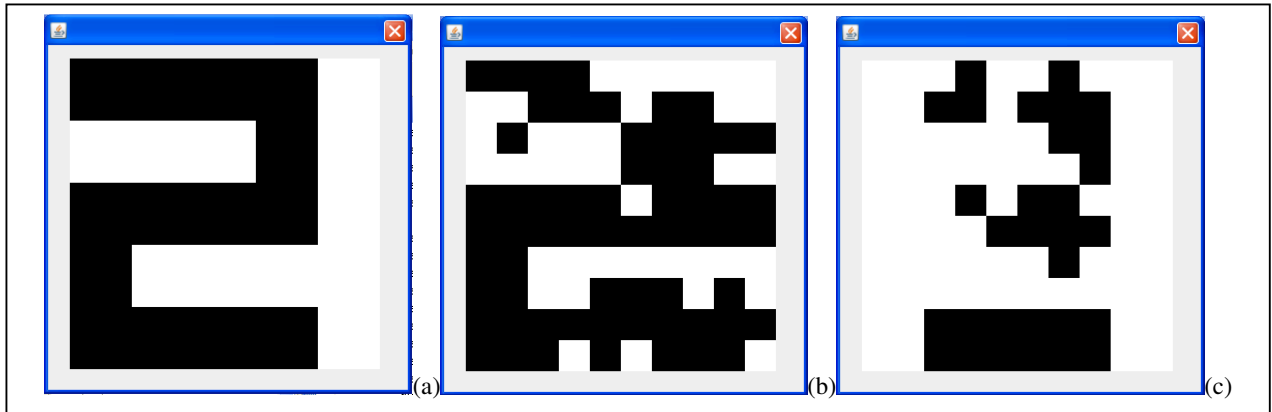


Figura 35. Exp. 7 – Padrão de treinamento (a) com 20% de ruído (b) e resposta da rede (c).

Tabela 10. Experimento 8 – Experimento mesclado (ruído de 20%)

Região	Quantidade de Neurônios	Função de Ativação	Threshold	Atuação da DA	Probabilidade (%)
Região 1	30	Logística	0.1	Atrofia Neuronal	75
Região 2	10	Tangente Hiperbólica	0.2	-	-
Região 3	15	Logística	0.1	Perda de Neurônios	75
Região 4	10	Tangente Hiperbólica	0.2	-	-
Região 5	30	Logística	0.1	Perda Sináptica	75
Região 6	5	Tangente Hiperbólica	0.2	-	-

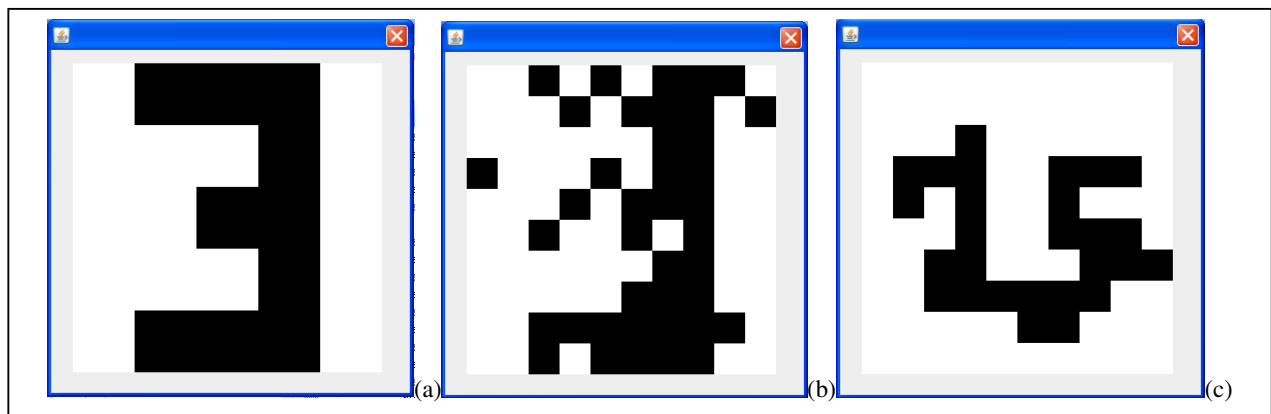


Figura 36. Exp. 8 – Padrão de treinamento (a) com 20% de ruído (b) e resposta da rede (c).

Capítulo 5

Conclusão

5.1 Discussão

O trabalho apresentado nesta monografia buscou realizar uma modelagem computacional da Doença de Alzheimer, que é uma disfunção de ordem neurológica responsável por mortes e diminuição na qualidade de vida de milhares de pessoas ao redor do mundo. Para alcançar esse objetivo, uma técnica de Inteligência Artificial foi aplicada a um domínio específico, no caso, redes neurais artificiais em um âmbito de neurociência computacional. Uma RNA com arquitetura baseada nas redes de Venn foi primeiramente treinada com o algoritmo do modelo de Hopfield. Após essa estrutura apresentar bons resultados no que diz respeito ao reconhecimento de padrões (em específico nas simulações, imagens binárias), uma modelagem computacional da DA foi implementada e aplicada à rede. Essa modelagem foi construída com inspiração nos aspectos neuropatológicos da doença, no que diz respeito às diversas frentes de ataque da patologia na estrutura e funcionalidade do cérebro. Como previsto na hipótese inicial, a aplicação da modelagem computacional prejudicou o bom funcionamento da rede neural treinada, o que foi comprovado nas diversas simulações.

Para a modelagem, e conseqüentemente, simulações, foram selecionadas após estudo cuidadoso da bibliografia, três diferentes formas de atuação da Doença de Alzheimer, a saber:

- Perda de neurônios;
- Atrofia neuronal;
- Perda sináptica.

Como se pôde perceber através das simulações, variando-se os diversos parâmetros da ferramenta de simulação produzida, a rede não conseguiu obter respostas satisfatórias quando da apresentação de imagens contidas nos padrões de treinamento do sistema. Isso ocorreu tanto nas

simulações independentes, isto é, levando-se em consideração as características da DA em separado, quanto nas simulações que aplicaram as diversas frentes de atuação da doença. Pôde-se perceber que os experimentos que causaram mais impacto à performance da rede foram aqueles obtidos pelas simulações que contemplaram todos os aspectos neuropatológicos modelados, os experimentos mesclados. Entretanto, deve-se observar que, para cada um dos aspectos modelados, a tendência é que a performance da rede neural piore na medida em que os parâmetros das simulações reflitam condições mais desfavoráveis. Essas condições ocorrem, por exemplo, com um ruído maior e com mais regiões afetadas pela doença.

Uma característica interessante, observada nas simulações, é explicitada nas figuras 31 e 32. Quando o aspecto atrofia neuronal foi aplicado à rede, o sistema tendeu a apresentar como resposta imagens semelhantes a outros padrões de treinamento, diferentes dos padrões de entrada. No experimento 3, a entrada foi a imagem representativa do número 0, e a rede apresentou como resposta uma imagem próxima à representativa do número 1. Por outro lado, no experimento 4, a rede apresentou como resposta uma imagem próxima à representativa do número 3 para uma entrada com a imagem do número 2.

De modo geral, foi possível fazer a associação de uma característica importante da Doença de Alzheimer, a perda de memória, com a modelagem computacional desenvolvida. Num sistema neural artificial de memória associativa, a aplicação da implementação da modelagem prejudicou essa característica apreendida com um treinamento. Em certo âmbito, esta é uma representação da atuação da DA no sistema neural biológico.

Outro ponto a destacar é a validação da utilização das redes de Venn em cenários patológicos, através da modelagem de uma doença neurológica nunca antes utilizada. Essa validação foi acompanhada da originalidade de se utilizar as redes de Venn com um algoritmo de treinamento de um modelo já conhecido na literatura, em específico o modelo de Hopfield. Essa faceta do trabalho serve como demonstração também da adaptabilidade das redes de Venn a diversos ambientes.

5.2 Trabalho futuros

A monografia aqui apresentada possui diversos desdobramentos possíveis. Tais desdobramentos têm como alvo generalizar e escalar os resultados atingidos. A seguir, alguns dos trabalhos sugeridos para complementar o desenvolvimento da modelagem computacional da Doença de Alzheimer.

- Na modelagem, a inspiração para a utilização das redes de Venn foi principalmente o conceito das unidades de processamento e das regiões. O modelo computacional pode ser extrapolado para contemplar outras características das redes de Venn, como por exemplo, a diferenciação das fibras. No caso deste trabalho, as conexões (representantes das fibras) foram de apenas um tipo, aproximando-se das *u-fibres* tipo 1;
- Aprofundar os estudos no que diz respeito aos aspectos neuropatológicos da Doença de Alzheimer. Assim, outras características dessa disfunção podem ser modeladas e simuladas;
- As simulações foram realizadas utilizando-se dados artificiais. A modelagem pode ser direcionada para contemplar a utilização de dados reais da doença, extraídos de outros trabalhos ou mesmo experimentalmente;
- A modelagem computacional da Doença de Alzheimer pode servir como inspiração para a modelagem de outras disfunções neurológicas. Um bom caminho para tanto é a modelagem da Doença de Parkinson, por exemplo.
- A modelagem computacional e as simulações foram implementadas em um módulo do GVNS2. Entretanto, esse módulo não está totalmente integrado com a interface gráfica do simulador. Após esse trabalho, o simulador poderá ser utilizado até mesmo por profissionais da área médica para realização de simulações com dados reais;
- O presente trabalho pode também contar com apoio de neurologistas para efetivamente iniciar estudos visando a calibração da ferramenta para casos reais da DA.

Referências Bibliográficas

- [1] <http://www.ncbi.nlm.nih.gov/sites/entrez>. Acessado em: 14 de Novembro de 2008, 10:40h.
- [2] Bergeron C. **Alzheimer's disease - neuropathological aspects**. Can J Vet Res. 1990 Jan;54(1):58-64. Review. PMID: 2407331 [PubMed - indexed for MEDLINE].
- [3] Duch W. **Therapeutic applications of computer models of brain activity for Alzheimer disease**. J. Medical Informatics and Technologies 5 (2000) 27-34.
- [4] Buarque de Lima Neto, F. **Modeling Neural Processing Using Venn-networks in Physiological and Pathological Scenarios**. (2002) PhD Thesis, Imperial College of Science, Technology and Medicine, Department of Electrical and Electronic Engineering. London, England.
- [5] Lima Neto, Fernando Buarque de; Wilde, Philippe de. **Modelling axonal delays caused by Multiple Sclerosis Plaques**. In: Workshop on Memory, Delays and Multistability, 2000, Montreal - Canada. Proc. of 2000 Workshop on Memory, Delays and Multistability.2000.
- [6] Alzheimer, A. **Ueber eine eigenartige Erkrankung der Hirnrinde**. Allgemeine Zeitschrift für Psychiatrie 1907; 64; 146 – 158.
- [7] Friedlander, Arthur A; et al. **Alzheimer's disease: Psychopathology, medical management and dental implications**. J Am Dent Assoc 2006; 137; 1240 – 1251.
- [8] Fitzpatrick AL, Kuller LH, Ives DG, et al. **Incidence and prevalence of dementia in the Cardiovascular Health Study**. J Am Geriatr Soc 2004; 52(2); 195 – 204.
- [9] Tang MX, Cross P, Andrews H, et al. **Incidence of AD in African-Americans, Caribbean Hispanics, and Caucasians in northern Manhattan**. Neurology 2001; 56(1):49 – 56.

- [10] Silverman JM, Ciresi G, Smith CJ, Marin DB, Schnaider-Beerli M. **Variability of familial risk of Alzheimer disease across the late life span.** Arch Gen Psychiatry 2005; 62(5):565 – 73.
- [11] Verdile G, Fuller S, Atwood CS, Laws SM, Gandy SE, Martins RN. **The role of beta amyloid in Alzheimer's disease: still a cause of everything or the only one who got caught?** Pharmacol Res 2004; 50(4): 385 – 409.
- [12] Terry RD. **Cortical morphometry in Alzheimer's disease.** In: Katzman R, ed. Banbury Report #15: Biological Aspects of Alzheimer's Disease. Cold Spring Harbor Laboratory, 1983: 95 – 99.
- [13] Mann DMA, Yates PO, Marcyniuk B. **Some morphometric observations on the cerebral cortex and hippocampus in presenile Alzheimer's disease, senile dementia of Alzheimer type and Down's syndrome in middle age.** J Neurol Sci 1985; 69: 139 – 169.
- [14] Hyman BT, Van Heusen GW, Damasio AR, Barnes CL. **Alzheimer's Disease: Cell-specific pathology isolates the hippocampal formation.** Science 1984; 225: 1168 – 1170.
- [15] Mann DMA. **Changes in protein synthesis.** In: Reisberg B, ed. **Alzheimer's Disease.** The Free Press, 1983: 107 – 115.
- [16] Doebler JA, Markesbery WR, Anthony A, Rhoads RE. **Neuronal RNA in relation to neuronal loss and neurofibrillary pathology in the hippocampus in Alzheimer's disease.** Neuropathol Exp Neurol 1987; 46: 28 – 39.
- [17] Goldman JE, Yen SH. **Cytoskeletal abnormalities in neurodegenerative disorders.** Ann Neurol 1986; 19: 209 – 223.
- [18] Hirano A, Zimmerman HM. **Alzheimer's neurofibrillary changes.** Arch Neurol 1962; 7: 227 – 242.
- [19] Baner C et al. **Accumulation of abnormally phosphorylated γ precedes the formation of neurofibrillary tangles in Alzheimer's disease.** Brain Res 1989; 477: 90 – 99.
- [20] Binder LI, Frankfurter A, Rebhun LI. **The distribution of tau in the mammalian central nervous system.** J Cell Biol 1985; 101: 1371 – 1378.

- [21] Katzman R, Terry RD et al. **Clinical, pathological and neurochemical changes in dementia: a subgroup with preserved mental status and numerous neocortical plaques.** Ann Neurol 1988; 23: 128 – 144.
- [22] McCulloch WS, Pitts W. **A logical calculus of the ideas immanent in nervous activity.** Bulletin of Mathematical Biophysics. 1989; 5: 115 – 133.
- [23] Haykin S. **Neural networks: a comprehensive foundation.** Prentice Hall; 2^oEd, 1999.
- [24] Braga AP, Carvalho APLF, Ludermir TB. **Redes Neurais Artificiais: Teoria e Aplicações.** Ed: ETC; 2^oEd, 2007.
- [25] Hopfield JJ. **Neural networks and physical systems with emergent collective properties.** Proc. Nat. Acad. Sci 1982; 79: 2554 – 8.
- [26] Hopfield JJ. **Neurons with graded response have collective computational properties like those of two-state neurons.** Proceedings of the National Academy of Sciences, USA, 1984; 81: 3088 – 3092.
- [27] Kandel, E; Schwartz J., Jessel, T. **Principles of Neural Sciences.** Fourth Edition, International Edition, New York: McGraw-Hill, 2000.
- [28] Levitan, I; Kaczmarek L. **The Neuron: Cell and Molecular Biology.** Second Edition, New York: Oxford University Press, 1997.
- [29] Horn D, Levy N, Ruppin E. **Neuronal-Based Synaptic Compensation: A Computational Study in Alzheimer’s Disease.** Neural Computation 1996; 8(6): 1227 – 1243.
- [30] Howard C, Finkel L. **Computational Approaches to Neurological Disease.** Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.7860>. Acessado em 14 de Novembro de 2008, 11:00h.
- [31] Horn D, Ruppin E, Usher M, Herрман M. **Neural network of memory deterioration in Alzheimer’s disease.** Neural Computation 1993; 5: 736 – 749.
- [32] Ruppin E, Reggia JA. **A Neural Model of Memory Impairment in Diffuse Cerebral Atrophy.** Br. Jour. of Psychiatry 1995; 166(1), 19 – 28.
- [33] Booch G. **UML, Guia do Usuário.** Campos: Rio de Janeiro, 2000
- [34] **Eclipse Platform (ver. 3.0.0 – Build id: 200406251208 Copyright® Eclipse contributors and others 2000, 2004).**