

Introdução às Redes Neurais Artificiais

Jorge M. Barreto

Laboratório de Conexionismo e Ciências Cognitivas
UFSC -Departamento de Informática e de Estatística
88040-900 - Florianópolis - SC

e-mail: barreto@inf.ufsc.br

4/2002

*Uma versão preliminar deste texto foi apresentada na Escola de , omputação da Região
Sul em , , , ,*

Sumário

1	Redes Neurais e Inteligência Artificial	6
1.1	Alguns Fatos Históricos das Redes Neurais	6
1.2	Que Esperar da IAC?	7
2	Fundamentos Biológicos	10
2.1	O Sistema Nervoso	11
2.2	Descobrimto de Neurônio	11
2.2.1	Organização geral	11
2.2.2	Potencial de Ação	12
2.2.3	Transmissão da Informação entre Neurônios	13
3	Vocabulário Básico	13
3.1	Modelos de Neurônios	13
3.1.1	Modelo de McCulloch-Pitts	14
3.1.2	Modelo Geral de Neurônio	14
3.2	Caracterização de RNA	16
4	Topologias das RNAs	17
4.1	Redes diretas (Feedforward)	19
4.2	Redes com ciclos	20
4.3	Redes simétricas	21
4.4	O que as RNAs não são!	22
5	Aprendizado	22
5.1	Aquisição do Conhecimento: Aprendizado	23
5.1.1	Classificação quanto à Independência de quem Aprende	23
5.1.2	Classificação Segundo Retroação do Mundo	23
5.1.3	Classificação quanto à Finalidade do Aprendizado	24
5.2	Regras de Aprendizado Conexionistas	24
5.2.1	Aprendizado Hebbiano	24
5.2.2	Regra Delta	25
5.2.3	Retropropagção	27
5.2.4	Aprendendo com a Natureza	28
5.2.5	Aprendizado Competitivo	29
5.2.6	Aprendizado Reforçado	29
5.2.7	Aprendizado Aleatório	30
5.2.8	Aprendizado Evolutivo	30

6	Máquina de Resolver Problemas	30
6.1	Tipos de Computadores	30
6.2	Teoria de Problemas	31
6.3	O Computador na Resolução de Problemas	32
6.4	Equivalência de Computabilidade	33
6.5	Pontos de Dúvida	35
6.6	Não Equivalência de Complexidade	35
6.7	Alguns Resultados Sobre Complexidade de RNA	36
6.8	Aprendizado de RNA como Paradigma de Programação	37
6.9	Um Pouco de Ficção	38
7	Aplicações das RNAs	38
7.1	Reconhecimento de Padrões	39
7.1.1	Em que Consiste o Reconhecimento de Padrões	39
7.1.2	Reconhecimento de Padrões como Classificador	39
7.1.3	Principais Aplicações	40
7.1.4	Reconhecimento de Caracteres	40
7.1.5	Reconhecimento de Faces	41
7.2	Sistemas Especialistas Conexionistas	41
7.2.1	SE Conexionistas	41
7.2.2	SE usando Redes Diretas	41
7.2.3	SE implementados com BAM	42
7.3	Controle de Processos	43
7.3.1	Controle Neural	43
7.3.2	Topologias de Controle	43
7.3.3	Malha Aberta	43
7.3.4	Controle com Retroação	44
7.3.5	Modelos Internos	44
7.4	Séries Temporais	45
7.5	Monitoramento	45
8	Implementação	46
8.1	Simulação de RNA	46
8.2	Implementações por Circuitos	47
8.2.1	Implementação da Sinapse	47
8.2.2	Implementação do Neurônio	47
8.2.3	Implementação do Aprendizado	48
8.2.4	Implementações Digitais versus Analógicas	48

<i>Redes Neurais Artificiais</i>	4
9 Epílogo	49
Referências bibliográficas	51

Redes Neurais Artificiais

J. M. Barreto

Resumo

Redes neurais artificiais podem ser consideradas como metodologia de resolver problemas característicos da inteligência artificial. Após apresentação de alguns fatos históricos e dos fundamentos biológicos são apresentados os conceitos de neurônio e rede neural artificial. Sendo a capacidade de aprender por exemplos a grande motivadora do uso de redes neurais, os principais paradigmas de aprendizado são apresentados. Segue-se uma comparação das possibilidades destas redes na resolução de problemas dando-se uma visão de computabilidade e complexidade em termos de redes neurais. Finalmente são apresentadas alguns campos de aplicação e como são atualmente estas redes implementadas.

1 Redes Neurais e Inteligência Artificial

Pode-se dizer que redes neurais artificiais consistem em um modo de abordar a solução de problemas de inteligência artificial. Neste caso, em lugar de tentar programar um computador digital de modo a fazê-lo imitar um comportamento inteligente (saber jogar xadrez, compreender e manter um diálogo, traduzir línguas estrangeiras, resolver problemas de matemática tais como se encontram nos primeiros anos dos cursos de engenharia, etc.) procura-se construir um computador que tenha circuitos modelando os circuitos cerebrais e espera-se ver um comportamento inteligente emergindo, aprendendo novas tarefas, errando, fazendo generalizações e descobertas, e frequentemente ultrapassando seu professor. Da mesma forma, estes circuitos neurais artificiais poderão se auto-organizar, quando apresentados a ambientes diversos, criando suas próprias representações internas e apresentar comportamentos imprevisíveis. E, melhor ainda, (ou pior) ter um comportamento que nem sempre pode-se prever e compreender, tal como hoje não compreendemos mecanismos do nosso próprio cérebro.

Ficção científica? Não! Trata-se sim de nova tecnologia que depois de um tempo de latência, emerge encontrando aplicações concretas, algumas das quais serão mencionadas mais adiante.

1.1 Alguns Fatos Históricos das Redes Neurais

O primeiro esforço conjunto para estudar inteligência artificial (IA) foi o encontro no “Darthmouth College”, em 1956. No livro publicado a seguir [83] com o título de “Automata Studies”, o primeiro artigo tratava de redes neurais como um paradigma da arquitetura computacional. Pode-se dizer que aí nasceram simultaneamente os dois paradigmas da inteligência artificial: *simbólica* e *conexionista*.

Na IAS (Inteligência Artificial Simbólica), o comportamento inteligente global é simulado, sem considerar os mecanismos responsáveis por este comportamento. Na IAC (Inteligência Artificial Conexionista) acredita-se que construindo máquina que imite a estrutura do cérebro ela apresentará inteligência. Progressivamente as duas correntes para IA separaram-se, e as pesquisas em redes neurais (corrente conexionista) andaram lentamente enquanto a corrente da manipulação simbólica se acelerou.

É interessante notar que um motivo para esta separação foi o livro de Minsky & Papert [71]. Este livro, entretanto, constitui um dos primeiros estudos sobre a complexidade do problema e a correspondente capacidade das redes neurais para resolvê-lo: “uma perceptron de uma única camada é incapaz de resolver o problemas linearmente não separáveis”. Possivelmente administradores dos USA, responsáveis por distribuir fundos de pesquisa concluíram que o assunto não era interessante e cortaram os investimentos em redes neurais. Os anos que seguiram o encontro Dartmouth foram anos de grande otimismo e o trabalho feito estava centralizado principalmente em pesquisas de laboratório. Entretanto, o progresso em muitos destes campos caminhava lentamente. Um exemplo estava no campo da tradução automática, onde os

problemas apresentavam-se muito mais complexos que o esperado. Por outro lado, muito se estava aprendendo sobre linguagens adequadas para pesquisar em IA! Entretanto, é necessário não esquecer que em alguns outros campos de aplicação a IA obteve sucesso, e que esses seus méritos não são mais considerados como pertencentes a IA. Como exemplos, temos o xadrez (sendo considerado agora como um jogo, e não como um desafio) e fazer matemática simbólica, onde diversos programas comerciais fazem nos esquecer que são resultado de técnicas de IA.

1.2 Que Esperar da IAC?

A IAS já deu muitos frutos, alguns atualmente tão populares que nem são mais considerados como produtos da IA. Exemplos são: jogo de xadrez, sistemas especialistas que fazem apoio à decisão médica, programas de matemática simbólica, etc.

E a IAC? Pode-se dizer que se espera da IAC uma performance melhor que a da IAS em problemas mal definidos, onde falta o conhecimento explícito de como realizar uma tarefa. Nestes o conhecimento é dado através de exemplos. Além disso, características encontradas nos seres vivos são esperadas e dificuldades em realizar tarefas de natureza intrinsecamente algorítmicas. As principais características são:

- **Degradação progressiva e qualidade.** Significa que a performance de um sistema baseado na IAC diminui lenta e monotonicamente em presença de informações falsas ou ausentes. Para ilustrar a idéia, tome-se a pesquisa em uma base de dados em que se deseje obter o nome de um artigo que seja do interesse de um certo leitor caracterizado por seu perfil de interesse. Usando técnicas conexionistas, na falta de um documento satisfazendo aos requisitos da busca, um mais próximo será escolhido.
- **Raciocínio por Default.** É outra propriedade intrínseca de RNA, coisa que requer bastante esforço se for usada a IAS.
- **Generalização.** Uma vez uma rede aprendendo um conceito ela é capaz de funcionar com conceitos similares que não foram aprendidos e isto sem esforço suplementar. Roisenberg [76], [77] estuda no caso de interceptar um objeto voador esta capacidade.
- **Raciocínio impreciso.** Mas, o mais importante é o raciocínio impreciso, que pode ser tratado na IAS pela a lógica nebulosa.

Em resumo, a IAC se baseia no seguinte princípio:

Princípio 1 Princípio da Réplica: *Se for construído um modelo suficientemente preciso do cérebro, este modelo apresentará um comportamento inteligente. Se apenas uma pequena parte do cérebro for reproduzida, a função exercida por esta parte emergirá do modelo.*

Atualmente as duas abordagens estão cada vez mais sendo usadas juntamente, e às duas se junta ainda uma outra família de abordagens: as inspiradas na evolução biológica e constituem os sistemas evolucionário, também chamados evolutivos¹ que serão abordados no capítulo ??, base da *Inteligência Artificial Evolucionária* ou IAE.

Também esta se baseia em um princípio conhecido com o nome de *Seleção Natural*, tirado dos trabalhos de Darwin [24] e que pode ser enunciado como:

Princípio 2 Princípio da Seleção Natural: *Dada uma população de indivíduos vivendo em um determinado ambiente, os mais aptos às condições de vida oferecidas, tem mais probabilidade de se reproduzir do que os menos aptos. Desta forma, com o correr do tempo e após gerações sucessivas a população tende a ficar cada vez mais adaptada ao ambiente.*

Este paradigma permite a resolução de problemas ser feita considerando várias soluções possíveis como os indivíduos da população e o problema a resolver como o ambiente. A adaptação seria então a qualidade da solução permitindo considerações sobre o quão “inteligente” seria a solução comparada com as demais [31].

Com o que foi dito é possível dividir as fases da história da IA² com nos seguintes períodos [9]:

1. **Época pré-histórica** (Nesta época nada se conhecia sobre os mecanismos da mente, nem sob o prisma fisiológico nem psicológico e por esta razão vai até 1875 quando Camillo Golgi visualizou o neurônio.)

Objetivo: Criar seres e mecanismos apresentando comportamento inteligente.

Metodologia e Conquistas: Mecanismos usando mecânica de precisão desenvolvida nos autômatos, mecanismos baseados em teares, etc. Apelo ao sobrenatural.

Limitações: Complexidade dos mecanismos, dificuldades de construção. Insucesso dos apelos ao sobrenatural.

2. **Época Antiga** (1875-1943) (Época em que a Lógica formal apareceu (Russel, Gödel, etc) bem como se passou a reconhecer o cérebro como órgão responsável pela inteligência. Hilbert imaginava um mundo paradisíaco, em que tudo poderia ser axiomatizado e reduzido à Lógica. Entretanto, assim como o final do século XIX viu o desmoronamento do mundo Euclidiano, Gödel abalou o mundo de Hilbert com seu teorema da incompletude da aritmética. Foi a época em que, tal como os filósofos gregos fizeram, são colocadas as bases da IAS e IAC, terminando com a publicação do trabalho de McCulloch e Pitts modelando o neurônio [66]):

¹Neste texto serão usados indistintamente e como sinônimos, os dois termos evolutivo e evolucionário.

²Note-se que os termos usados não se referem a períodos da história da humanidade e sim história da IA. Assim, o tremo “pré-história” nada tem a ver com a época em que seres humanos não deixaram documentos escritos.

Objetivo: Entender a inteligência humana.

Metodologia e Conquistas: Estudos de psicologia e de neurofisiologia. Nascimento da psicanálise.

Limitações: Grande distância entre as conquistas da psicologia e da neurofisiologia.

3. **Época Romântica** (1943-1956) (É o otimismo desordenado, que tal um jovem rapaz romântico³, crê que tudo é possível. Acaba com a reunião no Dartmouth College):

Objetivo: Simular a inteligência humana em situações pré-determinadas.

Metodologia e Conquistas: Inspiração na Natureza. Nascimento da Cibernética. Primeiros mecanismos imitando funcionamento de redes de neurônios. Primeiros programas imitando comportamento inteligente.

Limitações: Limitação das capacidades computacionais.

4. **Época Barroca** (1956-1969) (Tudo é fácil e será conseguido. O livro Perceptrons [71] mostra que nem tudo é possível. Além disto, grandes fabricantes de computadores, interessados em vender suas máquinas para aplicações de escritório, tiveram grande interesse em desmistificar o computador que na época chegou a ser chamado pelo grande público de “cérebro eletrônico”. Estes dois fatos marcaram o fim da época.):

Objetivo: Expandir ao máximo as aplicações da IA tanto usando a abordagem simbólica quanto a conexionista.

Metodologia e Conquistas: Perceptron. Primeiros sistemas especialistas usando a abordagem simbólica. Grandes esperanças da IAS.

Limitações: Dificuldades em técnicas de aprendizado de redes complexas.

5. **Época das Trevas** (1969-1981) (Paralização de quase todas as pesquisas em IA por falta de verbas. Acabou quando em outubro os japoneses anunciaram seus planos para a Quinta Geração de Computadores e em outro ambiente Hopfield publica célebre artigo sobre redes neurais. Uma característica interessante é que o renascimento de IA simbólica se fez em ambiente de computação e o de redes neurais em um ambiente interdisciplinar.

Assim como a Idade Média da História da humanidade viu florescer idéias novas, esta época não foi de total trevas. Nasceram as primeiras aplicações dos conjuntos nebulosos de Zadeh [87] nascendo o controle inteligente com Mamdani [54, 64]. Além disto os sistemas especialistas se firmaram com Shortliffe [84])

Objetivo: Encontrar para a IA aplicações práticas.

³Para satisfazer a uma minha aluna que teve a gentileza de ler e sugerir melhoras no texto, troquei “a jovem romantica” por rapaz “romântico” evitando conotação machista.

Metodologia e Conquistas: Sistemas especialistas. Aplicações principalmente em laboratórios. Os computadores usados principalmente para aplicações administrativas e numéricas. Interesse dos fabricantes de computadores de desmistificar a máquina levando a pouco interesse em IA⁴.

Limitações: Interesses econômicos.

6. **Renascimento** (1981-1987) (Começou a corrida para IA. Os resultados obtidos nas épocas anteriores atingiram o público em geral. Sistemas especialistas se popularizaram. Primeira conferência internacional em Redes Neurais marca o final do período. Note-se que redes neurais evoluiu independentemente da IA simbólica.):

Objetivo: Renascimento da IA, simbólica e conexionista.

Metodologia e Conquistas: Popularidade da linguagem Prolog, adotada pelos japoneses. Crescimento da importância da Lógica. Proliferação de máquinas suportando ferramentas para IA. Alguns poucos pesquisadores continuaram seus trabalhos em RNAs, Grossberg, Kohonen, Widrow, Hinton, etc. No final do período, trabalhos de Hopfield, do grupo PDP, etc., criaram condições para a fase seguinte no que diz respeito às RNAs.

Limitações: a IAS e a IAC evoluindo separadamente.

7. **Época Contemporânea** (1987 - atual): (Logo no início do período Gallant [36] publica seu célebre artigo sobre sistemas especialistas conexionistas. Foi o ponto de partida para a união das duas abordagens de IA, tornando a abordagem dirigida *problemas* a abordagem atual. É a abordagem adotada neste trabalho.

Objetivo: Alargamento das aplicações das IAs. Uso em tomografia, pesquisas em campos de petróleo, e bases de dados inteligentes.

Metodologia e Conquistas: Redes diretas como aproximador universal. Lógica nebulosa usada largamente em indústrias para controle inteligente. Sistemas especialistas se torna tecnologia dominada. Bons resultados em problemas mal definidos com sistemas usando hibridismo neural-nebuloso. Novo paradigma de programação: programação conexionista.

Limitações: Quem sabe???. Uma possibilidade é uma grande expansão das bases de dados inteligentes.

2 Fundamentos Biológicos

Supondo que grande parte dos leitores deste texto não estão mais familiarizados com conceitos biológicos em que as redes neurais se inspiram, pequena introdução aqui é feita.

⁴Um exemplo é a referência à IA como sendo “ignorância atrevida”, usual a título de humor.

2.1 O Sistema Nervoso

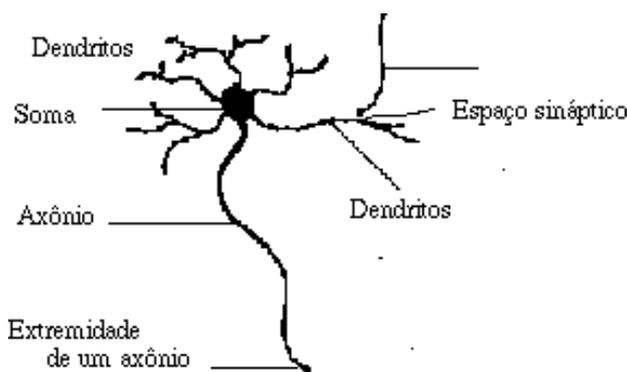
2.2 Descobrimto de Neurônio

Um dos primeiros passos na neuroanatomia foi a descoberta do italiano Camillo Golgi em 1875 [50]. Ele encontrou um método, aparentemente ao acaso, pelo qual apenas uma pequena porção de neurônios são corados durante um tempo, e essas células são completamente coradas. Com esse método é possível isolar e observar neurônios individuais. Golgi foi contemporâneo de Santiago Ramón y Cajal, um Espanhol, que dedicou sua vida usando o método de Golgi para cada parte do sistema nervoso. Nesta época as junções entre neurônios eram desconhecidas, principalmente porque o método de Golgi revelou um grande número de células isoladas mas sem sugerir junções entre estas no sentido de formar uma rede.

Entretanto os estudos de Cajal apresentaram dois resultados principais. Primeiro Cajal adotou a noção de sistema nervoso, postulando sobre a comunicação entre as células pela sinapse. A segunda foi de que a interconexão entre neurônios seria é altamente específica e estruturada. Ele descreveu diversas estruturas cerebrais. Isto fez com que as pesquisas se voltassem não apenas para a estrutura dos neurônios mas também para o desconhecido campo das muitas estruturas neuronais.

2.2.1 Organização geral

O sistema nervoso juntamente com o sistema endócrino proporciona muitas das funções de controle do corpo. O sistema nervoso de que o cérebro faz parte, controla as reações rápidas do corpo, como uma contração muscular (função motora) e controla a velocidade e equilibra a taxa de secreção de muitas glândulas endócrinas. Partes do corpo que têm as funções controladas pelo sistema nervoso tem tempo de resposta relativamente rápido. O sistema endócrino, por outro lado, controla muitas funções do metabolismo do corpo e sua atuação é mais lenta.



Um neurônio Existem dois tipos principais de células no cérebro, os neurônios e a glia. É comum atribuir aos neurônios as principais funções cerebrais já que a função da glia ainda tem muito a ser descoberto. Existem aproximadamente 10^{11} neurônios (um fator de 10 é razoável como expectativa de erro) no cérebro humano. Eles podem ser de diversos tipos. Um neurônio típico é apresentado na figura ao lado. O neurônio tem um corpo celular chamado

soma e diversas ramificações. As ramificações conhecidas como *dendritos*, conduzem sinais das extremidades para o corpo celular. Existe também uma ramificação, geralmente única, chamada *axônio*, que transmite um sinal do corpo celular para suas extremidades. As extremidades do

axônio são conectadas com dendritos de outros neurônios pelas *sinapses*. Em muitos casos, um axônio é diretamente conectado com outros axônios ou com o corpo de outro neurônio.

As sinapses tem um papel fundamental na memorização da informação e são principalmente as do córtex cerebral e algumas vezes de partes mais profundas do cérebro que armazenam esta informação. Pode-se imaginar que em cada sinapse, a quantidade de neurotransmissores que podem ser liberados para uma mesma frequência de pulsos do axônio representa a informação armazenada nesta sinapse.

Ora, pode-se imaginar, que seguindo um princípio frequentemente válido em biologia, o de que o uso de um órgão favorece seu desenvolvimento, que cada vez que uma sinapse é ativada e encontra ativado ou consegue ativar outro neurônio o número de neurotransmissores liberados aumenta na próxima vez que o neurônio for ativado. Isto representa um aumento da conexão entre os dois neurônios. Este processo chama-se *facilitação*. Um neurônio tem de 1000 a 10000 sinapses e pode receber informação de perto de 1000 outros neurônios.

O mecanismo de facilitação inspirou a conhecida Lei de Hebb: *A intensidade de uma conexão sináptica entre dois neurônios aumenta quando os dois neurônios estão excitados simultaneamente*. Note-se que a Lei de Hebb é básica de muitos algoritmos de aprendizagem de RNA.

2.2.2 Potencial de Ação

Existe dentro e fora da células concentrações diferentes de Na^+ e K^- que provocam um ponto de equilíbrio de -85 milivolts, o interior da célula negativo com relação ao exterior. Qualquer perturbação da membrana do neurônio provoca uma série de modificações que desaparecem também rapidamente, e durante as quais o potencial se torna positivo durante um curto espaço de tempo. A esta onda de variação de tensão chama-se *potencial de ação*.

A formação de um potencial de ação pode ser causado por uma estimulação elétrica, química, calor, etc. Um estímulo tem por efeito a destruição das propriedades dielétricas da membrana, em particular as permeabilidades tanto ao sódio como ao potássio, que são aumentadas permitindo a difusão destes ions através da membrana. Após um certo tempo as coisas voltam ao normal devido a mecanismos de transporte ativo (transporte de moléculas através da membrana celular contra o gradiente de concentração e com uso de energia). Esta fase chama-se *repolarização*. Logo após a repolarização a membrana passa por um período de tempo durante o qual ela não é mais sensível a outras perturbações e que se chama *período refratário*.

Este período refratário tem papel preponderante na transmissão de pulsos elétricos no axônio. Suponha que por uma razão qualquer apareça no axônio, perto do soma uma perturbação provocando um potencial de ação. Ele age com relação ao trecho do axônio um pouco mais longe do soma como uma perturbação, provocando o aparecimento de novo potencial de ação um pouco mais longe e assim por diante até a região junto à sinapse onde o potencial de ação tem por efeito liberar moléculas de neurotransmissores. Estes neurotransmissores liberados se difundem no espaço entre neurônios indo se colar na membrana de um outro neurônio, provocando uma

perturbação de membrana do outro neurônio, e o fenômeno continua.

2.2.3 Transmissão da Informação entre Neurônios

Quando o potencial de ação se propaga pelo axônio, chega a uma de suas terminações. Aí ele provoca modificações na membrana destas terminações, as *sinápses*. Isto permite a liberação de moléculas, de vários tipos com o nome genérico de *neurotransmissores* que se difundem no espaço entre o terminal do axônio e um outro neurônio, geralmente o terminal de um *dendrito*. Moléculas de neurotransmissor ao se colar ao dendrito provocam uma modificação na membrana deste que acaba, algumas vezes, provocando um potencial de ação, outras vezes, dificultando seu aparecimento. Este potencial de ação, se criado, se propaga até o soma do neurônio que recebeu o sinal alterando sua frequência de disparo.

Pode-se dizer portanto que a transmissão de informação entre neurônios depende do tipo de neurotransmissor e de sua abundância no terminal sináptico e da sensibilidade da membrana dendrítica à excitações. Desta forma modificando a intensidade com que um neurônio é capaz de excitar (ou inibir) um outro neurônio, depende de características sinápticas, e são estes valores de conexões que globalmente são responsáveis pelo comportamento da rede de neurônios. Mudando valores destas conexões muda-se o comportamento da rede. E estas mudanças de comportamento representam um *aprendizado da rede*.

Como o comportamento de uma rede neural depende diretamente dos valores de suas conexões sinápticas, o estudo de redes neurais tem também o nome de *Conexionismo*.

3 Vocabulário Básico

A terminologia usada em redes neurais artificiais apresenta variações, principalmente em textos em português. Neste trabalho usa-se a terminologia da Teoria de Sistemas tal como apresentada em [8].

3.1 Modelos de Neurônios

A construção de redes neurais artificiais (RNAs) tem inspiração nos neurônios biológicos e nos sistemas nervosos. Entretanto, é importante compreender que atualmente as RNAs estão muito distantes das redes neurais naturais (RNNs) e freqüentemente as semelhanças são mínimas. Se é verdade que o primeiro modelo de neurônio, proposto por McCulloch e Pitts em 1943 [66] é também um modelo simples, cabe ressaltar que a intenção era de imitar a realidade biológica, preocupação não compartilhada pelos muitos pesquisadores atuais. De fato, dois fatores diferentes motivam a pesquisa hoje em dia:

- O primeiro é modelar o sistema nervoso com suficiente precisão de tal modo a poder observar um comportamento emergente que sendo semelhante ao comportamento do ser

vivo modelado, possa servir de apoio às hipóteses usadas na modelagem.

- O segundo é construir computadores com um alto grau de paralelismo.

O trabalho na modelagem do sistema nervoso começou há um século aproximadamente. Depois do trabalho de McCulloch and Pitts [66], Hebb [43], e Rosenblatt [78], muitos cientistas se interessaram pelo campo. O desejo de construir neurocomputadores é mais recente [44].

3.1.1 Modelo de McCulloch-Pitts

Warren S. McCulloch era um fisiologista e conhecendo as ondas de potencial de membrana ele interpretou o funcionamento do neurônio como sendo um circuito binário. Seu modelo [65] é portanto binário e é apresentado na figura 1.

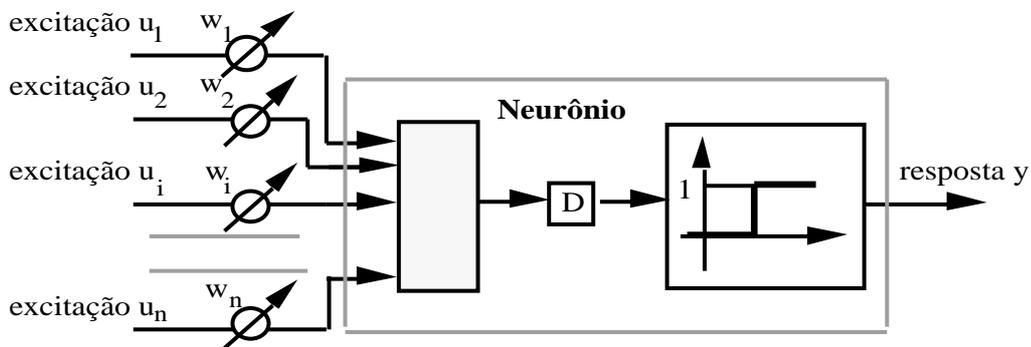


Figura 1: Modelo de McCulloch e Pitts

A entrada do neurônio é também binária e as várias entradas são combinadas por uma soma ponderada, produzindo a entrada efetiva do neurônio:

$$entrada_efetiva = \sum_{i=1}^n \omega_i u_i \quad (1)$$

O resultado na entrada efetiva sofre um retardo D (algumas vezes este retardo é desprezado tendo-se um neurônio estático) e serve de argumento a uma função chamada de *função de transferência* (neste caso de saída binária $\{0, 1\}$) para dar a resposta do neurônio.

3.1.2 Modelo Geral de Neurônio

O modelo geral de neurônio é mostrado na figura 2, sendo uma generalização do modelo de McCulloch e Pitts.

Neste modelo as entradas $w_i u_i$ são combinadas usando uma função Φ , para produzir um estado de ativação do neurônio que através da função η vai produzir a saída do neurônio (correspondente à frequência de descarga do neurônio biológico). Um valor auxiliar θ é geralmente usado para representar uma polarização, valor abaixo do qual a saída é nula.

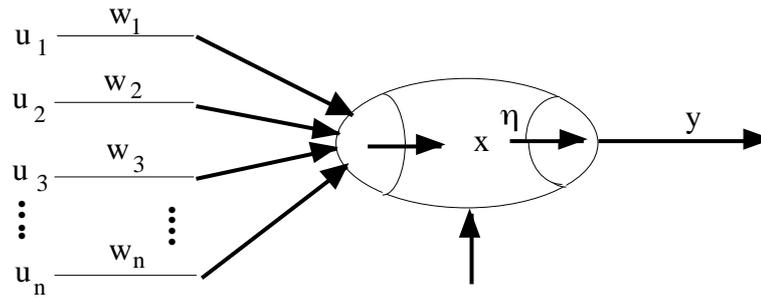


Figura 2: Neurônio artificial

Note-se que isso poderia também ser obtido por escolha adequada da função η , mas seria mais difícil de trabalhar. Note-se ainda que as conexões sinápticas são consideradas como externas ao modelo do neurônio, tal como ocorre no sistema nervoso biológico e não como fazendo parte do neurônio, como usado por alguns autores. Se este detalhe pode ter pouca importância aparente no estudo de uma RNA, proporciona a possibilidade de interpretar a matriz de conexões, chamada *matriz de conectividade* como a matriz de pesos de um grafo, o grafo representativo da rede neural.

Geralmente Φ é a soma das entradas. Algumas vezes o produto. Raramente uma outra função, se bem que isto seja possível. Geralmente costuma-se dar o nome *confluência* à combinação ponderada das entradas de um neurônio. A não linearidade do neurônio frequentemente é introduzida na função tangente hiperbólica, em degraus. A Figura 3 mostra algumas funções comumente usadas.

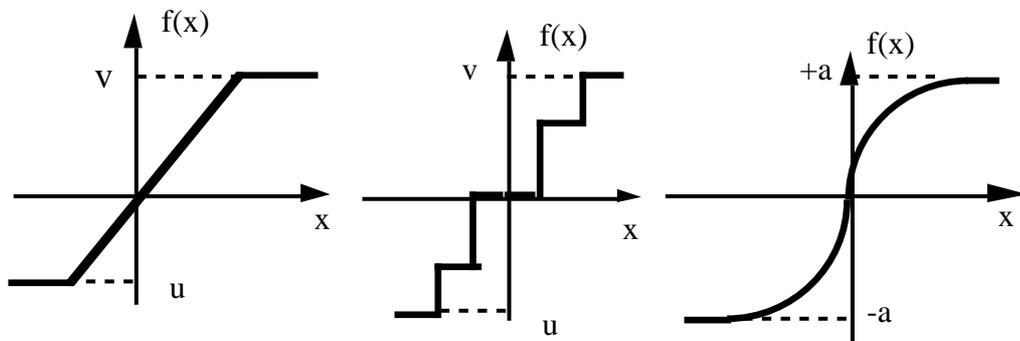


Figura 3: Não linearidades frequentemente usadas no modelo de neurônios de uma RNA.

O neurônio formal é um sistema dinâmico por ter memória materializada pelo retardo (ou equação diferencial). Um neurônio é estático quando o valor de x e de y se referem ao mesmo instante que as excitações ou seja, o retardo é nulo. O neurônio é dito dinâmico se para o cálculo de x em um determinado instante é necessário o conhecimento de x em um instante anterior no

caso do neurônio ser a tempo discreto.

Por esta definição nota-se que o modelo de neurônio proposto por McCulloch e Pitts é um sistema dinâmico se o retardo D não for nulo.

3.2 Caracterização de RNA

Informalmente uma rede neural artificial (RNA) é um sistema composto por vários neurônios. Estes neurônios estão ligados por conexões, chamadas conexões sinápticas. Alguns neurônios recebem excitações do exterior e são chamados *neurônios de entrada* e correspondem aos neurônios dos órgãos dos sentidos. Outros têm suas respostas usadas para alterar, de alguma forma, o mundo exterior e são chamados *neurônios de saída* e correspondem aos motoneurônios que são os neurônios biológicos que excitam os músculos. Os neurônios que não são nem entrada nem saída são conhecidos como *neurônios internos*. Estes neurônios internos à rede tem grande importância e são conhecidos na literatura saxônica como “*hidden*” fazendo com que alguns traduzam como “escondidos”.

Os neurônios internos são importantes por vários aspectos:

- **Importância biológica:** Por corresponder a uma atividade do sistema nervoso que pode apresentar uma independência de excitações externas. Com efeito, se entre estes neurônios houver ligações formando ciclos, e considerando ainda um certo tempo de resposta de um neurônio, após cessar toda excitação exterior pode haver nestes neurônios internos uma evolução de um vetor representativo da excitação destes neurônios. Esta excitação pode provocar uma evolução durante um tempo relativamente longo e pode ser interpretada como uma metáfora da mente, onde pensamentos vêm e voltam, sem estímulo exterior.
- **Importância matemática:** Desde que se provou que sem estes neurônios é impossível uma RNA resolver problemas classificados como linearmente não separáveis.

Para caracterizar uma RNA é importante especificar os seguintes pontos⁵:

- **Os componentes da rede: os neurônios:** ex; estáticos? dinâmicos?
- **A resposta de cada neurônio:** dicotômica? intervalo dos reais?
- **O estado global de ativação da rede:** vetor cujas componentes são as ativações dos neurônios?
- **A conectividade da rede dada pelos valores de conexões sinápticas:** que define a topologia da rede.
- **Como se propaga a atividade da rede:** síncrona? assíncrona?

⁵Inspirado em Rumelhart & al. [80].

- **Como se estabelece a conectividade da rede:** aprendizado.
- **O ambiente externo à rede:** estático? dinâmico? aleatório? determinístico?
- **Como o conhecimento é representado na rede:** localizado? distribuído?

4 Topologias das RNAs

De forma a definir as Redes Neurais Artificiais nós poderíamos, em princípio, estabelecer (e provar) um teorema mostrando que elas se constituem em sistemas dinâmicos, da mesma forma que foi feito para os neurônios. Todavia, um problema surgiria aqui: *nada seria dito acerca dos pesos das conexões entre os diferentes neurônios da rede*. Uma outra abordagem seria a de considerar uma rede neural como um sistema dinâmico complexo, onde:

Definição 1 *Um sistema dinâmico complexo é uma rede de sistemas interconectados.* ■

Da definição apresentada decorre que um sistema complexo pode ser representado por um grafo direcionado ou dígrafo, onde os vértices representam os sistemas componentes (subsistemas) e os arcos as interações entre subsistemas.

Esta será a abordagem utilizada aqui. No entanto, antes de desenvolver estes conceitos, faz-se necessário apresentar alguns conceitos de Teoria de Grafos. Estes conceitos são baseados naqueles apresentados por Harary [41].

Nota: É importante observar que, considerando que, em princípio, qualquer dígrafo possa dar lugar a uma topologia de RNA, esta abordagem vem sendo utilizada em textos surgidos nos últimos anos, como por exemplo [42], [52] entre outros. No entanto, De Azevedo [25] utilizou esta abordagem ainda em 1993.

Definição 2 *Um Grafo G consiste de um conjunto não vazio finito de vértices $V = v_i$ juntamente com um conjunto não ordenado de arcos A conectando certos pares de vértices. Cada par v_i, v_j de vértices em V é um arco de G , e o par v_i, v_j é dito “juntar” v_i e v_j .* ■

Definição 3 *Um Grafo G é Rotulado quando os p vértices são distinguíveis uns dos outros por nomes ou rótulos.* ■

Definição 4 *Um Grafo G é Arco Rotulado quando também os arcos são distinguíveis um dos outros por nomes ou rótulos.* ■

Tendo estabelecido o que significa “Grafo”, no nosso contexto, nós estamos prontos para definir as Redes Neurais Artificiais.

Definição 5 *Uma Rede Neural Artificial, RNA, é um Sistema Dinâmico Complexo representado por um grafo arco rotulado em que cada vértice é um Neurônio Artificial NA.* ■

Nesta definição, rótulos são, naturalmente, valores numéricos. Eles correspondem aos valores das conexões entre os diferentes neurônios. Todavia, eles podem ser interpretados, também, como os valores “fuzzy” entre as conexões. Neste caso, eles devem pertencer a um conjunto, que na maioria dos casos, é o conjunto $[0, 1]$ ⁶. Ambas interpretações são válidas para nossos propósitos. Todavia, se nós escolhermos a segunda interpretação nós poderíamos repensar a definição de Grafos e, por consequência, a de Redes Neurais, conforme segue:

Definição 6 *Um Grafo Nebuloso é um Grafo Arco Rotulado onde os rótulos são valores de um conjunto nebuloso.* ■

Definição 7 *Uma Rede Neural Nebulosa é ou:*

- *uma rede neural representada por um grafo nebuloso iu*
- *uma rede neural contendo ao mnenos um neurônio nebuloso.*

■

Tendo estabelecido definições precisas para Redes Neurais e Redes Neurais “Fuzzy” nós podemos definir diferentes tipos de redes. Isto é feito através de escolhas particulares dos conjuntos e funções envolvidas na definição de Redes Neurais como Sistemas Dinâmicos. Tem-se, por conseguinte:

Definição 8 *Uma Rede Neural Contínua no Tempo é uma rede neural definida em um subconjunto contínuo do eixo do tempo $T = \mathbb{R}$.* ■

Definição 9 *Uma Rede Neural Discreta no Tempo é uma rede neural definida em um subconjunto discreto do eixo do tempo $T = \mathbb{Z}$.* ■

Definição 10 *Uma Rede Neural Invariante no Tempo ou Rede Neural Estacionária é uma rede neural em que a função de transição Φ depende de um único elemento de T e a função de saída λ é independente de T .* ■

Neste trabalho nós consideramos ambos os tipos de redes, contínuas e discretas. Todavia, todas são invariantes no tempo para permitir uma fácil tratabilidade matemática.

Aqui uma questão surge. Relembremos a definição de “automatum”. Seguindo esta definição um “automatum” é um sistema dinâmico discreto e invariante no tempo. A questão que surge é: Pertencem as RNA à classe dos “automata” ?

⁶Outros intervalos de valores podem ser, também, considerados para conjuntos “fuzzy”.

Aliás, esta questão é muito importante posto que os Computadores baseados em Instrução (CBI) ⁷ estão intrinsecamente ligados a Teoria de “Automata”. A resposta é afirmativa. Pode ser claramente provado que qualquer rede neural discreta e invariante no tempo é um “automatum”. Este resultado permite que o formalismo que é usado para representar RNA e Computadores baseados em Redes Neurais (CBRN) seja o mesmo daquele usado para representar CBI’s. Este fato torna mais fácil o estudo da integração das duas abordagens quando do desenvolvimento de computadores híbridos.

Nesta direção, um resultado surpreendente é que qualquer “automatum” finito pode ser, essencialmente, substituído por uma RNA. A prova deste estabelecimento foi feita por McCulloch e Pitts [66]. Arbib apresentou, em [3], uma prova mais didática. A partir destes resultados é fácil mostrar as capacidades das RNA para memória e computação.

Até agora nós propusemos definições matemáticas para NA’s e RNA’s. Estas definições permitem o estudo de diferentes tipos particulares de RNA’s como sistemas dinâmicos. A abordagem dinâmica para RNA serve como um guia para o estudo da capacidade de memória e para formular idéias no sentido de uma Teoria da Computabilidade adaptada a RNA. A seguir serão apresentadas as topologias de RNA’s que podem ser derivados de nossos modelos formais.

Nota: Faz-se necessário dizer que algumas topologias particulares receberam maior atenção dos pesquisadores e são conhecidas com nomes específicos.

4.1 Redes diretas (Feedforward)

Definição 11 *Redes Diretas (“Feedforward”)* são aquelas cujo grafo não tem ciclos. ■

Freqüentemente é comum representar estas redes em camadas e, neste caso, são chamadas *redes em camadas*. Neurônios que recebem sinais de excitação são chamados da *camada de entrada*, ou primeira camada. Neurônios que têm sua saída como saída da rede pertencem a *camada de saída* ou última camada. Neurônios que não pertencem nem a camada de entrada nem a de saída são neurônios internos à rede podendo se organizar em uma ou mais camadas internas (“hidden layers”).

A figura ?? mostra uma rede direta com 3 camadas de neurônios. Observe que nesta figura os neurônios são apresentados com os seus diversos elementos constituintes conforme a figura ?. Estas redes são atualmente as mais populares, principalmente por existirem métodos de aprendizado bastante difundidos e fáceis de usar. Um método bastante usado, mas muito ineficiente, é o “backpropagation”. Por esta razão alguns autores chegam mesmo a chamar, impropriamente, este tipo de rede, de “backpropagation”. Além disto, estas redes são capazes de aproximar, com maior ou menor precisão, dependendo do número de neurônios da rede, qualquer função não-linear. Entretanto, mesmo no caso de usarem neurônios dinâmicos (equação diferencial de

⁷Para saber mais dos conceitos de CBI e CBRN veja [5], [10] [25], etc

primeira ordem ou a uma diferença finita), têm uma dinâmica muito limitada não podendo representar todos os sistemas dinâmicos.

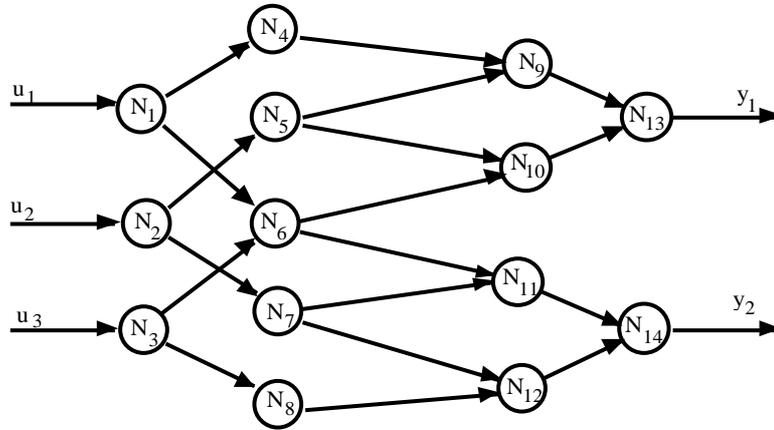


Figura 4: Uma rede direta com 3 camadas de conexões

Com efeito, seja por exemplo, uma rede síncrona de 4 camadas com neurônios definidos por uma equação contendo um retardo. Neste caso, a rede se comportará como um filtro não-linear FIR de ordem 4, sendo sua aproximação linear um sistema com todos os polos na origem do plano \mathbf{Z} não podendo aproximar convenientemente sistemas de resposta indicial de duração infinita.

4.2 Redes com ciclos

Definição 12 *Redes com ciclos (ou com realimentação, ou com retroação, ou com “feedback”) são aquelas cujo grafo de conectividade contém, ao menos, um ciclo.* ■

Um exemplo bem conhecido de rede com ciclos é a proposta por Hopfield [47].

Definição 13 *Redes recorrentes são aquelas que, além de apresentarem ciclos, envolvem neurônios dinâmicos.* ■

Por esta razão McCulloch chamou-as de “networks with cycles”, ou redes com ciclos. Duas destas redes têm particular importância: as redes propostas por Hopfield [47, 48] e as redes bi-direcionais, de Kosko [58], que podem ser usadas em um dos dois principais paradigmas de sistemas especialistas: treinamento com exemplos de uma rede direta e representação do conhecimento de modo localizado pelo uso de rede com ciclos, geralmente uma rede simétrica.

Com efeito, o mais popular paradigma usado na implementação de sistemas especialistas com redes neurais usa redes diretas e foi proposto por Gallant [36], existindo bastantes resultados neste domínio [37]. Baseia-se no fato de que as redes diretas são *aproximadores universais de*

funções. Assim, apresenta-se, na entrada da rede, os dados e ela é, então, treinada para a saída representar o parecer do sistema especialista. O funcionamento da RNA se torna uma metáfora de um ato reflexo que, depois de aprendido, se repete inconscientemente. E este aspecto, que é uma das forças do paradigma, pois *pode ser largamente empregado*, constitui-se, também, em um dos seus pontos fracos, pois, tal como um ato reflexo, *é difícil explicar o porquê do ato*. Assim é que, com o uso das redes diretas, existem dificuldades em extrair explicações de como o sistema chegou a uma conclusão⁸.

O segundo paradigma usa redes bidirecionais, caso particular das redes com ciclos, contendo neurônios dinâmicos [27]. Neste caso, tanto os dados, como os possíveis pareceres do especialista são representados pela ativação de neurônios, o conhecimento sendo representado por valores das intensidades de conexões sinápticas. Uma consulta é feita excitando neurônios representativos dos sintomas presentes no caso, deixando a rede evoluir até atingir um ponto de equilíbrio. A excitação de algum (ou alguns) neurônios, representando pareceres, será a resposta do sistema.

Este segundo paradigma é mais recente e ainda pouco explorado. Os principais pontos fracos são:

- é difícil saber se a rede vai, ou não, parar em um ponto de equilíbrio;
- o tempo correspondente ao transitório da rede pode ser longo.

As principais vantagens são:

- Uso de representação de conhecimento localizada, facilitando extração de explicações;
- ausência de método de aprendizagem ;
- transitório da rede pode ser interpretado como metáfora de raciocínio, podendo-se esperar deste paradigma, mais do que um simples ato reflexo.

4.3 Redes simétricas

Definição 14 *Uma rede simétrica é aquela cuja matriz de conectividade é uma matriz simétrica.* ■

Trata-se de um caso particular das redes com ciclos. Com efeito, os sistemas especialistas mencionados anteriormente e que usam redes com ciclos, usam redes simétricas. Isto foi feito para assegurar estabilidade do transitório da rede⁹.

⁸Note-se que no artigo de Gallant [36], sugestões de como extrair explicações são apresentadas. Mas é difícil usar estas sugestões em um caso geral.

⁹Pode ser mostrado que tal caso se trata de sistema discreto em que a aproximação linear tem polos de módulo menor que a unidade. Assim, conforme o teorema de Liapunov da estabilidade local [40] [39], o sistema terá, ao menos, um ponto de equilíbrio estável.

4.4 O que as RNAs não são!

Sim, as RNAs são inspiradas na redes neurais biológicas (RNB). Mas até onde esta inspiração é usada? Na realidade, freqüentemente esta inspiração é muito limitada e as RNA são uma caricatura da realidade biológica.

- **RNN não são circuitos digitais.** O modelo apresentado por McCulloch-Pitts [66] usava sinais binários. O neurônio biológico expressa sua ativação pela freqüência que emite pulsos e esta freqüência tem uma variação contínua entre dois valores positivos.
- **RNN não podem ter excitação negativa.** Alguns modelos usam valores de excitação negativa.
- **RNN não são homogêneas.** As RNN não possuem todos os seus neurônios de mesmo tipo como nas RNA, apenas em algumas regiões existe uma certa uniformidade no tipo de neurônios existentes nas RNN.
- **RNN não são circuitos síncronos ou assíncronos.** Com efeito, as RNB são sistemas a tempo contínuo, logo não cabe a classificação de síncrono ou assíncrono.
- **Nem neurônios nem sinapses tem dois valores.** Logo a semelhança com o “spin” do eletron não é válida [56].
- **Circuitos cerebrais não são capazes de cálculos recursivos.** Isto é consequência dos neurônios não serem sistemas discretos, levando a rede a não ser um autômato. Logo, equivalência com problemas solúveis por funções recursivas não tem sentido biológico. Entretanto, os neurônios das RNAs fazem das RNAs sistemas equivalentes à máquina de Turing e portanto capazes de resolver funções recursivas.

5 Aprendizado

Aprender é o ato que produz um comportamento diferente a um estímulo externo devido à excitações recebidas no passado e é de uma certa forma sinônimo de aquisição de conhecimento. Em IA é comum se falar de aprendizado pela máquina e aprender poder ser considerado como atributo fundamental de um comportamento inteligente.

RNA possuem a capacidade de aprenderem por exemplos, e fazerem interpolações do que aprenderam. No aprendizado conexionista não se procura obter regras como na abordagem simbólica da IA, mas determinar a intensidade de conexões entre neurônios. Como o conhecimento é armazenado nas conexões, o uso de RNA está intimamente ligado ao que se chama de *conexionismo*.

5.1 Aquisição do Conhecimento: Aprendizado

5.1.1 Classificação quanto à Independência de quem Aprende

Quanto à Independência de quem Aprende tem-se:

- Memorização.
- Por ser contado.
- Por exemplos.
- Por analogia.
- Por exploração e descoberta.

As RNAs aprendem principalmente por uma mistura dos tres últimos.

5.1.2 Classificação Segundo Retroação do Mundo

Um segundo modo de classificar o aprendizado é pela presença ou ausência de realimentação explícita do mundo exterior. Uma realimentação explícita significa que em certos intervalos de tempo um professor assinala erros e acertos. No caso que que a realimentação não é explícita é o aprendizado em ausência de professor. Costuma-se chamar estes dois casos de ensino *supervisionado* e *não supervisionado*.

- **Aprendizado Supervisionado**

Neste caso o ‘professor’ indica explicitamente um comportamento bom ou ruim. Por exemplo, seja o caso de reconhecimento de caracteres e para simplificar seja reconhecer entre um A ou X. Escolhe-se uma rede direta, com dois neurônios na camada de saída, uma ou várias camadas internas e uma conjunto de neurônios na camada de entrada capaz de representar com a precisão desejada a letra em questão. Apresentam-se estas letras sucessivamente a uma retina artificial constituída por uma matriz de elementos fotosensíveis, cada um ligado a um neurônio de uma RNA direta (feedforward). Observa-se qual dos dois neurônios de saída está mais excitado. Se for o que se convencionou representar a letra que for apresentada nada deve ser corrigido, caso contrario modifica-se os valores das conexões sinápticas no sentido de fazer a saída se aproximar da desejada. Foi exatamente isto que fez Rosenblatt com o seu Perceptron. Como a cada exemplo apresentado uma correção é introduzida depois de observar a saída da rede este é um caso de ensino supervisionado.

- **Aprendizado não Supervisionado** é quando para fazer modificações nos valores das conexões sinápticas não se usa informações sobre se a resposta da rede foi correta ou não. Usa-se por outro lado um esquema, tal que, para exemplos de coisas semelhantes, a rede responda de modo semelhante. Aprendizado não supervisionado se chama também

descobridor de regularidades ou *redes auto-organizadas* devido à propriedade básica de seu funcionamento.

O ensino supervisionado tem atraído atenção de muitos pesquisadores. Uma motivação talvez seja o fato que o aprendizado supervisionado pode ser encarado como um problema de otimização e usar ferramentas que já mostraram sua eficácia, tanto em programação linear e não linear. Basta para isso considerar o aprendizado com a minimização do erro entre a saída da rede e uma saída desejada. Entretanto pouco se usou até o momento em ferramentas de otimização de sistemas dinâmicos para treinamento de RNAs, tais como Programação Dinâmica e o Teorema do Máximo de Pontriagin.

5.1.3 Classificação quanto à Finalidade do Aprendizado

Um terceiro modo de classificação é quanto à finalidade do aprendizado. Assim temos um *auto associador*, um *hetero associador* ou um *detetor de regularidades*.

- Em um *auto-associador* uma coleção de exemplos é apresentada à rede, a qual é suposta memorizar os exemplos. Depois, quando um destes exemplos for novamente apresentado de modo deteriorado supõe-se que a rede restitua o original sem deterioração. Neste caso aprende-se a funcionar como um filtro.
- Um *hetero-associador* é uma variante do auto-associador em que se memoriza um conjunto de pares. O sistema aprende a reproduzir o segundo elemento do par mesmo que o primeiro se já apresentado de modo contendo pequenas alterações. Este hetero-associador é também conhecido como um reconhecedor de padrões, onde o primeiro elemento apresentado é elemento a reconhecer e o segundo um elemento do conjunto de padrões considerado.
- O *detetor de regularidades* é um reconhecedor de padrões em que os padrões possíveis não é definido a priori. O sistema deve se auto-organizar, e criar os possíveis padrões.

5.2 Regras de Aprendizado Conexionistas

Essencialmente o aprender de redes neurais consiste em colocar valores de conexões sinápticas. Em alguns casos estes valores são colocados representando um certo conhecimento, como no caso usado em sistemas especialistas. Em outros usa-se uma algoritmo para encontrá-los. A este algoritmo chama-se *algoritmo de aprendizagem*.

5.2.1 Aprendizado Hebbiano

A lei de Hebb, mencionada precedentemente é talvez a mais antiga regra de aprendizagem usada. Uma extensão desta lei é:

Ajuste a intensidade da conexão entre os neurônios A e B de uma quantidade proporcional ao valor da ativação simultânea dos dois neurônios. Se no entanto A tenta excitar B e não consegue a conexão é enfraquecida.

Uma característica importantíssima da lei de Hebb é que ela goza da *propriedade de localidade*. Isto significa que para alterar o valor de uma conexão sináptica apenas informações locais à sinapse em jogo são usadas dando plausibilidade biológica ao algoritmo.

Assim tem-se:

$$\Delta w_{ij} = \eta x_i o_j \quad (2)$$

onde:

w_{ij} intensidade da conexão entre os neurônios i e j

Δw_{ij} acréscimo da intensidade da conexão entre os neurônios i e j

η parâmetro definindo a intensidade da correção chamado *taxa de aprendizado*

x_i estado de ativação do neurônio i

o_j saída do neurônio j

Apesar da plausibilidade biológica o uso da lei de Hebb nesta forma apresenta vários inconvenientes costumando-se usar versões mais sofisticadas, tais como a Regra Delta.

Muitos pesquisadores em RNA não se preocupam com plausibilidade biológica e em muitos casos tem razão. Afinal aviões voam melhor que pássaros e os primeiros tiveram asas inspirados nos pássaros. E voam melhor por não terem de bater asas e sim usarem motores a jato, desconhecidos como solução biológica. Seria mesmo possível dizer que, se o objetivo é obter um artefato com propriedades de generalização, capazes de aprenderem a realizar tarefas mal definidas, a plausibilidade biológica pode ser desnecessária. Exemplos que se enquadram neste caso são inúmeros.

Entretanto suponha-se, por exemplo que se esteja interessado em um melhor compreensão de mecanismos envolvidos na inteligência. Usa-se um algoritmo que não é biologicamente plausível e tira-se conclusões sobre mecanismos mentais! Que validade terão estas conclusões se foram obtidas usando algo que é irreal? mesmo que os resultados reproduzam coisas que ocorrem no aprendizado de seres vivos o que se deveria concluir não é que se tem uma idéia mais precisa de como o aprendizado ocorreu, É que o aprendizado nada mais é do que criar um sistema com um comportamento determinado. Mas muitos sistemas internamente diferentes podem apresentar o mesmo comportamento externamente, logo o comportamento biológico que continua ignorado

5.2.2 Regra Delta

A expressão usada na Lei de Hebb é muito simplificada. Com efeito, considerando uma sinapse real tem-se:

- O valor da modificação da intensidade da conexão sináptica para mesmas excitações dos neurônios envolvidos pode variar com o tempo.
- A modificação da intensidade da conexão sináptica Δw_{ij} de w_{ij} pode depender de w_{ij} o que será um efeito não linear (anteriormente tem-se um sistema bi-linear por apresentar um produto de duas intensidades). Isto ocorre como um efeito de saturação do valor de conexão sináptica.
- Pode-se imaginar que a modificação da intensidade da conexão sináptica dependa também de neurônios vizinhos.

Um modelo um pouco mais completo seria:

$$\Delta w_{ij} = \Phi(x_i, d_i, w_{ij}, o_j, t) \quad (3)$$

Note-se que d_i não é local à sinapse, mas local com relação ao neurônio i .

Uma expressão um pouco mais simples seria obtida com a função Φ dependendo da diferença entre a excitação real do neurônio i e a que seria desejada. Com isto tem-se a *Regra de Widrow-Hoff* ou *Regra Delta* que pode ser expressa como:

$$\Delta w_{ij} = \eta(d_i - x_i)o_j \quad (4)$$

Existem muitas outras variantes para a regra de aprendizado expressa pela equação 3. Por exemplo, Grossberg propoz a seguinte regra de aprendizado:

$$\Delta w_{ij} = \eta x_i(o_j - w_{ij}) \quad (5)$$

Nesta regra de aprendizado toda a informação necessária à modificação de uma sinapse é local à sinapse, sendo portanto um regra de aprendizado plausível biologicamente.

É interessante notar que a regra delta implementa uma otimização em H_2 isto é, minimiza o erro médio quadrático. Com efeito, seja o erro médio quadrático correspondente ao exemplo 'p':

$$E_p = \frac{1}{2} \sum_j (d_{pj} - o_{pj})^2 \quad (6)$$

Este erro é função do valor das conexões e portanto para calcular seu mínimo deve-se encontrar os valores que anulam a primeira derivada. Assim tem-se:

$$\frac{\partial E_p}{\partial w_{ij}} = \frac{\partial E_p}{\partial o_{ip}} \partial o_{ip} \partial w_{ij} \quad (7)$$

$$= -(d_{pj} - o_{pj}) \frac{\partial o_{pj}}{\partial w_{ij}} \quad (8)$$

Considerando a aproximação de primeira ordem para a função $o = \Upsilon(w)$ e tomando o coeficiente de proporcionalidade como uma constante (o sinal negativo é escolhido para ter o mesmo sinal da expressão usada na regra delta) η , tem-se:

$$\frac{\partial E_p}{\partial w_{ij}} = \eta(d_{pj} - o_{pj}) \quad (9)$$

A Regra Delta é biologicamente plausível pois usa apenas informação local à sinapse para o aprendizado. Seu ponto de partida é a generalização da Lei de Hebb e efetua uma otimização que pode ser interpretada como o modelo matemático de um mecanismo de seleção.

5.2.3 Retropropagação

Retropropagação (“Backpropagation”) pode ser considerada como a generalização da Regra Delta para redes diretas com mais de duas camadas. Neste caso, ao menos uma camada de neurônios não está envolvida com a entrada ou saída e é portanto interna à rede. Esta camada e suas conexões quando aprendem a efetuar uma função, agem como se houvesse uma representação interna da solução do problema.

Para uma apresentação de como a retropropagação funciona recomenda-se a leitura do artigo original de Rumelhart et al. [81]. Sem entrar em detalhes a retropropagação é uma regra de aprendizado supervisionado. Apresenta-se à rede um exemplo e verifica-se a saída da rede, saída esta que é comparada à saída esperada dando um erro. Calcula-se o gradiente deste erro com relação aos valores sinápticos da camada de saída que é atualizada por um passo escolhido podendo-se então calcular o erro da saída da penúltima camada, e assim por diante propagando para trás o erro (origem do nome backpropagation) por todas as camadas de conexões. A seguir apresenta-se mais um exemplo (pode ser o mesmo repetido) e continua-se o processo até que o erro seja menor que uma tolerância desejada.

Esta é talvez a mais popular regra de aprendizado. A maioria dos programas para tratar RNA dispõem de uma implementação da backpropagation ou na forma original (usando gradiente) ou em uma forma modificada para melhorar a performance da regra. Além de ser a primeira regra inventada para efetuar treinamento supervisionado de redes diretas com mais de duas camadas e conseqüentemente não limitada a resolver problemas linearmente separáveis, tem ainda a vantagem de que, se a função de ativação for uma função analítica derivável, a derivada pode ser calculada explicitamente, evitando todos os problemas inerentes à derivação numérica. Tal é o caso da função logística.

$$y_{pi} = \frac{1}{1 + e^{-(\sum w_{ji}y_{pj} + \theta_j)}} \quad (10)$$

onde θ_j é a polarização (‘bias’).

Com efeito, chamando a expressão entre parêntesis na equação 10 net_{pi} a derivada é:

$$\frac{\partial y_{pj}}{\partial net_{pj}} = y_{pj}(1 - y_{pj}) \quad (11)$$

Conseqüentemente o erro em uma unidade de saída é dado por:

$$\delta_{pj} = (d_{pj} - y_{pj})y_{pj}(1 - y_{pj}) \quad (12)$$

e em uma unidade interna:

$$\delta_{pi} = y_{pj}(1 - y_{pj}) \sum_k k \delta_{pk} w_{kj} \quad (13)$$

O máximo da derivada ocorre para $y_{pj} = 0.5$ e o mínimo para 0 ou 1. Como na backpropagation a correção a cada passo é proporcional à derivada, os pesos mudam mais para as unidades que tem uma ativação de valor intermediário o que contribue para a convergência do algoritmo.

Convem ainda notar que retropropagação pode ser usada (teoricamente) para neurônios onde as entradas são combinadas usando multiplicação, para redes com realimentação ou recorrentes. No primeiro caso, como observado por Janson & Frenzel [51] a função erro pode apresentar muitos mínimos locais e não deve ser usada.

Finalmente cabe salientar que nem toda rede direta pode ser treinada pela retropropagação, pois devido ao cálculo da derivada é necessário que a função de ativação seja derivável (o Perceptron está excluído). Por outro lado é possível treinar redes com ciclos por retropropagação, como mostrado no artigo mencionado acima. Portanto, é impreciso chamar redes diretas de redes retropropagação como se faz freqüentemente na literatura.

5.2.4 Aprendendo com a Natureza

Em seu interessantíssimo livro, “Le hasard et la nécessité”, Monod [73] seguindo idéias Darwinianas [24], discute como aleatoriedade e luta pela vida servem para os seres vivos aprenderem e com isto evoluírem. A mesma idéia pode ser usada no treinamento de RNAs. Aplicada a populações esta idéia leva aos algoritmos genéticos. Aqui será apresentada a versão usada em estudo de manter a posição em pé.

O processo é o corpo da criança de pé. A entrada é o desejo de ficar nesta posição. As perturbações são do tipo vento, algum peso carregado, etc. A realimentação modela os sentidos. Incorpora-se informações visuais, do labirinto, proprioceptivas relativas às posições das juntas, etc. As informações do labirinto incluem as provenientes de órgãos vestibulares tais como os canais semicirculares capazes de detetar aceleração angular e os órgãos otolíticos capazes de detetar aceleração linear.

Usou-se rede direta como controlados. O comportamento global do sistema controlado é função do valor das conexões que são inicializadas aleatoriamente como uma matriz $W(0)$ e observa-se o comportamento durante um certo intervalo de tempo registrando-se o erro médio

quadrático $\epsilon(0)$. A seguir gera-se uma outra matriz aleatória ΔW criando uma nova matriz de conexões (η é um fator de correção):

$$W(1) = W(0) + \Delta W \cdot \eta \quad (14)$$

Simula-se novamente o sistema. Se o novo erro ϵ is for menor que $\epsilon(0)$, a nova matriz sináptica é adotada como $W(1)$. No caso contrário começa-se novamente tirado outra matriz aleatória ΔW .

Repete-se o mesmo algoritmo ate melhorar e atingir erro menor que uma tolerância quando se aumenta o tempo de observação.

O processo se repete até que apenas pequenas oscilações em torno da posição em pé sejam observadas.

5.2.5 Aprendizado Competitivo

No aprendizado competitivo, usado nas redes popularizadas por Kohonen [55] neuronios são inibidos por outros neuronios de modo a que a competição entre eles leva a apenas um acabar excitado. Assim, enquanto uma rede neural baseada em um aprendizado Hebbiano, vários neurônios de saída podem estar simultaneamente ativos, no caso do aprendizado competitivo, somente um neurônio de saída fica ativo de cada vez.

Fundamentalmente existem tres elementos que caracterizam o aprendizado competitivo:

1. Existe um conjunto de neurônios idênticos, ligados por valores de conexões sinápticas de valores distribuidos de modo aleatório.
2. Existe um valor máximo bem definido para a ativação dos neurônios.
3. Existe um mecanismo que permite que os neurônios entrem em competição pelo direito de permanecerem excitados.

No aprendizado competitivo entradas possuindo alguma semelhança tendem a excitar o mesmo neurônio na saída. Assim é que este paradigma de aprendizado pode servir para sugerir classificações, tal como estudado no caso do estudo do sono [21], [22], [20], [23] em que os vários tipos de sono é ainda assunto de debate.

5.2.6 Aprendizado Reforçado

No *Aprendizado Reforçado* ou “Reinforcement learning” consiste no aprendizado através do método da tentativa e erro de modo a otimizar um índice de performance chamado *signal de reforço*.

Este paradigma de aprendizado tem profunda motivação biológica, em que comportamentos provocando satisfação tem como consequencia um reforço das conexões que os produziram, e

aqueles provocando insatisfação uma modificação do valor das correspondentes conexões. Um exemplo é o estudo do controle e seu aprendizado da posição erecta [17].

5.2.7 Aprendizado Aleatório

O aprendizado é dito aleatório quando os passos no sentido de obter o comportamento aprendido se baseiam em valores tomados aleatoriamente que são testados para verificar sua adequabilidade. Assim, em essência o aprendizado aleatório segue os seguintes passos:

- Selecione os valores das conexões sinápticas de modo aleatório.
- Verifique o valor da performance da rede.
- Provoque uma variação aleatória nas conexões sinápticas e verifique o novo valor da performance da rede. SE melhorou retenha este novo valor de conexões. Caso contrário escolha um critério para escolher noiva variação.
- Verifique se um critério de parada especificado inicialmente foi satisfeito e neste caso pare o aprendizado.

De uma certa forma o aprendizado aleatório coincide com o aprendizado com a Natureza, mencionado acima.

5.2.8 Aprendizado Evolutivo

Aprendizado Evolutivo é o paradigma de aprendizado que, tirando inspiração da evolução biológica é capaz de modificar a topologia e os valores das conexões sinápticas de modo a fazer uma rede se tornar apta a resolver um problema. Este assunto pode servir tanto como algoritmo de aprendizado como para determinar a topologia da rede a ser usado para resolver determinado problema [7, 38].

6 Máquina de Resolver Problemas

O sucesso das RNAs faz crer que um computador usando estas redes, como bloco básico, possa resolver problemas que computadores que não usem esta tecnologia são incapazes, ou ao menos, teriam muita dificuldade para resolver. Isto nos leva a um estudo comparativo de possibilidades e torna-se conveniente precisar a terminologia empregada.

6.1 Tipos de Computadores

O estudo das possibilidades de RNAs na solução de problemas implica na existência de computadores usando estas redes. Chama-se *neurocomputador* um computador em que o funcionamento interno é feito por redes neurais munido de dispositivos de entrada e saída [44], [45].

Por outro lado, devemos mencionar computadores que não usam esta tecnologia, que possuem uma ou mais unidades centrais de processamento, memória.

Muitos destes computadores que não podem ser chamados de ‘convencionais’ por incluírem alto grau de sofisticação têm em comum o fato que seu funcionamento se baseia no conceito de *instrução*. Por esta razão eles serão chamados de *Computadores Baseados em Instruções* ou CBI como proposto em [5].

Em alguns casos pode ser conveniente usar simultaneamente as duas tecnologias tendo-se então um computador híbrido.

6.2 Teoria de Problemas

Informalmente pode-se dizer que IA serve para resolver problemas, imitando de uma certa forma a inteligência dos seres vivos (geralmente seres humanos). Mas o que é um problema? [6].

A palavra “problema” é tão familiar que pode até parecer estranho perguntar sobre seu significado. Durante muito tempo se tratou de problemas, achou-se a solução de muitos, provou-se que existem ou não solução para muitos problemas, e muitos ainda desafiam a ciência. Polya [74] sugere que antes de tentar buscar a solução de um problema procure-se responder as seguintes perguntas:

- Quais são os dados?
- Quais são as soluções possíveis?
- O que caracteriza uma solução satisfatória?

Estas perguntas podem ser tomadas como guia para formalizar a noção de problema [86]:

Definição 15 *Um problema é o objeto matemático $P = \{D, R, q\}$, consistindo de dois conjuntos não vazios, D os dados e R os resultados possíveis e de uma relação binária $q \subset D \times R$, a condição, que caracteriza uma solução satisfatória.*

Para ilustrar este conceito seja o caso de encontrar as raízes de um polinômio com coeficientes reais. Achar a solução do problema da busca das raízes de um polinômio com coeficientes reais consiste em associar a cada conjunto de coeficientes de um polinômio particular $p(x)$ de grau n , n números complexos c_n de modo a satisfazer a condição de que o valor de $p(x)$ fazendo $x = c_n$ para todo n seja nulo. Define-se então a solução de um problema como a função $f : D \rightarrow R$, tal que $\forall d \in D$ tem-se $(d, f(d)) \in q$.

O conceito de problema apresentado se refere a problemas em geral e não a exemplos específicos de problemas. Por exemplo achar as raízes do polinômio $2x^5 + 3x^2 + 3$ é um caso particular do problema de achar raízes de um polinômio de coeficientes reais.

Como a solução a um problema é uma função, se for possível implementar esta função tem-se a solução do problema. Este fato leva, na abordagem simbólica, à programação funcional e é a base da abordagem conexionista.

Com efeito, suponha-se implementada uma máquina abstrata tendo como primitivas um conjunto bem escolhido¹⁰ de funções, além de um mecanismo de construção de novas funções (recursividade seria um destes mecanismos). A função solução do problema poderia ser implementada em tal máquina, e esta solução estaria usando o paradigma funcional de programação.

Por outro lado, a abordagem conexionista é fundamentada no fato de que redes neurais, e em particular, redes diretas podem ser consideradas como aproximadores universais de funções [18]. Desta forma, basta criar uma rede direta, suficientemente rica e treiná-la para representar a função.

6.3 O Computador na Resolução de Problemas

O computador pode ser considerado como máquina de resolver problemas, logo, é natural imaginar que tanto a possibilidade de resolver um problema específico, como quanto vai ser gasto em recursos na tarefa, dependem da máquina usada. Ao fato de que um problema possa ser resolvido com recursos finitos chama-se *computabilidade* [53] e a quantidade de recursos envolvidos *complexidade*. Fala-se também em *computabilidade prática*; por exemplo, um problema que requeira um tempo de 100 anos do mais rápido computador disponível não é praticamente computável.

Para estudar se em um CBI e em um neurocomputador um problema é computável e qual a complexidade da solução, é necessário explicitar como um neurocomputador e um CBI enfrentam a tarefa de resolver problemas. Em ambos pode-se distinguir os três pontos seguintes:

a)-Em um CBI tem-se:

- 1 - o computador virtual (circuitos e programas),
- 2 - o ato de fazer o computador apto a resolver um problema específico (carregar o programa no computador),
- 3 - resolver o problema (rodar o programa).

b)-Em um neurocomputador tem-se:

- 1 - a rede de neurônios com entradas e saídas (simulado ou materialmente implementado),
- 2 - um meio de fixar os pesos das conexões, muitas vezes usando um algoritmo de aprendizagem (equivalente a carregar o programa),
- 3 - usar a rede já educada para resolver o problema com os dados a serem usados na entrada da rede (equivalente a rodar o programa).

A computabilidade de um problema depende dos pontos 1 e 2. Com efeito, a possibilidade de resolver um problema depende do apoio material que se dispõe e se existe um programa (caso de

¹⁰A expressão “bem escolhido” equivale, de modo informal a dizer “satisfazendo às seguintes condições:” e enumerar as condições necessárias para que a afirmação que se segue seja válida.

um CBI) ou se existe um conjunto de pesos de conexões (caso de um neurocomputador) capaz de resolver o problema. Por outro lado a complexidade do problema depende do ponto 3, ou seja rodar o programa ou excitar a rede com os dados a serem usados.

Se existe uma teoria de computabilidade e complexidade bem desenvolvida, esta teoria é voltada para os CBI. Com relação aos neurocomputadores existem apenas resultados isolados.

6.4 Equivalência de Computabilidade

A Tese de *Church-Turing* diz que todo problema computável pode ser resolvido por máquina de Turing. Se as redes neurais são ou não equivalentes a uma máquina de Turing e em conseqüência são capazes de resolver qualquer problema computável e apenas eles, tem despertado grande interesse recentemente.

Visto a luz dos trabalhos publicados por Arbib [2], [3] pode-se dizer que em termos de computabilidade CBI e neurocomputadores são equivalentes. Isso quer dizer que um neurocomputador não sabe resolver nenhum problema que não pudesse ser resolvido com um CBI e vice versa. Esta afirmação pode ser descrita mais precisamente por dois teoremas.

Theorema 1 *Todo problema que pode ser resolvido por uma rede neural poderá ser resolvido, com a precisão que se desejar, por um CBI.*

A prova deste resultado é fácil [2, 3]. Com efeito, sabe-se que toda RNA pode ser simulada em um CBI, geralmente usando um programa que efetua multiplicações matriciais, implementa funções, etc. E isto com a precisão desejada. Usando esta simulação e os dispositivos de entrada e saída do CBI tem-se um neurocomputador (simulado). Ora todo problema que este neurocomputador resolve está na realidade sendo resolvido por um CBI, que é a máquina hospedeira da simulação. Assim, pode-se dizer que todo problema resolvido por um neurocomputador pode ser resolvido por um CBI. ■

A recíproca deste teorema é:

Theorema 2 *Todo problema que pode ser resolvido por um CBI poderá ser resolvido, por uma RNA munida de convenientes dispositivos de entrada e saída.*

Com efeito, usando neurônios artificiais (e dos mais simples, aqueles que possuem apenas saídas binárias) é possível construir os circuitos lógicos ‘e’, ‘ou’ e ‘não’ além de circuitos bi-estáveis.

Com efeito, a figura 5 mostra como implementar estes circuitos básicos. O primeiro, implementando um circuito **não** consiste em um neurônio com entrada feita através de uma conexão inibitória, de valor sináptico unitário. O neurônio é, como em todos os três considerados na figura, definido por:

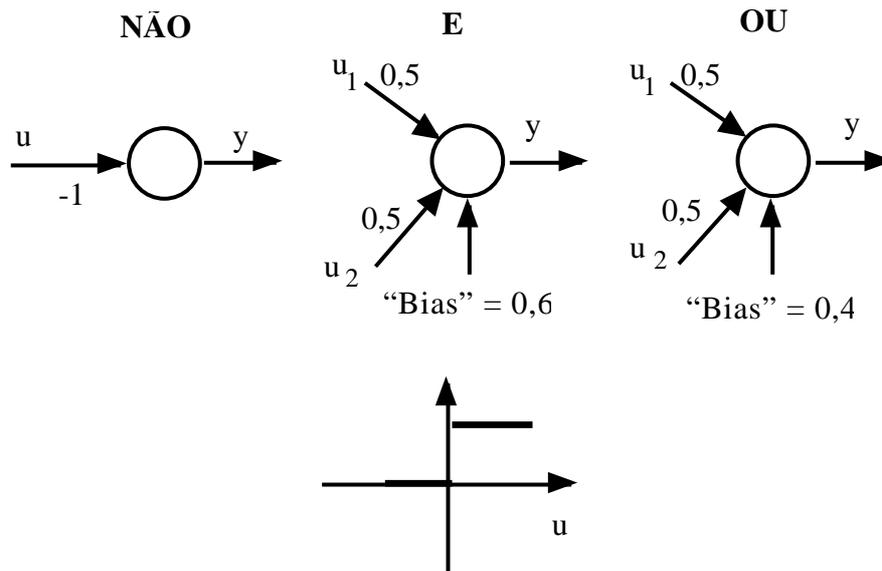


Figura 5: Circuitos **NÃO**, **E**, **OU** implementados com neurônios artificiais.

- entrada do neurônio u_T : soma dos produtos das excitações u pelos valores das conexões sinápticas;
- uma entrada auxiliar, chamada "bias" que é usada para alterar a função de saída do neurônio;
- neurônio estático, isto é, a saída ocorre no mesmo instante que ocorre a entrada (o tempo entre estes dois eventos é muito pequeno);
- a saída é obtida considerando a função de saída do neurônio, dada pela expressão abaixo onde u_T leva em consideração o "bias":

$$\text{se } u_T < 0 \text{ então } y = 0 \text{ senão } y = 1$$

Com estes circuitos pode-se construir um CBI [32] usando técnicas que os engenheiros de microeletrônica conhecem e juntando-se dispositivos de entrada e saída. Ora, seja um CBI, assim construído, programado para resolver um problema específico. Este problema estará sendo resolvido por um conjunto de neurônios artificiais ligados de um certo modo, logo por uma RNA. Logo, pode-se concluir que todo problema que pode ser resolvido por um CBI pode ser resolvido por um neurocomputador. ■

Desta forma fica provada, de modo intuitivo, a equivalência dos CBI e neurocomputadores em termos de computabilidade. Prova formal foi apresentada em [66] para o caso de neurônios com dois estados.

6.5 Pontos de Dúvida

Para concluir esta discussão sobre a equivalência de redes neurais artificiais (RNA) e máquina de Turing, é conveniente apresentar argumentos que mostram que esta equivalência é apenas aproximada.

- Existem RNAs cujos valores de ativação, saída e entradas são elementos de um intervalo dos números reais. Portanto a cardinalidade destes conjuntos é \aleph_1 , ou seja a cardinalidade do contínuo. Por outro lado, no caso da máquina de Turing, tem-se cardinalidades finitas, ou seja, conjuntos enumeráveis (memória interna tão grande quanto se deseje) sendo a cardinalidade no máximo \aleph_0 , ou seja a cardinalidade dos inteiros. Ora, como as cardinalidades são diferentes, não existe bijeção entre estes conjuntos, sendo as RNAs mais ricas que a máquina de Turing. Uma RNA pode ser simulada em máquina de Turing levando em consideração uma certa aproximação.
- No caso de neurônios naturais (ou biológicos), formadores das redes neurais naturais (RNN) ou redes neurais biológicas (RNB), a observação referente à diferença de cardinalidades deixa de ser válida. Com efeito, a transmissão de informação sináptica se faz por moléculas de neurotransmissores e, portanto, é quantizada, tendo cardinalidade finita ou no máximo \aleph_0 . Resta a dúvida com relação ao conjunto de excitações possíveis: será que a frequência de descarga dos neurônios pode variar continuamente, ou dará saltos entre frequências próximas?
- A maior parte das RNAs com que se trabalha, são redes síncronas. Ora, se isto é semelhante aos CBI que trabalham sincronizados por um relógio¹¹ tal não ocorre com as RNN em que o funcionamento é assíncrono.

Destas observações pode-se tirar várias conclusões, dentre as quais os teoremas e o corolário que existem redes neurais que não podem ser implementadas em CBI. Conseqüentemente existem problemas que podem ser resolvidos por neurocomputadores que não podem ser resolvidos pela Máquina de Turing.

O fato de uma rede neural suave poder ser aproximada com a precisão desejada por uma outra rede discreta, leva a definir uma equivalência em termos de computabilidade prática, dois computadores sendo equivalentes, são capazes de resolver os mesmos problemas. Entretanto, em termos de computabilidade prática, neurocomputadores e CBI são equivalentes.

6.6 Não Equivalência de Complexidade

Nas seções precedentes foi discutido o problema de computabilidade de CBI e neurocomputadores chegando-se a uma equivalência prática em termos de computabilidade. E em termos de complexidade?

¹¹Claro que em computação distribuída tal não é verdade.

Embora pouco seja conhecido sobre complexidade quando se usa um neurocomputador, sabe-se que em termos de complexidade as coisas são diferentes em termos de CBI e neurocomputadores.

Para ver que as coisas são diferentes basta considerar um exemplo simples. Seja o caso de um circuito implementando uma RNA direta síncrona com três camadas. Suponha-se ainda que ela foi treinada para associar padrões (por exemplo um sistema especialista de diagnóstico associando sintomas aos correspondentes diagnósticos). Como não há realimentações, o tempo para a rede dar uma resposta é sempre o mesmo: três ciclos de relógio! E isso para qualquer número de sintomas e doenças.

Ora, este resultado é completamente diferente se for usado um CBI. Por exemplo, se for usado Prolog, o tamanho da árvore de busca aumenta com o número de sintomas e diagnósticos. Poder-se-ia retrucar que no caso do neurocomputador, para muitos sintomas/diagnósticos possíveis o tamanho da rede aumenta, e isto é verdade ao menos em parte. Porque se uma rede maior pode fazer crer em um resultado mais preciso daquele que obtém se for usada uma rede menor, não se sabe ainda ao certo qual o tamanho da rede ideal para resolver um dado problema. Além disto, esta afirmação sobre precisão é baseada em resultados anteriores e pode ser considerada como uma heurística.

Existem muito poucos estudos sobre complexidade de problemas voltada para a solução por neurocomputadores. Note-se que não se trata de falar da complexidade de problemas ligados às RNA, tais como a complexidade de um algoritmo de aprendizado de redes.

A complexidade a que o presente estudo se refere é a quantidade de recursos em termos de RNA necessários para poder resolver um determinado problema, eventualmente considerando-se uma certa precisão. Estes recursos incluem o tipo de rede a ser escolhido, a topologia da rede, etc.

Um estudo que se tornou famoso foi o apresentado por Minsky e Papert em 1969 [71], [72] e que desviou a maioria das pesquisas de redes neurais para pesquisas em IA usando manipulação simbólica. Além deste trabalho, que pode ser considerado o primeiro grande passo na direção de uma teoria da complexidade voltada para RNA, alguns outros trabalhos isolados têm aparecido. Para exemplos, ver as referências que se seguem: [1], [67].

6.7 Alguns Resultados Sobre Complexidade de RNA

A complexidade das RNA diz respeito a dois pontos:

1. Dado um problema, definir a topologia da rede necessária para sua solução. Por exemplo, se a rede deve agir como modelo de um sistema de controle adaptativo, e a entrada da rede é a saída do processo, será possível usar uma rede direta?
2. Dado um problema que pode ser resolvido por uma determinada topologia de RNA, e uma precisão desejada, qual o tamanho mínimo da rede que deve ser usada?

Alguns teoremas com relação à complexidade das RNAs foram apresentados em [7] que permitem sugerir uma classificação baseada em separabilidade e dinâmica.

1. Problemas estáticos linearmente separáveis.

Trata-se de problemas envolvendo a implementação de uma função (por ser um problema estático) e que podem ser resolvidos por um perceptron de uma camada de conexões.

2. Problemas estáticos linearmente não separáveis.

Trata-se de problemas envolvendo a implementação de uma função (por ser um problema estático) e que podem ser resolvidos por uma rede direta, com neurônios estáticos, exigindo ao menos uma camada de neurônios internos.

3. Problemas dinâmicos com dinâmica finita.

Os problemas com dinâmica finita são aqueles que a duração da resposta do sistema após uma entrada dura um tempo finito. Um exemplo são os filtros FIR (“Finite Impulse Response”). Estes problemas pode ser resolvidos por rede direta com neurônios dinâmicos.

4. Problemas dinâmicos com dinâmica infinita.

Os problemas com dinâmica infinita são aqueles que a duração da resposta do sistema após uma entrada pode durar um tempo infinito. Um exemplo são os filtros IIR (“Infinite Impulse Response”). Estes problemas devem ser abordados por rede com retroação e com neurônios ou rede estática e conjunto de retardos. Neste caso o problema da estabilidade da rede, ou seja se a rede encontrará ou não solução e quanto tempo será necessário é problema em aberto.

6.8 Aprendizado de RNA como Paradigma de Programação

Será que a diferença em complexidade justifica esforços para construir neurocomputadores? A resposta pode ser encontrada no modo de fazer neurocomputadores aptos a resolver problemas.

O ato de fazer um CBI apto para resolver um problema é bem conhecido como a atividade de programar. Programar pode ser considerado como o ato de descrever um algoritmo ou como meio de se comunicar com o computador. Como modo de descrever um algoritmo fica impróprio falar em “programar um neurocomputador”, mas não se for considerado como o modo de se comunicar com ele. Assim considere-se o conceito de “programar” de um modo mais amplo que seja aplicável também aos neurocomputadores. Ora, neurocomputadores se preparam para resolver problemas ajustando os valores das conexões sinápticas entre seus neurônios, o que pode ser feito essencialmente de dois modos:

- Colocando diretamente os valores como uma representação do conhecimento com sugerido no livro de exercícios da série PDP [82], exemplo das gangs Jets e Sharks e que foi

modificado para uso em sistemas especialistas conexionistas com sucesso [25], [15], [13], [11].

- Usando um algoritmo de aprendizagem [46].

Mas se isso é também programar, que paradigma de programação será este?

É costume mencionar vários diferentes paradigmas de programação: imperativa, funcional, lógica, declarativa, orientada a objeto, etc. Estes paradigmas não são excludentes, existindo, por exemplo, funcional orientada objeto, imperativa com parte declarativa, etc. Entretanto, considerando a proximidade com o funcionamento de um CBI pode-se falar em imperativa e declarativa.

Na programação imperativa se especificam as ações que o computador deve efetuar para resolver determinado problema. Na programação declarativa declara-se o que deve ser uma solução para o problema e cabe ao computador transformar esta definição do problema em instruções imperativas a serem executadas. Por exemplo, um interpretador Prolog recebe um programa Prolog em estilo quase declarativo e o transforma, usando entre outras coisas o Princípio da Resolução proposto por Robinson [75] em um programa imperativo.

E programar um neurocomputador? Isso pode ser considerado como um novo paradigma de programação, em que não é mais necessário nem definir o algoritmo nem definir o problema precisamente. Basta introduzir no computador relações entre conceitos ou usar exemplos de problemas parecidos já resolvidos para serem usados na fase de aprendizado. A RNA, usando sua capacidade de generalização se torna capaz de resolver o problema desejado. Até o presente momento não se conhece teoria permitindo associar a precisão da solução a esta capacidade de generalização, o problema tendo sido abordado de modo experimental, usando simulação [76], [77].

6.9 Um Pouco de Ficção

Uma discussão de neurocomputadores e computadores híbridos motiva especulações. Quem não gostaria de um robô doméstico? Poderia limpar a casa, por a mesa, por a louça na máquina, trazer os chinelos depois de abrir a porta para seu mestre, e com as capacidades neurais de aprender e se adaptar a novas manias do mestre. Mas ele deveria aceitar programação declarativa também para que se pudesse colocar, por exemplo, as leis da robótica inventadas por Azimov... e nunca fazer mal a um ser humano.

7 Aplicações das RNAs

Atualmente as aplicações das RNAs estão invadindo todos os domínios, saindo das primeiras em reconhecimento de padrões, para ir a distribuição de energia elétrica, mercado de capitais,

aplicações navais, sistemas especialistas, etc. Neste capítulo serão abordadas algumas destas aplicações.

7.1 Reconhecimento de Padrões

7.1.1 Em que Consiste o Reconhecimento de Padrões

Reconhecimento de padrões é talvez uma das primeiras aplicações de redes neurais. Com efeito, o Perceptron de Rosenblatt [78, 79] foi concebido principalmente como instrumento capaz de reconhecer letras. A principal razão é que reconhecimento de padrões é uma tarefa geralmente desempenhada muito melhor usando as capacidades cognitivas do homem do que executando um algoritmo. Por exemplo, seres humanos são excelentes no reconhecimento de rostos, músicas, a caligrafia de alguém conhecido, etc. Cães são excelente em reconhecer odores e gatos são capazes de sentir o humor de pessoas fugindo daquelas que exprimem características agressivas. Isso pode ser atribuído a um sistema bastante desenvolvido de reconhecimento de padrões.

Por outro lado os esforços para fazer computadores baseados no conceito de instrução tem encontrado sérias dificuldades.

7.1.2 Reconhecimento de Padrões como Classificador

A figura 6 representa esquematicamente um reconhecedor de padrões. O transdutor é munido de um sensor que traduz a forma de energia suporte de informação sobre o objeto (ex: foto-elétrica ou células da retina se informação visual, terminações nervosas do ouvido interno ou microfone se informação sonora) e traduz esta forma de energia para outra capaz de ser processada (neurotransmissores e sinais elétricos de sistema biológico ou elétricos de circuitos artificiais). O processamento inclui geralmente uma primeira fase em que atributo relevantes são selecionados para processamento e este processamento age como uma função, associando ao valor de um conjunto de atributos relevantes um elemento de um conjunto de padrões possíveis, o qual é apresentado como resposta do classificador.

O processamento de sinais visuais tem grande importância na implementação de robôs autômos e requerem um processamento maciço. Devido a isto esta tem sido desde algum tempo a principal motivadora da implementação de RNAs em pastilhas de silício usando tecnologia VLSI [70].

O paradigma mais comum de aprendizado no caso do reconhecimento de padrões é o supervisionado, associado a uma rede direta multi-camadas. Devido a sua disponibilidade, a regra da retro-propagação é frequentemente usada bem como suas variantes. Entretanto bons resultados são obtidos também com o aprendizado competitivo tipo redes de Kohonen. Este último é principalmente interessante quando não se sabe quantas classes possíveis existem a identificar o que não é o caso do reconhecimento de padrões.

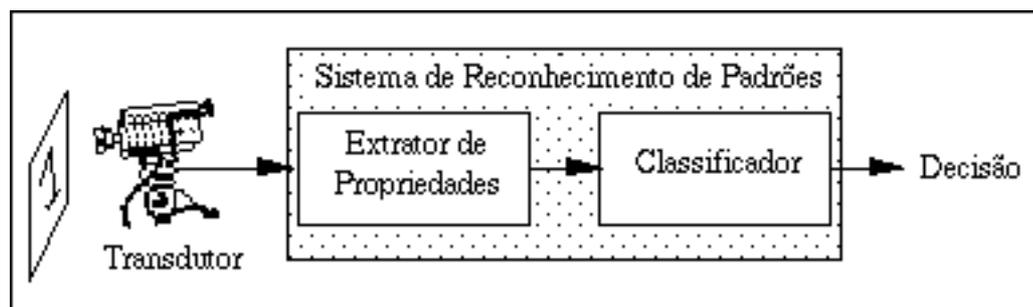


Figura 6: Visão de um sistema de reconhecimento de padrões.

7.1.3 Principais Aplicações

7.1.4 Reconhecimento de Caracteres

Reconhecimento de caracteres é uma aplicação bem sucedida de redes neurais desde o Perceptron de Rosenblatt. Muitos programas incluem alguma forma de reconhecimento de caracteres como programa de demonstração.

Normalmente a letra é apresentada a uma grade tal como a retina. Cada célula da grade serve de entrada a um neurônio da rede. Neurônios de saída da rede são associados a letras.

A camada de saída normalmente contém tantos neurônios quantos são as letras que se deseja identificar. Por exemplo para identificar todos os caracteres ASCII a rede terá 256 neurônios.

No caso apresentado, as letras são representadas na entrada da rede usando uma representação distribuída (a uma letra correspondem vários neurônios ativados) e na de saída uma representação localizada (a uma letra corresponde apenas um neurônio).

Durante a fase de treinamento apresentam-se sucessivamente as letras à rede em sua forma ideal e com algumas variações mais comuns. Depois de treinada, a rede identificará, aceitando letras com maior ou menor variação, as letras apresentadas.

Bastante sofisticado e dando resultados bastante bons, pode-se citar o *neocognitron* de Fukushima [34], evolução do *cognitron* [33]. Esta rede é do tipo multi-camadas. Permite reconhecer letras independentemente de sua posição e pode mesmo servir como modelo para o mecanismo biológico de reconhecimento de padrões visuais [35].

Uma variante do reconhecimento de caracteres é a filtragem de caracteres. Neste caso deseja-se poder apresentar na entrada da rede caracteres afetados de um ruído (mal representados) e ter na saída a mesma letra bem representada. Para este caso procede-se de modo inteiramente análogo com a diferença que a representação da saída é também distribuída.

O reconhecimento de caracteres foi também abordado com bastante sucesso por técnicas baseadas em manipulação simbólica, mas o reconhecimento de letras em posições diversas, o uso de caracteres distintos, o reconhecimento de letras manuscritas, por exemplo, continuam a ser

problemas que a abordagem simbólica encontra dificuldades em resolver.

7.1.5 Reconhecimento de Faces

Bem mais complexo que o reconhecimento de caracteres é o reconhecimento de faces. Esta aplicação pode, de modo análogo ao caso de reconhecimento de caracteres, ser usado para identificar a quem pertence um rosto visto em posições variadas e também reconstituir um rosto visto apenas parcialmente. Finalmente os dois problemas podem ser associados no reconhecimento de um rosto apresentado parcialmente. Os trabalhos de Kohonen são notáveis. Por exemplo, Kohonen [55] chega a mostrar o caso de um rosto que a rede sugeriu corretamente que o rosto deveria ser de um homem de óculos!

O reconhecimento de um rosto segue esquema semelhante ao do reconhecimento de uma letra. Note-se, no entanto, que geralmente o problema exige uma grade muito mais fina o que aumenta consideravelmente a quantidade de neurônios da rede.

O reconhecimento de faces ainda está em fase de pesquisa. Há no entanto um campo enorme de interesse ligado a problemas de segurança. Problema análogo é o da identificação de impressões digitais aplicação também em fase de pesquisa.

7.2 Sistemas Especialistas Conexionistas

7.2.1 SE Conexionistas

A IAS encontra dificuldades na solução de muitos problemas da vida real, devido a suas inconsistências, excessões e especificações incompletas.

É possível identificar dois tipos de domínios em que um SE pode atuar: artificiais e naturais. Se é geralmente fácil extrair regras para implementar um SE usando a IAS em sistemas pertencentes a um domínio artificial tal como panes de motores, é bastante difícil obter regras se o domínio é natural conhecido como no caso de diagnóstico médico, previsões financeiras, etc.

Um dos primeiros a usar sistemas especialistas conexionistas for Gallant (1988) [36] que considerou uma rede direta para a partir de 6 sintomas diagnosticar duas doenças e mostrou como é possível explicar o raciocínio para o caso estudado.

Posteriormente Azevedo [27, 28, 26, 25] mostrou como memórias associativas podem ser usadas em sistemas especialistas, trabalho continuado em [15, 13, 11]. Muito resta a fazer nestes dois paradigmas de implementação de sistemas conexionistas.

7.2.2 SE usando Redes Diretas

Nest caso consideram-se os exemplos disponíveis para o treinamento da rede por um algoritmo de aprendizado supervisionado. Como as redes diretas são capazes de aproximar uma função não linear qualquer, e imaginando que a solução do problema pode ser encontrada achando o valor da imagem de uma função, esta metodologia fornece solução satisfatória.

7.2.3 SE implementados com BAM

Neste caso supõe-se conhecidas explicitamente as relações entre sintomas e diagnósticos, tal como usual em um sistema especialista usando a IAS. Ora, é também possível incorporar exemplos.

Seja o caso de um diagnóstico médico. Inicialmente os conceitos são organizados em conjuntos de conceitos semelhantes: doenças, diagnósticos e pacientes.

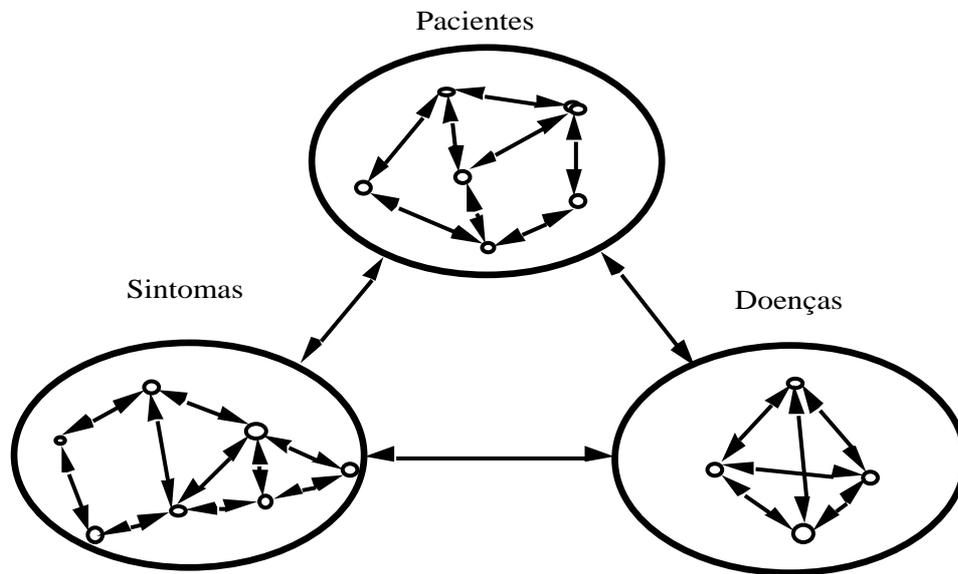


Figura 7: Conjuntos de doenças, diagnósticos e pacientes.

A cada conjunto de conceitos corresponde um cluster de neurônios. As conexões entre objetos são colocadas representando o que se conhece da relação entre dois objetos. Assim sabendo que um paciente apresenta um sintoma com uma certa gravidade, coloca-se a conexão entre este paciente e o sintoma com um valor que meça a gravidade do sintoma, ao menos subjetivamente.

Esta abordagem tem de interessante:

- Os exemplos são usados para colocar pesos nas conexões e não como elementos de um conjunto de treinamento.
- O conhecimento é localizado.
- O aprendizado se reduz à colocação dos pesos.
- A matriz sináptica é simétrica.
- A intensidade das conexões varia normalmente no intervalo $[-1 \ 1]$ representando o valor nebuloso da importância da relação entre os dois objetos, sendo de $[-1 \ 0]$ para inibição e de $[0 \ 1]$ para excitação.

- Uma consulta é feita excitando neurônios representativos de conceitos conhecidos e examinando o que aparecem excitados quando a rede atinge o equilíbrio.

Nota: Este paradigma tem sido usado em fase de protótipo para diagnóstico em reumatologia.

7.3 Controle de Processos

7.3.1 Controle Neural

Diz-se que um sistema de controle é neural quando se usa de alguma forma uma rede neural como parte do controlador. Atualmente esse assunto tem interessado largamente a pesquisa, principalmente no caso em que o modelo do processo a controlar não é disponível e se deseja usar as capacidades de aprendizado das RNA para obter a solução do problema. Para uma introdução ao assunto aplicada a problemas de robótica ver [12].

7.3.2 Topologias de Controle

Aqui se apresentam as topologias de controle mais comuns: malha aberta, realimentação, e modelo interno, com comentários de como uma RNA pode ser usada.

7.3.3 Malha Aberta

Trata-se da topologia mais simples possível. A finalidade é usada como entrada do sistema controlado e, se o controlador é escolhido como uma aproximação do modelo inverso do processo a controlar, a saída será uma aproximação da finalidade. Controle a malha aberta tem, geralmente, um funcionamento pobre, pois caso apareça uma perturbação no processo, o controlador não tem acesso a esta informação e não corrigirá o processo. Além disso toda imperfeição na implementação do modelo inverso se refletirá na saída do processo.

Uma das várias maneiras de treinar uma RNA para representar o modelo inverso é fazer vários ensaios com o processo a controlar obtendo pares entrada/saída. Depois usar estes pares como exemplos.

Neste ponto convém uma observação da maior importância, que, por sinal é frequentemente esquecida. Talvez a topologia de RNA mais utilizada seja a rede direta multi-camadas. Ora, se é bem sabido que esta rede pode ser treinada para aproximar uma função qualquer com uma precisão que depende do número de neurônios e topologia da rede, o processo a controlar frequentemente não pode ser representado por uma função! pois se trata de um *sistema dinâmico*. Logo é necessário usar uma RNA dinâmica o que se obtém usando uma rede com realimentação ou usando linha de retardo [25]. Além disso, o modelo inverso de um sistema dinâmico real é normalmente um sistema dinâmico irrealizável e conseqüentemente pode ser implementado apenas aproximadamente.

7.3.4 Controle com Retroação

A mais simples topologia permitindo o controlador sentir os efeitos de uma perturbação é usar a retroação.

Os dois modos principais de empregar uma RNA em um controle usando esta topologia é como controlador e como supervisor do controlador.

- Uma RNA pode ser usada diretamente na implementação do controlador. Usa-se um controlador para o processo durante algum tempo e registram-se suas reações formando um conjunto de treinamento. Depois de treinada uma rede neural com esse conjunto de treinamento, a RNA substitui o controlador.
- Como *supervisor*. Nesse caso essencialmente usa-se um controlador convencional e a RNA é usada para ajustar os parâmetros do controlador. Um tipo de controlador muito comum é o conhecido PID. Redes neurais podem ser usadas para implementar o comportamento de um PID mas podem também ser usadas para ajustar os pesos do PID. Esta opção tem a vantagem de deixar o sistema funcionar do modo que vinha funcionando e a RNA vem apenas melhorar o funcionamento global.

7.3.5 Modelos Internos

Esta topologia de controle popularizada pelo livro de Morari & Zafriou [63] tem grande motivação biológica.

Com efeito, supõe-se que muitos seres vivos constroem um modelo mental de seus corpos e do ambiente que os cerca. Baseado nestes modelos e por simulações mentais, ele decide que ações deve realizar. Por exemplo, um cavalo tem o modelo de seu corpo e visualizando o ambiente que o cerca pode decidir passar por uma porta com grande tristeza para o cavaleiro que não estando incluído neste modelo se arrisca a ficar dependurado na bandeira da porta...

Nesta topologia um modelo do processo recebe o mesmo sinal atuante que o processo e suas respostas são comparadas. Se existe diferença entre as respostas do modelo e do processo, pode-se atribuir que houve uma perturbação. Então esta diferença é realimentada para produzir um sinal de controle que corrija os efeitos da perturbação.

Esta topologia de controle pode ser implementada por uma RNA e treinada diretamente com sinais excitação e resposta do processo. Deve-se, no entanto, lembrar que esta estrutura é aplicável somente no caso de processos estáveis ou que tenham sido previamente estabilizados. Com efeito, em ausência de perturbação, se o modelo for perfeito, o sistema funciona a malha aberta. Portanto, se o processo a controlar for instável, o processo controlado também o será.

7.4 Séries Temporais

Uma aplicação de RNAs à séries temporais bastante útil é à previsão de mercado de capitais. Neste caso, usa-se uma rede direta que é treinada com valores de uma série temporal que ocorreram em um determinado intervalo de tempo e com saída como sendo um valor futuro da série.

Assim sejam os valores de uma série temporal dados por:

$$u_0, u_1, u_2, \dots, u_{k-3}, u_{k-2}, u_{k-1}, u_k$$

Usa-se como entrada da rede a sequência:

$$u_{k-p}, u_{k-p+1}, u_{k-p+2} \dots u_{k-3}, u_{k-2}, u_{k-1}$$

E para saída da rede, durante o treinamento: u_{k+q} Para $q = 0$ a rede vai aprender a estimar o próximo valor de uma série temporal a partir de uma sequência de p valores anteriores. Para outros valores de q valores futuros da série serão estimados.

É claro que a estimação será tanto melhor quanto mais correlacionados forem os valores da série, e também quanto melhor tenha sido o treinamento. Para gerar vários conjuntos de treinamento costuma-se observar o fenômeno durante um intervalo de tempo bem maior do que o que será usado no treinamento, podendo com isto usar vários pares de valores para o treinamento.

Séries temporais tem sido usadas em vários tipos diferentes de aplicações, indo desde mercado de capitais até previsões meteorológicas.

7.5 Monitoramento

Redes neurais podem ser muito úteis em monitoramento se explorada a capacidade de uma rede direta ter resposta praticamente imediata e esta solução deve ser cogitada em casos onde a rapidez de resposta é fator primordial.

Um exemplo é o da detecção de vibrações em um reator nuclear, como no caso concreto de centrais nucleares na Bélgica [14]. Vibrações em uma instalação nuclear em frequências próximas das frequências de ressonância de partes importantes da instalação podem ter resultados catastróficos. Elas devem ser detetadas urgentemente, antes que sua amplitude chegue a valores críticos.

Em muitas instalações costuma-se periodicamente fazer inspeções que consistem muitas vezes em registrar o ruído neutrônico. Supondo que este ruído é branco, qualquer alteração no espectro de frequência deste ruído registrado é indicação que existe uma vibração neste frequência. Havendo uma vibração do núcleo pode-se esperar que esta vibração se traduz por uma variação na espessura da proteção entre o núcleo e o captor. Isto faz variar a absorção de neutrons modulando o ruído neutrônico que deixa de se apresentar como ruído branco. A idéia foi colocar

uma RNA, treinada a detetar o espectro de frequência de um sinal, solução que deu perfeita satisfação.

8 Implementação

Pessoas ligadas às ciências da computação estão tão habituadas à idéia de uma CPU executando instruções sobre dados escritos e escrevendo resultados na memória que chegam a ter a idéia de que isto é natural e a única opção em computação.

A Natureza usa paradigma diferente para resolver problemas. Assim, sendo as RNA inspiradas na Natureza, é natural que sua implementação siga uma abordagem diferente. Essencialmente as implementações de RNAs podem ser feitas de tres modos distintos:

1. Simulação da RNA em um IBC através de técnicas de simulação.
2. Implementação por circuitos.
 - (a) Simulação com uso de circuitos especializados, tais como co-processadores.
 - (b) Implementação direta em circuitos. Neste caso os circuitos podem ser digitais ou analógicos.

8.1 Simulação de RNA

Um programa de simulação deve ter as seguintes características:

- **Capacidade de representar o sistema a simular.** No caso da simulação de uma rede neural, é necessário dispor de ferramentas para representar a rede, os valores das conexões sinápticas e implementar algoritmos de aprendizado. Essencialmente necessita-se implementar multiplicação matricial e uma função não linear.
- **Capacidade de representar a experiência de simulação.** Inclui-se aí a interface, possibilidade de funcionamento autônomo, etc.

Essencialmente pode-se usar:

- **Linguagens de uso geral** tais como Lisp, C, Oberon, estas duas últimas com facilidades de programação objeto, o que tem interessado os pesquisadores da área ultimamente.
- **Linguagens de simulação** que podem ser divididas em:
 - *linguagens de modelagem*
 - *linguagens de simulação no sentido restrito.*

Duas linguagens de simulação populares são Matlab e DESIRE/NEUNET desenvolvido por Granino Korn [57].

- **Programas de simulação no sentido restrito.** Estes programas geralmente apresentam interface amigável, havendo muitos deles de domínio público e que são acessíveis pela Internet. Exemplos são os do grupo PDP, Macbrain, Mactivation, NeuralWare.

8.2 Implementações por Circuitos

Implementar RNAs por simulação é um modo limitado por razões de performance. As tecnologias emergentes tem colaborado em muito para implementação de RNAs. Isto está bem evidente na última década quando os enormes avanços de tecnologia VLSI tem estimulado pesquisas em implementação em escala de uso industrial.

Essencialmente existem tres modos de implementar uma RNA como circuito autônomo:

1. Emulação por circuitos tais como co-processadores [68, 88]
2. Como Neurocomputadores incluindo estruturas especializadas em representar RNA e dispositivos de entrada e saída [68]
3. Circuitos dedicados [49, 69].

8.2.1 Implementação da Sinápsse

Essencialmente a sinápsse em uma RNA pode ser vista como um componente armazenando um coeficiente (valor da conexão sináptica) e cuja função é multiplicar este valor pelo valor da entrada dando sua saída. A sinápsse é portanto um multiplicador.

Em uma RNA totalmente interconectada com n neurônios, n_2 sinápses são necessárias. Em uma rede direta com p camadas, possuindo n_i neurônios na camada i , o número de sinápses S_r será:

$$S_r = n_1n_2 + n_2n_3 + \dots n_in_{i+1}\dots + n_{p-1}n_p = \sum_{i=1}^p n_{i-1}n_i$$

Como o número de sinápses é maior do que o número de neurônios, sua implementação é crucial na construção das RNAs.

8.2.2 Implementação do Neurônio

O neurônio pode ser dividido em dois circuitos o primeiro implementa a operação de *confluência* das entradas ponderadas pelos valores sinápticos. O segundo calcula a saída do neurônio.

8.2.3 Implementação do Aprendizado

A dificuldade na implementação de um algoritmo de aprendizado depende se é usada apenas informações locais ou não. No caso de usar apenas informações locais necessita-se de sub-circuitos com duas entradas, que são a excitação dos dois neurônios ligados pela sinápsse. A saída é usada para atualizar o valor da sinápsse. No caso de um algoritmo não local são necessárias q entradas, onde q é o número de sinais de que depende o algoritmo. A figura 8 mostra uma RNA direta por camadas usando um algoritmo não local. Vê-se que a sinápsse w_{13} para ser alterada usa algoritmo de aprendizado representado na figura pelo bloco L , o qual tem como entradas os valores das saídas y_i e das saídas desejadas Y_{id} como em um algoritmo de aprendizagem supervisionada.

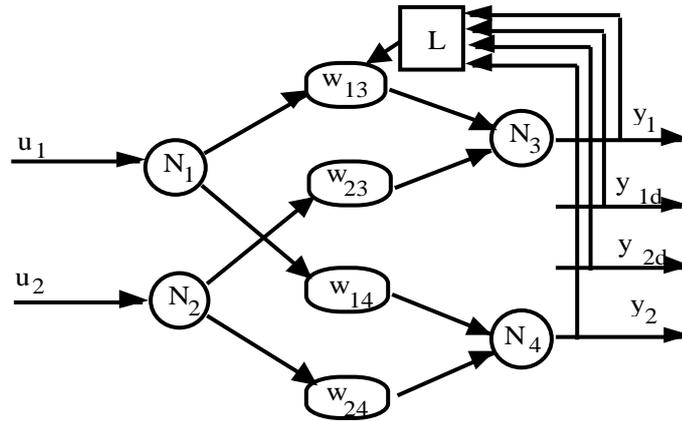


Figura 8: Rede com aprendizado não local.

Portanto os algoritmos que usam informação não local exigem um maior grau de complexidade de implementação. [49].

Muitas RNAs possuem um conjunto de neurônios que competem uns com os outros, de tal forma que a saída de todos menos um se mantém pequena [19]. Isto exige também um circuito com múltiplas entradas, as excitações dos neurônios que estão competindo.

8.2.4 Implementações Digitais versus Analógicas

Em uma implementação analógica tanto u_i entrada de neurônios quanto w_{ji} valores sinápticos tem valores em um segmento dos números reais que podem ser calculados usando a Lei de Ohm e a saída é uma corrente e a soma pode ser feita graças à Lei de Kirchhoff das correntes. A corrente resultante é em seguida convertida para tensão, entrada de operador não linear para dar a saída do neurônio. O cálculo é verdadeiramente em paralelo.

Quando a implementação é digital, então o cálculo vai requerer um grande número de multiplicadores e acumuladores, correspondendo a todos os neurônios da rede o que torna pouco realística tal solução. Entretanto soluções parcialmente digitais são possíveis em alguns casos.

A tabela que se segue apresenta uma comparação das duas tecnologias.

Implementação Digital versus Analógica

ANALÓGICA	DIGITAL
Requer muito conhecimento	Requer menos conhecimento
Ferramentas de CAD ¹² não são comuns	Ferramentas de CAD são comuns
Dífceis de testar	Fáceis de testar
Sensíveis ao ruído	Boa rejeição ao ruído
Densidade funcional elevada	Baixa densidade funcional
Fácil implementação de não linearidades	Difícil implementação de não linearidades
Regras de aprendizado local	Regras de aprendizado de difícil implementação
Próxima da realidade biológica	Diferente da realidade biológica

9 Epílogo

Uma palavra final adequada seria que o campo das RNAs é vasto. Aqui procurou-se dar ao leitor uma visão geral do assunto.

Acredita-se que as RNAs vieram para ficar. Mas redes neurais constituem um assunto novo e velho de pesquisa. Como todo assunto velho trata-se de assunto maduro com muitos anos de pesquisa apoiando os conhecimentos atuais. Como novo campo, serve de polo de atração para profissionais de vários domínios distintos se transformando em assunto interdisciplinar. Com efeito:

- Psicólogos estão vendo possibilidades de construir redes neurais artificiais e ver aparecer comportamentos emergentes tais como o aprendizado de novos conceitos dando uma ajuda na compreensão dos mecanismos do aprendizado.
- Neurofisiologistas estão interessados em ver as redes neurais artificiais como metáfora do cérebro permitindo experimentação com esta metáfora e melhorando o conhecimento dos mecanismos cerebrais.
- Cientistas cognitivos se empregam em usar as redes neurais artificiais para um melhor conhecimento dos mecanismos envolvidos no processo cognitivo.
- Engenheiros olham redes neurais artificiais como um caminho para, implementado estas redes em circuitos, ter computadores realmente paralelos e distribuídos. Muitos encontraram no aprendizado das redes neurais um campo para aplicar o que se conhece da teoria da otimização, incluindo o caso de sistemas dinâmicos e estáticos com algoritmos de programação não linear.
- Cientistas de computação encontraram um novo paradigma de programação e uma arquitetura distribuída. Explorar este paradigma é um desafio.

Leitura Complementar

Ao término deste texto é conveniente citar alguns livros de interesse que podem ser consultados para um aprofundar o assunto. Para uma introdução às bases do conexionismo, os trabalhos de Arbib [2], [3], [4], são claros, precisos e fazem uma ponte segura com conceitos biológicos. Mais populares são, no entanto, os livros do grupo PDP (“Parallel Distributed Processing”) [82], que se não podem ser ignorados. Recentemente Haykin [42] publicou livro didático bastante completo sobre o assunto. Kosko [59] trata de RNA e sistemas nebulosos realçando as semelhanças entre os dois campos de estudo. Em português recomenda-se [16], [7], [60].

No Brasil existem livro traduzido e alguns bons de autores nacionais. A segunda edio do Haykin est aumentada e disponivel [52]. Existe ainda os livro de Kovács (1996 e 1997) [60, 61] sem deixar de mencionar os de Loesch e o de Tafner e co-autores, ambos elementares [62], [85].

Mais recentemente (2000) o grupo de pesquisas do Recife produziu em cooperacção com São Carlos livro bem feito e mais profundo [30]. Ainda no mesmo ano apareceram outros livros voltados para redes neurais, dentre os quais, um, de três ex-doutorandos do autor do presente texto, F. M. De Azevedo, Lourdes M. Brasil e R. C. Limão de Oliveira, aos quais agradeço terem colocado em livro meus ensinamentos e idéias [29].

Referências

- [1] ABU-MOSTAFA, Y. S. & JACQUES, J. S. Information capacity of the Hopfield model. *IEEE Tans. on Information Theory* 31, 4 (July 1985), 461–464.
- [2] ARBIB, M. A. *Brains, Machines and Mathematics*. McGraw-Hill, 1964.
- [3] ARBIB, M. A. *Brains, Machines and Mathematics*, 2 ed. Springer, 1987.
- [4] ARBIB, M. A. *The Metaphorical Brain 2: Neural Networks and Beyond*. John Wiley and Sons, New York, 1989.
- [5] BARRETO, J. M. Neural networks learnig: A new programming paradigm. In *ACM International Conference: Trends and Directions in Expert Systems*, Orlando, Florida, October 29 to November 3 1990, p. ??
- [6] BARRETO, J. M. *Expert systems in biomedicine*. Universidade Católica de Louvain, Faculdade de Medicina, Programa de Informática Médica, Bruxelas, 1993.
- [7] BARRETO, J. M. *Conexionismo e a resolução de problemas*. UFSC, Dept. de Informática e Estatística, Florianópolis, maio 1996. Trabalho para concurso público para Professor Titular.
- [8] BARRETO, J. M. *Redes Neurais Artificiais*. Texto de Curso apresentado no II-CIPEEL, Universidade Federal de Santa Catarina, Dept. de Engenharia Elétrica, novembro 1996.
- [9] BARRETO, J. M. Introdução às redes neurais artificiais. In *V Escola Regional de Informática*. Sociedade Brasileira de Computação, Regional Sul, Santa Maria, Florianópolis, Maringá, 5-10/05 1997, p. 41–71.
- [10] BARRETO, J. M. *Inteligência Artificial no Limiar do Século XXI*, 3 ed. *ppp* Edições, Florianópolis, SC, 2001.
- [11] BARRETO, J. M. & DE AZEVEDO, F. M. Connectionist expert systems as medical decision aid. *AI in Medicine* 5, 9 (setembro 1993), 1–9.
- [12] BARRETO, J. M. & DE AZEVEDO, F. M. On the robot neural control. Relatório técnico., Projeto MUCOM (Multisensory Control of Movement) do Programa ESPRIT de Pesquisa Básica, da CEE (Comissão Econômica Europeia), Lab. of Neurophysiology, UCL, Bruxelas, maio 1993.
- [13] BARRETO, J. M.; DE AZEVEDO, F. M.; DE LIMA, W. C. & EPPRECHT, L. R. A neural network with fuzzy elements for a medical diagnosis. In *LATINCON'92, IEEE Latinoamerican Conference*, Santiago, Chile, 1992, p. 335–339.

- [14] BARRETO, J. M.; DE F. M. AZEVEDO; DE LIMA, W. C. & ZANCHIN, C. I. Neural network identification of resonance frequencies from noise. In *IX Congresso Brasileiro de Automatica*, Vitoria, Brazil, setembro 14-18 1992, p. 840–844.
- [15] BARRETO, J. M.; NEYER, M. D.; LEFÈVRE, P. & GOREZ, R. Qualitative physics versus fuzzy sets theory in modeling and control. In *IECON'91: IEEE International Conference on Industrial Electronics, Control and Instrumentation*, Kobe, Japão, outubro-novembro 1991, p. 1651–1656.
- [16] BARRETO, J. M.; NIEVOLA, J. & DE LIMA, W. C. A student freedom of a tutor system. In *MELECON'96; 8th IEEE Mediterranean Electrotechnical Conference*, Bari, Itália, 13-16/5 1995, IEEE, p. 1097–1100.
- [17] BARRETO, J. M. & PROYCHEV, T. Control of the standing position. Relatório técnico., Projeto MUCOM (Multisensory Control of Movement) do Programa ESPRIT de Pesquisa Básica, da CEE (Comissão Econômica Europeia), Lab. of Neurophysiology, UCL, Bruxelas, May 1994.
- [18] BECHTEL, A. & ABRAHAMSEN, A. *Connectionism and the mind*. Basil Blackwell, Cambridge, Massachusetts, 1991.
- [19] CARPENTER, G. A. & GROSSBERG, S. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics and Image Processing* 37 (1987), 54–115.
- [20] COIMBRA, A. F.; D'ANGELO, G.; MARINO-NETO; DE AZEVEDO, F. M. & BARRETO, J. M. Use of neural networks in brain state analysis. In *Neural Network Contact Group*, Louvain-la-Neuve, 30/11 1994.
- [21] COIMBRA, A. F.; D'ANGELO, G. G.; DE AZEVEDO, F. M. & JORGE M. BARRETO, J. M.-N. Electrographic analysis of brain states using neural networks. In *World Congress on Medical Physics and Biomedical Engineering*, Rio de Janeiro, 13-16/8 1994, vol. 1, p. 463.
- [22] COIMBRA, A. F.; MARINO-NETO; FREITAS, C. G.; DE AZEVEDO, F. M. & BARRETO, J. M. Automatic detection of sleep-waking states using Kohonen neural networks. In *1º Congresso Brasileiro de Redes Neurais*, Escola Federal de Engenharia de Itajubá, 24-27/10 1994.
- [23] COIMBRA, A. F.; MARINO-NETO, J.; DE AZEVEDO, F. M. & BARRETO, J. M. Brain electrographic state detection using combined unsupervised and supervised neural networks. In *Artificial Neural Nets and Genetic Algorithms*, N. C. S. e. R. A. D. W. Pearson, Ed. Springer Verlag, Viena, 1995, p. 76–79.

- [24] DARWIN, C. *The origin of species*. Grolier Enterprises Corp., Danbury, Connecticut, 1859.
- [25] DE AZEVEDO, F. M. *Contribution to the Study of Neural Networks in Dynamical Expert System*. Tese de Doutorado, Institut d'Informatique, FUNDP, Namur, Bélgica, 1993.
- [26] DE AZEVEDO, F. M.; BARRETO, J. M.; DE LIMA, W. C. & ZANCHIN, C. I. Teaching medical diagnosis with associative memories. In *Proc. XI Systems Science International Conference*, Wroclaw, Poland, setembro 22-25 1992, p. 28–29.
- [27] DE AZEVEDO, F. M.; BARRETO, J. M.; EPPRECHT, E. K.; EPPRECHT, L. R. & DE LIMA, W. C. Two approaches in case-based connectionist expert systems. In *AINN'91 Artificial Intelligence and Neural Networks*, Anahaim, Calgary, julho 1991, M. H. Hanza, Ed., Acta Press, p. 13–17.
- [28] DE AZEVEDO, F. M.; BARRETO, J. M.; EPPRECHT, L.; DE LIMA, W. C. & ZANCHIN, C. Neural network approach for medical diagnostic. In *Mini and Microcomputers in Medicine and Health Care 91*, Anahaim, Calgary, December 1991, D. Hudson, Ed., Acta Press, p. 16–18.
- [29] DE AZEVEDO, F. M.; BRASIL, L. M. & DE OLIVEIRA, R. C. L. *Redes Neurais*, 1 ed. Visual Books, Florianópolis, Sept. 2000.
- [30] DE PÁDUA BRAGA, A.; LUDERMIR, T. B. & DE LEON FERREIRA DE CARVALHO, A. C. P. *Redes neurais artificiais: teoria e aplicações*. Livros Técnicos e Científicos Editora, 2000.
- [31] FALQUETO, J.; DE LIMA, W. C.; DA SILVA BORGES, P. S. & BARRETO, J. M. The measurement of artificial intelligence: an IQ for machines? In *Proceedings of The International Conference on Modeling, Identification and Control*, Insbruck, Austria, fevereiro 2001, IASTED.
- [32] FRANKLIN, S. & GARZON, M. Neural computability. In *Progress in Neural Networks*, O. Omidvar, Ed. Ablex Publishing Co., New Jersey, 1991, p. 127–146.
- [33] FUKUSHIMA, K. Cognitron: A self-organizing multilayered neural network. *Biol. Cybernet.* 20 (1975), 121–136.
- [34] FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybernet.* 36 (1980), 193–202.
- [35] FUKUSHIMA, K.; MIYAKE, S. & ITO, T. Neocognitron: a neural network model for a mechanism of visual pattern recognition. *IEEE Trans. on Systems, man, and Cybernetics* 13, 5 (September/October 1983), 826–834.

- [36] GALLANT, S. Connectionist expert systems. *Communications of the ACM* 31, 2 (1988).
- [37] GALLANT, S. I. *Neural network learning and expert systems*. MIT press, Cambridge, Massachussets, 1993.
- [38] GARCIA, R.; DE AZEVEDO, F. M. & BARRETO, J. M. Artificial neural networks optimized by genetic algorithms. In *EXPERTSYS-95*, San Francisco, November 1995.
- [39] GIBSON, J. E. *Nonlinear Automatic Control*. McGraw-Hill, Nova York, 1963.
- [40] GILLE, J. C.; DECAULNE, P. & PÉLÉGRIN, M. *Méthodes modernes d'études des systèmes asservis*. Dunod, Paris, 1960.
- [41] HARARY, F. *Graph Theory*. Addison-Wesley, 1971.
- [42] HAYKIN, S. *Neural networks: a comprehensive foundation*. IEEE Press, New York, 1994.
- [43] HEBB, D. O. *The organization of behavior*. Wiley, New York, 1949.
- [44] HECHT-NIELSEN, R. Neurocomputing: Picking the human brain. *IEEE Spectrum* 25, 3 (1988), 36–41.
- [45] HECHT-NIELSEN, R. *Neurocomputers*. Addison-Wesley, Massachusetts, 1990.
- [46] HINTON, G. E. Connectionist learning procedures. *Artificial Intelligence* 40, 1-3 (1989), 185–234.
- [47] HOPFIELD, J. Neural Networks and Physical Systems with Emergent Collectives Computational Abilities. *Proceedings of the National Academy of Sciences* 79 (1982), 2554–2558.
- [48] HOPFIELD, J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences* 81 (1984), 3088–3092.
- [49] HOPFIELD, J. The effectiveness of analog neural network hardware. *Network* 1, 1 (1990), 27–40.
- [50] HUBEL, D. H. *The brain*. vol. 1. Freeman, San Francisco, 1979, ch. 1, p. 2–14.
- [51] JANSON, D. J. & FRENKEL, J. Training product unit neural networks with genetic algorithms. *IEEE Expert* 8, 5 (1993), 26–33.
- [52] JAYKIN, S. *Redes Neurais: Princípios e Práticas*. Bookman, 2001.
- [53] KFOURY, A. J.; MOLL, R. N. & ARBIB, M. A. *A Programming Approach to Computability*. Springer Verlag, 1982.

- [54] KING, P. & MAMDANI, E. The application of fuzzy control systems to industrial processes. *Automatica* 13 (1977), 235–242.
- [55] KOHONEN, T. *Self Organization and Associative Memory*. Springer-Verlag, Berlin, 1987.
- [56] KOHONEN, T. State of the art in neural computing. In *IEEE First International Conference on Neural Networks*, 1987, vol. 1, p. 77–90.
- [57] KORN, G. *Neural Networks Experiments on personal computers and workstations*. MIT Press, A Bradford Book, Cambridge, Massachusetts, 1992.
- [58] KOSKO, B. Bidirectional associative memories. *IEEE Trans. on Systems, Man and Cybernetics* 18, 1 (1988).
- [59] KOSKO, B. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice-Hall, 1992.
- [60] KOVÁCS, Z. L. *Redes neurais artificiais*. Edição Acadêmica, São Paulo, 1996.
- [61] KOVÁCS, Z. L. *O cérebro e sua mente: uma introdução á neurociência computacional*. Edição Acadêmica, São Paulo, 1997.
- [62] LOESCH, C. & SARI, S. T. *Redes neurais artificiais: fundamentos e modelos*. FURB, Blumenau, 1996.
- [63] M., M. & ZAFIRIOU, E. *Robust process control*. Prentice-Hall, Englewood Cliffs, New Jersey, 1975.
- [64] MAMDANI, E. Application of fuzzy algorithms for control of simple dynamic plant. *Proc. IEE* 121, 12 (1974), 1585–88.
- [65] MCCULLOCH, W. S. *Embodiments of Mind*. The MIT press, Cambridge, Massachusetts, 1965.
- [66] MCCULLOCH, W. S. & PITTS, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics* 5 (1943), 115–133.
- [67] MCELIECE, R. J.; POSNER, E. C.; RODEMICH, E. R. & VENKATESH, S. S. The capacity of Hopfield associative memory. *IEEE Trans. on Information Theory* 33, 4 (July 1987), 461–482.
- [68] MEAD, C. *Analog VLSI and Neural Systems*. Addison- Wesley, 1989.
- [69] MEAD, C. A. *Analog VLSI Neural Systems*. Addison-Wesley, 1988.

- [70] MEAD, C. A. & MAHOWALD, M. A. A silicon model of early visual processing. *Neural Networks 1*, 1 (1988).
- [71] MINSKY, M. L. & PAPERT, S. A. *Perceptrons: an introduction to computational geometry*. The MIT Press, Massachusetts, 1969.
- [72] MINSKY, M. L. & PAPERT, S. A. *Perceptrons: an introduction to computational geometry*, 3ª ed. The MIT Press, Massachusetts, 1988. Impressão modificada do original de 1969.
- [73] MONOD, J. *Le hasard et la nécessité*. Editions du Seuil, Paris, 1970.
- [74] POLYA, G. *A Arte de Resolver Problemas*. Intersciência, Tradução de “How to solve it: A New Aspect of Mathematical Method”, Princeton University Press, Rio de Janeiro, 1975.
- [75] ROBINSON, J. A. A machine oriented logic based on the resolution principle. *J. ACM 12*, 1 (1965), 23–41.
- [76] ROISENBERG, M.; BARRETO, J. M. & DE AZEVEDO, F. M. Generalization capacity in neural networks - the ballistic trajectory learning case. In *II Congresso Brasileiro de Redes Neurais*, Curitiba, 29/10-1/11 1995, p. 4 páginas.
- [77] ROISENBERG, M.; BARRETO, J. M. & DE AZEVEDO, F. M. Specialization versus generalization in neural network learning. In *IEEE MELECON'96 - 8th IEEE Mediterranean Electrotechnical Conference*, Bari, Italy, May 1996, p. 627–630.
- [78] ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review 65* (1958), 386–408.
- [79] ROSENBLATT, F. *Principles of Neurodynamics*. Spartan Books, 1959.
- [80] RUMELHART, D. E.; HINTON, G. E. & MCCLELLAND, J. L. A general framework for parallel distributed processing. In *Parallel distributed processing, vol 1: Foundations*, D. E. Rumelhart, J. L. McClelland and the PDP group, Ed. MIT Press, Cambridge, Massachusetts, 1987, p. 45–76.
- [81] RUMELHART, D. E.; HINTON, G. E. & WILLIAMS, R. J. Learning internal representations by error propagation. In *Parallel distributed processing, vol 1: Foundations*, D. E. Rumelhart, J. L. McClelland and the PDP group, Ed. MIT Press, Cambridge, Massachusetts, 1987, p. 318–362.
- [82] RUMELHART, D. E.; MCCLELLAND, J. L. & GROUP, P. *Parallel Distributed Processing, vol 1: Foundations*. The MIT Press, 1986.
- [83] SHANNON, C. E. & MCCARTY (EDT.), J. *Automata Studies*. Princeton University Press, New Jersey, 1956.

- [84] SHORTLIFFE, E. H. *MYCIN: a rule-based computer program for advising physicians regarding antimicrobial therapy selection*. Tese de Doutorado, Stanford University, California, 1974.
- [85] TAFNER, M. A.; DE XEREZ, M. & FILHO, I. W. R. *Redes neurais artificiais: introdução e princípios de neurocomputação*. Editoras ECO e Editora da FURB, Blumenau, 1996.
- [86] VELOSO, P. A. S. & VELOSO, S. R. M. Problem decomposition and reduction. In *Progress in Cybernetics and System Research*, F. P. R. Trappl, J. Klir, Ed., vol. VIII. Hemisphere, Washington DC, 1981, p. 199–203.
- [87] ZADEH, L. A. Fuzzy sets. *Information and Control* 8, 1 (January 1965), 338–353.
- [88] ZURADA, J. M. *Introduction to Neural Networks*. West Pub. Co., 1992.