

## Introdução

Este tutorial tem por objetivo apresentar uma introdução ao aprendizado artificial e automatizado (*machine learning*), focalizando-se sobre os aspectos referentes a uma técnica em particular, as redes neurais artificiais –R.N.A. Na primeira seção vamos discutir sobre a Inteligência Artificial, sobre a aquisição de conhecimentos e sobre a importância do aprendizado na construção de sistemas inteligentes. Na segunda seção iremos abordar as redes neurais artificiais (modelos conexionistas), onde vamos destacar: os diferentes tipos de redes e de algoritmos de aprendizado existentes; a representação do conhecimento neural; as características e limitações de uso deste tipo de técnicas, bem como mostraremos alguns exemplos de aplicações das RNAs. Para concluir, iremos discutir sobre os caminhos da pesquisa atual nesta área e tendências futuras no que diz respeito ao desenvolvimento dos sistemas inteligentes.

### 1. Inteligência Artificial e Aprendizado

Para podermos falar em Inteligência Artificial e Aprendizado Artificial, precisamos antes saber o que é “*inteligência*” e o que entendemos por “*aprendizado*”? Apesar de termos uma noção básica do que significam estas duas palavras, inteligência e aprendizado, temos uma grande dificuldade de defini-las em termos práticos e de forma bastante precisa.

#### 1.1. O que é Inteligência?

O que é inteligência? O que é um ser inteligente? Estas duas questões devem ser respondidas (ao menos de forma superficial) se quisermos então partir para a implementação dos sistemas inteligentes. A idéia de implementação dos sistemas inteligentes parte do pressuposto que iremos “copiar” a inteligência humana e colocá-la a nosso serviço através de implementações que automatizem este tipo de processo/comportamento.



Figura 1.1 – Inteligência: Natural e Artificial

O termo “*Artificial Intelligence*” (A.I. – I.A. em português) foi usado pela primeira vez em 1956 por McCarthy (e desenvolvido por grande pesquisadores como Marvin Minsky e Herbert Simon), e nada mais é do que uma tentativa de formalizar o eterno sonho da criação de um “cérebro eletrônico” (termo muito usado na ficção científica e mesmo na época inicial do desenvolvimento dos computadores). Encontramos também algumas definições interessantes na literaturas, tais como:

- (1) Definição de Inteligente: Dotado de inteligência, capaz de compreender, esperto, habilidoso [Larousse 99];

- (2) Definição de Inteligência: Faculdade de conhecer, de aprender, de conceber, de compreender: a inteligência distingue o homem do animal [Larousse 99];
- (3) Definição de Inteligência Artificial: Conjunto de teorias e de técnicas empregadas com a finalidade de desenvolver máquinas capazes de simular a inteligência humana [Larousse99];
- (4) Definição de Inteligência Artificial: A Inteligência Artificial é uma área de estudos da computação que se interessa pelo estudo e criação de sistemas que possam exibir um comportamento inteligente e realizar tarefas complexas com um nível de competência que é equivalente ou superior ao de um especialista humano [Nikolopoulos 97].

Destas definições acima, podemos fazer os seguintes comentários:

- As definições são usualmente recursivas, ou seja, um ser inteligente é aquele que possui inteligência, e a inteligência é a característica dos seres inteligentes;
- As definições possuem contradições, onde aparece que a inteligência distingue o homem do animal, pois só o homem é capaz de aprender, conhecer, compreender. Isto não é muito preciso/correto, pois os animais também podem aprender (a reconhecer o dono), conhecer (o caminho de volta para casa) e compreender (um comando de seu mestre). E podemos mesmo dizer que estas tarefas realizadas por animais são tipicamente tarefas consideradas na atualidade como “tarefas inteligentes” que os computadores ainda não realizam com um desempenho satisfatório.
- Uma pessoa com limitações (analfabeta, deficiente mental e/ou físico, com QI reduzido) poderá ser qualificada, pelos critérios acima, como inteligente? E no entanto ela é um ser inteligente com capacidades superiores as de um computador, ou você acha que, por exemplo, um computador conseguiria ir até uma farmácia comprar um remédio? (isto inclui tarefas de: planificação de trajetória, reconhecimento do caminho, tratamento de imprevistos, etc).

Esta discussão poderia se prolongar aqui de forma indefinida, mas acredito que um ponto importante a ser considerado é: se queremos reproduzir a inteligência humana, devemos ter consciência de quais aspectos da inteligência que gostaríamos que fossem reproduzidos, e que em nossa opinião, permitiriam o desenvolvimento de um sistema com propriedades ditas “inteligentes”. Esta pergunta foi feita em sala de aula para um conjunto de alunos da Unisinos, e o resultado foi bastante interessante, onde reproduzo aqui as respostas de forma a complementar esta discussão: O que é Inteligência?

- Associação de idéias e conceitos;
- Concluir coisas;
- Capacidade de aprendizado;
- Acúmulo de conhecimentos;
- Raciocínio: lógico, abstrato, dedução, analogia, indução, inferência, síntese, análise, ...;
- Uso de experiências e conhecimentos passados;
- Uso prático de conhecimentos e experiências;
- Tomada de decisões;
- Criar coisas novas (criatividade);
- Saber o que eu sei (capacidade de explicar algo);
- Saber que sei / Saber que não sei;
- Interação;
- Comunicação.

(espaço reservado para *voce* continuar esta reflexão!)

Vamos aqui usar a definição de inteligência (que nos convêm) que é baseada no seguinte ditado: “*Errar é humano, mas repetir o erro é burrice*”. Logo o conceito de inteligência (por oposição à burrice, e devido ao fato desta ser uma propriedade humana), indica que o ser humano (ou não) inteligente deve ser capaz de: adaptar-se, aprender, evoluir, corrigir seus erros. E assim estamos passando da questão “o que é um ser inteligente” para a questão seguinte “o que é aprendizado”.

### 1.1. O que é Aprendizado?

O *aprendizado natural* é também um conceito difícil de ser definido, mas também de grande importância para que possamos partir para a construção de sistemas inteligentes dotados da capacidade de aprendizado. Sugerimos ao leitor que faça aqui o uso de sua *inteligência natural* a fim de definir o que você entende por *aprendizado natural* e quais são as propriedades que estão associadas a este conceito. Depois sugerimos ao leitor que, de posse dessa sua definição/descrição, passe a analisar os próximos itens abordados neste tutorial, com um olhar crítico sobre o mesmo no que diz respeito a presença ou não destas propriedades associadas aos métodos que iremos apresentar.

Vamos tentar definir aqui brevemente o que entendemos (o que o **autor** entende) por aprendizado. Aprendizado é a capacidade de se adaptar, de modificar e melhorar seu comportamento e suas respostas, sendo portanto uma das propriedades mais importantes dos seres ditos inteligentes, sejam eles humanos ou não. A capacidade de aprender está ligada diretamente aos seguintes itens:

- **Adaptação** e mudança de comportamento de forma a evoluir (melhorar segundo algum critério). Um sistema, biológico ou artificial, que não seja capaz de evoluir ou de mudar seu comportamento diante de novas situações que lhe são propostas é um sistema sem inteligência;
- **Correção** dos erros cometidos no passado, de modo a não repeti-los no futuro. Este item está diretamente relacionado ao anterior, o sistema deve modificar seu comportamento caso o comportamento atual não satisfaça a algum tipo de exigência (onde a sobrevivência deve ser um dos quesitos mais importantes a serem considerados nos seres vivos);
- **Otimização**: melhoria da performance do sistema como um todo. O aprendizado pode implicar em uma mudança do comportamento que busque: a economia de energia gasta para realizar uma tarefa, a redução do tempo gasto numa tarefa, etc. Quando falamos em otimização, devemos lembrar que podemos ter quesitos contraditórios e opostos, onde teremos que maximizar ou minimizar custos de acordo com algum tipo de critério;
- **Interação** com o meio, pois é através deste contato com o mundo que nos cerca que podemos trocar experiências e/ou realizar experiência, de forma a adquirir novos conhecimentos;
- **Representação** do conhecimento adquirido. O sistema deve ser capaz de armazenar uma massa muito grande de conhecimentos, e isto requer uma forma de representar estes conhecimentos que permita ao sistema explorá-los de maneira conveniente. Como nossos recursos são limitados, devemos ter uma maneira de guardar conhecimentos e regras gerais, pois guardar tudo seria muito difícil (exige muita memória, dificulta o uso pela lentidão da consulta aos conhecimentos).

O aprendizado é um tema bastante interessante e amplo, onde insisto que o autor deste tutorial irá centrar a noção de inteligência no que diz respeito a capacidade de um sistema aprender. Visto que este tema é muito amplo, sugerimos a leitura de obras como a Sociedade da Mente [Minsky 85], Regras da Mente [Anderson 93] e Como a Mente Funciona [Pinker 99] como bases para uma discussão mais profunda sobre os questionamentos relativos a inteligência humana e aprendizado.

Como já foi dito, iremos nos concentrar no tema “aprendizado”, uma característica básica das redes neurais, mas antes de passarmos ao aprendizado conexionista (neural), achamos importante dar uma visão histórica sobre os métodos e conceitos básicos da Inteligência Artificial Simbólica (clássica).

### 1.3. Inteligência Artificial – Processamento Simbólico

A I.A. simbólica ficou conhecida nas décadas de 70/80 pela aparição dos sistemas especialistas, dos sistemas baseados em conhecimentos (KBS – *Knowledge based systems*), e da expansão do uso de linguagens como *Prolog* (Programação em Lógica). Nesta mesma época um dos grandes desafios à “inteligência” dos computadores eram jogos, destacando-se o jogo de Xadrez. Os japoneses também entraram na corrida pela criação de ferramentas de inteligência artificial, e propuseram a criação dos computadores de 5ª. Geração (computadores inteligentes, capazes de *escutar, falar e raciocinar*). Do outro lado do oceano, os americanos também criavam seus mega-projetos, onde um dos mais famosos foi o projeto CYC, que visava dotar um computador de conhecimentos, muitos conhecimentos de forma a torná-lo um grande “cérebro eletrônico” (o nome CYC vem da palavra *Encyclopaedia*).

Em meados dos anos 80 e no início dos anos 90 a I.A. atravessou (e ainda atravessa?) uma grande crise de identidade. Os resultados práticos obtidos com a I.A. simbólica, de certa maneira, eram decepcionantes. Os computadores progrediram em grandes passos (memória, velocidade, softwares, etc), e a I.A. não parecia estar acompanhando este progresso. A linguagem *Prolog* tinha seus atrativos, mas faltava alguma coisa (até a Borland chegou a lançar o seu Turbo *Prolog*, e o que houve com o T*Prolog*?). O projeto CYC esbarrou em problemas de gerência de uma grande base de conhecimentos, com muitas contradições, e até hoje está tentando representar o “senso comum” com um “bom senso”. O computador japonês de 5ª. Geração até sabe falar e escutar, mas ainda não compreende a linguagem humana. A síntese e o reconhecimento de voz são talvez os maiores avanços deste final de século, mas ainda enfrentam muitos problemas e não passam de *apenas mais uma* forma de comunicação homem-máquina, sem no entanto haver uma verdadeira interação do ponto de vista lingüístico.

E não podemos esquecer o Xadrez, pois o maior especialista do mundo, Kasparov, foi derrotado recentemente pelo *Deep Blue* da IBM. Mas o que significa realmente esta derrota? A I.A. atingiu os seus objetivos? O *Deep Blue* é uma máquina inteligente, e mais ainda, mais inteligente que toda a raça humana pois ganhou de nosso melhor representante? Não, de forma alguma podemos dizer que o *Deep Blue* é uma máquina inteligente... ele apenas possui uma capacidade de cálculo enorme, o que lhe deu uma grande vantagem sobre os seres humanos, mas seus conhecimentos, estes provêm dos seres humanos que lhe programaram. O *Deep Blue* é como uma grande calculadora, tão precisa e veloz que não tem ser humano que possa ganhar dele, mas tão burro quanto a calculadora. E como ele fez para ganhar, se o seu opositor era tão inteligente? Simples, ele podia prever todas as combinações possíveis de jogadas, em um número muito superior a capacidade humana (simples problemas combinatorial) e além disso armazenava em sua memória uma base de dados enorme de jogos clássicos de Xadrez já disputados por Kasparov e outros (simples banco de dados). Onde está a inteligência e a capacidade de aprendizado? Em uma das partidas o *Deep Blue* perdeu (entrou em *loop*), mas ele não se recuperou, foi resetado e reprogramado para evitar explicitamente aquela situação que lhe causou problemas.

Em 1999 o homem chegou lá, desenvolveu máquinas que sintetizam e reconhecem a voz e que ganham no Xadrez do maior especialista humano, mas que não apresentam muitas daquelas propriedades que foram discutidas nos itens 1.1 e 1.2. Deixamos para o leitor dar o seu veredito sobre a I.A. do final dos anos 90. Mas não seja tão cruel em seu veredito, pois muito, muito foi feito nestes últimos 15 anos.

Vejamos como evoluíram os sistemas especialistas. Inicialmente os Sistemas Especialistas (de 1ª. Geração [Nikolopoulos 97]) se baseavam em uma arquitetura como mostra a figura 1.2. O problema destes sistemas logo apareceu, a aquisição de conhecimentos não era automática e dependia do especialista e/ou engenheiro de conhecimento para que fossem adicionados novos conhecimentos ao sistema. Surgia assim o conhecido “gargalo da aquisição de conhecimentos” (*Knowledge Acquisition bottle-neck*) dos sistemas especialistas.

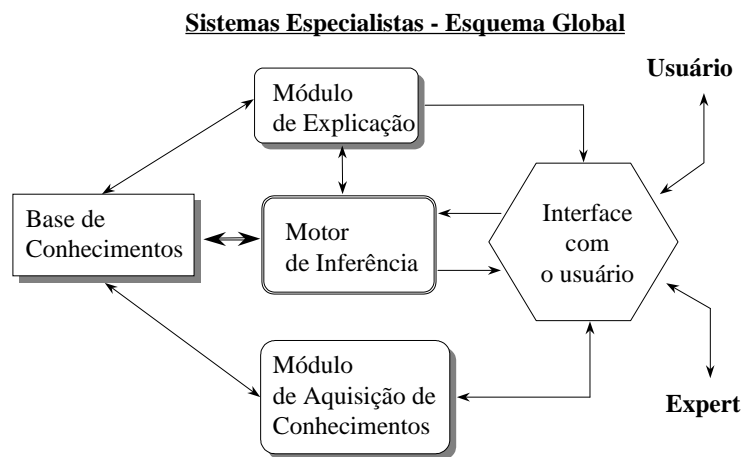


Figura 1.2 – Arquitetura dos Sistemas Especialistas

Os sistemas especialistas de 2ª. Geração introduziram a aquisição automática de conhecimentos, e então começamos a ouvir falar de aprendizado de máquinas simbólico (*symbolic machine learning*).

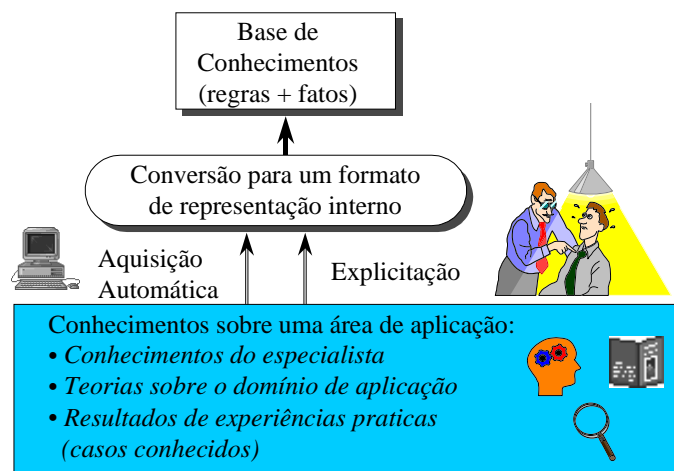


Figura 1.3 – Aquisição de Conhecimentos: Explicitação e Aprendizado Automático

A Inteligência Artificial começava a ter uma nova “cara” (vide fig. 1.4) onde deu-se mais importância a representação de conhecimentos e ao aprendizado, e não mais somente aos métodos de raciocínio automatizados (motor de inferência, processo de unificação, etc).

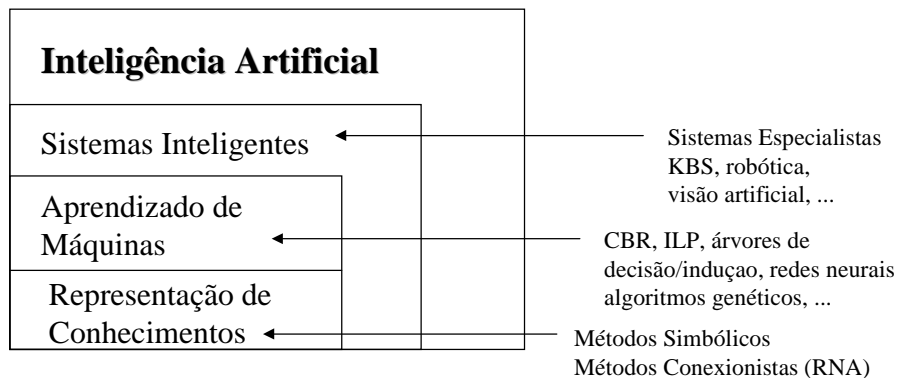


Figura 1.3 – Inteligência Artificial: Uma visão moderna

Como pode ser visto na figura 1.3, a representação de conhecimentos é uma espécie de divisor, que irá separar de um lado os métodos simbólicos e de outro os métodos conexionistas (redes neurais). Na realidade esta divisão não deve ser encarada como uma separação, mas sim como dois tipos de métodos que possuem cada qual suas peculiaridades e que devem ser entendidos e estudados de maneira a tirar o maior proveito de cada um. Fala-se muito atualmente em sistemas híbridos, multi-agentes ou múltiplas inteligências, como sendo uma direção para onde deve se orientar a I.A. do novo milênio [Osório 99].

#### 1.4. Inteligência Artificial – Aprendizado Simbólico

As ferramentas de I.A. simbólica, em sua evolução, também foram dotadas de mecanismos de aquisição automática de conhecimentos, onde podemos citar o exemplo da linguagem Prolog e dos sistemas especialistas (2ª. Geração) que foram dotados de mecanismos de aprendizado de máquinas. Os principais métodos (ou pelos menos os mais conhecidos) de aprendizado simbólico são:

- Aprendizado por analogia/por instâncias. Exemplo: Sistemas baseados em casos - CBR (*Case Based Reasoning*) [Mitchell 97, Kolodner 93];
- Aprendizado por Indução. Exemplos: Árvores de Decisão - ID3, C4.5, CN2 (IDT - *Induction of Decision Trees*) [Quinlan 92], e ILP - *Inductive Logic Programming (Prolog)* [Nilsson 98].
- Aprendizado por evolução/seleção. Exemplo: Algoritmos Genéticos - GA e GP (*Genetic Algorithms / Genetic Programming*) [Goldberg 89, Mitchell 97];
- Outros tipos de aprendizado: por reforço (*reinforcement learning*), não supervisionado, bayesiano e por explicações (*explanation based*) [Mitchell 97, Nilsson 98].

Estas ferramentas de aprendizado possuem fortes limitações: (1) usualmente assumem que os conhecimentos a serem adquiridos/manipulados e as informações disponíveis são corretos e devem estar completos em relação ao problema (o que dificilmente ocorre); (2) são orientadas para manipular informações simbólicas (informações qualitativas), onde os valores numéricos e contínuos (informações quantitativas) representam um problema difícil de ser tratado. Em relação a este último item, existem

tentativas de integrar ao processamento simbólico probabilidades (regras bayesianas) e incerteza (regras fuzzy), como forma de expandir as potencialidades deste tipo de sistemas.

#### **1.4. Inteligência Artificial – Alternativas ao aprendizado e processamento simbólico**

O processamento e aprendizado simbólico, devido as suas características básicas, possuem algumas limitações no que diz respeito a manipulação: de incertezas, de valores aproximados, de informações contraditórias, e de uma maneira geral, de informações quantitativas [Minsky 90]. O mundo que nos cerca é extremamente “impreciso” do ponto de vista computacional, e sendo assim, as ferramentas computacionais de I.A. devem portanto ser bastante robustas para poderem trabalhar neste ambiente de informações imprecisas (informações incorretas) e cheio de problemas imprevisíveis (informações incompletas).

Muitos pesquisadores, na sua busca da implementação de ferramentas e sistemas inteligentes, se orientaram para os sistemas baseados “realmente” no funcionamento do cérebro humano. Este tipo de enfoque levou muitos pesquisadores (entre eles Marvin Minsky, Seymour Papert e John Von Neumann) a estudarem novas formas de processamento de informações baseadas nos estudos neuro-fisiológicos do cérebro. Esta corrente de pesquisas tentava reproduzir os neurônios como elementos básicos do processamento de novas arquiteturas de máquinas inteligentes, ao invés de usar portas lógicas, bits e bytes controlados por uma Unidade Central.

Esperava-se que de elementos de processamento baseados em neurônios, conectados entre si com um grande número de ligações, e operando em paralelo, pudessem “emergir comportamentos inteligentes”. Este ramo da I.A. foi denominado de Inteligência Artificial Conexionista.

## 2. Redes Neurais Artificiais

### 2.1. Introdução

As Redes Neurais Artificiais (RNA), também conhecidas como métodos conexionistas, são inspiradas nos estudos da maneira como se organiza e como funciona o cérebro humano. Este tipo de método possui características peculiares de representação e de aquisição de conhecimentos, sendo considerado um método de nível sub-simbólico (em oposição aos métodos simbólicos, citados na seção anterior: árvores de decisão, CBR, KBS, etc).

Inicialmente vamos discutir sobre a representação de conhecimentos utilizada pelas Redes Neurais, para depois analisarmos a parte referente ao aprendizado destas. É importante salientar que existem diferentes tipos de RNAs e que cada um destes modelos diferentes possui suas características próprias quanto a forma de representar e de adquirir conhecimentos. Em função disto vamos primeiramente apresentar uma visão geral, classificando os diferentes modelos de RNAs para em seguida nos concentrarmos em um modelo mais específico: as redes neurais artificiais do tipo multi-nível baseada em Perceptrons (conhecidas como Multi-Layer Perceptron Nets ou Back-Propagation Nets). A figura 2.1 apresenta um exemplo deste modelo de rede conexionista.

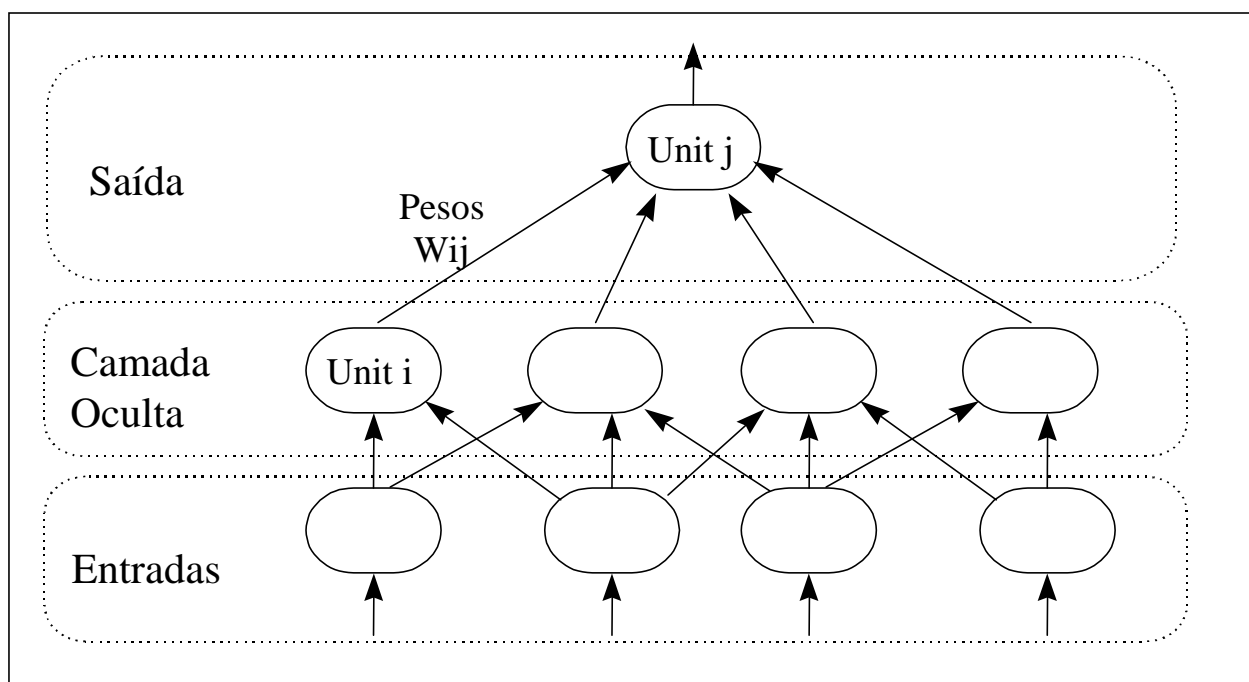


Figura 2.1. Exemplo de Rede Neural Artificial do tipo Multi-Nível

### 2.2. Conceitos Básicos, Origem e Evolução das Redes Neurais

Vamos discutir inicialmente como se estruturam as redes neurais e seus neurônios, passando depois para uma apresentação da origem e evolução dos estudos nesta área.



### 2.2.1. Representação de Conhecimentos

A representação de conhecimentos nas redes conexionistas, como diz o próprio nome, é fortemente ligada a noção de conexão entre neurônios (elementos processadores de informação) que interagem uns com os outros através destas ligações. O modelo conexionista possui sua origem nos estudos feitos sobre as estruturas de nosso cérebro – *sofrendo uma grande simplificação do modelo original* – onde encontramos no modelo artificial, que é simulado, elementos como os neurônios e as suas conexões, chamadas de sinapses. A proximidade e fidelidade dos modelos artificiais em relação ao modelo real é um *tema polêmico* que não vamos nos aventurar a discutir aqui, deixando apenas o registro da origem dos conceitos básicos que norteiam este campo de pesquisas.

O conhecimento de uma RNA está codificado na estrutura da rede, onde se destacam as conexões (sinapses) entre as unidades (neurônios) que a compõe. Nestas redes artificiais obtidas por simulação em computadores, associamos a cada conexão um peso sináptico (valor numérico) que caracteriza a força da conexão entre dois neurônios. O aprendizado em uma RNA é realizado por um processo de adaptação dos seus pesos sinápticos. As figuras 2.2 e 2.3 mostram a relação entre os neurônios naturais e o modelo de neurônio artificial.

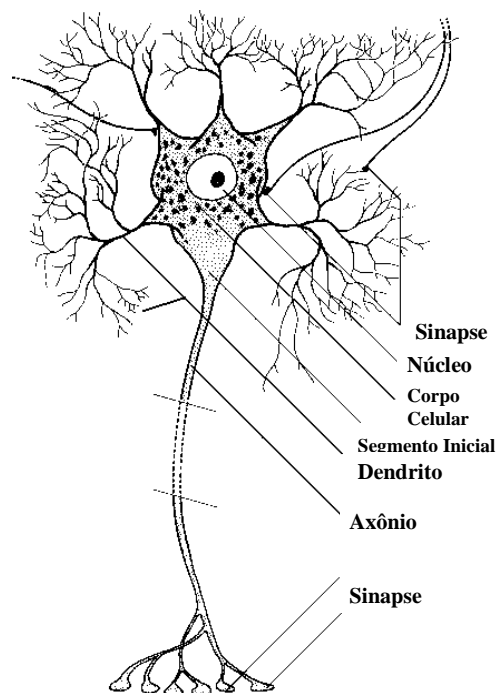


Figura 2.2. Exemplo de Neurônio Natural

Uma vez que os conhecimentos da rede estão codificados na estrutura de interconexões entre os neurônios e nos pesos associados a estas conexões, fica muito difícil para um ser humano realizar uma análise e interpretação dos conhecimentos adquiridos por uma RNA. Os conhecimentos das redes se resumem a um conjunto de valores numéricos descrevendo as conexões, e por conseqüência, estes valores

descrevem também o comportamento da rede. Entretanto, para um ser humano estes dados não fazem muito sentido. Na seção 2.3 vamos apresentar mais em detalhes os diferentes tipos de redes, onde a escolha do tipo de neurônio artificial utilizado é fundamental para se definir como será a representação interna dos conhecimentos da rede.

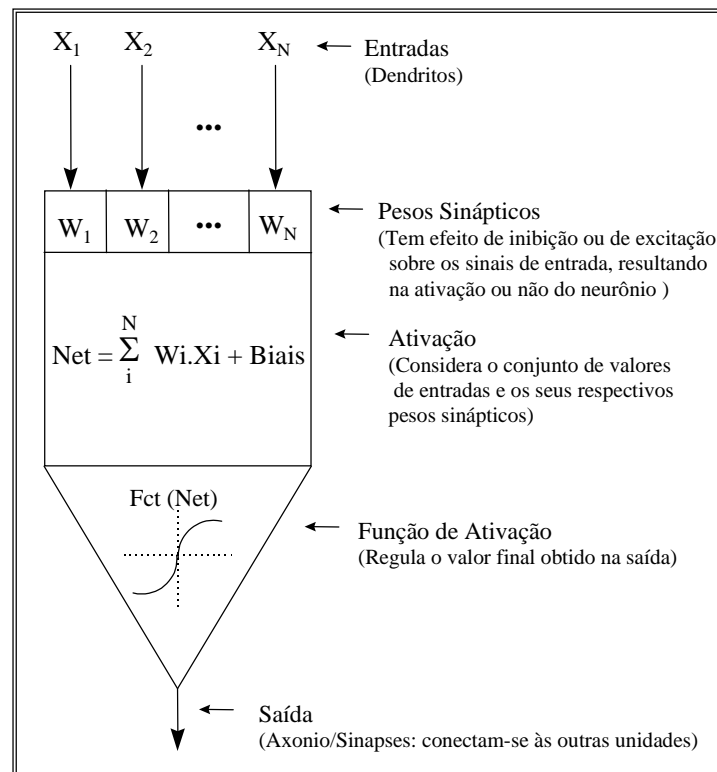


Figura 2.3. Exemplo de Neurônio Artificial

Apesar dos estudos sobre as redes neurais artificiais serem considerados como pertencentes à uma área “jovem”<sup>1</sup> de pesquisa, encontramos atualmente uma série de referências e exemplos de importantes aplicações práticas deste tipo de método de aprendizado, onde podemos citar alguns exemplos de obras relevantes na área, como por exemplo [Fiesler 97, Ripley 96, Bishop 95, Arbib 95, Krose 93, Freeman 92, Simpson 90, Faq 99a].

### 2.2.2. Origem e Evolução

Os primeiros estudos sobre as Redes Neurais Artificiais e propostas de modelos destas redes surgiram nos anos 40. Os primeiros modelos evoluíram bastante, onde alguns deles se destacaram e tornaram-se famosos, mas mesmo assim até hoje continuam sendo propostos novos modelos de redes neurais. O estudo dos primeiros modelos e de sua evolução nos ajuda a entender melhor as redes neurais, e seu estágio atual de evolução.

<sup>1</sup> Os primeiros estudos sobre as redes neurais artificiais remontam aos anos 40 (McCulloch-Pitts), mas foi somente na década de 80 que houve um grande desenvolvimento nesta área (Back-Propagation, Hopfield, Kohonen SOFM, ...)

O começo do estudo das redes neurais artificiais pode ser atribuído à criação do *Psychon* em 1943 por McCulloch e Pitts [McCulloch 43], sendo que alguns anos mais tarde, em 1949 D. O. Hebb publicava uma importante obra, o livro “*The Organization of Behaviour*” [Hebb 49], que influenciou vários modelos de RNAs de destaque na atualidade.

Em 1959, Frank Rosenblatt criou o *Perceptron* [Rosenblatt 59] que, como será visto neste trabalho, tem até hoje uma grande influência sobre os estudos das redes neurais, mostrando que apesar desta área de estudos ter crescido muito na atualidade, suas bases foram estruturadas juntamente com a criação dos fundamentos da ciência da computação. Alguns outros modelos similares ao Perceptron foram também desenvolvidos nesta época, como é o caso do *Adaline* (Adaptive Linear Element), criado por Bernard Widrow em 1962 [Widrow 62, 88, 90]. Os modelos do tipo Perceptron, incluindo o Adaline, são baseados no aprendizado supervisionado por correção de erros, uma classe muito importante de redes neurais artificiais, que possui uma larga aplicação na atualidade.

Em 1969 os modelos baseados no Perceptron receberam uma dura crítica feita por Minsky e Papert através de sua obra “*Perceptrons: An Introduction to Computational Geometry*” [Minsky 69]. Através deste livro, Minsky e Papert provaram matematicamente que os modelos de redes neurais baseados no Perceptron (redes de um só nível, o que na época era o tipo de rede de Perceptrons utilizado), **não** eram capazes de aprender uma simples função lógica do tipo “ou-exclusivo” (XOR = *Exclusive Or*). A função XOR possui um padrão de valores de entrada e de saída cuja associação não podia ser aprendida pelos modelos de redes baseados em Perceptron disponíveis naquela época. O impacto da publicação deste livro abalou profundamente as pesquisas realizadas nesta área de estudos.

O Madaline (Many Adaline), também criado por Widrow [Widrow 88, 90], podia de certa forma resolver o problema, mas o aprendizado não podia ser realizado de uma forma muito “natural” e automatizada, pois requeria a intervenção humana na construção da rede. Devido as críticas feitas e a falta de uma solução para os problemas apresentados, as redes neurais ficaram “esquecidas” por um certo tempo...

Somente na década de 80, surgiram novos modelos que deram um novo impulso as redes neurais. Em 1982 surgia um modelo importante de rede criado por J. Hopfield [Hopfield 82], onde este modelo começou a dar um novo impulso as redes neurais. O modelo que Hopfield criou era baseado em um tipo de rede diferente dos modelos baseados no Perceptron, sendo uma rede com conexões recorrentes e com um comportamento baseado na competição entre os neurônios, onde o aprendizado era não supervisionado. Outros modelos similares ao modelo de Hopfield surgiram pouco depois, onde podemos citar alguns como por exemplo: a máquina de Boltzmann [Hinton 84] e o BAM (*Binary Associative Memory*) [Kosko 87, 87a].

A década de 80 ficou também marcada profundamente pelo reaparecimento das redes baseadas em Perceptrons. Isto deveu-se ao desenvolvimento dos computadores, que eram mais velozes e permitiam realizar melhores simulações das redes neurais, bem como o desenvolvimento de modelos matemáticos que permitiram a solução do problema apontado por Minsky e Papert. Também podemos associar em parte este renascimento das redes neurais ao suposto desencanto com a I.A. clássica. O modelo que permitiu o ressurgimento das redes baseadas em Perceptrons foi o das redes multi-nível, onde o novo algoritmo de aprendizado chamado *Back-Propagation* resolveu em grande parte os problemas de aprendizado existentes até então. Este modelo foi desenvolvido por diferentes pesquisadores quase ao mesmo tempo, como D. Parker [Parker 82] e D. Rumelhart [Rumelhart 85], mas foi Rumelhart e Hinton que o tornaram este algoritmo famoso com a sua obra “*Parallel Distributed Processing - PDP*” [Rumelhart 86]. Este algoritmo, o Back-Propagation permitia realizar o aprendizado por correção de erros em uma rede com múltiplas camadas (níveis) e conseqüentemente resolveria o problema do XOR.

Além dos modelos de Hopfield e do modelo de redes multi-nível com Back-Propagation (chamado de *Multi-Layer Perceptron – MLP*), outro modelo importante que surgiu nesta década foi o modelo de Teuvo Kohonen [Kohonen 82, 87]. O modelo de Kohonen é muito interessante pois permite o aprendizado competitivo com uma auto-organização da rede neural, criando os chamados “mapas de atributos auto-organizáveis” (*self-organizing feature maps*).

Por fim, o último modelo de destaque neste período, foi o modelo ART (*Adaptive Resonance Theory*) criado por Gail Carpenter e Stephen Grossberg [Carpenter 93]. Este modelo possui um aprendizado do tipo não supervisionado, criando protótipos (*clusters*) dos padrões aprendidos. O modelo ART teve diversas versões posteriores, entre elas versões do tipo semi-supervisionado e com uso de conceitos da lógica nebulosa (Fuzzy-ART).

Os estudos sobre as redes neurais sofreram uma grande revolução a partir dos anos 80, conforme foi demonstrado acima. E, a partir dos anos 80, cada vez mais, esta área de estudos tem se destacado, seja pelas promissoras características apresentadas pelos modelos de redes neurais propostos, ou seja pelas condições tecnológicas atuais de implementação que permitem desenvolver arrojadas implementações de arquiteturas neurais paralelas em hardwares dedicados, obtendo assim ótimas performances destes sistemas (bastante superiores aos sistemas convencionais).

## 2.3. Modelos Conexionistas

### 2.3.1. Definição de uma Rede Neural Artificial

As *redes conexionistas* são formadas por um conjunto de unidades elementares de processamento de informações fortemente conectadas, que denominamos de neurônios artificiais. Uma RNA é constituída por um *grafo orientado e ponderado*. Os *nós* deste grafo são *autômatos simples*, os chamados *neurônios artificiais*, que formam através de suas conexões um autômato mais complexo, a rede neural, também conhecida como rede conexionista.

Cada unidade da rede é dotada de um estado interno, que nós vamos denominar de *estado de ativação*. As unidades podem propagar seu estado de ativação para as outras unidades do grafo, passando pelos arcos ponderados, que nós chamamos de *conexões*, *ligações sinápticas* ou simplesmente de *pesos sinápticos*. A regra que determina a ativação de um neurônio em função da influência vinda de suas entradas, ponderadas pelos seus respectivos pesos, se chama *regra de ativação* ou *função de ativação*.

As mudanças realizadas nos valores dos pesos sinápticos ou na estrutura de interconexão das unidades de uma rede, são responsáveis pelas alterações no comportamento de ativação desta rede. Estas alterações nas conexões e na estrutura da rede é o que nos permite realizar o *aprendizado* de um novo comportamento. Desta maneira vamos poder modificar o estado de ativação na saída da rede em resposta a uma certa configuração de entradas. Portanto, a rede é capaz de estabelecer associações de entrada-saída (estímulo e resposta) a fim de se adaptar a uma situação proposta. No caso de uma rede com aprendizado supervisionado (vide item sobre tipos de aprendizado), a rede deve adaptar os seus pesos de maneira à passar a responder de acordo com o exemplo dado, ou seja, gerando na sua saída um estado de ativação compatível para com o esperado. O método utilizado para modificar o comportamento de uma rede é denominado de *regra de aprendizado*.

A grande quantidade de modelos de redes conexionistas existentes torna difícil para nós a descrição exaustiva de todos eles. Se o leitor assim desejar, poderá se aprofundar em maiores detalhes sobre os diferentes modelos de RNAs em obras como o “Handbook of Neural Computation” [Fiesler 97]. Nós iremos nos concentrar aqui em diferenciar estes modelos, tomando como base as suas principais características.

### 2.3.2. Classificação e Propriedades

A grande quantidade de modelos existentes nos leva à uma análise de suas principais propriedades e diferenças em detrimento de uma análise caso à caso mais detalhada. Este estudo das principais propriedades das redes neurais nos permite compreender melhor as vantagens e/ou inconvenientes da escolha de um modelo em detrimento de um outro. Consideramos que não existe apenas uma maneira de classificar todos os modelos, mas de um modo geral devem ser considerados grupos de atributos, tais como: tipo de aprendizado, arquitetura de interconexões, forma interna de representação das informações, tipo de aplicação da rede, etc. Caso o leitor tenha o interesse de buscar uma proposta mais formal de classificação das redes neurais, esta pode ser encontrada em [Fiesler 97].

#### 2.3.2.1. *Aprendizado Conexionista*

O aprendizado conexionista é em geral um processo gradual e iterado, onde os pesos são modificados várias vezes, pouco à pouco, seguindo-se uma regra de aprendizado que estabelece a forma como estes pesos são alterados. O aprendizado é realizado utilizando-se um conjunto de dados de aprendizado disponível (base de exemplos). Cada iteração deste processo gradativo de adaptação dos pesos de uma rede neural, sendo feita uma apresentação completa do conjunto de dados, é chamada de *época de aprendizado*. Os métodos de aprendizado neural podem ser divididos em três grandes classes, segundo o grau de controle dado ao usuário

- **Aprendizado supervisionado:** o usuário dispõe de um comportamento de referência preciso que ele deseja ensinar a rede. Sendo assim, a rede deve ser capaz de medir a diferença entre seu comportamento atual e o comportamento de referência, e então corrigir os pesos de maneira a reduzir este erro (desvio de comportamento em relação aos exemplos de referência). O aprendizado supervisionado utiliza conhecimentos empíricos, habitualmente representados por um conjunto de exemplos etiquetados, ou seja, exemplos com pares de dados de entrada com a respectiva saída associada. A tabela verdade de uma operação booleana do tipo *AND* poderia ser considerada como um conjunto de exemplo de aprendizado, pois indica os valores de entrada e também a saída desejada. Nos casos de problemas de classificação, a saída é a classe à qual cada exemplo está associado. Exemplo de aplicação: reconhecimento de caracteres em uma aplicação do tipo OCR (*Optical Character Recognition*) [Osório 91].
- **Aprendizado semi-supervisionado:** o usuário possui apenas indicações imprecisas (por exemplo: sucesso/insucesso da rede) sobre o comportamento final desejado. As técnicas de aprendizado semi-supervisionado são chamadas também de aprendizado por reforço (reinforcement learning) [Sutton 98]. Para ser mais exato, neste tipo de aprendizado nós dispomos apenas de uma avaliação qualitativa do comportamento do sistema, sem no entanto poder medir quantitativamente o erro (desvio do comportamento em relação ao comportamento de referência desejado). Exemplo: aplicações em robótica autônoma, onde supondo uma situação hipotética, sabemos que seguir em frente não é possível pois existe um obstáculo, mas em compensação não temos uma medida numérica que indique para que lado seguir e exatamente como devemos proceder para desviar deste obstáculo.

- **Aprendizado não-supervisionado:** os pesos da rede são modificados em função de critérios internos, tais como, por exemplo, a repetição de padrões de ativação em paralelo de vários neurônios. O comportamento resultante deste tipo de aprendizado é usualmente comparado com técnicas de análise de dados empregadas na estatística (e.g. *clustering*). Exemplo: diferenciar tomates de laranjas, sem no entanto ter os exemplos com a sua respectiva classe etiquetada (e.g. self-organizing feature maps [Kohonen 87]).

O aprendizado conexionista em geral precisa de uma grande quantidade de dados, que nós agrupamos em uma *base de aprendizado*. Exemplos de bases de aprendizado podem ser encontrados na Internet no *UCI-ML repository* [UCI 99]. De acordo com a técnica de aprendizado utilizada, outros conjuntos de dados podem também ser necessários, principalmente para que se possa medir a validade do aprendizado realizado pela rede (e.g. cross-validation [Krogh 95]). Este conjunto de dados complementar é usualmente chamado de conjunto de *teste de generalização*. A figura 2.4 apresenta um gráfico típico da evolução do erro durante o aprendizado de uma rede neural, comparando a curva do erro (aprendizado supervisionado) referente à base de aprendizado com a curva do erro da base de teste de generalização.

Nós chamamos de *generalização* a capacidade de um modelo de aprendizado responder corretamente aos exemplos que lhe são apresentados, sendo que estes exemplos *NÃO* devem estar presentes na base de aprendizado. Um modelo que tem uma boa generalização é aquele modelo que responde corretamente aos exemplos contidos na base de aprendizado, mas também a outros exemplos diferentes daqueles da base de aprendizado, e que estão contidos em uma base de teste. A capacidade de generalizar é a principal capacidade buscada nas tarefas que envolvem aprendizado.

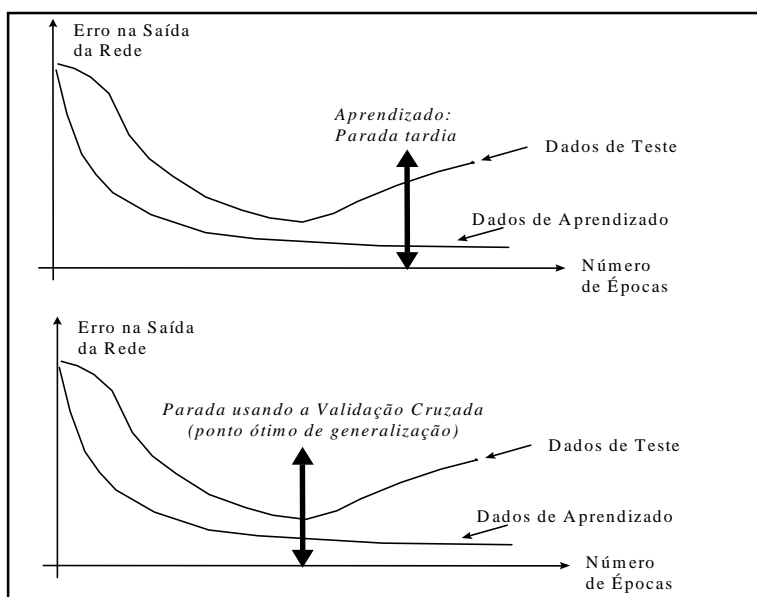


Figura 2.4. Aprendizado: Erro em relação ao conjunto de aprendizado e de teste

Uma rede pode se especializar demasiadamente em relação aos exemplos contidos na base de aprendizado. Este tipo de comportamento vai nos levar a um problema de aprendizado conhecido como super-aprendizado (*over-training / over-fitting*). Normalmente o over-fitting pode ser detectado/evitado através do uso de um teste de generalização por validação cruzada (*cross-validation*).

O aprendizado de um conjunto de dados pode ser realizado de diferentes formas, se considerarmos a maneira pela qual a rede é alimentada por estes dados:

- **Aprendizado instantâneo:** o conjunto de dados de aprendizado é analisado uma única vez e com isto o conjunto de pesos da rede é determinado de maneira imediata em uma única passagem da base de exemplos. Este modo de aprendizado também é conhecido como: *one single epoch learning / one shot learning*.
- **Aprendizado por pacotes:** o conjunto de dados de aprendizado é apresentado à rede várias vezes, de modo que possamos otimizar a resposta da rede, reduzindo os erros da rede e minimizando o erro obtido na saída desta. Este modo de aprendizado é caracterizado por trabalhar com uma alteração dos pesos para cada época, ou seja, para cada passagem completa de todos os exemplos base de aprendizado. O algoritmo de aprendizado deve reduzir pouco à pouco o erro de saída, o que é feito ao final de cada passagem (análise) da base de exemplos de aprendizado. Neste tipo de processo, podemos apresentar os exemplos na ordem em que se encontram, ou de modo mais usual, apresentar os dados em uma ordem aleatória. Outros tipos de seleção de exemplos para análise pelo algoritmo de aprendizado nos levam a métodos como a aprendizagem ativa (vide mais abaixo). Este método é conhecido pelo nome de batch-learning e constitui-se de um dos métodos mais utilizados.
- **Aprendizado contínuo:** o algoritmo de aprendizado leva em consideração continuamente os exemplos que lhe são repassados. Se o conjunto de dados é bem delimitado, chamamos este método de aprendizado on-line, e caso o conjunto de dados possa ir aumentando (sendo adicionados novos exemplos no decorrer do tempo), então chamamos este método de aprendizado incremental. O aprendizado on-line se opõe ao aprendizado por pacotes, pois ao contrário deste, para cada novo exemplo analisado já se realiza uma adaptação dos pesos da rede, com o objetivo de convergir na direção da solução do problema. O aprendizado contínuo incremental deve ser analisado sob o ponto de vista da aquisição dos dados (adição de novos exemplos na base de aprendizado), onde devemos prestar atenção para não confundir este tipo de aprendizado com o aprendizado incremental em relação a estrutura da rede (adição de novos neurônios no decorrer da simulação). O principal problema do aprendizado contínuo é a dificuldade de achar um bom compromisso entre a plasticidade e a estabilidade da rede. Uma rede com uma grande facilidade de adaptação pode “esquecer” rapidamente os conhecimentos anteriormente adquiridos e uma rede com uma grande estabilidade pode ser incapaz de incorporar novos conhecimentos.
- **Aprendizado ativo:** este modo de aprendizado assume que o algoritmo de adaptação da rede pode passar de uma posição passiva (apenas recebendo os dados do jeito como lhe são passados), para uma posição ativa. Sendo assim, assumimos que este algoritmo poderá vir a intervir sobre a forma como os dados lhe são repassados. Neste caso, a rede pode intervir e determinar assim quais dados que serão considerados e/ou desconsiderados, além também de determinar a ordem em que estes dados deverão ser considerados. A rede pode também vir a solicitar novos dados que julgue necessários para o bom aprendizado do problema proposto. Esta é uma área que vem sendo investigada com mais destaque recentemente.

A adaptação/otimização dos pesos também pode ser implementada por diferentes métodos, segundo o tipo de regra de aprendizado que for empregado. As regras de aprendizado<sup>2</sup> mais usadas são [Jodoin 94, Caudill 92, Simpson 90, Faq 99]:

---

<sup>2</sup> Maiores detalhes sobre a implementação de algoritmos de aprendizado neural podem ser encontradas nas seguintes obras [Osório 91, 92, 98] e na Internet em <http://www.inf.unisinos.br/~osorio/neural.html>

- Métodos de correção do erro, tais como a descida de uma superfície de erro baseada no gradiente. Exemplos de modelos deste tipo: Adaline, Perceptron, Back-Propagation, Cascade-Correlation;
- Métodos de aprendizado por reforço. Exemplos: Driver-Reinforcement Learning, AHC;
- Métodos de aprendizado por competição ou por auto-organização. Exemplos: Kohonen Self-Organizing Feature Maps, ART1;
- Métodos de aprendizado através da criação de protótipos ou *clusters*. Exemplos: RBF, ART1, ARN2;
- Métodos de aprendizado baseados em memórias associativas (auto-associativas ou hetero-associativas). Exemplos: Modelo de Hopfield, BAM.
- Métodos de aprendizado de seqüências temporais (redes recorrentes). Exemplos: SRN, BPTT, RTRL.

Existem alguns métodos que podem pertencer a duas categorias ao mesmo tempo, por exemplo, as redes com aprendizado do tipo ARN2 [Giacometti 95] que inclui neste modelo técnicas de aprendizado não-supervisionado, aprendizado supervisionado, adaptação por competição, e também através do uso de um método de criação de protótipos. Nas seções seguintes, nós vamos focar com mais atenção os modelos baseados no Perceptron: que possuem aprendizado supervisionado com descida do gradiente.

### 2.3.2.2. Tipos de Unidades

As unidades de uma rede – os neurônios artificiais – podem ser de diferentes tipos, de acordo com a função interna utilizada para calcular o seu estado de ativação. As principais diferenças são relativas ao tipo de função de ativação utilizada (e.g. linear, sigmoide assimétrica (exp), sigmoide simétrica (tanh), gaussiana, etc) [Jodoin 94a, Jodoin 94b]. Outro elemento importante diz respeito a forma como os neurônios armazenam as informações: unidades baseadas em protótipos, unidades do tipo Perceptron. Vamos diferenciar aqui estes dois tipos de maneiras de representar o conhecimento nas unidades de uma rede.

- **Redes à base de protótipos:** este tipo de rede utiliza neurônios que servem para representar protótipos dos exemplos aprendidos – as unidades tem uma representação interna que agrupa as características comuns e típicas de um grupo de exemplos [Orsier 95]. As redes baseadas em protótipos tem normalmente um aprendizado não supervisionado (com um ou mais protótipos associados à cada classe). Uma das vantagens deste tipo de redes é a possibilidade de fazer um aprendizado contínuo e incremental, uma vez que não é muito difícil de conceber um algoritmo capaz de aumentar a rede neural através da adição de novos protótipos. Os protótipos são também denominados de clusters, onde apresentamos um exemplo de rede a base de protótipos na figura 2.5. Este tipo de redes vão gerar uma representação dita *localista* de conhecimentos.



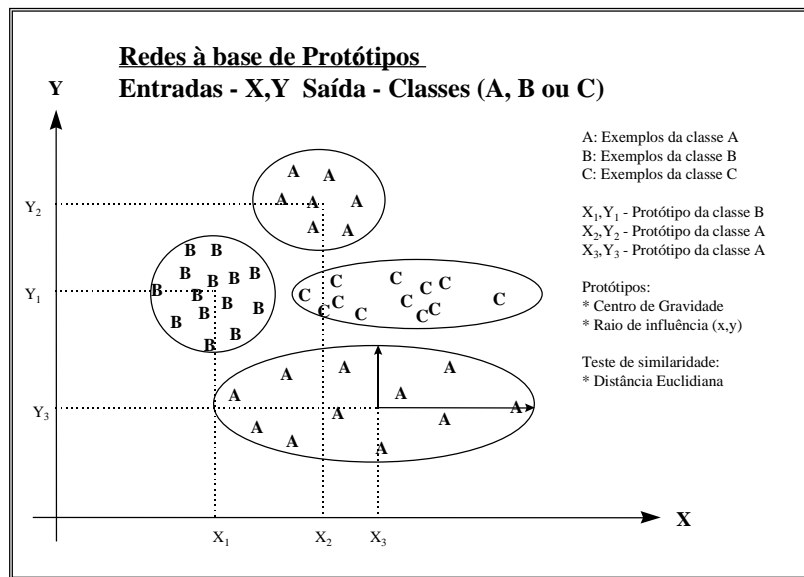


Figura 2.5. Protótipos de uma rede neural com duas entradas

- Redes à base de Perceptrons:** as unidades do tipo “Perceptron” foram criadas por Frank Rosenblatt em 1950. Este é um dos modelos de neurônios mais utilizados na atualidade. Ele é a base de diversos tipos de RNA com aprendizado supervisionado utilizando uma adaptação por correção de erros (usualmente baseada na descida da superfície de erro usando o gradiente). O modelo do Perceptron de múltiplas camadas (MLP – *Multi-Layer Perceptron*) tornou-se muito conhecido e aplicado, sendo na maior parte das vezes associado a regra de aprendizado do Back-Propagation [Jodoin 94, Widrow 90, Rumelhart 86]. A figura 2.6 apresenta um esquema da representação de conhecimentos nas redes baseadas em Perceptrons, e como este tipo de redes é capaz de classificar padrões, gerando planos (ou hiper-planos) de divisão do espaço em que se situam os exemplos.

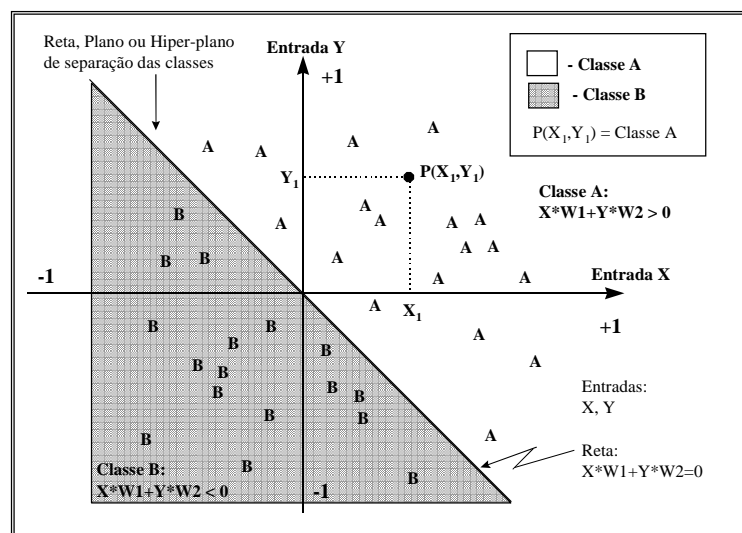


Figura 2.6. Separação de classes (classificação) através do uso de um Perceptron

## 2.3.2.3. Tipos de Arquiteturas de Conexão das Redes

As unidades de uma rede neural podem se conectar de diferentes modos, resultando em diferentes arquiteturas de interconexão de neurônios. A figura 2.7 apresenta alguns exemplos de possíveis maneiras de conectar os componentes de uma RNA. As arquiteturas de redes mais importantes são:

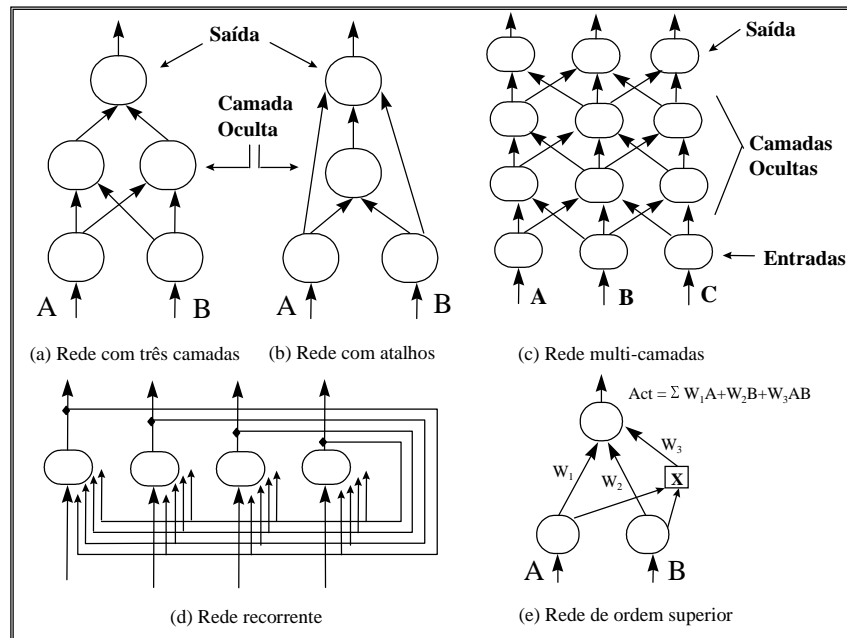


Figura 2.7. Arquiteturas de interconexão de neurônios em redes

- **Redes com uma única camada:** as unidades estão todas em um mesmo nível. Neste tipo de arquitetura, as unidades são conectadas diretamente às entradas externas e estas unidades servem também de saídas finais da rede. As redes de uma única camada possuem normalmente conexões laterais (entre os neurônios de uma mesma camada). Um exemplo deste tipo de arquitetura de redes são as redes do tipo “*Self-Organizing Feature Maps*” [Kohonen 87].
- **Redes com camadas uni-direcionais:** as unidades são organizadas em vários níveis bem definidos, que são chamados de camadas ou *layers*. Cada unidade de uma camada recebe suas entradas vindas à partir de uma camada precedente, e envia seus sinais de saídas em direção a camada seguinte. Estas redes são conhecidas como redes *feed-forward*. A Figura 2.7(a) mostra um exemplo de uma rede de três camadas uni-direcionais. Esta arquitetura de três camadas (entrada, camada oculta e saída) é muito usada em aplicações práticas das redes neurais. O modelo MLP [Widrow 90, Rumelhart 86] é composto em geral de uma arquitetura deste tipo, ou seja, com apenas uma camada oculta (*hidden layer*), mas nada nos impede de colocar mais de uma camada oculta entre a camada de entrada e a camada de saída de uma rede. Um outro tipo de interconexão utilizado em redes uni-direcionais são os atalhos (*short-cuts*) que permitem a conexão de uma unidade à outra em uma camada posterior, passando por cima de outras camadas intermediárias. O uso desta técnica vai nos permitir “saltar” por cima de uma camada até uma outra camada (vide figura 2.7(b)), à condição de não introduzir uma recorrência na rede, o que descaracterizaria esta rede como sendo do tipo feed-forward.

- **Redes recorrentes:** as redes recorrentes podem ter uma ou mais camadas, mas a sua particularidade reside no fato de que temos conexões que partem da saída de uma unidade em direção a uma outra unidade da mesma camada ou de uma camada anterior à esta. Este tipo de conexões permitem a criação de modelos que levam em consideração aspectos temporais e comportamentos dinâmicos, onde a saída de uma unidade depende de seu estado em um tempo anterior. Os laços internos ao mesmo tempo que dão características interessantes de memória e temporalidade as redes, tornam este tipo de redes muito instáveis, o que nos obriga a usar algoritmos específicos (e usualmente mais complexos) para o aprendizado destas redes. Um tipo particular de redes recorrentes são as redes totalmente conectadas, e um exemplo de modelo recorrente de uma única camada e totalmente conectado são as redes de Hopfield, representadas na figura 2.7(d).
- **Redes de ordem superior:** as unidades deste tipo de rede permitem a conexão direta entre duas ou mais de suas entradas, antes mesmo de aplicar a função de cálculo da ativação da unidade [Fiesler 94a]. Este tipo de rede serve para modelar “sinapses de modulação”, ou seja, quando uma entrada pode modular (agir sobre) o sinal que vem de uma outra entrada. Um modelo particular de rede de ordem superior são as redes tipo *Sigma-Pi* que foram apresentadas no livro PDP – Parallel Distributed Processing [Rumelhart 86], e que são representadas na figura 2.7(e).

A arquitetura de uma rede também pode ser classificada de acordo com a evolução desta no decorrer de sua utilização e desenvolvimento do aprendizado. Em função deste critério podemos ter os seguintes grupos:

- **Redes com estrutura estática:** a rede tem a sua estrutura definida antes do início do aprendizado. A quantidade de neurônios, assim como a sua estrutura de interconexões, não sofrem alterações durante a adaptação da rede. As únicas mudanças se realizam à nível dos pesos sinápticos, que são modificados durante o processo de aprendizado. Este tipo de modelo impõe uma dificuldade maior ao usuário: a determinação do número ideal de neurônios e de conexões a ser utilizado em uma determinada aplicação. Uma rede com poucas unidades e conexões tem forte chance de não ter sucesso em uma tarefa de aprendizado, não tendo condições de alcançar o melhor desempenho possível por falta de capacidade de representação de todos os conhecimentos envolvidos no problema tratado. Uma rede com muitas unidades pode ter também problemas de convergência e principalmente de generalização, pois quando se tem muita capacidade de armazenamento de informações em uma rede, esta tem uma tendência a decorar os exemplos no lugar de “aprendê-los” (generalizar os conhecimentos sobre o problema) [Fiesler 97, Krogh 95]. No caso deste tipo específico de redes, não existe um método formal que permita determinar o número exato e ótimo de unidades e conexões à serem empregadas no aprendizado de um determinado problema. As redes do tipo MLP com Back-Propagation, de acordo com o modelo proposto por Rumelhart, são redes do tipo estático.
- **Redes com estrutura dinâmica:** as redes que possuem uma estrutura dinâmica são redes onde o número de unidades e conexões pode variar no decorrer do tempo. Estas redes são também chamadas de *ontogênicas* [Fiesler 94b]. As modificações na estrutura da rede podem ser do tipo generativo (incremental) ou do tipo destrutivo (reduzidor por eliminação/simplificação). A escolha entre estes dois tipos de métodos é bastante polêmica: devemos começar com uma rede pequena e ir aumentando ela, ou devemos começar com uma rede bastante grande e ir reduzindo o seu tamanho posteriormente? Alguns autores defendem a idéia de uma criação construtiva de conhecimentos [Elman 93, Osório 98]. Do ponto de vista relacionado à carga de

processamento de dados necessária para as simulações neurais, a opção por uma rede pequena que adiciona pouco à pouco novas unidades e conexões é sem dúvida a de melhor performance, pois nas redes do tipo destrutivo uma grande parte do esforço de aprendizado acaba sendo depois destruído ao ser realizada a simplificação da rede. Apesar desta discussão, sobre qual dos dois tipos de redes com estrutura dinâmica que seria melhor usar, não possuir um consenso, podemos dizer que uma grande parte dos pesquisadores concorda que as redes ontogênicas em geral são um dos melhores métodos que existem para se escolher uma boa arquitetura para uma rede neural e assim resolver melhor um certo problema proposto. As redes do tipo Cascade-Correlation (CasCor [Fahlman 90]) são redes do tipo dinâmico e incremental.

O último ponto relevante que vamos abordar em relação a arquitetura das redes neurais está relacionado à modularidade [Ronco 96, Ronco 95, Amy 96, Rouzier 98, Jacobs 91]. As redes neurais podem trabalhar com arquiteturas modulares: elas podem ser constituídas por blocos com uma maior ou menor dependência entre eles. Existem diferentes maneiras de integrar e fazer cooperar os diferentes módulos de uma rede neural. Um primeiro método consiste em decompor o problema e obter assim módulos especializados para cada sub-problema. Um exemplo de aplicação deste tipo de método é o caso das aplicações de classificação em múltiplas classes, onde o problema de identificação de cada classe pode ser tratado por módulos separados, e então no lugar de ter um único classificador para os  $N$  exemplos em  $M$  classes, temos um classificador para cada uma das  $M$  classes. Outro tipo de método usado pelas redes modulares, mas mais complexo de ser implementado, é aquele onde os diferentes módulos vão tentar cooperar entre si a fim de juntos resolverem um problema. Neste tipo de método não são impostas tarefas particulares à módulos pré-especificados, deixando para a rede a tarefa de distribuir os conhecimentos e gerenciar a interação entre os módulos.

A modularidade é um problema relativo à escolha de uma arquitetura de rede, mas ela também pode ser ligada ao problema de particionamento dos dados de aprendizado (em um esquema semelhante ao usado na aprendizagem ativa, onde cada módulo poderia escolher que informações iria tratar). Para concluir, devemos salientar que a modularidade pode se tornar um aspecto muito importante a ser considerado segundo o tipo e a complexidade do problema à ser tratado.

#### 2.3.2.4. Tipos de Aplicações das Redes Neurais

As RNA podem ser aplicadas à diferentes tipos de tarefas, tais como: o reconhecimento de padrões (e.g. reconhecimento de faces humanas), a classificação (e.g. reconhecimento de caracteres - OCR), a transformação de dados (e.g. compressão de informações), a predição (e.g. previsão de séries temporais, como as cotações da bolsa de valores, ou também, uso para o diagnóstico médico), o controle de processos e a aproximação de funções (e.g. aplicações na área da robótica). Um grande número de exemplos de aplicações pode ser encontrado no *UCI-ML repository* [UCI 99]. Todas estas tarefas podem ser reagrupadas em dois grupos principais, segundo o tipo de saída fornecido pela rede neural e o comportamento que é buscado. Estes dois grupos são:

- **Redes para a aproximação de funções:** este tipo de redes devem ter uma saída com valores contínuos e usualmente são empregadas para realizar aproximações de funções (interpolação) [Chentouf 97]. Neste tipo de aplicações, as funções são representada por um conjunto de pontos-exemplo desta. Este tipo de redes é capaz de aprender uma função de transformação (ou de associação) de valores de entrada em valores de saída, usualmente estimando por interpolação as respostas para os casos que não aparecem na base de exemplos. Este tipo de problemas de aprendizado neural de funções é conhecido por ser uma aplicação de um *problema de regressão* [Bishop 97]. Em geral as funções à serem aprendidas pelas redes possuem tanto as entradas como as saídas indicadas através de valores contínuos (variáveis não discretas).

- **Redes para a classificação de padrões:** este tipo de redes deve atribuir para cada exemplo que lhe é fornecido uma classe ao qual este exemplo pertence. Portanto, a saída da rede é a classe associada ao exemplo e por consequência, as classes são valores discretos e não contínuos. A classificação é um caso particular da aproximação de funções onde o valor de saída da rede é discretizado e pertence a um conjunto finito de classes. No caso do aprendizado supervisionado, o conjunto de classes é bem definido e conhecido antes de ser iniciado o processo de aprendizado. Uma rede utilizada para fins de classificação deve possuir saídas discretas, ou então, deve implementar métodos de discretização de suas saídas (e.g. aplicação de um limiar de discriminação – *activation threshold*). As entradas da rede podem ser tanto contínuas, como também podem ser discretas, o que não deve interferir no fato desta rede ser usada para uma aplicação de classificação.

Seria muita pretensão de nossa parte se tentássemos classificar todos os diferentes modelos de redes neurais em apenas uma destas duas classes descritas acima. A maioria dos modelos pode ser adaptado para ser utilizado em um ou em outro tipo de aplicação, entretanto, alguns modelos são claramente mais adaptados a um tipo de tarefa que ao outro, como é o caso do Cascade-Correlation que foi desenvolvido basicamente apenas para tarefas de classificação.

## 2.5. Discussão sobre as Redes Conexionistas

As redes conexionistas utilizam métodos de aprendizado à partir de exemplos que possibilitam o ajuste dos pesos de suas conexões, resultando em um comportamento próximo ou até mesmo exatamente igual ao esperado. Esta modificação dos pesos da rede é feita de forma que a rede generalize os conhecimentos contidos na base de exemplos de aprendizado.

Uma boa definição das redes conexionistas é dada por Giacommetti [Giacometti 92]: as redes são capazes de aprender e representar o “saber fazer algo” (*savoir-faire*), que se traduz pelo seu comportamento após o processo de aprendizado de uma tarefa; este “saber fazer”, representado pelos **conhecimentos práticos** (*empirical knowledge ~ practical examples*) adquiridos pela rede, aparece aqui em oposição ao “saber sobre algo” (*savoir-que*) que representa os **conhecimentos teóricos** sobre um determinado assunto (*theoretical knowledge ~ symbolic rules*). Esta distinção entre o saber fazer uma tarefa e o conhecimento sobre a tarefa, é um dos pontos mais importantes da discussão sobre os sistemas híbridos, pois ambos os conhecimentos se completam um ao outro.

Antes de prosseguir em nossa análise sobre as redes conexionistas, cabe ressaltar que não seria possível fazer aqui uma análise relativa a *todos* os modelos de redes neurais, por isso nosso estudo se concentra principalmente em dois tipos principais de redes: as redes à base de protótipos com aprendizado supervisionado e as redes à base de Perceptrons (MLP) com aprendizado supervisionado. Nosso objetivo é fazer uma discussão orientada principalmente para estes dois tipos de redes, dada a sua importância. Devemos ressaltar também que nosso interesse é voltado às redes usadas para aplicações de classificação, visto que nosso estudo aborda aplicações do tipo sistemas especialistas, onde a aproximação de funções não é uma característica típica deste tipo de sistemas.

As redes conexionistas, em particular aquelas que tem sido aplicadas na construção de sistemas inteligentes, possuem as seguintes vantagens:

- **Conhecimento empírico:** o aprendizado à partir de exemplos é feito de uma maneira bastante simples e permite uma aquisição de conhecimentos de forma automática, muitas vezes de maneira bem mais fácil e confiável do que através de outros métodos de aquisição de conhecimentos (ajuda a resolver em parte o problema do “gargalo da aquisição de conhecimentos” bem conhecido dos sistemas especialistas);
- **Degradação progressiva:** as respostas dadas por uma rede se degradam progressivamente na presença de “perturbações e distorções” dos dados de entradas. Em geral, as redes obtêm uma boa generalização dos conhecimentos presentes na base de aprendizado e sendo assim são menos sensíveis a “perturbações” do que os sistemas simbólicos;
- **Manipulação de dados quantitativos:** o fato de se trabalhar com uma representação numérica dos conhecimentos implica que as redes são melhor adaptadas para a manipulação de dados quantitativos (valores contínuos). Grande parte dos problemas de nosso mundo real, necessitam do tratamento de informações medidas de forma quantitativa, onde uma representação qualitativa muitas vezes implica na perda de informações. As redes neurais são menos vulneráveis aos dados aproximativos e a presença de dados distorcidos ou incorretos que possam estar presentes na base de aprendizado. Esta capacidade de manipular dados aproximados e até mesmo inexatos é mais difícil de ser encontrada em outros métodos de aprendizado do tipo simbólico;
- **Paralelismo em larga escala:** as redes neurais são compostas de um conjunto de unidades de processamento de informações que podem trabalhar em paralelo. Apesar da maioria das implementações de RNAs serem feitas através de simulações em máquinas seqüenciais, é possível de se implementar (softwares e hardwares) que possam explorar esta possibilidade de ativação simultânea das unidades de uma rede. A maior parte das implementações de redes neurais simuladas em máquinas seqüenciais pode ser facilmente adaptada em uma versão paralela deste sistema.

As redes conexionistas apresentam um certo número de inconvenientes, do mesmo modo que os outros tipos de métodos de aprendizado. No caso específico das redes, temos limitações tais como:

- **Arquitetura e parâmetros:** não existe um método totalmente automático para que se possa escolher a melhor arquitetura possível para um problema qualquer. É bastante difícil de se encontrar uma boa topologia de uma rede, assim como os bons parâmetros de regulação do algoritmo de aprendizado. A evolução do processo de aprendizado é bastante influenciada por estes dois elementos: a arquitetura da rede e os parâmetros de regulação do algoritmo. O sucesso da rede depende bastante de uma boa escolha destes elementos, que variam muito de um problema para outro. Uma simples troca do conjunto de exemplos de uma base de aprendizado pode nos obrigar a reconfigurar toda a rede;
- **Inicialização e codificação:** os algoritmos de aprendizado conexionista são em geral muito dependentes do estado inicial da rede (devido à inicialização aleatória dos pesos) e da codificação dos dados da base de aprendizado. Uma má escolha dos pesos iniciais da rede, do método de codificação dos dados de entrada, ou mesmo, a ordem de apresentação destes dados, pode levar ao bloqueio do processo de aprendizado (e seu conseqüente fracasso), ou então, pode dificultar bastante o processo de convergência da rede na direção de uma boa solução;

- **Caixa preta:** os conhecimentos adquiridos por uma rede estão codificados no conjunto de valores dos pesos sinápticos, assim como pela maneira pela qual estas unidades se conectam. É extremamente difícil para um ser humano conseguir interpretar diretamente estes conhecimentos. As redes conexionistas são “caixas pretas” onde os conhecimentos ficam codificados de tal forma que estes são ininteligíveis para o utilizador ou até mesmo para um especialista. Uma rede não possui a capacidade de explicitar o tipo de raciocínio que lhe levou a obter uma certa resposta, ao contrário dos sistemas baseados em regras, que por sua vez podem facilmente mostrar a sequência de regras aplicadas na resolução de um problema;
- **Conhecimentos teóricos:** as redes neurais clássicas não permitem que se utilize os conhecimentos teóricos que possam estar disponíveis sobre um determinado problema que estejamos tratando. Como as árvores de decisão, as redes neurais são orientadas para a aquisição de conhecimentos empíricos (baseados em exemplos). Um modo simplista de se aproveitar algum conhecimento teórico pré-existente, consiste em se converter regras em exemplos (“protótipos” representativos destas regras). Entretanto, este tipo de método não nos garante que a rede será capaz de aprender corretamente estes exemplos, sendo assim, não podemos garantir que ao final do aprendizado todos os conhecimentos teóricos disponíveis estarão bem representados internamente na rede.

Estes tópicos listados acima não cobrem exhaustivamente todas vantagens e desvantagens das redes conexionistas, mas permitem que se tenha uma idéia das principais características deste tipo de sistemas. Podemos encontrar uma análise complementar a citada acima em outras obras da área, como por exemplo [Orsier 95, Towell 91, Fahlman 88]. No que diz respeito mais especificamente ao aprendizado usando redes neurais baseadas em MLP com Back-Propagation (um dos modelos de redes neurais mais utilizados na atualidade), podemos listar alguns dos pontos inconvenientes deste modelo:

- **Paralisia do aprendizado:** as redes do tipo MLP com Back-Propagation, devido à maneira como o algoritmo ajusta os pesos, tem a tendência à não mais corrigir estes pesos uma vez que a saída das unidades da rede forneçam valores próximos à 0 ou à 1 (isto se deve ao uso da aplicação da sigmoide e de sua derivada pelo algoritmo Back-Propagation). Este comportamento do algoritmo de aprendizado permite dar uma maior estabilidade ao processo de adaptação dos pesos, mas em compensação pode paralisar o aprendizado da rede. Este problema também é conhecido como o “*flat spot problem*”.
- **Instabilidade e esquecimento catastrófico:** de maneira inversa ao problema da paralisia, as redes também podem sofrer de um problema de instabilidade com a conseqüente perda dos conhecimentos anteriormente adquiridos. Uma vez que as redes realizam a minimização do erro de uma maneira não coordenada, ou seja, as unidades competem entre si a fim de reduzir o erro, isto pode levar a uma constante concorrência entre as unidades. Não importa se uma unidade se adaptou a fim de realizar uma pequena, mas importante tarefa, esta unidade vai continuar sempre tentando alterar os seus pesos a fim de minimizar o erro global ao máximo possível. Este tipo de “comportamento competitivo” pode nos levar à duas situações: (1) as unidades mudam constantemente de “opinião” durante o aprendizado (conhecido como o *moving target problem*), ou, (2) a rede perde grande parte dos conhecimentos já adquiridos ao tentarmos aprender um novo conjunto de exemplos de aprendizado (conhecido como o esquecimento catastrófico). Portanto este problema envolve a busca de um ponto de equilíbrio entre a grande plasticidade (capacidade de se adaptar) e a estabilidade (necessidade de manter as informações) das redes neurais. Redes neurais com uma grande plasticidade estão sujeitas a ficarem alterando indefinidamente os seus pesos, ou então, destruir uma boa configuração de pesos ao tentar adquirir novos conhecimentos.

- **Escolha dos parâmetros do algoritmo de aprendizado e a velocidade de convergência:** na maior parte das aplicações, a velocidade de convergência de uma rede em direção à um mínimo (local ou global) de erro é realizada de maneira muito lenta. As alterações dos pesos da rede devem ser feitas pouco à pouco de modo a garantir que não se “ultrapasse” o ponto ótimo de mínimo da curva de erro. O algoritmo de Back-Propagation é dotado de dois parâmetros –  $\alpha$  e  $\beta$  – que controlam respectivamente a velocidade de aprendizado (*learning speed*) e a inércia na descida da curva de erro (*momentum*). Estes dois parâmetros permitem que o processo de adaptação dos pesos da rede seja acelerado ou retardado, mas é preciso que eles sejam ajustados precisamente para que se obtenha bons resultados. Estes dois parâmetros, que devem ser fornecidos pelo usuário, são essenciais para o bom desempenho do processo de aprendizagem. O problema é que não possuímos métodos precisos de estimar estes valores. Além disso, os valores de  $\alpha$  e  $\beta$  são bastante dependentes do tipo de aplicação e da base de exemplos de aprendizado utilizada, devendo ser reconfigurados novamente caso o problema tratado seja alterado. Um valor de  $\alpha$  ou  $\beta$  que não seja muito bem escolhido pode levar ao fracasso toda a tentativa de se aprender uma base de dados. É por isso que normalmente o aprendizado de uma base de exemplos é feito com o uso de N conjuntos de configurações de parâmetros do algoritmo, para que se possa ter uma melhor chance de encontrar os valores adequados destes parâmetros.

Para concluir sobre as redes MLP, o algoritmo de Back-Propagation não é um algoritmo incremental, nem ao nível da base de exemplos, e muito menos ao nível da estrutura da rede. A arquitetura da rede é estática e conseqüentemente este continua à ser um problema a mais no que se refere ao aprendizado: como fazer para estimar o número ideal de neurônios para uma dada aplicação?

Apesar de todos estes problemas, o algoritmo Back-Propagation ainda é um dos métodos mais usados junto as redes neurais. Alguns pesquisadores, conscientes dos problemas deste algoritmo, propuseram técnicas para resolver ou reduzir estes problemas [Schiffmann 93, 94]. Podemos citar aqui alguns destes métodos aperfeiçoados de aprendizado: o RPROP [Riedmiller 93], o QuickProp [Fahlman 88], o Gradiente Conjugado (Scaled Conjugated Gradient) [Moller 90] e o Cascade-Correlation [Fahlman 90], bem como as técnicas ontogênicas de aprendizado [Fiesler 94].

## 2.6. Redes Neurais: A busca de uma solução ótima

As redes neurais possuem algumas limitações e problemas que citamos na seção anterior. Vamos listar aqui alguns pontos que devem ser considerados e discutidos no que se refere as redes neurais e a busca de uma solução ou melhoria do aprendizado neural:

- *Aprendizado incremental:* a rede neural deve ser capaz de adquirir novos conhecimentos, sem no entanto destruir os conhecimentos anteriormente adquiridos. A rede neural deve ser também capaz de adequar a sua estrutura aos requisitos do problema, sendo que esta deveria poder aumentar de tamanho (quantidade de neurônios) à medida que fosse aumentando a complexidade do problema tratado;
- *Estimativa da topologia:* deve-se buscar métodos que permitam ao usuário criar uma topologia de rede adequada para tratar um determinado problema. O ideal seria dotar as redes neurais de mecanismos que possibilitem que esta ajuste automaticamente a sua estrutura em função do problema tratado;



- *Estimativa dos parâmetros de aprendizado*: um bom algoritmo de aprendizado não deve ser muito dependente de parâmetros externos, e idealmente, a rede deveria poder de maneira automática ajustar todos os seus parâmetros para conseguir um resultado ótimo no aprendizado;
- *Introdução de conhecimentos a priori*: as redes neurais devem permitir que *conhecimentos a priori* sobre o problema possam ser inseridos de maneira a facilitar e adiantar o aprendizado de um determinado problema;
- *Instabilidade e velocidade*: um bom algoritmo de aprendizado deve ser o menos instável possível, com ótimas chances de adaptar os pesos da rede e convergir em direção a uma boa solução, minimizando o mais possível o erro na saída da rede. Este algoritmo deve permitir um aprendizado rápido e eficiente;
- *Abrir a “caixa preta”*: devemos buscar uma solução para o problema da falta de mecanismos para analisar os conhecimentos adquiridos pela rede. Devemos ser capazes de representar os conhecimentos adquiridos pela rede em um formato mais compreensível para os seres humanos;
- *Aprendizado ativo*: a rede neural deve ser capaz de dar um retorno sobre o processo de aprendizado, indicando quais os exemplos que devem ser tratados com uma maior prioridade, ou até mesmo, indicando a necessidade de mais exemplos de uma categoria específica para que o problema possa ser corretamente tratado;
- *Tratamento de informações temporais e contexto*: as redes neurais devem ser capazes de considerar o contexto (possuir memória) e assim poder também tratar informações que evoluem no decorrer do tempo, como por exemplo as séries temporais. As redes recorrentes parecem ser um caminho importante a seguir nesta direção, mas ainda restam problemas a serem resolvidos no que se refere a instabilidade e confiabilidade deste tipo de redes.

Estes itens citados acima seguem sendo pesquisados atualmente, e muitas propostas tem sido apresentadas a fim de solucionar (ou minimizar) os problemas ainda enfrentados pelas redes neurais. Apesar de termos problemas relacionados ao aprendizado neural sem serem completamente solucionados, este tipo de técnica tem adquirido uma importância cada vez maior junto à aplicações que necessitem de uma aquisição automática de conhecimentos. As redes neurais superam em muitos casos os demais métodos automáticos de aquisição de conhecimentos.

### **3. Conclusão e Perspectivas**

Neste trabalho apresentamos uma visão geral sobre os sistemas de I.A. e a necessidade do aprendizado para que um sistema inteligente possa ser considerado como tal. Enfocamos o aprendizado neural como sendo uma forma de aquisição de conhecimentos, que dadas as suas peculiaridades, possui um interesse particular na área de inteligência Artificial.

As principais características consideradas foram: a representação de conhecimentos, o paralelismo inerente as unidades da rede, a sua capacidade de adaptação, entre outros aspectos. As redes neurais também se apresentam como uma alternativa ao processamento simbólico de informações, podendo manipular informações do tipo quantitativo e qualitativo sem maiores problemas. Entretanto as redes neurais possuem ainda alguns pontos fracos a serem estudados, principalmente no que diz respeito a explicitação dos conhecimentos adquiridos e na dificuldade de convergência em relação a uma solução ótima.

Procuramos apresentar neste trabalho uma visão bastante ampla do assunto, levantando questionamentos e pontos em aberto para estudos futuros, de forma que o leitor possa ter ao mesmo tempo uma visão global da área, e também uma noção dos temas de pesquisa na atualidade neste domínio. Este trabalho visa ser uma fonte de questionamentos e idéias para novos trabalhos, onde a extensa relação de bibliografias remetem o leitor as demais obras da área que podem complementar os temas aqui abordados.

Concluindo, acreditamos que as pesquisas futuras nos levam em direção aos sistemas com múltiplas formas de aquisição e representação de conhecimentos, onde os sistemas híbridos, sistemas multi-agentes e sistemas com múltiplas inteligências são uma tendência. Devemos buscar a integração dos métodos simbólicos com os métodos conexionistas de forma a expandir as potencialidades dos “sistemas inteligentes”, para quem sabe assim poderemos realmente ter sistemas com as características (múltiplas!) relacionadas a inteligência que foram levantadas na primeira seção deste trabalho.

## APÊNDICES

### A.1. Exemplo de Conhecimentos Teóricos

If ((A or B) and Not(A and B)) Then Xor=1 Else Xor=0	Regras de Produção
---------------------------------------------------------	-----------------------

$XOR = (A \cdot \bar{B}) + (\bar{A} \cdot B)$	Fórmula Lógica
-----------------------------------------------	----------------

$XOR = (A \text{ Or } B) \text{ And Not } (A \text{ And } B)$ <p style="text-align: center; margin: 0;"><i>ou</i></p> $XOR = (A \text{ And Not } (B)) \text{ Or } (\text{Not } (A) \text{ And } B)$
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### A.2. Exemplo de Conhecimentos Práticos (Exemplos)

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

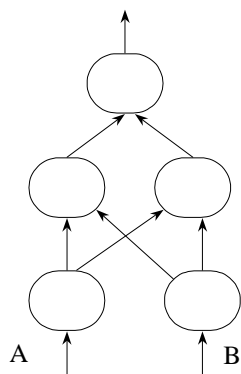
⇒ Conhecimentos Empíricos

Entradas : A e B

Saída : XOR (saída usada no aprendizado supervisionado)

Número de exemplos : 4

### A.3. Rede Neural – Solução para o problema do XOR



Topologia da rede neural usada no aprendizado do XOR

## **BIBLIOGRAFIA**

[Amy 96] AMY, Bernard. *Recherches et Perspectives dans le Domaine des Réseaux Connexionnistes*. Rapport de Recherche - DRET, Octobre 1996. Web: <http://www-leibniz.imag.fr/RESEAUX/public.html>  
Ftp: <ftp://ftp.imag.fr/pub/LEIBNIZ/RESEAUX-D-AUTOMATES/>

[Anderson 93] ANDERSON, J. R. *Rules of the Mind*. Erlbaum Ed., Hillsdale, NJ. 1993.

[Arbib 95] ARBIB, Michael A. (Editor) *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1995.

[Bishop 97] BISHOP, Christopher. *Classification and Regression*. In: Handbook of Neural Computation (section B6.2). E. Fiesler and R. Beale (Eds.) Institute of Physics and Oxford University Press. New York, NY - U.S.A., 1997. Web: <http://www.idiap.ch/publications/fiesler-96.1.bib.abs.html> or [http://www.oup-usa.org/acadref/nc\\_accs.html](http://www.oup-usa.org/acadref/nc_accs.html)

[Bishop 95] BISHOP, C.M. *Neural Networks for Pattern Recognition*, Oxford: Oxford University Press. ISBN 0-19-853849-9. 482 pages.1995.

[Carpenter 83] CARPENTER, G. & GROSSBERG, S. *A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine*. Computer Vision, Graphics and Image Processing, v.37, p.54-115, 1983.

[Caudill 92] CAUDILL, Maureen & BUTLER, Charles. *Understanding Neural Networks - Vol.1: Basic Networks, Vol.2: Advanced Networks*. Bradford Books, MIT Press, 1992.

[Chentouf 97] CHENTOUF, Rachida. *Construction de Réseaux de Neurones Multicouches pour l'Approximation*. Thèse de Doctorat en Sciences Cognitives, Laboratoire TIRF - INPG, Grenoble - France, Mars 1997.

[Elman 93] ELMAN, Jeffrey L. *Learning and Development in Neural Networks: The Importance of Starting Small*. Cognition, 48(1993), pp.71-99. 1993. Web: <http://crl.ucsd.edu/~elman/>  
Ftp: <ftp://crl.ucsd.edu/pub/neuralnets/cognition.ps.Z>

[Fahlman 88] FAHLMAN, Scott E. *An Empirical Study of Learning Speed in Back-Propagation Networks*. Carnegie Mellon University - CMU. Computer Science Technical Report CMU-CS-88-162. September 1988. Web: <http://www.cs.cmu.edu/Reports/index.html>

[Fahlman 90] FAHLMAN, S. E.; LEBIERE, C. *The Cascade-Correlation Learning Architecture*. Carnegie Mellon University - CMU, Computer Science Technical Report - CMU-CS-90-100. February 1990. Web: <http://www.cs.cmu.edu/Reports/index.html>  
Ftp: <ftp://archive.cis.ohio-state.edu/pub/neuroprose/fahlman.cascor-tr.ps.Z>

[Faq 99] FAQ. *Faq ANN - Comp.ai.neural-nets* 1999.  
Web: <http://www.cis.ohio-state.edu/hypertext/faq/usenet/ai-faq/neural-nets/>

[Faq 99a] FAQ. *Faq ANN - Comp.ai.neural-nets FAQ*, Part 4 of 7: Books, data, etc. 1999.

Web: <http://www.cis.ohio-state.edu/hypertext/faq/usenet/ai-faq/neural-nets/part4/faq-doc-1.html>

[Fiesler 94a] FIESLER, E. *Neural Networks Formalization and Classification*. Computer Standard & Interfaces, Special Issue on Neural Networks Standards, John Fulcher (Ed.). V.16, N.3. Elsevier Sciences Publishers, Amsterdam, June, 1994. Web: <http://www.idiap.ch/idiap-networks.html> .

Ftp: <ftp://ftp.idiap.ch/pub/papers/neural/fiesler.formalization.ps.Z>

[Fiesler 94b] FIESLER, Emile. *Comparative Bibliography of Ontogenic Neural Networks*. Proceedings of the International Conference on Artificial Neural Nets - ICANN'94. Sorrento, Italy, May 1994. Web: <http://www.idiap.ch/idiap-networks.html> .

Ftp: <ftp://ftp.idiap.ch/pub/papers/neural/fiesler.ontogenic-summary.ps.Z>

[Fiesler 97] FIESLER, E. & BEALE, R. *Handbook of Neural Computation*. Institute of Physics and Oxford University Press. New York, NY - U.S.A., 1997. Web: <http://www.idiap.ch/publications/fiesler-96.1.bib.abs.html> or [http://www.oup-usa.org/acadref/nc\\_accs.html](http://www.oup-usa.org/acadref/nc_accs.html)

[Freeman 92] FREEMAN, James A., SKAPURA, David M.. *Neural Networks: Algorithms, Applications, and Programming Techniques*. 1. ed., Reading: Addison-wesley, 1992. 401 p. il.

[Giacometti 95] GIACOMETTI, Arnaud. *ARN2 - A Prototype-Based Incremental Neural Network*. WNDA - Workshop on Neural Network Design and Analysis. University of Geneva. January 1995.

[Goldberg 89] GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing. 1989.

[Hebb 49] HEBB, D. O. *The Organization of Behaviour*. John Wiley & Sons. New York, 1949.

[Hinton 84] HINTON, G.; SEJNOWSKI, T; ACKLEY D. *Boltzmann Machines: Constraint Satisfaction Networks that Learn*. Carnegie-Mellon University – Tech. Report CMU-CS-84-119. 1984.

[Hopfield 82] HOPFIELD, J. J. *Neural Networks and Physical Systems with Emergent Computational Abilities*. In: Proceedings of the National Academy of Sciences, v.79, Washington, USA. p.2554-2558, April 1982.

[Jacobs 91] JACOBS, R. A., JORDAN, M. I., NOWLAN, S., and HINTON, G. E. (1991). *Adaptive mixtures of local experts*. In *Neural Computation*, (3), 1-12.

Web: <http://www.ai.mit.edu/projects/cbcl/people/jordan/jordan-hp.html>

[Jodouin 94a] JODOUIN, Jean-François. *Les Réseaux de neurones : Principes et définitions*. Editions Hermès, Paris, 1994.

[Jodouin 94b] JODOUIN, Jean-François. *Les Réseaux Neuromimétiques : Modèles et applications*. Editions Hermès, Paris, 1994.

[Kohonen 82] KOHONEN, Teuvo. *Self-Organized formation of Topologically Correct Feature Maps*. *Biological Cybernetics*, v.43, p.32-48. Jan. 1982.

[Kohonen 87] KOHONEN, Teuvo. *Self-Organization and Associative Memory*. Springer-Verlag Series in Information Science. 1987.

[Kolodner 93] KOLODNER, J. *Case-Based Reasoning*. Morgan Kaufmann Series in Representation and Reasoning, San Mateo, CA. 1993.

[Kosko 87] KOSKO, Bart. *Adaptive Bidirectional Associative Memories*. Applied Optics, v.26, p.4947-4960. Dec. 1987.

[Kosko 87a] Kosko, Bart. *Constructing an Associative Memory*. Byte, Highstown, v. 12, n.10, p.137-144. Sept. 1987.

[Krogh 95] KROGH, Anders & VEDELSBY, Jesper. *Neural Network Ensembles, Cross Validation and Active Learning* in NIPS - Advances in Neural Information Processing Systems, Vol. 7, pp. 231-238, The MIT Press, 1995), also available in the neuroprose archive as krogh.ensemble.ps.Z.

[Krose 93] KRÖSE, Ben J. & VAN DER SMAGT, Patrick. *An Introduction to Neural Networks*. University of Amsterdam, 1993. Web: <http://www.fwi.uva.nl/research/ias/> ou <ftp://ftp.wins.uva.nl/pub/computer-systems/aut-sys/reports/neuro-intro/>

[Larousse 99] LAROUSSE. Grande Enciclopédia Larousse Cultural. Editora Nova Cultural, 1999.

[McCulloch 43] McCULLOCH, W. S. & PITTS, W. *A Logical Calculus of the Ideas Imminent in Nervous Activity*. Bulletin of Mathematical Biophysics, 5, p.115-133, 1943.

[Minsky 85] MINSKY, Marvin. *The Society of Mind*. Simon & Schuster, New York, 1985. (possui uma edição traduzida para o português).

[Minsky 69] MINSKY, M. & PAPERT, S. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge. 1969.

[Minsky 90] MINSKY, Marvin. *Logical vs. Analogical or Symbolic vs. Connectionist or Neat vs. Scruffy*. In: Artificial Intelligence at MIT \_ Expanding Frontiers, P. Winston (ED.), vol 1. MIT Press (reprinted in AI Magazine). 1990. Web: <http://www.ai.mit.edu/people/minsky/minsky.html>

[Mitchell 97] MITCHELL, Tom M. *Machine Learning*. McGraw-Hill, 1997. Web: <http://www.cs.cmu.edu/afs/cs.cmu.edu/user/mitchell/ftp/tomhome.html>

[Moller 90] MOLLER, Martin F. *A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning*. Technical Report PB-339 - Computer Science Dept., University of Aarhus, Denmark. November 1990. Ftp: <ftp://archive.cis.ohio-state.edu/pub/neuroprose/moller.conjugate-gradient.ps.Z>

[Nilsson 98] NILSSON, Nils J. *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann Publishers. 1998. Web: <http://robotics.stanford.edu/people/nilsson/mlbook.html>

[Nikolopoulos 97] NIKOLOPOULOS, Chris. *Expert Systems – Introduction to First and Second Generation and Hybrid Knowledge Based Systems*. Marcel Dekker Inc. Press. 1997.

[Orsier 94] ORSIER, Bruno; AMY, Bernard; RIALLE, Vincent & GIACOMETTI, Arnaud. *A study of the hybrid system SYNHESYS*. Workshop ECAI-94 (European Conference on Artificial Intelligence) - Combining Connectionist and Symbolic Processing. Amsterdam, Agosto 1994.

[Orsier 95] ORSIER, Bruno. *Etude et Application de Systèmes Hybrides NeuroSymboliques*. Thèse en Informatique, Laboratoire LIFIA-IMAG, UJF - Grenoble, 1995.

Web: <http://www-leibniz.imag.fr/RESEAUX/public.html> .

Ftp: <ftp://ftp.imag.fr/pub/LEIBNIZ/RESEAUX-D-AUTOMATES/orsier.these.ps.gz>

[Osório 91] OSORIO, Fernando Santos. *Um Estudo sobre Reconhecimento Visual de Caracteres através de Redes Neurais*. Dissertação de Mestrado, CPGCC, UFRGS, Porto Alegre - Brasil. Outubro 1991.

[Osório 92] OSORIO, F. S. *Simulação de Redes Neurais Artificiais de Neurônios com Aprendizado Supervisionado*. Revista Scientia, Unisinos. v.3, n.1, p.45-66. 1992.

[Osório 98] OSORIO, Fernando Santos. *INSS: Un Système Hybride Neuro-Symbolique pour l'Apprentissage Automatique Constructif*. Thèse de Doctorat (Ph.D.) en Informatique. Laboratoire Leibniz – IMAG / INPG. Grenoble, France. 1998.

Web: <http://www-leibniz.imag.fr/RESEAUX/osorio/These/>

[Osório 99] OSORIO, F. S. & VIEIRA, R. Sistemas Híbridos Inteligentes. In: ENIA'99 – Encontro Nacional de Inteligência Artificial (Tutorial). Congresso da SBC'99. Rio de Janeiro, 1999.

Web: <http://www.inf.unisinos.tcche.br/~osorio/enia99/>

[Parker 82] PARKER, D. *Learning-Logic*. Invention Report S81-64, File 1, Office of Technology Licensing, Stanford University, Stanford. Oct. 1982.

[Pinker 99] PINKER, Steve. *Como a Mente Funciona*. Ed. Companhia das Letras, São Paulo, 1999.

[Quinlan 93] QUINLAN, J. R. *C4.5- Programs for Machine Learning*. Morgan Kauffman Publishers. San Mateo, CA. 1993. Web: <http://www.cs.su.oz.au/People/quinlan.html>

[Riedmiller 93] RIEDMILLER, Martin & BRAUN, Heinrich. *A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm*. Proceedings of the IEEE International Conference on Neural Networks. San Francisco - CA - USA. 1993. Web: <http://i11www.ira.uka.de/~riedml/> (or ~neuro)

[Ripley 96] RIPLEY, B.D. *Pattern Recognition and Neural Networks*, Cambridge: Cambridge University Press, ISBN 0-521-46086-7. 403 pages. 1996.

[Ronco 95] RONCO, Eric & GAWTHROP, Peter J., 1995. *Modular Neural Networks: a state of the art*. Tech. rep CSC-95026. University of Glasgow.

Web: <http://www.ee.usyd.edu.au/~ericr/pub/techrep.html>

[Ronco 96] RONCO, Eric; GOELLE, Henrik & GAWTHROP, Peter J., 1996. *Modular Neural Network and Self-Decomposition*. Tech. Rep CSC-96012. University of Glasgow.

Web: <http://www.ee.usyd.edu.au/~ericr/pub/techrep.html>

[Rouzier 98] ROUZIER, Sophie. *Réseaux Modulaires*. Thèse de Doctorat en Sciences Cognitives, Eq. Réseaux d'Automates - Lab. LEIBNIZ - IMAG, Grenoble - France, 1998.

[Rosenblatt 59] ROSENBLATT, R. *Principles of Neurodynamics*. Spartan Books. New York. 1959.

[Rumelhart 85] RUMELHART, D. ; HINTON, G.; WILLIAMS, R. *Learning Internal Representations by Error Propagation*. ICS Report 8506, Institute for Cognitive Science, University of California at San Diego, La Jolla. Sept. 1985.

[Rumelhart 86] RUMELHART, D.; HINTON, G. & WILLIAMS, R. *Learning Internal Representations by Error Propagation*. In : Rumelhart & McClelland: *Parallel Distributed Processing - Explorations in the Microstructure of Cognition - Vol.1: Foundations*. Cambridge: MIT Press, 1986.

[Schiffmann 93] SCHIFFMANN, W.; JOOST, M. & WERNER, R. *Comparison of Optimized Backpropagation Algorithms*. Proceedings of the European Symposium on Artificial Neural Networks, ESANN'93, Brussels, p.97-104, 1993. Web: <http://www.uni-koblenz.de/~schiff/publications.html>

[Schiffmann 94] SCHIFFMANN, W.; JOOST, M. & WERNER, R. *Optimization of the Backpropagation Algorithm for Training Multilayer Perceptrons*. Technical Report, University of Koblenz, Deutschland. September 1995. Web: <http://www.uni-koblenz.de/~schiff/publications.html>

[Simpson 90] SIMPSON, Patrick K. *Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations*. Pergamon Press, 1990.

[Sutton 98] SUTTON, Richard S. & BARTO, Andrew G. *Reinforcement Learning: An Introduction*. MIT Press (A Bradford Book), Cambridge, MA, 1998.

[Towell 91] TOWELL, Geoffrey. *Symbolic Knowledge and Neural Networks: Insertion, Refinement and Extraction*. Ph.D. Thesis, Computer Science Dept., University of Wisconsin-Madison, U.S.A. 1991.

Web: <http://www.cs.wisc.edu/~shavlik/uwml.html>

Ftp: [ftp://ftp.cs.wisc.edu/machine-learning/shavlik-group/\(towell.thesis.\\*.ps\)](ftp://ftp.cs.wisc.edu/machine-learning/shavlik-group/(towell.thesis.*.ps))

[UCI 99] UCI-ML. *Machine Learning Repository*. UCI – University of California Irvine. Dept. of Computer Science. Web: <http://www.ics.uci.edu/~mlearn/MLRepository.html>

[Widrow 62] WIDROW, Bernard. *Generalization and Information Storage in Networks of ADALINE Neurons*. In: *Self-Organization Systems*. Spartan Books. Washington: p.435-461, 1962.

[Widrow 88] WIDROW, Bernard & WINTER, R. *Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition*. IEEE Computer, New York, v.21, n.3, p.25-39, march 1988.

[Widrow 90] WIDROW, Bernard & LEHR, M. *30 Years of Adaptive Neural Networks : Perceptron, Madaline, and Back-Propagation*. Proceedings of the IEEE, New York, Vol.78, N.9, pp.1415-1441. September 1990.