Juiz de Fora, 14 a 17 de Setembro de 2008

Minicurso:

Componentes Principais Generalizadas Via Redes Neurais

Luiz P. Calôba

COPPE / Poli / UFRJ

caloba@ufrj.br www.lps.ufrj.br/~caloba/MinicursoCBA08.pdf

Resumo:

A Análise por Componentes Principais (PCA) ou Transformada de Kahunen-Löeve é provavelmente o método mais utilizado para compactação de informação. Vantagens adicionais além da redução da dimensionalidade são (a) o fornecimento de componentes ortogonais, descorrelacionadas, o que facilita uma série de processamentos posteriores da informação, e eventualmente (b) a redução do ruído presente na informação. Entretanto este tipo de compactação otimiza apenas a reconstrução da informação. Se uma informação $\underline{x}$ é compactada e preservada é porque, na maioria das vezes, ela vai ser utilizada para gerar algum resultado específico $\underline{y}$, isto é, utilizada como entrada para um mapeamento *M:* $\underline{x} \rightarrow \underline{y}$. E nem sempre a compactação que otimiza a reconstrução de $\underline{x}$ é a ótima para gerar $\underline{y}$. Neste mini-curso discutimos a utilização de redes neurais para compactar a informação $\underline{x}$ de modo a otimizar o resultado final desejado $\underline{y}$ mantendo as vantagens adicionais (a) e (b) mencionadas acima.

Representação por Componentes Principais

PCA - Principal Components Analysis

- Generalização

$$\underline{x} \xrightarrow{\eta} \underline{y}$$

$$\underline{x} \xrightarrow{\eta_1} \underline{z} \xrightarrow{\eta_2} \underline{\tilde{y}}$$

$$\dim \underline{z} < \dim \underline{x}, \dim \underline{y}$$

Compressão da Informação

Redução da Dimensionalidade

Descorrelação

$$r(z_i, z_j) = 0 \qquad \forall\, i, j$$

Simplificação do Processo

$$E(\underline{x}) = \underline{0}$$

$$E(\underline{y}) = \underline{0}$$

$$\underline{x} \longrightarrow \underline{x}' = \underline{x} - E(\underline{x})$$

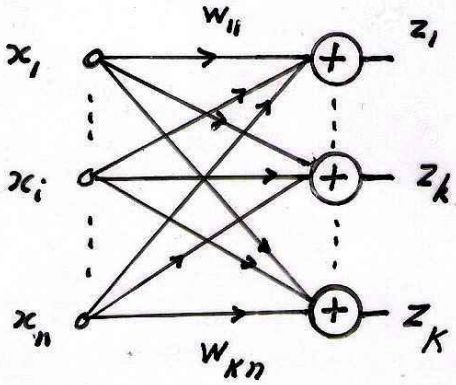
$$\underline{y} \longrightarrow \underline{y}' = \underline{y} - E(\underline{y})$$

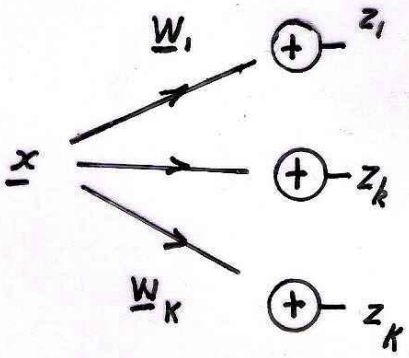
Tipos de mapeamentos:

$\mathcal{M}$	$\mathcal{M}_1$	$\mathcal{M}_2$	
Linear	L	L	*
	NL	NL	
Não Linear	L	NL	*
	NL	L	
	NL	NL	

Implementação dos mapeamentos

$\mathcal{M}_1$ linear $\quad \underline{z} = \underline{W}\,\underline{x}$





neurônios SEM BIAS

$$z_k = \underline{x}^t \underline{W}_k$$

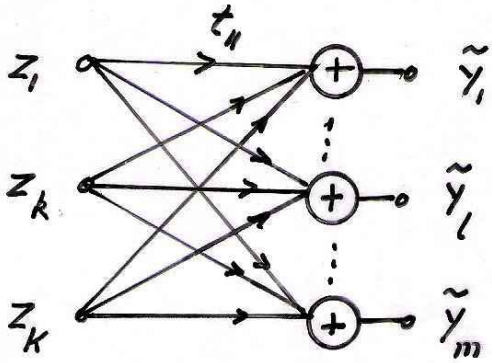
$$\underline{W}_k = \begin{bmatrix} w_{k1} \\ w_{k2} \\ \vdots \\ w_{kn} \end{bmatrix}$$

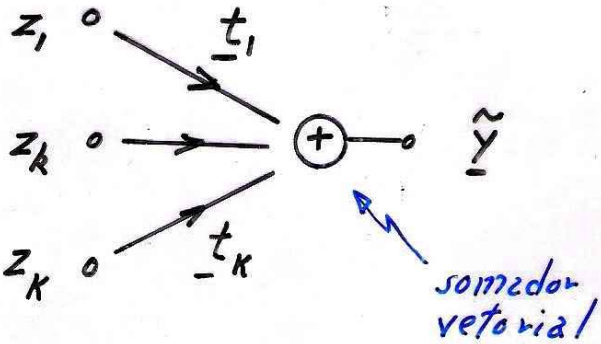
in-star de Grossberg

$$\underline{W} = [w_{ij}] = \begin{bmatrix} \underline{W}_1^t \\ \underline{W}_2^t \\ \vdots \\ \underline{W}_K^t \end{bmatrix}$$

m_2 linear

$$\underline{\tilde{y}} = \underline{T}\,\underline{z}$$





neurônios SEM BIAS

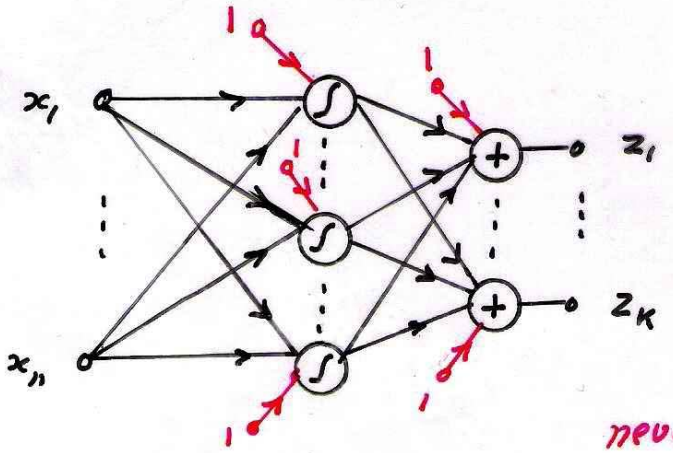
$$\underline{\tilde{y}} = \sum_{k=1}^{K} z_k\,\underline{t}_k$$

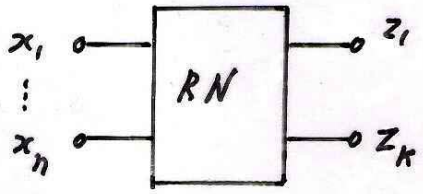
$$\underline{t}_k = \begin{bmatrix} t_{k1} \\ t_{k2} \\ \vdots \\ t_{kK} \end{bmatrix}$$

out-star de Grossberg

$$\underline{T} = [\,\underline{t}_1 \;\; \underline{t}_2 \;\; \cdots \;\; \underline{t}_K\,]$$

η_1 não linear $\underline{z} = \varphi(\underline{x})$

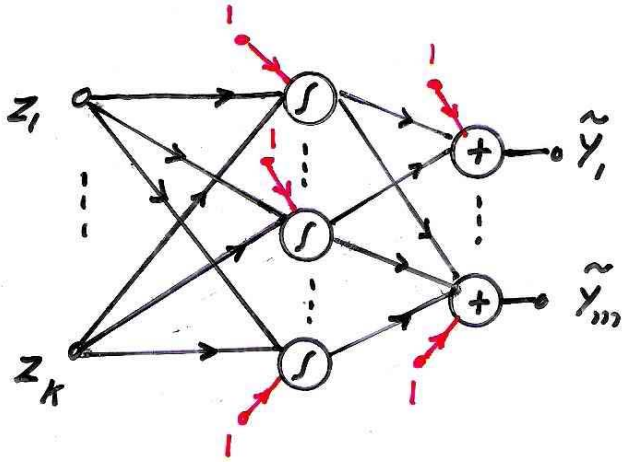


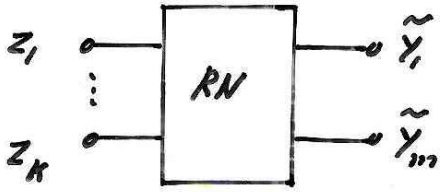


neurônios COM BIAS

η_2 não linear $\tilde{y} = \phi(\underline{z})$

similar

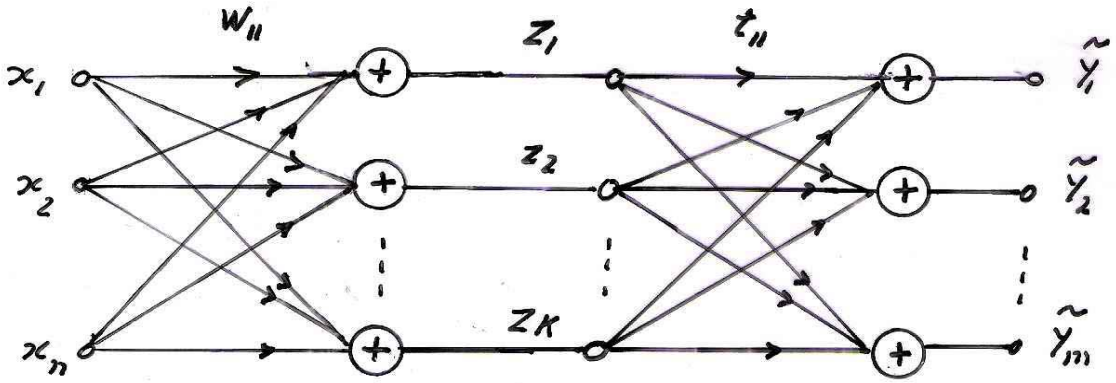


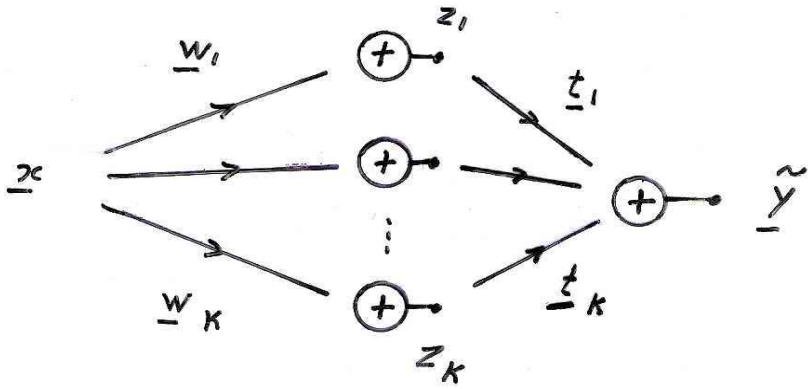


Mapeamento Linear

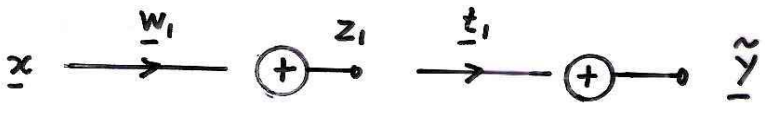
$\mathcal{M}_1$ linear $\quad \underline{z} = \underline{W}\,\underline{x}$

$\mathcal{M}_2$ linear $\quad \underline{\tilde{y}} = \underline{T}\,\underline{z}$



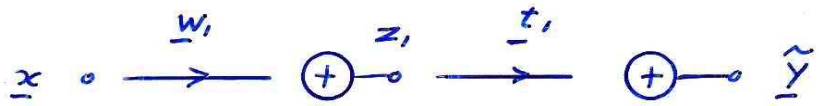


1ª componente



$z_1 = \underline{x}^t \underline{w}_1 \qquad \underline{\tilde{y}} = z_1\,\underline{t}_1 = \underline{x}^t \underline{w}_1\,\underline{t}_1$

Primeira componente



$$z_1 = \underline{x}^t \underline{w}_1 = \sum_{i=1}^{n} x_i w_{1i}$$

$$\underline{\tilde{y}} = z_1 \underline{t}_1 \qquad \tilde{y}_j = z_1 t_{j1}$$

Critério

$\underline{w}_1$, $\underline{t}_1$ /

$F = E\,|\underline{y} - \underline{\tilde{y}}|^2$ mínimo

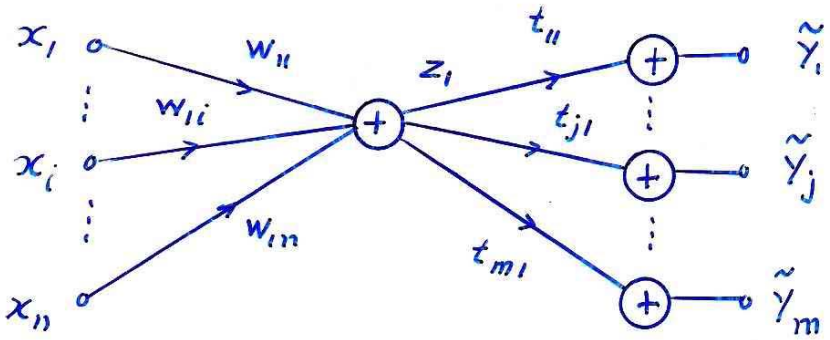
$$\Rightarrow \quad \nabla_{\underline{w}_1} F = \underline{0}$$

$$\nabla_{\underline{t}_1} F = \underline{0}$$

$$\Rightarrow \quad \frac{\partial F}{\partial w_{1i}} = 0 \qquad \forall i$$

$$\frac{\partial F}{\partial t_{j1}} = 0 \qquad \forall j$$

Treinamento da Rede da Primeira Componente



$$(\underline{x}, \underline{y}) \longrightarrow \underline{\tilde{y}} \simeq \underline{y} \qquad P \text{ pares}$$

$$F = \underset{\forall (\underline{x}, \underline{y})}{E} \; |\underline{y} - \underline{\tilde{y}}|^2$$

Gradiente descendente

$$\Delta w_{1i} = -\alpha \frac{\partial F}{\partial w_{1i}}$$

$$\Delta t_{j1} = -\alpha \frac{\partial F}{\partial t_{j1}}$$

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

$$= \sum_l \varepsilon_l^2$$

$$\frac{\partial \varepsilon^2}{\partial t_{j1}} = 2 \sum_l \varepsilon_l \frac{\partial \varepsilon_l}{\partial t_{j1}} = -2 \sum_l \varepsilon_l \frac{\partial \tilde{y}_l}{\partial t_{j1}}$$

$$\tilde{y}_l = z_1 t_{l1} \qquad \frac{\partial \varepsilon^2}{\partial t_{j1}} = -2 \varepsilon_j z_1$$

$$\frac{\partial \varepsilon^2}{\partial w_{1i}} = -2 \sum_l \varepsilon_l \frac{\partial \tilde{y}_l}{\partial w_{1i}}$$

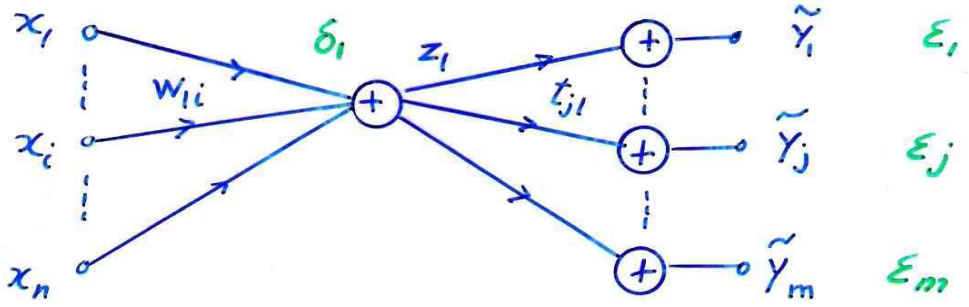
$$\tilde{y}_l = t_{l1} z_1 = t_{l1} \sum_{j=1}^{n} w_{1j} x_j$$

$$\frac{\partial \varepsilon^2}{\partial w_{1i}} = -2 \sum_l \varepsilon_l t_{l1} x_i = -2 x_i \underbrace{\sum_l \varepsilon_l t_{l1}}_{\delta_1}$$

$$\delta_1 = \sum_l \varepsilon_l t_{l1}$$

$$\frac{\partial \varepsilon^2}{\partial w_{1i}} = -2 x_i \delta_1$$

$$\delta_1 = \sum_{l=1}^{m} \varepsilon_l \, t_{l1}$$



$$\Delta w_{1i} = 2\alpha \, x_i \, \delta_1$$

$$\Delta t_{j1} = 2\alpha \, z_1 \, \varepsilon_j$$

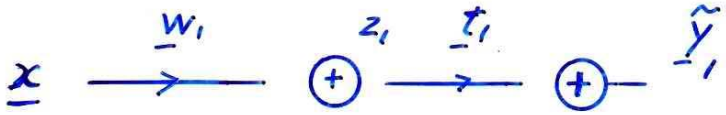
Normalização

$$\underline{\tilde{y}}_1 = z_1 \, \underline{t}_1 = \underline{x}^t \, \underline{w}_1 \, \underline{t}_1$$

$$\underline{w}_1 \longrightarrow \frac{1}{|\underline{w}_1|} \, \underline{w}_1$$

$$\underline{t}_1 \longrightarrow |\underline{w}_1| \, \underline{t}_1$$

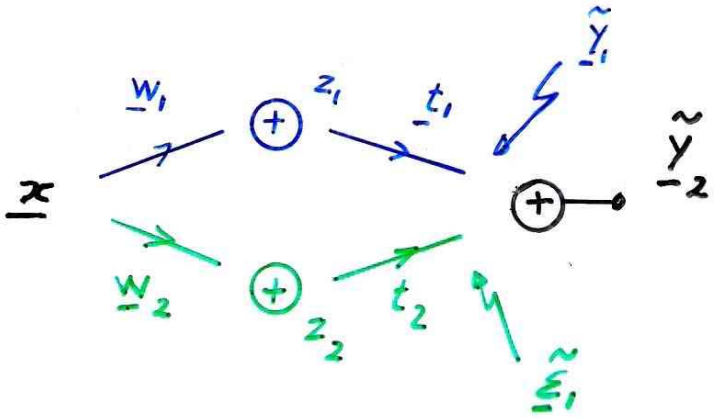
Primeira Componente



$$\underline{\varepsilon}_1 = \underline{y} - \underline{\tilde{y}}_1$$

$$\varepsilon_1^2 = |\underline{\varepsilon}_1|^2$$

Segunda componente



$$\underline{\varepsilon}_2 = \underline{y} - \underline{\tilde{y}}_2$$

$$\varepsilon_2^2 = |\underline{\varepsilon}_2|^2$$

$$\varepsilon_2^2 < \varepsilon_1^2$$

Treinamento

- "Congelar" $\underline{w}_1$ e $\underline{t}_1$

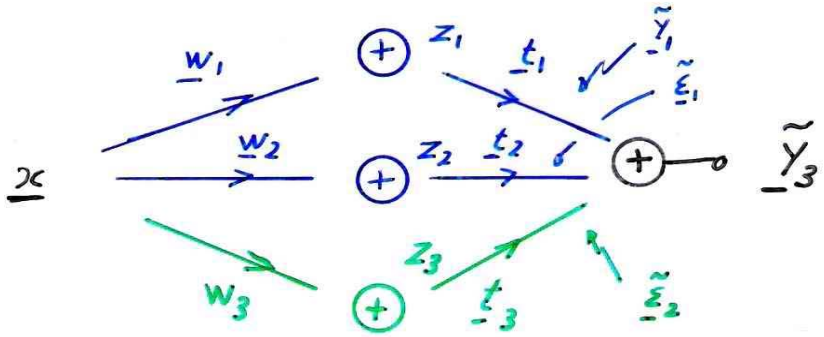
- Treinar $\underline{w}_2$ e $\underline{t}_2$

Terceira componente

Introduzir $\underline{w}_3$ e $\underline{t}_3$

Congelar $\underline{w}_1$, $\underline{t}_1$, $\underline{w}_2$ e $\underline{t}_2$

Treinar $\underline{w}_3$ e $\underline{t}_3$



$\tilde{\underline{y}}_1 \simeq \underline{y}$ $\qquad \underline{\varepsilon}_1 = \underline{y} - \tilde{\underline{y}}_1$

$\tilde{\underline{\varepsilon}}_1 \simeq \underline{\varepsilon}_1$ $\qquad \tilde{\underline{y}}_2 = \tilde{\underline{y}}_1 + \tilde{\underline{\varepsilon}}_1$

$$\underline{\varepsilon}_2 = \underline{y} - \tilde{\underline{y}}_2 =$$

$$\underline{y} - (\tilde{\underline{y}}_1 + \tilde{\underline{\varepsilon}}_1) =$$

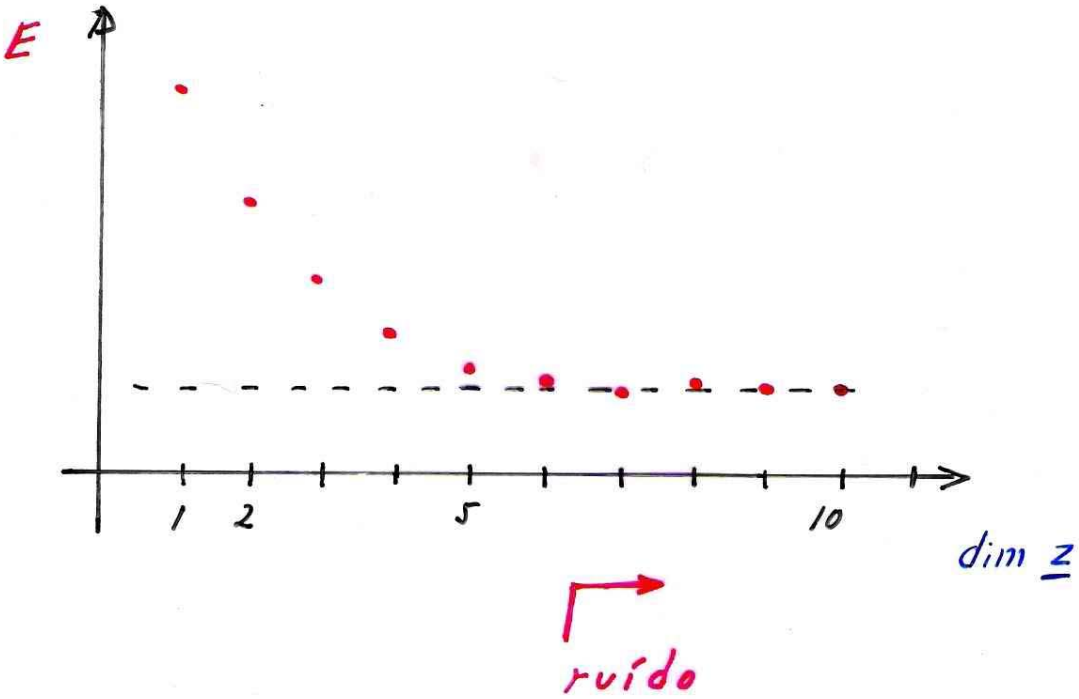
$$\underline{y} - [\underline{y} - \underline{\varepsilon}_1 + \tilde{\underline{\varepsilon}}_1] =$$

$$\underline{\varepsilon}_1 - \tilde{\underline{\varepsilon}}_1$$

$\tilde{\underline{\varepsilon}}_2 \simeq \underline{\varepsilon}_2$ $\qquad \tilde{\underline{y}}_3 = \tilde{\underline{y}}_2 + \tilde{\underline{\varepsilon}}_2$

Erro $E\,|\underline{y} - \underline{\tilde{y}}|^2$ versus

nº de componentes dim $\underline{z}$



Normalização

$$\underline{\tilde{y}}_1 = \underline{x}^t \underline{w}_1 \underline{t}_1$$

$$\underline{w}_1 \longrightarrow \frac{1}{|\underline{w}_1|} \underline{w}_1$$

$$\underline{t}_1 \longrightarrow |\underline{w}_1| \underline{t}_1$$

e

$$\underline{w}_2 \longrightarrow \frac{1}{|\underline{w}_2|} \underline{w}_2$$

$$\underline{t}_2 \longrightarrow |\underline{w}_2| \underline{t}_2$$

etc...

Ortogonalidade

$\underline{w}_1$, $\underline{t}_1$ / $|\underline{\varepsilon}_1|$ mínimo

$$\underline{\tilde{y}}_1 = \underline{x}^t \underline{w}_1 \underline{t}_1 = \underline{y}_1 - \underline{\varepsilon}_1$$

$$\underline{\tilde{y}}_2 = \underbrace{\underline{x}^t \underline{w}_1 \underline{t}_1}_{\underline{\tilde{y}}_1} + \underbrace{\underline{x}^t \underline{w}_2 \underline{t}_2}_{\underline{\tilde{\varepsilon}}_1}$$

Se $\underline{w}_2 = \alpha \underline{w}_1 + \underline{w}_\perp$

$$\underline{\tilde{y}}_2 = \underline{x}^t \underline{w}_1 \underline{t}_1 + \underline{x}^t \alpha \underline{w}_1 \underline{t}_2 + \underline{x}^t \underline{w}_\perp \underline{t}_2$$

$$= \underline{x}^t \underline{w}_1 (\underline{t}_1 + \alpha \underline{t}_2) + \underline{x}^t \underline{w}_\perp \underline{t}_2$$

$$= \underline{t}_1$$

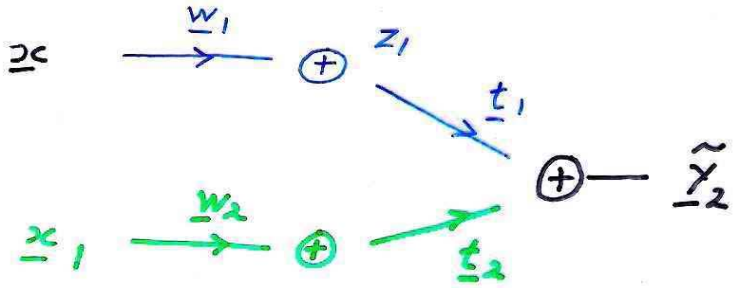
$\therefore \quad \alpha = 0 \qquad \underline{w}_2 = \underline{w}_\perp$

Garantindo a ortogonalidade

$$\underline{w}_{2\ inicial} = \underline{r} - \underline{r}^t \underline{w}_1 \underline{w}_1$$

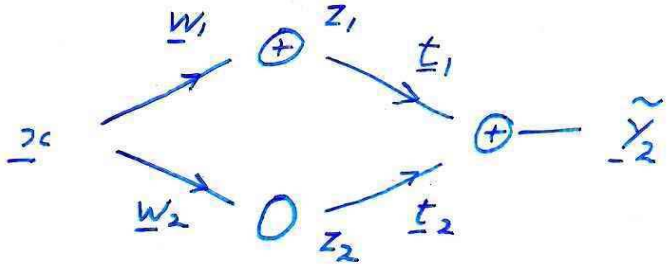
$\underline{r}$ = randomico

Treinamento de $\underline{w}_2$



$$\underline{x}_1 = \underline{x} - \underline{x}^t \underline{w}_1 \underline{w}_1$$

Operação

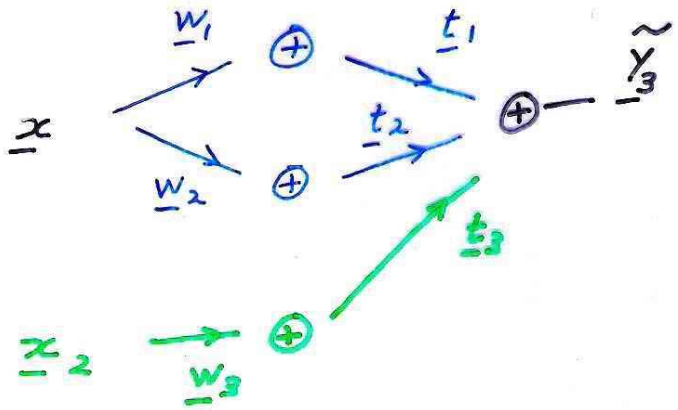


Treinamento de $\underline{W}_3$

$\underline{W}_{3\ inicial}$ =

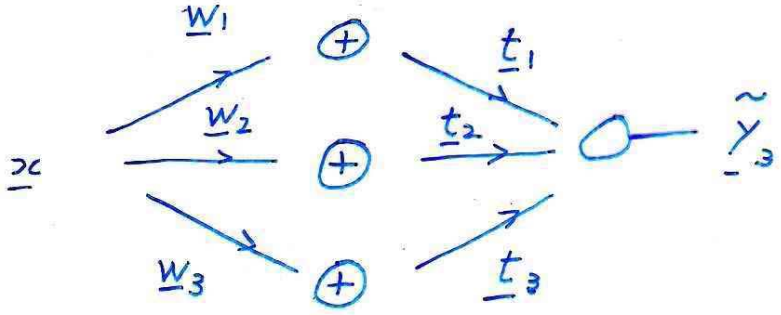
$$\underline{r} - \underline{r}^t \underline{w}_1 \underline{w}_1 - \underline{r}^t \underline{w}_2 \underline{w}_2$$

Treinamento

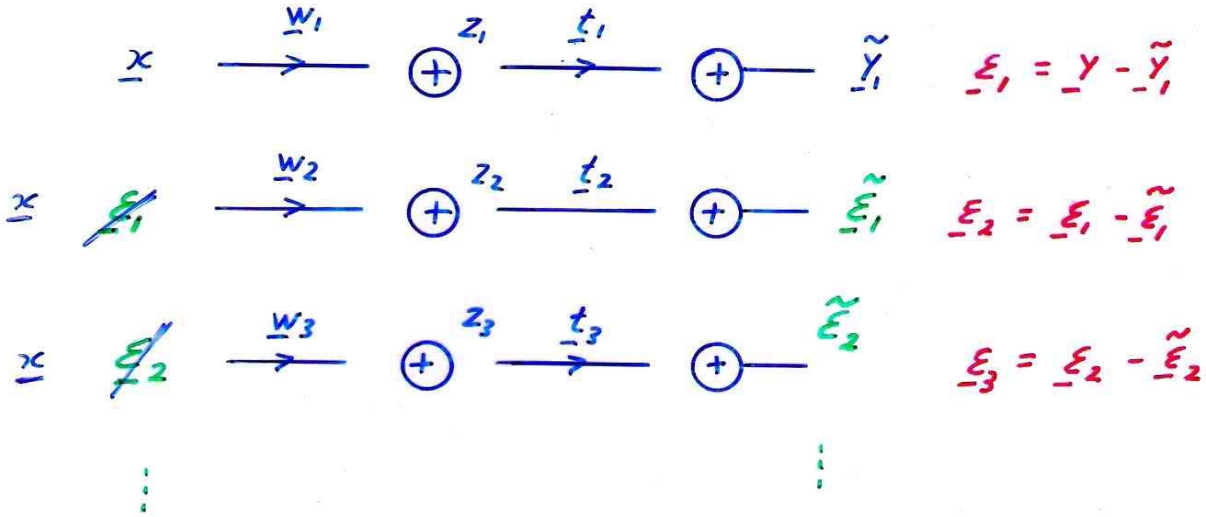


$$\underline{x}_2 = \underline{x}_1 - \underline{x}_1^t \underline{w}_2 \underline{w}_2$$

Operação



Treinamento Alternativo

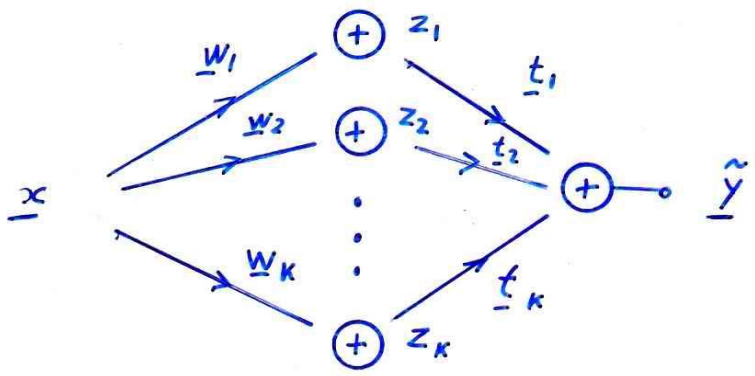


convergência mais rápida

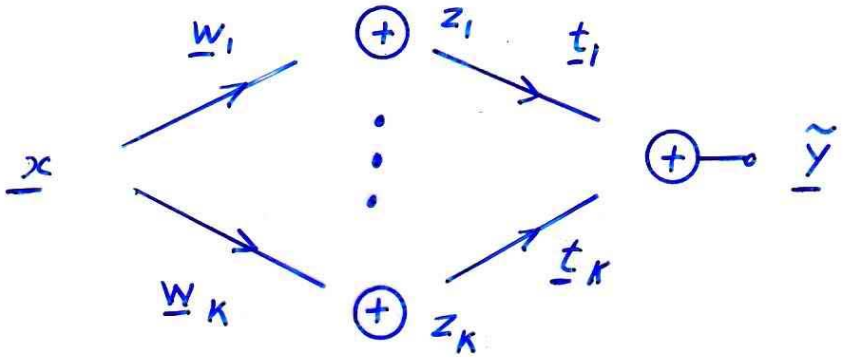
processamento paralelo

Operação

$$\tilde{y} = \tilde{y}_K = \tilde{y}_1 + \tilde{\varepsilon}_1 + \tilde{\varepsilon}_2 + \dots + \tilde{\varepsilon}_{K-1}$$



Outro treinamento alternativo



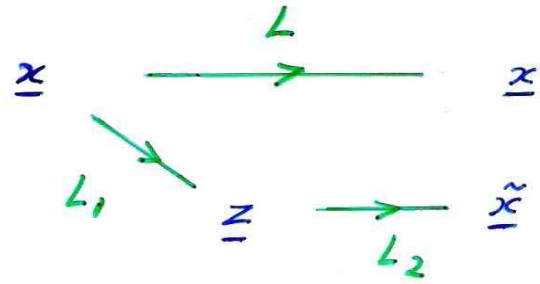
Treinamento simultâneo de todas as sinapses

↓ mais rápido

$[\underline{W}]$ é apenas uma base para o espaço das componentes principais.

Compressão da informação

para (auto) reprodução



L $\quad \underline{x} = [I]\,\underline{x}$

L_1 $\quad \underline{z} = [\underline{W}]\,\underline{x}$

L_2 $\quad \underline{\tilde{x}} = [T]\,\underline{z} = [W^{-1}]\,\underline{z}$

Transformada de Karhunen - Loève

Componentes Principais propriamente ditas

Rede: BP auto supervisionada

Outros processos neurais

Oja , Sanger - Oja

Se normalizarmos

$$|\underline{w}_i| = \underline{w}_i^t \underline{w}_i = 1 \qquad \forall i$$

E como $\underline{w}_i$'s são ortogonais

$$\underline{w}_i^t \underline{w}_j = 0 \qquad \forall i \neq j$$

$[W]$ é "ortogonal"

$$[W][\underline{w}^t] = [I] \quad \text{ou}$$

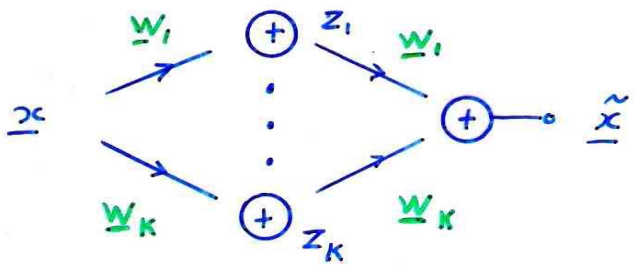
$$[W^{-1}] = [W^t]$$

logo

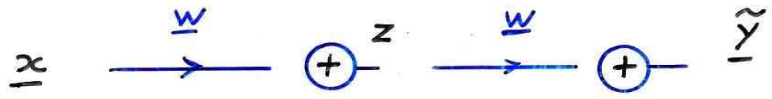
$$[T] = [W^{-1}] = [W^t]$$

ou

$$\underline{t}_i = \underline{w}_i \qquad \forall i$$



Treinamento BP - Regra delta



$$z = \sum_j w_j x_j \qquad \tilde{y}_i = z\, w_i$$

$$\frac{d\tilde{y}_i}{dw_k} = \begin{cases} w_i \dfrac{dz}{dw_k} = w_i x_k & i \neq k \\ z + w_k \dfrac{dz}{dw_k} = z + w_k x_k & i = k \end{cases}$$

$$\varepsilon^2 = \sum_i \varepsilon_i^2 = \sum_i (y_i - \tilde{y}_i)^2$$

$$\frac{d\varepsilon^2}{dw_k} = \sum_i 2\,\varepsilon_i \frac{d\varepsilon_i}{dw_k} = -2 \sum_i \varepsilon_i \frac{d\tilde{y}_i}{dw_k} =$$

$$= -2 \left[\sum_{i \neq k} \varepsilon_i \frac{d\varepsilon_i}{dw_k} + \varepsilon_k \frac{d\varepsilon_k}{dw_k} \right] =$$

$$= -2 \left[\sum_{i \neq k} \varepsilon_i w_i x_k + \varepsilon_k (z + w_k x_k) \right]$$

$$= -2 \left[\varepsilon_k z + x_k \sum_i \varepsilon_i w_i \right]$$

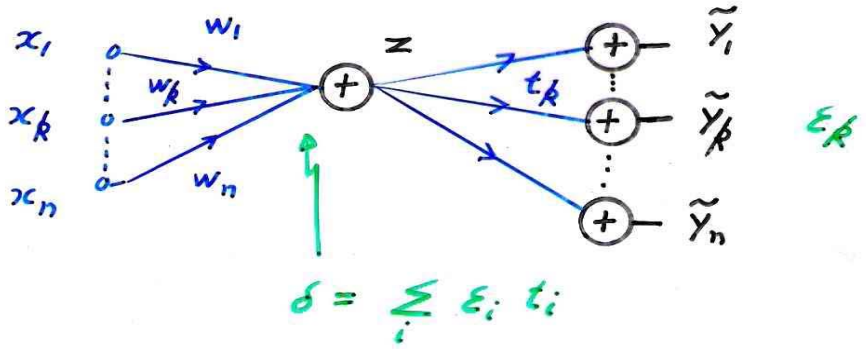
$$\Delta w_k = E\left\{ -\alpha \frac{d\varepsilon^2}{dw_k} \right\} =$$

$$= 2\alpha\, E\left\{ z\,\varepsilon_k + x_k \sum_i \varepsilon_i w_i \right\}$$

Obs: $|\underline{W}_{inicial}| = 1$

Obs:

Sinapses independentes



$$\Delta t_k = 2\alpha\, E\left\{ z\,\varepsilon_k \right\} \qquad (1)$$

$$\Delta w_k = 2\alpha\, E\left\{ x_k\,\delta \right\} \qquad (2)$$

Sinapses iguais $t_k = w_k$

$$\Delta w_k = \Delta t_k = 2\alpha\, E\left\{ z\,\varepsilon_k + x_k\,\delta \right\}$$

(1) + (2)

Obs:

Mapeamento linear

⇓

F quadrática

↳ treinamento

↓

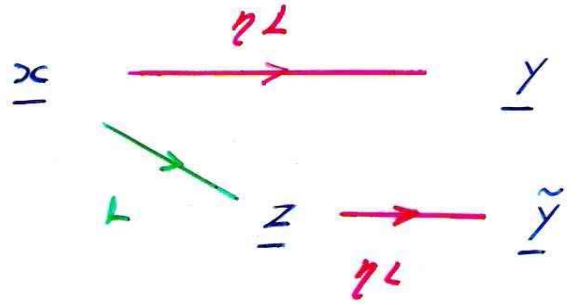
rápido

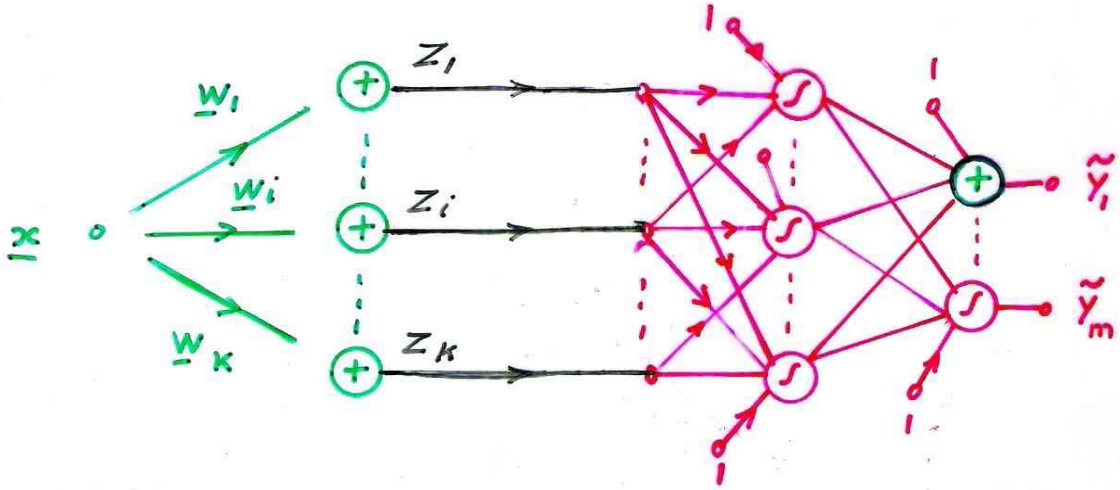
sem mínimos locais

sem overtraining

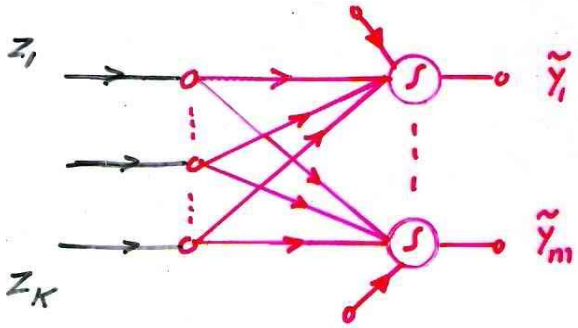
∃ solução analítica

Mapeamento não Linear I



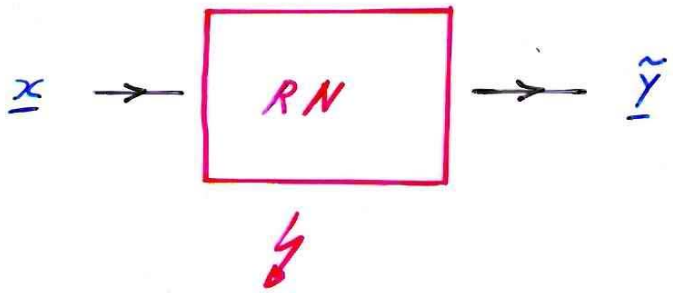


sem bias com bias



Dimensionando a RN

Mapear SEM comprimir



1 - 2 camadas

Q # neurônios camada intermediária

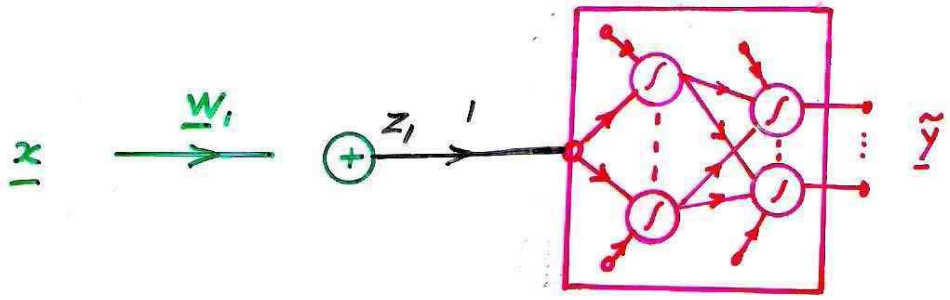
$\Rightarrow \quad \dim \underline{z} \leq Q$

1 camada $\Rightarrow \quad \dim \underline{z} \leq \dim \underline{y}$

Componentes para

ATUAÇÃO independente

1-



RN com uma entrada

Treinar todas as sinapses,

$\underline{W}_1$ e RN

normalizar $\underline{W}_1 \rightarrow \dfrac{\underline{W}_1}{|\underline{W}_1|}$

2 - Segunda Componente

$$\underline{x}_1 = \underline{x} - \underline{x}^t \underline{w}_1 \underline{w}_1$$

$$\underline{w}_{2\ inicial} = \underline{r} - \underline{r}^t \underline{w}_1 \underline{w}_1$$

retornar ao passo 1 fazendo

$$\underline{x}_1 \longrightarrow \underline{x}$$

$$\underline{w}_2 \longrightarrow \underline{w}_1$$

$$z_2 \longrightarrow z_1$$

3 - Terceira Componente

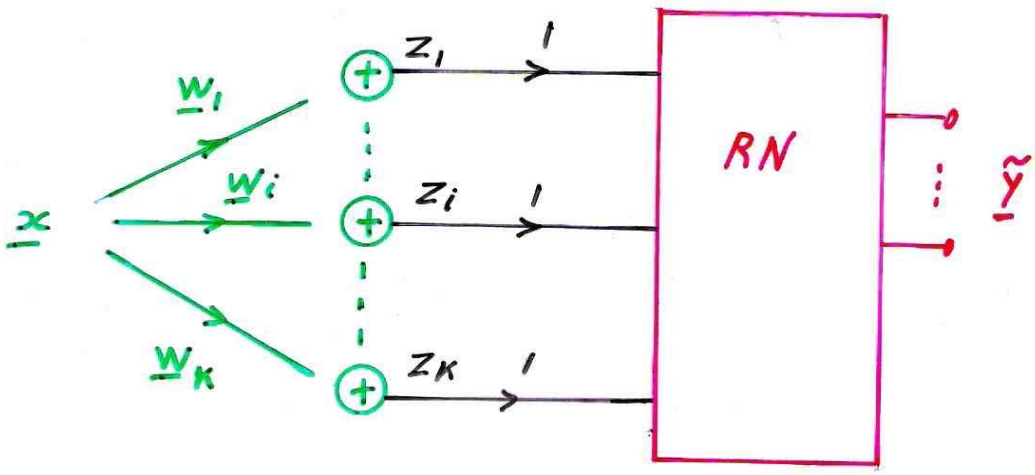
...

Rede para ATUAÇÃO conjunta

1 - Determinar $\underline{W}_1, \underline{W}_2, \ldots, \underline{W}_K$

2 - Treinar RN,

mantendo $\underline{W}_i$ congelados



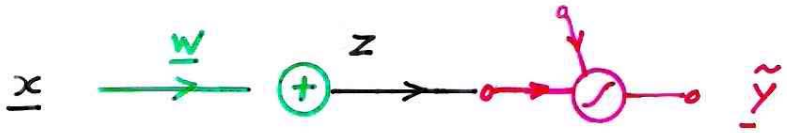
Componentes calculadas para

ATUAÇÃO independente

Usadas em ATUAÇÃO conjunta

Um caso particular:

o classificador linear



CDA - Canonical discrimination analisys

Discriminador linear ótimo

Discriminador de Fischer

Componentes para ATUAÇÃO conjunta ou cooperativa

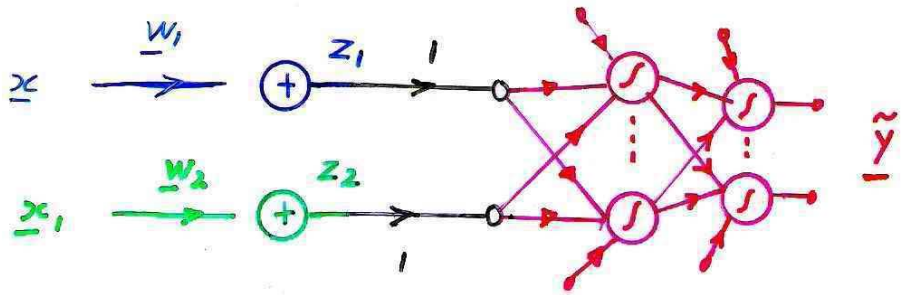
1 - Passo igual ao treinamento para componentes com atuação independente.

2 - Congelar $\underline{W}_1$ $\quad |\underline{W}_1| = 1$

Acrescentar $\underline{W}_2$ para gerar z_2

Acrescentar uma entrada à RN

Treinar $\underline{W}_2$ e TODA a RN (inclusive as sinapses conectadas à z_1)



$$\underline{x}_1 = \underline{x} - \underbrace{\underline{x}^t \underline{W}_1}_{z_1} \underline{W}_1$$

$$\underline{W}_{2\,inicial} = \underline{r} - \underline{r}^t \underline{W}_1 \underline{W}_1$$

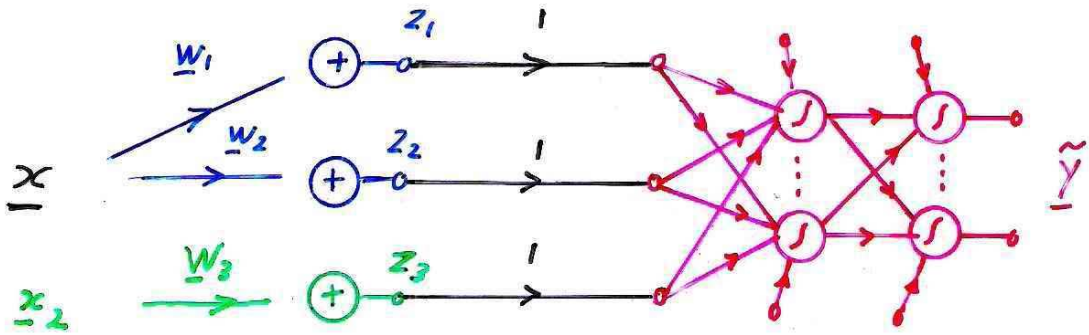
3 - Congelar $\underline{W}_1$ e $\underline{W}_2$ $\quad |\underline{W}_2| = 1$

Acrescentar $\underline{W}_3$ para gerar z_3

Acrescentar uma entrada a RN

Treinar $\underline{W}_3$ e TODA a RN

(inclusive as sinapses conectadas a z_1 e z_2)



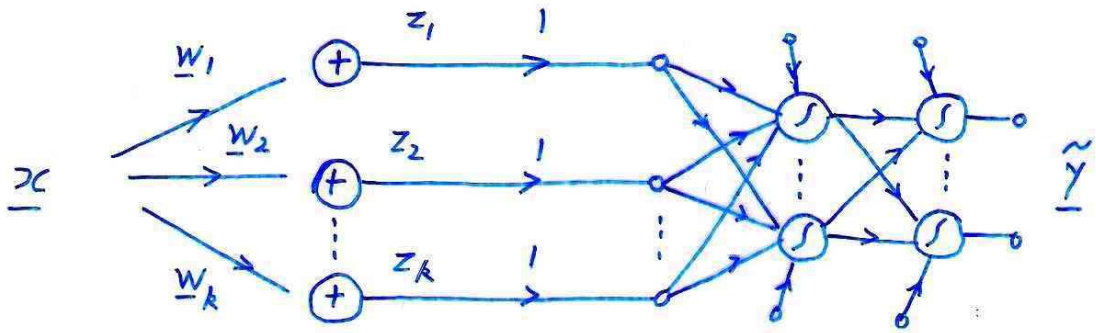
$$\underline{x}_2 = \underline{x}_1 - \underline{x}_1^t \underline{W}_1 \underline{W}_1$$

$$\underline{W}_{3\,inicial} = \underline{r} - \underline{r}^t \underline{W}_1 \underline{W}_1 - \underline{r}^t \underline{W}_2 \underline{W}_2$$

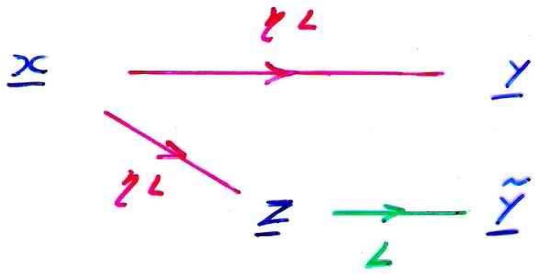
Obs : dim z ≤

neuronios camada intermediaria

Operação : última rede treinada



Mapeamento não Linear II



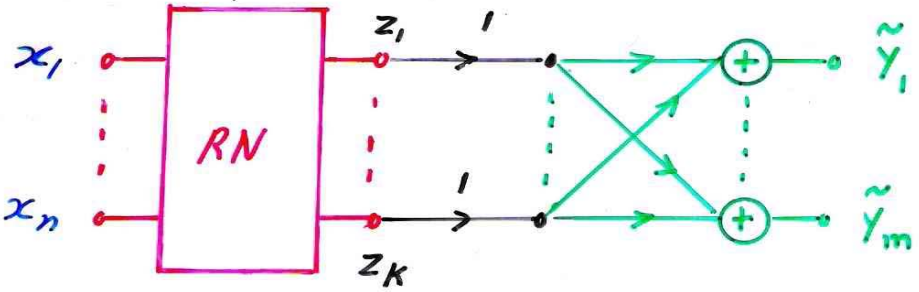
1 - Comprimir $\underline{y}$

$$\underline{z} = \underline{W}\,\underline{y} \qquad \underline{\tilde{y}} = \underline{W}^t\,\underline{z}$$

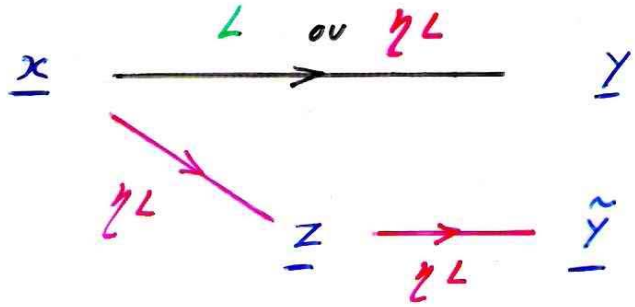
2 - Realizar o mapeamento

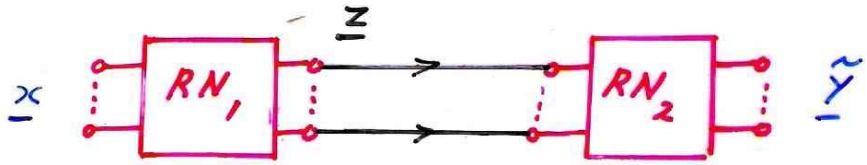
$$\underline{z} = \varphi(\underline{x}) \quad \text{via RN}$$

3 - Rede final



Mapeamento não Linear III





Como RN são mapeadores

universais

⇓

teóricamente dim z = 1

satisfaz !

Conclusões:

Neste mini-curso discutimos a generalização do conceito de componentes principais objetivando a compressão da informação de entrada $\underline{\boldsymbol{x}}$ de modo a otimizar a saída $\underline{\boldsymbol{y}}$ de um mapeamento qualquer $\underline{\boldsymbol{x}}: \rightarrow \underline{\boldsymbol{y}}$, linear ou não linear. Os métodos de compressão estudados utilizam redes neurais feedforward com aprendizado supervisionado tipo retropropagação do erro. Nos casos de mapeamentos não lineares a não linearidade pode ser introduzida no mapeamento de compressão ou de descompressão da informação.

Fim.

Obrigado pelo interesse.