

Minicurso CBA 2008:***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-Löve 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

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

$$\underline{x} \xrightarrow{M_1} \underline{z} \xrightarrow{M_2} \underline{\tilde{y}}$$

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

Compressão da Informação

Redução da Dimensionalidade

Descorrelação

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

Simplificação do Processo

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

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

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

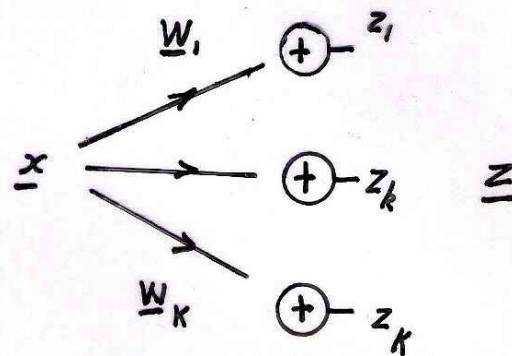
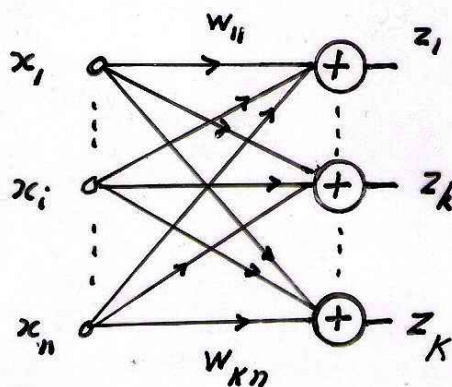
$$\underline{y} \rightarrow \underline{y}' = \underline{y} - E(\underline{y})$$

Tipos de mapeamentos:

m	m_1	m_2	
Linear	L	L	*
	NL	NL	
Não Linear	L	NL	*
	NL	L	
	NL	NL	

Implementação dos mapeamentos

m_1 linear $\underline{z} = \underline{W} \underline{x}$



neurônios SEM BIAS

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

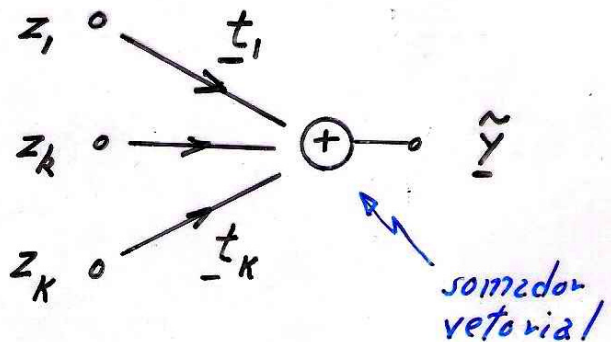
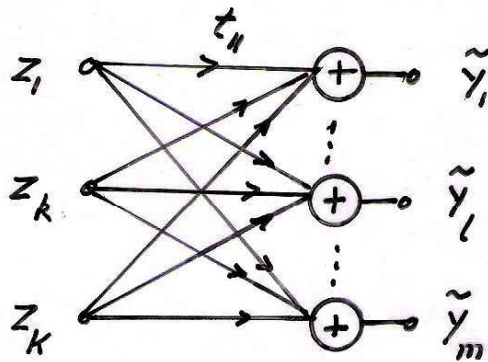
$$\underline{w}_k = \begin{bmatrix} w_{k1} \\ w_{k2} \\ \vdots \\ w_{kn} \end{bmatrix}$$

in-ster
de Grossberg

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

η_2 linear

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



neurônios SEM BIAS

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

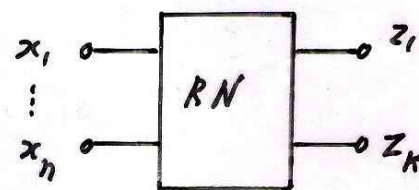
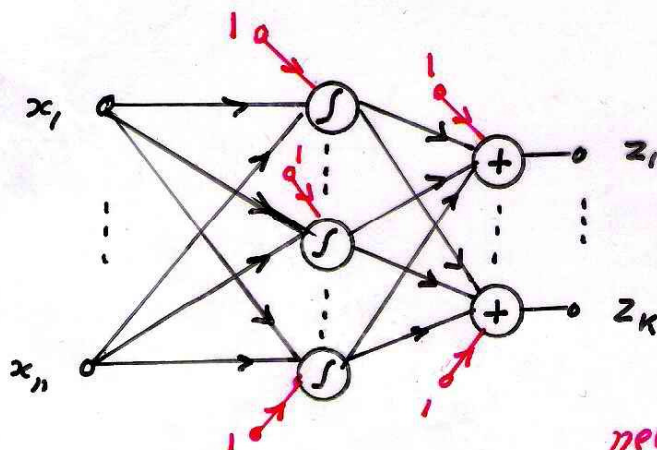
$$\underline{t}_k = \begin{bmatrix} t_{k1} \\ t_{k2} \\ \vdots \\ t_{kK} \end{bmatrix}$$

out-star
de Grossberg

$$\underline{T} = [\underline{t}_1 \quad \underline{t}_2 \quad \dots \quad \underline{t}_K]$$

η_1 não linear

$$\underline{z} = \varphi(\underline{x})$$

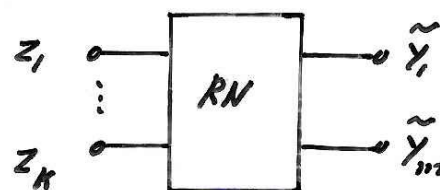
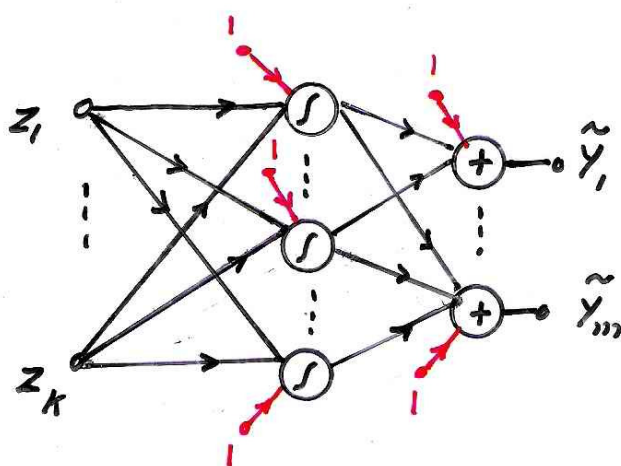


neurônios COM BIAS

η_2 não linear

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

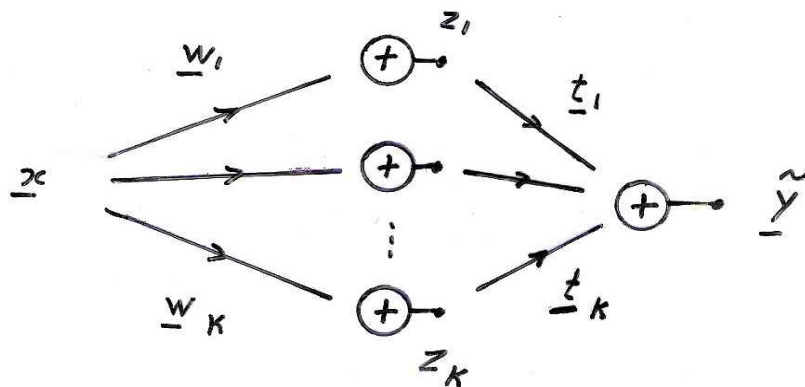
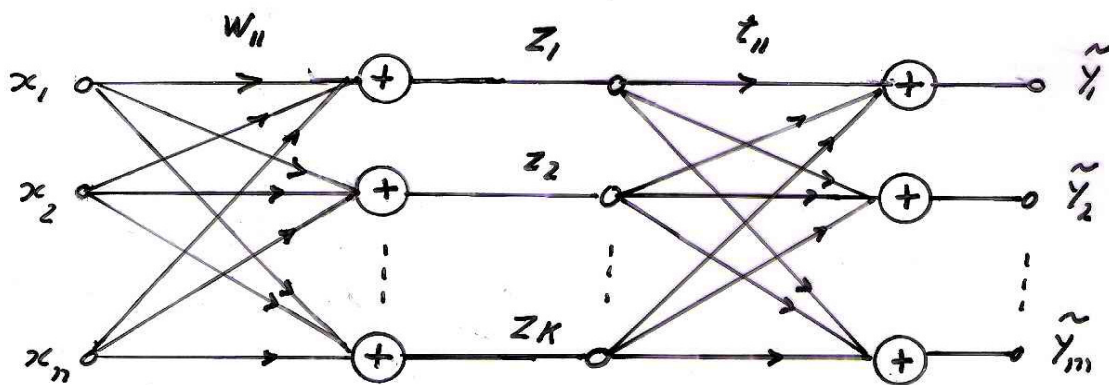
similar



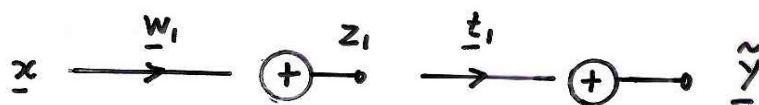
Mapeamento Linear

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

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

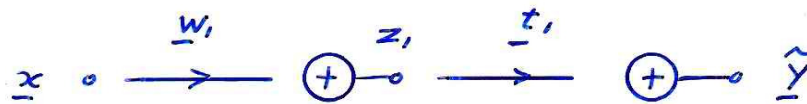


1ª componente



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

Primeira componente



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

$$\tilde{y} = z_1 \underline{t}_1 \quad \tilde{y}_j = z_1 t_{j,1}$$

Critério

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

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

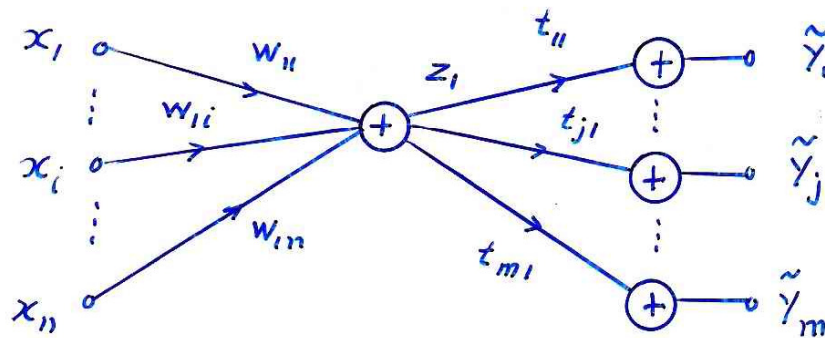
$$\Rightarrow \nabla_{\underline{w}_1} F = \underline{0}$$

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

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

$$\frac{\partial F}{\partial t_{j,1}} = 0 \quad \forall j$$

Treinamento da Rede da Primeira Componente



$$(\underline{x}, \underline{y}) \rightarrow \underline{\tilde{y}} \approx \underline{y} \quad P \text{ pares}$$

$$F = E \sum (\underline{y} - \underline{\tilde{y}})^2$$

$$F(\underline{x}, \underline{y})$$

Gradiente descendente

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

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

$$\begin{aligned}\varepsilon^2 &= |\underline{y} - \underline{\tilde{y}}|^2 = \sum_{l=1}^m (\gamma_l - \tilde{\gamma}_l)^2 \\ &= \sum_l \varepsilon_l^2\end{aligned}$$

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

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

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

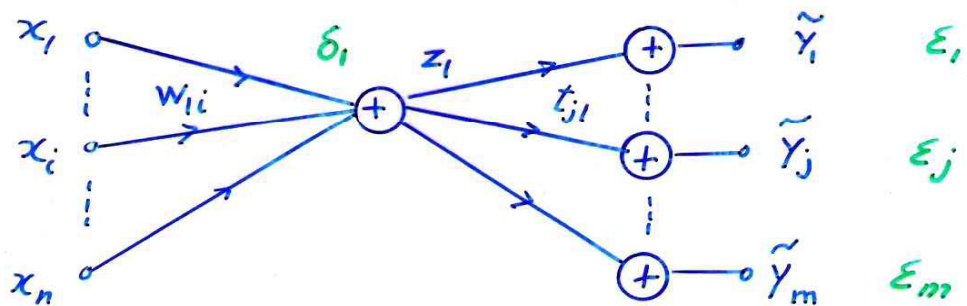
$$\tilde{\gamma}_l = t_{l1} z_1 = t_{l1} \sum_{j=1}^n w_{1j} x_j$$

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

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

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

$$\delta_i = \sum_{l=1}^m \epsilon_l t_{li}$$



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

$$\Delta t_{j1} = 2\alpha z_1 \epsilon_j$$

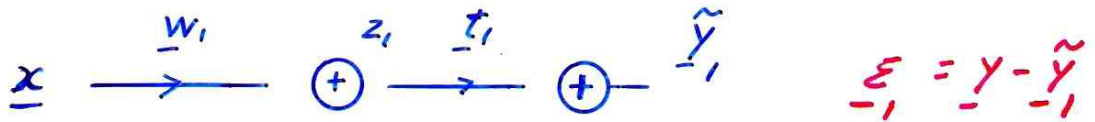
Normalização

$$\tilde{y}_i = z_i t_{i1} = \underline{x}^t \underline{w}_i t_{i1}$$

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

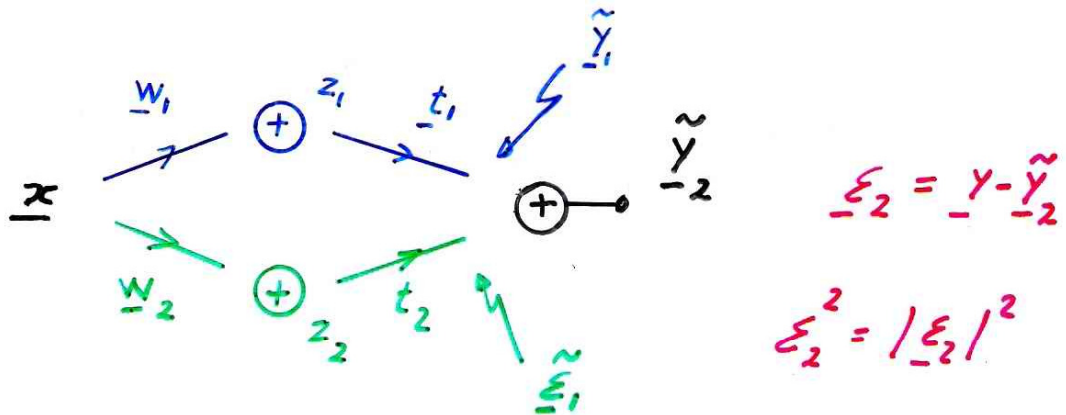
$$\underline{t}_i \longrightarrow |\underline{w}_i| \underline{t}_i$$

Primeira Componente



$$\epsilon_1^2 = |\underline{\epsilon}_1|^2$$

Segunda componente



$$\epsilon_2^2 < \epsilon_1^2$$

Treinamento

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

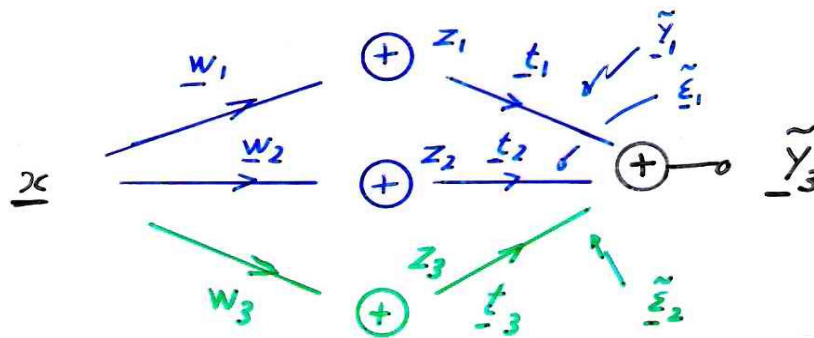
- Treinar $\underline{w}_2$ e $\underline{t}_2$

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 $\underline{t}_3$



$$\tilde{y}_1 \approx y$$

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

$$\tilde{\epsilon}_1 \approx \underline{\epsilon}_1$$

$$\tilde{y}_2 = \tilde{y}_1 + \tilde{\epsilon}_1$$

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

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

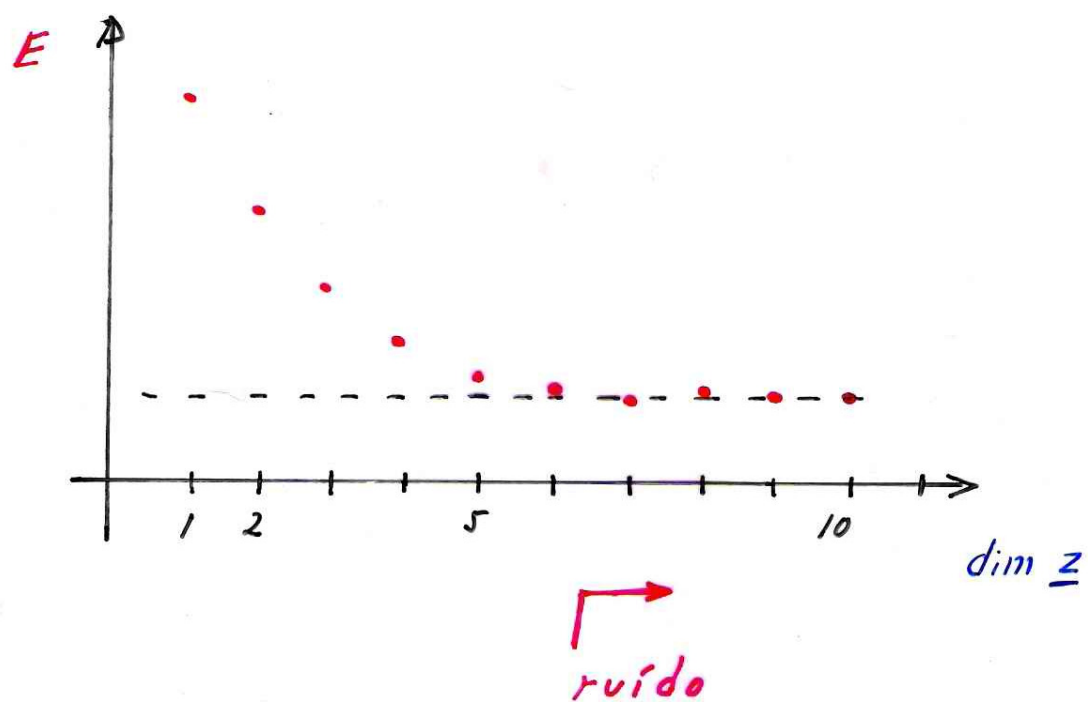
$$y - [y - \underline{\epsilon}_1 + \tilde{\epsilon}_1] =$$

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

$$\tilde{\epsilon}_2 \approx \underline{\epsilon}_2$$

$$\tilde{y}_3 = \tilde{y}_2 + \tilde{\epsilon}_2$$

Erro $E \|\underline{y} - \tilde{\underline{y}}\|^2$ versus
nº de componentes $\dim \underline{z}$



Normaliza $\tilde{z}_0$

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

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

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

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

$\underline{w}_1, \underline{t}_1 \quad / \quad |\underline{\epsilon}_1| \text{ mínimo}$

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

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

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

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

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

$$= \underline{t}_1$$

$$\therefore \alpha = 0$$

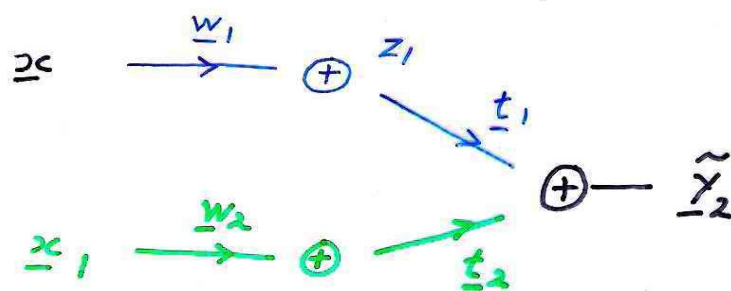
$$\underline{w}_2 = \underline{w}_\perp$$

Garantindo a ortogonalidade

$$\underline{w}_2_{inicial} = \underline{r} - \underline{r}^t \underline{w}_1 \underline{w}_1$$

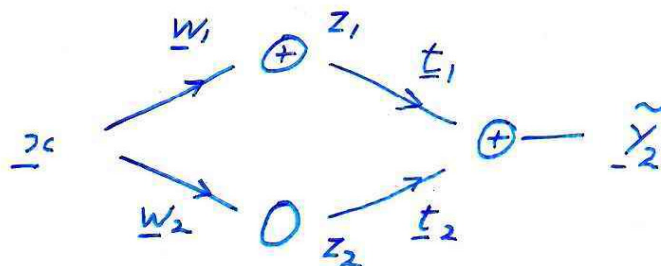
$$\underline{r} = \text{randomico}$$

Treinamento de $\underline{w}_2$



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

Operação

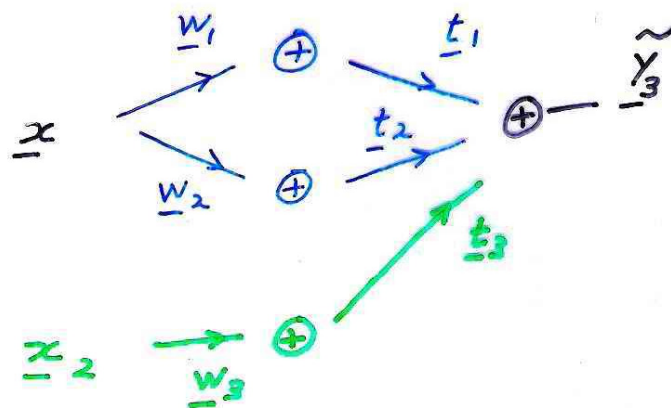


Treino de $\underline{w}_3$

$$\underline{w}_3 \text{ inicial} =$$

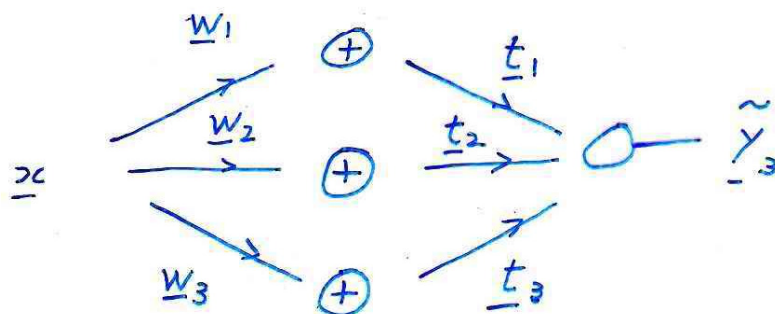
$$\underline{r} - \underline{r}^t \underline{w}_1 \underline{w}_1 - \underline{r}^t \underline{w}_2 \underline{w}_2$$

Treino

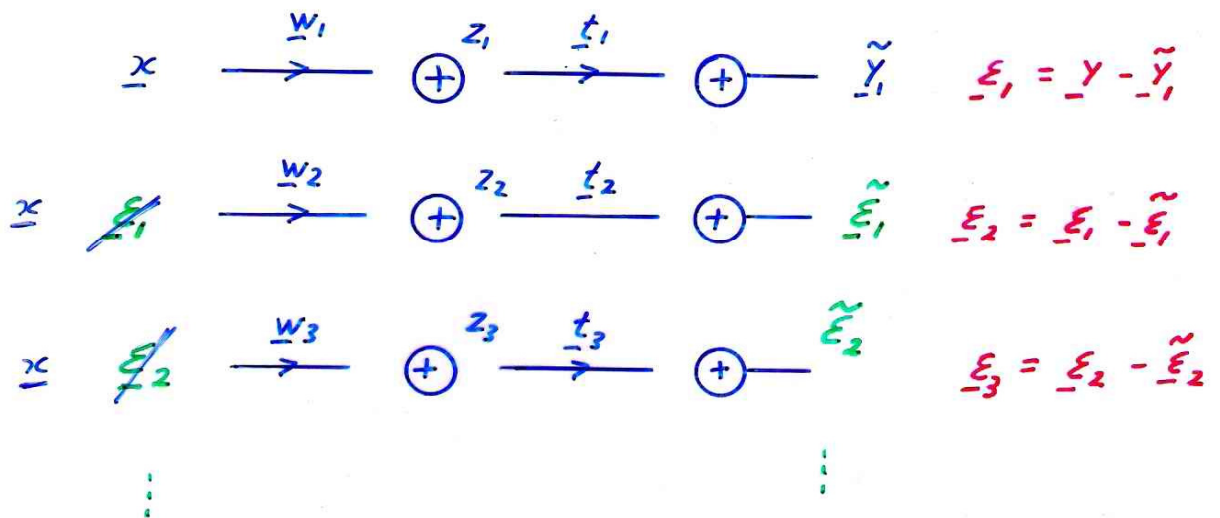


$$\underline{x}_2 = \underline{x}_1 - \underline{x}_1^t \underline{w}_2 \underline{w}_2$$

Operação



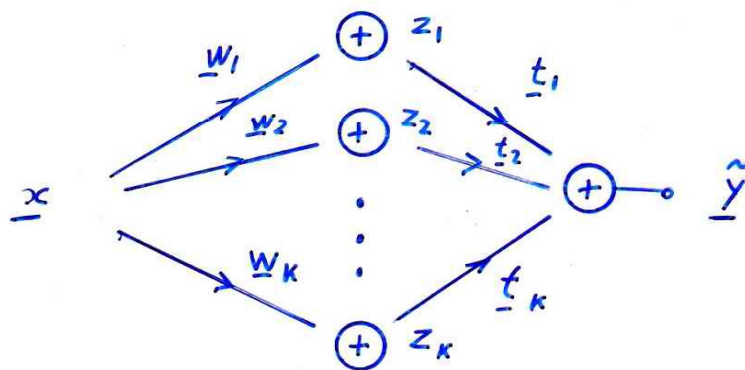
Treinamento alternativo



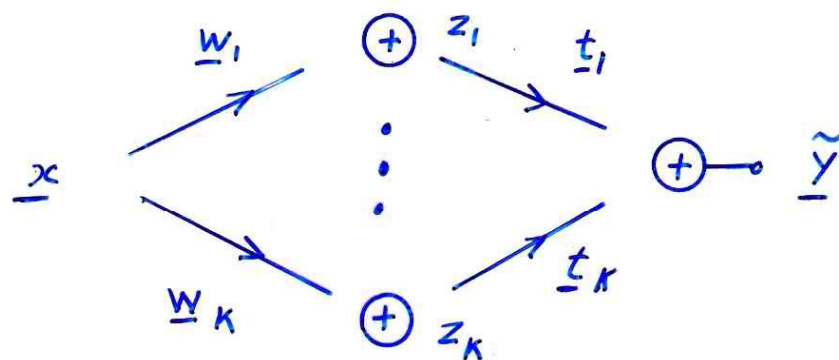
convergência mais rápida
processamento paralelo

Operação

$$\tilde{y} = \tilde{y}_k = \tilde{y}_1 + \underline{\tilde{e}}_1 + \underline{\tilde{e}}_2 + \dots + \underline{\tilde{e}}_{k-1}$$



Outro treinamento alternativo



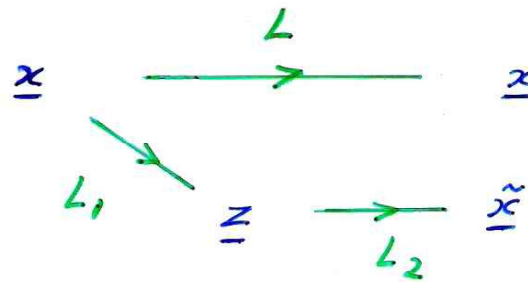
Treinamento simultâneo de
todas as sinapses



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 ditos

Rede: BP auto supervisionada

Outros processos neurais

Oja , Sanger - Oja

Se normalizarmos

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

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

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

$[W]$ é "ortogonal"

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

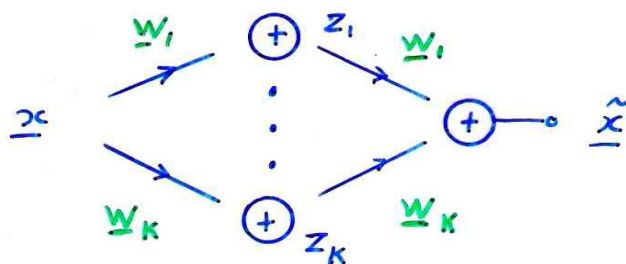
$$[W^{-1}] = [W^t]$$

logo

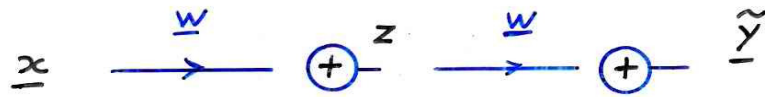
$$[T] = [W^{-1}] = [W^t]$$

ou

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



Treinamento BP - Regra de H2



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

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

$$\mathcal{E}^2 = \sum_i \mathcal{E}_i^2 = \sum_i (y_i - \tilde{y}_i)^2$$

$$\frac{d\mathcal{E}^2}{dw_k} = \sum_i 2\mathcal{E}_i \frac{d\mathcal{E}_i}{dw_k} = -2 \sum_i \mathcal{E}_i \frac{d\tilde{y}_i}{dw_k} =$$

$$= -2 \left[\sum_{i \neq k} \mathcal{E}_i \frac{d\mathcal{E}_i}{dw_k} + \mathcal{E}_k \frac{d\mathcal{E}_k}{dw_k} \right] =$$

$$= -2 \left[\sum_{i \neq k} \mathcal{E}_i w_i x_k + \mathcal{E}_k (z + w_k x_k) \right]$$

$$= -2 \left[\mathcal{E}_k z + x_k \sum_i \mathcal{E}_i w_i \right]$$

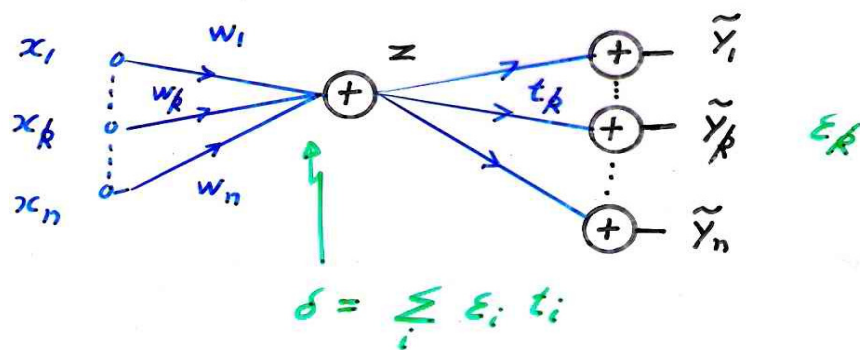
$$\Delta w_k = E \left\{ -\alpha \frac{d\epsilon^2}{dw_k} \right\} =$$

$$= 2\alpha E \left\{ z \epsilon_k + x_k \sum_i \epsilon_i w_i \right\}$$

Obs: $|w_{inicial}| = 1$

Obs:

Sinapses independentes



$$\Delta t_k = 2\alpha E \left\{ z \epsilon_k \right\} \quad (1)$$

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

Sinapses iguais $t_k = w_k$

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

(1) + (2)

Obs:

Mapeamento linear



F quadrática

↳ treinamento



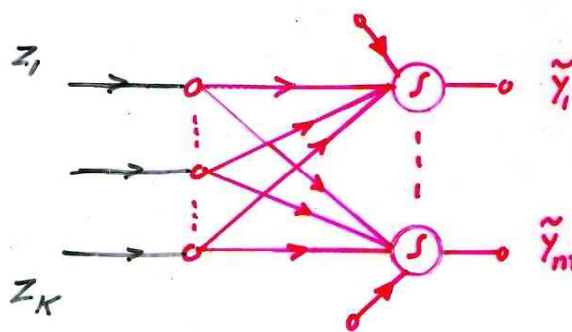
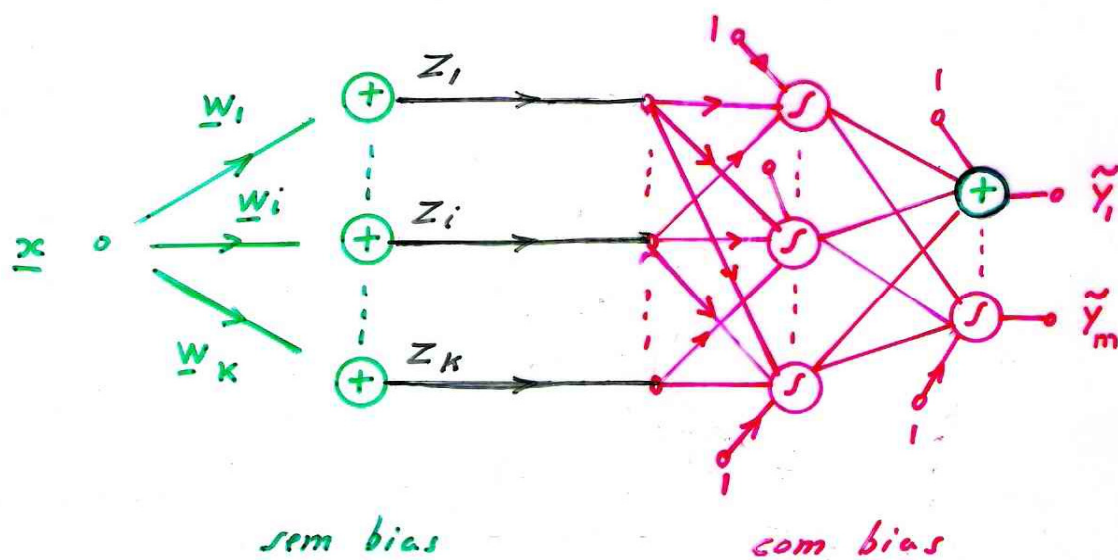
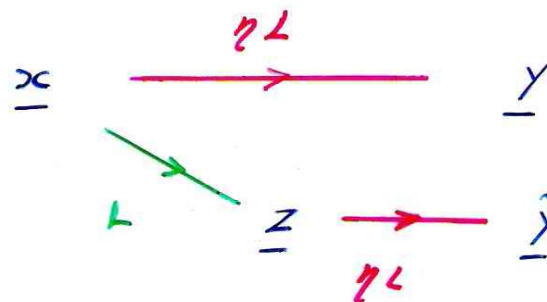
rápido

sem mínimos locais

sem overtraining

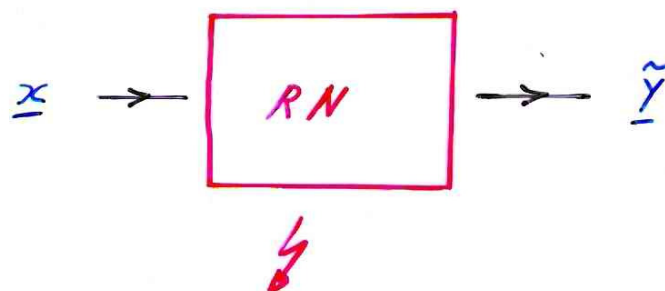
∃ solução analítica

Mapeamento não Linear I



Dimensionando a RN

Mapear SEM comprimir



1 - 2 camadas

φ # neurônios camada intermediária

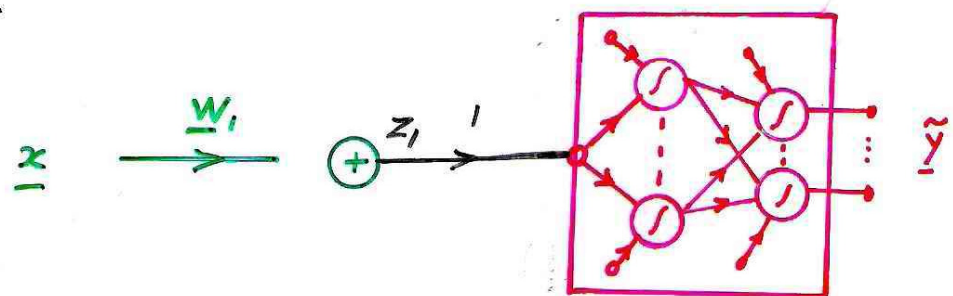
$$\Rightarrow \dim \underline{z} \leq \varphi$$

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

Componentes para

ATUAÇÃO independente

1-



RN com
uma entrada

Treinar todas as sinapses,

$\underline{w}_i$ e RN

normalizar $\underline{w}_i \rightarrow \frac{\underline{w}_i}{|\underline{w}_i|}$

2 - Segunda Componente

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

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

retornar ao passo 1 fazendo

$$\underline{x}_1 \rightarrow \underline{x}$$

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

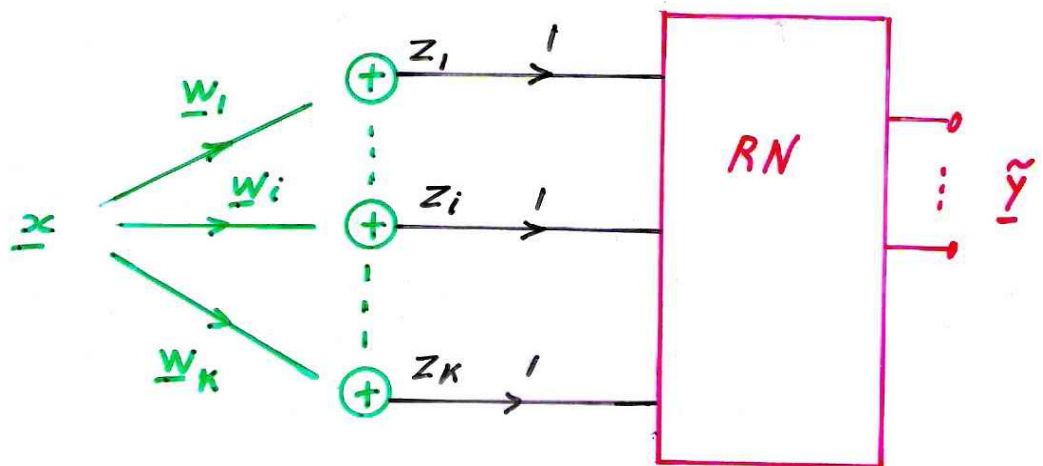
$$\underline{z}_2 \rightarrow \underline{z}_1$$

3 - Terceira Componente

...

Rede para ATUAÇÃO conjunta

- 1 - Determinar $\underline{w}_1, \underline{w}_2, \dots, \underline{w}_k$
- 2 - Treinar RN,
mantendo $\underline{w}_i$ congelados



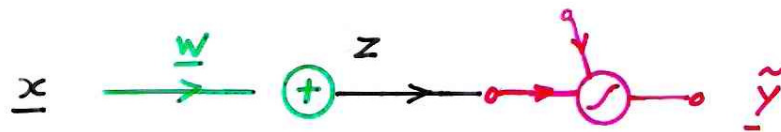
Componentes calculados para

ATUAÇÃO independente

Usada em ATUAÇÃO conjunta

Um caso particular:

o classificador linear



CDA - Canonical discrimination
analysis

Discriminador linear ótimo

Discriminador de Fischer

Componentes para ATUAÇÃO conjunta ou cooperativa

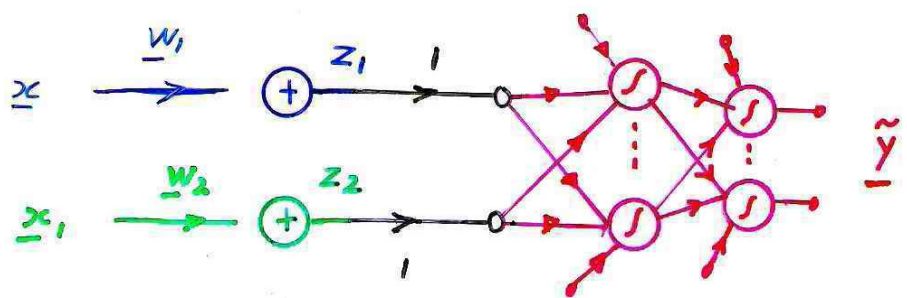
1 - Passo igual ao treinamento para
componentes com atuação independente.

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

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

Aumentar uma entrada à RN

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



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

$$\underline{W}_2_{\text{inicial}} = \underline{r} - \underline{r}^t \underline{W}_1 \underline{W}_1$$

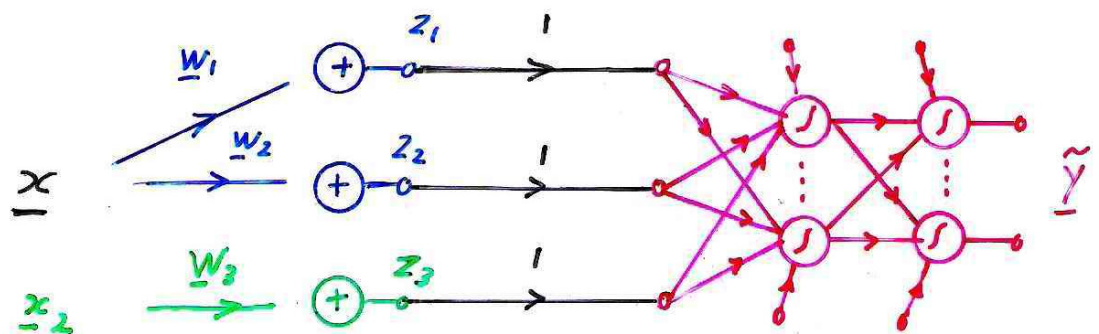
3 - Congelar $\underline{W}_1$ e $\underline{W}_2$ $|\underline{W}_2|=1$

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

Aumentar uma entrada a RN

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

(inclusive as sinapses conectadas
a z_1 e z_2)



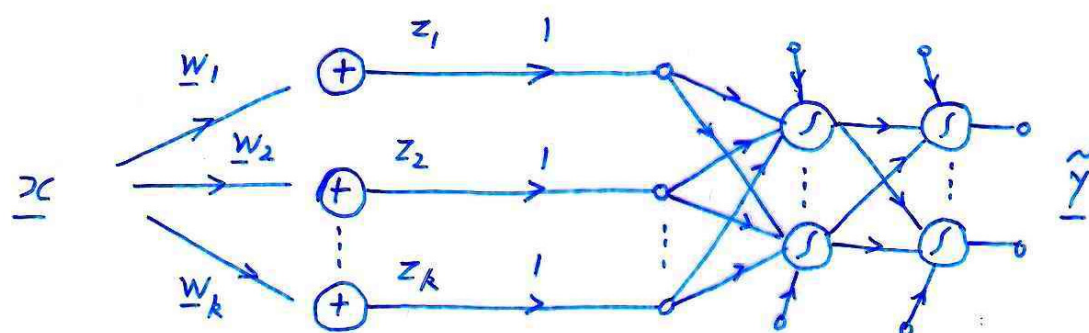
$$\underline{x}_2 = \underline{x}_1 - \underline{x}_1^t \underline{W}_1 \underline{W}_1$$

$$\underline{W}_3_{\text{inicial}} = \underline{I} - \underline{I}^t \underline{W}_1 \underline{W}_1 - \underline{I}^t \underline{W}_2 \underline{W}_2$$

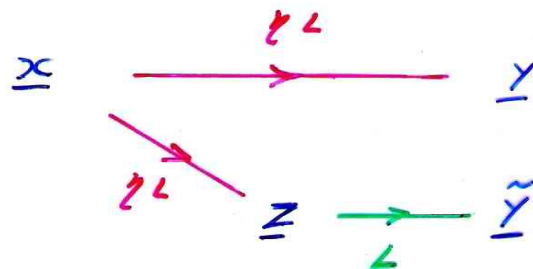
Obs : $\dim z \leq$

neurônios camada intermediária

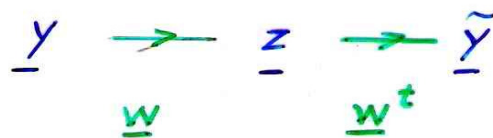
Operação : Última rede treinada



Mapeamento não Linear II



1 - Computar $\underline{y}$

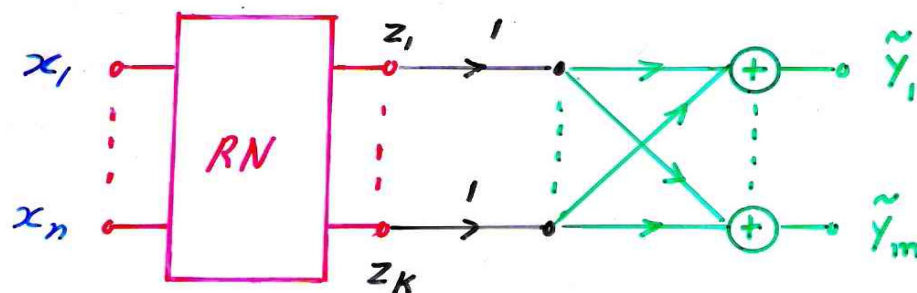


$$\underline{z} = \underline{W} \underline{y} \quad \underline{\tilde{y}} = \underline{W}^t \underline{z}$$

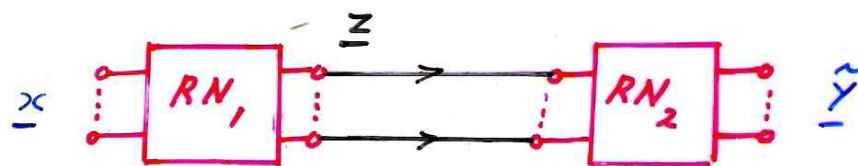
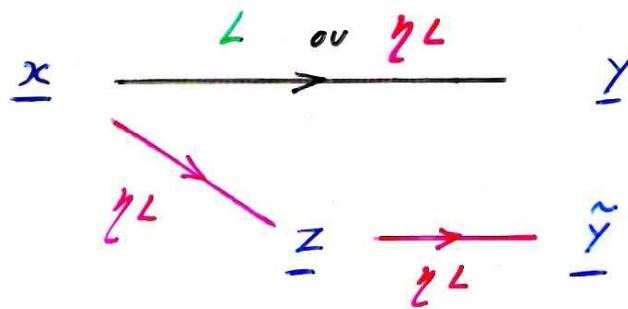
2 - Realizar o mapeamento

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

3 - Rede final



Mapeamento não Linear III



Como RN são mapeadores
universais



teóricamente $\dim \underline{z} = 1$

satisfaz !

Conclusões:

Neste mini-curso discutimos a generalização do conceito de componentes principais objetivando a compressão da informação de entrada $\underline{x}$ de modo a otimizar a saída $\underline{y}$ de um mapeamento qualquer $\underline{x}: \rightarrow \underline{y}$, 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.