

## 1. Introdução

A tendência científica contemporânea prossegue seu perfil inovador em todas as áreas de pesquisa existentes. Um forte estímulo é inserido ao meio, graças à característica contínua dos projetos acadêmicos. Todas as áreas científicas se encontram, quebrando assim, a barreira multidisciplinar, que de certa forma ainda compõe o meio acadêmico.

A universidade tem papel fundamental no estudo de ciências específicas, que tratam de assuntos atuais e aprimora-se a partir de sua própria evolução, proposta por programas científicos, grupo de pesquisadores, parcerias empresariais de várias áreas.

As organizações, cada vez mais, tem se interessado no percurso traçado por tais ciências, levando o interesse empresarial ao meio acadêmico, e de si, extraíndo todo respaldo tecnológico. Muitas tecnologias desenvolvidas proporcionam a execução de determinados procedimentos organizacionais, que antes eram inviáveis ou impossíveis de serem exercidos.

De acordo com Póvoa (2005), no mundo globalizado, as empresas devem estar atentas para a concorrência acirrada. Para uma organização sobreviver, e expandir a sua atuação, deve oferecer produtos de qualidade com preços competitivos de mercado. Para se firmarem no mercado, as empresas devem usar tecnologia em todos os seus processos (compra, produção, transporte, etc.).

A dinâmica de mercado atual não permite um comportamento empresarial passivo, sem as devidas mudanças e adaptações ao ambiente globalizado. A parceria empresa-ciência colabora para a maneira como as ferramentas evoluem e para a elaboração de uma boa estrutura organizacional. O diferencial das empresas vem sendo sustentado por variadas propostas de melhoria em seu ambiente de produção. Os processos e metodologias presentes, muitas vezes, não satisfazem a demanda atual, e isso significa que estarão ainda mais obsoletos para uma demanda futura. Pesquisas especializadas ao meio podem ser empregadas com o intuito de ajustar os resultados de acordo com a realidade e necessidade da

empresa.

Novas tecnologias podem se transformar, segundo Johansen e Swigart (1994), da noite para o dia, de símbolos de status em grilhões. Expandindo-se as empresas em rede pelo globo, a necessidade dos gerentes de contato (entenda-se controle) aumenta também.

De acordo com Chiavenato (2004), a competitividade tornou-se mais intensa entre as organizações. O mercado de capitais passou a migrar volatilmente de um continente para outro em segundos, à procura de novas oportunidades de investimentos, ainda que transitórias.

Esse ambiente é propício ao estudo aprimorado e diversificado de metodologias compatíveis ao meio e ao mesmo tempo contribuindo para a evolução das mesmas. Com isso, o rumo de algumas ciências vem se cruzando durante o decorrer do tempo, juntado assuntos de interesse comum e assim traçando caminhos de pesquisa com relevância considerável.

O ambiente que favorece tal situação é muito estimulado na engenharia, mais especificamente, na engenharia de produção, que visa aprimorar através da contextualização científica, o tratamento e implementação de determinados procedimentos visando a melhoria. A otimização envolvida no ambiente organizacional, porém, nem sempre é compatível com o conjunto de ferramentas existentes. Isso implica na necessária adaptação que essa ciência sofre com o decorrer de sua evolução. Ciências como a Pesquisa Operacional, Estatística, Logística, estão consideravelmente inseridas neste meio.

A Pesquisa Operacional visa estudar minuciosamente o comportamento de um problema específico, e através de um conjunto de ferramentas, tais como as heurísticas, é capaz de interpretar o comportamento do sistema real em um modelo representativo, bem como, através do mesmo, encontrar uma solução plausível, que otimiza as variáveis de decisão envolvidas.

A própria Logística se enquadra nesse perfil. Segundo Ballou (1993) a logística associa estudo e administração dos fluxos de bens e serviços e da

informação que os põe em movimento. Caso fosse viável produzir todos os bens e serviços no ponto onde eles são consumidos ou caso as pessoas desejassem viver onde as matérias-primas e a produção se localizam, então a logística seria pouco importante. Mas isto é uma utopia. Uma região tende a especializar-se na produção daquilo que tiver vantagem econômica para fazê-lo. Isto cria um hiato de tempo e espaço entre fontes de matéria-prima e produção e entre produção e consumo. Vencer tempo e distância na movimentação de bens ou na entrega de serviços de forma eficaz e eficiente é tarefa da logística. Ou seja, sua missão é colocar as mercadorias ou serviços certos no lugar, no instante correto e na condição desejada, ao menor custo possível.

Em se tratando de estatística, Carvalho (2004) afirma que esta metodologia, este método, consiste em uma série de etapas, iniciando pela coleta das informações (dos dados) que, após coletadas, passarão por uma organização e apresentação. Chega-se, daí, a uma fase complementar, na qual se dará a análise daqueles dados (já organizados e descritos). Esta análise dos dados coletados funcionará como um meio, pelo qual se chegará a uma conclusão. Esta, por sua vez, ensejará uma tomada de decisão.

Em alguns ramos mais específicos, a Pesquisa Operacional trabalha com conhecimentos extraídos de outras ciências. Como exemplo, o Algoritmo Genético se baseia no comportamento hereditário humano para se concentrar em resultados mais compatíveis à realidade.

Em outro plano, tem-se o estudo das Redes Neurais, que através do comportamento padronizado de determinadas situações, é capaz de prever um resultado coerente e muitas vezes, passível de melhorias. Apresentam performance considerável e são baseados no comportamento do cérebro humano, e por isso, intitulados como Redes Neurais Artificiais (RNA). Estes algoritmos, de forma análoga às ferramentas estatísticas, são usados para interpretar um grande conjunto de dados para abstrair informações pertinentes à tomada de decisão. Metodologias como essas são muito importantes para a ascensão das empresas, tornando-as mais vigorosas e competitivas.

Segundo Haykin (2001), é evidente que uma rede neural extrai seu poder

computacional através, primeiro, de sua estrutura paralelamente distribuída e segundo, de sua habilidade de aprender e, portanto de generalizar. A generalização se refere ao fato de a rede neural produzir saídas adequadas para entradas que não estavam presentes durante o treinamento (aprendizagem). Estas duas capacidades de processamento de informação tornam possível para as redes neurais resolver problemas complexos (de grande escala), que são atualmente intratáveis.

A principal e breve definição que pode ser associada ao comportamento assumido pelas Redes Neurais Artificiais é o de desenvolver algoritmos, incorporados à sistemas de informação (programas de computador), capazes de simular o comportamento neural humano. Muito se associa esse ramo de pesquisa à Inteligência Artificial (IA).

De acordo com Schildt (1989), um programa inteligente é aquele que exhibe comportamento similar ao de um ser humano quando confrontado com um problema análogo. Não é necessário que o programa realmente solucione, ou procure solucionar o problema da mesma maneira que um ser humano.

A intenção de alcançar o comportamento real do cérebro humano tem fundado novas pesquisas no campo científico neural, visando espelhar-se cada vez mais na realidade. Quando novas descobertas sobre o comportamento real do cérebro são publicadas, cria-se um ambiente propício para a criação de modelos e adaptações, buscando com isso, atingir resultados melhores.

As Redes Neurais Artificiais possuem um escopo matemático caracterizando as adaptações necessária para que seja obtida uma aplicabilidade ao contexto da engenharia. Entretanto, todo seu desenvolvimento origina-se do modelo biológico. Espera-se que a continuidade de seus princípios biológicos forneça respaldo para alcançar maior nível de evolução como ferramenta.

## 1.1 Objetivo

Como principal objetivo, aspira-se relacionar o avanço da ciência biológica

neural sob o contexto cognitivo e biológico, aos algoritmos já existentes da Engenharia de Produção, mais especificamente, as Redes Neurais Artificiais.

Todo respaldo dos modelos artificiais hoje existentes foi adquirido através de estudos desenvolvidos no comportamento real do cérebro humano. Ao longo de sua evolução, os algoritmos artificiais seguiram um caminho de influência matemática, distanciando-se, de certa forma, do comportamento biológico associado. Em alguns pontos, marcados por novas descobertas do funcionamento biológico, os modelos artificiais retornaram à inspiração inicial, adaptando novos algoritmos e buscando maior proximidade do comportamento do cérebro.

Esta dissertação centra-se na alteração de alguns dos algoritmos inseridos nos modelos neurais artificiais – perceptron e backpropagation – a partir de novas descobertas e tendências do campo de pesquisa biológico, visando conquistar um maior desempenho – tratando-se de resultados alcançados e tempo – no seu comportamento.

O princípio biológico apresentado no Capítulo 4 mostra que durante o reaprendizado, o cérebro fortalece as conexões entre os neurônios especificamente para a situação. Ou seja, durante o aprendizado de um conceito, Matemática, por exemplo, as conexões sinápticas favorecidas não são necessariamente as mesmas do aprendizado do Português. Isto ocorre devido às particularidades existentes nos dados do aprendizado da Matemática, que são distintos do Português. Observe a Figura 1.1.

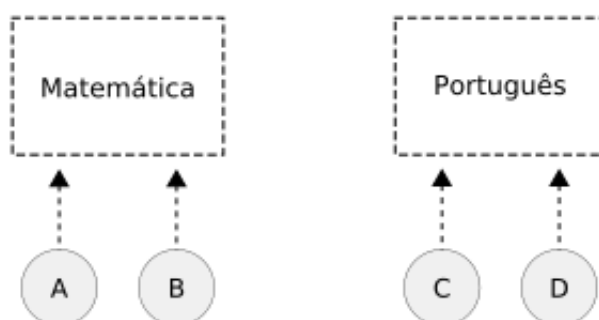


Figura 1.1 – Aprendizado de conceitos

Essas particularidades mostradas na Figura 1.1 serão abordadas no

Capítulo 6, onde as alterações propostas serão aplicadas ao algoritmo neural artificial com aprendizado backpropagation. Este é o principal objetivo do trabalho.

## 1.2 Justificativa

Todo o estudo coerente com as pesquisas relacionadas às Redes Neurais Artificiais, em instância anterior, teve respaldo e concretização (ou avanço de conhecimento) biológica. Sabe-se que um modelo nunca é capaz – comumente – de tratar com perfeição o comportamento avaliado, e sim, trata de fornecer uma avaliação ou observação detalhada do mesmo sob alguns aspectos limitadores – opcionais ou não. Isso é uma característica inata dos modelos e somente é verdadeira por dois possíveis motivos:

- O pesquisador (ou grupo de pesquisadores) avalista entende perfeitamente o processo por parâmetros conhecidos, mas com modelagens complexas, inviáveis ou desconhecidas. Através desta visão, o modelo, ou parte, é construído com intuito de simular o comportamento real do procedimento.
- O pesquisador não conhece por completo a estrutura existente no comportamento real do processo. Isso impossibilita a construção, mais próxima da realidade, de um modelo compatível.

O estudo das redes neurais, em geral, está associado ao segundo caso descrito. Já existe uma massa considerável de conhecimento relacionado ao comportamento biológico do cérebro e suas principais funções, tais como aprendizado, reconhecimento de padrões, manipulação e armazenamento de grande quantidade de informações, visão, estrutura motor, etc. Contudo, ainda não é completamente suficiente para que um modelo artificial construído para simular seu comportamento seja capaz de expressar com exatidão o processo biológico.

Durante a contextualização científica dos estudos das redes neurais, expressa com detalhes no capítulo 3, vários fatores biológicos foram levados em consideração para a construção de simples modelos neurais, mas pela falta do entendimento real, são necessárias muitas – às vezes, predominantes – adaptações matemáticas. Ou seja, na tentativa de modelar o cérebro, encontra-se

uma limitação científica causada pelo *pouco* que se conhece do processo biológico (funcionamento real do cérebro), então, modificações matemáticas (e ciências correlatas, como engenharia e estatística) plausíveis são estipuladas para que seja possível o programa de um modelo coerente.

Certamente, quando se estabelece uma estrutura mais confiável sob o comportamento real do processo, no caso, o cérebro, novos algoritmos podem ser programados (ou adaptações dos já existentes) com aspiração em diminuir a distância do comportamento do modelo e sua inspiração.

Como justificativa principal, este trabalho visa aprimorar o comportamento de alguns modelos já existentes, mais especificamente, o algoritmo de aprendizagem e estrutura celular do Perceptron e Perceptron Multi-camada (Backpropagation) propostos respectivamente por Rosenblatt (1956), e Rumelhart, Hinton e Williams (1986). A melhoria é aplicada a partir de pesquisas biológicas recentes – que fornecem o contexto do aprendizado real induzido pela proteína CaMKII, aqui, principalmente descritas por Coudry e Freire (2005), Santos, Milano e Rosat (1998) e Rosler e Quevedo (1998). Sendo isso realizado com intuito de diminuir a distância entre o modelo (Rede Artificial) e o processo real (Cérebro).

### 1.3 Organização do Trabalho

O Capítulo 2, apresentado a seguir, mostrará um breve estudo associado ao comportamento geral dos modelos neurais existentes. Tópicos de ênfase às possíveis aplicações também são salientados.

Serão também enfatizadas, as principais características biológicas de interesse na prática da modelagem (aprendizagem), assim como, a representação destes conceitos nos modelos artificiais.

No Capítulo 3, uma detalhada descrição proveniente de todo o estudo associado aos modelos artificiais, bem como, pontos biológicos considerados na criação dos algoritmos, é apresentado no Estado da Arte. Alguns dos fatores terão maior relevância no restante da dissertação, e justamente por esse fato, terão uma

abordagem maior.

O Capítulo 4 mostra os estudos biológicos pertinentes às sugestões apresentadas e abordadas neste trabalho. Em seguida, o Capítulo 5 descreve o modelo de aplicação da rede neural com aprendizado backpropagation. O conteúdo apresentado no Capítulo 6 destaca-se por se tratar dos ajustes sugeridos ao algoritmo de retropropagação. Toda a importância biológica será ressaltada e as alterações matemáticas serão abordadas.

No Capítulo 7 são realizados testes para o modelo convencional da rede de retropropagação e para sua versão alterada. Os testes terão importância na comprovação empírica das alterações. Por fim, no Capítulo 8 são apresentadas conclusões da dissertação.



## 2. Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA, ou simplesmente Redes Neurais) fornecem à Engenharia, um conjunto de ferramentas – também chamadas de algoritmos – capazes de analisar o comportamento substancial de determinado ambiente e através disso, estabelecer ou caracterizar um padrão de identificação para o mesmo.

Por ser um algoritmo com característica adaptativa ao problema – no sentido de modificar seu comportamento enfocando o ambiente envolvido, a Rede Neural precisa ser modelada e ajustada criteriosamente à cada caso.

O propósito principal deste capítulo é identificar o ambiente favorável à adoção dessa tecnologia, assim como destacar de forma substancial algumas características e metodologias empregadas nos algoritmos neurais.

### 2.1 Ambiente envolvido

Muitos sistemas envolvem uma grande concentração de dados relacionados ao seu comportamento ambiental. Em todo momento, esses dados caracterizam, seja de forma positiva ou negativa, o andamento e perspectiva do mesmo. Com o destaque de uma série de padrões pertinentes ao seu funcionamento, pode-se dizer que, cria-se um levantamento histórico que capacita o estabelecimento precipitado de seu comportamento.

Em visão mais prática, o sistema se comporta obedecendo ao conjunto de padrões ao qual pertence. Em quase todas as situações, seus resultados encaixam-se dentro dessa perspectiva – sendo que, as exceções, apesar de existirem, de alguma forma representariam uma anomalia em seu funcionamento.

Com isso, gera-se uma quantidade massiva de dados que, por questões práticas, impossibilita o estudo e conseqüente compreensão através de metodologias convencionais. Dessa forma, o sistema possui caráter abrangente e

complexo. Ferramentas que possibilitam o tratamento extensivo de uma grande quantidade de informações serão necessárias.

As Redes Neurais Artificiais, por essência, são capazes de agrupar dados semelhantes e ponderá-los com o objetivo de padronizar determinado comportamento. Essa é uma característica intuitiva do algoritmo, dada a capacidade que o mesmo tem de ajustar-se ao problema – situação trabalhada.

Isso torna a Rede Artificial uma ferramenta altamente capacitada e multidisciplinar. Conforme descrito por Ramalho (2003), as redes neurais artificiais oferecem uma boa abordagem para problemas que requeiram ajuste de funções, inclusive reconhecimento, identificação, associação ou classificação de padrões. Encontram aplicações em diversas áreas tais como Administração, Química, Medicina, Controle de Qualidade, Linhas de Produção, Segurança e Telecomunicações.

Dentro deste contexto, uma abordagem específica de elementos pertinentes ao ambiente do sistema envolvido é necessária para determinar se o comportamento é ou não adequado. Essa análise é, de fato, uma das características trabalhadas pelos algoritmos artificiais que compõem o estudo das redes neurais no ambiente computacional.

Pode-se destacar como áreas afins desse ambiente a relação proposta por Demuth (2007), descrita abaixo:

*Aeroespacial:* Atividades e soluções envolvidas no mercado aéreo. Abordagens como pilotos automáticos de alta performance, simulações das possíveis rotas aéreas, sistemas de controle aéreo, simulação de componentes de aviões, e detecção de problemas nos aviões.

*Aplicações Financeiras:* Em decisões relacionadas à avaliação da aplicação de crédito, as empresas financeiras necessitam apurar uma quantidade de dados com demasiada complexidade. Um perfil adequado ao uso das RNA.

*Verificação da Atividade de Cartões de Crédito:* Todos os clientes das empresas de

cartões de crédito possuem um perfil de gastos construído gradativamente com o uso do cartão. Através desse perfil é possível estipular um padrão de comportamento para os clientes. Ferramentas construídas com algoritmos neurais artificiais são capazes de identificar mudanças nesse comportamento, criando possíveis indicações de fraude ou roubo do cartão.

*Segurança:* Muitas ferramentas de uso governamental relacionadas à defesa pública usufruem de algoritmos RNA. Problemas como reconhecimento de objetos, soluções de biometria, reconhecimento facial, extração de características, supressão de ruídos, entre outros, são abordados.

*Eletrônicos:* Para o desenvolvimento de ferramentas capazes de analisar o layout e as características de chips integrados, controle de processos, visão e voz eletrônicos, modelos não lineares, etc, o uso de heurísticas é muito comum.

*Entretenimento:* Uma área bastante satisfatória no mercado de hoje, altamente influenciada pelo uso da Inteligência Artificial – conceito multidisciplinar que usa recursos e algoritmos das Redes Neurais Artificiais é a área dos jogos. Várias atividades virtuais simulam o comportamento no mundo real e por isso são complexas e de difícil programação. As RNA também são utilizadas em animações, efeitos especiais, e previsão de mercado. Em geral, a área de entretenimento é muito favorecida.

*Finanças:* Atividades como predição de preço da moeda, análise de portfólio, análise de crédito, recomendação de empréstimos, análise financeira, etc, possui características intrínsecas ao campo das Redes Neurais Artificiais.

*Industrial:* A vida útil de um equipamento e seu desgaste contínuo, como também a avaliação de sua exposição ao ambiente industrial, são problemas comuns e de difícil mensuração, no qual a utilização de algoritmos neurais se torna interessante.

*Sistemas de Controle:* Com escopo também industrial, esse tema foi tratado separadamente devido a sua importância no contexto apresentado. Há mais de 20 anos, alguns sistemas de controle baseiam-se no uso das Redes Neurais Artificiais em virtude da presença de alto grau de não-linearidade, incerteza e imprecisão.

Sistemas de Controle permitem a definição de um controlador que altera o comportamento de um sistema dinâmico para modelagem de seu funcionamento real – ou seja, processo de modelar o contexto.

*Manufatura:* Através dos recursos heurísticos oferecidos pelos algoritmos de RNA, a manufatura é favorecida nos seguintes tópicos: modelagem dinâmica de sistemas de processos químicos, planejamento e gerenciamento, análise da vida útil de equipamentos, análise de design de produtos químicos, análise de qualidade de chips de computador, previsão da qualidade de papel, testes de bebidas, diagnóstico de equipamentos e processos, análise do design de produtos, etc. Em geral, a abordagem centra-se no controle do processo de manufatura.

*Medicina:* Na medicina, existem diversos tipos de análises que envolvem julgamento de uma grande massa de dados, entre esses, podem-se destacar a análise de células cancerígenas, otimização de tempo de transplantes, melhorias no atendimento médico, etc.

*Petróleo e Gás:* Dentro do ambiente universitário tem-se desenvolvido muitas pesquisas relacionadas à extração de petróleo e gás. Uma das fortes tendências está relacionada à busca e localização de possíveis poços de petróleo, assim como, no teste de qualidade de seus derivados. De acordo com (Pinheiro, 1996), a inferência de propriedades visa fornecer uma boa estimativa de propriedades de derivados de petróleo (ponto final de ebulição, pressão de vapor, etc.). Essas propriedades podem ser determinadas por analisadores de processo ou análises de laboratório. Contudo, esses sistemas nem sempre apresentam resultados satisfatórios ou na frequência necessária para permitir o controle. Porém, se o valor estimado de uma determinada propriedade estiver disponível, o mesmo pode ser utilizado para permitir o controle ou a otimização do processo produtivo.

*Robótica:* A utilização de heurísticas, especificamente as Redes Neurais Artificiais, na engenharia robótica e mecânica possui grande relevância. Problemas de trajetória, visão eletrônica, percepção do ambiente, são freqüentemente tratados dentro deste contexto.

*Linguística:* A compreensão da fala com coerência é um problema que envolve

muita complexidade. As RNA são constantemente aplicadas em reconhecimento de fala, compreensão de palavras, reconhecimento de voz, sintetizadores de voz para transposição de texto em voz, entre outras. Esse estudo também está relacionado à robótica.

*Telecomunicações:* Sistemas de análise de dados, serviços de informação automatizados, tradução em tempo real, pesquisas relacionadas ao tráfego e infraestrutura de rede, identificação e extração de ruídos no sinal, entre outros, trabalham com o tratamento e interpretação de dados sob uma forma complexa. As Redes Neurais são necessárias nesse escopo.

*Transporte:* Uma forte tendência existe na área de transporte para o uso de algoritmos que otimizam os recursos e transações envolvidas. Segundo Póvoa (2005), a logística vem se destacando como fator primordial de sobrevivência empresarial. Um completo sistema logístico abrange o processo de movimentação de matéria-prima (e outros insumos necessários à produção) de fornecedores a fábrica, o movimento destes produtos para vários centros de distribuição ou depósitos, e a entrega destes produtos ao consumidor final. Além de estudos relacionados à logística, as Redes Neurais podem ser aplicadas em estudos sobre a alocação de recursos e a detecção de problemas nos veículos.

Comumente qualquer aplicação que necessite da interpretação, agrupamento e avaliação de uma grande quantidade de dados está inserida na ciência das Redes Neurais Artificiais e atualmente é vista como fator importante para a otimização dos recursos da organização.

## 2.2 Conceitos biológicos

Todo o estudo neural artificial tem como base o comportamento neural biológico, tendo como principal participante, o neurônio.

O sistema nervoso biológico é uma poderosa máquina capaz de comandar todo o processo de recebimento de informações, interpretação das mesmas, e conseqüente resposta do sistema. Toda sua estrutura é formada para aceitar a

existência do feedback de seu comportamento, que se ajusta com o decorrer do tempo, sua interpretação dos dados.

O centro do sistema nervoso é o cérebro, que, em pontos práticos pode ser entendido como o computador de carbono mais poderoso que existe na atualidade, e que, ainda deverá manter-se nesse perfil durante muito tempo – devido sua distância dos computadores de silício modernos. Sua enorme massa de neurônios fornece por essência um alto grau de processamento paralelo e auto-adaptativo. Segundo Held (2006), os processadores dual-core estão presentes no mercado, e os quad-core estão-se iniciando. Nos anos subseqüentes o número de núcleos na arquitetura dos processadores irá crescer, almejando alcançar processadores vastamente poderosos. Estas são as máquinas que entrarão no desempenho dos teraflops (grande poder de processamento) com as potencialidades necessitadas nas aplicações do amanhã.

Vista a realidade computacional que trabalha com quatro núcleos de processamento, um ambiente com bilhões de mini-processadores, intitulados neurônios, certamente apresenta um comportamento aprimorado, quando comparado ao seu concorrente não-biológico.

Toda a entrada das informações é derivada do corpo humano e seus sentidos básicos. Essa entrada é capturada pelos receptores do sistema nervoso central. Um fluxo de informação trafega para a rede neural. Uma característica importante que ocorre nesse momento é a inferência que o cérebro realiza com os dados. Após a interpretação e julgamento dos dados, os atuadores (parte que compõe o neurônio) recebem o sinal da rede e emitem a resposta do sistema.

Pode-se considerar como base de funcionamento do sistema nervoso neural os elementos responsáveis pela recepção e inferência dos dados, e ainda aquele que exprime, através dos atuadores, os resultados do sistema.

Observe na Figura 2.1 uma representação em forma de diagrama do sistema nervoso neural. Os estímulos externos ao sistema, adquiridos pelo corpo humano, estão representados pela seta de entrada e são atendidos pelos receptores. O tráfego dos dados prossegue seu percurso para o sistema neural,

que extrai o que lhe é importante, toma decisões apropriadas e envia o resultado para o terceiro elemento. Os atuadores são responsáveis pela composição das respostas. Através deles o sistema nervoso demonstra sua posição ao ambiente.



Figura 2.1 – Sistema Nervoso – Haykin (2001)

Observa-se que o comportamento do sistema também é alimentado pelo feedback que, na Figura 2.1, está expresso através das setas com origem na direita e destino na esquerda. Neste contexto, o feedback representa a característica auto-adaptativa do sistema neural. Isso significa, em outras palavras, que o modelo neural interpreta as características do ambiente e é capaz de modificar-se, concebendo assim, o *aprendizado*.

Ao contrário dos modelos computacionais comumente utilizados, em que um programa precisa ser escrito para resolver um dado problema, as Redes Neurais aprendem a resolver problemas através da sua interação com o meio externo, conforme afirmado por Braga *et al.* (2000). Segundo Barreto (2002), 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.

De acordo com Santos, Milano e Rosat (1998), a retenção e evocação de uma informação aprendida é chamada de memória que, junto com o processo de aprendizagem, envolve a aquisição de novos padrões funcionais no sistema nervoso, como consequência da apresentação de uma situação nova ou repetida, a qual é expressa por mudanças de respostas do organismo. Ou seja, todo o processo que é inserido no contexto de aprendizagem de um sistema nervoso neural modificará a estrutura do sistema de acordo com sua interpretação dos dados do ambiente. Tudo posteriormente tratado por esse sistema será

influenciado pelas adaptações prévias.

Toda essa estrutura é formada pelos neurônios e suas interligações. Pode-se considerá-lo como elemento primordial para a elaboração do aprendizado. Um neurônio é uma unidade de processamento de informação que é fundamental para a operação de uma rede neural, Haykin *et al.* (2001).

Com embasamento na neuroanatomia, o neurônio é uma célula biológica de grande importância no sistema nervoso central. Segundo Arone, Philippi e Vono (1994), os neurônios são as unidades básicas do sistema nervoso e possuem as propriedades de excitação e condução. Sua estrutura celular é composta pelos seguintes elementos: corpo celular, dendritos, sinapses, corpúsculo de Nissl, axônios, mielina, neurilema e nódulo de Ranvier, conforme ilustrado na Figura 2.2. Cada um desses componentes possui características e funções específicas no neurônio. Tem-se abaixo uma breve descrição biológica dos elementos.

*Corpo Celular e Corpúsculo de Nissl:* Também conhecido como *pericário* ou *soma*, o corpo celular do neurônio comporta-se, juntamente com o corpúsculo de Nissl, como catalisador. Ou seja, elemento que tem a propriedade de acelerar ou retardar a velocidade de uma reação química sem se alterar no decorrer deste processo. Na prática, o corpo celular do neurônio recebe os dados de entrada e realiza uma espécie de *somatório*. Outras características biológicas existem, mas não trazem importância científica pertinente para este estudo.

*Dendritos:* O corpo do neurônio possui ramificações para que os sinais de entrada possam encontrar seu núcleo. Essas ramificações são conhecidas como dendritos. De acordo com Barreto (2002), as ramificações conhecidas como dendritos, conduzem sinais das extremidades para o corpo celular. Nessas extremidades, encontram-se as sinapses.

*Axônio:* Assim como os dendritos, o axônio oferece ao neurônio uma estrutura de tráfego de sinais eletroquímicos, porém, com o papel de saída. Conforme apresentado nos tópicos anteriores, o neurônio recebe os estímulos de entrada através dos dendritos, é capaz de analisar os elementos em seu corpo e por fim, enviar sua influência para os outros milhares de neurônios vizinhos através do



canal de saída denominado axônio. Segundo Portugal e Fernandes (1995), cada célula possui uma única saída (axônio), a qual pode se ramificar em muitas ligações colaterais (cada ramificação possuindo o mesmo sinal de saída do neurônio). Todos os elementos do neurônio se interagem, conforme proposto por Soto (1999): os neurônios reais têm um corpo celular, um axônio e muitos dentritos, sendo o axônio uma protuberância que transporta a saída do neurônio até aos dentritos de outros neurônios. Os dentritos são protuberâncias que proporcionam uma área abundante e assim facilitam a conexão com os axônios de outros neurônios.

*Mielina e Neurilema:* Algumas fibras nervosas são envolvidas por uma substância formada pelas células de Schwann. Essa substância fornece a capacidade de uma condução satisfatória que permite a condução dos impulsos de forma mais rápida. A mielina está presente nessa camada. Segundo Arone, Philippi e Vono (1994), a mielina é uma fonte nutritiva e isolante da célula nervosa. A neurilema está presente imediatamente por fora da camada de mielina. Ela é uma membrana que envolve a bainha de mielina. Serve como tubo protetor, através do qual, novas fibras pode crescer, Arone, Philippi, Vono *et al.* (1994). Toda essa membrana envolve o axônio. De acordo com Prado (2004), os axônios são cobertos por bainha neurilemal de Schwann. Nas fibras mais mielinizadas, a célula de Schwann, através de uma rotação, forma uma estrutura multilaminar que envolve, com uma bainha de mielina, um único axônio.

*Nódulo de Ranvier:* O nódulo de Ranvier é referenciado como o contato da neurilema com a parte condutora do axônio, Arone, Philippi, Vono *et al.* (1994). Em prática, a extensão do axônio que não está envolvida pela membrana das células de Schwann, sendo denominada nódulo de Ranvier.

*Sinapses:* Todos os neurônios do cérebro estão conectados. A sinapse pode ser entendida como a mediadora dessas conexões. Estima-se que, em média, cada neurônio possui 10.000 conexões. Em cada sinapse, o neurônio é capaz de armazenar informações pertinentes à sua interação com o ambiente. Isso significa, em outras palavras, que num primeiro momento, o neurônio não possui influência sobre o sinal que recebe do sistema nervoso. A partir do momento que interage com o meio, ele é capaz de modificar o poder de suas sinapses, influenciando a

próxima entrada com o intuito de excitá-la ou inibi-la.

Portanto, os neurônios estão interligados pelas sinapses. Quando um sinal é disparado, ele atravessa uma sinapse que, de acordo com o seu grau de influência, aumentará ou diminuirá o valor do mesmo.

Nessa região ocorre a liberação de substâncias químicas que irão excitar ou inibir o próximo neurônio, modificando sua atividade elétrica. O estímulo é enviado ao longo do axônio, permitindo o fluxo de  $\text{Na}^+$  para o interior da membrana e o de  $\text{K}^+$  para o exterior. O axônio volta ao estado original quase imediatamente (em milésimos de segundo), ficando pronto para receber um novo estímulo. As principais substâncias químicas que atravessam as sinapses são acetilcolina e adrenalina, Arone, Philippi, Vono *et al.* (1994).

Ao observar, a Figura 2.2, constata a representação dos elementos descritos. A partir dessa representação pode-se associar com mais facilidade o tipo de interação que cada um dos componentes representa para o todo, no caso, o neurônio.

Através dessa abordagem, pode-se entender que o processo de inferência e conseqüente aprendizagem fornecidos pelas células neurais é complexo e envolve o processamento distribuído oferecido essencialmente pela estrutura do córtex. Em resumo, a operação inicia-se quando estímulos originados da rede neural (ou seja, outros neurônios) atravessam as sinapses, que injetam determinado grau de influência no sinal concretizando uma excitação ou inibição para o neurônio receptor. Fica claro que esta influência é definida pelas interações com o ambiente decorrentes da vida do neurônio. Posteriormente os sinais eletroquímicos, mais precisamente compostos por sódio e potássio, avançam pelos dendritos e finalmente chegam ao corpo celular. O neurônio recebe tais sinais e expõe sua influência para o restante da rede através do canal de saída oferecido pelo axônio e suas membranas protetoras. Ao final deste axônio existem milhares de filamentos que estão ligados, por intermédio das sinapses, aos próximos neurônios.

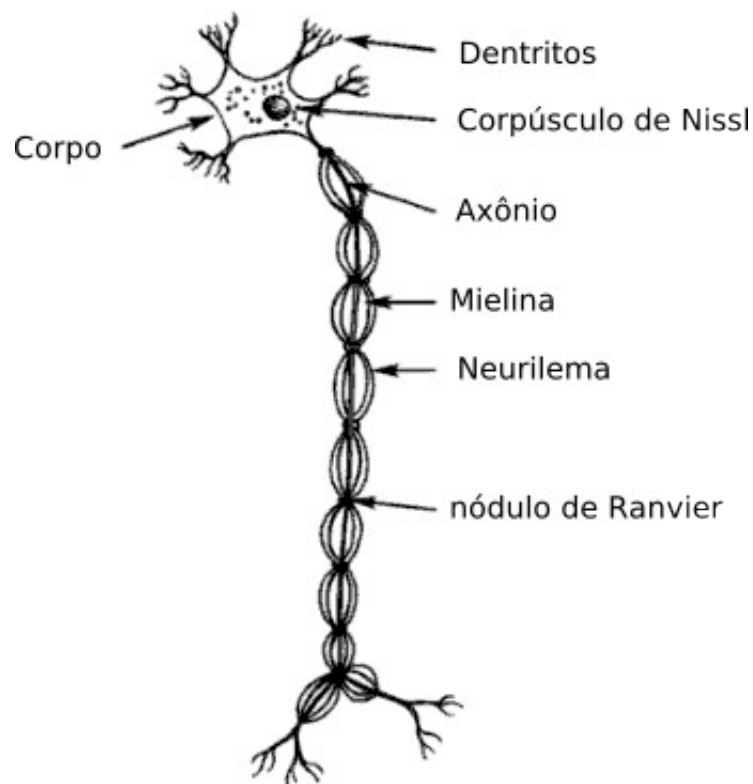


Figura 2.2 – Neurônio. Arone, Philippi e Vono (1994).

Em uma abordagem geral do neurônio, pode-se destacar como essencial, entre todos os seus elementos, apenas o canal receptivo (dentritos), o corpo celular, o axônio e sinapses. Através desses elementos é possível concretizar o comportamento do neurônio perante a rede neural biológica. De acordo com a abordagem proposta por Paula (2000), a maioria dos neurônios, independentemente do seu tipo, possuem em comum quatro regiões funcionais: um elemento de recepção ou entrada, um elemento de ativação, um componente condutor e um elemento emissor. A organização funcional dos neurônios, portanto, pode ser representada por um neurônio modelo. Cada componente gera um sinal característico: a entrada, a ativação e os sinais propagáveis são elétricos; enquanto que, o sinal emitido é a liberação de um transmissor químico na fenda sináptica. Durante o decorrer deste trabalho, principalmente na abordagem matemática da estrutura neural, apenas esses quatro componentes primários serão relevantes.

## 2.3 Neurônio Artificial

Certamente nem todo aparato biológico presente no neurônio proposto por Arone, Philippi e Vono (1994), representado na Figura 2.2, servirá como componente para o modelo fictício, conhecido como *Neurônio Artificial*. Os fatores de maior importância são os *dentritos*, *sinapses*, *corpo do neurônio* e *axônio*. A importância biológica agregada a esses componentes permanece presente e, obviamente, servi como base para a contextualização matemática proposta a seguir:

*Dentritos*: Mantém a característica de elemento que permite a entrada dos estímulos (dados) no neurônio. Vários autores, descritos posteriormente, sugerem a apresentação dos estímulos como um vetor formado por elementos escalares, positivos ou negativos, em uma unidade de tempo discreta. Pela notação matemática definida neste trabalho, o vetor de entrada, também denominado *vetor-chave*, é referenciado por  $x$ . A representação do tempo, em geral, será abordada como  $n$ , limitada pelo primeiro instante 1 e última ocorrência de iteração  $t$ .

*Sinapses*: Estes elementos são de grande importância, tanto no contexto biológico, quanto artificial, pois é através deles que o neurônio é capaz de ponderar os valores de entrada, influenciando-os com sua própria abstração do ambiente e conseqüentemente apresentando à rede sua participação no resultado. Sob a contextualização matemática, as sinapses são conhecidas como *pesos* ou *pesos sinápticos* e, geralmente estão associadas ao vetor de valores escalares  $w$ . Sob o contexto de muitos neurônios (rede), os pesos são apresentados através da *matriz sináptica*  $W$ .

*Corpo do Neurônio*: O principal objetivo do corpo do neurônio é receber todos os sinais de entrada e realizar uma espécie de validação, no qual avalia o potencial dos estímulos e produz uma saída significativa quando necessário. Na realidade, a saída será apresentada somente se os valores de entrada satisfizerem um *limiar*. Em caso negativo, a saída apresentada não produz importância para os próximos neurônios. Na forma matemática, o corpo do neurônio estabelece um somatório de todos os estímulos já ponderados pelas respectivas sinapses. Isso pode ser entendido como uma combinação linear entre os vetores de entrada  $x$  e a matriz de

pesos  $W$ . Através da equação (2.1), esse comportamento pode ser entendido. O valor produzido pelo somatório é conhecido como *campo local induzido* e é representado por  $v$ .

$$v = \sum_{i=1}^m w_i x_i \quad (2.1)$$

Observe que  $m$  representa a extensão do vetor de entrada e que, neste caso, existe apenas um neurônio associado que por consequência produzirá um valor escalar como *campo local induzido*. Outro objetivo biológico é estabelecer um limiar como intuito de limitar a resposta do neurônio. Na forma matemática, esse limiar é produzido por funções apropriadas, comumente conhecidas como *função de ativação* ou *função de disparo*. Existem várias funções sob o contexto das redes neurais artificiais. A escolha mais apropriada é realizada de acordo com o modelo trabalhado e a problemática envolvida. Respectivamente, através das equações (2.2), (2.3) e (2.4), estão representadas as funções de ativação: *Função de Limiar*, *Função Linear* e *Função Sigmóide*.

$$f(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases} \quad (2.2)$$

$$f(v) = v \quad (2.3)$$

$$f(v) = \frac{1}{1 + e^{(-\alpha v)}} \quad (2.4)$$

O comportamento da *função sigmóide* trabalha com um parâmetro  $\alpha$  denominado *parâmetro de inclinação* – associado à inclinação da curva produzida pela função. O parâmetro de inclinação é responsável pelo comportamento da função sigmóide e conseqüentemente pela importância aplicada em  $v$ , o que justifica uma atenção em particular na sua definição durante a elaboração da rede.

*Axônio*: O axônio representa o sinal de saída produzido pelo neurônio. No modelo, ele poderá ter um comportamento escalar, ou vetorial, estando diretamente associado ao número de neurônios na *camada de saída* – as redes são organizadas em camadas, conforme demonstrado adiante. A representação da saída da rede é denotada por  $y$ . Através da equação (2.5), pode-se observar o

comportamento matemático de um neurônio artificial.

$$y=f\left(\sum_{i=1}^m w_i x_i\right) \quad (2.5)$$

Esta metodologia é favorável a muitos modelos artificiais que serão brevemente descritos no próximo capítulo. Uma notação interessante também adotada é a notação gráfica, aqui representada através da Figura 2.3.

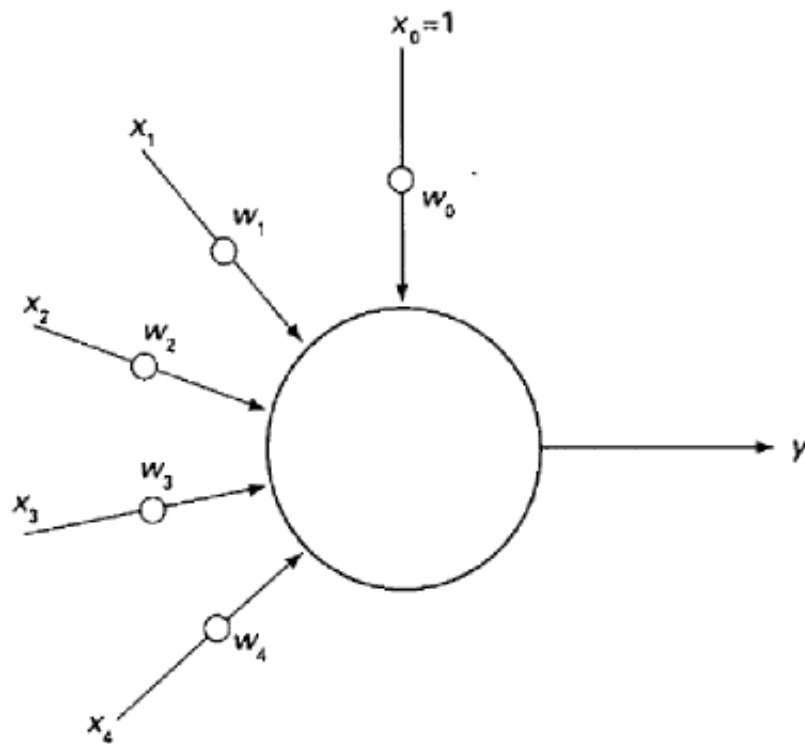


Figura 2.3 – Neurônio Artificial. Veelenturf (1995).

Observa-se na Figura 2.3 que, além dos estímulos naturais associados aos índices 1-4, existe a apresentação do elemento  $x_0$  que por padrão, assume o valor 1. Este indicador representa o *bias*. O bias pode ser considerado como um peso, com valor de entrada fixo em 1. Ele adiciona seu valor ao produto  $wx$  e oferece uma alternativa de deslocamento para o comportamento da *função de ativação*, Demuth, Beale, Hagan *et al.* (2007). Com acréscimo do bias, o campo local induzido sofre conseqüente alteração, conforme observado na equação (2.6).

$$v = \sum_{i=1}^m w_i x_i + b \quad (2.6)$$

Este é um comportamento base para um neurônio artificial. Outros modelos serão apresentados no decorrer do estudo.

## 2.4 Arquiteturas de Rede

Os neurônios artificiais sozinhos não apresentam grande poder de processamento. Geralmente, em uma rede neural artificial, muitos neurônios estão conectados entre si, com o intuito de oferecer a característica de processamento paralelo, presente na rede biológica. As formas nas quais, os neurônios estão organizados, é dita como *Arquitetura da Rede*.

A maneira pelas quais os neurônios de uma rede neural estão estruturados está intimamente ligada com o algoritmo de aprendizagem usado para treinar a rede, Haykin *et al* (2001). Dependendo da arquitetura adotada, na fase de treinamento, a saída da rede é afetada pela atividade de todos os neurônios, e a correção dos parâmetros atinge potencialmente todas as conexões, Timoszczuk *et al*. (2004). Fundamentalmente, pode-se classificar as estruturas de redes neurais em três tipos distintos: *Redes alimentadas adiante com Camada Única*, *Redes alimentadas diretamente com Múltiplas Camadas* e *Redes Recorrentes*.

De acordo com Barbosa (2005) e Haykin (2001), na forma mais simples de uma rede neural em camadas, tem-se uma camada de nós de entrada que se projeta sobre uma camada de saída de neurônios, mas não vice-versa. Esta é uma arquitetura típica, conforme apresentada na Figura 2.4. O número de neurônios existentes nesta estrutura está associado unicamente, aos neurônios da camada de saída, pois, os nós de entrada não representam nenhum tipo de processamento. Na realidade, os nós de entrada estabelecem o vetor de entrada  $x$ , onde nenhum comportamento da rede é estabelecido.

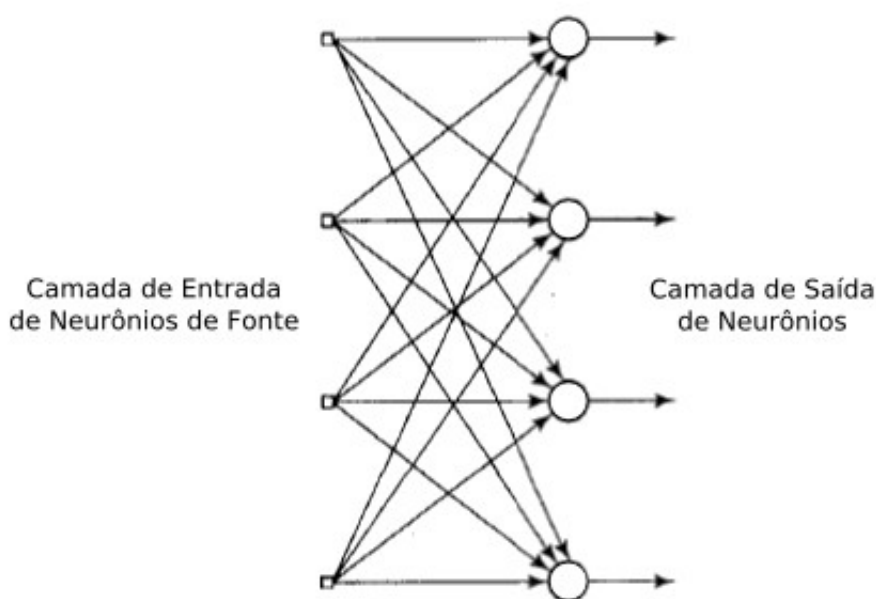


Figura 2.4 – Rede Alimentada adiante com uma camada. Haykin (2001).

O segundo modelo de arquitetura, conforme sugerido pelo nome, é composto por várias camadas. Em comparação com o modelo de camada única, pode-se destacar que a *Rede Alimentada Adiante com Múltiplas Camadas* também apresenta uma camada de entrada e saída. Sua distinção está na presença de camadas ocultas que intermeiam os estímulos e saída da rede. No comportamento sugerido por Haykin (2001), os nós de fonte da camada de entrada da rede fornecem os respectivos elementos do padrão de ativação (vetor de entrada), que constituem os sinais de entrada aplicados aos neurônios (nós computacionais) na segunda camada (isto é, a primeira camada oculta). Os sinais da saída da segunda camada são utilizados como entradas para a terceira camada, e assim por diante para o resto da rede. Outros autores, assim como Demuth, Beale e Hagan (2007), consideram como *camada de entrada* a primeira camada oculta e não associa os vetores de estímulos (a entrada da rede) como uma camada em separado. Uma estrutura favorável à resolução de problemas complexos é formada por apenas duas camadas ocultas – ou, na notação sugerida por Demuth, Beale e Hagan (2007), uma camada oculta, sendo a primeira, a camada de entrada – e uma camada de saída. Adicionando-se uma ou mais camadas ocultas, tornamos a rede capaz de extrair estatísticas de ordem elevada. Observe uma estrutura de rede alimentada adiante com múltiplas camadas através da Figura 2.5.



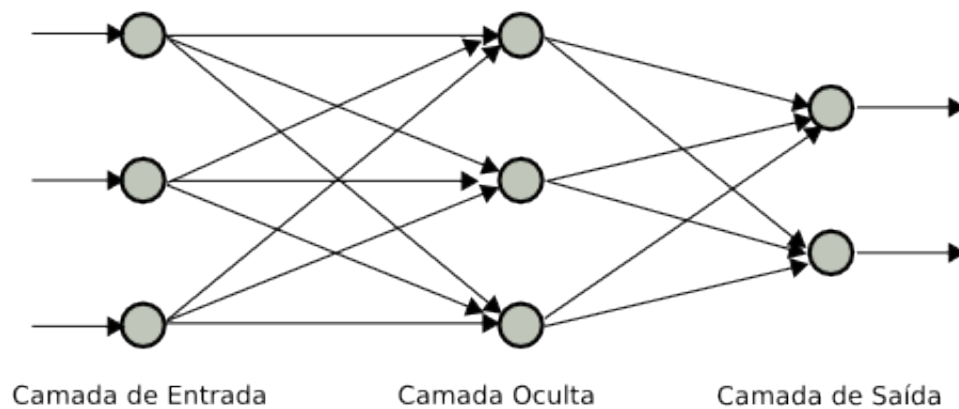


Figura 2.5 – Rede Alimentada adiante com múltiplas camadas. Boothe (2002).

Observe que a notação sugerida na Figura 2.5 trata a primeira camada oculta como camada de entrada. Por fim, a terceira notação conceitual pode ser observada na Figura 2.6, onde uma *Rede Recorrente* é apresentada.

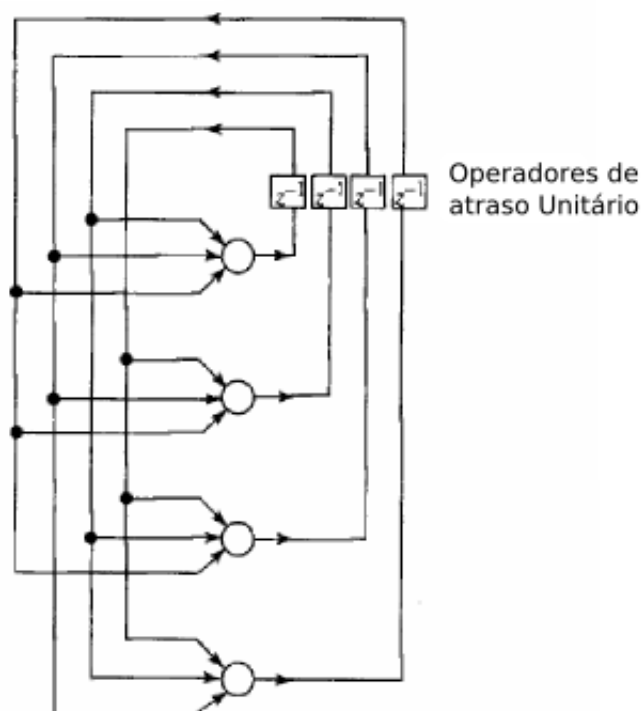


Figura 2.6 – Rede Recorrente. Haykin (2001).

Uma rede neural recorrente se distingue de uma rede neural alimentada adiante por ter pelo menos um laço de *realimentação*. Uma rede recorrente pode

consistir, por exemplo, de uma única camada de neurônios com cada neurônio alimentando seu sinal de saída de volta para as entradas de todos os outros neurônios, Haykin *et al.* (2001). Esta é a perspectiva apresentada na Figura 2.6. Além disso, os laços de realimentação envolvem o uso de ramos particulares compostos de *elementos de atraso unitário* (elementos, representados por  $z^{-1}$ , que fornecem as alterações relacionadas aos períodos em que os dados são apresentados à rede), o que resulta em um comportamento dinâmico não-linear, admitindo-se que a rede neural contenha unidades não-lineares, Haykin *et al.* (2001). As redes dinâmicas podem ser divididas em dois tipos de categorias: aquelas que apresentam comportamento *feedforward*, ou de acordo com a notação trabalhada, alimentadas diretamente; ou aquelas que possuem *feedback*, ou redes recorrentes, Demuth, Beale, Hagan *et al.* (2007). As redes dinâmicas geralmente são mais favoráveis ao comportamento representado pelo tempo, muito comum em representação de sistemas dinâmicos.

## 2.5 Aprendizagem

Muitos fatores presentes no ambiente formado pelas redes neurais artificiais as tornam ferramentas extremamente eficazes, contudo, dentre tais qualidades a que mais se destaca é a característica inata, herdada do contexto biológico, conhecida como *aprendizagem*. A aprendizagem é um processo fabuloso, no qual o sistema é capaz de adquirir conhecimento a partir de suas interações com o ambiente em que está envolvido. Em geral, essa é uma característica exclusiva dos seres vivos, somente acessível pelos comportamentos assumidos pelo cérebro. A rede neural artificial, em sua busca pela representação de tais comportamentos, também incorpora, com devidas adaptações matemáticas, o aprendizado.

A noção de aprendizado, apesar de muito importante, é extremamente simples. Ela caracteriza uma alteração sofrida pelo sistema (rede neural biológica ou artificial) a partir de sua vivência. Todos os estímulos apresentados à rede são por ela captados, e avaliados. A partir desses dados ela é capaz de se ajustar e conseqüentemente, tornar-se mais apta a trabalhar com entradas de mesma conceituação. Isso, em termos mais práticos, significa que a rede neural é capaz de ajustar seus pesos sinápticos (elementos de ponderação geralmente associados

com a memória) de acordo com o comportamento induzido pelos estímulos. Isso trará a ela, no futuro, uma visão modificada dos estímulos. Ou seja, ela *aprendeu* através das entradas e agora é capaz de interpretá-las de forma diferente. Uma boa analogia é dada da seguinte forma:

*(1) Uma criança queima sua mão ao tocar a tampa quente de uma panela. A próxima interação com esse objeto será distinta. (2) A criança não correrá o risco de se queimar novamente, tendo adquirido o conhecimento necessário em uma instância anterior.*

Pode-se entender como *treinamento* o fato (1), onde o sistema recebe os dados do ambiente e a partir dele monta seu conhecimento. O instante (2), o *aprendizado* já está concretizado. Isso significa que o sistema é capaz de interpretar os dados do ambiente com uma nova perspectiva. Segundo a definição de Haykin (2001), a aprendizagem pode ser conceitualmente definida como:

*Aprendizagem é um processo pelos quais os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira com que a modificação dos parâmetros ocorre.*

Na prática o aprendizado envolve uma modificação dos pesos sinápticos. O método de definição dos valores dos pesos (treinamento) é uma característica de distinção importante para as redes neurais artificiais. Por conveniência, podem-se distinguir dois tipos de treinamento: *aprendizado supervisionado* e *aprendizado não-supervisionado*, Fausett *et al.* (1994).

*Aprendizado supervisionado*: Neste tipo de treinamento (ou aprendizado) a rede neural recebe seus estímulos e condiciona-os através dos pesos. O resultado da combinação linear é acrescido pelo bias, como resultado final o campo local induzido. A resposta da rede (limitação do campo local induzido pela função de ativação presente) é comparada com um elemento alvo, ou *saída desejada*. A existência de uma distância entre o *valor alvo* e a *saída real* da rede é sempre presente; isso é denominado *erro*. Esse erro é usado para mensurar de forma apropriada uma possível modificação para os pesos sinápticos, com intuito de

diminuir a distância entre a próxima saída real e o alvo. Esse comportamento é conhecido como *aprendizado supervisionado*, pois existe uma entidade *supervisora* que apresenta para a rede a saída desejada.

Em termos conceituais, pode-se considerar o *professor* como tendo conhecimento sobre o ambiente, com este conhecimento sendo representado por um conjunto de *exemplos de entrada-saída*. Entretanto, o ambiente é desconhecido pela rede neural de interesse. Suponha agora que o professor e a rede neural sejam expostos a um vetor de treinamento (isto é, um exemplo) retirado do ambiente. Em virtude do seu conhecimento prévio, o professor é capaz de fornecer à rede neural uma resposta desejada para aquele vetor de treinamento. Na verdade, a resposta desejada representa a ação ótima a ser realizada pela rede neural. Os parâmetros da rede são ajustados sob a influência combinada entre a resposta desejada e a resposta real da rede, Haykin *et al.* (2001). O que pode ser observado, através da Figura 2.7, sobre o comportamento deste tipo de aprendizado.

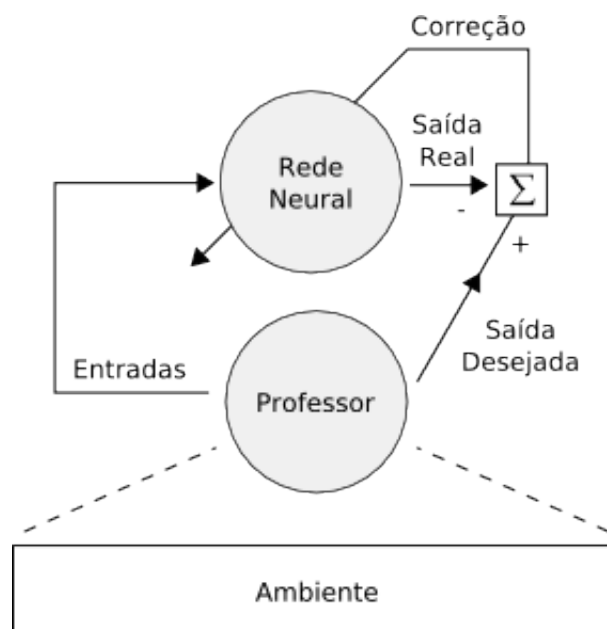


Figura 2.7 – Aprendizado supervisionado.

*Aprendizado não-supervisionado*: De forma distinta do aprendizado supervisionado, onde um professor estabelece para a rede, uma seqüência de treinamento, no

aprendizado não-supervisionado, a rede é capaz de abstrair as características implícitas no ambiente e reestruturar seu comportamento – ou, em outras palavras, ajustar seus pesos sinápticos. Uma seqüência de vetores de entrada é provida, mas nenhum vetor de alvo é especificado. A rede modifica os pesos que mais se aproximarem dos vetores de entrada associados à mesma unidade de saída, Fausett *et al.* (1994). Na aprendizagem não-supervisionada ou *auto-organizada*, não há um professor externo ou um crítico para supervisionar o processo de aprendizado. Em vez disso, são dadas condições para que a Rede possa realizar uma *medida independente da tarefa* da qualidade da representação que a rede deve aprender, e os parâmetros livres da rede são otimizados em relação a esta medida, Haykin *et al.* (2001). Esse tipo de comportamento é facilmente entendido através da representação gráfica expressa na Figura 2.8.

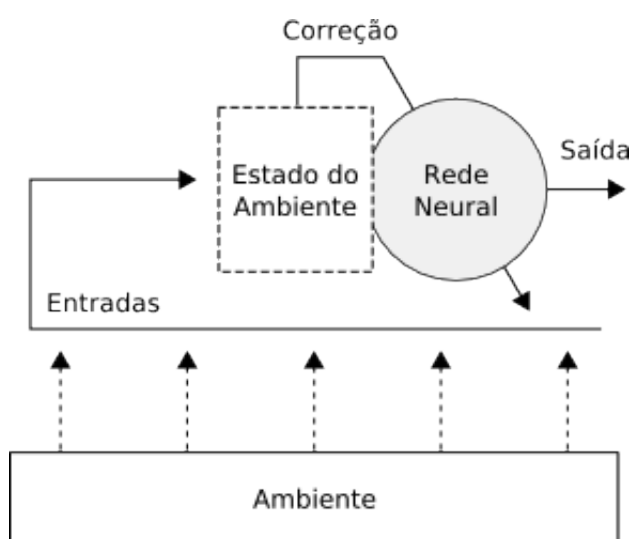


Figura 2.8 – Aprendizado não-supervisionado.

Do ponto de vista prático, os dois comportamentos, sugerem algumas vantagens e normalmente são direcionados a aplicações específicas. Contudo, nos dois casos apresentados – e suas derivações – a rede é capaz de adquirir conhecimento através do ambiente. Esse é o fator de maior importância agregado à inteligência das redes neurais artificiais, mesmo estando potencialmente distante do comportamento real do cérebro.

### 3. Estado da Arte

O tratamento em separado do referencial teórico agregado aos conceitos neurais estudados neste trabalho é justificado pela grande massa de informação com relativa importância para a dissertação. Alguns tópicos de maior interesse serão abordados na íntegra, com exceção dos modelos de maior destaque neste estudo – *perceptron* e *backpropagation*, que serão abordados em detalhes, nos capítulos posteriores.

Sob o ponto de vista matemático, todo o percurso concluído pela ciência das Redes Neurais Artificiais baseia-se em estudos pertinentes à área biológica, sendo presente, em alguns pontos específicos, a necessária adaptação do modelo através de comportamentos numéricos que, em maioria, são aplicados aos algoritmos de aprendizagem (ou treinamento). Certamente a distância entre o comportamento real e o modelo de aprendizagem neural artificial (como em todos os modelos matemáticos) é extremamente grande – fator que impulsiona o uso de ferramentas matemáticas distintas.

O estudo biológico do funcionamento do sistema nervoso iniciou-se com as concepções adquiridas no trabalho pioneiro oferecido por um importante histologista e médico chamado Santiago Ramón y Cajál, no início do século passado, López Piñero *et al.* (2006). Em meados de 1904, através da publicação de sua obra de caráter e conhecimento maior, intitulada *Texturas do sistema nervoso dos homens e dos vertebrados*, ele introduziu a idéia dos neurônios como constituintes estruturais do cérebro, Haykin *et al.* (2001). Através desse trabalho, o autor, juntamente com o italiano Golgi, conquistou o Prêmio Nobel in Physiology or Medicine (1906).

Suas descobertas, conforme apresentado em Ramón y Cajal (1906), concentram-se nos neurônios. Em sua definição, as células nervosas são entidades morfológicas. No trabalho científico, foi aplicado o método de Golgi – associado ao criador Camillo Golgi – no cérebro. A partir daí foi possível observar relativamente o arranjo dos terminais nervosos celulares. Essas fibras, além de serem diversas vezes ramificadas e presentes em torno do axônio, sempre

prosseguem em direção ao corpo neural, ou para as extensões protoplasmáticas ao redor. Nesse estudo, foi inserido nas pesquisas neurobiológicas, as descobertas sobre os processos conectivos das células pertencentes à medula espinhal, como também, as mudanças básicas sofridas pelos neurônios durante o funcionamento do sistema nervoso, Ramón y Cajal *et al.* (1906).

A comunidade acadêmica biológica continuou seu avanço em pesquisas pertinentes à área. A próxima descoberta de caráter importante para a contextualização das Redes Neurais Artificiais só ocorreu, praticamente, meio século depois das descobertas de Ramón e Cajal. Neste ponto, segundo Haykin, (2001), tem início a era moderna das redes neurais através dos conceitos introdutórios pertinentes à construção matemática do comportamento neural biológico, associados aos autores McCulloch e Pitts, no ano de 1943.

Warren McCulloch foi um psiquiatra e neuroanatomista por treinamento; passou 20 anos refletindo sobre a representação de um evento no sistema nervoso. Walter Pitts foi um prodígio que se associou a McCulloch em 1942, Haykin *et al.* (2001).

O trabalho inicial desenvolvido por esses pesquisadores data de 1943 e versa sobre a concepção de tratamento linear através de uma estrutura de neurônio artificial simples, conhecida como Unidade Lógica de Limiar (Linear Threshold Unit, ou simplesmente *LTU*). Segundo Mangasarian (1993), a mais simples e antiga rede neural é a *Linear Threshold Unit*. Este estudo combina as ciências da neuro-psicologia e lógica matemática, usando uma característica de disparo denominada *tudo-ou-nada*, para modelos de neurônio num comportamento de tempo discreto, Arbib *et al.* (2003).

McCulloch e Pitts projetaram o que hoje se considera como as primeiras redes neurais. Esses pesquisadores reconheceram que a combinação de muitos neurônios simples numa rede neural é a fonte de aumento do poder computacional Fausett (1994).

O princípio básico concebido por McCulloch e Pitts em 1943, segundo Mangasarian (1993) e Höffgen, Simon e Horn (1995), baseia-se na adoção de uma

função de ativação por limiar (linear threshold function) que condiciona o resultado do neurônio em  $\{1,0\}$ , dentro do espaço real  $n$ -dimensional,  $R^n$ . Uma notação sugerida por Haykin (2001) representa a *função de ativação por limiar* como  $\varphi$ , onde seu comportamento pode ser expresso conforme apresentado na equação (3.1), abaixo.

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq \theta \\ 0, & \text{se } v < \theta \end{cases} \quad (3.1)$$

A expressão (3.1) define o comportamento da função, que, está limitado ao escopo  $\{1,0\}$ .  $\theta$  define o limiar (threshold) da função.  $v$  expressa o resultado da combinação linear e é, segundo Haykin (2001), denominado *campo local induzido*. Sob o aspecto biológico, o campo local induzido representa a chegada dos estímulos ponderados pelas sinapses ao corpo do neurônio. O comportamento matemático é expresso pela equação (3.2), a seguir:

$$v(n) = \sum_{j=1}^m w_{kj} x_j(n), k=1 \quad (3.2)$$

O conjunto de  $j$  elementos  $x$ , associa-se aos estímulos de entrada do neurônio. Observa-se que eles são ponderados pelos respectivos pesos sinápticos  $w$ , durante o tempo discreto  $n$ . De acordo com Fausett (1994), os pesos de um neurônio são definidos para que o neurônio execute uma função particular de lógica simples, com neurônios diferentes que executam funções diferentes.

Assim, como na biologia, o neurônio artificial McCulloch-Pitts trabalha associado ao *tempo*. A idéia básica é dividir o tempo como unidades comparáveis para um período refratário de modo que, em cada período de tempo, um valor é gerado pelo axônio do neurônio em questão, Arbib *et al.* (2003). Certamente o neurônio McCulloch-Pitts opera em um tempo discreto  $n=1,2,3,\dots$ , representando as iterações, enquanto o comportamento biológico trabalha na casa dos milissegundos. O fluxo de informação através da rede supõe uma etapa de unidade de tempo para que o sinal viaje para o próximo neurônio. Essa instância de tempo permite que a rede modele alguns processos biológicos, como uma percepção de calor e frio, Fausett *et al.* (1994).



A principal abordagem do neurônio McCulloch-Pitts refere-se à circuitos lógicos. De acordo com Arbib (2003), os pesos e a limiar podem ser definidos para que os neurônios realizem o comportamento das funções lógicas AND, OR e XOR. Como resultado, o neurônio projetado por McCulloch e Pitts são suficientes para construir redes que podem funcionar como um circuito de controle para um computador, exibindo saídas computacionais de complexidade arbitrária. A Figura 3.1 (a) demonstra o comportamento do neurônio para resolver a lógica AND. Respectivamente, as Figuras 3.1 (b) e (c) exibem o comportamento das estruturas lógicas OR e XOR.

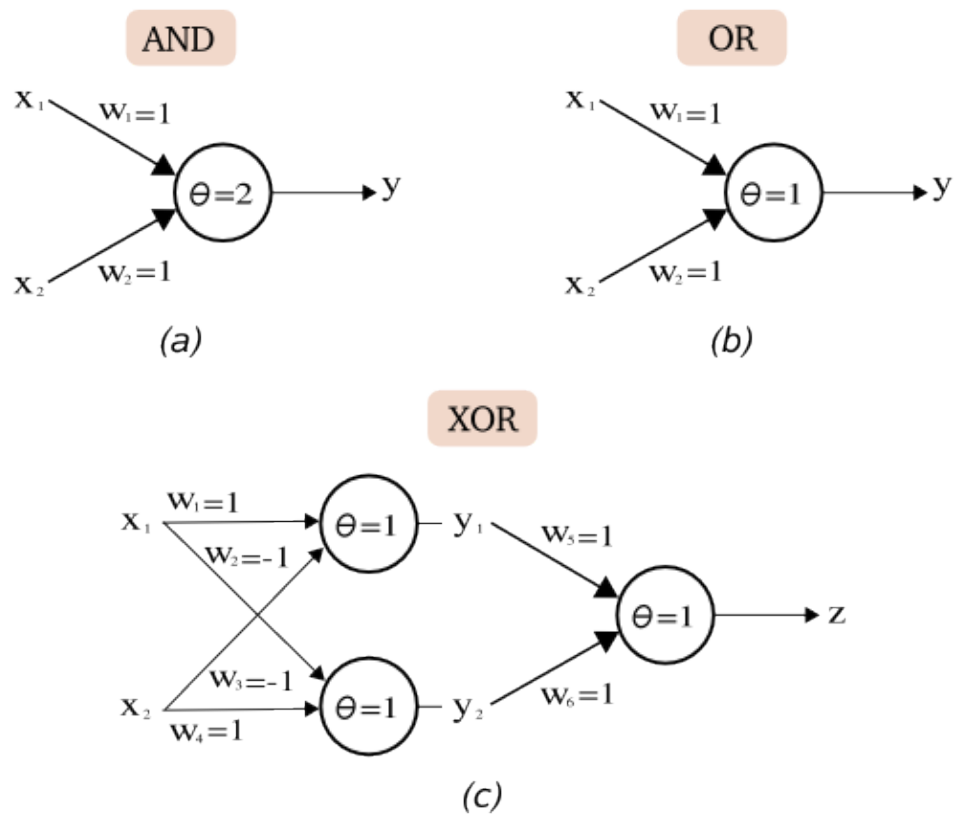


Figura 3.1 – Circuitos Lógicos AND, OR e XOR

Os dentritos  $x_1$  e  $x_2$  assumem as possibilidades lógicas conforme a Tabela 3.1. Observe também o comportamento das lógicas AND, OR e XOR.

$x_1$	$x_2$	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Tabela 3.1 – Estruturas Lógicas AND, OR e XOR

Ao observar, a Figura 3.1 percebe-se que, as estruturas lógicas pertinentes ao AND e OR são formadas por apenas uma unidade de processamento (neurônio). Através de seus respectivos resultados apresentados na Tabela 3.1, pode-se facilmente comprovar que com uma reta – associada ao neurônio linear – é possível separar seus disparos (resultados) em dois grupos distintos.

A sinapse  $w$  é um vetor de pesos fixos em  $R^n$  e o limiar  $\theta$  é um número real fixo. Geometricamente uma LTU pode ser representada pelo plano  $w \cdot x = \theta$  em  $R^n$ . Cada plano pode ser dividido em dois espaços de conjuntos exclusivos. Segundo essa definição, observa-se, de forma óbvia, que uma LTU pode ser usada para discriminar dois conjuntos exclusivos definidos como  $A$  e  $B$  em  $R^n$ , cada um definindo seus valores em um dos espaços. Essa definição é conhecida como *linearmente separáveis*, Mangasarian *et al* (1993).

Todavia, o comportamento lógico XOR não poderá ser resolvido com um neurônio. Isso implica que, a Unidade Lógica de Limiar sugerida por McCulloch e Pitts não é capaz de resolver problemas que não são linearmente separáveis.

O próximo desenvolvimento significativo das redes neurais veio em 1949, com a publicação do livro de Hebb, denominado *The Organization of Behavior*, no qual foi apresentada pela primeira vez uma formulação explícita de uma regra de aprendizagem fisiológica para a *modificação sináptica*, Haykin *et al.* (2001). O *postulado de Hebb* é a mais antiga e simples regra de aprendizado de uma rede neural, Fausett *et al.* (1994).

Conhecido como o pai da neuropsicologia e das redes neurais, Donald Olding Hebb representa um forte marco no conceito da aprendizagem. Iniciou-se, através de sua contribuição, a compreensão de como o funcionamento dos

neurônios condiciona alguns processos psicológicos, tais como a aprendizagem. Ele foi uma figura de grande importância para área de psicologia.

Donald Olding Hebb encarava a psicologia como uma ciência biológica e sua proposta sobre as células conjuntas rejuvenesceu o interesse na psicologia. Desde sua morte, suas idéias exercem um crescimento no interesse pela mente (ciência cognitiva), cérebro (neurociência), e em como implementar o comportamento da mente no cérebro, Klein *et al.* (1999).

Segundo Hebb (1949), o problema de compreender o comportamento é um problema de entender a ação total do sistema nervoso, e vice-versa. Seu esforço interdisciplinar em defender este problema psicológico tratava-se de sua principal abordagem, Klein *et al.* (1999).

Em sua perspectiva de explicar a organização do comportamento humano e animal sob uma abordagem do principal dispositivo biológico envolvido, o cérebro, Hebb apresentou uma única teoria disposta em três postulados.

(a) A conexão existente entre os neurônios, conhecida como sinapse, é favorecida no mesmo grau das atividades decorrentes do comportamento pré e pós-sináptico. De acordo com Haykin (2001), a definição da *sinapse de Hebb* é aquela que usa um mecanismo dependente do tempo, altamente local e fortemente interativo para aumentar a eficiência sináptica como uma função da correlação entre as atividades pré-sináptica e pós-sináptica. Em sua publicação, Hebb introduziu esse conceito de acordo com a citação abaixo:

*Quando um axônio da célula A está perto o suficiente para excitar uma célula B e participa do seu disparo repetida ou persistentemente, então algum processo de crescimento ou modificação metabólica acontece em uma das células ou em ambas, de tal forma que a eficiência de A como uma das células que dispara para B é aumentada.*

(b) Grupos de neurônios com tendência para disparar ao mesmo tempo de um conjunto de células em atividade pode persistir após provocar o evento e servir para representá-lo, Klein *et al.* (1999). Alguns autores consideram que este

postulado é de grande importância conceitual.

(c) Um processo do pensamento, ou *fase-da-seqüência*, é composta por uma série de eventos que, de acordo com Hebb (1949), com alguma repetição freqüente, de uma simulação particular, irá conduzir ao desenvolvimento lento de um *conjunto de células* (cell-assembly) – uma estrutura de células compreendidas no córtex – e outras regiões específicas, Klein *et al.* (1999).

Em síntese, a regra de Hebb, diante do postulado (a), oferece um conceito de aumento da eficiência sináptica durante a interação de dois neurônios. No esquema de aprendizado de Hebb, a conexão entre dois neurônios é reforçada se ambos dispararem ao mesmo tempo, Arbib *et al.* (2003). De acordo com Fausett (1994), Hebb propôs que o aprendizado ocorre, pela modificação da força sináptica (peso) de uma maneira em que se dois neurônios interconectados forem ativados ao mesmo tempo, sua conexão seria acrescida. A mais simples formulação acontece conforme apresentado na equação (3.3).

$$\Delta w_{kj}(n) = \eta y_k(n) x_j(n) \quad (3.3)$$

Diante desta formulação,  $\Delta w_{kj}$  representa a modificação do peso sináptico. O estímulo  $x_j$  assume um valor de entrada para o neurônio  $k$  (sinal pré-sináptico) e  $y_k$  representa a saída do neurônio (sinal pós-sináptico). Nesta representação, conforme sugerido por Haykin (2001), a correção sináptica é influenciada pela *taxa de aprendizagem*  $\eta$ . Assim como no modelo McCulloch-Pitts, o neurônio de Hebb trabalha em iterações discretas representadas por  $n$ .

Em um modelo mais antigo, de performance original sugerida por Hebb, a taxa de aprendizagem  $\eta$  não está presente. O algoritmo de aprendizagem *Hebbiana* está apresentado na figura (3.2). Observe a inserção do *bias*, elemento com comportamento análogo ao peso sináptico, porém com a entrada fixa em 1. As entradas do tipo *bias* não são explicitamente usadas na formulação original do aprendizado de Hebb, Fausett *et al.* (1994). Todavia, aborda-se o elemento *bias* nesta formulação devido ao posterior treinamento bipolar apresentado.

**Passo 1.**

Inicializar os pesos sinápticos e bias:

$$w_{kj}=0, \forall j,j=1 \dots m \text{ e } k=1$$

$$b_k=0, k=1$$

**Passo 2.**

Para cada vetor de treinamento  $s$  e valor alvo  $t$ , executar os passos 3, 4 e 5, expostos abaixo.

**Passo 3.**

Definir as entradas:

$$x_j=s_j, \forall j,j=1 \dots m$$

**Passo 4.**

Definir o valor da unidade de saída:

$$y_k=t_k, k=1$$

**Passo 5.**

Ajuste dos pesos e bias (aprendizado):

$$\Delta w_{kj}(n)=x_j(n)y_k(n), \forall j,j=1 \dots m \text{ e } k=1$$

$$w_{kj}(n+1)=w_{kj}(n)+\Delta w_{kj}(n), \forall j,j=1 \dots m \text{ e } k=1$$

$$\Delta b_k(n)=y_k(n), k=1$$

$$b_k(n+1)=b_k(n)+\Delta b_k(n), k=1$$

Figura 3.2 – Algoritmo de Hebb

Observe que, nessa formulação (sem a presença de  $\eta$ ), o comportamento binário dos estímulos e respostas do neurônio limitam seu funcionamento. A fórmula não é capaz de distinguir quando os dados de treinamento são *ligados* (sob a notação binária, 1) e o valor alvo *desligado* (0), assim como, quando as entradas e alvo são *desligados*, Fausett *et al.* (1994). Uma alternativa ao tratamento binário é a representação bipolar.

A Tabela 3.2 representa o comportamento das lógicas binárias apresentadas na Tabela 3.1, sob o contexto bipolar, sugerido por Fausett (1994). Durante a

resolução da lógica *AND*, o desenvolvimento do algoritmo de Hebb, percorre quatro iterações, onde seus elementos assumem valores de acordo com a demonstração da Figura 3.3.

$x_1$	$x_2$	AND	OR	XOR
-1	-1	-1	-1	-1
-1	1	-1	1	1
1	-1	-1	1	1
1	1	1	1	-1

Tabela 3.2 – Estruturas Lógicas AND, OR e XOR (Bipolar)

#### ***Iteração 0.***

$$\begin{aligned} w_{11} &= 0, w_{12} = 0, b_1 = 0 \\ x_1 &= -1, x_2 = -1, y_1 = -1 \\ \Delta w_{11} &= 1, \Delta w_{12} = 1, \Delta b_1 = -1 \end{aligned}$$

#### ***Iteração 1.***

$$\begin{aligned} w_{11} &= 1, w_{12} = 1, b_1 = -1 \\ x_1 &= -1, x_2 = 1, y_1 = -1 \\ \Delta w_{11} &= 1, \Delta w_{12} = -1, \Delta b_1 = -1 \end{aligned}$$

#### ***Iteração 2.***

$$\begin{aligned} w_{11} &= 2, w_{12} = 0, b_1 = -2 \\ x_1 &= 1, x_2 = -1, y_1 = -1 \\ \Delta w_{11} &= -1, \Delta w_{12} = 1, \Delta b_1 = -1 \end{aligned}$$

#### ***Iteração 3.***

$$\begin{aligned} w_{11} &= 1, w_{12} = 1, b_1 = -3 \\ x_1 &= 1, x_2 = 1, y_1 = 1 \\ \Delta w_{11} &= 1, \Delta w_{12} = 1, \Delta b_1 = 1 \end{aligned}$$

#### ***Resultado***

$$w_{11} = 2, w_{12} = 2, b_1 = -2$$

Figura 3.3 – Algoritmo de Hebb resolvendo a lógica AND

Considerando o comportamento da equação (3.4), pode-se encontrar a equação da reta do resultado apresentado na Figura 3.3, conforme a equação (3.5). Observe que a lógica *AND* é linearmente separável, e por isso, com possível

resolução pelo algoritmo de Hebb.

$$b_1 + w_{11}x_1 + w_{12}x_2 = 0 \quad (3.4)$$

$$x_2 = -x_1 + 1 \quad (3.5)$$

No gráfico da Figura 3.6, de abscissa  $x_1$  e ordenada  $x_2$ , onde os elementos *desligados* (-1) e *ligados* (1) estão, respectivamente representados pelos símbolos - e +, a reta expressa na equação (3.5) resulta na separação linear dos grupos distintos.

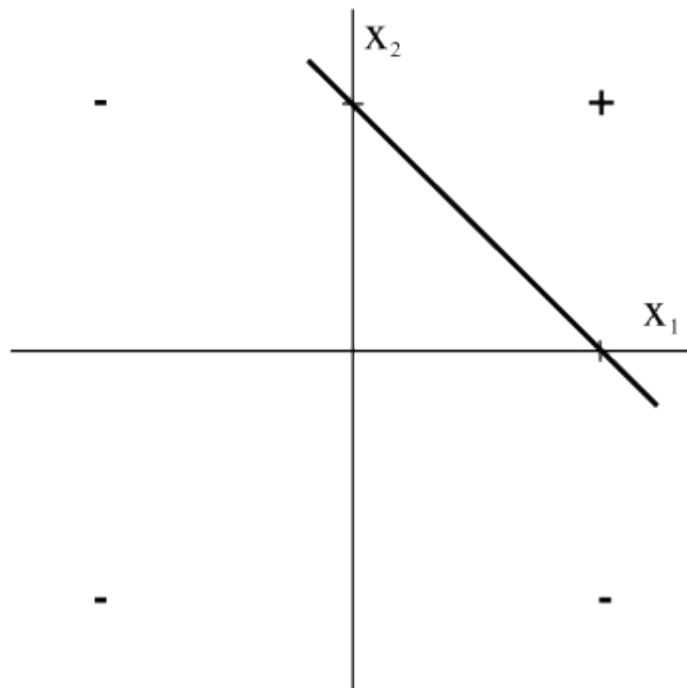


Figura 3.4 – Separação linear da lógica AND

O problema do neurônio original proposto por Hebb, conforme descrito por Arbib (2003) e Haykin (2001) é que, com a aplicação repetida do sinal de entrada, as sinapses irão continuamente e exponencialmente ficar mais fortes, chegando ao ponto da saturação. Neste momento, nenhuma informação será armazenada e a capacidade de seleção estará perdida.

Existem contribuições científicas que oferecem alterações ao modelo original de Hebb. Uma solução apresentada por Malsburg (1973) normaliza as sinapses,

conforme expresso na equação (3.6).

$$\frac{w_{kj} + \Delta w_{kj}}{\sum_{i=1}^m (w_{ki} + \Delta w_{ki})} \quad (3.6)$$

Na equação (3.6),  $i$  percorrem todas as  $m$  entradas do neurônio. Esta nova regra não somente aumenta a força das sinapses com as entradas correlacionadas da atividade celular, como pode diminuir a força sináptica das conexões que não cresceram, Arbib *et al.* (2003).

Outra alternativa ao problema pertinente à hipótese de Hebb refere-se aos conceitos introduzidos por Sejnowski (1977). Nesta hipótese, também conhecida como *hipótese da covariância*, sinais pré-sinápticos e pós-sinápticos são substituídos pelos desvios dos sinais pré-sinápticos e pós-sinápticos em relação aos seus respectivos valores médios em certo intervalo de tempo, Haykin *et al.* (2001). A fórmula (3.7) representa a modificação de Sejnowski, onde  $\bar{x}$  representa o valor médio no tempo do sinal pré-sináptico  $x_j$  e, de forma análoga,  $\bar{y}$  representa a média no tempo do sinal pós-sináptico  $y_k$ . Observe também a adoção da taxa de aprendizado.

$$\Delta w_{kj}(n) = \eta (x_j - \bar{x})(y_k - \bar{y}) \quad (3.7)$$

Independentemente das alterações sugeridas por Malsburg (1973) e Sejnowski (1977), há uma forte evidência fisiológica para a aprendizagem hebbiana na área do cérebro chamada *hipocampo*. O hipocampo desempenha um papel importante em certos aspectos de aprendizagem e memória. Esta evidência fisiológica torna a aprendizagem hebbiana bastante atrativa, Haykin *et al.* (2001).

Diante deste postulado, os conceitos introduzidos por Hebb adquiriram espaço científico na neurociência e outras ciências correlatas. Vários artigos apresentaram formulações práticas para a aprendizagem Hebbiana, sendo que, com devidas modificações. De forma análoga à Malsburg (1973) e Sejnowski (1977), outros autores abordaram a necessidade de se aplicar a característica



inibitória no neurônio de Hebb, evitando a saturação das sinapses e mal funcionamento da rede.

Em um trabalho de investigação, proposto por Amari, em 1971, que pretende entender mais aspectos da informação processada no sistema nervoso através dos muitos elementos lineares aleatórios que compõem as redes neurais, mostra que a informação é carregada pelo nível de atividade de uma rede com as taxas designadas para a *excitação* dos elementos. Por um longo período, uma oscilação estável existente é mostrada em um sistema composto por dois tipos de elementos, isto é, elementos *excitatórios* e *inibitórios*, que são conectados aleatoriamente, Amari *et al.* (1971).

Um dos primeiros trabalhos publicados que apresentaram a necessidade da característica inibitória nos neurônios, denominado *Tests on a cell assembly theory of the action of the brain, using a large digital computer*, foi proposto por Rochester, Holland, Haibt e Duda (1956). O artigo envolve a simulação computacional da teoria dos conjuntos celulares propostos por Hebb (1949) e mostra a clara necessidade que se deve adicionar a inibição para que a teoria realmente funcione, Haykin *et al.* (2001).

De acordo com a abordagem de Timoszczuk (2004), vários pesquisadores trabalharam sobre a idéia de Hebb, dentre eles Ashby (1952), Minsky (1954) e Uttley (1956).

Segundo a literatura de Mohan, Mehrotra e Ranka (1997), Uttley demonstrou que redes neurais com conexões modificáveis aprendem a classificar padrões com pesos sinápticos representando probabilidades condicionais. Ele desenvolveu um separador linear em que os pesos foram ajustados usando a *teoria da informação* de Shannon (1948) – que por sua vez, sob o escopo de um espaço ordenado, a teoria da informação é apropriada para a medição de incertezas. Esta é uma hipótese em que a eficiência de uma sinapse variável do sistema nervoso depende da relação estatística entre os estados flutuantes em ambos os lados daquela sinapse, Haykin *et al.* (2001).

Em um artigo clássico publicado em 1948, Claude Shannon estabeleceu os

fundamentos da teoria da informação. O trabalho original de Shannon sobre a teoria da informação, e seu refinamento por outros autores, foi uma resposta direta às necessidades de engenheiros eletricitas para projetar sistemas de comunicação que sejam tanto eficientes como confiáveis, Haykin *et al.* (2001).

De acordo com a abordagem de Intrator (2003), Shannon considerou a informação como a perda da incerteza e a definiu como uma função de distribuição de probabilidade.

A contribuição inserida pelo psiquiatra William Ross Ashby, através do artigo *Design for a Brain: The Origin of Adaptive Behavior*, que no mesmo ano (1948) também foi publicado como livro, caracterizava o comportamento humano de uma forma conceitual e com termos matemáticos. De acordo com seu próprio trabalho, Ashby havia tentado deduzir que algumas propriedades eram necessárias para que o sistema nervoso se comportasse mecanicamente e adaptativamente, Ashby *et al.* (1948).

Ashby estava certo que muito do comportamento neural humano poderia ser esclarecido mecanicamente. Em seu trabalho ele propôs um meio de imitar a habilidade do cérebro para produzir comportamento adaptativo, ou, em outras palavras, o conceito de aprendizagem, Maccorduck *et al.* (2004). Segundo Haykin, (2001), o trabalho trata da noção básica de que o comportamento adaptativo não é inato mas sim aprendido, e que através da aprendizagem o comportamento de um animal (sistema) normalmente muda para melhor. O livro enfatizava os aspectos dinâmicos do organismo vivo como uma máquina e o conceito correlacionado de estabilidade.

Muitos outros autores trabalharam com propostas científicas neste sentido, mas geralmente abandonavam a questão quando alguns postulados não se enquadravam com perfeição no âmbito da pesquisa, Ashby *et al.* (1948).

Em 1954, Marvin Minsky escreveu uma tese de doutorado em redes neurais na University of Pinceton, intitulada *Theory of Neural-Analog Reinforcement Systems and Its Application to the Brain-Model Problem*, Haykin *et al.* (2001). O trabalho fixou-se na investigação de um tipo de treinamento de reforço.

Em sua tese, Minsky descreveu a *Calculadora Neural-Análoga Estocástica do Reforço*, conhecida como SNARC (Stochastic Neural-Analog Reinforcement Calculator). As unidades de seu sistema são canais simples com pesos únicos que determinam a probabilidade de passar um pulso através do canal. A probabilidade é aumentada se a passagem de um pulso for seguida por uma recompensa. Minsky demonstrou seu sistema em uma tarefa de aprendizagem, Anderson *et al.* (1986). Segundo Randlov (2001), a idéia de um sinal interno de reforço secundário vem da psicologia de aprendizagem animal, entre outras coisas. Um reforço secundário é um estímulo que seria emparelhado com um reforço preliminar que conseqüentemente irá aumentar o reforço de propriedades similares. Minsky foi o primeiro a sugerir que este princípio poderia ser usado na inteligência artificial.

Em um trabalho posterior, Minsky retratou os conceitos de rede neurais sob a concepção de inteligência artificial. Esse estudo foi publicado no artigo *Steps Toward Artificial Intelligence* no ano de 1961. Segundo sua própria definição em Minsky (1961), os problemas de programação heurística – problemas com resolução computacional realmente difícil – são divididos em cinco principais áreas: busca, reconhecimento de padrões, aprendizado, planejamento e indução. Neste trabalho ele realiza um minucioso estudo em cada uma dessas áreas com abordagem direcionada à RNA.

Deve-se considerar que, em Minsky (1961), a terminologia *atribuição de crédito* foi introduzida. O problema da atribuição de crédito está relacionado com a definição de valores aos eventos pertinentes a um sistema, sendo que, cada evento envolve um número extensivo de possibilidades. Ou seja, para cada decisão a ser tomada, uma excitação ou inibição, respectivamente para acerto ou erro, deveria ser associada aos elementos de ponderação para que o próximo comportamento ganhasse benefício. Porém, num ambiente onde uma única tarefa envolve uma grande quantia de decisões, a atribuição do crédito seria impossibilitada. Observe a citação elaborada por Minsky (1961), a seguir:

*Jogando um jogo complexo como o xadrez, ou escrevendo um programa de computador, temos um critério definitivo de sucesso – no jogo, por exemplo, se ganha ou perde. Mas no decorrer do jogo, cada sucesso*

*final (ou falha) é associado com um grande número de decisões internas. Se o funcionamento for bem sucedido, como nós podemos atribuir crédito para o sucesso entre a grande quantidade de decisões?*

Todo este problema é diretamente associado à utilização do modelo sugerido por Rosenblatt (1958), denominado *Perceptron*, o qual será citado mais adiante.

No ano de 1967, foi publicado o livro de Minsky, *Computation: Finite and Infinite Machines*. Este livro, escrito de forma clara, estendeu os resultados de 1943 de McCulloch e Pitts e os colocou no contexto da teoria dos autômatos e da teoria da computação, Haykin *et al.* (2001). O estudo desenvolvido nesta publicação de Minsky gira em torno do estabelecimento da equivalência entre redes neurais com conexões cíclicas e uma classe abstrata de dispositivos computacionais chamados *finite-state machines* ou *finite automata*, Forcada *et al.* (2002).

Um modelo de comportamento representado por um número finito de estados, transações entre os estados, e ações, é conhecido como *finite-state machine* (FSM). De acordo com a definição de Lawson (2004), o termo *finite automata* descreve uma classe de modelos computacionais que são caracterizados por terem um número finito de estados. A característica de armazenamento da FSM é denominado estado. Através dos estados podem-se estudar as mudanças ocasionadas no sistema e compará-las com o comportamento atual. Uma transação é entendida como uma mudança de estado na qual estabelece regras que, quando cumpridas, permitem a transição. Por fim, para descrever as atividades, a *finite automata* utiliza-se da ação.

Por questões de conveniência didática, um importante trabalho realizado por Minsky (juntamente com Papert), datado em 1969, que apresenta circunstâncias limitadoras à metodologia proposta por Rosenblatt (1958), será descrito posteriormente ao levantamento teórico de Rosenblatt.

Uma importante etapa associada às RNA é constatada na construção de uma máquina de filtros não-lineares, elaborada por Dennis Gabor, em conjunto com outros autores. Segundo a definição de Boker (1996), as técnicas similares

àquelas usadas na predição não-linear provaram ser notavelmente eficazes em identificar e em remover o erro aditivo em simulações de sistemas dinâmicos não-lineares. Este processo tem raízes nas técnicas de filtros não-lineares que foram propostos por Gabor, Wilby, e Woodcock.

O projeto da máquina dos filtros lineares baseou-se nos conceitos do *filtro adaptativo não-linear* introduzidos pelo próprio Gabor alguns anos antes (mais especificamente, em 1954). Ele construiu essa máquina com a ajuda dos colaboradores Wilby e Woodcock. A aprendizagem era realizada alimentando-se a máquina com amostras de um processo estocástico, juntamente com a função-alvo que a máquina deveria produzir, Haykin *et al* (2001).

A perspectiva de resolução dos problemas por filtros lineares fica restrita a problemas mais simples. As equações derivadas das teorias econômicas, por exemplo, geralmente possuem comportamento não-linear. Não se pode derivar a expressão explícita para o algoritmo de filtragem, exceto para os casos em que a distribuição é normal e, a mensuração e equação transitória, sejam lineares, Tanizak *et al*. (1996).

Em 1956, de acordo com Cordeschi (2002) e Lingireddy e Brion (2005), Taylor desenvolveu as redes de memória associativa com base na teoria de aprendizagem de Hebb. Grande parte das pesquisas sobre sistemas paralelos foi introduzida por Taylor e seu sucessor Steinbruch, de publicações datadas respectivamente nos anos de 1956 e 1961, através do desenvolvimento de memórias distribuídas capazes de aprender por associação entre os padrões de entrada, ou através do aprendizado fornecido pela associação entre os padrões de entrada e respostas categorizadas. Muitos modelos recentes associaram-se a este tipo de problema, Dawson *et al*. (1998). Segundo Haykin (2001), o trabalho de Steinbruch, com base em Taylor, introduziu a *matriz de aprendizagem*; esta matriz consiste de uma rede planar de chaves interpostas entre arranjos de receptores *sensoriais* e atuadores *motores*.

Na essência, conforme a definição anterior de Dawson (1998), a memória associativa está relacionada a dois tipos de aprendizado. Pode-se caracterizá-los como *auto-associação* e *hetero-associação*. Na auto-associação, o processo

aquisição de conhecimento da rede enquadra-se no aprendizado não-supervisionado. Um conjunto de vetores de entrada, respectivo aos padrões, são apresentados à rede, aspirando a formação da memória. Por fim, com o treinamento finalizado, deve-se mostrar à rede um padrão distorcido ou incompleto, no qual ela será capaz de resgatar o formato original. Sob o aspecto hetero-associativo, a rede apresenta um aprendizado supervisionado, onde um conjunto de padrões de entrada está associado a um conjunto de padrões de saída.

Uma memória associativa é uma memória inspirada no cérebro, que aprende por *associação*. Desde Aristóteles, se sabe que a associação é uma característica proeminente da memória humana, e todos os modelos de cognição utilizam associação de uma forma ou de outra como a operação básica, Haykin *et al* (2001). Outros trabalhos realizados posteriormente basearam-se neste princípio, tendo como metodologia principal o aprendizado do *produto externo*. Antes da abordagem do produtor externo, alguns conceitos devem ser explanados.

Muitas variáveis compõem o comportamento do processo de aprendizagem. Entre elas pode-se destacar a *memória*. Esta pode ser entendida como um conjunto de alterações sinápticas decorrentes da interação de um sistema (ser vivo) com seu ambiente. Através da memória o sistema deve ser capaz de ter um comportamento ponderado pelo conhecimento associado à memória, ou seja, ter o comportamento influenciado pelas experiências do passado. Isso justifica de forma explícita a necessidade de recorrer ao conhecimento adquirido. Portanto, pode-se afirmar que o acesso à memória deve ser possível para que o sistema possa comportar-se de forma diferenciada.

Segundo a definição de Gupta, Homma e Jin (2003), pode-se distinguir a memória em dois tipos: Memória de curto prazo (STM, da notação *short-term memory*) e Memória de longo-prazo (LTM, *long-term memory*). Sob o aspecto biológico, pode-se definir:

*Memória de curto prazo*: Toda a informação na memória sensorial não é perdida até aproximadamente metade de um segundo. Esse processo é preservado por um escopo maior de tempo pela retransmissão inata do cérebro.

*Memória de longo prazo:* Dita aquela memória que armazena informações permanentemente ou por um prazo maior de tempo. Geralmente é adquirida com a aquisição contínua da memória de curto prazo. Na biociência é caracterizada como aprendizado.

Do ponto de vista artificial, considera-se que a memória associativa, além de distribuída, trabalha com estímulos e respostas – fato que permitirá a associação. Os estímulos são conhecidos como *padrões-chave* e as respostas como *padrões memorizados*. A recuperação da memória está implicitamente ligada ao valor agregado ao estímulo, ou seja, ele determina o seu local de armazenamento e endereço para recuperação, Haykin *et al* (2001).

O neurônio é uma unidade limitada sob o contexto de armazenamento de informação – memória. Contudo, a característica distribuída da rede neural é capaz de, com um ruído estocástico presente, representar e recuperar a memória.

De forma simples, a memória é adquirida através do mapeamento entre um padrão-chave e um padrão armazenado, e concebido pelos vários neurônios distribuídos na rede. Considere que  $x_k$  é um padrão de atividade de entrada e que  $y_k$  é ma padrão de atividade de saída. Observados os respectivos padrões tem-se a formação da memória associativa, conforme apresentado na Figura 3.5.

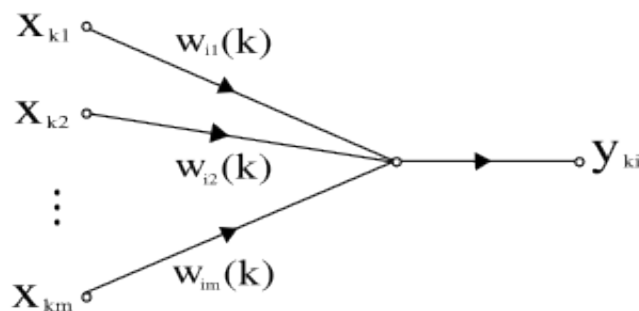


Figura 3.5 – Fluxo de grafo do neurônio  $i$ . (Haykin, 2001)

O comportamento de aprendizado associativo aqui tratado está relacionado com os vetores  $x_k$  e  $y_k$ , respectivamente apresentados na forma expandida através das equações (3.8) e (3.9). Observe que o padrão-chave  $x_k$  é representado por um

vetor de dimensionalidade  $m$ . De forma análoga, o padrão memorizado  $y_k$  possui dimensão  $m$ . Pode-se então considerar que a rede possui dimensão  $m$ .

$$x_k = [x_{k1}, x_{k2}, \dots, x_{km}]^T, \quad \forall k, k=1, 2, \dots, m \quad (3.8)$$

$$y_k = [y_{k1}, y_{k2}, \dots, y_{km}]^T, \quad \forall k, k=1, 2, \dots, m \quad (3.9)$$

Cada um dos elementos presentes nos padrões de entrada-saída assumem valores positivos ou negativos – característica extraída do contexto biológico, sendo que, um valor significativo para o disparo teria caráter matemático positivo, do contrário, seria negativo. Através da notação matricial, pode-se representar a relação existente entre  $x_k$  e  $y_k$  conforme a equação (3.10) – deve-se supor que a rede possui um comportamento linear.

$$y_k = W(k) x_k, \quad k=1, 2, \dots, m \quad (3.10)$$

Aqui,  $W(k)$  é a matriz de peso determinada pelo par de entrada-saída. Conforme apresentado na Figura 3.5, a saída  $y_{ki}$  do neurônio  $i$  devido à combinação dos elementos do padrão-chave  $x_k$  é expressa na equação (3.11), abaixo.

$$y_{ki} = \sum_{j=1}^m w_{ij}(k) x_{kj}, \quad i=1, 2, \dots, m \quad (3.11)$$

Observe que  $w_{ij}(k)$  onde  $j=1, 2, \dots, m$  representa os pesos sinápticos do neurônio  $i$  correspondente ao  $k$ -ésimo par de padrão associados. Utilizando a notação matricial para o comportamento de  $y_{ki}$  tem-se o resultado expresso na equação (3.12).

$$y_{ki} = [w_{i1}(k), w_{i2}(k), \dots, w_{im}(k)] \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{km} \end{bmatrix}, \quad i=1, 2, \dots, m \quad (3.12)$$

Considerando o comportamento de  $y_{ki}$  de índice  $m$ -por-1, pode-se reformular



a equação (3.12) em (3.13). Observe que a matriz de componentes  $W(k)$ , expressa na equação (3.10) é uma matriz *m-por-m*.

$$\begin{bmatrix} y_{k1} \\ y_{k2} \\ \vdots \\ y_{km} \end{bmatrix} = \begin{bmatrix} w_{11}(k) & w_{12}(k) & \cdots & w_{1m}(k) \\ w_{21}(k) & w_{22}(k) & \cdots & w_{2m}(k) \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1}(k) & w_{m2}(k) & \cdots & w_{mm}(k) \end{bmatrix} \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{km} \end{bmatrix} \quad (3.13)$$

As apresentações individuais dos  $q$  pares de padrões associados  $x_k \rightarrow y_k, k=1,2,\dots,q$ , produzem valores correspondentes da matriz individual, ou seja,  $W(1), W(2),\dots, W(q)$ . Reconhecendo que esta associação de padrões é representada pela matriz de pesos  $W(k)$ , pode-se definir uma *matriz de memória m-por-m* que descreve a soma das matrizes de pesos para o conjunto inteiro de associações de padrões, Haykin *et al* (2001), conforme (3.14).

$$M = \sum_{k=1}^q W(k) \quad (3.14)$$

Encontra-se portanto a matriz  $M$  que representa a *experiência total* assumida pela memória como resultado das apresentações de  $q$  padrões de entrada-saída. A estruturação de  $M$  ocorre pelo somatório de todas representações  $W(k)$ . De início, na associação do padrão de entrada-saída 1,  $M_0$  é zero – devido ao valor inicial dos pesos sinápticos, que é nulo. Ao final de  $q$  associações,  $M_q$  é equivalente à  $M$  apresentada na equação (3.14). Entendido esse princípio, pode-se representar a matriz de memória, de forma recursiva, pela equação (3.15), expressa abaixo.

$$M_k = M_{k-1} + W(k), \quad k=1,2,\dots,q \quad (3.15)$$

Note que, durante as associações,  $W(k)$  perde sua identidade quando é adicionado à  $M_{k-1}$ . Isso significa que na representação completa da memória  $M$ , o comportamento de uma associação de entrada-saída denotada por  $W(k)$  torna-se um elemento do todo, e conseqüentemente não representa mais uma associação individual. Observe também que, quanto maior o valor de  $q$ , menos significativo é a importância de  $W(k)$ .

Uma aproximação do comportamento de  $M$  é dado pela equação (3.16), onde  $y_k x_k^T$  representa o *produto externo* entre o padrão-chave  $x_k$  e o padrão memorizado  $y_k$ , Anderson *et al.* (2005). O produto externo é uma estimativa da matriz de pesos  $W(k)$  que mapeia o padrão de saída  $y_k$  para o padrão de entrada  $x_k$ , Haykin *et al.* (2001). Observe que  $\hat{M}$  apresenta a mesma dimensionalidade de  $M$ .

$$\hat{M} = \sum_{k=1}^q y_k x_k^T \quad (3.16)$$

Um processo de aprendizagem *local* inserido no contexto de (3.16) pode ser considerado como uma generalização do aprendizado proposto por Hebb (1949).

Observe que  $y_{ki} x_{kj}$  representa um termo do produto externo  $y_k x_k^T$ , onde  $x_{kj}$  é o elemento de origem  $j$  da camada de entrada – ou sinal pré-sináptico do peso  $w_{ij}(k)$  – e  $y_{ki}$  é o elemento de saída do neurônio  $i$  – ou sinal pós-sináptico do peso  $w_{ij}(k)$  na  $k$ -ésima associação. Essa regra também é denominada como *regra do produto externo* devido à operação matricial usada para construir a matriz de memória  $\hat{M}$ . Correspondentemente, uma regra de aprendizado assim construída é chamada de *memória por matriz de correlação*, Haykin *et al* (2001). De forma análoga à representação exposta na equação (3.15), a regra do produto externo também pode ser representada pela forma recursiva, de acordo com (3.17).

$$\hat{M}_k = \hat{M}_{k-1} + y_k x_k^T, \quad k=1,2,\dots,q \quad (3.17)$$

Conforme discutido, uma característica de grande importância pertinente à memória é a *recordação*. Suponha que o produto externo  $\hat{M}$  foi exposta ao conjunto completo de padrões-chave associados aos padrões memorizados. Destaca-se do conjunto de entrada um padrão-chave qualquer, denominado  $x_j$ , no qual espera-se encontrar o valor correspondente  $y_j$  através do resultado  $y$  expresso na equação (3.18), formalizada a seguir.

$$y = \hat{M} x_j \quad (3.18)$$

De acordo com o comportamento da equação (3.16),  $\hat{M}$  poderá ser substituído e conseqüentemente a equação (3.19) será formada. Reestruturando a equação (3.19), pode-se encontrar o comportamento sugerido na equação (3.20).

$$y = \sum_{k=1}^m y_k x_k^T x_j \quad (3.19)$$

$$y = \sum_{k=1}^m (x_k^T x_j) y_k \quad (3.20)$$

Observe que  $x_k^T$  é um vetor de dimensão *1-por-m* e que  $x_j$ , um vetor pertencente ao conjunto de entradas, é *m-por-1*. Isso significa que o produto interno  $x_k^T x_j$  é um escalar. Ao destacar o comportamento sugerido pelo índice *j*, encontra-se a equação (3.21) de comportamento análogo à (3.20).

$$y = (x_j^T x_j) y_j + \sum_{\substack{k=1 \\ k \neq j}}^m (x_k^T x_j) y_k \quad (3.21)$$

De acordo com Haykin (2001), normalizando cada um dos padrões-chave  $x_1, x_2, \dots, x_q$  encontra-se a energia unitária conforme apresentado na equação (3.22). Este conceito será usado adiante.

$$E_k = \sum_{l=1}^m x_{kl}^2$$

$$E_k = x_k^T x_k \quad (3.22)$$

$$E_k = 1, \quad k = 1, 2, \dots, q$$

Pode-se considerar a equação (3.21) de forma simplificada, onde se iguala o segundo elemento do somatório à  $v_j$ . Esta formulação é expressa na equação (3.23). Na equação (3.24) é representado o valor atribuído à  $v_j$ .

$$y = y_j + v_j \quad (3.23)$$

$$v_j = \sum_{\substack{k=1 \\ k \neq j}}^m (x_k^T x_j) y_k \quad (3.24)$$

Observe em (3.23) que o elemento  $y_j$  representa a resposta *desejada* e compõe o valor da resposta real  $y$ . O segundo termo  $v_j$ , resultado da interferência cruzada entre o valor  $x_j$  e todos os padrões-chave da memória, representa um *vetor de ruído*. Observe ainda que, se o vetor de ruído apresentar um valor nulo, a resposta desejada se iguala à resposta obtida. Isso acontece nos casos formalizados adiante.

De acordo com a definição de Vidyamurthy (2004), o co-seno de um ângulo entre um par de vetores (neste caso,  $x_j$  e  $x_k$ ) é definido como o produto interno entre os vetores dividido pelo produto de suas normas euclidianas, conforme mostrado na equação (3.25).

$$\cos(x_k, x_j) = \frac{x_k^T x_j}{\|x_k\| \|x_j\|} \quad (3.25)$$

Sabe-se que a norma euclidiana de  $x_k$  (assim como  $x_j$ ) é a raiz quadrada da energia do vetor  $x_k$ . Estando de acordo com as regras definidas na equação (3.22), o comportamento de  $\|x_k\|$  é equivalente à  $(x_k^T x_k)^{1/2}$ , e conseqüentemente igual à  $E_k^{1/2}$ . Pode-se, portanto redefinir o comportamento da equação (3.25) através da equação (3.26).

$$\cos(x_k, x_j) = x_k^T x_j \quad (3.26)$$

Observe que, através da formulação da equação (3.16), o comportamento da equação (3.24) pode ser redefinido, conforme mostrado na fórmula (3.27). Por conseqüência, a equação (3.23) também será afetada.

$$v_j = \sum_{\substack{k=1 \\ k \neq j}}^m \cos(x_k, x_j) y_k \quad (3.27)$$

Através desta formulação, fica explícito que o valor esperado  $y_j$  iguala-se ao valor real  $y$  quando os vetores  $x_k$  e  $x_j$  formarem um ângulo reto (perpendiculares) entre si no sentido euclidiano, sendo que  $k$  deve ser diferente de  $j$ . Ou seja, quando os vetores padrões-chave forem *ortogonais*, a expressão abaixo, representada através da fórmula (3.28), se torna verdadeira. A condição existente para que o produto externo seja ortogonal é representado na equação (3.29).

$$\cos(x_k, x_j) = 0, \quad k \neq j \quad (3.28)$$

$$x_k^T x_j = \begin{cases} 1, & k=j \\ 0, & k \neq j \end{cases} \quad (3.29)$$

Certamente, existe um número de padrões que podem ser associados à matriz de correlação com confiança – no sentido de que possam ser acessados posteriormente sem a presença drástica do ruído presente, ou ainda, livre dele. Este fator é determinado pelo posto da matriz, Haykin *et al.* (2001). Assim, o número de padrões que podem ser armazenados de forma confiável é dado pelo posto da matriz de correlação  $\hat{M}$ . O posto da matriz é o número de linhas (ou colunas) que são linearmente independentes, Poole, Salerno *et al.* (2004). Observe abaixo a definição de Haykin (2001):

*O número de padrões que podem ser armazenados de forma confiável em uma memória por matriz de correlação nunca pode exceder a dimensionalidade do espaço de entrada.*

Essa definição baseia-se no estudo anterior, onde a matriz de correlação  $\hat{M}$  é de ordem  $m$ -por- $m$ , e conseqüentemente com um limite de associações inferior (ou igual) à  $m$  (sua dimensionalidade).

Todo avanço baseada na memória associativa proposta por Taylor (1956) iniciou-se seis anos mais tarde, em 1972, através dos autores Anderson, Kohonen e Nakano que, de maneira independente, introduziram a idéia da *memória por*

*matriz de correlação* adquirida pela regra de aprendizagem do *produto externo*, conforme apresentado nos postulados e equações anteriores, Haykin *et al.* (2001).

Outra grande etapa consistente na contextualização das redes neurais é firmada na introdução do *Perceptron*. Segundo Fausett (2004), a regra de aprendizado do perceptron é uma regra de aprendizado mais poderosa do que a regra sugerida por Hebb. O trabalho centra-se na demonstração de uma nova metodologia de aprendizado supervisionado introduzida por Rosenblatt (1956). Contudo, o coroamento do trabalho de Rosenblatt foi o chamado *teorema da convergência do perceptron*, cuja primeira demonstração foi delineada em um trabalho posterior do autor, mais precisamente, em 1960, Haykin *et al.* (2001). A concepção do aprendizado supervisionado, segundo Mohan, Mehrotra e Ranka (1997), está intimamente relacionada com o *perceptron*, e também foi designada nos trabalhos de Rosenblatt.

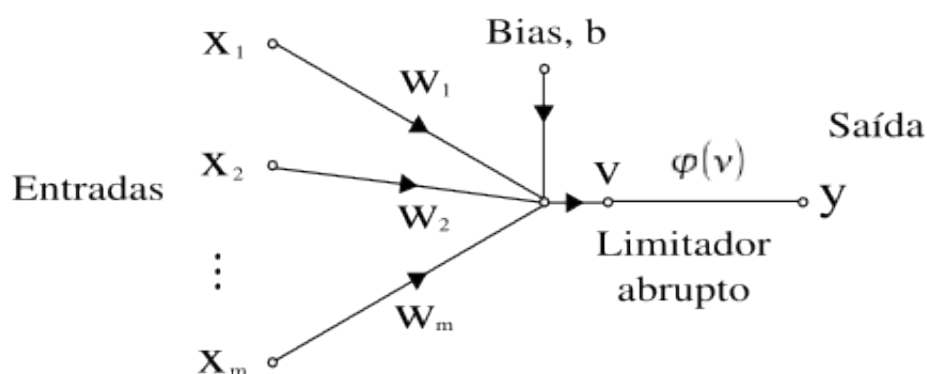


Figura 3.6 – Fluxo de grafo do *Perceptron*. Haykin (2001)

Rosenblatt define que um *perceptron* é uma máquina que aprende, usando exemplos, para atribuir vetores de entrada (*amostras*) à diferentes classes, usando uma função linear das entradas, Mohan, Mehrotra, Ranka *et al.* (1997). Perceptrons podem ser entendidos como uma rede neural que aprendem com a *experiência*, usando uma regra de correção de erro projetada para mudar os pesos de cada unidade quando produz resultados errôneos aos estímulos que são apresentados à rede, Arbib *et al.* (2003). Pode-se considerar os perceptrons como a forma artificial mais simples capaz de realizar a classificação de padrões linearmente separáveis,

ou seja, pode distinguir dois padrões (hipóteses) que se encontram em lados opostos de um *hiperplano*. O formato mais simples do perceptron – que atende à realidade acima – é composto por um único neurônio, formado por pesos *sinápticos* e *bias*. Esse modelo está representado, na forma de grafos de fluxo, na Figura 3.6.

As entradas  $x_1, x_2, \dots, x_m$  são associadas aos respectivos pesos sinápticos  $w_1, w_2, \dots, w_m$  onde é calculada uma combinação linear. Os bias apresentam uma entrada padrão 1 e pondera seu valor. Segundo Haykin (2001), os bias têm o efeito de aumentar ou diminuir a entrada líquida da função de ativação, dependendo se ele é positivo ou negativo, respectivamente. O resultado da combinação linear acrescida do bias é expresso na equação (3.30).

$$v = \sum_{i=1}^m w_i x_i + b \quad (3.30)$$

O resultado  $v$  da equação (3.30), ou *campo local induzido*, é aplicado ao *limitador abrupto* – função de ativação. Várias considerações são importantes para a compreensão do modelo proposto por Rosenblatt, tais como *teorema da convergência do perceptron*, algoritmo do *mínimo quadrado médio*, além de conteúdos referentes à *otimização irrestrita* – método de Newton, Gauss-Newton, entre outros. Contudo, por conveniência do desenvolvimento deste trabalho, esses conceitos serão abordados nos próximos capítulos de forma exclusiva.

Em 1960, Widrow e Hoff introduziram o *algoritmo do mínimo quadrado médio* (LMS – Least Mean-Square) e o usaram para formular o Adaline (adaptive linear element – elemento linear adaptativo). A diferença entre o perceptron e o Adaline está no procedimento de aprendizagem, Haykin *et al.* (2001). A Adaline é um dispositivo consistente de um elemento de processamento simples. O termo Adaline é um acrônimo; entretanto, seu significado modificou-se durante os anos. Inicialmente era chamado de *Adaptive Linear Neuron*, depois se tornou *Adaptive Linear Element*, conforme Freeman, Skapura *et al.* (1991).

Observe na Figura 3.7 o modelo essencial da Adaline. Pode-se entender que uma Adaline completa é constituída por um combinador linear adaptável, conforme

mostrado na linha tracejada da Figura 3.7, e uma função de saída bipolar.

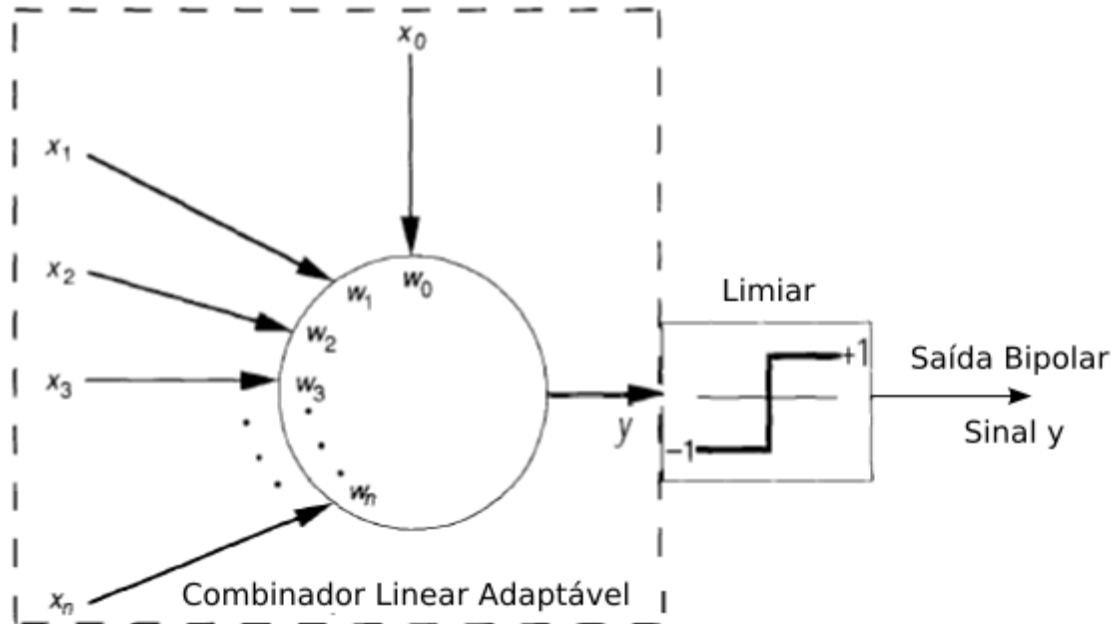


Figura 3.7 – Adaline. Freeman, Skapura (1991)

O modelo de aprendizagem LMS trabalha, conforme representado na Figura 3.7, com vetor de entrada, vetor de pesos e saída bipolar desejada (com instância entre 0 e 1). Seu desenvolvimento centra-se na busca por um vetor de pesos que pode, com sucesso, associar cada vetor de entrada com o valor desejado, Freeman, Skapura *et al.* (1991). Similar à regra de aprendizado do *perceptron*, o algoritmo do mínimo quadrado médio é um exemplo de treinamento supervisionado, em que a regra de aprendizagem é caracterizada pela definição de exemplos para o comportamento desejado da rede, Demuth, Beale, Hagan *et al.* (2007). Na definição da equação (3.31),  $x$  são as entradas da rede,  $d$  os alvos (valor desejado), caracterizados nas iterações  $1, 2, \dots, n$ .

$$\{x_1, d_1\}, \{x_2, d_2\}, \dots, \{x_n, d_n\} \quad (3.31)$$

Para cada entrada apresentada à rede, a saída da rede é comparada com o valor alvo. O erro é calculado com a diferença entre a saída desejada e a saída da rede, de acordo com Demuth, Beale, Hagan *et al.* (2007). *Minimizando* este erro, encontrar-se-á um vetor de pesos otimizado capaz de aproximar, com um erro



tolerável, os vetores de entrada aos respectivos alvos.

Todavia, os modelos mais simples da adaline possuem arquitetura de camada única. Um modelo mais complexo de rede neural composto por camadas treináveis com múltiplos elementos adaptativos, sugerido por Widrow e seus estudantes, em 1962, é denominado *multiple-adaline*, ou simplesmente *Madaline*, Haykin *et al.* (2001). *Madalines* são extensões multi-camadas das *Adalines*, Fausett *et al.* (1994). *Madaline* é um acrónimo para muitas *Adalines*. Organizada em uma arquitetura multi-camadas, a *Madaline* representa uma estrutura geral das redes neurais artificiais, Freeman, Skapura *et al.* (1991). Através dessa estrutura a *Madaline* é capaz de resolver o problema de lógica *XOR*, por exemplo. Nos próximos capítulos este assunto será novamente abordado.

Durante o período clássico do perceptron, nos anos 1960, parecia que as redes neurais poderiam realizar qualquer coisa, Haykin *et al.* (2001). Todavia, uma pesquisa realizada por Minsky e Papert, em 1969, que demonstrava as limitações dos perceptrons (isto é, redes de única camada) bloqueou um avanço mais intensivo das pesquisas durante alguns anos, Fausett *et al.* (1994). Em seu livro, *Marvin Minsky e Seymour Papert*, fizeram uma análise formal da capacidade do modelo neural perceptron. Eles descobriram que como um dispositivo lógico, o perceptron não poderia resolver um simples problema de lógica booleana *XOR*, Boothe *et al.* (2002), ou seja, eles forneceram uma simples prova geométrica que o modelo não pode resolver um problema simples que envolve a determinação em que uma imagem geométrica binária está conectada, Bishop *et al.* (1995).

Ainda neste mesmo trabalho, Minsky e Papert colocaram que, não havia motivo aparente para acreditar que redes formadas por múltiplas camadas de perceptrons, pudessem herdar as boas qualidades do perceptron simples. Estas colocações desestimularam sobremaneira os pesquisadores que atuavam na área, fazendo com que as pesquisas sobre RNAs sofressem uma quase que completa paralisação durante a década de 70, Timoszczuk *et al.* (2004). De fato, somente uns punhados dos pioneiros originais mantiveram seu comprometimento com as redes neurais. De uma perspectiva da engenharia, podem-se considerar os anos 70 como uma década de adormecimento para as redes neurais, Haykin *et al.* (2001).

Segundo Fausett (1994) e Timoszczuk (2004), nesta época, as contribuições favoráveis à ciência das redes neurais artificiais estão associadas aos seguintes autores: *Kohonen, Anderson, Grossberg, Carpenter e Malrsburg*.

*Von del Malrsburg* em seu trabalho *Self-organization of orientation-sensitive cells in the striate cortex*, publicado em Malrsburg (1973), foi o primeiro pesquisador a demonstrar o princípio da auto-organização. Seu trabalho ofereceu uma metodologia relacionada à formação das redes auto-organizáveis, motivada pelo mapeamento topológico do cérebro, Timoszczuk *et al.* (2004). *Stephen Grossberg* desenvolveu, em publicações nos anos de 1972 e 1976, trabalhos relacionados à aprendizagem competitiva, Haykin *et al.* (2001). De acordo com Fausett (1994), *Gail Carpenter* desenvolveu, juntamente com *Grossberg*, uma teoria de redes neurais auto-organizáveis que, em trabalhos futuros, propiciaram uma metodologia conhecida como *ressonância adaptativa (ART – adaptive resonance theory)*. Através do trabalho desenvolvido na *Helsinki University of Technology*, *Teuvo Kohonen*, em 1972, introduziu seus modelos de memórias associativas, Timoszczuk *et al.* (2004). Paralelamente ao trabalho de Kohonen, as contribuições de *James Anderson*, em seu trabalho pela *Brown University*, abordaram aspectos sobre as redes com memória associativa. Ele desenvolveu estas idéias em seu trabalho *Brain-State-in-a-Box*, consistindo de uma rede associativa simples acoplada a uma dinâmica não-linear, Fausett *et al.* (1994) e Haykin *et al.* (2001).

Alguns trabalhos de grande impacto para as RNA surgiram na década de 80. Isso caracterizou uma reaparição do interesse científico pertinente às redes neurais. Uma publicação de influência, que trata de uma classe particular de redes neurais com realimentação, foi introduzida por *Hopfield*. O tratamento de redes neurais como sistemas dinâmicos é uma característica comum, e já foi abordada por vários autores. Contudo, através de um artigo publicado em 1982, *John Hopfield* atraiu a atenção de muitos a este campo de estudo, Arbib *et al.* (2003). O princípio dos trabalhos de Hopfield encontra-se em publicações anteriores referentes às redes recorrentes com memória auto-associativa. Em seu estudo, Hopfield trabalha com diversas idéias já abordadas a respeito destas redes e apresenta uma análise matemática completa baseada em modelos da rotação de Ising (modelo matemático em mecanismos estatísticos). Este tipo de rede é

conhecida como *redes, Hopfield, Anderson et al. (1995).*

Como comportamento padrão, as redes Hopfield são capazes de interpretar entradas, associando-as a um dos dados de treinamento, e representá-las sem o fator *ruído* – quando presente. Ou seja, no *aprendizado* a rede é estimulada com padrões de entrada, pelos quais é capaz de adquirir seu conhecimento. Com a apresentação posterior de uma entrada ruidosa, a rede retornará um dos padrões de entrada mais coerentes. Segundo a definição de Arbib (2003), uma rede Hopfield é composta por unidades com pesos simétricos ( $w_{ij} = w_{ji}$ ) sem auto-conexões ( $w_{ii} = 0$ ) e com atualização assíncrona. Em outras palavras, ela consiste de  $N$  neurônios interconectados que atualiza seus valores de ativação de forma assíncrona e independente dos outros neurônios, Anderson *et al.* (1995). Por exemplo, suponha que  $s_i$  define o estado (0 ou 1) da  $i$ -ésima unidade. Para cada passo de tempo uma unidade é escolhida aleatoriamente. Se a unidade  $i$  é escolhida,  $s_i$  apresenta o valor 1 se e somente se  $\sum w_{ij}s_j \geq \theta_i$ . Caso contrário, é apresenta o valor 0, Arbib *et al.* (2003).

Segundo Anderson (1995), o estado do sistema é dado pelos valores da ativação  $y = (y_k)$ . A entrada da rede  $s_k(t+1)$  do neurônio  $k$  no ciclo  $t+1$  é constituído conforme apresentado na equação (3.32), a seguir.

$$s_k(t+1) = \sum_{j \neq k} y_j(t) w_{jk} + \theta_k \quad (3.32)$$

Uma simples função de limiar é aplicada à entrada da rede para obter o novo valor de ativação  $y_i(t+1)$  na iteração  $t+1$ . Essa função é descrita na equação (3.33).

$$y_k(t+1) = \begin{cases} +1, & \text{se } s_k(t+1) > U_k \\ -1, & \text{se } s_k(t+1) < U_k \\ y_k(t), & \text{caso contrário} \end{cases} \quad (3.33)$$

Para simplificar, pode-se definir  $U_k = 0$ , mas isso não é necessário. Um neurônio  $k$  da rede Hopfield é dito estável no momento  $t$ , se estiver de acordo com

as equações (3.32) e (3.33). Segundo a definição de Arbib (2003), Hopfield definiu uma medida chamada *energia* para sua rede, conforme apresentado na equação (3.34).

$$E = -1/2 \sum_{ij} s_i s_j w_{ij} + \sum_i s_i \theta_i \quad (3.34)$$

Este não é um comportamento físico (*energia física*) da rede neural, mas é um quantitativo matemático que, em muitos casos, realiza para a dinâmica neural o que a energia potencial faz para os mecanismos de Newton. Em geral, um sistema mecânico move seu estado de energia potencial para baixo, assim como, no contexto caracterizado pela Figura 3.8, uma esfera tende a mover-se para baixo.

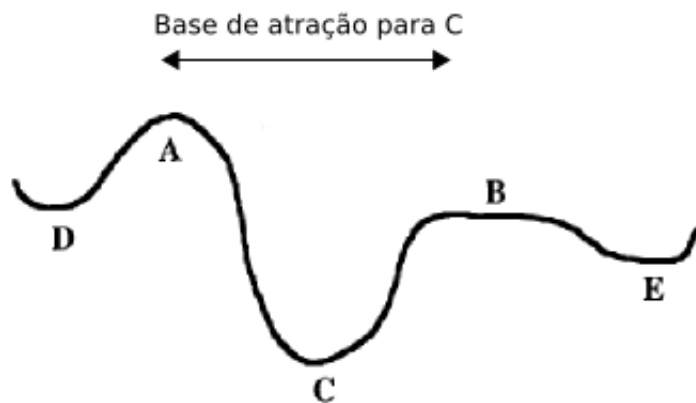


Figura 3.8 – Paisagem de Energia. Arbib (2003)

No caso da Figura 3.8, o ponto C é chamado de atrator e compreende todos os estados cuja dinâmica tende para C. Hopfield mostrou que suas redes simétricas com atualização assíncrona possuem uma propriedade familiar, Arbib *et al.* (2003).

Assim, o trabalho publicado por Hopfield, em 1982, desencadeou um grande interesse da física teórica pela modelagem neural, transformando com isso a área de redes neurais. Foi neste mesmo artigo que pela primeira vez o princípio do armazenamento da informação em redes dinamicamente estáveis foi explicado. Além disso, Hopfield mostrou que ele havia se baseado no modelo de vidro de *spins* da mecânica estatística para examinar o caso especial das redes recorrentes

com conexões simétricas, garantindo com isso, sua convergência para uma condição estável, Haykin *et al.* (2001).

Um trabalho baseado nas contribuições de Hopfield é, segundo Golden (1996), mostrado por Cohen e Grossberg. Em 1983, eles apresentaram um princípio geral para estimar a estabilidade de uma memória *endereçável por conteúdo*, Haykin *et al.* (2001). A rede de tempo contínuo de classe *Cohen-Grossberg* inclui a característica de *tempo-contínuo* da rede Hopfield como um caso especial, Golden *et al.* (1996). Uma característica distintiva de uma rede neural de atratores é o modo natural como o *tempo*, uma dimensão essencial para a aprendizagem, se manifesta na dinâmica não-linear da rede, Haykin *et al.* (2001).

O próximo trabalho de interesse científico foi publicado no ano de 1982, por *Teuvo Kohonen*. Este trabalho contém um estudo teórico e simulações computacionais de um novo processo de auto-organização. A principal descoberta trata de uma simples rede de elementos físicos adaptativos, receptores de sinais de um espaço de tempo primário, onde a representação dos sinais é automaticamente mapeada na definição da saída de tal maneira que, as respostas adquirem a mesma ordem topológica existente nos eventos preliminares. Em outras palavras, foi descoberto um princípio que facilitasse a formação automática de mapas topológicos observando as características dos eventos, de acordo com Kohonen *et al.* (1982).

O mapa auto-organizável proposto por Kohonen é uma rede engenhosa construída em torno de uma grade uni- ou bidimensional de neurônios para capturar as características importantes contidas em um espaço de entrada (dados) de interesse. Dessa forma, ele fornece uma representação estrutural dos dados de entrada pelos vetores de pesos dos neurônios como protótipos, conforme Haykin *et al.* (2001).

As características do aprendizado pertinente ao modelo de Kohonen apresentam peculiaridades interessantes. Os mapas auto-organizáveis apresentam um método não-linear em que as características podem ser obtidas através de um processo de *aprendizado não-supervisionado*, Arbib *et al.* (2003). Conhecido como *aprendizado competitivo*, o modelo de Kohonen estabelece uma competição entre

os neurônios. O elemento que apresentar os valores sinápticos mais próximos dos estímulos da rede, em um determinado instante, será definido como *neurônio vencedor*. Este vencedor terá seus pesos ajustados para um valor ainda mais próximo dos sinais encontrados no estímulo de entrada (de mesmo instante).

O processo competitivo entre os neurônios envolve a descoberta do neurônio vencedor. De acordo com o trabalho de Haykin (2001), considere que  $m$  represente a dimensão do espaço de entrada. Considere que um padrão (associado a um vetor) de entrada selecionado aleatoriamente do espaço de entrada seja representado pela equação (3.35).

$$x = [x_1, x_2, \dots, x_m]^T \quad (3.35)$$

O vetor de peso sináptico de cada neurônio da grade tem a mesma dimensão que o espaço de entrada. Considere que o vetor de peso sináptico do neurônio  $j$  seja representado pela equação (3.36).

$$w_j = [w_{j1}, w_{j2}, \dots, w_{jm}]^T, \quad j = 1, 2, \dots, l \quad (3.36)$$

O elemento  $l$  representa o número total de neurônios. Para encontrar o melhor casamento do vetor de entrada  $x$  com os vetores de pesos sinápticos  $w_j$ , com suposição de que o mesmo *limiar* seja aplicado a todos os neurônios, escolha o *produto interno*  $w_j^T x$  que apresente maior valor ou, de forma equivalente, minimize a distância euclidiana entre os vetores  $x$  e  $w_j$ , Haykin *et al.* (2001). A equação (3.37) é capaz de encontrar o índice  $i(x)$  do neurônio vencedor.

$$i(x) = \arg \min_j \|x - w_j\|, \quad j = 1, 2, \dots, l \quad (3.37)$$

De acordo com a definição de Demuth, Beale e Hagan (2007), os pesos do neurônio vencedor (uma fileira da matriz dos pesos de entrada) são ajustados com a regra de aprendizado de Kohonen. Supondo que o  $i$ -ésimo neurônio seja vencedor, os elementos da respectiva linha ( $i$ -ésima linha) da matriz dos pesos de entrada serão ajustados conforme a equação (3.38) representada abaixo.

$${}_i w(n) = {}_i w(n-1) + \alpha(x(n) - {}_i w(n-1)) \quad (3.38)$$

A taxa de aprendizado é representada por  $\alpha$ . O vetor de pesos correspondente ao neurônio vencedor  $i$  é denotado por  $w$ . O vetor de entrada é  $x$  e  $n$  caracteriza o período.

O modelo de Kohonen recebeu muito mais atenção em um contexto analítico e em relação às aplicações na literatura que o modelo de *Willshaw* e *von der Malsburg* (que era diferente em alguns aspectos), e tornou-se uma referência para a avaliação de outras inovações neste campo, Haykin *et al.* (2001). Os trabalhos introduzidos por Kohonen são aplicados às áreas de *reconhecimento da fala*, solução para o *problema do caixeiro viajante* e composição musical, Fausett *et al.* (1994).

Através dos trabalhos introduzidos por Kirkpatrick, Gelatt e Vecchi, referentes a um procedimento denominado *recozimento simulado*, foi concebido o desenvolvimento de uma máquina estocástica conhecida como *máquina de Boltzmann*, pelos autores Ackley, Hinton e Sejnowski, no ano de 1985, Haykin *et al.* (2001). Esses pesquisadores envolveram-se no desenvolvimento de uma rede neural não-determinística, isto é, uma rede que modifica seus pesos ou ativações com base em uma função de densidade de probabilidade, Fausett *et al.* (1994).

A máquina de Boltzmann é uma rede recorrente com unidades estocásticas de conexões simétricas para a qual o algoritmo de aprendizagem de Hebb é utilizado para reduzir a distância Kullback-Liebler (utilizada para calcular a distância entre duas distribuições de probabilidade) entre a distribuição da probabilidade das entradas e os valores gerados pela rede, Hinton, Sejnowski *et al.* (1999).

Apesar do algoritmo de aprendizagem da máquina de Boltzmann não ter se mostrado tão eficiente do ponto de vista computacional como o algoritmo de *retro-propagação* (apresentado adiante), ele superou o impasse psicológico, mostrando que a especulação de *Minsky* e *Papert*, em 1969, não estava corretamente embasada, Haykin *et al.* (2001).

Outro trabalho favorável a este meio foi introduzido por Braitenberg (1984) através da publicação de seu livro. *Valentino Braitenberg*, um pesquisador da neurociência, introduziu alguns conceitos que se associaram ao seu próprio nome. Os *veículos de Braitenberg* são apresentados de forma ideal, ilustrando alguns pontos teóricos fundamentais, tais como o problema *quadro de referência* (frame-of-reference – uma perspectiva particular para observação do ambiente). Eles também provem uma perspectiva interessante para os problemas de comportamento segmentado e seleção de ação, Pfeifer, Scheier *et al.* (1999). Em seu trabalho, Braitenberg defende o *princípio do desempenho auto-organizado, direcionado a objetivo*: obtém-se um melhor entendimento de um processo complexo pela síntese de mecanismos elementares putativos do que por uma análise de cima para baixo, Haykin *et al.* (2001).

Os veículos de Braitenberg são instâncias de uma metodologia sintética. Braitenberg propôs um estudo dos princípios da inteligência através da construção de agentes sucessivamente mais complexos, Pfeifer, Scheier *et al.* (1999).

O próximo tópico importante foi proposto por *Rumelhart, Hinton e Williams*, em 1986, e é conhecido algoritmo da *retro-propagação* (traduzido do termo original *Backpropagation*). O algoritmo Backpropagation é uma *arquitetura adaptável*, e não somente uma regra local de ajuste sináptico. Ela faz uso da posição do neurônio na rede para indicar como seus pesos sinápticos serão ajustados, Arbib *et al.* (2003). Um ambiente favorável para as aplicações do Backpropagation é centrado em problemas onde tenta-se automatizar aplicações complexas de reconhecimento de padrão, Freeman, Skapura *et al.* (1991).

A tarefa do Backpropagation é treinar um Perceptron multi-camadas (feedforward): uma rede que possui suas unidades distribuídas entre as camadas, onde cada unidade fornece valores somente às entradas das unidades da camada seguinte, Arbib *et al.* (2003). Observe na Figura 3.9 o comportamento típico de uma Backpropagation.



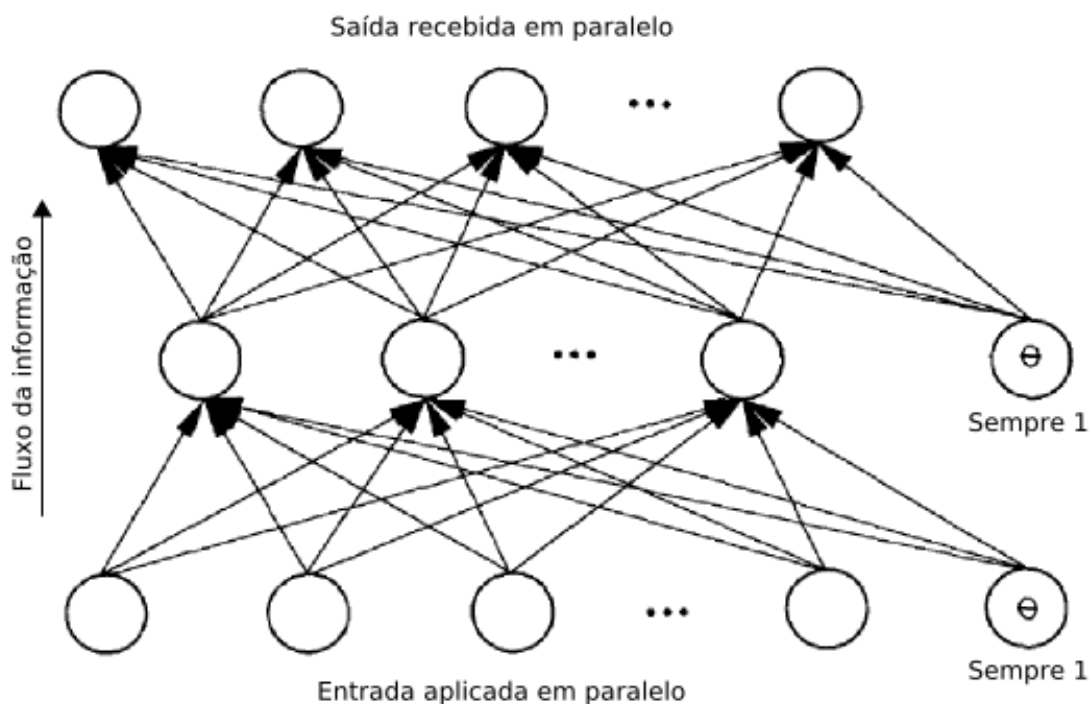


Figura 3.9 – Backpropagation. Freeman e Skapura (1991)

A primeira camada corresponde a unidades de entrada fixas. Na sequência podem existir diversas camadas formadas por unidades de treinamento escondidas, que carregam suas próprias representações (ponderações). Esse tipo de camada é conhecido como *camada escondida*. Finalmente, existe uma camada de saída, com treinamento análogo. Para construir essa estrutura é necessário, portanto, que o valor de saída definido por todas as camadas, exceto a de saída (que independe dessa regra), seja apresentado em uma escala contínua, geralmente estabelecida entre os limites  $[0,1]$ , Arbib *et al.* (2003).

O grande diferencial deste modelo centra-se no método de aprendizado – na verdade, Backpropagation é um tipo de aprendizado que pode ser utilizado pelo Perceptron multi-camadas. Basicamente, a aprendizagem por retro-propagação do erro consiste de dois passos através das diferentes camadas da rede: um passo para frente, a *propagação*, e um passo para trás, a *retro-propagação*. No *passo para frente*, um padrão de atividade (vetor de entrada) é aplicado aos nós sensoriais da rede e seu efeito se propaga através da rede, camada por camada. Finalmente, um conjunto de saídas é produzido como a resposta real da rede. Durante o passo de propagação, os pesos sinápticos da rede são todos *fixos*.

Durante o *passo para trás*, por outro lado, os pesos sinápticos são todos ajustados de acordo com uma regra de correção de erro, Haykin *et al.* (2001).

Por conveniência da organização do trabalho, a contextualização detalhada do algoritmo de retro-propagação, aqui, brevemente descrito, será demonstrada em separado nos próximos capítulos. Isto se deve ao fato de que, toda a proposta de melhoria realizada neste estudo será enfatizado sobre os modelos *Perceptron* e *Perceptron multi-camadas* (sob o aspecto da retro-propagação).

Outro trabalho baseado na *teoria da informação*, no período posterior às observações de *Minsky* e *Papert*, foi concebido por Ralph Linkser, no ano de 1988. Ele ofereceu uma nova metodologia para uma rede auto-organizável. De acordo com Haykin (2001), o princípio é concebido para preservar o máximo de informação sobre os padrões de atividade das entradas, sujeito a limitações como as conexões sinápticas e o intervalo dinâmico das sinapses. Todo seu trabalho foi baseado no sistema visual biológico. O sistema visual é o sistema perceptivo de maior estudo nos mamíferos. A informação visual é processada em estágios. Aspectos simples, tal como o contraste e orientação no ambiente, são analisados como estágios separados. As características mais complexas são analisadas posteriormente. Outros aspectos do processamento visual, como a cor e a noção de movimento, são processadas em paralelo com a análise das formas, Linsker *et al.* (1988). Esses conceitos, entre outros, foram analisados do ponto de vista artificial em seu trabalho.

Também em 1988, Broomhead e Lowe descreveram um procedimento para o projeto de redes alimentadas adiante, em camadas utilizando *funções de base radial* (RBF, *radial basis function*), às quais fornecem uma alternativa aos perceptrons de múltiplas camadas, Haykin *et al.* (2001). As *redes de base radial* – nome associado à metodologia que utiliza as funções de base radial – podem requerer um número maior de neurônios do que o padrão de rede Backpropagation feedforward, mas mesmo assim podem ocupar uma fração do tempo de treinamento usado nessas redes, Demuth, Beale, Hagan *et al.* (2007). Problemas como *regressão linear* e *classificação de padrões* são comumente tratados pelas redes de bases radial, devido à embasamento estatístico que ela possui – muitas vezes considerada como um *aproximador universal*.

Uma função de base radial, representada por  $\phi$ , resulta em uma saída simétrica em torno de um centro associado  $\mu_c$ . Esta representação é definida na equação (3.39), onde  $\|\cdot\|$  é a norma do vetor, Howlett, Jain *et al.* (2001)

$$\phi_c(\mathbf{x}) = \phi(\|\mathbf{x} - \mu_c\|; \sigma) \quad (3.39)$$

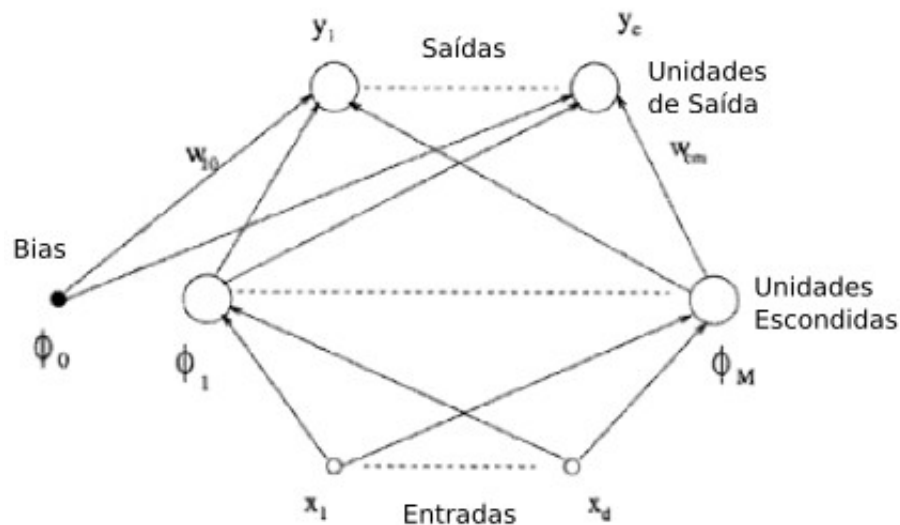


Figura 3.10 – Rede com Função Base Radial. Howlett e Jain (2001).

A construção de uma rede de função de base radial, em sua forma básica, envolve três camadas com papéis totalmente diferentes. A camada de entrada é constituída por nós de fonte (unidades sensoriais) que conectam a rede ao seu ambiente. A segunda camada, a única camada oculta da rede, aplica uma transformação não-linear do espaço de entrada para o espaço oculto; na maioria das vezes o espaço oculto é de alta *dimensionalidade*. A camada de saída é linear, fornecendo a resposta da rede ao padrão (sinal) de ativação aplicada à camada de entrada, Haykin *et al.* (2001). A contextualização sugerida por Haykin (2001), descrita anteriormente, pode ser observada através da Figura 3.10.

A próxima contribuição considerável é dada no início dos anos 90 através das *Máquinas de Vetor de Suporte* (SVM – Support Vector Machines). Esse estudo, desenvolvido por Vapnik e co-autores, abordava uma classe de redes de aprendizagem supervisionada poderosa do ponto de vista computacional e

geralmente são usadas para problemas de reconhecimento de padrões, regressão e estimação de densidade, Haykin *et al.* (2001). De forma análoga à rede neural artificial, ou similar a ela, a Máquina de Vetor de Suporte possui a habilidade e conhecimento de aproximador universal para quaisquer funções multi-variadas e com qualquer grau de exatidão desejado. Conseqüentemente, existe um interesse em modelar processos desconhecidos, parcialmente desconhecidos, com comportamento não-linear, simples ou complexos, Wang *et al.* (2005). Segundo Haykin (2001), uma característica inovadora das máquinas de vetor de suporte é o modo natural pelo qual a *dimensão de Vapnik-Chervonenkis (V-C)* é incorporado no seu projeto.

De fato, além do escopo intelectual e científico apresentado no decorrer deste capítulo, a ciência das redes neurais artificiais engloba muitos outros textos e pesquisas durante o período descrito e subsequente. A característica multidisciplinar, conforme observado é inata, e tende a absorver conteúdo da matemática, física, computação, psicologia, engenharia e principalmente neurociência. Apesar de breve, o conteúdo explícito neste capítulo será necessário para a proposta firmada neste projeto.

#### 4. Contextualização Neurológica

O cérebro humano, para se tornar apto a trabalhar e inferir respostas apropriadas ao ambiente em que vive, concebe a reorganização das redes de neurônios e o reforço das conexões sinápticas associadas ao estímulo do aprendizado. Esta característica é conhecida como plasticidade neural, Coudry e Freire (2005). Dentre vários fenômenos pertinentes à plasticidade neural, pode-se destacar a potenciação de longa duração – LTP, do inglês *long-term potentiation*, Milano e Rosat *et al.* (1998) – que consiste de um modelo celular dos processos de memória atuantes em um importante conjunto de células nervosas, geralmente em sinapses glutamatérgicas. Através do LTP, a amplitude da resposta sináptica cresce e pode ser mantida ativa durante dias, sendo que isso ocorre através de estímulos externos – do ambiente.

De acordo com Santos, Milano e Rosat (1998), estudos em modelos de animais têm mostrado que a formação da memória envolve uma série de alterações bioquímicas em várias áreas do sistema nervoso central (SNC), entre as quais se destaca o hipocampo. Os eventos bioquímicos envolvidos na formação da memória incluem, inicialmente, a ativação de receptores glutamatérgicos dos tipos N-metil-D-aspartato (NMDA) e metabotrópico (mGluRs), e a ativação de cascatas bioquímicas nos neurônios. Entre as proteínas cerebrais envolvidas nessas cascatas, destacam-se a proteína quinase A (PKA), a proteína quinase C (PKC), a proteína quinase dependente de GMPc (PKG) e a cálcio-calmodulina quinase II (CaMKII).

Centrando-se na influência das cascatas bioquímicas, é importante ressaltar que sua ocorrência está diretamente relacionada aos estímulos provocados pelo ambiente, que de forma mais clara, podem ser entendidos como elementos do aprendizado. O processo ocorre graças à mediação da fosforilação – principalmente conduzida pela proteína CaMKII, Rosler, Quevedo *et al.* (1998).

Através desta reação, os canais sinápticos estarão mais preparados para a condução dos estímulos. Em um segundo momento, novas reações envolvendo os mesmos elementos ocorrerão por influência de um estímulo análogo ao primeiro.

Pode-se então inferir que, na prática, quando um tipo de estímulo é realizado várias vezes, uma reestruturação no cérebro ocorre, buscando estimular o aprendizado de determinado conceito.

De forma mais clara, o processo de aprendizado, inicialmente, é capaz de reestruturar bioquimicamente a organização da rede neural concebendo as futuras inferências relacionadas com o assunto aprendido. Ou seja, durante o aprendizado de um novo conceito, o cérebro se organiza apropriadamente para tratar sobre aquele conceito. Durante o processo de reaprendizagem, o cérebro é capaz de reforçar os elementos envolvidos na inferência do aprendizado. Isso o torna mais apto a tratar aquele tipo de ocorrência. Em outras palavras, reforça as ligações entre os neurônios de forma a facilitar o entendimento do assunto aprendido. Este tipo de reforço sináptico, de certa forma, contribui para o aprendizado humano.

Sob perspectiva empírica, mostrada no Capítulo 7, a representação dessas características biológicas em modelos artificiais existentes trouxe novos benefícios para a relação desempenho/resultados das Redes Neurais Artificiais. Espera-se que através dessas modificações, os modelos fiquem mais próximos da realidade, apresentando diferenciações em sua absorção dos elementos de entrada, para benefício de um resultado mais satisfatório.

## 5. Backpropagation

Tem-se neste capítulo, uma referência mais completa sobre a característica – na verdade, procedimento de aprendizado – *backpropagation* de possível utilização nas Redes Neurais Artificiais. Em síntese, pode-se destacar, sobre esta forma de aprendizado supervisionado, que as grandezas encontradas na diferenciação entre a resposta esperada e a obtida (em outras palavras, o erro), são aplicadas a todo o escopo da rede, influenciando os elementos presentes nas camadas escondidas e iniciais – de forma satisfatória e apropriada.

Através dessa modificação, os pesos sinápticos serão definidos com a aspiração de inferir novos resultados, sempre mais próximos aos resultados alvo – esperados. Durante toda a trajetória da Rede Backpropagation, sua minimização do erro é tratada por diversos algoritmos de busca, cada um com sua particularidade e aplicação. Os resultados obtidos por esse tipo de rede são satisfatórios, tornando-a, do ponto de vista da otimização e aproximação, uma Rede Neural bastante confiável.

### 5.1 Características

O backpropagation é um algoritmo de aprendizado utilizado por uma rede Perceptron de múltiplas camadas. A filosofia da rede Perceptron é mantida nesta rede, preservando a idéia de aprendizado supervisionado, no qual a rede é capaz de se ajustar e aproximar dos resultados esperados. A melhoria que a técnica de aprendizado backpropagation acrescenta no modelo, está centrada na possibilidade de correção dos erros – diferenças entre um valor ideal e o atual – inerente aos pesos sinápticos, de todas as camadas.

O processo de aprendizagem backpropagation modifica a importância de seus pesos através de um fluxo *inverso* ao da propagação do sinal. A propagação do sinal ocorre quando as entradas trafegam pela rede sendo adequadas pelos pesos sinápticos e limitadas pelas funções de ativação. Esse fluxo tem origem na camada de entrada, e subsequência nas camadas escondidas.

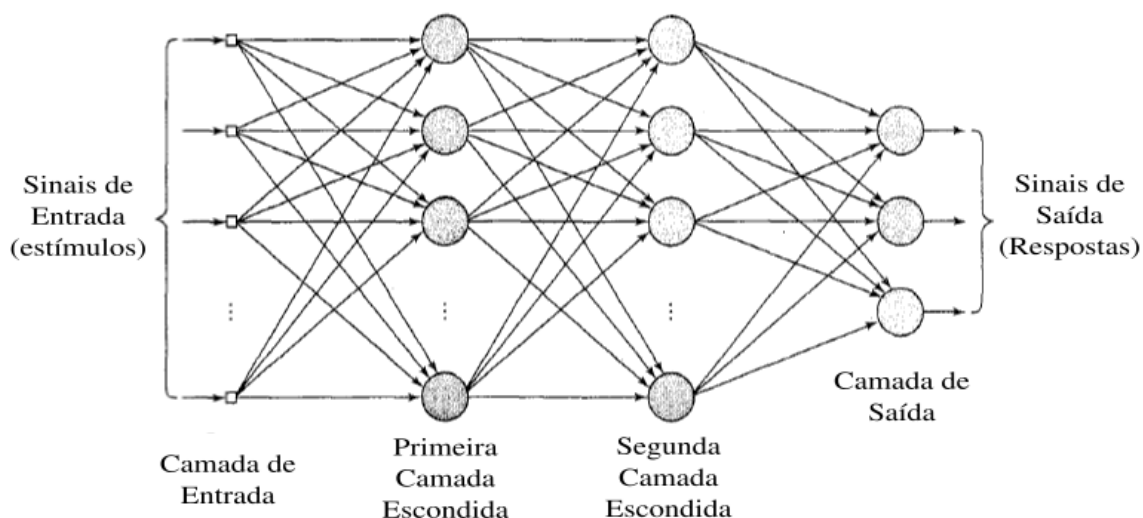


Figura 5.1 – Perceptron com múltiplas camadas. Haykin (2001).

O fluxo apresentado na Figura 5.1 corresponde ao tráfego do sinal de entrada para a saída da rede, com as devidas alterações no valor. O processo de propagação é finalizado na última camada da rede, onde é obtido o sinal de saída. Através da diferença entre o alvo e o sinal de saída da rede, é calculado o valor do erro. Por consequência do algoritmo backpropagation, esse erro é propagado na ordem inversa (retropropagação), alterando os pesos de todas as camadas. A última camada será consideravelmente afetada. As duas camadas escondidas (primeira camada escondida e segunda camada escondida, conforme visualizado na Figura 5.1) sofrerão adaptações de intensidade inferior às alterações aplicadas à camada de saída. De acordo com Haykin (2001), a camada de entrada apenas representa o vetor dos elementos de estímulo, sem a presença de quaisquer valores de influência para a rede – no caso, os pesos sinápticos. Por isso ela não sofrerá alteração.

A atualização sequencial dos pesos é o método preferido para a implementação em tempo de execução do algoritmo de retropropagação, conforme Haykin *et al.* (2001). Para este modo de operação, o algoritmo circula através da amostra de treinamento  $\{x(n), d(n)\}_{n=1}^N$  como segue:

(a) *Inicialização*: Assumindo que nenhuma informação prévia esteja disponível,



retire os pesos sinápticos e lineares de uma distribuição uniforme cuja média é zero e cuja variância é escolhida para que o desvio padrão dos campos locais induzidos dos neurônios esteja na transição entre as partes linear e saturada da função de ativação sigmóide.

(b) *Apresentação dos Exemplos de Treinamento:* Apresente uma época de exemplos de treinamento à rede. Para cada exemplo do conjunto, ordenado de alguma forma, realize a seqüência de computações para frente e para trás descritas, respectivamente, nos tópicos à seguir.

(c) *Computação para Frente (Propagação):* Suponha que um exemplo de treinamento da época seja representado por  $(x(n), d(n))$ , com o vetor de entrada  $x(n)$  (sinais de entrada) aplicado à camada de entrada de nós sensoriais e o vetor resposta desejada  $d(n)$  apresentado à camada de saída de nós computacionais. Calcule os campos locais induzidos e os sinais funcionais da rede prosseguindo para frente através da rede, camada por camada. O campo local induzido  $v_j^{(l)}(n)$  para o neurônio  $j$  na camada  $l$  é

$$v_j^{(l)}(n) = \sum_{i=0}^{m_0} w_{ji}^{(l)}(n) y_i^{(l-1)}(n) \quad (5.1)$$

onde  $y_i^{(l-1)}(n)$  é o sinal (função) de saída do neurônio  $i$  na camada superior  $l-1$ , na iteração  $n$ , e  $w_{ji}^{(l)}(n)$  é o peso sináptico do neurônio  $j$  da camada  $l$ , que é alimentado pelo neurônio  $i$  da camada  $l-1$ . Para  $i=0$ , temos  $y_0^{(l-1)}(n) = +1$  e  $w_{j0}^{(l)}(n) = b_j^{(l)}(n)$  é o bias aplicado ao neurônio  $j$  na camada  $l$ . Assumindo-se o uso de uma função sigmóide, o sinal de saída do neurônio  $j$  na camada  $l$  é

$$y_j^{(l)} = \varphi_j(v_j(n)) \quad (5.2)$$

Se o neurônio  $j$  está na primeira camada oculta (i.e.,  $l=1$ ), faça

$$y_j^{(0)}(n) = x_j(n) \quad (5.3)$$

onde  $x_j(n)$  é o  $j$ -ésimo elemento do vetor de entrada  $x(n)$ . Se o neurônio  $j$  está na camada de saída (i.e.,  $l=L$ , onde  $L$  é denominado a *profundidade* da rede), faça

$$y_j^{(L)} = o_j(n) \quad (5.4)$$

Calcule o erro

$$e_j(n) = d_j(n) - o_j(n) \quad (5.5)$$

onde  $d_j(n)$  é o  $j$ -ésimo elemento do vetor resposta desejada  $d(n)$ .

(d) *Computação para Trás (Retropropagação)*: Calcule os  $\delta$  (i.e., gradientes locais) da rede, definidos por

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(l)}(n) \varphi_j'(\nu_j^{(L)}(n)) & \text{para o neurônio } j \text{ da camada de saída } L \\ \varphi_j'(\nu_j^{(l)}(n))' \sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) & \text{para o neurônio } j \text{ da camada oculta } l \end{cases} \quad (5.6)$$

onde o apóstrofe em  $\varphi_j'(\cdot)$  representa a diferenciação em relação ao argumento. Ajuste os pesos da rede nas camadas através da regra delta

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n) \quad (5.7)$$

onde  $\eta$  é a taxa de aprendizagem. Segundo Rumelhart, Hinton e Williams (1986), existe uma alternativa à regra delta, que utiliza  $\alpha$  como constante de momento. A constante de momento  $\alpha$  afeta o ajuste dos pesos de acordo com a atual iteração da rede. Isso oferece um comportamento dinâmico à taxa de aprendizagem, que no modelo (5.7) permanece estática.

(e) *Iteração*: Itere as computações para frente e para trás, das seções anteriores, apresentando novas épocas de exemplos de treinamento para a rede, até que seja

satisfeito o critério de parada.

Os tópicos anteriores compõem o formato principal do algoritmo de aprendizado backpropagation, sendo que, a ordem de apresentação dos exemplos de treinamento deve ser aleatória, de época para época. Os parâmetros de momento, sugeridos por Rumelhart, Hinton e Williams (1986) na regra alternativa de ajuste dos pesos, e da taxa de aprendizagem tipicamente são ajustados (e normalmente reduzidos) quando o número de iterações de treinamento aumenta, Haykin *et al.* (2001).

## 5.2 Inicialização dos Pesos

O algoritmo trabalhado na seção anterior mostra um tipo de inicialização para os pesos sinápticos. Todavia, outras formas podem ser utilizadas para proporcionar características distintas à rede, ocasionando particularidades na convergência do modelo de acordo com a aplicação.

Segundo Fausett (1994), outras formas de inicialização dos pesos são plausíveis, tais como a *inicialização aleatória* e a *inicialização de Nguyen-Widrow*. A escolha do tipo de inicialização terá influência direta na convergência, tratando-se de performance e precisão alcançada. Os valores iniciais para os pesos não podem ser muito grandes, ou os sinais de entrada iniciais das unidades poderão cair na região onde a derivada da função de ativação sigmóide tem um valor muito pequeno (região de saturação). E para o caso oposto, no qual os valores iniciais são muito pequenos, os sinais de entrada das unidades serão definidos como zero, proporcionando um aprendizado extremamente lento.

Podem-se utilizar os procedimentos de inicialização dos pesos (e bias, sabendo-se que a bias está associada ao peso de índice zero) descritos a seguir, para evitar situações que possam prejudicar a performance e convergência do modelo de rede, Fausett *et al.* (1994).

*Inicialização Aleatória:* Definir valores aleatórios para os pesos entre -0.5 e 0.5 (ou entre -1 e 1). Os valores podem ser positivos ou negativos, de forma que serão

adaptados no decorrer do treinamento. Uma simples alteração na inicialização aleatória, desenvolvida por Nguyen e Widrow (1990), é mostrada adiante.

*Inicialização Nguyen-Widrow:* Apesar da simples alteração aplicada à inicialização aleatória, pode-se obter uma melhoria considerável no desempenho da rede. A pesquisa é baseada na análise geométrica das respostas dos neurônios escondidos para uma entrada simples.

Os pesos associados aos neurônios da camada escondida para a camada de saída são inicializados aleatoriamente entre -0.5 e 0.5, como no caso normal. Já para os pesos associados à camada de entrada, que no caso das definições de Haykin (2001), são tratados como primeira camada escondida, deve-se utilizar a seqüência de equações a seguir. Inicialmente os pesos da primeira camada são inicializados aleatoriamente, em seguida, seus valores serão modificados pelas fórmulas propostas.

$$\|v_j(\text{antigo})\| = \sqrt{v_{1j}(\text{antigo})^2 + v_{2j}(\text{antigo})^2 + \dots + v_{nj}(\text{antigo})^2} \quad (5.8)$$

$$v_{ij} = \frac{\beta v_{ij}(\text{antigo})}{\|v_j(\text{antigo})\|} \quad (5.9)$$

onde  $n$  é o número de unidades de entrada (primeira camada escondida),  $p$  o número de unidades escondidas (segunda camada escondida) e  $\beta$  é o valor definido por  $\beta = 0.7(p)^{1/n} = 0.7\sqrt[n]{p}$ . A bias  $v_{0j}$  será definida com um valor aleatório entre  $-\beta$  e  $\beta$ . Observe que existem algumas distinções entre a formalização utilizada por Fausett (1994) e Haykin (2001), onde, respectivamente, os pesos são tratados pela notação  $v$  e  $w$ .

### 5.3 Parâmetros da Rede

Existem algumas particularidades que devem ser observadas a respeito da configuração de uma rede com aprendizado de retropropagação. Em primeira instância pode-se observar a importância do critério de parada. Segundo Haykin (2001), não se pode demonstrar que o algoritmo de retropropagação convergiu e

não existem critérios bem-definidos para encerrar a sua operação. Em vez disso, existem alguns critérios razoáveis, cada um com o seu mérito prático, particular, que podem ser usados para encerrar o ajuste dos pesos.

Em se tratando da minimização, deve-se observar a convergência relacionada ao erro. Suponha que  $w^*$ , vetor de peso, represente um mínimo, seja ele, local ou global. Uma condição necessária para que  $w^*$  seja mínimo é que o vetor gradiente  $g(w)$  (i.e., a derivada parcial de primeira ordem) da superfície de erro em relação ao vetor de peso  $w$  seja zero em  $w=w^*$ . Considerando o critério  $w=w^*$ , o tempo de aprendizado pode ser longo. Deve-se também salientar a complexidade computacional envolvida no cálculo do vetor gradiente  $g(w)$ .

Outra propriedade do mínimo que se pode utilizar é o fato de que a função de custo ou medida de erro  $\varepsilon_{med}(w)$ , observada na fórmula abaixo, é estacionária no ponto  $w=w^*$ . Estacionária no sentido de permanecer em  $w=w^*$ , uma vez que o valor é encontrado, independente das iterações posteriores.

$$\varepsilon_{med} = \frac{1}{2N} \sum_{n=1}^N \sum_{j \in C} e_j^2(n) \quad (5.10)$$

Há outro critério de convergência útil e teoricamente fundamentando. Após cada iteração de aprendizagem, a rede é testada pelo seu desempenho de generalização. O processo de aprendizagem é encerrado quando o desempenho de generalização for adequado, ou quando ficar aparente que o desempenho de generalização alcançou o máximo Haykin *et al.* (2001).

Outra importante característica que deve ser observada é o tempo (durabilidade de execução) de treinamento necessário para que uma rede de retropropagação realize uma convergência considerável. Durante a fase de treinamento, busca-se um ponto de convergência que satisfaça duas características importantes: memorização e generalização. Pode-se observar que as duas características devem ser adquiridas em conjunto, evitando casos comuns, em que a rede tenha boa capacidade na utilização da memorização, mas não realiza a generalização de maneira satisfatória. De forma análoga, o contrário também é possível, onde uma rede possui grande capacidade de generalização,

mas pouca flexibilidade nos casos onde a memorização é necessária.

Essas características podem ser em parte, tratadas durante o treinamento, quando são utilizados dois grupos distintos de padrões (sinais de entrada/saída desejada): dados de treinamento e dados de teste. O comportamento da taxa de erro pode ser observado para caracterizar o poder de generalização/memorização da rede durante o treinamento. Quando o erro está sendo minimizado, o treinamento deve continuar, sem provocar inconsistência na rede. Todavia, quando o erro encontrar um ponto estável e em seguida apresentar crescimento – em relação ao comportamento anterior, significa que a rede está aumentando o nível de memorização de forma irregular, e por consequência, prejudicando a capacidade de generalização, Fausett *et al* (1994).

Outra observação de relativa importância, que possui grande impacto na comunidade científica, é o número de camadas ocultas na rede, tal como o número de neurônios presentes em cada camada. De forma simplificada, pode-se chegar à conclusão que esta característica está intrínseca na forma empírica de organização da rede, ou seja, é o tipo de parâmetro que deve ser observado, durante testes, efetuados no treinamento, definindo assim, valores satisfatórios com os resultados esperados. Existem diversos estudos associados a esse contexto, porém, a base teórica relacionada não fará parte do escopo deste trabalho.

## 5.4 Algoritmo

De forma resumida, nesta seção, tratar-se-á dos passos do algoritmo de treinamento backpropagation relacionado à rede perceptron multi-camadas. Aqui será trabalhado com a notação sugerida por Haykin (2001), mesmo quando a referência estiver associada à outra literatura. Isso tornará a coesão do texto mais conveniente para o entendimento, devido a uma única notação.

Deve-se ressaltar que o algoritmo de treinamento backpropagation (conforme tratado no tópico 5.1 deste capítulo) é resolvido em duas fases: a fase de propagação – responsável pela transmissão sequencial dos sinais de entrada, propagando-os pela rede, até que seja obtido o sinal de saída – e a fase de

retropropagação – responsável pelo cálculo do erro gerado pela rede, e correção das influências sinápticas (pesos) das camadas. Esses procedimentos são realizados em  $n$  iterações, com expectativa de atender algum critério de parada específico – geralmente associado à minimização do erro.

O algoritmo abaixo demonstra, sob notação matemática, o comportamento algorítmico da rede, em passos a serem executados, orientados pela indentação do texto. O procedimento será realizado até que a *Energia Média do Erro Quadrático* seja minimizada. O exemplo abaixo trata de uma rede com uma camada oculta e uma camada de saída, onde, respectivamente, existem três neurônios caracterizados com a função sigmóide, e dois neurônios com a função linear. Estes termos foram definidos por conveniência da apresentação do algoritmo.

#### **Parâmetros iniciais da rede:**

*Parâmetro que identifica a atual iteração na rede:*  $n$

*Índice dos sinais de entrada:*  $i$

*Índice dos neurônios da primeira camada (oculta):*  $j$

*Índice dos neurônios da segunda camada (saída):*  $k$

*Função de ativação sigmóide da primeira camada (oculta):*  $\varphi^{(1)}(.)$

*Função de ativação linear da segunda camada (saída):*  $\varphi^{(2)}(.)$

*Inicialização aleatória dos pesos e bias da primeira camada (oculta):*

$W_{ji}^{(1)}(n)$  e  $b_j^{(1)}(n)$  são inicializados com valores aleatórios

*Inicialização aleatória dos pesos e bias da segunda camada (de saída):*

$W_{kj}^{(2)}(n)$  e  $b_k^{(2)}(n)$  são inicializados com valores aleatórios

*O critério de parada da rede será definido pela Energia Média do Erro Quadrático*

$$\varepsilon_{med} = \frac{1}{N} \sum_{n=1}^N \varepsilon(n) \quad \text{onde} \quad \varepsilon(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n)$$

*onde C inclui todos os sinais de entrada da rede*

*O critério de parada da rede é  $\theta$  e a taxa de treinamento é dada por  $\eta$*

#### **Procedimento de propagação:**

*Cálculo do Campo Local Induzido da primeira camada oculta:*

$$v_j^{(1)}(n) = b_j^{(1)} + \sum_{i=1}^m w_{ji}^{(1)}(n) x_i(n), \quad \forall j$$

*Resultado da primeira camada oculta:*

$$y_j(n) = \varphi^{(1)}[v_j^{(1)}(n)], \quad \forall j$$

*Cálculo do **Campo Local Induzido** da camada de saída :*

$$v_k^{(2)}(n) = b_k^{(2)} + \sum_{i=1}^j w_{ki}^{(2)}(n) y_i(n), \quad \forall k$$

*Resultado da camada de saída :*

$$o_k(n) = \varphi^{(2)}[v_k^{(2)}(n)], \quad \forall k$$

*Cálculo do erro :*

$$e_k(n) = d_k(n) - o_k(n), \quad \forall k$$

***Procedimento de retropropagação:***

*Cálculo do gradiente da camada de saída :*

$$\delta_k^{(2)}(n) = e_k(n) \varphi'^{(2)}[v_k^{(2)}(n)], \quad \forall k$$

*Cálculo da correção dos pesos e bias da camada de saída :*

$$\Delta W_{kj}^{(2)}(n) = \eta \delta_k^{(2)}(n) y_j(n), \quad \forall k \quad e \quad \Delta b_k^{(2)}(n) = \eta \delta_k^{(2)}(n), \quad \forall k$$

*Cálculo do gradiente da primeira camada oculta :*

$$\delta_j^{(1)}(n) = \varphi'^{(1)}[v_j^{(1)}(n)] \sum_k \delta_k^{(2)}(n) w_{kj}^{(2)}(n), \quad \forall j$$

*Cálculo da correção dos pesos e bias da primeira camada oculta :*

$$\Delta W_{ji}^{(1)}(n) = \eta \delta_j^{(1)}(n) x_i(n), \quad \forall j \quad e \quad \Delta b_j^{(1)}(n) = \eta \delta_j^{(1)}(n), \quad \forall j$$

*Ajuste dos pesos e bias :*

$$W_{kj}^{(2)}(n+1) = W_{kj}^{(2)}(n) + \Delta W_{kj}^{(2)}(n), \quad \forall k, \quad \forall j$$

$$W_{ji}^{(1)}(n+1) = W_{ji}^{(1)}(n) + \Delta W_{ji}^{(1)}(n), \quad \forall j, \quad \forall i$$

$$b_k^{(2)}(n+1) = b_k^{(2)}(n) + \Delta b_k^{(2)}(n), \quad \forall k$$

$$b_j^{(1)}(n+1) = b_j^{(1)}(n) + \Delta b_j^{(1)}(n), \quad \forall j$$

$$\text{Calcular } \varepsilon_{med} = \frac{1}{N} \sum_{n=1}^N \varepsilon(n) \quad e \quad \text{fazer } n = n + 1$$

*Repetir procedimento de **propagação e retropropagação** enquanto  $\varepsilon_{med} > \theta$*

Algoritmo do aprendizado backpropagation.



Outros conceitos que podem ser observados estão associados aos parâmetros utilizados pela rede em sua etapa de treinamento, que implicam diretamente na performance do processo e contribuem para a minimização do erro – atribuída ao cálculo da Energia Média do Erro Quadrático.

Um dos parâmetros é definido dentro da função de ativação sigmóide, representado por  $\alpha$ , denominado *Parâmetro de Inclinação*. Este parâmetro contribui para o resultado do cálculo do Campo Local Induzido da primeira camada oculta, no exemplo anterior. O parâmetro de inclinação é responsável pelo comportamento da função sigmóide – comportamento da curvatura da função.

A Taxa de Aprendizado  $\eta$  implica na velocidade de convergência relacionada à otimização do gradiente descendente, que é aplicada às matrizes de pesos sinápticos da rede. Uma taxa de aprendizado pequena contribui para uma otimização mais precisa, porém demorada. O contrário favorece um procedimento mais rápido, contudo implica em uma menor precisão. A melhor definição deste parâmetro é encontrada através da análise empírica do modelo.

Por fim, pode-se destacar o critério de parada  $\theta$ . Ele será comparado, em cada iteração, à Energia Média do Erro Quadrático  $\varepsilon_{med}$  para que seja estabelecido um momento onde a convergência da rede (minimização de  $\varepsilon_{med}$ ) tenha alcançado um ponto satisfatório, próximo ao mínimo global (ou, provavelmente, local).

## 6. Backpropagation Alterada

Todo estudo associado às alterações propostas em relação ao modelo mencionado no capítulo anterior, originam-se das perspectivas biológicas observadas no Capítulo 4. O primeiro tópico deste capítulo observará, com certa abstração, que importância os conceitos biológicos trazem sob o aprendizado humano. A notação química será desnecessária, porém maiores detalhes podem ser observados nos trabalhos de Rosler e Quevedo (1998) e Milano e Rosat (1998), ou inclusive, no Capítulo 4, já mencionado.

### 6.1 Perspectiva Biológica

De fato, milhares de variáveis estão associadas ao aprendizado humano. A característica da plasticidade neural biológica, não só atribuída ao homem racional, é fator marcante na proficiência da memorização e generalização proferida pelos seres vivos. Para induzir o entendimento de determinado conceito, o cérebro é capaz de realizar modificações em suas conexões intracelulares, de forma a aumentar o fluxo das ligações existentes e/ou criar novos caminhos de comunicação entre os neurônios.

Por conveniência, o entendimento do processo de aprendizagem, será apresentada uma simulação fictícia, onde um ser concentra-se no aprendizado de dois tópicos distintos, aqui tratados como, *Objetos de Conhecimentos*, como pode ser observado na Figura 6.1.



Figura 6.1 – Objetos de Conhecimento.

Das muitas variáveis relacionadas com o aprendizado do *Objeto de Conhecimento Matemática*, podem-se ressaltar similaridades com as do

aprendizado do *Português*. Porém, o cérebro é capaz de se auto-organizar, de forma a proporcionar uma assimilação dos conceitos sem prejudicar – de início – o tratamento e entendimento do outro. A Figura 6.2 mostra que o aprendizado dos *Objetos de Conhecimento* são associados a determinadas características do ambiente – os sinais de entrada, sendo que, existem distinções intrínsecas nas variáveis – aqui bem destacadas para denotar o exemplo tratado.

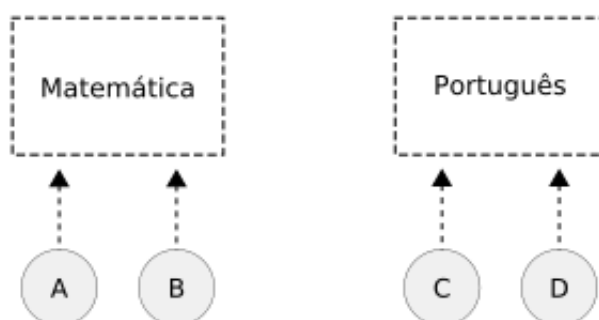


Figura 6.2 – Aprendizado dos Objetos de Conhecimento.

Atente que foram destacadas 4 variáveis denotadas pelas letras A, B, C e D, onde, respectivamente, as duas primeiras estão agregadas ao aprendizado do *Objeto de Conhecimento Matemática* e as duas últimas ao *Objeto de Conhecimento Português*.

A característica plasticidade oferece ao neurônio o poder de fortalecer as ligações sinápticas, favorecendo os sinais químicos associados a determinado aprendizado. Isso significa que, quando está se lidando com variáveis pertinentes ao aprendizado de *Matemática*, o cérebro trabalhará com as ligações criadas para este conceito, fortalecendo-as e favorecendo o aprendizado do conteúdo. Isso é possível por conta das particularidades existentes nos fatores de entrada. Ou seja, ao distinguir que os elementos particulares A e B estão sendo absorvidos pelos sinais sensoriais (visão ou audição, por exemplo), o cérebro interpreta os mesmos com uma certa distinção, tentando fortalecer somente as ligações sinápticas pertinentes àquele aprendizado.

Observe a Figura 6.3. O processo de aprendizagem está focado na obtenção do *Objeto de Conhecimento Matemática*. Os sinais de entrada, são representados pelos círculos A e B. As ligações sinápticas associadas ao processo

estão tracejadas, de forma a destacar quais ligações estão sendo fortalecidas.

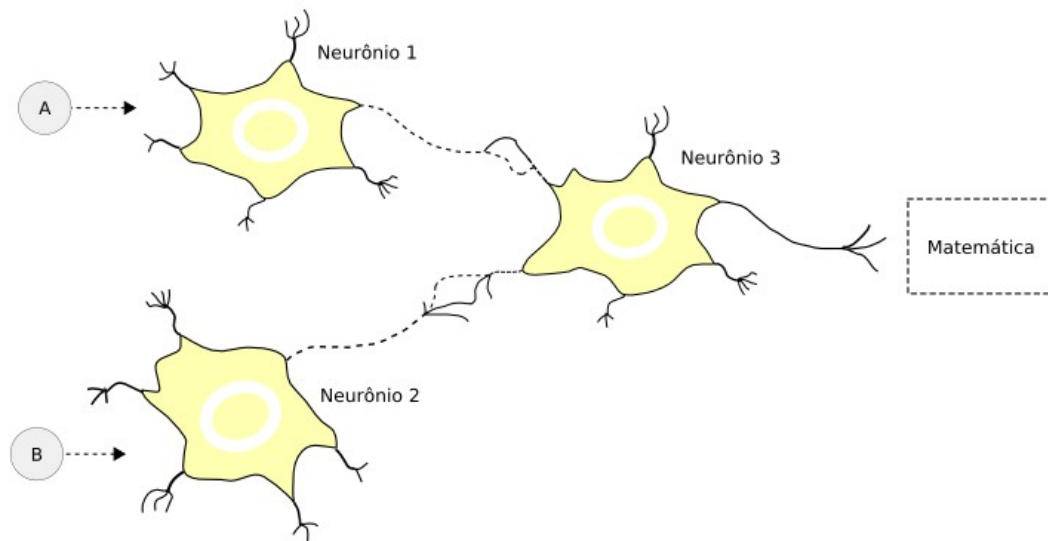


Figura 6.3 – Aprendizado da Matemática.

Sabe-se que o conceito A está sendo reservado ao Neurônio 1, e B ao Neurônio 2. Observe que, como ligação sináptica entre os Neurônios 1 e 3, existem dois percursos (linha tracejada). O cérebro é capaz de inferir que, como o aprendizado volta-se à Matemática, somente o percurso inferior entre os Neurônios 1 e 3 está sendo utilizado. De forma similar, pode-se observar o percurso definido entre os Neurônios 2 e 3.

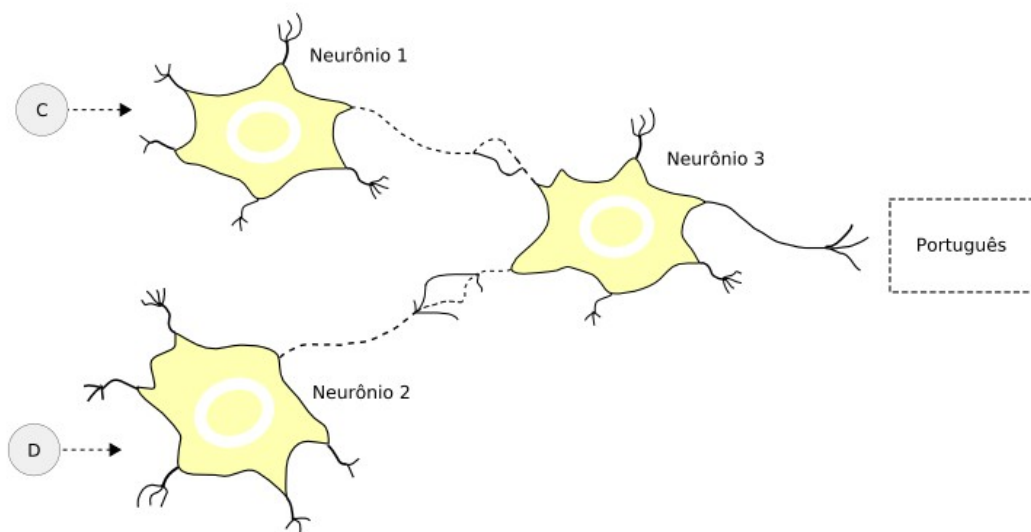


Figura 6.4 – Aprendizado do Português.

Perceba que todo o esforço de melhoria sináptica é aplicado justamente ao caminho tracejado, sem comprometer as outras ligações. Na Figura 6.4, onde é proferido o aprendizado do Português, as conexões utilizadas são distintas do primeiro exemplo. Este é justamente o fator diferencial que será desenvolvido em termos matemáticos para aperfeiçoar o treinamento backpropagation.

## 6.2 Objetos de Conhecimento

Um Objeto de Conhecimento estabelece determinado conceito que será aprendido pela rede. No aprendizado supervisionado, as saídas esperadas são responsáveis pela definição dos Objetos de Conhecimento. Em um trabalho de reconhecimento de padrões, por exemplo, cada padrão pré-estabelecido, representará um Objeto de Conhecimento. Em notação matemática, os Objetos de Conhecimentos serão representados por  $\rho$ .

Em uma rede com dois padrões a serem classificados, o valor de  $\rho$  é 2. Quando a rede estiver encarregada de proferir melhorias nos pesos sinápticos a respeito do Objeto de Conhecimento 1, ou seja,  $\rho=1$ , as matrizes de peso a serem modificadas serão particulares. Quando  $\rho=2$ , outras matrizes deverão ser modificadas (ajustadas).

A rede biológica é capaz de controlar qual ligação sináptica deverá ser trabalhada pelo sinal de entrada. De forma análoga, o modelo matemático ajustará as matrizes correspondentes de acordo com a entrada da rede. Isto será possível através dos Campos de Atração, introduzidos à seguir.

## 6.3 Campos de Atração

A característica de aprendizagem inata do cérebro, associada ao tratamento específico do Objeto de Conhecimento, será tratada aqui, como Campo de Atração. Através dos Campos de Atração, a rede tratará os sinais de entrada com determinadas ligações sinápticas. De forma biológica, isso significa que, na Figura

6.3, o Campo de Atração conduziu a melhoria sináptica, entre os neurônios 1 e 3, pelo caminho tracejado. De forma análoga, o treinamento representado na Figura 6.4, trabalha pelos caminhos em destaque.

O Campo de Atração, pela notação matemática, é observado através da equação 6.1. O cálculo da Matriz  $A_{i\rho}$  será dado pela média aritmética de todos os valores de entrada com alvo  $\rho$ . O índice  $i$  está associado às entradas da primeira camada oculta, que variam entre 1 e  $m$ . Espera-se encontrar a média dos valores de entrada que proporcionam um alvo específico. Isto será necessário para que seja possível realizar a classificação dos dados do ambiente. Por exemplo, os valores para o aprendizado de Português, conforme na Figura 6.2, possuem determinado comportamento. A Matriz  $A_{i\rho}$  define esse comportamento, que será utilizado durante o treinamento da rede para caracterizar os dados de entrada como sendo associados ao Português – quando se aproximarem da média armazenada em  $A_{i\rho}$  - ou à Matemática – valores próximos de  $A_{i\rho+1}$ .

$$A_{i\rho} = \frac{1}{N_\rho} \sum_{n=1}^N (x_i(n) d_\rho(n)), \quad \forall i, \quad i=1,2,\dots,n \quad (6.1)$$

Deve-se ressaltar que, como se trata de um grupo de dados aplicados à classificação de padrões, os possíveis valores assumidos por  $d_\rho(i)$ ,  $i=1,2,\dots,n$  serão 0 e 1. Isso implica no resultado do somatório, pois as entradas associadas ao resultado esperado  $\rho$  serão multiplicadas por 1 e as entradas do valor alvo  $\rho+1$  serão multiplicadas por 0, não tendo influência direta no somatório. Este cálculo é necessário para se obter a média das entradas exclusivamente relacionada ao alvo  $\rho$  - que será 1, e anular as entradas de  $\rho+1$  - multiplicadas por 0 e conseqüentemente anuladas. O valor de  $N_\rho$  é demonstrado abaixo.

$$N_\rho = \sum_{n=1}^N d_\rho(n), \quad \forall \rho \quad (6.2)$$

Desta forma, a divisão realizada na equação 6.1, com denominador  $N_\rho$ , que indica exatamente o número de elementos com valor 1 (ou seja, elementos

associados ao Campo de Conhecimento em questão), trará o valor da média das entradas (1,2,...,m) para o Campo de Conhecimento tratado.

A média aritmética dos valores de entrada será usada para o cálculo da distância euclidiana. Para cada iteração de treinamento operado pela rede neural artificial, a distância euclidiana será calculada conforme a equação a seguir.

$$d_{\rho}(n) = \sqrt{\sum_{i=1}^m [A_{i\rho} - x_i(n)]^2}, \quad \forall \rho \quad (6.3)$$

O menor dos valores  $d_{\rho}$  encontrado, na iteração  $n$ , implicará na escolha do percurso sináptico adequado ao treinamento. Esta afirmação implica na existência de  $\rho$  matrizes de pesos para cada neurônio da primeira camada oculta. Cada neurônio, terá uma matriz sináptica  ${}_{\rho}W_{ji}(n)$  para cada valor de  $\rho$ .

Durante o aprendizado (ou utilização da rede após etapa de treinamento), na iteração  $n$ , os valores de  $d_{\rho}$  serão calculados. O menor dos valores mostrará, através do índice  $\rho$ , qual das matrizes  ${}_{\rho}W_{ji}(n)$  será utilizada para influenciar os sinais de entrada, e conseqüentemente, será ajustada pela regra de aprendizado backpropagation.

Deve-se observar, pois, que o Campo de Atração só será aplicado aos estímulos da camada diretamente conectada aos sinais de entrada da rede (primeira camada oculta). As demais camadas ocultas, e a camada de saída, terão comportamento idêntico ao apresentado no Capítulo 5.

#### 6.4 Algoritmo

Se comparado com o algoritmo de aprendizado backpropagation apresentado no Capítulo 5, o modelo alterado trará as seguintes modificações:

##### ***Alterações aplicadas à fase inicial:***

*Definir o valor de  $\rho$  de acordo com o número de padrões:* Nesta etapa somente

será definido o número de Objetos de Conhecimento.

*Definição dos Campos de Atração:* Cálculos pertinentes à definição da matriz  $A_{i\rho}$ , que define os valores de atração utilizados nas iterações de treinamento da rede. Cada coluna da matriz representa um dos Objetos de Conhecimento.

*Definição das matrizes sinápticas para cada Objeto de Conhecimento:* Conforme observado, cada neurônio da primeira camada oculta terá  $\rho$  matrizes de pesos sinápticos. Cada uma dessas matrizes deve ser, de início, idêntica à matriz de pesos padrão do treinamento backpropagation.

### ***Alterações aplicadas à fase de propagação:***

*Cálculo das distâncias euclidianas  $d_\rho$ :* Após efetuar o cálculo, conforme apresentado na equação 6.3, deve-se considerar o menor dos valores encontrados, e através do índice, selecionar a matriz  ${}_\rho W_{ji}(n)$  correspondente à iteração.

*Realizar os cálculos de propagação da primeira camada oculta:* Deve-se apenas se atentar para a utilização de  ${}_\rho W_{ji}(n)$ , como matriz de pesos sinápticos.

### ***Alterações aplicadas à fase de retropropagação:***

*Ajuste da matriz sináptica:* De forma similar aos tópicos anteriores, todo ajuste de minimização deverá ser aplicado somente à matriz  ${}_\rho W_{ji}(n)$ , correspondente ao Objeto de Conhecimento  $\rho$ .

No algoritmo abaixo, será demonstrado o comportamento algorítmico da rede backpropagation alterada.

### ***Parâmetros iniciais da rede:***

*Parâmetro que identifica a atual iteração na rede:*  $n$

*Objetos de conhecimento:*  $\rho$

*Índice dos sinais de entrada:*  $i$

*Índice dos neurônios da primeira camada (oculta):*  $j$

*Índice dos neurônios da segunda camada (saída):*  $k$

*Função de ativação sigmóide da primeira camada (oculta):*  $\varphi^{(1)}(.)$

*Função de ativação linear da segunda camada (saída):*  $\varphi^{(2)}(.)$

*Inicialização aleatória dos pesos e bias da primeira camada (oculta):*

$W_{ji}^{(1)}(n)$  e  $b_j^{(1)}(n)$  são inicializados com valores aleatórios

*Inicialização aleatória dos pesos e bias da segunda camada (de saída):*

$W_{kj}^{(2)}(n)$  e  $b_k^{(2)}(n)$  são inicializados com valores aleatórios



*O critério de parada da rede será definido pela Energia Média do Erro Quadrado*

$$\varepsilon_{med} = \frac{1}{N} \sum_{n=1}^N \varepsilon(n) \quad \text{onde} \quad \varepsilon(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n)$$

*onde  $C$  inclui todos os sinais de entrada da rede*

*O critério de parada da rede é  $\theta$  e a taxa de treinamento é dada por  $\eta$*

*Cálculo dos Campos de Atração:*

$$A_{i\rho} = \frac{1}{N_\rho} \sum_{n=1}^N (x_i(n) * d_\rho(n)), \quad \forall i \quad \text{onde} \quad N_\rho = \sum_{n=1}^N d_\rho(n), \quad \forall \rho$$

*Definição das matrizes sinápticas associadas aos Objetos de Conhecimento:*

$${}_\rho W_{ji}(n) = W_{ji}^{(1)}(n), \quad \forall \rho$$

### ***Procedimento de propagação:***

*Cálculo das distâncias Euclidianas:*

$$d_\rho(n) = \sqrt{\sum_{i=1}^m [A_{i\rho} - x_i(n)]^2}, \quad \forall \rho$$

*Definição do índice da menor distância Euclidiana:*

$$\hat{\rho} = \operatorname{argmin} [d_\rho, \quad \forall \rho]$$

*Cálculo do **Campo Local Induzido** da primeira camada oculta:*

$$v_j^{(1)}(n) = b_j^{(1)} + \sum_{i=1}^m \hat{\rho} w_{ji}(n) x_i(n), \quad \forall j$$

*Resultado da primeira camada oculta:*

$$y_j(n) = \varphi^{(1)}[v_j^{(1)}(n)], \quad \forall j$$

*Cálculo do **Campo Local Induzido** da camada de saída:*

$$v_k^{(2)}(n) = b_k^{(2)} + \sum_{i=1}^j w_{ki}^{(2)}(n) y_i(n), \quad \forall k$$

*Resultado da camada de saída:*

$$o_k(n) = \varphi^{(2)}[v_k^{(2)}(n)], \quad \forall k$$

*Cálculo do erro:*

$$e_k(n) = d_k(n) - o_k(n), \quad \forall k$$

**Procedimento de retropropagação:**

*Cálculo do gradiente da camada de saída :*

$$\delta_k^{(2)}(n) = e_k(n) \varphi' [v_k^{(2)}(n)], \quad \forall k$$

*Cálculo da correção dos pesos e bias da camada de saída :*

$$\Delta W_{kj}^{(2)}(n) = \eta \delta_k^{(2)}(n) y_j(n), \quad \forall k \quad e \quad \Delta b_k^{(2)}(n) = \eta \delta_k^{(2)}(n), \quad \forall k$$

*Cálculo do gradiente da primeira camada oculta :*

$$\delta_j^{(1)}(n) = \varphi' [v_j^{(1)}(n)] \sum_k \delta_k^{(2)}(n) w_{kj}(n), \quad \forall j$$

*Cálculo da correção dos pesos e bias da primeira camada oculta :*

$$\Delta W_{ji}^{(1)}(n) = \eta \delta_j^{(1)}(n) x_i(n), \quad \forall j \quad e \quad \Delta b_j^{(1)}(n) = \eta \delta_j^{(1)}(n), \quad \forall j$$

*Ajuste dos pesos e bias :*

$$W_{kj}^{(2)}(n+1) = W_{kj}^{(2)}(n) + \Delta W_{kj}^{(2)}(n), \quad \forall k, \quad \forall j$$

$$\hat{\rho} W_{ji}^{(1)}(n+1) = \hat{\rho} W_{ji}^{(1)}(n) + \Delta W_{ji}^{(1)}(n), \quad \forall j, \quad \forall i$$

$$b_k^{(2)}(n+1) = b_k^{(2)}(n) + \Delta b_k^{(2)}(n), \quad \forall k$$

$$b_j^{(1)}(n+1) = b_j^{(1)}(n) + \Delta b_j^{(1)}(n), \quad \forall j$$

$$\text{Calcular } \varepsilon_{med} = \frac{1}{N} \sum_{n=1}^N \varepsilon(n) \quad e \quad \text{fazer } n = n + 1$$

*Repetir procedimento de **propagação e retropropagação** enquanto  $\varepsilon_{med} > \theta$*

**Algoritmo do aprendizado backpropagation alterado**

Através das alterações propostas neste capítulo, pode-se encontrar grande impacto computacional durante a etapa de treinamento da rede, melhorando o tempo do processo e a quantidade de iterações. O Capítulo 7 abordará alguns testes computacionais relacionados aos algoritmos da rede backpropagation comum e da rede alterada, assim como tabelas, com informações sobre os resultados encontrados e as devidas comparações.

## 7. Avaliação Empírica das Redes

Neste Capítulo os algoritmos observados nos Capítulos 5 e 6, serão utilizados em uma situação prática, na qual se busca encontrar termos empíricos para a avaliação de performance das redes. A rede neural artificial, com algoritmo de aprendizado supervisionado backpropagation, será comparada com sua versão alterada, com variações nos parâmetros e precisão do critério de parada apresentado na equação 5.10, no Capítulo 5.

### 7.1 Arquitetura e Característica das Redes

Ambas as redes serão testadas com a mesma arquitetura, sob a amostra de dados aleatória idêntica. Vale ressaltar que o objetivo principal deste Capítulo firma-se na obtenção de valores, com característica comprobatória, para os dois modelos apresentados. Desta forma, a formalização da arquitetura e o conjunto de dados necessitam ser os mesmos.

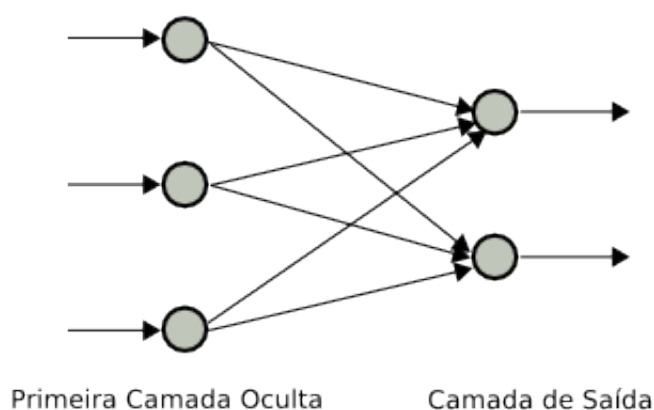


Figura 7.1 – Arquitetura da Rede.

Conforme visualizado na Figura 7.1, a arquitetura adotada pelas redes terá duas camadas: uma camada oculta, representando a primeira camada da rede, e uma camada de saída. A primeira camada oculta está em contato direto com os sinais de entrada (estímulos da rede). Essa camada será formada por três neurônios, cada um deles associado a uma função de ativação sigmóide. A

camada de saída é formada por dois neurônios. Trabalha-se com funções lineares nessa camada.

O vetor de entrada, para cada iteração, possui dimensão três – ou seja, em cada iteração, a rede recebe três valores de entrada. A rede apresenta dois valores de saída. Quando o primeiro neurônio de saída apresenta o valor 1, espera-se que o outro calcule o valor 0. Isso é definido pelo conjunto dos valores alvo, composto por um vetor de dois elementos, em iteração. No total, os dois vetores, de entrada e saída alvo, são compostos por 500 exemplos, apresentados no Apêndice I. Para cada exemplo, um único elemento do vetor alvo apresenta valor nulo – sendo o outro, o associado ao padrão utilizado (com valor 1). Na classificação de padrões, utiliza-se esse comportamento. Uma rede com dois elementos de saída tratará da classificação dos dados de entrada em dois padrões distintos. Os dados estarão enquadrados no primeiro padrão quando o primeiro elemento de saída assumir o valor 1 e o segundo elemento assumir o valor 0. De forma análoga, o segundo padrão será referenciado quando a rede apresentar 0 e 1 para o primeiro e segunda valor de saída, respectivamente.

A rede será treinada até que o critério de parada  $\varepsilon_{med}$  seja inferior ou igual à  $\theta$ . Neste ponto, espera-se que o valor real apresentado pela rede seja próximo ao valor alvo. O treinamento utilizado, em ambos os casos, será o backpropagation, exceto pelas alterações propostas no Capítulo 6, em uma das redes utilizadas.

## 7.2 Arquitetura Computacional

Para executar os experimentos foi utilizado o aplicativo MATLAB R2006a, da empresa The MathWorks. Em todos os casos testados, para as duas redes, foi utilizado um computador Dell, com processador Intel Dual Core de 1.66 Ghz, 1024 de memória principal. Foi utilizado o sistema operacional Microsoft Windows XP. Os testes foram preferencialmente realizados sob a mesma plataforma para deixar que as diferenças encontradas entre as redes sejam unicamente associadas à performance do algoritmo e não ao poder computacional empregado. O código fonte utilizado encontra-se no Apêndice II.

### 7.3 Parâmetros das Redes

Os pesos e bias da rede foram inicializados com valores aleatórios, variando entre -0.5 e 0.5. Para o algoritmo backpropagation alterado o valor inicial da matriz de pesos da primeira camada oculta foi propagada para as matrizes associadas aos Objetos de Conhecimento, conforme visualizado em detalhes na Figura 6.5.

As funções de ativação lineares existentes na camada de saída não trabalham com parâmetros específicos, por característica própria. Porém, a função sigmóide apresentada na camada de saída, trabalha com um termo característico: o parâmetro de inclinação  $\alpha$ . Em todos os testes realizados, o parâmetro de inclinação adotado será  $\alpha=0.8$ .

Foi executada uma seqüência de testes, onde a taxa de aprendizado  $\eta$  assumiu os valores 0.2, 0.3 e 0.4. A taxa de aprendizado, conforme já observado, proporciona o poder de aprendizado da rede, com mais ou menos intensidade. O critério de parada terá seu valor igual à  $10^{-7}$  ou  $10^{-8}$ . Estes valores foram adotados para compor os testes apresentados, porém sem nenhuma particularidade específica.

### 7.4 Análise dos Resultados

A Tabela 7.1, mostra os valores encontrados para os testes realizados com as redes. Utiliza-se aqui  $\theta=10^{-8}$ , como critério de parada, e a taxa de aprendizado  $\eta=0.4$ .

	Backpropagation	Backpropagation Alterada
<b>Critério de Parada <math>\theta</math></b>	0.00000001	0.00000001
<b>Parâmetro de Inclinação <math>\alpha</math></b>	0.8	0.8
<b>Taxa de Aprendizado <math>\eta</math></b>	0.4	0.4
<b>Número de Iterações</b>	26208	1757
<b>Tempo (Segundos/Minutos)</b>	94,0499 / 1:34	6,3798 / 0:06
<b>E<sub>med</sub> (Erro)</b>	0.00000000984724	0.00000000819718

Tabela 7.1 – Primeiro Exemplo

Observe que, em todos os parâmetros relacionados à otimização e tempo de execução, a rede alterada apresentou melhores resultados. Os testes efetuados, relativamente, foram resolvidos em poucas iterações. Para a rede alterada, por exemplo, com 1757 iterações, o grupo de dados foi apresentado, por completo, três vezes. O próximo exemplo tratará da mesma otimização (sob mesmo critério), porém com uma taxa de aprendizado suavemente menos impactante:  $\eta=0.3$ . Observe que, apenas por essa modificação, a rede apresentou uma performance consideravelmente lenta, se comparado com os outros testes efetuados.

	Backpropagation	Backpropagation Alterada
<b>Critério de Parada <math>\theta</math></b>	0.00000001	0.00000001
<b>Parâmetro de Inclinação <math>\alpha</math></b>	0.8	0.8
<b>Taxa de Aprendizado <math>\eta</math></b>	0.3	0.3
<b>Número de Iterações</b>	276207	48663
<b>Tempo (Segundos/Minutos)</b>	992,9161 / 16:33	174,6669 / 2:55
<b><math>E_{med}</math> (Erro)</b>	0.00000000998938	0.00000000997125

Tabela 7.2 – Segundo Exemplo

Pode-se observar uma melhoria considerável na performance da rede alterada, em relação à rede backpropagation comum. A rede alterada encontrou uma Energia do Erro Médio Quadrático inferior, em 48663 iterações, ao encontrado pela backpropagation, com 276207 iterações. O tempo de execução também foi otimizado.

Nos casos apresentados, aspirava-se encontrar  $\theta=10^{-8}$ . Os próximos exemplos a serem tratados, trabalham com um critério de parada mais suave, o que minimiza o tempo de aprendizado, tendo como custo, uma menor precisão. O terceiro exemplo, apresentado na Tabela 7.3, utiliza uma taxa de aprendizado menor, definida como  $\eta=0.2$ . Como houve folga em relação à precisão da rede, o número de iterações será, considerando o valor de  $\eta$ , menor.

Com os valores encontrados neste último exemplo, pode-se destacar que o nível da taxa de aprendizado não é diretamente proporcional à precisão da minimização do erro. Se comparados os valores de  $\varepsilon_{med}$  das Tabelas 7.1 e 7.3,

observa-se que, com o parâmetro  $\eta=0.4$ , a rede apresentou melhor comportamento, inclusive em performance. Em ambos os casos, a alteração sugerida neste estudo obteve melhorias interessantes, do ponto de vista da otimização.

	Backpropagation	Backpropagation Alterada
<b>Critério de Parada <math>\theta</math></b>	0.0000001	0.0000001
<b>Parâmetro de Inclinação <math>\alpha</math></b>	0.8	0.8
<b>Taxa de Aprendizado <math>\eta</math></b>	0.2	0.2
<b>Número de Iterações</b>	177816	77605
<b>Tempo (Segundos/Minutos)</b>	638,1226 / 10:38	278,5681 / 4:39
<b>E<sub>med</sub> (Erro)</b>	0.00000009972402	0.00000009946938

Tabela 7.3 – Terceiro Exemplo

O último dos exemplos (exemplo 4) tratará do treinamento com critério de parada idêntico ao anterior, ou seja,  $\theta=10^{-7}$ , porém com a taxa de aprendizado definida como  $\eta=0.3$  - um valor intermediário, entre os testes realizados acima. Esta alteração proporcionou o encontro de um ponto de convergência de forma mais rápida do que o modelo apresentado na Tabela 7.3, devido ao maior ajuste atribuído aos pesos. Em cada iteração, os pesos sinápticos são ajustados por  $w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n)$ , onde  $\eta$  afeta o comportamento da regra. Em comparação ao exemplo anterior, onde  $\eta=0.2$ , este experimento oferece um ajuste de maior consideração à rede.

	Backpropagation	Backpropagation Alterada
<b>Critério de Parada <math>\theta</math></b>	0.0000001	0.0000001
<b>Parâmetro de Inclinação <math>\alpha</math></b>	0.8	0.8
<b>Taxa de Aprendizado <math>\eta</math></b>	0.3	0.3
<b>Número de Iterações</b>	27705	5250
<b>Tempo (Segundos/Minutos)</b>	99,5809 / 1:40	18,9312 / 0:19
<b>E<sub>med</sub> (Erro)</b>	0.00000009959973	0.00000009239615

Tabela 7.4 – Quarto Exemplo

O exemplo apresentado na Tabela 7.4 será complementado, por conveniência da comparação entre os modelos tratados, com gráficos apontando a minimização do erro no ajuste dos pesos sinápticos. A Figura 7.2, à seguir, mostra

a otimização associada ao algoritmo backpropagation comum.

Observe, conforme apresentado no percurso da linha de ajuste dos pesos, que a otimização obteve declínio mais acentuado no início do processo – anterior às 5000 iterações. A partir disso, o processo apresentou uma minimização mais suave. Ao final das 27705 iterações, o critério de parada foi superado.

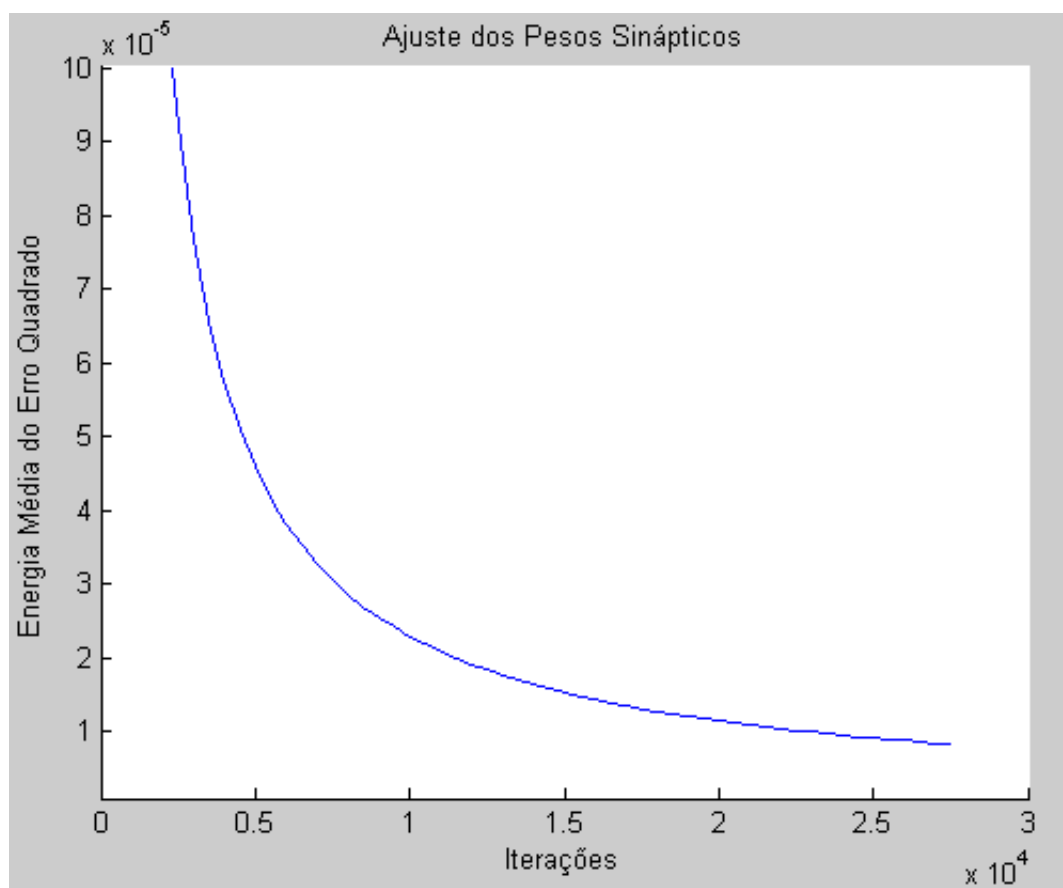


Figura 7.2 – Ajuste dos Pesos da Rede Backpropagation

Note, porém, que o ajuste dos pesos apresentado pela rede alterada, obteve uma otimização altamente acentuada, alcançando o critério de parada com 5250 iterações. Ao observar as Figuras 7.2 e 7.3, fica claro que as alterações propostas no Capítulo 6, oferecem uma melhoria de performance considerável para a rede backpropagation. Nos quatro exemplos apresentados, a rede alterada obteve um nível melhor de minimização, com um número menor de iterações, processadas num período de tempo inferior ao da rede comum.



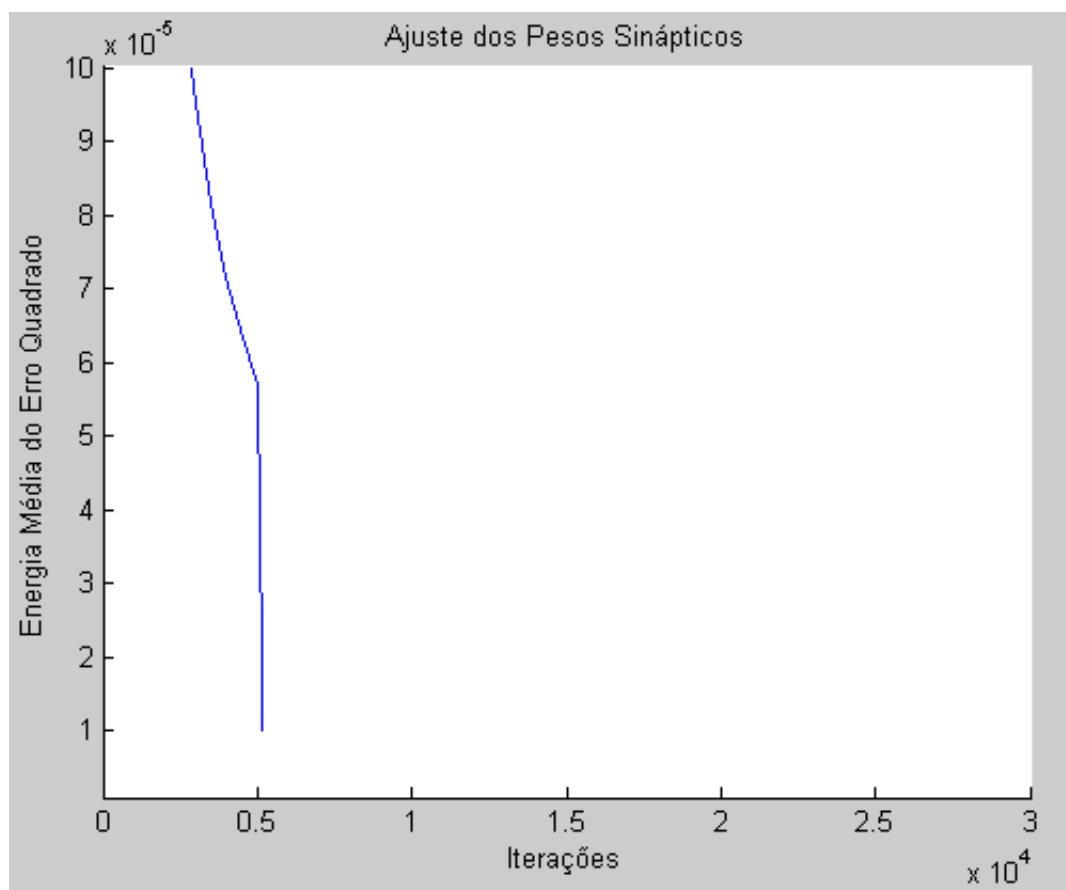


Figura 7.3 – Ajuste dos Pesos da Rede Backpropagation Alterada

As perspectivas exibidas nas Figuras 7.2 e 7.3 mostram com clareza o diferencial dos algoritmos se tratando do número de iterações necessárias para encontrar um erro satisfatório.

## 8. Conclusões

Do ponto de vista prático, todo o comportamento alcançado pela rede neural, com o treinamento repetitivo, é aprimorado, no sentido de favorecer o aprendizado. A análise biológica favorece o conceito estabelecido, que indica particularidades existentes no processo de aprendizado dos seres vivos. Toda a alteração sináptica decorrente da assimilação dos estímulos já apresentados ao corpo neural será reforçada com certo nivelamento, ajustando o valor representativo dos elementos químicos existentes nas ligações sinápticas. O nível da reaprendizagem afetará fortemente as conexões previamente contempladas com os ajustes químicos iniciais. Ou seja, o processo induzido na reapresentação do mesmo estímulo proporciona a capacidade de maior assimilação do mesmo. A forma biológica do processo trabalha com esta perspectiva.

O estudo realizado neste trabalho contribuiu para que o comportamento artificial utilize questões biológicas mais recentes, introduzindo os conceitos aos modelos existentes. Nos experimentos realizados, não foram encontradas limitações em relação à utilização do algoritmo. Em essência, o modelo trabalha com a classificação de padrões dos dados, onde o treinamento supervisionado é requerido. Esse fato é confirmado pela associação dos valores alvo com os Objetos de Conhecimento, tratados no Capítulo 6. A rede não foi elaborada para tratar situações onde não exista um conjunto de dados alvo, ou seja, não é capaz de trabalhar com o treinamento não-supervisionado. Outras limitações não foram encontradas.

O escopo apresentado, sob a modelagem neural do treinamento de retropropagação, indicou que o aprendizado deveria ser tratado distintamente, quando associado ao trabalho de padronização. O cérebro biológico favorece estas preliminares. O trabalho apresentado tende a direcionar o estudo dos modelos neurais artificiais para este mesmo percurso. Assim, a dissertação centrou-se na concepção de análises empíricas, realizadas com intuito de comprovar perspectivas computacionais e de otimização associadas às alterações propostas. Todavia, cria-se a abertura para a implementação dos novos modelos em projetos futuros, intermitentes ou complementares a este, ou ainda, a utilização das

importâncias observadas, em estudos e casos reais.

As principais vantagens fornecidas pela alteração do algoritmo backpropagation, em relação ao seu modelo convencional, estão centradas na performance. Conforme observado no Capítulo 7, mais especificamente nos quatro experimentos realizados, a rede alterada teve grande melhoria em relação ao tempo de execução e o número de iterações. O quarto experimento, com resultados mais expressivos, teve uma diferença aproximada de 20.000 iterações.

Para experimentos futuros, poderia ser considerada a possibilidade de aplicação do modelo alterado em situações reais, com valores fornecidos em bases de benchmark ou em estudos específicos – associados à academia ou ao meio empresarial. Deve-se também observar a possibilidade de aplicação da lógica fuzzy na definição das fronteiras entre os Campos de Atração. Essencialmente, a abertura para novos projetos firma-se na utilização de um contexto supervisionado de treinamento. Contudo, pode-se considerar a elaboração de redes de treinamento não-supervisionado a partir dos conhecimentos biológicos apresentados, criando assim, novos modelos neurais artificiais.

Pode-se concluir que, a estimativa do reaprendizado, quando atribuído ao escopo artificial, favorece o comportamento dos modelos. Nas práticas empíricas, o comportamento da rede teve melhorias consideráveis, atribuindo ênfase à análise teórica realizada. Isso significa que se obteve melhoria de performance com a aproximação do modelo neural ao comportamento biológico.

## 9. Referência Bibliográfica

- Amari, S.I. (1971) *Characteristics of randomly connected threshold-element networks and network systems*. Proceedings of the IEEE. NC State University, Raleigh, USA.
- Anderson, C.W. (1986) *Learning and Problem Solving with Multilayer Connectionist Systems*. Dissertation. University of Massachusetts.
- Anderson, J.A. (1995) *An Introduction to Neural Networks*. MIT Press.
- Arbib, M.A. (2003) *The Handbook of Brain Theory and Neural Networks*. The MIT Press. Cambridge, Massachusetts, London, England.
- Arone, E.M. Philippi, M.L.S. Vono, Z.E. (1994) *Enfermagem médico-cirúrgica aplicada ao sistema nervoso*. Senac. São Paulo, SP.
- Ashby, W.R. (1952) *Design for a Brain*. New York, John Wiley.
- Ballou, R.H. (1993) *Logística Empresarial: transporte, administração de materiais e distribuição física*. Atlas.
- Barbosa, C.D. (2005) *Uma aplicação de Redes Neurais Artificiais à classificação dos municípios do Estado do Rio de Janeiro segundo indicadores sociais*. Universidade Estadual do Norte Fluminense Darcy Ribeiro, UENF. Campos dos Goytacazes, RJ.
- Barreto, J.M. (2002) *Introdução às Redes Neurais Artificiais*. Laboratório de Conexionismo e Ciências Cognitivas. UFSC. Florianópolis, SC.
- Bishop, C.M. (1995) *Neural Networks for Pattern Recognition*. Oxford University Press.
- Boker, S.M. (1996) *Linear and Nonlinear Dynamical Systems Data Analytic Techniques and an Application to Developmental Data*. Dissertation. University of Virginia.
- Boothe, R.G. (2002) *Perception of the Visual Environment*. Springer.
- Braga, A.P. Carvalho, A.C.P.L.F. Ludermir, T.B. (2000) *Redes Neurais Artificiais: Teoria e aplicações*. LTC. Rio de Janeiro, RJ.
- Braitenberg, V. (1984) *Vehicles: Experiments in Synthetic Psychology*. The MIT Press; Reprint edition.
- Carvalho Filho, S. (2004) *Estatística básica para concursos: teoria e 150 questões*. Impetus. Niterói, RJ.
- Chiavenato, I. (2004) *Recursos humanos: o capital humano das organizações*. 8. ed. Atlas. São Paulo, SP.
- Cordeschi, R. (2002) *The Discovery of the Artificial: Behavior, Mind, and Machines*

*Before and Beyond Cybernetics*. Springer.

Coudry, M.I.H. Freire, F.M.P. (2005) *O Pequeno Grande W*. Universidade Estadual de Campinas. Campinas, SP.

Dawson, M.R. (1998) *Understanding Cognitive Science*. Blackwell Publishing.

Demuth, H. Beale, M. Hagan, M. (2007) *Neural Network Toolbox 5: User's Guide*. The MathWorks, Inc. Natick, MA.

Fausett, L. (1994) *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice Hall.

Forcada, M.L. (2002) *Neural Networks: Automata and Formal Models of Computation*. Dept. Llenguatges i Sistemes Informàtics, Universitat d'Alacant. Spain.

Freeman, J.A. Skapura, D.M. (1991) *Neural Networks: algorithms, applications, and programming techniques*. Addison-Wesley Publishing Company.

Golden, R.M. (1996) *Mathematical Methods for Neural Network Analysis and Design*. MIT Press.

Gupta, M.M. Homma, N. Jin, L. (2003) *Static and Dynamic Neural Networks: from fundamentals to advanced theory*. Wiley-IEEE.

Haykin, S. (2001) *Redes Neurais: princípios e práticas*. 2. ed. Bookman. Porto Alegre, RS.

Hebb, D.O. (1949) *The Organization of Behavior: A Neuropsychological Theory*. New York, Wiley.

Held, J. Bautista, J. Koehl, S. (2006) *From a Few Cores to Many: A Tera-scale Computing Research Overview*. Intel Corporation. Santa Clara, CA.

Hinton, G.E. Sejnowski, T.J. (1999) *Unsupervised Learning: Foundations of Neural Computation*. MIT Press.

Höffgen, K.U. Simon, H.U. Horn, K.S.V. (1995) *Robust Trainability of Single Neurons*. Journal of Computer and System Sciences. Academic Press.

Howlett, R.J. Jain, L.C. (2001) *Radial Basis Function Networks 2: New Advances in Design*. Springer.

Intrator, N. (2003) *Information Theory and Visual Plasticity*. The Handbook of Brain Theory and Neural Networks. The MIT Press. Cambridge, Massachusetts, London, England.

Johansen, R. Swigart, R. (1994) *Depois da Reengenharia: Motivando e Valorizando o Profissional na Empresa Redimensionada*. Objetiva. Rio de Janeiro, RJ.

Klein, R.M. (1999) *The legacy Hebb*. Canadian Journal of Experimental Psychology. Ottawa, Ontario.

- Kohonen, T. (1982) *Self-organized formation of topologically correct feature maps*. Department of Technical Physics, Helsinki University of Technology, Espoo, Finland.
- Lawson, M.V. (2004) *Finite automata*. Department of Mathematics. School of Mathematical and Computer Sciences, Heriot-Watt University. Scotland.
- Lingireddy, S. Brion, G.M. (2005) *Artificial Neural Networks in Water Supply Engineering*. ASCE Publications.
- Linsker, R. (1988) *Self-organization in a perceptual network*. Computer Magazine.
- López Piñero, J.M. (2006) *Santiago Ramon y Cajal*. Publ. Universitat de Valencia. Valencia, ES.
- Maccorduck, P. (2004) *Machines Who Think: : a Personal Inquiry Into the History and Prospects of Artificial Intelligence*. AK Peters, Ltd.
- Malsburg, von der C. (1973) *Self-organization of orientation-sensitive cells in the striate cortex*. Kybernetik, Biological Cybernetics. New York.
- Mangasarian, O.L. (1993) *Mathematical Programming in Neural Networks*. ORSA. Journal on Computing.
- Minsky, M.L. (1954) *Theory of neural-analog reinforcement systems and its application to the brain-model problem*. PhD Thesis. Princeton University.
- Minsky, M.L. (1961) *Steps Toward Artificial Intelligence*. Research Lab. of Electronics. Massachusetts Institute of Technology, MIT.
- Mohan, C.K. Mehrotra, K. Ranka, S. (1997) *Elements Of Elements of Artificial Neural Networks*. MIT Press.
- Nguyen, D. Widrow, B. (1990) *Improving the Learning Speed of Two-Layer Neural Networks by Choosing Initial Values of the Adaptive Weights*. International Journal Conference on Neural Network. San Diego, CA.
- Paula, M.B. (2000) *Reconhecimento de Palavras Faladas usando Redes Neurais Artificiais*. Universidade Federal de Pelotas. UFPEL. Pelotas, RS.
- Pfeifer, R. Scheier, C. (1999) *Understanding Intelligence*. The MIT Press.
- Pinheiro, G.R.V. (1996) *Redes Neurais Aplicadas na Inferência de Propriedades de Derivados de Petróleo*. Pontifícia Universidade Católica do Rio de Janeiro, PUC-RJ. Rio de Janeiro, RJ.
- Poole, D. Salerno, M. (2004) *Álgebra Linear*. Thomson Learning Ibero.
- Portugal, M.S. Fernandes, L.G.L. (1995) *Redes Neurais Artificiais e Previsão de Séries Econômicas: Uma Introdução*. UFRGS. Porto Alegre, RS.

- Póvoa, C.L.R. (2005) *Heurística Grasp para o problema de Roteamento de Veículos com Multi-compartimentos e sua integração com o Sistema de Informação Geográfica Geo-Rota*. UENF. Campos dos Goytacazes, RJ.
- Prado, M.M.B. (2004) *Estudo sobre a parestesia do nervo alveolar inferior pós cirurgias de terceiros molares inferiores*. Dissertação de Mestrado. USP. São Paulo, SP.
- Ramalho, J.P. Issa, F.M.R. Pereira, M.C. Brandi, S.D. Garcia, C. Junior, F.K. (2003) *Uma Proposta para Comparação de Performance entre Redes Neurais Artificiais*. Universidade Católica Dom Bosco, UCDB - Campo Grande, MS.
- Ramón y Cajal, S. (1906) *The structure and connexions of neurons*. Nobel Lecture. Amsterdam, Netherlands.
- Randlov, R. (2001) *Solving Complex Problems with Reinforcement Learning*. Ph.D. Thesis. University of Copenhagen.
- Rochester, N. Holland, J. Haibt, L. Duda, W. (1956) *Tests on a cell assembly theory of the action of the brain, using a large digital computer*. IEEE Transactions on Information Theory. vol. 2.
- Rosenblatt, M. (1956) *Remarks on some nonparametric estimates of a density function*. Annals of Mathematical Statistics.
- Rosenblatt, M. (1958) *The Perceptron: A probabilistic model for information storage and organization in the brain*. Psychological Review.
- Rosler, R. Quevedo, J. (1998) *Neuroquímica da Memória*. Universidade Federal do Rio Grande do Sul, UFRGS. Porto Alegre, RS.
- Rumelhart, D.E. G.E. Hinton, R.J. Williams. (1986) Learning representations of back-propagation errors. Nature, London.
- Santos, D.L. Milano, M.E. Rosat, R. (1998) *Exercício Físico e Memória*. Universidade Federal do Rio Grande do Sul, UFRGS. Porto Alegre, RS.
- Schildt, H. (1989) *Inteligência Artificial Utilizando a Linguagem C*. ed. MacGraw-Hill. São Paulo, SP.
- Sejnowski, T.J. (1977) *Storing covariance with nonlinearly interacting neurons*. Journal of Mathematical Biology. New York.
- Shannon, C.E. (1948) *A Mathematical Theory of Communication*. The Bell System Technical Journal. vol. 27.
- Soto, C.P. (1999) *Redes Neurais Temporais para o Tratamento de Sistemas Variantes no Tempo*. PUC-RJ. Rio de Janeiro, RJ.
- Tanizak, H. (1996) *Nonlinear Filters: Estimation and Applications*. Springer.
- Timoszczuk, A.P. (2004) *Reconhecimento Automático do Locutor com Redes Neurais*

*Pulsadas*. Universidade de São Paulo, USP. Tese de Doutorado. São Paulo.

Uttley, A.M. (1956) *A Theory of the Mechanism of Learning Based on the Computation of Conditional Probabilities*. Proceedings of the International Conference on Cybernetics. Paris. Gauthier-Vellair.

Veelenturf, L.P.J. (1995) *Analysis and Applications of artificial neural networks*. Prentice Hall.

Vidyamurthy, G. (2004) *Pairs Trading: Quantitative Methods and Analysis*. Wiley Finance.

Wang, L. (2005) *Support Vector Machines: Theory and Applications*. Springer.



## Apêndice I – Dados utilizados pelas Redes

As colunas de dados apresentadas neste apêndice exibem respectivamente os valores de entrada e os valores alvo da rede. Cada linha está associada a uma iteração da rede. Cada coluna representa um elemento do vetor. O vetor de entrada possui três elementos e o vetor alvo é formado por dois elementos.

### Valores de entrada da rede:

### Valores alvo da rede:

2.3846	1.5050	0.5980	0	1
2.3672	1.5676	0.5998	0	1
2.3681	1.4759	0.5917	0	1
2.0503	0.8291	0.7043	1	0
2.3305	1.4379	0.5981	0	1
2.3682	1.4606	0.5946	0	1
2.3151	1.5396	0.5962	0	1
2.0860	0.8146	0.7059	1	0
2.0497	0.8100	0.7082	1	0
2.3645	1.5636	0.5934	0	1
2.3342	1.4579	0.5966	0	1
2.3534	1.5454	0.5969	0	1
2.3838	1.5136	0.5963	0	1
2.3703	1.5093	0.5956	0	1
2.3695	1.5243	0.5921	0	1
2.3957	1.5045	0.5912	0	1
2.0173	0.8020	0.7027	1	0
2.0252	0.8124	0.7074	1	0
2.0137	0.8988	0.7089	1	0
2.3199	1.4597	0.5934	0	1
2.0284	0.8531	0.7006	1	0
2.0988	0.8417	0.7042	1	0
2.0516	0.8666	0.7043	1	0
2.3226	1.5160	0.5924	0	1
2.3530	1.5281	0.5979	0	1
2.0380	0.8217	0.7068	1	0
2.3461	1.5136	0.5921	0	1
2.0059	0.8397	0.7005	1	0
2.0415	0.8695	0.7087	1	0
2.0015	0.8232	0.7097	1	0
2.0990	0.8211	0.7044	1	0
2.0498	0.8786	0.7064	1	0
2.0320	0.8040	0.7073	1	0
2.3412	1.5489	0.5973	0	1
2.3440	1.5867	0.5932	0	1
2.0213	0.8161	0.7063	1	0

2.0134	0.8793	0.7061	1	0
2.3630	1.4741	0.5942	0	1
2.0451	0.8956	0.7003	1	0
2.0313	0.8987	0.7038	1	0
2.3683	1.4186	0.5996	0	1
2.0612	0.8391	0.7002	1	0
2.0016	0.8810	0.7059	1	0
2.3058	1.4735	0.5937	0	1
2.0718	0.8307	0.7008	1	0
2.0454	0.8558	0.7035	1	0
2.0154	0.8324	0.7070	1	0
2.0728	0.8522	0.7055	1	0
2.3121	1.4902	0.5928	0	1
2.3893	1.4546	0.5975	0	1
2.3866	1.4465	0.5920	0	1
2.0908	0.8768	0.7024	1	0
2.0050	0.8922	0.7064	1	0
2.0191	0.8156	0.7017	1	0
2.0171	0.8006	0.7044	1	0
2.3340	1.4628	0.5963	0	1
2.3393	1.5183	0.5988	0	1
2.0038	0.8541	0.7087	1	0
2.3934	1.4529	0.5984	0	1
2.0873	0.8762	0.7065	1	0
2.3967	1.5330	0.5913	0	1
2.3010	1.4274	0.5918	0	1
2.3430	1.5781	0.5927	0	1
2.0687	0.8654	0.7017	1	0
2.0156	0.8809	0.7042	1	0
2.3856	1.4980	0.5918	0	1
2.0461	0.8543	0.7045	1	0
2.0412	0.8098	0.7001	1	0
2.0297	0.8951	0.7069	1	0
2.3650	1.5966	0.5945	0	1
2.3400	1.4398	0.5937	0	1
2.0733	0.8624	0.7001	1	0
2.0420	0.8246	0.7079	1	0
2.3920	1.5689	0.5963	0	1
2.0621	0.8269	0.7019	1	0
2.3905	1.5138	0.5937	0	1
2.3234	1.5098	0.5907	0	1
2.0335	0.8344	0.7039	1	0
2.3627	1.5398	0.5960	0	1
2.0414	0.8345	0.7084	1	0
2.3372	1.4851	0.5941	0	1
2.3566	1.5433	0.5949	0	1
2.0776	0.8511	0.7019	1	0
2.0701	0.8017	0.7081	1	0
2.0704	0.8515	0.7011	1	0
2.3665	1.4731	0.5986	0	1
2.0567	0.8177	0.7067	1	0
2.3999	1.5923	0.5994	0	1

2.0360	0.8451	0.7026	1	0
2.3597	1.4099	0.5943	0	1
2.3701	1.5925	0.5925	0	1
2.3740	1.4864	0.5937	0	1
2.3803	1.4168	0.5905	0	1
2.0916	0.8398	0.7025	1	0
2.3873	1.5027	0.5927	0	1
2.0422	0.8039	0.7007	1	0
2.0553	0.8708	0.7086	1	0
2.3336	1.5360	0.5995	0	1
2.0357	0.8502	0.7043	1	0
2.0562	0.8383	0.7011	1	0
2.0898	0.8245	0.7079	1	0
2.0815	0.8330	0.7020	1	0
2.0273	0.8374	0.7054	1	0
2.3060	1.4178	0.5973	0	1
2.3409	1.4948	0.5909	0	1
2.0596	0.8671	0.7048	1	0
2.3597	1.4323	0.5917	0	1
2.0956	0.8404	0.7003	1	0
2.3812	1.5220	0.5930	0	1
2.3092	1.4850	0.5962	0	1
2.3166	1.5666	0.5916	0	1
2.0452	0.8043	0.7015	1	0
2.3870	1.5539	0.5956	0	1
2.0621	0.8048	0.7064	1	0
2.0247	0.8647	0.7019	1	0
2.3491	1.4819	0.5954	0	1
2.0611	0.8929	0.7031	1	0
2.3608	1.4350	0.5938	0	1
2.0246	0.8413	0.7051	1	0
2.3465	1.5083	0.5906	0	1
2.0342	0.8598	0.7031	1	0
2.3412	1.4572	0.5961	0	1
2.0503	0.8278	0.7031	1	0
2.3112	1.4887	0.5953	0	1
2.3015	1.5328	0.5928	0	1
2.0282	0.8738	0.7071	1	0
2.3784	1.5972	0.5953	0	1
2.0903	0.8549	0.7080	1	0
2.3829	1.4333	0.5961	0	1
2.3521	1.5436	0.5943	0	1
2.3461	1.4891	0.5991	0	1
2.3443	1.4733	0.5970	0	1
2.3852	1.5519	0.5905	0	1
2.3558	1.4028	0.5940	0	1
2.0816	0.8023	0.7022	1	0
2.0704	0.8478	0.7093	1	0
2.0713	0.8772	0.7045	1	0
2.3172	1.5938	0.5964	0	1
2.3049	1.5511	0.5911	0	1
2.0286	0.8749	0.7093	1	0

2.0131	0.8059	0.7070	1	0
2.3848	1.4419	0.5954	0	1
2.0081	0.8149	0.7056	1	0
2.3319	1.4750	0.5913	0	1
2.0372	0.8926	0.7020	1	0
2.0049	0.8433	0.7012	1	0
2.3522	1.4234	0.5923	0	1
2.3375	1.5647	0.5995	0	1
2.0598	0.8051	0.7029	1	0
2.0889	0.8898	0.7007	1	0
2.3234	1.5866	0.5994	0	1
2.3264	1.5999	0.5979	0	1
2.3498	1.4581	0.5933	0	1
2.3958	1.5533	0.5933	0	1
2.0131	0.8905	0.7001	1	0
2.0288	0.8183	0.7099	1	0
2.0017	0.8181	0.7062	1	0
2.3560	1.4488	0.5918	0	1
2.3263	1.5507	0.5934	0	1
2.3214	1.5204	0.5940	0	1
2.3660	1.4367	0.5936	0	1
2.0170	0.8460	0.7062	1	0
2.3686	1.5355	0.5912	0	1
2.3013	1.4621	0.5922	0	1
2.0307	0.8073	0.7068	1	0
2.3074	1.4141	0.5999	0	1
2.3227	1.5033	0.5954	0	1
2.3703	1.5165	0.5949	0	1
2.3074	1.4386	0.5962	0	1
2.3276	1.5542	0.5969	0	1
2.3638	1.5973	0.5950	0	1
2.0948	0.8172	0.7092	1	0
2.0113	0.8188	0.7091	1	0
2.3156	1.4244	0.5924	0	1
2.0722	0.8348	0.7075	1	0
2.0663	0.8117	0.7027	1	0
2.3419	1.4426	0.5996	0	1
2.0081	0.8149	0.7034	1	0
2.3466	1.5828	0.5977	0	1
2.3862	1.5313	0.5911	0	1
2.0488	0.8007	0.7037	1	0
2.3531	1.4363	0.5950	0	1
2.3422	1.5321	0.5933	0	1
2.0957	0.8808	0.7011	1	0
2.0565	0.8031	0.7002	1	0
2.3870	1.4054	0.5948	0	1
2.0192	0.8284	0.7025	1	0
2.3934	1.4274	0.5948	0	1
2.3895	1.5885	0.5966	0	1
2.0437	0.8529	0.7015	1	0
2.0136	0.8468	0.7073	1	0
2.3399	1.4717	0.5971	0	1

2.0869	0.8374	0.7024	1	0
2.0978	0.8359	0.7023	1	0
2.3681	1.5332	0.5987	0	1
2.0022	0.8738	0.7012	1	0
2.0069	0.8147	0.7018	1	0
2.3032	1.5468	0.5946	0	1
2.3276	1.4737	0.5999	0	1
2.3889	1.5732	0.5975	0	1
2.0569	0.8841	0.7059	1	0
2.0331	0.8341	0.7086	1	0
2.0568	0.8020	0.7079	1	0
2.3153	1.5666	0.5981	0	1
2.0639	0.8331	0.7077	1	0
2.0380	0.8558	0.7048	1	0
2.0608	0.8824	0.7000	1	0
2.3790	1.5027	0.5979	0	1
2.3103	1.4315	0.5959	0	1
2.0408	0.8947	0.7094	1	0
2.3150	1.4769	0.5969	0	1
2.0169	0.8103	0.7032	1	0
2.0734	0.8589	0.7040	1	0
2.3506	1.4339	0.5948	0	1
2.0641	0.8984	0.7084	1	0
2.3803	1.5396	0.5954	0	1
2.3083	1.5641	0.5981	0	1
2.0445	0.8987	0.7031	1	0
2.3875	1.5671	0.5967	0	1
2.3881	1.4959	0.5944	0	1
2.0616	0.8338	0.7062	1	0
2.0685	0.8490	0.7071	1	0
2.3515	1.5212	0.5903	0	1
2.3822	1.4636	0.5941	0	1
2.0130	0.8746	0.7080	1	0
2.3668	1.4027	0.5944	0	1
2.3455	1.5810	0.5972	0	1
2.0065	0.8523	0.7098	1	0
2.3922	1.5122	0.5935	0	1
2.3773	1.4212	0.6000	0	1
2.3542	1.4014	0.5955	0	1
2.0196	0.8213	0.7062	1	0
2.3016	1.5782	0.5924	0	1
2.0907	0.8241	0.7038	1	0
2.3331	1.5008	0.5944	0	1
2.0767	0.8220	0.7048	1	0
2.0802	0.8529	0.7020	1	0
2.0580	0.8333	0.7068	1	0
2.3943	1.5540	0.5926	0	1
2.0866	0.8009	0.7050	1	0
2.3629	1.5585	0.5955	0	1
2.3524	1.4343	0.5987	0	1
2.0219	0.8895	0.7014	1	0
2.0457	0.8212	0.7028	1	0

2.3225	1.5818	0.5999	0	1
2.0589	0.8458	0.7065	1	0
2.0313	0.8769	0.7042	1	0
2.0299	0.8328	0.7094	1	0
2.0343	0.8437	0.7012	1	0
2.0169	0.8721	0.7056	1	0
2.3486	1.5904	0.5977	0	1
2.3479	1.5053	0.5921	0	1
2.3193	1.5819	0.5908	0	1
2.3013	1.5535	0.5905	0	1
2.3813	1.5848	0.5980	0	1
2.3674	1.5854	0.5966	0	1
2.3594	1.5231	0.6000	0	1
2.3982	1.5799	0.5931	0	1
2.3440	1.5402	0.5939	0	1
2.0300	0.8144	0.7011	1	0
2.0292	0.8903	0.7040	1	0
2.3333	1.5888	0.5916	0	1
2.0258	0.8957	0.7001	1	0
2.0574	0.8256	0.7081	1	0
2.3638	1.4503	0.5986	0	1
2.3652	1.5892	0.5918	0	1
2.3930	1.4620	0.5973	0	1
2.3536	1.4327	0.5979	0	1
2.0217	0.8348	0.7005	1	0
2.0229	0.8333	0.7031	1	0
2.3307	1.5441	0.5905	0	1
2.0131	0.8932	0.7013	1	0
2.3166	1.5823	0.5986	0	1
2.0617	0.8731	0.7022	1	0
2.0713	0.8451	0.7094	1	0
2.0330	0.8296	0.7094	1	0
2.0582	0.8120	0.7075	1	0
2.0380	0.8274	0.7016	1	0
2.3956	1.4392	0.5922	0	1
2.3613	1.4325	0.5997	0	1
2.3289	1.5942	0.5905	0	1
2.3228	1.5917	0.5932	0	1
2.3055	1.5200	0.5961	0	1
2.0215	0.8818	0.7008	1	0
2.3007	1.5578	0.5998	0	1
2.3878	1.4705	0.5928	0	1
2.3968	1.4311	0.5984	0	1
2.0313	0.8971	0.7036	1	0
2.3027	1.5587	0.5900	0	1
2.0110	0.8377	0.7013	1	0
2.3310	1.4270	0.5978	0	1
2.3397	1.4270	0.5976	0	1
2.3928	1.4782	0.5949	0	1
2.0093	0.8978	0.7016	1	0
2.0845	0.8121	0.7019	1	0
2.3991	1.5424	0.5913	0	1

2.3480	1.4992	0.5971	0	1
2.0061	0.8738	0.7019	1	0
2.3917	1.4247	0.5999	0	1
2.3370	1.5397	0.5911	0	1
2.0594	0.8843	0.7032	1	0
2.0233	0.8992	0.7040	1	0
2.0650	0.8915	0.7077	1	0
2.0970	0.8285	0.7078	1	0
2.0238	0.8804	0.7026	1	0
2.0714	0.8022	0.7064	1	0
2.3546	1.5696	0.5920	0	1
2.3668	1.5342	0.5918	0	1
2.3970	1.4974	0.5918	0	1
2.0642	0.8694	0.7066	1	0
2.0358	0.8062	0.7049	1	0
2.3091	1.5348	0.5949	0	1
2.3222	1.5450	0.5993	0	1
2.0964	0.8792	0.7016	1	0
2.3638	1.4000	0.5966	0	1
2.3275	1.4089	0.5991	0	1
2.0410	0.8183	0.7087	1	0
2.3023	1.5454	0.5915	0	1
2.3729	1.5910	0.5934	0	1
2.0742	0.8655	0.7088	1	0
2.3347	1.4119	0.5928	0	1
2.3958	1.4314	0.5958	0	1
2.3094	1.4900	0.5913	0	1
2.0392	0.8747	0.7035	1	0
2.3743	1.5302	0.5906	0	1
2.0833	0.8530	0.7063	1	0
2.3058	1.5084	0.5954	0	1
2.0863	0.8145	0.7047	1	0
2.0787	0.8344	0.7000	1	0
2.0131	0.8505	0.7004	1	0
2.3227	1.4656	0.5910	0	1
2.3314	1.4503	0.5957	0	1
2.3842	1.4369	0.5949	0	1
2.3452	1.4651	0.5962	0	1
2.3886	1.5523	0.5912	0	1
2.0457	0.8201	0.7013	1	0
2.0065	0.8625	0.7037	1	0
2.3484	1.5939	0.5966	0	1
2.0253	0.8415	0.7052	1	0
2.3163	1.4973	0.5950	0	1
2.0843	0.8194	0.7086	1	0
2.3610	1.5131	0.5939	0	1
2.3103	1.4317	0.5959	0	1
2.0560	0.8731	0.7078	1	0
2.0388	0.8969	0.7059	1	0
2.0559	0.8799	0.7009	1	0
2.3933	1.4519	0.5980	0	1
2.3049	1.5212	0.5945	0	1

2.3096	1.5274	0.5956	0	1
2.0066	0.8626	0.7025	1	0
2.3925	1.5259	0.5912	0	1
2.0642	0.8202	0.7044	1	0
2.3981	1.4192	0.5947	0	1
2.3546	1.4569	0.5963	0	1
2.0065	0.8455	0.7084	1	0
2.0145	0.8828	0.7007	1	0
2.3824	1.4268	0.5912	0	1
2.3515	1.5927	0.5988	0	1
2.0048	0.8620	0.7041	1	0
2.3401	1.4842	0.5962	0	1
2.3907	1.5340	0.5904	0	1
2.3163	1.5497	0.5963	0	1
2.0454	0.8961	0.7056	1	0
2.3372	1.5586	0.5920	0	1
2.3383	1.4506	0.5966	0	1
2.0968	0.8520	0.7037	1	0
2.3765	1.4754	0.5910	0	1
2.0183	0.8632	0.7092	1	0
2.0516	0.8910	0.7074	1	0
2.3005	1.5206	0.5904	0	1
2.3397	1.5463	0.5932	0	1
2.3979	1.4408	0.5941	0	1
2.3952	1.4521	0.5949	0	1
2.3636	1.4802	0.5951	0	1
2.3750	1.4252	0.5996	0	1
2.3371	1.5387	0.5906	0	1
2.0478	0.8871	0.7048	1	0
2.3946	1.4735	0.5967	0	1
2.0773	0.8703	0.7018	1	0
2.0691	0.8736	0.7046	1	0
2.3844	1.5763	0.5930	0	1
2.0756	0.8025	0.7040	1	0
2.3131	1.5449	0.5910	0	1
2.3171	1.4086	0.5952	0	1
2.0094	0.8350	0.7095	1	0
2.3458	1.5074	0.5993	0	1
2.0494	0.8582	0.7029	1	0
2.0290	0.8246	0.7010	1	0
2.3077	1.5442	0.5924	0	1
2.0658	0.8190	0.7037	1	0
2.3306	1.4741	0.5929	0	1
2.3168	1.5627	0.5953	0	1
2.0722	0.8005	0.7036	1	0
2.3731	1.5299	0.5932	0	1
2.3008	1.5308	0.5905	0	1
2.3613	1.5566	0.6000	0	1
2.3797	1.5284	0.5982	0	1
2.0529	0.8781	0.7055	1	0
2.0058	0.8412	0.7042	1	0
2.0186	0.8936	0.7007	1	0



2.3310	1.5888	0.5902	0	1
2.0555	0.8011	0.7069	1	0
2.3242	1.5620	0.5907	0	1
2.3129	1.5374	0.5970	0	1
2.3647	1.4928	0.5908	0	1
2.0242	0.8340	0.7033	1	0
2.0477	0.8531	0.7071	1	0
2.3240	1.5434	0.5913	0	1
2.0411	0.8575	0.7095	1	0
2.0881	0.8301	0.7031	1	0
2.3829	1.5941	0.5970	0	1
2.0998	0.8559	0.7001	1	0
2.3292	1.5384	0.5951	0	1
2.0083	0.8804	0.7098	1	0
2.0366	0.8861	0.7001	1	0
2.3641	1.5475	0.5997	0	1
2.0102	0.8107	0.7076	1	0
2.3583	1.5371	0.5903	0	1
2.0550	0.8223	0.7061	1	0
2.3989	1.4097	0.5901	0	1
2.3205	1.5825	0.5933	0	1
2.3462	1.4097	0.5954	0	1
2.0800	0.8711	0.7070	1	0
2.3259	1.5426	0.5928	0	1
2.3733	1.5245	0.5901	0	1
2.3152	1.4407	0.5918	0	1
2.0058	0.8461	0.7019	1	0
2.0599	0.8708	0.7009	1	0
2.0507	0.8116	0.7062	1	0
2.0046	0.8048	0.7017	1	0
2.3827	1.5223	0.5915	0	1
2.0114	0.8351	0.7011	1	0
2.0473	0.8317	0.7013	1	0
2.0464	0.8929	0.7058	1	0
2.0566	0.8745	0.7024	1	0
2.0016	0.8615	0.7076	1	0
2.3575	1.4816	0.5980	0	1
2.0512	0.8287	0.7087	1	0
2.3742	1.5990	0.5913	0	1
2.0486	0.8097	0.7002	1	0
2.3637	1.5888	0.5996	0	1
2.3286	1.5355	0.5941	0	1
2.0065	0.8121	0.7049	1	0
2.3891	1.5525	0.5934	0	1
2.0971	0.8829	0.7014	1	0
2.0755	0.8687	0.7046	1	0
2.3339	1.4019	0.5967	0	1
2.3247	1.5988	0.5909	0	1
2.0350	0.8722	0.7009	1	0
2.0241	0.8098	0.7032	1	0
2.0573	0.8815	0.7074	1	0
2.0112	0.8616	0.7086	1	0

2.3857	1.5639	0.5975	0	1
2.3898	1.5208	0.5986	0	1
2.3161	1.5766	0.5902	0	1
2.3601	1.4724	0.5977	0	1
2.0450	0.8357	0.7025	1	0
2.0468	0.8503	0.7007	1	0
2.3767	1.4091	0.5983	0	1
2.3777	1.4417	0.5975	0	1
2.3397	1.4961	0.5949	0	1
2.3625	1.5251	0.5901	0	1
2.0359	0.8724	0.7068	1	0
2.0509	0.8723	0.7058	1	0
2.3823	1.5883	0.5956	0	1
2.3423	1.5992	0.5939	0	1
2.3944	1.5824	0.5919	0	1
2.3690	1.4617	0.5944	0	1
2.0637	0.8231	0.7005	1	0
2.0115	0.8154	0.7017	1	0
2.3037	1.4625	0.5918	0	1
2.0235	0.8413	0.7092	1	0
2.3330	1.4411	0.5964	0	1
2.0976	0.8283	0.7064	1	0
2.0783	0.8534	0.7023	1	0
2.0318	0.8211	0.7063	1	0
2.0660	0.8462	0.7092	1	0
2.0781	0.8673	0.7074	1	0
2.3438	1.5184	0.5989	0	1
2.0319	0.8378	0.7089	1	0
2.3964	1.4040	0.5989	0	1
2.0403	0.8903	0.7029	1	0
2.0679	0.8020	0.7040	1	0
2.0821	0.8868	0.7084	1	0
2.0233	0.8933	0.7033	1	0
2.3575	1.4274	0.5913	0	1
2.3329	1.4976	0.5957	0	1
2.3261	1.5966	0.5903	0	1
2.3949	1.4903	0.5967	0	1
2.0008	0.8834	0.7083	1	0
2.0643	0.8678	0.7003	1	0
2.0355	0.8256	0.7030	1	0
2.0181	0.8585	0.7087	1	0
2.0625	0.8945	0.7040	1	0
2.0302	0.8848	0.7031	1	0
2.3003	1.4875	0.5932	0	1
2.0823	0.8244	0.7016	1	0
2.3552	1.5050	0.5908	0	1
2.0442	0.8955	0.7096	1	0
2.0013	0.8448	0.7093	1	0

## Apêndice II – Código Fonte

Os códigos apresentados neste apêndice foram desenvolvidos no aplicativo MATLAB, de versão R2006a. Todos os arquivos precisam trabalhar em conjunto para que o código realize o processamento de forma correta. Os dados apresentados no Apêndice I também são necessários.

### Arquivo da rede backpropagation: BACK2.m

```
% REDE BACKPROPAGATION

load('C:\Documents and Settings\guilherme\Desktop\UENF3\Bj1.mat');
load('C:\Documents and Settings\guilherme\Desktop\UENF3\Bj2.mat');
load('C:\Documents and Settings\guilherme\Desktop\UENF3\d.mat');
load('C:\Documents and Settings\guilherme\Desktop\UENF3\Wji1.mat');
load('C:\Documents and Settings\guilherme\Desktop\UENF3\Wji2.mat');
load('C:\Documents and Settings\guilherme\Desktop\UENF3\x.mat');

% 3 neurônios na camada de entrada (sinais de entrada)
% 3 neurônios na primeira camada oculta
% 2 neurônios na camada de saída
% Wji1: Pesos da primeira camada oculta (destino j e origem i)
% Wji2: Pesos da segunda camada oculta (destino j e origem i)
% Bj1: Biases da primeira camada oculta
% Bj2: Biases da segunda camada oculta
% sigmoid( alpha , v ) é uma função com a entrada do parâmetro de
% inclinação alpha e o campo local induzido v
% dsigmoid( alpha , v ) é a derivada de sigmoid

% ALTERAÇÕES NA REDE
rho = 2; % Objetos de conhecimento
% Campos de atração: obj para cada neurônio da primeira camada oculta
A_j_rho = [ 0 0 ; 0 0 ; 0 0 ]; % Matriz Ajr
% Os campos de atração são aplicados somente na primeira camada oculta
% Matriz dos pesos do primeiro campo de atração (rho)
Wji_rho1 = [ 0 0 0 ; 0 0 0 ; 0 0 0 ];
% Matriz dos pesos do segundo campo de atração (rho)
Wji_rho2 = [ 0 0 0 ; 0 0 0 ; 0 0 0 ];
sigma = 0; % índice vencedor
d1 = 0; % Distância euclidiana de rho 1
d2 = 0; % Distância euclidiana de rho 2

% Variáveis de DEBUG
td1 = 0;
td2 = 0;

criterio = 0.0000001; % critério de parada
n = 1; % iteração de treinamento
bias = 1;
alpha = 0.8; % parâmetro de inclinação da função sigmoide
eta = 0.3; % taxa de aprendizado
% Matriz do ajuste dos pesos da primeira camada oculta
dWji1 = [ 0 0 0 ; 0 0 0 ; 0 0 0 ];
```

```

% Matriz do ajuste dos pesos da camada de saída
dWji2 = [ 0 0 0 ; 0 0 0 ];
% Vetor do ajuste dos biases da primeira camada oculta
dBj1 = [ 0 0 0 ];
% Vetor do ajuste dos biases da camada de saída
dBj2 = [ 0 0 ];
% Vetor dos gradientes da primeira camada oculta
GRADj1 = [ 0 0 0 ];
% Vetor dos gradientes da camada de saída
GRADj2 = [ 0 0 ];
% Energia Média do Erro Quadrado
E_med = 10;
% Energia Total do Erro
E_total = 10;
t = 1; % Contador para cálculo da Energia Média do Erro Quadrado

% Critério de parada: -1 deve continuar, 0 parar
parada = 10;

% Ajuste inicial das Matrizes de pesos associadas aos rho
Wji_rho1 = Wji1; % Matriz do primeiro campos de atração
Wji_rho2 = Wji1; % Matriz do segundo campos de atração

disp( sprintf('\n :: Ajuste dos campos de atração') );
% Ajuste dos campos de atração
for i=1:500
% se o campo d( n , 1 ) for 1, estamos interagindo com rho 1
    if ( d( i , 1 ) == 1 ) % rho = 1
        td1 = td1 + 1;
        A_j_rho( 1 , 1 ) = A_j_rho( 1 , 1 ) + x( i , 1 );
        A_j_rho( 2 , 1 ) = A_j_rho( 2 , 1 ) + x( i , 2 );
        A_j_rho( 3 , 1 ) = A_j_rho( 3 , 1 ) + x( i , 3 );
    else % rho = 2
        td2 = td2 + 1;
        A_j_rho( 1 , 2 ) = A_j_rho( 1 , 2 ) + x( i , 1 );
        A_j_rho( 2 , 2 ) = A_j_rho( 2 , 2 ) + x( i , 2 );
        A_j_rho( 3 , 2 ) = A_j_rho( 3 , 2 ) + x( i , 3 );
    end
end
% Média da matriz de atração
A_j_rho( : , 1 ) = A_j_rho( : , 1 ) / td1;
A_j_rho( : , 2 ) = A_j_rho( : , 2 ) / td2;

disp( sprintf('\n :: Inicializando Treinamento') );

while ( E_med > criterio )

    % Camada de entrada não possui neurônios, somente a matriz de dados x

    %%%%%%%%%%%%%%% Propagação
    % :: Primeira camada oculta
    % Cálculo da Distância Euclidiana para descobrir rho
    d1 = sqrt( (A_j_rho(1,1)-x(n,1))^2 + (A_j_rho(2,1)-x(n,2))^2 +
(A_j_rho(3,1)-x(n,3))^2 );
    d2 = sqrt( (A_j_rho(1,2)-x(n,1))^2 + (A_j_rho(2,2)-x(n,2))^2 +
(A_j_rho(3,2)-x(n,3))^2 );

    % Cálculo do campo local induzido (primeira camada oculta)
    if ( d1 < d2 ) % rho = 1
        v1_1 = x(n,1)* Wji_rho1(1,1) + x(n,2)* Wji_rho1(1,2) + x(n,3)*
Wji_rho1(1,3) + bias*Bj1(1);

```

```

        v2_1 = x(n,1)* Wji_rho1(2,1) + x(n,2)* Wji_rho1(2,2) + x(n,3)*
Wji_rho1(2,3) + bias*Bj1(2);
        v3_1 = x(n,1)* Wji_rho1(3,1) + x(n,2)* Wji_rho1(3,2) + x(n,3)*
Wji_rho1(3,3) + bias*Bj1(3);
    else % rho = 2
        v1_1 = x(n,1)* Wji_rho2(1,1) + x(n,2)* Wji_rho2(1,2) + x(n,3)*
Wji_rho2(1,3) + bias*Bj1(1);
        v2_1 = x(n,1)* Wji_rho2(2,1) + x(n,2)* Wji_rho2(2,2) + x(n,3)*
Wji_rho2(2,3) + bias*Bj1(2);
        v3_1 = x(n,1)* Wji_rho2(3,1) + x(n,2)* Wji_rho2(3,2) + x(n,3)*
Wji_rho2(3,3) + bias*Bj1(3);
    end
    % Cálculo da função de ativação para encontrar as saídas da primeira
    % camada oculta
    y_1 = sigmoid( alpha , v1_1 );
    y_2 = sigmoid( alpha , v2_1 );
    y_3 = sigmoid( alpha , v3_1 );

    % :: Camada de saída
    % Cálculo do campo local induzido (camada de saída)
    v1_2 = y_1 * Wji2( 1 , 1 ) + y_2 * Wji2( 1 , 2 ) + y_3 * Wji2( 1 , 3 )
+ bias*Bj2(1);
    v2_2 = y_1 * Wji2( 2 , 1 ) + y_2 * Wji2( 2 , 2 ) + y_3 * Wji2( 2 , 3 )
+ bias*Bj2(2);
    % Cálculo da função de ativação para resultados da camada de saída
    o_1 = linear( v1_2 );
    o_2 = linear( v2_2 );
    %%%%%%%%%%%%%%% Fim da Propagação

    %%%%%%%%%%%%%%% Retropropagação
    % Cálculo do erro da camada de saída
    e1_2 = d( n , 1 ) - o_1;
    e2_2 = d( n , 2 ) - o_2;
    % Cálculo do delta Wji2 (correção dos pesos da camada de saída)
    % GRADj2 = ej * dsigmoid( vj );
    % dWji2 = eta * GRADj2 * yi
    GRADj2( 1 ) = e1_2*dlinear( v1_2 );
    dWji2( 1 , 1 ) = eta * GRADj2( 1 ) * y_1;
    dWji2( 1 , 2 ) = eta * GRADj2( 1 ) * y_2;
    dWji2( 1 , 3 ) = eta * GRADj2( 1 ) * y_3;
    GRADj2( 2 ) = e2_2*dlinear( v2_2 );
    dWji2( 2 , 1 ) = eta * GRADj2( 2 ) * y_1;
    dWji2( 2 , 2 ) = eta * GRADj2( 2 ) * y_2;
    dWji2( 2 , 3 ) = eta * GRADj2( 2 ) * y_3;
    % Cálculo do delta Bj2 (correção dos biases na camada de saída)
    % dBj2 = eta * ( ej * dsigmoid( vj ) ) * bias
    dBj2( 1 ) = eta * ( e1_2*dlinear( v1_2 ) ) * bias;
    dBj2( 2 ) = eta * ( e2_2*dlinear( v2_2 ) ) * bias;
    % Cálculo do delta Wji1 (correção dos pesos da primeira camada oculta)
    % GRADj1 = dsigmoid( vj ) * SOMATORIOk( GRADk2 * Wkj2 )
    % dWji1 = eta * ( GRADj1 ) * xi
    GRADj1( 1 ) = dsigmoid( alpha , v1_1 ) * ( ( GRADj2( 1 )*Wji2( 1 , 1 )
) * ( GRADj2( 2 )*Wji2( 2 , 1 ) ) );
    dWji1( 1 , 1 ) = eta * GRADj1( 1 ) * x( n , 1 );
    dWji1( 1 , 2 ) = eta * GRADj1( 1 ) * x( n , 2 );
    dWji1( 1 , 3 ) = eta * GRADj1( 1 ) * x( n , 3 );
    GRADj1( 2 ) = dsigmoid( alpha , v2_1 ) * ( ( GRADj2( 1 )*Wji2( 1 , 2 )
) * ( GRADj2( 2 )*Wji2( 2 , 2 ) ) );
    dWji1( 2 , 1 ) = eta * GRADj1( 2 ) * x( n , 1 );
    dWji1( 2 , 2 ) = eta * GRADj1( 2 ) * x( n , 2 );
    dWji1( 2 , 3 ) = eta * GRADj1( 2 ) * x( n , 3 );

```

```

GRADj1( 3 ) = dsigmoid( alpha , v3_1 ) * ( ( GRADj2( 1 )*Wji2( 1 , 3 )
) * ( GRADj2( 2 )*Wji2( 2 , 3 ) ) );
dWji1( 3 , 1 ) = eta * GRADj1( 3 ) * x( n , 1 );
dWji1( 3 , 2 ) = eta * GRADj1( 3 ) * x( n , 2 );
dWji1( 3 , 3 ) = eta * GRADj1( 3 ) * x( n , 3 );
% Cálculo do delta Bj1 (correção dos biases na primeira camada oculta)
% dBj1 = eta * GRADj1 * bias
dBj1( 1 ) = eta * GRADj1( 1 ) * bias;
dBj1( 2 ) = eta * GRADj1( 2 ) * bias;
dBj1( 3 ) = eta * GRADj1( 3 ) * bias;
% Ajuste dos pesos da camada de saída
Wji2( 1 , 1 ) = Wji2( 1 , 1 ) + dWji2( 1 , 1 );
Wji2( 1 , 2 ) = Wji2( 1 , 2 ) + dWji2( 1 , 2 );
Wji2( 1 , 3 ) = Wji2( 1 , 3 ) + dWji2( 1 , 3 );
Wji2( 2 , 1 ) = Wji2( 2 , 1 ) + dWji2( 2 , 1 );
Wji2( 2 , 2 ) = Wji2( 2 , 2 ) + dWji2( 2 , 2 );
Wji2( 2 , 3 ) = Wji2( 2 , 3 ) + dWji2( 2 , 3 );
% Ajuste dos pesos da primeira camada oculta
if ( d1 < d2 ) % rho = 1
    Wji_rho1( 1 , 1 ) = Wji_rho1( 1 , 1 ) + dWji1( 1 , 1 );
    Wji_rho1( 1 , 2 ) = Wji_rho1( 1 , 2 ) + dWji1( 1 , 2 );
    Wji_rho1( 1 , 3 ) = Wji_rho1( 1 , 3 ) + dWji1( 1 , 3 );
    Wji_rho1( 2 , 1 ) = Wji_rho1( 2 , 1 ) + dWji1( 2 , 1 );
    Wji_rho1( 2 , 2 ) = Wji_rho1( 2 , 2 ) + dWji1( 2 , 2 );
    Wji_rho1( 2 , 3 ) = Wji_rho1( 2 , 3 ) + dWji1( 2 , 3 );
    Wji_rho1( 3 , 1 ) = Wji_rho1( 3 , 1 ) + dWji1( 3 , 1 );
    Wji_rho1( 3 , 2 ) = Wji_rho1( 3 , 2 ) + dWji1( 3 , 2 );
    Wji_rho1( 3 , 3 ) = Wji_rho1( 3 , 3 ) + dWji1( 3 , 3 );
else % rho = 2
    Wji_rho2( 1 , 1 ) = Wji_rho2( 1 , 1 ) + dWji1( 1 , 1 );
    Wji_rho2( 1 , 2 ) = Wji_rho2( 1 , 2 ) + dWji1( 1 , 2 );
    Wji_rho2( 1 , 3 ) = Wji_rho2( 1 , 3 ) + dWji1( 1 , 3 );
    Wji_rho2( 2 , 1 ) = Wji_rho2( 2 , 1 ) + dWji1( 2 , 1 );
    Wji_rho2( 2 , 2 ) = Wji_rho2( 2 , 2 ) + dWji1( 2 , 2 );
    Wji_rho2( 2 , 3 ) = Wji_rho2( 2 , 3 ) + dWji1( 2 , 3 );
    Wji_rho2( 3 , 1 ) = Wji_rho2( 3 , 1 ) + dWji1( 3 , 1 );
    Wji_rho2( 3 , 2 ) = Wji_rho2( 3 , 2 ) + dWji1( 3 , 2 );
    Wji_rho2( 3 , 3 ) = Wji_rho2( 3 , 3 ) + dWji1( 3 , 3 );
end
% Ajuste das biases da camada de saída
Bj2( 1 ) = Bj2( 1 ) + dBj2( 1 );
Bj2( 2 ) = Bj2( 2 ) + dBj2( 2 );
% Ajuste das biases da primeira camada oculta
Bj1( 1 ) = Bj1( 1 ) + dBj1( 1 );
Bj1( 2 ) = Bj1( 2 ) + dBj1( 2 );
Bj1( 3 ) = Bj1( 3 ) + dBj1( 3 );
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Fim da Retropropagação

% Calculo da Energia Média do Erro Quadrado
E_total = ( ( e1_2*e1_2 ) + ( e2_2*e2_2 ) ) / 2;
if ( t == 1 )
    E_med = E_total;
else
    E_med = ( E_med + E_total ) / t;
end

% Critério de parada
% parada = ( sqrt(e1_2*e1_2) + sqrt(e2_2*e2_2) ) / 2;

n = n+1; % próxima iteração
t = t+1; % contador

```

```

if ( n > 499 ) % Repetir Lote de Dados
    n = 1;
    % parada = 0;
    disp( sprintf(' :: 500 Iterações :: ERRO: %6.8f', E_med') );
end
end

disp( sprintf(' :: A rede realizou %i iterações', t) );
disp( sprintf(' :: Energia Média do Erro Quadrado: %2.14f', E_med) );
disp( sprintf(' :: Diferença 1: %6.8f', e1_2 ) );
disp( sprintf(' :: Diferença 2: %6.8f', e2_2 ) );
disp( sprintf('\n') );
disp( sprintf(' ### TESTE DA REDE ###') );
n = 1;
%%%%%%%%%%%%%% Propagação
% :: Primeira camada oculta
% Cálculo do campo local induzido (primeira camada oculta)
v1_1 = x( n , 1 ) * Wji1( 1 , 1 ) + x( n , 2 ) * Wji1( 1 , 2 ) + x( n ,
3 ) * Wji1( 1 , 3 ) + bias*Bj1(1);
v2_1 = x( n , 1 ) * Wji1( 2 , 1 ) + x( n , 2 ) * Wji1( 2 , 2 ) + x( n ,
3 ) * Wji1( 2 , 3 ) + bias*Bj1(2);
v3_1 = x( n , 1 ) * Wji1( 3 , 1 ) + x( n , 2 ) * Wji1( 3 , 2 ) + x( n ,
3 ) * Wji1( 3 , 3 ) + bias*Bj1(3);
% Cálculo da função de ativação para encontrar as saídas da primeira
% camada oculta
y_1 = sigmoid( alpha , v1_1 );
y_2 = sigmoid( alpha , v2_1 );
y_3 = sigmoid( alpha , v3_1 );

% :: Camada de saída
% Cálculo do campo local induzido (camada de saída)
v1_2 = y_1 * Wji2( 1 , 1 ) + y_2 * Wji2( 1 , 2 ) + y_3 * Wji2( 1 , 3 )
+ bias*Bj2(1);
v2_2 = y_1 * Wji2( 2 , 1 ) + y_2 * Wji2( 2 , 2 ) + y_3 * Wji2( 2 , 3 )
+ bias*Bj2(2);
% Cálculo da função de ativação para resultados da camada de saída
o_1 = linear( v1_2 );
o_2 = linear( v2_2 );
%%%%%%%%%%%%%% Fim da Propagação
disp( sprintf(' Saída Real 1: %6.8f', o_1 ) );
disp( sprintf(' Saída Real 2: %6.8f', o_2 ) );
disp( sprintf(' Saída Esperada 1: %6.8f', d( n , 1 ) ) );
disp( sprintf(' Saída Esperada 2: %6.8f', d( n , 2 ) ) );
e1_2 = d( n , 1 ) - o_1;
e2_2 = d( n , 2 ) - o_2;
disp( sprintf(' Diferença 1: %6.8f', e1_2 ) );
disp( sprintf(' Diferença 2: %6.8f', e2_2 ) );
disp( sprintf(' #####') );
disp( sprintf('\n') );

```

### Arquivos com valores de Bias: Bj1.mat e Bj2.mat

Bj1 =

```

0.4030
-0.2655
-0.0962

```

Bj2 =

```

-0.1739
-0.1048

```

### Arquivos com valores de Pesos: Wji1.mat e Wji2.mat

Wji1 =		Wji2 =			
0.3178	-0.0679	-0.2969	0.1856	-0.2156	0.0568
0.3114	-0.3595	-0.0824	0.3311	-0.1280	-0.1621
-0.0136	0.0754	-0.2960			

### Arquivos com função sigmóide:

```
function y = sigmoid(alpha,v)
y = logsig(alpha*v);
end
```

### Arquivos com função linear:

```
function y = linear(v)
y = purelin(v);
end
```

### Arquivos com função a derivada da função sigmóide:

```
function y = dsigmoid(alpha,v)
y = alpha*v*(1-v);
end
```

### Arquivos com função a derivada da função linear:

```
function y = dlinear(v)
y = dpurelin(v);
end
```