

## 1º Congresso Brasileiro de Redes Neurais

Escola Federal de Engenharia de Itajubá  
Itajubá, 24 a 27 de outubro de 1994

### UMA ARQUITETURA PARCIALMENTE RECURSIVA APLICADA A PROBLEMAS DE CLASSIFICAÇÃO

Marcello Baptista de Martino\*

#### Resumo:

As Redes Neurais apresentam-se como uma promissora técnica para aplicações dos Sistemas Elétricos de Potência. Grande parte destas aplicações podem utilizar alguma forma de Classificação em sua resolução. Assim, discutimos e levantamos os requisitos necessários à aplicação de Redes Neurais em Classificação. Apresentamos uma nova arquitetura parcialmente recursiva baseada nos modelos de *Jordan* e *Elman*, selecionamos o algoritmo "Back-Propagation Through Time" para ser utilizado na arquitetura proposta e, finalmente, comprovamos sua eficácia em um exemplo de classificação fornecido por *Telfer*.

#### I - Introdução

A Inteligência Artificial tem começado a desempenhar um importante papel nos Sistemas Elétricos de Potência, tendo sido utilizada com sucesso em diversas aplicações. Embora a maior parte das aplicações existentes utilize Sistemas Especialistas, certas classes de problemas não têm tido soluções totalmente satisfatórias por causa da inexistência de um modelo matemático adequado (e.g. Controle), da ocorrência de complexas classificações (e.g. Detecção de Falhas) ou do elevado tempo de processamento (e.g. Análise de Contingência). As Redes Neurais devido à sua capacidade de aprendizado através de exemplos, de processamento em tempo real e de generalização, apresentam-se como uma solução promissora a ser explorada.

Os levantamentos realizados sobre as aplicações existentes de técnicas de Inteligência Artificial concluíram que o sucesso destas depende da qualidade do conhecimento adquirido e representado. De um modo geral, as aplicações existentes podem ser divididas em 3 tipos:

- as que buscam otimizar uma ou mais variáveis do sistema (otimização),
- as que relacionam eventos ou estados do sistema à conclusões ou ações sobre este (classificação),
- as que realizam ambas as tarefas.

A Classificação é uma poderosa estratégia de organização do conhecimento que pode ser utilizada em diversas aplicações (e.g. monitoração, predição, interpretação e controle). Assim, o estudo de Redes Neurais em Classificação é fundamental para a resolução de grande parte desses problemas.

\* CEPEL - Centro de Pesquisas de Energia Elétrica  
Av. Hum. sem número, Cidade Universitária, Rio de Janeiro, RJ, BRASIL, CEP 21.491-590  
E-Mail: martino@fund.cepel.br

## II - Redes Neurais em Classificação

Uma Rede Neural pode funcionar de dois modos distintos: como um associador de padrões ou como um detector de regularidades. O paradigma Classificador pode ser considerado como um caso particular da associação de padrões, cujo objetivo é aprender a classificar os padrões dos exemplos em categorias distintas, de modo a estar apto a classificar outros padrões quaisquer. Um dos mecanismos mais utilizados neste paradigma é o aprendizado competitivo, no qual os elementos processadores competem entre si para determinar o vencedor. Este paradigma resolve, na verdade, um caso particular da classificação, denominado categorização.

Uma das primeiras formas de representação do conhecimento relativo a problemas de Classificação denomina-se Redes Semânticas, e pode ser modelada por um grafo cujos vértices representam os conceitos envolvidos e cujas arestas representam o relacionamento entre estes conceitos.

Uma Rede Semântica normalmente apresenta arcos de generalização e especialização, organizando os conceitos em uma taxonomia [Woods,91]. A taxonomia pode ser interpretada como uma estrutura para agrupar indivíduos de acordo com características comuns, e a classificação como o processo pelo qual novos conceitos são localizados ou adicionados a uma taxonomia existente.

Tradicionalmente, para resolver problemas de classificação, é criada uma hierarquia de módulos pré-treinados de Redes Neurais para categorização, compondo uma Rede Neural Hierárquica. Tal solução envolve a criação de uma arquitetura diferente para cada problema.

Para que se possa efetivamente utilizar uma única arquitetura de Redes Neurais na resolução de problemas gerais de classificação, deve-se ter o cuidado de analisar as características particulares inerentes a estes problemas.

## III - Levantamento dos Requisitos

Primeiramente, é preciso considerar que o especialista, por ser a pessoa mais capacitada na resolução do problema em questão, é quem determina quais são os conceitos significativos e fornece os exemplos adequados. Assim, a rede neural a ser utilizada deve possuir um mecanismo de *aprendizado supervisionado*.

No paradigma classificador, os padrões de saída são diferentes dos padrões de entrada (*Equação 1*), indicando a utilização de redes *hetero-associativas* em sua resolução.

$$E \Rightarrow S$$

*Equação 1*

onde o símbolo " $\Rightarrow$ " representa uma dependência conceitual dos conceitos de saída **S** em relação aos conceitos de entrada **E**.

Em problemas de categorização, os conceitos de entrada são totalmente independentes dos conceitos de saída. Porém, em problemas de classificação, um especialista muitas vezes se utiliza de *conceitos duais* (conceitos que às vezes servem de entrada e outras, de saída) nos exemplos (*Equação 2*) para sua resolução.

$$(E+D) \Rightarrow (D+S)$$

*Equação 2*

onde o símbolo **D** representa os conceitos duais.

Neste tipo de exemplos, os conceitos duais podem ser utilizados tanto como conceitos de entrada quanto como conceitos de saída. Um conceito dual pode, inclusive, ser utilizado para obter outro conceito dual, indicando que a rede deve ser *recursiva*.

Embora os conceitos duais devam ser utilizados como entrada durante o treinamento, na utilização da rede apenas os conceitos de entrada estarão disponíveis, implicando em *diferentes conjuntos de treinamento e testes*. A rede

deve ser capaz de obter padrões de saída **S** e padrões duais **D**, a partir de padrões de entrada **E** (Equação 3).

$$E \Rightarrow (D+S) \quad \text{Equação 3}$$

Finalmente, até mesmo os exemplos a serem utilizados durante o treinamento não são "completos", ou seja, o especialista não fornece exemplos em que todas as entradas resultem em todas as saídas. Pelo contrário, ele procura selecionar exemplos gerais e, a partir destes, se especializa em exemplos mais detalhados. Esta característica requer que exista algum mecanismo de *aprendizado a partir de exemplos de dados incompletos*. Esta não é uma limitação relativa apenas à problemas de classificação, mas comum a todas as aplicações em Redes Neurais.

#### IV - Arquitetura Proposta

A maioria dos modelos conhecidos possuem aprendizado supervisionado e são hetero-associativas, porém não apresentam uma topologia recursiva. Um exemplo típico é a rede "Back-Propagation", atualmente o modelo mais utilizado em aplicações práticas no mundo.

Os modelos recursivos são pouco utilizados e sua aplicação prática é ainda mais restrita, pois apresentam sérios problemas de convergência e estabilização.

Uma abordagem recente utiliza redes com a maioria das conexões em uma única direção ("feed-forward") e algumas, cuidadosamente selecionadas, na outra direção ("feed-back"), sendo denominadas *parcialmente recursivas*.

A rede de *Jordan* [Jordan,89] é uma arquitetura parcialmente recursiva (Figura 1), na qual uma camada extra, denominada *camada de contexto*, copia o padrão de ativação da camada de saída no instante anterior.

A camada de contexto é realimentada por si mesma, de modo a acumular os traços dos valores passados.

As conexões recursivas permitem que a camada escondida utilize suas próprias respostas anteriores para orientar seu comportamento subsequente, criando uma memória na rede. Com uma entrada fixa, sua rede pode ser treinada para gerar uma seqüência de saídas, além de poder ser treinada para reconhecer e distinguir diferentes seqüências de entradas. Uma rede deste tipo é capaz de aprender a emular um Autômato de Estados Finitos.

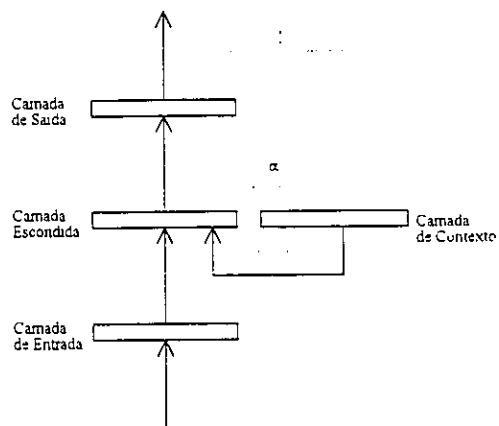


Figura 1 - Rede de Jordan

A rede de *Elman* possui uma arquitetura similar [Elman,90], porém mais simples, na qual a camada de contexto copia os valores da camada escondida ao invés da camada de saída (Figura 2).

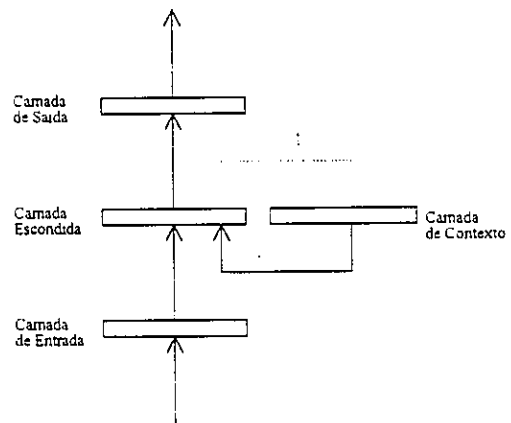


Figura 2 - Rede de Elman

Nesta arquitetura, a camada de contexto armazena o estado anterior da camada escondida, provendo a rede de memória. A camada escondida deve mapear tanto as entradas quanto o estado

anterior nas saídas desejadas. Os padrões internos são salvos como contexto e este, por sua vez, é utilizado na determinação do próximo padrão interno. Deste modo, os padrões internos são sensíveis ao contexto.

Baseados nos modelos de *Jordan* e *Elman*, desenvolvemos uma arquitetura parcialmente recursiva de redes neurais (*Figura 3*) que atende aos requisitos referentes à arquitetura levantados.

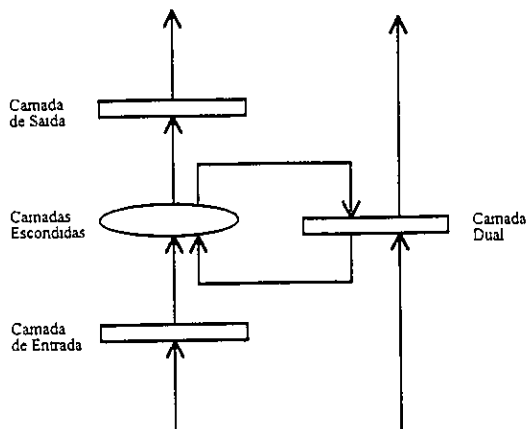


Figura 3 - Arquitetura Proposta

A arquitetura sugerida, além de prover a rede de memória, permite treiná-la com o contexto desejado. Este contexto é aproveitado durante a determinação do próximo contexto, e assim sucessivamente.

## V - Seleção do Algoritmo

As redes desenvolvidas por *Jordan* e *Elman* possuem somente conexões recursivas de peso fixo e permitem a utilização do algoritmo de aprendizado "*Back-Propagation*" (*BP*). Distintamente destas, a arquitetura proposta envolve conexões recursivas adaptáveis, não permitindo a utilização deste algoritmo.

O algoritmo "*Back-Propagation*" pode ser estendido para Redes Neurais arbitrárias [Pineda,89], denominando-se "*Recurrent Back-Propagation*", cuja dificuldade é garantir a convergência para estados estáveis, assim como em qualquer rede recursiva. O "*Back-Propagation*" pode ser considerado um caso particular do "*Recurrent Back-Propagation*" (*RBP*).

No mesmo trabalho em que criticou os Perceptrons, *Minsky* mostrou que para qualquer rede neural recursiva existe uma rede "feed-forward" correspondente, com o mesmo funcionamento (em um intervalo finito de tempo). A rede sem realimentação equivalente [Rumelhart,86] pode ser treinada através de uma versão discretizada do "*Back-Propagation*", conhecido como "*Back-Propagation Through Time*".

O algoritmo anterior pode ser adaptado para aprender enquanto os padrões são apresentados, possibilitando sua utilização com seqüências de tamanho indeterminado e em tempo real [Zipser,89], sendo denominado "*Real Time Recurrent Learning*" (*RTRL*). Porém, requer a manutenção, para cada intervalo de tempo, de  $N^4$  derivadas.

Existe, ainda, um outro algoritmo denominado "*Time-Dependent Recurrent Back-Propagation*", capaz de treinar uma rede recursiva genérica em intervalos contínuos de tempo [Pearlmutter,89], que pode ser visto tanto como uma extensão do *RTRL* para intervalos contínuos de tempo quanto como uma extensão do *RBP* para seqüências dinâmicas.

Como o problema em questão não exige intervalos contínuos de tempo nem aprendizado em tempo real, selecionamos o algoritmo "*Back-Propagation Through Time*" para ser utilizado na arquitetura sugerida.

Infelizmente, o algoritmo sugerido minimiza o erro entre os padrões obtidos e os desejados, tornando necessário que os dados dos exemplos de treinamento sejam completos.

## VI - Back-Propagation Through Time

A rede original é discretizada no tempo através da replicação de cada unidade de processamento em intervalos de tempo. As conexões ligam uma unidade em um instante  $t$  a outra em um instante  $t+1$ . A atualização de cada unidade no instante  $t+1$  é dada por uma função do estado das unidades no instante  $t$  (e.g. Equação 4).

$$U_i(t+1) = g \left( \sum w_{ij} U_j(t) + \varepsilon_i(t) \right)$$

Equação 4

De um modo geral, o procedimento deste algoritmo consiste em apresentar o padrão de entrada para o sistema, permitir que ele rode por um determinado número de iterações, comparar o resultado com a saída desejada, retropropagar os erros obtidos pelo mesmo número de iterações e atualizar os pesos.

Assim, uma de suas maiores dificuldades está em determinar um número de iterações que seja suficientemente grande para garantir a resposta desejada e suficientemente pequeno para que não torne o procedimento inviável, como em seqüências compridas ou de tamanho indeterminado. No caso dos problemas de Classificação este número pode ser determinado através da altura da árvore de hierarquia ou da profundidade do grafo correspondente.

Devido à replicação das unidades, o principal problema desse algoritmo é a necessidade de elevados recursos computacionais. Apesar disso, esta restrição se aplica somente à fase de treinamento, pois, uma vez treinada a rede discretizada no tempo, a rede recursiva original com os pesos obtidos pode ser utilizada.

Assim como toda rede recursiva, a arquitetura proposta pode ser igualmente discretizada no tempo (Figura 4).

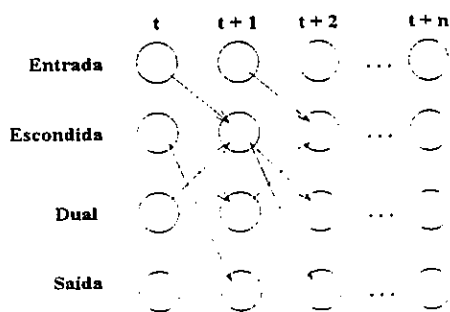


Figura 4 - Discretização da Arquitetura

É possível demonstrar que, independentemente do problema, a arquitetura pode ser discretizada em apenas dois intervalos de tempo.

### VII - Testes

Como teste utilizamos uma ligeira variação do exemplo de um artigo [Telfer,91] publicado no periódico *Neural Networks*, em que especialistas levantaram uma classificação (Figura 5) para determinação de diferentes aeronaves.

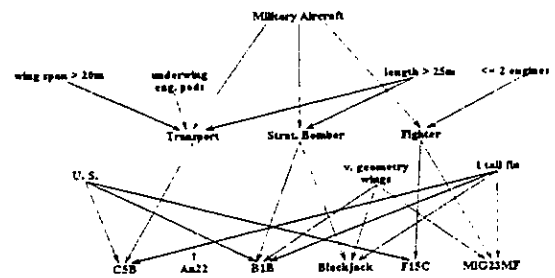


Figura 5 - Exemplo (Telfer,91)

No exemplo, foram utilizados conceitos duais (Transporte, Bombardeiro e Caça) a partir dos conceitos de entrada para determinar os conceitos de saída (C5B, An22, B1B, BlackJack, F15C e Mig23MF).

Para implementação deste exemplo na arquitetura proposta, associamos cada conceito a um elemento processador. Cada elemento das camadas de entrada, dual e saída representa, respectivamente, um conceito de entrada, dual e de saída. Assim, a rede neural construída tem 8 elementos de entrada, 3 duais e 6 de saída. Além destes, foram utilizados 4 elementos na camada escondida.

Para treinar uma rede neural normal (e.g. "Back-Propagation") é preciso levantar os exemplos completos do mapeamento entre os conceitos de entrada e saída. Para treinar a arquitetura proposta é necessário incluir também os conceitos duais nos exemplos, que podem ser obtidos diretamente da hierarquia de conceitos (Figura 5).

	1	2	3	4	5	6	7	8	9
Militar	1	1	1	1	1	1	1	1	1
Envergadura>20m	1	1	1	1	0	0	1	0	0
Tanque nas Asas	1	1	0	0	0	0	1	1	0
Comprimento>25m	1	1	1	1	0	0	1	1	0
Motores<3	0	0	0	0	1	1	0	0	1
Americano	1	0	1	0	1	0	0.3	0.3	0.3
Asas em V	0	0	1	1	0	1	0.3	0.3	0.3
Quilhas=1	1	0	1	1	0	1	0.3	0.3	0.3
Transporte	1	1	0	0	0	0	1	0	0
Bombardeiro	0	0	1	1	0	0	0	1	0
Caça	0	0	0	0	1	1	0	0	1
C5B	1	0	0	0	0	0	0.5	0	0
An22	0	1	0	0	0	0	0	0	0
B1B	0	0	1	0	0	0	0	0	0
BlackJack	0	0	0	1	0	0	0	0.5	0
F15C	0	0	0	0	1	0	0	0	0.5
Mig23MF	0	0	0	0	0	1	0	0	0.5

Tabela 1 - Treinamento e Testes

Para ensinar a rede foram utilizados os seis exemplos (Tabela 1: exs.1 a 6) fornecidos (um para cada aeronave). Para testá-la foram utilizados três outros exemplos (Tabela 1: exs.7 a 9) que servem para determinar se a rede aprendeu corretamente a hierarquia desejada. Nestes exemplos, os três conceitos de entrada que não foram utilizados pelos especialistas (Americano, AsasEmV, Quilhas=1) na determinação dos conceitos duais são representados por um valor médio (0.3). Os testes mostram que a rede foi capaz de determinar com precisão (valor 1) o tipo (Transporte, Bombardeiro ou Caça) e de sugerir uma aeronave (valor 0.5).

O simulador utilizado foi o SNNSv3.1 ("Stuttgart Neural Network Simulator"), da Universidade de Stuttgart, em uma "workstation" da DEC. Devido à implementação desse algoritmo neste simulador, foram necessárias 6 iterações ao invés de 3 (altura da hierarquia) para cada exemplo. Também foram realizados testes com sucesso na Detecção de Falhas em Transformadores. Os resultados destes testes serão em breve divulgados em artigos mais específicos.

### VIII - Conclusões

A arquitetura proposta pode ser interpretada de dois modos distintos: como uma rede parcialmente recursiva com memória determinada externamente, ou

como uma arquitetura única equivalente a redes neurais "feed-forward" hierárquicas.

O desenvolvimento de um algoritmo para aprendizado a partir de exemplos com dados incompletos representaria um avanço significativo na aplicabilidade das Redes Neurais, e permitiria a plena utilização da arquitetura proposta.

Um dos resultados obtidos mais interessantes é que o maior problema do algoritmo selecionado (a determinação do número necessário e suficiente de intervalos de tempo) desaparece quando da aplicação deste na arquitetura proposta.

Finalmente, os testes efetuados demonstraram a eficácia da utilização da arquitetura em problemas de classificação, e demonstraram seu potencial para utilização em aplicações ainda mais complexas (e.g. Controle).

### IX - Bibliografia:

- [Elman,90] "Finding Structure in Time". J.L.Elmán, Cognitive Science, vol.14, no.2, pp.179-211
- [Jordan,89] "Generic Constraints on Underspecified Target Trajectories", M.I.Jordan. IJCNN'89, vol.1, pp.217-225
- [Pearlmutter,89] "Learning State Space Trajectories in Recurrent Neural Networks". B.A.Pearlmutter, ICNN' 89, vol.II, pp.365-372
- [Pineda,89] "Recurrent Back-Propagation and the Dynamical Approach to Adaptive Neural Computation". F.J.Pineda, Neural Computation, vol.1, pp.161-172
- [Rumelhart,86] "Parallel Distributed Processing: Explorations in the Microstructure of Cognition". D.E.Rumelhart & J.L.McClelland, The MIT Press. Cambridge. Massachusetts
- [Telfer,91] "Neural Closure Associative Processor", B.Telfer & D.Casasent, Neural Networks, vol.4, pp.589-598
- [Woods,91] "Understanding Subsumption and Taxonomy: A Framework for Progress". W.A.Woods, "Principles of Semantic Networks: Explorations in the Representation of Knowledge". J.F.Sowa, M.Kaufmann Publishers Inc.
- [Zipser,89] "Encoding Sequential Structure: Experience with the Real-Time Recurrent Learning Algorithm". D.Zipser & A.W.Smith. IEEE ICNN'89, vol.I, pp.645-648