



Universidade do Minho  
Escola de Engenharia

## **Redes Neurais**

Mestrado Integrado em Engenharia Informática

Sistemas de Representação de Conhecimento e Raciocínio  
2º Semestre\2016-2017

A70938 Diogo Meira Neves  
A70676 Marcos de Morais Luís  
A71625 Nelson Arieira Parente  
A70951 Pedro Miguel Lopes Pereira

18 de Maio de 2017  
Braga

## Resumo

Este documento relata o trabalho realizado para a componente prática da unidade curricular de **Sistemas de Representação de Conhecimento e Raciocínio**, nomeadamente o terceiro exercício do trabalho de grupo. Pretende-se a utilização e aplicação da linguagem de programação *R* para a resolução de problemas, neste caso, continuando a debater representação de conhecimento e raciocínio

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>4</b>
<b>2</b>	<b>Contextualização</b>	<b>5</b>
<b>3</b>	<b>Preliminares</b>	<b>5</b>
<b>4</b>	<b>Normalização e Analise de Dados</b>	<b>6</b>
4.1	Normalização . . . . .	6
4.2	Análise . . . . .	7
<b>5</b>	<b>Implementação</b>	<b>9</b>
5.1	Importância dos atributos . . . . .	9
5.2	Redes Neurais . . . . .	10
5.2.1	Formulas . . . . .	10
5.2.2	Definição da rede neuronal . . . . .	11

# 1 Introdução

Foi-nos proposta a realização deste projeto no âmbito da unidade curricular de **Sistemas de Representação de Conhecimento e Raciocínio**, sendo que o mesmo, tem como objetivo motivar nos para a utilização de sistemas não simbólicos na representação de conhecimento e no desenvolvimento de mecanismos de raciocínio, utilizando mais designadamente Redes Neurais Artificiais (RNAs) para a resolução de problemas.

## 2 Contextualização

Este projeto tem como tema incidente a biometria sendo a ciência e a tecnologia de medição e análise de dados biológicos. Em tecnologia da informação, a biometria refere-se a tecnologias que medem e analisam as características do corpo humano, tais como o ADN, impressões digitais, retinas e íris, padrões de voz, padrões faciais e medições de mão, para fins de autenticação. Neste trabalho focaremos as características comportamentais mais especificamente a relação entre homem-computador, mais especificamente a interação do homem através do teclado e do rato. Como tal foram fornecidos dados de vários tipos de parâmetros que são recolhidos de maneira invisível e não intrusiva para a realização de testes.

## 3 Preliminares

Quebrando a dependência da representação de conhecimento através do uso de símbolos, foi criada a representação de conhecimento sub-simbólicos. Neste trabalho apenas se abordaram um “ramo” deste amplo tema, Redes neuronais artificiais (RNAs). Para permitir uma melhor compreensão do tema de seguida explicar-se-ão conceitos essenciais à compreensão do abordado ao longo do relatório. Redes neuronais artificiais são estruturas de resolução de problemas que quebram a dependência da utilização de símbolos. Baseia-se na conexão entre unidades de processamento e a sua nomenclatura é herdada da biologia. Desta forma uma rede neuronal é constituída por:

- Neurónio: unidades de processamento;
- Dentríte: associadas aos neurónios, recebem a informação que depois é processada;
- Axónio: também associados aos neurónios, são responsáveis pela passagem de informação.

Um neurónio pode possuir várias dentrites mas apenas um axónio. À passagem de informação dá-se o nome de transferência/sinapse, esta apenas ocorre caso o estado de excitação dos neurónios seja suficiente. Este estado é regulado pela informação que chega ao neurónio. A rede neuronal recebe então n parâmetros de um caso como input e, faz esta informação percorrer a sua rede até que é retornado um ou mais valor/valores de output. A aprendizagem da rede é definida pela regra de transferência que a rede neuronal implementa, isto será outro parâmetro que decidirá o funcionamento da rede. O cálculo do valor de ativação dos neurónios é influenciado pela informação que chega aos mesmos, pelos dados de input e pelo valor de ativação anterior (armazenado em memória). Apesar do uso das redes neuronais ser bastante vantajoso é de notar que todos os valores obtidos são apenas aproximações e que existe uma dependência na existência de “pré-conhecimento”, ou seja, são necessários casos de treino com informação já real. 4. Descrição do Trabalho e Análise de Resultados A documentação, assim como o exercício, está dividida em três partes principais, que correspondem aos três desafios propostos. Na primeira parte é pedido que se tente treinar a rede para que esta responda com o maior grau de aproximação aos resultados disponibilizados, a segunda parte da documentação incide sobre o desafio de transformar a escala de fadiga fornecida (níveis de 0 a 7) para uma escala que apenas identifica se existe ou não fadiga (dois níveis 1 identifica fadiga e 0 identifica a sua ausência), finalmente, na terceira e última parte é pedido que se crie uma escala que melhor se adegue ao problema.

## 4 Normalização e Análise de Dados

### 4.1 Normalização

Foi nos fornecido um ficheiro em que contém uma tabela com valores relativos às biométricas utilizadas para deteção de exaustão, sendo elas:

- Performance.KDTMean
- Performance.MAMean
- Performance.MVMean
- Performance.TBCMMean
- Performance.DDCMean
- Performance.DMSMean
- Performance.ADMSLMean
- Performance.AEDMean
- ExhaustionLevel
- Performance.Task

Os primeiros 8 dos 10 parâmetros estão definidos num intervalo entre  $[-1,1]$ , o nível de exaustão tem valores entre  $[1,7]$  cada um com um significado próprio. O ultimo parâmetro está definido por uma de 3 tags sendo estas "Work", "Office" e "Programming", houve uma necessidade de normalizar isto para valores inteiros, a primeira gama de valores utilizados foram respetivamente 0,1 e 2 mas verificou-se que ao associar o valor 0 a uma das 3 tags podia dar resultados errados devido ao valor lógico e matemático que o 0 representa, então posteriormente decidiu-se unanimemente passar a gama de valores utilizados para 1,2 e 3.

Após esta modificação, chegou-se à conclusão que de certa maneira os dados estão agrupados por tag de performance ou seja por outras palavras os testes "works" estão todos seguidos e assim por diante o que iria originar problemas em termos de representação de resultados pois ao dividirmos os dados de input em treino teste, não iriam ser similares em termos de consistência de numero de amostras de cada tipo de tags. Tendo isto em conta procedeu-se ao "baralhamento" dos dados para assim dispersar qualquer padrão existente.

## 4.2 Análise

Pretende-se que seja realizado um estudo que envolva a identificação da exaustão reconhecida e a tarefa em execução. Para tal houve um estudo prévio relativamente aos dados fornecidos com o intuito de se perceber em que níveis de exaustão existia mais casos identificados e também por sua vez relativamente à tarefa em execução qual delas tinha mais casos apontados.

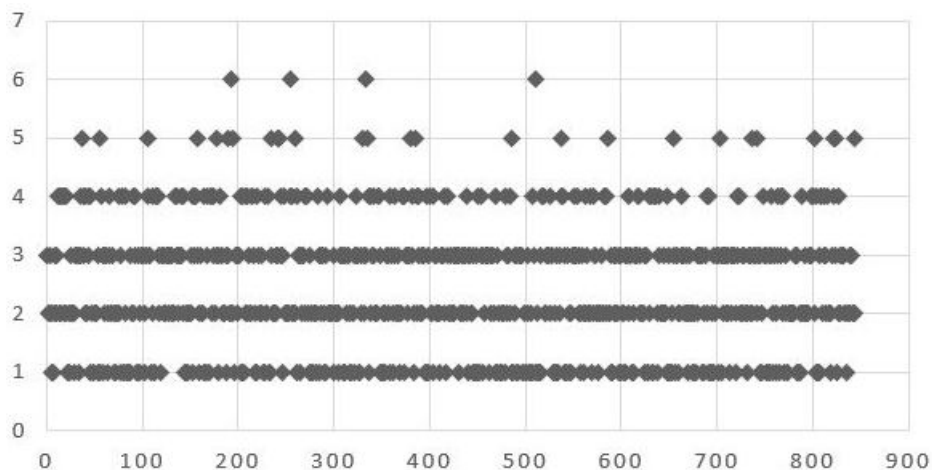


Figura 1: Nº de amostras por nível de fadiga - Gráfico de Dispersão

Foi utilizado como primeiro gráfico um gráfico de dispersão para mostrar a relação entre as amostras e o seu nível de fadiga correspondente, como se pode observar existe uma maior ocorrência nos primeiros 4 níveis sendo eles :

- [1] - Totalmente bem;
- [2] - Responsivo, mas não no pico;
- [3] - Ok, normal;
- [4] - Em baixo de forma/do normal, a sentir-se em baixo;

Por outro lado, os últimos 3 níveis tem um numero muito baixo de ocorrências cerca de 10 por cento do total de dados.

- [5] - Sentido moleza, perdendo o foco;
- [6] - Muito difícil concentrar, meio tonto;
- [7] - Incapaz de funcionar, pronto a “desligar”.

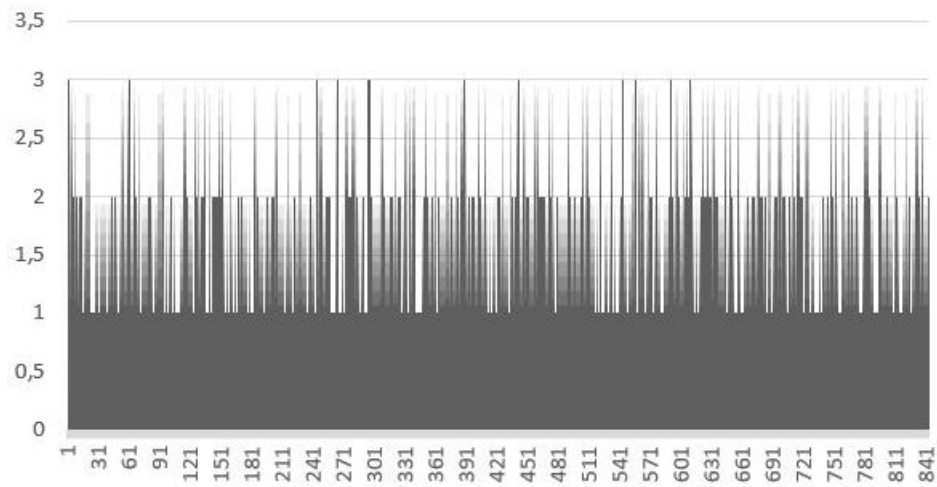


Figura 2: Nº de amostras por tarefa em execução

Relativamente à tarefa em execução procedeu-se ao mesmo estudo para se poder observar neste caso a relação entre o numero de amostras e a tarefa em execução , após a criação do gráfico de área , chegou-se à conclusão que existe mais casos de 1 e 2 , sendo o 3 a tarefa com menos ocorrência. Em termos da numeração , temos de seguida a discriminação de cada numero , ja antes referida na parte da normalização.

- [1] - Work;
- [2] - Office;
- [3] - Programming.



## 5 Implementação

### 5.1 Importância dos atributos

Previamente foi feito um estudo da relevância de cada atributo para a representação de conhecimento do problema em análise. Em primeiro foi feito a partir da formula para o nível de fadiga, para tal numa primeira versão foi preciso utilizar todos os atributos disponiveis para ver qual a relevancia de cada um.

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

- Tabela gerada relativamente à importância dos atributos em relação ao nível de fadiga

		KDTMean	MAMean	MVMean	TBCMean	DDCMean	DMSMean	AEDMean	ADMSLMean	Task
1	( 1 )	" "	" "	" "	" "	" "	" "	" "	" "	"*"
2	( 1 )	" "	"*"	" "	" "	" "	" "	" "	" "	"*"
3	( 1 )	" "	"*"	"*"	" "	" "	" "	" "	" "	"*"
4	( 1 )	" "	"*"	"*"	" "	"*"	" "	" "	" "	"*"
5	( 1 )	" "	"*"	"*"	" "	"*"	"*"	" "	" "	"*"
6	( 1 )	" "	"*"	"*"	" "	"*"	"*"	" "	"*"	"*"
7	( 1 )	"*"	"*"	"*"	" "	"*"	"*"	" "	"*"	"*"
8	( 1 )	"*"	"*"	"*"	"*"	"*"	"*"	" "	"*"	"*"
9	( 1 )	"*"	"*"	"*"	"*"	"*"	"*"	"*"	"*"	"*"

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

- Tabela gerada relativamente à importância dos atributos em relação à tarefa em execução

		KDTMean	MAMean	MVMean	TBCMean	DDCMean	DMSMean	AEDMean	ADMSLMean	FatigueLvl
1	( 1 )	" "	" "	" "	" "	" "	" "	" "	" "	"*"
2	( 1 )	"*"	" "	" "	" "	" "	" "	" "	" "	"*"
3	( 1 )	"*"	" "	" "	" "	" "	"*"	" "	" "	"*"
4	( 1 )	"*"	" "	" "	" "	"*"	"*"	" "	" "	"*"
5	( 1 )	"*"	" "	" "	" "	"*"	"*"	"*"	" "	"*"
6	( 1 )	"*"	" "	" "	" "	"*"	"*"	"*"	"*"	"*"
7	( 1 )	"*"	"*"	" "	" "	"*"	"*"	"*"	"*"	"*"
8	( 1 )	"*"	"*"	"*"	" "	"*"	"*"	"*"	"*"	"*"
9	( 1 )	"*"	"*"	"*"	"*"	"*"	"*"	"*"	"*"	"*"

## 5.2 Redes Neurais

### 5.2.1 Formulas

Após o reconhecimento da importância dos atributos passou-se à definição das formulas que serão passadas como argumento no treino da rede neuronal. Relativamente quer ao nível de fadiga quer à tarefa em execução após vários testes chegamos à conclusão de definir três formulas, a primeira contendo todos os atributos, uma segunda com apenas os quatro com mais relevância e depois por final uma ultima em que só tem o atributo mais importante. Também se concebeu a experiencia de realizar uma formula que combinava no output as duas questões levantadas no problema experimental utilizando como input todos os outros 8 atributos com gama entre [-1,1].

- Formulas para o nível da Fadiga

```
[-----Formula c/ todos os Atributos-----]
```

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

```
[-----Formula c/ 4 Atributos-----]
```

```
formulafadiga2 <- FatigueLevel ~ Performance.MAMean+Performance.MVMean+  
Performance.DDCMean+Performance.Task
```

```
[-----Formula c/ 1 Atributo-----]
```

```
formulafadiga3 <- FatigueLevel ~ Performance.Task
```

- Formulas para a tarefa em execução

```
[-----Formula c/ todos os Atributos-----]
```

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

```
[-----Formula c/ 4 Atributos-----]
```

```
formulatarefa2 <- Performance.Task ~ Performance.KDTMean+Performance.DDCMean+  
Performance.DMSMean+FatigueLevel
```

```
[-----Formula c/ 1 Atributo-----]
```

```
formulatarefa3 <- Performance.Task ~ FatigueLevel
```

- Formula para a tarefa em execução e o nível de fadiga

```
[-----Formula c/ todos os Atributos-----]
```

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

### 5.2.2 Definição da rede neuronal

Para o treino da rede neuronal foi utilizado em R a função `neuralnet`, os parâmetros definidos foram:

- `formula` : formula definida como as vistas anteriormente na secção 5.2.1.
- `data` : data que contém as variáveis da formula
- `hidden` : um vetor de inteiros que define quantos nodos por cada camada
- `threshold` : um valor numérico que especifica o `threshold` , ou valor de erro no qual a função pára
- `lifesign` : String que especifica o quanto vai ser impresso durante a execução
- `linear.output` : Booleano que especifica a utilização dos nodos exteriores

API :

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