CPE 723 - OTIMIZAÇÃO NATURAL

Luiz P. Calôba

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

Módulo I - Otimização Contínua

Parte 1 – Métodos de Otimização

Aproximação local (Taylor), Gradiente ∇, Hessiana H

Extremos, mínimo, máximo

Propriedades ∇ e H

Métodos analíticos

Métodos numéricos recursivos

Gradiente descendente, passo, passo variável, ótimo

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

Newton amortecido, Quasi Newton

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

Esta é a primeira edição desta apostilha. Críticas, comentários e informações sobre erros localizados são muito bem-vindas.

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.

Cichocki & Unbehauen:

Neural Networks for Optimization and Signal Processing

Andrzej Cichocki
Warsaw University of Technology
Poland

Rolf Unbehauen
Universität Erlangen-Nürnberg
Germany

Parte 1 – Métodos de Otimização Contínua "clássica"

Noções Elementares de Otimização

I – Função objetivo – a minimizar

Um problema muito comum em diversas áreas é, dada uma função escalar F de n variáveis w_1, w_2, ..., w_i , ... w_n , descobrir qual o valor destas variáveis tal que o valor da função seja mínimo.

$$F(w_1, w_2, \ldots, w_n) = F(\underline{w}) \qquad \underline{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

Nosso problema é então encontrar o minimante $\underline{w}^*$ de F tal que

$$F(\underline{w}^*) \leq F(\underline{w}) \qquad \forall\ \underline{w} \neq \underline{w}^* \qquad \text{ou}$$

$$\underline{w}^* = \underset{\forall \underline{w}}{Arg\ Min}\ F(\underline{w})$$

Em nosso caso de interesse os domínios são reais, F é não negativa e tem derivadas de primeira e segunda ordem em relação à quaisquer das variáveis w_i

Propriedades de F

$$F(\underline{w}) \text{ e } \underline{w} \text{ são reais. } F(\underline{w}) \geq 0$$

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

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

Os aspectos simplificadores do nosso problema são que F e $\underline{w}$ são reais, F é contínua e derivável, e não há restrições sobre os valores de cada w_i . Os complicadores são que F pode ser não linear, não convexa, e $\underline{w}$ pode ter dimensão muito elevada.

Mínimos Locais e Globais

Buscamos o valor de $\underline{w}^*$ tal que $F(\underline{w}^*) \leq F(\underline{w})$ para qualquer outro $\underline{w}$ diferente de $\underline{w}^*$. Neste caso $F(\underline{w}^*)$ e chamado de um mínimo global da função, mas sua determinação implica em conhecer todo o domínio infinito de $\underline{w}$, o que somente é possível se a forma analítica de F for conhecida.

Mas é útil conhecer o mínimo de uma função em um domínio limitado, digamos em uma esfera de raio ε e centro em $\underline{w}^*$. Este ponto é chamado de mínimo local, porque é mínimo apenas em um domínio limitado.

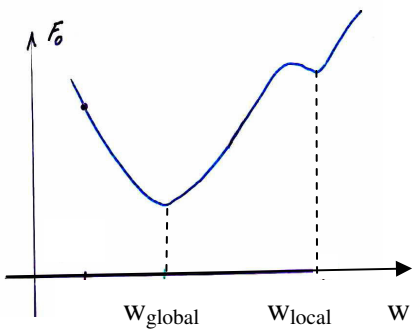
Para um mínimo global

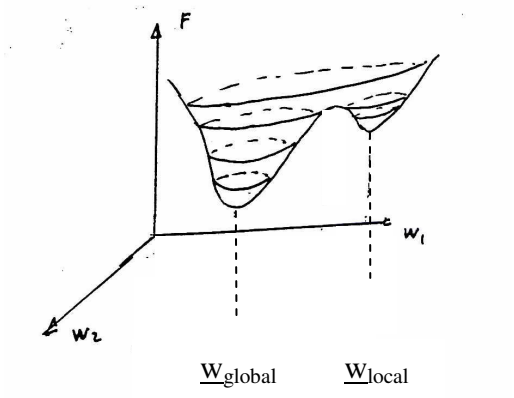
$$F(\underline{w}^*) \leq F(\underline{w}) \qquad \forall\ \underline{w} \neq \underline{w}^*$$

e para um mínimo local

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

As funções $F_0(w)$ e $F(w_1, w_2)$ esboçadas nos gráficos abaixo apresentam mínimos globais e locais.





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

Qualquer função que admita derivadas pode ser aproximada em uma esfera de raio ε centrada em um ponto $\underline{w}_0$ por uma série de Taylor

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

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

Na expressão da série acima estão explicitados os acréscimos de ordem zero, de primeira e de segunda ordem. Termos de ordem mais elevada são normalmente desconsiderados e não estão explicitados. Uma notação vetorial, mais simples e elegante, pode ser utilizada. Considere os vetores acréscimo $\underline{\Delta w}$ e gradiente $\underline{\nabla}(\underline{w}_0)$, e a matriz Hessiana $\underline{H}(\underline{w}_0)$ definidos abaixo

$$\underline{\Delta w} = \left[\Delta w_i\right] = \begin{bmatrix} \Delta w_1 \\ \Delta w_2 \\ \ldots \\ \Delta w_n \end{bmatrix} \qquad \underline{\nabla}(\underline{w}_0) = \left[g_i\right] = \left[\left.\frac{\partial F}{\partial w_i}\right|_{\underline{w}_0}\right] = \begin{bmatrix} \left.\frac{\partial F}{\partial w_1}\right|_{\underline{w}_0} \\ \left.\frac{\partial F}{\partial w_2}\right|_{\underline{w}_0} \\ \ldots \\ \left.\frac{\partial F}{\partial w_n}\right|_{\underline{w}_0} \end{bmatrix}$$

$$\underline{H}(\underline{w}_0) = \left[h_{ij}\right] = \left[\left.\frac{\partial^2 F}{\partial w_i \, \partial w_j}\right|_{\underline{w}_0}\right] = \begin{bmatrix} \left.\frac{\partial^2 F}{\partial w_1^2}\right|_{\underline{w}_0} & \left.\frac{\partial^2 F}{\partial w_1 \, \partial w_2}\right|_{\underline{w}_0} & \ldots & \left.\frac{\partial^2 F}{\partial w_1 \, \partial w_n}\right|_{\underline{w}_0} \\ \left.\frac{\partial^2 F}{\partial w_2 \, \partial w_1}\right|_{\underline{w}_0} & \left.\frac{\partial^2 F}{\partial w_2^2}\right|_{\underline{w}_0} & \ldots & \left.\frac{\partial^2 F}{\partial w_2 \, \partial w_n}\right|_{\underline{w}_0} \\ \ldots & \ldots & \ldots & \ldots \\ \left.\frac{\partial^2 F}{\partial w_n \, \partial w_1}\right|_{\underline{w}_0} & \ldots & \ldots & \left.\frac{\partial^2 F}{\partial w_n^2}\right|_{\underline{w}_0} \end{bmatrix}$$

Note que o i-ésimo componente do gradiente, g_i, é a derivada da função objetivo em relação à i-ésima variável, w_i. E a componente h_{ij} da Hessiana é a derivada segunda da função objetivo em relação à w_i e w_j. $\underline{H}$ é simétrica.

Observando a expressão da aproximação pela série de Taylor da função verificamos que os acréscimos de primeira ordem correspondem ao produto interno dos vetores acréscimo e gradiente, e que os acréscimos de segunda ordem são dados pela metade da Hessiana pré e pós multiplicada pelo vetor acréscimo.

$$\sum_i \left.\frac{\partial F}{\partial w_i}\right|_{\underline{w_0}} \Delta w_i = \underline{\Delta w}^t \, \underline{\nabla}(\underline{w}_0) \qquad e \qquad \sum_i \sum_j \left.\frac{\partial^2 F}{\partial w_i \, \partial w_j}\right|_{\underline{w}} \Delta w_i \Delta w_j = \underline{\Delta w}^t \; \underline{H}(\underline{w}_0)\, \underline{\Delta w}$$

Então

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

A ordem das derivadas usadas dá a ordem da aproximação. Quanto maior a ordem maior o entorno ε no qual a aproximação é válida. Em nosso estudo usaremos apenas aproximações de primeira e segunda ordem.

Aproximação linear ou de 1ª. ordem

A aproximação de primeira ordem envolve apenas o termo constante $F(\underline{w}_0)$ e as derivadas de primeira ordem.

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

Se a própria função $F(\underline{w})$ é linear a aproximação de 1ª. ordem é exata. O gradiente $\underline{\nabla}(\underline{w})$ é constante e as derivadas de ordem maior que 1 são nulas, i.e., a Hessiana $\underline{H}(\underline{w})$ é nula para todo $\underline{w}$.

$$\underline{\nabla}(\underline{w}) = \text{ctte}(\underline{w}) \quad e \quad \underline{H}(\underline{w}) = \underline{0}$$

Aproximação quadrática ou de 2ª. ordem

Envolve o termo constnte e as derivadas de primeira e segunda ordem apenas.

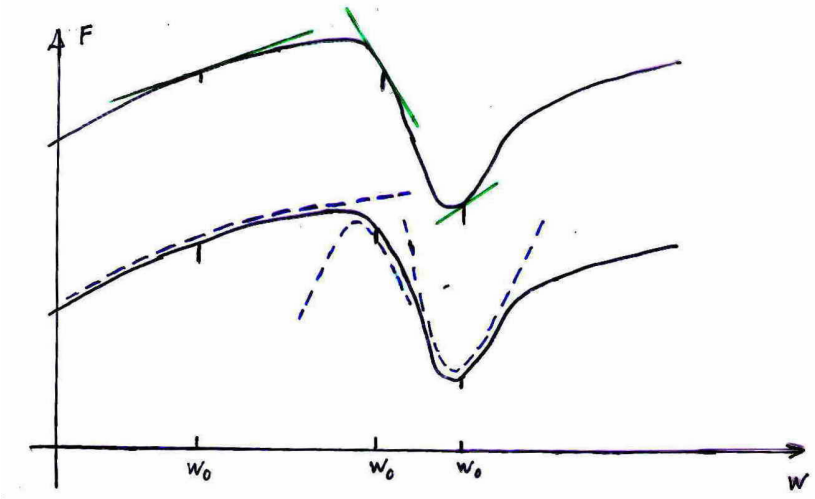
$$F(\underline{w}_0 + \Delta w) \simeq F(\underline{w}_0) + \underline{\Delta w}^{\,t}\ \underline{\nabla}\,(\underline{w}_0) + \frac{1}{2}\,\underline{\Delta w}^{\,t}\ \underline{H}(\underline{w}_0)\,\Delta\underline{w}$$

Se F(w) é quadrática a aproximação de 2ª. ordem é exata. A Hessiana $\underline{H}(\underline{w})$ é constante e as derivadas de ordem maior que dois são nulas.

$$\underline{H}(\underline{w}) = \text{ctte}\,(\underline{w})$$

Validade das aproximações

A região em torno de $\underline{w}_0$ em que uma aproximação é válida com um erro menor que δ depende da forma da função no entorno do ponto, isto é, depende do acréscimo na função, e não do acréscino nas variáveis independentes. A figura abaixo esboça aproximações de primeira e segunda ordem para uma função, onde fica claro que os domínios onde a aproximação vale dependem do comportamento da função na região. Estes domínios não são obrigatoriamente esferas, nem simétricos em relação a $\underline{w}_0$. É fácil verificar, também, que dependem da direção considerada para funções de mais de uma variável. Uma melhor aproximação para estes domínios seria um elipsóide, não uma esfera.



Entretanto a série de Taylor garante que o erro cometido na aproximação é sempre menor que o primeiro termo desprezado na série. Assim, podemos aceitar que a aproximação de primeira ordem

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

é válida enquanto

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

como $| \underline{\Delta w} | \leq \varepsilon$ sempre é possível escolher ε suficientemente pequeno para que a aproximação de primeira ordem seja válida com erro menor ou igual a δ em uma esfera de raio ε centrada em $\underline{w}_0$.

Aproximação do Gradiente

O gradiente também é uma função, e admite aproximação. Pelos mesmos critérios anteriores a aproximação de primeira ordem de cada termo do gradiente no entorno de $\underline{w}_0$ pode ser escrita

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

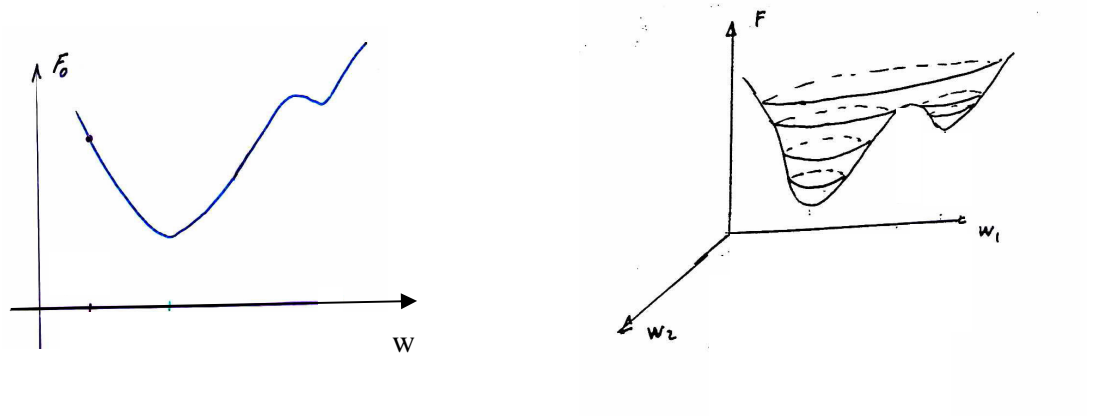
Em forma vetorial a aproximação de primeira ordem do gradiente é:

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

Funções de duas ou mais variáveis

Funções de uma variável são muito simples de visualizar em um sistema cartesiano, porque são representadas por linhas. Usualmente utilizamos um espaço bidimensional com um eixo horizontal para representar a variável independente F e um vertical para representar a variável dependente w.

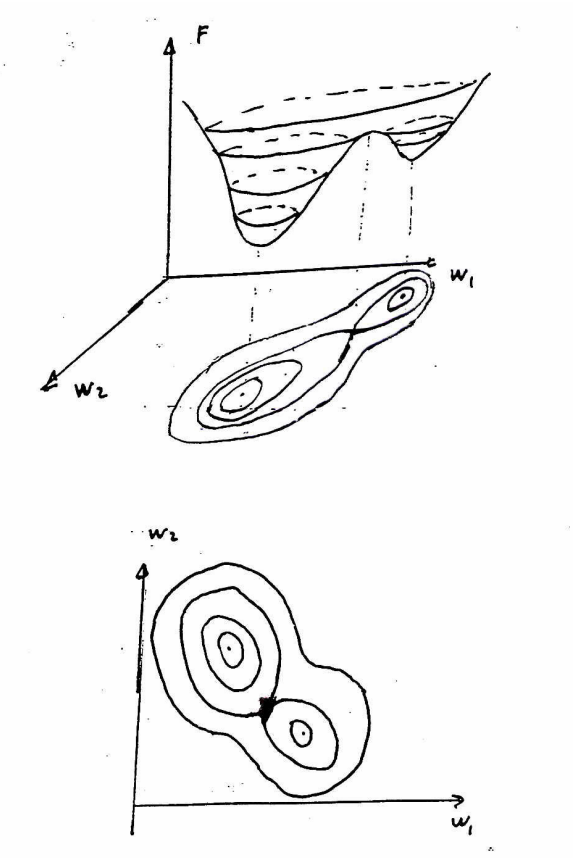
Funções de duas variáveis são mais complexas de visualizar, porque são representadas por superfícies em um espaço tridimensional e requerem uma visualização em perspectiva. Usualmente o eixo vertical para representar a variável dependente F e dois eixos representando em perspectiva um plano horizontal, domínio das variáveis independentes w_1 e w_2..



Visualização por Curvas de Nível

Na área de topografia, onde a visualização de relevos tridimensionais é fundamental, a solução encontrada é visualizar estas superfícies tridimensionais pelas suas curvas de nível. Para isto são criados diversos planos horizontais, cada um cortando o eixo vertical em diferentes valores (cotas, em topografia). As intersecções destes planos com a superfície que representa a função são as curvas de nível. Estas curvas de nível são então projetadas sobre o plano norizontal, gerando uma representação em duas dimensões da superfície tridimensional..

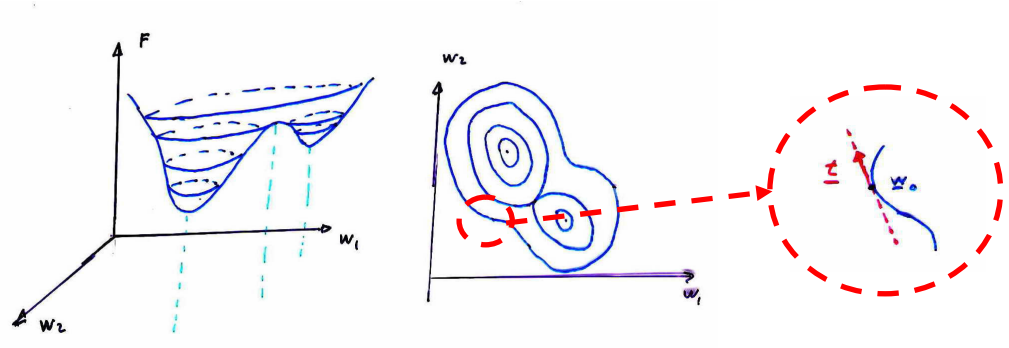
Generalizando, as curvas de nível são projeções no espaço das variáveis independentes $\underline{w}$ do lugar geométrico onde a função assume um determinado valor. Assim, para funções de uma variável as curvas de nível degeneram em pontos, paras duas variáveis são curvas, para três variáveis são superfícies no espaço tridimensional e para mais que três variáveis são hipersuperfícies em espaços hiperdimensionais.



Curvas e Superfícies de nível

Considere a reta **t** tangente à uma curva de nível no ponto $\underline{w}_0$. Esta tangente representa a aproximação de primeira ordem da curva de nível no ponto $\underline{w}_0$. Pela própria definição de curva de nível pequenos deslocamentos à partir de $\underline{w}_0$ sobre a reta **t** não alteram o valor de F, isto é

$$F(\underline{w}_0 + \alpha\,\underline{t})\big|_{\alpha<<} \cong F(\underline{w}_0) \qquad \left.\frac{dF}{d\underline{w}}\right|_{\underline{t}} = 0 \qquad \left.\frac{d}{d\alpha} F(\underline{w}_0 + \alpha\,\underline{t})\right|_{\alpha=0} = 0$$



Logo a derivada da função na direção da tangente à curva de nível é nula.

Para o caso de superfícies de nível a reta tangente se generaliza em um plano (ou hiperplano) tangente $\underline{\pi}$, e novamente pequenos deslocamentos à partir de $\underline{w}_0$ sobre $\underline{\pi}$ não alteram o valor de F, isto é

$$F(\underline{w}_0 + \alpha\, \underline{t})\Big|_{\substack{\underline{t}\in\underline{\pi} \\ \alpha<<}} \cong F(\underline{w}_0) \qquad \frac{dF}{d\underline{w}}\bigg|_{\underline{t}\in\underline{\pi}} = 0 \qquad \frac{d}{d\alpha} F(\underline{w}_0 + \alpha\, \underline{t})\Big|_{\substack{\underline{t}\in\underline{\pi} \\ \alpha=0}} = 0$$

Logo a derivada da função em qualquer direção contida no plano tangente à uma superfície de nível é nula.

III – Extremos:

Mínimo

O objetivo de nosso estudo é descobrir os mínimantes $\underline{w}^*$ (e os mínimos) de uma função F($\underline{w}$), isto é, o ponto em que

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

. No entorno de um minimante a função pode ser aproximada por

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

O acréscimo ΔF da função é dado por dois termos, um de primeira e outro de segunda ordem.

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

Para | $\Delta\underline{w}$ | muito pequenos o termo de segunda ordem pode ser desconsiderado, e

$$\Delta F \cong \underline{\Delta w}^t \, \underline{\nabla}(\underline{w}^*) = \left|\underline{\Delta w}\right| \left|\underline{\nabla}(\underline{w}^*)\right| \cos \angle \underline{\Delta w}, \underline{\nabla}(\underline{w}^*)$$

Como $\angle \underline{\Delta w}, \underline{\nabla}(\underline{w}^*) \in [0, 2\pi]$, $\cos \angle \underline{\Delta w}, \underline{\nabla}(\underline{w}^*) \in [-1,1]$, e a condição necessária e suficiente para que o acréscimo seja não negativo é

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

Para $|\,\underline{\Delta w}\,|$ um pouco maiores o termo de segunda ordem usualmente domina, e além disto o gradiente é nulo, então

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

E a condição para que o acréscimo ΔF seja positivo é

$$\underline{\Delta w}^t \, \underline{H}(\underline{w}^*) \, \underline{\Delta w} > 0$$

Isto é, que $\underline{H}(w^*)$ seja definida positiva. Uma matriz real é dita definida positiva quando pré e pós multiplicada por um mesmo vetor não nulo resulta em um número positivo, qualquer que seja o vetor.

As condições para que um ponto w^* seja minimante de uma função são então:

$$\underline{\nabla}\left(\underline{w}^*\right) = \underline{0} \qquad e \qquad \underline{H}(\underline{w}^*) \text{ definida positiva}$$

Extremos

Um ponto $\underline{w}^*$ em que o gradiente é nulo é chamado extremo da função, é classificado dependendo da Hessiana e, em conseqüência, do tipo de acréscimo ΔF na função a que um pequeno deslocamento conduz (determinado pela Hessiana).

Tipos de Extremos:

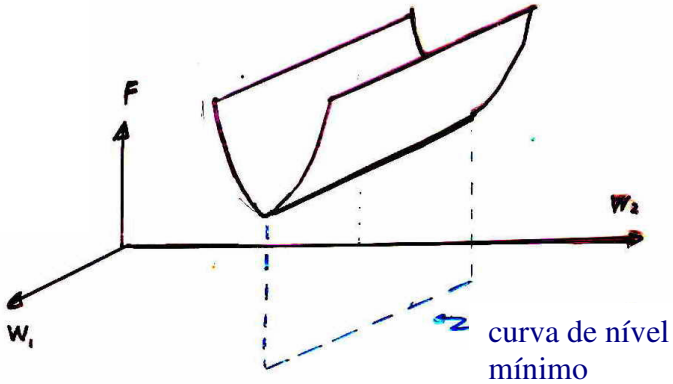
$\nabla(\underline{w}^*)$	$\underline{H}(\underline{w}^*)$	$\Delta F = F(\underline{w}^* + \underline{\Delta w}) - F(\underline{w}^*) = \frac{1}{2}\underline{\Delta w}^t \underline{H}(\underline{w}^*)\underline{\Delta w}$	Tipo de Extremo $\underline{w}^*$
$\underline{0}$	definida positiva	positivo	mínimo
$\underline{0}$	semidefinida positiva	positivo ou nulo	calha
$\underline{0}$	definida negativa	negativo	máximo
$\underline{0}$	semidefinida negativa	negativo ou nulo	cumeeira
$\underline{0}$	não definida	positivo, negativo ou nulo	sela

Um exemplo de cada tipo de extremo é esboçado a seguir em um espaço bidimensional, dim $\underline{w}$ = 2. LG significa Lugar Geométrico. Máximos e mínimos são pontos, independente da dimensão do espaço. Calhas, cumeeiras e selas são curvas no espaço bidimensional e superfícies no espaço multidimensional, dim $\underline{w}$ > 2. Calhas também são mínimantes.

H semidefinida positiva

$\Delta F \geq 0$

CALHA

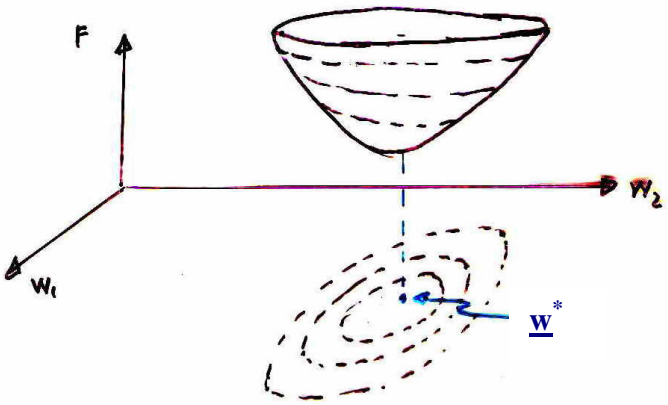


curva de nível mínimo
LG de $\underline{w}^*$

H definida positiva

$\Delta F > 0$

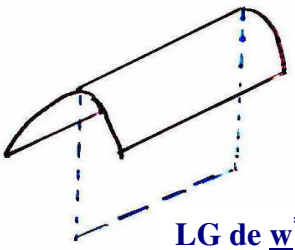
MÍNIMO
propriamente dito



H semidefinida negativa

$\Delta F \leq 0$

CUMEEIRA

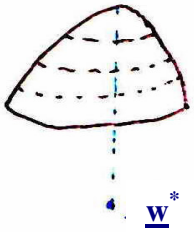


LG de $\underline{w}^*$

H definida negativa

$\Delta F < 0$

MÁXIMO

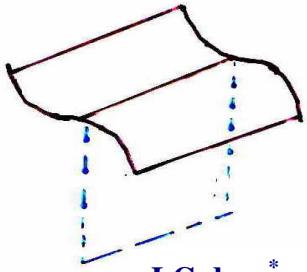


$\underline{w}^*$

H não definida

$\Delta F \geq 0$ ou $\Delta F < 0$

SELA



LG de $\underline{w}^*$

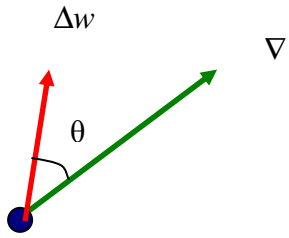
IV – Propriedades do Gradiente

O gradiente será a nossa principal ferramenta nos processos de procura de extremos de uma função, e portanto vale a pena examinar algumas de suas propriedades. Um acrécimo $\underline{\Delta w}$ (pequeno o suficiente para que a aproximação de primeira ordem valha) aplicado no ponto de operação $\underline{w}_0$ provoca um acréscimo ΔF na função

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

$$\Delta F \cong \underline{\Delta w}^t \, \nabla(\underline{w}_0) = |\underline{\Delta w}| \; |\underline{\nabla} F(\underline{w}_0)| \cos\theta$$

$$\theta = \angle \underline{\Delta w}, \underline{\nabla} F(\underline{w}_0)$$



1.a) Para $|\underline{\Delta w}|$ constante (e pequeno o suficiente) o gradiente indica a direção e o sentido de deslocamento ($\theta = 0$) do ponto de operação que provoca o máximo acréscimo em F.

1.b) O deslocamento na mesma direção e no sentido contrário do gradiente indica a direção ($\theta = \pi$) de máximo decréscimo em F.

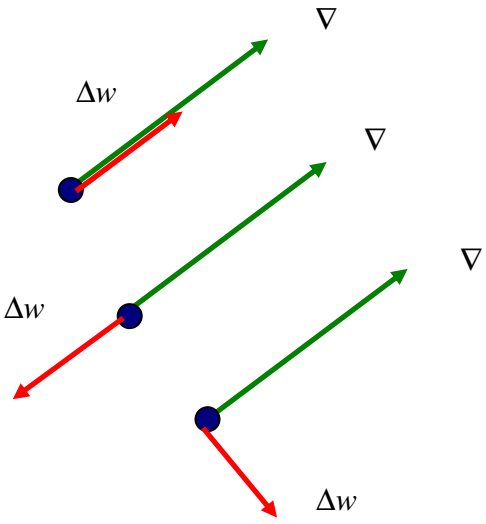
1.c) O deslocamento em qualquer direção ortogonal ao gradiente ($\theta = \pm \pi / 2$) não altera o valor de F.

$\theta = 0$ ⇒ ΔF max

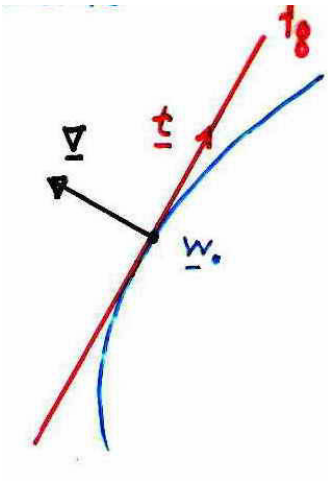
$\theta = \pi$ ⇒ ΔF min

(-ΔF max)

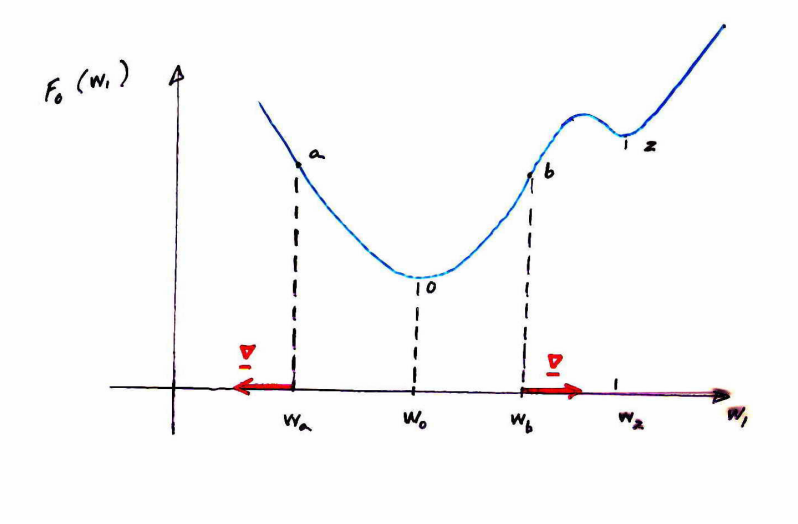
$\theta = \pm \frac{\pi}{2}$ ⇒ $\Delta F = 0$

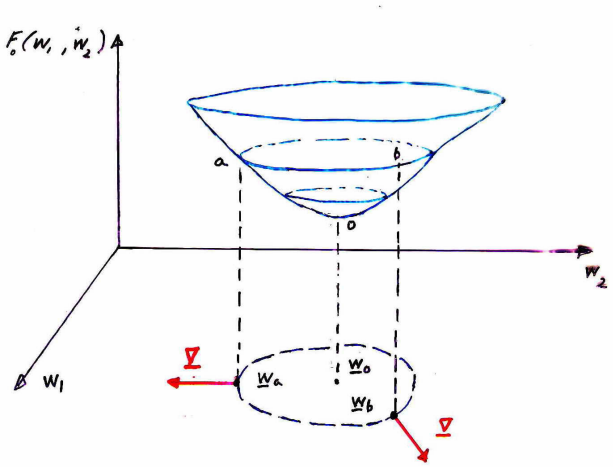


2 – O gradiente é ortogonal ao plano tangente à superfície de nível que passa pelo ponto. Vimos anteriormente que quaisquer pequenos deslocamentos de $\underline{w}_0$ sobre o plano tangente à superfície passando por $\underline{w}_0$ não alteram o valor de F, logo estes deslocamentos tem que ser ortogonais ao gradiente no ponto. Mas se o gradiente é ortogonal a quaisquer vetores contidos no plano tangente, então é ortogonal ao plano tangente.



Nas figuras à seguir esboçamos o gradiente para duas funções F em dois pontos diferentes





V. Otimização

Nosso problema é encontrar um ponto $\underline{w}^*$ onde $F(\underline{w}^*)$ seja mínima.

VI – Otimização: Solução Analítica

Dos itens anteriores um minimante (local ou global) da função é o ponto $\underline{w}^*$ onde o gradiente é nulo e a Hessiana é definida positiva

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

A - Funções Lineares

Funções lineares tem a forma

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

onde $\underline{a}$ é um vetor e b um escalar. A função é representada por uma reta, um plano ou um hiperplano, dependendo da dimensão de $\underline{w}^*$. Fazendo $\underline{w}_0 = \underline{0}$, $\Delta\underline{w} = \underline{w}$ vemos que a aproximação de primeira ordem no entorno da origem é a própria função. O gradiente da função é constante e não se anula, e a Hessiana é nula.

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

$$\nabla(\underline{w}) = \underline{a} = \text{ctte}(\underline{w}) \qquad \underline{H}(\underline{w}) = \underline{0}$$

O mínimo deste caso degenerado ocorre para $|\Delta\underline{w}|$ tendendo a infinito e não tem interesse prático.

B – Funções Quadráticas

Funções quadráticas tem a forma

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

onde $\underline{A}$ é uma matriz, $\underline{b}$ é um vetor e c é um escalar. A função é representada por uma parábola, paraboloide ou hiper-paraboloide, dependendo da dimensão de $\underline{w}$. Fazendo $\underline{w}_0 = \underline{0}$, $\Delta\underline{w} = \underline{w}$ e vemos que a aproximação de segunda ordem no entorno da origem é a própria função.

$$F(\underline{0} + \underline{w}) = F(\underline{0}) + \underline{w}^t \nabla(\underline{0}) + \frac{1}{2}\underline{w}^t \underline{H}(\underline{0})\underline{w}$$

$$\nabla(\underline{0}) = \underline{b}$$

$$\underline{H}(\underline{w}) = 2\,\underline{A} = \text{ctte}(\underline{w})$$

Em um ponto $\underline{w}$ qualquer o gradiente é

$$\nabla(\underline{0} + \underline{w}) = \nabla(\underline{0}) + H(\underline{w})\,\underline{w} = \underline{b} + 2\,\underline{A}\,\underline{w}$$

O ponto $\underline{w}^*$ em que o gradiente se anula é dado por

$$\nabla(\underline{w}^*) = \underline{0} \qquad \Rightarrow \qquad \underline{w}^* = -\,\underline{H}^{-1}(\underline{0})\,\nabla(\underline{0}) = -\,\frac{1}{2}\,\underline{A}^{-1}\,\underline{b}$$

Igual dedução pode ser feita a partir de um ponto $\underline{w}_0$ qualquer. Neste caso o ponto $\underline{w}^*$onde o gradiente se anula é dados por

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

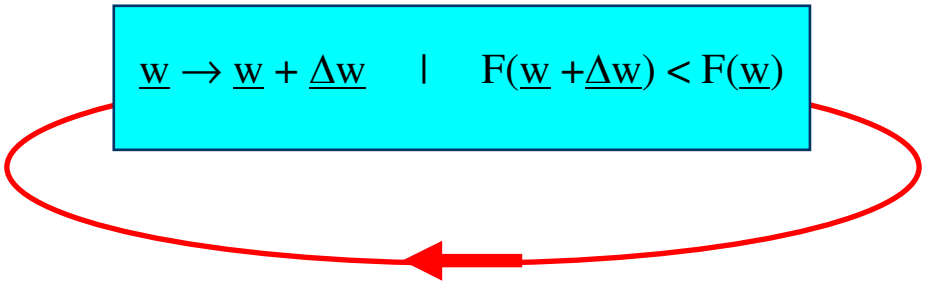
Esta equação é a base do método de Newton, ou método de Newton-Raphson. O problema é que o cálculo das derivadas de segunda ordem que compõem $\underline{H}$ pode ser trabalhoso, e a inversão de $\underline{H}$ pode ser numericamente mal condicionada, especialmente nos casos em que a dimensão de $\underline{w}$ é muito grande.

C – Funções de ordens mais elevadas e/ou transcendentes

Operar algebricamente com funções de ordem superior a dois ou com funções transcendentes pode ser extremamente complexo. Nestes casos a solução é usualmente obtida por métodos numéricos, que serão vistoa à seguir.

VII – Otimização: Métodos Numéricos Recursivos

O princípio dos métodos numéricos recursivos de determinação dos mínimos de uma função é extremamente simples: a partir de um ponto de operação inicial $\underline{w}$ dá-se um acréscimo $\underline{\Delta w}$ que move o ponto de operação para $\underline{w} + \underline{\Delta w}$, onde a função objetivo tem valor menor que no ponto anterior. Como a função é limitada inferiormente, repete-se o processo até que um minimante seja alcançado.



O acréscimo $\underline{\Delta w}$ é definido por dois parâmetros, o passo $\alpha > 0$ que controla o tamanho do acréscimo, e o vetor $\underline{d}$, que define a direção do acréscimo;

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

Métodos Numéricos Recursivos - Algorítmo geral

Inicialização dos parâmetros e variáveis

n= 1; $\underline{w}_1 = \ldots$

Até que o critério de parada seja satisfeito

Passo de treinamento n

calcular as variáveis intermediárias eventualmente necessárias

calcular $\underline{d}_n$

calcular α_n

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

n = n + 1

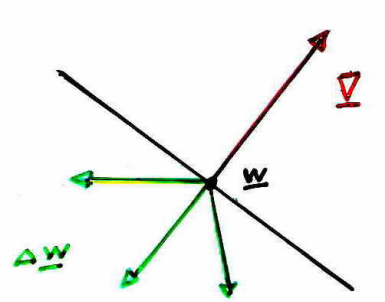
Condições necessárias

Para que o processo funcione é necessário que cada novo ponto de operação tenha um valor de F menor que o anterior. Se α for suficientemente pequeno para garantir que a aproximação de primeira ordem é válida o acréscimo em F será

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

e ΔF será negativo se

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

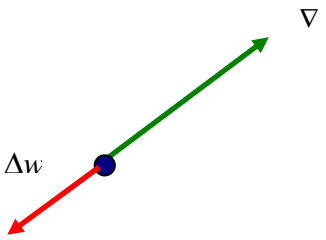


As condições para que o processo convirja são que (1) α seja suficientemente pequeno para que a aproximação de primeira ordem seja válida e (2) o ângulo entre a direção do acréscimo e o gradiente no ponto esteja entre 90 e 270 graus.

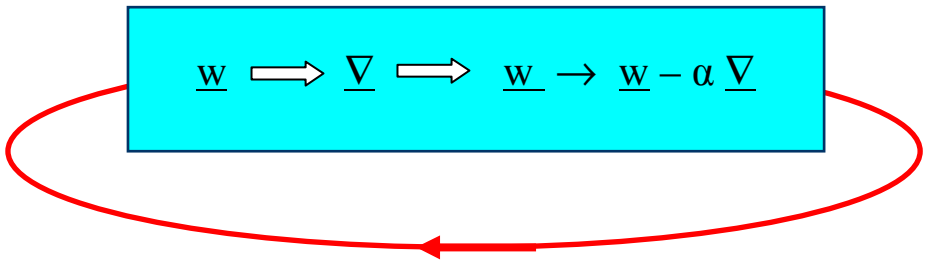
VIII – Otimização: Métodos Numéricos Recursivos de primeira ordem

Método do Gradiente Descendente ou Método da Descida pelo Gradiente

Fixado um α suficientemente pequeno, uma vez que diversas direções $\underline{d}$ satisfazem a restrição, nos perguntamos qual a "melhor" direção para este acréscimo, qual a direção que conduz à um maior decréscimo em F para um mesmo $|\Delta \underline{w}|$. Para que o decréscimo $-\Delta F$ seja máximo é necessário que $\cos \angle \underline{d}, \underline{\nabla} = -1$, isto é, que $\angle \underline{d}, \underline{\nabla} = 180^o$ ou ainda que $\underline{d} = - \underline{\nabla}$. Para o máximo decaimento em um único passo o deslocamento do ponto de operação deve ter a mesma direção e sentido contrário ao do gradiente. Neste caso o método é chamado "do gradiente descendente" ou "de descida pelo gradiente".

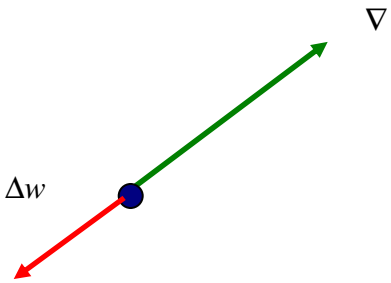


O método de Descida pelo Gradiente ou Gradiente Descendente consiste em calcular o gradiente no ponto de operação e mover este ponto de operação uma pequena distância na mesma direção e sentido contrário ao gradiente. Repetir a operação seguidamente até que um mínimo seja alcançado. O algoritmo esta esboçado na figura abaixo



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

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



O algoritmo é apresentado a seguir. Os valores numéricos são adequados para o treinamento tipo retropropagação (backpropagation) de Redes Neurais.

Gradiente Descendente - Algorítmo

Inicialização

n = 1; $\alpha \approx .1$

para as i = 1, 2, …, V variáveis $w_{i\,(1)} \in [+.2, -.2]$ randômicos

Até que o critério de parada seja satisfeito

Passo de treinamento n

Para i = 1, 2, …, V

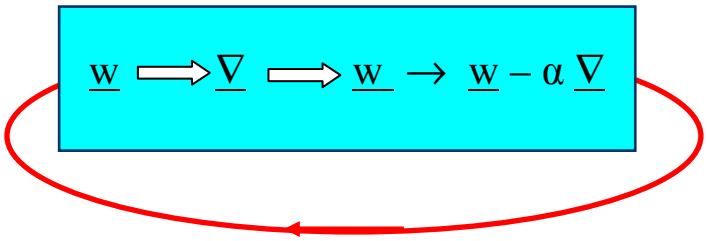
Calcular $\left.\frac{\partial F}{\partial w_i}\right|_n$

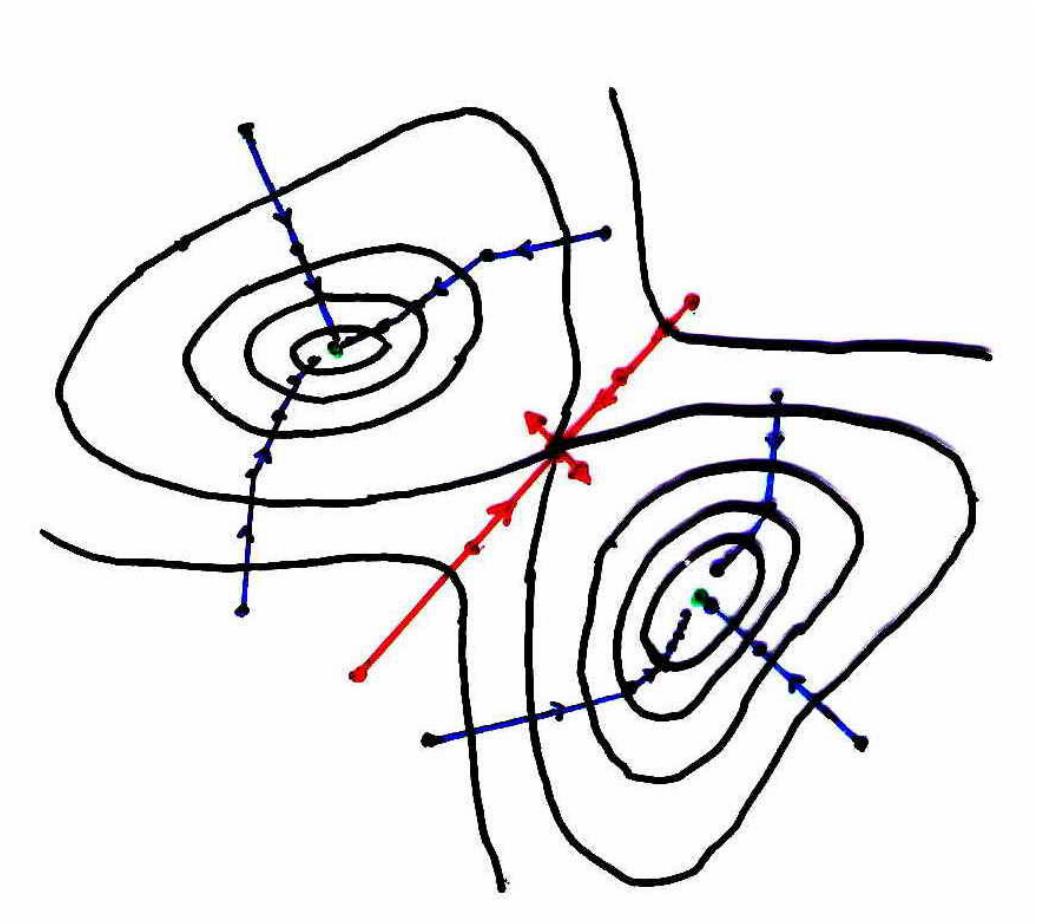
Fazer $w_{i\,(n+1)} = w_{i\,(n)} - \alpha \left.\frac{\partial F}{\partial w_i}\right|_n$

n = n + 1

Evolução do ponto de Operação no Método do Gradiente Descendente

A figura abaixo mostra a evolução do ponto de operação sobre curvas de nível para vários pontos de partida diferentes. Note que o mínimo que será atingido depende do ponto de partida do algoritmo: é sempre aquele do vale onde se encontra o ponto de partida. Não há garantia alguma de que seja o mínimo global.





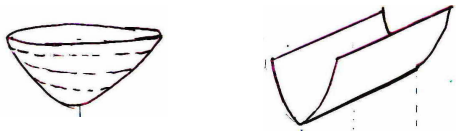
Critério de Parada - Fim expontâneo do processo

O minimante nunca é atingido exatamente devido à precisão numérica, o ponto de operação fica saltando de um lado para outro do minimante. O tamanho do salto dependendo do valor de α e das características da função no entorno do minimante. Mas o valor esperado do acréscimo após vários passos é nulo: o processo acaba quando o valor esperado no acréscimo do ponto de operação após vários passos é nulo, i.e., quando o (valor esperado do) gradiente é praticamente nulo.

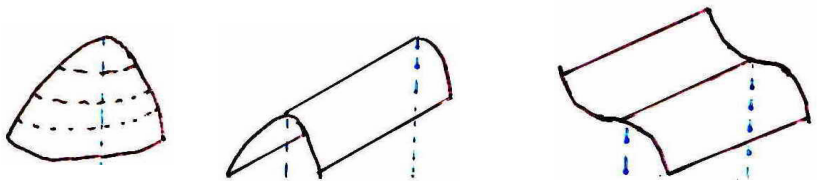
$$E(\Delta \underline{w}) = E[-\alpha \nabla(\underline{w})] = \underline{0} \qquad \Rightarrow \qquad \nabla(\underline{w}) \approx \underline{0}$$

A função está então em um extremo, mas que tipo de extremo ? Como o processo desloca o ponto de operação reduzindo a função este extremo será obrigatóriamente um mínimo ou uma calha. Máximos e pontos de sela são pontos instáveis para este método, à menos que sejam atingidos exatamente, o que é numericamente improvável.

pontos estáveis: mínimo, calha



pontos instáveis: máximo, cumeeira, sela



Em vez do valor médio do gradiente nulo os dois critérios de parada mais utilizados na prática são:

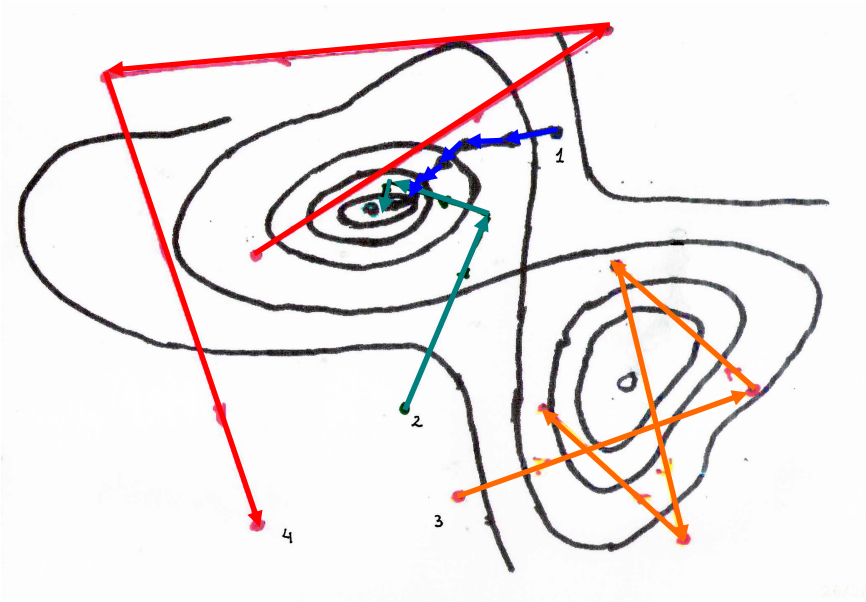
1 - um número máximo de passos preestabelecido é atingido $n \geq n_{max}$ ou

2 - o módulo do gradiente para de decair passo a passo ou

3 - o módulo do gradiente cai abaixo de um valor mínimo pré estabelecido, $|\nabla(\underline{w})| \leq \text{mod}_{\,min}$.

Passo de Treinamento α

"A escolha do passo de treinamento α é uma arte", diz Bernard Widrow. Se o passo escolhido for muito pequeno o processo fica muito lento, e se for escolhido grande demais oscila ou pode mesmo divergir. Na figura abaixo são mostradas várias trajetórias de pontos de operação com vários passos. Na azul (1) o passo é excessivamente pequeno, na verde (2) é ótimo, na laranja (3) é grande mas o processo ainda converge e na vermelha (4) é tão grande que o processo diverge.



Efeito do passo de treinamento α: um exemplo simples:

Na seqüência de figuras abaixo mostramos um caso muito simples, a trajetória do ponto de operação da função F = w^2 a partir de w_0 = 1 para vários passos de treinamento. Na figura à esquerda o passo é muito pequeno, na do meio é ótimo e na da direita são muito grandes: a trajetória azul mostra que o passo ultrapassa o mínimo, mas ainda converge, enquanto que na vermelha o processo diverge. Para esta função

$$F(w) = w^2 \qquad \nabla(w) = 2w \qquad H(w) = 2$$

$$w_0 = 1 \qquad \nabla(w_0) = 2 \qquad H(w_0) = 2$$

$$\Delta w = -\alpha \nabla(w_0) = -2\alpha$$

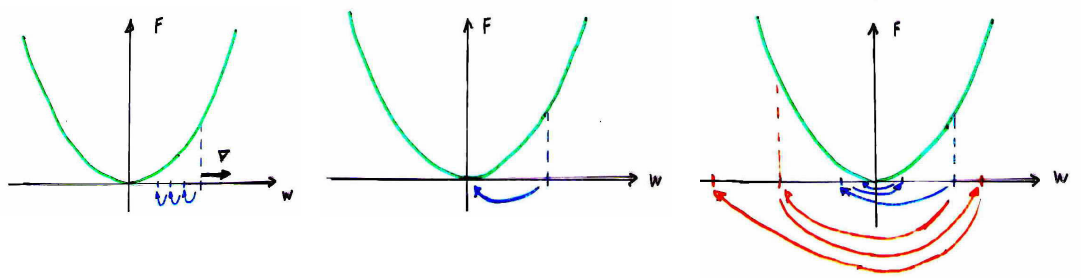
$$\nabla(w_0 + \Delta w) = \nabla(w_0) + H\,\Delta w = 2 + 2(-2\alpha) = 2(1 - 2\alpha)$$

Neste caso o valor ótimo de α que anula o gradiente em um único passo é

$$\nabla(w_0 + \Delta w) = 2(1 - 2\alpha) = 0 \qquad \alpha_{ótimo} = 0{,}5$$

O processo diverge quando

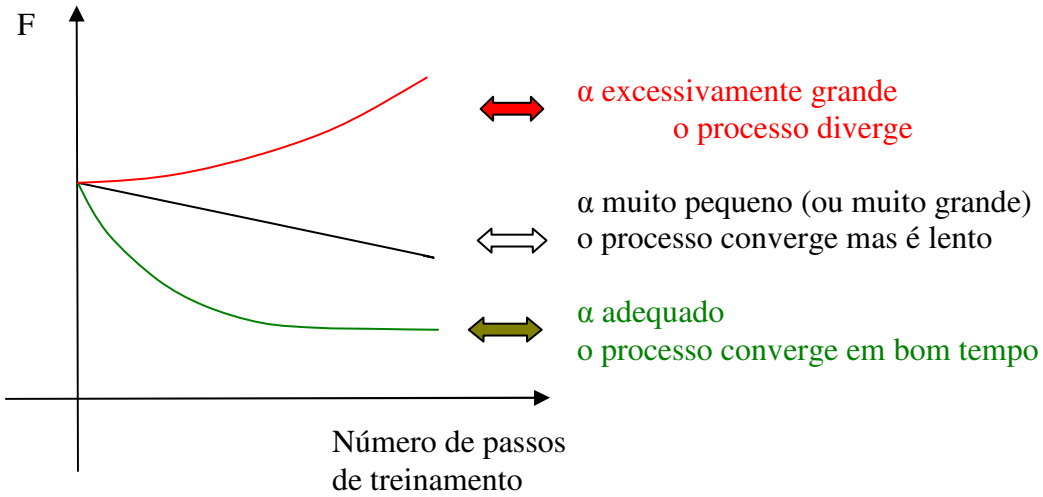
$$|\nabla(w_0 + \Delta w)| > |\nabla(w_0)| \qquad |2(1 - 2\alpha)| > |2| \qquad \alpha_{crítico} = 1$$



O valor adequado de α depende do processo. No treinamento de redes neurais um valores típicos para α estão entre 0,05 e 0,1, mas podem variar de caso para caso, claro.

Evolução do erro ao longo do treinamento:

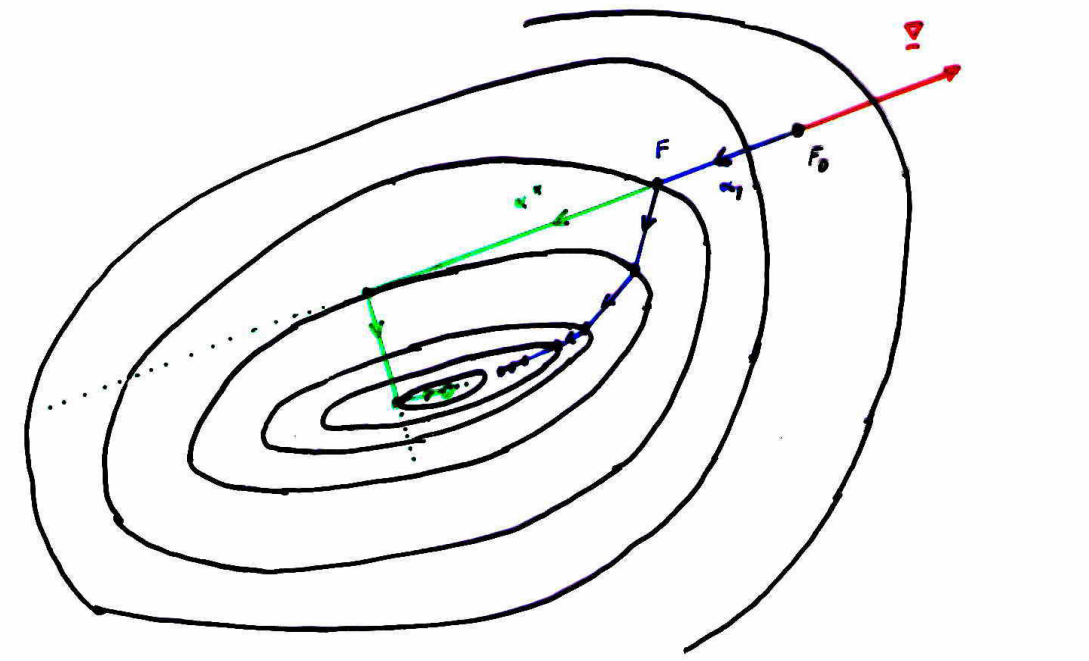
Uma forma prática de verificar a adequabilidade do valor de α é acompanhar a evolução da função objetivo ao longo do treinamento.



O acompanhamento da evolução da função objetivo ao longo do treinamento é fundamental, é a melhor forma de verificar se o processo esta evoluindo bem, determinar o ponto de parada, etc.

Otimização em Linha - Passo Ótimo de Treinamento α^*

Considere que a aproximação de segunda ordem da função é válida. Quando realizamos o deslocamento do ponto de operação ao longo de uma direção $\underline{d}$ adequada o valor da função varia com uma função de segundo grau, decresce, passa por um mínimo e posteriormente cresce. A forma mais eficiente de minimizar a função seria utilizar um α que levasse o ponto de operação para este mínimo da função. Na figura abaixo $\underline{d} = -\underline{\nabla}$; a trajetória azul mostra a descida por gradiente com α constante, e a verde usando α que maximiza o decréscimo da função, **$-\Delta F$.**



Nossa questão agora é: que $\alpha = \alpha^*$ maximiza $-\Delta F$? Este processo é chamado Otimização em linha, porque busca o ótimo sobre uma trajetória ou "linha". Considere que a aproximação de segunda ordem é válida.

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

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

$$F(\underline{w}_0 + \alpha \, \underline{d}) \approx F(\underline{w}_0) + \alpha \, \underline{d}^t \, \underline{\nabla} + \frac{1}{2} \alpha^2 \, \underline{d}^t \, \underline{H} \, \underline{d} \qquad (1)$$

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

Para $\alpha = \alpha^*$ $\Rightarrow$ **$-\Delta F$** é máximo e $\dfrac{\partial F}{\partial \alpha} = \dfrac{\partial \Delta F}{\partial \alpha} = 0$

$$\alpha^* = \underset{\alpha > 0}{\text{Arg}} \left[\text{Min } F(\underline{w}_0 + \alpha \, \underline{d}) \right]$$

$$\frac{\partial}{\partial \alpha} F = \underline{d}^t \, \underline{\nabla} + \alpha \, \underline{d}^t \, \underline{H} \, \underline{d} = 0$$

$$\alpha^* = - \frac{\underline{d}^t \, \underline{\nabla}}{\underline{d}^t \, \underline{H} \, \underline{d}} \qquad (2)$$

O cálculo de α^* necessita de $\underline{H}$, que pode ser complicada de determinar. Existe, entretanto uma maneira de contornar o problema, porque na verdade necessitamos não de $\underline{H}$, mas apenas do valor de $\underline{d}^t \, \underline{H} \, \underline{d}$.

Utilizando $\alpha = \alpha_1$ arbitário aplicamos um deslocamento $\alpha_1 \, \underline{d}$ e calculamos o valor de $F = F_1$ no ponto de operação $\underline{w}_0 + \alpha_1 \, \underline{d}$

$$F_1 = F_0 + \alpha_1 \, \underline{d}^t \, \underline{\nabla} + \frac{1}{2} \alpha_1^2 \, \underline{d}^t \, \underline{H} \, \underline{d}$$

donde

$$\underline{d}^t \, \underline{H} \, \underline{d} = \frac{2}{\alpha_1^2} \, (F_1 - F_0 - \alpha_1 \, \underline{d}^t \, \underline{\nabla}) \qquad (3)$$

aplicando (3) em (2)

$$\alpha^* = - \frac{\alpha_1^2 \, \underline{d}^t \, \underline{\nabla}}{2 \left(F_1 - F_0 - \alpha_1 \, \underline{d}^t \, \underline{\nabla} \right)}$$

e o valor de α^* é determinado sem necessitar do cálculo de $\underline{H}$. O acréscimo (negativo) na função é dado por

$$\Delta F = \frac{1}{2}\,\alpha^*\,\underline{d}^t\,\underline{\nabla}$$

A seguir apresentamos o algorítmo para o caso desenvolvido acima, em que a aproximação de segunda ordem de $F(\underline{w})$ no domínio de busca é válida

Otimização em linha - Algorítmo

á partir do ponto de operação $\underline{w}_n$

calcule $\underline{\nabla}_n$ e $F(\underline{w}_n)$
estabeleça o $\underline{d}_n$ desejado
arbitre α_{teste} suficientemente pequeno
desloque o ponto de operação para $\underline{w}_{teste} = \underline{w}_n + \alpha_{teste}\,\underline{d}_n$
calcule $F(\underline{w}_{teste})$

calcule $\alpha_n = -\dfrac{\alpha_{teste}^2\,\underline{d}^t\,\underline{\nabla}_n}{2\left[F(\underline{w}_{teste}) - F(\underline{w}_n) - \alpha_{teste}\,\underline{d}^t\,\underline{\nabla}_n\right]} > 0$

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

calcule $\underline{\nabla}_{n+1}$ e $F(\underline{w}_{n+1})$

se $F(\underline{w}_{n+1}) - F(\underline{w}_n) \approx \frac{1}{2}\,\alpha_n\,\underline{d}^t\,\underline{\nabla}_n$ e $\underline{d}_n{}^t\,\underline{\nabla}_{n+1} \approx 0$

o ponto encontrado é válido, fim.

caso contrário outros métodos numéricos são necessários

Algumas observações sobre este processo:

1. o cálculo de α^* supõe que a aproximação de segunda ordem de F é válida. Se o α^* encontrado for muito grande provavelmente a aproximação (e o α^* encontrado) não serão válidos.

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

3. A condição usada para calcular α^* leva a extremos, não obrigatóriamente à mínimos. Mostra-se que se $\alpha^* < 0$ o extremo é um máximo, não um mínimo.

4. Se o algorítmo acima não é válido métodos numéricos de busca em linha para funções não quadráticas devem ser usados.

Passo variável controlando o decréscimo de F

O passo variável permite também controlar a forma com que a função objetivo decresce em cada passo de treinamento. No método do gradiente descendente o decréscimo da função é dado por

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

No caso usual

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

Para obter decréscimo ΔF constante usamos

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

Para um decréscimo percentual $\frac{\Delta F}{F}$ constante usamos

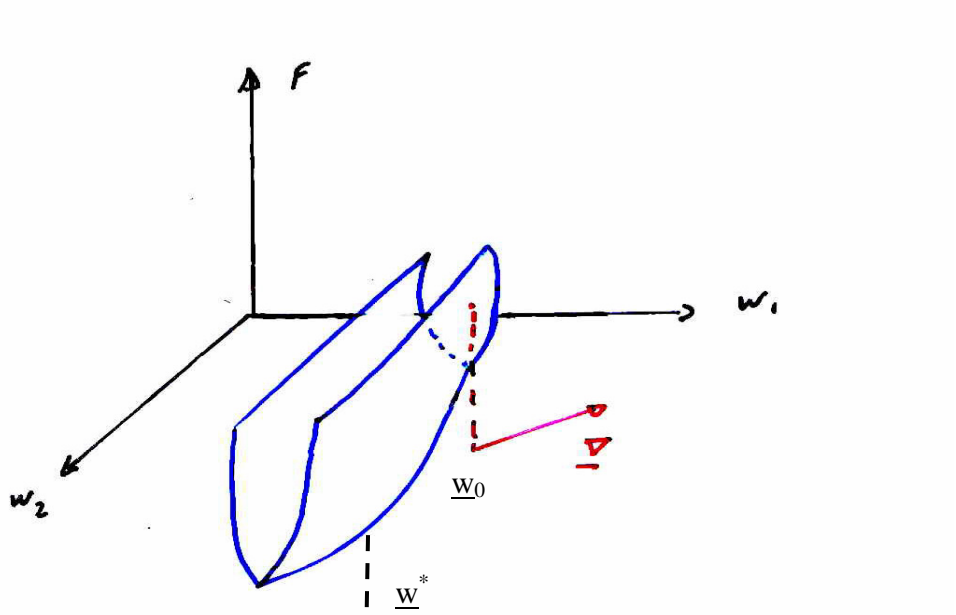
$$\alpha = \alpha_0 \frac{F}{\left\| \underline{\nabla} \right\|^2} \quad \Rightarrow \qquad \frac{\Delta F}{F} = - \alpha_0 \quad \text{ctte}$$

É necessário lembrar, entretanto, que este controle do decréscimo somente é válido enquanto α for suficientemente pequeno para que a aproximação de primeira ordem seja válida.

Passo Variável acelerando a convergência - Retropropagação Resiliente

Acelerando o processo – passo variável por direção

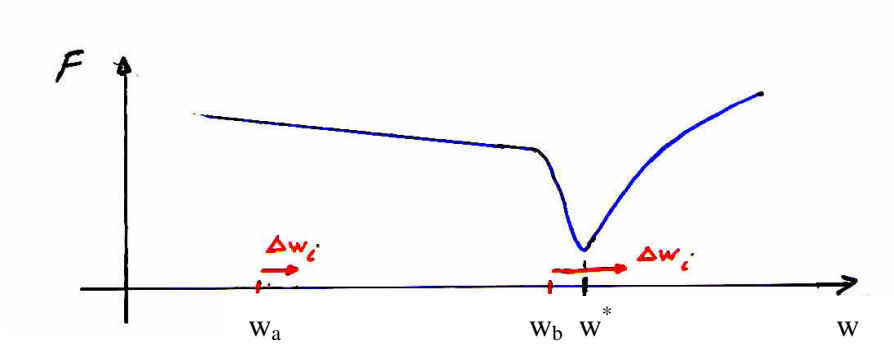
A função esboçada abaixo mostra que em alguns casos pode haver nítida vantagem em utilizar um passo α diferente em cada direção. A figura é uma calha de paredes abruptas com um fundo em parábola com uma leve inclinação. O minimante $\underline{w}^*$ se situa no centro deste fundo. Considere o ponto de operação $\underline{w}_0$. Na direção w_1 o módulo da componente do gradiente $\frac{\partial F}{\partial w_1}$ é grande, e o α utilizado deveria ser pequeno para o deslocamento de $\underline{w}$ não ultrapassar o fundo da calha, onde se encontra o minimante $\underline{w}^*$. Por outro lado, na direção w_2 o módulo da componente do gradiente $\frac{\partial F}{\partial w_2}$ é pequeno, e o α utilizado deveria ser grande para que o minimante $\underline{w}^*$ fosse alcançado mais rapidamente, em menos passos.



Acelerando o processo – passo variável pela forma de $F(w_i)$

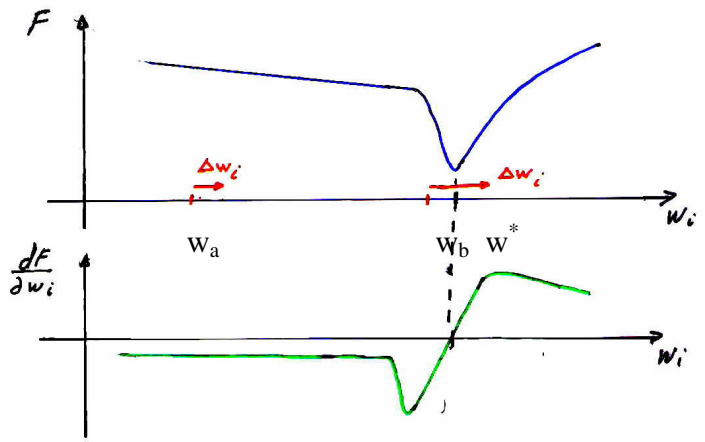
A figura abaixo apresenta a variação de F com um parâmetro w. Observamos que para valores de w no entorno de w_a o gradiente é pequeno; deveríamos então utilizar um

passo α grande para atingir o minimante mais rapidamente. Entretanto, no entorno de w_b o gradiente é grande e o passo α deveria ser pequeno para não ultrapassar $\underline{w}^*$.



Pelas figuras anteriores vemos claramente que utilizar um passo diferenciado por direção e pelas características da função nesta direção pode levar a uma considerável aceleração no processo. Este método e suas variações são conhecidos como Retropropagação resiliente, Super SAB, Método de Silva e Almeida, etc. Apresentamos aqui uma versão com as características mais importantes destes métodos.

Cada direção w_i é tratada de forma independente e tem um passo próprio α_i. O princípio é aumentar o passo α_i enquanto o deslocamento estiver ocorrendo no sentido do minimante, e reduzir α_i quando o minimante for ultrapassado. O deslocamento em uma mesma direção é caracterizado pela manutenção do sinal de dF/dw_i em dois passos consecutivos, e a ultrapassagem do minimante pela troca de sinal de dF/dw_i . A figura abaixo mostra a variação da função e de sua derivada em uma direção w_i., mostrando que esta troca de sinal quando o minimante é ultrapassado. É conveniente fixar um limite superior para α_i.



O algoritmo é apresentado a seguir. Os valores numéricos apresentados na inicialização são adequados para o treinamento tipo retropropagação (backpropagation) de Redes Neurais.

BP Resiliente - Algorítmo

Inicialização

n = 1 $\alpha_{max} \approx .5$ $a \approx 1.05$ $b \approx .9$

para as i = 1, 2, …, V variáveis

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

Até que o critério de parada esteja satisfeito

Passo de treinamento n

Para i = 1, 2, …, V

calcule $\dfrac{\partial F}{\partial w_i}(n)$

$$\alpha_i(n) = \begin{cases} a\,\alpha_i(n-1) & \text{se } \operatorname{sign}\dfrac{\partial F}{\partial w_i}(n) = \operatorname{sign}\dfrac{\partial F}{\partial w_i}(n-1) \\ & \quad mas \text{ se } \alpha_i(n) > \alpha_{max} \; faça \; \alpha_i(n) = \alpha_{max} \\ b\,\alpha_i(n-1) & \text{se } \operatorname{sign}\dfrac{\partial F}{\partial w_i}(n) \neq \operatorname{sign}\dfrac{\partial F}{\partial w_i}(n-1) \end{cases} \qquad (1)$$

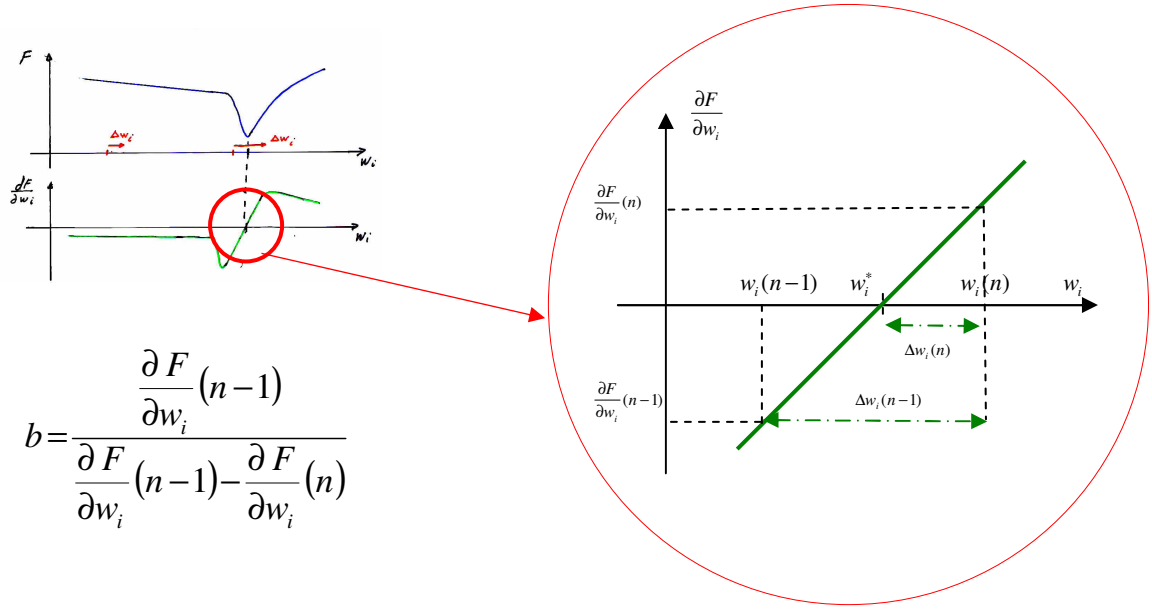
$$\Delta w_i(n) = -\,\alpha_i(n)\,\frac{\partial F}{\partial w_i}(n)$$

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

n = n + 1

Uma alternativa interessante quando muda o sinal da derivada $\dfrac{\partial F}{\partial w_i}$ é considerar que, como estamos próximos do minimante, a aproximação de segunda ordem para a função e a aproximação de primeiras ordem para o gradiente são válidas. A componente

$\frac{\partial F}{\partial w_i}$ do gradiente no entorno do minimante esta esboçada na figura abaixo, o que, usando semelhança de triângulos, nos permite calcular o valor de b que deve levar diretamente ao ponto onde o mesmo é nulo, o minimante $\underline{w}^*$.



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

isto é, no algoritmo do BP Resiliente a linha (1) pode ser substituída pela linha abaixo

$$\alpha_i(n) = \frac{\frac{\partial F}{\partial w_i}(n-1)}{\frac{\partial F}{\partial w_i}(n-1) - \frac{\partial F}{\partial w_i}(n)} \alpha_i(n-1) \qquad \text{se } \operatorname{sign}\frac{\partial F}{\partial w_i}(n) \neq \operatorname{sign}\frac{\partial F}{\partial w_i}(n-1) \qquad (1)$$

Note que para o caso em que o valor das derivadas não muda de sinal também existe uma fórmula análoga que fornece o valor de *a* que leva ao minimante em um único passo, mas somente se a aproximação de segunda ordem for válida. Isto é similar à otimização em linha vista anteriormente. Mas com o sinal das derivadas constante não sabemos se estamos perto do minimante ou não, nem se a aproximação de segunda ordem é válida ou não. Se a aproximação não for válida o resultado encontrado não terá sentido e pode inclusive provocar a divergência do processo. Por exemplo, aplicar esta técnica no

ponto w_a das figuras anteriores levará à um resultado errôneo. Por isto não é recomendável usar este procedimento para determinar um valor de *a* ótimo.

O BP Resiliente representa um ótimo compromisso para associar a simplicidade dos métodos de primeira ordem e a rapidez dos de segunda ordem.

Métodos de 1ª. Ordem (Resumo)

Todos os métodos numéricos de otimização consistem em sucessivamente dar passos $\Delta\underline{w}_n$ no ponto de operação que reduzem a função objetivo F.

$$\Delta\underline{w}_n = \alpha_n \underline{d}_n$$

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

No método do gradiente descendente, o mais simples deles, o deslocamento é dado na mesma direção e sentido contrário ao do gradiente, $\underline{d}_n = -\underline{\nabla}_n$, e o passo é mantido constante, $\alpha_n = \alpha_0 = \text{ctte}_n$. Este método pode ser tornado mais eficiente se o valor ótimo de α_n for determinado através de uma otimização em linha.

No método BP Resiliente α é variável por sinapse e por passo. O BP Resiliente acelera muito o processo mantendo cada passo simples, sendo o mais recomendável.

IX – Otimização: Métodos Numéricos Recursivos de Segunda Ordem

Os métodos de ordem mais elevada são mais sofisticados matematicamente e permitem determinar o minimante com um número bem menor de passos. Entretanto, cada passo é mais complexo que nos métodos de primeira ordem, e existe uma possibilidade maior do processo divergir. A maioria dos métodos de segunda ordem envolve o uso da inversa da Hessiana. O problema é que o cálculo das derivadas de segunda ordem que compõem $\underline{H}$ pode ser trabalhoso, e a inversão de $\underline{H}$ pode ser numericamente mal condicionada, especialmente nos casos em que a dimensão de $\underline{w}$ é muito grande.

Todos os métodos de segunda ordem que veremos consistem em dar um passo $\Delta\underline{w}_n$ no ponto de operação em uma direção $\underline{d}_n$ e utilizar uma otimização em linha para determinar o tamanho do passo, α^*.

$$\Delta\underline{w}_n = \alpha_n \underline{d}_n$$

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

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

Métodos de 2ª ordem - Algorítmo genérico

Inicialização

$\underline{w}_0 = \ldots$ n = 1 ....

Até que o critério de parada esteja satisfeito

Passo de treinamento n

calcule $\underline{d}_n$

otimização em linha

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

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

calcule as variáveis auxiliares eventualmente necessárias

n = n + 1

Os métodos variam apenas na forma como $\underline{d}$ é determinado. Discutiremos aqui os métodos básicos e alguns dos métodos mais conhecidos e/ou utilizados.

Método de Newton ou de Newton Raphson

Como foi visto anteriormente, se a função é quadrática ou se a aproximação quadrática é válida a partir de um ponto $\underline{w}_0$ o minimante pode ser calculado por

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

Esta equação é a base do método de Newton. Como a região em que a aproximação quadrática é válida é limitada usamos em cada passo n a direção dada pela aproximação no ponto

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

mas realizamos uma otimização em linha para determinar o valor de α_n .

$$\alpha^* = \underset{\alpha>0}{\text{Arg}}\,[\text{Min } F(\underline{w}_0 + \alpha\,\underline{d})]$$

Os problemas deste método são que exigem a determinação e a inversão de $\underline{H}_n$.

Métodos Pseudo-Newton

O problema do cálculo e da inversão de $\underline{H}$ é contornado aproximando a Hessiana pela sua diagonal,

$$\underline{\tilde{H}} \approx diag[\underline{H}]$$

que é mais fácil de calcular e inverter, mas que infelizmente na maioria dos casos não é uma aproximação suficientemente boa.

Métodos de Newton amortecidos

Método de Levenberg - Marquadt

Os métodos de Newton amortecidos fornecem ferramentas para a inversão da Hessiana.

O método de Levenberg-Marquadt propõe inicialmente simplificar o cálculo da Hessiana utilizando uma aproximação que evita as derivadas de segunda ordem. Se a função objetivo é um quadrado

$$F(\underline{w}) = \varepsilon^2(\underline{w})$$

$$\frac{\partial F}{\partial w_i} = 2\varepsilon \frac{\partial \varepsilon}{\partial w_i}$$

$$\frac{\partial^2 F}{\partial w_i \, \partial w_j} = 2\frac{\partial \varepsilon}{\partial w_i}\frac{\partial \varepsilon}{\partial w_j} + 2\varepsilon \frac{\partial^2 \varepsilon}{\partial w_i \, \partial w_j} \cong 2\frac{\partial \varepsilon}{\partial w_i}\frac{\partial \varepsilon}{\partial w_j}$$

e então o gradiente e a aproximação $\tilde{H}$ da Hessiana podem ser obtidos apenas com as derivadas de primeira ordem

$$\underline{\nabla} = \frac{\partial \varepsilon^2}{\partial w_i}\Bigg] = 2\varepsilon \frac{\partial \varepsilon}{\partial w_i}\Bigg]$$

$$\underline{H} = \left[\frac{\partial^2 \mathrm{F}}{\partial \mathrm{w_i}\, \partial \mathrm{w_j}}\right] \cong \left[2\frac{\partial \varepsilon}{\partial \mathrm{w_i}} . \frac{\partial \varepsilon}{\partial \mathrm{w_j}}\right] = \underline{\tilde{H}}$$

O processo também vale para o caso em que F é uma soma de M quadrados $\varepsilon_m^2(\underline{w})$, m = 1, ..., M

$$F(\underline{w}) = \sum_{m=1}^{M} \varepsilon_m^2(\underline{w})$$

E neste caso o gradiente e a aproximação da Hessiana são dados por

$$\underline{\nabla} = 2\sum_{m=1}^{M} \varepsilon_m \frac{\partial \varepsilon_m}{\partial w_i} \Bigg]$$

$$\underline{H} = \left[\frac{\partial^2 \mathrm{F}}{\partial \mathrm{w}_\mathrm{i}\, \partial \mathrm{w}_\mathrm{j}} \right] \cong \left[2\sum_{m=1}^{M} \frac{\partial \varepsilon_m}{\partial \mathrm{w}_\mathrm{i}} . \frac{\partial \varepsilon_m}{\partial \mathrm{w}_\mathrm{j}} \right] = \underline{\tilde{H}}$$

Note que o gradiente e a aproximação da Hessiana são calculados para cada função $\varepsilon_p^2(\underline{w})$ de forma independente, e em seguida somados para obter o gradiente e a aproximação da Hessiana de F(w). Este caso será útil na sessão seguinte, Ajuste de Parâmetros, onde as funções $\varepsilon_p^2(\underline{w})$ serão os "erros instantâneos" para cada par entrada-saída..

Resta agora o problema da inversão da aproximação da Hessiana, $\tilde{H}$. Este problema é resolvido com uma segunda aproximação da Hessiana a partir de $\tilde{H}$. $\tilde{H}$ é aproximada por uma matriz diagonal dominante (ou próxima disto) $\tilde{\tilde{H}}$ tal que sua inversão seja possível, numericamente bem condicionada. Uma matriz $[h_{ij}]$ é dita diagonal dominante se para todas as suas linhas i (ou colunas) $h_{ii} \geq \sum_{\forall j \neq i} \left| h_{ij} \right|$. Matrizes diagonal dominantes são facilmente inversíveis. A aproximação proposta é

$$\underline{H} \approx \underline{\tilde{H}} \approx \underline{\tilde{\tilde{H}}} = \left[\underline{\tilde{H}} + \lambda \underline{I} \right]$$

onde $\underline{I}$ é a matriz identidade e λ um número positivo suficientemente grande para garantir a inversão de $\underline{\tilde{\tilde{H}}}$ mas também suficientemente pequeno para que a aproximação seja válida. A medida em que o processo evolui e a aproximação de segunda ordem começa a ser mais

precisa o valor de λ vai sendo reduzido para que as aproximações sejam ainda mais precisas. Mas reduzir λ muito rapidamente para acelerar o processo pode tornar a inversão de $\underline{\tilde{\tilde{H}}}$ numericamente mal condicionada e fazer todo o processo divergir.

Obtida a inversa da Hessiana aproximada, $\underline{\tilde{\tilde{H}}}^{-1}$, o restante do processo é convencional,

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

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

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

O algorítmo será apresentado na Parte II, para a aplicação em Ajuste de Parâmetros

Métodos Quase Newton

Método BFGS: Broyden, Fletcher, Goldfarb e Shanno

Os métodos quase Newton contornam o problema da inversão tentando obter diretamente uma aproximação para $\underline{H}^{-1}$. A idéia básica é obter uma aproximação $\underline{G}$ para $\underline{H}^{-1}$ á partir da fórmula da variação do gradiente com o acréscimo no ponto de operação.

Fórmula básica

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

$$(\underline{w}_{n+1} - \underline{w}_n) = \underline{G}(\underline{\nabla}_{n+1} - \underline{\nabla}_n) \qquad onde \ \ \underline{G} \cong \underline{H}_n^{-1}$$

A partir daí é possível calcular uma fórmula recursiva para $\underline{G}$

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

Uma fórmula alternativa para $\underline{G}_{n+1}$ é proposta por Barnes e Rosen:

$$\underline{G}_{n+1} = \underline{G}_n + \frac{\left(\underline{d}_n - \underline{G}_n \underline{y}_n\right)\left(\underline{d}_n - \underline{G}_n \underline{y}_n\right)^t}{\left(\underline{d}_n - \underline{G}_n \underline{y}_n\right)^t \underline{y}_n} \quad onde \quad \underline{y} = \underline{\nabla}_{n+1} - \underline{\nabla}_n$$

O algorítmo para o método é:

BFGS- Algorítmo

Inicialização

$\underline{w}_0 = \ldots$ n = 1

calcule

$\underline{\nabla}_0 = \ldots \quad \underline{d}_0 = -\underline{\nabla}_0 \quad \underline{G}_0 = \underline{I}$

faça a otimização em linha

$$\alpha_0 = \underset{\alpha>0}{\text{Arg}}\left[\text{Min } F(\underline{w}_0 + \alpha \underline{d}_0)\right]$$

$$\underline{w}_1 = \underline{w}_0 + \alpha_0 \underline{d}_0$$

Até que o critério de parada esteja satisfeito

Passo de treinamento n

calcule

$$\underline{\nabla}_n = \ldots$$

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

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

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

faça a otimização em linha

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

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

n = n + 1

Método do Gradiente Conjugado

Método de Fletcher – Reeves

O método do gradiente conjugado não usa a Hessiana, e para V variáveis w_i teóricamente exige apenas V passos para convergir.

Método do Gradiente Conjugado - Algorítmo

Inicialização

$\underline{w}_0 = \ldots$ n = 1

calcule

$\underline{\nabla}_0 = \ldots \quad \underline{d}_0 = -\underline{\nabla}_0$

faça a otimização em linha

$$\alpha_0 = \underset{\alpha>0}{\text{Arg}}\left[\text{Min F}(\underline{w}_0 + \alpha\,\underline{d}_0)\right]$$

$$\underline{w}_1 = \underline{w}_0 + \alpha_0 \underline{d}_0$$

Até que o critério de parada esteja satisfeito

Passo de treinamento n

calcule

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

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

faça a otimização em linha

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

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

n = n + 1

A fórmula acima para o cálculo β_n foi proposta por Leonard e Kramer. Uma fórmula alternativa foi proposta por Polak e Ribière

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

Métodos de 2ª. Ordem (Resumo)

Todos os métodos numéricos de otimização consistem em sucessivamente dar passos $\Delta\underline{w}_n$ no ponto de operação que reduzem a função objetivo F.

$$\Delta\underline{w}_n = \alpha_n \underline{d}_n$$

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

Nos métodos de segunda ordem a direção do deslocamento $\underline{d}$ é uma aproximação envolvendo a inversa da Hessiana

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

e o passo α é obtido por uma otimização em linha.

$$\alpha^* = \underset{\alpha>0}{\text{Arg}} \left[\text{Min } F(\underline{w}_0 + \alpha \underline{d})\right]$$

Todos os métodos práticos analisados utilizam apenas derivadas de primeira ordem. O método de Levenberg-Marquadt (Newton amortecido) utiliza duas aproximações sucessivas da Hessiana, a primeira utilizando apenas derivadas de primeira ordem e a segunda para permitir a inversão na matriz. O método BFGS (quase Newton) obtém diretamente uma aproximação da inversa da Hessiana, contornando o problema da inversão da matriz. Finalmente, o método do Gradiente Conjugado não utiliza a Hessiana e realiza a busca em princípio com um número pré determinado de passos.

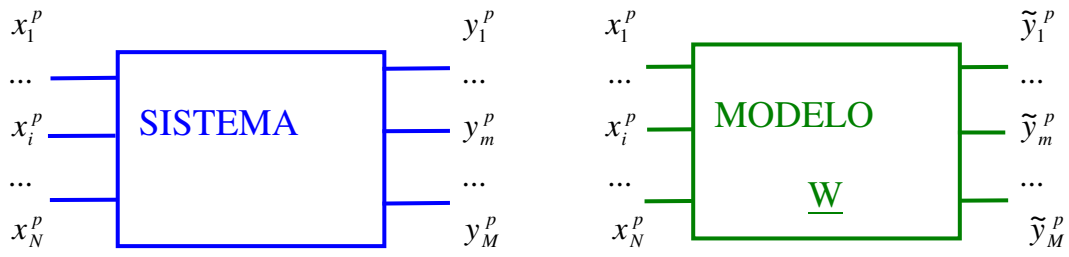
Parte II

Ajuste dos Parâmetros (ou treinamento, ou aprendizado) de um Modelo como um processo de otimização

I Sistemas e Modelos

Nas ciências da terra e nas engenharia podemos dizer que conhecemos um sistema quando conhecemos um modelo matemático para o mesmo. Um sistema físico recebe variáveis em sua entrada (que representamos por um vetor de entrada $\underline{x}$) e gera variáveis em sua saída (que representamos por um vetor de saída $\underline{y}$) . É portanto um mapeador do espaço de entrada, o domínio de $\underline{x}$, $D_{\underline{x}}$, no espaço de saída, o domínio de $\underline{y}$, $D_{\underline{y}}$. Usualmente $\underline{x}$ e $\underline{y}$ são multidimensionais e N = dim ($\underline{x}$) > dim ($\underline{y}$) = M. Cada entrada específica $\underline{x}^p$ gera uma saída específica $\underline{y}^p$. Cada conjunto ($\underline{x}^p$; $\underline{y}^p$) constitui um par entrada-saída do sistema. O conjunto de todos os P pares entrada-saída conhecidos define numericamente o mapeamento $\underline{x} \rightarrow \underline{y}$ realizado pelo sistema.

Um modelo é um conjunto de relações (em nosso caso serão equações) tais que recebendo como entrada um vetor $\underline{x}^p$ gera como saída um vetor $\underline{\tilde{y}}^p$ que deve ser uma aproximação aceitável de $\underline{y}^p$. E estas equações tem associado a elas um conjunto de parâmetros $\underline{w}$.



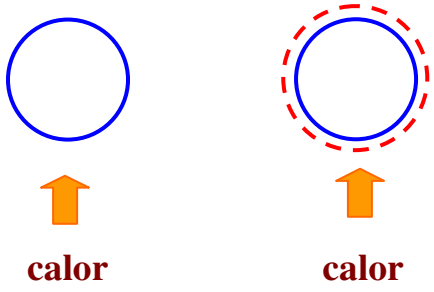
$$\underline{y} = \varphi(\underline{x}) \qquad\qquad \underline{\tilde{y}} = \tilde{\varphi}(\underline{w}, \underline{x})$$

$$onde \quad \underline{y} \cong \underline{\tilde{y}} \quad \forall \quad \underline{x} \in D_{\underline{x}}$$

Se conhecemos a fenomenologia do sistema, isto é, a forma como o sistema processa as variáveis de entrada (e também suas eventuais variáveis internas) então montamos o sistema de equações do modelo de forma a reproduzir matematicamente este processamento. Este modelo é chamado de "caixa branca" porque conhecemos as equações no seu interior e o modo como estão ligadas ao sistema. O ajuste do modelo, neste caso, consiste em ajustar seus parâmetros $\underline{w}$.

Há casos em que não conhecemos a maneira como o sistema processa as variáveis, conhecemos apenas o mapeamento entrada-saída dado pelos P pares entrada-saída. Neste caso utilizamos um conjunto de equações capaz de realizar qualquer mapeamento (um aproximador universal) ajustando seus parâmetros baseado apenas nos pares entrada-saída disponíveis. Este será um modelo tipo "caixa preta".

Considere um exemplo trivial: uma quantidade de gás ideal está contida em um recipiente indeformável. Sabemos que a lei de Boltzmann rege a relação entre a temperatura T (em ^{o}C), a pressão P (em N/m^2) e o volume V (em m^3) do gás conforme a equação $PV = Nk(T + T_0)$. Se quisermos construir um modelo que nos informe a pressão dada uma determinada temperatura podemos usar a equação linear $P = w_1 T + w_0$. Os parâmetros w_0 e w_1 podem ser calculados conhecidos o número N de átomos de gás contidos no recipiente, a constante k de Boltzmann e o valor do zero absoluto T_0 em ^{o}C. Ou ajustados a partir do conhecimento de alguns pares entrada-saída, (T^p , P^p). Este modelo é uma caixa branca.



Continuemos com nosso exemplo trivial, mas considere agora que o gás ideal está contido em um recipiente de plástico maleável que se expande em função da pressão interna e da temperatura, de uma forma complexa da qual não conhecemos os detalhes,

$V = \Phi(T, P)$. Sabemos que a lei de Boltzmann $PV = Nk(T + T_0)$ continua válida, mas agora não somos mais capazes de escrever um sistema de equações simples que relacione P com T. Entretanto, se conhecermos alguns pares entrada-saída (T^p , P^p) e algum sistema de equações suficientemente genérico (por exemplo, uma rede neural feedforward adequadamente dimensionada) podemos ajustar os parâmetros do siatema de equações (as sinapses da rede neural, no caso) e obter um modelo que nos fornece o mapeamento $P = \varphi(T)$. Este modelo será uma caixa preta.

Modelos que mesclam conhecimento fenomenológico e estruturas empíricas são chamados caixas cinza.

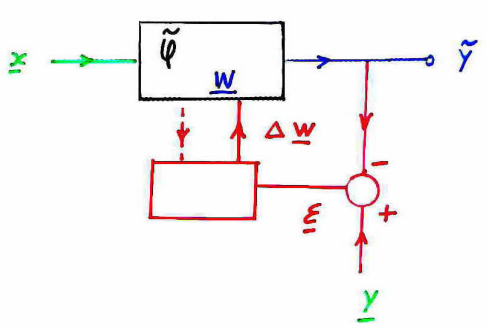
II - Treinamento ou Ajuste dos Parâmetros

Qualquer que seja o modelo consideramos que no processo de ajuste dos parâmetros de forma a casar o modelo ao sistema já conhecemos a arquitetura do modelo, i.e., as equações que o definem, mas não conhecemos o valor dos parâmetros w_i destas equações. O ajuste dos parâmetros do modelo pode ser visto como um problema de otimização: encontrar o vetor $\underline{w}$ que minimiza uma função objetivo que meça a discrepância entre a saída do sistema e a saída do modelo, para todos os pontos do domínio da entrada, $D_{\underline{x}}$.

$$\textit{Sistema}: \quad \underline{y} = \varphi(\underline{x}) \qquad\qquad \textit{Modelo}: \quad \tilde{\underline{y}} = \tilde{\varphi}(\underline{w}, \underline{x})$$

$$\underline{w} = Arg\ Min\ Discrepância\left[\underline{y}(\underline{x}),\ \tilde{\underline{y}}(\underline{x}, \underline{w})\right] \quad \forall\ \underline{x} \in D_x$$

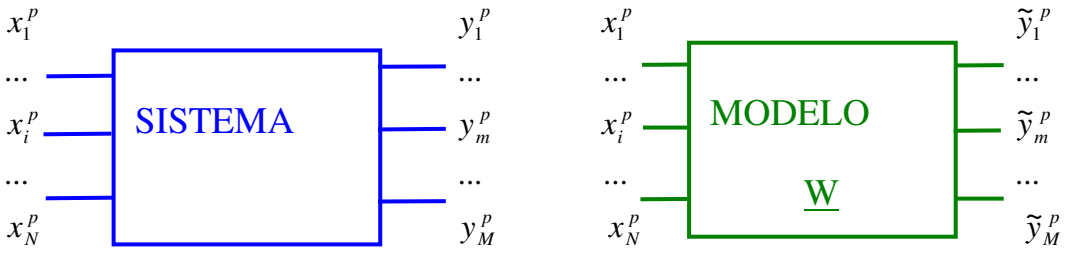
O princípio do ajuste dos parâmetros do modelo é simples: aplicamos ao modelo e à planta a mesma entrada $\underline{x}$ e comparamos as saídas. A diferença ou erro entre elas é utilizada em um algorítmo que altere o valor dos parâmetros do



modelo de modo a reduzir a discrepância entre as saídas para aquela entrada. A aplicação deste processo repetidas vezes e com diferentes entradas ajustará os parâmetros para produzir um erro pequeno para qualquer entrada.

Erro na saída - Função objetivo à minimizar

Para cada entrada $\underline{x}^p$ a planta produz uma saída $\underline{y}^p$. Cada par numérico ($\underline{x}^p$, $\underline{y}^p$), p = 1, 2, ..., P é chamado um par entrada-saída da planta. O conjunto de todos os P pares entrada-saída define numericamente a relação funcional entrada-saída da planta que queremos reproduzir no modelo. O modelo, por sua vez, para cada uma das entradas $\underline{x}^p$ produz uma saída $\underline{\tilde{y}}^p$ que deve ser uma aproximação suficientemente boa da saída da planta, $\underline{y}^p$. A discrepância entre $\underline{y}^p$ e $\underline{\tilde{y}}^p$ é o erro que queremos minimizar ao ajustar os parâmetros $\underline{w}$ do modelo. A discrepância ou erro entre dois vetores pode ser definida de diversas maneiras, sendo a mais comum o erro quadrático $\left(\varepsilon^p\right)^2$:



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

onde M é a dimensão de $\underline{y}$ e ε_m^p é o erro na m-ésima saida

$$\varepsilon_m^p = y_m^p - \tilde{y}_m^p \qquad \textit{onde} \quad y_m^p = y_m^p\left(\underline{x}^p\right) \quad \textit{e} \quad \tilde{y}_m^p = \tilde{y}_m^p\left(\underline{w}, \underline{x}^p\right)$$

$\left(\varepsilon^p\right)^2$ é chamado de erro quadrático “instantâneo” (em contraposição ao erro quadrático médio) porque se refere apenas a um par. Como queremos minimizar o erro

para todos os P pares entrada-saída podemos escolher para função objetivo F(w) à ser minimizada o erro quadrático médio (ou erro médio quadrático) entre a saída da planta e a do modelo para todos os P pares que caracterizam a planta.

$$F(\underline{w}) = \underset{p}{E}\left(\varepsilon^p\right)^2 = \frac{1}{P}\sum_{p=1}^{P}\left(\varepsilon^p\right)^2$$

Cálculo do Gradiente

Qualquer que seja o processo de busca do minimante que formos utilizar precisaremos das derivadas de primeira ordem de F em relação à cada parâmetro w_i. Mas como calcular a derivada de uma função objetivo aparentemente complexa F(w) em relação à cada um dos seus parâmetros w_i ?

$$\frac{\partial}{\partial w_i} F(\underline{w}) = \frac{\partial}{\partial w_i} \underset{p}{E}\left(\varepsilon^p\right)^2$$

Este cálculo é uma mera questão algébrica. Considere inicialmente que:

$$\frac{\partial}{\partial w_i} \underset{p}{E}\left(\varepsilon^p\right)^2 = \frac{\partial}{\partial w_i} \frac{1}{P}\sum_{p=1}^{P}\left(\varepsilon^p\right)^2 = \frac{1}{P}\sum_{p=1}^{P}\frac{\partial}{\partial w_i}\left(\varepsilon^p\right)^2 = \underset{p}{E}\frac{\partial}{\partial w_i}\left(\varepsilon^p\right)^2$$

O que a expressão acima nos diz é que o gradiente do valor esperado do erro quadrático é o valor esperado do gradiente do erro quadrático "instantâneo", isto é, o gradiente do erro quadrático para cada par isoladamente.

$$\underline{\nabla}\,\underset{p}{E}\left(\varepsilon^p\right)^2 = \underset{p}{E}\,\underline{\nabla}\left(\varepsilon^p\right)^2$$

Isto significa que podemos calcular o gradiente do erro quadrático para cada par independentemente dos demais e tomar a média para obter o gradiente de F(w). Nosso problema esta portanto simplificado para calcularmos, para cada par p,

$$\frac{\partial}{\partial w_i}\left(\varepsilon^p\right)^2$$

Continuando o desenvolvimento

$$\frac{\partial}{\partial w_i}\left(\varepsilon^p\right)^2 = \frac{\partial}{\partial w_i}\sum_{m=1}^{M}\left(\varepsilon_m^p\right)^2 = \sum_{m=1}^{M}\frac{\partial}{\partial w_i}\left(\varepsilon_m^p\right)^2 =$$

$$= 2\sum_{m=1}^{M}\varepsilon_m^p\frac{\partial}{\partial w_i}\varepsilon_m^p \qquad onde \qquad \varepsilon_m^p = y_m^p - \tilde{y}_m^p$$

$$= -2\sum_{m=1}^{M}\varepsilon_m^p\frac{\partial\,\tilde{y}_m^p}{\partial w_i}$$

onde o termo $-2\varepsilon_m^p$ deriva da **formula do erro** usada, no nosso caso o e.m.q.

e o termo $\frac{\partial\,\tilde{y}_m^p}{\partial w_i}$ depende das **equações do modelo** usado.

III - Gradiente Descendente - Cálculo do Acréscimo

Para um único par entrada-saída o acréscimo a ser aplicado em cada w_i pelo processo do gradiente descendente seria

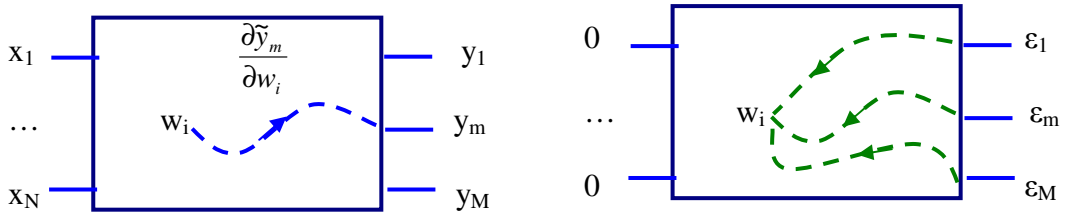
$$\Delta w_i^p = -\alpha\frac{\partial}{\partial w_i}\underset{p=1}{E}\left(\varepsilon^p\right)^2 = -\alpha\frac{\partial}{\partial w_i}(\varepsilon)^2 = 2\alpha\sum_{m=1}^{M}\varepsilon_m\frac{\partial\,\tilde{y}_m}{\partial w_i}$$

$$\Delta w_i^p = 2\alpha\delta_i^p \qquad onde \qquad \delta_i^p = \sum_{m=1}^{M}\varepsilon_m\frac{\partial\,\tilde{y}_m}{\partial w_i}$$

Interpretemos agora δ_i^p. O termo $\frac{\partial\,\tilde{y}_m}{\partial w_i}$ mede a relação entre pequenas variações que ocorrem na saída $\tilde{y}_m$ provocadas por pequenas variações introduzidas no parâmetro w_i. É chamado ganho para pequenos sinais de w_i até $\tilde{y}_m$: é o ganho de w_i até $\tilde{y}_m$ se o modelo tivesse sido linearizado, isto é, se todos os seus elementos não lineares tivessem sido substituidos pelas derivadas no ponto de operação dado pela entrada x.. Considere

agora que o modelo foi linearizado, e que o sentido de transmissão de todos os seus componentes foi invertido. Podemos chamar esta estrutura de "Modelo Associado". δ_i^p é o sinal que obteríamos em w_i se alimentássemos as saídas do modelo associado com os erros que apareceram nas saídas do modelo original, isto é, se fizéssemos os erros retornarem das saídas até o parâmetro em estudo. Por este motivo δ_i^p é chamado de erro retropropagado, o que deu à este método o nome de "retropropagação do erro" quando aplicado em redes neurais.

Modelo Original Modelo Associado



Chamemos o acéscimo a ser aplicado em w_i considerando apenas o par p de $\Delta w_i^p = 2\alpha\delta_i^p$. Para os P pares o acréscimo a ser aplicado ao parâmetro é a média dos acréscimos calculados para cada par

$$\Delta w_i = -\alpha \frac{\partial}{\partial w_i} \underset{p}{E}\left(\varepsilon^p\right)^2 = \underset{p}{E}\,\Delta w_i^p = 2\alpha \underset{p}{E}\,\delta_i^p$$

$$\Delta w_i = 2\alpha \frac{1}{P} \sum_{p=1}^{P} \sum_{m=1}^{M} \varepsilon_m^p \frac{\partial\, \tilde{y}_m^p}{\partial\, w_i}$$

IV - Gradiente Descendente - Formas de Aplicação dos Acréscimos

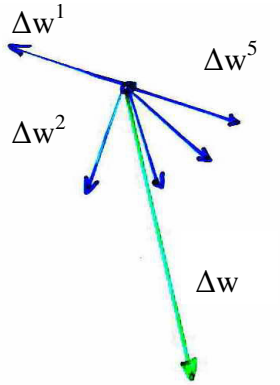
Processo em Batelada – Batch

O processamento descrito acima é chamado de batelada. Neste processamento em batelada inicialmente cada par entrada saída é processado de forma independente, calculado-se o acréscimo para cada parâmetro w_i , Δw_i^p , devido apenas a este par, como feito acima.

Em seguida, aplica-se em cada parâmetro w_i a média dos acréscimos computados para para ele com cada par. No passo n

$$\Delta w_i(n) = \underset{p}{E}\,\Delta w_i^p$$

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



Geometricamente o processo é exemplificado na figura ao lado com cinco pares: cada par p indica a direção ótima de deslocamento $\Delta\underline{w}^p$ para minimizar o seu erro quadrático instantâneo (em azul), e a direção ótima $\Delta\underline{w}$ para minimizar o emq (em verde) é a média das direções para minimizar os erros quadráticos instantâneos.

Se o número de pares é muito grande (P >> 200) o processo pode se tornar muito lento. Neste caso pode-se (1) reduzir o número de pares entrada-saída que serão utilizados no treinamento, mas tomando-se o cuidado de não perder a generalização ou (2) empregar o processo de treinamento por lotes, apresentado à seguir

Batelada - Algorítmo

Até que o critério de parada seja satisfeito

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

calcular $\underline{\tilde{y}}^p = \varphi\left(\underline{x}^p\right)$

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

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

outro par

calcular $\Delta w_i = -\alpha \mathop{E}_{p} \frac{\partial \varepsilon^{p2}}{\partial w_i}$

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

reiniciar

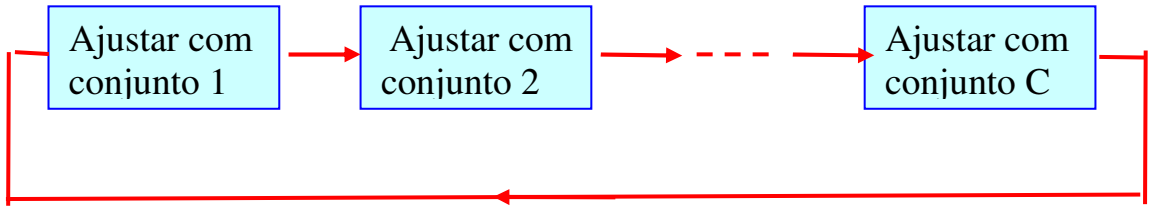
Processo por Lotes

Se o número de pares entrada-saída for muito grande (P >> 200) é possível descartar alguns pares para reduzir o tempo de ajuste, mas com este descarte sempre se alguma generalização.

Uma alternativa é alocar os pares aleatoriamente em C conjuntos, de forma que cada conjunto contenha cerca de 200 pares.

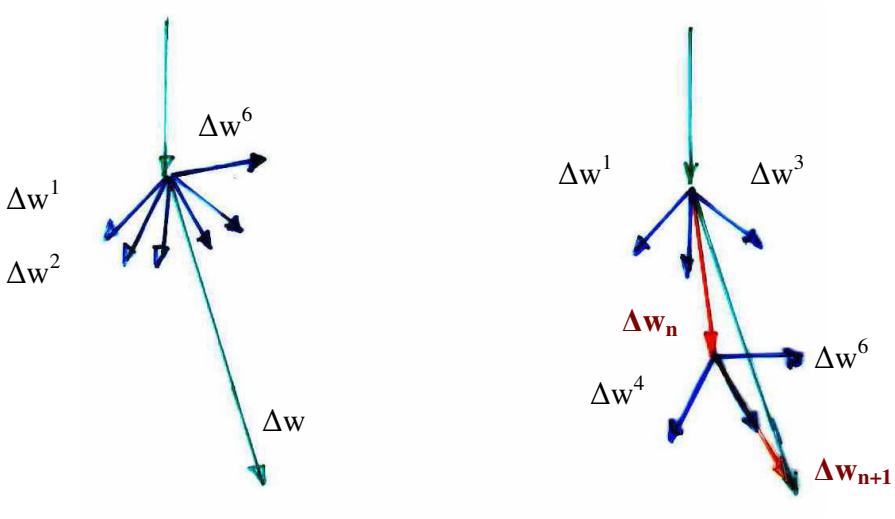
$C \approx P / 200$

Realiza-se o ajuste dos parâmetros em batelada com o primeiro conjunto, em seguida com o segundo, até o C-ésimo conjunto, quando reiniciamos o processo com o conjunto 1.



Este processo permite reduzir o tempo de ajuste conservando a generalização oferecida por todos os pares entrada-saída disponíveis.

Geometricamente o processo é exemplificado na figura abaixo lado com seis pares entrada-saída. Na figura a esquerda, no processamento em batelada única, o deslocamento do ponto de operação é o valor médio (em verde) da direção ótima para cada um dos seis pares (em azul), mas só ocorre após a aplicação dos seis pares. Na figura à direita, se dividirmos em dois lotes o deslocamento do ponto de operação (em vermelho) ocorre a cada tres pares, mais rápido.



BATELADA LOTES

Lotes - Algorítmo

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

Processo em "Regra Delta"

A regra delta é um processo por lotes com apenas um par por lote, isto é, à cada par entrada-saída apresentado o ponto de operação é atualizado o que torna o processo muito rápido.

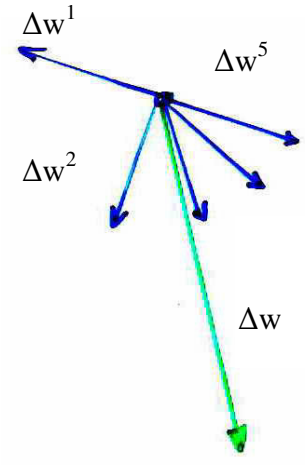
No processo regra delta no passo n esta disponível um par entrada saída. Calculamos o acréscimo que seria aplicado por regra delta em cad parâmetro w_i devido a este par e o aplicamos

$$\Delta w_i\,(n) = -\alpha \frac{\partial}{\partial w_i} \varepsilon^2$$

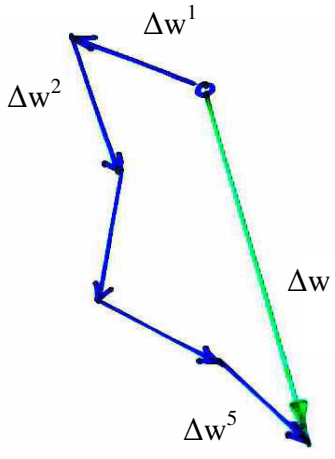
$$w_i\,(n+1) = w_i\,(n) + \Delta w_i\,(n)$$

Como o gradiente instantãneo de um par pode diferir muito do gradiente do emq a trajetoria oscila bastante. Embora em média F decresça, pode inclusive crescer em determinados passos.

A figura abaixo esboça a evolução do ponto de operação de um processo em batelado com um em regra delta, para cinco pares. Se o passo é pequeno os dois processos resultam em pontos finais muito próximos.



BATELADA



REGRA DELTA

A regra delta é utilizada em aplicações on line, quando o sistema é variante no tempo e a atualização em tempo real é necessária, e também quando não há muita memória disponível para estocar os pares entrada-saída.

Regra Delta - Algorítmo

Até que o critério de parada seja satisfeito

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

calcular $\underline{\tilde{y}}^p = \varphi\left(\underline{x}^p\right)$

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

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

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

outro par

Processo em Regra Delta com Momento

O momento é um processo utilizado normalmente junto a regra delta e serve para reduzir as oscilações de trajetória mantendo a atualização do ponto de operação a cada par entrada-saída aplicado. O momento associa a rapidez e necessidade de pouca memória da regra delta com a estabilidade de trajetória da batelada.

No processo de regra delta com momento no passo n esta disponível um par entrada saída. Calculamos o acréscimo que seria aplicado por regra delta em cada parâmetro w_i devido a este par:

$$\Delta w_i^{RD}(n) = -\alpha \frac{\partial}{\partial w_i} \varepsilon^2$$

Mas o acréscimo que é aplicado é uma media ponderada entre este acréscimo e o acréscimo aplicado no passo anterior:

$$\Delta w_i(n) = \beta \Delta w_i(n-1) + (1-\beta)\Delta w_i^{RD}(n) \qquad 0 < \beta < 1$$

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

A preservação de parte do acréscimo anterior mantém parte da trajetória anterior do ponto de operação e lembra o momento de inércia da mecânica. Daí o nome momento.

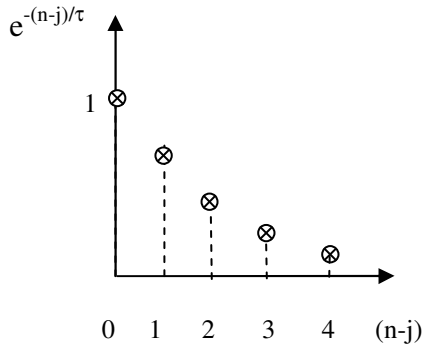
Para n grande pode-se mostrar que o acréscimo $\Delta w_i(n)$ aplicado em n é a média dos acréscimos "regra delta" ocorridos nos passos anteriores j , $\Delta w_i^{RD}(j)$, ponderados exponencialmente pelo atraso (n-j) com que ocorreram.

$$\Delta w_i(n) \cong (1-\beta) \sum_{j=1}^{n} \beta^{n-j} \Delta w_i^{RD}(j)$$

A contribuição de cada par é ponderado por β^{n-j}, isto é, decai exponencialmente com uma constante de tempo τ em relação ao atraso (n-j) com que ocorreu. A figura abaixo mostra a forma do decaimento do ponderador β^{n-j}.

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

$$onde \quad \tau = -\frac{1}{\ln\beta} \approx \left.\frac{1}{1-\beta}\right|_{\beta\cong 1}$$



Após quatro constantes de tempo a contribuição do termo pode ser negligenciada. Desta forma o processo regra delta com momento pode ser considerado como um processo por lotes com $N_0 \approx 4\tau$ pares por lote. O número de pares é controlado por β.

$$N_0 \approx 4\tau \approx \frac{4}{1-\beta} \qquad ou \qquad \beta = 1 - \frac{4}{N_0}$$

Valores típicos são $N_0 \approx 40$ e $\beta \approx 0{,}9$. Como mencionado anteriormente a regra delta é utilizada em aplicações on line, quando o sistema é variante no tempo e a atualização em tempo real é necessária. O momento suaviza a trajetória do ponto de operação. Quanto maior β mais suave é a trajetória, mas mais tempo leva para que o modelo acompanhe uma mudança no sistema.

Regra Delta com Momento - Algorítmo

Inicialização n=1; $\Delta w_i^{RD}(0) = 0 \quad \forall\, i$

Até que o critério de parada seja satisfeito

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

calcular

$$\underline{\tilde{y}}^{\,p} = \varphi\left(\underline{x}^{\,p}\right)$$

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

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

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

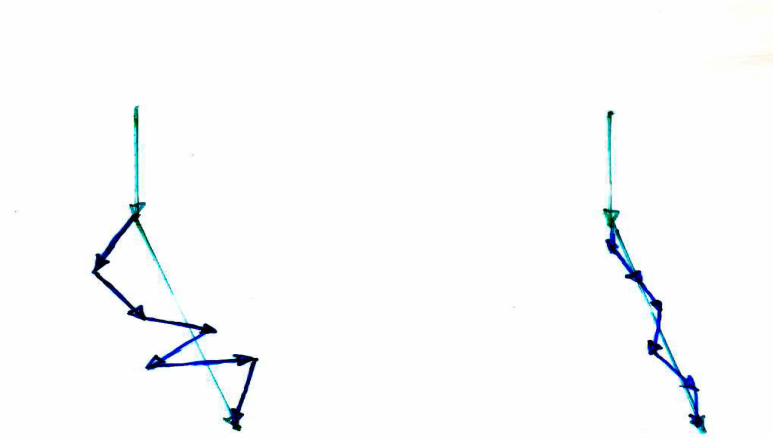
$$w_i(n+1) = w_i(n) + \Delta w_i(n) \quad \forall i$$

$$n = n + 1$$

outro par

A figura abaixo esboça trajetórias de ajuste em batelada, lotes, regra delta com momento e sem momento:

Batelada Lotes



Regra Delta sem momento

Regra Delta com momento

Método BP Resiliente em Ajuste de Parâmetros de um Modelo.

Na página seguinte apresentamos o algorítmo para o uso do método de backpropagation resiliente em ajuste dos parâmetros de um modelo. Os parâmetros propostos são adequados para o treinamento de redes neurais feedforward.

Método de Levemberg-Marquadt em Ajuste de Parâmetros de um Modelo.

O método de Levemberg-Marquadt tornou-se muito popular após sua inclusão no tool box de redes neurais do Matlab. Duas páginas à seguir apresentamos o algorítmo para o uso deste método no ajuste dos parâmetros de um modelo. Os parâmetros propostos são adequados para o treinamento de redes neurais feedforward.

BP Resiliente - Algorítmo

Inicialização

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

para as i = 1, 2, …, V variáveis

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

Até que o critério de parada esteja satisfeito

Passo de treinamento n

Cálculo do gradiente

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

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

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

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

outro par

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

Cálculo do acréscimo

para i = 1, 2, …, V

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

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

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

n = n + 1

Levenberg- Marquadt - Algorítmo

Inicialização

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

Até que o critério de parada esteja satisfeito

Passo de treinamento n

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

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

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

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

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

outro par

calcular $$\frac{\partial F}{\partial w_i}(n) = g_{i\,(n)} = \underset{p}{E} \frac{\partial \varepsilon^{p2}}{\partial w_i}$$

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

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

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

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

realize a otimização em linha

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

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

n = n + 1

Parte 3 - Comentários Finais

I - Outras funções objetivo - modificações

Composição do gradiente

No ajuste de um modelo a função objetivo F(.) a ser minimizada é uma composição para todos os pares entrada-saída de uma função f(.) que mede a dissimilaridade para cada par entre o valor obtido $\underline{\tilde{y}}$ e valor desejado $\underline{y}$ para a saída. Usualmente esta composição é o valor esperado de f(.) para todos os pares entrada-saída e a função dissimilaridade f(.) é o erro quadrático na saída.

$$F(\underline{w},\underline{y},\underline{\tilde{y}}) = F[f(\underline{y},\underline{\tilde{y}})] \qquad \underline{\tilde{y}} = \underline{\tilde{y}}(\underline{x},\underline{w}) \qquad f(\underline{w},\underline{y},\underline{\tilde{y}})$$

No cálculo das componentes do gradiente para um par entrada-saída $(\underline{x},\underline{y})$, usando a regra da cadeia

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

onde na expressão acima $\frac{\partial F}{\partial \tilde{y}_m}$ depende apenas da função objetivo, e $\frac{\partial \tilde{y}_m}{\partial w_i}$ depende apenas das equações do modelo.

Erro médio quadrático ponderado

Uma generalização simples mas com bastante aplicação é considerar como função objetivo o erro médio quadrático ponderado da saída, onde o ponderador a(m,k) pode ser função da saída m, do par p aplicado, etc.

$$\mathrm{F}(\underline{\mathrm{w}}) = \mathrm{E}_{\mathrm{p}} \sum_{\mathrm{m}} \mathrm{a}(\mathrm{p},\mathrm{m})\,\varepsilon_{\mathrm{m}}^2 \qquad \textit{onde} \qquad \varepsilon_{\mathrm{m}}^2 = (y_m - \tilde{y}_m)^2$$

donde

$$\frac{\partial F}{\partial w_i} = -2 \underset{p}{E} \sum_{m=1}^{M} a(p,m)\, \varepsilon_m \frac{\partial \tilde{y}_m}{\partial w_i}$$

Saídas com erros com relevâncias diferentes

Por exemplo, se em um modelo com três saídas o erro na primeira saída é cinco vezes mais relevante que o das outras duas devemos incluir este critério no treinamento usando pesos diferenciados por saída. No caso usaríamos a(p,1) = 5 e a(p,2) = a(p,3) =1.

Erro médio relativo quadrático:

Em ciências é muito comum o erro de interesse ser definido como o erro relativo. Mas devemos lembrar que alguns modelos (neurais, entre outros) trabalham com variáveis normalizadas (escaladas) x, y, diferentes das variáveis do mundo real, X, Y. Um escalamento muito usado é o chamado escalamento estatístico,

$$y_m = \frac{1}{\sigma_m}(Y_m - \mu_m) \qquad \text{e evidentemente} \qquad \tilde{y}_m = \frac{1}{\sigma_m}(\tilde{Y}_m - \mu_m)$$

onde σ_m *e* μ_m são respectivamente o desvio padrão e a média de Y_m. O erro médio relativo quadrático a ser minimizado é:

$$F = \underset{p}{E} \sum_{m=1}^{M} \left(\frac{Y_m - \tilde{Y}_m}{Y_m} \right)^2 = \underset{p}{E} \sum_{m=1}^{M} \frac{1}{(y_m + \sigma_m \mu_m)^2} (y_m - \tilde{y}_m)^2$$

que é da forma

$$F(\underline{w}) = \underset{p}{E} \sum_{m=1}^{M} a(p,m) \varepsilon_m^2 \qquad \text{onde} \qquad a(p,m) = \frac{1}{(y_m + \sigma_m \mu_m)^2}$$

Domínios com baixa população

O domínio de definição do mapeamento ($\underline{x}$, $\underline{y}$) pode ser subdividido em várias regiões ou subdomínios de igual dimensão mas eventualmente com populações muito diferentes. O caso típico são classificadores em que as diversas classes tem populações muito diferentes. Como o critério é de minimização do erro médio regiões com baixa população são mal aprendidas, i.e., mapeadas com erros elevados. Uma forma de contornar este problema é utilizar pesos inversamente proporcionais à população da região a que o par pertence. Neste caso usamos

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

onde *Pop (p)* é a população da região a que o par p pertence e P_{max} = Max [Pop(p)]. O efeito é como se todas as regiões tivessem uma população P_{max}.

Outras funções objetivo

Diversas outras funções objetivo podem ser utilizadas. O erro médio de ordens mais elevadas F_{oma} penaliza mais fortemente os erros maiores, mas as superfícies de F são mais abruptas e a convergência é mais problemática.

$$F_{oma}(\underline{w}) = \underset{p}{E}\left\{\sum_{m}(y_m - \tilde{y}_m)^{2n}\right\} \qquad n = 2,3,\ldots$$

O erro médio absoluto F_{abs} não prioriza os maiores erros como o emq. Não é derivável na origem e necessita procedimentos especiais.

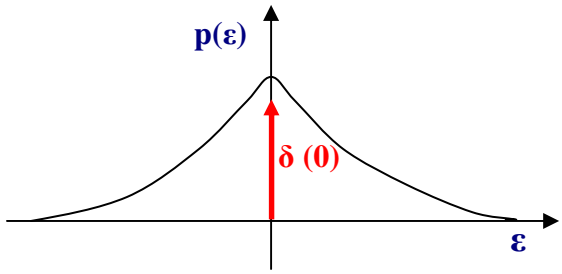
$$F_{abs}(\underline{w}) = \underset{p}{E}\left\{|\,\underline{y} - \underline{\tilde{y}}\,|\right\} = \underset{p}{E}\left\{\sqrt{\sum_{m}(y_m - \tilde{y}_m)^2}\right\}$$

O MAPE - Mean Absolute Percentual Error deve ser aplicado sobre as variáveis não escaladas, necessitando adaptação para as variáveis escaladas, como o erro relativo médio

quadrático. Não prioriza os maiores erros e não é derivavel na origem, necessitando de procedimentos especiais.

$$F_{MAPE}(\underline{w}) = \underset{p}{E}\left\{\frac{|\underline{Y} - \underline{\tilde{Y}}|}{|\underline{Y}|}\right\} = \underset{p}{E}\left\{\sqrt{\frac{\sum_m (Y_m - \tilde{Y}_m)^2}{\sum_m (Y_m)^2}}\right\}$$

Uma função objetivo que merece atenção especial é a entropia. O objetivo na minimização do erro ε é obter o histograma de p(ε) com média nula (fácil) e o mais estreito possível, o ideal sendo p(ε) = δ (0) (delta de Dirac, erro nulo). Se a distribuição do erro p(ε) for Gaussiana, o objetivo é alcançado minimizando o erro médio quadrático, a variância de p(ε). Se a distribuição do erro p(ε) não for Gaussiana, o objetivo é alcançado minimizando a entropia (ver os trabalhos de J.C. Príncipe)



II - Crítica durante o treinamento - acompanhamento do erro

Para diversos modelos (incluindo redes neurais) as superfícies de erro F(w) podem ter platôs que provocam a paralisia do processo e formas complexas que retardam temporáriamente a redução do erro e dando a falsa impressão do término do processo. O meio prático de acompanhar o processo é observar a evolução de F ao longo do treinamento.

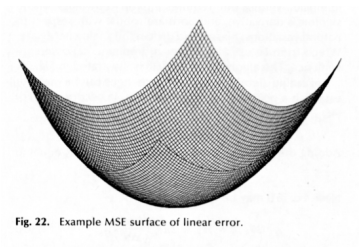


Fig. 22. Example MSE surface of linear error.

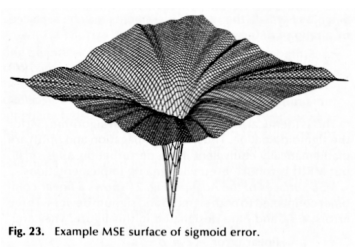


Fig. 23. Example MSE surface of sigmoid error.

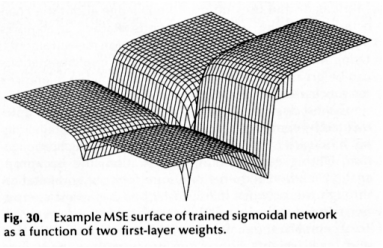
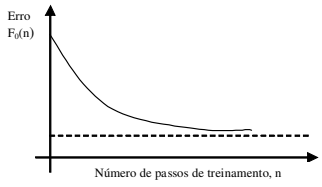
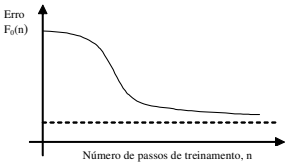
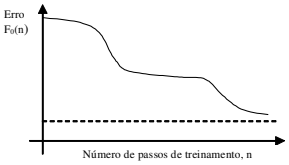


Fig. 30. Example MSE surface of trained sigmoidal network as a function of two first-layer weights.

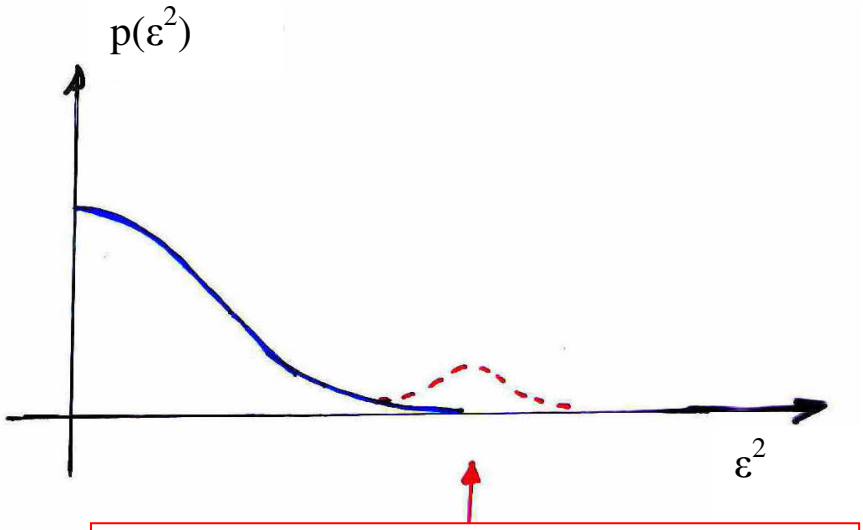






III - Crítica pós treinamento

Pelo critério do erro médio quadrático o histograma do erro quadrático ao término do treinamento deve ter a forma aproximada de uma semi-gaussiana. O aparecimento de modas com erros maiores possivelmente indica regiões de baixa população ou dados errados (intrusos, outlayers). Os pares que geram estes erros permitem localizar as regiões com problemas: são os possíveis pertencentes às regiões de baixa população ou possíveis intrusos e seus vizinhos.



Por enquanto

FIM

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

CPE 721 no próximo período.