

Redes Neurais Feedforward – Revisão

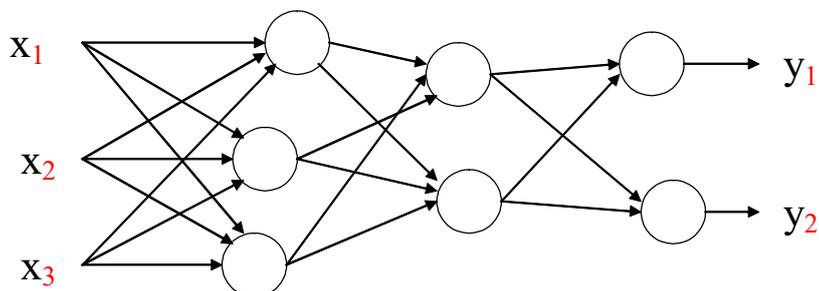
Neurônio - elemento de processamento



$$u_i = \sum_{j=1}^N w_{ij} x_j + w_{i0} = \underline{w}_i^t \underline{x} + w_{i0}$$

$$s_i = u_i \quad \text{ou} \quad s_i = \text{tgh}(u_i)$$

Arquitetura da Rede

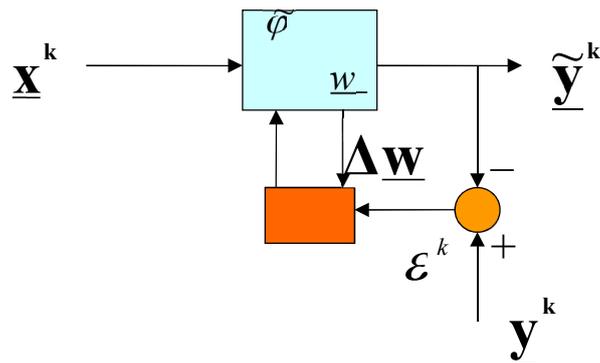


Rede feedforward

Aprendizado

(ou treinamento, ou ajuste de parâmetros)

como um processo de otimização



Aprendizado =
minimização do erro na saída

Erro a minimizar:

$$\varepsilon^{2^k} = \|\underline{y}^k - \tilde{\underline{y}}^k\|^2 = \sum_{l=1}^m (y_l^k - \tilde{y}_l^k)^2$$

Erro médio quadrático:

$$F_0 = E(\varepsilon^{2^k}) = \frac{1}{P} \sum_{k=1}^P \varepsilon^{2^k}$$

$$F_0 = E(\varepsilon^{2^k}) = F_0(\underline{\mathbf{w}}) \geq 0$$

Otimização via Métodos Numéricos Recursivos

$$\underline{w} \rightarrow \underline{w} + \underline{\Delta w} \quad | \quad F(\underline{w} + \underline{\Delta w}) < F(\underline{w})$$

$\underline{\Delta w}$??

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

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

controla
 direção $\underline{\Delta w}$

Gradiente:

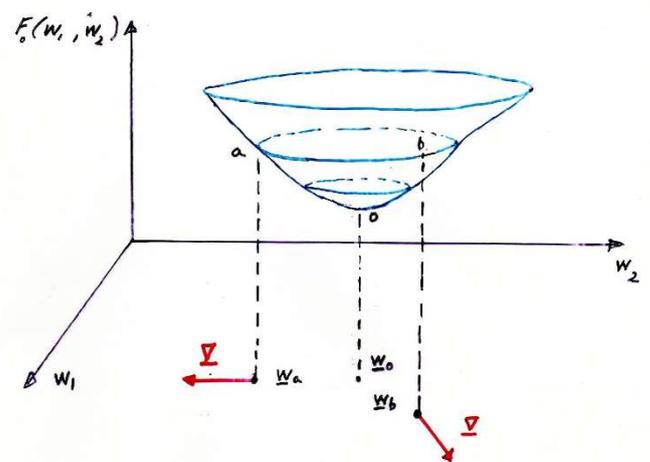
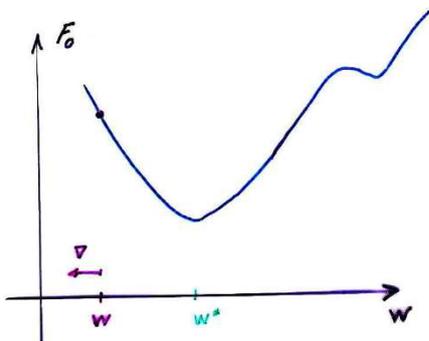
$$F_0(w_1, w_2, \dots) = F(\underline{w})$$

$$\underline{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \end{bmatrix} \quad \nabla_{\underline{w}} F_0 = \underline{\nabla} = \begin{bmatrix} \frac{\partial F_0}{\partial w_1} \\ \frac{\partial F_0}{\partial w_2} \\ \vdots \end{bmatrix}$$

Gradiente descendente

ou descida por gradiente

$$\underline{w} \longrightarrow \Delta \underline{w} = -\alpha \nabla(\underline{w}) \longrightarrow \underline{w} = \underline{w} + \Delta \underline{w}$$



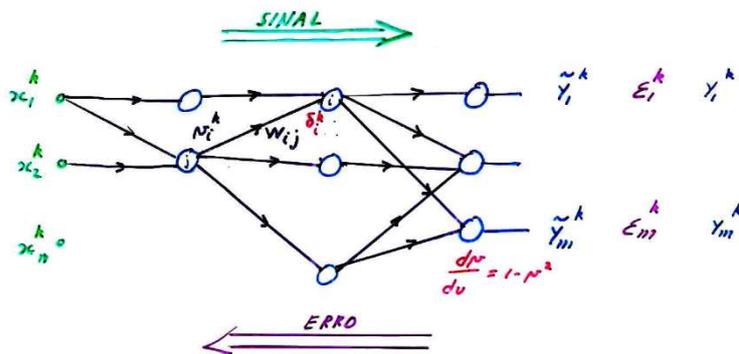
Complexidade de cálculo:

$$\Delta w_{ij} = -\alpha \frac{\partial F}{\partial w_{ij}}$$

Fim do processo:

$$\nabla \underline{w}_{otimo} = \underline{0} \quad \longrightarrow \quad \Delta \underline{w}_{otimo} = \underline{0}$$

Error Backpropagation - Resumo:



$$\frac{\partial \mathcal{E}^{2^k}}{\partial W_{ij}} = -2v_j^k \delta_i^k$$

$$\nabla F_0 = \frac{\partial F_0}{\partial W_{ij}} \Bigg|_{\forall k} = -2 \mathbf{E} \left(v_j^k \delta_i^k \right)$$

$$\Delta W_{ij} = 2\alpha \mathbf{E} \left(v_j^k \delta_i^k \right)$$

Resumo do Processo de Treinamento - Batelada

I - Pré processamento

Selecionar as entradas à usar

Compactar as entradas, se necessário

Normalizar as entradas e saídas, verificar intrusos

Montar os conjuntos de pares entrada-saída de treinamento, teste e validação

Dimensionar a rede, atribuir valores iniciais às sinapses

Definir o método de treinamento

Definir o critério de parada do treinamento (overtraining) e o intervalo de teste de parada N

II - Fase de Treinamento

BP resiliente com controle de overtraining

Até que o critério de parada esteja satisfeito

Aplicar N épocas de treinamento

Época de Treinamento

Para cada par entrada-saída do conjunto de treinamento

Aplicar \underline{x} na rede; calcular v_j , f'_i , y_{l-rede} , ϵ_l

Criar a rede associada;

calcular os erros retropropagados δ_i

calcular $v_j \delta_i$ para cada sinapse

Outro par

Calcular o valor médio $E[v_j \delta_i]$ e α_{ij} de para cada sinapse w_{ij}

Atualizar as sinapses com $\Delta w_{ij} = 2 \alpha_{ij} E[v_j \delta_i]$

Aplicar os pares de treinamento, validação e teste

Plotar os erros; verificar critério de parada

Fim

III - Crítica Pós Treinamento

Histograma dos erros e

Correções de anomalias

IV – Processos Pós Treinamento

Relevância das entradas

Absorção do escalamento

Poda da Rede, etc.