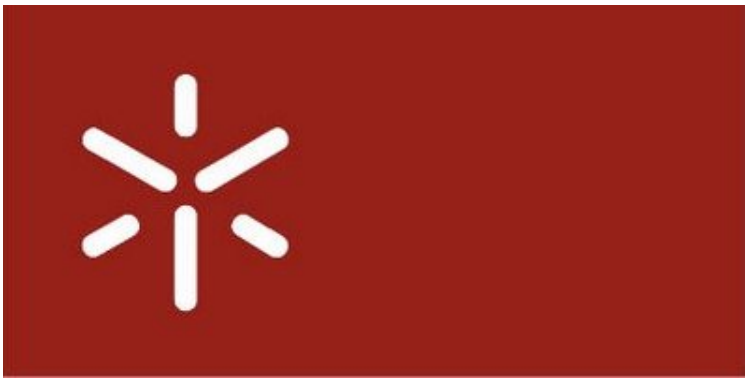


Universidade do Minho

Mestrado Integrado de Engenharia Informática

Sistemas de Representação Conhecimento e Raciocínio

Exercício 3 - Conhecimento não simbólico: Redes Neuronais Artificiais

Trabalho realizado por:

Adriana Guedes	A74545
Guilherme Guerreiro	A73860
Marco Barbosa	A75278
Ricardo Certo	A75315

GRUPO 1

21 de Maio de 2017

Resumo

Este trabalho foi realizado no âmbito da Unidade Curricular de Sistemas de Representação de Conhecimento e Raciocínio e consiste na utilização de Redes Neuronais Artificiais (RNA) para a identificação do nível de exaustão num Ser-Humano tendo em conta métricas relativas á utilização do teclado e do rato do computador.

O trabalho consiste na utilização de RNA's para a identificação de diferentes níveis de exaustão de um Ser Humano tendo em conta métricas relativas á sua interação com um computador. As métricas utilizadas neste estudo são as seguintes:

- "**Performance.KDTMean**" – tempo médio entre o momento em que a tecla é pressionada para baixo e o momento em que é largada;
- "**Performance.MAMean**" – aceleração do manuseamento rato em determinado momento. O valor da aceleração é calculado através da velocidade do rato (pixel/milissegundos) sobre o tempo de movimento (milissegundos);
- "**Performance.MVMean**" – velocidade do manuseamento do rato em determinado momento. A distância percorrida pelo rato (em píxeis) entre uma coordenada C1 (x1; y1) e uma C2 (x2; y2) correspondentes a time1 e time2, sobre o tempo (em milissegundos);
- "**Performance.TBCMean**" – tempo entre dois *clicks* consecutivos, entre eventos consecutivos MOUSE_UP e MOUSE_DOWN;
- "**Performance.DDCMean**" – período de tempo entre dois eventos MOUSE_UP consecutivos;
- "**Performance.DMSMean**" – distância média em excesso entre o caminho de dois *clicks* consecutivos;
- "**Performance.ADMSLMean**" – distância média das diferentes posições do ponteiro entre dois pontos durante um movimento, e o caminho em linha reta entre esses mesmos dois pontos;
- "**Performance.AEDMean**" – esta métrica é semelhante à anterior, no sentido em calculará a soma da distância entre dois eventos MOUSE_UP e MOUSE_DOWN consecutivos;
- "**ExhaustionLevel**" – nível subjetivo de exaustão mental;
- **"Performance.Task"** – identificação da tarefa em execução no momento da recolha dos dados.

Para possibilitar a realização deste trabalho foi-nos fornecido um ficheiro exaustao.csv que continha dados respetivos às métricas referidas em cima. Os dados recebidos contém 844 observações que podem ser usadas para o treino da nossa RNA.

Conteúdo

1 Introdução

O trabalho prático descrito neste relatório consiste na utilização de uma RNA (Rede Neuronal Artificial) com objetivo de identificar diferentes níveis de exaustão de acordo com uma fórmula definida previamente, fórmula esta que utiliza como parâmetros métricas relativas à utilização do rato e do teclado do computador.

A linguagem que utilizamos para implementar todo este trabalho foi a linguagem **R** , e para isso utilizados o RStudio como ferramenta.

Ao longo deste relatório vamos apresentar todas as decisões tomadas pelo grupo e todas as conclusões consoante os resultados obtidos.

2 Preliminares

De forma a conseguirmos realizar tudo o que nos é proposto no enunciado, com base nas aulas da Unidade Curricular de Sistemas de Representação de Conhecimento e Raciocínio, temos de possuir conhecimentos sobre RNA's de forma a aplicarmos da melhor maneira o conhecimento adquirido e obter as respetivas conclusões sobre os novos dados. Foi ainda necessário adquirirmos bases sobre a linguagem de programação R.

2.1 Redes Neuronais Artificiais

2.2 O que são RNA's ?

RNA's são um conceito da computação que visa trabalhar no processamento de dados de maneira parecida com o cérebro humano. O cérebro funciona como um processador extremamente complexo e que têm a capacidade de realizar processamentos em paralelo a uma velocidade extremamente alto sendo que ainda não existe nenhum computador no mundo capaz de realizar o que o cérebro humano faz.

Nas RNA's, a ideia é realizar o processamento de informações tendo em conta a organização dos neurónios do cérebro. Tal como o cérebro as RNA's devem ser capazes de aprender e tomar decisões.

Assim, uma RNA pode ser vista como um esquema de processamento capaz de armazenar conhecimento baseado em aprendizagem/experiência.

2.2.1 Funcionamento

As RNA's são criadas a partir de algoritmos projetados para um determinado fim. É impossível criar um algoritmo desse sem ter conhecimento de modelos matemáticos capazes de simular o processo de aprendizagem do cérebro humano.

Basicamente, uma RNA assemelha-se ao cérebro em dois pontos: o conhecimento é obtido através de etapas de aprendizagem e pesos sinápticos são usados para armazenar o conhecimento. Podemos então afirmar que as RNA's têm presentes na sua constituição uma série de neurónios artificiais que serão conectados entre si, formando uma rede com elementos de processamento.

Existem várias formas de se desenvolver uma rede neuronal artificial. Ela deve ser montada de acordo com o problema a ser resolvido.Na sua arquitetura são determinados o número de camadas usadas (as camadas são formadas por neurônios), a quantidade de neurônios em cada camada, o tipo de sinapse utilizado, etc.

2.2.2 Aplicações

As RNA's podem ser aplicadas para resolver um grande número de problemas. Um bom exemplo desta aplicação são os softwares de reconhecimento de voz, que precisam de aprender a conhecer a voz de determinadas pessoas, também são usadas em robôs que desarmam bombas. Mas no geral as RNA's são usadas principalmente em aplicações mais complexas como o mercado financeiro, etc.

3 Normalização e Análise de Dados

3.1 Identificação dos Níveis de fadiga

O primeiro problema que nos foi proposto no enunciado foi que treinássemos a rede neuronal de forma a esta conseguir identificar os 7 níveis de fadiga distintos com a maior precisão possível.

Os níveis de fadiga predefinidos para identificação são os seguintes:

1. Totalmente bem;
2. Responsivo, mas não no pico;
3. Ok, normal;
4. Em baixo de forma/do normal, a sentir-se em baixo;
5. Sentido moleza, perdendo o foco;
6. Muito difícil concentrar, meio tonto;
7. Incapaz de funcionar, pronto a “desligar”.

Para responder a este problema o primeiro passo seria treinar a rede neuronal com alguns dos casos dos quais possuíamos dados.

3.2 Normalização

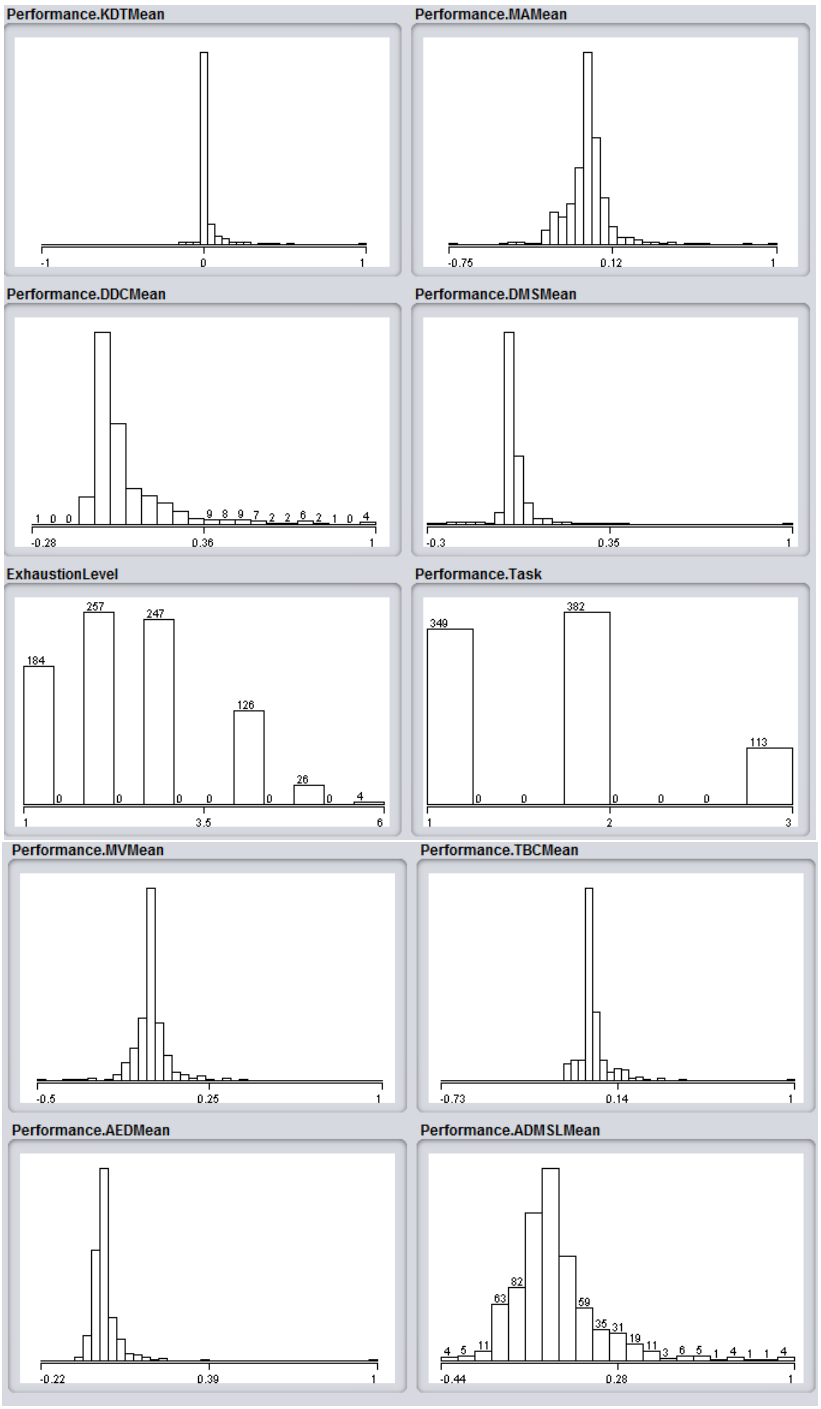
Tivemos a necessidade de normalizar os valores do ficheiro que nos foi fornecido relativo às biométricas utilizadas para a deteção de exaustão.

Os primeiro 8 parâmetros estão definidos num intervalo entre -1 e 1 , o nível de exaustão possuí valores entre 1 e 7 e o último parâmetro está definido por 3 strings sendo estas ”Work”, ”Oficce”e ”Programming”. Neste último parâmetro tivemos a necessidade de normalizar estas strings para valores inteiros, assim a string ”Work”vai tomar o valor 1 , a ”Office”vai tomar o valor 2 e por fim a ”Programming”vai tomar o valor 3. Escolhemos os valores 1, 2 e 3 pelo facto se escolhêssemos o 0, 1 e 2 poderíamos vir a ter problemas devido ao valor lógico e matemático que 0 representa.

Depois deste processo ainda tivemos necessidade de baralhar os dados fornecidos uma vez que os dados vinham agrupados pela sua string, pois se não realizasse-mos isso iríamos possuir alguns problemas no que toca à representação de resultados, pois ao dividirmos os dados de input em treino e em teste, as amostras dos dados não iriam ser similares em termos de consistência do número de amostras de cada tipo de strings.

3.3 Análise de Dados

Os dados fornecidos vão-nos servir de base para a realização de um estudo que envolve a identificação do nível de exaustão e respetiva tarefa em execução de uma dada pessoa. Antes de realizarmos qualquer processo fizemos um breve estudo sobre os dados fornecidos com o intuito de percebermos melhor o problema em questão. Na figura seguinte podemos verificar a forma de como os nosso dados variam.



Depois de analisar os dados o grupo em consenso chegou á decisão de dividir o ficheiro em dois sets, um para o treino da amostra inicial e outro para teste. A amostra ficou dividida na seguinte forma:

```
#divisão dos dados para teste e treino
```

```
trainset <- dataset[1:600,]
testset <- dataset[601:844,]
```

4 Níveis de Exaustão

4.1 Importância dos Atributos

Num momento anterior foi realizado um estudo á cerca da relevância que cada atributo poderia ter na resolução do problema em questão. Numa primeira fase consideramos que todos os atributos iriam ter a mesma relevância e daí resultou a seguinte fórmula:

```
formula <- ExhaustionLevel ~
    Performance.KDTMean + Performance.MAMean +
    Performance.MVMean + Performance.TBCMean +
    Performance.DDCMean + Performance.DMSMean +
    Performance.AEDMean + Performance.ADMSLMean
```

De seguida podemos observar na imagem a respetiva importância de cada atributo em relação ao nível de exaustão:

```
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         Performance.KDTMean Performance.MAMean Performance.MVMean Performance.TBCMean Performance.DDCMean
1  ( 1 ) " "                 " "                " "                " "                "*"
2  ( 1 ) " "                 "*"                " "                " "                "*"
3  ( 1 ) " "                 "*"                "*"                " "                "*"
4  ( 1 ) "*"                 "*"                "*"                " "                "*"
5  ( 1 ) "*"                 "*"                "*"                " "                "*"
6  ( 1 ) "*"                 "*"                "*"                " "                "*"
7  ( 1 ) "*"                 "*"                "*"                "*"                "*"
8  ( 1 ) "*"                 "*"                "*"                "*"                "*"
         Performance.DMSMean Performance.AEDMean Performance.ADMSLMean
1  ( 1 ) " "                 " "                 " "
2  ( 1 ) " "                 " "                 " "
3  ( 1 ) " "                 " "                 " "
4  ( 1 ) " "                 " "                 " "
5  ( 1 ) "*"                 " "                 " "
6  ( 1 ) "*"                 " "                 "*"
7  ( 1 ) "*"                 " "                 "*"
8  ( 1 ) "*"                 "*"                 "*"
>
```

Figura 1: Importância de cada atributo em relação ao nível de exaustão

Por fim podemos observar a fórmula que diz respeito à tarefa em execução e a respetiva imagem que possui a importância de cada atributo:

```
formula <- Performance.Task ~
            Performance.KDTMean + Performance.MAMean +
            Performance.MVMean + Performance.TBCMean +
            Performance.DDCMean + Performance.DMSMean +
            Performance.AEDMean + Performance.ADMSLMean
```

```
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         Performance.KDTMean Performance.MAMean Performance.MVMean Performance.TBCMean Performance.DDCMean
1  ( 1 ) "*"                 " "                 " "                 " "                 " "
2  ( 1 ) "*"                 " "                 " "                 " "                 " "
3  ( 1 ) "*"                 " "                 " "                 " "                 "*"
4  ( 1 ) "*"                 " "                 " "                 " "                 "*"
5  ( 1 ) "*"                 " "                 " "                 " "                 "*"
6  ( 1 ) "*"                 " "                 " "                 "*"                 "*"
7  ( 1 ) "*"                 " "                 "*"                 "*"                 "*"
8  ( 1 ) "*"                 "*"                 "*"                 "*"                 "*"
         Performance.DMSMean Performance.AEDMean Performance.ADMSLMean
1  ( 1 ) " "                 " "                 " "
2  ( 1 ) "*"                 " "                 " "
3  ( 1 ) "*"                 " "                 " "
4  ( 1 ) "*"                 "*"                 " "
5  ( 1 ) "*"                 "*"                 "*"
6  ( 1 ) "*"                 "*"                 "*"
7  ( 1 ) "*"                 "*"                 "*"
8  ( 1 ) "*"                 "*"                 "*"
```

Figura 2: Importância de cada atributo em relação à tarefa em execução

Como podemos observar a importância de um atributo nas fórmulas depende daquilo que estamos a calcular nesse preciso momento. Pois um atributo pode desempenhar um papel importante no cálculo de um fórmula e noutra ser o atributo menos importante.

4.2 Fórmulas

Depois de tidos em conta os resultados obtidos decidimos elaborar as seguintes fórmulas em que o número de atributos presentes varia tendo sempre em consideração o grau de importância de cada atributo. Depois de efetuarmos alguns testes achamos por bem fazer 5 fórmulas para cada um , sendo que uma delas possuí todos os atributos e as outras variam de 3 a 6 no número de atributos.

- Nível de Exaustão

```
formula <- ExhaustionLevel ~
 Performance.KDTMean +
 Performance.MAMean +
 Performance.MVMean +
 Performance.TBCMean +
 Performance.DDCMean +
 Performance.DMSMean +
 Performance.AEDMean +
 Performance.ADMSLMean


formula3 <- ExhaustionLevel ~
 Performance.MAMean +
 Performance.MVMean +
 Performance.DDCMean


formula4 <- ExhaustionLevel ~
 Performance.KDTMean +
 Performance.MAMean +
 Performance.MVMean +
 Performance.DDCMean
```

```
formula5 <- ExhaustionLevel ~
 Performance.KDTMean +
 Performance.MAMean +
 Performance.MVMean +
 Performance.DDCMean +
 Performance.DMSMean


formula6 <- ExhaustionLevel ~
 Performance.KDTMean +
 Performance.MAMean +
 Performance.MVMean +
 Performance.DDCMean +
 Performance.DMSMean +
 Performance.ADMSLMean
```

- Tarefa em execução

```
formula <- Performance.Task ~
 Performance.KDTMean + Performance.MAMean +
 Performance.MVMean + Performance.TBCMean +
 Performance.DDCMean + Performance.DMSMean +
 Performance.AEDMean + Performance.ADMSLMean


formula3 <- Performance.Task ~
 Performance.KDTMean +
 Performance.DDCMean +
 Performance.DMSMean


formula4 <- Performance.Task ~
 Performance.KDTMean +
 Performance.DDCMean +
 Performance.DMSMean +
 Performance.AEDMean


formula5 <- Performance.Task ~
 Performance.KDTMean +
 Performance.DDCMean +
 Performance.DMSMean +
 Performance.AEDMean +
 Performance.ADMSLMean


formula6 <- Performance.Task ~
 Performance.KDTMean +
```

```
Performance.DDCMean +
Performance.DMSMean +
Performance.AEDMean +
Performance.ADMSLMean +
Performance.TBCMean
```

4.3 Treino

4.3.1 Neuralnet

Para conseguirmos realizar o treino da RNA usamos a função neuralnet:

neuralnet(formula,data,hidden,threshold, algoritmo, lifesign, linear.output)

Os respetivos parâmetros são:

- formula : formula usada para treinar a rede
- data : dataset responsável por conter as variáveis da fórmula
- hidden : um valor que define quantos nodos escondidos vamos possuir
- threshold : valor de erro que é responsável por fazer a função parar a sua execução
- algoritmo : especificação de um algoritmo para a realização do treino
- lifesign : especifica o que vai ser impresso durante a execução
- linear.output : Booleano que especifica a utilização de nodos exteriores

4.4 Sets

Para a realização dos testes após treinarmos a rede precisamos de definir sets de modo em que as colunas dos atributos que vamos utilizar na fórmula estejam disponíveis. Definimos 5 subsets tanto para o nível de exaustão como para a tarefa em execução. Os sets são:

- Nível de Exaustão

```
vartest <- subset(testset, select = c("Performance.KDTMean","Performance.MAMean",
   "Performance.MVMean","Performance.TBCMean","Performance.DDCMean",
   "Performance.DMSMean","Performance.AEDMean","Performance.ADMSLMean"))


vartest3 <- subset(testset, select = c("Performance.MAMean" ,   "Performance.MVMean",
   "Performance.DDCMean"))


vartest4 <- subset(testset, select = c("Performance.MAMean" , "Performance.MVMean",
   "Performance.DDCMean","Performance.KDTMean"))


vartest5 <- subset(testset, select = c("Performance.MAMean" , "Performance.MVMean",
   "Performance.DDCMean",
   "Performance.KDTMean" ,
```

```
   "Performance.DMSMean"))


vartest6 <- subset(testset, select = c("Performance.MAMean" , "Performance.MVMean",
   "Performance.DDCMean" , "Performance.KDTMean" ,
   "Performance.DMSMean" , "Performance.ADMSLMean"))
```

- Tarefa em execução

```
vartest <- subset(testset, select = c("Performance.KDTMean","Performance.MAMean",
   "Performance.MVMean","Performance.TBCMean",
   "Performance.DDCMean","Performance.DMSMean",
   "Performance.AEDMean","Performance.ADMSLMean"))


vartest3 <- subset(testset, select = c("Performance.KDTMean" , "Performance.DDCMean",
   "Performance.DMSMean"))


vartest4 <- subset(testset, select = c("Performance.KDTMean" , "Performance.DDCMean",
   "Performance.DMSMean" , "Performance.AEDMean"))


vartest5 <- subset(testset, select = c("Performance.KDTMean" , "Performance.DDCMean" ,
   "Performance.DMSMean" , "Performance.AEDMean" ,
   "Performance.ADMSLMean"))


vartest6 <- subset(testset, select = c("Performance.KDTMean" , "Performance.DDCMean" ,
   "Performance.DMSMean" , "Performance.AEDMean" ,
   "Performance.ADMSLMean" , "Performance.TBCMean"))
```

4.5 Testes

Nesta primeira parte os testes foram realizados em relação aos 7 níveis de fadiga e a tarefa em execução relativamente a esses níveis. De seguida, vamos encontrar as duas tabelas que dizem respeito ao processo de treino das RNA utilizando sempre paramêtros diferentes.

4.5.1 Níveis de Exaustão

Nº treino	Fórmula	Rede Neuronal	Threshold	Steps	Erro	Algoritmo	RMSE
1	formula	(10)	0.1	21675	209.74	slr	1.26580
2	formula	(2,4)	0.1	30943	266.06	slr	1.23310
3	formula3	(4,8)	0.1	62472	261.93	sag	1.18896
4	formula3	(10)	0.1	6163	285.72	rprop+	1.21783
5	formula3	(4,8)	0.5	757	303.68	sag	1.12961
6	formula4	(12,2)	0.1	——-	———	rprop+	1.40636
7	formula4	(12,2)	0.1	——-	———	sag	1.38396
8	formula5	(2,4)	0.05	7337	296.46	rprop-	1.23326

Tabela 1: Tabela de Treino e RMSE - Nível de Exaustão

4.5.2 Tarefa em Execução

Nº treino	Fórmula	Rede Neuronal	Threshold	Steps	Erro	Algoritmo	RMSE
1	formula	(10)	0.1	9105	58.270	slr	0.53378
2	formula	(2,4)	0.1	647	110.63	slr	0.60459
3	formula3	(4,8)	0.1	28	139.69	sag	0.66245
4	formula3	(5)	0.1	209	138.90	rprop+	0.66194
5	formula3	(2,2,4)	0.5	17	144.14	sag	0.66585
6	formula4	(10,2)	0.1	8346	88.43	rprop+	0.62512
7	formula4	(10,2)	0.5	155	132.04	sag	0.66164
8	formula5	(2,4)	0.05	1402	132.8	rprop-	0.66860

Tabela 2: Tabela de Treino e RMSE - Tarefa em execução

5 Existência ou Ausência de Exaustão

Nesta segunda fase do trabalho prático é-nos proposto que alteremos a escala de exaustão passando dos 7 níveis iniciais para apenas 2. Estes dois níveis vão representar a ausência de exaustão (1) e a existência exaustão (2). Os níveis 1,2 e 3 representam a ausência de exaustão, por outro lado o 4,5,6 e 7 representam a existência de exaustão. Sendo assim , tivemos que efetuar uma nova construção da nossa RNA.

Usando o script concebido na linguagem R , efetuamos a transformação para apenas considerarmos 2 níveis de fadiga. Esta transformação é efetuada recorrendo aos seguintes comandos:

```
#transformar valores de exaustão em 1(não exausto) ou 2(exausto)
  dataset2$ExhaustionLevel[dataset2$ExhaustionLevel <= 3] <- 1
  dataset2$ExhaustionLevel[dataset2$ExhaustionLevel > 3] <- 2
```

5.1 Importância dos Atributos

Num momento anterior foi realizado um estudo á cerca da relevância que cada atributo poderia ter na resolução do problema em questão. Numa primeira fase consideramos que todos os atributos iriam ter a mesma relevância e daí resultou a seguinte fórmula:

```
form <- ExhaustionLevel ~
    Performance.KDTMean + Performance.MAMean +
    Performance.MVMean + Performance.TBCMean +
    Performance.DDCMean + Performance.DMSMean +
    Performance.AEDMean + Performance.ADMSLMean
```

De seguida podemos observar na imagem a respetiva importância de cada atributo em relação ao nível de exaustão.

```
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         Performance.KDTMean Performance.MAMean Performance.MVMean Performance.TBCMean Performance.DDCMean Performance.DMSMean Performance.AEDMean
1  ( 1 ) "*"                 " "                " "                " "                " "                " "                " "
2  ( 1 ) "*"                 " "                " "                "*"                " "                " "                " "
3  ( 1 ) "*"                 " "                " "                " "                " "                "*"                " "
4  ( 1 ) "*"                 " "                " "                "*"                " "                "*"                " "
5  ( 1 ) "*"                 " "                " "                "*"                "*"                "*"                " "
6  ( 1 ) "*"                 "*"                "*"                "*"                " "                "*"                " "
7  ( 1 ) "*"                 "*"                "*"                "*"                "*"                "*"                " "
8  ( 1 ) "*"                 "*"                "*"                "*"                "*"                "*"                "*"
         Performance.ADMSLMean
1  ( 1 ) " "
2  ( 1 ) " "
3  ( 1 ) "*"
4  ( 1 ) "*"
5  ( 1 ) "*"
6  ( 1 ) "*"
7  ( 1 ) "*"
8  ( 1 ) "*"
```

5.2 Fórmulas

Depois de tidos em conta os resultados obtidos decidimos elaborar as seguintes fórmulas em que o número de atributos presentes varia tendo sempre em consideração o grau de importância de cada atributo. Depois de efetuarmos alguns testes achamos por bem fazer 5 fórmulas, sendo que uma delas possuí todos os atributos e as outras variam de 3 a 6 no número de atributos.

```
form <- ExhaustionLevel ~ Performance.KDTMean + Performance.MAMean +
   Performance.MVMean + Performance.TBCMean +
   Performance.DDCMean + Performance.DMSMean +
   Performance.AEDMean + Performance.ADMSLMean


form3 <- ExhaustionLevel ~ Performance.KDTMean + Performance.DMSMean +
       Performance.ADMSLMean


form4 <- ExhaustionLevel ~ Performance.KDTMean + Performance.DMSMean +
         Performance.ADMSLMean + Performance.TBCMean


form5 <- ExhaustionLevel ~ Performance.KDTMean + Performance.DMSMean +
        Performance.ADMSLMean + Performance.TBCMean +
        Performance.DDCMean


form6 <- ExhaustionLevel ~ Performance.KDTMean + Performance.DMSMean +
     Performance.ADMSLMean + Performance.TBCMean +
     Performance.DDCMean + Performance.MAMean
```

5.3 Treino

5.3.1 Neuralnet

Para conseguirmos realizar o treino da RNA usamos a função neuralnet:

neuralnet(formula,data,hidden,threshold, algoritmo, lifesign, linear.output)

Os respetivos parammetros são:

- formula : formula usada para treinar a rede
- data : dataset responsável por conter as váriaveis da formula
- hidden : um valor que define quantos nodos escondidos vamos possuir
- threshold : valor de erro que é responsável por fazer a função parar a sua execução
- algoritmo : especificação de um algoritmo para a realização do treino
- lifesign : especifica o que vai ser impresso durante a execução
- linear.output : Booleano que especifica a utilização de nodos exteriores

5.4 Sets

Para a realização dos testes após treinarmos a rede precisamos de definir sets de modo em que as colunas dos atributos que vamos utilizar na fórmula estejam disponíveis. Definimos 5 subsets para o nível de exaustão que são os seguintes:

```
varteste <- subset(testset2, select = c("Performance.KDTMean",
    "Performance.MAMean","Performance.MVMean","Performance.TBCMean",
    "Performance.DDCMean","Performance.DMSMean","Performance.AEDMean",
    "Performance.ADMSLMean"))


varteste3 <- subset(testset2, select = c("Performance.KDTMean" ,
     "Performance.DMSMean" , "Performance.ADMSLMean"))


varteste4 <- subset(testset2, select = c("Performance.KDTMean" ,
     "Performance.DMSMean" , "Performance.ADMSLMean" ,
     "Performance.TBCMean"))


varteste5 <- subset(testset2, select = c("Performance.KDTMean" ,
     "Performance.MAMean" , "Performance.MVMean" ,
     "Performance.DDCMean" , "Performance.DMSMean"))


varteste6 <- subset(testset2, select = C("Performance.KDTMean" ,
    "Performance.DMSMean" , "Performance.ADMSLMean" ,
    "Performance.TBCMean" , "Performance.DDCMean" ,
    "Performance.MAMean"))
```

5.5 Testes

Nesta segunda parte os testes foram realizados em relação aos 2 níveis de exaustão definidos anteriormente. De seguida, vamos encontrar uma tabela que diz respeito ao processo de treino das RNA utilizando sempre parâmetros diferentes.

5.5.1 Níveis de Exaustão

Nº treino	Fórmula	Rede Neuronal	Threshold	Steps	Erro	Algoritmo	RMSE
1	form	(10)	0.1	3224	34.69	slr	0.4572
2	form3	(2,4)	0.1	18	42.64	slr	0.4153
3	form3	(2,4)	0.1	66	42.11	sag	0.4122
4	form3	(2,4)	0.1	50	42.21	rprop+	0.4128
5	form3	(2,4)	0.1	24	42.66	rprop-	0.4146
6	form	(2,4)	0.08	1124	39.09	slr	0.4226
7	form4	(2,4,6)	0.01	63283	32.38	rprop+	0.4475
8	form4	(5)	0.5	20	42.29	sag	0.4128
9	form4	(5,2)	0.05	49	42.08	rprop-	0.4125
10	form5	(15,10)	0.1	——-	——	rprop-	0.4151

Tabela 3: Tabela de Treino e RMSE - Nível de Exaustão em 2 níveis

6 Melhor Escala de Identificação de Exaustão

Nesta terceira e última fase do trabalho prático é-nos proposto que implementemos uma nova escala de identificação da exaustão de modo a conseguirmos um menor erro possível. Como podemos verificar na análise de dados existem muito pouco atributos do quarto nível para cima inclusive , por isso decidimos agrupa-los todos num só nível e assim passamos a ficar com apenas 4 níveis em que agora o número de amostras de cada nível ficou mais equilibrado. Usando o script concebido na linguagem

R , efetuamos a transformação para apenas considerar-mos 4 níveis de fadiga. Esta transformação é efetuada recorrendo aos seguintes comandos:

```
dataset3$ExhaustionLevel[dataset3$ExhaustionLevel == 1] <- 1
dataset3$ExhaustionLevel[dataset3$ExhaustionLevel == 2] <- 2
dataset3$ExhaustionLevel[dataset3$ExhaustionLevel == 3] <- 3
dataset3$ExhaustionLevel[dataset3$ExhaustionLevel == 4] <- 4
dataset3$ExhaustionLevel[dataset3$ExhaustionLevel == 5] <- 4
dataset3$ExhaustionLevel[dataset3$ExhaustionLevel == 6] <- 4
dataset3$ExhaustionLevel[dataset3$ExhaustionLevel == 7] <- 4
```

6.1 Importância dos Atributos

Num momento anterior foi realizado um estudo á cerca da relevância que cada atributo poderia ter na resolução do problema em questão. Numa primeira fase consideramos que todos os atributos iriam ter a mesma relevância e daí resultou a seguinte fórmula:

```
ff <- ExhaustionLevel ~ Performance.KDTMean + Performance.MAMean +
      Performance.MVMean + Performance.TBCMean +
      Performance.DDCMean + Performance.DMSMean +
      Performance.AEDMean + Performance.ADMSLMean
```

De seguida podemos observar na imagem a respetiva importância de cada atributo em relação ao nível de exaustão.

```
Performance.ADMSLMean     FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         Performance.KDTMean Performance.MAMean Performance.MVMean Performance.TBCMean Performance.DDCMean Performance.DMSMean Performance.AEDMean
1  ( 1 ) " "                 " "                " "                " "                "*"                " "                " "               
2  ( 1 ) " "                 "*"                "*"                " "                " "                " "                " "               
3  ( 1 ) " "                 "*"                "*"                " "                "*"                " "                " "               
4  ( 1 ) "*"                 "*"                "*"                " "                "*"                " "                " "               
5  ( 1 ) "*"                 "*"                "*"                " "                "*"                "*"                " "               
6  ( 1 ) "*"                 "*"                "*"                " "                "*"                "*"                " "               
7  ( 1 ) "*"                 "*"                "*"                " "                "*"                "*"                "*"               
8  ( 1 ) "*"                 "*"                "*"                "*"                "*"                "*"                "*"               
         Performance.ADMSLMean
1  ( 1 ) " "                  
2  ( 1 ) " "                  
3  ( 1 ) " "                  
4  ( 1 ) " "                  
5  ( 1 ) " "                  
6  ( 1 ) "*"                  
7  ( 1 ) "*"                  
8  ( 1 ) "*"
```

6.2 Fórmulas

Depois de tidos em conta os resultados obtidos decidimos elaborar as seguintes fórmulas em que o número de atributos presentes varia tendo sempre em consideração o grau de importância de cada atributo. Depois de efetuarmos alguns testes achamos por bem fazer 5 fórmulas, sendo que uma delas possuí todos os atributos e as outras variam de 3 a 6 no número de atributos.

```
ff <- ExhaustionLevel ~ Performance.KDTMean + Performance.MAMean +
  Performance.MVMean + Performance.TBCMean +
  Performance.DDCMean + Performance.DMSMean +
  Performance.AEDMean + Performance.ADMSLMean


ff3 <- ExhaustionLevel ~ Performance.MAMean +
  Performance.MVMean + Performance.DDCMean


ff4 <- ExhaustionLevel ~ Performance.MAMean + Performance.MVMean +
  Performance.DDCMean + Performance.KDTMean


ff5 <- ExhaustionLevel ~ Performance.MAMean + Performance.MVMean +
  Performance.DDCMean + Performance.KDTMean +
  Performance.DMSMean


ff6 <- ExhaustionLevel ~ Performance.MAMean + Performance.MVMean +
  Performance.DDCMean + Performance.KDTMean +
  Performance.DMSMean + Performance.ADMSLMean
```

6.3 Treino

6.3.1 Neuralnet

Para conseguirmos realizar o treino da RNA usamos a função neuralnet:

neuralnet(formula,data,hidden,threshold, algoritmo, lifesign, linear.output)

Os respetivos parâmetros são:

- formula : formula usada para treinar a rede
- data : dataset responsável por conter as váriaveis da formula
- hidden : um valor que define quantos nodos escondidos vamos possuir
- threshold : valor de erro que é responsável por fazer a função parar a sua execução
- algoritmo : especificação de um algoritmo para a realização do treino
- lifesign : especifica o que vai ser impresso durante a execução
- linear.output : Booleano que especifica a utilização de nodos exteriores

6.4 Sets

Para a realização dos testes após treinarmos a rede precisamos de definir sets de modo em que as colunas dos atributos que vamos utilizar na fórmula estejam disponíveis. Definimos 5 subsets para o nível de exaustão que são os seguintes:

```
vartt <- subset(testset3, select = c("Performance.KDTMean","Performance.MAMean",
    "Performance.MVMean","Performance.TBCMean",
    "Performance.DDCMean","Performance.DMSMean",
    "Performance.AEDMean","Performance.ADMSLMean"))


 vartt3 <- subset(testset3, select = c("Performance.MAMean" , "Performance.MVMean" ,
     "Performance.DDCMean"))


 vartt4 <- subset(testset3, select = c("Performance.MAMean" , "Performance.MVMean" ,
     "Performance.DDCMean" , "Performance.KDTMean"))


 vartt5 <- subset(testset3, select = c("Performance.MAMean" , "Performance.MVMean" ,
      "Performance.DDCMean" , "Performance.KDTMean" ,
      "Performance.DMSMean"))


 vartt6 <- subset(testset3, select = c("Performance.MAMean" , "Performance.MVMean" ,
     "Performance.DDCMean" , "Performance.KDTMean" ,
     "Performance.DMSMean" , "Performance.ADMSLMean"))
```

6.5 Testes

Nesta terceira parte os testes foram realizados em relação aos 4 níveis de exaustão definidos anteriormente. De seguida, vamos encontrar uma tabela que diz respeito ao processo de treino das RNA utilizando sempre parâmetros diferentes.

6.5.1 Níveis de Exaustão

Nº treino	Fórmula	Rede Neuronal	Threshold	Steps	Erro	Algoritmo	RMSE
1	ff	(20,15)	0.1	——	——	rprop-	2.0540
2	ff	(2,2)	0.04	7722	249.75	slr	1.1200
3	ff3	(4,2)	0.08	13269	265.22	slr	1.0882
4	ff3	(10,5)	0.1	9935	207.62	rprop+	1.2211
5	ff4	(12,6)	0.1	——	——	rprop+	1.2427
6	ff4	(12,6)	0.1	——	——	rprop-	1.4000
7	ff4	(12,6)	0.1	——-	——	sag	1.2389
8	ff4	(12,6)	0.1	——	——	slr	1.2305
9	ff4	(4,3)	0.02	92024	244.72	slr	1.1213

Tabela 4: Tabela de Treino e RMSE - Nível de Exaustão em 4 níveis

7 Conclusão

A elaboração deste projeto funcionou como uma forma de consolidação da matéria lecionada ao longo das aulas desta Unidade Curricular, mais concretamente sobre o tema de Redes Neuronais. Assim, seria esperada a correta utilização de sistemas não simbólicos na representação do conhecimento através de RNA's e a sua otimização.

Posto isto, neste exercício foi necessário tratamos os dados fornecidos pelo docente normalizando os mesmos de forma a aumentarmos a eficiência e diminuirmos o RMSE, efetuamos os testes à nossa RNA necessários e os cálculos para chegarmos aos resultados apresentados no presente documento.

O grupo chegou à conclusão que todo o conceito de Redes Neuronais com o passar do tempo cada vez mais se vai tornando num conceito de extrema importância no mundo da computação, responsável pela solução de muitos problemas complexos.