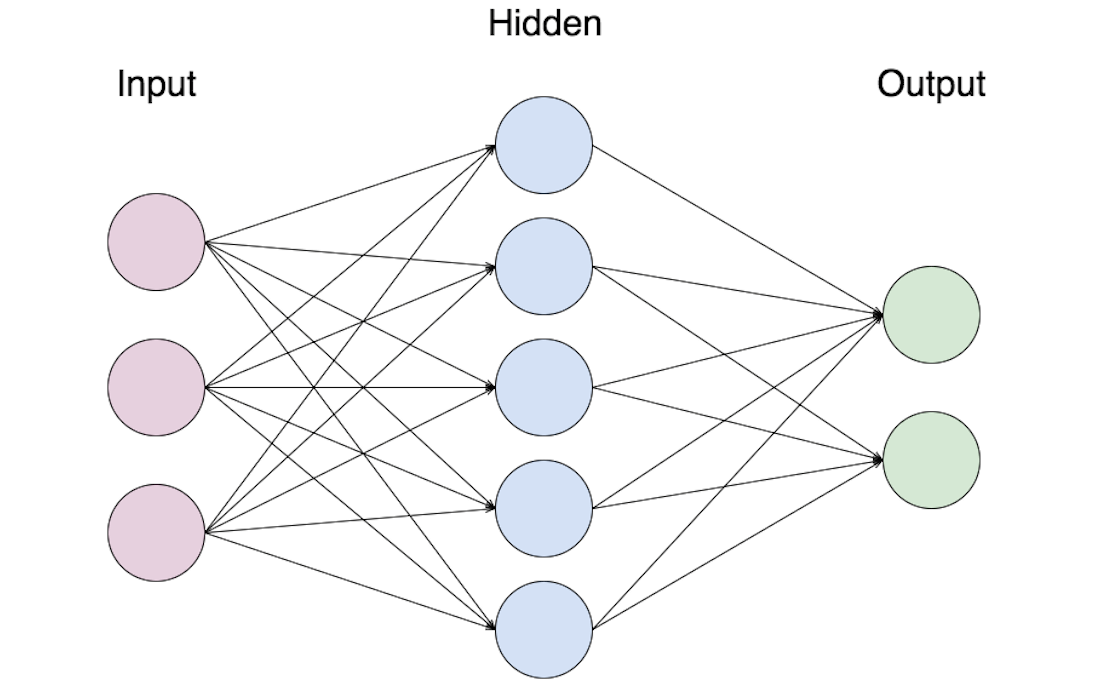
Uma imagem com texto

Descrição gerada automaticamente

Relatório do Trabalho Prático

sobre

Redes Neuronais



Conhecimento e Raciocínio

**Trabalho realizado por:**

Rúben Santos – 2019116244

Pedro Brás –

Índice

[Introdução 3](#_Toc132964013)

[Decisões tomadas 4](#_Toc132964014)

[Tratamento das Imagens 4](#_Toc132964015)

[Targets e Inputs 4](#_Toc132964016)

[Guardar Net’s 4](#_Toc132964017)

[Treino e Estudo Estatístico 5](#_Toc132964018)

[Alínea a) 5](#_Toc132964019)

[Alínea b) 5](#_Toc132964020)

[*Uma só Net* 5](#_Toc132964021)

[*Nets Separadas* 5](#_Toc132964022)

[Alínea c) 5](#_Toc132964023)

[Aplicação 6](#_Toc132964024)

[Conclusões 7](#_Toc132964025)

[Bibliografia 8](#_Toc132964026)

# Introdução

Este trabalho foi realizado no âmbito da Unidade Curricular Conhecimento e Raciocínio e tem como objetivo o desenvolvimento e estudo estatístico de redes neuronais para identificação de imagens de dígitos de 0 a 9 e operadores de adição, subtração, multiplicação e divisão.

O estudo é feito recorrendo a redes neuronais do tipo feedforward com alteração de diversos parâmetros para efeitos de comparação e teste.

# Decisões tomadas

## Tratamento das Imagens

Para proceder ao tratamento das imagens primeiro tivemos de definir os patos para as poder aceder. Depois de ter o path para as pastas, percorremos todas as imagens dentro delas e para cada uma, a fim de otimizar a leitura das imagens definimos um tamanho de 25x25. De seguida procede-se à sua transformação em matrizes binárias para que possamos obter os respetivos targets e inputs para essa matriz binária (imagem).

Todos os ficheiros além do trainFunction\_d, que corresponde à alínea d), recorrem a dois for loops para percorrer o número de pastas e o número de ficheiros. Para a alínea d) como recorre a uma funcionalidade que apenas requer a leitura de uma imagem separada, criamos outro ficheiro para retornar a matriz binaria e o target de apenas essa imagem em específico.

## Targets e Inputs

No fim de ter obtido todas as matrizes binarias das imagens, são gerados 14 vetores em que cada um deles corresponde a uma pasta. Esta atribuição é feita pela ordem de leitura das pastas, ou seja, são 14 vetores que correspondem às pastas dos dígitos de 0 a 9, add, div, mul e sub.

## Guardar Net’s

Para cada alínea a rede é treinada 10 vezes para que se possa obter uma média do valor global de precisão e do valor de precisão teste. Com esses valores comparamos as redes ao longo das execuções e vamos sempre guardando a rede com melhores valores para depois ser guardada dentro da pasta ‘networks’.

# Treino e Estudo Estatístico

## Alínea a)

Esta alínea testa o dataset da pasta ‘start’ que é composta por 5 imagens de cada dígito e operador.

Uma imagem com mesa

Descrição gerada automaticamente

Como se pode ver, o número de neurónios e camadas escondidas tem um papel importante neste teste, visto que com o aumento dos mesmos, os valores de precisão também tiveram um aumento.

Para este teste o melhor resultado foi com 3 camadas escondidas e 10 neurónios em cada uma delas, obtendo assim uma média de 75.14% de precisão global e uma média de 67.27% de precisão de teste após as 10 execuções.

Apesar de termos obtido um valor aceitável para o teste anterior, não foi onde obtivemos o melhor desempenho de todos.

Uma imagem com mesa

Descrição gerada automaticamente

Através da alteração dos parâmetros de divisão conseguimos obter valores mais aceitáveis para a precisão global.

Para este teste o melhor resultado obtido foi de 85.71% para a precisão global e 67.1% para a precisão de teste.

Agora onde obtivemos de facto o melhor resultado para ambas a precisões foi nos testes das funções de ativação.

Uma imagem com mesa

Descrição gerada automaticamente

A combinação da função logsig com a purelin e apenas 1 camada com 10 neurónios produziu um resultado com precisão global de 90% e precisão de teste de 85.45%.

Ao alterar as funções de treino foi onde obtivemos os piores tempos de execução, sendo extremamente demorados e a atingirem quase sempre 1000 épocas nas primeiras iterações para as funções traingd e trainbfg. Para além do tempo de execução elevado os resultados foram muito baixos comparando com a função trainlm que demonstrou uma melhor aprendizagem.

Uma imagem com mesa

Descrição gerada automaticamente

## Alínea b)

### *Uma só Net*

Esta alínea testa o dataset da pasta ‘train que é composta por 50 imagens de cada dígito e operador.

Como se pode ver, à semelhança da alínea anterior o número de neurónios e camadas escondidas tem um papel importante neste teste, visto que com o aumento dos mesmos, os valores de precisão também tiveram um aumento.

Uma imagem com mesa

Descrição gerada automaticamente

Para este teste o melhor resultado foi com 3 camadas escondidas e 10 neurónios em cada uma delas, obtendo assim uma média de 86.19% de precisão global e uma média de 83.62% de precisão de teste após as 10 execuções.

Estes valores são significativamente melhores do que os obtidos no teste da alínea anterior.

Para as funções treinos os resultados obtidos foram semelhantes aos da alínea anterior sendo que o tempo de execução foi extremamente elevado para as funções traingd e trainbfg e, em geral, os resultados foram muito baixos.

Uma imagem com mesa

Descrição gerada automaticamente

Mais uma vez a combinação de funções de ativação que se destacou foi a logsig e purelin, que permitiram obter uma precisão global de 84.4% e precisão de teste de 80.29%, que apesar de ser um valor bom não foi o melhor resultado obtido.

Uma imagem com mesa

Descrição gerada automaticamente

Também testamos o desempenho dos testes ao colocar o parâmetro de validação a 0 e os resultados para a função traingd foram melhores tanto em precisão como em tempo de execução, demorando muito menos para obter os resultados.

### Uma imagem com mesa Descrição gerada automaticamente

Por fim, a alteração dos parâmetros de divisão também demonstrou uma grande influência nos resultados, sendo o melhor resultado obtido 85.14% para a precisão global e 80.86% para a precisão de teste.

Uma imagem com mesa

Descrição gerada automaticamente

Para este dataset o melhor resultado obtido foi através do teste com 3 camadas escondidas e 10 neurónios em cada uma delas, com os parâmetros de divisão train=0.7, val=0.15 e test=0.15, com recurso às funções de ativação tansig para a primeira camada, tansig para a segunda camada, purelin para a ultima camada e trainln como função de treino.

### *Nets Separadas*

Ao alterar o número de camadas escondidas e neurónios podemos atingir resultados perto de 100% para a precisão global, sendo que o melhor teste obteve resultados de 97.5% para a precisão global e 16.67% para a precisão de teste para a rede dos dígitos e, 97.5% de precisão global e 23.33% de precisão de teste para a rede dos operadores.

Uma imagem com mesa

Descrição gerada automaticamente

Como nas alíneas anteriores, tempo de execução e precisão dos testes muito prejudicada para a rede dos dígitos. Para a rede dos operadores os resultados foram muito bons comparando com as redes das alíneas anteriores.

Uma imagem com mesa

Descrição gerada automaticamente

Ao alterar as funções de ativação foi novamente com a combinação da função logsig com a purelin que obtivemos os melhores resultados para ambas as redes neste teste, sendo que para a rede dos dígitos obtivemos uma precisão global de 93.58% e precisão de teste de 91.07%, enquanto para a rede dos operadores obtivemos 97.9% na precisão global e apenas 13.33% na precisão de teste.

Uma imagem com mesa

Descrição gerada automaticamente

Para a alteração dos parâmetros de divisão de treino também obtivemos bons resultados em ambas as redes sendo que a rede dos dígitos obteve resultados muito bons em quase todos os testes.

Uma imagem com mesa

Descrição gerada automaticamente

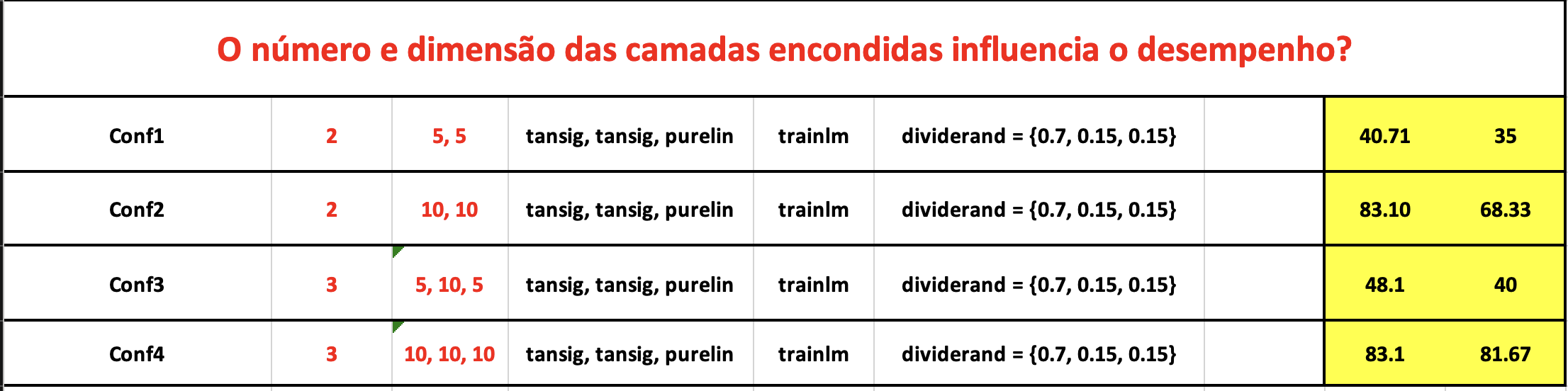
Em geral, a rede dos operadores divido a ser uma rede com uma complexidade de aprendizagem menor, obteve muitos bons resultados em todos os testes, enquanto a rede dos dígitos devido a ter um maior numero de imagens acabou por não ter um desempenho igual, mas comparando com o teste em que foi feito tudo na mesma rede, os resultados ao separar foram muito melhores.

## Alínea c)

Esta alínea testa o dataset da pasta ‘custom\_draw que é composta por 3 imagens de cada dígito e operador.

Este dataset foi criada através da nossa GUI em que fomos nós que utilizados a funcionalidade de desenhar para criar as imagens.

Ao alterar a dimensão das camadas escondidas e o número de neurónios por cada percebemos que os resultados foram muito satisfatórios para duas camadas com 10 neurónios em cada uma, e 3 camadas com 10 neurónios em cada uma, obtendo em ambos uma média de precisão global de 83.10%.



À semelhança das alíneas anterior, tempo de execução muito elevado para as funções traingd e trainbfg, porém para as restantes funções obtivemos melhores resultados do que nos outros datasets.

Uma imagem com mesa

Descrição gerada automaticamente

Os testes de função de ativação mostram um comportamento semelhante aos testes feitos com outros datasets em que a combinação da função logsig e purelin é que demonstra melhores resultados, obtendo uma média de precisão global de 87.62% e 86.67% para a média de precisão de teste.

Uma imagem com mesa

Descrição gerada automaticamente

Na divisão de exemplos o que teste que se voltou a destacar foi colocando o valor train=0.9 e os restantes a 0.05, obtendo uma média de precisão global de 89.05% e 75% para a média de precisão de teste, sendo este o teste que permitiu uma maior precisão global para este dataset.

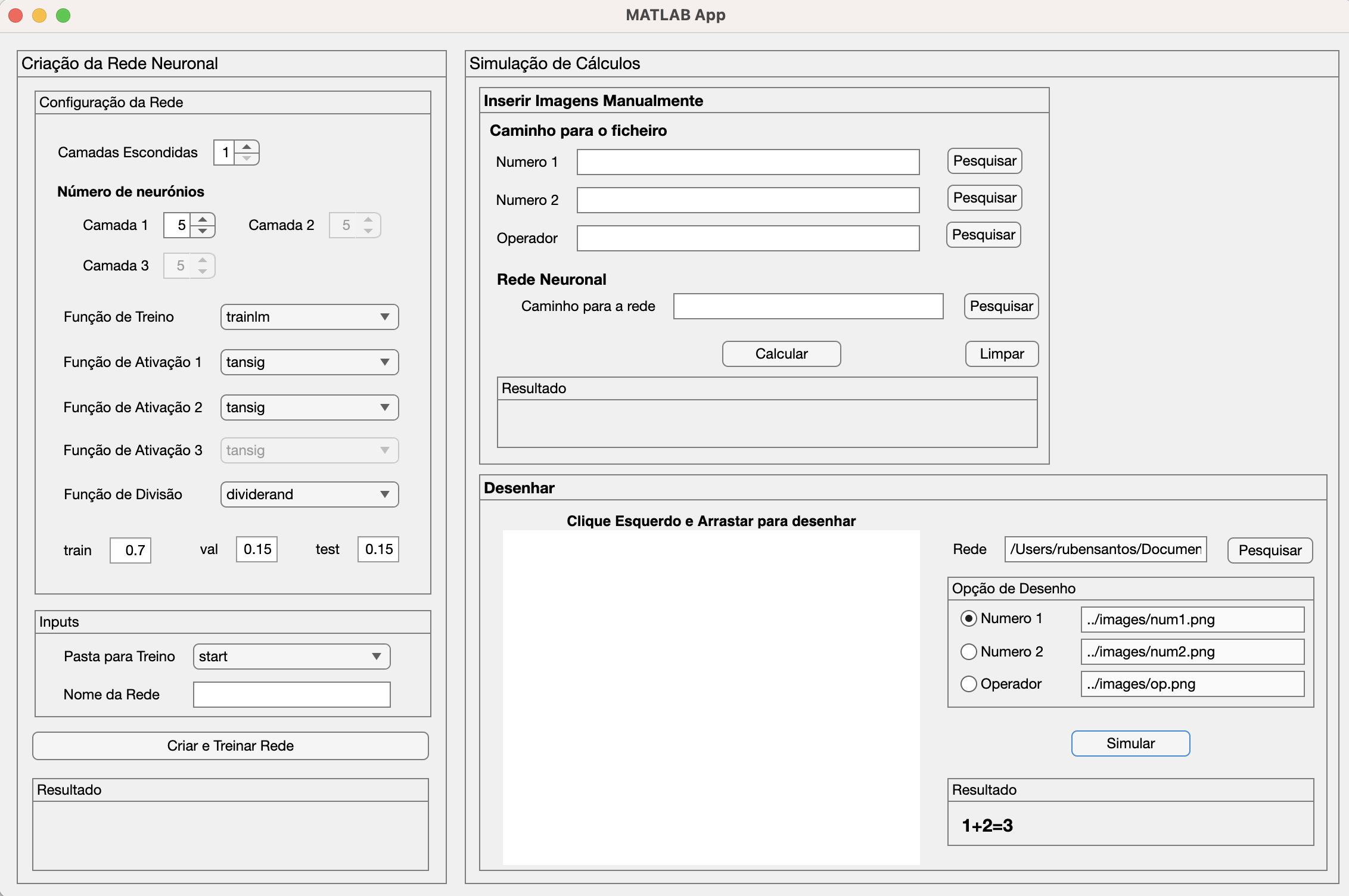
Uma imagem com mesa

Descrição gerada automaticamente

# Aplicação

A nossa aplicação é distribuída em 3 partes, sendo elas:

* Criar uma rede neuronal
  + No painel ‘Criação de Rede Neuronal’ pode criar uma rede neuronal com os parâmetros que pretender, dar um nome à rede e selecionar qual dataset quer treinar (start, train ou custom\_draw).
* Simular um cálculo de forma manual ao escolher as imagens
  + No painel ‘Inserir Imagens Manualmente’ pode pesquisar no seu computador, ou nos datasets fornecidos, uma imagem para o número 1, número 2 e operador. Depois basta selecionar qual a rede que pretende para avaliar as imagens.
  + O output será mostrado no painel ‘Resultado’, em que a ordem é: numero1 operador numero2 = resultado da operação.
* Simular um cálculo ao desenhar os dígitos
  + À semelhança de introduzir as imagens manualmente, no painel ‘Desenhar’ é o utilizador que desenha no quadrado branco ao pressionar o botão esquerdo do rato e arrastar para criar uma forma. Para escolher qual dígito se pretende desenhar basta selecionar o radio button correspondente (Número 1, Número 2 ou Operador). Também deve selecionar qual rede pretende usar.
  + O output será mostrado no painel ‘Resultado’, em que a ordem é: numero1 operador numero2 = resultado da operação.



# Conclusões

Com este trabalho prático podemos concluir que não há apenas uma parametrização predefinida que obtenha os melhores resultados. Para determinados datasets o melhor é testar vários tipos de parametrização e tirar as conclusões a partir daí.

Concluímos que o trabalho sobre o tema redes neuronais foi terminado com sucesso ao conseguir executar todas as tarefas pedidas no enunciado e com isso evoluímos o nosso conhecimento sobre redes neuronais.

# Bibliografia