Universidade Federal do Rio Grande do Sul
Escola de Engenharia
Departamento de Engenharia Química
Laboratório de Simulação - Laboratório de Controle e Integração de Processos

Notas do Curso de Controle Avançado de Processos

Prof. Dr. Argimiro R. Secchi
Prof. Dr. Jorge O. Trierweiler

– Identificação de Modelos Não Lineares –

- Redes Neuronais
- Redes de Modelos Locais

Conteúdo:

1. Estrutura de Modelos

A classificação do modelo matemático na estimação de parâmetros depende de como os parâmetros aparecem nas equações. A forma geral para modelos lineares pode ser escrita como:

$$y(x_1, x_2, ..., x_n) = \theta_1\ f_1(x_1, x_2, ..., x_n) + \theta_2\ f_2(x_1, x_2, ..., x_n) + ... + \theta_p\ f_p(x_1, x_2, ..., x_n) \qquad (1)$$

onde $f_i(x_1, x_2, ..., x_n)$ são vetores de formas funcionais conhecidas (podendo ser não linear nos x_i's), θ_i são os parâmetros do modelo, y é o vetor das variáveis dependentes e x_i são as variáveis independentes. Por exemplo:

$y(x) = \theta_1 + \theta_2\ x$	linear em θ e x
$y(x) = \theta_1 + \theta_2\ x + \theta_3\ x^2$	linear em θ e não linear em x

A forma geral para modelos não lineares pode ser escrita como:

$$y(x_1, x_2, ..., x_n) = f(x_1, x_2, ..., x_n; \theta_1, \theta_2, ..., \theta_p) \qquad (2)$$

Por exemplo:

$y(x_1, x_2) = \theta_1 + \theta_3\ (\theta 2\ x1 + x_2)$	não linear em θ e linear em x
$y(x) = \theta_1 \exp\ (\theta_2\ x)$	não linear em θ e x

Portanto, conhecida a estrutura do modelo, estimar os parâmetros do modelo significa encontrar o conjunto de θ_i's que faz com que o modelo reproduza os dados experimentais da melhor forma possível.

Caso a estrutura do modelo não esteja totalmente definida, tem-se um problema de identificação do processo associado ao problema de estimação de parâmetros, onde procura-se também determinar a estrutura do modelo mais adequada para representar os dados experimentais. Em um contexto mais completo, a identificação do processo consiste de um procedimento iterativo incluindo as seguintes etapas:

- planejamento de experimentos;
- seleção da estrutura do modelo;
- estimação de parâmetros;
- validação do modelo.

O ponto de partida para a identificação do processo é a pesquisa bibliográfica, onde procura-se acumular o conhecimento das diversas pesquisas realizadas para o desenvolvimento de modelos do processo. Como resultado pode-se obter uma estrutura hierárquica de modelos variando desde modelos detalhados e complexos a modelos simples usados para propósitos exploratórios e obtenção de características qualitativas do comportamento do sistema.

O planejamento de experimentos é fundamental para a obtenção de um modelo apropriado com um mínimo de experimentação. Por exemplo, o método da tentativa-e-erro para o projeto e execução de experimentos além de ser de alto custo e consumo de tempo, é um método auto-destrutivo. Para estimar os parâmetros do modelo com uma pequena região de confiabilidade, algo mais que uma simples análise dos resultados experimentais faz-se necessário. A Figura 1 mostra um estratégia eficiente para experimentações, onde um modelo proporciona as bases para novos experimentos que levarão a um novo modelo.

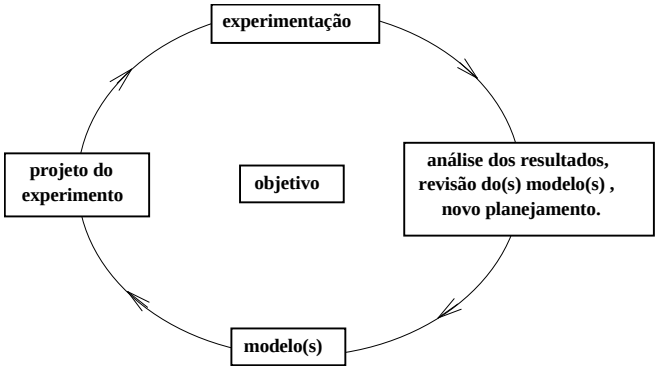

Figura 1. *Estratégia de experimentação eficiente.*

A seleção da estrutura do modelo depende fundamentalmente do conhecimento do sistema e do grau de informação que se deseja obter do mesmo. Outros aspectos que devem ser levados em conta são: a simplicidade do modelo, as características numéricas favoráveis à estimação de seus parâmetros e a dimensão do modelo (número de variáveis e parâmetros e número de equações).

Dependendo do conhecimento prévio do sistema, a estrutura do modelo pode ser classificada dentre os seguintes categorias:

1) Modelos "caixa-branca": quando o modelo é conhecido perfeitamente, com base em conhecimentos prévios e aspectos teóricos;
2) Modelos "caixa-cinza": quando algum aspecto teórico é disponível, mas vários parâmetros precisam ser determinados a partir de dados observados. Pode-se ainda subdividir em modelagem teórica e semi-empírica. No primeiro caso a estrutura do modelo é construída com embasamento teórico, e no segundo caso os sinais medidos sofrem transformações não lineares com base teórica antes de entrar em uma estrutura "caixa-preta".
3) Modelos "caixa-preta": quando o modelo é totalmente empírico, mas escolhido com base em experiências bem sucedidas em situações similares.

Os modelos "caixa-preta" lineares, que já foram abordados no módulo de identificação de sistemas lineares, possuem a seguinte estrutura geral:

$$A(q)\, y(t) = \frac{B(q)}{F(q)} u(t) + \frac{C(q)}{D(q)} e(t) \tag{3}$$

onde $q^{-1}\, x(t) = x(t-1)$, e alguns casos especiais são conhecidos como:

- modelo Box-Jenkins (BJ): $A = 1$;
- modelo ARIMAX: $F = 1$, $D = 1 - q^{-1}$;
- modelo ARMAX: $F = D = 1$;
- modelo ARX: $F = C = D = 1$;
- modelo *output-error* (OE): $A = C = D = 1$;
- modelo FIR: $A = F = C = D = 1$.

O preditor associado à Equação (3) pode ser escrito na seguinte forma de regressão linear:

$$\hat{y}(t \mid \theta) = \theta^T \varphi(t, \theta) \tag{4}$$

onde os regressores, isto é, os componentes de $\varphi(t, \theta)$, são dados por:

$u(t–k)$	entradas passadas, associados ao polinômio $B(q)$
$y(t–k)$	saídas passadas, associados ao polinômio $A(q)$
$\varepsilon(t–k \mid \theta) = y(t–k) - \hat{y}(t-k \mid \theta)$	erros de predição, associados ao polinômio $C(q)$
$\hat{y}_u(t-k \mid \theta) = A(q)\hat{y}(t-k \mid \theta)$	saídas simuladas com u passados, associadas ao polinômio $F(q)$
$\varepsilon_u(t–k \mid \theta) = y(t–k) - \hat{y}_u(t-k \mid \theta)$	erros de simulação, associados ao polinômio $D(q)$.

Uma estrutura geral para os modelos "caixa-preta" não lineares tem a seguinte forma:

$$y(t) = g(u^{t-1}, y^{t-1}) + v(t) \tag{5}$$

onde $x^t = [x(1)\ x(2)\ ...\ x(t)]$, $x = u, y$, são as entradas e saídas observadas, e $v(t)$ leva em conta o fato de que $y(t)$ não é função somente de dados passados. Entretanto, deve-se ter uma estrutura onde $v(t)$ é pequeno, de modo que $g(u^{t-1}, y^{t-1})$ seja uma boa predição de $y(t)$.

O preditor associado à Equação (5), na forma de regressão não linear, é escrito como:

$$\hat{y}(t \mid \theta) = g(\varphi(t), \theta) \tag{6}$$

onde g é alguma função não linear parametrizada por θ.

Seguindo a nomenclatura usada para os modelos lineares, tem-se os casos especiais não lineares:

- modelo NBJ: utiliza $u(t–k)$, $\varepsilon(t–k \mid \theta)$, $\hat{y}_u(t-k \mid \theta)$ e $\varepsilon_u(t–k \mid \theta)$ como regressores. Neste caso as saídas simuladas são obtidas substituindo os regressores ε e ε_u por zeros;
- modelo NARMAX: utiliza $u(t–k)$, $y(t–k)$ e $\varepsilon(t–k \mid \theta)$ como regressores;
- modelo NARX: utiliza $u(t–k)$ e $y(t–k)$ como regressores;
- modelo NOE: utiliza $u(t–k)$ e $\hat{y}_u(t-k \mid \theta)$ como regressores. Neste caso, a saída do modelo é também $\hat{y}(t \mid \theta)$;
- modelo NFIR: utiliza somente $u(t–k)$ como regressores.

As estruturas NBJ, NARMAX e NOE são **estruturas recorrentes**, pois parte do vetor de regressão é formado por saídas passadas do modelo.

Uma maneira natural de construir o mapeamento não linear $g(\varphi, \theta)$ é através de expansão em famílias de funções:

$$g(\varphi, \theta) = \sum \alpha_k\, g_k(\varphi) \tag{7}$$

onde g_k são chamados de funções bases. A maioria dos modelos não lineares "caixa-preta" são estruturados com g_k obtidos através de uma parametrização de uma única **função base mãe**, $\kappa(x)$. Nestes casos, pode-se escrever as funções bases na seguinte forma geral:

$$g_k(\varphi) = \kappa(\varphi, \beta_k, \gamma_k) \tag{8}$$

onde β_k e γ_k são parâmetros com características diferenciadas. Em geral, β_k está relacionado com a escala ou alguma propriedade direcional de $g_k(\varphi)$, e γ_k é um parâmetro de posição ou translação.

Na Tabela 1 são listados alguns exemplos de expansão de funções escalares, $f(x)$, dada pela Equação (7), quando as funções bases tem a forma $g_k(\varphi) = \kappa[\beta_k\,(\varphi - \gamma_k)]$.

Tabela 1. Expansão de funções escalares.

expansão	$\kappa(x)$	β_k	γ_k	α_k
série de Fourier	$\cos(x)$	freqüências	fases	$\langle f(x),\cos(x)\rangle / \| \cos(x) \|^2$
constante por partes (a)	1 se $0 \le x < 1$, 0 caso contrário	$1/\Delta$	k	$f(k\Delta)$
(a) suavizada	$r(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$	$1/\Delta$	k	$f(k\Delta)$
constante por partes (b)	0 se $x < 0$, 1 se $x \ge 0$	$1/\Delta$	k	$f(k\Delta)-f[(k\text{-}1)\Delta]$
(b) suavizada	$\sigma(x) = \frac{1}{1+e^{-x}}$	$1/\Delta$	k	$f(k\Delta)-f[(k\text{-}1)\Delta]$

Os quatro últimos exemplos da Tabela 1 utilizam **funções bases locais**, pois seus gradientes possuem suporte limitado, isto é, suas variações estão concentradas em um intervalo finito. Por outro lado, o primeiro exemplo utiliza **funções bases globais**.

Para o caso multivariável, $\varphi = [\varphi_1\ \varphi_2\ \ldots\ \varphi_d]$, as funções bases são geralmente construídas como extensões do caso monovariável como segue:

1- Produto tensorial: $g_k(\varphi) = \prod \kappa_i(\varphi_i, \beta_{ik}, \gamma_{ik})$;

2- Função radial: $g_k(\varphi) = \kappa(\|\varphi - \gamma_k\|_{\beta_k})$, em uma dada norma $\|\cdot\|_{\beta_k}$, ex: $\|\varphi\|^2_{\beta_k} = \varphi^T \beta_k \varphi$;

3- Função "ridge": $g_k(\varphi) = \kappa(\beta_k^T \varphi + \gamma_k)$.

Note que a função "ridge" é constante para todo φ no subespaço $\{\varphi \in \mathbb{S}^d / \beta_k^T \varphi = \text{const.}\}$. Como conseqüência, mesmo se a função base mãe, κ, tiver suporte local, a função base g_k não terá suporte limitado neste subespaço (função base semi-global).

A Tabela 2 apresenta algumas estruturas populares de modelos nesta formulação.

Tabela 2. Exemplos de estruturas de modelos.

estrutura de modelo	**função base mãe, $\kappa(x)$**	**função base, $g_k(\varphi)$**
wavelets	base ortonormal	$g_{jk}(\varphi) = 2^{j/2}\,\kappa(2^j\varphi - k)$
kernel	funções em forma de sino	$\kappa[(\varphi - \gamma_k)/\Delta]$, γ_k = pontos no $\mathbb{S}^d$
interpolação	função pulso unitário	$\kappa(\|\varphi - \gamma_k\|_\infty)$, γ_k = centros de hipercubos
fuzzy	funções membro	$\prod \kappa_i[\beta_{ik}(\varphi_i - \gamma_{ik})]$
Volterra	x	$\prod \kappa(\beta_{ik}\,\varphi_i)$
rede neuronal sigmoidal	$\sigma(x)$	$\kappa(\beta_k^T \varphi + \gamma_k)$
RBF	$r(x)$	$\kappa(\|\varphi - \gamma_k\|/\sigma_k)$, γ_k = centros da RBF
rede modelos locais lineares	$r(x)$, $y_L(x) = \theta^T x$	$r(\|\varphi_r - \gamma_k\|/\sigma_k)\ \theta_k^T \varphi_L$

As estruturas de modelos com expansão em funções bases são freqüentemente referidas como **redes**, primeiro porque geralmente a função base mãe é repetida em um número elevado de vezes na expansão; segundo porque uma ilustração gráfica da estrutura tem a forma de uma rede.

No caso de estruturas em redes de multi-camadas, a extensão é direta, bastando definir os regressores para cada camada. Então, para a i-ésima camada, com *m* funções bases, tem-se:

$$\varphi_k^{i+1} = g_k^i(\varphi^i) = \kappa(\varphi^i, \beta_k^i, \gamma_k^i) \tag{9}$$

onde $\varphi^{i+1} = [\varphi_1^{i+1} \;\; \varphi_2^{i+1} \cdots \varphi_m^{i+1}]$ é o vetor de regressão para o camada i+1. A Figura 2 ilustra uma estrutura de rede *feedforward* com duas camadas internas (ou **escondidas** - "*hidden*").

Portanto, para construir um modelo "caixa-preta" não linear, os seguintes passos devem ser executados:

1. Selecionar os regressores φ;
2. Selecionar a(s) função(ões) base(s) mãe(s), κ;
3. Fazer a expansão da função base mãe no espaço dos regressores, do tipo radial, "ridge", produto tensorial, ou uma outra função multidimensional específica;
4. Determinar o número de funções bases, g_k, e o número de camadas da estrutura;
5. Determinar os valores dos parâmetros de dilatação e localização, β_k e γ_k, respectivamente;
6. Determinar os parâmetros de coordenação, α_k, da expansão de $g(\varphi, \theta)$.

Basicamente, há duas possibilidades para determinação dos parâmetros dos passos 5 e 6:

a) Estimar todos os parâmetros α, β e γ simultaneamente;

b) Tratar os parâmetros β e γ separadamente, como por exemplo usar valores predeterminados como no caso das *wavelets*. Neste caso, a estimação dos α_k é um problema de regressão linear, para redes mono-camadas.

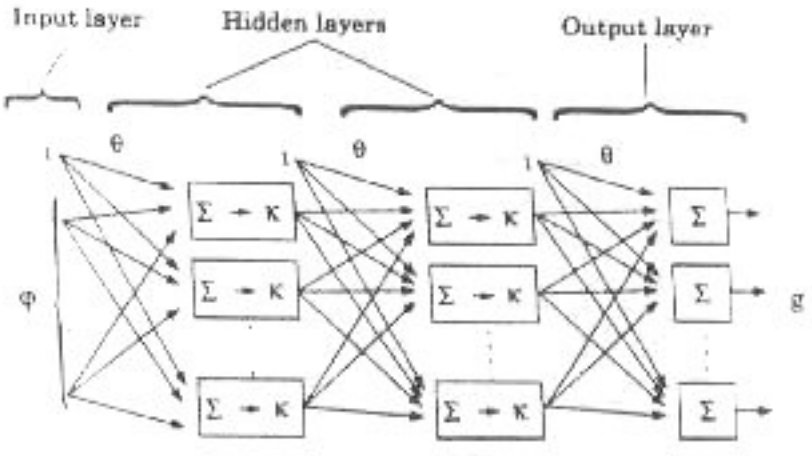

Figura 2. Rede *feedforward* com duas camadas escondidas.

2. Notas Sobre Estimação de Parâmetros

Considerando que a estrutura do modelo esteja determinada, o problema de estimação de parâmetros é no final das contas um problema de otimização, onde uma função objetivo deve ser formulada.

Na técnica dos mínimos quadrados procura-se minimizar os quadrados dos desvios do modelo em relação aos dados observados:

$$\hat{\sigma}_y^2 = \min_{\alpha} S(\alpha) = \min_{\alpha} \frac{1}{\nu} \sum_{i=1}^{n} \varepsilon_i^2 = \min_{\alpha} \frac{1}{\nu} \sum_{i=1}^{n} [y_i - f(x_i;\alpha)]^2 \tag{10}$$

onde $\nu = n - p$ é o número de graus de liberdade, isto é, número de experimentos menos os números de parâmetros a serem estimados. Observa-se que se o erro experimental for o mesmo para todos os dados, então $\hat{\sigma}_y^2$ é uma estimativa da variança experimental de *y*.

No caso de se conhecer a função densidade de probabilidade, $p(x,y)$ o problema é posto na seguinte forma:

$$\min_{\alpha} S(\alpha) = \min_{\alpha} E\{[y - f(x;\alpha)]^2\} = \min_{\alpha} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [y - f(x;\alpha)]^2 p(x,y) dx dy \tag{11}$$

A técnica da máxima verossimilhança é baseada na função densidade de probabilidade:

$$p(y/x;\alpha) = \frac{1}{\sqrt{2\pi}\sigma_y} \exp\left[-\frac{1}{2\sigma_y^2} [y - f(x;\alpha)]^2 \right] \tag{12}$$

isto é, admite-se que a variável medida *y* apresenta distribuição normal em torno da média, $f(x)$. Se os erros experimentais não estão correlacionados (eventos independentes), pode-se construir a função de verossimilhança:

$$P(\alpha/y,x) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_{y_i}} \exp\left[-\frac{1}{2\sigma_{y_i}^2} [y_i - f(x_i;\alpha)]^2 \right] \tag{13}$$

e os parâmetros podem ser estimados de forma a maximizar a função de verossimilhança, isto é, maximizar a probabilidade de encontrar os dados experimentais obtidos. As estimativas pela máxima verossimilhança são eficientes e portanto consistentes, contudo não são necessariamente livres de erros sistemáticos.

Rescrevendo a função de verossimilhança na forma logarítmica:

$$\ln P(\alpha/y,x) = \sum_{i=1}^{n} \ln \frac{1}{\sqrt{2\pi}\sigma_{y_i}} - \sum_{i=1}^{n} \frac{1}{2\sigma_{y_i}^2} [y_i - f(x_i;\alpha)]^2 \tag{14}$$

observa-se que maximizar $P(\alpha / y, x)$ em relação aos parâmetros α é equivalente a:

$$\min_{\alpha} S(\alpha) = \min_{\alpha} \sum_{i=1}^{n} \frac{1}{\sigma_{y_i}^2} [y_i - f(x_i;\alpha)]^2 \tag{15}$$

que pode ser visto como um problema de mínimos quadrados ponderados pela recíproca do erro experimental. No caso em que o erro experimental é o mesmo para todos os dados pode-se obter uma estimativa da variança experimental de *y* pela maximização de $P(\alpha, \sigma_y / y, x)$ em relação ao parâmetro σ_y, obtendo-se:

$$\hat{\sigma}_y^2 = \frac{1}{n}\sum_{i=1}^{n}[y_i - f(x_i;\alpha)]^2 = \frac{n-1}{n}\sigma_y^2 \tag{16}$$

que é uma estimativa com erro sistemático de σ_y^2.

O grande mérito da máxima verossimilhança é permitir a consideração de erros nas variáveis independentes de forma consistente. Admitindo-se que tanto a variável medida y_i quanto a variável medida x_i apresentam distribuição normal em torno de suas médias, $f(\bar{x}_i)$ e $\bar{x}_i$, respectivamente, pode-se construir a função de verossimilhança para erros experimentais não correlacionados:

$$P(\alpha, \bar{x} / y, x) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_{y_i}} \exp\left[-\frac{1}{2\sigma_{y_i}^2}[y_i - f(\bar{x}_i;\alpha)]^2\right] \frac{1}{\sqrt{2\pi}\sigma_{x_i}} \exp\left[-\frac{1}{2\sigma_{x_i}^2}(x_i - \bar{x}_i)^2\right] \tag{17}$$

que ao ser maximizada em relação aos parâmetros α_i's e $\bar{x}_i$'s equivale a:

$$\min_{\alpha,\bar{x}} S(\alpha, \bar{x}) = \min_{\alpha,\bar{x}} \left\{ \sum_{i=1}^{n} \frac{1}{\sigma_{y_i}^2}[y_i - f(\bar{x}_i;\alpha)]^2 + \sum_{i=1}^{n} \frac{1}{\sigma_{x_i}^2}(x_i - \bar{x}_i)^2 \right\} \tag{18}$$

Conclui-se então que o método da máxima verossimilhança se reduz ao método dos mínimos quadrados quando as variáveis independentes são isentas de erro ($\bar{x}_i = x_i$) e quando os erros nas variáveis dependentes são constantes, com a diferença que o método dos mínimos quadrados, não gera uma estimativa com erro sistemático para a variança experimental de *y*. Por outro lado, o método da máxima verossimilhança permite valorizar mais aos dados com menores erros experimentais.

Outras técnicas de estimação como as variáveis instrumentais (IV) e o método do erro de predição (PEM), podem ser encontrados na literatura.

Uma estimação completa, do ponto de vista estatístico, deve fornecer além de estimativas para os parâmetros, estimativas para suas variânças e covarianças (isto é, a matriz de covariança) e estimativas para a variança experimental. Pois desta forma, pode-se determinar:

- se o modelo é adequado (uso do teste F e covarianças);
- os intervalos de confiança dos parâmetros (uso das variânças);
- se os parâmetros são significativamente diferentes de zero (uso do teste t);
- o intervalo de predição do modelo (análise dos erros de predição).

A matriz de covariança entre variáveis aleatórias $x_1, x_2, ..., x_N$ é definida como:

$$V = \begin{bmatrix} \sigma_1^2 & \sigma_{1,2}^2 & \cdots & \sigma_{1,N}^2 \\ \sigma_{1,2}^2 & \sigma_2^2 & \cdots & \sigma_{2,N}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1,N}^2 & \sigma_{2,N}^2 & \cdots & \sigma_N^2 \end{bmatrix} = E\{\Delta x \Delta x^T\} \tag{19}$$

onde $\sigma^2_{i,j} = E\{(x_i - \bar{x}_i)(x_j - \bar{x}_j)\} = Co\,\mathrm{var}\{\Delta x_i, \Delta x_j\}$ e $\sigma^2_i = E\{(x_i - \bar{x}_i)^2\} = Var\{\Delta x_i\}$.

Define-se também o coeficiente de correlação:

$$\rho_{i,j} = \frac{\sigma^2_{i,j}}{\sigma_i \sigma_j} \quad , \quad -1 < \rho_{i,j} < 1 \tag{20}$$

que corresponde a $\sigma^2_{i,j}$ normalizada, e a matriz de correlações $C = [\rho_{i,j}]$. Forte correlação entre os parâmetros *i* e *j* são caracterizados por valores elevados de $|\rho_{i,j}|$, sendo que um valor negativo de $\rho_{i,j}$ indica que um desvio positivo no parâmetro *i* provoca um desvio negativo no parâmetro *j*, de modo a acomodar os resultados.

Analisando o problema de estimação para um modelo genérico (não linear nos parâmetros), admitindo que as variáveis independentes não estão sujeitas a erros, a matriz de covariança pode ser obtida linearizando a equação do modelo em torno dos valores esperados dos parâmetros, $\alpha = E\{\hat{\alpha}\}$:

$$f(x;\hat{\alpha}) \approx f(x;\alpha) + \frac{\partial f}{\partial \alpha}\Delta\alpha \tag{21}$$

com $\Delta\alpha = \alpha - \hat{\alpha}$. Deste modo a equação de predição $\tilde{y} = f(x;\hat{\alpha}) + \tilde{\varepsilon}$, onde $\tilde{\varepsilon}$ representa os erros de predição, pode ser rescrita na forma:

$$\Delta\tilde{y} \approx \frac{\partial f}{\partial \alpha}\Delta\alpha + \tilde{\varepsilon} \quad , \quad \text{com} \quad \Delta\tilde{y} = \tilde{y} - f(x;\alpha) \tag{22}$$

Como a estimação dos parâmetros equivale a minimizar a função objetivo $S(\alpha;x,y)$, então:

$$\nabla_\alpha S(\hat{\alpha};x,y) = 0 \tag{23}$$

Da mesma forma, na minimização de $S(\hat{\alpha} + \Delta\alpha;\, x,\, y + \Delta y)$ tem-se:

$$\nabla_\alpha S(\hat{\alpha} + \Delta\alpha; x, y + \Delta y) = 0 \tag{24}$$

onde $\Delta y = y - f(x;\hat{\alpha}) = \varepsilon$ e portanto:

$$\nabla_\alpha S(\hat{\alpha} + \Delta\alpha; x, y + \Delta y) = \nabla_\alpha S(\hat{\alpha};x,y) + \frac{\partial}{\partial y}\nabla_\alpha S(\hat{\alpha};x,y)\Delta y + \frac{\partial}{\partial \alpha}\nabla_\alpha S(\hat{\alpha};x,y)\Delta\alpha = 0 \tag{25}$$

Lembrando que:

$$\frac{\partial}{\partial \alpha}\nabla_\alpha S(\hat{\alpha};x,y) = \nabla^2_\alpha S(\hat{\alpha};x,y) = H_\alpha(\hat{\alpha};x,y) \tag{26}$$

é a matriz Hessiana de $S(\alpha;x,y)$, e definindo:

$$G = \frac{\partial}{\partial y}\nabla_\alpha S(\hat{\alpha}; x, y) = \begin{bmatrix} \frac{\partial^2 S}{\partial \alpha_1 \partial y_1} & \frac{\partial^2 S}{\partial \alpha_1 \partial y_2} & \cdots & \frac{\partial^2 S}{\partial \alpha_1 \partial y_n} \\ \frac{\partial^2 S}{\partial \alpha_2 \partial y_1} & \frac{\partial^2 S}{\partial \alpha_2 \partial y_2} & \cdots & \frac{\partial^2 S}{\partial \alpha_2 \partial y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 S}{\partial \alpha_p \partial y_1} & \frac{\partial^2 S}{\partial \alpha_p \partial y_2} & \cdots & \frac{\partial^2 S}{\partial \alpha_p \partial y_n} \end{bmatrix} \tag{27}$$

tem-se então:

$$G\,\Delta y + H_\alpha\,\Delta\alpha = 0 \tag{28}$$

ou

$$\Delta\alpha = -H_\alpha^{-1} G \Delta y = -H_\alpha^{-1} G \varepsilon \tag{29}$$

A Equação (29) corresponde a equação do método Newton, pois $G(\hat{\alpha}; x, y)\varepsilon = \nabla_\alpha S(\hat{\alpha}; x, y)$.

Por exemplo, para um modelo linear do tipo: $f(x; \alpha) = x^T \alpha$, e usando o critério dos mínimos quadrados (na forma matricial),

$$S(\alpha) = \frac{1}{\nu}\sum_{i=1}^{n} \varepsilon_i^2 = \frac{1}{\nu}\sum_{i=1}^{n} [y_i - x_i^T \alpha]^2 = \frac{1}{\nu}(Y - X\,\alpha)^T (Y - X\,\alpha)$$

tem-se: $\nabla_\alpha S(\hat{\alpha}; X, Y) = -\frac{2}{\nu} X^T (Y - X\,\hat{\alpha}) = 0$, $\nabla_\alpha^2 S(\hat{\alpha}; X, Y) = H_\alpha(\hat{\alpha}; X, Y) = \frac{2}{\nu} X^T X$ e

$\frac{\partial}{\partial y}\nabla_\alpha S(\hat{\alpha}; X, Y) = G(\hat{\alpha}; X, Y) = -\frac{2}{\nu} X^T$.

Portanto, $\Delta\alpha = -H_\alpha^{-1} G\varepsilon = (X^T X)^{-1} X^T \varepsilon$ ou simplesmente $\hat{\alpha} = (X^T X)^{-1} X^T Y$, pois o modelo é linear.

Para problemas com *r* variáveis dependentes, a matriz *G* é formada por *n* x *r* colunas, onde o *j*-ésimo conjunto de *n* colunas corresponde as derivadas do $\nabla_\alpha S(\hat{\alpha}; x, y)$ em relação a *j*-ésima variável dependente. Pois neste caso o vetor *y* teria a seguinte forma:

$$y = [y_{11}\ y_{12} \ldots y_{1n}\ y_{21}\ y_{22} \ldots y_{2n}\ \ldots y_{r1}\ y_{r2} \ldots y_{rn}]^T$$

A equação acima mostra como os desvios experimentais, ε, se propagam até os parâmetros α, e:

$$\Delta\alpha\Delta\alpha^T = \left(-H_\alpha^{-1} G\varepsilon\right)\left(-H_\alpha^{-1} G\varepsilon\right)^T = H_\alpha^{-1} G\varepsilon\varepsilon^T G^T (H_\alpha^{-1})^T$$

$$E\{\Delta\alpha\Delta\alpha^T\} = H_\alpha^{-1} G E\{\varepsilon\varepsilon^T\} G^T (H_\alpha^{-1})^T \tag{30}$$

portanto a matriz de covariança dos parâmetros é dada por:

$$V_\alpha = H_\alpha^{-1} G V_\varepsilon G^T (H_\alpha^{-1})^T \tag{31}$$

onde V_ε é a matriz de covariança dos erros experimentais. Observa-se pela equação acima, que a incerteza experimental (V_ε) se transforma em incerteza nos parâmetros (V_α) após ser filtrada pela função objetivo (S) e modelo (f) utilizados, através das matrizes das derivadas H_α e G. A correlação existente entre os parâmetros e os erros experimentais pode ser observada através da matriz de covariança entre os mesmos:

$$V_{\alpha,\varepsilon} = E\{\Delta\alpha, \varepsilon\} = -H_\alpha^{-1} G E\{\varepsilon\varepsilon^T\} = -H_\alpha^{-1} G V_\varepsilon \tag{32}$$

A região de confiança dos parâmetros é a elipsóide dada por:

$$\Delta\alpha^T V_\alpha^{-1} \Delta\alpha = \eta^2 \tag{33}$$

onde $\eta = t_{\frac{1-\delta}{2}}$, isto é, t^* tal que $p(t > t^*) = \dfrac{1-\delta}{2}$, para um grau de certeza δ.

Por exemplo, para uma região com 99 % de confiança nos parâmetros tem-se $\eta \approx 3$. O intervalo de confiança dos parâmetros é dado por:

$$\hat{\alpha}_i - \eta\, \sigma_i \leq \alpha_i < \hat{\alpha}_i + \eta\, \sigma_i$$

desde que o parâmetro $\hat{\alpha}_i$ tenha distribuição normal em torno de α_i.

A matriz de covariança dos erros de predição é obtida a partir da equação de predição em termos da variável desvio:

$$\Delta\tilde{y}\Delta\tilde{y}^T = \left(\frac{\partial f}{\partial \alpha}\Delta\alpha + \tilde{\varepsilon}\right)\left(\frac{\partial f}{\partial \alpha}\Delta\alpha + \tilde{\varepsilon}\right)^T$$

definindo a matriz de sensitividade:

$$B = \frac{\partial f}{\partial \alpha} \tag{34}$$

$$\Delta\tilde{y}\Delta\tilde{y}^T = (B\Delta\alpha + \tilde{\varepsilon})(\Delta\alpha^T B^T + \tilde{\varepsilon}^T) = B\Delta\alpha\Delta\alpha^T B^T + B\Delta\alpha\tilde{\varepsilon}^T + \tilde{\varepsilon}\Delta\alpha^T B^T + \tilde{\varepsilon}\tilde{\varepsilon}^T$$

$$V_{\tilde{y}} = E\{\Delta\tilde{y}\Delta\tilde{y}^T\} = BV_\alpha B^T + BE\{\Delta\alpha\tilde{\varepsilon}^T\} + E\{\tilde{\varepsilon}\Delta\alpha^T\}B^T + V_{\tilde{\varepsilon}} \tag{35}$$

Como os erros de predição não estão correlacionados com os parâmetros (ou, erros passados e futuros não estão correlacionados):

$$E\{\Delta\alpha\tilde{\varepsilon}^T\} = E\{\tilde{\varepsilon}\Delta\alpha^T\} = 0 \tag{36}$$

e portanto:

$$V_{\tilde{y}} = BV_\alpha B^T + V_{\tilde{\varepsilon}} \tag{37}$$

mostrando que as predições são afetadas tanto pelos erros passados, através dos erros nos parâmetros (V_α), quanto pelos erros futuros ($V_{\tilde{\varepsilon}}$).

3. Notas Sobre Otimização

O problema de estimação de parâmetros pode ser tratado como um problema de otimização sem restrição, isto é, o problema que está sendo resolvido é:

$$\min_{x \in \Re^p} S(x)$$

onde x é o vetor de parâmetros a ser determinado. Os métodos existentes para a solução deste problema podem ser agrupados em duas categorias:

1) métodos que não usam derivadas (ou métodos de busca);

2) métodos que usam derivadas (ou métodos analíticos ou métrica variável).

Como regra geral, na solução de problemas sem restrição, os métodos que usam derivadas convergem mais rapidamente que os métodos de busca. Por outro lado, os métodos de busca não requerem regularidade e continuidade da função objetivo e, principalmente o cálculo de derivadas primeira ou segunda de $S(x)$.

Os métodos de busca monovariável são, em geral, usados como métodos auxiliares para a determinação do tamanho do passo dos métodos analíticos. Dentre os diversos métodos deste tipo, cita-se:

- método da seção áurea (“golden section”);
- método de Coggins ou DSC-Powell (interpolação polinomial);
- método da bisseção.

Os métodos de busca mulivariáveis são métodos competitivos com os métodos analíticos em função das vantagens e desvantagens citadas acima. Como exemplo destes métodos tem-se:

- método dos poliedros flexíveis;
- método de Hooke & Jeeves;
- método da busca seccionada.

Como a otimização sem restrições é equivalente a encontrar a solução do sistema de equações não-lineares $F(x) = \nabla S(x) = 0$, pode-se utilizar todos os métodos disponíveis para a solução de $F(x) = 0$. Por exemplo, na utilização do método de Newton-Raphson, a matriz Jacobiana é a própria matriz Hessiana.

Apesar da literatura referenciar somente os métodos quasi-Newton, que utilizam aproximações para o cálculo da matriz Hessiana, como métodos da métrica variável, nestas notas o termo “métrica variável” engloba todos os métodos que utilizam a primeira e/ou a segunda derivada da função objetivo. Estes métodos têm como equação básica para o processo iterativo:

$$x^{k+1} = x^k - \alpha_k \, W(x^k) \, \nabla S(x^k) \tag{38}$$

onde α_k é o tamanho do passo

$W(x^k)$ é a matriz direção (inversa da matriz Hessiana ou uma aproximação desta)

A seguir serão descritos os métodos mais conhecidos desta classe, seguidos de uma generalização dos métodos da métrica variável.

1. Métodos do gradiente

Utilizam somente a primeira derivada da função objetivo, caso em que $W(x^k) = I$:

$$x^{k+1} = x^k - \alpha_k \nabla S(x^k) \tag{39}$$

Quando α_k é escolhido de modo a minimizar:

$$g_k(\alpha) = S(x^k - \alpha \nabla S(x^k)) \ , \ \alpha > 0 \tag{40}$$

tem-se o método da maior descida ("steepest descent"), cujo algoritmo básico pode ser escrito da seguinte forma.

algoritmo

1) Escolher um ponto inicial x^o, $k = 0$

2) Calcular $d_k = -\nabla S(x^k)$

3) Encontrar α_k tal que $S(x^k + \alpha_k d_k) = \min_{\alpha>0} g_k(\alpha) = S(x^k + \alpha d_k)$

4) Calcular $x^{k+1} = x^k + \alpha_k d_k$

5) **Se** o critério de convergência não foi satisfeito, **então** $k \leftarrow k + 1$ (ir para 2)

6) FIM.

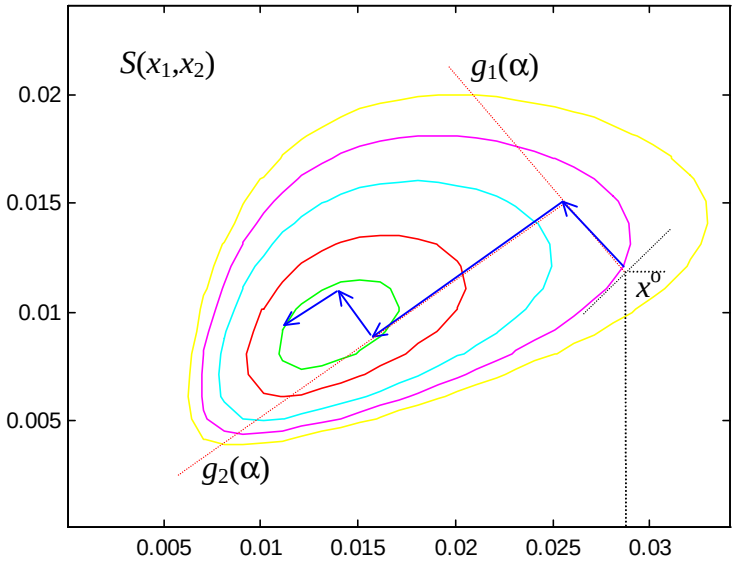

Figura 3. Exemplo de otimização pelo método do gradiente.

A minimização de $g_k(\alpha)$, também chamada de busca em linha ("linesearch"), pode ser realizada com o uso de qualquer método de minimização univariável. Para ilustrar esta função, a Figura 3 mostra as curvas de níveis de uma função objetivo bi-dimensional com a trajetória do método dos gradientes. A determinação do tamanho do passo por uma busca em linha para este exemplo está ilustrada na Figura 4, onde mostra a função $g_2(\alpha)$ do problema acima:

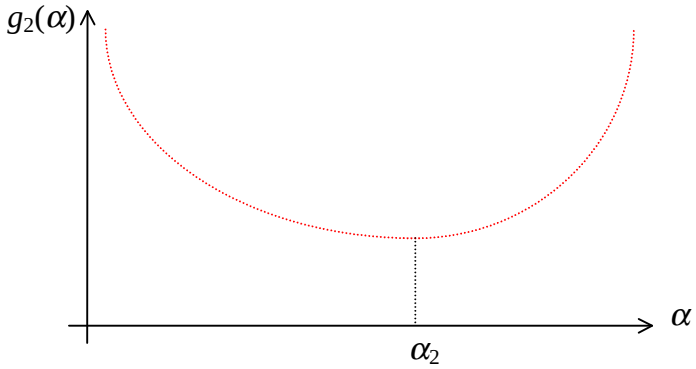

Figura 4. Exemplo de busca em linha para a determinação do tamanho do passo.

Aproximando $S(x)$ por uma função quadrática:

$$S(x^{k+1}) \approx S(x^k) + \nabla^T S(x^k)(x^{k+1} - x^k) + \frac{1}{2}(x^{k+1} - x^k)^T H(x^k)(x^{k+1} - x^k)$$

ou de forma similar:

$$g_k(\alpha) = S(x^k + \alpha d_k) \approx S(x^k) + \alpha \nabla^T S(x^k) d_k + \frac{1}{2}\alpha^2 d_k^T H(x^k) d_k$$

que minimizando em relação a α, $\frac{dg_k}{d\alpha} = 0$, resulta:

$$\alpha^* = \alpha_k = -\frac{\nabla^T S(x^k) d_k}{d_k^T H(x_k) d_k} = \frac{d_k^T d_k}{d_k^T H(x_k) d_k} \qquad (41)$$

Contudo, a Equação (41) não é utilizada para o cálculo de α nos métodos gradientes, pois exigiria o cálculo da segunda derivada da função objetivo. Neste caso, utiliza-se, em geral, métodos de busca para a sua seleção.

2. Método de Newton

Faz uso da segunda derivada da função objetivo, caso em que $W(x^k) = [H(x^k)]^{-1}$:

$$x^{k+1} = x^k - \alpha_k [H(x^k)]^{-1} \nabla S(x^k) \qquad (42)$$

que é resultado da minimização da aproximação de $S(x)$ por uma função quadrática:

$$S(x) \approx S(x^k) + \nabla^T S(x^k)\Delta x^k + \frac{1}{2}(\Delta x^k)^T H(x^k)\Delta x^k \qquad (43)$$

onde $\Delta x^k = x - x^k$, na direção Δx^k, isto é: $\frac{\partial S}{\partial \Delta x_i^k} = 0$

$$\Delta x^k = -[H(x^k)]^{-1} \nabla S(x^k)$$

Neste caso α_k ou é um parâmetro de relaxação do processo iterativo $0 < \alpha_k \leq 1$, ou é um fator de correção da inversa da matriz Hessiana, caso esta não seja atualizada em todas as iterações.

Em particular, quando o método de Newton é utilizado para a solução de problemas de mínimos quadrados, ele é comumente referenciado na literatura como método de Gauss-Newton. Sendo que uma das aplicações é a solução de sistemas de equações não-lineares, $F(x) = 0$, transformados em problemas de mínimos quadrados ao procurar minimizar o quadrado dos resíduos, isto é,

$$S(x) = F^T(x)F(x) = \sum_{i=1}^{n} f_i^2(x)$$

neste caso $\nabla S(x^k) = J^T(x^k)\, F(x^k)$ e $H(x^k) = J^T(x^k)\, J(x^k)$, onde $J(x) = \left[\frac{\partial f_i}{\partial x_j}\right]_{i,j}$ é a matriz Jacobiana do sistema.

3. Método do gradiente conjugado

Utiliza somente a primeira derivada da função objetivo, gerando uma seqüência de direções que são combinações lineares do gradiente:

$$d_{k+1} = \varepsilon_{k+1}\, d_k - \nabla S(x^{k+1}) \tag{44}$$

onde a nova direção é conjugada com a direção anterior com respeito a Hessiana, isto é:

$$(d_{k+1})^T\, H(x_k)\, d_k = 0 \tag{45}$$

e $x^{k+1} = x^k + \alpha_k\, d_k$, onde α_k é obtido de forma similar ao método da maior descida. Para calcular ε_{k+1}, faz-se a aproximação quadrática de $S(x)$, Equação (43), de onde obtém-se:

$$\nabla S(x) = \nabla S(x^k) + H(x^k)\,(x - x^k) \tag{46}$$

e portanto: $\nabla S(x^{k+1}) - \nabla S(x^k) = H(x^k)\,(x^{k+1} - x^k) = H(x^k)\,\alpha_k\, d_k$, que multiplicado por d_{k+1} à esquerda, resulta:

$$(d_{k+1})^T\,[\nabla S(x^{k+1}) - \nabla S(x^k)] = \alpha_k\,(d_{k+1})^T\,\mathrm{H}(x^k)\, d_k = 0$$

substituindo a Equação (44) na expressão acima, tem-se:

$$[\varepsilon_{k+1}\, d_k - \nabla S(x^{k+1})]^T\,[\nabla S(x^{k+1}) - \nabla S(x^k)] = 0\,,$$

mas devido as direções conjugadas: $(d_k)^T\,\nabla S(x^{k+1}) = 0$ e $\nabla^T S(x^{k+1})\,\nabla S(x^k) = 0$, resultando em:

$$\varepsilon_{k+1} = -\frac{\nabla^T S(x^{k+1})\nabla S(x^{k+1})}{d_k^T\,\nabla S(x^k)} = \frac{\nabla^T S(x^{k+1})\nabla S(x^{k+1})}{\nabla^T S(x^k)\nabla S(x^k)} \tag{47}$$

a última igualdade resulta do fato de $d_k = \varepsilon_k\, d_{k-1} - \nabla S(x^k)$, que multiplicado por $\nabla S(x^k)$ à direita:

$$(d_k)^T\,\nabla S(x^k) = \varepsilon_k\,(d_{k-1})^T\,\nabla S(x^k) - \nabla^T S(x^k)\,\nabla S(x^k) = -\nabla^T S(x^k)\,\nabla S(x^k),$$

pois $(d_{k-1})^T\,\nabla S(x^k) = 0$, pela mesma razão acima.

algoritmo

1) Escolher um ponto inicial x^o

2) Calcular $d_o = -\nabla S(x^o)$, $k = 0$

3) Encontrar α_k tal que $S(x^k + \alpha_k d_k) = \min_{\alpha>0} g_k(\alpha) = S(x^k + \alpha d_k)$

4) Calcular $x^{k+1} = x^k + \alpha_k d_k$ e $\nabla S(x^{k+1})$

5) **Se** o critério de convergência não foi satisfeito, **então** FIM.

6) Calcular $d_{k+1} = -\nabla S(x^{k+1}) + d_k \dfrac{\nabla^T S(x^{k+1}) \nabla S(x^{k+1})}{\nabla^T S(x^k) \nabla S(x^k)}$, $k \leftarrow k + 1$

7) **Se** $k = n$, isto é, realizou n direções L.I. **então** fazer $x^o = x^k$ e (ir para 2)
senão (ir para 3)

4. Métodos da métrica variável

A partir da Equação (46), pode-se tirar a seguinte relação:

$$x^{k+1} - x^k = \Delta x^k = [H(x^k)]^{-1} [\nabla S(x^{k+1}) - \nabla S(x^k)]$$

Fazendo uso de uma aproximação da inversa da matriz Hessiana:

$$[H(x^k)]^{-1} \approx \omega\, W(x^{k+1}) = \omega\, [W(x^k) + \Delta W(x^k)]$$

onde ω é um fator de escala e $\Delta W(x^k) = W(x^{k+1}) - W(x^k)$, resulta em:

$$\Delta x^k = \omega\, [W(x^k) + \Delta W(x^k)]\, [\nabla S(x^{k+1}) - \nabla S(x^k)] ,$$

que rearranjando tem-se:

$$\Delta W(x^k) \Delta f(x^k) = \frac{\Delta x^k}{\omega} - W(x^k) \Delta f(x^k) \qquad (48)$$

onde $\Delta f(x^k) = \nabla S(x^{k+1}) - \nabla S(x^k)$. Uma solução geral para a Equação (48) é dada por:

$$\Delta W(x^k) = \frac{1}{\omega} \frac{\{\Delta x^k, u^k\}}{< u^k, \Delta f(x^k) >} - \frac{\{W(x^k) \Delta f(x^k), v^k\}}{< v^k, \Delta f(x^k) >} \qquad (49)$$

onde u^k e v^k são vetores arbitrários,

$\{x, y\} = x\ y^T$ é o produto transposto (uma matriz)

$<x, y> = x^T y$ é o produto escalar (um escalar)

Conseqüentemente, pode existir uma infinidade de métodos que utilizam a Equação (38), dependendo da escolha de ω, u^k e v^k. O parâmetro α_k pode ser minimizado ("linesearch") ou escolhido adequadamente.

algoritmo

1) Escolher x^o, ω, u^k e v^k

2) Calcular $W(x^o)$, $k = 0$

2) Calcular $d_k = -W(x^k)\,\nabla S(x^k)$

3) Encontrar α_k tal que $S(x^k + \alpha_k\, d_k) = \min_{\alpha > 0} g_k(\alpha) = S(x^k + \alpha\, d_k)$

4) Calcular $x^{k+1} = x^k + \alpha_k\, d_k$, $S(x^{k+1})$ e $\Delta f(x^k) = \nabla S(x^{k+1}) - \nabla S(x^k)$

5) **Se** o critério de convergência foi satisfeito, **então** FIM.

6) calcular $W(x^{k+1})$, $k \leftarrow k + 1$, verificar a necessidade de *reset* e (ir para 3)

onde *reset* é uma reinicialização de $W(x^k)$ devido a problemas de convergência. No caso do método dos gradientes conjugados, o *reset* acontece a cada n iterações (n direções linearmente independentes).

Definindo:

$W = W(x^k)$	$W^{-1} = [W(x^k)]^{-1}$
$\Delta x = \Delta x^k$	$\Delta f = \Delta f(x^k)$
$d = \Delta x - W\,\Delta f$	$\hat{d} = \Delta f - W^{-1}\,\Delta x$
$\delta = \langle \Delta x, \Delta f \rangle$	$\hat{\delta} = \langle \Delta f , \Delta f \rangle$
$\gamma = \langle \Delta f , W\,\Delta f \rangle$	$\hat{\gamma} = \langle \Delta x, W^{-1}\,\Delta x \rangle$
$A = \{\Delta x, \Delta x\} / \delta$	$\hat{A} = \{\Delta f, \Delta f\} / \delta$
$B = \{W\,\Delta f, W\,\Delta f\} / \gamma$	$\hat{B} = \{W^{-1}\,\Delta x, W^{-1}\,\Delta x\} / \hat{\gamma}$
$E = \{\Delta x, \Delta f\}$	

é apresentado na Tabela 3, a seguir, alguns métodos de otimização que utilizam a primeira e/ou a segunda derivada da função objetivo. Os métodos de Greenstadt e Marquardt-Levenberg são variantes do método de Newton, que procuram manter a Hessiana sempre positiva definida, ou pela troca de sinal dos valores característicos ($C \rightarrow \underline{C}$) ou pela adição de β nos seus elementos da diagonal, respectivamente.

Tabela 3. Métodos da métrica variável

método	$W(x^{k+1})$
Steepest descent	I
Newton	$[H(x^{k+1})]^{-1}$
Greenstadt	$D^{-1}\,\underline{C}^{-1}\,D^{-1}$; $D = \text{diag}\,(\lvert W^{-1}\rvert_{i,j}^{1/2})$ e $C^{-1} = D\,W\,D$
Marquardt-Levenberg	$D^{-1}\,(C + \beta\,\text{I})^{-1}\,D^{-1}$; $\beta > -\text{min}$ (valor caract. de C)
Broyden	$W + \dfrac{\{d,d\}}{< d,\Delta f >}$
Davidon-Fletcher-Powell (DFP)	W + A – B
Pearson I	$W + \dfrac{\{d,\Delta x\}}{\delta} = W + A - \gamma\hat{B}^{-1}E^T$
Pearson II	$W + \dfrac{\{d,W\Delta f\}}{\gamma} = \left(1+\dfrac{\delta}{\gamma}\right)W - B$
Newton Projetado (ou Zoutendijk)	$W - B$
Greenstadt-Goldfarb I (GGI)	$W + \dfrac{1}{\hat{\delta}}\left(\{d,\Delta f\} + \{\Delta f,d\} - \dfrac{\delta}{\hat{\delta}} < \Delta f,d > \hat{A}\right)$
Greenstadt-Goldfarb II (GGII)	$W + \dfrac{EW + WE^T}{\gamma} - \left(1+\dfrac{\delta}{\gamma}\right)B$
Broyden-Fletcher-Goldfarb-Shanno (BFGS ou GGIII)	$W + \left(1+\dfrac{\gamma}{\delta}\right)A - \dfrac{EW + WE^T}{\delta}$
Fletcher ou $(\text{DFP})^{-1}$	$\left[W^{-1} + \left(1+\dfrac{\hat{\gamma}}{\delta}\right)\hat{A} - \dfrac{E^T W^{-1} + W^{-1}E}{\delta}\right]^{-1}$
$(\text{BFGS})^{-1}$	$\left[W^{-1} + \hat{A} - \hat{B}\right]^{-1}$
$(\text{Broyden})^{-1}$	$\left[W^{-1} + \dfrac{\{\hat{d},\hat{d}\}}{\{\hat{d},\Delta x\}}\right]^{-1}$
Goldstein-Price	aproximação por diferenças-finitas

4. Redes Neuronais

4.1 Generalidades

No Capítulo 1 foi feita uma abordagem generalizada das estruturas de modelos não lineares tipo "caixa-preta", usando uma nomenclatura clássica da matemática estatística. Para apresentar a estrutura de redes neuronais nesta mesma nomenclatura, é necessário definir alguns sinônimos freqüentemente utilizados nos textos de redes neuronais e lógica difusa:

- estimação = treinamento ("*train*"), aprendizagem ("*learn*");
- validação = generalização;
- estrutura do modelo = topologia ou arquitetura da rede;
- dados = conjunto ("*set*");
- sobre-ajuste = sobre-treinamento;
- sensitividade paramétrica = grau de ativação do neurônio;
- re-estimar = recordar ("*recall*");
- tamanho do passo = taxa de aprendizagem;
- parâmetros = pesos, *bias*, níveis ("*thresholds*");
- função base mãe, $\kappa(x)$ = função de transferência;
- função base, $g_k(\varphi) = \kappa(\varphi, \beta_k, \gamma_k)$ = neurônio ou elementos de processamento;
- regressor = padrão de entrada ("*input pattern*");
- sobre-relaxações sucessivas (SOR) = momentum;
- iteração = época ou repetição ("*epoch*");
- estimação recursiva = adaptação;
- uma estrutura particular de modelo = *perceptron* (adequado para classificação);
- função objetivo = função de desempenho;
- algoritmo de otimização = professor ("*teacher*");
- curva da função objetivo x iterações = curva de aprendizado;
- regra da cadeia eficiente = retro-propagação.

Nos textos sobre este assunto, tem-se uma definição de rede neuronal como:

"É um sistema de computação construído a partir de um número de nós ou elementos de processamento, ou ainda neurônios, simples e altamente interconectados, que processam informação pela resposta de seu estado dinâmico a entradas externas."

Mantendo a consistência com o Capítulo 1, a definição teria a seguinte forma:

"Rede neuronal é uma classe de estruturas de modelo não linear tipo caixa-preta."

Sendo que as funções bases são geralmente dos tipos *ridge* e radial. A Figura 5 ilustra a estrutura geral destas funções bases, onde v é o vetor de entradas (regressor), w_j é o vetor de pesos (parâmetro de escala para a função *ridge* e de deslocamento para a função radial), b_j é o *bias* (parâmetro de deslocamento para a função ridge ou de escala para a função radial), $\kappa(x_j)$ é a função de

transferência (função base mãe) e z_j é a saída (valor da função base g_j para um dado conjunto de parâmetros e regressor). O argumento da função de transferência, x_j, vai depender do tipo da função utilizada:

$x_j = w_j^T v + b_j$ (função "*ridge*") $x_j = b_j \|v - w_j\|$ (função radial).

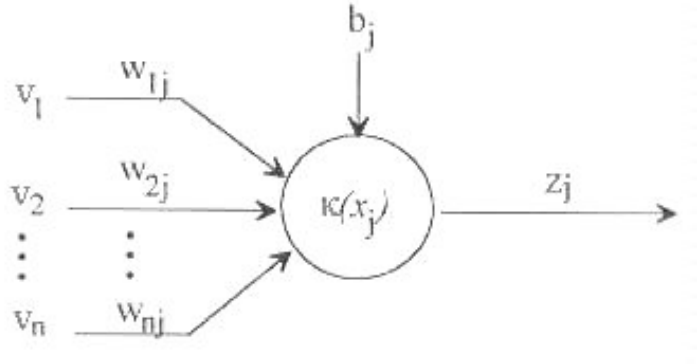

Figura 5. Função base $z_j = g_j(v,w_j,b_j)$.

4.2 Propriedades

1. As características do sistema são distribuídas ao longo de toda a rede, com alta uniformidade. Fazendo com que a remoção de uma função base tenha uma baixa taxa de degradação do ajuste. Também proporciona uma maior capacidade de filtrar ruídos e minimizar os efeitos de dados incompletos ou inconsistentes;
2. Capacidade e facilidade de reajuste dos parâmetros a novas situações sem grande investimentos;
3. Facilidade de obter informações das características do sistema através das interconexões entre as funções bases e seus pesos associados;
4. Possuiu uma abstração automatizada típica de estruturas tipo "caixa-preta", com pouca necessidade de especialistas;
5. Após ajuste, possui um grande potencial de utilização *on-line* devido seu baixo custo computacional, quando comparado com modelos não lineares teóricos;
6. Por resultar de uma expansão em funções bases, possui uma estrutura naturalmente paralela, que pode ser explorada nos sistemas de computação;
7. Estrutura MIMO;
8. Elevado número de parâmetros para serem ajustados, ocasionando um alto custo computacional durante a estimação. É recomendado na literatura, como ponto de partida, utilizar uma rede com duas camadas internas com 30 nós na primeira camada e 15 na segunda, o que resulta em um modelo com $495 + 31\ m + 16\ n$ para um sistema com m entradas e n saídas;
9. Devido ao elevado número de parâmetros, é também necessário um grande número de dados experimentais para o ajuste;
10. Por ser um modelo tipo "caixa-preta", apresenta baixa capacidade de extrapolação, sendo, portanto, necessário um projeto de experimento adequado ou um banco de dados com ampla faixa de operação;
11. As estimativas dos parâmetros não estão totalmente livres de erros sistemáticos (*bias*) e de soluções locais (mínimos locais);
12. A qualidade do ajuste depende fortemente do tipo de normalização dos dados de entrada e saída;
13. Modelo não linear.

4.3 Funções bases mães

Na figura abaixo são apresentados alguns tipos de funções bases mães usados em estruturas de redes neuronais.

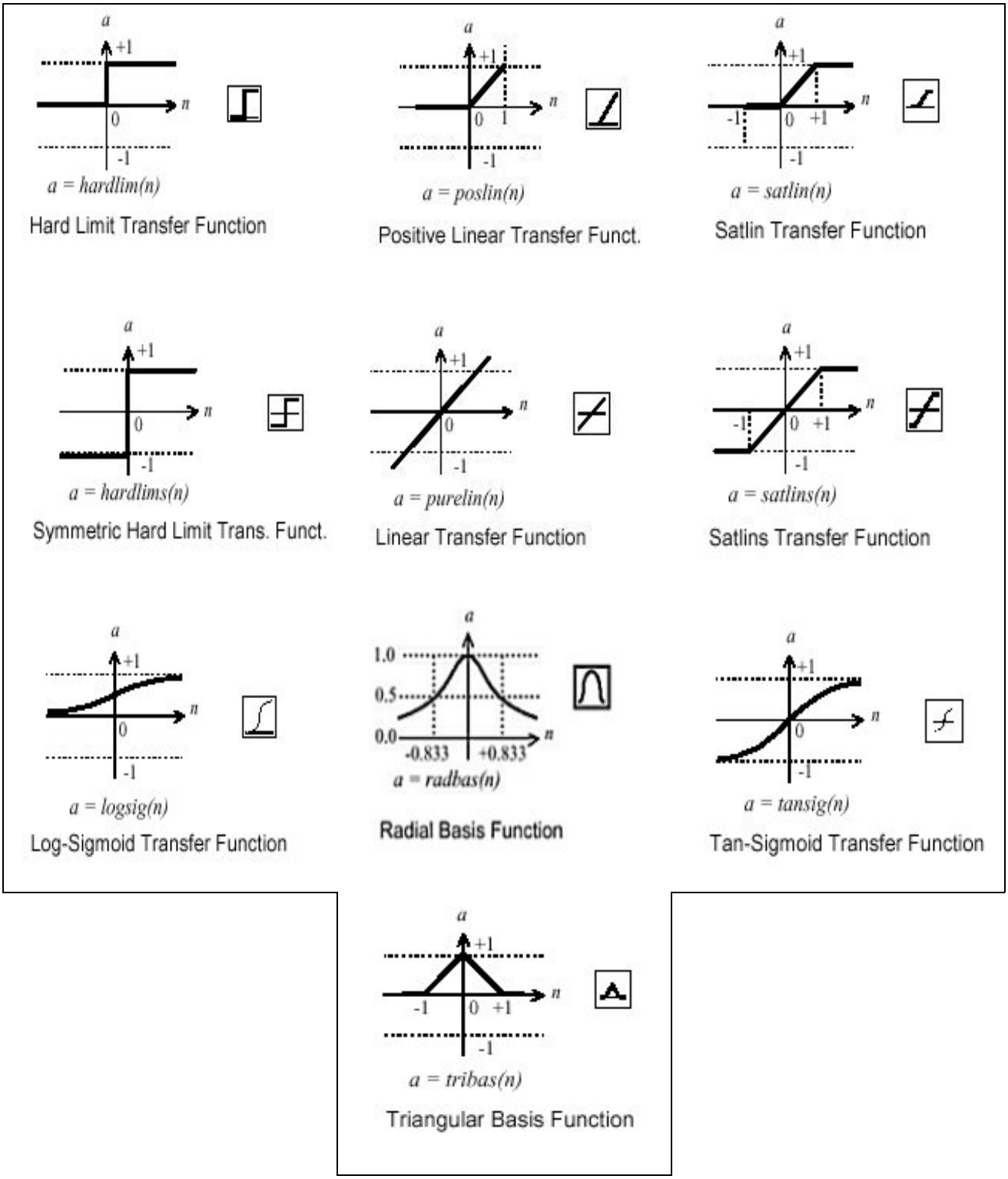

Figura 6. Alguns tipos de funções bases mães.

O argumento da função base depende do tipo de construção utilizado: produto tensorial, função radial, função *ridge*, etc. Este argumento é chamado de **ativação total** do nó na terminologia de redes. Outros termos utilizados são a **inibição** e **excitação** do nó dependendo se o argumento tende para -∞ ou +∞, respectivamente.

As funções bases sigmoide e tangente hiperbólica são geralmente utilizadas nas camadas internas das redes FNN ("*Feedforward Neural Network*") e RNN ("*Recurrent Neural Network*"), pois são funções contínuas, monótonas e finitas quando o argumento tende para ± ∞, e suas derivadas também são contínuas. Estas propriedades são favoráveis para um ajuste mais eficiente dos parâmetros da rede. A função gaussiana (radial) também apresenta as mesmas propriedades, com exceção da monotonicidade, sendo muito utilizada nas camadas internas das redes RBF ("*Radial-Basis-Function Network*"). A Tabela 4 apresenta as fórmulas destas funções.

Tabela 4. Fórmulas de algumas funções bases.

função base	**fórmula**	**derivada**	**intervalo**
sigmoide	$\kappa(x) = \frac{1}{1+e^{-x}}$	$\kappa'(x) = \kappa(x)[1-\kappa(x)]$	$0 < \kappa(x) < 1$
tangente hiperbólica	$\kappa(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	$\kappa'(x) = 1-[\kappa(x)]^2$	$-1 < \kappa(x) < 1$
gaussiana	$\kappa(x) = \exp\left(-\frac{x^2}{2}\right)$	$\kappa'(x) = -x\,\kappa(x)$	$0 < \kappa(x) < 1$

Comparando a função sigmoide com a tangente hiperbólica tem-se o seguinte:

a) a inclinação da tangente hiperbólica em torno do ponto de inflexão é muito maior que a da sigmoide. Deste modo a tangente hiperbólica pode distinguir melhor entre pequenas variações no valor de entrada, gerando uma resposta mais não linear. Para a região em torno da origem, a resposta da tangente hiperbólica é aproximadamente 4 vezes maior que a sigmoide. A Figura 7 mostra estas duas funções superpostas;

b) a tangente hiperbólica tem uma resposta com o mesmo sinal do argumento, ao passo que a sigmoide sempre tem resposta positiva.

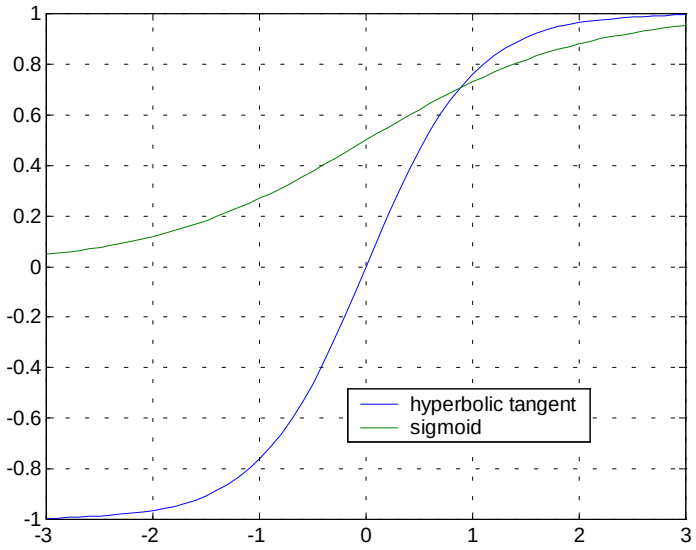

Figura 7. Funções sigmoide e tangente hiperbólica.

4.4 Estrutura das redes

A estrutura (**topologia**) da rede refere-se a como as funções bases (nós) estão interconectadas. Esta estrutura é formada pela organização dos nós em camadas, conectando-os e ponderando as interconexões (combinação dos regressores com o uso de alguma construção, ex: *ridge*, radial). Na Figura 8 estão ilustradas as diferentes formas de conexões: **intra-camadas**, **inter-camadas** e **recorrentes**, dependendo se um regressor (entrada) de uma função base origina-se de outra função base (saída) da mesma camada, de outra camada ou da própria função base, respectivamente.

As conexões inter-camadas podem ainda ser do tipo ***feedforward*** ou ***feedback***, dependendo se o sentido da conexão é da entrada para a saída ou vice-versa, respectivamente. A Figura 9 ilustra estas duas situações.

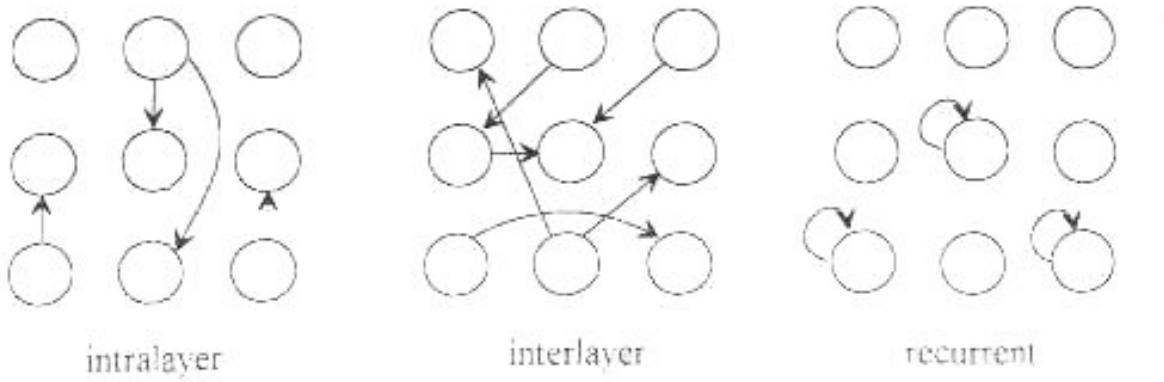

Figura 8. Opções de conexão em uma rede.

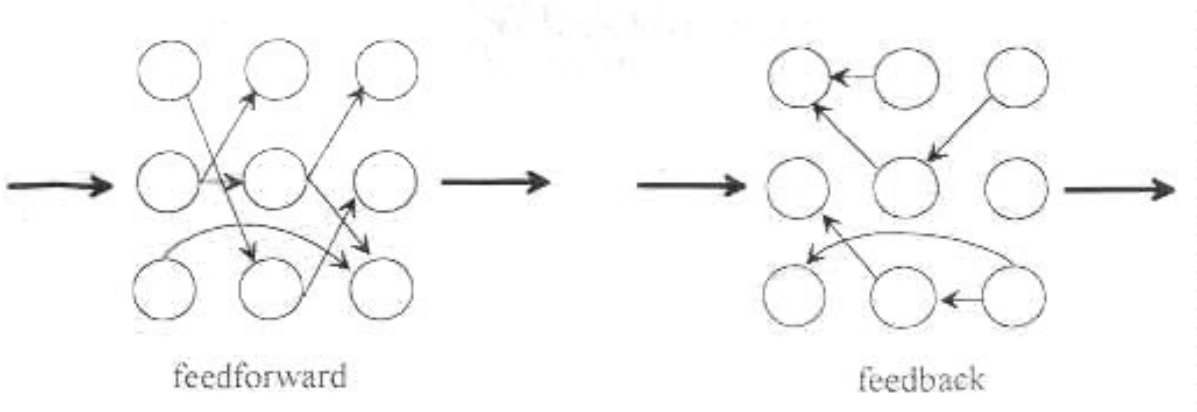

Figura 9. Conexões *feedforward* e *feedback*.

Uma forma de se referenciar ao tamanho da rede é através da **configuração das camadas internas** (ou escondidas). Por exemplo, uma estrutura do tipo 30:20:10 está associada a uma rede com três camadas internas, contendo 30 funções bases na primeira camada, 20 na segunda e 10 na terceira camada, **numerando da entrada para a saída**.

4.5 Alguns tipos de redes

1. Redes de Classificação (RBF)

Geralmente utilizada para identificar falhas operacionais, para categorizar diferentes características de processos estacionários ou transientes e para determinar a robustez do sistema a perturbações. Estes tipos de problemas sugerem naturalmente o uso de funções bases radiais. A Figura 8 ilustra a estrutura de uma rede RBF ("*Radial-Basis-Function Network*") com uma camada

interna. Na camada de saída tem-se funções *ridge* com a função base mãe na forma de tangente hiperbólica.

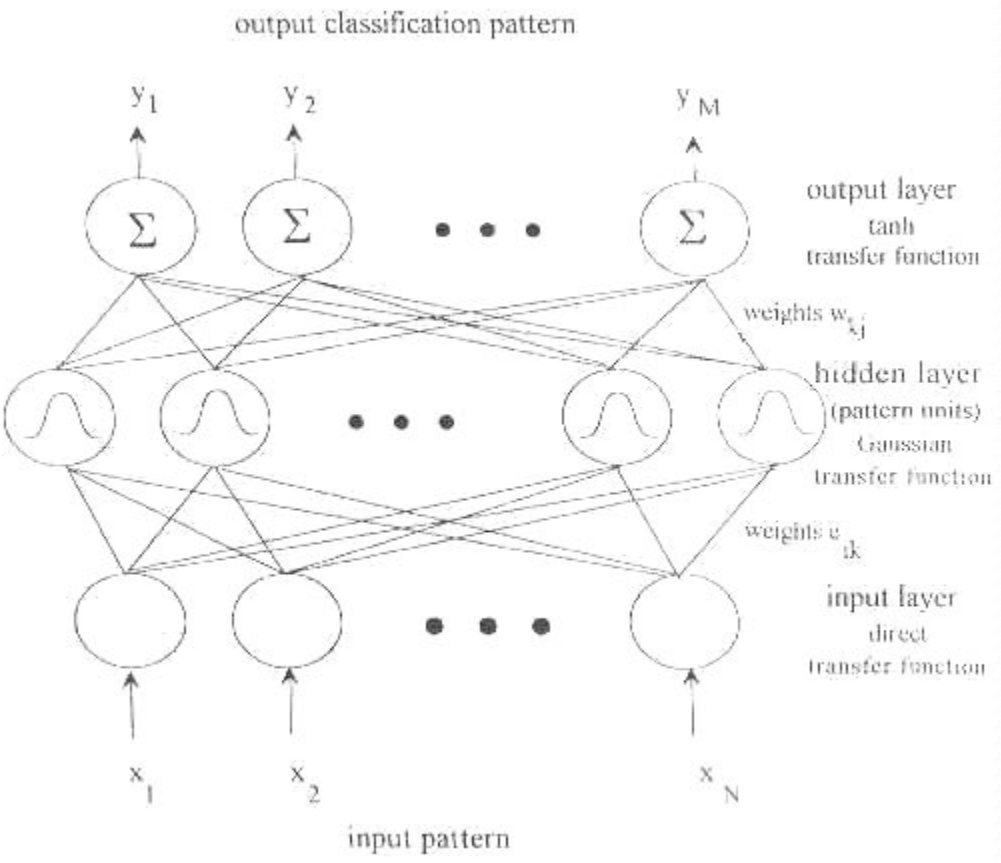

Figura 8. Rede RBF com *N* entradas e *M* saídas e com uma camada interna.

2. Redes *Feedforward* (FNN)

Muito utilizada para predizer valores de variáveis relacionadas com o desempenho de um processo, através de um conjunto de variáveis operacionais provenientes de dados de planta ou de laboratório. Combinadas com métodos estatísticos, pode ser também utilizada para aplicações de controle de qualidade na identificação das principais variáveis operacionais e suas regiões de operação na otimização do desempenho do processo. Outra aplicação importante é a construção de "*software-based sensors*" (ou "*soft sensors*"). A Figura 2 ilustra a estrutura de uma rede FNN. Este tipo de rede é também conhecida, equivocadamente, como rede de retro-propagação, pois este termo está relacionado com o método utilizado para a estimação dos parâmetros.

3. Redes Recorrentes (RNN)

Também são redes utilizadas para predição, mas com uma estrutura modificada, usando laços *feedback* ou recorrentes, para manipular dados dependentes do tempo. Este tipo de rede utiliza valores passados de entrada e saída da rede para predição da resposta do sistema para determinadas condições de operação. A Figura 9 ilustra a estrutura de uma rede RNN discreta.

As redes recorrentes de Grossberg/Hopfield possuem um estrutura similar a da Figura 9, mas representam as dinâmicas de entrada e saída das funções bases por meio de equações diferenciais ordinárias de primeira ordem.

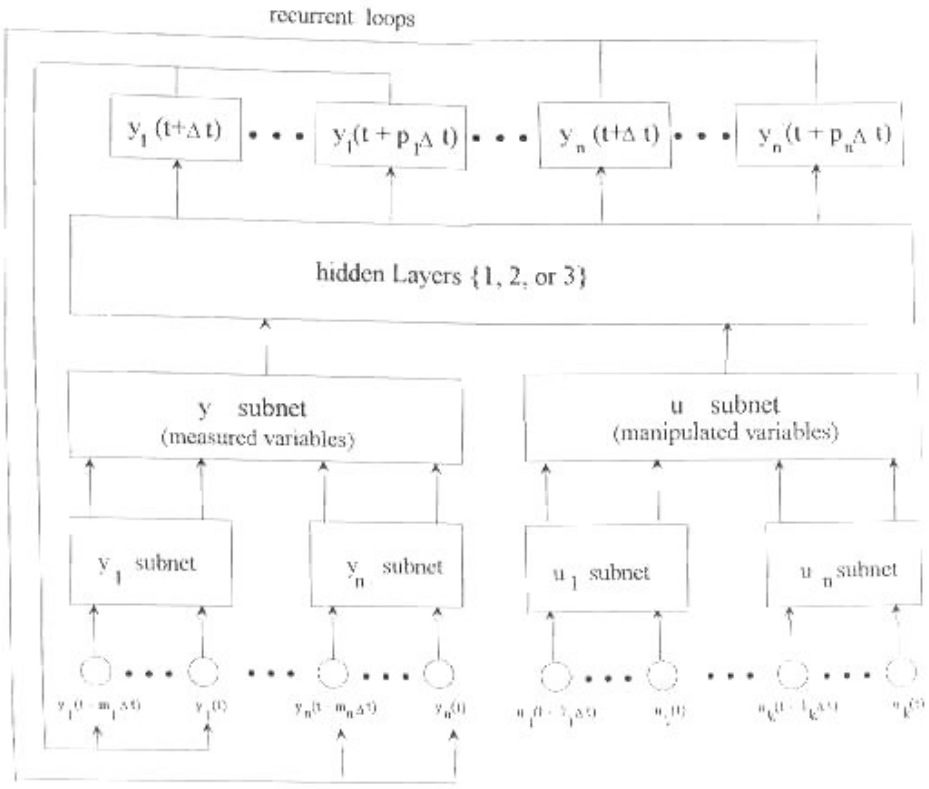

Figura 9. Rede RNN para processos transientes.

4.6 Normalização

A normalização dos dados de entrada e saída de uma rede é uma fator fundamental para a qualidade da estimação dos parâmetros. Isto se torna ainda mais vital com o uso de funções bases localmente limitadas, como é o caso das funções sigmoide e tangente hiperbólica, pois elas perdem a sensibilidade a valores elevados em seus argumentos. Além disto, quando usadas na camada de saída, limitam os valores de saída para o seus domínios (0, 1) e (-1, 1), respectivamente, o que torna estes valores dependentes da normalização utilizada.

Os tipos de normalização mais utilizados são (ilustrados nas páginas seguintes):

1. Intervalo [0,1] absoluto: $x_{i,norm} = \dfrac{x_i}{x_{i,\max}}$ (50)

2. Intervalo [0,1] relativo: $x_{i,norm} = \dfrac{x_i - x_{i,\min}}{x_{i,\max} - x_{i,\min}}$ (51)

3. Intervalo [-1,1] relativo à média: $x_{i,norm} = \dfrac{x_i - x_{i,avg}}{R_{i,\max}}$ (52)

onde $R_{i,\max} = \max(x_{i,\max} - x_{i,avg}\,,\, x_{i,avg} - x_{i,\min})$ e $x_{i,avg}$ é o valor médio de x_i.

O problema da primeira normalização é que, quando o valor mínimo de $x_i > 0$, ela não utiliza todo o intervalo de variação significante da função base. Por outro lado, a interpretação dos resultados fica mais direta, sem a necessidade de transformações dos dados. A última normalização pode resultar em um ajuste mais rápido dos parâmetros, pois se o valor médio corresponder a uma condição normal de operação, e ambos os dados de entrada e saída usarem o mesmo tipo de normalização, então os parâmetros serão ajustados para compensarem os desvios dos valores nominais.

4.7 Estimação dos parâmetros da rede

Para exemplificar o processo de estimação dos parâmetros de uma rede, é mostrado a seguir o método do gradiente aplicado a uma rede FNN com uma camada interna (Figura 10) e com funções *ridge*, onde fica claro o uso eficiente da regra da cadeia (referenciada como retro-propagação).

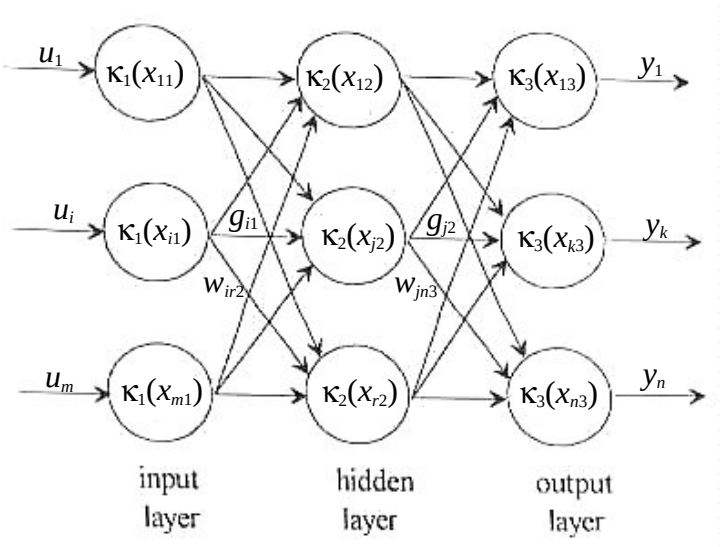

Figura 10. Rede FNN com uma camada interna (sem a representação do bias).

Usando o método dos mínimos quadrados, a função objetivo tem a forma:

$$S(w) = \frac{1}{\nu}\sum_{i=1}^{N}\sum_{k=1}^{n}\varepsilon_{k,i}^2 = \frac{1}{\nu}\sum_{i=1}^{N}\sum_{k=1}^{n}[y_{k,i} - \hat{y}_{k,i}(w)]^2 \tag{53}$$

De modo a facilitar o entendimento, sem perda de generalidade, será considerado somente um vetor de entrada (N = 1), eliminando assim o índice k da Equação (53), e ν = 2 para ser cancelado no cálculo do gradiente. Com isto, a função objetivo se transforma em:

$$S(w) = \frac{1}{2}\sum_{k=1}^{n}\varepsilon_k^2 = \frac{1}{2}\sum_{k=1}^{n}[y_k - \hat{y}_k(w)]^2 \tag{54}$$

Apesar do bias da função base não ter sido representado na Figura 10, ele será tratado como um parâmetro a ser estimado na forma implícita:

$$g_{ic} = \kappa_c(x_{ic}). \quad , \quad x_{ic} = \sum_{k=0}^{s} w_{kic}\, g_{k,c-1} \tag{55}$$

onde o índice c refere-se ao número da camada;
o índice i refere-se à função base;
o índice k refere-se às "s" entradas da função base;
quando k = 0 tem-se w_{0ic} = bias e $g_{0,c\text{-}1}$ = 1;
x_{i1} pode ser tratado como o próprio u_i, ou como uma fórmula de normalização do mesmo;
$\hat{y}_k = g_{k3}$.

Usando o método do gradiente para o ajuste dos parâmetros, tem-se:

$$w^{h+1} = w^h - \eta_h \nabla S(w^h) \tag{56}$$

onde η_h é o tamanho do passo na h-ésima iteração (ver Capítulo 3 para sua determinação).

Aplicando a regra da cadeia para o cálculo do gradiente da função objetivo em relação aos parâmetros resulta em:

$$\frac{\partial S}{\partial w_{jp3}} = -\sum_{k=1}^{n} \varepsilon_k \frac{\partial y_k}{\partial x_{k3}} \frac{\partial x_{k3}}{\partial w_{jp3}} = -\varepsilon_p \frac{d\kappa_3}{dx_{p3}} g_{j2} \quad , \quad \frac{\partial x_{k3}}{\partial w_{jp3}} = \delta_{kp} g_{j2} \tag{57}$$

$$\frac{\partial S}{\partial w_{iq2}} = -\sum_{k=1}^{n} \varepsilon_k \frac{\partial y_k}{\partial x_{k3}} \sum_{j=1}^{r} \frac{\partial x_{k3}}{\partial g_{j2}} \frac{\partial g_{j2}}{\partial x_{j2}} \frac{\partial x_{j2}}{\partial w_{iq2}} = -\sum_{k=1}^{n} \varepsilon_k \frac{d\kappa_3}{dx_{k3}} w_{qk3} \frac{d\kappa_2}{dx_{q2}} g_{i1} \ , \ \frac{\partial x_{j2}}{\partial w_{iq2}} = \delta_{jq} g_{i1} \tag{58}$$

onde δ_{ij} é a função delta de Kronecker ($\delta_{ij} = 1$ se $i = j$ e $\delta_{ij} = 0$ se $i \neq j$).

Utilizando a Equação (56) do método do gradiente tem-se:

$$w_{jp3}^{h+1} = w_{jp3}^{h} + \eta_h \left(\varepsilon_p \frac{d\kappa_3}{dx_{p3}} g_{j2} \right)_h \quad , \quad j = 0,1,...,r \quad \text{e} \quad p = 1,2,...,n \tag{59}$$

$$w_{iq2}^{h+1} = w_{iq2}^{h} + \eta_h \left(\sum_{k=1}^{n} \varepsilon_k \frac{d\kappa_3}{dx_{k3}} w_{qk3} \frac{d\kappa_2}{dx_{q2}} g_{i1} \right)_h \quad , \quad i = 0,1,...,m \quad \text{e} \quad q = 1,2,...,r \tag{60}$$

O termo “retro-propagação do erro através da rede”, utilizado na terminologia de redes, resulta deste aumento gradual de termos devido à regra da cadeia (carregando os erros de predição) a medida que se caminha no sentido da saída para a entrada. É comum também chamar os termos abaixo, ξ, que multiplicam os regressores da respectiva camada, de equações do erro LMS (“Least-Mean-Squares”).

$$\xi_{p3} = \varepsilon_p \frac{d\kappa_3}{dx_{p3}} \quad \rightarrow \quad w_{jp3}^{h+1} = w_{jp3}^{h} + \eta_h \left(\xi_{p3} g_{j2} \right)_h \tag{61}$$

$$\xi_{q2} = \sum_{k=1}^{n} \varepsilon_k \frac{d\kappa_3}{dx_{k3}} w_{qk3} \frac{d\kappa_2}{dx_{q2}} \quad \rightarrow \quad w_{iq2}^{h+1} = w_{iq2}^{h} + \eta_h \left(\xi_{q2} g_{i1} \right)_h \tag{62}$$

Por outro lado, pela Equação (27) tem-se:

$$\frac{\partial^2 S}{\partial \hat{y}_p \, \partial w_{jp3}} = \frac{d\kappa_3}{dx_{p3}} g_{j2} \quad \rightarrow \quad w_{jp3}^{h+1} = w_{jp3}^{h} + \eta_h \left(\varepsilon_p \frac{\partial^2 S}{\partial \hat{y}_p \, \partial w_{jp3}} \right)_h \tag{63}$$

$$\frac{\partial^2 S}{\partial \hat{y}_k \, \partial w_{iq2}} = \frac{d\kappa_3}{dx_{k3}} w_{qk3} \frac{d\kappa_2}{dx_{q2}} g_{i1} \quad \rightarrow \quad w_{iq2}^{h+1} = w_{iq2}^{h} + \eta_h \left(\sum_{k=1}^{n} \varepsilon_k \frac{\partial^2 S}{\partial \hat{y}_k \, \partial w_{iq2}} \right)_h \tag{64}$$

que mostram a direção da correção dos erros de predição.

Os gradientes das saídas em relação às entradas, ou matriz de transferência do processo (e matriz dos ganhos do processo para o estado estacionário), também pode ser obtida usando a regra da cadeia, resultando em:

$$\frac{\partial y_k}{\partial u_i} = \frac{\partial y_k}{\partial x_{k3}} \sum_{j=1}^{r} \frac{\partial x_{k3}}{\partial g_{j2}} \frac{\partial g_{j2}}{\partial x_{j2}} \sum_{q=1}^{m} \frac{\partial x_{j2}}{\partial g_{q1}} \frac{\partial g_{q1}}{\partial x_{q1}} \frac{\partial x_{q1}}{\partial u_i} = \frac{d\kappa_3}{dx_{k3}} \sum_{j=1}^{r} w_{jk3} \frac{d\kappa_2}{dx_{j2}} w_{ij1} \frac{d\kappa_1}{dx_{i1}} \frac{dx_{i1}}{du_i} \tag{65}$$

onde o valor de $\frac{dx_{i1}}{du_i}$ vai depender da definição de x_{i1} (fórmula de normalização).

Como todo método iterativo, a estimação dos parâmetros destes modelos não lineares necessita de uma estimativa inicial. Como em uma estrutura de rede convencional todas as funções bases de uma dada camada tem os mesmos regressores, então uma estimativa inicial dos parâmetros com uma distribuição uniforme seria muito natural. Contudo, este tipo de distribuição pode não ser muito eficiente, pois como todas as funções bases iniciam com o mesmo grau de importância, a velocidade de convergência para uma solução, geralmente com pesos diferenciados entre estas funções, provavelmente será baixa. Outras alternativas mais eficientes são distribuições não uniformes regulares (como por exemplo a gaussiana) ou completamente aleatórias, em geral dentro do intervalo [-1, 1].

Outra observação sobre a estimação dos parâmetros diz respeito a reformulação da função objetivo. Uma prática usada em redes neuronais é a penalização da função objetivo com a grandeza dos parâmetros, conhecida como **regularização**, que modifica a função objetivo ponderando a média dos quadrados dos parâmetros, isto é:

$$S(w) = \frac{\omega}{\nu} \sum_{i=1}^{N} \sum_{k=1}^{n} \varepsilon_{k,i}^2 + \frac{(1-\omega)}{p} \sum_{j=1}^{p} w_j^2 \tag{66}$$

onde ω é conhecida como razão de desempenho e w_j é o vetor de todos os p parâmetros da rede. Esta regularização resulta em uma rede com parâmetros menores (em valor absoluto), suavizando a resposta da rede e reduzindo a possibilidade de sobre-ajuste em detrimento da qualidade do ajuste aos pontos observados.

5. Redes de Modelos Locais

(artigos [6,7] + Maurício)

Referências:

[1] D.R. Baughman, Y.A. Liu, “Neural Networks in Bioprocessing and Chemical Engineering”, Academic Press, San Diego, 1995.

[2] L. Ljung, “System Identification: Theory for the User”, Prentice-Hall, New York, 1987.

[3] A.R. Secchi, “Estimação de Parâmetros”, Notas do Curso de Especialização em Processos Químicos”, COPPE/UFRJ, OPP Petroquímica S.A., Triunfo, RS, 1997.

[4] A.R. Secchi, “Técnicas de Otimização”, Notas do Curso de Especialização em Processos Petroquímicos”, UFRGS, Porto Alegre, RS, 1997.

[5] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarsson, A. Juditsky, “Nonlinear Black-box Modeling in System Identification: a Unified Overview”, Automatica, **31** (12) 1691-1724 (1995).

[6] J.O. Trierweiler, U. Neumann, “Rede de Modelos Locais: Uma solução simples para problemas complexos” , COBEQ 98, Trab338, Porto Alegre, Brazil (1998).

[7] J.O. Trierweiler, A.R. Secchi, “Exploring the Potentiality as Using Multiple Model Approach in Nonlinear Model Predictive Control”, in “Progress in Systems and Control Theory” (Ed. F. Allgöwer and A. Zheng) (1999).