

Redes Neurais Artificiais

Simulação: Base de Dados, Codificação, Parâmetros
Aprendizado: Generalização, Avaliação de Resultados

Unisinos - 2001/2

Curso de Informática

Disciplina: Redes Neurais

Prof. Fernando Osório



Centro de Ciências Exatas e Tecnológicas - C6 / E-mail: osorio@exatas.unisinos.br
Curso de Informática - Web: <http://inf.unisinos.br/~osorio/neural.html>

F. OSÓRIO - UNISINOS 2001

TEMAS DE ESTUDO: REDES NEURAIIS ARTIFICIAIS

Modelo de Redes Neurais: MLP com BP

- Multi-Layer Perceptron com Back-Propagation
- Algoritmo de Aprendizado e suas variações (inércia, esquecimento, etc)

Simulação: MLP BackProp

- Definição da topologia e interconexão dos elementos da rede
- Definição da inicialização dos pesos sinápticos
- Definição do tipo de função de transferência dos neurônios (sigmoide, linear, ...)
- Definição dos parâmetros básico: Alfa (Velocidade), μ / Momentum (Inércia),
- Definição de parâmetros avançados: *weight decay*, *temperature*, *sp-offset*
- Definição do modo de aprendizado: por exemplo / por lote, ordenado / aleatório
- Definição do critério de parada: máximo de épocas, limite de erro
- Exemplos de simuladores: Nevprop, Neusim, SNNS, PDP++, outros

Simulação: Arquivos de Aprendizado e de Teste de Generalização

- Arquivo de Aprendizado, Teste de Generalização e Saída
- Validação cruzada (cross-validation)
- Critério de parada do aprendizado: parada tardia e generalização

Simulação: Avaliação dos Resultados

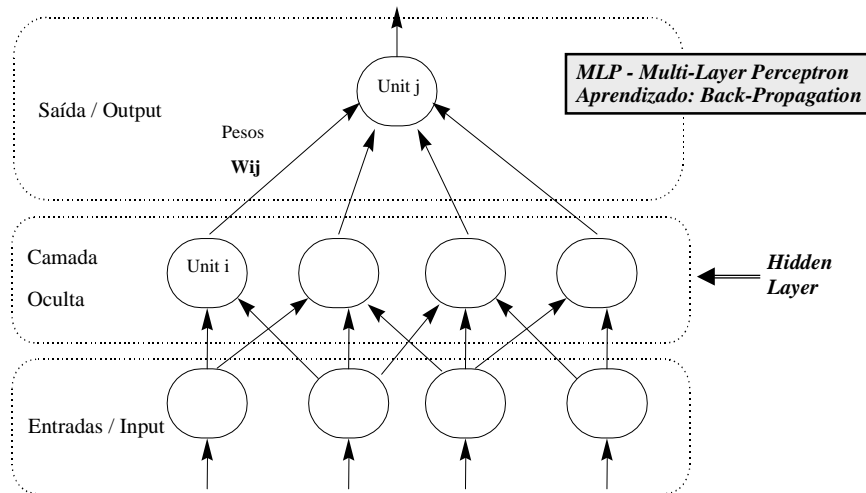
- Grau de Acerto: saída binária / saída contínua, saída única / saídas múltiplas
- Criação de um experimento cientificamente válido: resultados reproduzíveis

F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:

MLP / BackProp

- Redes Multi-Nível: Aprendizado de problemas - Classes não linearmente separáveis
Rumelhart, Hinton & Williams 1986 - Livro "PDP - Parallel Distributed Processing"



F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:

Algoritmo de Aprendizado Back-Propagation

- Rede com uma camada única:

Ativação

$$Y_i = F_{\text{transf}}(\sum X_i * W_{ij} + \text{Bias}) \quad \therefore \quad X_i = \text{Entrada 'i'}$$
$$W_{ij} = \text{Peso da entrada 'i' do neurônio 'j'}$$
$$F_{\text{transf}}(S_{\text{POND}}) = \text{Sigmoid}(S_{\text{POND}}) \quad Y = \frac{1}{1 + \text{Exp}(-T \cdot S_{\text{pond}})}$$

$T = \text{Temperatura da sigmoide}$

Adaptação dos Pesos

$$\text{Erro}_i = D_i - A_i \quad \therefore \quad D_i = \text{Saída desejada (desired) do neurônio}$$
$$A_i = \text{Saída efetiva (actual) do neurônio}$$

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij} \quad \therefore \quad W_{ij}(t) = \text{Peso } W_{ij} \text{ no tempo 't'}$$
$$\therefore \quad W_{ij}(t+1) = \text{Peso no tempo 't+1' (após adaptação)}$$

$$\Delta W_{ij} = \alpha \cdot \text{Erro}_i \cdot F'_{\text{transf}}(S_{\text{POND}}) \cdot X_i \quad \therefore \quad F'_{\text{transf}}(X) = \text{Derivada da fç de transferência}$$
$$F'_{\text{transf}}(X) = F(X) \cdot (1 - F(X)) \quad [\text{Sigmoide}]$$
$$\alpha = \text{Velocidade de aprendizado}$$

F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:*Algoritmo de Aprendizado Back-Propagation*

• Rede com uma única camada:

Ativação

$$Y_i = F_{\text{transf}} (\sum X_i * W_{ij} + \text{Bias}) \quad \therefore \quad X_i = \text{Entrada 'i'}$$

$$F_{\text{transf}} (S_{\text{POND}}) = \text{Sigmoid}(S_{\text{POND}}) \quad \therefore \quad W_{ij} = \text{Peso da entrada 'i' do neurônio 'j'}$$

$$Y = \frac{1}{1 + \text{Exp}^{(-T.Spond)}} \quad T = \text{Temperatura da sigmoide}$$

Adaptação dos Pesos

$$\text{Erro}_i = D_i - A_i \quad \therefore \quad D_i = \text{Saída desejada (desired) do neurônio}$$

$$A_i = \text{Saída efetiva (actual) do neurônio}$$

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}(t) \quad \therefore \quad W_{ij}(t) = \text{Peso } W_{ij} \text{ no tempo 't'}$$

$$\therefore \quad W_{ij}(t+1) = \text{Peso no tempo 't+1' (após adaptação)}$$

Momentum = Inércia

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}(t) + \mu \cdot \Delta W_{ij}(t-1) \quad \therefore \quad \mu = \text{Fator de inércia (momentum)}$$

F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:*Algoritmo de Aprendizado Back-Propagation*• Rede com mais de uma camada:**Ativação => Feed-Forward:** Ativa a rede a partir das entradas e propaga para as saídas**Adaptação dos Pesos - Output Layer => Similar a rede de uma única camada****Adaptação dos Pesos - Hidden Layer => Propagação da “culpa no erro da camada seguinte”****IF** Neurônio_i ∈ Saída**THEN** Erro_Estimado_i = Saída_Desejada_i - Saída_Efetiva_i**ELSE** Erro_Estimado_i = $\sum W_{kj} \cdot \text{Erro_Estimado}_j$ **Onde:** W_{jk} = Peso do neurônio ‘j’ para a entrada ‘k’ (a entrada ‘k’ vem do neurônio ‘i’)Erro_Estimado_j = Erro do neurônio ‘j’ (camada de saída)

Este neurônio possui uma conexão de entrada vinda do neurônio ‘i’

Erro_Estimado_i = Erro do neurônio ‘i’ (camada oculta / intermediária),

Este neurônio possui uma conexão de saída ligada ao neurônio ‘j’

Segue com o cálculo... $\Delta W_{ij} = \alpha \cdot \text{Erro}_i \cdot F'_{\text{transf}}(S_{\text{POND}}) \cdot X_i$

F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:

Algoritmo de Aprendizado Back-Propagation

• Características Adicionais:

Esquecimento = *Weight Decay*

- Pequena perda do valor dos pesos (esquecimento)
- Perda 1: o peso é diminuído de forma a tender para zero
- Perda 2: o peso é diminuído de forma a tender para o seu valor inicial
- O valor do weight decay deve ser uma fração da correção ΔW aplicada aos pesos

Tipo de Aplicação

- Classificação: usar sigmoide em todas as unidades
- Aproximação de funções: não usar a sigmoide na saída de rede (usualmente linear)

Ajuste de parâmetros

- Velocidade de aprendizado: 0 a 1 (use um valor baixo)
- Inércia: 0 a 1 (use um valor alto, ou então não use)

Comportamento de aprendizado anômalos:

- Dependência do “ponto de partida” (sem garantia de sucesso)
- Erro com variação irregular => Parâmetros mal configurados - Precisa convergir!
- Paralisia do aprendizado: incapacidade, mínimo local, ou *flat spot*

F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:

Simulação de Redes Neurais

Configurando a Simulação: MLP BackProp

- Definição da topologia e interconexão dos elementos da rede
Nro. de neurônios, disposição e interconexão dos neurônios
- Definição da inicialização dos pesos sinápticos
Intervalo de valores de inicialização, semente de nros. aleatórios
- Definição do tipo de função de transferência dos neurônios
Sigmoide, linear, gaussiana, etc
- Definição dos parâmetros básico
Alfa (Velocidade), μ / Momentum (Inércia)
- Definição de parâmetros avançados
Weight decay, *temperature (sigmoid)*, *sp-offset (sigmoid prime offset)*
- Definição do modo de aprendizado (reajuste dos pesos e entrada dos exemplos)
Exemplo-à-exemplo ou época-à-época, exemplos ordenados ou aleatórios
- Definição do critério de parada:
Máximo de épocas, limite de erro

F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:*Simulação de Redes Neurais***Configurando o NEVPROP**

XOR Problem Example

Net: 2-2-1
Epochs: 3000 (average)

To use with NevProp

#

IBMorDOS 0 UseQuickProp 0 EpochWiseUpdate 0
BestByCindex 0 MinEpochs 100 BeyondBestEpoch 2.0
WeightRange 0.01 HyperErr 0 SigmoidPrimeOffset 0.1
Epsilon 0.05 SplitEpsilon 0 Momentum 0.9
WeightDecay 0 ScoreThreshold 0.4
QPMMaxFactor 1.75 QPModeSwitchThreshold 0.0
Ninputs 2 Nhidden 2 Noutputs 1
UnitType 2
Connectcalls 2
1 2 3 4
3 4 5 5
NTrainingPatterns 4 NTestingPatterns 0
0.0 0.0 0
0.0 1.0 1
1.0 0.0 1
1.0 1.0 0

F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:*Simulação de Redes Neurais***Configurando o NEUSIM**

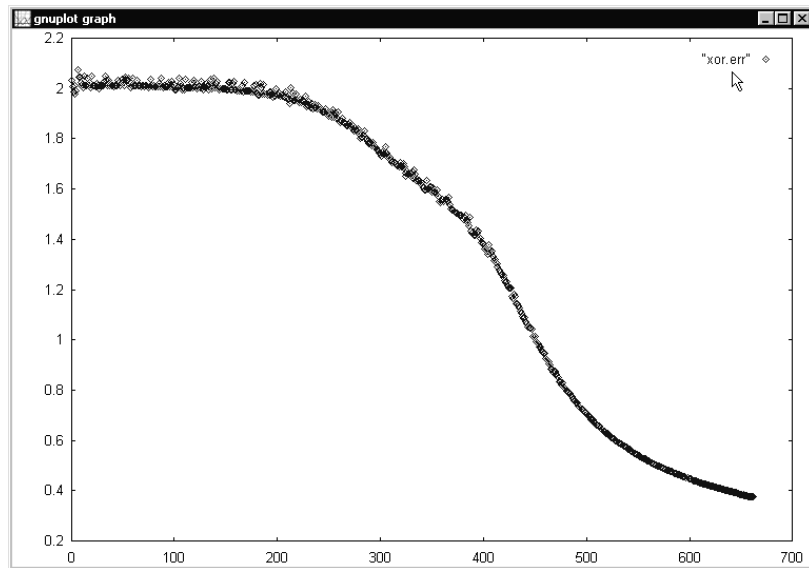
```
%%  
%% NEUSIM Configuration File - XOR with Back-Propagation learning  
%%  
Task            LT            % Learning + Final Test  
MaxEpochs 2000  
NInputs        2  
NOutputs       1  
BackProp       2                      % Learning method: backpropagation with 2 hidden units  
%  
% Learning parameters  
%  
RndRange       1.0  
Epsilon        0.1  
Momentum       0.9  
MaxErr          0.4  
StopCrit       1                      % Stop when 100% correct (learning data)  
SPOffset       0.1  
DataOrder      1                      % Order of data presentation: randomly  
%  
% Report  
%  
RepLevel       128  
ErrLevel       0  
RepFreq        100  
UserFreq       10  
UserRep        1  
%%  
%% End of CFG  
%%
```

F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:

Simulação de Redes Neurais

Usando o NEUSIM - Curva do Erro: Problema do Xor

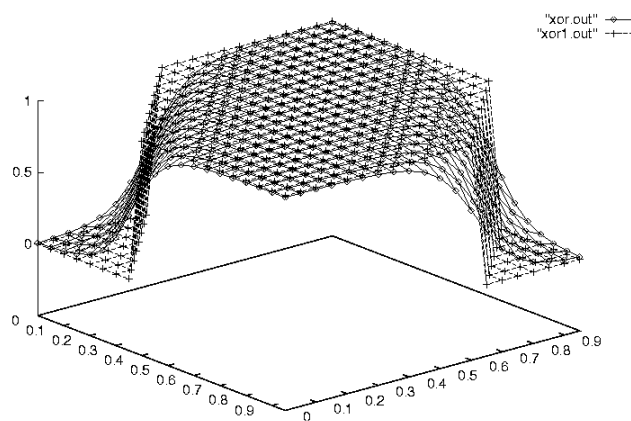


F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:

Simulação de Redes Neurais

Usando o NEUSIM - Curva de Resposta / Generalização: Problema do Xor



F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:*Simulação de Redes Neurais**Simulação: Arquivos de Aprendizado e de Teste de Generalização*

• Arquivo de Aprendizado, Teste de Generalização e Saída

Dados Disponíveis	Dados de Aprendizado	Dados de Teste	Saída
1.2 2.4 0	1.2 2.4 0	1.2 1.8 0	1.31 2.25 ?
1.2 1.2 0	1.2 1.2 0	1.3 1.2 1	
1.2 1.8 0	1.3 2.4 1	1.4 1.2 0	<i>Recall</i>
1.3 2.4 1	1.3 1.8 1		
1.3 1.2 1	1.4 2.4 0	<i>Generalization</i>	
1.3 1.8 1	1.4 1.8 0		
1.4 1.2 0			
1.4 2.4 0	<i>Learn</i>		
1.4 1.8 0			

• Validação cruzada (*cross-validation*)

- Simula Aprendizado + Teste de Generalização simultâneos
- Acompanha a curva de erro: determina a “Best Epoch” (parada tardia)

• Critério de parada do aprendizado: Generalização

- Realiza o aprendizado com validação cruzada
- Repete a simulação EXATAMENTE nas mesmas condições
- Ajusta o Random Seed / Número máximo de épocas = Best Epoch

F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:*Simulação de Redes Neurais**Simulação: Arquivos de Aprendizado e de Teste de Generalização*

• Codificação dos Dados:

Entrada: Binário, Discreta, Contínua (Normalizada ou não - Definir intervalo)

Saída: Binária, Discreta, Contínua, Codificada (Normalizada ou não)

• Pré-processamento e Pós-Processamento: diferentes técnicas

Considerando o contexto

Considerando o tempo

Normalização dos dados

• Métodos de codificação: Valores discretos

	1	2	3
Entradas mutuamente exclusivas	1 0 0 0	/ 0 1 0 0	/ 0 0 1 0
Codificação binária	0 0 0 1	/ 0 0 1 0	/ 0 0 1 1
Termômetro	1 0 0 0	/ 1 1 0 0	/ 1 1 1 0
Gaussiana (intervalo com ou sem sobreposição)	1 0 0 0	/ 0.5 1.0	0.5 0.0

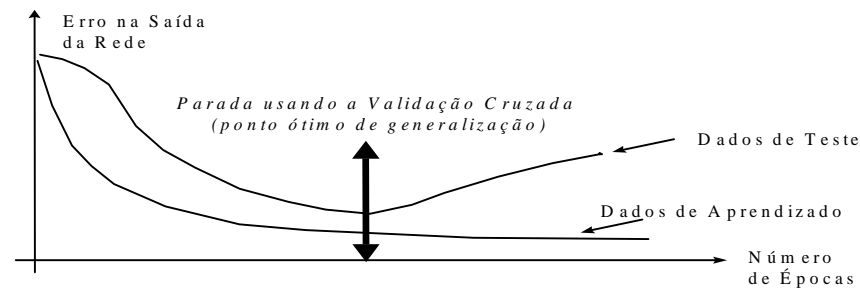
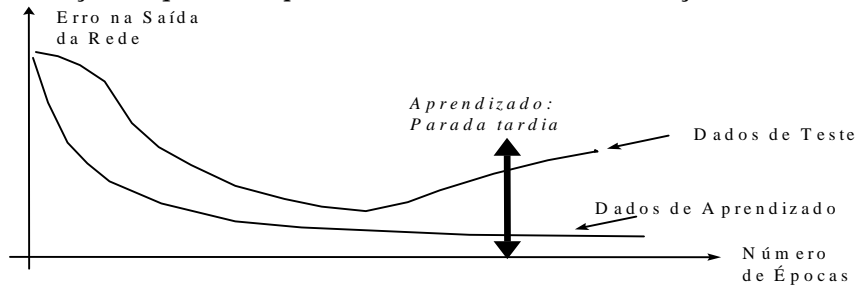
...

F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:

Simulação de Redes Neurais

Simulação: Arquivos de Aprendizado e de Teste de Generalização



F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:

Simulação de Redes Neurais

Simulação: Avaliação dos Resultados

• Performance:

- Tempo: número de épocas
- Precisão: erro residual, grau de acertos (%)
- Generalização e Tamanho da Rede

• Grau de Acerto:

- Saída binária \times saída contínua - Contagem de acertos?
- Saída binário com indecisão: diagnóstico - Sim / Não Sei / Não
- Tabela: Sim (OK), Sim (Erro), Não (OK), Não (Erro)
- Saída única \times saídas múltiplas - Contagem de acertos?
(Winner Take All, Lista de Candidatos Ordenada)

• Criação de um experimento cientificamente válido:

- Resultados reproduzíveis = Múltiplas simulações!
- Registro: estatística dos resultados (média, desvio)
- Descrição detalhada do experimento: configuração / parâmetros

F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:

Simulação de Redes Neurais

MLP Vantagens...

- Solução de problemas não linearmente separáveis
- Sistema adaptativo - Aprendizado baseado na experiência
- Degradação progressiva
- Generalização
- Aproximação: dados inexatos / dados incorretos ou inconsistentes
- Paralelismo e robustez
- Aplicações: classificação e regressão não linear

MLP Desvantagens...

- Definição da arquitetura: neurônios, camadas e interconexões?
- Parâmetros: excesso e dependência - velocidade, momentum, sp-offset, weight decay
- Codificação dos dados: qual a melhor?
- Inicialização dos pesos: 'sorte' e 'azar' na inicialização
- Paralisia do aprendizado: mínimos locais e o flat spot
- Convergência lenta e incerta (depende de vários fatores)
- Instabilidade: esquecimento catastrófico (moving target problem)
- Dilema da plasticidade *versus* estabilidade

F. OSÓRIO - UNISINOS 2001

REDES NEURAIS:

Simulação de Redes Neurais

MLP Desvantagens...

- Caixa preta: o que a rede sabe, o que a rede não sabe? Explicar respostas!
- Redes estáticas *versus* Conhecimento dinâmico
- Não aproveita todos os conhecimentos sobre o problema (teoria?)

MLP Alternativas para acelerar...

- RPROP [Riedmiller]
- QuickProp [Fahlman]
- Conjugated Scaled Gradient [Moller]
- Cascade-Correlation [Fahlman]
- Silva e Almeida
- Métodos de 2a. Ordem (vide FAQ)

MLP Alternativas para melhorar...

- Sistemas Híbridos
 - KBANN: Knowledge Based Artificial Neural Networks [Towell]
 - INSS: Incremental Neuro-Symbolic System [Osorio]
 - Projeto FAGNIS
 - Projeto HMLT

F. OSÓRIO - UNISINOS 2001

TEMAS DE PESQUISA SOBRE REDES NEURAIS ARTIFICIAIS:

*** PAPERS / DOCUMENTAÇÃO SOBRE MLP - BP:**

- **Obra clássica:**
Rumelhart, D.; Hinton, G & Williams, R.
Learning Internal Representations by Error Propagation. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition - Vol. 1. Cambridge: MIT Press, 1986.
- **Lista de Bibliografia** - <http://www.inf.unisinos.br/~osorio/neural/bp-speedup/biblio-mlp+.pdf>
- **Bibliografia** - <http://www.inf.unisinos.br/~osorio/neural/docs/> (cópias locais)
- **Acelerando o Back-Propagation** - <http://www.inf.unisinos.br/~osorio/neural/bp-speedup/>
- **Kangaroo** - <http://www.inf.unisinos.br/~osorio/neural/docs/kangaroo.txt>
- **Artigos sobre Redes Neurais na Web**
Neuroprose: <http://www.enm.bris.ac.uk/research/neural/archives.html>
ftp - [ftp.cis.ohio-state.edu](ftp://ftp.cis.ohio-state.edu) (anonymous) - /pub/neuroprose/
Citeseer Nec: <http://citeseer.nj.nec.com/cs>

• SOFTWARES:

- | | |
|-----------|---|
| • PDP++ | http://www.cnbc.cmu.edu/PDP++/PDP++.html (Linux e Windows) |
| • SNNS | http://www-ra.informatik.uni-tuebingen.de/SNNS/ (Linux) |
| • NevProp | Nevada Propagation Software - Disponível na Unisinos (Linux e Dos) |
| • INSS | Contactar Osório... Disponível na Unisinos (Linux e Dos / Windows) |
| • Outros | http://www.inf.unisinos.br/~osorio/neural/software.html |