CAPÍTULO 3

Técnicas Subsimbólicas: Redes Neurais

Contribuíram: *Martin Buch, Universidade de Kaiserslautern*

3.1. Filosofia Geral da Aplicação do Raciocínio Subsimbólico a Padrões: Redes Neurais Aprendendo Dados, Classificadores e Agrupadores

Como vimos na berve visão geral do aprendizado em IA fornecida no capítulo anterior, chamamos de Raciocínio Subsimbólico ao processamento de informação em um nível onde os padrões representam conjuntos de dados, mas onde não podemos associar um significado imediato a cada dado ou elemento do padrão processado. Isto significa que trabalhamos com dados, que em seu conjunto podem ser chamados de informação e assumir um significado, mas onde não sabemos ou não podemos determinar o significado de cada parte do conjunto de informação em separado. O mesmo vale para as representações internas, intermediárias desses dados: usamos sistemas que representam essa informação, mas eles são sistemas *black box* - caixa preta - produzindo classificações dos padrões e resultados numéricos, sem no entanto fazer isto de forma explícita. As redes neurais são o melhor exemplo de um sistema subsimbólico: mesmo que cada parte de um padrão apresentado a uma rede tenha um significado explícito, associável a um símbolo de nosso modelo do mundo real, a representação interna dos dados no procesador (a rede) não é explícita e não possui significado. Um método subsimbólico tipicamente é **incapaz** de

explicar porque chegou a uma determinada conclusão, uma vez que um mapeamento explícito de causa-e-efeito não existe.

Fica mais fácil ilustrar este conceito através de um exemplo. Imagine um sistema militar para classificar tipos de navios ("inimigos") a partir do padrão de ruído emitido por estes, da forma como é captado por um submarino ("nosso"). Hipotetize que nós sabemos que um determinado padrão de ruídos corresponde a um porta-aviões de determinada classe de uma determinada nacionalidade. Este padrão inclue um conjunto de freqüências e as variações de amplitude dessas freqüências, além de algumas outras informações. Nós podemos associar um conjunto de símbolos a esse padrão: "Porta-Aviões modelo Banheirão de Corto Maltese", mas classificamos o padrão como um todo. Nós não sabemos dizer que papel tem uma freqüência X qualquer do ruído neste padrão ou que parte mecânica do navio em questão ela representa. Talvez não saibamos nem mesmo, se vamos continuar conseguindo classificar o padrão, caso retiremos os valores correspondentes a esta freqüência do padrão.

Este é um exemplo típico. Mesmo em situações onde os dados possuem um significado conhecido, como no caso de dados de um paciente cardíaco potencial, onde eu sei o significado da freqüência cardíaca, mas onde eu não sei o relacionamento entre a freqüência cardíaca e a chance deste paciente ser vítima de um infarto numa determinada situação futura. É esse relacionamento, que eu não posso mapear de forma explícita, que eu quero que um sistema subsimbólico mapeie implicitamente para mim. E os sistemas subsimbólicos fazem isto, mas o fazem de forma fechada, sem gerar mapas, tabelas estatísticas ou conjuntos de regras de como criam este mapeamento.

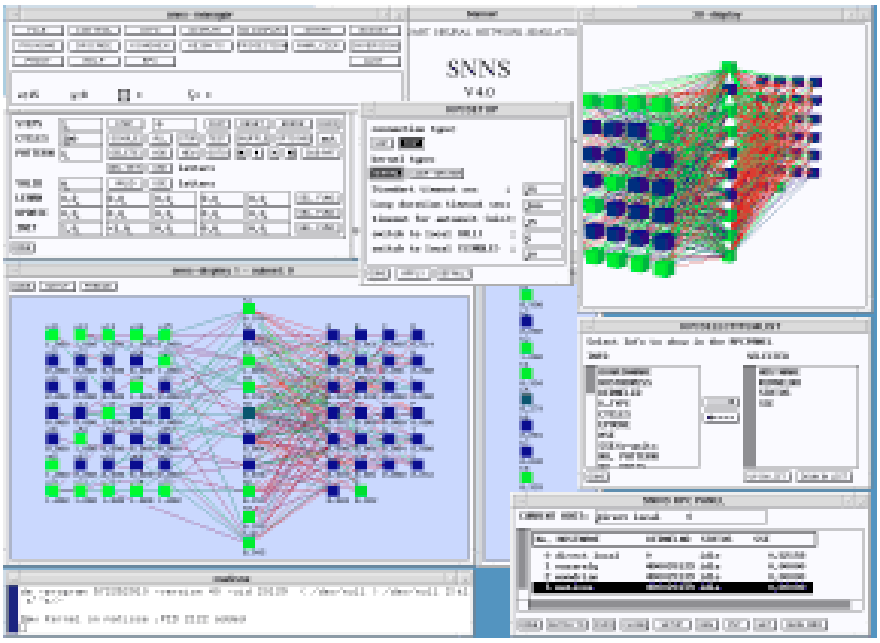
As Redes Neurais Artificiais são o mais difundido e popular conjunto de métodos subsimbólicos, sendo em geral caixas-pretas por excelência. Existem outros métodos que não vamos abordar aqui. O fato das redes neurais serem caixas pretas muitas vezes é citado como uma de suas desvantagens. Neste capítulo nós vamos ver que isto é relativo. Este capítulo pressupõe que você já viu Teoria das Redes Neurais na cadeira de Inteligência Artificial do Curso de Ciências da Computação ou de Sistemas de Informação da UFSC e que você tem o conhecimento teórico básco sobre os métodos: aqui nós vamos ver

aplicações de redes neurais e técnicas de integração das mesmas em sistemas mais complexos.

3.2. O Simulador SNNS - Stuttgarter Neural Network Simulator

O SNNS é um dos melhores simuladores de Redes Neurais existentes. Por isso nós vamos vê-lo aqui. O objetivo de alocarmos um capítulo a ele é o de prover ao aluno com uma ferramenta poderosa para a execução dos exercícios propostos, livrando-o da necessidade de ter de programar ele mesmo as redes.

Figura 3.1. **Interface de Usuário do SNNS Padrão para Unix/Linux**



O SNNS possui outra vantagem: após treinada um rede, você pode gerar com o SNNS um arquivo em linguagem "C" contendo a rede treinada, utilizando o utilitário **snns2c** fornecido juntamente com a versão Unix/ Linux. Este arquivo compilado pode ser utilizado como prorama standa-

lone ou então como biblioteca (.dll ou .so) linkada ao programa aplicativo que você for usar. Isto é uma vantagem para as aplicações que você vai desenvolver, pois permite que, se você for desenvolver as aplicações em Smalltalk, você utilize o pacote "DLL & C connect" para usar estas redes na sua aplicação em Smalltalk ou, que se você for desenvolver a sua aplicação em “C”, integre a rede diretamente ao seu código.

O SNNS possui duas versões: SNNS Padrão e JavaNNS. A versão padrão é fornecida em código-fonte e pode ser compilada para qualquer plataforma Unix/Linux. Ela possui uma série de utilitários que permitem a integração das redes neurais treinadas com o sistema em programas aplicativos. O JavaNNS é uma nova implementação em Java com finalidade exclusiva de ensino. Está disponível sob a forma de arquivo .jar para plataforma MS Windows e oferece apenas o ambiente de treinamento e teste interativo de redes neurais, não oferecendo nenhuma ferramente de integração das redes em programas aplicativos.

3.3. Classificadores: Usando Aprendizado Supervisionado para Reconhecer Padrões

Há dois modelos de redes neurais utilizados na prática como classificadores passíveis de serem gerados através de aprendizado supervisionado. Ambos os modelos baseiam-se nos Perceptrons *feed-forward*[1], variando o número de camadas e a função de ativação e, por conseguinte, a regra de aprendizado*: a)* As Redes Backpropagation[2] e b) as Redes de Base Radial.

Por serem as mais utilizadas e, do ponto de vista prático, mas mais importantes, vamos nos ocupar aqui das Redes Backpropagation, também chamadas **redes-BP**. As redes de Base Radial, conhecidas

1. Conceito referido por alguns autores em Língua Portuguesa como “Alimentação Adiante”. Como na literatura internacional de computação em muitas línguas, não somente a inglesa, este termo já se estabeleceu como termo técnico, vamos nos abster aqui de traduções de eufonia questionável.
2. Retropropagação. A rede em si na verdade é um perceptron de três ou mais camadas utilizando uma função quaselinear como função de ativação. Chamamos de redes backpropagation porque o algoritmo de aprendizado utilizado é assim chamado.

também por **redes-RBF** podem ser usadas, em teoria, para representar os mesmos tipos de problemas que uma rede Backpropagation equivalente. Alguns autores argumentam que são mais eficientes durante o treinamento. Por outro lado, a compreensão de seu algoritmo de aprendizado envolve uma matemática bastante mais complexa. Como são utilizadas para resolver o mesmo tipo de problemas que os onde Backpropagation encontra aplicação, sem vantagens consideráveis na qualidade do resultado final, vamos ignorá-las aqui. No final desta seção há uma comparação entre redes-RBF e redes-BP, extraída de [Haykin] e um comentário nosso sobre redes-RBF e sua relação com Nearest Neighbour e métodos que utilizam NN, como IBL*.

Como esta disciplina pressupõe que você já viu o assunto Redes Neurais, vamos aqui apenas recordar alguns conceitos matemáticos importantes para que você entenda a nossa discussão mais adiante de como se deve aplicar corretamente Backpropagation.

3.3.1. Simbolismo matemático

Existem várias convenções amtemáticas para a nomenclatura dos elementos de uma rede neural. Durante todo o capítulo de Redes Neurais vamos utilizar a simbologia matemática descrita em [Hertz et.ali.].

Um conjunto de treinamento é um conjunto p de padrões, todos de mesmo tamanho, cada qual dividido em duas partes: **vetor de entrada** ξ e **vetor de saída** ζ. O vetor de saída representa a atividades esperada nos neurônios de saída quando é apresentado o vetor de entrada nos neurônios de entrada da rede. Matematicamente o conjunto de treinamento pode ser definido como:

$$\{ \langle \xi^{\mu} \zeta^{\mu} \rangle | (\mu = 1, \ldots, p) \}$$

Os elementos são os seguintes:

ξ^{μ} Vetor de entrada do padrão de treinamento μ apresentado à camada de entrada.

ξ_k^{μ} Valor de entrada deste padrão para o neurônio k da camada de entrada.

ζ^{μ} Vetor de saída esperado do padrão de treinamento μ apresentado à camada de entrada.

ζ_i^{μ} Valor de saída deste padrão para o neurônio *i* da camada de saída.

w_{ij} Peso da conexão dirigida do neurônio *j* para o neurônio *i*. Lê-se "peso da conexão que *i* recebe de *j*".

h_j^{μ} Entrada total do neurônio *j* para o padrão μ. Não confunda com ativação.

V_j^{μ} Sinal de saída (ativação) do neurônio interno (hidden) *j* para o padrão μ.

O_i^{μ} Sinal de saída (ativação) do neurônio da camada de saída (*output*) *i* para o padrão μ.

g Função de ativação. Computa a ativação de um neurônio dada uma entrada *h*.

3.3.2. Princípios Básicos das Redes Backpropagation

A idéia básica de toda rede neural *feedforward* com aprendizado supervisionado é que, durante o seu treinamento, aplicamos à sua camada de entrada o padrão que desejamos que seja aprendido e propagamos a ativação ξ^{μ} gerada por este padrão na camada de neurônios de entrada, camada a camada, até gerarmos uma ativação nos neurônios da camada de saída.

Cada neurônio *i* de uma camada está tipicamente ligado a todos os neurônios *j* da camada anterior e recebe o sinal *h* de todos estes, cada qual ponderado pelo peso w_{ij} da conexão correspondente.

Cada neurônio possui uma **função de ativação** *g*() cuja variável independente é a entrada *h* do neurônio.

Todo neurônio *i* possui um **sinal de saída**, que será denominado V_i se o sinal for dirigido para outra camada da rede (neurônios de entrada e internos) ou O_i se o neurônio for de saída. A relação entre o sinal de saída e a ativação *g*(*h*) do neurônio tipicamente é a identidade. A separação da ativação e do sinal de saída em duas variáveis permite

realizar-se de forma elegante o sincronismo de toda a rede (inerentemente paralela) quando se simula a mesma em uma máquina de von Neumann (seqüencial): Atualizamos os valores de saída dos neurônios de uma camada somente depois de termos calculado todas as ativações desta camada. Isto é especialmente importante em redes BP recorrentes e outros modelos de rede onde neurônios possuem ligações com neurônios da própria camada.

O objetivo do treinamento supervisionado é modificar os pesos das conexões da rede de tal forma que a saída O^{μ} gerada para o vetor de entrada ξ^{μ} do padrão μ pela rede seja o mais próximo possível do vetor de saída ζ^{μ} deste padrão, de forma que no futuro, quando apresentarmos um outro vetor similar a ξ^{μ}, a rede produza uma resposta o mais próxima possível de ζ^{μ}.

Para realizarmos esta modificação dos pesos, representamos o erro de saída da rede como uma função do conjunto dos pesos, E(**w**), e utilizamos a técnica denominada **descida em gradiente** (*gradient descent* ou *Gradientenabstieg*) para realizar **alterações iterativas** dos pesos de forma a **reduzir o erro**. Para isto, representamos inicialmente E(**w**) como uma função-custo baseada na soma dos quadrados dos erros:

$$E(w) = \frac{1}{2}\sum_{i\mu} \langle \zeta_i^{\mu} - O_i^{\mu} \rangle^2 = \frac{1}{2}\sum_{i\mu} \langle \zeta_i^{\mu} - \sum_k w_{ik}\xi_k^{\mu} \rangle^2$$

A função acima representa o erro do resultado apresentado pela rede como uma função da diferença entre o resultado esperado e os pesos das ligações entre a camada de saída e a anterior e a entrada desta camada. Como os valores de entrada e de saída são dados, o conjunto de variáveis independentes é formado pelos pesos e o erro está representado como função dos pesos.

A variação do erro em função dos pesos pode então ser representada pelo vetor das **derivadas parciais do erro em função dos pesos**, também denominado **gradiente** do erro: $\frac{\partial E}{\partial w_{ik}}$.

Este vetor possui a propriedade de apontar no sentido do maior acréscimo de erro. Portanto para reduzir E(w) da forma mais rápida possível

movemo-nos no sentido contrário e definimos a modificação dos pesos como:

$$\Delta w_{ik} = -\eta \frac{\partial E}{\partial w_{ik}} = \eta \sum_{\mu} \langle \zeta_i^{\mu} - O_i^{\mu} \rangle \xi_k^{\mu}$$

Chamamos a esta fórmula de **regra-delta**.

Como esta modificação dos pesos deve ser gradativa, para evitar que uma modificação brusca de pesos faça a rede desaprender outro padrão já aprendido, utilizamos os seguintes princípios:

- introduzimos uma taxa de aprendizado η, tipicamente de valor < 0,2
- apresentamos os padrões de treinamento em ordem aleatória, garantindo que tenhamos apresentado todos antes de reapresentarmos algum.

Podemos também representar a alteração dos pesos individualmente para cada padrão μ:

$$\Delta_{w_{ik}}^{\mu} = \eta \langle \zeta_i^{\mu} - O_i^{\mu} \rangle \xi_k^{\mu}$$

$$\Delta_{w_{ik}}^{\mu} = \eta \, \delta_i^{\mu} \, \xi_k^{\mu} \text{ onde:}$$

$$\delta_i^{\mu} = \zeta_i^{\mu} - O_i^{\mu}$$

No treinamento de redes neurais, ao invés de falarmos de iterações, chamamos a cada ciclo de apresentação de todos os padrões de **época**. Antes de iniciar uma nova época, reorganizamos os padrões em uma nova ordem aleatória. A cada época todos os padrões são apreseentados.

A regra-delta, tal qual está representada acima, ainda não nos permite treinar a rede, apenas expressa um vetor de modificação dos pesos para

a camada de saída. Vamos agora recapitular brevemente a regra de aprendizado das redes-BP.

3.3.3. Aprendizado das Redes-BP

O fato que fez com que a pesquisa em redes neurais ficasse quase 20 anos (1968 - 1984) parada no cenário internacional foi o seguinte conjunto de fatos:

- para que uma rede neural feedforward possa representar uma função qualquer (universalidade representacional) ela necessita de pelo menos uma camada intermediária, além da camada de entrada e da de saída e a função de ativação de pelo menos parte dos neurônios deve ter caráter não-linear.
- para que a rede possa aprender, é necesário que possamos calcular a derivada do erro em relação aos pesos em cada camada, de tràs para frente, de forma a minimizar a função custo definida na camada de saída. Para isto ser possível, a função de ativação deve ser derivável.

Já o modelo de rede de McCulloch & Pitts possuía um caráter representacional quase universal, mas não apresentava a possibilidade de treinar-se redes com camadas intermediárias, pois a função de limiar por eles desenvolvida não era derivável.

A regra de aprendizado do modelo do Perceptron podia treinar redes com quantas camadas se quisesse, mas estas camadas intermediárias eram inúteis pois a função de ativação linear do neurônio do Perceptron fazia com que camadas adicionais representasssem transformações lineares sobre transformações lineares, o que é equivalente a uma única transformação linear e incapaz de representar problemas linearmente inseparáveis.

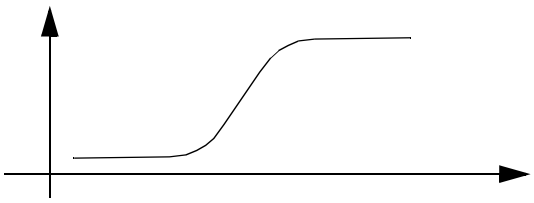
Como resolver este problema? Através de uma idéia muito simples: introduzir uma não linearidade “bem comportada” através de **funções quaselineares contínuas e deriváveis**. Com isto conseguimos:

- introduzir uma não linearidade sem no entanto alterar de forma radical a resposta da rede (ela se comporta de forma similar a uma rede linear para casos “normais”) e

- possibilitar o cálculo da derivada parcial do erro em relação aos pesos (o que nós queríamos desde o começo) de uma forma elegante e generalizável para todas as camadas.

As funções de ativação quaselineares que se costuma utilizar têm a forma geral do desenho abaixo.

Figura 3.2. **Gráfico da forma geral da função de ativação quaselinear**



Esta família de funções possue comportamento assintótico, significando que para valores valores muito grandes ou muito pequenos de *y* elas tendem a um valor constante. Além disso é derivável e, próxima de sua parte "mais normal", possue um comportamento muito semelhante ao de uma reta inclinada, como em uma função linear.

As duas funções mais utilizadas são a **tangente hiperbólica** e a **função logística**, mostradas abaixo, juntamente com suas derivadas:

$$g(h) = \tanh\langle \beta h \rangle$$
$$g'(h) = \beta\langle 1 - g^2(h) \rangle$$

$$g(h) = \frac{1}{1 - \exp^{-2\beta h}}$$
$$g'(h) = 2\beta g(h)\langle 1 - g(h) \rangle$$

O parâmetro β é um parâmetro livre com $\beta > 0$ que permite parametrizar o comportamento da função. Na prática utilizamos geralmente β=1.

Estas duas funções possuem a propriedade adicional de se poder definir suas derivadas em função delas mesmas, significando que se pode reuti-

lizar o valor já calculado para a ativação ao se realizar a retropropagação do erro, sem necessidade de novos cálculos complexos.

A função tangente hiperbólica possui domínio no intervalo aberto]-1, +1[e a função logística no domínio]0, 1[. Podemos escolhê-las de acordo com nosso domínio de aplicação ou fazer, como o fazem a maioria dos simuladores de redes-BP, tomar a função logística como função padrão e suprir valores negativos de saída através da adição de uma constante. Essa constante é muitas vezes provida através de um neurônio virtual a mais em cada camada, denominado *bias neuron*, que possui saída constante e negativa.

Dessa forma podemos reescrever a função-custo:

$$E(w) = \frac{1}{2}\sum_{i\mu} \langle \zeta_i^\mu - O_i^\mu \rangle^2 = \frac{1}{2}\sum_{i\mu} \langle \zeta_i^\mu - g\langle \sum_k w_{ik}\xi_k^\mu \rangle \rangle^2$$

Derivando, podemos agora reescrever a regra-delta:

$$\Delta w_{ik} = -\eta \frac{\partial E}{\partial w_{ik}} = \eta \sum_\mu \langle \zeta_i^\mu - g\langle h_i^\mu \rangle \rangle g'\langle h_i^\mu \rangle \xi_k^\mu$$

E a regra para modificação dos pesos imediatamente após a apresentação de cada padrão:

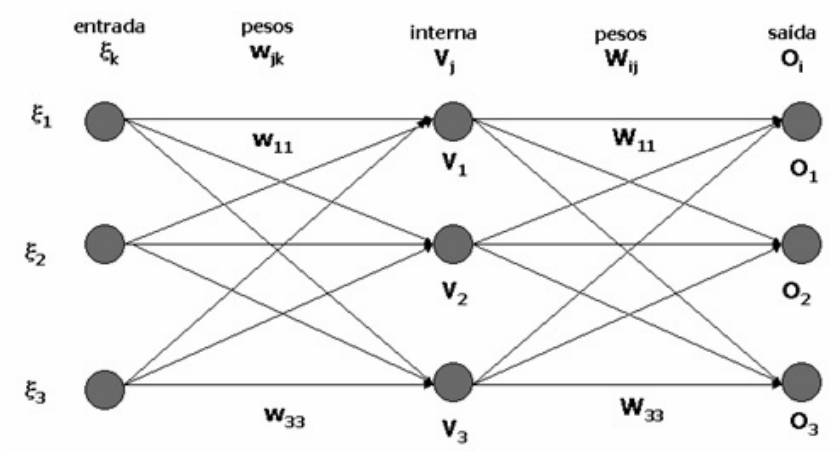
$$\Delta_{w_{ik}}^{\mu} = \eta\, \delta_i^\mu\, \xi_k^\mu \text{ onde:}$$
$$\delta_i^\mu = \langle \zeta_i^\mu - O_i^\mu \rangle \; g'\langle h_i^\mu \rangle$$

3.3.3.1. Aprendizado para Redes de Várias Camadas

As fórmulas deduzidas acima nos permitem implementar uma descida em gradiente, realizando o aprendizado, em redes de duas camadas. Como já discutimos, porém, necessitamos de redes com pelo menos uma camada interna. Para extender o aprendizado a redes de uma camada interna, temos de primeiramente extender a nossa função-custo de forma que ela diferencie entre pesos de neurônios internos e pesos de neurônios de saída.

Para isto observemos a figura 3.3 abaixo, que representa uma rede de 9 neurônios divididos em três camadas. Você pode observar na figura que

Figura 3.3. **Rede feedforward típica mostrando nomenclatura utilizada.**



os pesos k -> j são representados por $\mathbf{w_{jk}}$ e os pesos j -> i são representados por $\mathbf{W_{ij}}$.

Lembrando que a nossa função-custo tem a forma geral:

$$E(w) = \frac{1}{2}\sum_{i\mu} \langle \zeta_i^\mu - O_i^\mu \rangle^2$$

...e que desejamos minimizar E(w) modificando gradualmente o vetor dos pesos w = (w_{11},....,W_{11},....).

Para isso, vamos inicialmente definir como surge o vetor de saída da rede O_i:

- Depois que o vetor de entrada ξ^μ foi apresentado à camada de entrada, cada neurônio da camada intermediária recebe a entrada definida por:

$$h_j^\mu = \sum_k w_{jk}\xi_k^\mu$$

- produzindo como saída a ativação de um neurônio interno:

$$V_j^\mu = g(h_j^\mu) = g\langle \sum_k w_{jk}\xi_k^\mu \rangle$$

- o que faz com que um neurônio de saída possua a entrada:

$$h_i^\mu = \sum_i W_{ij} V_j^\mu = \sum_i W_{ij} g\langle \sum_k w_{jk}\xi_k^\mu \rangle$$

- e produz como saída da rede:

$$O_i = g\langle h_i^\mu \rangle c = g\langle \sum_i W_{ij} V_j^\mu \rangle = g\langle \sum_i W_{ij}\, g\langle \sum_k w_{jk}\xi_k^\mu \rangle \rangle$$

- o que, por sua vez nos permite escrever a função-custo da seguinte forma:

$$E(w) = \frac{1}{2}\sum_{i\mu} \langle g\langle \sum_i W_{ij}\, g\langle \sum_k w_{jk}\xi_k^\mu \rangle \rangle \rangle^2$$

Esta função E(**w**) é contínua e diferenciável para todos os w_{jk},, W_{ij}. Isto permite que se utilize o método de descida em gradiente para construir o algoritmo de aprendizado.

Faremos a descida em gradiente em etapas, uma para cada conjunto de pesos entre camadas. Para uma rede BP com uma camada interna, isto é feito de trás para frente (por isso retropropagação do erro) em duas etapas, uma para W_{ij} e outra para w_{jk}.

Para W_{ij} fica como segue:

Com isso, a regra-delta para a adaptação dos pesos da camada de saída fica:

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}} = \eta \sum_\mu \langle \zeta_i^\mu - O_i^\mu \rangle \cdot g'\langle h_i^\mu \rangle V_j^\mu = \eta \sum_\mu \delta_i^\mu V_j^\mu$$

$$\delta_i^\mu = g'\langle h_i^\mu \rangle \cdot \langle \zeta_i^\mu - O_i^\mu \rangle$$

$$\frac{\delta E}{\delta W_{ij}} = \sum_{\mu} \frac{\delta E}{\delta O_i} \cdot \frac{\delta O_i}{\delta W_{ij}}$$

$$= -\sum_{\mu} \langle \zeta_i^{\mu} - O_i^{\mu} \rangle \cdot \frac{\delta O_i}{\delta W_{ij}}$$

$$= -\sum_{\mu} \langle \zeta_i^{\mu} - O_i^{\mu} \rangle \cdot g'\langle h_i^{\mu} \rangle \frac{\delta h_i^{\mu}}{\delta W_{ij}}$$

$$= -\sum_{\mu} \langle \zeta_i^{\mu} - O_i^{\mu} \rangle \cdot g'\langle h_i^{\mu} \rangle V_j^{\mu}$$

Observe que esta é praticamente a mesma regra que havíamos escrito antes, somente substituímos os valores do vetor de entrada pelos valores da ativação dos neurônios internos.

Para o caso dos pesos *entrada -> interno* nós vamos ter de aplicar a regra da cadeia também para as derivadas internas:

$$\frac{\delta E}{\delta w_{jk}} = \sum_{\mu} \frac{\delta E}{\delta V_j^{\mu}} \cdot \frac{\delta V_j}{\delta w_{jk}}$$

$$= \sum_{\mu} \left(\sum_{i} \frac{\delta E}{\delta O_i} \cdot \frac{\delta O_i}{\delta V_j^{\mu}} \right) \cdot \frac{\delta V_j^{\mu}}{\delta W_{ij}} \quad \text{com a regra da cadeia}$$

$$= \sum_{\mu} \left(\sum_{i} -\langle \zeta_i^{\mu} - O_i^{\mu} \rangle \cdot g'\langle h_i^{\mu} \rangle \frac{\delta h_i^{\mu}}{\delta V_j^{\mu}} \right) \cdot g'\langle h_j^{\mu} \rangle \frac{\delta h_j^{\mu}}{\delta w_{jk}}$$

$$= -\sum_{\mu i} \langle \zeta_i^{\mu} - O_i^{\mu} \rangle \cdot g'\langle h_i^{\mu} \rangle W_{ij} \cdot g'\langle h_j^{\mu} \rangle \xi_k^{\mu}$$

Com isso, a regra-delta para a adaptação dos pesos da camada interna fica:

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial w_{jk}} = \eta \sum_{\mu} \delta_j^{\mu} \xi_k^{\mu}$$

$$\delta_j^{\mu} = g'\langle h_j^{\mu} \rangle \cdot \sum_i W_{ij} \cdot \delta_i^{\mu}$$

Observe que esta fórmula possui caráter genérico extensível a qualquer camada interna, pois se criarmos um buffer para os deltas da adaptação da camada anterior, podemos utilizá-los para a atual, de trás para frente.

Dessa forma, a **regra-delta generalizada** para a *m*-ésima camada de uma rede backpropagation fica assim (considerando-se a camada de entrada como camada 0 e a camada de saída como camada *M*):

$$\Delta w_{pq}^{m} = \eta \sum_{\mu} \delta_p^{m,\mu} \; V_q^{m-1,\mu}$$

$$\delta_p^{m,\mu} = \begin{cases} g'(h_j^{M,\mu}) \cdot (\zeta_p^{\mu} - O_p^{\mu}), & \text{para camada de saída = M} \\ g'\langle h_p^{m,\mu} \rangle \cdot \sum_r w_{rp}^{m+1} \cdot \delta_r^{m+1,\mu}, & \text{senão (m < M)} \end{cases}$$

3.3.3.2. Algoritmo de Aprendizado para Redes de Várias Camadas

Em termos práticos, o que vai nos interessar para formular o **algoritmo de aprendizado por retropropagação do erro** (*error backpropagation*) é a **regra delta generalizada**, como foi apresentada acima. A partir dela é possível formular-se todo o algoritmo de aprendizado.

O algoritmo é como segue:

Algoritmo Backpropagation

Dados:

- uma rede feedforward de *M* camadas
- um conjunto de conexões ponderadas por pesos e dirigidas da camada *m*-1 para a camada *m*: w_{ij}^{m}
- uma função de ativação não-linear contínua e diferenciável no domínio dos valores a serem treinados na rede
- uma taxa de aprendizado η
- um conjunto de treinamento contendo ξ^{μ} entradas e ζ^{μ} saídas.

Execução:

1. Inicialize os pesos w^m das camadas $m = 0, \ldots, M$ com valores aleatórios e pequenos.
2. Escolha um padrão μ do conjunto de treinamento:
   Entrada da rede: ξ^{μ}
   Saída desejada: ζ^{μ}
   Apresente o padrão à rede: $V_k^0 = \xi_k^{\mu}$ para todos os neurônios de entrada.
3. Propague a ativação através das m = 1,...,M camadas restantes da rede:

   $$V_i^m = g(h_i^m) = g(\sum_j w_{ij}^m V_j^{m-1})$$

   até que as saídas da rede tenham sido calculadas.
4. Calcule os δ para a camada de saída:

   $$\delta_i^{\mu} = g'\langle h_i^{\mu}\rangle \cdot \langle \zeta_i^{\mu} - O_i^{\mu}\rangle$$

   para todo *i*.
5. Itere pelas camadas anteriores, de trás para frente, calculando os δ através da retropropagação do erro:

   $$\delta_p^{m,\mu} = g'\langle h_p^{m,\mu}\rangle \cdot \sum_r w_{rp}^{m+1} \cdot \delta_r^{m+1,\mu}$$

   para todo neurônio *p* da camada *m* - 1.

6. Determine a variação dos pesos para todas as camadas:

$$\Delta w_{pq}^{m} = \eta \sum_{\mu} \delta_{p}^{m} \; V_{q}^{m-1}$$

para todas as conexões entre neurônios.

7. Determine os novos pesos das conexões:

$$w_{pq}^{m} = w_{pq}^{m} + \Delta w_{pq}^{m}$$

8. Retorne ao passo 2 e tome o próximo padrão.

3.3.4. O que aprende uma Rede-BP ?

Como citamos em outras partes deste texto, uma rede backpropagation, ao contrário de redes-RBF ou classificadores baseados em Nearest Neighbour, como IBL, **aprende uma função capaz de mapear a entrada à saída**, caso esta exista. Se o conjunto de treinamento for inconsistente a rede não aprenderá nada ou aprenderá cada exemplo individual do conjunto de treinamento, caso a criemos grande o suficiente.

Em princípio, o mapeamento entrada-saída em uma rede-BP está distribuído sobre o total dos pesos e conexões da rede, sendo bastante difícil associarmos um determinado neurônio e suas conexões a uma determinada classe.

Do ponto de vista matemático, existem várias interpretações do significado dos pesos aprendidos por uma rede neural. Uma discussão detalhada deste assunto foge do escopo de uma disciplina de graduação e nós remetemos à literatura, principalmente [Hertz et.ali.].

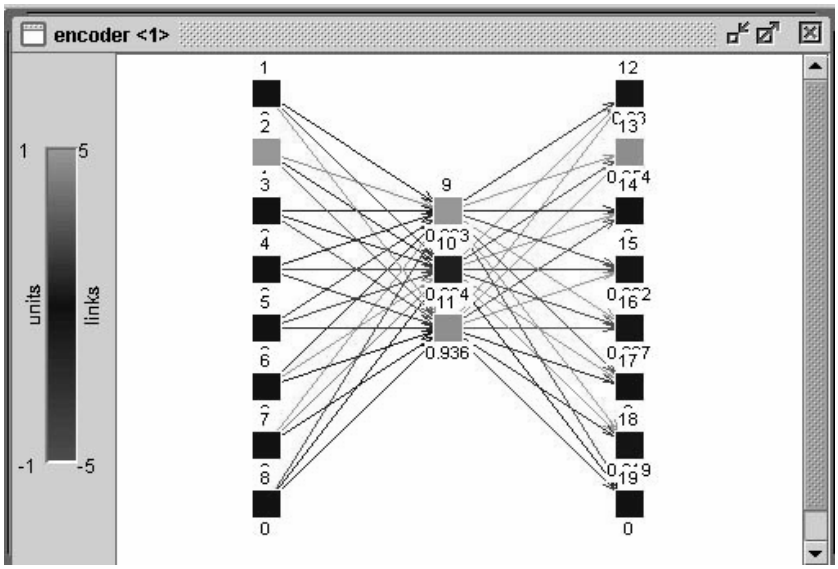
Existem, porém, algumas situações interessantes, onde o aprendizado da rede neural pode ser visualizado e podemos realmente associar um ou um conjunto de neurônios a uma determinada classe. Isto tende a acontecer quando o conjunto de treinamento contém classes realmente muito bem comportadas.

O exemplo clássico para este comportamento é o **encoder** (codificador). Este exemplo está incluído na coleção de exemplo sprontos do SNNS e sugerimos ao leitor que faça alguns experimentos com ele. O encoder é um exemplo onde podemos fazer os padrões de ativação dos neurônios da

camada interna representarem uma compressão dos dados de treinamento e ainda utilizar esta compressão de dados como um código representando os mesmos.

O encoder toma um valor de entrada de 0 a 7 e aprende a asociá-lo a mesma saída, possuíndo 8 neurônios de entrada, um para cada valor e 8 neurônios de saída, com a mesma representação. Na camada intermediária possui apenas 3 neurônios. O objetivo é que, ao aprender a associação entrada-saída, ele **codifique** os dados. A figura abaixo mostra o encoder recebendo o número 1 como padrão de entrada (01000000) e representando internamente este número como 101.

Figura 3.4. **Encoder**



O conjunto de treinamento do encoder não é só linearmente separável, mas é também linearmente independente e, portanto, um conjunto extremamente fácil de aprender, que não necessitaria de uma camada interna, podendo ser representado por um perceptron simples. Mesmo assim é interessante de se observar o fenômeno da representação interna. Este fenômeno porém não ocorre sempre dessa forma, com a rede “inventando” seu próprio código binário. Às vezes os pesos se

distribuem de uma forma tal na rede que não é possível uma interpretação visual da “representação interna”.

3.3.4.1. Exercício: Observando o Encoder

Vamos ver com que freqüência o encoder realmente aprende uma representação interna que para nós, humanos, faz “sentido”. Carregue o exemplo do encoder no SNNS. Reinicialize a rede e treine-o. Bastam 100 épocas pois o conjunto é aprendido extremamente rápido. Feito isto, vá para o modo “updating” e repasse todo o conjunto de treinamento pela rede. Foi possível criar-se um arepresnetação interna similar a algum código binário conhecido ? Repita este processo várias vezes, reincializando, treinando e testando a rede para ver como ela se comporta.

3.3.5. Desenvolvimento de Aplicações e Resultados Práticos do Uso de Backpropagation

Esta seção é dedicada a aspectos práticos do desenvolvimento de aplicações que utilizem uma rede neural como classificador de padrões, em particular uma rede-BP. Além disso será sugerido aqui um exercício que deverá ser resolvido pelo aluno.

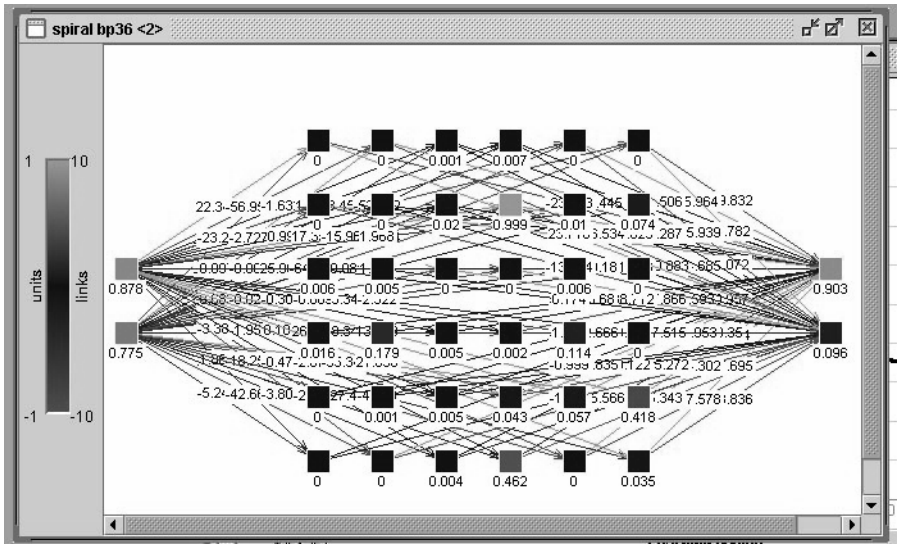
Importante: O texto abaixo está estruturado de forma a complementar o assunto visto em aula de laboratório. O material não pretende ser auto-explicativo nem didaticamente autosuficiente: esta seção supõe que a aula a que ela se refere foi assistida pelo aluno e serve apenas de referência.

3.3.5.1. Aprendizado de Conjuntos Intrincados

O conjunto de dados distribuído sob a forma de duas espirais duplas, como já foi dito anteriormente, é um conjunto clássico de teste para redes neurais. Vamos tentar avaliar a competência de todos so métodos que veremos nesta disciplina utilizando este conjunto de dados. Até agora você viu como técnicas simples como kNN e como métodos simbólicos que utilizam Nearest Neighbour, como IBL se comportam frente a este conjunto de dados. Nesta seção veremos como se comporta uma rede-BP frente ao mesmo conjunto de dados. O objetivo desta comparação é fornecer-lhe dados para avaliar a performance e adequação desta técnica.

Para isto utilizaremos uma rede neural como mostra a figura abaixo. Esta rede está organizada em 3 camadas, possuindo apenas uma camada interna. A organização desta camada interna como uma matriz é somente um subterfúgio gráfico para fazer a rede caber na janela do simulador, a posição de um neurônio nesta matriz, ao contrário do que ocorrerá mais adiante em redes de Kohonen, não possui nenhum significado.

Figura 3.5. **Rede-BP para teste do conjunto espiral dupla mostrada no visualizador de rede do SNNS**

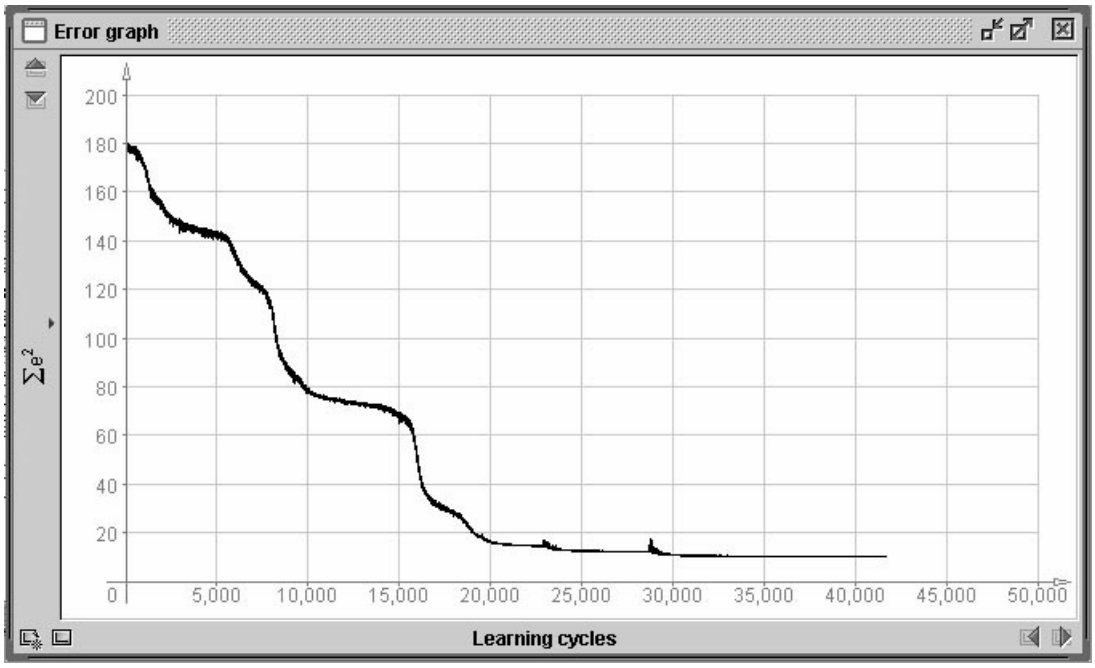


A camada da esquerda é a camada de entrada, a camada da direita é a camada de saída. Ambas possuem dois neurônios. Na de entrada são apresentadas as coordenadas x e y do ponto na espiral e na de saída, há um neurônio para cada classe (braço da espiral). O neurônio que apresentar a maior atividade representa a classe resultante da classificação de um ponto apresentado.

A próxima figura representa o gráfico de variação do erro E(**w**) durante o treinamento desta rede. A abcissa representa o número de épocas (ciclos sobre todos os 360 padrões) e a ordenada o erro global E(**w**) da rede. Nós interrompemos o treinamento em 42.000 épocas, o que para

um conjunto de treinamento de 360 padrões, perfaz **15.120.000** iterações. Observe que o erro se reduz em “saltos”. Isto é comum em métodos de descida em gradiente e significa que o processo encontrou uma “ravina” na paisagem de gradientes e começou a descê-la até encontrar outro “platô”.

Figura 3.6. **Gráfico do erro global E(w) da rede da figura anterior.**

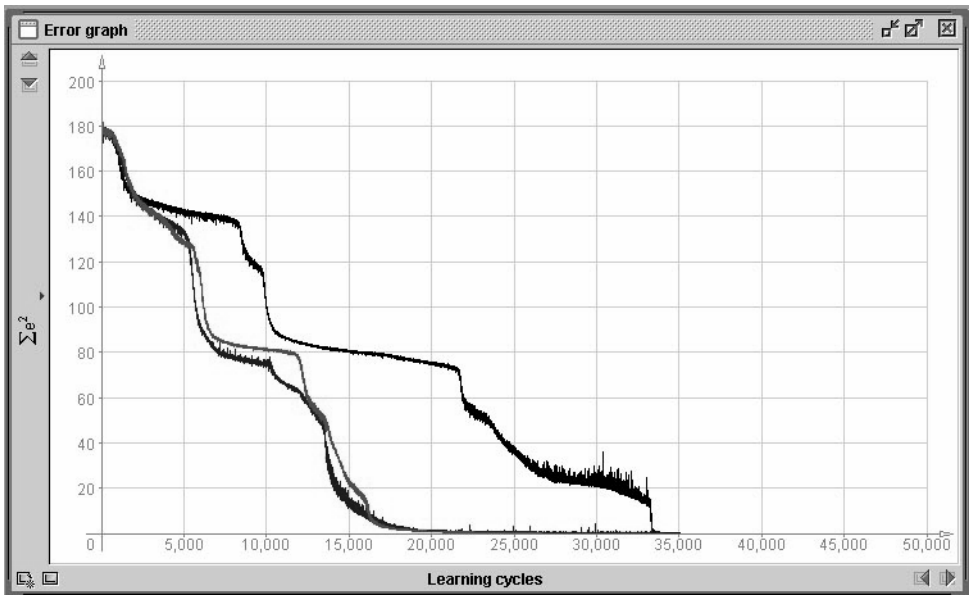


Decidimos abortar o treinamento em 42.000 ciclos pois o erro não apresentou mais redução significativa durante 20.000 ciclos, o que significa que a rede ou encontrou o máximo de fidelidade possível no mapeamento da função que desejamos que aprenda ou então que encontrou um **mínimo local** na paisagem de erro.

Um mínimo local é um pequeno “vale” na paisagem de erro definida por E(w), mas que não contém o menor erro possível. Como as regiões ao redor de um mínimo local são regiões que possuem todas um erro menor e o método de descida em gradiente tenta sempre encontrar um novo conjunto de valores de pesos que reduza o erro e nunca um que aumente,

a rede nunca sairá dalí. Mínimos locais são o maior problema que deparamos em métodos de descida em gradiente. Como o estado inicial do sistema é aleatório e a ordem de apresentação dos padrões também, o caminho que leva ao mínimo erro global pode passar por um mínimo local ou não, dependendo do acaso. Quando nos deparamos com um mínimo local onde a rede "encalhou", a única opção que temos utilizando o algoritmo de rede-BP tradicional é a de reinicializar a mesma e treinar novamente desde o início. Ná próxima figura vemos três situações de gráfico de erro completamente diferentes geradas após 50.000 épocas pela mesmo rede com o mesmo conjunto de dados. Apenas foram geradas inicializações diferentes, porém com os mesmos parâmetros. Nós vamos discutir questão dos mínimos locais e outros aspectos importantes como oscilações na próxima seção.

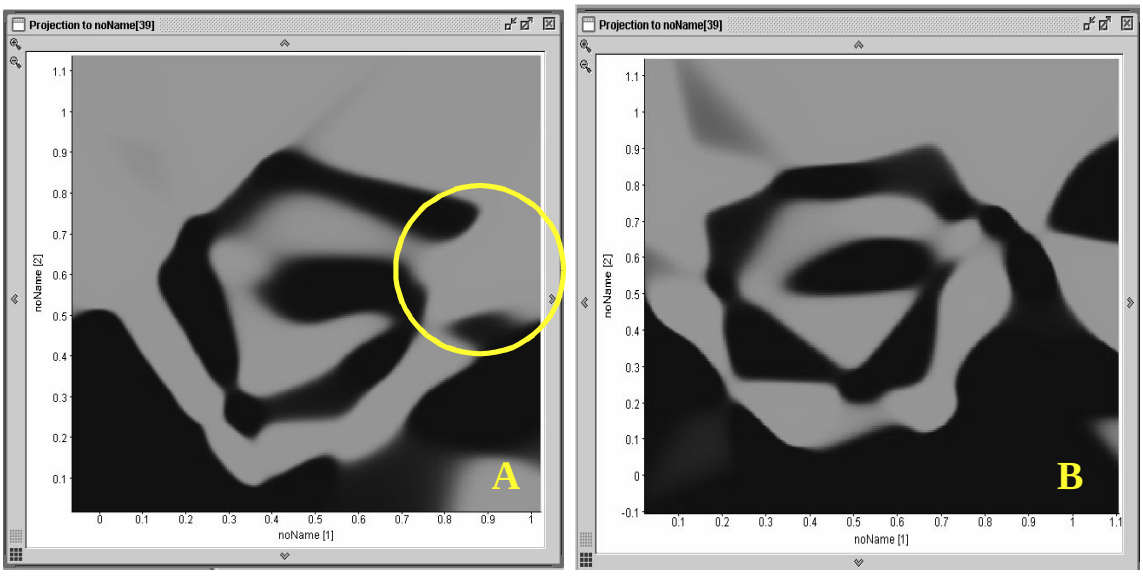
Figura 3.7. **Comparação de variação aleatória de performance de treinamento em dependência da inicialização**



Nos três casos acima, apesar da performance de aprendizado ter sido bastante diferente, em nenhum dos três casos a rede encontrou um mínimo local e estacionou em um platô de erro constante. Todas as três redes atingiram um erro próximo de 0.

Na figura abaixo vemos o resultado da projeção de duas superfícies de decisão geradas por duas das mostradas anteriormente. Cada projeção é gerada através de uma ferramenta do SNNS que permite associar-se graficamente dois neurônios de entrada a um de saída. O gráfico resultante mostra a ativação do neurônio de saída para todas as possíveis combinações de valores de entrada nos dois neurônios selecionados. No caso específico da espiral dupla, onde existem apenas dois neurônios de entrada e dois neurônios de saída, esta projeção corresponde à superfície de decisão gerada pela rede. Compare este resultado com a performance do algoritmo IBL, observando que cada rede que gerou este resultado teve de ser treinada durante 20 minutos em um computador com 512 MB de RAM e um processador AMD 1800+ e que o algoritmo IBL (1, 2 ou 3) toma uma fração de segundo para realizar a mesma coisa em um computador com essas especificações.

Figura 3.8. **Duas projeções das superfícies de decisão geradas pelas redes anteriores**



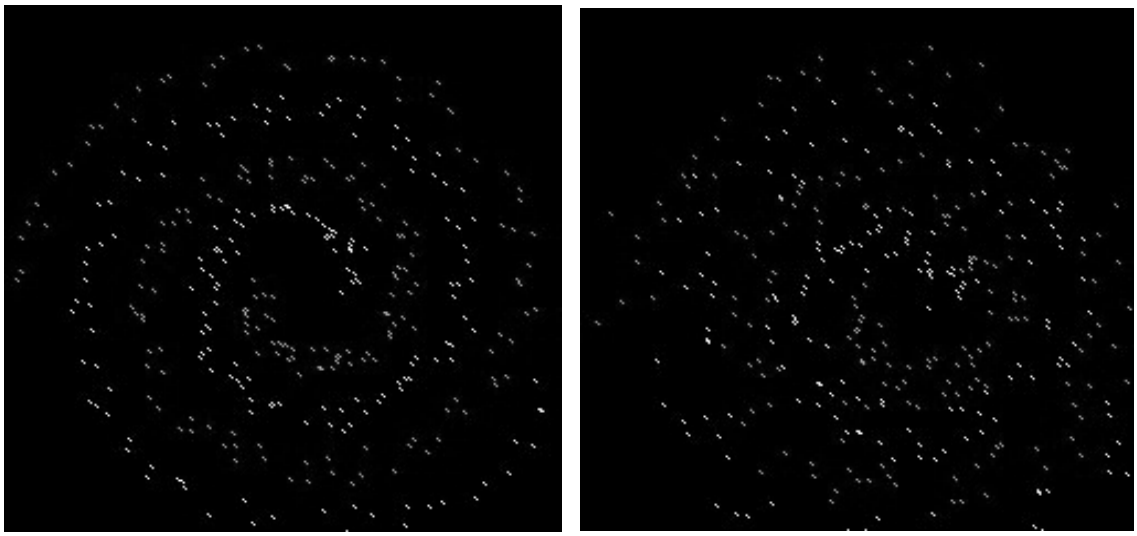
Observe que na rede (A) há uma área mapeada incorretamente. A rede (B) aprendeu a espiral com qualidade aceitável. A rede (A) é a primeira rede neural treinada, cuja treinamento foi interrompido em função de uma mínimo local. Aqui podemos observar que o mínimo local que a rede

encontrou a impediu de aprender uma parte do problema, fazendo com que a superfície de decisão ficasse incompleta.

3.3.5.2. Aprendizado de Conjuntos Intrincados Apresentando Erro

Para termos uma idéia como se comporta uma rede-BP nos casos onde existe algum erro nos conjuntos de dados, retomemos as nossas espirais geradas apresentando desvio aleatório de posição dos dados do conjunto de treinamento de até 15 e de até 30 pixel.

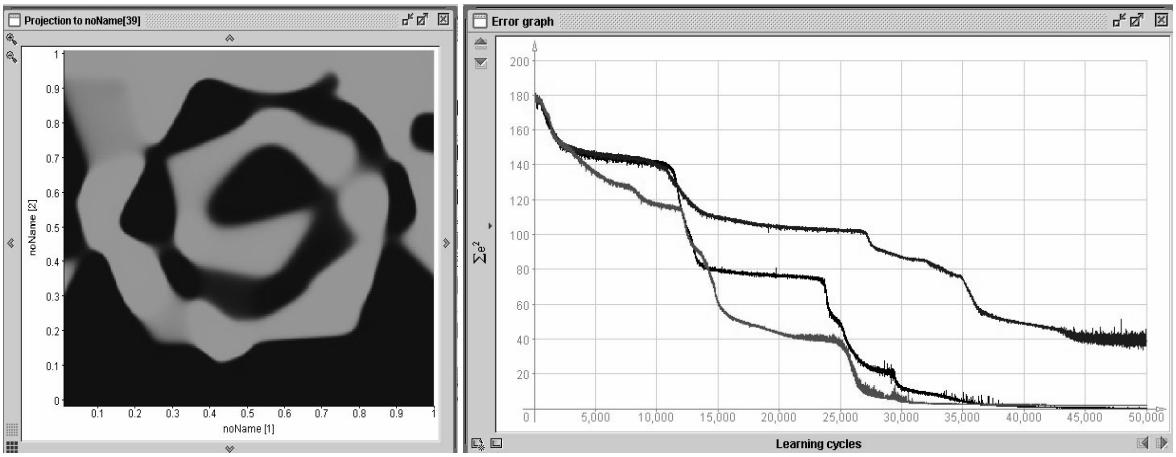
Figura 3.9. **Espirais com erro de até 15 e de até 30 pixel na posição dos pontos**



O resultado do treinamento de três redes com o conjunto de dados de treinamento com erro de até 15 pixel pode ser visto abaixo. Observe que a superfície de decisão gerada (mostra a rede cuja curva de treinamento está em preto) é bastante boa. As curvas de treinamento mostram que uma das redes treinadas (curva em azul), além de mostrar um comporta-

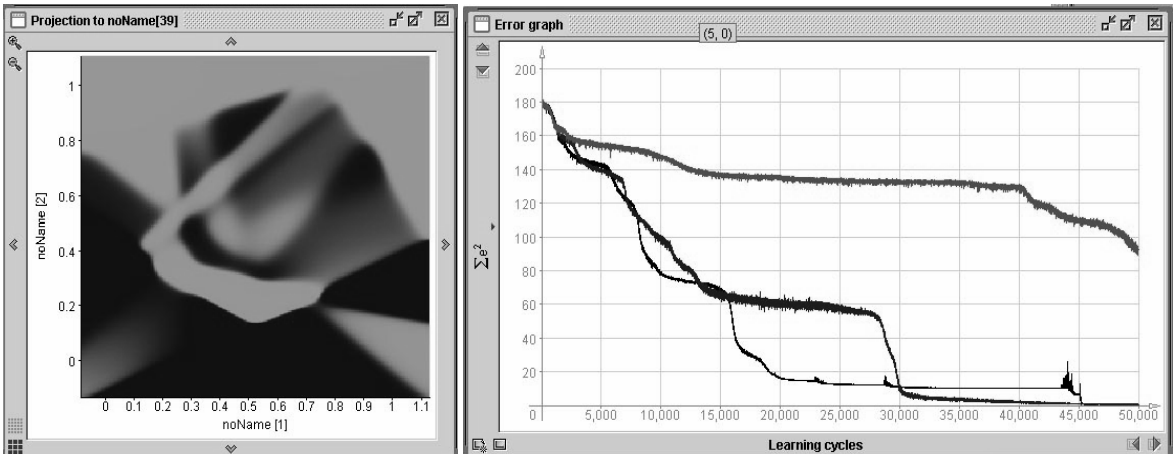
mento de aprendizado bastante ruim, com muitas oscilações, provavelmente vai cair em um mínimo local.

Figura 3.10. **Comparação de 3 redes treinadas com o conjunto com erro de 15 pixel**



Se utilizarmos o conjunto de treinamento gerado com erro de até 30 pixel, a qualidade do aprendizado deteriora bastante, como vemos na superfície de decisão abaixo. As curvas comparam a performance de uma rede treinada com dados com erro de 30 pixel (vermelho), erro de até 15 pixel e uma treinada com dados sem erro.

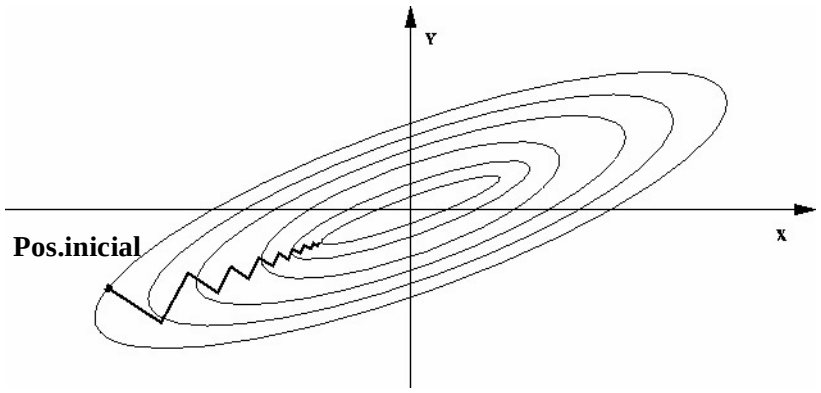
Figura 3.11. **Comparação de 3 redes treinadas com o conjunto com erro de 30 pixel**



3.3.5.3. Descida em Gradiente, Mínimos Locais e outros aspectos práticos do treinamento de uma rede-BP

Como citamos brevemente antes, o processo de redução do erro E(w) pertence a uma categoria de métodos matemáticos denominado Métodos de Descida em Gradiente (*Gradient Descent Methods*). Nocaso específico das redes-BP, que nos interessa, podemos imaginar a idéia de que o erro é uma (hiper) superfície em um espaço definido pelos pesos [w] da rede neural. O estado atual da rede (conjunto de valores específico dos pesos das conexões entre os neurônios) é um ponto sobre esta superfície. O nosso objetivo é mover este ponto através da alteração dos valores do spesos de forma a encontrar uma posição onde o erro seja o menor possível. O movimento é realizado sempre no sentido de reduzir-se o erro, ou seja sempre descendo a superfície de erro de forma que o próximo ponto seja uma posição mais funda nesta superfície, até encontrar uma posição de onde não sejja possível descer-se mais. Se imaginarmos uma situação 3D, onde há apenas dois pesos definindo os espaços x e Y e a coordenada z sendo definida pelo erro, podemos imaginar o processo como o mostra a figura abaixo.

Figura 3.12. **Idéia da descida em gradiente**



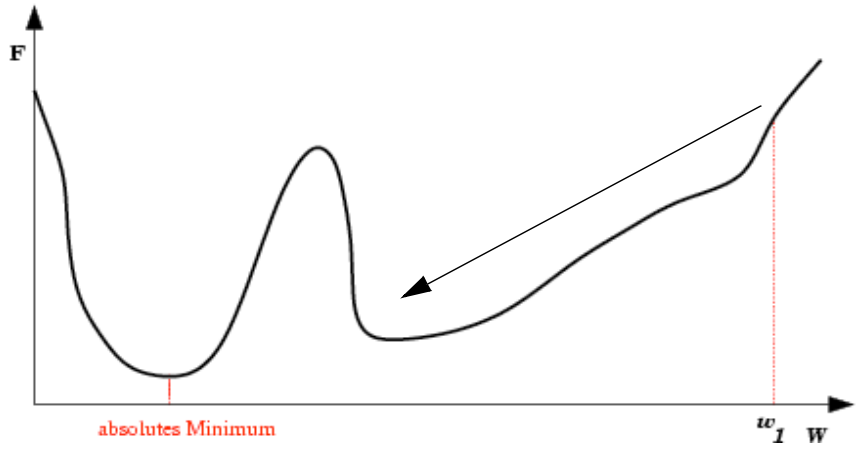
Na prática teremos situações muito mais complexas do que essa, onde os pesos definem um espações 20,100 ou até 1000 dimensões. Para entender o conceito, porém, costumamos visualizá-lo em situações simplificadas. A forma mais ismples de visualizar o processo de descida em gradiente é a situação onde temos apenas um peso para ser modifi-

cado. Isto nos permite representar a descida em gradiente em um gráfico 2D, onde a abcissa é o único peso w e a ordenada o erro global E(w) em função dos valores de w. Este exemplo é simplificado mas permite uma excelente visualização das situações mais importantes neste contetxo.

O material abaixo é um resumo do tutorial de redes neurais disponibilizado pela **Neuro-Fuzzy AG** (Grupo de Trabalho Neuro-Fuzzy) do Departamento de Matemática da Universidade de Muenster, Alemanha. As imagens originais são GIFs animados que mostram o processo descrito textualmente para cada figura. Não podemso repetir tudo aqui, por isso sugerimos que o leitor olhe no site da dsciplina para visualizar melhor o que está sendo explicado aqui.

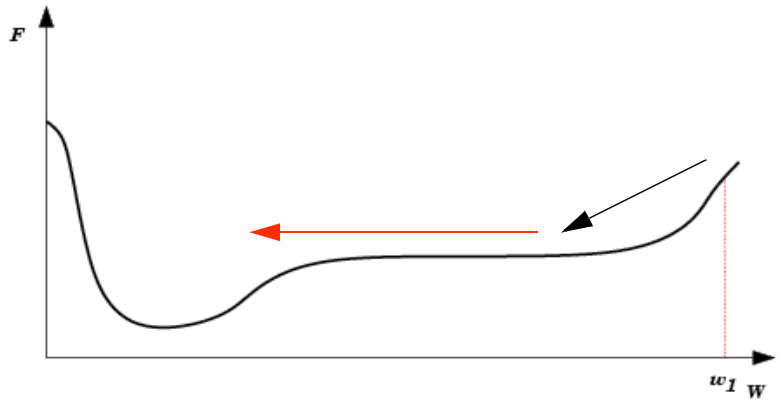
Mínimo Local: Partindo-se da configuração de pesos inicial *w1*, o método de descida em gradiente não encontrará a solução (mínimo global).

Figura 3.13. **Mínimo Local**



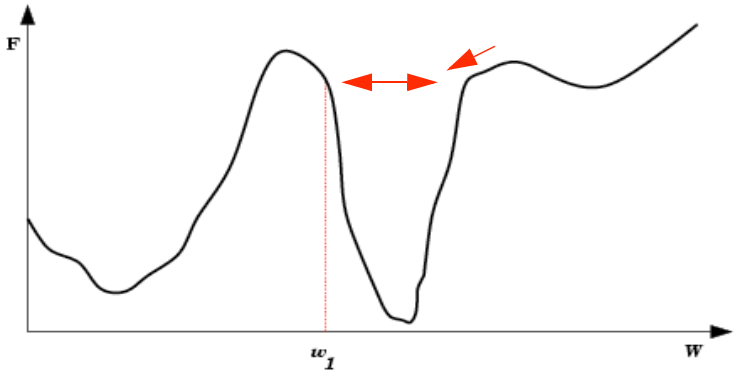
Platô encontrado em E(w)durante treinamento da rede neural. Durante um longo período não haverá mudanças significativas em E(**w**). Após um tempo, porém, o mínimo absoluto (global) é encontrado.

Figura 3.14. **Platô**



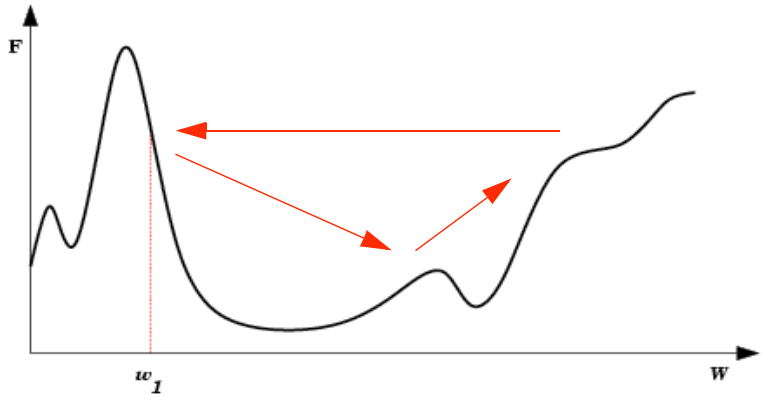
Oscilações ocorrem quando o processo de descida de gradiente cai em uma ravina de onde não sai mais. Também é uma espécie de mínimo local. O passo de modificação dos pesos (taxa de aprendizado) é grande demais para que a rede caia na ravina, mas pequena demais para sair do mínimo. Ao conrário do mínimo local comum, aqui a rede não encontra um estado estável.

Figura 3.15. **Oscilações**



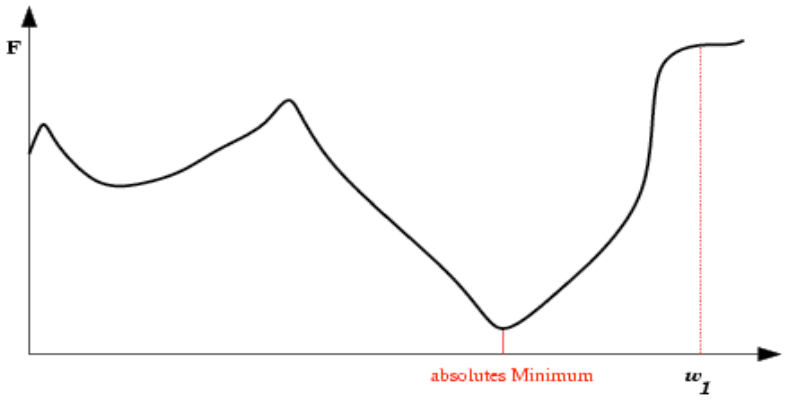
A **Oscilação Indireta** é uma situação de oscilação mais complexa, onde a rede também fica "presa" em um mínimo local e não encontra um estado estável. Neste caso porém, existem estados intermediários entre os estados extremos da oscilação, onde o erro por momentos se permite reduzir.

Figura 3.16. **Oscilações Indiretas**



Saída do mínimo ótimo para um subótimo. Se a mudança dos pesos se inicia numa área de gradiente muito grande, os primeiros ajustes podem ser excessivamente grandes e levar a rede a passar do vale onde está o mínimo global, para uma região com um mínimo local.

Figura 3.17.



3.3.6. Breve Comparação entre Redes-BP, redes-RBF e métodos simbólicos utilizando Nearest Neighbour

Nesta seção realizaremos uma breve comparação entre redes-RBF e redes-BP, extraída de [Haykin] e um comentário nosso sobre redes-RBF e sua relação com Nearest Neighbour e métodos que utilizam NN, como IBL*.

As redes-RBF e os perceptrons de múltiplas camadas treinados com algoritmo backpropagation (redes-BP) são ambos exemplos de redes feedforward não lineares treinadas através de aprendizado supervisionado, sendo aproximadores universais. Isto quer dizer, se existe uma função capaz de mapear o conjunto de vetores de entrada no conjunto de vetores de saída (classes) desejado, este mapeamento é garantido ser possível de ser aprendido, dispondo-se de memória e tempo de processamento suficientes.

Esta equivalência garante que sempre existirá uma rede-RBF capaz de reproduzir o comportamente de uma rede-BP específica e vice-versa. As diferenças entre os modelos são as seguintes:

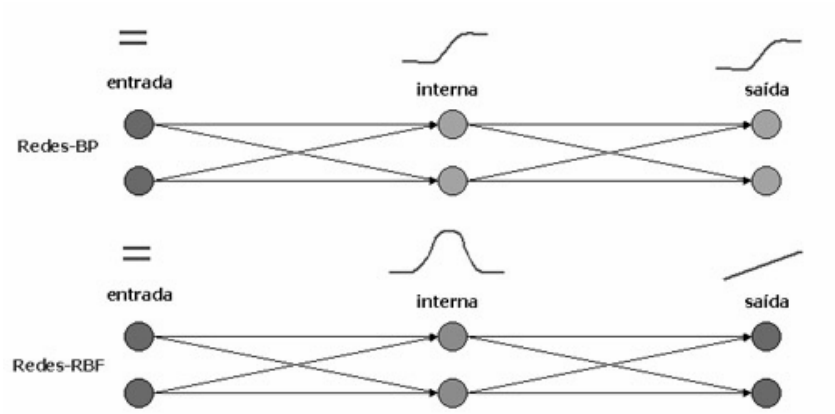
1. Uma rede-RBF clássica terá sempre uma única camada interna (*hidden layer*), enquanto uma rede-BP pode ter várias.
2. As redes-BP são homogêneas e todos os neurônios possuem o mesmo modelo, compartilhando a mesma função quaselinear de ativação, não importando em qual camada se encontram[1]. Os neurônios de uma rede-RBF possuem funções de ativação diferentes, sendo que a camada interna possue geralmente uma função gaussiana e a camada de saída uma função linear (que pode ser a identidade). Isto significa que a camada interna de ma rede-RBF é não linear, mas sua camada de saída é linear. No rede-BP todas as camadas são não-lineares.
3. A função de ativação dos neurônios da camada interna de uma rede-RBF calcula a **distância euclideana** entre o vetor de entrada e o centro daquele neurônio. A função de ativação de uma rede-BP calcula o **produto interno** do vetor de entrada pelo vetor de pesos daquele neurônio.

1. Com exceção da camada de entrada, que em ambos os modelos serve apenas para distribuir o sinal de entrada à camada interna e onde a função de ativação é sempre a função identidade.

4. Como conseqüência do ponto acima, as redes-BP constroem **aproximações globais de um único mapeamento de entrada-saída não-linear**, aproximando uma função que representa este mapeamento, enquanto que as redes-RBF constroem **aproximações locais para mapeamentos de entrada-saída não-lineares** utilizando modelos não-lineares com decaimento exponencial (p.ex. funções gaussianas) para este fim.

   Em resumo, uma rede-BP tenta encontrar uma única função não-linear capaz de representar o problema sendo treinado, enquanto que uma rede-RBF realiza um conjunto de aproximações locais não-lineares (uma por neurônio interno) baseadas em distância euclideana e delimitadas por uma função gaussiana que sejam linearmente separáveis e possam ser representadas pela função de mapeamento linear implementada entre a camda interna e a camada de saída.

Figura 3.18. **Visão geral da distribuição das funções de ativação em redes-BP e -RBF: Acima de cada camada está representado o tipo de função de ativação típico.**



Considere agora o seguinte: Sabemos que métodos utilizando Nearest Neighbour, como os algoritmos da família IBL ou kNN realizam um mapeamento muito parecido com a descrição das redes-RBF dada acima. A diferença está no fato de que não existe uma suavização da distância euclideana através de uma curva gaussiana (ou função similar). A não-linearidade que permite ao modelo baseado em NN representar problemas complexos é introduzida pela heurística de limiar móvel dada pela regra de escolha do protótipo mais similar. Traduzindo em outras palavras, o

modelo utilizado pelo algoritmo IBL, por exemplo, é um mapeamento local não linear porque, apesar da distância euclideana representar um mapeamento linear, o fato de aplicarmos uma decisão binária ao resultado da nossa comparação de distâncias (tomando o protótipo mais próximo) representa uma não-linearidade similar ao de um limiar (*threshold*).

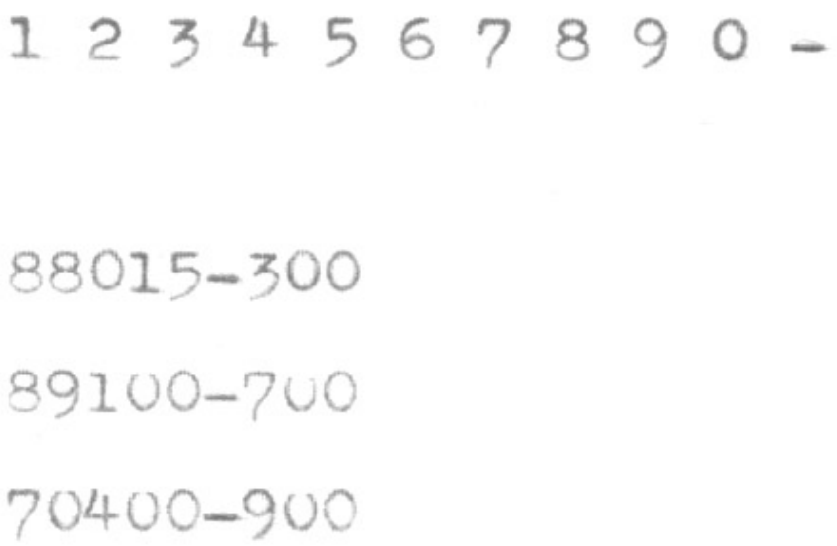
Sugerimos que você tome alguns problemas clássicos (espiral dupla, por exemplo) e outros com os quais você esteja familiarizado e realize uma comparação entre a performance de redes-RBF com várias topologias e a performance de vários modelos IBL, como IBL2, IBL3, etc. Qual a diferença no tempo de treinamento ? Qual a diferença na taxa de erros de classificação ? Qual a tolerância a erros ? Acreditamos que os resultados que você vai obter vão lhe ensinar bastante sobre reconhecimento de padrões.

3.4. Uma Aplicação de Reconhecimento de Padrões Simples: Reconhecer CEPs em Cartas

Tarefas que parecem mais complexas do que a classificação de dados sintéticos como o da espiral dupla, são, muitas vezes, bastante mais simples. Um exemplo é o caso do reconhecimento ótico de caracteres impressos. Suponha um sistema para litura automática do CEP em cartas. O sistema pressupõe que o CEP foi batido à máquina. Nosso exercício aqui pressupõe também que os passos iniciais, onde o CEP deverá ser identificado na carta, recortado da imagem da carta e normalizado para uma mesma resolução (número de pixel) são realizados por um algoritmo que não interessa no momento (vamos vê-los mais tarde no capítulo de processamento de imagens).

Um exemplo de um conjunto de números + hífem impresso por uma máquina de escrever e possíveis CEPs escritos com ela está abaixo.

Figura 3.19. **CEPs**



Para criar um pequeno sistema de OCR (reconhecimento ótico de caracteres) vamos supor que a primeira linha do escaneado na figura anterior será nossa amostra para treinamento. Para preparar esta amostra para que uma rede neural possa aprendê-la temos de primeiramente seguir os seguintes passos:

1. Decidir por uma representação. Neste caso vamos escolher representar os valores diretamente através de seus valores de pixel. Como a camada de entrada de uma rede-BP é unidimensional, vamos representar a matriz de uma imagem por linhas, cada linha ao lado da anterior.

Figura 3.20. **Os algarismos**

2. Aumento do contraste. As imagens da amostra (escaneadas com os tons de cinza do papel) são de baixa qualidade. Simplesmente aumentamos o contratste da imagem para tornálos mais visíveis.

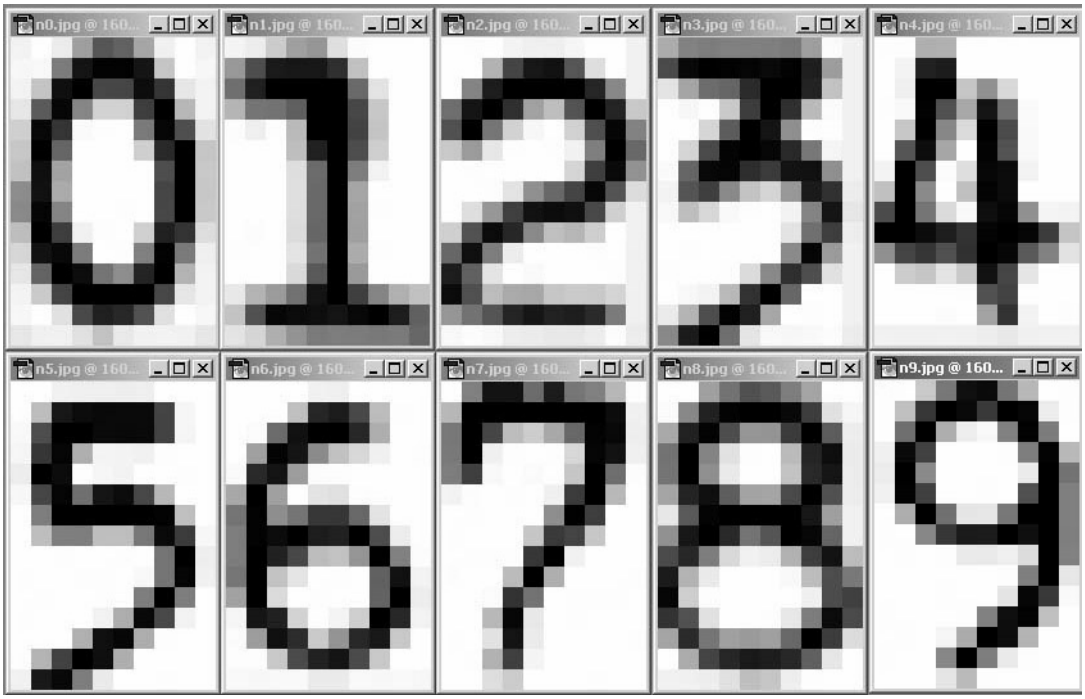
Figura 3.21. **Os algarismos com mais contraste**

3. Redução da resolução. Se pretendemos representar cada algarismo pelos seus pixels, não podemos fazer isto numa resolução onde cada algarismo ocupa uma matriz 70x100 ou similar. Temos de realizá-lo de forma simplificada. Fá-lo-emos inicialmente com uma matriz de aproximadamente 10x15.

Figura 3.22. **Os algarismos em baixíssima resolução**

4. Normalização: Como os tipos da máquina de escrever são de tamanho diferente, escolhemos um tamanho-meta e “normalizamos” as imagens para este tamanho. O tamanho-meta será 10x15 pixel, significando 150 neurônios na camada de entrada da rede. Agora tomamos cada número, recortamos exatamente os pixeis não brancos e redimensionamos a imagem resultante para 10x15.

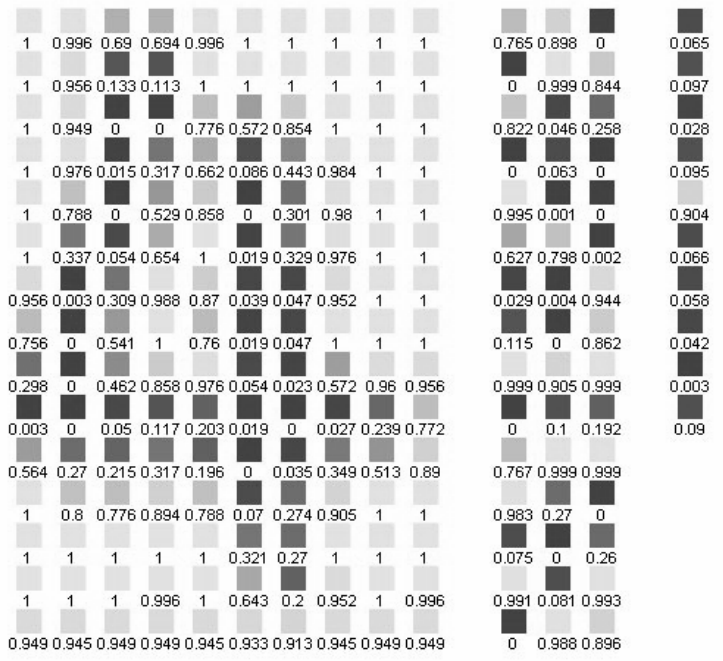
Figura 3.23. **Os algarismos normalizados**



Tudo o que temos de fazer ainda é criar arquivos de dados para treinamento da rede. Nós fizemos um método simples em Smalltalk que lê um arquivo de dados binário e incluios dados em um arquivo ASCII na sintaxe do SNNS. Como entrada para este método usamos um conjunto de dados gerado a partir das imagens acima salvando-as em formato .RAW (só bytes e mais nada) em um editor de figuras. A cada figura carregada neste método, associamos manualmente um valor de saída.

Agora podemos criar uma rede-BP no SNNS com 150 neurônios de entrada, 10 de saída e um número qualquer (no nosso exemplo: 30) na camada intermediária, como na figura abaixo:

Figura 3.24. **Ativação da camada de entrada ao apresentar-se um padrão**



3.4.1. Trabalho

Passado em sala de aula.

3.5. Agrupadores: Usando Aprendizado Não Supervisionado para Organizar Padrões

Já vimos anteriormente, quando discutimos o conceito de **aprendizado indutivo**, a diferença entre um ***classificador*** e um ***agrupador***. As redes-BP vistas anteriormente são o exemplo mais consagrado de modelos de redes neurais atuando como classificadores: a classe a que pertence cada padrão é um dado intrínsico do conjunto de treinamento e o que queremos que seja aprendido é exatamente um mapeamento entre a informação e a classe que associamos a ela de antemão.

Um outro grupo de problemas é aquele onde não sabemos de antemão a qual classe um padrão pertence nem quais são as classes em que o nosso problema se divide. Queremos que um mecanismo de reconhecimento de padrões seja capaz de detectar *semelhanças* entre padrões apresentados e que *agrupe* esses padrões durante o aprendizado de tal forma, que possamos utilizar o resultado do aprendizado de duas maneiras distintas:

a) como uma forma de *abstração* dos padrões apresentados, onde associamos cada *grupo* "descoberto" pelo método a uma *classe* ou *categoria* e

b) como um *classificador auto-organizante*, onde podemos utilizar a informação codificada durante o agrupamento dos padrões em categorias como mecanismo de classificação de novos padrões, apresentados em um estágio posterior.

A forma mais tradicional de se realizar esta tarefa é através da utilização dos métodos da Estatística Multivariada, principalmente da Análise de Agrupamentos e da Análise de Discriminantes. Estas são técnicas desenvolvidas mais ou menos durante a década de 1950 e possuem algumas limitações e algumas vantagens. Veremos isto no capítulo correspondente.

No campo das redes neurais existem três modelos clássicos de redes neurais que podem funcionar como agrupadores: *Competitive Learning*, os *Mapas Auto-Organizantes de Kohonen - SOM* e os modelos baseados na *Teoria da Ressonância Adaptativa - ART*. Desses três modelos, o modelo de Kohonen, também conhecido como Rede de Kohonen é o modelo matematicamente mais elegante e também o de maior aplicação prática. É este modelo que nós vamos ver como exemplo de agrupadores neurais nesta disciplina.

3.5.1. O Modelo de Kohonen e Quantização de Vetores

O pesquisador finlandês Teuvo Kohonen possui uma longa história de pesquisas no campo de modelos para descoberta de interrelacionamentos intrínsecos em distribuições de padrões, que se iniciou com pesquisas na área da Análise de Componentes Principais e quantização de atributos em conjuntos de vetores (pesquisa descrita em seu primeiro livro na área) e foi evoluindo no sentido de tentar descrever modelos cada vez mais plausíveis do ponto de vista biológico. O objetivo de suas pesquisas, na década de 1980, passou a ser o de descobrir um modelo de auto-organização de informações em um processo de aprendizado indutivo capaz de ser usado como modelo para o aprendizado e organização de informações no neocórtex cerebral de um animal superior [Koho88].

O modelo deveria ser capaz de explicar como estímulos similares são aprendidos e agrupados em áreas próximas no cérebro de um animal e levam a uma posterior categorização desses estímulos e à fomação subseqüente de um modelo de mundo, criando uma explicação implícita do que foi percebido através dos agrupamentos de estimulos relacionados em categorias na memória e realizando o processo inteligente da *abstração.*

Estas pesquisas foram inspiradas nas descobertas que as neurociências estavam realizando sobre o fato de que conceitos similares parecem estar representados em áreas próximas no cérebro humano e que essa localização espacial deveria ser uma característica do aprendizado. Em função desse objetivo inspirado em modelos biológicos de aprenidzado, o modelo de Kohonen é talvez o modelo de rede neural mais próximo de um modelo de aprendizado biológico[1].

Kohonen publicou vários artigos sobre o seu modelo, cada qual descrevendo-o sob aspectos um pouco diferentes. Talvez o mais importante desses artigos, e um dos menos conhecidos, seja o artigo publicado por

1. Mesmo assim, a plausibilidade biológica é refutada por um passo muito importante do algoritmo de Kohonen: a determinação do vencedor. Este passo do aprendizado de Kohonen poderia ser explicado através de um mecanismo de emissão de neurotransmissores inibidores por parte de um neurônio ativado de forma muito forte por um estímulo externo (o vencedor). Esta explicação não serve como modelo, porém, para uma determinação global do vencedor, pois um efeto desses, caso existisse (o que não ocorre, ao que se sabe), seria necessariamente local.

Kohonen e Helge Ritter, da Universidade de Munique, em 1989 na revista *Biological Cybernetics* [KR89]. Neste artigo eles descrevem o modelo em detalhes, explicando o conceito de vizinhança e da função de vizinhança e aplicam o modelo a dois exemplos, um de aprendizado indutivo de conceitos e outro de aprendizado lingüístico. É o único artigo escrito por Kohonen onde a função de vizinhança é discutida em detalhes. O fato de esta revista na época ser lida quase exclusivamente por um público de biólogos e ciberneticistas, tornou a função de vizinhança utilizada por Kohonen um dos aspectos menos conhecidos de seu modelo e em muitos livros sobre redes neurais essa função sequer é citada.

Consideramos o artigo de Kohonen e Ritter um marco tão importante na história das redes neurais e uma explicação tão perfeita sobre o modelo, que vamos reproduzi-lo aqui na íntegra, através da sua tradução para o Português realizada por Maricy Caregnato e Emerson Fedechen, do CPGCC da UFSC. Esta tradução será entremeada de comentários nossos e de exemplos de reprodução dos experimentos de Kohonen e Ritter com o SNNS.

3.5.2. Os Mapas Auto-Organizantes de Kohonen

Teuvo Kohonen e Helge Ritter
Biological Cybernetics, 61, 241-254, Elsevier, Amsterdam, 1989

Tradução: Maricy Caregnato e Emerson Fedechen, CPGCC, UFSC.

3.5.2.1. Resumo

A formação auto organizável de mapas topográficos para dados abstratos, tais como palavras, está demonstrada neste trabalho.Os relacionamentos semânticos nos dados são refletidos por suas distancias relativas no mapa. Duas simulações diferentes baseadas em modelos de redes neurais que implementam o algoritmo de mapas de atributos auto organizáveis são demonstrados. Para ambas o novo ingrediente essencial é a inclusão de contexto no qual cada símbolo aparece dentro de dados de entrada. Isto habilita a rede neural a detectar a "similaridade lógica" entre palavras na estatística de seus contextos. Na primeira demonstração o contexto simplesmente consiste de um conjunto de valores de atributos que ocorrem em conjunção com as palavras. Na segunda demonstração, o

contexto é definido pelas seqüências nas quais as palavras ocorrem, sem considerar nenhum dos atributos associados. Proposição verbal simples consiste de substantivos, verbos e advérbios tem sido analisados dessa forma. Frases ou cláusulas envolvem algumas dessas abstrações que aparecem no pensamento, isto é, a categoria mais comum, nas quais as palavras são agrupadas automaticamente em ambas simulações. Também argumentamos que um processo similar pode estar no funcionamento do cérebro.

3.5.2.2. Hipóteses sobre a representação interna de Elementos da lingüística e estruturas

Um dos problemas mais intrigantes na teoria de redes neurais artificiais e biológicas , é dimensionar um simples sistema adaptativo para tornar-se hábil a encontrar abstrações, invariâncias, e generalizações de dados crus.

Muitos resultados interessantes em reconhecimento de padrões (percepção artificial de imagens, acústica, e outros padrões) já tem sido obtido. Extração de atributos de elementos de dados relatados geometricamente ou fisicamente, contudo, ainda é tarefa muito concreta, pelo menos no princípio. Um objeto de estudo quanto mais abstrato e enigmático processa informação cognitiva que divide com elementos de conhecimento e seus relacionamentos; isto é freqüentemente identificado com a capacidade de usar linguagens. O propósito do presente trabalho é estudar se isto é possível para criar abstrações em redes neurais artificiais, tal que elas, pelo menos na forma primitiva, refletiriam algumas propriedades de representações cognitivas e lingüísticas e relações.

Em particular estamos mostrando aqui novos resultados que demonstram que um processo auto organizável está realmente apto a criar uma rede neural topograficamente ou geometricamente organizando mapas que mostram relações semânticas entre dados simbólicos. Isto pode ser adequado para chamadas como representações de mapas semânticos auto organizáveis.

Estamos relatando nossos resultados para a base fundamental da cognição , chamada, categorização de observações. Como as conexões

dessas idéias de teorias fundamentais de conhecimento podem por outro lado permanecerem obscuras, isso pode ser próprio para começar com uma pequena revisão de um fundo filosófico, chamada, a teoria das categorias como o último framework de abstração.

3.5.2.2.1. Categorias e suas relações para representações neurais e lingüísticas

Os conceitos mais gerais de abstração que são necessários para representar o mundo empírico são chamadas categorias; elementos de redução básica e formas de pensamento e comunicação podem também ser encontrados em todas as linguagens primitivas como também as mais desenvolvidas.

As categorias estão supostas a abranger todo o domínio de conhecimento, e parta formar as bases de conhecimento. Aristóteles de fato já distinguiu dez categorias. As mais comuns de todas são: 1) Itens (objetos), 2) Qualidades (propriedades) 3) Estados (ou mudanças de estado) 4) Relacionamentos (espacial, temporal e outros).

Nas linguagens a categoria 1 corresponde aos substantivos, a categoria 2 aos adjetivos, e categoria 3 aos verbos. Para a representação da categoria 4, diferentes linguagens usam advérbios, preposições, pós posições , pontos finais, inflexões, ou sintaxe (ordem das palavras). Naturalmente muitas classes de palavras auxiliares são necessárias para inter relatar frases e cláusulas, para indicar modalidades lógicas, como também para facilitar inferência dedutiva e indutiva.

O profundo significado original metafísico de "categoria" foi perdido no uso comum desta palavra. "Categorias" são freqüentemente identificadas como classes de itens como animais , plantas, etc. Mais exatamente tais classes somente constituem subcategorias da Categoria1.

Desde que representações de categorias ocorreram em todas as linguagens, muitos recursos tem estipulado que os elementos semânticos mais profundos de uma linguagem podem ter uma representação fisiológica em um domínio neural; e se eles são independentes de uma história cultural diferente, isso conclui que tais representações devem ser herdadas geneticamente.

Na época que a predisposição genética de elementos de linguagem foi sugerida, não havia mecanismo conhecido que teria explicado as origens das abstrações em informações neurais processada outra então evolue.. Isto não foi desde que a modelagem "redes neurais" alcançasse o nível presente quando pesquisadores começaram a descobrir de propriedades abstratas de representações internas dos sinais de modelos na rede física. Lá existe pelo menos duas classes de modelos com este potencial: a rede backpropagation e a map self-organizing. O encontrado indica que as representações internas de categorias podem ser deriváveis de relações e regras mútuas de um sinal primário ou elementos de dados.

Contudo o propósito deste paper não é afirmar que todas as representações no cérebro biológico somente são adquiridas pelo aprendizado. Os princípios adaptativos discutidos abaixo podem ser considerados como frameworks teóricos, e a primeira faze do aprendizado é a forma mais simples. É totalmente possível que um processo similar esteja trabalhando em um ciclo genético, por outro lado esses mecanismos explícitos são difíceis para imaginar.

Isso agora será próprio para abordar o problema de mapas semânticos auto organizáveis usando dados que contém informações implícitas relatando simples categorias; se mais tarde forem detectados automaticamente , podemos pensar que o passo significante em direção ao processamento lingüístico auto organizável foi feito.

Um aspecto pode ser ainda enfatizado. Isso talvez não seja razoável para procurar por elementos de linguagens no cérebro. A visão mais fundamental é que as funções fisiológicas são esperadas para refletir a organização categórica e não tanto as formas lingüísticas detalhadas.

3.5.2.2.2. Exemplos de modelos de redes neurais para representações internas

Redes semânticas.

Para a materialização reta de representações internas a rede semântica foi sugerida. Na sua forma original elas compreendem uma estrutura gráfica com nodos e links. Os nodos podem ser itens ou conceitos (grupos de atributos), enquanto os links indicam relações: as mais

simples relações binárias representam as co-ocorrências de itens em eventos observados, links rotulados descrevem suas relações qualificadas. As redes semânticas supostamente tem a contrapartida um por um em células neurais e suas interconexões. Por onde um processo de busca seria interpretado como ativação expansiva nessa rede neural. Na visão neuro-fisiológica contemporânea dado como grau de especificidade a resolução espacial é altamente improvável em biologia. Mais um tem que compreender que modelos de rede neural do cérebro , significam semântica de predisposição para os nodos e links terem sido postulados; como um "mapeamento " não é derivado de nenhum processo auto organizável.

Camadas internas em redes-BP

Se na atualidade famílias de redes neurais "feedforward" com errros de propagação de fundo significa que podem ser considerados como modelos biológicos ou não, células ou nodo nas suas camadas escondidas freqüentemente parecem aprender respostas que são específicas para algumas qualidades abstratas de informações de entrada. Contudo, deve ser enfatizado que a propagação de fundo é crucialmente baseado em aprendizado supervisionando. O estímulo de saída em relação ao de entrada, são forçados para dar valores por otimização de parâmetro de pesos internos dos nodos na rede . Em uma rede multi-nível com dados estruturados pode acontecer que para alcançar a otimização global , alguns nodos de camadas internas tornam-se afinados para representar alguns tipos de "eigendata" de sinais que ocorrem, que representam a "generalização" ou "abstrações ". Foi demonstrado recentemente que os pesos de vetores da camada escondida podem convergir para valores que codificam itens lingüísticos de acordo com suas regras semânticas . Essas regras estão definidas explicitamente no processo de aprendizagem.

Mapas de auto organização (características topológicas)

A forma mais genuína de auto organização é o aprendizado competitivo que tem a capacidade de encontrar agrupamentos das informações primárias , eventualmente em modo de organização hierárquica. Em um sistema de características de células sensitivas o aprendizado competitivo significa que um número de células está comparando os mesmos sinais de entrada com seus parâmetros internos , e a célula com o melhor competidor (winner) é então auto ajustada a esta entrada. Desta forma diferentes células aprendem diferentes aspectos da sua entrada , que podem ser

considerados como a mais simples forma de abstração.O mapa de auto organização é um adiantado desenvolvimento do aprendizado competitivo em que a célula de melhor entrada também ativa seus vizinhos topográficos na rede para fazer parte no afinamento da mesma entrada. Um acerto, não significa resultado óbvio coletivo , o aprendizado coletivo assume a rede neural como uma falha de duas dimensões. As diferentes células tornam-se ajustados a diferentes entradas em uma moda ordenada , definindo características de sistemas de coordenadas através da rede. Após o aprendizado, cada entrada obtém uma resposta localizada , qual posição no papel reflete a mais importante "coordenada característica"da entrada. Isso corresponde a uma projeção não linear do espaço de entrada na rede que faz a melhor relação de vizinhança entre elementos explícitos geometricamente. Particularmente se os dados são agrupados hierarquicamente , uma representação muito explícita está localizada na mesma estrutura gerada. Enquanto mapas auto organizáveis como foram usados para muitas aplicações para visualizar dados agrupados , muitas possibilidades intrigantes são diretamente possíveis de criar um processo de representação topográfica da semântica de relação não métrica implicando em dados lingüísticos.

3.5.2.3. As funções de processamento da informação estão localizadas no cérebro? Justificação do modelo.

Contra retirada geral e a favor da localização

Geralmente a psicologia comportamental enfatiza a natureza globall e holística do mais alto processamento de informações humana. Algum procura neurofisiológica encontradas realmente precisa suportar essa visão. Distribuição de resultados de aprendizagem na massa celular do cérebro foi descoberta em experimentos clássicos de de Lashley em 1938, que por um longo tempo o cérebro foii interpretado como sendo uma caixa preta com mais ou menos componentes eqüipotenciais que podem ser repassados aos outros. Uma visão extrema carrega todas as tentativas para isolar e localizar funções cognitivas no cérebro como uma maneira moderna de fenologia.

È verdade que em um processo que conduz a percepção ou ação, muitas partes do cérebro estão envolvidas em uma iteração ou moda recursiva.

Isto contudo, poderia ser dito de algum dispositivo ou mecanismo que foi designado para representar uma tarefa particular, , e precisa da cooperação de todos estes componentes. Contudo, isso seria absurdo negar, na visualização de dados neurofisiológicos, o cérebro contém partes, redes, e mesmo simples células neurais que representam funções parciais específicas. Lá existem registros de vários tipos de células de atributos sensitivos ou lugares que respondem a qualidades específicas de estímulo sensorial, e o neurônio motor que controla músculos particulares são localizados certamente. As funções globais obviamente seguem da cooperação de componentes muito grandes desse tipo. A quantia de paralelismo e redundância no processamento podem ser enormes. No resto da questão somente interessa o grau ou perspicácia da localização, como também uma organização hierárquica possível de tais funções localizadas.

Técnicas para determinar localização e suas críticas

No final do século IXX, a organização topográfica detalhada do cérebro, especialmente o cortex, já foi deduzível de déficits funcionais e falhas comportamentais que foram induzidas por vários tipos de defeitos causados acidentalmente, adequado para tumores, mal formações , hemorragias ou lesões causadas artificialmente . Uma técnica moderna causa lesões controláveis e reversíveis, é para estimular uma parte em particular na superfície cortical por pequenas correntes elétricas, através disso eventualmente induzem efeitos inibitórios e excitatórios, mas de qualquer forma uma função local assume um distúrbio. Se tal estímulo confinado globalmente então sistematicamente interrompe uma habilidade cognitiva específica tais como objetos, lá existe a menor indicação que o lugar correspondente é essencial para aquela tarefa. Esta técnica foi criticada freqüentemente pelo fato que carrega para todos os estudos nas lesões. Por outro lado uma lesão similar no mesmo lugar sempre causaria a mesma deficiência, e a mesma deficiência nunca foi produzida por um outro tipo de lesão, ela não é logicamente possível usar como dado como uma prova conclusiva para localização; a parte principal da função pode residir em outro lugar, enquanto a lesão pode destruir somente uma conexão do controle vital para ela. Hughlings Jackson já declarou "Para localizar os danos que destroem a fala e para localizar a fala são duas coisas diferentes "

Uma outra forma controlável para a determinação da localização é comprimir quimicamente ou herdar o processo que causa o engatilhamento dos neurônios , ou seja, usar pequenos retalhos embebidos em striquinina. Esta técnica foi usada com sucesso para mapear, isto é, funções sensoriais primárias.

O método mais simples é localizar uma resposta para armazenar o potencial ou encadeamento de impulsos neurais associados com ele. Apesar de desenvolver técnicas multi-eletródo geniais, este método não detectou todas as respostas em uma área desde que o encadeamento neural seja homogêneo , a união faz um neurônio particular ser mais eventual, especilamente de um sensor primário e de áreas associativas , foi feito por várias técnicas registradas eletrofisiológicamente. Evidencias mais conclusivas para localização podem ser obtidas por modernas técnicas imaginárias que mostram diretamente a distribuição espacial da ativação do cérebro associado com a função alcançando uma resolução espacial de alguns milímetros. Os dois métodos principais que são baseados em traçadores radioativos são eles: Positron Emission Tomography(PET), e auto radiografia do cérebro através de conjuntos de colimadores muito pequenos (câmara gama). PET revelam mudanças no uptake oxigênio metabolismo fosfato. O método de câmara gama detecta mudanças diretamente no fluxo sanguíneo cerebral. Os fenômenos correlate com a ativação neural local, mas eles não estão hábeis a seguir rapidamente os fenômenos. Em magnetoencephalography (MEG), o baixo campo magnético causado por respostas neurais é detectado, e por computação desses recursos, as respostas neurais podem ser diretamente ser analisada com razoável rapidez , com uma resolução espacial de junção de milímetros. A principal desvantagem é que somente tais dipoles atuais são detectáveis, as que estão em paralelo na superfície do crânio, isto é, principalmente o silco do córtex que pode ser estudado com este método.

Parece existir uma técnica não ideal que sozinha seria usada para mapear todas as respostas neurais. Ela é necessária para combinar estudos anatômicos, eletrofisiológicos, imaginários e histoquímicos.

Mapas topográficos em Áreas sensoriais

Genericamente, dois tipos de mapas fisiológicos são distinguíveis no cérebro: aqueles que soa claramente ordenados, e aqueles que são quase

randomicamente organizados, respectivamente. Mapas que formam uma imagems contínuas ordenada de algumas "superfícies receptivas" podem ser encontradas na visão, e córtices somatosensoriais no cerebelo , e em certo núcleo. A escala local no fator de sublimação desses mapas depende da importância comportamental de sinais particulares , ou seja, imagens de parte foveal da retina , a ponta dos dedos e os lábios são sublimes em relação as outras partes. Há assim um mapeamento "quasi-conformal"da "superfície" dentro do cérebro.Também há mais mapas abstratos, ordenados, contínuos em muitas outras áreas sensoriais primárias , tais como o tonotopic ou mapas de freqüência auditiva. Isso é uma característica comum de tais mapas que são confinados para uma área menor, raramente excedendo 5mm de diâmetro, como isso é justificado para usar o modelo dela no qual a rede total é assumida homogeneamente estruturada. Sobre uma área , um mapeamento espacialmente ordenado ao longo de uma ou duas dimensões de atributos importantes de um sinal sensorial é usualmente discernível.

Fisiologistas também usam a palavra "mapa" para respostas não ordenadas para estímulos sensoriais contanto que estes sejam localizáveis espacialmente, até se eles forem randomicamente dispersos em cima de uma área de vários centímetros quadrados e muitos tipos diferentes de respostas forem encontrados na mesma área. Respostas visuais mais complexas encontradas em níveis mais altos são mapeadas desta forma: por instância, células foram detectadas respondendo seletivamente a faces.

Evidências para localização de função lingüística :

Foi conhecido no início do século que a afasia sensorial é causada por lesão nas parte superior e posterior do lobo temporal no cérebro chamada área de Wernicke; mas com técnicas modernas de tratamento de imagem somente uma localização muito mal feita de funções da linguagem tem sido possível. Praticamente toda a função sistemática de alta resolução mapeada foi feita por um método de simulação.

É muito mais difícil localizar lingüísticas em funções semânticas no cérebro do que para mapear as áreas sensoriais primárias. Primeiro, ele ainda não esta claro para quais aspectos da linguagem as dimensões características podem corresponder. Segundo, como foi notado recentemente como um mapeamento pode ser disperso.Terceiro, resposta para

elementos lingüísticos podem somente ocorrer dentro "time windows". Quarto, as técnicas experimentais usadas em animais estudados sendo usualmente evasivos, não podem ser aplicados a seres humanos, a menos que exista indicação de uma operação cirúrgica. Contudo, o significado entre evidencias experimentais já é avaliável suportando a visão do grau mais alto da localização nas funções da linguagem.

PET da imagem tem revelado que durante a tarefa de processar simples palavras , diversos lugares de cortes corticais são ativados simultaneamente. Estes não estão todos localizados na área de Wernicke :algumas partes do lobo frontal e as áreas associativas podem mostrar respostas simultaneamente também, especialmente em locais obviamente associados com percepção visual e auditiva , articulação e planejamento de tarefas.

Ao invés de estudar representações internas , localização de lugares relacionados a processos semânticos precisam de melhor resolução ao invés de um milímetro tão difícil de registrar mesmo por estímulos de mapas, entretanto este método não pode detectar algum pico de atividade temporal, isso pode apenas produzir bloqueio temporário reversível do processo em uma região confinada a um milímetro quadrado. Estimulações repetidas da mesma área causa uma espécie de deficiência temporária , isto é, erros em nomear objetos, ou dificuldade em recolecionar da memória de padrões verbais curtas. Contudo, a estimulação de algumas outra áreas apenas 5mm já separados podem induzir tipos completamente diferentes de deficiência ou sem efeito algum. Adicionalmente estes são casos de pacientes bilíngües onde nomeados pelo mesmo objeto e prejudicado em apenas uma das linguagens dependendo da área que está sendo estimilada. Isso parece como se a função da linguagem fosse organizada como um mosaico de módulos localizados .

Outra evidência indireta para um mapeamento estruturado está disponível em diversos casos nas deficiências selecionadas como resultado de pancadas ou cérebros feridos. Exemplos incluem deficiências no uso de palavras concretas por abstratas , inamimado por animado ou deixando objetos e comida contra palavras animadas. Lá existe relatório bem documentado em impairements seletivos relatando quais subcategorias como objetos internos , partes do corpo, frutas, vegetais.

Análise de qual informação tem direcionado a conclusão que existe módulos separados no cérebro por uma "palavra lexicamente visual" e a palavra lexicamente fonética para reconhecimento da palavra em semântica léxica para o significado da palavra como uma saída léxica para palavras articuladas, respectivamente cada um desses módulos pode ser independentemente falho.

As falhas categoricamente relatadas acima parecem relatar danos causado seletivamente para a "léxica semântica ". Estas observações não podem prover evidências conclusivas para a localização de classes semânticas sem a léxica, porque em todos esses casos não foi possível avaliar a extensão espacial precisamente no tecido afetado no cérebro. Nonetheles isso parece justificado para aquele estado de falha seletiva em que um grande número de casos, seria muito difícil explicar se a organização semântica aparente da observação não estivasse em alguma forma ponderada no layout espacial do sistema.

3.5.2.4. Representação de dados topologicamente relacionados em um mapa auto organizável

Algum modelo sugerido para a formação auto organizável de representações internas (como células de características sensitivas) precisa também estar apto para fazer relações essenciais entre itens de dados explícitos. Uma forma intrigante de alcançar isso é a formação de mapas espaciais, que talvez sejam o local mais conhecido de representações.

Vários anos atrás , um dos autores (Kohonen) desenvolveu um modelo de adaptação neural capaz de fazer formação não supervisionada de mapas espaciais para vários tipos diferentes de dados. Nesta seção primeiro mostraremos o modelo de equação (simplificado) e então explicaremos como um mapa de estrutura preservada de dados relatados hierarquicamente é gerada por ele. Descrição mais detalhada do processo e seu fundamentos podem ser encontradas na publicação original e também alguns desenvolvimentos recentes (Kohonen 1982 a-c, 1984; Cotrell and Fort 1986; Ritter and Schulten 1986, 1988, 1989)

O modelo assume um conjunto de neurônios adaptativos interagindo lateralmente, geralmente arranjados como uma lâmina em duas dimensões.Os neurônios são conectados como um feixe comum de fibras de

entrada. Algum padrão de atividade surge nas dadas fibras de entrada para a excitação de algum grupo de neurônios locais. Depois do aprendizado, as posições espaciais de grupos específicos excitados em um mapeamento de padrões de entrada em uma lâmina bidimensional , o último tendo a propriedade do mapa topográfico, isto é, ele representa as relações de distancia de alta dimensão do espaço dos sinais de entrada aproximadamente como distancia de relacionamento nas laminas neurais bidimensionais. Esta propriedade considerável segue de interações laterais assumidas e bastante simples de baixa adaptação biologicamente justificada. De fato, parece que os requerimentos principais auto organizáveis são: (I) os neurônios são expostos a um número suficiente entradas diferentes (II) para cada entrada, as conexões de entradas sinápticas somente o grupo excitado é afetado (III)atualização similar é imposed em muitos neurônios adjacentes e (IV) o resultado ajustado é tal que o aumento da mesma resposta para a subseqüente , entrada similar suficiente.

Matematicamente o padrão de atividade das entradas está described por um vetor x n-dimensional onde n é o número de linhas de entrada. A resposta do neurônio r é especificada por um vetor w_r n-dimensional , eventualmente correspondendo ao vetor ao vetor de eficácias sinápticas e isso é medido pelo produto $x|w_r$.

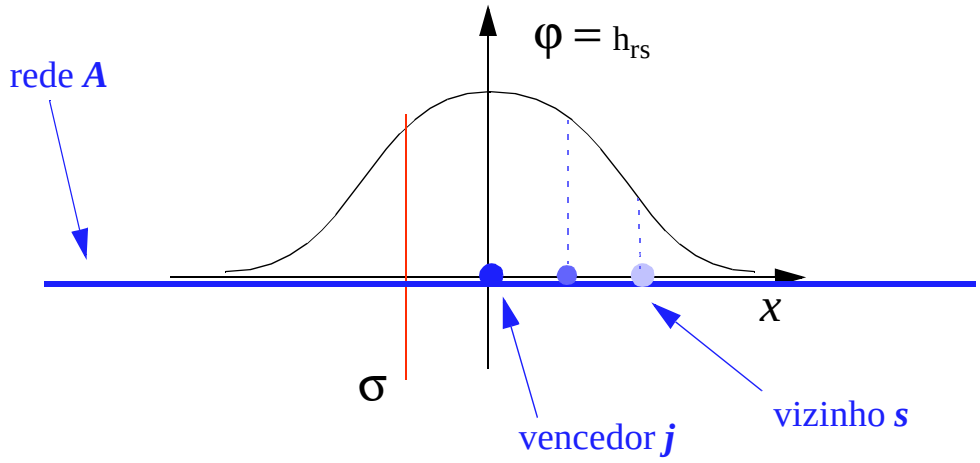
Para a eficiência do processo e conveniência matemática, todos os vetores de entrada são sempre normalizados para tamanho único, considerando que o w_r não precisa ser normalizado explicitamente no processo , cedo ou tarde o processo os normalizará automaticamente. Os neurônios estão arranjados em uma grade bi-dimensional, e cada neurônio está rotulado pela sua grade bi-dimensional de posição r. O grupo de neurônios excitados é escolhido para estar centralizado no neurônio s para que x. ws seja o máximo. Esta forma e extensão são descritas por uma função h_{rs} , cujo valor é a excitação do neurônio r se o centro do grupo estiver em s. Esta função pode ser constante para todo o r em uma "zona de vizinhança" em torno de s e zero, como em uma simulação presente em que são supostas para descrever o mapeamento mais natural. Neste caso hrs será o maior em r=s e declínio para zero

com distância decrementada ||r-s||. A melhor modelagem realista escolhida para h_{rs} é:

$$h_{rs} = e^{-\frac{\|r-s\|^2}{\sigma(t)^2}} \qquad (1)$$

isto é , a distancia Gaussiana ||r-s|| cuja variância $\sigma^2/2$ controlará os radianos do grupo.

Figura 3.25. **Forma da função de vizinhança**



Os ajustes correspondentes para a entrada *X* devem ser dados por:

$$W_r^{novo} = \lambda h_{rs}(X - W_r^{velho}) \qquad (2)$$

A equação (2) pode ser justificada assumindo a tradicional lei de Hebb para modificações sinápticas, e um processo adicional "active" não linear de esquecimento para a força sináptica. A equação (2) foi desejada propriamente de algumas adaptações de confinamento para a vizinhança do neurônio s e responde melhor ao x.

Nós devemos presentear aqui alguma prova para que estas condições realmente conduzem para uma organização ordenada do mapa. Para o

presente propósito é suficiente dizer que os mapas resultantes são projeções não lineares no espaço de entrada nessa superfície com duas propriedades: (I) os relacionamentos de distância entre a fonte de dados são preservados pelas suas imagens no mapa tão fielmente quanto possível. Contudo, um mapeamento de um espaço alto-dimensional para um baixo-dimensional geralmente alterará mais distancias e só preservará o mais importante relacionamento de vizinhança entre os itens de dados , isto é a topologia de suas distribuições. Este é o fator comando da informação de uma representação reduzida em que detalhes irrelevantes são ignorados. (II) Se diferentes vetores de entrada aparecem com diferentes freqüências , o mais freqüente será mapeado para domínios maiores a custa das menos freqüentes.

Estes resultados em uma localização muito econômica de recursos de memória para itens de dados concessões com descobrimentos fisiológicos.

Se os dados formam agrupamentos no espaço de saída, isto é, se há regiões com várias freqüências e ao mesmo tempo muitos dados similares (I) e (II) assegurará que os dados do agrupamento são mapeados para um domínio de localização comum no mapa. Além disso, o processo arranjará a mútua posição desses domínios em que a forma para capturar a topologia completa do agrupamento é possível. Desta forma, como agrupamentos hierárquicos podem ser melhorados, uma pensamento freqüentemente capaz de representar uma forma de abstração.

3.5.2.5. Mapas de auto organização semântica.

Simbologia de mapas de auto organização.

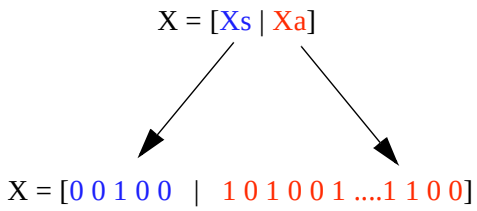
Na demonstração descrita em (kohonen 1982c) e (kohonen 1984), os mapas de auto organização principalmente refletem as relações de distancias métricas entre vetores de representações de padrões. Como informações são características dos mais baixos níveis de percepção, em linguagem particular e raciocínio, parece sobrar no processo de símbolos discretos. daqui nós devemos entender como o cérebro pode representar entidades simbológicas . Em vista da localização parecer vista neste nível, nós devemos particularmente expor como o mapea-

mentos de símbolos pode ser formada em qual relação lógica ocupa lugares vizinhos.

Um pensamento pode aplicar as leis de adaptação neuronal a um conjunto de símbolos que podem criar um mapa topográfico que mostra a distância lógica entre os símbolos como comparados em dados contínuos. Para a similarieade mais tarde sempre mostrar de uma jeito natural, como diferenças métricas entre seus códigos contínuos. Isto não é mais verdadeiro para a simbologia de itens discretos, como palavras para quais nenhuma métrica foi definida.

Isto não é verdade para discrição, itens simbólicos, como palavras, para as quais nenhuma métrica foi definida. Isto está no mais natural símbolo que seu significado é dissociado do seu código. Daqui a relação lógica entre diferentes símbolos deseja-se em geral não ser diretamente detectáveis pelos seus códigos e não pode assim presumir nenhuma relação métrica entre os símbolos, mesmo quando eles representam itens similares. Como seria então possível mapeá-los topograficamente? A resposta é que o símbolo, ao menos no processo de aprendizagem poderia ser frequentemente apresentado em contexto semelhante, i.e. em conjuctura com todos ou parte dos valores atribuídos ao item que ele codifica, ou com outro, correlacionando símbolos.

O modelo básico do sistema para mapas simbólicos aceita cada dado do vetor x como uma concatenação de dois (ou mais) campos, um especificando o código simbólico, denotado por xb e o outro, o conjunto de atributos, denotado por xa, respectivamente.



(3)

A equação 3 ilustra em equação vetorial que a decodificação da parte simbólica e a parte atributo pode formar um vetor somado com dois componentes ortogonais. A idéia central de mapas simbólicos, é que as duas partes são ponderadas apropriadamente como a norma da parte

atributo predominada sobre a parte simbólica durante o seu processo de organização; o mapeamento topográfico desse momento principalmente reflete os relacionamentos métrico do conjunto de atributos. Deste modo, as entradas para sinais simbólicos, de qualquer forma, são ativados todo o tempo, traços de memórias deles são formadas para a entrada correspondente de outras células do mapa que foi selecionado (ou atualmente forçado) pela parte atributo. Se então, durante a recognição de dados de entrada, o sinal dos atributos são perdidos ou estão muito fracos as mesmas unidades do mapa são selecionadas à base da parte simbólica unicamente. Deta forma os símbolos vêm codificados dentro de um ordem espacial refletindo suas similaridades lógicas (ou semânticas).

Atributos podem ser variáveis com valores escalares discretos ou valores contínuos ou eles podem alcançar propriedades qualitativas como "bom" ou "ruim". Isto é simplesmente para assumir que a identidade de cada atributo é clara nas suas posições no "campo atributo" xa, por meio do qual a presença ou falta de uma propriedade qualitativa pode ser indicada por um valor binário, dizendo 0 ou 1 respectivamente. Então a (desnormalizada) similaridade entre dois conjuntos de atributos podem ser definidos em termos do número de atributos comuns para vários conjuntos, ou equivalências, como produto ponto dos respectivos vetores atributos.

Para ilustrar isto com um modelo concreto de simulação, considere o dado fornecido na fig.1. cada coluna é uma muito esquemática descrição de um animal, baseado na presença (=1) ou falta (=0) ou algum dos 13 diferentes atributos dados à esquerda. Alguns atributos, como "penas" e "2 pernas" são combinados, indicando mais diferenças significantes que os outros atributos.

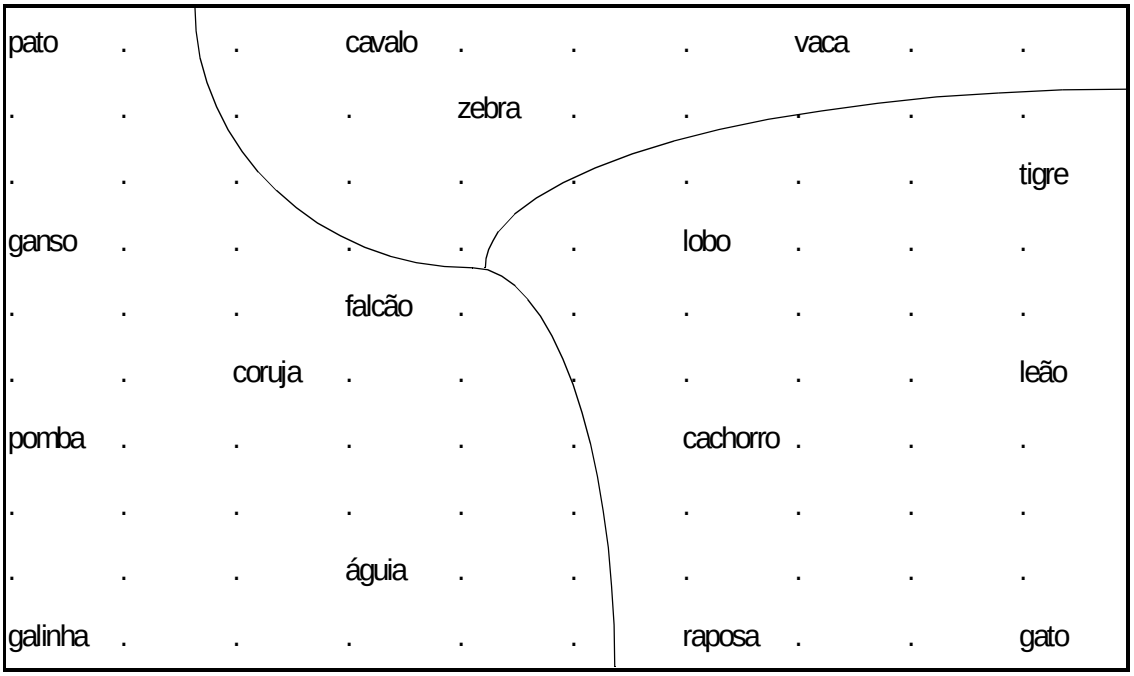
A seguir, nós vamos pegar cada coluna para o campo atributo xa do animal indicado no topo. O próprio nome do animal não pertence a xa mas ao invés disso especifica a parte do símbolo xs do animal. Selecionar o código do símbolo pode ser feito de uma varidade de formas. Entretanto, nós agora queremos ter certeza que o código dos símbolos indiquem alguma informação sobre similaridades entre os itens. Daqui nós escolhemos para a parte simbólica do k-th animal um vetor d-dimensional, o qual k-th componente tem um valor fixo de a, e dos

Figura 3.26. **Nomes de animais e seus atributos**

		pomba	galinha	pato	ganso	coruja	falcão	águia	raposa	cachorro	lobo	gato	tigre	leão	cavalo	zebra	vaca
é	pequeno	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	médio	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	grande	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
tem	2 pernas	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	4 pernas	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	pêlo	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	cascos	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	crina	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	penas	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
gosta	caçar	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	correr	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	voar	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
	nadar	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

quais componentes remanescentes são zeros. Este d é o número de itens (d = 16 em nosso exemplo). Para esta escolha, a distância métrica entre dois vetores xs é o mesmo, irrespectivo dos símbolos codificados. O parâmetro a pode ser interpretado como medindo a "intensidade" de entrada dos campos simbólicos e isso determina a realtiva influência da parte simbólica comparada com a parte atributo. Como nós procuramos o último que irá predominar, nós escolhemos um valor para a = 0.2 para nossa simulação. Combinando xa e xs de acordo com (3), cada animal foi codificado por um 29-dim vetor de dados x = [xs, xa]t (*elevado a t*). Finalmente cada vetor de dado foi normalizado a um único tamanho. Embora isso é apenas um significado técnico para garantir uma boa estabilidade no processo de auto-organização, sua contraparte biológica

Figura 3.27. **Depois que a rede treinou com os dados de entrada, apresenta-se os nomes dos animais isoladamente. Um agrupamento de acordo com a similaridade é gerado.**



poderá ser intensificada a normalização dos padrões de atividade de entrada.

Os membros do conjunto de dados assim obtidos foram apresentados iterativelmente e em uma ordem randomica para uma rede planar de 10 x 10 neuronios ´sujeita a um processo de adaptação descrito a seguir. A conecção inicial força entre os neurônios e seus n = 29 linhas de entrada onde são escolhidos os pequenos valores randomicos. i. e. nenhuma ordem prioritária foi imposta. Entretanto depois de um processo de 2000 apresentações, cada "célula" torna-se mais ou menos responsável por uma das combinações de atributos de ocorrência e simultâneamente para um dos nomes de animais também. Se nóes testarmos agora qual célula dá a resposta mais forte se apenas o nome do animal é apresentado como dado de entrada (i.e. x =[x_s,0]t (*elevado

Figura 3.28. **Tabela com os dados de treinamento utilizados por Kohonen e Ritter, mostrando a parte x_a de atributos do vetor atenuada em 80%. Observe que x_a é a matriz transposta da mostrada na fig. 3.26**

1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2	0,0	0,0	0,2	0,0	0,0	0,0	0,0	0,2	0,0	0,0	0,2	0,0
0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2	0,0	0,0	0,2	0,0	0,0	0,0	0,0	0,2	0,0	0,0	0,0	0,0
0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2	0,0	0,0	0,2	0,0	0,0	0,0	0,0	0,2	0,0	0,0	0,0	0,2
0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2	0,0	0,0	0,2	0,0	0,0	0,0	0,0	0,2	0,0	0,0	0,2	0,2
0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2	0,0	0,0	0,2	0,0	0,0	0,0	0,0	0,2	0,2	0,0	0,2	0,0
0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2	0,0	0,0	0,2	0,0	0,0	0,0	0,0	0,2	0,2	0,0	0,2	0,0
0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2	0,0	0,2	0,0	0,0	0,0	0,0	0,2	0,2	0,0	0,2	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2	0,0	0,0	0,2	0,2	0,0	0,0	0,0	0,2	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2	0,0	0,0	0,2	0,2	0,0	0,0	0,0	0,0	0,2	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2	0,0	0,0	0,2	0,2	0,0	0,2	0,0	0,2	0,2	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,2	0,0	0,0	0,0	0,2	0,2	0,0	0,0	0,0	0,2	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2	0,0	0,2	0,2	0,0	0,0	0,0	0,2	0,2	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,2	0,0	0,2	0,2	0,0	0,2	0,0	0,2	0,2	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,2	0,0	0,2	0,2	0,2	0,2	0,0	0,0	0,2	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,2	0,0	0,2	0,2	0,2	0,2	0,0	0,0	0,2	0,0	0,0
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,2	0,0	0,2	0,2	0,2	0,0	0,0	0,0	0,0	0,0	0,0

a t*), nós obtemos o mapa mostrado na fig. 3.27 (os pontos indicam neurônios com respostas fracas)

Isto é altamente aparente que a ordem espacial das respostas foi capturada a essencial "família de relacionamentos" entre os animais. Células respondendo para, e.g. "birds" ocupam a parte esquerda da rede, "hunters" como também "tiger", "lion" e "cat" recolhem para a direita, mais "peacefull" espécies como "zebra", "horse",e "cow" agregam ao meio superior. Dentro de cada conjunto, um novo agrupamento de acordo com a similaridade é discernido. A fig. 3 mostra o resultado de um "traçado simulado da penetração do eletrodo" para a mesma rede. Ela difere da fig. 2 em que agora cada célula tem sido marcada pelo símbolo que é seu melhor estímulo, i. e., extrai a melhor resposta para aquela célula. Isto faz o parcelamento do "território neural" em domínios específicos para estes itens visíveis de entrada. Hierarquia deste modo é representada por domínios aninhados. A classe geral (e.g. "bird") ocupa um largo território, no qual ele mesmo é diferenciado em subdomínios aninhados, correspondendo a mais itens especializados ("owl","duck", "hen", etc.). Embora fortemente idealizado, este resultado é muito sugestivo de como o sistema de auto-organização para guiar espacialmente a formação de traços de memória em tal maneira que seu layout físico final forma uma imagem direta da hierarquia do mais importante "conceito de realcionamentos".

Figura 3.29. **"mapeamento da atividade cerebral" para a rede na fig.3.27. Cada célula é marcada pelo nome do animal gerando a melhor resposta. Células respondendo ao mesmo nome de animal formam domínios, os quais são agrupados de acordo com a similaridade entre os animais.**

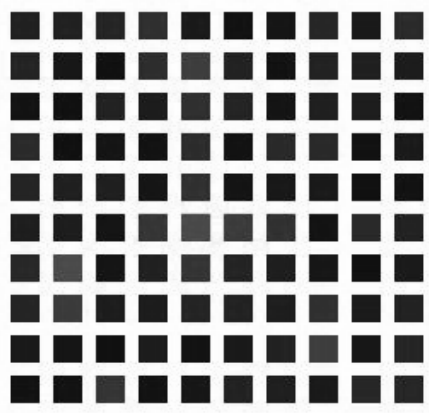
pato	pato	cavalo	cavalo	zebra	zebra	vaca	vaca	vaca	vaca
pato	pato	cavalo	zebra	zebra	zebra	vaca	vaca	tigre	tigre
ganso	ganso	ganso	zebra	zebra	zebra	lobo	lobo	tigre	tigre
ganso	ganso	falcão	falcão	falcão	lobo	lobo	lobo	tigre	tigre
ganso	coruja	falcão	falcão	falcão	lobo	lobo	lobo	leão	leão
pomba	coruja	coruja	falcão	falcão	cachorro	cachorro	cachorro	leão	leão
pomba	pomba	coruja	coruja	coruja	cachorro	cachorro	cachorro	cachorro	leão
pomba	pomba	águia	águia	águia	cachorro	cachorro	cachorro	cachorro	gato
galinha	galinha	águia	águia	águia	raposa	raposa	raposa	gato	gato
galinha	galinha	águia	águia	águia	raposa	raposa	raposa	gato	gato

O resultado da realização de uma simulação no SNNS utilizando estes dados pode ser vista na figura abaixo. Na Figura 3.30.é mostrada a ativação da rede após apresentação apenas da parte simbólica do padrão de número 13 (leão). Aqui foi utilizada uma rede de Kohonen de 10x10 com uma camada de entrada de 29 neurônios, como descrito no experimento. Observe a atividade de neurônios agrupada em um cluster em torno de um neurônio com ativação mais forte.

Regras Baseadas em Mapas Semânticos

No exemplo do mapa animal, a regra do contexto foi ainda muito simples: A decodificação simbólica foi relacionada a um conjunto de atributos estaticamente explícitos. Em linguagem natural, e obviamente em qualquer percepção natural também, os itens e seus atributos, e

Figura 3.30. **Resultado de treinamento de uma rede com SNNS: ativação após apresentação da parte simbólica do padrão #13 (leão). O vermelho representa ativação mais alta.**



obviamente algum estado informativo usualmente ocorrem em uma sequência temporal. O conceito do contexto então precisa ser ampliado e a dimensão do tempo dosado também. Talvez o caminho mais simples para fazer isto, é definir o contexto de cada item com todos os outros itens (juntos com suas ordens seriais) que ocorrem em um certo "período de tempo" ao redor do item selecionado.

Neste trabalho nós não perderemos tempo com uma representação física concreta de sinais, de qualquer forma os padrões são temporais, como se fala, ou espaciais, como em texto. Para as conversões séries - paralelo, redes neurais podem usar caminhos com diferentes tempos, estados próprios que dependem das sequências, ou de algum outro mecanismo implementado na memória short-term. Aqui nós mostramos concentrando-se apenas ans similaridades entre as exdpressões que levanta-se de ocorrências condicionais de suas partes,e simplesmente imagina que trios ou pares de palavras podem de algum modo ser apresentadas às portas de entrada do sistema.

Linguagens contém muitos outros níveis de significado. Isto é possível para construir casos, onde the devida "janela" para o enetendimento da palavra tem que compreender uma sequencia inteira. Em outra mão, a possiblidade de formar gramáticas demosntra que uma significante parte da estrutura da linguagem ainda manifesta-se em um imenso baixo nível,

abaixo para padrões de palavras e conclusões. Detecção de tal estrutura "escala curta" poderá ser o foco de nosso interesse nesta seção e nós demostraremos que a inclusão de um muito limitado contexto de palavras permite o modelo básico da rede (1) a formar mapas semânticos, no qual as palavras itens são agrupadas de acordo com a semântica categorias (objetos, atividades, qualificações, etc.) e simples similaridade.

Figura 3.31. **a Lista as palavras usadas (nomes, verbos e advérbios), b padrões de sentenças e c alguns exemplos de sentenças de três palavras geradas**

Bob/Jim/Mary	1
cavalo/cachorro/gato	2
cerveja/água	3
carne/pão	4
corre/caminha	5
trabalha/fala	6
visita/telefona	7
compra/vende	8
gosta/odeia	9
bebe/come	10
muito/pouco	11
rápido/lentamente	12
frequentemente/raramente	13
bem/mal	14

(a)

Padrões de sentenças

1	5	12	1	9	2	2	5	14
1	5	13	1	9	3	2	9	1
1	5	14	1	9	4	2	9	2
1	6	12	1	10	3	2	9	3
1	6	13	1	11	4	2	9	4
1	6	14	1	10	12	2	10	3
1	6	15	1	10	13	2	10	12
1	7	14	1	10	14	2	10	13
1	8	12	1	11	12	2	10	14
1	8	2	1	11	13	1	11	4
1	8	3	1	11	14	1	11	12
1	8	4	2	5	12	2	11	13
1	9	1	2	5	13	2	11	14

(b)

Mary gosta de comer
Jim fala bem
Mary gosta de Jim
Jim come frequentemente
Mary compra carne
cachorro bebe rapido
cavalo odeia carne
Jim come raramente
Bob compra carne
gatos caminham suavemente
Jim come pão
Gato odeia Jim
Bob vende cerveja
(etc)

©

Para a nossa demonstração, nós usamos um conjunto de 3 sequências de palavras randomicamente geradas construídas do vocabulário da Fig. 4 a. O vocabulário contém nomes, verbos e advérbios, e cada classe contém várias subdivisões, como nome de pessoas, animais e objetos inanimados em uma categoria de nomes. Essas distinções são em parte de uma gramática, em parte da semântica natural . De qualquer forma, por razões discutidas na seção 4.1, eles mostraram não ser discerníveis de um código de palavras próprias mas apenas de um contexto de onde as palavras são usadas. Em linguagem natural, como um contexto poderia conter uma rica variedade de experiências sensoriais. Nesta demonstração muito limitada, entretanto, nós poderemos apenas pegar

no cliente o contexto fornecido pelo ambiente textual imediatamente adjacente de cada palavra corrente. Isso irá retornar que mesmo este contexto extremamente restrito será suficiente para fazer saber alguma estrutura semântica interessante. É claro que isto requer que cada sentença não seja totalmente randomica, mas obedeça algumas últimas regras rudimentares de gramática e semântica com exatidão. Isto é assegurado por restringir a seleção randomica a um conjunto de 39 padrões de sentenças "legais" apenas. Cada padrão é um trio de números da figura 4b. Uma sentença é construída pela escolha de uma tripla e substituindo cada número por uma das palavras com o mesmo número na fig. 4.a. Este resultado é um total de 498 diferentes sentenças de palavras triplas, alguns dos quais são dados na fig 4c. (Se aquelas indicações são verdadeiras ou não não nos interessa: nós estamos apenas interessados exatidão semântica).

Nesta demonstração muito simples, supôs-se que o contexto de uma palavra seria suficientemente definida pelo par formado pelos seus predecessores e sucessores imediatos. (Para ter tais pares também para a primeira e última palavra da sentença, nós decidimos que as sentenças serão concatenadas em uma ordem randômica da sua produção.) Para o vocabulário de 30 palavras na fig 4a nós poderíamos ter procedido como na seção 4.1 e representado cada para por um vetor de 60-dim com dois não-zeros de entrada. Para uma codificação mais otimizada, de qualquer forma, como explicado mais detalhadamente no apêndice I, nós assumimos para cada palavra, um vetor randômico 7-dim de tamanho único, escolhido fora do conjunto para cada palavra independentemente para uma distribuição probabilística isotropica. Daqui cada par predecessor/sucessor foi representado por um codigo vetorial de 14-dim.

Isso aconteceu em todos os nossos experimentos computacionais que preferencialmente demos atenção para cada cláusula separadamente, uma estratégia de aprendizagem muito mais eficiente foi considerar cada palavra neste contexto médio sob um conjunto de cláusulas possíveis, antes apresentando isso ao algoritmo de aprendizado. O (significado) contexto de uma palavra foi deste modo definido primeiramente como média sobre 10.000 sentenças de todos os códigos vetoriais de pares predecessor/sucessor cercando essa palavra. O trigésimo resultado da 14-dim "contexto médio de palavras", normalizada a um único comprimento, assumiu uma regra similar como campo de atributos xa na simulação

prévia. Cada "campo de atributo" foi combinado com um 7-dim "campo simbólico", xs consistindo em um código vetorial para a sua palavra, mas adequada ao comprimento a. Neste momento, o uso do vetor de código randômico quase garantiu que o campo simbólico xs não saiba nenuma informação sobre relacionamentos de similaridade entre as palavras. Como antes, o parâmetro a determinou a influência relativa da parte simbólica em comparação a parte contextual e teria o conjunto de a = 0.2.

Figura 3.32. **"Mapa Semântico" obtido na rede de 10 x 15 células**

.	água	.	carne	.	.	.	.	cachorro	cavalo
cerveja	.	.	.	.	pão	.	.	.	.
.	.	.	.	.	.	.	.	.	gato
.	.	.	Pequeno	.	.	.	.	.	.
rápido	.	.	.	.	.	rara-mente	.	.	Bob
.	.	.	.	muito	.	.	.	Jim	.
lenta-mente	.	freqüen-temente	.	.	.	.	.	.	.
.	.	.	.	.	.	come	.	.	Mary
bem	.	.	.	trabalha	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
mal	.	.	Fala	.	.	.	.	.	telefona
.	.	.	.	.	compra	.	.	visita	.
.	.	.	.	.	.	vende	.	.	.
.	.	corre	.	.	.	.	.	.	.
bebe	.	.	.	caminha	.	.	odeia	.	gosta

Para esse experimento uma grade planar de 10 x 15 neurônios formais seriam usados. Como antes, cada neurônio inicialmente faria apenas conecções randômicas fracas ao n = 21 linhas de entrada do sistema, então novamente, nenhuma ordem inicial seria apresentada.

Depois de 2000 apresentações de entrada as respostas dos neurônios das partes simbólicas somente seriam testadas. Na fig. 5, o quadro simbólico foi escrito para mostrar o local onde o sinal do símbolo x = [xs,0]t (*elevado a T*) deu a resposta do máximo. Nós claramente vemos que os contextos tem "canalizado" os itens das palavras às posições de memória das quais refletem as relações gramáticas e semânticas. Palavras de mesmo tipo, i. e. nomes, verbos e advérbios tem segregado em separado, grandes domínios.

O "mapa semântico" obtido em uma rede de 10 x15 células depois de 2000 representações de pares de palavras-contexto derivados de 10.000 sentenças randômicas do tipo mostrado na fig. 4c. Nomes, verbos e advérbios são segregados dentro de diferentes domínios. Dentro de cada domínio um agrupamento adicional concorda com aspectos do significado como discernimento.

Cada um desses domínios é mais adiante subdividido por similaridade no nível de semântica. Por instância, nomes de pessoas e animais tendem a ser aglomerados em subdomínios em comum "domínio do substântivo", refletindo em co-ocorrências diferentes com, e.g. verbos como "correr" e "telefonar". Advérbios com significado oposto tendem a ser particularmente fechados juntos, como o oposto deles significa assegurar a eles o uso máximo do espaço comum. O agrupamento de verbos indicam diferenças nos caminhos, eles podem co-ocorrer com advérbios, pessoas, animais e objetos não animados como e.g. "comida".

Figura 6 mostra o resultado de um outro experimento, baseado no mesmo vocabulário e mesmo padrão de sentença como antes. De qualquer forma, nesta simulação o contexto de uma palavra foi restrita apenas ao seu predecessor. (O contexto agora consiste de um vetor de 7-dim). Mesmo isto sendo muito limitado, provou como sendo suficiente para produzir um mapa com aproximadamente similar as propriedades como na fig 5. Isto mostra que as regularidades apresentadas são um tanto robustas para trocas nos detalhes da codificação tão grande quanto o contexto capturar uma quantidade suficiente da estrutura lógica subjacente.

Figura 3.33. **Este mapa foi obtido pelo mesmo procedimento da fig 05, mas com um cotexto mais restrito que inclui apenas o predecessor imediato de cada palavra.**

vende

Pode-se discutir que a estrutura resultante no mapa tinha sido artificialmente criada por uma escolha pre-planejada da sequência de padrões reservadas na entrada. De qualquer forma, isso é facilmente verificado nos padrões da fig. 4b quase que completamente até a exaustão das possibilidade de combinação das palavras da fig 4a em uma semanticidade bem formada de sentenças de 3 palavras (um leitor astuto pode verificar alguns "casos de linha semânticas" não cobertas, como "dog eats cat"). Isto pode tornar isso claro que todos padrões de sentenças selecionados estavam realmente determinados pelas restrições inerentes na semanticidade correta usada pelas palavras, e não vice-versa. Além

disso, uma porcentagem significativa das palavras vizinhas estendem-se através das bordas das sentenças randomicamente concatenadas. Nesta concatenação foi irrestrita, tais vizinhos foram largamente irrelacionados a estrutura semântica e gramatical das sentenças, e constituíram um tipo de "ruído" no decorrer do processo. Isso é importante observar que este ruído não disfarça as regularidades se não forem apresentadas nas cláusulas.

De qualquer forma, o que importante observar está exatamente aqui. Alguma semântica realística de mapas cerebrais, precisariam de um modelo hierárquico probabilístico muito mais complicado. A finalidade de um simples modelo artificial usado neste trabalho foi apenas demostrar o potencial de um processo auto organizacional par formar mapas abstratos. Em particular, os resultados da simulação, como está, não poderia ser usado como referência para comparação topográfica direta com áreas do cérebro. Como uma comparação entre a fig. 5 e fig.6 mostram, existem muitos caminhos quase equivalentes, nos quais um conjuntos de relacionamentos de similaridades podem ser apresentados no mapa. Consequentemente os mapas gerados pelo modelo não são únicos, a menos que restrições adicionais, como e.g. condições limiares ou alguma ordem inicial grosseria for imposta. Estes podem então inicialmente "polarizar" o sistema que então converge a um outro único mapa.

3.5.2.6. Discussão

Um dos mecanismos biológicos que atualmente tem sido mal compreendido, é a habilidade do cérebro de formar abstrações de experiências sensoriais primárias em gigantescos níveis de generalizações.

Isto já é bem conhecido que em um baixo nível percentual, informações sensoriais primeiramente vêm organizadas dentro de mapas sensoriais ordenados topograficamente, e isto tem também já sido demostrado teoricamente que cada mapa pode ser formado adaptativamente, refletindo uma métrica mútua dos relacionamentos e estatísticas dos dados de entrada. Este mesmo princípio tem sido aplicado com sucesso considerável para exigir tecnicas padrões de farefas de recognição como discurso recognitivo.

Neste trabalho nós temos agora mostrado que o princípio de mapas de auto organização podem também ser extendidos para níveis mais altos de processamento, onde os relacionametos entre itens são mais súbitos e menos aparentes em suas características intrísecas, uma propriedade que é característica de expressões simbólicas. Simbolos em geral, não contém componentes metricamente relatáveis. Consequentemente, mapas de significância topográfica de símbolos não devem mostrar por muito tempo as características intrísecas, mas ao invés disso as similaridades lógicas das suas entradas. Isto voltará, entretanto que mapeamento organizados de dados simbólicos podem seguir as mesmas leis básicas de adaptação, fornecendo que os dados simbólicos de entrada são apresentados juntos com uma quantidade suficiente de contexto, que então definem os relacionamentos de similaridade entre eles. Se as descrições simbólicas deixam traços de memória nos mesmos neurônios dos quais os sinais contextuais convergem, os mesmos neurônios então tornam-se também sensíveis aos sinais simbólicos em uma ordem espacial que também reflete sua lógica de similaridade.

Símbolos tocam um regra particularmente importante em linguagens. Neste trabalho nós demos dois exemplos de simulação que demonstram a formação auto-organizada de mapas semânticos, no qual relacionamentos semânticos entre palavras tem sido codificadas em posições relativas espaciais de localização de respostas. Nossos mapas artificiais são parcelados na hierarquiedade aninhado a domínios refletindo diferentes categorias de palavras. Este parcelamento emerge totalmente do co-ocorrente contexto sensorial e palavras. Em nossas simulações o contexto sensorial foi restrito a um simples conjunto de atributos ou palavras adjacentes em sentenças. O tipo simples de clausulas utilizadas neste experimento ocorrem em todas as linguagens, suas primitivas uniformes. Isto é consequentemente também do interesse para notar qual dado experimental (sect.2) indica organizações similares as área do cérebro relacionadas a linguagem de processamento. Especialmente a debilidade da linguagem categoria-específica discutidas na sect.2. (Warrington and McCarthy 1987) visto para refletir o mais similar em um nível filosófico.

Na primeira simulação nós usamos inicialmente atributos explicitos, deste modo assumindo que algum mecanismo neural já tinha gerado-os. A filosofia subjacente do nosso trabalho é que uma tendência auto-

organizadora similar poderia existir em todos os níveis de processamento; ilustrando isto, de qualquer forma, é apenas possível se os sinais tem algum significado para nós.

O termo "mapa semântico" usado neste trabalho, não é ainda referido a "compreensão mais elevada da palavra"; palavras estão apenas sendo agrupadas conforme o seu contexto local. Devido a grande correlação entre contexto local e significado da palavra, entretanto isto aproxima a ordenação semântica encontrada na linguagem natural, o qual presumidamente não pode ainda ser generalizada em cada fase aprendida. Isto é uma questão intrigante se algum estágio de processamento subsequente poderá criar um ordenamento que reflete significados de um nível mais elevado - dos quais poderá facilitar totalmente o entendimento da significado das palavras - por algum tipo de interação do basico processo de auto-organização.

Nosso modelo enfatiza a regra do arranjo espacial de neurônios, um aspecto apenas considerado em muitas poucas abordagens modeladas. Entretanto nós não gostamos de dar a impressão que nós nos opomos a visão de redes neurais como sistemas distribuídos. As interconecções massivas responsáveis pela interação lateral tão bem como os engramas relacionando para a memória associativa são certamente disseminado sobre uma grande área da rede.

Em outra mão, isto mostra-se inevitável que alguma tarefa de processamento complexo precisa algum tipo de segregação de informação em partes separadas, e localização do mais robusto e eficiente caminho para encontrar esta meta. Os mapas semânticos oferecem um mecanismo eficiente para gerar uma segregação significativa de informação simbólica uniforme em um nível razoavelmente alto de semânticas, e eles tem qualidade mais recente de ser o único baseado em aprendizado não-supervisionado. Se nós ainda necessitarmos considerar um timing relativo de sinais. (cf. von der Malsburg and Bienenstock 1986) remanescem o mais recente objetivo do estudo.

Existem outras novas razões não para negligenciar os arranjos espaciais das unidades de processamento. Por instância, a anatomia dos conjuntos de circuitos neurais restringem a realização da conectividade entre unidades. Mais a fundo, sinais cerebrais nãos e apoiam unicamente em transmissão de sinais axonais emitidos em distâncias selecionáveis, mas

também emprega difusão de neurotransmissores e neuromoduladores, em todas semelhanças, estas restrições poderiam limitar a implementação de muitos mecanismos computacionais, a menos que este obstáculo esteja aliviada pela eficiente organização espacial oferecida pelos mapas.

De um ponto de vista hardware, se isto fosse esperado que a minimização dos custos de conectividade poderia fortalecer este tipo de design de rede neural. Isto poderia dar um indício porque uma organização topográfica é tão difundida no cérebro. Outros argumentos para localização são que a segregação espacial de representações fazem então mais lógica, pela redução de etapas para a sua inferência mútua, e logicamente de itens simbolicos similares, sendo espacialmente adjacentes, podem invocar um outro associativamente, como expressado nas leis clássicas de associação.

Uma outra observação pode ser necessária. Nossas simulações não poderão ser pegas como uma sugestão que cada palavra é representada por uma então chamada "célula mãe" no cérebro. Cada palavra é um pedaço complexo de informação provavelmente redundante codificada por uma população neuronal inteira (e várias vezes em separado "lexica", cf. 2.4). Tudo em um grande modelo idealizado usado em nossas simulações, isto não é um simples neurônio mas um subconjunto inteiro de células, cercar o mais responsável deles, que pega o mais adequado a palavra (cf. fig 3). Estes subconjuntos podem então ser engajados em novos processamentos, não capturado pelo modelo básico. O número de células atribuídas a cada subconjunto também depende da frequência das ocorrências das palavras. Isto é análoga ao caso que a frequência de ocorrência de estímulos determina o fator local ampliado em um mapa sensorial (Kohonen op. Cit., Ritter and Schulten 1986). Similarmente palavras frequentes poderiam recrutar células de um grande território neural e ser mais redundantemente representado. Como consequência, as mais frequentes palavras poderão ser menos suscetíveis aos danos locais. Esta complies com observações empíricas nos pacientes do curso, por meio do que as palavras familiares tem mais chances de sobreviver que as raras.

Finalmente, nós gostaríamos de apresentar um noção filosófica intrigante. Como indicado anteriormente, existem vária evidências bioló-

gicas e justificações teóricas para o funcionamento do cérebro, requisitando representação de seus dados de entrada por significativas partes processadas em localizações separadas espacialmente. A idéia sobre categorias fundamentais postuladas para a interpretação e entendimento do mundo mais obviamente levanta da formação prioritária de cada representação no próprio mundo biológico do cérebro.

3.5.2.7. Variação da Função de Vizinhança durante o Treinamento

Kohonen e Ritter sugerem, em um anexo de seu artigo, que, para tornar o processo de aprendizado cada vez mais local, à medida em que o processo de aprendizado avança, o tamanho da variância definida para o "sino de Gauss"seja reduzido gradualmente. Isto tem como efeito que o treinamenento que o neurônio vencedor sofre afeta uma vizinhança cada vez menor. A idéia por detrás deste procedimento é simular o efetito de que primeiramente uma rede aprende conceitos de forma grosseira e global, achando um único local para armazenar estes conceitos. À medida que o tempo passa, porém, o aprendizado se torna mais local e nunces e ajustes finos para variedades de um conceito são realizadas numa pequena vizinhança. A técnica de começar-se com uma vizinhança grande, que é reduzida modificando-se a variância da curva de Gauss tem esse efeito: primeiramente a rede aprende de maneira geral a organizar padrões similares em grupos localizados em regiões específicas da rede, depois a rede passa a refinar este mapeamento de maneira cada vez mais localizada.

Na prática, a fórmula sugerida por Ritter e Kohonen é a mostrada na Eq. 4 abaixo:

$$\sigma(t) = \sigma_i \left(\frac{\sigma_f}{\sigma_i} \right)^{\frac{t}{t_{max}}} \qquad (4)$$

onde t_{max} representa o número total de épocas (1 época = apresentação de todos os padrões para a rede).

3.6. O que aprende uma Rede de Kohonen ?

Vimos até agora que:

- uma Rede de Kohonen é inspirada na forma como se supõe que redes neurais naturais aprendem e
- o modelo originou-se a partir das pesquisas anteriores de Teuvo Kohonen em Análise de Componentes Principais e Quantização de Vetores.

Para fundamentar uma aplicação na prática de Redes de Kohonen como um mecanismo para o aprendizado auto-organizante de padrões e seu posterior uso para classificação de padrões, é importante analisarmos a capacidade representacional e a forma de representação da informação em um Mapa Auto-Organizante.

Na prática, uma rede de Kohonen toma um conjunto de dados em um espaço de dados **V** qualquer e os representa de forma discretizada através de um neurônio (e eventualmente sua vizinhança) no espaço de um Mapa Auto-Organizante **A**. Esta transformação de um espaço de representação para outro é denominada **mapeamento** ϕ, podendo ser representada por:

$$\phi: V \rightarrow A, (x \in V) \rightarrow (\phi(x) \in A)$$

A condição para que este mapeamento seja uma boa representação do espaço vetorial é que:**,** .

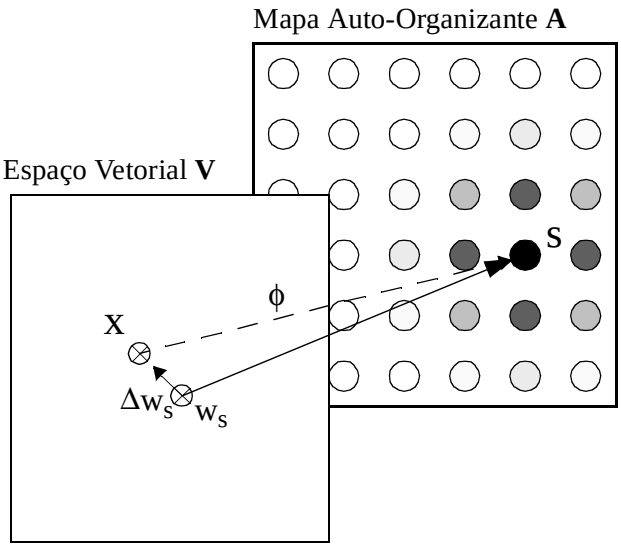
$$\left\| W_{\phi(x)} - X \right\| = min \left\| W_r - X \right\|, r \in A$$

onde W é um vetor de pesos da rede A

Este mapeamento está ilustrado na Figura 3.34..

O espaço vetorial V é um espaço qualquer com a dimensionalidade do número de variáveis de um padrão X desse espaço. O vetor de pesos w_s do neurônio vencedor S pertencente a A representa uma aproximação da função de mapeamento ϕ que associa pontos do espaço vetorial V a neurônios em A. Δw_s é o erro dessa aproximação representado no espaço vetorial V.

Figura 3.34. **Mapeamento V -> A.**



Com isso, vimos como ocorre o mapeamento de entre o espaço vetorial e o espaço do Mapa Auto-Organizante da rede de Kohonen.

Supondo agora, que os dados em V possuem uma distribuição ***d*** qualquer, como é gerada a função de mapeamento ϕ de forma a refletir esta distribuição ?

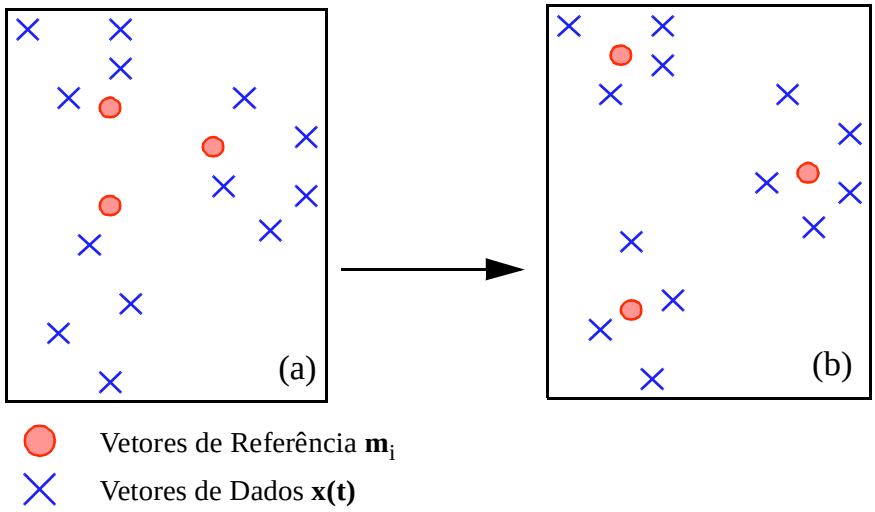
3.6.1. Qualidades Matemáticas do Modelo de Kohonen

Existem várias interpretações matemáticas da forma como uma rede de Kohonen aprende e de como devemos interpretar o mapeamento ϕ gerado após o aprendizado da rede. Helge Ritter em sua tese de doutorado (Univ. de Munique, 1988) analisou em detalhe ambos. Nós vamos reproduzir aqui, omitindo os detalhes matemáticos, a sua interpretação da representação.

O conceito básico de representação em uma rede de Kohonen baseia-se na idéia de Componentes Principais. A Análise de Componentes Principais é uma técnica de análise de distribuição de dados onde se procura encontrar vetores de referência que representem de uma forma mais ou menos

adequada conjuntos de vetores de uma distribuição de dados. Possui utilidade em mineração de dados e para decifrar códigos baseados em índices. A Figura 3.35. dá um exemplo de três vetores de referência $\mathbf{m_i}$ encontrados para aproximar uma distribuição de dados dividida em grupos.

Figura 3.35. **Representação de agrupamentos de dados expressando uma função x(t) em um espaço n-dimensional qualquer através de vetores de referência $\mathbf{m_i}$**



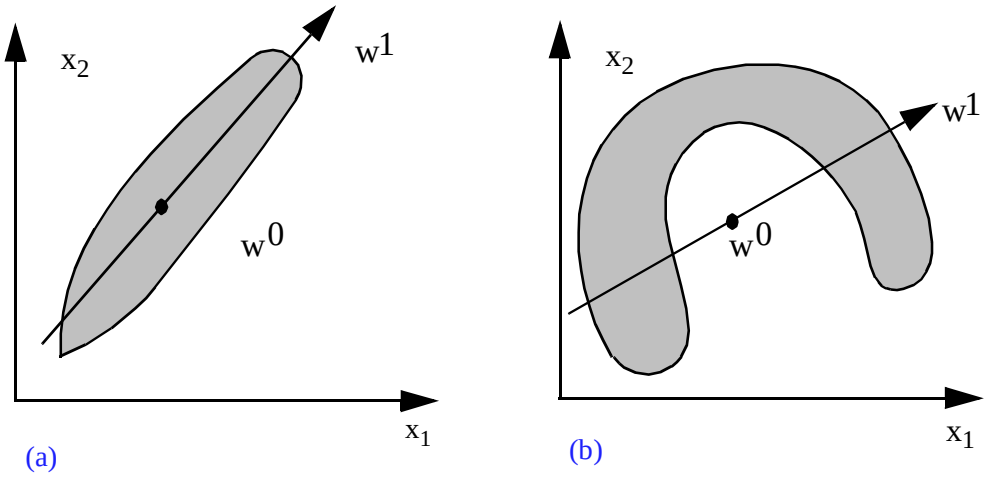
O que uma rede de Kohonen representa após o aprendizado pode ser considerado como uma generalização dessa idéia.

Se nós observarmos uma distribuição de dados representando, por exemplo, todos os pares de valores de duas variáveis x_1 e x_2 que pertençam à categoria c_j, poderemos ter um *scatter plot* como mostrado em (a) ou em (b) na Figura 3.36., dependendo de como os dados se distribuem

Podemos representar a componente principal desta distribuição de dados através de um único ponto w^0 no espaço vetorial, que representará exatamente o "centro de massa" da distribuição[1], ou através de um

1. Se a distribuição é conhecida, podemos calcular w0 usando exatamente o método de cálculo do centro de massa da Física, atribuindo uma massa qualquer, não nula, a cada um dos pontos do conjunto.

Figura 3.36. **Duas distribuições de dados e suas componentes principais**



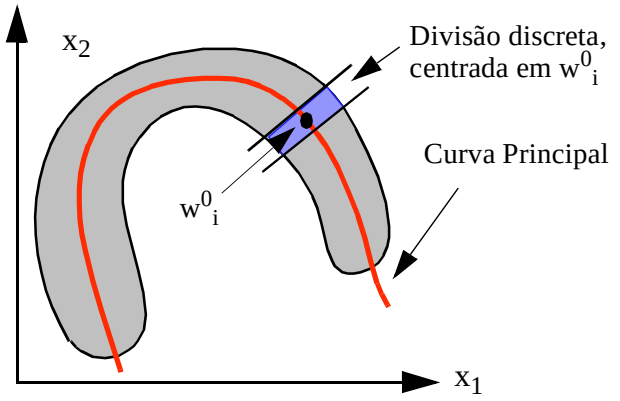
vetor w^1 que representa o "eixo principal" da distribuição[1], indicando a sua tendência. Isto pode ser realizado através de várias técnicas estatíticas, entre outras pela Análise Fatorial, utilizada quando a nossa distribuição de dados representa várias classes.

O problema de uma representação deste tipo ocorre quando temos uma distribuição de dados como em (b). Numa situação como essa, o centro da distribuição é um ponto em V que não pertence à distribuição e o eixo principal da distribuição é uma descrição muito pobre e falha do real comportamento desta. É o caso de distribuições de dados com tendências não-lineares, que nós já abordamos no capítulo 1, quando falamos de Nearest Neighbour.

Para representarmos adequadamente uma distribuição de dados como a representada em (b) necessitamos de uma representação não-linear da distribuição, dada por uma **curva principal** da distribuição, como é mostrado na Figura 3.37.

1. Das mesma forma, se a distribuição é conhecida, podemos utilizar o método de cálculo do eixo de massa principal da Física para obter w1.

Figura 3.37. **Curva Principal**



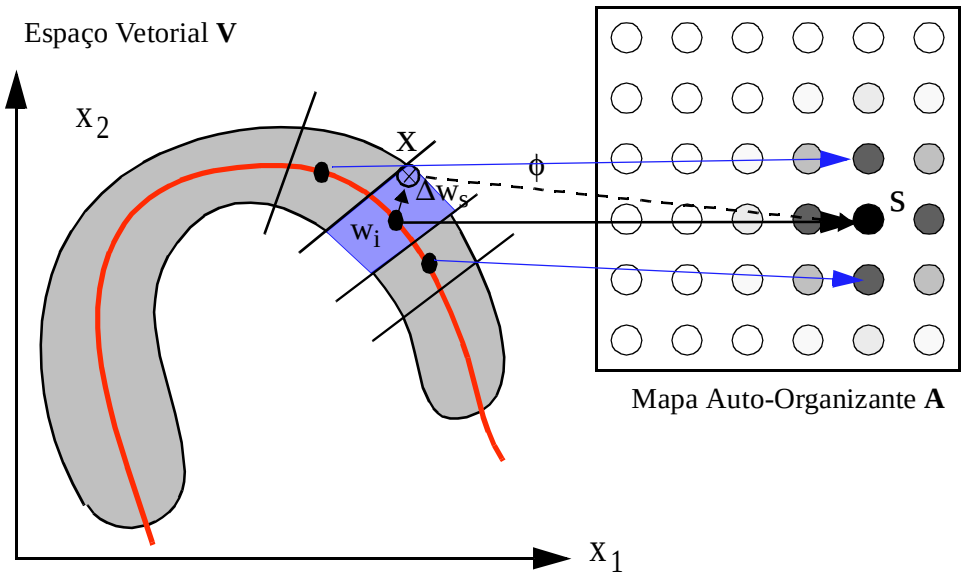
O cálculo exato de uma curva principal, porém, pode ser um processo matemático extremamente custoso, envolvendo interpolação polinomial ou outra técnica.

Quando discutimos *Nearest Neighbour*, no capítulo 1, e algoritmos que o utilizam, como IBL, no capítulo 2, vimos que existe a possibilidade de se aproximar um mapeamento de uma distribuição deste tipo através da divisão desta área curva em pedaços discretos, representados através de um conjunto de protótipos w^0_i. Isto está muito bem exemplificado pela facilidade com que IBL representa o problem ada espiral exatamente implementando esta técnica. Para gerarmos um conjunto de protótipos w^0_i deste tipo, porém, é necessário que a distribuição seja **conhecida**. Isto é fácil, quando temos, de antemão, associada a cada padrão, a sua categoria. Mas como proceder quando não conhecemos a distribuição dos dados nem quais classes existem ?

É aqui que a utilização de Redes de Kohonen se torna interessante: Helge Ritter demonstrou que uma rede de Kohonen aprende exatamente uma representação não linear discretizada deste tipo, sem necessidade de que se forneça de antemão as classes a que pertence cada padrão, realizando uma espécie de Análise Fatorial Não-Linear Discretizada. O resultado do processo de aprendizado, quando a convergência ocorreu adequadamente, é um mapeamento de subconjuntos da distribuição de dados a neurônios específicos da Rede A, que passam a fungir como

protótipos para esses subconjuntos. Regiões vizinhas da distribuição são mapeadas para neurônios vizinhos no mapa de Kohonen A. O mecanismo de escolha do vencedor, similar a idéia do Nearest Neighbour, é o que garantre a não-linearidade da capacidade de representação da rede depois de treinada, agindo como uma função limiar, intrinsecamente não-linear, que determina as fronteiras entre cada subárea (subvolume) da distribuição mapeada. Isto pode ser visto na figura abaixo, onde uma classe é representada por um agrupamento (cluster) de neruônios em torno do vencedor S. O vencedor S representa com a maior aproximação o padrão X apresentado à rede..

Figura 3.38. **Representação discretizada de uma distribuição não-linear de padrões aprendida por uma rede de Kohonen segundo Ritter.**



3.7. Explorando Dados Agrupados em Redes

Apropriedade estrutural das redes de Kohonen que vimos até agora coloca a pergunta: Não podemos de alguma forma aproveitar o fato de as redes de Kohonen estruturarem e organizarem topologicamente a informação aprendida ?

A resposta a essa pergunta é **sim**. Ao contrário das redes-BP, que necessariamente têm de ser encaradas como classificador de caixa-preta, as informações (e as abstrações) "aprendidas" por uma rede de Kohonen podem ser exploradas após o treinamento da rede e utilizadas das mais variadas formas.

Para finalizar este capítulo, vamos ver duas aplicações de redes de Kohonen.

A primeira delas se refere à utilização da informação contida em uma rede de Kohonen para guiar a busca de dados ainda desconhecidos que tenham a mais forte relação com um **contexto** atual de informação incompleta. Na verdade, trata-se da utilização de uma rede de Kohonen treinada como uma máquina de inferência neural que guia um processo de busca no sentido de se seguir pelo menor caminho na árvore de busca. Este trabalho foi realizado por nós no início da década de 1990 e apresenta uma solução para o problema de se explorar o espaço de possíveis soluções de maneira eficiente utilizando-se redes neurais.

A segunda é uma aplicação de treinamento de um braço-robô desenvolvida por Helge Ritter em 1989, onde se utiliza uma terceira camada de saída para uma rede de Kohonen, que controla um braço-robô de forma a que se mova para um ponto visualizado fornecido como dado de entrada. Esta não é propriamente uma aplicação de reconhecimento de padrões pura, mas mostra como extender uma rede de Kohonen através de uma camada de saída, aspecto pouco comentado na literatura.

3.8. Utilizando Mapas Auto-Organizantes como Máquinas de Inferência: KoDiag

No início da década de 1990 foi desenvolvido, pelo grupo de Sistemas Especialistas de Kaiserslautern, um sistema híbrido, denominado KoDiag [RW94, RW93, Wan93, RWW92, WR92], para diagnóstico baseado em casos utilizando uma Rede Neural de Kohonen modificada para dinamicamente atribuir pesos a atributos em função do contexto do problema e, assim, também realizar diagnóstico por meio do levantamento dirigido de atributos para o novo caso.

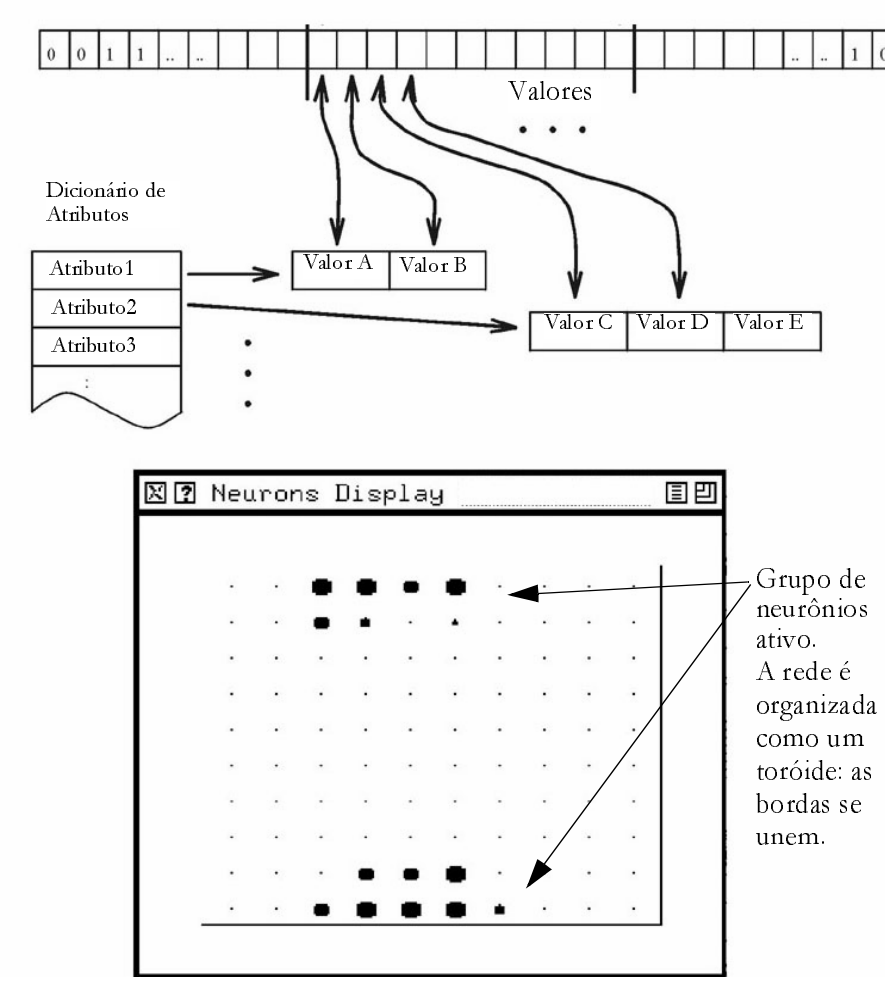
Para a implementação de KoDiag foi utilizado o princípio do mapeamento topológico da rede neuronal de Kohonen para o armazenamento de casos: casos mais similares são armazenados em áreas topologicamente próximas na rede. KoDiag utilizou pela primeira vez uma representação não simbólica para a base de casos, que era aprendida pela rede e ficava armazenada na mesma de forma sintética.

KoDiag utiliza a base de casos de PATDEX [Wes91, Wes93] para o treinamento da rede neural. Os casos são codificados como padrões especiais, e são representados de forma explícita: variáveis, valores de variáveis e diagnóstico. Dessa forma, o padrão de treinamento possui 3 partes: a primeira representa todas as variáveis do sistema; a segunda, todos os possíveis valores dessas variáveis; e a terceira, todos os diagnósticos. Dessa forma, um caso a ser treinado é representado com dois valores para cada variável: um para indicar a existência da mesma e outro para indicar seu valor.

KoDiag podia realizar aquisição incremental dirigida de dados para diagnóstico com a apresentação de dados incompletos da situação atual. Caso a informação não bastasse para o diagnóstico, o agrupamento de neurônios ativado por esta apresentação era analisado e o valor da variável ainda desconhecida mais fortemente correlacionada a este agrupamento era solicitado para ser levantado pelo usuário. Um exemplo de um padrão em PATDEX pode ser visto abaixo:

KoDiag é considerado um sistema de Raciocínio Baseado em Casos, em contraste com outros sistemas que também utilizam a codificação de casos em redes neurais, por utilizar uma interpretação autônoma passo a

Figura 3.39. **KoDiag: no alto é mostrado o padrão de treinamento. No diagnóstico, somente a primeira parte era utilizada. Abaixo, o agrupamento ativo após apresentação do caso atual [RW94]**



passo dos dados aprendidos na rede e guiar o usuário de forma inteligente no processo de levantamento de dados.

Figura 3.40. **Código Smalltalk descrevendo um caso de defeito em uma máquina CNC utilizado por KoDiag**

```
PortableCase
    newCase: #Toolarm10
    withEnvironment: #(#(#IoStateIN32 #logical0) #(#Code #I41)
        #(#ToolarmPosition #back) #(#IoStateOUT30 #logical0)
        #(#IoStateOUT28 #logical1) #(#Valve21Y2 #switched)
        #(#IoStateIN37 #logical1) )
    describes: #IoCardFaultAtIN32i59
```

3.8.1. Qual é o objetivo de KoDiag ?

O ponto de partida para o desenvolvimento de KoDiag (Diagnóstico com Redes de Kohonen) foi a necessidade de se possibilitar o levantamento incremental de variáveis de um problema em uma situação de diagnóstico descrita como um padrão composto por pares atributo-valor: em muitas situações onde é necessário efetuar-se um diagnóstico de um problema não se possui de antemão valores para todas as variáveis do problema e tampouco todas as variáveis são necessárias para se determinar o diagnóstico correto para todas as situações.

Um exemplo é o domínimo de aplicação-exemplo de KoDiag: Diagnóstico de falhas em tornos de comando numérico:

- O estado de um torno CNC pode ser descrito por uma quantidade bastante grande de variáveis, como por exemplo a temperatura do óleo em diversas partes hidráulicas, o estado de diversos fusíveis, a mensagem de erro sinalizada no display de comando, o estado de desgaste da ferramenta de corte, etc.
- Quando um torno deixa de funcionar, dependendo do erro, apenas algumas dessas variáveeis terão relevância para se obter um diagnóstico correto da falha.
- Levantar o valor para todas é um processo desnecessário e custoso, uma vez que implica em tempo e, eventualmente, na necessidade de se desmonatr partes do torno ou de se realizar testes complexos para levantar o estado de uma peça ou parte mecânica.

- Partindo-se de um conjunto de variáveis iniciais, como por exemplo o código de erro mostrado pela máquina, é importante guiar o processo de levantamento de valores para as outras variáveis cujo valor ainda é desconhecido de maneira a otimizar o processo de busca de um diagnóstico. Para isso é preciso encontrar um caminho mínimo no espaço de pares variável-valor de maneira e levantar apenas os valores da variáveis relevantes ao contexto de falha atual e evitar levantar valores para variáveis desnecessárias, como por exemplo o estado do fusível da fonte de alimentação em uma situação onde o óleo de um componente hidráulico superaqueceu.

Esse procedimento é fácil de realizar em um sistema especialista para diagnóstico convencional baseado em regras ou em um sistema baseado em casos que utiliza uma matriz de relevância para conjuntos de pares variável-valor. Em muitas istuações, porém, a relevância de variáveis para contexto é desconhecida e precisamos de uma técnica capaz de agrupar as informações de forma a representar esses contextos. A rede de Kohonen é ideal para isso pois possibilita a exploração da informação codificada na rede através da:

- apresentação incremental de padrões incompletos, que são considerados como o contexto atual ,
- da exploração da informação da rede, tomando-se os neurônios ativados por este contexto e seguindo-se os pesos de volta para a camada de entrada e vendo-se com qual valor ainda desconhecido este contexto correlaciona masi fortemente.

3.8.2. Como funciona KoDiag ?

KoDiag explora o fato de que uma rede de Kohonen mapeia valores de variáveis a contextos representados por grupos de neurônios na rede ativados por um padrão incompleto de forma a possibilitar o levantamento dirigido de novas informações.

A codificação de informação em KoDiag funciona da seguinte forma:

- Cada padrão é constituído de três partes: diagnóstico, variáveis, valores de variáveis.
- Para cada diagnóstico há um neurônio de entrada

- Para cada variável representando uma parte da máquina responsável por uma falha, há um neurônio de netrada.
- Para cada valor que essa variável pode assumir há também um neurônio de entrada. Para variáveis de domínios contínuos, discretiza-se o domínio em faixas de valores.

A rede é treinada com padrões que contém valor “1” para dois neurônios de entrada de cada variável que participa deste padrão: um valor para indicar a participação desta variável neste caso e outro para indicar qual valor que esta variável assumiu neste caso particular. Variáveis cujo valor é irrelevante para a situação são representadas por “0”. Dessa forma estamos associando a variável, indeopendentemente de seu valor à situação.

O processo de diagnóstico, depois de treinada a rede, funciona então da seguinte maneira:

1. É apresentado à rede um padrão incompleto inicial, contendo apenas os pares variável-valor capazes de serem levantados de forma fácil, por exemplo: a mensagem de erro emitida pela máquina e fo fato de o óleo em um mancal estar superaquecido.
2. Este contexto inicial é propagado pela rede e observa-se quais neurônios respondem a ele com uma ativação mínima. Este é o conjunto dos Active-Neurons.
3. Propagamos a ativação destes neurônios de volta para a parte dos diagnósticos da camada de entrada usando os pesos das conexões existentes. Se há um neurônio de diagnóstico que obteve ativação acima de um limiar mínimo e ao mesmo tempo diferente o suficiente dos outros, consideramos que o contexto foi suficiente para se chegar a um diagnóstico inequívoco e paramos.
4. Se isto não acontece, propagamos a ativação dos ActiveNeurons de volta para a parte das variáveis da camada de entrada usando os pesos das conexões existentes. Buscamos o neurônio representando uma variável cujo valor ainda é desconhecido que tenha obtido a maior ativação. Este é um neurônio que durante o treinamento foi associado ao contexto atual de forma forte.

5. Levantamos o valor da variável representada este neurônio, detalhando o nosso contexto. Este par atributo-valor é então apresentado isoladamente à rede.
6. Determinamos o conjunto de nerurônios do mapa de Kohonen ativados por este par, que é denominado ChosenNeurons. Estes são os neurônios do mapa associados a este contexto atributo-valor independentemente dos outros valores que o acompanham.
7. Realizamos a intersecção entre o conjunto anterior, ActiveNeurons e os ChosenNeurons, levando ao novo conjunto de ActiveNeurons, cujo tamanho é bastante menor. Este procedimento de redução progressiva do conjunto de neurônios do mapa considerados garante a convergência do processo.
8. Novamente propagamos a ativação dos ActiveNeurons para a parte de diagnóstico da camada de entrada, retornando ao passo 3. O processo termina quando o teste do passo 3 for satisfeito ou os ActiveNeurons representarem um conjunto vazio.

Figura 3.41. **O processo de KoDiag pode ser visto abaixo**

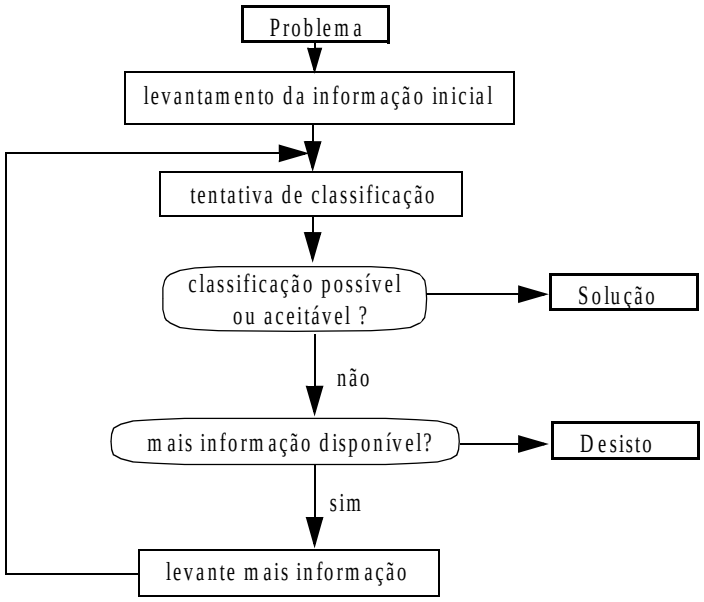
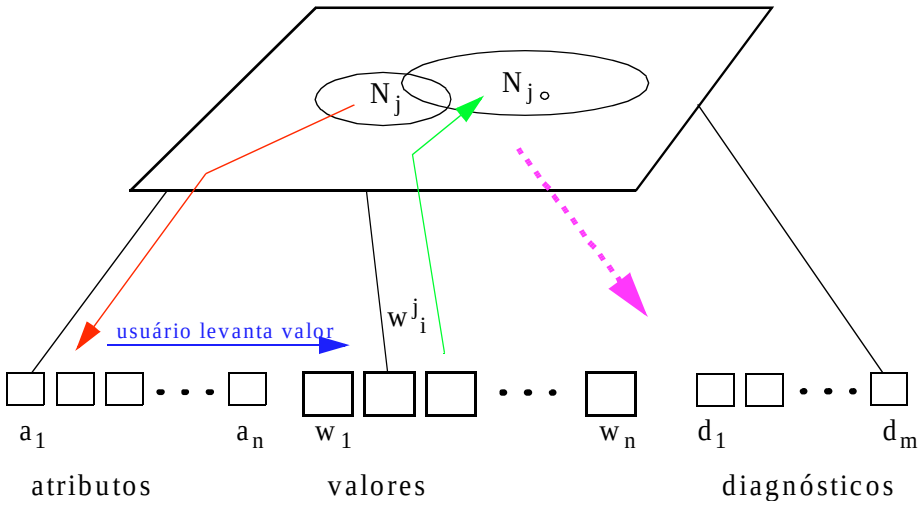


Figura 3.42. **Seleção de Neurônios**



KoDiag foi testado com o mesmo conjunto de daods do sistema PATDEX, um sistem ade RBC estado-da-arte na época e os resultados obtidos foram bastante similares, tanto em termos de levantamento de variávesi quanto de resultados de classificação em função de perda de informação.

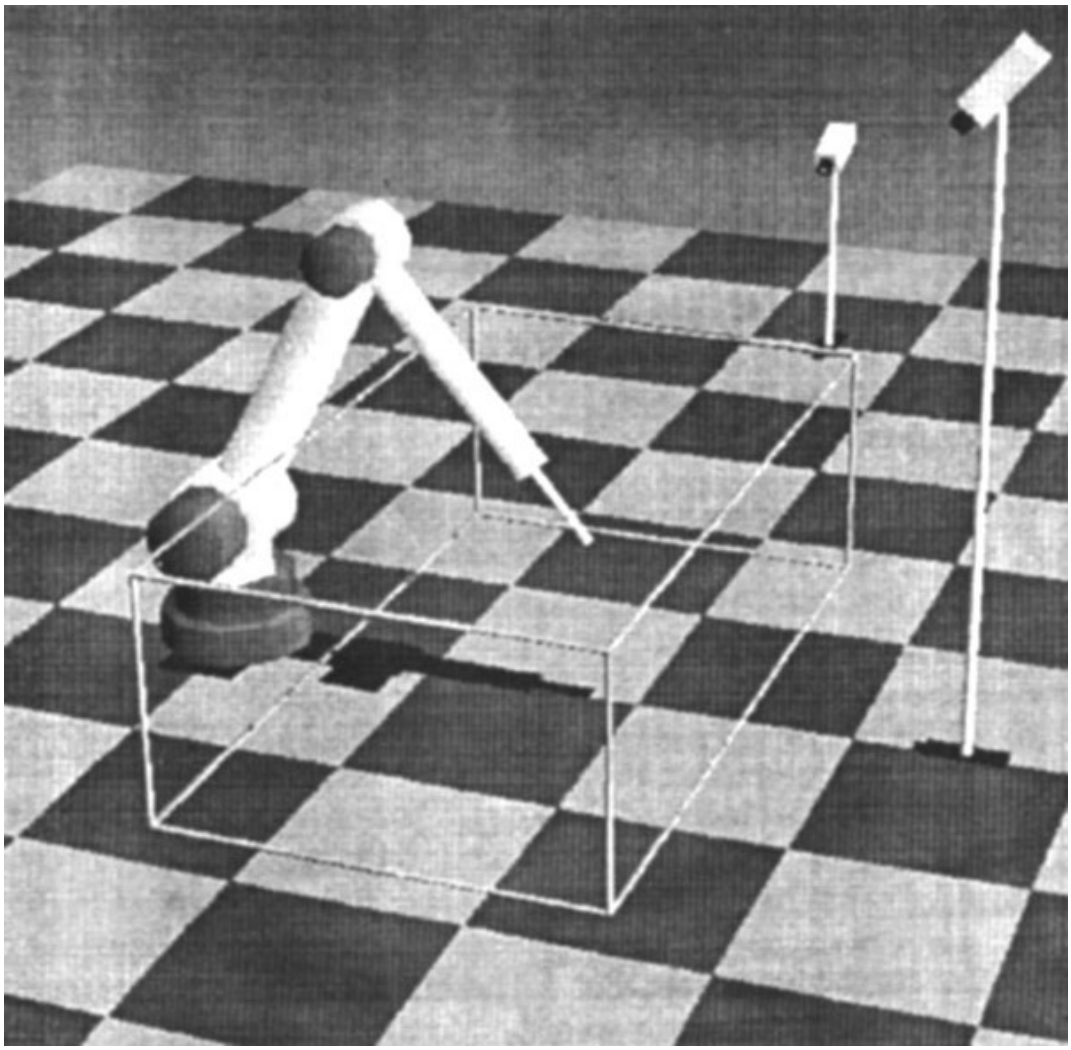
3.9. Utilizando Redes de Kohonen para a coordenação visumotora de um braço de Robô

Motivação: simulação das „cartas motoras“ encontradas em cérebros de mamíferos utilizando uma rede de Kohonen tridmensional com uma camada extra de saída representando os comandos de movimento do braço-robô.

Técnica: Modelar uma projeção $\Phi : V \rightarrow U$ de um espaço „sensórial“ V em um espaço de „comandos motores“ U.
a) criar uma projeção ϕ dos „estímulos sensoriais“ *V* em uma carta tridimensional *A*

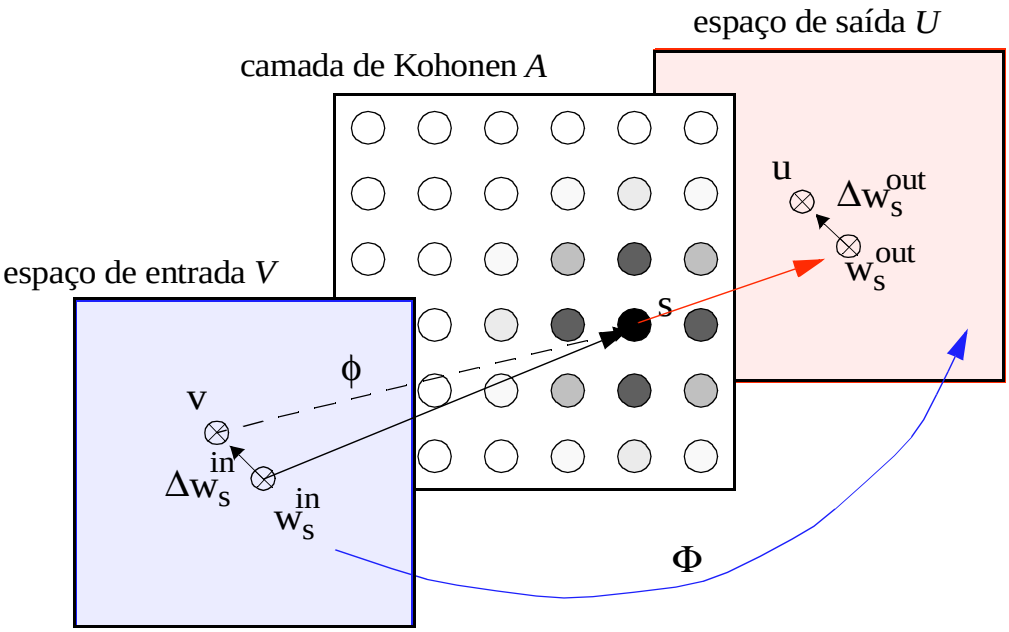
b) criar uma projeção da carta em um espaço de „comandos motores“ *U*.

Figura 3.43. **Desenho da simulação utilizada por Helge Ritter**



O modelo geral da projeção gerada pode ser visto na próxima figura. O objetivo é utilizar-se dois mapeamentos: um entre a camada de entrada e a de Kohonene outro entre camada de Kohonen e a de saída.

Figura 3.44. **Mapeamento esperado**



Algoritmo geral de treinamento:

1. Registre a próxima ação de controle (v, u)
2. Vencedor: calcule a posição na carta s := $\phi_w(v)$ que corresponde ao mapeamento do estímulo visual v na carta.
3. Execute um passo de treinamento:

$$\Delta w_r^{(in)} = w_r^{(in)} + \alpha h_{rs}(v - w_r^{(in)})$$

4. Execute um passo de treinamento:

$$\Delta w_r^{(out)} = w_r^{(out)} + \alpha^{out} h_{rs}^{out}(u - w_r^{(out)})$$

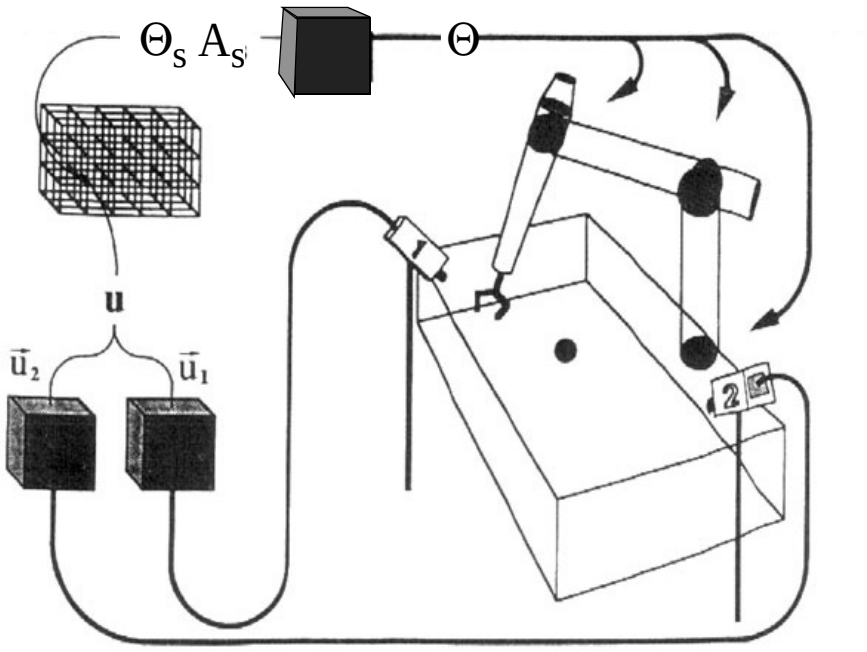
5. retorne a 1

Estrutura da simulação para os movimentos do braço do robô:

- Posição do objeto dada por um vetor 4-dimensional $(\vec{u}_1, \vec{u}_2)$

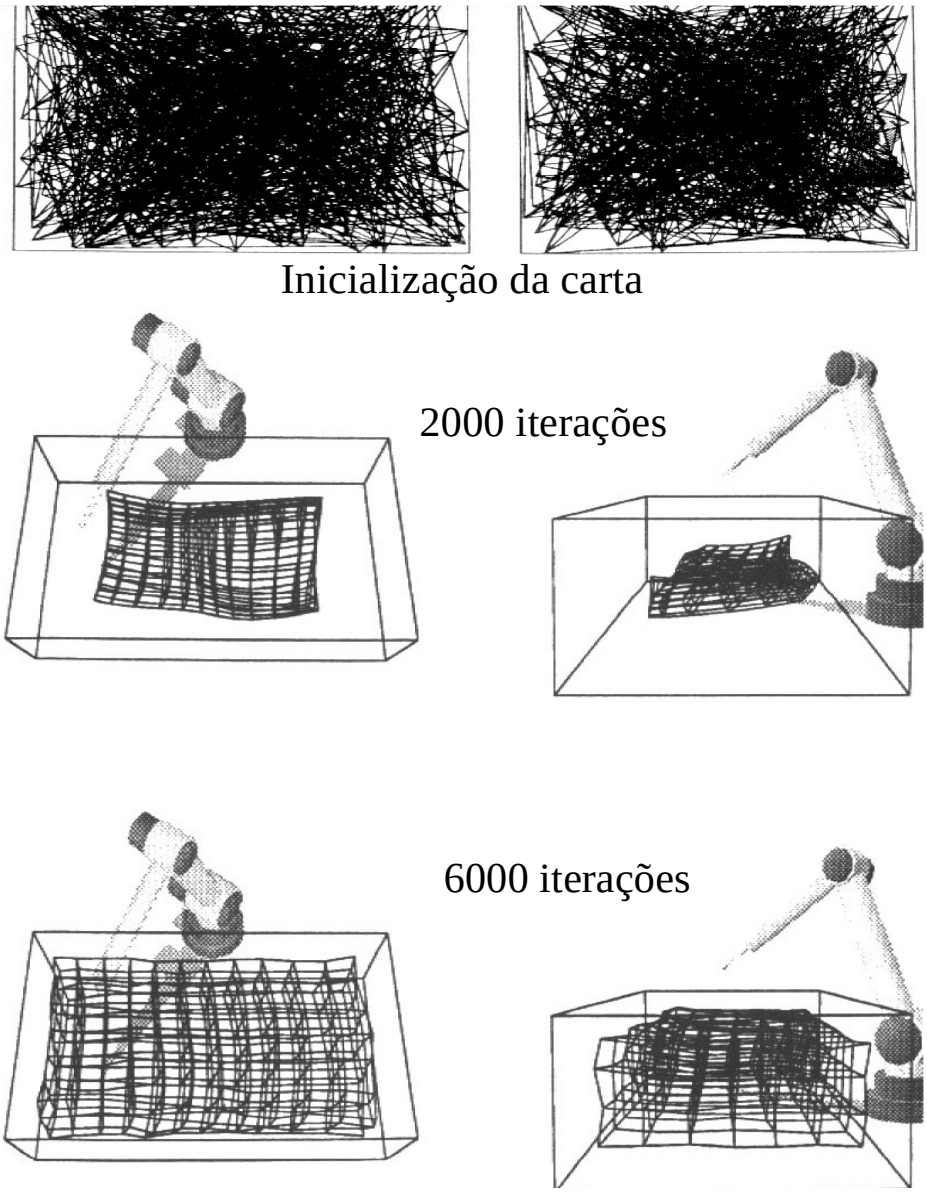
Posição do braço do robô dada por um conjunto de 3 ângulos das juntas do braço.

Figura 3.45. **Estrutura da simulação para os movimentos do braço do robô**



Resultados do treinamento podem ser vistos na próxima figura: com o passar do tempo a rede vai aprendendo a configuração tridimensional do espaço de movimentação do braço-robô e cada ponto do espaço na rede tridimensional é mapeado a um conjunto de ângulos das juntas do braço robô na camada de saída que movimenta o braço para aquela posição.

Figura 3.46. **Treinamento**



3.10. Referências

[Koho88] Kohonen,Teuvo: Self-Organization and Associative Memory, Springer, 1988 (2. edition)

[KR89] Kohonen,T., Ritter, H.; Self-Organizing Semantic Maps, Biol. Cybern., 61, 241-254, (1989)

[RW93] J. Rahmel, A. von Wangenheim. The KoDiag System: Case-Based Diagnosis with Kohonen Networks. In Proceedings of the I International Workshop on Neural Networks Applications and Tools, Liverpool, IEEE Computer Society Press, 1993.

[RW94] J. Rahmel, A. von Wangenheim. KoDiag: A Connectionist Expert System. In Proceedings of the IEEE International Symposium on Integrating Knowledge and Neural Heuristics, Pensacola, Florida, 1994.

[RWW92] J. Rahmel, A. von Wangenheim, S. Wess. KODIAG: Fallbasierte Diagnose mit KOHONEN Netzen. In Proceedings of the GI Workshop Fälle in hybriden Systemen, Germany, 1992.

[Wan93] A. von Wangenheim. Fallbasierte Klassifikation mit Kohonen Netzen. In Proceedings of the Workshop "Fälle in der Diagnostik", XPS-93, Germany, Februar 1993.

[Wes91] S. Wess. PATDEX/2: Ein fallbasiertes System zur technischen Diagnostik. SEKI-Working Paper SWP91/01, Department of Computer Science, University of Kaiserslautern, Germany, 1991.

[Wes93] S. Wess. PATDEX - Inkrementelle und wissensbasierte Verbesserung von Aehnlichkeitsurteilen in der fallbasierten Diagnostik. In Proceedings of the 2. German Workshop on Expertsystems, Germany, 1993.

[WR92] A. von Wangenheim, J. Rahmel. Fallklassifikation und Fehlerdiagnose mit Kohonen-Netzen. In Proceedings of the Workshop "Ähnlichkeit von Fällen", Universität Kaiserslautern, Germany, 1992.