isec
Politécnico de Coimbra

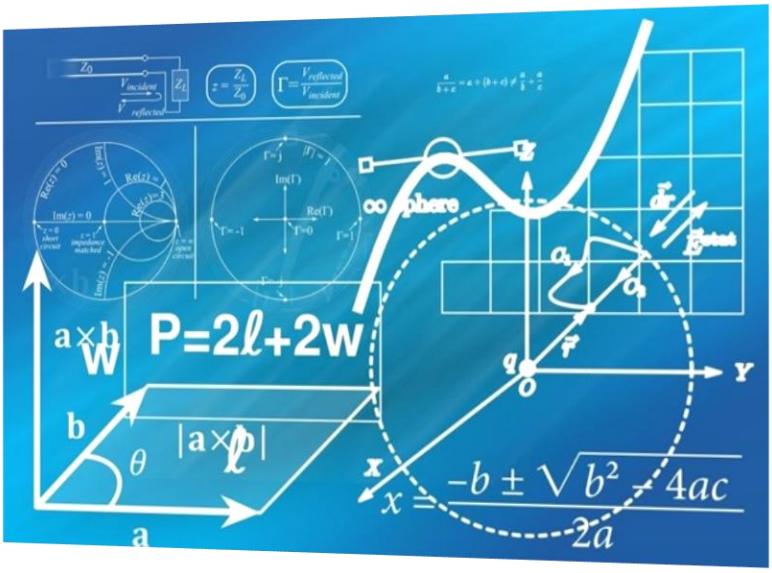
Licenciatura em Engenharia Informática

Conhecimento e Raciocínio

RELATÓRIO

TRABALHO PRATICO

Redes Neurais



Grupo:

Afonso Da Silva: **2021133861**

Frederico Quelhas: **2022135081**

Coimbra – 29 de março de 2024

ÍNDICE

INTRODUÇÃO **2**

1. Preparação do Dataset **3**

2. Estudo e Análise de Redes Neuronais **4**

CONCLUSÃO **10**

INTRODUÇÃO

Este relatório apresenta um estudo realizado na disciplina de Conhecimento e Raciocínio, focado na aplicação de redes neuronais para classificar condições médicas, especificamente Cirroses. O objetivo é explorar a eficácia de diferentes configurações de redes neuronais em diagnósticos médicos, utilizando um dataset específico para avaliar a precisão e a generalização desses modelos em um contexto prático.

1. Preparação do Dataset

O dataset TRAIN foi analisado e preparado para utilização em redes neuronais. Os atributos foram convertidos em valores numéricos ou booleanos, e os valores em falta foram tratados com base em raciocínio baseado em casos.

1. Identificação do Atributo Alvo (Ponto b)

O primeiro passo foi identificar o atributo que corresponde à saída desejada (target) da rede neuronal. Determinamos que a coluna **Stage** com base nas informações da descrição dos DATASETS. No entanto, observamos a presença de valores em falta identificados como NA nessa coluna.

2. Implementação do Sistema de Raciocínio Baseado em Casos (Ponto c)

Para lidar com os valores em falta (NA), implementamos um sistema de raciocínio baseado em casos. Essa abordagem utiliza apenas a fase de RETRIEVE para encontrar o caso mais semelhante ao caso com valor em falta e usa os valores desse caso para preencher os valores em falta.

A função **retrieveTP**, desenvolvida para essa finalidade, realiza as seguintes operações:

- Percorre os dados do dataset.
- Identifica os valores em falta (NA).
- Substitui esses valores pela média dos valores até o momento em que foi encontrado.
- Realiza algumas conversões de valores específicos de algumas colunas, como por exemplo substituir 'D' por 2, 'CL' por 1 e 'C' por 0 na terceira coluna.

3. Chamada da Função para Preparação do Dataset

Para efetuar a preparação do dataset, a função **cbrTP** foi desenvolvida. Esta função é responsável por:

- Carregar os dados do arquivo CSV.
- Chamar a função **retrieveTP** para realizar o preenchimento dos valores em falta.
- Exibir os dados modificados, agora prontos para serem utilizados no estudo e análise de desempenho das redes neurais feedforward.

2. Estudo e Análise de Redes Neuronais

a) Durante o estudo, exploramos diversas configurações de redes neuronais para identificar a mais adequada para classificação em datasets médicos. As redes foram configuradas com diferentes números de camadas ocultas (1 a 3), número de neurônios por camada (10, 20, 30), e variadas funções de ativação (tansig, logsig, purelin) e treino (trainlm, trainbr, traingdx). A escolha dessas configurações visa compreender o impacto da complexidade da rede e do tipo de aprendizado no desempenho e na generalização.

Função treino:

No estudo das funções de treino com o dataset 'Start', traingd se destacou como a mais eficaz, equilibrando treino e teste de forma mais consistente em comparação com trainoss, trainbfg, e trainscg, que mostraram baixa precisão de teste, indicando problemas de generalização.

A função de treino influencia o desempenho?								
Conf1	1	10	tansig, purelin	traingd	dividerand = {0.7, 0.15, 0.15}		55.00	25.00
Conf2	1	10	tansig, purelin	trainbfg	dividerand = {0.7, 0.15, 0.15}		42.50	12.50
Conf3	1	10	tansig, purelin	trainscg	dividerand = {0.7, 0.15, 0.15}		40.00	12.50
Conf4	1	10	tansig, purelin	trainoss	dividerand = {0.7, 0.15, 0.15}		47.50	12.50

funções de ativação:

Na análise das funções de ativação no dataset 'Start', compet e elliotsig foram as mais eficazes, equilibrando bem o treino (80.00%) e teste (50.00%). Outras combinações, como tansig e logsig, mostraram alguma generalização, enquanto logsig e purelin, e hardlim variantes indicaram problemas de overfitting ou eficácia limitada no treino e nenhuma generalização

As funções de ativação influenciam o desempenho?							
Conf1	1	10	logsig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	55.00	0
Conf2	1	10	tansig, logsig	trainlm	dividerand = {0.7, 0.15, 0.15}	55.00	25.00
Conf3	1	10	compet, elliotsig	trainlm	dividerand = {0.7, 0.15, 0.15}	80.00	50.00
Conf4	1	10	elliotsig, compet	trainlm	dividerand = {0.7, 0.15, 0.15}	20.00	0.00
Conf5	1	10	hardlim, hardlims	trainlm	dividerand = {0.7, 0.15, 0.15}	30.00	0.00
Conf6	1	10	hardlims, hardlim	trainlm	dividerand = {0.7, 0.15, 0.15}	10.00	0.00

O entanto apesar de se ter obtido valores bons em geral, não é ideal efetuar um estudo apenas a este dataset devido ao baixo número de stage, sendo que deve ser tomado apenas como comparação para dataset mais extensos.

b) Nesta alínea foram feitos testes o dataset da pasta ‘train’.

Utilizando o dataset TRAIN preparado, diversas arquiteturas de redes foram analisadas, assim como diferentes funções de treino e de ativação. As três melhores redes foram salvas para análise posterior.

Influência do número e dimensão das camadas escondidas:

Aumentar o número e dimensão das camadas ocultas nem sempre melhora o desempenho da rede neural. Configurações com duas camadas ocultas não demonstraram melhorias significativas em relação a uma única camada. Além disso, um aumento excessivo no número de neurônios e camadas pode até piorar o desempenho, como observado na Configuração 4.

O número e dimensão das camadas encondidas influencia o desempenho?							
Conf1	2	5, 5	tansig, tansig, purelin	trainlm	dividerand = {0.7, 0.15,	51.72	39.34
Conf2	2	10,10	tansig, tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	51.47	26.23
Conf3	3	5, 10, 5	tansig, tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	52.70	49.18
Conf4	3	10, 10, 10	tansig, tansig, purelin	trainlm	dividerand = {0.7, 0.15, 0.15}	47.55	36.07

Portanto, com base nos resultados fornecidos, parece que a configuração por defeito com uma única camada oculta e 10 neurônios oferece um desempenho competitivo em comparação com as configurações com múltiplas camadas ocultas. A introdução de mais camadas ocultas ou mais neurônios não necessariamente melhora o desempenho da rede neural neste contexto.

Influência da função de treino:

No próximo teste, foram testadas várias parametrizações dos parâmetros de divisão de treino, sendo que neste teste foi possível obter alguns melhores resultados do que no teste anterior em que se alterou a dimensão das camadas escondidas e número de neurónios por camada.

A função de treino influencia o desempenho?							
Conf1	1	10	tansig, purelin	traingd	dividerand = [0.7, 0.15, 0.15]	41.91	40.98
Conf2	1	10	tansig, purelin	trainbfg	dividerand = [0.7, 0.15,	55.39	47.54
Conf3	1	10	tansig, purelin	trainscg	dividerand = [0.7, 0.15,	52.21	47.54
Conf4	1	10	tansig, purelin	trainoss	dividerand = [0.7, 0.15,	49.26	57.38

Influência das funções de ativação:

Ajustar o parâmetro de validação, eliminando-o e variando as proporções de treino e teste, influenciou positivamente o desempenho das redes neurais. A configuração que equilibrou treino e teste (Conf2) e aquela que priorizou teste sobre treino (Conf3) demonstraram melhorias significativas na precisão de teste, sugerindo que uma maior exposição a dados de teste pode melhorar a generalização da rede. A eliminação da validação parece ter beneficiado o desempenho geral, com a Conf3 alcançando a maior precisão de teste de 43.36%.c) As redes obtidas foram testadas utilizando o dataset TEST, avaliando-se a capacidade de generalização e aprendizagem. Diferentes funções de ativação também influenciam o desempenho da rede neural.

O parâmetro de validação influencia o desempenho?							
Conf1	1	10	tansig, purelin	traingd	dividerand = [0.7, 0.	45.83	24.59
Conf2	1	10	tansig, purelin	traingd	dividerand = [0.5, 0.	40.44	37.75
Conf3	1	10	tansig, purelin	traingd	dividerand = [0.3, 0.	43.38	43.36

Registro e Avaliação das Três Melhores Redes Neuronais

Descrição do Processo

O script MATLAB trainFeedForward é utilizado para explorar uma variedade de configurações de redes neuronais feedforward e identificar as três com melhor desempenho. O processo segue várias etapas detalhadas:

Carregamento do Dataset: Inicia-se carregando os dados de entrada e alvos (targets) do arquivo 'trainModifiedNew.csv', onde os alvos são codificados em one-hot para adequação ao problema de classificação.

Configuração de Parâmetros:

Camadas Ocultas e Neurônios: Experimenta-se com diferentes números de camadas ocultas (1 a 3 camadas) e quantidades de neurônios (10, 20, 30).

Funções de Treinamento: Varia-se entre trainlm, trainbr, e traingdx.

Funções de Ativação: Testa-se tansig, logsig, e purelin.

Divisão dos Dados: Define-se proporções distintas para treinamento, validação e teste.

Criação e Treinamento de Redes:

Para cada combinação de configurações, uma rede é criada com a função feedforwardnet.

A rede é configurada com funções de divisão aleatória dos dados e proporções especificadas.

Cada camada da rede é configurada com uma função de ativação específica.

Avaliação de Desempenho:

Após o treinamento, a rede é simulada usando a função sim.

A precisão é calculada comparando as saídas da rede com os alvos codificados em one-hot.

As três redes com as maiores precisões são identificadas e armazenadas.

Armazenamento de Configurações e Redes:

As configurações das três melhores redes são armazenadas em uma estrutura, juntamente com informações detalhadas sobre a configuração de cada rede.

As redes são salvas em best_network_1.mat, best_network_2.mat e best_network_3.mat para uso futuro.

Exibição dos Resultados:

As configurações e precisões das três melhores redes são exibidas no console para revisão.

Uso do Ficheiro TESTE para Análise de Generalização

Esta seção visa avaliar a capacidade de generalização das três melhores redes neuronais identificadas anteriormente. O foco é entender como essas redes performam em dados nunca vistos anteriormente, simulando uma situação real de aplicação onde a rede deve fazer previsões sobre novos casos.

Para esta análise, utilizamos o dataset contido no TESTE.csv, que inclui exemplos que não foram usados durante o treinamento das redes. A metodologia adotada inclui as seguintes etapas:

Preparação dos Dados:

O dataset TESTE é dividido em variáveis de entrada e de target. O target representa as categorias de classificação que cada exemplo pertence, sendo crucial para a avaliação da precisão das redes.

Avaliação das Redes:

As três redes salvas anteriormente são carregadas. As redes são utilizadas para prever os resultados no dataset TESTE sem qualquer treino adicional, garantindo que a avaliação reflete a capacidade de generalização.

A precisão é calculada como a percentagem de previsões corretas em relação ao total de previsões feitas, comparando as saídas das redes com os targets reais.

Resultados

Os resultados obtidos na avaliação de generalização das redes são os seguintes:

Precisão da Rede 1: 40%

Precisão da Rede 2: 30%

Precisão da Rede 3: 20%

Estas métricas de acerto indicam como cada rede conseguiu lidar com novos dados, destacando diferenças significativas em suas capacidades de prever corretamente as categorias de classificação.

CONCLUSÃO

Este estudo explorou a eficácia de diferentes configurações de redes neuronais na classificação de datasets médicos, especificamente focado em Cirroses. A preparação cuidadosa do dataset TRAIN e o tratamento inovador de dados faltantes estabeleceram uma fundação robusta para a análise subsequente.

Descobrimos que configurações mais simples de redes neuronais frequentemente igualam ou superam configurações mais complexas em termos de desempenho e generalização. Este achado sugere que, em contextos médicos, redes mais simples podem ser preferíveis devido à sua eficiência computacional e facilidade de interpretação.

Os testes de generalização no dataset TESTE revelaram que, embora as redes conseguissem lidar com novos dados, todas apresentaram espaço para melhoria. A rede mais eficiente alcançou uma precisão de 40%, destacando a necessidade de otimizações futuras para aprimorar a precisão e a aplicabilidade prática das redes neuronais em diagnósticos médicos.