

CPE 723 - OTIMIZAÇÃO NATURAL

3 módulos:

OTIMIZAÇÃO CONTÍNUA - Calôba

RECOZIMENTO SIMULADO - Gabriel

ALGORÍTMOS GENÉTICOS - Alexandre

Objetivos

Pré-requisitos

Avaliação

Por módulo { séries exercícios (1-2)
teste (1)



Conceito final

Módulo I - Otimização Contínua

Parte 1 – Métodos de Otimização

Aproximação local (Taylor), Gradiente $\underline{\nabla}$, Hessiana $\underline{H}$

Extremos, mínimo, máximo

Propriedades $\underline{\nabla}$ e $\underline{H}$

Métodos analíticos

Métodos numéricos recursivos

Gradiente descendente, passo, passo variável, ótimo

Métodos de 2ª. Ordem: Gradiente conjugado, Newton,
Newton amortecido, Quasi Newton

Módulo I - Otimização Contínua

Parte 2 – Treinamento como um Processo de Otimização

Erro

Treinamento

Batelada,
Lotes,
Regra Delta, Momento
Outros métodos

Parte 3 - Comentários Finais

Outras funções objetivo

Acompanhamento do treinamento

Crítica pós treinamento

Referências bibliográficas principais:

Cichocki, Unbehauen – “Neural Networks for Optimization and Signal Processing”, Cap. 1 e 3, Wiley, 1993.

Chong, Zak – “An Introduction to Optimization”, Part II, Wiley, 2001.

Addy, Dempster – “Introduction to optimization methods”, Chapman and Hall, 1974.

www.lps.ufrj.br/~caloba/cpe723

[cpe723_slides.pdf](#)

[cpe723_texto.pdf](#)

[cpe723_exercícios.pdf](#)

[cichocki&unbehauen.pdf](#)

Neural Networks for Optimization and Signal Processing

Andrzej Cichocki
Warsaw University of Technology
Poland

Rolf Unbehauen
Universität Erlangen-Nürnberg
Germany

* Chapter 1. Mathematical Preliminaries of Neurocomputing	1
* 1.1. Linear Matrix Algebra	1
* 1.1.1. Matrix Representations and Notations	1
* 1.1.2. Inner and Outer Product	3
1.1.3. Linear Independence of Vectors	4
1.1.4. Rank of a Matrix	5
* 1.1.5. Positive and Negative Definite Matrices	5
* 1.1.6. The Inverse and Pseudoinverse of Matrices	5
1.1.7. Orthogonality, Unitary Matrices, Conjugate Vectors	7
1.1.8. Eigenvalues and Eigenvectors	7
1.1.9. Vector and Matrix Norms	9
1.1.10. Singular Value Decomposition (SVD)	10
1.1.11. Condition Numbers	12
1.1.12. The Kronecker Product	14
1.2. Elements of Multivariable Analysis	15
1.2.1. Sets	15
* 1.2.2. Functions	16
* 1.2.3. Differentiation of a Scalar Function with Respect to a Vector	17
* 1.2.4. The Hessian Matrix	17
1.2.5. The Jacobian Matrix	18
* 1.2.6. Chain Rule	19
* 1.2.7. Taylor Series Expansion and Mean Value Theorems	20
1.3. Lyapunov's Direct Method	21
1.4. Unconstrained Optimization Algorithms	23
* 1.4.1. Necessary and Sufficient Conditions for an Extremum	23
* 1.4.2. Dynamic Gradient Systems	23
* 1.4.3. Newton's Methods	25
* 1.4.4. The Quasi-Newton Methods	27
* 1.4.5. The Conjugate Gradient Method	25
1.5. Constrained Nonlinear Programming Problems	25
1.5.1. Kuhn-Tucker Conditions	25
1.5.2. Lagrange Multipliers and Kuhn-Tucker Conditions for Constrained Minimization with Mixed Constraints	31

* 3.1. The Use of Systems of Ordinary Differential Equations in Unconstrained Optimization Problems – Trajectory-Following Methods	89
3.1.1. Basic Iterative Gradient Descent Algorithms	89
3.1.2. Continuous-Time Realization of Iterative Algorithms	91
3.1.3. Basic Gradient Systems	93
3.1.4. Continuous-Time Algorithm with Prespecified Convergence Speed	96
* 3.2. Unconstrained Optimization by Applying a System of Second-Order Differential Equations	102
3.3. Branin's Method	104
○ 3.4. Optimization Networks Using a Combination of Deterministic and Random Search – Stochastic Gradient Algorithms	107
○ 3.5. Boltzmann Machine and Simulated Annealing	113
○ 3.6. Mean-Field Annealing Algorithm	117
3.7. Back-Propagation Learning Algorithms	122
3.7.1. Learning of the Single Layer Perceptron	122
3.7.2. Standard Back-Propagation Algorithm for the Multilayer Perceptron	127
* 3.7.3. Back-Propagation Algorithm with Momentum Updating	132
* 3.7.4. Batch Learning Algorithm	133
* 3.7.5. Comparison of the On-Line and the Batch Procedures	135
3.7.6. Back-Propagation Algorithm with a Variable Number of Neurons in the Hidden Layers	136
* 3.8. Back-Propagation Algorithms with Non-Euclidean Error Signals	137
* 3.9. Speeding up the Back-Propagation Learning Algorithms	142
3.9.1. Back-Propagation Learning Algorithm with an Adaptive Slope of the Activation Functions	143
* 3.9.2. Search-Then-Converge Strategy	144
3.9.3. Averaging Procedure	145
* 3.9.4. Global Adaptation of the Learning Rate and/or Momentum Rate	146
* 3.9.5. Local Adaptation of Learning Rates	148
* 3.9.6. Quickprop	151
3.10. Generalized Learning Algorithm for a Single Neuron	152
3.10.1. Generalized LMS Learning Rule	154
3.10.2. Potential Learning Rule	156
3.10.3. Correlation Learning Rule	156
3.10.4. Hebbian Learning Rule	158
3.10.5. Oja's Learning Rule	158
3.10.6. Standard Perceptron Learning Rule	159
3.10.7. Generalized Perceptron Learning Rule	159
Questions and Problems for Chapter 3	160
3.11. References and Sources for Further Reading	162

Parte 1 – Métodos de Otimização Contínua “clássica”

$$F(w_1, w_2, \dots, w_n) = F(\underline{w})$$

$$\underline{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

Propriedades

$$F(\underline{w}) \geq 0$$

$$\frac{\partial F}{\partial w_i} \quad \exists \quad \forall w_i, \quad \forall \underline{w}$$

$$\frac{\partial^2 F}{\partial w_i \partial w_j} \quad \exists \quad \forall w_i, w_j, \quad \forall \underline{w}$$

Como encontrar $\underline{w}^*$ tal que

$$F_0(\underline{w}^*) \leq F_0(\underline{w}) \quad \forall \underline{w} \neq \underline{w}^* \quad ?$$

Otimização de F_0

não linear

não convexa

w ordem elevada

irrestrita

contínua, derivável

Como fazer ???

Noções Elementares de Otimização

I – Função objetivo – a minimizar

$$F(w_1, w_2, \dots) = F(\underline{w})$$

$$\frac{\partial F}{\partial w_i} \text{ existe } \forall w_i$$

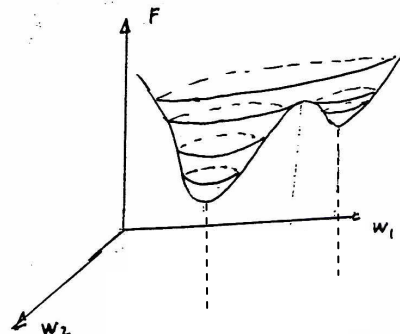
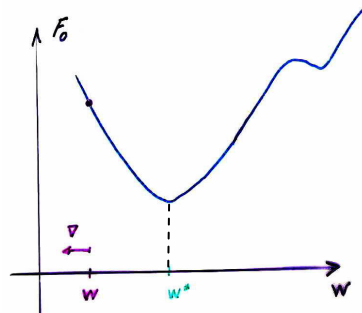
Mínimo, minimante $\underline{w}^*$

$$F(\underline{w}^*) < F(\underline{w}^* + \underline{\Delta w}) \quad \forall \underline{\Delta w} \mid |\underline{\Delta w}| < \varepsilon$$

→ mínimo local

Mínimo global

$$F(\underline{w}^*) \leq F(w) \quad \forall w \neq \underline{w}^*$$



II – Aproximações no entorno de um ponto qualquer $\underline{w}_0$ (Taylor)

$$\underline{w} = \underline{w}_0 + \underline{\Delta w} \quad |\underline{\Delta w}| \leq \varepsilon$$

$$F(\underline{w}_0 + \underline{\Delta w}) = F(\underline{w}_0) + \sum_i \left. \frac{\partial F}{\partial w_i} \right|_{\underline{w}_0} \Delta w_i + \frac{1}{2} \sum_i \sum_j \left. \frac{\partial^2 F}{\partial w_i \partial w_j} \right|_{\underline{w}_0} \Delta w_i \Delta w_j + \dots$$

considerando

$$\underline{\Delta w} = [\Delta w_i] \quad \left[\nabla F(\underline{w}_0) = \left. \frac{\partial F}{\partial w_i} \right|_{\underline{w}_0} \right] \quad \text{gradiente}$$

$$\underline{H}(\underline{w}_0) = \left[\left. \frac{\partial^2 F}{\partial w_i \partial w_j} \right|_{\underline{w}_0} \right] \quad \text{Hessiana (simétrica)}$$

$$F(\underline{w}_0 + \Delta \underline{w}) = F(\underline{w}_0) + \sum_i \left. \frac{\partial F}{\partial w_i} \right|_{\underline{w}_0} \Delta w_i + \frac{1}{2} \sum_i \sum_j \left. \frac{\partial^2 F}{\partial w_i \partial w_j} \right|_{\underline{w}_0} \Delta w_i \Delta w_j + \dots$$



$$\underline{\Delta w}^t \underline{\nabla} F(\underline{w}_0)$$



$$\underline{\Delta w}^t \underline{H}(\underline{w}_0) \underline{\Delta w}$$

Então

$$F(\underline{w}_0 + \Delta \underline{w}) = F(\underline{w}_0) + \underline{\Delta w}^t \underline{\nabla} F(\underline{w}_0) + \frac{1}{2} \underline{\Delta w}^t \underline{H}(\underline{w}_0) \underline{\Delta w} + \dots$$

Aproximação linear ou de 1ª. ordem

$$F(\underline{w}_0 + \underline{\Delta w}) \simeq F(\underline{w}_0) + \underline{\Delta w}^t \underline{\nabla} F(\underline{w}_0)$$

Obs: Se $F(\underline{w})$ é linear a aproximação de 1ª. ordem é exata. $\underline{\nabla} F(\underline{w}) = \text{ctte}(\underline{w})$ e as derivadas de ordem > 1 são nulas, i.e, a Hessiana $\underline{H}(\underline{w})$ é nula para todo $\underline{w}$.

Aproximação quadrática ou de 2ª. ordem

$$F(\underline{w}_0 + \underline{\Delta w}) \simeq F(\underline{w}_0) + \underline{\Delta w}^t \underline{\nabla} F(\underline{w}_0) + \frac{1}{2} \underline{\Delta w}^t \underline{H}(\underline{w}_0) \underline{\Delta w}$$

Obs: Se $F(\underline{w})$ é quadrática a aproximação de 2ª. ordem é exata. $\underline{H}(\underline{w}) = \text{ctte}(\underline{w})$ e as derivadas de ordem > 2 são nulas

Aproximação do Gradiente

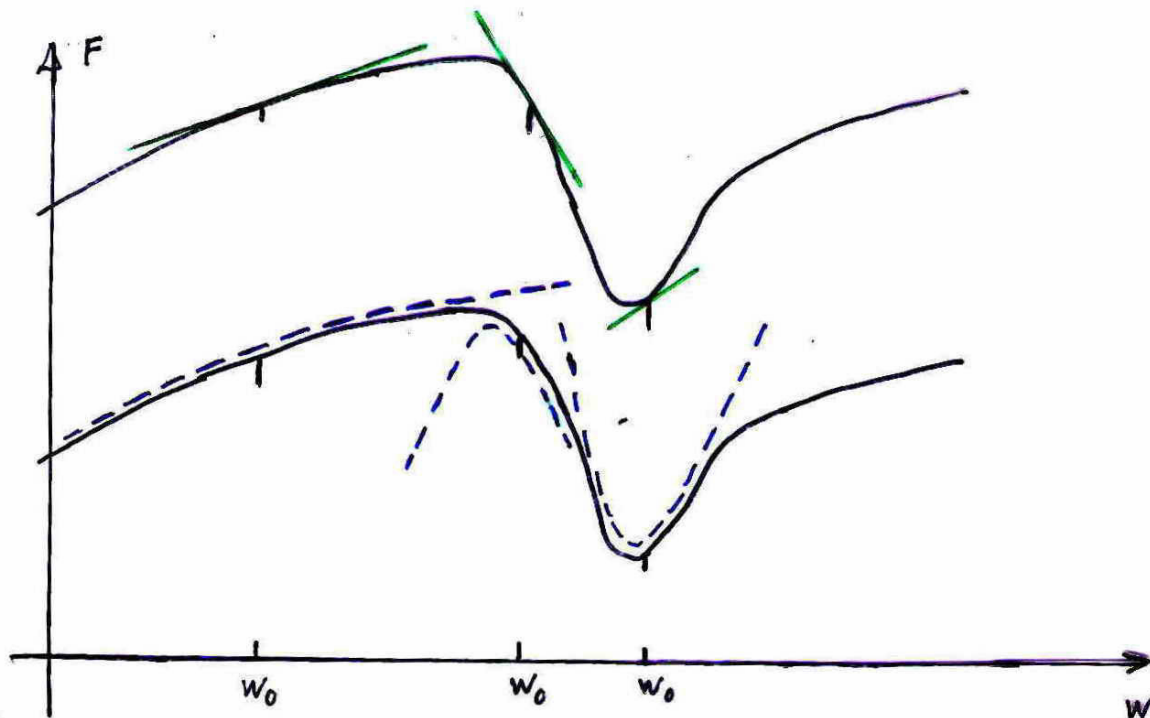
$$F(\underline{w}_0 + \Delta \underline{w}) \cong F(\underline{w}_0) + \sum_i \left. \frac{\partial F}{\partial w_i} \right|_{\underline{w}_0} \Delta w_i$$

$$\left. \frac{\partial F}{\partial w_i} \right|_{\underline{w}_0 + \Delta \underline{w}} \cong \left. \frac{\partial F}{\partial w_i} \right|_{\underline{w}_0} + \sum_j \frac{\partial}{\partial w_j} \left. \frac{\partial F}{\partial w_i} \right|_{\underline{w}_0} \Delta w_j$$

$$\underline{\nabla} F(\underline{w}_0 + \Delta \underline{w}) \cong \underline{\nabla} F(\underline{w}_0) + \underline{H}(\underline{w}_0) \Delta \underline{w}$$

Funções de ordem mais alta

validade das aproximações ?



A validade da aproximação

depende do acréscimo na função,

e não do acréscimo nas variáveis independentes!

e.g., a aproximação de primeira ordem

$$F(\underline{w}_0 + \underline{\Delta w}) \simeq F(\underline{w}_0) + \underline{\Delta w}^t \underline{\nabla} F(\underline{w}_0)$$

é válida enquanto

$$\left| \frac{1}{2} \underline{\Delta w}^t \underline{H}(\underline{w}_0) \underline{\Delta w} \right| \ll \left| \underline{\Delta w}^t \underline{\nabla} F(\underline{w}_0) \right|$$

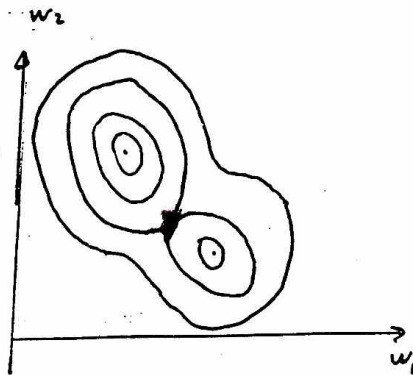
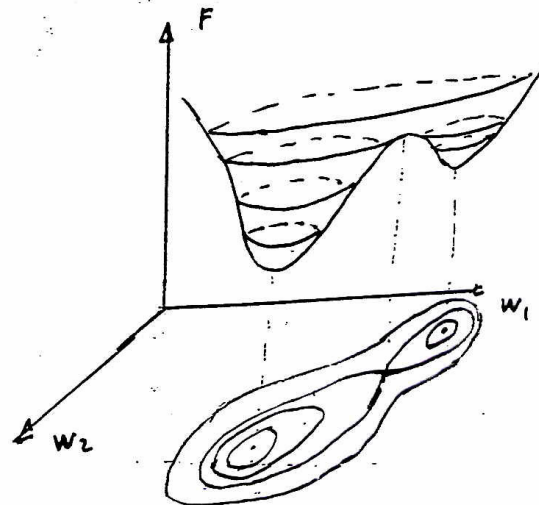
ou, alternativamente, enquanto

$$\left| \frac{1}{2} \underline{\Delta w}^t \underline{H}(\underline{w}_0) \underline{\Delta w} \right| \ll \epsilon$$

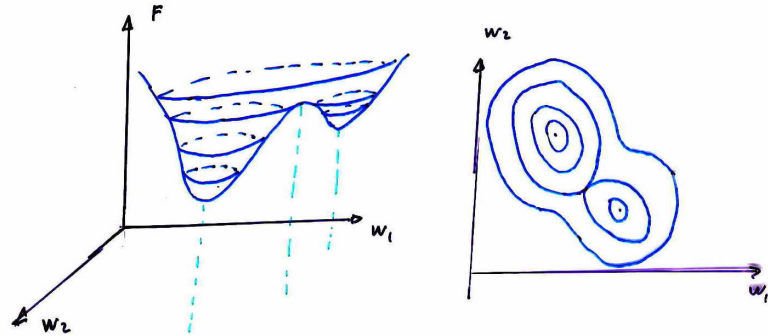
Funções de duas variáveis

Visualização por

Curvas de Nível



Curvas / Superfícies de nível

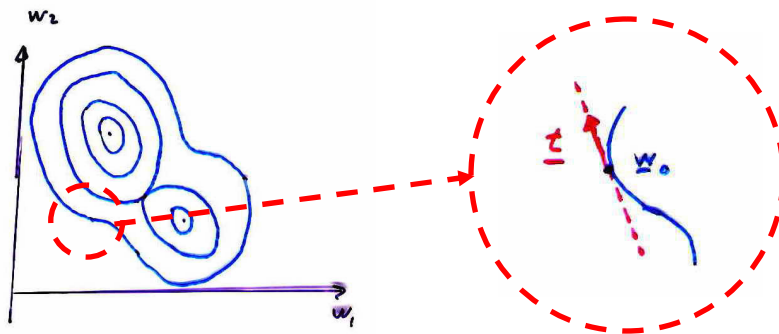
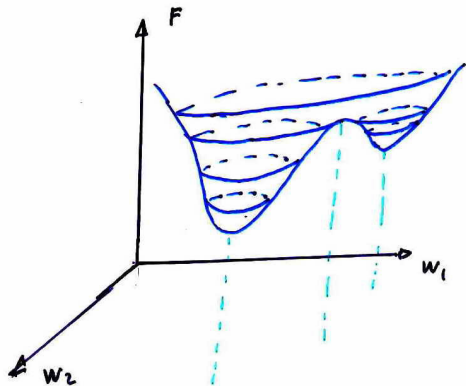


$\underline{w}$
(dim 2) $\rightarrow$ curva de nível
(dim 2) $\rightarrow$ reta t_g
(dim 2)

$\underline{w}$
(dim 3) $\rightarrow$ superfície de nível
(dim 3) $\rightarrow$ plano t_g
(dim 3)

$\underline{w}$
(dim n) $\rightarrow$ superfície de nível
(dim n) $\rightarrow$ hiperplano t_g
(dim n)

Propriedade:



$$\left. \frac{dF}{d\underline{w}} \right|_{\underline{t}} = 0$$

$$\left. \frac{d}{d\alpha} F(\underline{w}_0 + \alpha \underline{t}) \right|_{\alpha=0} = 0$$

III – Extremos:

Mínimo, minimante $\underline{w}^*$

$$F(\underline{w}^* + \underline{\Delta w}) > F(\underline{w}^*) \quad \forall \underline{\Delta w} \quad | \quad |\underline{\Delta w}| \leq \varepsilon$$

$$F(\underline{w}^* + \underline{\Delta w}) \cong F(\underline{w}^*) + \underline{\Delta w}^t \underline{\nabla F} + \frac{1}{2} \underline{\Delta w}^t \underline{H} \underline{\Delta w}$$

$$\underline{\Delta w}^t \underline{\nabla F} = |\underline{\Delta w}| |\underline{\nabla F}| \cos \angle \underline{\Delta w}, \underline{\nabla F} > 0$$

+ + +/-

$$\Rightarrow \underline{\nabla F}(\underline{w}^*) = \underline{0}$$

$$\underline{\Delta w}^t \underline{H} \underline{\Delta w} > 0$$

$$\Rightarrow \underline{H}(\underline{w}^*) \text{ definida positiva}$$

Nos extremos

$$\underline{\nabla F}(\underline{w}^*) = \underline{0}$$

$$F(\underline{w}^* + \underline{\Delta w}) = F(\underline{w}^*) + \Delta F$$

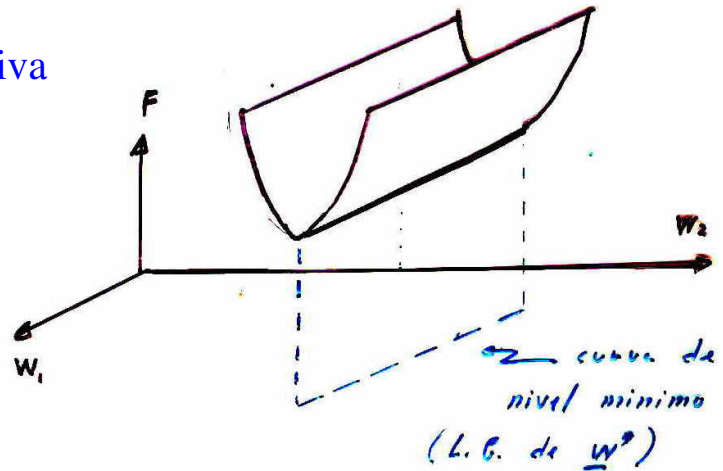
$$= F(\underline{w}^*) + \frac{1}{2} \underline{\Delta w}^t \underline{H} \underline{\Delta w}$$

$$\Delta F = \frac{1}{2} \underline{\Delta w}^t \underline{H} \underline{\Delta w} \quad \left\{ \begin{array}{ll} > 0 & \underline{H} \quad \text{definida positiva} \\ \geq 0 & \underline{H} \quad \text{semi-definida positiva} \\ \leq 0 & \underline{H} \quad \text{semi-definida negativa} \\ < 0 & \underline{H} \quad \text{definida negativa} \\ \pm & \underline{H} \quad \text{n\~ao definida} \end{array} \right.$$

H semidefinida positiva

$$\Delta F \geq 0$$

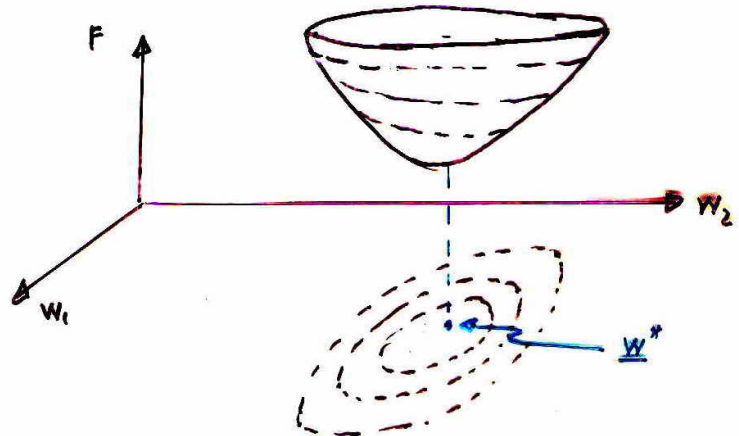
CALHA



H definida positiva

$$\Delta F > 0$$

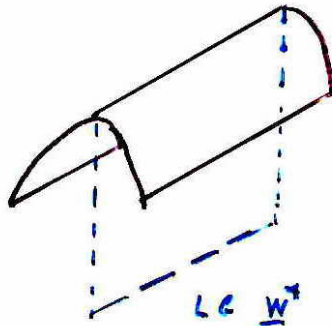
MÍNIMO
propriamente dito



H semidefinida
negativa

$$\Delta F \leq 0$$

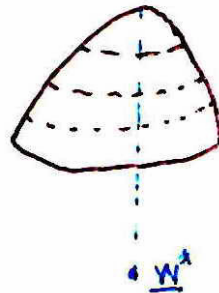
CUMEEIRA



H definida negativa

$$\Delta F < 0$$

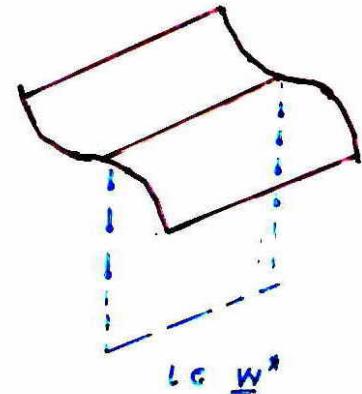
MÁXIMO



H não definida

$$\Delta F \geq 0 \text{ ou } \Delta F < 0$$

SELA



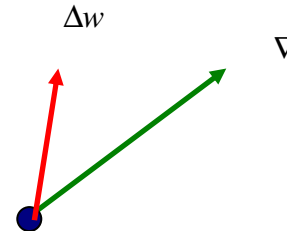
IV – Propriedades do Gradiente

1) Para $|\Delta \underline{w}| < \varepsilon$ e constante o gradiente indica a direção de máximo ΔF

vale a aproximação de primeira ordem $|\Delta \underline{w}| \leq \varepsilon$

$$F(\underline{w}_0 + \Delta \underline{w}) = F(\underline{w}_0) + \Delta F \cong F(\underline{w}_0) + \Delta \underline{w}^t \underline{\nabla} F$$

$$\Delta F \cong \Delta \underline{w}^t \underline{\nabla} F = |\Delta \underline{w}| |\underline{\nabla} F| \cos \theta$$

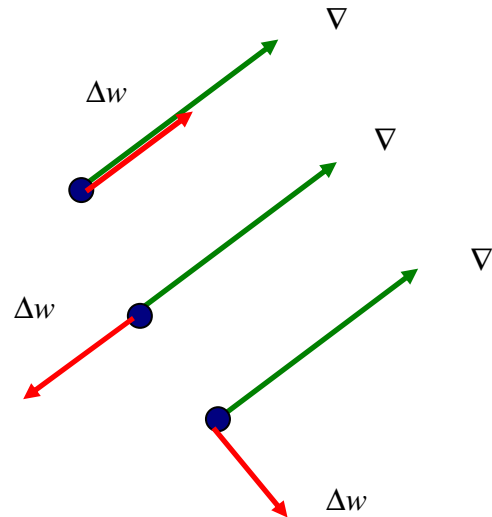


$$\Delta F \cong \underline{\Delta \mathbf{w}}^t \nabla F = |\underline{\Delta \mathbf{w}}| |\nabla F| \cos \theta$$

$$\theta = 0 \quad \longrightarrow \quad \Delta F \text{ max}$$

$$\theta = \pi \quad \longrightarrow \quad \Delta F \text{ min} \\ (-\Delta F \text{ max})$$

$$\theta = \pm \frac{\pi}{2} \quad \longrightarrow \quad \Delta F = 0$$



2) O gradiente é ortogonal ao plano tangente à superfície de nível no ponto

$$\underline{\Delta w} \in \pi_{tg} \quad \underline{\Delta w} = \alpha \underline{t}$$

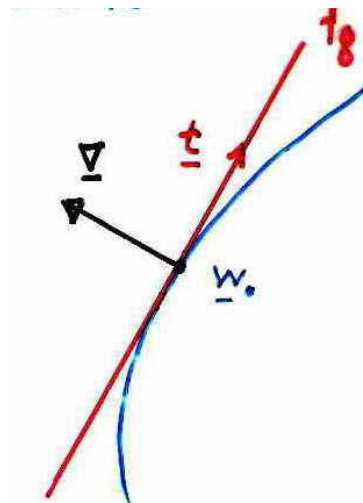
$$\left. \frac{d}{d\alpha} F(\underline{w}_0 + \alpha \underline{t}) \right|_{\forall \underline{t} \in \pi_{\epsilon}} = 0$$

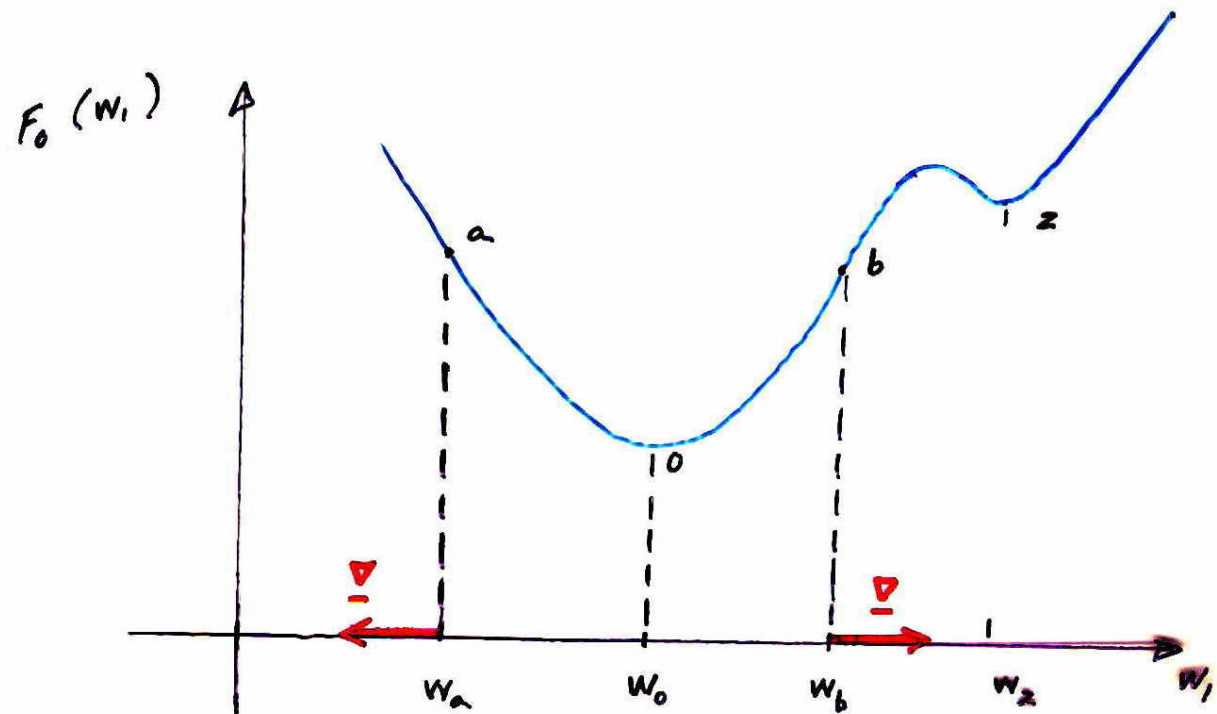
$$\Delta F \cong \underline{\Delta w}^t \nabla F$$

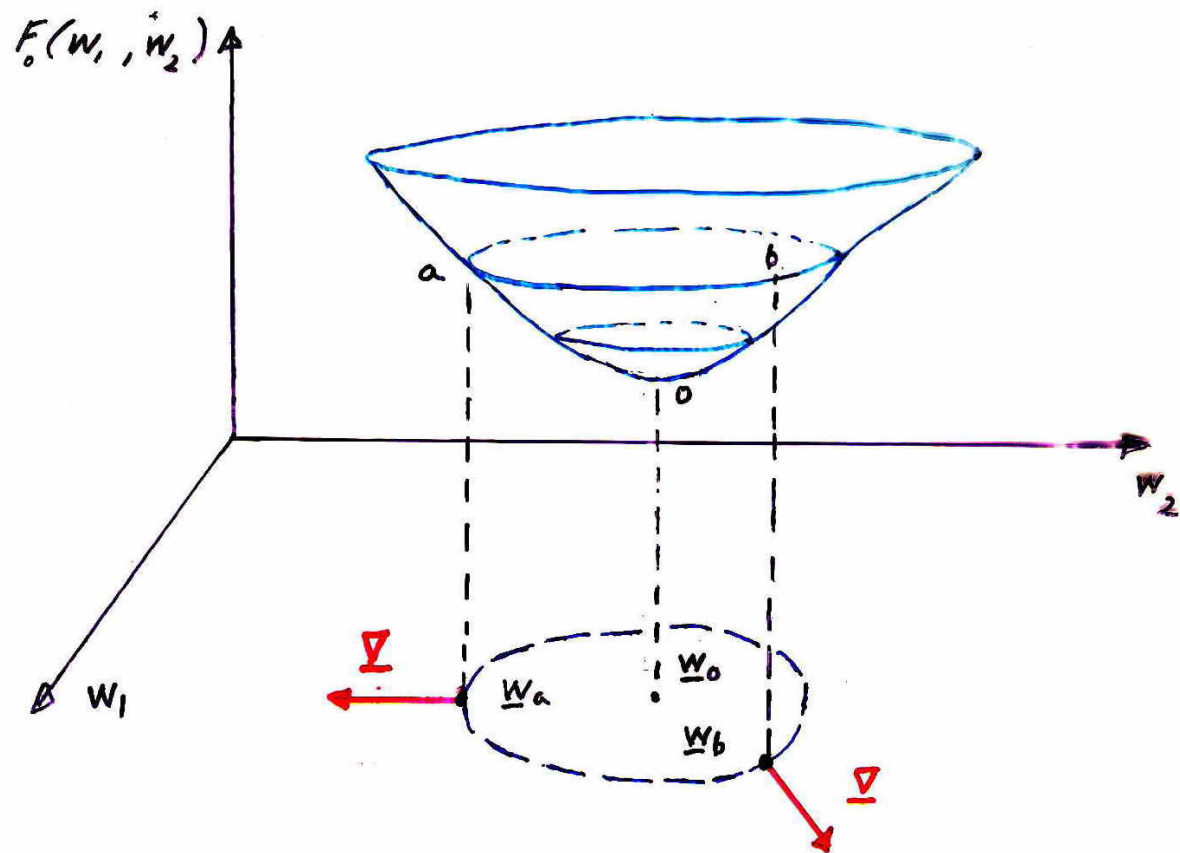
$$\Delta F = 0 \rightarrow \underline{\Delta w} \perp \nabla F$$

$$\underline{\Delta w} \in \pi_{\perp \nabla}$$

$$\pi_t \equiv \pi_{\perp \nabla}$$







V. Otimização

$$\underline{w}^* \mid F(\underline{w}^*) \text{ é mínima}$$

VI – Otimização: Solução Analítica

$$\underline{w}^* \mid \nabla F(\underline{w}^*) = \underline{0} \text{ e}$$
$$\underline{H}(\underline{w}^*) \text{ definida positiva}$$

A - Funções Lineares

$$F(\underline{w}) = \underline{a}^t \underline{w} + b \quad \text{hiper-plano}$$

$$\underline{\nabla} F = \underline{a} = \text{constante}$$

$$\underline{\nabla} F = 0 \quad ? \quad \text{solução} \quad \underline{w}^* \longrightarrow \infty$$

B – Funções Quadráticas

$$F(\underline{w}) = \underline{w}^t \underline{A} \underline{w} + \underline{b}^t \underline{w} + c$$

$$F(\underline{w}_0 + \underline{\Delta w}) = F(\underline{w}_0) + \underline{\Delta w}^t \underline{\nabla}(\underline{w}_0) + \frac{1}{2} \underline{\Delta w}^t \underline{H}(\underline{w}_0) \underline{\Delta w}$$

$$\underline{\nabla}(\underline{w}_0 + \underline{\Delta w}) = \underline{\nabla}(\underline{w}_0) + \underline{H}(\underline{w}_0) \underline{\Delta w}$$

$$\underline{\nabla} = \underline{0} \quad \Rightarrow \quad \underline{\Delta w} = - \underline{H}^{-1}(\underline{w}_0) \underline{\nabla}(\underline{w}_0)$$

$$\underline{w}^* = \underline{w}_0 - \underline{H}^{-1}(\underline{w}_0) \underline{\nabla}(\underline{w}_0)$$

Newton

Newton Raphson

Problema: H, e principalmente H⁻¹

Evitando $\underline{H}$

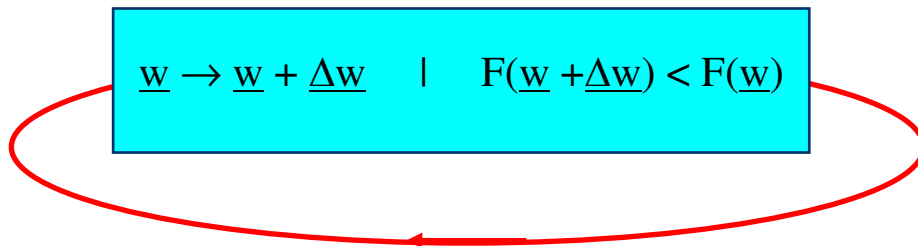
Pseudo Newton $\underline{H} \approx \text{diag } \underline{H} \implies \underline{H}^{-1}$

Quase Newton $\sim \underline{H}^{-1}$

C – Funções de ordens mais elevadas e/ou transcendentais

muito complexo $\implies$ Métodos Numéricos Recursivos

VII – Otimização: Métodos Numéricos Recursivos



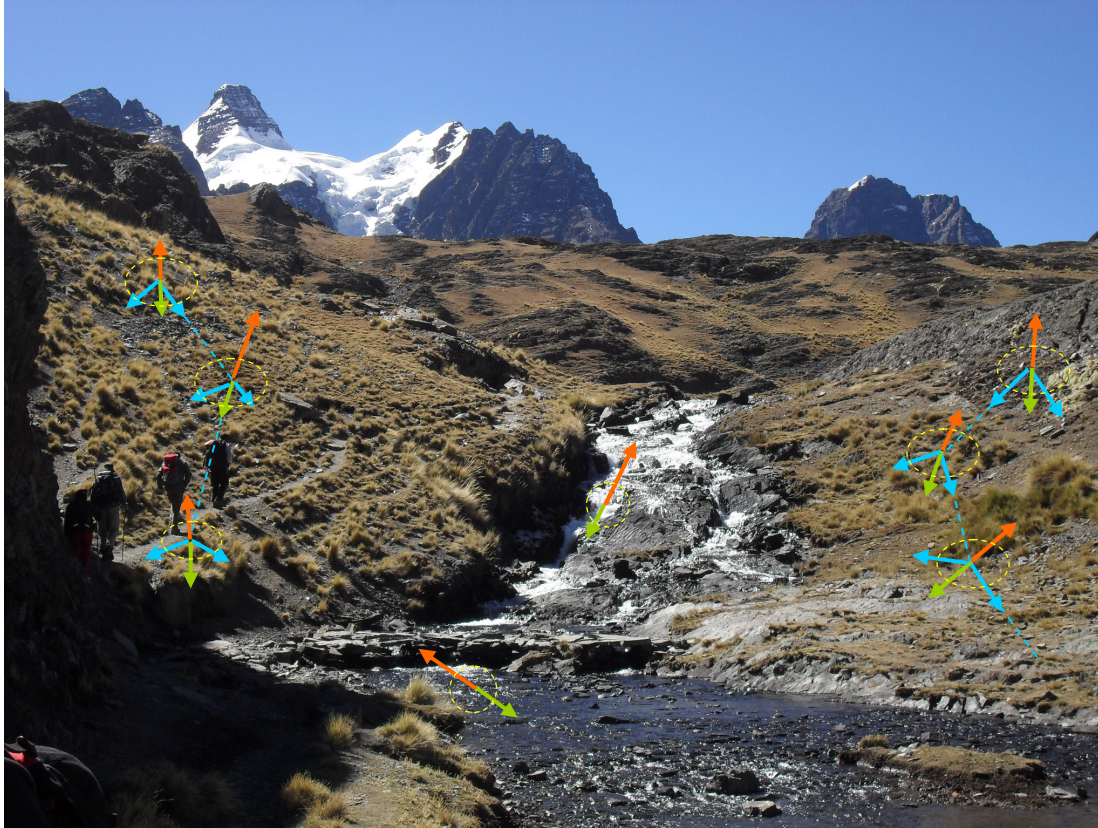
Δw ??

$$\underline{\Delta w} = \alpha \underline{d}$$

Passo
controla $|\underline{\Delta w}|$
 $\alpha > 0$, pequeno

controla
direção Δw

Descida continuada, permanente - **decisão local**



Condições necessárias

α pequeno, vale a aproximação de primeira ordem

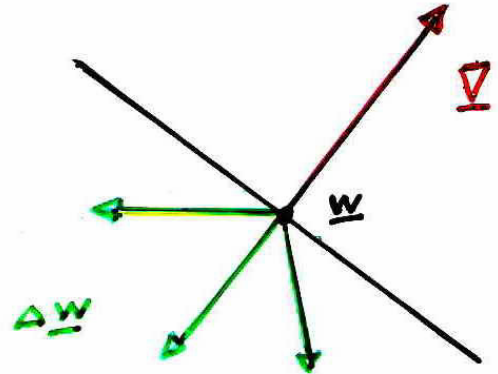
$$F(\underline{w} + \Delta \underline{w}) \approx F(\underline{w}) + \Delta \underline{w}^t \nabla(\underline{w})$$

$$\Delta F \approx \Delta \underline{w}^t \nabla(\underline{w})$$

$$= \alpha \|\underline{d}\| \|\nabla(\underline{w})\| \cos \angle \underline{d}, \nabla$$

direção $\underline{d}$

$$\angle \underline{d}, \nabla \in (90^\circ, 270^\circ)$$



Direção mais eficaz ?

Para $|\Delta \underline{w}|$ constante

qual $\underline{d}$ para máximo $-\Delta F$?

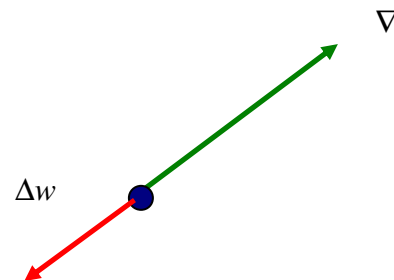
$$F(\underline{w} + \Delta \underline{w}) = F(\underline{w}) + \Delta \underline{w}^t \nabla$$

$$\Delta F = \Delta \underline{w}^t \nabla = |\Delta \underline{w}| |\nabla| \cos \angle \underline{d}, \underline{\nabla}$$

constant $(-1, 1)$

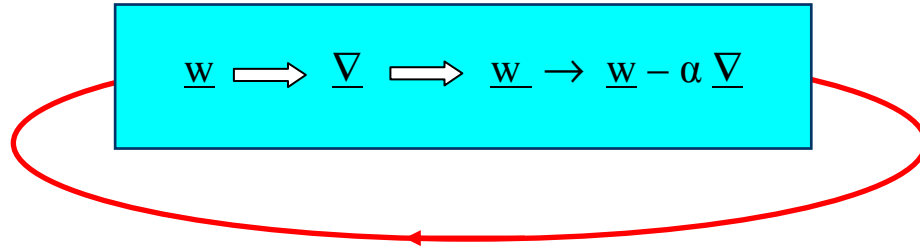
$$\underline{d} = - \underline{\nabla}$$

$$\Delta \underline{w} = - \alpha \underline{\nabla} \Rightarrow - \Delta F \text{ max}$$



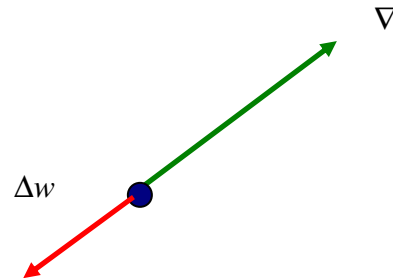
Método do Gradiente Descendente ou

Método da Descida pelo Gradiente

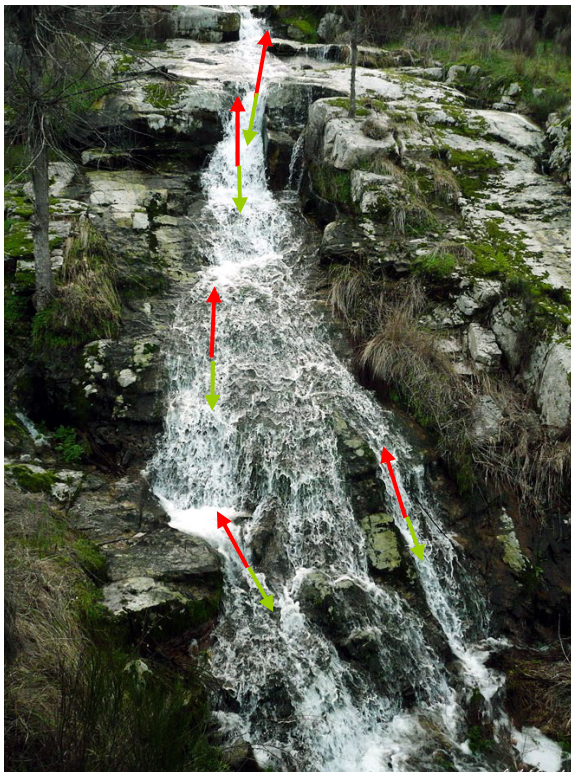


$$\Delta \underline{w} = -\alpha \underline{\nabla}$$

$$\Delta F = \Delta \underline{w}^t \underline{\nabla} = -\alpha |\underline{\nabla}|^2$$



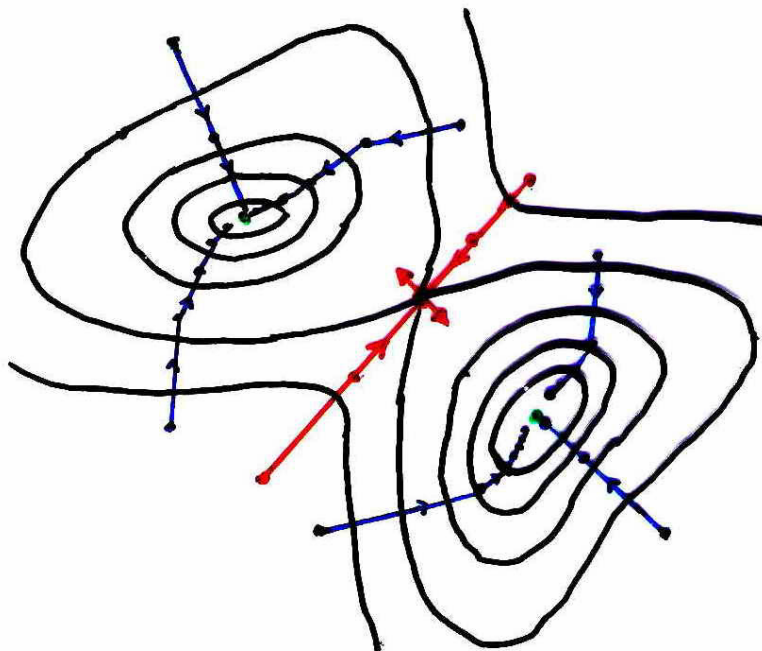
Gradiente Descendente



Método do

Gradiente Descendente

$$\underline{w} \rightarrow \underline{\nabla} \rightarrow \underline{w} \rightarrow \underline{w} - \alpha \underline{\nabla}$$



Algoritmo

Até que o critério de parada seja satisfeito

Para cada variável w_i

Calcular $\frac{\partial F}{\partial w_i}$

Fazer $w_{i\text{ novo}} = w_{i\text{ antigo}} - \alpha \frac{\partial F}{\partial w_i}$

Fim (natural) do processo

$$\underline{w} = \underline{w}_0$$

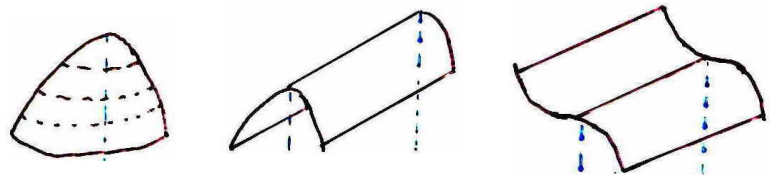
$$E(\Delta \underline{w}) = \underline{0} \quad \Rightarrow \quad \nabla(\underline{w}_0) = \underline{0}$$

O que é $\underline{w}_0$??

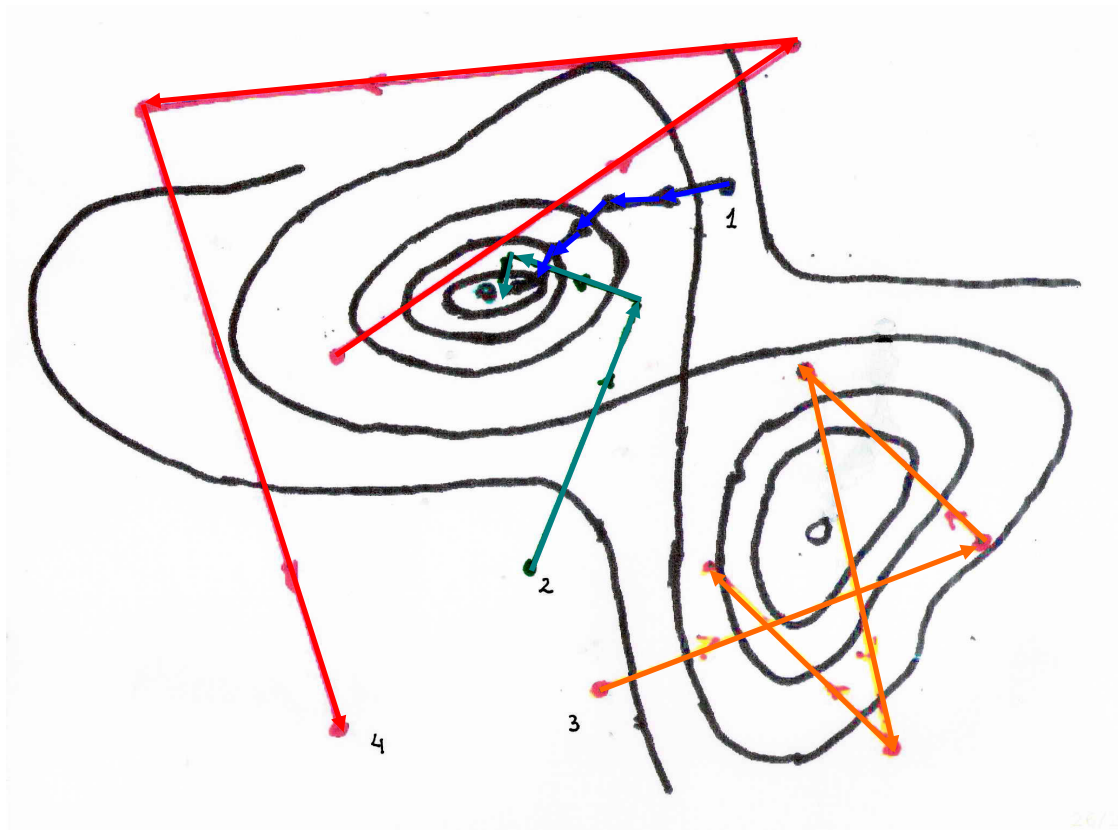
pontos estáveis: mínimo, calha



pontos instáveis: máximo, cumeeira, sela

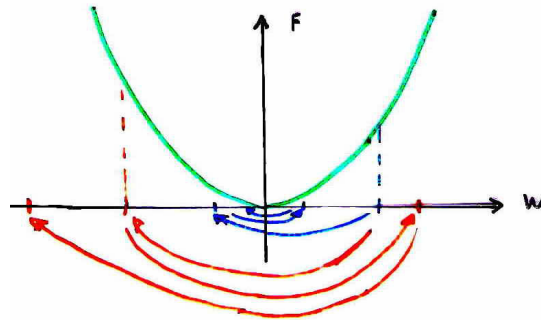
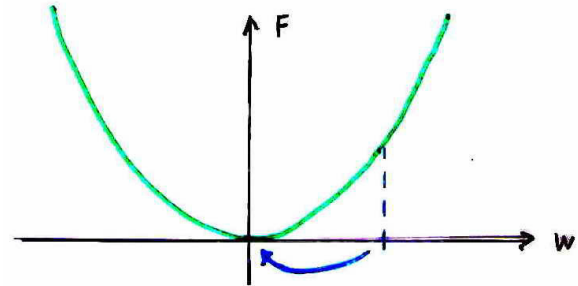
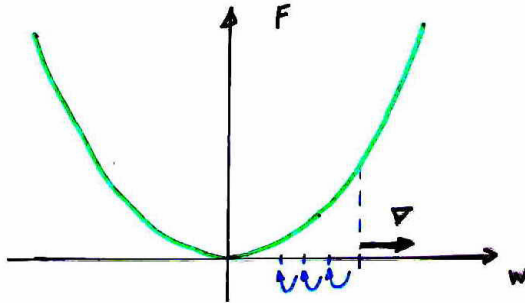


Passo de Treinamento α

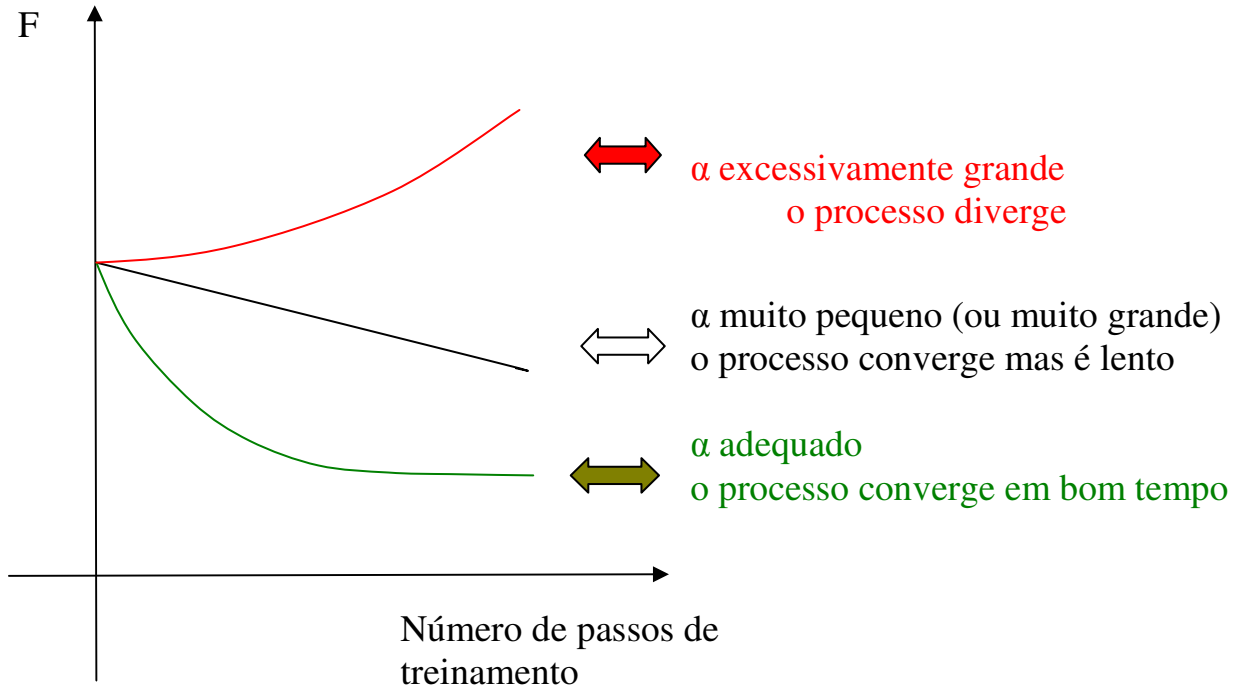


Efeito do passo de treinamento α

Exemplo simples: $F = w^2$



Evolução do erro ao longo do treinamento:



Passo de Treinamento α

$\alpha \ll$ convergência lenta

$\alpha \gg$ F_0 oscila muito, convergência lenta, **pode divergir**

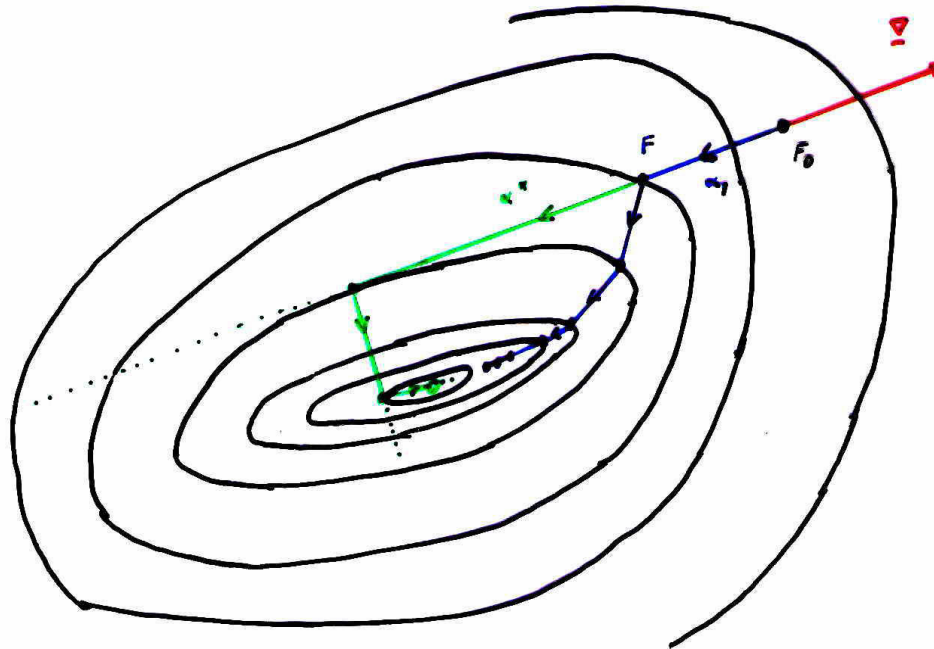
$\Rightarrow \alpha_{\text{ótimo}} ??$

“a escolha de α é uma arte” – Bernard Widrow

No treinamento de redes neurais

$0 < \alpha \leq 1$ $\alpha_{\text{tipico}} = .05, .1$

*Passo Ótimo de Treinamento α^**



Que $\alpha = \alpha^*$ maximiza $-\Delta F$?? **Otimização em linha.**

Passo ótimo α^*

$$\Delta \underline{\mathbf{w}} = -\alpha \underline{\nabla}$$

$$F(\underline{\mathbf{w}}_0 + \Delta \underline{\mathbf{w}}) \approx F(\underline{\mathbf{w}}_0) + \underline{\nabla}^t \Delta \underline{\mathbf{w}} + \frac{1}{2} \Delta \underline{\mathbf{w}}^t \underline{\mathbf{H}} \Delta \underline{\mathbf{w}}$$

$$F(\underline{\mathbf{w}}_0 - \alpha \underline{\nabla}) \approx F(\underline{\mathbf{w}}_0) - \alpha \underline{\nabla}^t \underline{\nabla} + \frac{1}{2} \alpha^2 \underline{\nabla}^t \underline{\mathbf{H}} \underline{\nabla} \quad (1)$$

$$\Delta F \approx -\alpha \underline{\nabla}^t \underline{\nabla} + \frac{1}{2} \alpha^2 \underline{\nabla}^t \underline{\mathbf{H}} \underline{\nabla}$$

Que $\alpha = \alpha^*$ maximiza $-\Delta F$?? Otimização em linha.

$$\alpha = \alpha^* \Rightarrow -\Delta \mathbf{F} \text{ é máximo} \quad \frac{\partial F}{\partial \alpha} = \frac{\partial \Delta F}{\partial \alpha} = 0$$

$$\alpha^* = \underset{\alpha^* > 0}{\text{Arg}} [\text{Min}_{\alpha} F(\underline{\mathbf{w}}_0 - \alpha^* \underline{\nabla})]$$

$$\frac{\partial}{\partial \alpha} F = -|\underline{\nabla}|^2 + \alpha \underline{\nabla}^t \underline{\mathbf{H}} \underline{\nabla} = 0$$

$$\alpha^* = \frac{|\underline{\nabla}|^2}{\underline{\nabla}^t \underline{\mathbf{H}} \underline{\nabla}} \quad (2)$$

necessita de H !

Contornando o problema de H:

$$\text{Aplicar } \alpha = \alpha_1 \quad \underline{w}_0 \rightarrow \underline{w}_0 - \alpha_1 \underline{\nabla} \quad F_0 \rightarrow F_1$$

$$\text{De (1)} \quad F_1 = F_0 - \alpha_1 \underline{\nabla}^t \underline{\nabla} + \frac{1}{2} \alpha_1^2 \underline{\nabla}^t \underline{H} \underline{\nabla}$$

$$\therefore \quad \underline{\nabla}^t \underline{H} \underline{\nabla} = \frac{2}{\alpha_1^2} (F_1 - F_0 + \alpha_1 |\underline{\nabla}|^2) \quad (3)$$

aplicando (3) em (2)

$$\alpha^* = \frac{|\underline{\nabla}|^2}{\underline{\nabla}^t \underline{H} \underline{\nabla}} \quad (2)$$

$$\alpha^* = \frac{\alpha_1^2 |\underline{\nabla}|^2}{2(F_1 - F_0 + \alpha_1 |\underline{\nabla}|^2)} \quad \text{e}$$

$$\Delta F = \frac{1}{2} \alpha^* |\underline{\nabla}|^2$$

Obs:

1 - α^* leva a extremos. Se $\alpha^* < 0$ o extremo é um máximo.

2 - o passo de ensaio, $-\alpha_1 \underline{\nabla}$, levanta as características da função no entorno de $\underline{w}_0$ com raio $|\alpha_1 \underline{\nabla}|$. Se $\alpha^* \gg \alpha_1$ possivelmente o α^* encontrado não é válido.

Passo variável controlando o decréscimo de F

$$\Delta F \approx - \Delta \underline{w}^t \underline{\nabla} = - \alpha \|\underline{\nabla}\|^2$$

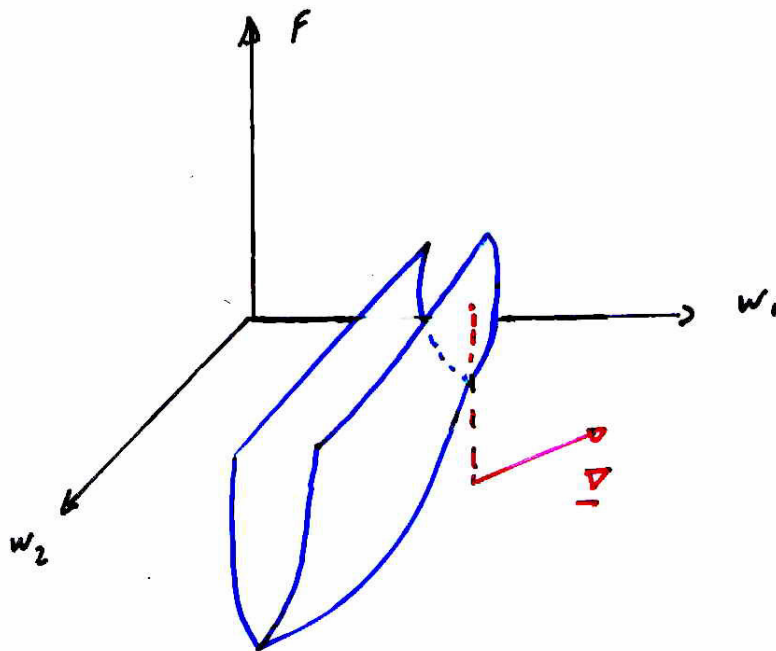
$$\alpha = \alpha_0 \quad \text{ctte} \quad \Rightarrow \quad \Delta F = - \alpha \|\underline{\nabla}\|^2$$

$$\alpha = \alpha_0 \frac{1}{\|\underline{\nabla}\|^2} \quad \Rightarrow \quad \Delta F = - \alpha_0 \quad \text{ctte}$$

$$\alpha = \alpha_0 \frac{F}{\|\underline{\nabla}\|^2} \quad \Rightarrow \quad \frac{\Delta F}{F} = - \alpha_0 \quad \text{ctte}$$

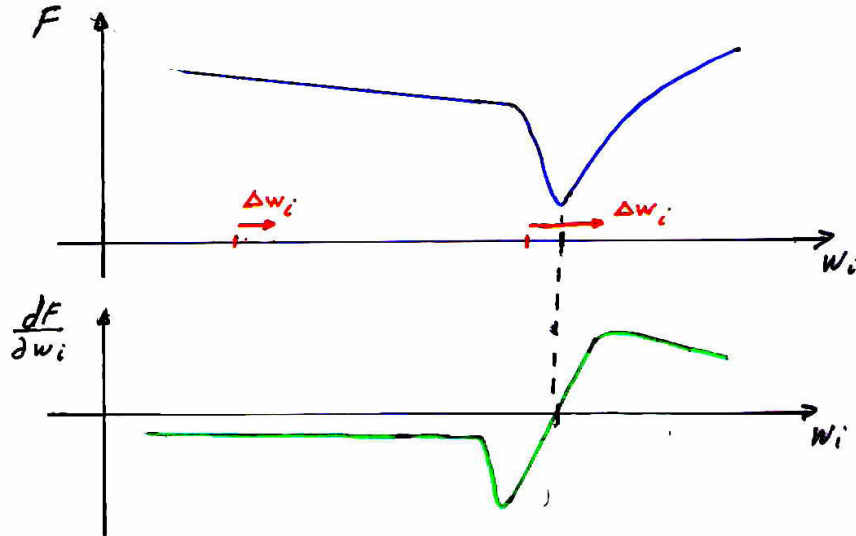
Acelerando o processo – passo variável

α diferenciado por variável



Acelerando o processo – passo variável

α diferenciado por ponto



$$\Delta w_i = -\alpha \frac{\partial F}{\partial w_i}$$

Resilient Backpropagation (modificado)

(Silva e Almeida, Super SAB, etc.)

$$\alpha \text{ variável} \left\{ \begin{array}{l} \text{por variável } w_i \\ \text{por passo de treinamento } n \end{array} \right.$$

$$\Delta w_i(n) = - \alpha_i(n) \frac{\partial F}{\partial w_i}(n)$$

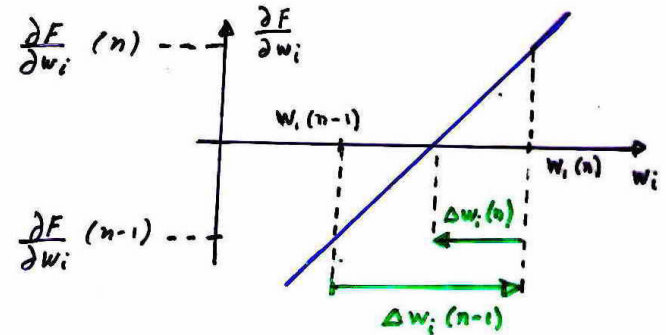
$$\alpha_i(n) = \left\{ \begin{array}{ll} a \alpha_i(n-1) & \text{se } \text{sign} \frac{\partial F}{\partial w_i}(n) = \text{sign} \frac{\partial F}{\partial w_i}(n-1) \\ b \alpha_i(n-1) & \text{se } \text{sign} \frac{\partial F}{\partial w_i}(n) \neq \text{sign} \frac{\partial F}{\partial w_i}(n-1) \end{array} \right.$$

se $\alpha_i(n) > \alpha^*$ faça $\alpha = \alpha^*$

$$\alpha_i(0) \approx .05 \quad \alpha^* \approx .2 \quad a \approx 1.05 \quad b \approx .9$$

Outra alternativa

$$b = \frac{\frac{\partial F}{\partial w_i}(n-1)}{\frac{\partial F}{\partial w_i}(n-1) - \frac{\partial F}{\partial w_i}(n)}$$

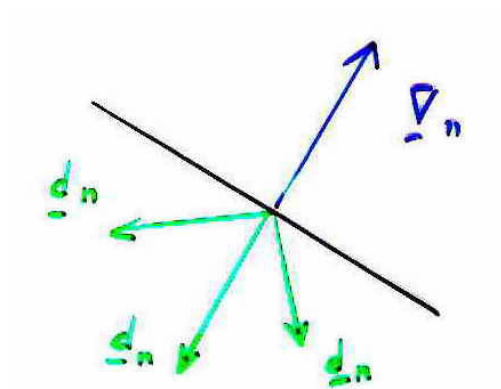


Métodos de 1ª. Ordem (Resumo)

$$\underline{w}_{n+1} = \underline{w}_n + \alpha_n \underline{d}_n$$

Para $F(\underline{w}_{n+1}) < F(\underline{w}_n)$

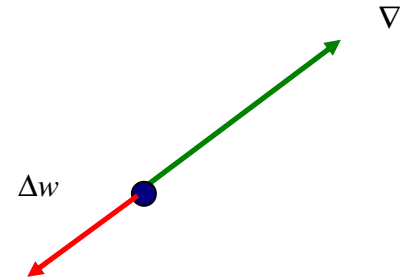
$$\underline{d}_n^t \underline{\nabla}_n < 0$$



Gradiente Descendente

$$\underline{w}_{n+1} = \underline{w}_n + \alpha_n \underline{d}_n$$

$$\underline{d}_n = - \underline{\nabla}_n \quad \alpha_n = \alpha_0$$



Passo ótimo

$$\alpha_n = \underset{\alpha_n > 0}{\text{Arg}} [\text{Min } F(\underline{w}_n + \alpha_n \underline{d})]$$

BP Resiliente: α variável por sinapse e por passo

MÉTODOS DE SEGUNDA ORDEM:
MAIS SOFISTICADOS / MAIS EFICIENTES (?)

Newton $\underline{\mathbf{H}}$

Newton Amortecido $\approx \underline{\mathbf{H}}$

Quasi Newton $\approx \underline{\mathbf{H}}^{-1}$

Gradiente Conjugado

Newton

$$\underline{w}^* = \underline{w}_1 - \underline{H}_1^{-1} \underline{\nabla}_1$$

$$\underline{w}_{n+1} = \underline{w}_n + \alpha_n \underline{d}_n$$

$$\underline{d}_n = - \underline{H}_n^{-1} \underline{\nabla}_n$$

$$\alpha_n = \underset{\alpha_n > 0}{\text{Arg}} [\text{Min } F(\underline{w}_n + \alpha_n \underline{d})]$$

$$\underline{H}_n^{-1} \text{ existe ? } \acute{\text{e}} \text{ d.p. ?}$$

Newton Amortecido

Marquadt – Levenberg

para garantir a inversão de $\underline{H}$



$$\underline{H}_n \rightarrow \underline{H}_n + \lambda \underline{I}$$

$$\lambda > 0$$

$$\lambda \rightarrow 0$$

Levenberg – Marquadt

$$F(\underline{w}) = \frac{1}{2} \sum_k \varepsilon_k^2$$

$$\underline{H}_n = [h_{ij}]$$


$$h_{ij} = \frac{\partial^2 F}{\partial w_i \partial w_j} \approx \sum_k \frac{\partial \varepsilon_k}{\partial w_i} \cdot \frac{\partial \varepsilon_k}{\partial w_j}$$


$$\underline{H}_n \rightarrow \underline{H}_n + \lambda \underline{I}$$

$$\lambda > 0 \quad \lambda \rightarrow 0$$

$$\underline{w}_{n+1} = \underline{w}_n + \alpha_n \underline{d}_n$$

$$\underline{d}_n = - \left[\left[h_{ij} \right] + \lambda \underline{I} \right]^{-1} \underline{\nabla}_n$$

$$\alpha_n = \underset{\alpha_n > 0}{\text{Arg}} \left[\text{Min } F(\underline{w}_n + \alpha_n \underline{d}) \right]$$

Quasi Newton

Fórmulas básicas:

$$\underline{\nabla}_{n+1} = \underline{\nabla}_n + \underline{H}_n (\underline{w}_{n+1} - \underline{w}_n)$$

$$\text{onde } \underline{w}_{n+1} - \underline{w}_n = \Delta \underline{w}$$

$$\underline{w}_{n+1} - \underline{w}_n = \underline{G}_n (\underline{\nabla}_{n+1} - \underline{\nabla}_n)$$

$$\text{onde } \underline{G}_n \approx \underline{H}_n^{-1}$$

BFGS – Broyden – Fletcher – Goldfarb – Shanno

$$\underline{w}_{n+1} = \underline{w}_n + \alpha_n \underline{d}_n$$

$$\underline{d}_n \approx \underline{w}_{n+1} - \underline{w}_n = - \underline{G}_n \underline{\nabla}_n$$

$$\alpha_n = \underset{\alpha_n > 0}{\text{Arg}} [\text{Min } F(\underline{w}_n + \alpha_n \underline{d})]$$

$$\underline{y} = \underline{\nabla}_{n+1} - \underline{\nabla}_n$$

$$\underline{G}_{n+1} = \left[\underline{I} - \frac{\underline{d}_n \underline{y}_n^t}{\underline{d}^t \underline{y}_n} \right] \underline{G}_n \left[\underline{I} - \frac{\underline{y}_n \underline{d}_n^t}{\underline{d}_n^t \underline{y}_n} \right] + \frac{\underline{d}_n \underline{d}_n^t}{\underline{d}_n^t \underline{y}_n}$$

$$\underline{G}_n \approx \underline{H}_n^{-1}$$

Gradiente Conjugado - não usa H

Fletcher – Reeves

$\underline{d}_i$ – uma só vez

$$\underline{w}_{n+1} = \underline{w}_n + \alpha_n \underline{d}_n$$

$$\underline{d}_n = \beta_n \underline{d}_{n-1} - \underline{\nabla}_n$$

$$\alpha_n = \underset{\alpha_n > 0}{\text{Arg}} [\text{Min}_{\alpha_n > 0} F(\underline{w}_n + \alpha_n \underline{d})]$$

Leonard & Kramer

$$\beta_n = \frac{|\underline{\nabla}_n|^2}{|\underline{\nabla}_{n-1}|^2}$$

Polak – Ribière

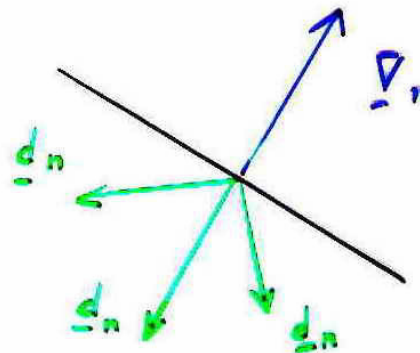
$$\beta_n = \frac{(\underline{\nabla}_n - \underline{\nabla}_{n-1})^t \underline{\nabla}_n}{|\underline{\nabla}_{n-1}|^2}$$

Métodos de 2ª. Ordem (Resumo)

$$\underline{w}_{n+1} = \underline{w}_n + \alpha_n \underline{d}_n$$

$$\alpha_n = \underset{\alpha_n > 0}{\text{Arg}} [\text{Min } F(\underline{w}_n + \alpha_n \underline{d})]$$

$\underline{d}_n$???



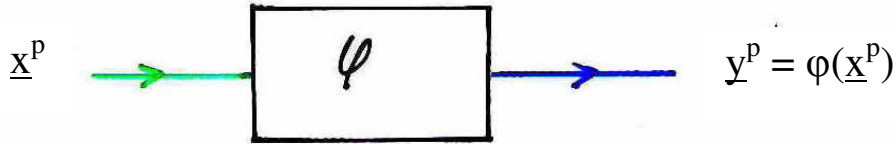
Parte II

Aprendizado

(ou treinamento, ou ajuste de parâmetros)

como um processo de otimização

Sistema à modelar / simular



Conhecimento do Sistema

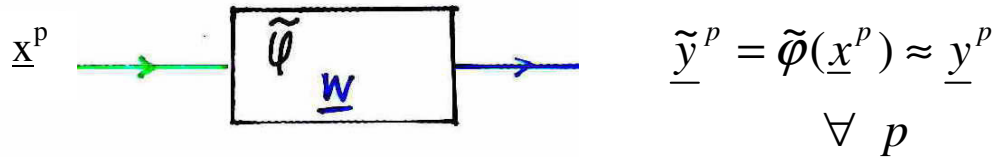
Fenomenológico – equações – caixa branca

Funcional – relação entrada/saída – caixa preta

Pares entrada-saída

$$(\underline{x}^p, \underline{y}^p) \quad p = 1, \dots, P$$

Modelo Matemático Fenomenológico / Numérico - Simulador

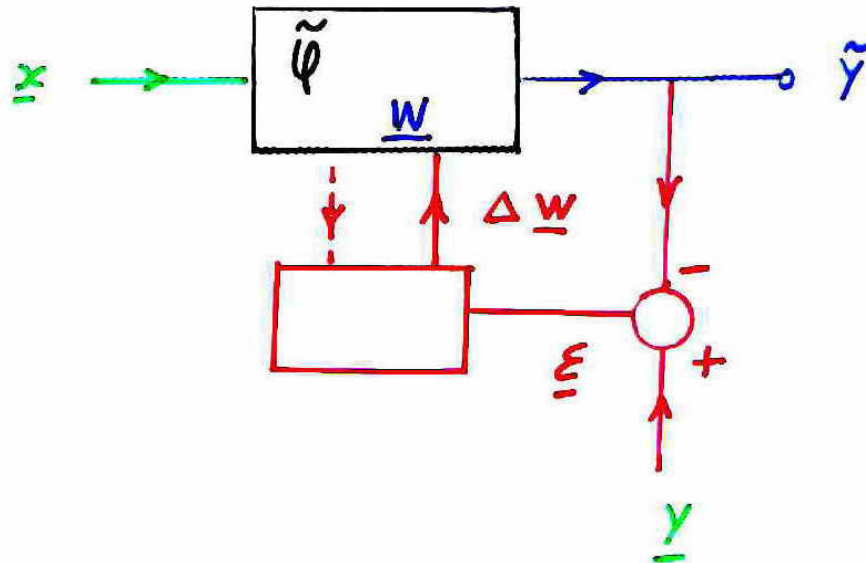


Modelo Fenomenológico – Caixa branca

Modelo Numérico Geral (e.g. rede neural) – Caixa Preta

Ajuste dos parâmetros $\underline{w}$?

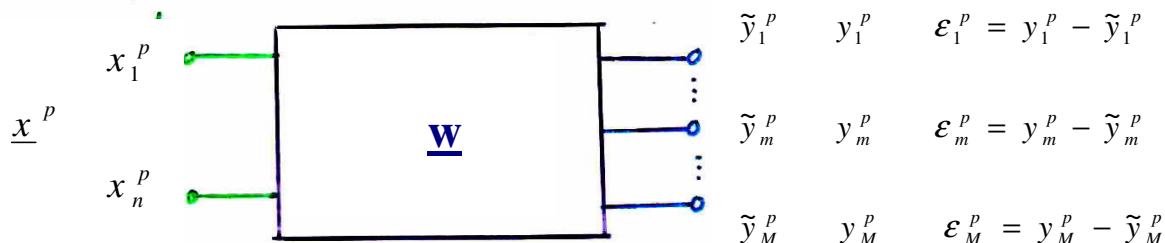
Treinamento, Aprendizado ou Ajuste dos Parâmetros



como um processo de otimização

Erro na saída  **Função objetivo à minimizar**

A saída é multidimensional:



Erro quadrático na saída para o k-ésimo par entrada-saída k

$$(\epsilon^p)^2 = \left| \underline{y}^p - \underline{\tilde{y}}^p \right|^2 = \sum_{m=1}^M \left(y_m^p - \tilde{y}_m^p \right)^2 = \sum_{m=1}^M \left(\epsilon_m^p \right)^2$$

onde

$$\epsilon_m^p = y_m^p - \tilde{y}_m^p \quad e \quad \tilde{y}_m^p = \tilde{y}_m^p(\underline{w})$$

Erro quadrático médio para todos os P pares entrada-saída

$$(\underline{x}^p, y^p) \quad p = 1, \dots, P$$

Erro médio quadrático:

$$F(\underline{w}) = E_p(\varepsilon^p)^2 = \frac{1}{P} \sum_{p=1}^P (\varepsilon^p)^2$$

Como calcular

$$\frac{\partial}{\partial w_i} E_p(\varepsilon^p)^2 \quad ???$$

$$\frac{\partial}{\partial w_i} E(\epsilon^p)^2 = \frac{\partial}{\partial w_i} \frac{1}{P} \sum_{p=1}^P (\epsilon^p)^2 = \frac{1}{P} \sum_{k=1}^P \frac{\partial}{\partial w_i} (\epsilon^p)^2 = E \frac{\partial}{\partial w_i} (\epsilon^p)^2$$

$$\underline{\nabla}_p E(\epsilon^p)^2 = E \underline{\nabla}_p (\epsilon^p)^2$$

Propriedade importante: o gradiente do valor esperado do erro quadrático é igual ao valor esperado do gradiente do erro quadrático

Consequência:

Calcularemos separadamente para cada par

$$\frac{\partial}{\partial w_i} (\varepsilon^p)^2 \quad \forall p$$

e tomaremos a média

$$\frac{\partial}{\partial w_i} E_p (\varepsilon^p)^2 = E_p \frac{\partial}{\partial w_i} (\varepsilon^p)^2$$

Continuando

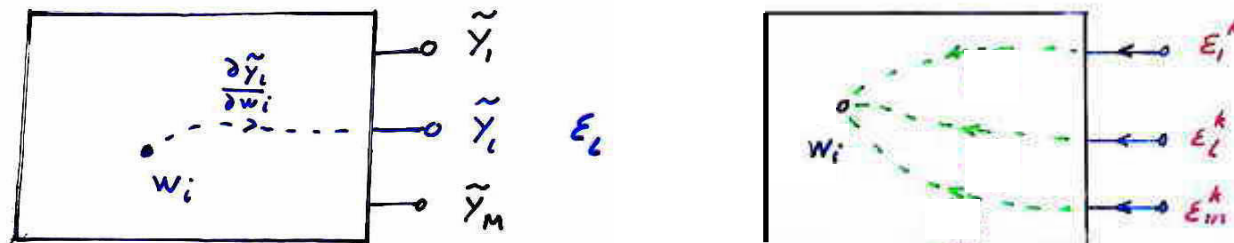
$$\begin{aligned}\frac{\partial}{\partial w_i} (\epsilon^p)^2 &= \frac{\partial}{\partial w_i} \sum_{m=1}^M (\epsilon_m^p)^2 = \sum_{m=1}^M \frac{\partial}{\partial w_i} (\epsilon_m^p)^2 = \\ &= 2 \sum_{m=1}^M \epsilon_m^p \frac{\partial}{\partial w_i} \epsilon_m^p \quad \text{onde} \quad \epsilon_m^p = y_m^p - \tilde{y}_m^p \\ &= -2 \sum_{m=1}^M \epsilon_m^p \frac{\partial \tilde{y}_m^p}{\partial w_i}\end{aligned}$$

onde o termo $-2\epsilon_m^p$ deriva da **formula do erro** usada

e o termo $\frac{\partial y_m^p}{\partial w_i}$ depende das **equações do modelo** usado

Para um único par entrada-saída o acréscimo a ser aplicado em w_i seria

$$\Delta w_i^p = -\alpha \frac{\partial}{\partial w_i} E(\varepsilon^p)^2 = -\alpha \frac{\partial}{\partial w_i} (\varepsilon)^2 = 2\alpha \sum_{m=1}^M \varepsilon_m \frac{\partial \tilde{y}_m}{\partial w_i}$$



$$\Delta w_i^p = 2\alpha \delta_i^p \quad \text{onde} \quad \delta_i^p = \sum_{m=1}^M \varepsilon_m \frac{\partial \tilde{y}_m}{\partial w_i} \quad - \text{ erro retropropagado}$$

error backpropagation

Para os P pares:

$$\Delta w_i = \alpha \frac{\partial}{\partial w_i} E(\varepsilon^k)^2 = -2\alpha \frac{1}{P} \sum_{k=1}^P \sum_{l=1}^m \varepsilon_l^k \frac{\partial \tilde{y}_l^k}{\partial w_i}$$

Formas de Aplicação

Processo em Batelada – Batch

Batelada - Algoritmo

Até que o critério de parada seja satisfeito

para cada par $(\underline{x}^p, \underline{y}^p)$ $p = 1, \dots, P$

calcular $\tilde{\underline{y}}^p = \phi(\underline{x}^p)$

$$\varepsilon_m^p = y_m^p - \tilde{y}_m^p \quad \text{e} \quad \frac{\partial \tilde{y}_m^p}{\partial w_i} \quad \forall 1, i$$

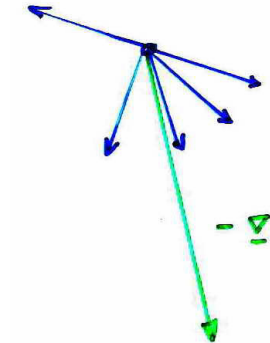
$$\frac{\partial \varepsilon^{p2}}{\partial w_i} = -2 \sum_{m=1}^M \varepsilon_m^p \frac{\partial y_m^p}{\partial w_i} \quad \forall i$$

outro par

calcular $\Delta w_i = -\alpha \sum_p \frac{\partial \varepsilon^{p2}}{\partial w_i}$

$$w_i(n+1) = w_i(n) + \Delta w_i$$

reiniciar



Se $P \gg (P > \sim 200)$

$\Rightarrow$ **molto lento**

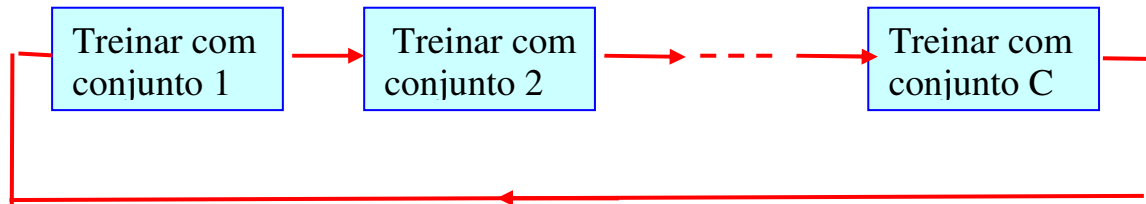
Processo por Lotes

Se $P > \sim 200$

Alocar os pares entrada-saída aleatoriamente em C conjuntos,

$$C \approx P / 200$$

Algoritmo



Lotes - Algoritmo

Até que o critério de parada seja satisfeito

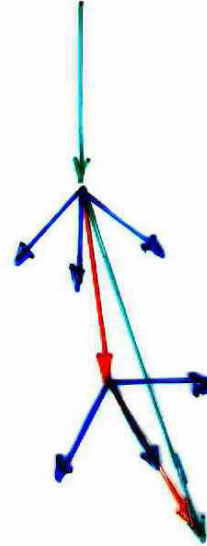
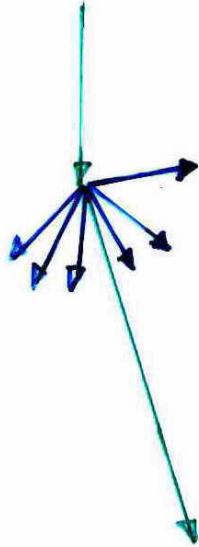
Treinar (em batelada) usando os pares do conjunto 1

Treinar (em batelada) usando os pares do conjunto 2

...

Treinar (em batelada) usando os pares do conjunto C

retornar



Reduz o tempo

Preserva a generalização mas

Diminui (pouco) a estabilidade do treinamento

Processo em “Regra Delta” -

uso “on line”

Atualiza os parâmetros a cada par entrada-saída apresentado

Regra Delta - Algoritmo

Até que o critério de parada seja satisfeito

para cada par $(\underline{x}^p, \underline{y}^p)$ $p = 1, \dots, P$

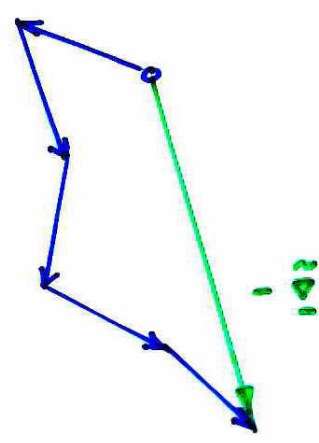
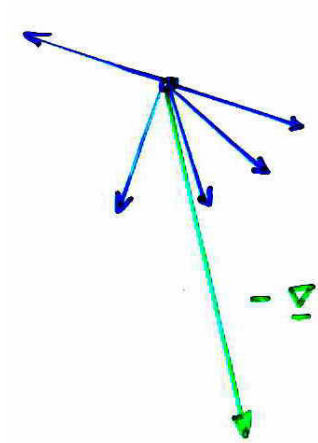
calcular $\tilde{y}^p = \phi(\underline{x}^p)$

$$\varepsilon_m^p = y_m^p - \tilde{y}_m^p \quad \text{e} \quad \frac{\partial \tilde{y}_m^p}{\partial w_i} \quad \forall 1, i$$

$$\frac{\partial \varepsilon^{p2}}{\partial w_i} = -2 \sum_{m=1}^M \varepsilon_m^p \frac{\partial y_m^p}{\partial w_i} \quad \forall i$$

$$w_i(n+1) = w_i(n) - \alpha \frac{\partial \varepsilon^{p2}}{\partial w_i} \quad \forall i$$

outro par



Rápido mas F oscila !

Normalmente usado com momento

Treinamento com Momento

Regra delta:

$$\Delta w_{i_D}(n) = 2 \alpha \sum_{l=1}^m \varepsilon_l \frac{\partial \tilde{y}_l}{\partial w_i}$$

Momento

$$\Delta w_i(n) = \beta \Delta w_i(n-1) + (1-\beta) \Delta w_{i_D}(n)$$

$$\beta_{\text{típico}} \approx .9$$

intermediário entre regra delta e batelada (lote)

$$\text{lote} \quad N_0 \approx \frac{4}{1 - \beta}$$

$$n = 1 \quad \Delta w_D(1)$$

$$\Delta w(1) = \beta \Delta w(0) + (1-\beta) \Delta w_D(1)$$

$$= (1-\beta) \Delta w_D(1)$$

$$n = 2 \quad \Delta w_D(2)$$

$$\Delta w(2) = \beta \Delta w(1) + (1-\beta) \Delta w_D(2) =$$

$$= \beta(1-\beta) \Delta w_D(1) + (1-\beta) \Delta w_D(2)$$

$$n = 3 \quad \Delta w_D(3)$$

$$\Delta w(3) = \beta \Delta w(2) + (1-\beta) \Delta w_D(3) =$$

$$= \beta^2(1-\beta) \Delta w_D(1) + \beta(1-\beta) \Delta w_D(2) + (1-\beta) \Delta w_D(3)$$

$$n = 4 \quad \Delta w_D(4)$$

$$\Delta w(4) = \beta^3(1-\beta) \Delta w_D(1) + \beta^2(1-\beta) \Delta w_D(2) + \beta(1-\beta) \Delta w_D(3) + (1-\beta) \Delta w_D(4)$$

...

$$n \quad \Delta w_D(n)$$

$$\Delta w(n) = \beta^{n-1}(1-\beta) \Delta w_D(1) + \beta^{n-2}(1-\beta) \Delta w_D(2) + \dots$$

$$\dots + \beta^{n-1}(1-\beta) \Delta w_D(i) + \dots + (1-\beta) \Delta w_D(n)$$

$$\Delta w(n) = (1-\beta) \sum_{i=1}^n \Delta w_D(i) \beta^{n-i}$$

Contribuições em cada passo:

Média ponderada exponencialmente pelo atraso

$n - i$ = atraso da i -ésima entrada em relação ao instante atual n

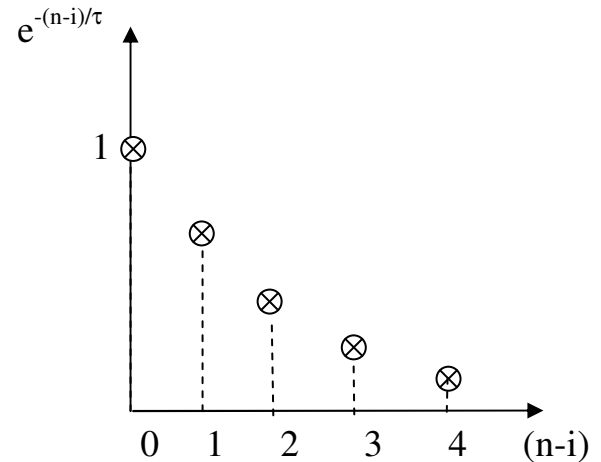
No passo n

$$\Delta w(n) = (1-\beta) \sum_{i=1}^n \Delta w_D(i) \beta^{n-i}$$

$$\beta^{n-i} = e^{(n-i) \ln \beta} = e^{-\frac{(n-i)}{\tau}}$$

$$\tau = -\frac{1}{\ln \beta} \approx \frac{1}{1-\beta} \quad \text{se } \beta \approx 1$$

$$\Delta w(n) = (1-\beta) \sum_{i=1}^n \Delta w_D(i) e^{-(n-i)/\tau}$$

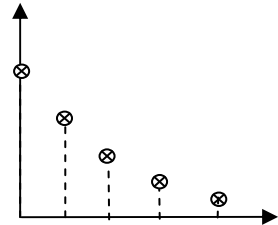


média ponderada exponencialmente pelo atraso

Contribuições significativas: $\text{peso} > .02 \implies n-i < 4 \tau$

Máximo atraso significativo (lote significativo):

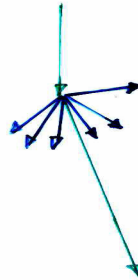
$$N_0 \cong 4\tau \cong \frac{4}{1-\beta}$$



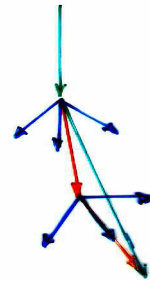
Contribuição total de $\Delta w_D(n)$:

$$\begin{aligned} \Delta w_D(n) (1-\beta) (1 + \beta + \beta^2 + \dots) &= \\ &= \Delta w_D(n) (1-\beta) \frac{1}{1-\beta} = \Delta w_D(n) \end{aligned}$$

Compensa a “instabilidade” do
processo de regra delta



batelada



lotes



regra delta



momento

Ajuste de Modelo por Regra Delta com Momento - Algoritmo

Inicialização $n=1$; $\alpha = .1$ $\beta = .9$ $\Delta w_i^{RD}(0) = 0 \quad \forall i$

para as $i = 1, 2, \dots, V$ variáveis $w_i(1) \in [+.2, -.2]$ randômicos;

Até que o critério de parada seja satisfeito

para cada par $(\underline{x}^p, y^p)$ $p = 1, \dots, P$

calcular

$$\underline{\tilde{y}}^p = \phi(\underline{x}^p)$$

$$\varepsilon_m^p = y_m^p - \tilde{y}_m^p \quad \text{e} \quad \frac{\partial \tilde{y}_m^p}{\partial w_i} \quad \forall m, i$$

Regra Delta com Momento - Algoritmo (continuação)

$$\Delta w_i^{RD}(n) = 2\alpha \sum_{m=1}^M \varepsilon_m^p \frac{\partial y_m^p}{\partial w_i} \quad \forall i$$

$$\Delta w_i(n) = \beta \Delta w_i(n-1) + (1-\beta) \Delta w_i^{RD}(n) \quad \forall i$$

$$w_i(n+1) = w_i(n) + \Delta w_i(n) \quad \forall i$$

$$n = n + 1$$

outro par

Ajuste de modelo por BP Resiliente - Algoritmo

Inicialização

$$n = 1 \quad \alpha_{\max} \approx .5 \quad a \approx 1.05$$

para as $i = 1, 2, \dots, V$ variáveis

$$w_i(1) \in [+.2, -.2] \text{ randômicos}; \quad \alpha_i(1) = .1 \quad \frac{\partial F}{\partial w_i}(0) = 0$$

Até que o critério de parada esteja satisfeito

Passo de treinamento n

BP Resiliente - Algoritmo (continuação)

Cálculo do gradiente

para cada par $(\underline{x}^p, \underline{y}^p)$ $p = 1, \dots, P$

calcular $\tilde{\underline{y}}^p = \phi(\underline{x}^p)$

$$\varepsilon_m^p = y_m^p - \tilde{y}_m^p \quad \text{e} \quad \frac{\partial \tilde{y}_m^p}{\partial w_i} \quad \forall m, i$$

$$\frac{\partial \varepsilon^{p2}}{\partial w_i} = -2 \sum_{m=1}^M \varepsilon_m^p \frac{\partial y_m^p}{\partial w_i} \quad \forall i$$

outro par

calcular $\frac{\partial F}{\partial w_i}(n) = E_p \frac{\partial \varepsilon^{p2}}{\partial w_i}$

BP Resiliente - Algoritmo (continuação)

Cálculo do acréscimo

para $i = 1, 2, \dots, V$

$$\alpha_i(n) = \begin{cases} \alpha_i(n-1) & \text{se } \text{sign} \frac{\partial F}{\partial w_i}(n) = \text{sign} \frac{\partial F}{\partial w_i}(n-1) \\ & \text{mas se } \alpha_i(n) > \alpha_{\max} \text{ faça } \alpha_i(n) = \alpha_{\max} \\ \frac{\frac{\partial F}{\partial w_i}(n-1)}{\frac{\partial F}{\partial w_i}(n-1) - \frac{\partial F}{\partial w_i}(n)} \alpha_i(n-1) & \text{se } \text{sign} \frac{\partial F}{\partial w_i}(n) \neq \text{sign} \frac{\partial F}{\partial w_i}(n-1) \end{cases}$$

$$\Delta w_i(n) = -\alpha_i(n) \frac{\partial F}{\partial w_i}(n)$$

$$w_i(n+1) = w_i(n) + \Delta w_i(n)$$

$$n = n + 1$$

Ajuste de Modelo por Levenberg- Marquadt - Algoritmo

Inicialização

$n = 1$; para as $i = 1, 2, \dots, V$ variáveis $w_i(1) \in [+.2, -.2]$ randômicos

Até que o critério de parada esteja satisfeito

Passo de treinamento n

Levenberg- Marquadt - Algoritmo (continuação)

Cálculo do gradiente g_i e da aproximação da Hessiana $[\tilde{h}_{ij}]$

para cada par $(\underline{x}^p, \underline{y}^p)$ $p = 1, \dots, P$

calcular $\tilde{\underline{y}}^p = \phi(\underline{x}^p)$

$$\varepsilon_m^p = y_m^p - \tilde{y}_m^p \quad \text{e} \quad \frac{\partial \tilde{y}_m^p}{\partial w_i} \quad \forall m, i$$

$$\frac{\partial \varepsilon^p}{\partial w_i} = - \sum_{m=1}^M \frac{\partial y_m^p}{\partial w_i} \quad \frac{\partial \varepsilon^{p2}}{\partial w_i} = - 2 \sum_{m=1}^M \varepsilon_m^p \frac{\partial y_m^p}{\partial w_i} \quad \forall i$$

outro par

calcular $\frac{\partial F}{\partial w_i}(n) = g_{i(n)} = E_p \frac{\partial \varepsilon^{p2}}{\partial w_i}$

$$\frac{\partial^2 F}{\partial w_i \partial w_j}(n) \cong \tilde{h}_{ij(n)} = 2 E_p \frac{\partial \varepsilon^p}{\partial w_i} \frac{\partial \varepsilon^p}{\partial w_j}$$

Levenberg- Marquadt - Algoritmo (continuação)

Cálculo e aplicação do passo $\Delta \underline{w}$

escolha λ_n adequado e inverta $\underline{\tilde{H}}_n = [\underline{\tilde{H}}_n + \lambda_n \underline{I}]$

calcule $\underline{d}_n = -\underline{\tilde{H}}_n^{-1} \underline{g}_n$

realize a otimização em linha

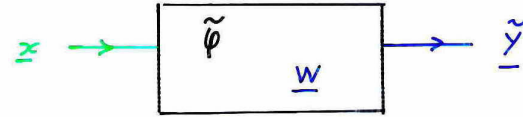
calcule $\alpha^* = \underset{\alpha > 0}{\text{Arg}} [\text{Min } F(\underline{w}_n + \alpha \underline{d}_n)]$

$$\underline{w}_{n+1} = \underline{w}_n + \alpha^* \underline{d}_n$$

$$n = n + 1$$

Comentários finais

Composição de F como erro



$$F(\underline{y}, \underline{\tilde{y}})$$

$$\underline{\tilde{y}}(\underline{w})$$

$$\frac{\partial F}{\partial w_i} = \frac{\partial F}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial w_i}$$

onde $\frac{\partial F}{\partial \tilde{y}}$ depende da função erro utilizada e $\frac{\partial \tilde{y}}{\partial w_i}$ depende das equações do modelo utilizado

no caso de saídas múltiplas

$$\frac{\partial F}{\partial w_i} = \sum_{m=1}^M \frac{\partial F}{\partial \tilde{y}_m} \frac{\partial \tilde{y}_m}{\partial w_i}$$

Outros critérios de erro

Erro quadrático médio ponderado

$$F = E_p \sum_m a(p, m) \varepsilon_m^2$$

$a(m)$ escalamento de y_m

$a(p)$ população

Saídas com relevâncias diferentes

Erro relativo médio quadrático

$$F = E_p \left(\frac{Y_m - \tilde{Y}_m}{Y_m} \right)^2 \quad Y - \text{variáveis não normalizadas}$$

Domínios com baixa população

$$a(p, m) = \frac{P_{\max}}{Pop(p)}$$

Crítica (acompanhamento) durante o treinamento

Evolução do Erro durante o treinamento

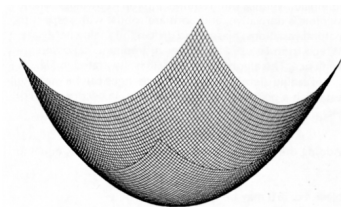


Fig. 22. Example MSE surface of linear error.



Fig. 23. Example MSE surface of sigmoid error.

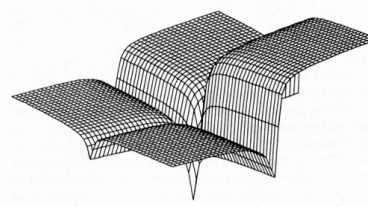
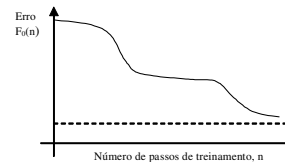
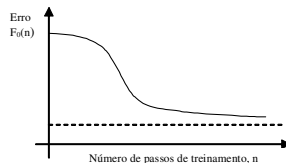
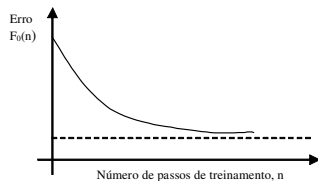
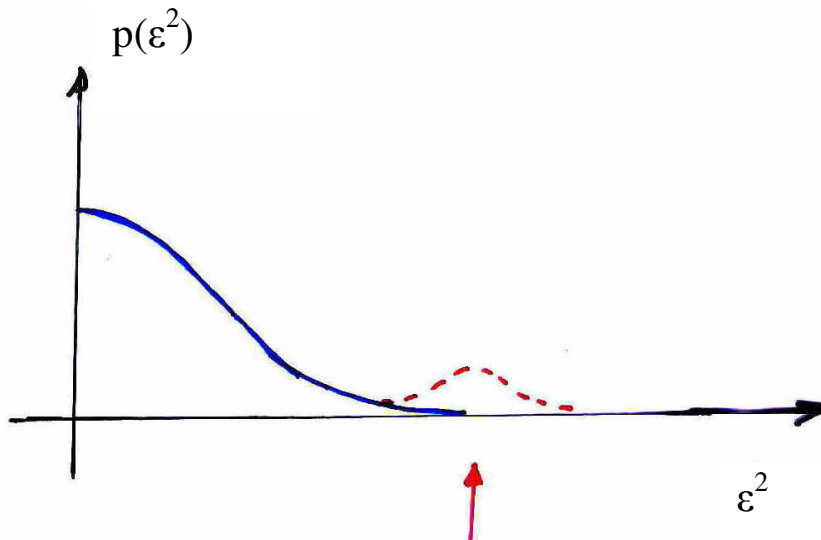


Fig. 30. Example MSE surface of trained sigmoidal network as a function of two first-layer weights.



Paralisia (o ponto de partida é crítico)

Crítica pós treinamento



possivelmente devido à
elementos de região com baixa população ou
intruso (outlayer) e seus vizinhos

Por enquanto

FIM

Mas mais detalhes (importantes) serão vistos para redes neurais em

CPE 721

no próximo período.