A red and white logo

Description automatically generated

Redes Neuronais

Conhecimento e Raciocínio

Carolina Vilas Boas Veloso- a2021140780

Luís Miguel Martins Duarte- a2021137789

Table des matières

[Introdução 3](#_Toc166450598)

[O Nosso DataSet 3](#_Toc166450599)

[BALENCEADO: 5](#_Toc166450600)

[INSTÂNCIAS: 5](#_Toc166450601)

[CBR 5](#_Toc166450602)

[Retrieve 5](#_Toc166450603)

[Treino e estudo estatístico 7](#_Toc166450604)

[Pasta Start 7](#_Toc166450605)

[Pasta Train 7](#_Toc166450606)

# Introdução

Este trabalho foi realizado no âmbito da Unidade Curricular, Conhecimento e Raciocino, no 2º Semestre da Licenciatura em Engenharia Informática no ano letivo 2023/2024. Consiste no desenvolvimento e estudo estatístico de redes neuronais para classificar o nosso Dataset escolhido, neste caso o da Hepatite.

# O Nosso DataSet

O objetivo principal ao trabalhar com este Dataset é desenvolver modelos de classificação capazes de prever com precisão a categoria de Hepatite de um paciente com base nas suas características clínicas e demográficas. Estes modelos podem ser úteis para ajudar no diagnóstico precoce, triagem de pacientes e planeamento do tratamento para indivíduos com Hepatite.

O dataset possui cerca de cinco classes, representadas pelos valores 0, 0s, 1, 2 e 3. Cada classe corresponde a um estágio ou tipo diferente de hepatite:

* Blood Donor (0)
* Suspect Blood Donor (0s)
* Hepatitis (1)
* Fibrosis (2)
* Cirrhosis (3)

Ao resolver esse problema, o objetivo é identificar com precisão em qual categoria de problema hepático um paciente se enquadra, proporcionando assim uma base para intervenções médicas adequadas e oportunas.

Este Dataset apresenta ainda cerca de 12 atributos, sendo eles:

1. Age (Idade)
   * Representa a idade do paciente a ser avaliado.
2. Sex (Sexo)
   * Representa o sexo do paciente a ser avaliado.
3. ALB (Albumina)
   * Representa uma proteína presente no sangue que desempenha um papel importante no transporte de várias substâncias, incluindo hormônios, medicamentos e íons. Níveis anormais de albumina podem indicar problemas no fígado, rins ou outras condições médicas.
4. ALP (Fosfatase Alcalina)
   * É uma enzima encontrada em vários tecidos do corpo, incluindo fígado, ossos e intestinos. Níveis elevados de fosfatase alcalina no sangue podem indicar problemas hepáticos, ósseos ou biliares.
5. ALT (Alanina Aminotransferase)
   * É uma enzima encontrada principalmente no fígado. Elevações nos níveis de ALT podem indicar danos ou doenças hepáticas, como hepatite ou cirrose.
6. AST (Aspartato Aminotransferase)
   * É uma enzima presente em vários tecidos, incluindo músculos e fígado. Elevações nos níveis de AST podem ser indicativas de danos ao fígado, mas também podem estar presentes em condições como infarto do miocárdio ou lesão muscular.
7. BIL (Bilirrubina Total)
   * É um pigmento amarelo produzido pela quebra das células vermelhas do sangue. A bilirrubina total é a quantidade total de bilirrubina presente no sangue. Níveis elevados podem indicar problemas no fígado, vesícula biliar ou problemas na quebra das células vermelhas do sangue.
8. CHE (Colinesterase)
   * É uma enzima que desempenha um papel na degradação de certas substâncias químicas no corpo, incluindo alguns pesticidas e medicamentos. Níveis anormais de colinesterase podem indicar exposição a substâncias tóxicas ou danos nos órgãos.
9. CHOL (Colesterol)
   * É uma substância  acastanhada encontrada no sangue e em todas as células do corpo. Existem diferentes tipos de colesterol, como o colesterol HDL (bom) e LDL (ruim). Níveis elevados de colesterol LDL estão associados a um maior risco de doenças cardíacas.
10. CREA (Creatinina)
    * É um resíduo produzido pelo metabolismo muscular. Os níveis sanguíneos de creatinina são usados ​​como um indicador da função renal. Elevações nos níveis de creatinina podem indicar problemas renais.
11. GGT (Gama-Glutamil Transferase)
    * É uma enzima encontrada em vários tecidos do corpo, incluindo fígado, rins e pâncreas. Elevações nos níveis de GGT podem indicar danos no fígado ou problemas biliares.
12. PROT (Proteína Total)
    * É uma medida da quantidade total de proteína no sangue, incluindo albumina e outras proteínas. Níveis anormais de proteína total podem ser indicativos de desnutrição, problemas renais, hepáticos ou outros distúrbios médicos.

## BALENCEADO:

O meu Dataset não é balanceado visto que existe uma descrepância enorme entre o número de casos por classe:

* Blood Donor (0) -> 529
* Suspect Blood Donor (0s) -> 6
* Hepatitis (1) -> 21
* Fibrosis (2) -> 18
* Cirrhosis (3) -> 29

## INSTÂNCIAS:

O ficheiro Train apresenta 606 instâncias, enquanto que o ficheiro Start apresenta 10 instâncias.

606 , ficheiro train

10, ficheiro start

## CBR

Utilizamos o peso 4 para a idade, porque a nosso ver quanto maior a idade, mais peso tem no cálculo da similaridade. Para o sexo usamos um peso de 2, porque consideramos que o sexo nao tenha assim tanto peso para o cálculo. Para o resto dos atributos utilizamos um peso de 6, pois consideramos que são todos igualmente importantes para a deteção da doença.

## Retrieve

**Inicialização:**

Define os fatores de ponderação (weighting\_factors) para calcular a similaridade entre os casos.

Inicializa listas vazias para armazenar os índices dos casos recuperados e as suas similaridades.

Remove atributos ausentes:

Remove os fatores de ponderação correspondentes aos atributos ausentes.

**Cálculo da similaridade:**

Calcula a similaridade entre os casos usando uma média ponderada das distâncias, normalizada pelo somatório dos fatores de ponderação.

Usando ou calculate\_linear\_distance ou calculate\_euclidean\_distance

**Retorno:**

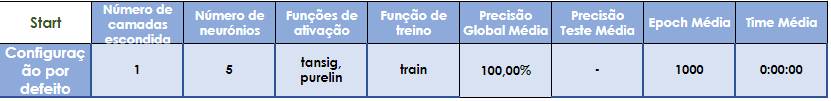
Retorna os índices dos casos recuperados, as similaridades correspondentes e o próprio new\_case.

# Treino e estudo estatístico

## Pasta Start

Para começar, treinou-se a pasta start, com uma rede neuronal feedforward, com uma camada de 5 neurónios, tansig e purelin para função de ativação e train para função de treino.

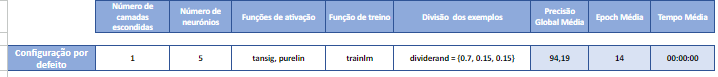
Alcançou-se um total de 1000 epochs e uma média de 100% de precisão.



Com base nestes resultados, podemos concluir que este modelo apresenta um desempenho como seria espeado na classificação das amostras, sugerindo que é capaz de capturar com eficiência as padrões presentes nos dados a realizar a classificação correta da maioria das amostras.

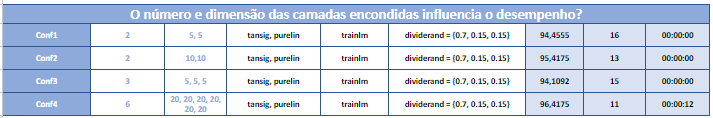
## Pasta Train

**Classificação por defeito**

Com base no treino da pasta anterior, implementamos mais estudos usando diferentes parametrizações.

Com base nestes resultados, podemos concluir que o modelo da rede neuronal, tem uma boa performance na classificação das amostras. A precisão global de 94,19% mostra que ele consegue classificar corretamente bem mais de metade das amostras do conjunto. Logo existe um desempenho alto.

**Número e dimensão de camadas**



1) Duas camadas escondidas com cinco neurónios cada:

- Precisão global média: 94,45%

Com base nos resultados, observamos um desempenho satisfatório do modelo. A alta precisão global indicaa que o modelo teve facilidade em classificar corretamente as amostras do conjunto de dados. A combinação de duas camadas escondidas com apenas cinco neurónios pode ser suficiente para capturar a complexidade dos padrões presentes nos dados.

2) Duas camadas escondidas com dez neurónios cada:

- Precisão global média: 95,42%

Nesse caso, observamos uma melhoria na precisão global em comparação com a configuração anterior. Aumentar o número de neurónios nas camadas escondidas permitiu uma melhor aprendizagem dos padrões.

3) Três camadas escondidas com cinco neurónios cada:

- Precisão global média: 94,11%

Com a adição de uma terceira camada escondida, não notamos uma melhoria significativa na precisão global, comparando com os dois casos anteriores. Isto indica que o modelo não foi capaz de aprender melhor os padrões dos dados e generalizar mais efetivamente para novas amostras.

4) Seis camadas escondidas com vinte neurónios cada:

- Precisão global média: 96,42%

Com esta configuração, observamos uma alta precisão global em comparação com a maioria das configurações anteriores. Aumentar tanto o número de camadas escondidas quanto o número de neurónios em cada camada permitiu ao modelo aprender representações mais complexas dos dados, mas pode ter ocorrido um problema na sua generalização.

A white and blue grid with black text

Description automatically generated with medium confidence**Funções de Treino**

1) Uma camada escondida com cinco neurónios e função de treino traingdx:

- Precisão global média: 90,52%

Considerando a função de treino traingdx, uma variação de retro propagação do gradiente com atualização adaptativa de taxa de aprendizagem e momento. Esta função de treino ajusta os pesos da rede neuronal iterativamente, procurando minimizar o erro entre as saídas previstas e os valores reais. No entanto, mesmo com o uso desta função de treino, os resultados obtidos apesar de serem satisfatórios, não são os mais eficazes.

**GDX** - algoritmo de treino baseado no método do gradiente descendente com momento. Ele utiliza a direção do gradiente ponderada pelo momento para atualizar os pesos da rede neuronal. O momento ajuda a acelerar o treinamento, especialmente em áreas com curvatura mais suave da função de erro.

2) Uma camada escondida com cinco neurónios e função de treino traingdm:

-Precisão global média: 88,85%

Considerando a função de treino traingdm, baseado no método do gradiente descendente com momento, os resultados obtidos apesar de serem bastante superiores a 50%, ficam aquém dos resultados obtidos pelas outras funções de treino.

**GDM** - algoritmo de treino para redes neuronais, baseado no método do gradiente descendente com momento. Usa uma abordagem adaptativa para ajustar a taxa de aprendizagem e incorpora um termo de momento para acelerar o treino.

3) Uma camada escondida com cinco neurónios e função de treino trainscg:

- Precisão global média: 91,19%

O algoritmo trainscg utiliza o método de otimização de gradiente conjugado escalado (SCG) para treinar a rede neuronal. Esta função de treino acabou por ser a mais eficaz em termos de resultados, tendo por isso sido tanto os resultados como o desempenho bastante satisfatórios.

**SCG** - algoritmo de treino eficiente que utiliza uma combinação de técnicas de gradiente conjugado e atualização adaptativa da taxa de aprendizagem. Ajusta os pesos da rede neuronal por meio de uma busca direta em direção à solução ótima, evitando a necessidade de calcular a matriz Hessiana completa.

4) Uma camada escondida com cinco neurónios e função de treino traincgf:

- Precisão global média: 90,16%

O algoritmo traincgf utiliza o método de otimização de gradiente conjugado com Fletcher-Reeves para treinar a rede neuronal. Apesar de um leve retrocesso nos resultados, o desempenho continua a ser bastante satisfatório.

**CGF** - um algoritmo de treino baseado no método de gradiente conjugado e utiliza a direção do gradiente para atualizar os pesos da rede neuronal. Ele é uma variação do método de Fletcher-Powell que pode melhorar a eficiência do treino em algumas situações.

5) Uma camada escondida com cinco neurónias e função de treino trains:

-Precisão global média: 87,96%

Considerando a função de treino trains, estes foram os piores resultados obtidos nos treinos todos, no entanto considero que ainda se podem considerar que tenha tido um desempenho satisfatório.

**S** - Método de treinamento para redes neurais que prioriza a atualização incremental dos pesos da rede em uma ordem sequencial. Ajusta os pesos gradualmente à medida que novos exemplos de treino são introduzidos. Isso permite uma adaptação contínua da rede aos padrões nos dados.

Função de Ativação

A blue and white grid with black text

Description automatically generated

1) Uma camada escondida com cinco neurónios, funções de ativação logsig e purelin:

- Precisão global média: 93,95%

Esta configuração resultou numa precisão global alta. A função de ativação logsig na camada oculta pode ter ajudado a capturar os padrões dos dados.

2) Uma camada escondida com cinco neurónios, funções de ativação purelin e logsig:

- Precisão global média: 89,48%

Observamos uma precisão global relativamente alta, mas comparando com a anterior bem mais pequena. A função de ativação purelin na camada oculta pode ter limitado a capacidade do modelo de aprender padrões complexos nos dados.

3) Uma camada escondida com cinco neurónios, funções de ativação tansig e tansig:

- Precisão global média: 94,02%

Neste caso, observamos uma precisão global relativamente alta. A função de ativação tansig nas camadas ocultas pode ter permitido que o modelo aprendesse de forma mais eficaz os padrões dos dados, resultando num desempenho melhor.

4) Duas camadas escondidas com cinco neurónios cada, funções de ativação logsig, tansig e purelin:

- Precisão global média: 94,60%

Com esta configuração, a precisão global continua relativamente alta. O uso de duas camadas ocultas parece ter melhorado um pouco o desempenho do modelo.

5) Uma camada escondida com cinco neurónios, funções de ativação tansig e logsig:

- Precisão global média: 89,43%

Aqui, também observamos uma precisão global alta, mas relativamente mais baixa comparando com as outras. A configuração das funções de ativação pode não ter permitido que o modelo aprendesse adequadamente os padrões dos dados.

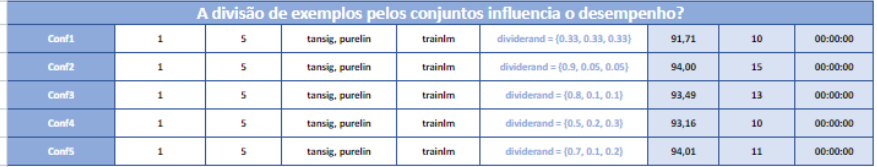
6) Uma camada escondida com dez neurónios, funções de ativação purelin e purelin:

- Precisão global média: 93,71%

Com esta configuração, observamos uma precisão global alta.

Em resumo, as diferentes configurações de camadas escondidas, neurónios e funções de ativação resultaram em desempenhos variados. A configuração com duas camada escondida, cinco neurónios e funções de ativação "logsig", "tansig" e “purelin” pareceu ter o melhor desempenho, alcançando uma precisão global mais alta em comparação com as outras configurações testadas.

**Divisão de exemplos**



Divisão dos exemplos. A função de divisão por defeito (dividerand) cria os 3 conjuntos de treino, validação e teste, respetivamente, com 70%, 15% e 15% dos exemplos. Estes valores podem ser alterados através das variáveis pertencentes ao objeto net.divideParam.

1. dividerand = {0.33, 0.33, 0.33}
   1. Precisão global média: 91,71%
2. dividerand = {0.9, 0.05, 0.05}
   1. Precisão global média: 94,00%
3. dividerand = {0.6, 0.4, 0.4}
   1. Precisão global média: 93,49%
4. dividerand = {0.8, 0.1, 0.1}
   1. Precisão global média:93,16%
5. dividerand = {0.5, 0.2, 0.3}
   1. Precisão global média:94,01%

Observa-se que, em geral, as configurações apresentaram uma precisão global moderada, variando entre 91,71% e 94,01%.