Trabalho Prático 3

Previsão de Crises Epilépticas

Computação Adaptativa (MEI)

CNSD (MIEB)

Alexandre Sousa (MIEB) - 2004107017

Marco Simões (MEI) - 2006125287

Sérgio Santos (MEI) - 2006125508

Departamento de Engenharia Informática

Faculdade de Ciências e Tecnologia

Universidade de Coimbra

Novembro, 2009

# Introdução

O objectivo deste Trabalho Prático consiste no desenvolvimento de um programa de previsão de crises epilépticas, através de redes neuronais. Sendo-nos fornecidos dados para teste e treino das redes, e utilizando a Neural Networks Toolbox do Matlab, resta-nos desenvolver uma plataforma que permita a experimentação das diversas abordagens possíveis. Essa plataforma deverá conter uma interface gráfica que permita a realização dessas mesmas experiências, e visualização dos resultados respectivos.

Adicionalmente, como os dados fornecidos incluem um grande número de características (320) recolhidas de pacientes, será necessário verificar qual o número de características mais eficaz e quais características em particular. A abordagem a este desafio é deixada em aberto, sendo possível usar várias técnicas de selecção de características.

Neste documento serão apresentados os principais detalhes relativos à implementação da plataforma, como executá-la, que testes foram realizados, quais os seus resultados e que conclusões podemos depreender destes.

# Desenvolvimento

A plataforma desenvolvida é composta por diversas secções, que serão descritas em seguida.

### Interface

Foi desenvolvida uma interface gráfica para permitir a experimentação das várias configurações possíveis. A interface foi desenvolvida através da ferramenta *guide*, e possui a capacidade de definir praticamente todas as propriedades variáveis do problema consideradas. Por exemplo, é possível definir o tipo de rede utilizada, qual o número de características escolhidas e como estas são escolhidas e qual dos ficheiros de dados são usados para treino e para teste.

Para além disso permite executar o treino da rede especificada e o posterior teste. Essas acções irão executar as funções **trainNetwork** e **testNetwork**, respectivamente. Os resultados são apresentados nessa mesma interface, incluindo um gráfico de comparação entre as crises previstas pela aplicação e as anotadas nos dados fornecidos.

### Selecção de características

Foram implementadas várias técnicas de selecção de características, com o intuito de avaliar as melhores características a serem usadas, segundo diversos parâmetros. Todas as técnicas estão condicionadas pelo número de entradas/características que se pretende fornecer à rede neuronal criada. As técnicas poderão ser utilizadas simultaneamente, pela ordem em que serão apresentadas a seguir.

Ratio short/long wave

As características estão representadas sempre em dois formatos: short e long wave. Numa tentativa de redução do número de características, pensámos em utilizar o ratio entre estas duas ondas, na esperança de que a relação entre estas possa ser mais significante que os seus valores absolutos.

Agrupamento

Tendo em conta que existem várias características muito semelhantes entre si, uma opção para diminuir o número de características será agrupar essas características semelhantes numa só. As características que resultam desses agrupamentos contêm a média das características agrupadas. A técnica utilizada foi a k-means clustering, já implementada pela toolbox de análise estatística do Matlab.

Correlação

Tendo em conta que existe muita informação partilhada por todas as características, uma possível solução seria retirar as características que partilham mais informação com todas as outras, uma vez que a maior parte dessa informação estará já representada nas restantes características. Assim, calculamos o somatório dos índices de correlação de cada característica com todas as outras e escolhemos apenas as com um total inferior, ou seja, mais distintas.

Informação Mútua

Finalmente, uma das propriedades mais relevantes é a relação de uma determinada característica com o resultado esperado da rede. Essa relação pode ser avaliada através da informação mútua entre ambos. Logo, podemos seleccionar como características mais relevantes as que maior *informação mútua* têm com o *target*.

### Normalização

Uma das opções que implementámos tem a ver com a normalização dos dados. Os dados de input são bastante esparsos, obtendo valores de 0 a 1e10. Desta forma, decidimos atacar também esta característica, permitindo na interface ao utilizador escolher se deseja ou não normalizar estes dados, isto é, colocá-los no intervalo [0 1].

### Formato de output

A abordagem inicial considerava como output da rede quatro neurónios. Nesta abordagem, a rede treina-se para, no fundo, detectar quatro características diferentes. Para simplificar o problema, uma vez que o objectivo é detectar apenas a fase pré-ictal, criámos a possibilidade de escolher um formato de output de apenas um neurónio, que tenderá para 0 se não for a fase pré-ictal ou para 1 se for.

Desta forma, reduzimos um problema de classificação de 4 características para duas. Isto obrigou-nos também a formatar os targets, passando de valores de 1 a 4 para uma estrutura mais binária, apenas composta por 0s e 1s.

### Cálculo de sensibilidade e especificidade

Para concluir, falta referirmos como é realizado o cálculo das percentagens de sensibilidade e especificidade. Estas são calculadas apenas entre as fases pré-ictais previstas e “reais”, ou seja, apenas sobre a classe 2. Os valores de positivos verdadeiros e falsos, e negativos verdadeiros e falsos, são contabilizados ponto a ponto, sendo o valor de sensibilidade e especificidade calculados com esses mesmos totais, segundo as fórmulas:





# Execução

Para abrir a interface gráfica da aplicação, e poder configurá-la para a criação, treino e teste de redes neuronais para a previsão de crises epilépticas, basta executar a função **interface**. Deverá surgir uma janela semelhante a esta:

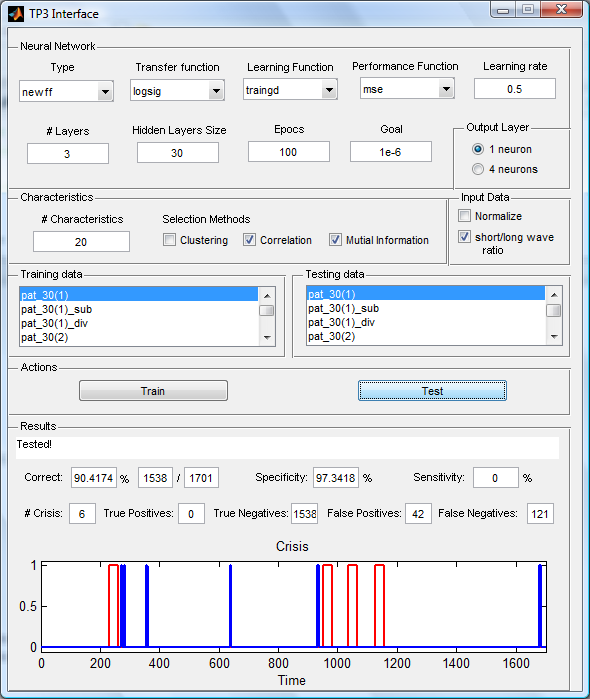


Figura - Interface gráfica da aplicação

Foi também desenvolvida uma função que permite executar um grande número de treinos e testes de seguida, exportando os resultados para um ficheiro *csv*: a função **batchTestNetworks**. A configuração dos testes a serem executados encontra-se dentro da própria função. Esta só deverá ser executada para a realização de estudos mais elaborados, como é o caso do que foi feito durante este projecto, e será apresentado na secção seguinte.

# Resultados

Este trabalho teve uma grande componente de testes, uma vez que era bastante orientado aos resultados.

Desta forma, alem do grande número de testes de aceitação, seguem em anexo os resultados dos seguintes testes:

* *CharacteristicsStats.xlsx* – Este ficheiro contém um elevado número de testes, efectuados sobre a rede *newff*, com a função de aprendizagem *trainlm*, uma camada escondida de tamanho 30 neurónios. Neste ficheiro podemos avaliar os resultados da aplicação dos diferentes métodos de selecção de características, bem como o número de características em si, normalização dos dados e formato de output.
* *NetworksAndLearnings.xlsx* – Este ficheiro contém os testes efectuados para verificar qual a melhor rede e qual a melhor topologia associada, isto é, número de camadas, learning functions, etc.
* *LongShortRatio.xlsx* – Este ficheiro contém os testes efectuados para verificar a aplicação da relação entre a onda longa e comprida em vez das próprias ondas como características.

A elevada dimensão dos dados de testes não nos permite incorporá-los no relatório. No entanto, estes seguem em anexo, sendo sugerido ao leitor que vá fazendo o paralelismo com estes à medida que avança no capítulo das conclusões.

# Conclusões

Através de todos os resultados que obtivemos, dos testes descritos na secção anterior, pudemos obter várias conclusões relativas à problemática do projecto. Separámo-las nas referentes ao tipo e configurações de redes neuronais, e nas referentes ao processo de selecção das características a fornecer como entrada à rede.

### Estrutura de rede neuronal

Neste trabalho foram testadas inúmeras redes com inúmeras topologias. No entanto, a maioria destas acabou por se revelar más para a classificação desejada. A maioria fazia *over-fitting* dos dados, o que leva a resultados díspares, com valores de sensibilidade e especificidade completamente opostos.

Ao fim de muitos testes concluímos que a única rede neuronal que conseguia uma relação mais satisfatória entre estas duas componentes era a “feed forward”, apresentada na toolbox do matlab como ***newff***. No entanto, isto apenas se verifica com a função de aprendizagem ***trainlm***. Para que os resultados sejam significativos, esta deve apresentar pelo menos uma camada escondida, assumindo valores semelhantes com uma ou duas camadas escondidas, sobressaindo um pouco a especificidade usando apenas uma. Tal pode ser consultado no ficheiro *NetworksAndLearnings.xlsx*.

O número de neurónios utilizados nas camadas escondidas também é um ponto importante, do qual concluímos que este deve ser ligeiramente maior que o número de inputs. No ficheiro NetworksAndLearnings.xlsx podemos ver os resultados da aplicação de 30 neurónios e de 100 na mesma rede, para 20 características de entrada, revelando-se a primeira uma melhor estrutura.

As funções de transferência desempenham também, tendo o logsig e o tansig melhores valores em relação aos obtidos por funções como purelin ou hardlim. No entanto, nas estruturas newff, a camada de saída é utiliza sempre purelin, sendo o logsig aplicado apenas às camadas escondidas.

Existem ainda outros factores com influência nos resultados, sendo estes a normalização dos dados, a estrutura do output e a simplificação das características através do short/long wave ratio.

No caso da normalização, o ficheiro *CharacteristicsStats.xlsx* revela os seguintes valores médios:

|  |  |  |
| --- | --- | --- |
|  | **normalization** | **no normalization** |
| Specificity | 65% | 85% |
| Sensibility | 37% | 16% |

Como podemos verificar, a normalização dos dados homogenia mais as características de valiação, diminuindo, afortunadamente, a especificidade, valor que ganha depois na sensibilidade. Desta forma podemos concluir que a aplicação da normalização dos dados, em média, favorece os nossos interesses classificativos.

Outra das tentativas de melhoramento dos resultados foi a diminuição do número de classes, passando de quatro (inter-ictal, pre-ictal, ictal, pos-ictal) para duas (pré-ictal, não pré-ictal). Na tabela seguinte analisamos os resultados médios destas duas abordagens:

|  |  |  |
| --- | --- | --- |
|  | **one neuron** | **four neuron** |
| Specificity | 63% | 88% |
| Sensibility | 37% | 16% |

Como podemos observar, a versão de um neurónio equilibra mais os resultados, aumentado significativamente o valor médio da sensibilidade das redes. Assim, apesar da perda na especificidade, o equilíbrio trazido aos dados faz desta abordagem um trunfo importante na detecção das crises.

Finalmente, para simplificação dos dados, foram substituídos os dados short wave e long wave das características pelo seu respectivo rácio, na esperança que esta relação fosse suficiente para caracterizar as diferentes fazes. O quadro seguinte mostra os resultados médios desta aplicação nos diferentes pacientes (cf. : *LongShortRatio.xlsx*):

|  |  |  |
| --- | --- | --- |
|  | **short/long wave ratio** | **no short/long wave ratio** |
| Specificity | 85% | 87% |
| Sensibility | 15% | 12% |

Da análise do quadro acima apresentado podemos concluir que a influência deste método é mínima, causando diferenças pouco significativas nos resultados. Desta forma, a sua utilização (uma vez que reduz o número de características para metade) é aconselhada.

### Selecção de características

Iremos agora analisar os resultados das várias técnicas de selecção de características implementadas e do número de características que obtém melhores resultados. Foram realizados os testes utilizando uma tipologia de rede que geralmente obteve bons resultados, pelos dados que recolhemos.

Através dos valores máximo obtidos não foi possível depreender qualquer padrão. Esse facto leva-nos a crer que os possíveis ganhos, com técnicas de selecção de características serão pouco acentuados. Avançámos então para um estudo dos valores médios para todas as execuções efectuadas, envolvendo cada uma das técnicas. Seguem-se os resultados desse mesmo estudo:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | **NOTHING** | **cluster** | **no cluster** | **correlation** | **no correlation** | **mutual information** | **no mutual information** |
| Specificity | 80% | 77% | 74% | 78% | 73% | 74% | 77% |
| Sensibility | 21% | 25% | 28% | 23% | 30% | 29% | 24% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **cluster + correlation** | **cluster + mutual info** | **correlation + mutual info** | **cluster + correlation**  **+ mutual info** |
| Specificity | 81% | 79% | 83% | 92% |
| Sensibility | 21% | 21% | 20% | 9% |

Como é possível observar, as diferenças são ligeiras entre cada uma das combinações. No entanto, notou-se uma eficácia ligeiramente superior na especificidade (mantendo um equilíbrio com a sensibilidade) na utilização de correlação com informação mútua, em relação à não utilização de qualquer técnica.

Em termos da influência do número de características de entrada na rede neuronal, os resultados agregados obtidos foram os seguintes:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | **2 Characteristics** | | **30 Characteristics** | | **100 Characteristics** |
| Specificity | | 72% | | 75% | | 79% |
| Sensibility | | 32% | | 26% | | 21% |
|  |  | |  | |  | | |

Existe um maior equilíbrio entre os dois parâmetros com a utilização de menos características. Este facto pode ser justificado pela maior tendência de *overfitting* da rede neuronal, na presença de mais entradas, especialmente se estas tiverem uma forte correlação entre ambas.

### Melhores resultados globais

O principal problema na elaboração deste trabalho foi conseguir uma estrutura de rede neuronal que nos garantisse um valor de **especificidade** e **sensibilidade** elevados. Conseguimos soluções que nos garantiram máxima especificidade, bem como máxima sensibilidade. No entanto, o objectivo era maximizar ambos os campos, objectivo esse que se revelou bastante complicado.

Nos dados de entrada torna-se difícil identificar com clareza a zona *pré-ictal*, o que levou a maior parte das vezes as redes a fazerem *overfitting* dos dados, tendendo demasiado para uma boa especificidade ou o oposto, revelando claras dificuldades em encontrar um equilíbrio de bom desempenho.

A melhor configuração, que nos permitiu obter os resultados de especificidade 90%e de sensibilidade 56%, foi a seguinte:

**Network Type:** 'newff'

**Transfer Function:** 'tansig'

**Learning Function:**  'trainlm'

**Performance Function:** 'mse'

**Learning Rate:**  '0.3'

**Number of Layers:** 2

**Hidden Layer Size:**  30

**Number of Characteristics:** 30

**Charact.s Selection Methods:** clustering

**Output Format:**  4 neuron