

Universidade do Minho

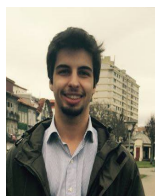
# Sistemas de Representação de Conhecimento e Raciocínio

MIEI - 3º ANO - 2º SEMESTRE

UNIVERSIDADE DO MINHO

## TRABALHO III REDES NEURONAIS ARTIFICIAIS

GRUPO 10



Dinis Peixoto  
A75353



José Bastos  
A74696



Ricardo Pereira  
A74185



Marcelo Lima  
A75210

16 de Novembro de 2017

# 1. *Resumo*

Este relatório é o resultado da elaboração do terceiro e último exercício da componente prática exigida na unidade curricular *Sistemas de Representação de Conhecimento e Raciocínio*. Com o tema *Conhecimento não simbólico: Redes Neurais Artificiais* e totalmente diferente dos exercícios anteriores, a realização deste exercício pretende motivar os alunos para a utilização de sistemas não simbólicos na representação de conhecimento e no desenvolvimento de mecanismos de raciocínio, nomeadamente, Redes Neurais Artificiais (RNAs) para a resolução de problemas.

Assim, tendo por base um vasto conjunto de dados recolhidos da interação humano-computador através dos dispositivos físicos rato e teclado é exigido que seja realizado um estudo que envolva a identificação da exaustão reconhecida, assim com a tarefa de execução.

Estes dados fornecidos correspondem a várias biométricas comportamentais de fácil recolha e, de certa forma, invisíveis ao utilizador. No seu total são 10 biométricas distintas entre as quais, o tempo médio entre o momento em que uma tecla é pressionada para baixo e o momento em que é largada, o período de tempo entre dois eventos MOUSEUP consecutivos, etc.

Desta forma, a solução para este estudo passará por estudar os atributos mais significativos para a representação do conhecimento do problema em análise, identificar as topologias de rede mais adequadas e selecionar as regras de aprendizagem para treinar essas mesmas redes, tudo isto analisando a capacidade de resolução do problema por uma ou mais RNAs.

Tal como dito anteriormente, o objetivo passa por conseguir identificar a exaustão e a tarefa de execução. Em relação à primeira serão identificados os 7 níveis de exaustão, será identificada a simples existência ou ausência de exaustão e por último será encontrada a melhor escala de identificação de exaustão. Relativamente à tarefa de execução, serão exploradas soluções que permitam classificar as ocorrências numa das três atividades work, office ou programming. Tudo isto será elaborado recorrendo ao ambiente de análise de dados R.

# Conteúdo

<b>1</b>	<b>Resumo</b>	<b>1</b>
<b>2</b>	<b>Introdução</b>	<b>3</b>
<b>3</b>	<b>Preliminares</b>	<b>4</b>
3.1	O que é uma Rede Neuronal Artificial? . . . . .	4
3.2	Comparação e constituição biológica . . . . .	4
3.3	Benefícios das RNAs . . . . .	5
<b>4</b>	<b>Descrição do Trabalho e Análise de Resultados</b>	<b>6</b>
4.1	Análise dos dados e Normalização . . . . .	6
4.1.1	Análise . . . . .	7
4.1.2	Normalização . . . . .	8
4.1.3	Rede neuronal . . . . .	8
4.2	Os 7 níveis de exaustão . . . . .	9
4.2.1	Relevância dos atributos . . . . .	9
4.2.2	Fórmulas . . . . .	10
4.2.3	Nível de Fadiga . . . . .	11
4.2.4	Tarefa . . . . .	11
4.3	A existência ou ausência de exaustão . . . . .	12
4.3.1	Escala . . . . .	12
4.3.2	Relevância dos atributos . . . . .	12
4.3.3	Fórmulas . . . . .	12
4.3.4	Nível de Fadiga . . . . .	13
4.3.5	Tarefa . . . . .	14
4.4	Melhor escala de identificação de exaustão . . . . .	14
4.4.1	Escala . . . . .	14
4.4.2	Relevância dos atributos . . . . .	14
4.4.3	Fórmulas . . . . .	14
4.4.4	Nível de Fadiga . . . . .	15
<b>5</b>	<b>Conclusões e Sugestões</b>	<b>16</b>
<b>6</b>	<b>Referências</b>	<b>17</b>

## 2. *Introdução*

Este último exercício, proposto pela unidade curricular de *Sistemas de Representação de Conhecimento e Raciocínio*, é um pouco distinto dos anteriores, uma vez que este se debruça num tema um pouco diferente que os restantes que é o *Conhecimento não simbólico: Redes Neurais Artificiais*.

Assim, tendo por base um enorme conjunto de dados fornecidos pelo docente, com um vasto conjunto de biométricas comportamentais relativas à interação humano-computador através dos dispositivos físicos rato e teclado, teremos de ser capazes de fazer um estudo de modo a identificar não só os diferentes níveis de exaustão reconhecidos, como também a respetiva tarefa em execução durante estas mesmas interações.

Tudo isto será elaborado através de um processo sequencial com todas as etapas necessárias à sua realização, as quais serão de seguida abordadas.

### 3. *Preliminares*

Deixando a dependência da representação de conhecimento através do uso de símbolos foi criada a representação de conhecimento sub-simbólicos. Neste documento abordaremos apenas um ramo muito amplo deste tema, as Redes Neurais Artificiais, denominadas por RNAs. Para uma melhor compreensão do seguinte tema explicar-se-ão conceitos essenciais que serão abordados mais profundamente ao longo do presente relatório.

#### 3.1 O que é uma Rede Neuronal Artificial?

*As Redes Neurais Artificiais, também designadas por sistemas conexionistas, são modelos simplificados do sistema nervoso central do ser humano. Trata-se de uma estrutura extremamente interconectada de unidades computacionais, frequentemente designadas por neurónios ou nodos, com capacidade de aprendizagem.*

*Uma RNA é um processador eminentemente paralelo, composto por simples unidades de processamento, que possui uma propensão natural para armazenar conhecimento empírico e tornar-lo acessível ao utilizador.*

O comportamento de uma RNA, tal como podemos facilmente reparar é baseado no do cérebro do ser humano, desta forma em ambos:

- O conhecimento é adquirido a partir de um processo de aprendizagem.
- O conhecimento é armazenado nas conexões entre os nodos, também designadas por sinapses.

#### 3.2 Comparação e constituição biológica

Como vimos anteriormente, e uma vez que toda representação tem uma profunda inspiração biológica, a constituição não fugirá à regra, como tal estas apresentam, na sua generalidade:

- **Neurónio:** unidades de processamento;
- **Dentrite:** associadas aos neurónios, recebem a informação previamente para que esta possa ser efetivamente processada;
- **Axónio:** também associados aos primeiros, são responsáveis pela passagem completa de informação.

Um neurónio pode possuir várias dentrites mas apenas um único axónio. A passagem de informação tem o nome de sinapse, e esta apenas ocorrerá caso o estado de excitação

dos neurónios seja suficiente, sendo este regulado conforme a informação que chega ao neurónio.

A rede neuronal recebe então um ou vários parâmetros de um caso como *input* e percorre esta informação recebida através da sua rede até que é retornado como *output*, ou ou mais valores (*não significando que com um input se terá necessariamente apenas um output*). A aprendizagem da rede é definida pela regra de transferência que a rede neuronal implementa, isto será outro parâmetro que decidirá o funcionamento desta. O cálculo do valor de ativação dos neurónios é influenciado pela informação recebida por estes, pelos dados de *input* e pelo valor de ativação posterior, respectivamente armazenado em memória.

### 3.3 Benefícios das RNAs

O poder computacional de uma RNA baseia-se em dois aspectos fundamentais: o facto de apresentar uma topologia que premeia o paralelismo e a sua capacidade de aprendizagem e generalização. Estas não são, no entanto, todo o conjunto dos seus benefícios:

- ***aprendizagem e generalização*** - conseguindo descrever o todo a partir de algumas partes, constituindo-se como formas eficientes de aprendizagem e armazenamento de conhecimento;
- ***processamento maciçamente paralelo*** - permitindo que tarefas complexas sejam realizadas num curto espaço de tempo;
- ***transparência*** - podendo ser vistas como uma caixa negra que transforma vectores de entrada em vectores de saída, via uma função desconhecida;
- ***não linearidade*** - atendendo a que muitos dos problemas reais a equacionar e resolver são de natureza não linear.

## 4. Descrição do Trabalho e Análise de Resultados

### 4.1 Análise dos dados e Normalização

Posteriormente à complexa e elaborada análise de dados obtidos e respectiva normalização, devemos primeiramente fazer uma compreensão profunda às identidades e domínios dos conjuntos de dados.

O conjunto de biométricas comportamentais escolhidas para o estudo foram as *Keystroke Dynamics* e *Mouse Dynamics*, obtidas através do uso do teclado e rato, respectivamente. Estas, como já sabemos revelaram ser as mais adequadas para a deteção de exaustão mental, uma vez que resultam da interação computador-humano da forma mais natural possível, evitando fatores externos, assim como de rápida identificação. Além de todos estes aspectos extremamente importantes ao estudo, estas biométricas apresentam ainda um outro de grande relevância, o facto de ser praticamente invisíveis ao utilizador, garantindo dados completamente imparciais.

Seguem-se as biométricas utilizadas para deteção de exaustão:

- **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 ( $x_1$ ;  $y_1$ ) e uma C2 ( $x_2$ ;  $y_2$ ) 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 mento da recolha dos dados.

Os diferentes níveis de exaustão mental devem também ser tidos em conta, a escala a seguir apresentada são uma representação da escala *USAFSAM Fatigue scale*, desenvolvida pelo Dr. *Layne Perelli* no *USA Army*.

A escala apresenta o seguinte formato:

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*.

#### 4.1.1 Análise

É pretendida a realização de um estudo envolvente da identificação da exaustão reconhecida e respectiva tarefa em execução. Para tal, previamente foi realizado um estudo relativamente aos dados fornecidos com o simples intuito de se perceber em quais níveis de exaustão existiria mais casos identificados e da mesma forma, relativamente à tarefa em execução, quais delas teriam mais casos assinalados.

Segue-se um gráfico capaz de mostrar a relação entre o número de amostras por nível de fadiga. Facilmente notamos a exponencial diferença entre os distintos níveis de fadiga. Nos primeiros quatro níveis o número de amostras é, realmente, elevado enquanto que nos restantes três, os níveis mais superiores, o número de amostras passa a ser mais escasso, até que no último nível chega mesmo a ser nulo.

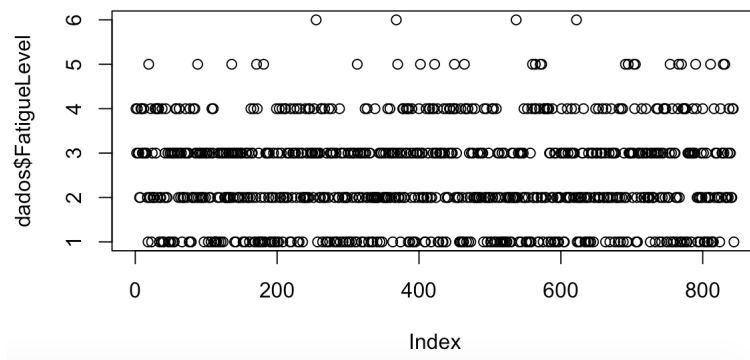


Figura 4.1: Número de amostras por nível de fadiga.



Relativamente à tarefa em execução, o processo efetuado foi idêntico comparando agora o número de amostras por tarefa em execução. Por sua vez os resultados também não foram, de todo, surpresa, embora mais consistentes que o caso anterior. De facto, as tarefas 1 (*Work*) e 2 (*Office*) partilham um elevado número de amostras, enquanto que a tarefa 3 (*Programming*) mostrou-se com uma diminuição bastante significativa comparativamente às adversárias.

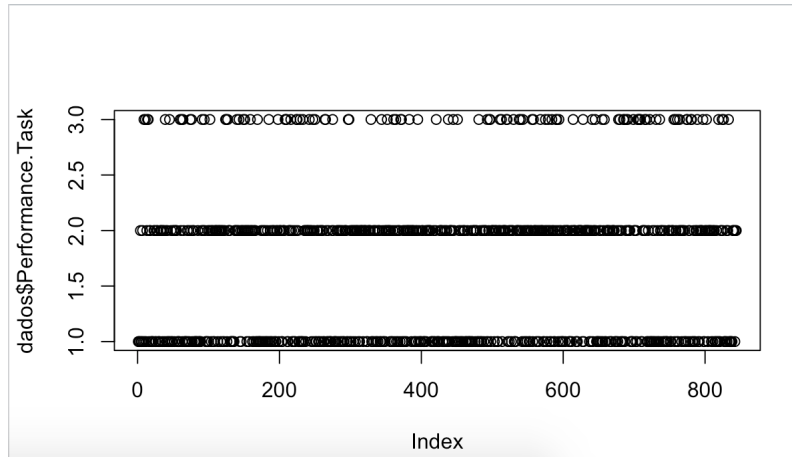


Figura 4.2: Número de amostras por tarefa em execução.

#### 4.1.2 Normalização

As biométricas anteriormente mencionadas necessitaram de ser normalizadas de modo a uma melhor obtenção de resultados.

Os primeiros 8 parâmetros (*Performance*) encontram-se definidos no intervalo  $[-1, 1]$ , o 9º parâmetro (*ExhaustionLevel*) no intervalo  $[1, 7]$  e por fim, o último parâmetro encontra-se identificado por uma das seguintes três tags: *Work*, *Office* ou *Programming*. Neste último caso houve necessidade de normalizar estes para valores inteiros, ao que o grupo optou por utilizar o intervalo  $[0, 2]$ , algo que não foi possível uma vez que devido ao valor lógico que o valor 0 representa. Esta situação foi facilmente ultrapassada incrementando a posição inicial do intervalo a uma unidade, ficando assim  $[1, 3]$ .

Além de tudo isto e uma vez que os dados estão agrupados e seguem um padrão específico, de modo a tornar-mos tudo mais realista e imparcial optamos por normalizar esta situação efetuando uma leitura aleatória aos mesmos, deixando assim de ser uma leitura consistente e conforme um padrão previamente definido.

#### 4.1.3 Rede neuronal

Depois de uma pesquisa sobre a quantidade de nodos e camadas a utilizar, chegamos à conclusão que, de forma a conseguirmos abranger os vários casos da topologia:

- A primeira rede será constituída por um número de nodos por camada superior ao número de parâmetros.
- A segunda será constituída por um número de nodos por camada inferior ou igual ao número de parâmetros.

Como ferramenta para realizar o treino da RNA recorreremos à função *neuralnet*:

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

Os parâmetros acima apresentados são:

- formula - fórmula utilizada para treinar a rede.
- data - dataset que contém as variáveis da fórmula.
- hidden - define o número de nodos escondidos que iremos usar.
- threshold - valor de erro que irá parar a execução da função.
- algoritmo - algoritmo responsável pela realização do treino.
- lifesign - responsável por especificar aquilo que será impresso durante a execução.
- linear.output - bool responsável por especificar a utilização de nodos exteriores.

## 4.2 Os 7 níveis de exaustão

### 4.2.1 Relevância dos atributos

Com as tabelas seguintes é visível, os atributos mais relevantes tanto para o *FatigueLevel* como para a *Task*.

		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 4.3: Resultado obtido pelo *RStudio* para os atributos com maior relevância do *FatigueLevel*.

Para este caso, é visível que os 4 atributos mais relevantes são: Performance.KDTMean, Performance.MAMean, Performance.MVMMean e Performance.DDCMean.

		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 4.4: Resultado obtido pelo *RStudio* para os atributos com maior relevância da *Task*.

Já para a *Task* os 4 atributos mais relevantes são: Performance.KDTMean, Performance.DDCMean, Performance.DMSMean e Performance.AEDMean.

Para uma melhor análise dos atributos, recorreremos ao programa *Weka*, obtendo os seguintes resultados:

```
Selected attributes: 2,5,10 : 3
Performance.MAMean
Performance.DDCMean
```

Figura 4.5: Resultado obtido pelo *Weka* para os atributos com maior relevância do *FatigueLevel*.

```
Selected attributes: 1,4,5,9 : 4
Performance.KDTMean
Performance.TBCMean
Performance.DDCMean
```

Figura 4.6: Resultado obtido pelo *Weka* para os atributos com maior relevância da *Task*.

Por fim, podemos verificar que os atributos obtidos, são quase idênticos, o que permite-nos concluir que os atributos apresentados são os mais relevantes.

### 4.2.2 Fórmulas

A fórmula nº1 corresponde à fórmula, que possui todos as métricas como atributos (inputs), sendo apresentada assim :

```
formulaFatigue <- FatigueLevel ~ Performance.KDTMean + Performance.MAMean + Performance.MVMean +
Performance.TBCMean + Performance.DDCMean + Performance.DMSMean +
Performance.AEDMean + Performance.ADMSLMean
```

Figura 4.7: Formula do *FatigueLevel*.

Quanto à fórmula nº2, apenas possui as 4 métricas mais relevantes consoante o resultado da análise efectuada anteriormente, obtendo a seguinte fórmula :

```
formulaFatigue <- FatigueLevel ~ Performance.KDTMean + Performance.MAMean + Performance.MVMean + Performance.DDCMean
```

Figura 4.8: Formula nº2 do *FatigueLevel*.

A fórmula nº3 possui todos métricas como atributo para a *Task*, obtendo a seguinte fórmula :

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

Figura 4.9: Formula nº3 da *Task*.

Para a fórmula nº4, apenas possui as 4 métricas mais relevantes consoante o resultado da análise efectuada anteriormente, isto é :

```
formulaTask <- Performance.Task ~ Performance.KDTMean + Performance.DDCMean + Performance.DMSMean + Performance.AEDMean
```

Figura 4.10: Formula nº4 da *Task*.

### 4.2.3 Nível de Fadiga

Nº Teste	Fórmula	Hidden	Algoritmo	Threshold	Steps	Error	Rmse
1	formula1	c(4,2)	sag	0.1	23441	245.02164	1.171684
2	formula1	c(4,2)	slr	0.1	3079	289.33774	1.163357171
3	formula1	c(3)	rprop-	0.1	2273	317.99526	1.104909842
4	formula2	c(10,5)	sag	0.1	86280	247.80187	1.164260809
5	formula2	c(5,2)	rprop+	0.05	4856	300.98618	1.148802841
6	formula2	c(3)	slr	0.1	1013	355.03094	1.10416036

Tabela 4.1: Tabela com os resultados do nível da fadiga.

### 4.2.4 Tarefa

Nº Teste	Fórmula	Hidden	Algoritmo	Threshold	Steps	Error	Rmse
1	formula1	c(4,2)	slr	0.1	5712	91.99873	0.6089917352
2	formula1	c(3)	rprop+	0.05	3113	97.99757	0.6563277567
3	formula1	c(8,3)	rprop-	0.05	6710	60.44245	0.5474435597
4	formula2	c(3)	sag	0.1	6228	132.04266	0.8341141448
5	formula2	c(4,2)	slr	0.1	37417	107.31762	0.8717923318
6	formula3	c(5)	rprop+	0.1	81	135.22678	0.8560583322

Tabela 4.2: Tabela com os resultados sobre a tarefa em execução.

## 4.3 A existência ou ausência de exaustão

### 4.3.1 Escala

Perante esta situação o grupo optou por normalizar a escala anterior, a *USAFSAM Fatigue scale* para uma escala com apenas dois valores possíveis: 1 e 2. Desta forma, todos os níveis na escala primária iguais ou inferiores a 3, correspondendo ao nível respectivamente aos níveis: *Totalmente bem*; *Responsivo, mas não no pico* e *Ok, normal* - passaram a corresponder ao **nível 1 da escala normalizada**. Por sua vez, os restantes níveis: *Em baixo de forma/do normal, a sentir-se em baixo*; *Sentido moleza, perdendo o foco*; *Muito difícil concentrar, meio tonto* e *Incapaz de funcionar, pronto a desligar* - passaram a corresponder ao **nível 2 da escala normalizada**.

```
#normalizar fatiguelevel para 1 e 2|
dados$FatigueLevel[dados$FatigueLevel <= 3] <- 1
dados$FatigueLevel[dados$FatigueLevel > 3] <- 2
```

Figura 4.11: Excerto do *script* de normalização da escala.

Esta decisão foi tomada de modo a conseguir quase que uma escala *booleana*, isto é, a existência (1) ou não (2) de exaustão.

### 4.3.2 Relevância dos atributos

Nesta nova escala, reparamos que a ordem dos atributos mais relevantes para a *Task* permaneceu igual. No entanto, para o *FatigueLevel* reparamos em algumas mudanças.

		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 4.12: Resultado obtido pelo *RStudio* para os atributos com maior relevância do *FatigueLevel*, agora noutra escala.

### 4.3.3 Fórmulas

A fórmula nº1 apresenta todos os atributos e já foi representada anteriormente :

```
formulaFatigue <- FatigueLevel ~ Performance.KDTMean + Performance.MAMean + Performance.MVMean +  
Performance.TBCMean + Performance.DDCMean + Performance.DMSMean +  
Performance.AEDMean + Performance.ADMSLMean
```

Figura 4.13: Formula do *FatigueLevel*.

Quanto a fórmula nº2, apenas possuí as 4 métricas mais relevantes para esta determinada análise, obtendo a seguinte fórmula :

```
formulaFatigue <- FatigueLevel ~ Performance.KDTMean + Performance.MAMean + Performance.MVMean + Performance.TBCMean
```

Figura 4.14: Formula nº2 do *FatigueLevel*.

Como se pode constatar, neste caso apenas um dos atributos mudou comparativamente ao caso dos 7 níveis, em vez do *Performance.DDCMean* está representado o *Performance.TBCMean*.

A fórmula nº3 também já foi apresentada anteriormente, possuindo todos as métricas como atributo para a *Task*, obtendo a seguinte fórmula :

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

Figura 4.15: Formula nº3 da *Task*.

Para a fórmula nº4, possuí as 4 métricas mais relevantes consoante o resultado da análise efectuada anteriormente, isto é :

```
formulaTask <- Performance.Task ~ Performance.KDTMean + Performance.DDCMean + Performance.DMSMean + Performance.AEDMean
```

Figura 4.16: Formula nº4 da *Task*.

Como se pode constatar, neste caso ficou igual à outra fórmula dos 7 níveis, pois os atributos mais relevantes são os mesmos.

#### 4.3.4 Nível de Fadiga

Nº Teste	Fórmula	Hidden	Algoritmo	Threshold	Steps	Error	Rmse
1	formula6	c(3)	slr	0.1	317	41.31133	0.4075243403
2	formula6	c(4,2)	rprop-	0.1	1175	35.51564	0.4179063407
3	formula6	c(8,4,2)	sag	0.1	stepmax	-	-
4	formula7	c(4,2)	slr	0.1	1779	41.21287	0.4257320821
5	formula7	c(8,2)	rprop-	0.05	8264	27.961	0.4424085538
6	formula7	c(10,4)	rprop-	0.1	1297	30.76818	0.4520066356

Tabela 4.3: Tabela com os resultados do nível da fadiga.

### 4.3.5 Tarefa

Nº Teste	Fórmula	Hidden	Algoritmo	Threshold	Steps	Error	Rmse
1	formula8	c(4,2)	slr	0.1	1532	89.78971	0.6683721355
2	formula8	c(5)	rprop+	0.1	1982	76.45972	0.6954182607
3	formula9	c(10,5)	rprop-	0.1	12386	77.58656	0.7995900589
4	formula9	c(3)	sag	0.1	4103	129.14823	0.8097763302

Tabela 4.4: Tabela com os resultados sobre a tarefa em execução.

## 4.4 Melhor escala de identificação de exaustão

### 4.4.1 Escala

Nesta fase, uma vez mais foi necessária a implementação de uma nova escala de identificação que de certa forma seja capaz de melhorar a sua compreensão e diminuir a quantidade de erros presentes. Isto foi conseguido através de uma cuidadosa seleção de níveis de fadiga associando quando um destes níveis a um nível abstrato criado pelo grupo, isto é, o grupo escolheu quantos níveis de fadiga e em cada um destes quando níveis primários incluir. Para tal foi utilizado o método *descretize* que auxilia a visualizar o número de categorias onde existe um equilíbrio no número de amostras.

Chegamos então à conclusão que existe um equilíbrio em 5 níveis, embora sejam apenas 4 considerados uma vez que o último nível corresponde ao nível primário 7, onde o número das amostras é completamente nulo. Desta forma, partindo dos 7 níveis primários teremos os seguintes níveis:

- nível 1 - [1];
- nível 2 - [2];
- nível 3 - [3];
- nível 4 - [4,6];
- nível 5 - [7] (não utilizado).

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

Figura 4.17: Função utilizada.

### 4.4.2 Relevância dos atributos

Na relevância dos atributos, como se pode constatar permanece igual ao capítulo anterior apresentado

### 4.4.3 Fórmulas

As fórmulas desta parte também ficam iguais ao do capítulo anterior.

#### 4.4.4 Nível de Fadiga

Nº Teste	Fórmula	Hidden	Algoritmo	Threshold	Steps	Error	Rmse
1	formula1	c(10,5)	slr	0.1	stepmax	-	-
2	formula1	c(3)	slr	0.1	2813	263.08188	1.050617604
3	formula1	c(5,2)	rprop+	0.1	20674	205.43987	1.134621861
4	formula2	c(3)	rprop-	0.1	4878	275.91696	1.190364119
5	formula2	c(5,2)	slr	0.1	14927	255.44683	1.229045559
6	formula2	c(8,4)	sag	0.1	50297	216.70133	1.243112498

Tabela 4.5: Resultados dos testes.



## 5. *Conclusões e Sugestões*

Sendo este o terceiro e último exercício que compõe o trabalho prático da Unidade Curricular, a sua elaboração foi um pouco mais demorada e complexa que os anteriores.

Com este exercício deixamos de lado aquele que é o Conhecimento simbólico, baseado na lógica, e iniciamos o estudo do Conhecimento não simbólico, baseado na capacidade de aprendizagem, nunca antes trabalhado pelo grupo.

Com este tipo de conhecimento surgem assim as Redes Neurais Artificiais, concebidas com base num modelo simplificado do sistema nervoso central dos seres humanos, as quais tivemos de aprofundar, entendendo o seu funcionamento de modo a elaborarmos este exercício.

O enunciado apresentado, para além de ser um pouco abstrato, continha uma série de requisitos que tinham de ser obrigatoriamente cumpridos e nos quais nos debruçamos da forma mais correta possível para os atingir.

Durante a sua elaboração deparamo-nos com várias incertezas e dificuldades que, depois de alguma dedicação e consulta externa, conseguimos ultrapassar da melhor forma.

Assim, a elaboração deste exercício revelou ser de extrema importância e bastante útil para todo o grupo, uma vez que nos permitiu pôr em prática os conhecimentos adquiridos durante as várias aulas e ao mesmo tempo aperfeiçoar e aprofundar esses mesmos conhecimentos e ferramentas das quais tiramos proveito, enriquecendo-nos com uma experiência que certamente será bastante útil no futuro.

Concluindo, pensamos que o resultado final decorrente da resolução do exercício proposto vai de encontro aquilo que era esperado, na medida em que nos empenhamos da melhor maneira possível.

## 6. *Referências*

- [1] - Yampolskiy, Roman V., and Venu Govindaraju. "Behavioural biometrics: a survey and classification." *International Journal of Biometrics* 1.1 (2008): 81-113.
- [2] - Perelli, Layne P. *Fatigue Stressors in Simulated Long-Duration Flight. Effects on Performance, Information Processing, Subjective Fatigue, and Physiological Cost.* No. SAM-TR-80-49. SCHOOL OF AEROSPACE MEDICINE BROOKS AFB TX, 1980.
- [3] - Pimenta A., Carneiro D., Novais P., Neves J., *Detection of Distraction and Fatigue in Groups through the Analysis of Interaction Patterns with Computers, Intelligent Distributed Computing VIII*, Springer-Verlag - *Studies in Computational Intelligence*, David Camacho, Lars Braubach, Salvatore Venticinque and Costin Badica (Eds) Vol. 570, pp 29-39, ISBN: 978-3-319-10421-8, 2014.
- [4] - Pimenta A., Carneiro D., Novais P., Neves J., *Monitoring Mental Fatigue through the Analysis of Keyboard and Mouse Interaction Patterns, Hybrid Artificial Intelligent Systems - 8th International Conference HAIS 2013*, Jeng-Shyang Pan, Marios M. Polycarpou, Michał Woźniak, André C. P. L. F. de Carvalho, Héctor Quintián, Emilio Corchado (eds), *Lecture Notes in Computer Science*, Vol 8073, ISBN 978-3-642-40845-8, pp 222-231, 2013.