

# Universidade do Minho

## Exercício 3

Licenciatura em Engenharia Informática

Sistemas de Representação de Conhecimento e Raciocínio

2ºSemestre (2014/2015)

67673 André Geraldes

67665 Patrícia Barros

67709 Sandra Ferreira

Braga

Março de 2015

# **Resumo**

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

# Índice

RESUMO	2
ÍNDICE	3
ÍNDICE DE FIGURAS	4
ÍNDICE DE TABELAS	5
INTRODUÇÃO	6
PRELIMINARES	7
DESCRIÇÃO DO TRABALHO	8
<b>Identificação dos Níveis de Fadiga</b>	<b>9</b>
Análise de Resultados	9
<b>Identificação da Presença ou Ausência de Fadiga</b>	<b>12</b>
Análise de Resultados	12
<b>Melhor Escala para a Identificação da Fadiga</b>	<b>15</b>
Dificuldades Encontradas	15
Análise de Resultados	16
CONCLUSÕES	18

## Índice de Figuras

Figura 1 - Rede neuronal geradora da solução ótima(1ª alínea).....	11
Figura 2- Rede neuronal geradora da solução ótima (2ª alínea).....	13
Figura 3 - Rede neuronal gerada (3ª alínea).....	16

# Índice de Tabelas

Tabela 1 - Cálculo do erro com distintas definições da rede com 8 métricas a serem utilizadas (1ª alínea) .....	10
Tabela 2 - Cálculo do erro com distintas definições da rede com 5 métricas a serem utilizadas (1ª alínea) .....	10
Tabela 3 - Comparação do resultado esperado com o resultado obtido (1ª alínea) .....	11
Tabela 4 - Cálculo do erro com distintas definições da rede com 8 métricas a serem utilizadas (2ª alínea) .....	12
Tabela 5 - Cálculo do erro com distintas definições da rede com 5 métricas a serem utilizadas (2ª alínea) .....	13
Tabela 6 - Comparação do resultado esperado com o resultado obtido (2ª alínea) .....	14
Tabela 7- Comparação do resultado esperado com o resultado obtido (3ª alínea) .....	17

# Introdução

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

A linguagem utilizada para desenvolver este trabalho será a linguagem **R**.

Neste relatório são apresentadas todas as decisões tomadas pelo grupo e conclusões tiradas em relação aos resultados obtidos.

## Preliminares

De forma a conseguirmos realizar o trabalho proposto foi necessário, através das aulas da Unidade Curricular de Sistemas de Representação de Conhecimento e Raciocínio, possuímos conhecimentos base sobre **RNA's** de forma a conhecermos a forma de utilizar o conhecimento que já possuímos sobre algo para obter conclusões sobre novos dados. Foi ainda necessário conhecermos a linguagem de programação **R**.

## Descrição do Trabalho

Como referido anteriormente este trabalho consiste na utilização de uma **RNA** para a identificação de diferentes níveis de fadiga num Ser-Humano tendo em conta certas métricas relativas à 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 pixéis) 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;
- **Performance.Task:** identificação da tarefa que estava ser executada no momento da recolha dos dados.

Para possibilitar a realização deste trabalho foram-nos fornecidos dados respetivos a observações destas métricas e respetivo nível de fadiga. Os dados que recebemos contém 844 observações que podem ser usadas para o treino da nossa **RNA**.



## Identificação dos Níveis de Fadiga

O primeiro desafio que nos foi proposto neste trabalho foi que treinássemos a rede de forma a esta conseguir identificar 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. Tomamos a decisão de utilizar 600 observações para o treino da rede e as restantes para testar.

```
trainset<-dataset[1:600,]
```

De seguida testamos a rede com diferentes características: variamos o número de neurónios, variamos o algoritmo utilizado e variamos as métricas utilizadas na fórmula de cálculo. Para cada definição da rede foi calculado o desvio padrão do erro entre o valor que deveria ser calculado e o que de facto a rede calculou.

### Análise de Resultados

Segue-se uma tabela com os resultados obtidos quando utilizadas as 8 métricas (todas menos a *Performance.Task*). Todos os resultados foram calculados utilizando um *threshold* de 0.1.

	ALGORITMO	NRº CAMADAS	NRº NODOS 1ª CAMADA	NRº NODOS 2ª CAMADA	ERRO
1	backprop	2	16	8	2.1860
2	rprop+	2	16	8	2.1747

3	rprop-	2	16	8	2.1078
4	backprop	2	20	15	1.7748
5	rprop+	2	20	15	2.0417
6	rprop-	2	20	15	2.1524

Tabela 1 - Cálculo do erro com distintas definições da rede com 8 métricas a serem utilizadas (1ª alínea)

De seguida apresentamos uma tabela semelhante mas agora para testes que realizamos utilizando apenas algumas das variáveis que tínhamos à nossa disposição, que consideramos mais relevantes: *Performance.ADMSLMean*, *Performance.DMSMean*, *Performance.DDCMean*, *Performance.TBCMean* e *Performance.MVMean*.

	ALGORITMO	NRº CAMADAS	NRº NODOS 1ª CAMADA	NRº NODOS 2ª CAMADA	ERRO
1	backprop	2	10	5	1.4344
2	rprop+	2	10	5	1.3982
3	rprop-	2	10	5	2.1078
4	backprop	2	20	15	1.89909

Tabela 2 - Cálculo do erro com distintas definições da rede com 5 métricas a serem utilizadas (1ª alínea)

Facilmente observamos que os resultados utilizando apenas as 5 variáveis escolhidas por nós são mais favoráveis do que utilizando todas as 8. Daqui se conclui que algumas das métricas não são tão influenciadoras do nível de fadiga como outras.

Podemos observar ainda que, em relação à Tabela 2, o algoritmo mais favorável é o algoritmo *rprop+* e que o aumento do número dos nodos das camadas não traz benefício para o resultado, pelo contrário

Posto isto apresentamos agora o desenho da rede correspondente ao melhor resultado obtido.

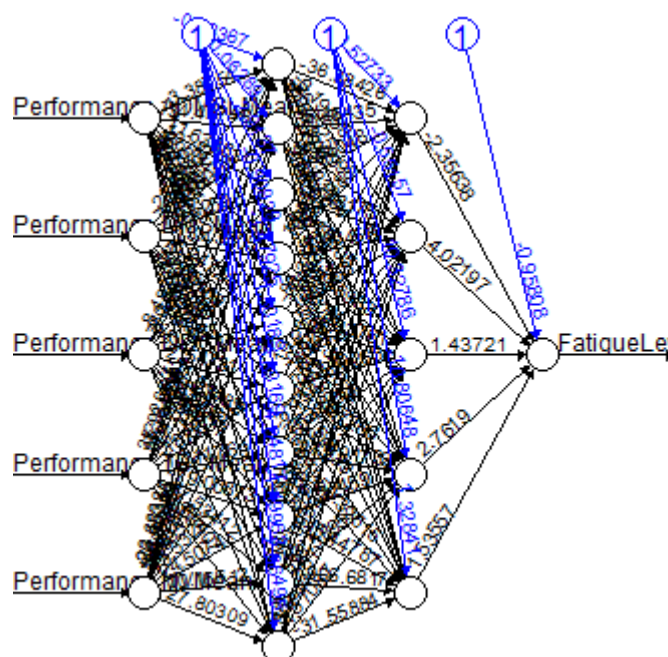


Figura 1 - Rede neuronal geradora da solução ótima(1ª alínea)

Apresentamos ainda um excerto de uma tabela que, utilizando essa mesma rede, compara o resultado esperado com o resultado que foi de facto obtido.

row.names	OutputEsperado	Output	
1	601	3	3.005
2	602	1	1.397
3	603	1	2.005
4	604	2	1.997
5	605	2	2.168
6	606	1	1.642
7	607	1	2.582
8	608	3	2.713
9	609	3	1.903
10	610	4	2.300
11	611	4	2.924
12	612	3	1.393
13	613	3	1.505
14	614	3	3.026
15	615	3	4.426
16	616	3	2.989
17	617	3	2.914

Tabela 3 - Comparação do resultado esperado com o resultado obtido (1ª alínea)

## Identificação da Presença ou Ausência de Fadiga

Outro desafio que nos foi proposto foi treinar a rede para que esta seja capaz de identificar a presença ou ausência de fadiga.

Através da análise dos 7 níveis de fadiga existentes concluímos que a presença de fadiga corresponde aos níveis 4, 5, 6 e 7 e que os restantes níveis correspondem a uma pessoa não fatigada.

Para treinarmos a rede desta forma, formatamos os nossos dados de entrada para que o resultado presente na variável *Fatigue.Level* fosse 0 ou 1, caso apresentasse fadiga ou não, respetivamente.

```
dataset2$FatigueLevel[dataset2$FatigueLevel <= 3] <- 0  
dataset2$FatigueLevel[dataset2$FatigueLevel > 3] <- 1
```

Após feita esta formatação procedemos exatamente da mesma forma que a descrita na secção anterior: treinamos a rede e testamo-la com diferentes definições.

### Análise de Resultados

Apresentamos então uma tabela com os resultados utilizando todas as 8 variáveis.

	ALGORITMO	NRº CAMADAS	NRº NODOS 1ª CAMADA	NRº NODOS 2ª CAMADA	ERRO
1	backprop	2	15	10	0.5899
2	rprop+	2	15	10	0.6273
3	rprop-	2	15	10	0.7935
4	backprop	2	5	3	0.8273
5	rprop+	2	5	3	0.4154
6	rprop-	2	5	3	0.4959

Tabela 4 - Cálculo do erro com distintas definições da rede com 8 métricas a serem utilizadas (2ª alínea)

Segue-se a tabela respetiva aos testes efetuados com apenas 5 variáveis (as mesmas 5 variáveis da secção anterior).

	ALGORITMO	NRº CAMADAS	NRº NODOS 1ª CAMADA	NRº NODOS 2ª CAMADA	ERRO
1	backprop	2	5	3	0.4099
2	rprop+	2	5	3	0.4099
3	rprop-	2	5	3	0.4198
4	backprop	2	15	10	0.5876

Tabela 5 - Cálculo do erro com distintas definições da rede com 5 métricas a serem utilizadas (2ª alínea)

Mais uma vez observamos que o melhor resultado se obtém utilizando apenas as 5 variáveis seleccionadas por nós, sendo que neste caso temos duas hipóteses para o algoritmo a utilizar: *backprop* ou *rprop+*.

Novamente notamos que o aumento do número de nodos nas duas camadas não traz benefício à rede.

De seguida mostramos o desenho da rede com o melhor resultado, utilizando o algoritmo *backprop*.

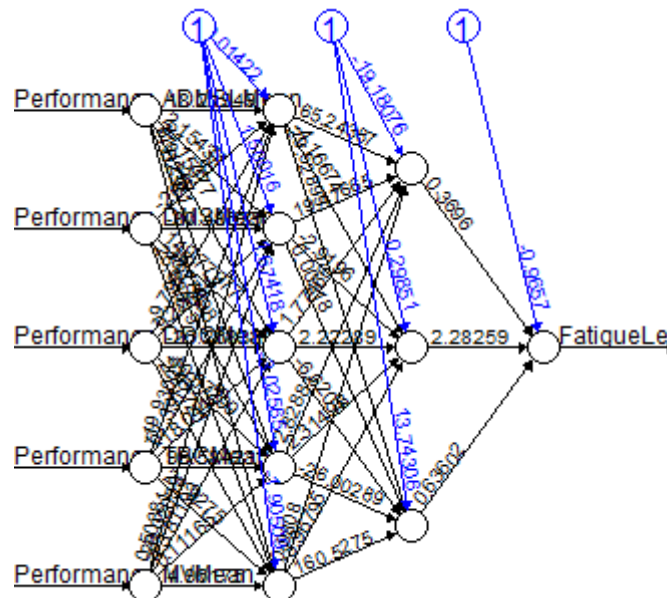


Figura 2- Rede neuronal geradora da solução ótima (2ª alínea)

Segue-se uma tabela comparativa dos resultados que se esperavam obter com os obtidos.

	row.names	OutputEsperado	Output
1	601	0	0.130
2	602	0	0.028
3	603	0	0.235
4	604	0	0.128
5	605	0	0.271
6	606	0	0.082
7	607	0	-0.175
8	608	0	0.003
9	609	0	0.048
10	610	1	0.044
11	611	1	0.133
12	612	0	0.106
13	613	0	0.077
14	614	0	0.043
15	615	0	0.072
16	616	0	0.128
17	617	0	-0.036

Tabela 6 - Comparação do resultado esperado com o resultado obtido (2ª alínea)

## **Melhor Escala para a Identificação da Fadiga**

O último desafio que nos foi proposto foi que criássemos uma nova escala de fadiga, uma escala que seja mais exata e conduza a menos erros: a melhor escala de fadiga.

Para isso, tendo em conta que o número de observações registadas para valores de fadiga acima do nível 3 é muito baixo, e tendo em conta que é a partir desse nível que se considera que a pessoa esteja realmente fatigada decidimos que dividiríamos a escala em 4 níveis distintos, sendo que os primeiros três são correspondentes aos primeiros três níveis descritos no enunciado do projeto e o quarto nível engloba os restantes. Assim sendo passa a nossa escala a ser a seguinte:

1. Totalmente bem;
2. Responsivo, mas não no pico;
3. Ok, normal;
4. Fatigado.

Trata-se de uma escala mais simples mas consideramos que tendo em conta os dados que possuímos seria mais adequada do que a escala com os 7 valores em que, o último nível não tinha sequer nenhuma observação.

### **Dificuldades Encontradas**

Para a realização desta alínea começamos por decidir utilizar o *software* WEKA, apresentado por um dos docentes da Unidade Curricular numa aula prática para perceber melhor como estava organizado o nosso *dataset* e como se dividiam os dados, de forma a tentarmos perceber exatamente em quantos níveis deveríamos dividir o nível de fadiga. Efetuamos ainda alguma pesquisa sobre *clustering*, mas infelizmente, devido à proximidade da data da entrega quando tomamos esta decisão não conseguimos analisar os dados produzidos pelo WEKA da melhor forma, não conseguindo assim chegar a nenhuma conclusão dessa forma e tendo optado pela solução descrita anteriormente.

## Análise de Resultados

Para testar a rede com esta nova escala optamos por utilizar apenas as cinco variáveis que já tínhamos observado resultarem melhor do que as 8. Da mesma forma decidimos utilizar de partida o algoritmo *rprop+* pois observamos nas alíneas anteriores que é o que obtém melhores resultados. Utilizamos duas camadas de nodos com 8 e 5 nodos cada uma, respetivamente.

O erro que obtivemos com estas configurações foi de 0.6069.

Segue-se o desenho da rede.

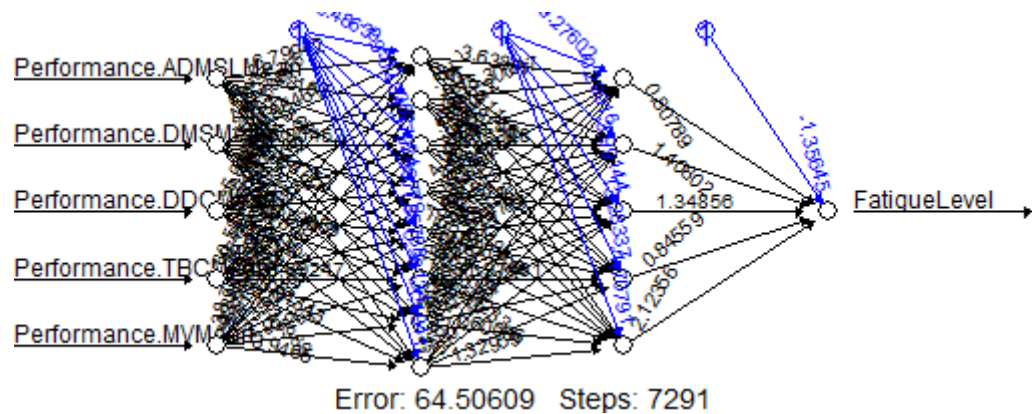


Figura 3 - Rede neuronal gerada (3ª alínea)

De seguida é apresentado um excerto de uma tabela comparativa dos resultados esperados com os resultados obtidos.

	row.names	OutputEsperado	Output
1	601	2	1.5278919561
2	602	1	1.0457096445
3	603	1	0.5447792886
4	604	1	1.0532750458
5	605	1	1.1260782481
6	606	1	1.2514452512
7	607	1	1.2708713547
8	608	2	2.0558811812
9	609	2	1.9576973398
10	610	2	2.0434712772
11	611	2	2.1329206560
12	612	2	1.2604599718
13	613	2	1.3994052285



	row.names	OutputEsperado	Output
14	614	2	1.5666224342
15	615	2	1.6205390199
16	616	2	2.0690036880
17	617	2	1.2676143434

Tabela 7- Comparação do resultado esperado com o resultado obtido (3ª alínea)

## **Conclusões**

A realização deste projeto foi uma forma de interiorizarmos toda a matéria lecionada nas aulas desta Unidade Curricular sobre o tema de Redes Neurais e aplicarmos os conhecimentos aprendidos anteriormente.

Não foi um trabalho de muito fácil realização dado tratar-se de um tema que dá ao a muitas possibilidades diferentes de resolução e a muita subjetividade, trazendo assim alguma confusão na altura da tomada de decisões, levando a que estas tenham sido encontradas maioritariamente por tentativa-erro.

Pessoalmente, consideramos todo o conceito das Redes Neurais um conceito interessante e útil para os mais variados fins.