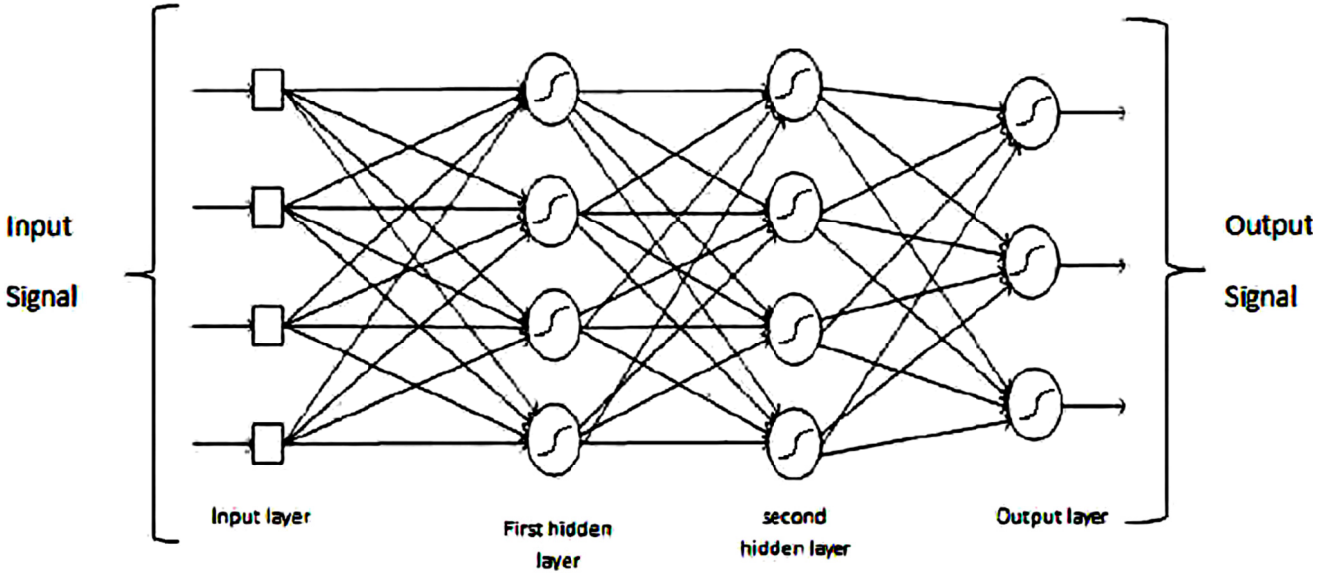
Radial Basis Functions

Star Trails over Mexican Hat Rock, Utah

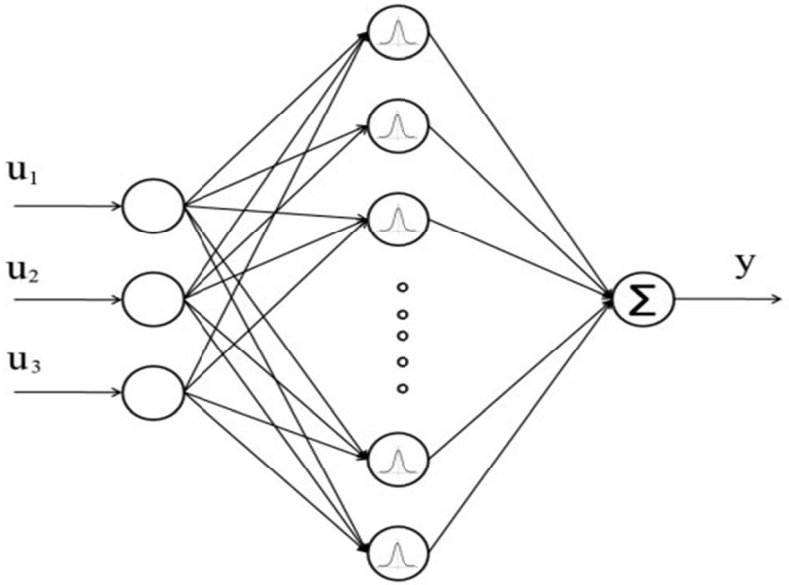
Fonte [3]

Multilayer Perceptrons x Radial Basis Function

RNA MLP



RNA RBF

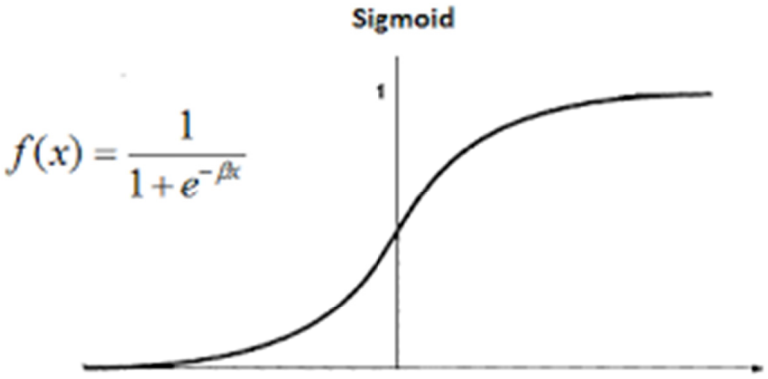


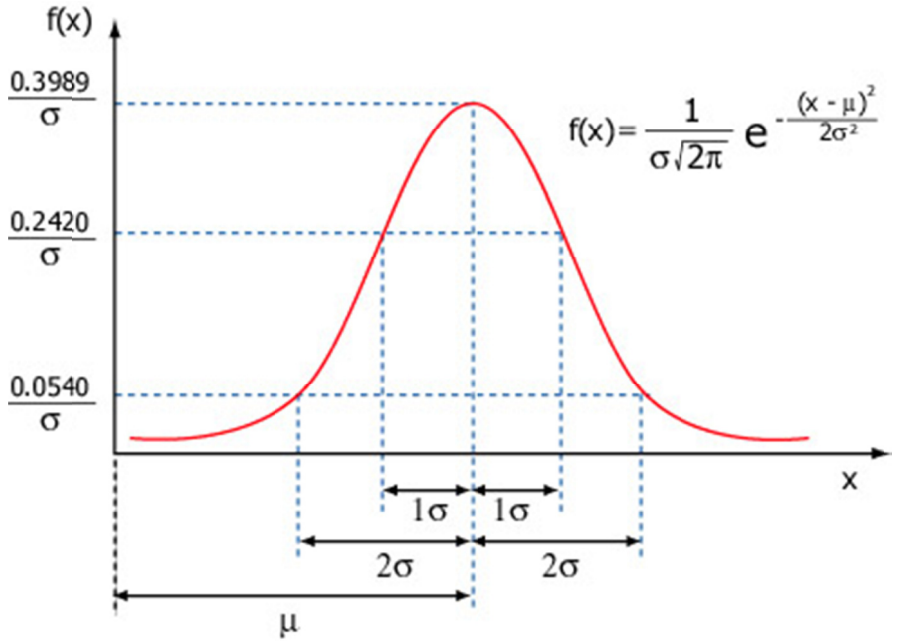
- ➔ Ambas são redes supervisionadas
- ➔ Ambas são aproximadoras universais
- ➔ Mas as arquiteturas e algoritmos de aprendizagem são muito diferentes

Diferenças entre MLPs e RBFs:

1. Unidades escondidas em MLPs dependem de somas ponderadas das entradas, transformadas por funções de ativação monotônicas (Função Sigmoidal, não-linear e continuamente diferenciável: Função Logística e Função Tangente Hiperbólica.)

   Em RBFs, a ativação de uma unidade escondida é determinada por uma função não-linear da distância entre o vetor de entrada e um vetor de referência. As unidades escondidas de uma RBF possuem funções de ativação que são funções localizadas e que apresentam base radial definida sobre seu domínio.





2. MLP forma uma representação distribuída no espaço de valores de ativação para as unidades escondidas: Para um dado vetor de entrada, muitas unidades escondidas contribuem para a determinação do valor de saída, razão pela qual um MLP tende a resultar em uma aproximação global que contempla estatísticas de ordem superior.

   Durante o treinamento, as funções representadas pelas unidades escondidas são tais que, quando linearmente combinadas pela camada final de pesos, gerem as saídas corretas para um intervalo grande de possíveis valores de entrada. A "interferência" e o "acoplamento cruzado" entre as unidades escondidas levam a resultados durante o processo de treinamento da rede que são altamente não-lineares, resultando em problemas de mínimos locais ou em regiões quase planas na função de erro, fatores estes que podem levar a uma convergência muito lenta do procedimento de treino, mesmo com a utilização de estratégias avançadas de otimização.

   RBF, com funções de base localizadas, forma uma representação no espaço de unidades escondidas que é local com respeito ao espaço de entrada porque, para um dado vetor de entrada, tipicamente apenas algumas unidades escondidas apresentarão ativações significantes. Por esta razão, uma RNA RBF tende a produzir uma aproximação local do processo que está sendo aprendido.

3. MLPs têm muitas camadas e um complexo padrão de conectividade.

   RBFs têm uma arquitetura simples, consistindo de duas camadas, em que a primeira camada contém os parâmetros das funções de base radial e a segunda camada forma combinações lineares das ativações das funções de base radial para gerar a saída.

4. Os parâmetros em uma rede MLP são ajustados no âmbito de uma única estratégia global de aprendizado supervisionado. Este tipo de treinamento apresenta um relativamente alto custo computacional, decorrente da necessidade de retro-propagação do erro, o que faz com que as redes MLPs apresentem uma característica de aprendizado lenta. No entanto, a performance de generalização de uma rede MLP é, em geral, robusta.

   A arquitetura de uma rede RBF permite que o aprendizado ser efetuado em dois estágios, com os vetores-centro das funções de base radial sendo determinados primeiramente por meio de técnicas não-supervisionadas (algoritmo K-means, por exemplo), usando para tal apenas os dados de entrada e, subsequentemente, adaptando todos os parâmetros livres da RBF (vetores-centro e variâncias das funções de base radial e sinapses) por métodos supervisionados, de rápida convergência.

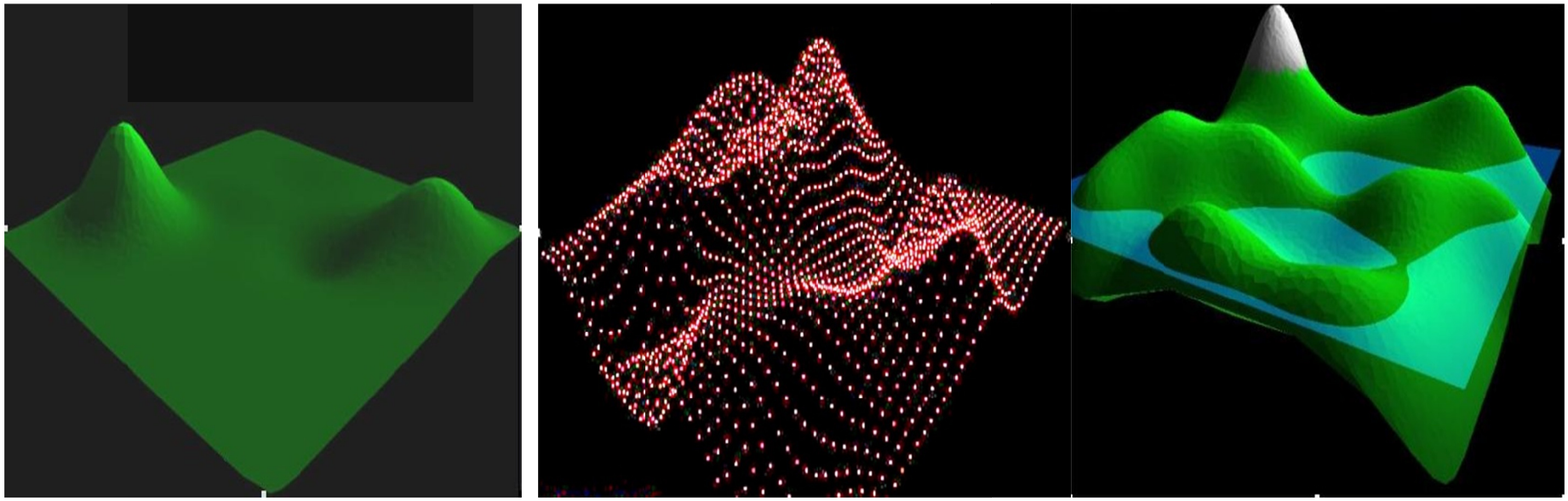
   A diferente estratégia de treinamento e a consequente diferença de velocidade de treinamento entre as duas redes faz com que as redes MLP se mostrem **menos** adequadas do que as redes RBF, quando estivermos interessados em operações dinâmicas, que envolvam aprendizado continuado (como, por exemplo, em predição de séries temporais, equalização de canal e aplicações dinâmicas *on-the-fly*).

5. No contexto de aproximação de funções, sob idênticas condições do ambiente no qual estão inseridas, de uma forma geral pode-se afirmar que:
   - o erro final atingido por uma rede RBF é menor do que o erro final atingido por uma rede MLP;
   - a convergência de uma rede RBF pode chegar a ser uma ordem de grandeza mais rápida do que a convergência de uma rede MLP;
   - a capacidade de generalização de uma RNA MLP é, em geral, superior à capacidade de generalização de uma RNA RBF.

Interpretação do aprendizado e generalização em uma rede neural RBFs:

Aprendizado: equivale a ajustar uma superfície não-linear ao conjunto de dados de treino, em um espaço multi-dimensional, de acordo com algum critério estatístico.

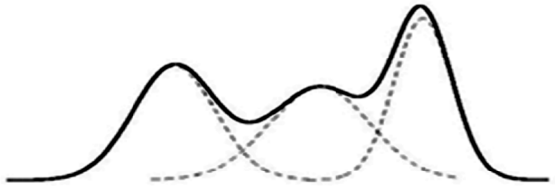
Generalização: equivale a usar esta superfície multi-dimensional para interpolar outros pontos que não pertençam ao conjunto de treino, mas estejam em sua vizinhança.

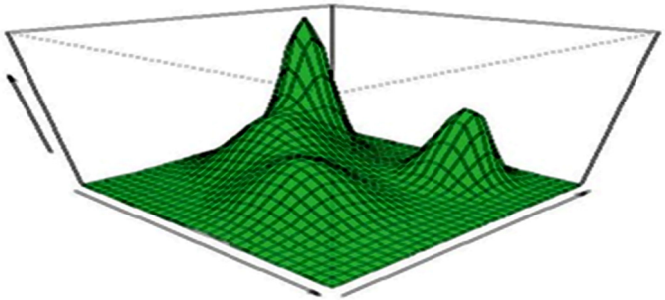


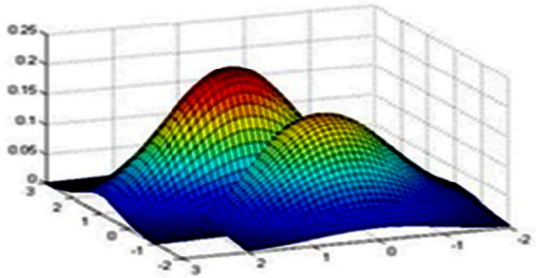
Fontes: [1] e [2]

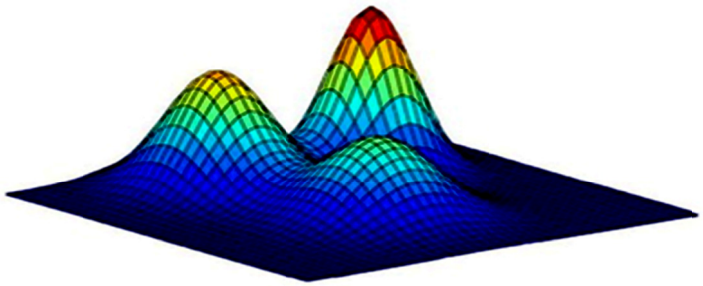
Aproximação de funções e de processos com redes neurais RBFs

- RBFs aproximadoras possuem em sua arquitetura uma camada escondida cujas funções de ativação dos neurônios são funções de base radial, as quais derivam seu nome da representação gráfica da função (ver figura).
- Os neurônios da camada escondida de uma rede RBF representam um conjunto de funções que constitui uma base arbitrária no espaço por eles formado, e em cujo espaço o conjunto de entrada pode ser expandido. Os dados representados através de RBFs são, portanto, expandidos com referência a um conjunto finito de funções de base radial, análogo à expansão através de senos e cossenos em Séries de Fourier.
- Cada função de base radial é centrada em uma particular coordenada do espaço multi-dimensional do conjunto de vetores (dados) de entrada. Cada uma destas coordenadas usualmente define o centro de uma – dentre várias possíveis – região de maior aglomeração de pontos (*cluster*) do espaço de dados de entrada.









Historicamente, RBFs foram originalmente desenvolvidas p/ interpolação multi-dimensional

Segundo B. Mulgrew, o problema da interpolação de dados pode ser assim formulado: dado um conjunto de vetores $\{\underline{u}_j\}$, $\underline{u} \in \Re^M$, e um conjunto de respectivos escalares $\{y_j\}$, busca-se uma função $F(.)$, tal que,

$$y_j = F(\underline{u}_j), \ \forall j \tag{5.1}$$

Mas, desde que definida analiticamente, é factível que a função $F(.)$ possa mapear vetores $\underline{u} \in \Re^M$ que não pertençam ao conjunto original no conjunto de pontos *y* associados, generalizando a aproximação.

Uma possível solução para o mapeamento analítico é escolher $F(\underline{u})$, tal que:

$$F(\underline{u}) = \sum_{k=0}^{K-1} w_k \ \phi\left(\left\| \underline{u} - \underline{u}_k \right\|^2 \right) \tag{5.2}$$

onde $\phi\left(\left\| \underline{u} - \underline{u}_k \right\|^2 \right)$ é uma função escalar radialmente simétrica, tendo $\underline{u}_k$ como centro da k –ésima função de base radial, $k = 0,1,\cdots K-1$.

Observação: O operador $\|\cdot\|$ é usualmente a norma Euclidiana – ou Norma L2 – e mede o módulo do vetor argumento, isto é, a distância Euclidiana da ponta do vetor à sua origem.

A norma Euclidiana de $\underline{u} \in \Re^M$ é expressa por $\|\underline{u}\| = \sqrt{\sum_{m=0}^{M-1} (u_m)^2} = \sqrt{\underline{u}^T \cdot \underline{u}}$

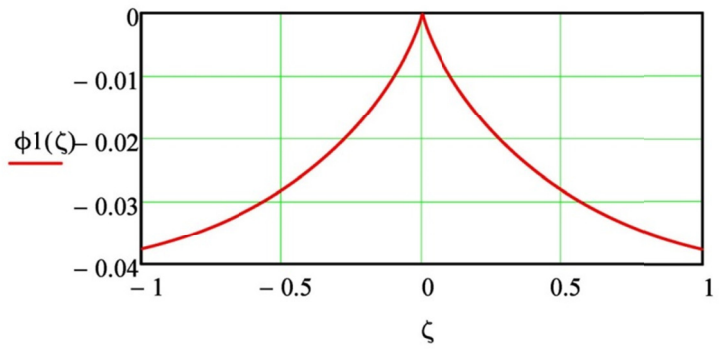
Funções analíticas (dentre várias) usualmente utilizadas como funções de base radial

Spline

$$\phi 1(\zeta) := \frac{|\zeta|}{\sigma^2} \cdot \log\left(\frac{|\zeta|}{\sigma}\right)$$

$\sigma := 4 \quad \Delta := 1$

$\zeta := -\Delta, -0.999 .. \Delta$

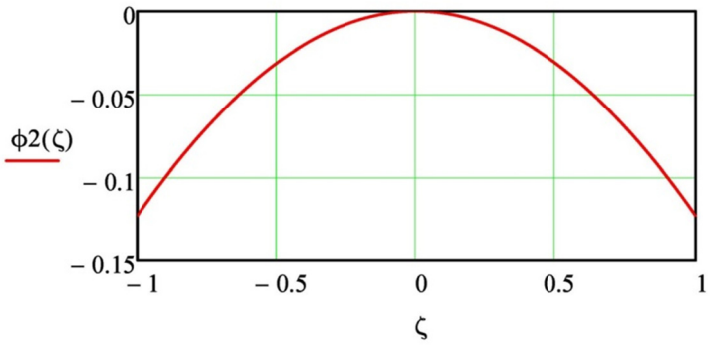


Multi-Quadrática

$$\phi 2(\zeta) := \sigma - \sqrt{\zeta^2 + \sigma^2}$$

$\sigma := 4 \quad \Delta := 1$

$\zeta := -\Delta, -0.999 .. \Delta$

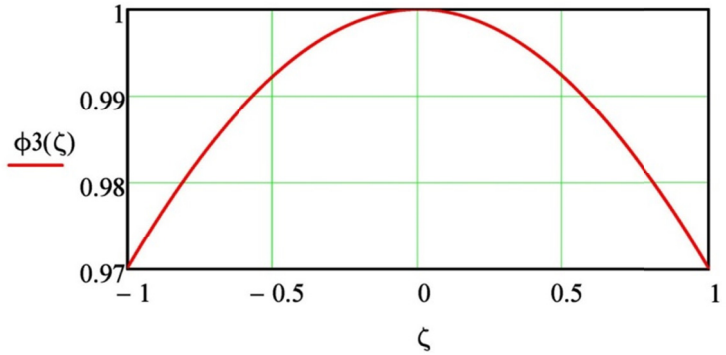


Multi-quadrática inversa

$$\phi 3(\zeta) := \frac{\sigma}{\sqrt{\zeta^2 + \sigma^2}}$$

$\sigma := 4 \quad \Delta := 1$

$\zeta := -\Delta, -0.999 .. \Delta$

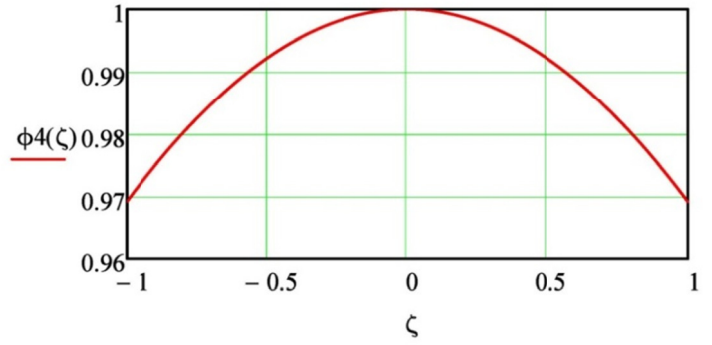


Gaussiana

$$\phi 4(\zeta) := e^{\left[-\left(\frac{\zeta}{\sqrt{2}\cdot\sigma}\right)^2\right]}$$

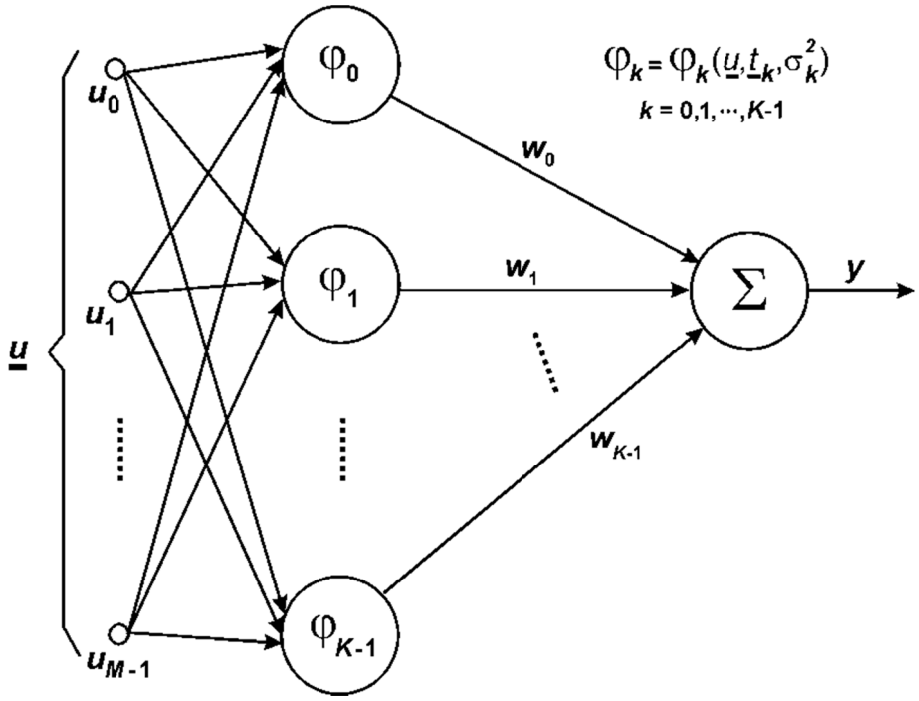
$\sigma := 4 \quad \Delta := 1$

$\zeta := -\Delta, -0.999 .. \Delta$



O parâmetro σ determina o raio da região de sensibilidade no domínio da k –ésima função de base radial, em torno de seu centro $\underline{u}_k$, região cujos vetores $\underline{u}$ a ela pertencentes resultam em magnitude significativa da resposta da k –ésima função. Por exemplo , um vetor $\underline{u}$ muito afastado do centro $\underline{u}_k$ resulta em uma fraca resposta da k –ésima função de base radial caso o raio σ delimite uma região de sensibilidade muito estreita. No caso da função Gaussiana o parâmetro σ corresponde ao desvio-padrão.

Arquitetura de uma RNA Radial Basis Function



Note a camada de M nós-fonte (que conectam a rede a seu ambiente externo), à qual é apresentado o vetor de entrada $\underline{u}(n) \in \Re^M$.

Note também a única camada *hidden* de K neurônios não-lineares (usualmente gaussianos), cada um deles computando uma função da distância entre o vetor de entrada e o centro da função de base radial associada.

O mapeamento não-linear do k-ésimo neurônio é expresso por uma função de ativação Gaussiana, da forma

$$\varphi_k(n) = \varphi_k\left(\underline{u}(n), \underline{t}_k(n), \sigma_k^2(n)\right) =$$

$$= \exp\left[-\frac{1}{\sigma_k^2(n)}\left\|\underline{u}(n) - \underline{t}_k(n)\right\|^2\right],$$

onde:

$\underline{u}(n) \in \Re^M$ representa o vetor de entrada $\underline{u}$ no instante n,
M é o número de nós na camada de entrada = dimensão dos vetores de entrada;

$\underline{t}_k(n) \in \Re^M$ representa o vetor-centro da k-ésima função de base radial no instante n,
$k = 0,1,\cdots,K-1$, onde K é o número de funções de base radial;

$\sigma_k^2(n) \in \Re$ é a variância (quadrado do desvio padrão σ) da k-ésima função φ_k no instante n.

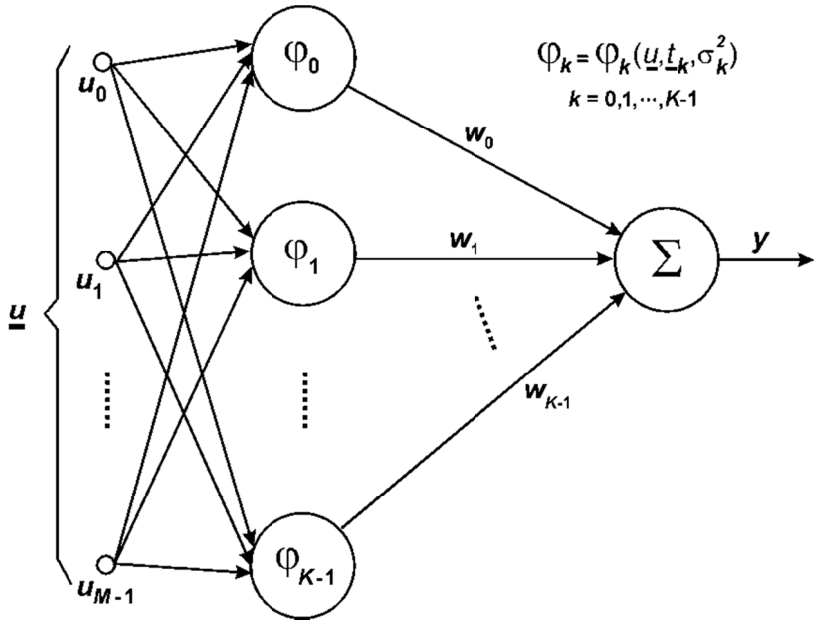
A camada de saída da rede neural é usualmente formada por um único neurônio linear, definido como um combinador linear das saídas das funções de base radial.

A saída *y* da rede RBF é, portanto, a soma das saídas de cada Gaussiana, ponderadas pelos respectivos pesos sinápticos w_k, de tal forma que a combinação linear é expressa por

$$y = \sum_{k=0}^{K-1} w_k \varphi_k\left(\underline{u}, \underline{t}_k, \sigma_k^2\right)$$

O termo $\varphi_k\left(\underline{u}, \underline{t}_k, \sigma_k^2\right)$ é a k –ésima função de base radial, que computa o quadrado da distância Euclidiana $D_k^{\,2} = \left\| \underline{u} - \underline{t}_k \right\|^2$ entre $\underline{u}$ e o centro $\underline{t}_k$ da k –ésima função de base radial.

O sinal de saída produzido pelo *k*-ésimo neurônio escondido é, portanto, devido à função $\exp(\cdot)$ e ao operador $(\cdot)^2$, sendo portanto uma função não-linear da distância D_k .



A transmitância w_k da k –ésima sinapse conecta o sinal de saída do *k*-ésimo neurônio escondido ao neurônio linear na saída da rede.

À equação para *y* pode, em alguns casos, ser ainda acrescido um termo constante de polarização ou *bias*.

Transformação não-linear: é definida pelo conjunto de funções de base radial φ_k .

Transformação linear: é definida pelo conjunto de sinapses w_k, $k = 0, 1, \cdots, K-1$.

O procedimento de aprendizado de uma RNA RBF ajusta os parâmetros livres da RBF, que são:

as variâncias σ_k^2 , os centros $\underline{t}_k$ e os pesos sinápticos w_k .

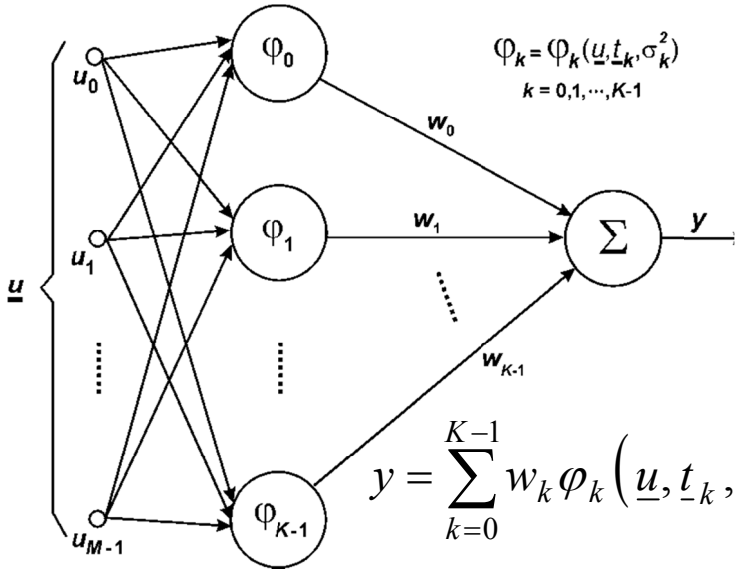
O aprendizado ou treinamento consiste em determinar estes parâmetros de tal forma que, dado um conjunto de estímulos $\underline{u}$ na entrada, as saídas *y* se aproximem o mais possível do conjunto de valores desejado.

Diferentes algoritmos podem ser utilizados para o ajuste dos parâmetros livres das redes RBF. Por exemplo:

- o algoritmo *k-means* pode ser utilizado para a inicialização e/ou atualização dos centros das funções de base radial;
- o algoritmo de Moore-Penrose para pseudo-inversão de matrizes pode ser utilizado para a determinação dos pesos sinápticos;
- o algoritmo Gradiente Estocástico pode ser aplicado na atualização dos pesos da rede RBF, das variâncias e dos centros das funções de base radial.

Possíveis Algoritmos de Aprendizado para Ajuste dos Parâmetros Livres

Centros das RBF	Pesos Sinápticos	Variâncias dos centros
Constante: Por conhecimento prévio e inferência a partir do conjunto de vetores de treino.	Grad. Estocástico (LMS) Supervisionado: usa $e(n) = d(n) - y(n)$	Constante: Por conhecimento prévio e inferência a partir do conjunto de vetores de treino.
"Clusterização" pelo algoritmo *k-means*. Não-supervisionado.	Pseudo Inversa por decomposição em valores singulares (Moore-Penrose): $\underline{w}(n) = \mathbf{\Phi}^{-1}(n) . \underline{d}(n)$ Vide função pinv() do Matlab em http://www.ece.northwestern.edu/local-apps/matlabhelp/techdoc/ref/pinv.html	Gradiente Estocástico (LMS). Supervisionado: usa $e(n) = d(n) - y(n)$
Gradiente Estocástico (LMS). Supervisionado: usa $e(n) = d(n) - y(n)$		$\sigma_k^2(n) = \xi_k(n) . \max_{a,b} \left\{ \left\| \underline{t}_a(n) - \underline{t}_b(n) \right\|^2 \right\}$ onde $\xi_k(n)$ é fixo ou ajustado pelo LMS.



$$y = \sum_{k=0}^{K-1} w_k \varphi_k\left(\underline{u}, \underline{t}_k, \sigma_k^2\right)$$

$$\begin{bmatrix} \varphi_0(n) & \cdots & \varphi_{K-1}(n) \\ \vdots & \ddots & \vdots \\ \varphi_0(n-N+1) & \cdots & \varphi_{K-1}(n-N+1) \end{bmatrix} \begin{bmatrix} w_0(n) \\ \vdots \\ w_{K-1}(n) \end{bmatrix} = \begin{bmatrix} d(n) \\ \vdots \\ d(n-N+1) \end{bmatrix}$$

$\mathbf{\Phi}(n)$ $\quad$ $\underline{w}(n)$ $\quad$ $\underline{d}(n)$

onde N é o número de vetores de entrada $\underline{u}(n)$ do conjunto de treino, sendo K não necessariamente igual a N.

Diferentes heurísticas de treinamento resultam da combinação dos algoritmos para atualização de centros, variâncias e pesos sinápticos mostrados na Tabela anterior.

A complexidade computacional do processo de aprendizado de uma rede RBF é, em geral, menor que a complexidade computacional do processo de aprendizado de uma rede MLP, principalmente porque os processos de aprendizagem para os centros, as variâncias e os pesos sinápticos podem ser encadeados sequencialmente, possibilitando que o aprendizado das redes RBF seja otimizado pela resultante divisão de tarefas.

Neste contexto, uma possível heurística de aprendizado é:

- um algoritmo não-supervisionado estima inicialmente os centros (algoritmo *K*-means)
- uma posterior estimativa inicial da distância do vetor de entrada com respeito a cada centro inicializa as variâncias dos centros
- os pesos sinápticos são determinados pelo algoritmo de pseudo-inversão de Moore-Penrose .

Após a estimativa inicial de parâmetros da rede, um ajuste mais fino e dinâmico é aplicado a esta estimativa, utilizando técnicas de gradiente aplicadas ao ajuste adaptativo de todos os parâmetros livres (centros, variâncias e sinapses).

Sistemas dinâmicos, com operação *on-the-fly* para aproximação de funções e de processos, usualmente adotam como algoritmo de aprendizado supervisionado a técnica de gradiente *steepest descent* (regra delta) – LMS , que é aplicada ao ajuste adaptativo de todos os parâmetros livres (centros, variâncias e sinapses). O objetivo é minimizar a função de custo dada pelo valor esperado do erro quadrático entre a saída da RBF e a saída desejada para o processo, ou seja, $J = E\left\{\left(d(n) - y(n)\right)^2\right\}$.

Heurística sugerida p/ o processo de aprendizado de RNA Radial Basis Function

Inicialização

1. Centros das funções de base radial

   Nesta abordagem, os centros das funções de base radial são primeiramente inicializados pelo algoritmo *k-means*.

Algoritmo *k-means*

- A cada iteração n, o algoritmo *k-means* determina as distâncias entre o vetor $\underline{u}(n)$ pertencente ao conjunto de entrada e cada um dos centros $\underline{t}_k(n)$.
- Ao centro $\tilde{k}$ que corresponder à menor distância – Equação (5.6) – é aplicada a atualização mostrada na Equação (5.7).

$$\tilde{k}(\underline{u}) = \arg\min_k \| \underline{u}(n) - \underline{t}_k(n) \|; \quad k = 0,1,\cdots,K-1 \tag{5.6}$$

$$\underline{t}(n+1) = \begin{cases} \underline{t}_k(n) + \eta[\underline{u}(n) - \underline{t}_k(n)]; & k = \tilde{k}(\underline{u}) \\ \underline{t}_k(n); & \text{outros casos} \end{cases} \tag{5.7}$$

onde η é a razão de atualização do algoritmo *k-means*.

➢ As iterações prosseguem até que $\| \underline{t}_k(n) - \underline{t}_k(n+1) \| \to \varepsilon, \ \forall k$, onde ε é um número muito pequeno.

2. Variância das funções de base radial

Após a inicialização dos centros pelo algoritmo *k-means*, é definida a variância inicial comum a todos os centros pela Equação (5.8).

$$\sigma^2 = d_{\max}^2 \left(\underline{t}_i - \underline{t}_j \right); \quad \text{com} \quad i, j = 0,1,\cdots, K-1 \tag{5.8}$$

Na Equação (5.8), $d_{\max}^2 \left(\underline{t}_i - \underline{t}_j \right)$ expressa o quadrado da maior distância Euclidiana entre os centros , isto é, $\max\left\{ \left\| \underline{t}_i - \underline{t}_j \right\|^2 \right\}$.

3. Pesos Sinápticos

Os pesos sinápticos são inicializados com zero.

Treinamento

Para cada vetor $\underline{u}(n)$ do conjunto de treino apresentado à entrada da rede, a saída $y(n)$ da rede RBF é determinada, e é avaliada a diferença entre este valor de saída e aquele desejado $d(n)$, conforme

$$e(n) = d(n) - y(n) \tag{5.9}$$

O erro assim obtido é utilizado para a posterior atualização até a convergência dos centros, pesos sinápticos e variância.

As equações de atualização, baseadas no algoritmo *steepest descent* (regra delta) , são expressas em (5.10), (5.11) e (5.12), e encontram-se derivadas no Apêndice A.

$$w_k(n+1) = w_k(n) + \mu_w e(n)\varphi_k(n) \tag{5.10}$$

$$\underline{t}_k(n+1) = \underline{t}_k(n) + 2\mu_t e(n) w_k(n)\varphi_k(n)\frac{\underline{u}(n) - \underline{t}_k(n)}{\sigma_k^2(n)} \tag{5.11}$$

$$\sigma_k^2(n+1) = \sigma_k^2(n) + \mu_\sigma e(n) w_k(n)\varphi_k(n)\frac{\left\|\underline{u}(n) - \underline{t}_k(n)\right\|^2}{\left(\sigma_k^2(n)\right)^2} \tag{5.12}$$

Observe que, nas equações (5.10), (5.11) e (5.12) os parâmetros μ_w, μ_t e μ_σ são, respectivamente, as razões de aprendizado da adaptação dos pesos sinápticos, dos centros das funções de base radial e das variâncias dos centros.

A apresentação de *N* vetores de dados $\underline{u}(n)$ à entrada da RBF, $n = 0,1,\cdots,N-1$, constitui uma época de treino.

Ao final de cada época, o conjunto de vetores de dados é embaralhado aleatoriamente com uniforme distribuição de probabilidades, para evitar que a rede aprenda o padrão sequencial de apresentação dos vetores de treino. Isto porque estamos interessados na capacidade de generalização da rede com relação ao conjunto de dados em si, e a mesma ordem de apresentação dos vetores a cada época de treino poderia prejudicar tal capacidade de generalização.

O treinamento de uma RBF através das Equações (5.10) a (5.12) é continuado até a sua convergência, situação em que o valor obtido para o erro de aproximação é menor que um valor máximo permitido ε .

Critério de avaliação do erro de aproximação

Para avaliar a capacidade de aproximação das redes RBF é definido o MSEA, ou seja, o Erro Médio Quadrático de Aproximação dado por

$$\text{MSEA} = \frac{1}{N}\sum_{n=0}^{N-1}\left(d(n) - y(n)\right)^2 \qquad (5.13)$$

Sumário da heurística de treino de uma RNA RBF no contexto de aproximação de funções e processos:

Gradiente (*steepest descent*): w_k , $\underline{t}_k$ e σ_k^2 são ajustados até a convergência em um processo de aprendizado supervisionado baseado na minimização de $J = E\left\{(d(n) - y(n))^2\right\}$ através da regra delta.

I - Inicialização:

1. Subtrair o vetor média do conjunto de N vetores de treino.

2. Normalizar a i-ésima componente de cada vetor de treino pelo desvio padrão do conjunto de N valores formado pela i-ésima componente de todos os N vetores de treino.

3. Normalizar o conjunto de N saídas desejadas para o intervalo [-1,+1].

4. Inicializar os centros das funções de base radial:

$$\underline{t}_k(n=0) \rightarrow \textit{k-means}$$

$$\underline{t}(n+1) = \begin{cases} \underline{t}_k(n) + \eta\left[\underline{u}(n) - \underline{t}_k(n)\right]; & k = \tilde{k}(\underline{u}) \\ \underline{t}_k(n); & \text{outros casos} \end{cases}$$

$$\tilde{k}(\underline{u}) = \arg\min_k \left\| \underline{u}(n) - \underline{t}_k(n) \right\|; \quad k = 0,1,\cdots,K-1$$

$$\left\| \underline{t}_k(n) - \underline{t}_k(n+1) \right\| \rightarrow \varepsilon, \ \forall k$$

5. Inicializar os pesos sinápticos: $\underline{w}_k(n=0) \rightarrow 0$; $k = 0,1,\cdots,K-1$

6. Inicializar as variâncias dos centros:

$$\sigma_k^2(n=0) \rightarrow d_{\max}^2\left(\underline{t}_i - \underline{t}_j\right), \forall k; \quad i,j = 0,1,\cdots,K-1, \ \text{onde} \ d_{\max}^2\left(\underline{t}_i - \underline{t}_j\right) = \max\left\{\left\|\underline{t}_i - \underline{t}_j\right\|^2\right\}$$

II - Treinamento:

1. Apresentar vetor $\underline{u}(n) \in$ ao conjunto de treino à entrada da RBF.

2. Calcular saída da rede através de

$$y(n)=\sum_{k=0}^{K-1} w_k(n)\varphi_k(n)=\sum_{k=0}^{K-1} w_k(n)\exp\left[-\frac{1}{\sigma_k^2(n)}\left\|\underline{u}(n)-\underline{t}_k(n)\right\|^2\right]$$

$\Rightarrow$ Se há *bias* B, $y(n)=\sum_{k=0}^{K-1} w_k(n)\varphi_k(n)+w_b B$

3. Determinar o erro instantâneo de treinamento na iteração *n* através de $e(n)=d(n)-y(n)$

4. Atualizar os pesos sinápticos w_k :

$$w_k(n+1)=w_k(n)+\mu_w e(n)\varphi_k(n)$$

Atualizar os centros das funções de base radial $\underline{t}_k$:

$$\underline{t}_k(n+1)=\underline{t}_k(n)+2\mu_t e(n)w_k(n)\varphi_k(n)\frac{\underline{u}(n)-\underline{t}_k(n)}{\sigma_k^2(n)}$$

Atualizar as variâncias dos centros das funções de base radial $\sigma_k^{\,2}$:

$$\sigma_k^2(n+1)=\sigma_k^2(n)+\mu_\sigma e(n)w_k(n)\varphi_k(n)\frac{\left\|\underline{u}(n)-\underline{t}_k(n)\right\|^2}{\left(\sigma_k^2(n)\right)^2}$$

5. Incrementar n $(n=n+1)$. Se n é tal que todos os N vetores do conjunto de treino foram apresentados à entrada da RBF (uma época), executar o procedimento II.6, caso contrário executar o procedimento II.1.

6. Embaralhar aleatoriamente a ordem dos vetores $\in$ ao conjunto de treino.

7. Determinar e avaliar o Erro Médio Quadrático de Aproximação dado por $\mathrm{MSEA}=\frac{1}{N}\sum_{n=0}^{N-1}(d(n)-y(n))^2$. Se $\mathrm{MSEA}<\varepsilon \Rightarrow$ II.8, caso contrário $\Rightarrow$ II.1.

8. Armazenar os parâmetros da rede (w_k , $\underline{t}_k$ e σ_k^2).

Referências:

[1] Figura obtida em 17/05/2017 no site:
https://www.mathworks.com/matlabcentral/fileexchange/52580-radial-basis-function-neural-networks--with-parameter-selection-using-k-means-?requestedDomain=www.mathworks.com

[2] Figura obtida em 17/05/2017 no site:
http://www.it.uu.se/research/scientific_computing/project/rbf

[3] Foto *Star Trails over Mexican Hat Rock, Utah*, obtida em 17/05/2017 no site: *https://www.flickr.com/photos/ironrodart/33679728172/*

RNAs *RBF* no Contexto de Aproximação de Funções: Exemplo

$\underline{t}_k$ inicializados pelo *k-means*	Atualização até a convergência dos $\underline{t}_k$, σ_k^2 e w_k $\Rightarrow$Gradiente Estocástico.

➢ Estabelecer a relação analítica entre os oito primeiro algarismos representados em base binária e os valores que definem o quadrado de um décimo de sua representação em base decimal.

Mapeamento $F:\Re^3 \to \Re$ que se deseja aproximar.

$\underline{u} \in \Re^3$			$F(\underline{u})$
u_0	u_1	u_2	
0	0	0	0.0
0	0	1	0.01
0	1	0	0.04
0	1	1	0.09
1	0	0	0.16
1	0	1	0.25
1	1	0	0.36
1	1	1	0.49

Aproximação por mapeamento linear

$$F(\underline{u}) \overset{\underline{w}?}{=} \underline{w}^T \underline{u} = w_0 u_0 + w_1 u_1 + w_2 u_2$$

sendo $\underline{w} = [w_0 \;\; w_1 \;\; w_2]^T$ o vetor que define $F(\underline{u})$ obtido da solução do sistema de equações

$$\begin{aligned}
0w_0 + 0w_1 + 0w_2 &= 0.0 \\
0w_0 + 0w_1 + 1w_2 &= 0.01 \\
0w_0 + 1w_1 + 0w_2 &= 0.04 \\
0w_0 + 1w_1 + 1w_2 &= 0.09 \\
1w_0 + 0w_1 + 0w_2 &= 0.16 \\
1w_0 + 0w_1 + 1w_2 &= 0.25 \\
1w_0 + 1w_1 + 0w_2 &= 0.36 \\
1w_0 + 1w_1 + 1w_2 &= 0.49
\end{aligned}$$

sem solução – não existe $\underline{w} = [w_0 \;\; w_1 \;\; w_2]^T$ que atenda simultaneamente todas as Equações.

Aproximação por mapeamento não-linear (expresso pela rede RBF)

Conjunto de treino da RBF (Tabela):

$N = 8$ vetores $\underline{u}(n) \in \Re^3 \quad n = 0,1,\cdots,N-1 \quad d(n) = F(\underline{u}(n))$

RNA RBF:

$\underline{u}(n) \in \Re^3$, $M = 3$ (nós entrada).

$K = 3$ (funções de base radial para formar a superfície de aproximação).

Inicialização:

$\underline{t}_k \to$ *k-means* ($\eta = 0.1$)

$\sigma_k^2 \to \sigma_k^2 = d_{\max}^2 (\underline{t}_i - \underline{t}_j); \quad i, j = 0,1,\cdots,K-1$

Treinamento:

$$w_k(n+1) = w_k(n) + \mu_w e(n) \varphi_k(n)$$

$$\underline{t}_k(n+1) = \underline{t}_k(n) + 2\mu_t e(n) w_k(n) \varphi_k(n) \frac{\underline{u}(n) - \underline{t}_k(n)}{\sigma_k^2(n)}$$

$$\sigma_k^2(n+1) = \sigma_k^2(n) + \mu_\sigma e(n) w_k(n) \varphi_k(n) \frac{\|\underline{u}(n) - \underline{t}_k(n)\|^2}{(\sigma_k^2(n))^2}$$

($\mu_w = 0.1$, $\mu_t = 0.1$ e $\mu_\sigma = 0.1$)

Época: apresentação dos $N = 8$ vets. de treino

Para o treinamento da RBF, o conjunto de treino é normalizado para $[-1,1]$ (precaução para evitar *overflow* das variáveis de ponto flutuante ao longo da operação do Gradiente Estocástico).

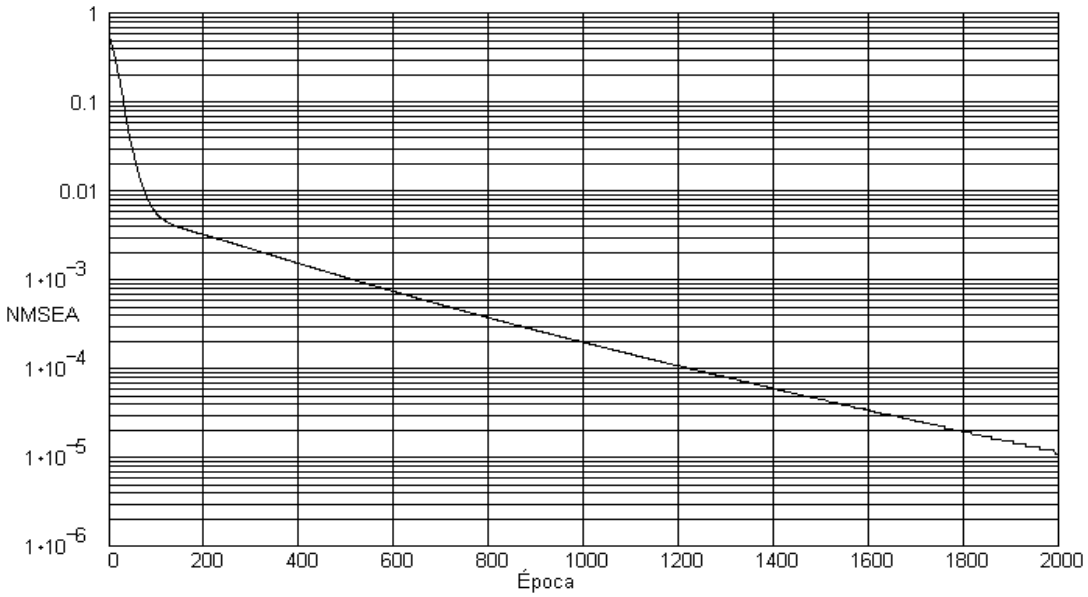
- Cada componente dos vetores $\underline{u}(n) \in \Re^3$ é norm. através da transf. $\theta_u : \Re \to \Re$, $\theta_u(x) = 2x - 1$.
- Cada valor do $\{d(n)\}$, $n = 0,1,\cdots, N-1$ é norm. através da transf. $\theta_d : \Re \to \Re$, $\theta_d(x) = 4.08163x - 1$.

- Após 2000 épocas de treino, a rede RBF apresenta a relação analítica dada pela equação abaixo, como aproximação para o mapeamento $F(\underline{u})$ (a Eq. inclui as desnormalizações θ_u e θ_d).

$$F(\underline{u}) = \frac{-1.948417 \cdot e^{\frac{-\left\| (2\underline{u}-\underline{1}) - \begin{bmatrix} -0.069259 \\ -0.962758 \\ 0.015931 \end{bmatrix} \right\|^2}{2.774243}}}{4.08163} + \frac{2.111299 \cdot e^{\frac{-\left\| (2\underline{u}-\underline{1}) - \begin{bmatrix} 1.265731 \\ 0.180856 \\ 0.316373 \end{bmatrix} \right\|^2}{2.372715}}}{4.08163} + \frac{-1.045209 \cdot e^{\frac{-\left\| (2\underline{u}-\underline{1}) - \begin{bmatrix} -1.234718 \\ 1.019335 \\ -0.118491 \end{bmatrix} \right\|^2}{1.934487}}}{4.08163} + 1$$

Representação obtida pela rede RBF para o mapeamento $F(\underline{u})$.

$$F\left([0\,0\,0]^T\right) = -7.739 \times 10^{-6}$$
$$F\left([0\,0\,1]^T\right) = 0.01$$
$$F\left([0\,1\,0]^T\right) = 0.04$$
$$F\left([0\,1\,1]^T\right) = 0.09$$
$$F\left([1\,0\,0]^T\right) = 0.16$$
$$F\left([1\,0\,1]^T\right) = 0.25$$
$$F\left([1\,1\,0]^T\right) = 0.36$$
$$F\left([1\,1\,1]^T\right) = 0.49$$



Evolução do NMSEA à medida que as épocas de treinamento se sucedem.

Apêndice A

Derivação das equações de atualização para w_k, $\underline{t}_k$ e σ_k^2 a partir da Regra Delta

Derivação de $\Delta w_k(n)$

Sabemos (do Capítulo 3) que a equação de atualização para os pesos sinápticos a partir do algoritmo Gradiente Estocástico pode ser expressa por

$$w_p(n+1) = w_p(n) - \mu_w \nabla_p \mathrm{J}(n); \quad p = 0,1,\cdots,K-1, \tag{5.14}$$

onde o parâmetro μ_w é a razão de aprendizado (ou passo de adaptação) dos pesos sinápticos e a função de custo J é expressa por

$$\mathrm{J} = \frac{1}{2}e^2 = \frac{1}{2}(d-y)^2, \tag{5.15}$$

em que foi considerada a expressão para o erro dada por (5.9).

Observa-se a partir da Equação (5.14) que, para que possamos determinar $\Delta w_p(n)$ precisamos encontrar $\nabla_p \mathrm{J}$. Desta forma, de (5.15) podemos escrever

$$\nabla_p \mathrm{J} = \frac{\partial \mathrm{J}}{\partial w_p} = \frac{1}{2}\frac{\partial}{\partial w_p}(d-y)^2 \tag{5.16}$$

em que

$$\frac{\partial}{\partial w_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial w_p}(d-y). \tag{5.17}$$

Na Equação (5.5) encontramos a expressão para a saída y da rede RBF que, substituída em (5.17) conduz a

$$\frac{\partial}{\partial w_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial w_p}\left(d - \sum_{i=0}^{K-1} w_i \varphi_i\right) \tag{5.18}$$

que pode ser expandida em

$$\frac{\partial}{\partial w_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial w_p}\left(d - w_0\varphi_0 - w_1\varphi_1 - \ldots - w_p\varphi_p - \ldots - w_{K-1}\varphi_{K-1}\right) \tag{5.19}$$

Como a saída desejada d não depende dos pesos sinápticos, e a derivada $\partial/\partial w_p$ dos termos expandidos só existirá para $w_i = w_p$, teremos

$$\frac{\partial}{\partial w_p}(d-y)^2 = 2(d-y)\left(-\varphi_p\right) \tag{5.20}$$

Substituindo (5.20) em (5.16) encontraremos

$$\nabla_p \mathrm{J} = \frac{\partial \mathrm{J}}{\partial w_p} = \frac{1}{2}\frac{\partial}{\partial w_p}(d-y)^2 = \frac{1}{2}\left[2(d-y)\left(-\varphi_p\right)\right] = (d-y)\left(-\varphi_p\right) \tag{5.21}$$

Desta forma, substituindo (5.21) em (5.14) obteremos a equação de atualização para os pesos sinápticos, que pode ser expressa por

$$w_p(n+1) = w_p(n) - \mu_w(d-y)\left(-\varphi_p\right) = w_p(n) + \mu_w\, e(n)\varphi_p(n) \tag{5.22}$$

Observe que a Equação (5.22) é igual à Equação (5.10).

Derivação de $\Delta \underline{t}_k(n)$

Passemos agora à derivação da equação de atualização para os vetores centro das funções de base radial, a partir do algoritmo Gradiente Estocástico. Sabemos, do Capítulo 3, que

$$\underline{t}_p(n+1) = \underline{t}_p(n) - \mu_t \underline{\nabla}_p \, \mathrm{J}(n); \quad p = 0,1,\cdots,K-1 \tag{5.23}$$

onde o parâmetro μ_t é a razão de aprendizado (ou passo de adaptação) dos vetores centro das funções de base radial e a função de custo J é, novamente, expressa pela Equação (5.15).

Observa-se, a partir da Equação (5.23) que, para que possamos determinar $\Delta \underline{t}_p(n)$ precisamos encontrar $\underline{\nabla}_p \, \mathrm{J}$. Desta forma, de (5.15) podemos escrever

$$\underline{\nabla}_p \mathrm{J} = \frac{\partial \mathrm{J}}{\partial \underline{t}_p} = \frac{1}{2} \frac{\partial}{\partial \underline{t}_p} (d-y)^2 \tag{5.24}$$

onde

$$\frac{\partial}{\partial \underline{t}_p} (d-y)^2 = 2(d-y) \frac{\partial}{\partial \underline{t}_p} \left[d - \sum_{i=0}^{K-1} w_i \varphi_i \right] \tag{5.25}$$

Na Equação (5.4) encontramos a expressão para as funções de base radial φ_i. A partir de (5.4) podemos escrever

$$\frac{\partial}{\partial \underline{t}_p} (d-y)^2 = 2(d-y) \frac{\partial}{\partial \underline{t}_p} \left[d - \sum_{i=0}^{K-1} w_i \exp\left[-\frac{1}{\sigma_i^2} \left\| \underline{u} - \underline{t}_i \right\|^2 \right] \right] \tag{5.26}$$

Expandindo o somatório presente em (5.26), encontraremos

$$\frac{\partial}{\partial \underline{t}_p}(d-y)^2 = 2(d-y)\times \tag{5.27}$$

$$\frac{\partial}{\partial \underline{t}_p}\left[d - w_0 e^{-\frac{\|\underline{u}-\underline{t}_0\|^2}{\sigma_0^2}} - w_1 e^{-\frac{\|\underline{u}-\underline{t}_1\|^2}{\sigma_1^2}} - \ldots - w_p e^{-\frac{\|\underline{u}-\underline{t}_p\|^2}{\sigma_p^2}} - \ldots - w_{K-1} e^{-\frac{\|\underline{u}-\underline{t}_{K-1}\|^2}{\sigma_{K-1}^2}} \right]$$

Como a saída desejada d não depende dos centros das funções de base radial e a derivada $\partial/\partial \underline{t}_p$ dos termos expandidos só existirá para $\underline{t}_i = \underline{t}_p$, teremos

$$\frac{\partial}{\partial \underline{t}_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial \underline{t}_p}\left[-w_p \exp\left(-\frac{\|\underline{u}-\underline{t}_p\|^2}{\sigma_p^2} \right) \right] = \tag{5.28}$$

$$= 2(d-y)(-2)w_p\left(\frac{\underline{u}-\underline{t}_p}{\sigma_p^2} \right)\exp\left[-\frac{\|\underline{u}-\underline{t}_p\|^2}{\sigma_p^2} \right]$$

Substituindo (5.28) em (5.24),

$$\underline{\nabla}_p \mathrm{J} = \frac{1}{2}\left\{ 2(d-y)(-2)w_p\left(\frac{\underline{u}-\underline{t}_p}{\sigma_p^2} \right)\exp\left[-\frac{\|\underline{u}-\underline{t}_p\|^2}{\sigma_p^2} \right] \right\} = \tag{5.29}$$

$$= -2w_p(d-y)\left(\frac{\underline{u}-\underline{t}_p}{\sigma_p^2} \right)\exp\left[-\frac{\|\underline{u}-\underline{t}_p\|^2}{\sigma_p^2} \right]$$

Levando o resultado de $\underline{\nabla}_p \mathrm{J}$ obtido em (5.29) à Equação (5.23), teremos

$$\underline{t}_p(n+1) = \underline{t}_p(n) + 2\,\mu_t\, w_p(n) e(n)\left(\frac{\underline{u}(n)-\underline{t}_p(n)}{\sigma_p^2(n)} \right)\varphi_p(n) \tag{5.30}$$

Observe que a Equação (5.30) é igual à Equação (5.11).

Derivação de $\Delta\sigma_k^2(n)$

Tendo derivado as equações de atualização para os pesos sinápticos e para os vetores centro das funções de base radial a partir do algoritmo Gradiente Estocástico, resta-nos derivar a equação de atualização para as variâncias dos centros das funções de base radial.

Assim, fazendo $\alpha_p = \sigma_p^2$, partiremos de

$$\alpha_p(n+1) = \alpha_p(n) - \mu_\sigma \nabla_p \mathrm{J}(n); \quad p = 0,1,\cdots,K-1 \tag{5.31}$$

onde μ_σ é o parâmetro razão de aprendizado (ou passo de adaptação) das variâncias dos centros das funções de base radial. A função de custo J é expressa pela Equação (5.15).

A partir de (5.31) observa-se que, para a determinação de $\Delta\alpha_p(n)$, precisamos encontrar $\nabla_p \mathrm{J}$. Desta forma, a partir de (5.15) podemos escrever

$$\nabla_p \mathrm{J} = \frac{\partial \mathrm{J}}{\partial \alpha_p} = \frac{1}{2}\frac{\partial}{\partial \alpha_p}(d-y)^2 \tag{5.32}$$

onde

$$\frac{\partial}{\partial \alpha_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial \alpha_p}\left[d - \sum_{i=0}^{K-1} w_i \varphi_i\right] \tag{5.33}$$

Na Equação (5.4) encontramos a expressão para as funções de base radial φ_i. A partir de (5.4) podemos escrever

$$\frac{\partial}{\partial \alpha_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial \alpha_p}\left[d - \sum_{i=0}^{K-1} w_i \exp\left[-\frac{1}{\alpha_i}\left\| \underline{u} - \underline{t}_i \right\|^2 \right] \right] \tag{5.34}$$

Expandindo o somatório presente em (5.34), encontraremos

$$\frac{\partial}{\partial \alpha_p}(d-y)^2 = 2(d-y)\times \tag{5.35}$$

$$\frac{\partial}{\partial \alpha_p}\left[d - w_0 e^{-\frac{\left\| \underline{u}-\underline{t}_0 \right\|^2}{\alpha_0}} - w_1 e^{-\frac{\left\| \underline{u}-\underline{t}_1 \right\|^2}{\alpha_1}} - \ldots - w_p e^{-\frac{\left\| \underline{u}-\underline{t}_p \right\|^2}{\alpha_p}} - \ldots - w_{K-1} e^{-\frac{\left\| \underline{u}-\underline{t}_{K-1} \right\|^2}{\alpha_{K-1}}} \right]$$

Como a saída desejada d não depende das variâncias das funções de base radial e a derivada $\partial/\partial \alpha_p$ dos termos expandidos só existirá para $\alpha_i = \alpha_p$, teremos

$$\frac{\partial}{\partial \alpha_p}(d-y)^2 = 2(d-y)\frac{\partial}{\partial \alpha_p}\left[-w_p \exp\left(-\frac{\left\| \underline{u} - \underline{t}_p \right\|^2}{\alpha_p} \right) \right] \tag{5.36}$$

Como $\frac{\partial}{\partial x}\left(\exp\left[-\frac{c}{x} \right] \right) = \frac{c}{x^2}\exp\left[-\frac{c}{x} \right]$, (5.36) pode ser escrita sob a forma

$$\frac{\partial}{\partial \alpha_p}(d-y)^2 = 2(d-y)\left(-w_p\right)\left(\frac{\left\| \underline{u} - \underline{t}_p \right\|^2}{\left(\alpha_p\right)^2} \right)\exp\left[-\frac{\left\| \underline{u} - \underline{t}_p \right\|^2}{\alpha_p} \right] \tag{5.37}$$

Levando o resultado de (5.37) à Equação (5.32), teremos

$$\nabla_p \mathrm{J} = \frac{\partial \mathrm{J}}{\partial \alpha_p} = \frac{1}{2}\left\{ 2(d-y)(-w_p)\left(\frac{\left\| \underline{u} - \underline{t}_p \right\|^2}{(\alpha_p)^2} \right) \exp\left[-\frac{\left\| \underline{u} - \underline{t}_p \right\|^2}{\alpha_p} \right] \right\} = \tag{5.38}$$

$$= (d-y)(-w_p)\left(\frac{\left\| \underline{u} - \underline{t}_p \right\|^2}{(\alpha_p)^2} \right) \exp\left[-\frac{\left\| \underline{u} - \underline{t}_p \right\|^2}{\alpha_p} \right]$$

Substituindo o resultado obtido para $\nabla_p \mathrm{J}$ em (5.38) na Equação (5.31), teremos

$$\alpha_p(n+1) = \alpha_p(n) - \mu_\sigma (d-y)(-w_p)\left(\frac{\left\| \underline{u} - \underline{t}_p \right\|^2}{(\alpha_p)^2} \right) \exp\left[-\frac{\left\| \underline{u} - \underline{t}_p \right\|^2}{\alpha_p} \right] \tag{5.39}$$

Considerando que $e(n) = d(n) - y(n)$ e $\alpha_p = \sigma_p^2$, teremos

$$\sigma_p^2(n+1) = \sigma_p^2(n) + \mu_\sigma \, e(n) \, w_p(n) \varphi_p(n) \left(\frac{\left\| \underline{u}(n) - \underline{t}_p(n) \right\|^2}{(\sigma_p^2)^2} \right) \tag{5.40}$$

Observe que a Equação (5.40) é igual à Equação (5.12).