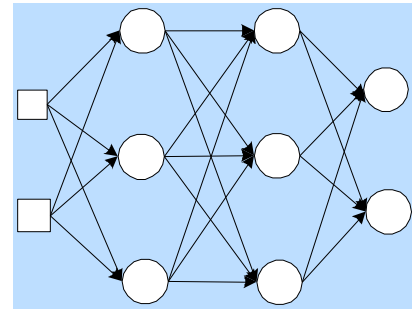


Redes Multi-camada

Classificação de ruído em biosinais



Computação Adaptativa

Aulas práticas



*Departamento de Engenharia Informática
Faculdade de Ciências e Tecnologia
Universidade de Coimbra*

Jorge Henriques

Novembro, 2013

Índice

Resumo

1. Redes neuronais: matlab

- 1.1 Definição do conjunto de dados
- 1.2 Definição da Estrutura
- 1.3 Especificação dos parâmetros relativos ao treino
- 1.4 Treino: cálculo dos pesos e desvios
- 1.5 Verificação
- 1.6 Validação: capacidade de generalização

2 Problema a resolver

- 2.1 Definição
 - Sinal PPG – Características*
- 2.2 Dados para treino e validação
- 2.3 Dados para “concurso”

3 Trabalho

- 3.1 Dados de entrada / características
- 3.3 Classificador: rede neuronal feedforward
- 3.4 Validação: SE e SP
- 3.2 Entrega
- 3.3 Concurso

Resumo

Pretende-se neste trabalho estudar a viabilidade de utilização de redes neuronais em problemas de classificação.

Em concreto, serão analisadas as capacidades das redes **multicamada/feedforward** em detectar a presença de ruído em sinais biológicos (**PPG-fotopletismograma**), conhecendo algumas das suas características (amplitudes, variações, ...).

Nota:

Agradece-se ao aluno Ricardo Couceiro a disponibilização e tratamento dos dados!

1. Redes neuronais: matlab

A definição de uma rede neuronal, assim como o seu treino, é possível de implementar facilmente recorrendo à custa da **toolbox Neural Networks**, pertencente ao Matlab.

1.1 Definição do conjunto de dados

Assume-se a existência de um conjunto de dados de entrada (**P**) e correspondentes saídas desejadas (**T**) para o treino, eventualmente normalizadas, de dimensões:



P: (**R**, **Q**) **T**: (**S**, **Q**)

R: número de entradas

S: número de saídas

Q: número de padrões de treino

Exemplo de definição em Matlab do conjunto de dados de treino **{P, T}**:

```
%  DADOS 
P= [
    88 86 83 80 79 67 99 78
    66 67 60 78 77 67 62 59
    89 78 76 80 82 89 81 82];
```

```

T= [ 12 15 16 17 18 13 11 13];
%----- normalização
P= 0.01*P;
T= 0.05*T;
minx=0;
maxx=1;
%----- dimensões
[R,Q] =size(P) ;
[S,Q] =size(T) ;
% DADOS

```

1.2 Definição da Estrutura

Assume-se, neste caso, uma rede *feedforward* (**net**), a ser criada pela função (**newff**):

- uma camada interna, com **M** neurónios
- funções de activação $\sigma(\cdot)$ -tangente hiperbólica (**tansig**) na camada interna
- funções de activação lineares (**purelin**) na de saída
- existência de desvio na camada interna
- sem desvio na camada de saída.

As matrizes de ligação (**W1** e **W2**) e o desvio na camada interna (**B1**) são inicializados aleatoriamente, sendo utilizado como método de treino o Backpropagation (**traingd**).

Exemplo de definição da rede em Matlab (M=6 neurónios na camada interna):

```

% ESTRUTURA
M= 6;
net = newff( P, T,[M S],{'tansig','purelin'},'traingd' );
% ESTRUTURA

```

1.3 Especificação dos parâmetros relativos ao treino

É possível especificar diversos parâmetros relacionados com o treino da rede. Entre estes, destaca-se o número máximo de épocas de treino, de quanto em quantas iterações é visualizado o resultado do treino, o objectivo e o critério a ser minimizado. No quadro seguinte mostra-se ainda como é possível atribuir à rede valores iniciais para os pesos e desvios.

[illegible]

1.4 Treino: cálculo dos pesos e desvios

Para efectuar o treino da rede recorre-se à função (**train**). Os pesos e desvio resultantes do processo de treino obtidos podem ser acedidos, como se mostra a seguir:

```
% [ ] TREINO [ ]
```

```
[net,tr] = train(net,P,T);
```

```
W1=net.IW{1,1};  
W2=net.LW{2,1};  
B1=net.b{1,1};
```

```
% [ ] TREINO [ ]
```

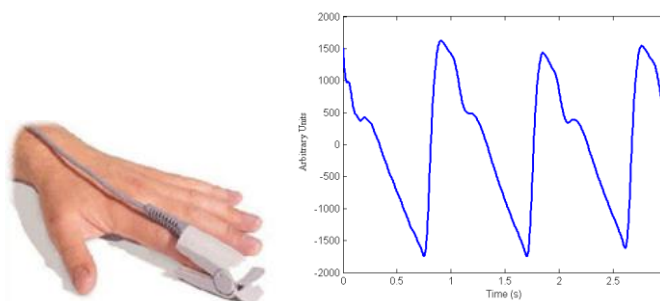
1.5 Verificação

Pode-se depois verificar o sucesso/insucesso do treino, isto até que ponto os valores proporcionados pela rede se aproximam dos valores desejados (**T**). Para o cálculo da saída da rede (**A**) o MatLab disponibiliza a função (**sim**).

2 Problema a resolver

2.1 Definição

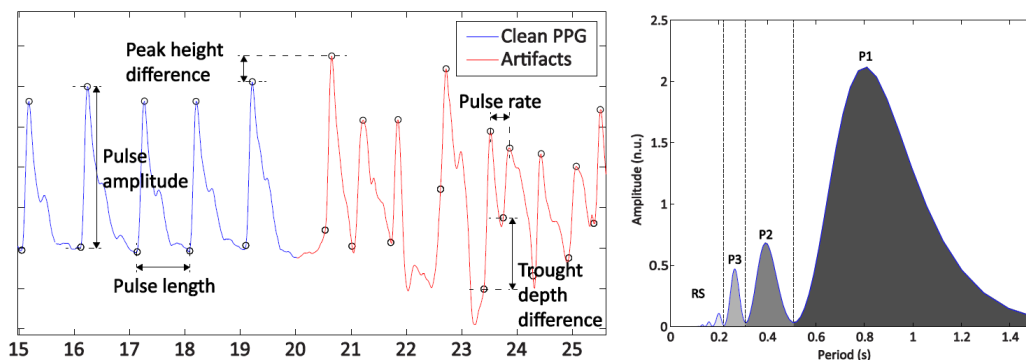
Pretende-se, com base na informação recolhida sobre sinal (extraída batimento a batimento), implementar uma rede neuronal capaz de classificar cada batimento como normal ou ruído. Se o batimento for considerado normal é numa fase posterior processado; cada seja considerado como ruído é descartado.



Para tal utiliza-se uma rede **feedforward**, com pelo menos uma camada interna, função de activação tangente hiperbólica/sigmoidal e linear respectivamente na(s) camada(s) interna(s) e de saída.

Sinal PPG – Características

Admita que são extraídas/calculadas 26 características em cada batimento (para mais detalhes ver artigos fornecidos)



2.2 Dados para treino e validação

Consideram-se os seguintes dados para a aprendizagem e validação das redes neuronais (contidos no ficheiro dataset1.mat) :

- **X1**: Matriz de dimensão (12331x26) - são disponibilizados **Q=12331** padrões de treino, neste caso os valores correspondentes às características (**R=26**) extraídas de 12331 batimentos.
- **T1**: Matriz de dimensão (12331x1) – correspondendo à classificação em questão, valores possíveis {0,1}, respectivamente normal ou ruído.
- **caracteristicas_names** – definição das características. Note-se que apesar de disponibilizado não é necessário o conhecimento destes parâmetros.

2.3 Dados para “concurso”

Além dos dados disponibilizados, em que se conhece a saída desejada, é disponibilizado outro conjunto de dados (contidos no ficheiro dataset2.mat)

- **X2**: Matriz de dimensão (602x26) - correspondentes às características de 602 batimentos.

No entanto para este conjunto de dados não se conhecem as saídas desejadas (**T2**), isto é a classificação normal/ruído. Assim sendo é impossível saber se a rede é capaz de funcionar adequadamente com este conjunto de dados.

As respectivas saídas desejadas (**T2.dat**, de dimensão 602x1) serão apenas disponibilizadas após a entrega dos trabalhos práticos.

Para o grupo que melhor resultados alcançar neste conjunto de dados haverá uma bonificação de 20%.

3 Trabalho

3.1 Dados de entrada / características

1. Terá interesse normalizar os dados?
2. Será preferível/possível considerar todas as características ou apenas parte delas?

Explore os seguintes métodos de redução de dimensionalidade possíveis de utilizar:

- Cálculo de correlação → Matlab: *corrcoef(x,y)*
- Análise de componentes principais → Matlab: *princomp(X)*

3.3 Classificador: rede neuronal feedforward

Recorra ao Matlab para definir/treinar a rede neuronal (secção 2). Entre outros, tenha em atenção:

- Dados de treino / validação
- Número de neurónios na camada interna
- Método de treino (gradiente, Levenberg-Marquardt, ...)

3.4 Validação: SE e SP

Deve ter em conta tanto o erro de treino como validação.

Como medidas de validação considere a sensibilidade (SE) e a especificidade (SP).

3.2 Entrega

O trabalho e respectivo relatório deverão ser entregues até **03 de Dezembro 2013**. Deve ser entregue:

- **Por mail:** Ficheiro zipado com todos os ficheiros necessários
 - Nome: AnacletoCaroço_DesidérioAbelha.zip (nomes dos elementos do grupo)
 - Deve haver um ficheiro principal (executável) de nome **tp2.m**
- **Em papel:** Relatório **sucinto** relativo às conclusões principais.

3.3 Concurso

Para os grupos que quiserem participar no concurso (bonificação 20%) deve ser entregue (por mail) a classificação às entradas T2.

Espera-se um vector de 602 elementos, valores possíveis $\{0,1\}$.

Boa classificação!