

# Detecção de crises epiléticas

***Redes MLNN***

*César Teixeira*

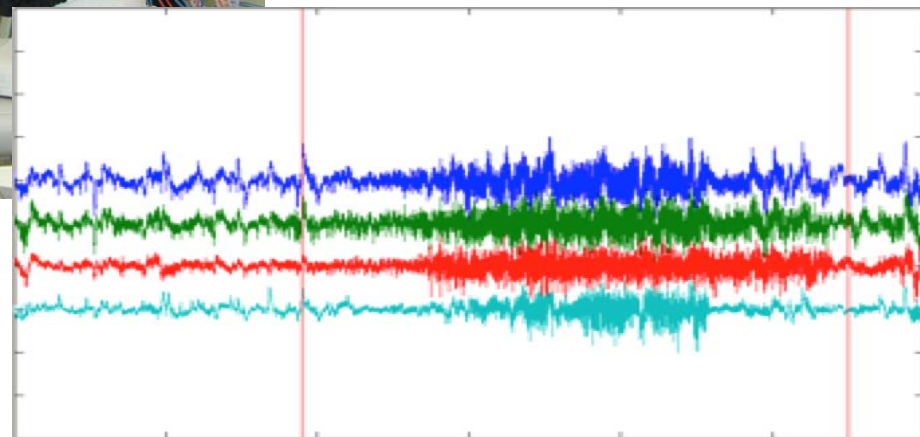
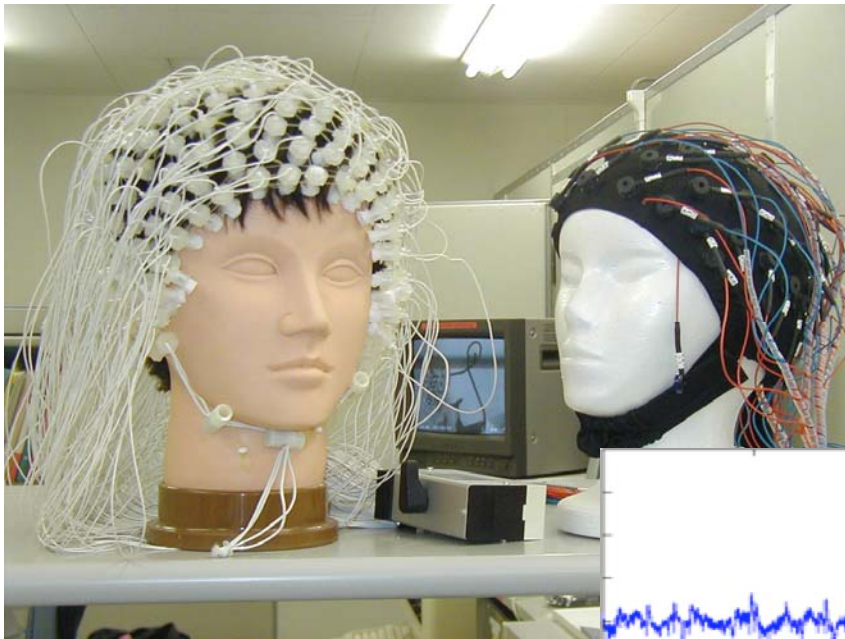
# Sumário

- 1. O problema
- 2. Rede MLNN
- 3. Avaliação dos resultados
- 4. Processamento



# *Epilepsia*

## ■ *EEG - ElectroEncefaloGrama*



# ***Epilepsia***

## ■ ***Classificação***

- *1 -inter-ictal, estado normal*
- *2- pré-ictal, aproxima-se uma crise*
- *3- ictal, está a decorrer uma crise*
- *4- pós-ictal, acabou a crise e vai passar ao estado normal*

- ***0/1 – Não ictal / Ictal***

# Epilepsia

- **Dados: Projecto EPILEPSIAE:** [www.epilepsiae.eu](http://www.epilepsiae.eu)

- *Segmentos de EEG de duração de 2s com 50% de sobreposição entre segmentos*

- *Para cada segundo*

- **29 características** do espectro de frequência do sinal de EEG

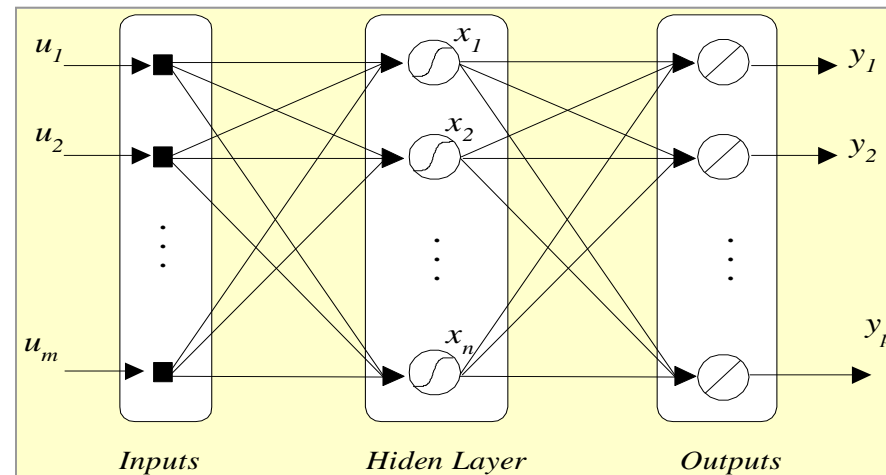


# Sumário

- 1. O problema
- 2. Rede MLNN
  - Estrutura
  - Treino
- 3. Avaliação Resultados
- 4. Processamento

## 2.1 Estrutura

- **Restringimos o estudo a (??):**
  - MLNN – multilayer neural network (**multicamada**)
  - FF – FeedForward (**de avanço**)
  - BP – Backpropagation (**retropapagação**)



## 2.1 Estrutura

- O número de *entradas e de saídas* é usualmente fixado pelo problema a resolver.
  - 29 entradas
  - 2 saídas (*eventualmente uma*)

- **Matlab:**

- `>> newfeedforward`

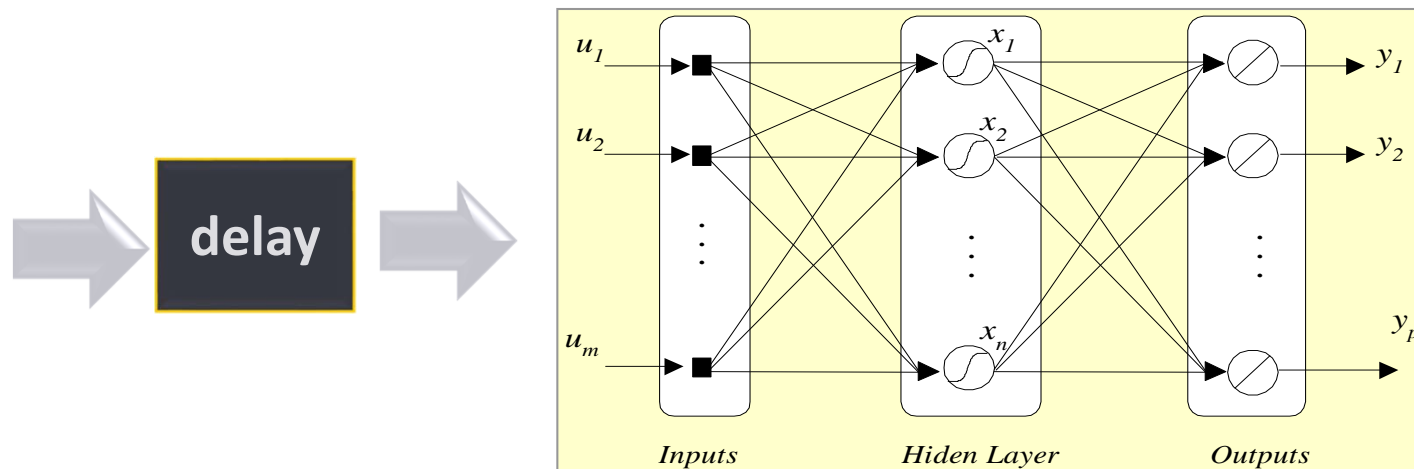
- **E na camada interna ?**

- Número de neurónios ?



## 2.1 Estrutura

- *Uso de uma rede RBF?*
- *Introdução de atrasos nas entradas ?*



- *Matlab*

■ `net=layrecnet(layerDelays,hiddenSizes,trainFcn)`

## 2.2 Treino

- *Backpropagation* = gradiente
  - Método de primeira ordem (primeira derivada)

$$\Theta(k+1) = \Theta(k) - \alpha \mathbf{e} \mathbf{z}^T$$

- Levenberg-Marquardt
  - Método segunda ordem

$$H \approx H_a = \left( \frac{\partial F}{\partial \mathbf{w}} \right)^T \frac{\partial F}{\partial \mathbf{w}} + \gamma I = G^T G + \gamma I$$

# Sumário

- 1. O problema
- 2. Rede MLNN
- 3 Avaliação Resultados
  - Métricas
  - Treino/teste/validação
- 4 Processamento

## 3.1 Métricas

### SENSIBILIDADE

$$SE = \frac{TP}{TP + FN}$$

### ESPECIFICIDADE

$$SP = \frac{TN}{TN + FP}$$

	<i>Epilepsia</i> (1)	<i>NOT Epilepsia</i> (0)
<i>Detect</i> (1)	TP	FP
<i>NOT Detect</i> (0)	FN	TN
	SE	SP

*TP – Verdadeiros Positivos*

*TN – verdadeiros negativos*

*FP – Falsos Positivos*

*FN – Falsos negativos*

## 3.2 Treino/validação/teste

- Dados de treino / validação / teste
  - *A rede deve ser capaz de generalizar !!*
  - Dados de treino, validação e teste

## 3.2 Treino/validação/teste

### ■ Dados de treino / validação

- Devem existir crises quer nos dados de treino, quer nos de teste, e nos de validação sempre que possível
- Por exemplo para um paciente com 9 crises
  - Considerar para treino as primeiras 6 crises (70%)
  - As restantes 3 para teste (30%)

### ■ *Matlab*

- Divisão do conjunto de dados em treino, validação e teste
- `>> divideind`

# Sumário

- 1. O problema
- 2. Rede MLNN
- 3. Avaliação Resultados
- 4 Processamento
  - Pré-processamento
    - Normalização
    - Redução dimensionalidade
  - Pós-processamento

## 4.1 Normalização

- ***Gama dos dados de entrada e de saída***
  - De preferência os dados devem ser restringidos=normalizados a uma determinada gama
  - Exemplos
    - Valor [mínimo, máximo] = [0 , 1]
    - ***Média nula, desvio padrão = 1***



## 4.2 Redução da dimensionalidade

- *E no caso do numero de parâmetros (=entradas) ser elevado?*

- ***Seleccção de parâmetros mais relevantes***

- Redução da dimensionalidade

- Limitação:

- Pode conduzir à perda da interpretabilidade dos dados de entrada

## 4.2 Redução da dimensionalidade

### ■ *Várias técnicas*

- 1. Análise de correlação
- 2. Pesquisa
  - para a frente
  - para trás
- 3. Análise de Componentes Principais

## 4.2 Redução da dimensionalidade

### ■ 1. *Análise de Correlação*

#### ■ *Seleccção de parâmetros (entradas)*

- *Correlação*
- *Informação mutua*

```
■ >> corrcoef( parâmetro, classificação)
```

#### ■ *Valor na gama [-1, 1]*

#### ■ *Se -1 ou 1 perfeitamente correlacionada*

- *(importante)*

#### ■ *Se 0 completamente descorrelacionado*

- *(não interessa)*

## 4.2 Redução da dimensionalidade

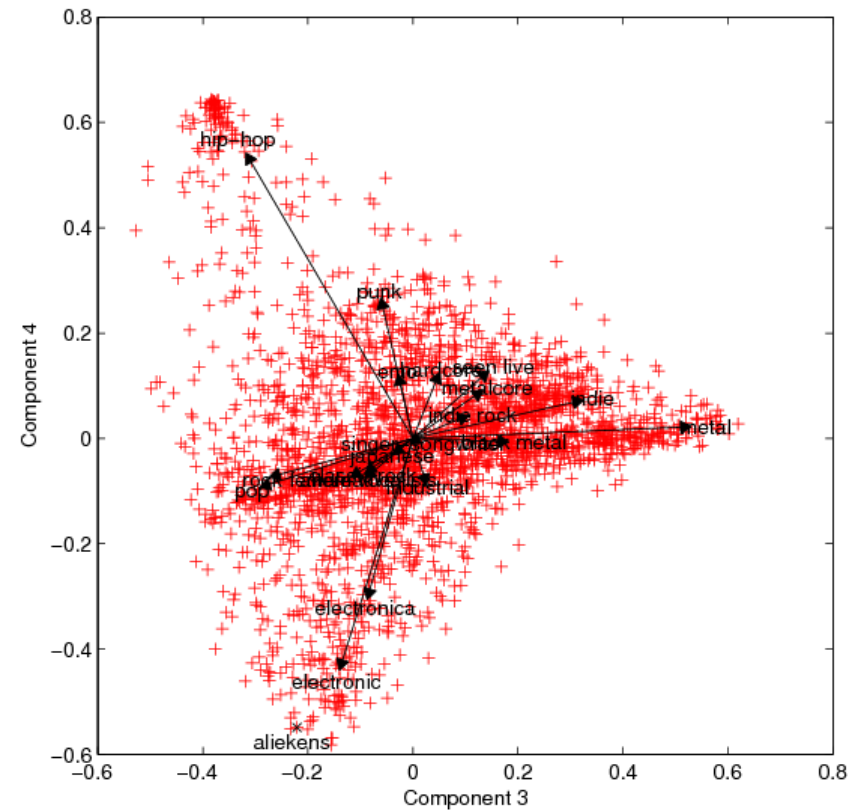
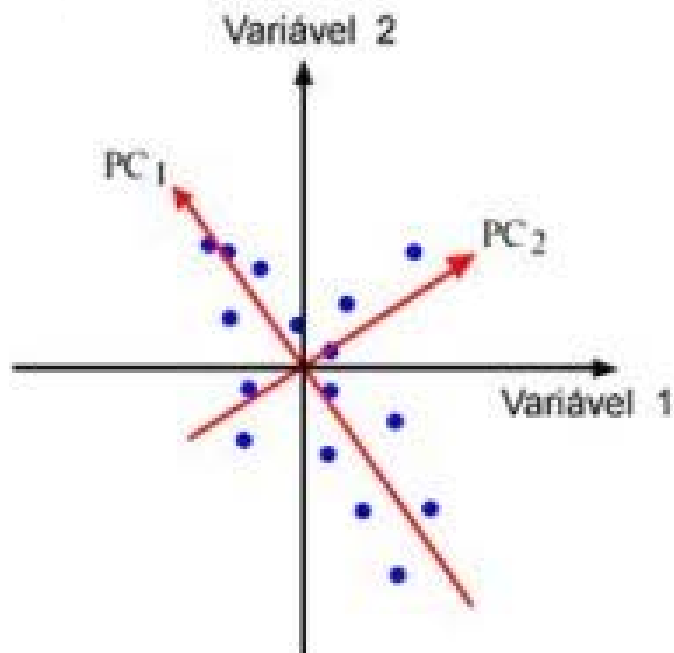
- **2.1 Pesquisa para Frente (Forward selection)**
- A pesquisa é iniciada sem qualquer parâmetro sendo estes adicionados um a um
  - Cada parâmetro é adicionado isoladamente e o resultado obtido é avaliado segundo um determinado critério (geralmente erro)
  - O parâmetro que conduz ao melhor resultado é adicionado

## 4.2 Redução da dimensionalidade

- **2.2 Pesquisa para trás (*Backward elimination*)**
- O processo é iniciado com todo o conjunto de todos os parâmetros, eliminando um deles de cada vez
- O processo é análogo ao *Forward selection*

## 4.2 Redução da dimensionalidade

### ■ 3. *Análise de componentes principais*



## 4.2 Redução da dimensionalidade

- *As componentes principais (PC) são direcções independentes*
  - As PCs ordenadas de acordo com quantidade da variância dos dados originais que elas contêm (ordem decrescente)
  - A primeira PC “explica” (contém) a maior percentagem da variabilidade do conjunto de dados original
  - A segunda PC define a próxima maior parte, e assim sucessivamente
- Em geral, apenas algumas das primeiras PCs são responsáveis pela maior parte da variabilidade do conjunto de dados
- O restante das PCs tem uma contribuição insignificante
- ***Devem considerar-se apenas as componentes principais***

## 4.2 Redução da dimensionalidade

### ■ *Matlab*

- $[PC, \textit{reducedData}, EIG] = \text{princomp}(X);$

- PC = Componentes principais

- *reducedData* – transformação dos dados X

- EIG – valores próprios



## 4.2 Redução da dimensionalidade

### ■ *EIG - Valores próprios*

- *Por ordem decrescente*
- *Logo, devem-se escolher os primeiros valores próprios até um dado valor*
  - *Ex: dimensão 5, escolha dos três primeiros*
  - **[ 0.8 0.4 0.1 0.001 0.0002 0.0004 ]**
- *As bases principais/características correspondem aos vectores próprios associados*
  - *Perde-se interpretabilidade !!!*

# Sumário

- 1. O problema
- 2. Rede MLNN
- 3. Avaliação Resultados
- 4 Processamento
  - Pré-processamento
  - **Pós-processamento**

# Conclusões

- *Especificar uma rede para a detecção de crises de epilepsia*
  - *Estrutura ?*
  - *Treino/validação ?*
  - *Pré e pós processamento (redução features / classificação selectiva)*
  - *Métricas: SE e SP*
  
- *Cada grupo:*
  - *Dados de dois pacientes (29 features por segundo)*
  - *Se um grupo quiser testar em mais de dois pacientes, poderá fazê-lo, basta solicitar os dados.*

## 4.3 Pós-processamento

- *Numa abordagem inicial, deve classificar ponto a ponto, isto é, cada ponto deve ser associado a uma das 2 classes: **crise/não crise***
- *Numa segunda abordagem:*
  - *Só se deteta uma crise se houver por exemplo 10 pontos consecutivos na classe 1 (ictal),*
  - *ou pelo menos 5 dos últimos 10 são da classe 1.*
  - *Se houver um pequeno número de pontos da classe 1 isolados, pode-se considerar como não crise, (devidos a outros fatores).*