

Redes Neurais Artificiais

MLP: Modelos Avançados - Acelerando o Back-Propagation
Modelos: Rprop, Quick-Prop, Cascade-Correlation

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: *MLP: Modelos Avançados*

MLP Desvantagens...

- Definição da arquitetura: nro. de neurônios, camadas e interconexões?
- Parâmetros em excesso: velocidade, momentum, sp-offset, weight decay
- Dependente da inicialização dos pesos
- Paralisia do aprendizado: mínimos locais e o flat spot
- Minimização do erro: lenta e incerta (depende de vários fatores)
- Instabilidade: esquecimento catastrófico (moving target problem)
- Dilema da plasticidade *versus* estabilidade
- Redes estáticas *versus* Conhecimento dinâmico

MLP - Aumentando a performance do Back-Propagation

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

F. OSÓRIO - UNISINOS 2001

ALGORITMO RPROP: MLP: Modelos Avançados - RPROP

Referência:

Martin Riedmiller & Heinrich Braum

A Direct Adaptive Method for Faster BackPropagation Learning - The RPROP Algorithm

RPROP: Adaptive Learning Algorithm => Resilient Propagation

Ajuste do parâmetro de velocidade de aprendizado durante o aprendizado

“Cada vez que a derivada parcial de um determinado peso W_{ij} troca de sinal, indicando que a última atualização dos pesos foi grande demais e o algoritmo pulou sobre o mínimo local, o fator de ajuste individual e específico deste peso (Δ_{ij}) é diminuído por um fator η^- . Se a derivada manteve o mesmo sinal, o fator de ajuste individual do peso é ligeiramente aumentado de modo a acelerar a convergência.”

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^+ * \Delta_{ij}^{(t-1)} & : \text{ if } \frac{\partial E}{\partial w_{ij}}^{(t-1)} * \frac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ \eta^- * \Delta_{ij}^{(t-1)} & : \text{ if } \frac{\partial E}{\partial w_{ij}}^{(t-1)} * \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ \Delta_{ij}^{(t-1)} & : \text{ else} \end{cases} \quad \Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)} & : \text{ if } \frac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ +\Delta_{ij}^{(t)} & : \text{ if } \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ 0 & : \text{ else} \end{cases}$$

where $0 < \eta^- < 1 < \eta^+$

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)}$$

F. OSÓRIO - UNISINOS 2001

ALGORITMO QUICKPROP: MLP: Modelos Avançados - QuickProp

Referência:

Scott E. Fahlman

An Empirical Study of Learning Speed in Back-Propagation Networks

September 1988. CMU Technical Report - CMU-CS-88-162.

QUICKPROP:

“Back-propagation and its relatives work by calculating the partial first derivative of the overall error with respect to each weight. Given this information we can do gradient descent in weight space. If we take infinitesimal steps down the gradient, we are guaranteed to reach a local minimum, and it has been empirically determined that for many problems this local minimum will be a global minimum, or at least a "good enough" solution for most purposes.

Of course, if we want to find a solution in the shortest possible time, we do not want to take infinitesimal steps; we want to take the largest steps possible without overshooting the solution. Unfortunately, a set of partial first derivatives collected at a single point tells us very little about how large a step we may safely take in weight space. If we knew something about the higher-order derivatives -- the curvature of the error function -- we could presumably do much better!

...

It is a second-order method, based loosely on Newton's method, but in spirit it is more heuristic than formal. I then make the assumption that the error vs. weight curve for each weight can be approximated by a parabola whose arms open upward”

F. OSÓRIO - UNISINOS 2001

ALGORITMO QUICKPROP: MLP: Modelos Avançados - QuickProp

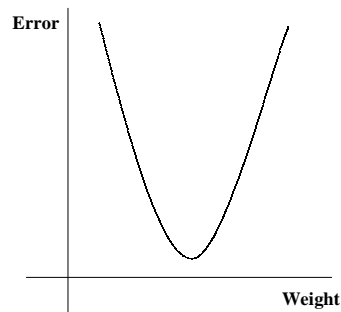
Referência:

Scott E. Fahlman
An Empirical Study of Learning Speed in Back-Propagation Networks
September 1988. CMU Technical Report - CMU-CS-88-162.

QUICKPROP:

$$\Delta w(t) = \frac{S(t)}{S(t-1) - S(t)} \Delta w(t-1)$$

where S(t) and S(t-1) are the current and previous values of $\partial E / \partial w$.



Vantagem:

- Convergência muito rápida

Problemas do QuickProp:

- Convergência rápida demais, pode levar a um estouro da capacidade de representação numérica usada.

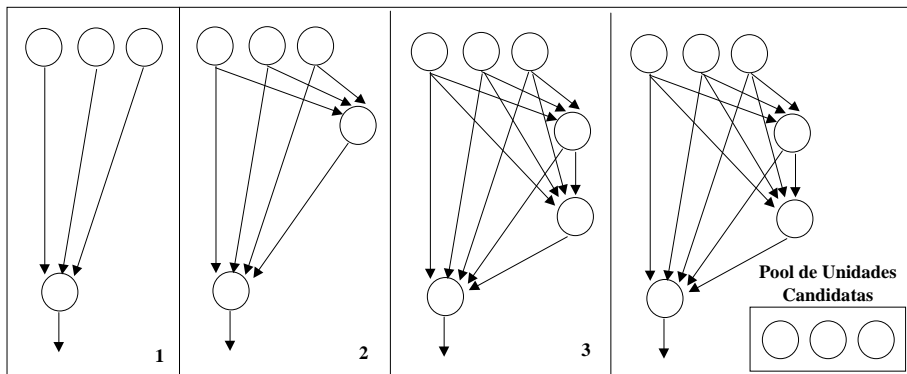
F. OSÓRIO - UNISINOS 2001

ALGORITMO CASCADE-CORRELATION: MLP: Modelos Avançados - CASCOR

Referência:

Scott E. Fahlman & Christian Lebiere
The Cascade-Correlation Learning Architecture.
February 1990. CMU Technical Report - CMU-CS-90-100.

CASCOR: Rede Neural Construtiva - Baseado no uso do QuickProp



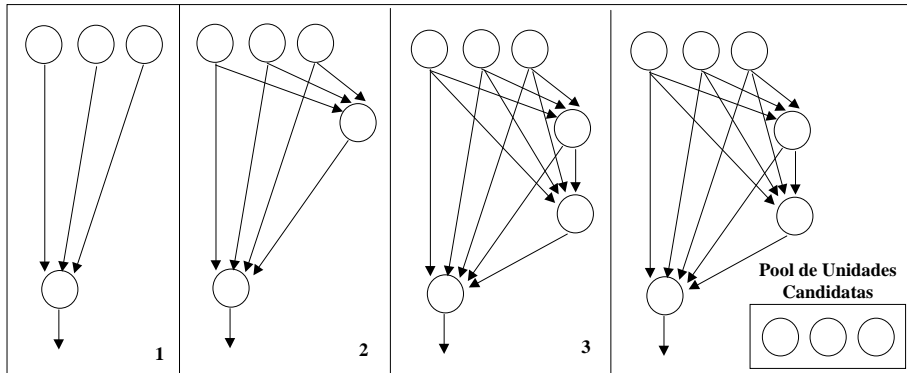
Aprendizado: alternância unidades de saída e unidades candidatas
Limitações: Batch Learning - Classificação (saídas binárias)

F. OSÓRIO - UNISINOS 2001

ALGORITMO CASCADE-CORRELATION: MLP: Modelos Avançados - CASCOR

Aprendizado:

- Ajusta os pesos dos neurônios de saída a fim de minimizar o erro, até a “paciência estourar”, ou parando o aprendizado se os objetivos forem alcançados;
- Cria um pool de unidades candidatas, conectadas a todas unidades das camadas superiores, e maximiza a correlação entre o erro residual e a saída desta unidade (feature detector), parando o ajuste dos pesos quando a “paciência estourar”;
- Adiciona a melhor unidade candidata, conectando ela a saída da rede e congelando seus pesos;
- Volta ao início do algoritmo.

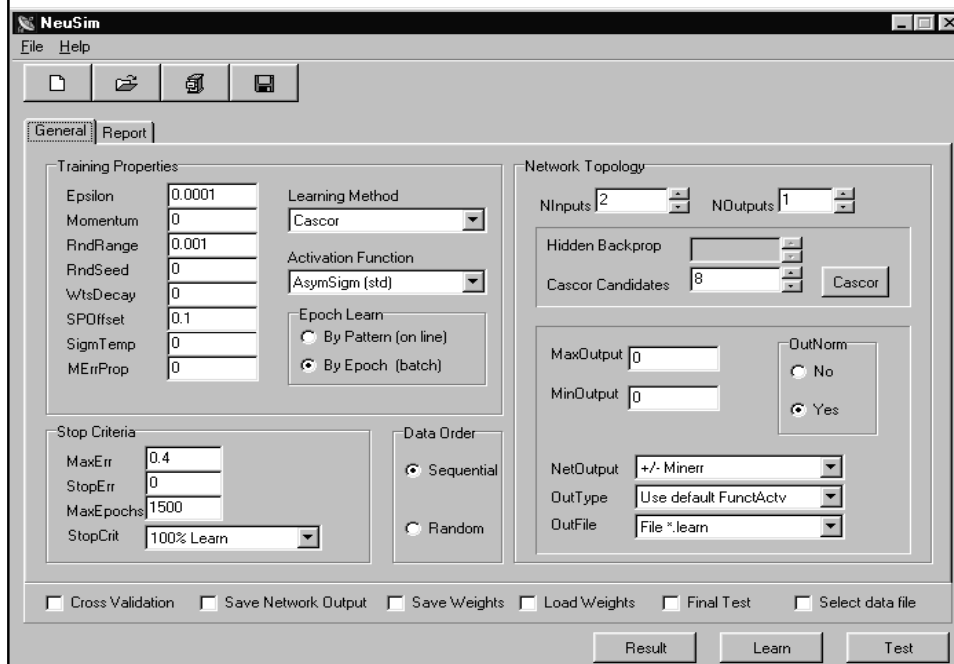


Aprendizado: rápido e de fácil configuração

Limitações: Batch Learning - Classificação (saídas binárias)

F. OSÓRIO - UNISINOS 2001

SIMULADOR: NEUSIM inclui modelos avançados - CASCOR e QUICKPROP



TEMAS DE PESQUISA SOBRE REDES NEURAIAS ARTIFICIAIS: MLP

*** 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> (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>