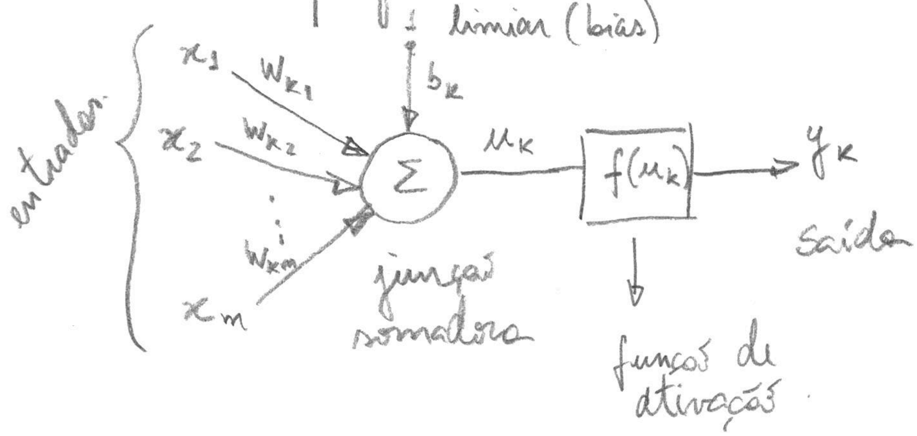
Redes Neurais Artificiais

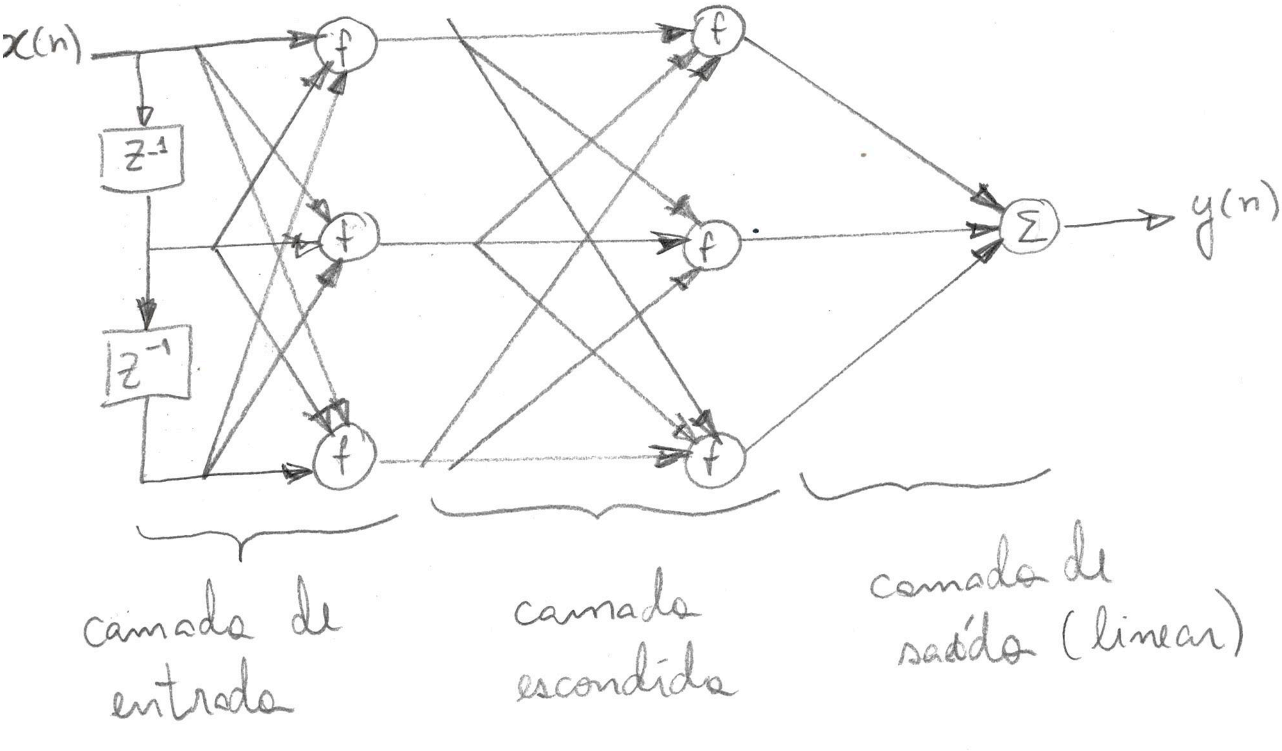
Modelo simplificado de um neurônio:



A saída do neurônio k é dada por

$$y_k = f(u_k) = f\left(\sum_{i=1}^{m} w_{k_i} x_i + b_k\right)$$

Uma rede neural é obtida a partir da interligação de vários neurônios, dando origem à estrutura conhecida como multilayer perceptron (MLP).



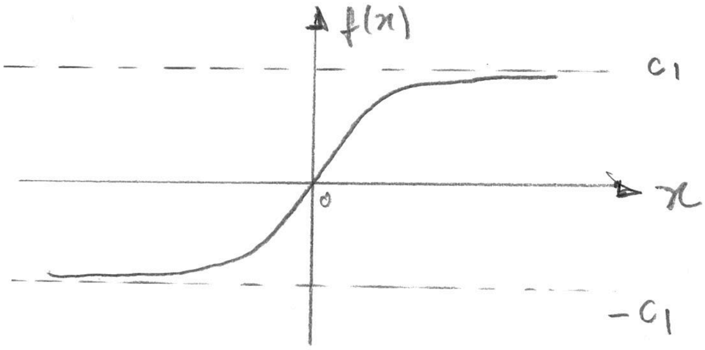
A saída de cada neurônio na MLP é dada pela expressão

$$y_{\ell,i}(n) = f_{\ell,i}\left\{\sum_{j=0}^{(N_{\ell-1}-1)} w_{\ell,i,j}(n)\, y_{\ell-1,j}(n) - b_{\ell,i}(n)\right\}$$

em que: $w_{\ell,i,j}(n)$ são os pesos que conectam o sinal de saída $y_{\ell-1,j}(n)$ do j-ésimo neurônio da camada $(\ell-1)$ à entrada do neurônio i da camada ℓ, para $\ell = 0, 1, \ldots, L-1$ e $i = 0, 1, \ldots, N_\ell - 1$. Note que N_ℓ é o número de neurônios na ℓ-ésima camada e que L é o número de camadas. (na figura anterior, $L = 3$).

A não-linearidade da MLP advém da função de ativação dos neurônios, que geralmente corresponde à tangente hiperbólica ou à função logística ou sigmoid. Usaremos a segunda!

$$f(x) = sgm(x) = \frac{2c_1}{1 + e^{-c_2 x}} - c_1$$



c_1 e c_2 são constantes arbitrárias.

Note que a somatória no argumento de $f(\cdot)$ é o produto interno entre o vetor de pesos sinápticos e o vetor de entradas do neurônio. Desse modo, quanto mais colineares forem esses vetores, maior a ativação do neurônio.

Os parâmetros de uma MLP são os pesos sinápticos e os limiares de cada neurônio. Eles podem ser ajustados de modo a minimizar o erro quadrático instantâneo, $e^2(n) = [d(n) - y(n)]^2$.

O backpropagation é um algoritmo muito usual no treinamento de MLP's e é baseado no gradiente do erro quadrático instantâneo.

Para os pesos da camada de saída, a equação de atualização é dada por:

$$w_{L-1,i,j}(n+1) = w_{L-1,i,j}(n) + 2\mu_{L-1} e(n)\, sgd\left\{sgm^{-1}\left[y_{L-1,j}(n)\right]\right\} sgm\left[y_{L-2,j}(n)\right]$$

em que $i = 0, 1, \ldots, N_{L-2}-1$ e $j = 0, 1, \ldots, N_{L-1}-1$. Como há apenas uma saída na rede, $N_{L-1} = 1$. O parâmetro μ_{L-1} é o passo de adaptação da última camada. A função $sgd(\cdot)$ corresponde à derivada da função sigmoide

$$sgd(x) = \frac{c_2}{2c_1}\left[c_1^2 - sgm^2(x)\right]$$

Aplicando a regra da cadeia para derivar a função objetivo em função do peso desejado, é possível retropropagar o erro até a saída dos neurônios internos:

$$e_{l,j}(n) = \text{sgd}\left[\sum_{j=0}^{N_{l-1}-1} w_{l,i,j}(n)\, y_{l-1,j}(n)\right] \sum_{i=0}^{N_{l}-1} w_{l+1,i,j}(n)\, e_{l+1,i}(n)$$

As equações de atualização dos pesos sinápticos das camadas internas e dos limiares são:

$$w_{l,i,j}(n+1) = w_{l,i,j}(n) + 2\mu_l\, e_{l,j}(n)\, y_{l-1,j}(n)$$

$$b_{l,i}(n+1) = b_{l,i}(n) - 2\mu_l\, e_{l,j}(n)$$

para $i = 0, 1, \ldots N_{l-1}-1$ e $j = 0, 1, \ldots, N_l - 1$

Assim, para cada novo vetor de entrada, primeiramente obtém-se a saída de cada neurônio da rede. Em seguida, calculam-se os erros e atualizam-se os pessos sinápticos e os limiares de cada neurônico, da ultima camada até a primeira.

Observações:

1) A MLP é uma estrutura de filtragem muito flexível e poderosa, com capacidade de aproximação universal, isto é, ela é capaz de aproximar uma função com precisão arbitrária em um domínio compacto.

2) Não é trivial determinar o número de neurônios e camadas necessários para se obter um desempenho adequado.

3) Por se tratar de um problema de otimização não-linear, existe a possibilidade de convergência para um mínimo local da função objetivo.

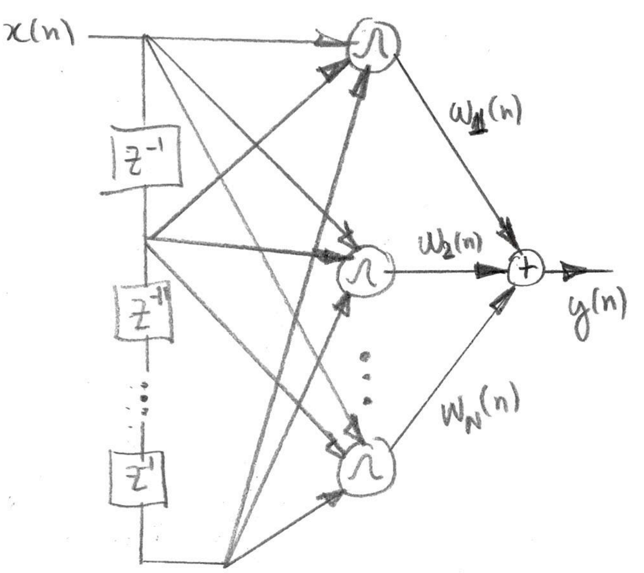
4) O algoritmo de adaptação é significativamente mais complexo que os vistos até o momento. Além disso, seu desempenho depende fortemente das inicializações.

Redes Neurais com Função de Ativação de Base Radial

Uma função de ativação de base radial (Radial Basis Function, RBF) é caracterizada por apresentar uma resposta que decresce (ou cresce) monotonicamente com a distância a um ponto central. Exemplo: função gaussiana

Diferenças em relação à rede MLP:

- As RBF's sempre apresentam uma única camada intermediária
- Neurônios de saída são sempre lineares.
- Os neurônios da camada intermediária têm função de base radial como função de ativação.



Saída do neurônio i:

$$f_i(\underline{x}(n)) = \exp\left[\frac{-\|\underline{x}(n) - \underline{c}_i\|^2}{\sigma_i^2}\right]$$

Assim, a saída da rede é dada por:

$$y(n) = \sum_{i=1}^{N} w_i \, f_i(\underline{x}(n)) = \underline{w}^T \underline{f}(n)$$

em que: $\underline{w} = [w_1 \; w_2 \; \dots \; w_N]^T$

$$\underline{f}(n) = [f_1(\underline{x}(n)) \;\; f_2(\underline{x}(n)) \dots f_N(\underline{x}(n))]^T$$

Os parâmetros da RBF são os centros das funções de ativação, $\underline{c}_i$, suas dispersões σ_i e os pesos da camada de saída. As dispersões determinam a região de influência da função de ativação.

Note que se os centros e dispersões forem fixos e apenas os pesos da camada de saída forem ajustáveis, o problema de otimização se torna linear nos parâmetros, como no caso dos filtros de Volterra.

Para otimização de todos os parâmetros do filtro usando o gradiente estocástico, temos as seguintes equações de atualização:

$$w_i(n+1) = w_i(n) + \mu_w\, e(n)\, f_i(\underline{x}(n))$$

$$\sigma_i(n+1) = \sigma_i(n) + \mu_\sigma\, e(n)\, w_i(n)\, \frac{\|\underline{x}(n) - \underline{c}_i(n)\|^2}{\sigma_i^3(n)} \cdot f_i(\underline{x}(n))$$

$$\underline{c}_i(n+1) = \underline{c}_i(n) + \mu_c\, e(n)\, f_i(\underline{x}(n))\, \underline{w}_i(n)\, \frac{\underline{x}(n) - \underline{c}_i(n)}{\sigma_i^2(n)}$$

Observações:

1) As redes RBF apresentam capacidade de aproximação universal

2) As funções de ativação RBF têm influência muito mais localizada que, por exemplo, a função sigmóide usada na MLP. Como conseqüência, a convergência para mínimos locais é um problema mais grave que no caso da MLP, de modo que não é aconselhado usar-se o algoritmo do gradiente estocástico a menos que se disponha de uma boa inicialização para os parâmetros, principalmente os centros.

3) Também devido à localidade da função de ativação, o número de neurônios necessário para o funcionamento adequado da RBF aumenta exponencialmente com o número de entradas.

Equalizador MAP ou Bayesiano

• O equalizador MAP é a estrutura de memória finita (feedforward) que minimiza a probabilidade de erro de símbolo na transmissão.

Objetivo: $\hat{s}(n-d) = \arg\max_{s} p(s(n-d)=s \mid \underline{x}(n))$

$$= \arg\max_{s} p(\underline{x}(n) \mid s(n-d)=s) \cdot P(s(n-d)=s)$$

Considerando uma modulação binária equiprovável, com $s(n) \in \{-1; +1\}$, $P(s(n-d)=s) = 0,5$ para qualquer valor de s. Resta obter $p(\underline{x}(n) \mid s(n-d)=s)$. Usando a lei da probabilidade total, temos:

$$p(\underline{x}(n) \mid s(n-d)=+1) = \sum_{\underline{s}_j \in S_d^+} p(\underline{x}(n) \mid \underline{s}_j)\, p(\underline{s}_j) \qquad (1)$$

em que $\underline{s}_j$ é a sequência transmitida que produziu o vetor de saída do canal $\underline{x}(n)$ e S_d^+ é o conjunto de todas as sequências transmitidas para as quais $s(n-d)=+1$. Como todas as sequências são equiprováveis, $p(\underline{s}_j) = \frac{1}{N_s}$, em que N_s é o número total de sequências distintas que podem ser transmitidas.

O vetor de saída do canal pode ser escrito como:

$$\underline{x}(n) = H\,\underline{s}(n) + \underline{\eta}(n)$$

em que:

$$\underline{x}(n) = [\,x(n) \quad x(n-1) \quad \dots \quad x(n-M+1)\,]^T$$

$$H = \begin{bmatrix} h_0 & h_1 & \dots & h_{N-1} & 0 & \dots & 0 \\ 0 & h_0 & h_1 & \dots & & & 0 \\ \vdots & & & & & & \vdots \\ 0 & \dots & & & \dots & h_{N-2} & h_{N-1} \end{bmatrix} \qquad M \times (N+M-1)$$

matriz de convolução.

$$\underline{s}(n) = [\,s(n) \quad s(n-1) \quad \dots \quad s(n-(N+M-2))\,]^T$$

$$\underline{\eta}(n) = [\,\eta(n) \quad \eta(n-1) \dots \quad \eta(n-(N+M-1))\,]^T \rightarrow \text{vetor de ruído}$$

Note que na ausência de ruído o vetor de saída do canal assume um número finito de valores, de acordo com a sequência transmitida.

$$\underline{x}(n) = H\,\underline{s}(n) = \underline{c}(n) \,.$$

Esses vetores são chamados de estados do canal.

$$\underline{c}_j = H\,\underline{s}_j$$

Desse modo, podemos escrever

$$p(\underline{x}(n)|\underline{s}_j) = p(\underline{x}(n)|\underline{c}_j)$$

Como $\underline{\eta}(n) = \underline{x}(n) - H\underline{s}(n) = \underline{x}(n) - \underline{c}_j$, para $\underline{s}(n) = \underline{s}_j$, e

$$p_\eta(\underline{\eta}(n)) = (2\pi\sigma_\eta^2)^{-M/2} \exp\left(\frac{-\|\underline{\eta}(n)\|^2}{2\sigma_\eta^2}\right)$$

Temos:

$$p(\underline{x}(n)|\underline{c}_j) = p_\eta(\underline{x}(n) - \underline{c}_j) = (2\pi\sigma_\eta^2)^{-M/2} \exp\left(-\frac{\|\underline{x}(n) - \underline{c}_j\|^2}{2\sigma_\eta^2}\right) \quad (2)$$

Substituindo esse resultado em (1),

$$p(\underline{x}(n)\,|\,s(n-d) = +1) = \frac{1}{N_s} \sum_{\underline{c}_j \in C_d^+} (2\pi\sigma_\eta^2)^{-M/2} \exp\left(\frac{-\|\underline{x}(n) - \underline{c}_j\|^2}{2\sigma_\eta^2}\right) \quad (3)$$

em que C_d^+ é o conjunto de todos os estados do canal para os quais $s(n-d) = +1$.

Equivalentemente,

$$p(\underline{x}(n)\,|\,s(n-d) = -1) = \frac{1}{N_s} \sum_{\underline{c}_j \in C_d^-} (2\pi\sigma_\eta^2)^{-M/2} \exp\left(\frac{-\|\underline{x}(n) - \underline{c}_j\|^2}{2\sigma_\eta^2}\right)$$

Definindo

$$y(n) = f_{MAP}(\underline{x}(n)) = p(\underline{x}(n)|s(n-d) = +1) - p(\underline{x}(n)|s(n-d) = -1)) \quad (4)$$

$f_{MAP}(\underline{x}(n)) > 0$ quando $p(\underline{x}(n)|s(n-d)=+1) > p(\underline{x}(n)|s(n-d)=-1)$

e

$f_{MAP}(\underline{x}(n)) < 0$ quando $p(\underline{x}(n)|s(n-d)=+1) < p(\underline{x}(n)|s(n-d)=-1)$

Ou seja, a saída do equalizador é positiva quando é mais provável que um símbolo +1 tenha sido transmitido e negativa quando é mais provável que um símbolo -1 tenha sido enviado. Assim, aplicando uma função sinal de decisão, $\hat{s}(n-d) = \mathrm{sgn}(y(n))$, será sempre estimado o valor mais provável para o símbolo transmitido.

Substituindo (2) e (3) em (4), temos.

$$y(n) = \underbrace{\frac{1}{N_s}\left(2\pi\sigma_\eta^2\right)^{-M/2}}_{\text{constante que não altera o sinal de } y(n)} \cdot \left[\sum_{\underline{c}_j \in C_d^+} \exp\left(\frac{-\|\underline{x}(n)-\underline{c}_j\|^2}{2\sigma_\eta^2}\right) - \sum_{\underline{c}_j \in C_d^-} \exp\left(\frac{-\|\underline{x}(n)-\underline{c}_j\|^2}{2\sigma_\eta^2}\right)\right]$$

Juntando as duas somatórias e suprimindo a constante multiplicativa,

$$y(n) = \sum_{j=1}^{N_s} w_j \exp\left(\frac{-\|\underline{x}(n)-\underline{c}_j\|^2}{2\sigma_\eta^2}\right)$$

em que $w_j = +1$ se $\underline{c}_j \in C_d^+$ e $w_j = -1$ se $\underline{c}_j \in C_d^-$.

Note como o mapeamento entrada-saída do equalizador Bayesiano se assemelha a uma rede neural RBF.:

- centros: estados do canal.
- dispersão: variância do ruído.
- pesos da camada de saída: vetor de +1s e -1s $\underline{W}$.
- número de neurônios: número de estados (N_s)

$$N_s = 2^{N+M-1}$$

$\hookrightarrow$ aumenta exponencialmente com o número de entradas. (N)