

Assignment 2 – Prediction and detection of epileptic seizures

Grupo 1 - Diogo A. Rosário (uc2023185395), Arthur N. Itacarambi (nº2020115569)

Universidade de Coimbra - FCT - Prof. António Dourado Pereira Correia



UNIVERSIDADE D
COIMBRA

1 Introdução

Este relatório é o resultado de um trabalho proposto no âmbito da disciplina de **Aprendizagem Computacional**. Apresentamos neste relatório uma visão periférica do problema em estudo *Prediction and detection of epileptic seizures*. Utilizando ferramentas de *Machine Learning*, disponibilizadas pelo matlab (aplicação externa), iremos discutir os resultados obtidos dos modelos treinados. Nos próximos capítulos será discutido o passo-a-passo que os autores seguiram para chegar aos presentes resultados, o que esses resultados representam e o que podemos concluir sobre eles.

2 Modelação do *Dataset*

Para criar o target do dataset usamos um programa autoral “correctTarget.m”, este programa gerou o arquivo “Target.mat” que também foi usado em todos os modelos apresentados neste relatório. O target consiste em uma matriz com 3 colunas e o número de linhas do dataset, a primeira coluna representa os instantes em que o paciente está em um estado *interictal*, a segunda coluna um estado *preictal* e a terceira coluna um estado *ictal*. O estado em que o paciente se encontra naquele instante é representado pelo número 1 na coluna correspondente. Para chegar nesse vetor foi nos disponibilizado um vetor *target* que possuía um número n^o de linhas onde 0 representava um estado non-ictal e 1 representava um estado *ictal*.

O data set também foi dividido em duas partes, uma para treino (85 por cento) e para teste (15 por cento) para todos os modelos apresentados e testados nesse relatório.

Obs: Qualquer outra alteração no dataset foi feito por ferramentas diferentes e descritas nos *scripts* disponibilizados.

3 Resultados

Os resultados encontram-se disponíveis na pasta “resultados/” em formato Excell. Cada resultado possui nas últimas colunas da folha, os resultados da sensibilidade e da especificidade de cada modelo treinado, tanto por *Point by Point* como por *Seizure by Seizure*. Os melhores modelos treinados para cada arquitetura são os que se encontram sublinhados a verde. É importante também notar que são denominados como “melhores modelos” os que têm a sensibilidade e especificidade mais alta e que a diferença entre estes dois valores não seja muito alta (cerca de 0.3 / 0.4).

4 Análise de Resultados

4.1 Análise de Redes Razas

4.1.1 feedforwardnet

Todos os modelos foram treinados de maneira batch, sendo que a função de treino utilizada foi a *trainlm*. No entanto, é importante destacar que na aplicação desenvolvida por este grupo, o utilizador tem a habilidade de introduzir outras funções de treino que pretenda.

Observando os resultados que se encontram no ficheiro excell **shallowNets(Panilha feedforwardnet)**, conseguimos reparar que:

1. A aplicação do equilíbrio de classes revelou-se essencial para atenuar a desigualdade na distribuição das classes. Ao reduzir significativamente a frequência de eventos interictais, esta estratégia promoveu uma representação mais equilibrada dos dados, reforçando a robustez do modelo e minimizando possíveis tendências que poderiam prejudicar a generalização.
2. A introdução da metodologia de *Error Weights* destacou-se como uma abordagem eficaz para aprimorar a sensibilidade do modelo, concentrando-se nas fases preictal e ictal. A atribuição de pesos diferenciados aos erros proporcionou um acerto mais preciso, resultando numa maior importância dada a eventos críticos. Este ajuste contribuiu diretamente para uma performance aprimorada.
3. Em contraste, o impacto dos autoencoders revelou uma observação interessante. Surpreendentemente, não identificámos uma grande diferença no desempenho do modelo com a incorporação de autoencoders. Apesar das expectativas de melhorias, os resultados não indicaram uma vantagem significativa. Esta observação sugere que, para o contexto específico deste estudo, os autoencoders podem não ter desempenhado um papel tão preponderante quanto as estratégias de equilíbrio de classes e *Error Weights*.
4. Além disso, é interessante notar que os melhores resultados foram obtidos em diferentes tipos de funções de ativação ("*purelin*", "*logsig*", "*tansig*"). Observou-se também que, quanto mais camadas a *network* possuía, melhores eram os resultados.
5. Notavelmente, observamos que o número de neurónios por camada não apresentou melhorias nos resultados.

Para finalizar este ponto, todos os melhores modelos que estão construídos com esta arquitetura, na generalidade, são melhores para **detetar** um episódio epiléptico.

4.1.2 layrecnet

Observando os resultados que se encontram no ficheiro excell **shallowNets(Panilha Delays)**, conseguimos reparar que tal como no ponto anterior, o uso de: 1) equilíbrio de classes; 2) Metodologia de *Error Weights*; mostraram-se bastante importantes para obter melhores resultados pela mesma razão que foi descrita no ponto anterior.

Foi também usado autoencoders mas não se observou grandes diferenças nos resultados.

Para finalizar este ponto, todos os melhores modelos que estão construídos com esta arquitetura, na generalidade, são melhores para **detetar** um episódio epiléptico.

4.2 Análise de Redes Profundas

4.2.1 CNN's

Observando os resultados que se encontram no ficheiro excell **CNN**, conseguimos reparar que:

1. O uso do equilíbrio de classes revelou-se mais uma vez crucial. A sua aplicação não só contrabalançou a disparidade na distribuição das classes, como também contribuiu para uma performance mais estável e consistente entre as diferentes arquiteturas. Este ajuste teve um impacto positivo na capacidade do modelo em generalizar para eventos menos frequentes, reforçando a confiabilidade das previsões.

2. Observamos que o aumento no número de *layers* resultou em melhorias significativas em casos específicos, especialmente em situações em que a hierarquia de características desempenhou um papel crucial na discriminação entre eventos. No entanto, é importante notar que este benefício não foi uniforme em todas as configurações.
3. A escolha entre pooling "*average*" e "*max*" revelou-se como um fator determinante na otimização da sensibilidade e especificidade do modelo. Especificamente, a opção "*average*" destacou-se por equilibrar de forma notável os valores dessas métricas cruciais. Esta abordagem minimizou desequilíbrios entre a capacidade do modelo em identificar eventos positivos e negativos, resultando em uma performance mais equitativa e abrangente.
4. Mais uma vez, o uso de autoencoders não se mostrou altamente relevante nos resultados obtidos. Apesar das expectativas de melhoria na representação latente dos dados, os autoencoders não contribuíram de maneira significativa para a eficácia do modelo na detecção de eventos epiléticos.
5. A observação de que não apenas o tamanho do *Pooling*, mas também o número de filtros e o tamanho deles tiveram um impacto mínimo nos resultados.
6. A combinação eficaz destes três elementos - equilíbrio de classes, otimização do número de *layers* e seleção do tipo de *pooling* - resultou em melhorias substanciais na capacidade do modelo em prever eventos epiléticos.

Para finalizar este ponto, os dois melhores modelos que estão construídos com esta arquitetura, na generalidade, são melhores para **prever** um episódio epilético. No entanto o modelo que está identificado no excell com um verde mais escuro, também é relevante para **detetar** casos epiléticos.

4.2.2 LSTM

Observando os resultados que se encontram no ficheiro excell **LSTM**, conseguimos reparar que:

1. O uso do equilíbrio de classes mais uma vez relevou-se importante pela razão que foi descrita em todas as arquiteturas anteriores.
2. Mais uma vez, o uso de autoencoders não se mostrou altamente relevante nos resultados obtidos. Neste caso até pioram o resultado final.
3. O uso de mais ou menos neurónios nas *hidden layers*, não afetou significativamente o resultado da previsão nem da deteção de episódios.

Para finalizar este ponto, os dois melhores modelos que estão construídos com esta arquitetura são melhores para **detetar** um episódio epilético.

5 Guia da Aplicação

5.1 Treino e teste de redes rasas

Ao rodar o arquivo "GUI.matapp" a primeira tela que aparece ao utilizador:

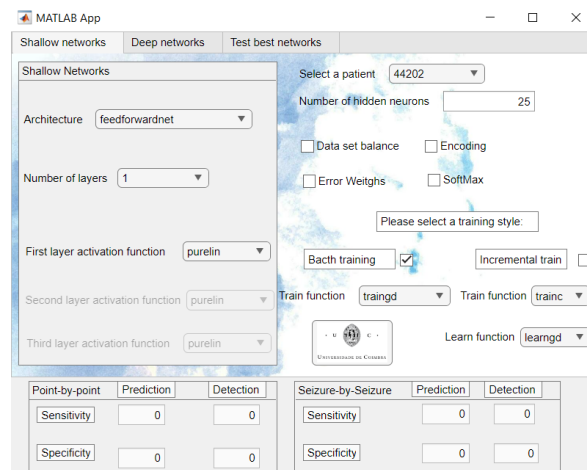


Figure 1: Primeira página da App.

Nessa tela o utilizador se depara com diversas opções pra treinar as redes rasas. A app foi construída de uma forma que as opções selecionadas já excluem a possibilidade de outras opções serem utilizadas, fazendo com o que o utilizador não precise de se preocupar ao treinar os modelos com os parâmetros que seleciona. Após selecionar as opções o utilizador deve apenas clicar no logotipo da Universidade de Coimbra, isso vai fazer com que o utilizador veja duas janelas distintas: a janela de treino de redes neuronais do Matlab e um esquema do modelo proposto. Na app, os resultados dos testes aparecem na parte inferior da janela.

5.2 Treino e testes de redes profundas

Assim como na janela anterior as opções e a execução do código se faz da mesma maneira, de novo, o utilizador não precisa se preocupar com os parâmetros que introduz. Os resultados vão aparecer na parte inferior da janela.

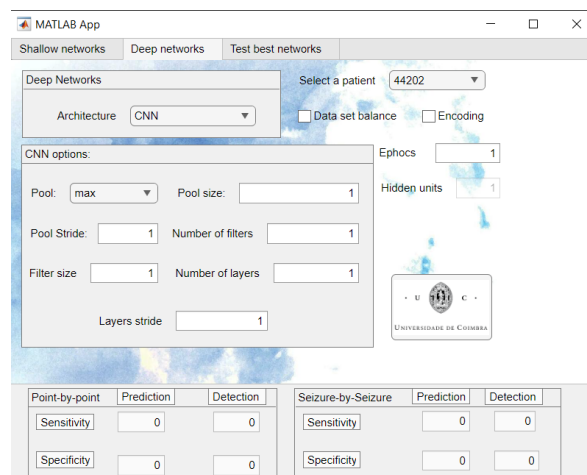


Figure 2: Segunda página da App.

5.3 Melhores modelos alcançados

Nesta janela o utilizador apenas precisa apertar qualquer um dos botões disponíveis nessa janela para alcançar os resultados dos modelos que, novamente, aparecem na parte inferior da janela.

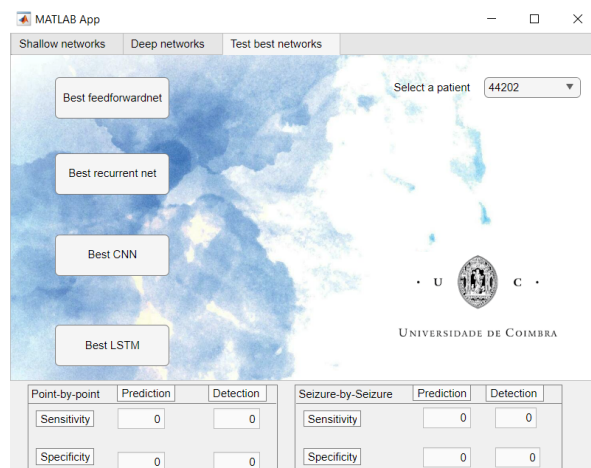


Figure 3: Terceira página da App.

6 Conclusão

A conclusão deste estudo sobre a detecção de episódios epiléticos por meio de Aprendizagem Computacional destaca várias descobertas cruciais. De forma geral, a implementação do equilíbrio de classes e a atribuição de *Error Weights* (no caso de redes rasas) mostraram-se fundamentais para alcançar resultados mais equilibrados e precisos na previsão e detecção de episódios epiléticos. Contrariamente às expectativas, a utilização de autoencoders na nossa análise, não demonstrou relevância significativa nos resultados.

A modelação de redes rasas emergiram como arquiteturas boas para a detecção de episódios epiléticos. Por outro lado, as *CNNs* revelaram-se mais aptas para a previsão e a sua detecção. As *LSTM* mostraram-se particularmente eficientes na detecção.

O equilíbrio cuidadoso entre redes rasas, *CNNs* e *LSTM*, aliado a estratégias de equilíbrio de classes, destaca-se como um guia valioso para avanços significativos na detecção e previsão desses eventos cruciais na epilepsia.

Além disso, é relevante mencionar que a abordagem de calcular a previsão e detecção *Point by Point*, utilizando sensibilidade e especificidade, foi aplicada com sucesso. Além disso, a metodologia "*Seizure by Seizure*" também foi implementada, proporcionando uma avaliação abrangente e generalizada da performance dos modelos em relação aos episódios epiléticos. Essa análise multifacetada fortalece a robustez das conclusões, oferecendo uma visão mais completa e refinada do desempenho dos modelos em diferentes cenários clínicos.

Finalmente, é necessário enfatizar que os melhores modelos treinados encontram-se sublinhados a verde em cada *excell* corresponde.