

---

Sistemas de Representação de Conhecimento e Raciocínio

**Exercício 3**

Redes Neurais Artificiais

---

André Gonçalves  
A75625

Bruno Cancelinha  
A75428

José Silva  
A74576

Marcelo Miranda  
A74817

22 de Maio de 2017



**Universidade do Minho**

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Análise e Tratamento de Dados</b>	<b>2</b>
2.1	Biométricas . . . . .	2
2.2	Normalização . . . . .	3
2.3	Relevância de Cada Métrica . . . . .	3
<b>3</b>	<b>Descrição do Trabalho</b>	<b>4</b>
3.1	Testar redes neurais . . . . .	4
3.1.1	Exaustão . . . . .	4
3.1.2	Tarefa . . . . .	5
3.2	Identificação da presença ou ausência de exaustão . . . . .	5
3.3	Nova escala de exaustão . . . . .	6
<b>4</b>	<b>Conclusão</b>	<b>7</b>
<b>A</b>	<b>Gráficos de Análise de Dados</b>	<b>8</b>
A.1	Relação dos Dados com a Exaustão . . . . .	8

## Resumo

Depois de dois exercícios, onde aprofundamos a representação de conhecimento, vamos agora afastarmos-nos do ***PROLOG*** e debruçarmos-nos sobre ***redes neuronais artificiais*** (RNAs).

# Capítulo 1

## Introdução

Neste terceiro exercício para a unidade curricular de Sistemas de Representação e Conhecimento e Raciocínio, vamos tomar partido de ***Redes Neurais Artificiais*** de forma a, dado um conjunto de dados relativos à utilização do computador, poder prever os níveis de exaustão do utilizador e até a tarefa que este está a desempenhar.

Redes Neurais Artificiais são sistemas capazes de, dado um conjunto de dados, produzir um resultado. Esta definição é muito genérica, descreve apenas um sistema de software qualquer, mas o que distingue uma RNA de um programa convencional é a sua versatilidade. Ao contrário de um programa, uma RNA não necessita de ser programada, não vamos ter linhas de código a especificar como tratar os dados recebidos.

Uma RNA é capaz de identificar padrões no *input* e produzir o *output* que melhor se adequa, para isso, em primeiro lugar, é necessário treinar a rede para poder identificar esses padrões que "ligam" o *input* ao *output* desejado. Para além deste treino, a RNA deva ser desenhada para melhor se adequar ao tipo de dados que recebe. Assim, vamos ter de manter especial atenção à quantidade de conhecimento com que iremos treinar a rede, bem como a como esta está desenhada afim de produzir o *output* esperado.

Depois da nossa Rede Neuronal estar desenhada e ter passado pelo treino devido, é necessário testa-la. Para isso, fazemos passar os nossos dados pela RNA para saber o seu *output* que é depois comparado com o *output* correto, calculamos a ***RMSE*** (*Root Mean Square Error*) para termos o valor de erro da nossa rede.

Vamos passar por todos estes passos, explicar as nossas decisões e finalizar com as conclusões que retivemos no relatório que se segue.

# Capítulo 2

## Análise e Tratamento de Dados

Foi-nos entregue um conjunto de dados relativos a biométricas comportamentais, resultantes da interação humano-computador, para desenvolver uma Rede Neuronal Artificial capaz de identificar os níveis de exaustão do utilizador, bem como a tarefa que este realiza.

### 2.1 Biométricas

#### **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 pixels) 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 que pode variar entre 1 (melhor caso) e 7 (o mais exausto possível).

### **Performance.Task**

Identificação da tarefa em execução no momento da recolha dos dados.

## **2.2 Normalização**

Para a rede neuronal poder funcionar corretamente, é necessário que todos os valores sejam normalizados (i.e., todos os valores devem estar entre -1 e 1 ou entre 0 e 1). Isto significa que até mesmo a **Performance.Task** deve ser colocada em forma de números, para isto, primeiro identificamos que a **Task** só contém três tipos de valores ("`office`", "`work`" e "`programming`") e convertemos esses valores em 0, 0.5 e 1, visto que optamos por colocar todos os valores entre 0 e 1.

## **2.3 Relevância de Cada Métrica**

Nem todas as métricas são relevantes quer para calcular o valor da exaustão, quer para decifrar a tarefa a ser realizada. Para isso, é necessário saber filtrar quais as mais relevantes para o cálculo de cada um destes atributos. Aproveitando o uso da *linguagem R*, que é uma linguagem largamente utilizada para análise estatística, podemos traçar um gráfico que relaciona quer a exaustão, quer a tarefa, a cada uma das métricas. Todos estes gráficos estarão disponíveis em apêndice.

Baseando-nos no quão bem relacionadas estão as métricas com o objetivo, chegamos à conclusão que as mais relevantes são as seguintes. Para a **Tarefa**: **ADMSL** e **DMS**. Para a **Exaustão**: **AED**, **DMS** e **MA**. Assim, vamos focarmos-nos principalmente nestas métricas para o cálculo respetivo, não pondo de parte a hipótese de adicionar métricas novas ou remover algumas das escolhidas.

# Capítulo 3

## Descrição do Trabalho

Depois de uma rigorosa análise dos dados e consequente processamento, vamos agora dar uso a esse conhecimento e desenhar uma RNA capaz de o prever. Para isso, tivemos de separar a informação recebida, parte para treino, parte para testar a própria rede. Esta separação foi feita de forma a garantir que tanto os testes como o treino tinham a mesma quantidade de informação e igualmente relevante, para isso, dividimos a tabela dada em função da linha, para treino as linhas pares, para testes as linhas ímpar.

### 3.1 Testar redes neuronais

#### 3.1.1 Exaustão

Tendo construído uma rede neuronal para o cálculo da exaustão, fizemos vários testes para ajustar a RNA de forma a ter o melhor valor de erro.

Algoritmo	Nodos	KDT	MA	MV	TBC	DDC	DMS	ADMSL	AED	RMSE
rprop+	6,3		✓				✓		✓	0.2431
rprop+	6,2		✓				✓		✓	0.2122
rprop+	6,3,2		✓				✓		✓	0.1810
rprop+	6,3,2			✓			✓		✓	0.1810
rprop+	6,2			✓			✓		✓	0.1806
rprop+	6,2			✓		✓	✓		✓	0.1793
rprop+	6,3,2			✓		✓	✓		✓	0.1802
rprop+	4,2			✓		✓	✓		✓	0.1788
rprop+	4			✓		✓	✓		✓	0.1782
rprop+	6			✓		✓	✓		✓	0.1773

Tabela 3.1: Testes para a exaustão e respetiva média de rmse

Os valores com o asterisco (\*) significa que os dados relativos a essa métrica foram processados previamente de forma a tentar dispersar de maneira uniforme os seus valores. Isto foi feito em R da seguinte maneira:

---

```
1 dados$Performance.DMSMean <- as.numeric(discretize(dados$Performance.DMSMean
, method = "frequency", categories = 10))
```

---

### 3.1.2 Tarefa

Os valores de erro para o cálculo da tarefa não diversificaram muito, portanto realizamos poucos testes.

Algoritmo	Threshold	Nodos	KDT	MA	MV	TBC	DDC	DMS	ADMSL	AED	RMSE
rprop+	0.01	4,2						✓	✓		0.3338
rprop+	0.01	2,2						✓	✓		0.3338
sag	0.1	4,2		✓			✓	✓*	✓		0.3229
slr	0.1	4,2		✓			✓	✓	✓		0.3338
rprop+	0.1	4,2					✓	✓	✓		0.3338
rprop+	0.1	2					✓	✓	✓		0.3332

Tabela 3.2: Testes para o cálculo da tarefa

## 3.2 Identificação da presença ou ausência de exaustão

Foi-nos também pedido para planear uma RNA capaz de detetar presença ou ausência de exaustão. Tomando partido da *Linguagem R*, alteramos o conhecimento que nos foi passado. Assim, todos os valores de exaustão acima de 3, passaram a ter o valor de 1 (representando exaustão), e qualquer outro passou a ter 0 (representando ausência de exaustão).

---

```
1 dados$ExhaustionLevel[dados$ExhaustionLevel <= 3] <- 0
2 dados$ExhaustionLevel[dados$ExhaustionLevel > 3] <- 1
```

---

Algoritmo	Threshold	Nodos	KDT	MA	MV	TBC	DDC	DMS	ADMSL	AED	RMSE
backprop	0.1	4,2			✓		✓	✓		✓	0.001
rprop+	0.01	2,2					✓	✓		✓	0.004

Tabela 3.3: Testes da rede neuronal que deteta presença ou ausência de exaustão

O algoritmo de *backprop* utilizado usava um *learningrate* de 0.3.

Ficamos imensamente satisfeitos com os resultados, chegando até mesmo ter erros de 0 em alguns testes, recordamos que os valores de RMSE acima são médias.



### 3.3 Nova escala de exaustão

A escala de exaustão usada anteriormente concentrava os valores numa classe muito pouco dispersada. Por exemplo, havia muitos valores de exaustão no nível 3 mas pouquíssimos no nível 6, aliás, nos dados que nos foram fornecidos nem estava presente nenhum exemplo com exaustão a nível 7. Assim, usando novamente a "magia" de **R**, usamos uma função que é capaz de "nivelar" um conjunto de dados baseando na frequência com que aparecem no conjunto de dados. Estamos a falar da função **discretize**, que aplicamos nos níveis de exaustão, de forma a dividir estes em 5 categorias, da seguinte maneira:

---

```
1 dados$ExhaustionLevel <- as.numeric(discretize(dados$ExhaustionLevel, method  
= "frequency", categories = 5))
```

---

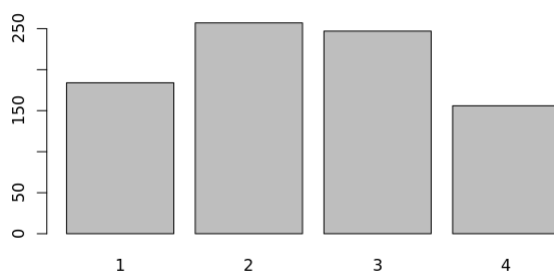


Figura 3.1: Variedade e frequência dos valores de exaustão na sua nova escala

É certo que estamos a reduzir o conhecimento a que temos acesso, mas este pequeno esforço irá compensar no resultado final da rede. Não havia necessidade de tratar casos tão específicos e raros como seria os valores de 7 ou, até mesmo, valores de 6, podem portanto ser aglomerados na mesma categoria.

# Capítulo 4

## Conclusão

Deixamos assim terminado o nosso último exercício para a unidade curricular de SRCR, onde aprofundamos os nossos conhecimentos de Redes Neurais Artificiais na linguagem R, que é altamente usada no campo de *Sistemas Inteligentes* e *Machine Learning*.

Este trabalho levou-nos a questionar fortemente a relevância dos dados que nos são entregues, a gerir qual a melhor maneira de "desenhar" uma RNA e qual a melhor maneira de a treinar. Foi muitas vezes necessário reduzir o conhecimento para produzir melhores resultados, quer na seleção dos dados relevantes para rede, quer na criação de uma nova escala para representar a exaustão. Familiarizamos-nos com os algoritmos de treino mais frequentes para RNAs e desenvolvemos a nossa aptidão para tomada de decisões face aos dados que nos são apresentados.

Conseguimos cumprir o objetivo, desenvolvendo uma rede neuronal capaz de identificar, com um valor baixo de erro, os níveis de exaustão de um utilizador, bem como a tarefa que este realiza, utilizando apenas as biométricas da interação humano-computador.

# Apêndice A

## Gráficos de Análise de Dados

### A.1 Relação dos Dados com a Exaustão

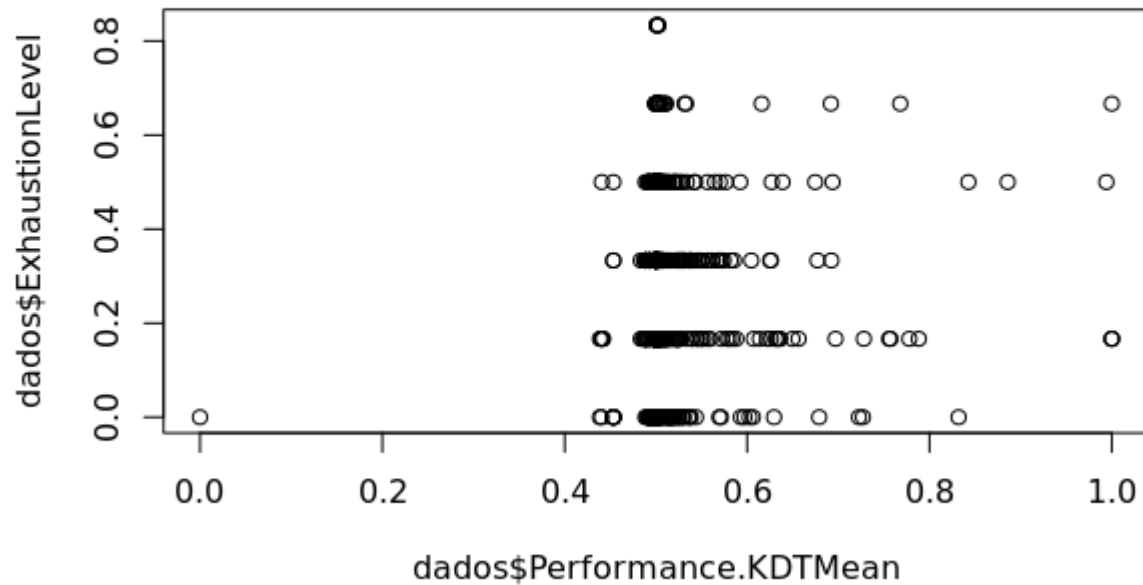


Figura A.1: Relação entre Performance.KDTMean e Exaustão