Universidade do Minho

Escola de Engenharia

Licenciatura em Engenharia Informática

Sistemas de Representação de Conhecimento e Raciocínio

Ano Lectivo de 2014/2015

Exercício 3

Grupo 29:

Adriano Teixeira, a67663

Carlos Morais, a64306

Filipe Ribeiro, a64315

João Farinha, a69302

Maio, 2015

Resumo

No decorrer dos trabalhos realizados na unidade de Sistemas de Representação de Conhecimento e Raciocínio, neste relatório é descrito a evolução do terceiro trabalho proposto, é apresentado os diferentes processos da criação e utilização de Redes Neuronais Artificiais(RNA). Com base em um conjunto de dados biométricos da interação humano-computador com o objetivo de classificar o nível de fadiga.

Para a implementação do problema é utilizada a linguagem de programação R, com a utilização do ambiente de desenvolvimento RStudio apresentado nas aulas praticas.

No desenvolvimento desde relatório serão apresentados todos os passos para a implementação do problema proposto, será feita uma demonstração dos resultados obtidos sendo feita no final uma conclusão de todo o trabalho realizado.

Índice

Índice de Figuras

1. Introdução

Neste exercício é pretendido a utilização e analise de sistemas sub-simbólicos para representação de conhecimento, nomeadamente a utilização de Redes Neuronais Artificiais.

Tendo como base um conjunto de dados fornecidos, pretendesse identificar de melhor forma a nível de fadiga presentes nos mesmos.

Utilizando a biblioteca presente na linguagem R *neuralnet,* abordada nas aulas, em primeiro lugar foi criado uma rede neuronal com base num conjunto de dados de treino retirados dos dados fornecidos, após isso questionamos a rede para determinar o nível de fadiga para uma conjunto de testes, sendo que, para o resultado de cada um destes, atém de indicar o nível de fadiga também seja representado a informação simplesmente de existência ou ausência de fadiga, representado como ”*True*” e “*False*” respectivamente.

Além da rede apresentada foi criado uma outra que, ao contrario da anterior, fosse capaz de determinar o tipo de tarefa que um individuo realizou

No final do relatório serão apresentadas algumas conclusões finais sobre o trabalho.

2. Desenvolvimento

2.1. Tratamento dos dados fornecidos

Numa primeira fase tivemos que ler os dados fornecidos,

```
fadigaData <- read.csv("exercicio3.csv")
```

, partindo *fadigaData* obtido dividimos os 300 primeiros registos para a conseguirmos treinar a rede (*trainset*), os restantes seram utilizados para questionar a rede, e posteriormente analise dos resultados obtidos (*testeset*).

```
trainset <- fadigaData[1:300, ]

testset <- fadigaData[301:844, ]
```

Sendo que o *testset* criado contem a informação sobre a *LevelFadiga* e Performance.Task que é o que se pretende obter, porá corrigir esse problema criamos um *subset* partindo do *testset*, apenas com as biométricas utilizadas.

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

2.2. Determinar o nível de Fadiga

2.2.1 Utilização de 8 Biométricas

Após termos tratado todos os dados, foi criado a rede neuronal de forma a conseguir classificar o nível de fadiga, o principal desafio foi escolher numero de neurónios intermédios a utilizar, tendo como objetivo um menor numero de erro. Na rede criada foi utilizado três conjuntos de neurónios intermédios com a utilização de uma *threshold* de 0.01.

```
fadigaMental <- neuralnet(FatigueLevel~Performance.KDTMean + Performance.MAMean +
                          Performance.MVMean + Performance.DMSMean +
                          Performance.ADMSLMean + Performance.AEDMean +
                           Performance.TBCMean + Performance.DDCMean,
                          trainset, hidden=c(8,10,16), threshold = 0.01)

print(fadigaMental)
```

Error Reached	Threshold	Steps
0.8588743222	0.04922794597	36315

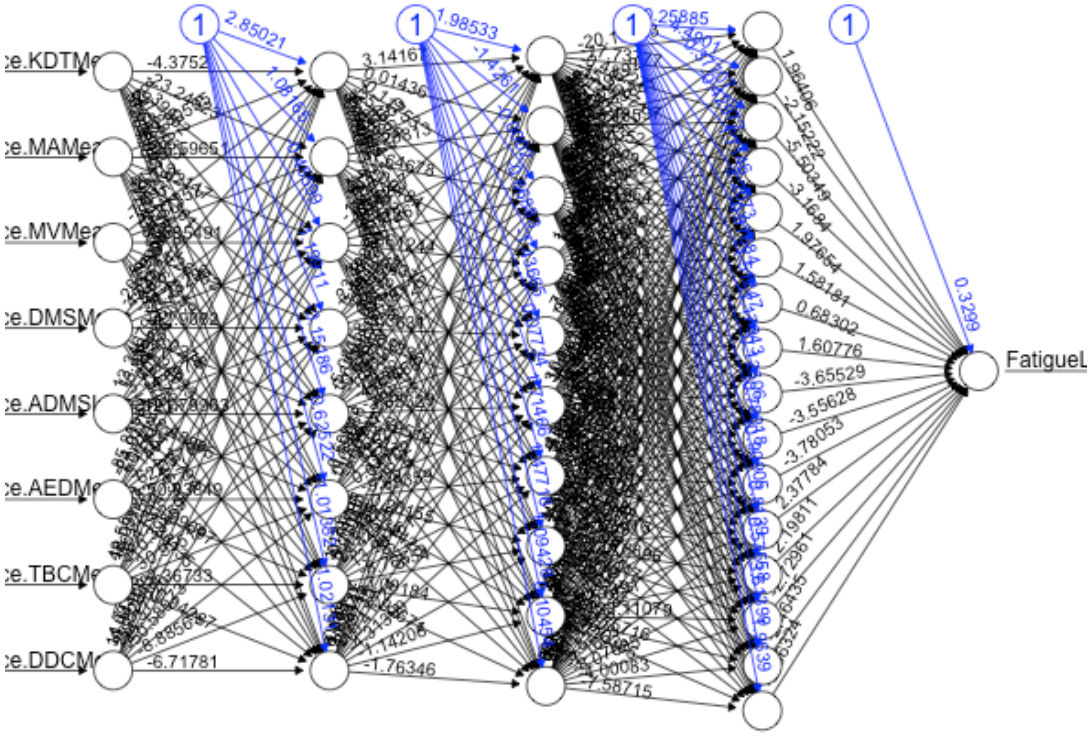

Figure 1 Rede Neuronal FatigueLevel com 8 Biométricas

Após a criação da rede, questionamos a mesma para o *subset_test* criado anteriormente,

```
fadigaMental.results <-compute(fadigaMental, subset_test)
```

Para compararmos o resultado obtido na rede, criamos uma *data.frame* contendo na primeira coluna o nível de fadiga nos dados originais e na segunda a nível obtido após questionar a rede. Além disso acrescentamos uma terceira coluna, que, com base nos resultados obtidos através da rede neuronal, representa de existe ou não fadiga, isto é, se o nível de cansaço for superior a 3 existe fadiga caso contrario não.

```
results <- data.frame(FatigueLevel= testset$FatigueLevel,
                      NetOutput = round(fadigaMental.results$net.result))

results["Fadiga"] <- results$ Fadiga <- ifelse(results$NetOutput>3,"True","False")
```

Exemplo de output:

	FatigueLevel	NetOutput	Fadiga
301	3	3	False
302	3	3	False
303	4	2	False
304	4	3	False
305	4	6	True
306	4	1	False
...	...	...	...

2.2.2 Utilização de 9 Biométricas

Em continuação da rede anterior, utilizamos a biométrica relativa ao tipo de tarefa também como neurónio de entrada, para isso foi necessário converter cada um dos tipos de tarefas para inteiro, sendo, *Work*: -1, *programming*: 0 e *office*: 1. De seguida foi criado uma nova rede neuronal.

Para criar a rede neuronal foi necessário construir um novo *subtest* contendo todas as biométricas utilizadas.

```
subset_test_2 <- subset(testset, select = c("Performance.KDTMean","Performance.MAMean",
                                            "Performance.MVMean","Performance.DMSMean",
                                            "Performance.DDCMean","Performance.ADMSLMean",
                                            "Performance.AEDMean","Performance.TBCMean",
                                             "Performance.Task"))
```

```
fadigaMental_2 <- neuralnet(FatigueLevel~Performance.KDTMean + Performance.MAMean +
                                         Performance.MVMean + Performance.DMSMean +
                                         Performance.ADMSLMean + Performance.AEDMean +
                                         Performance.TBCMean + Performance.DDCMean +
                                         Performance.Task, trainset, hidden=c(8,10,16),
                                         threshold = 0.05)

    # IMPRIMIR INFORMAÇÃO DA REDE
    print(fadigaMental_2)
```

Error Reached	Threshold	Steps
0.5860171434	0.04449010908	69518

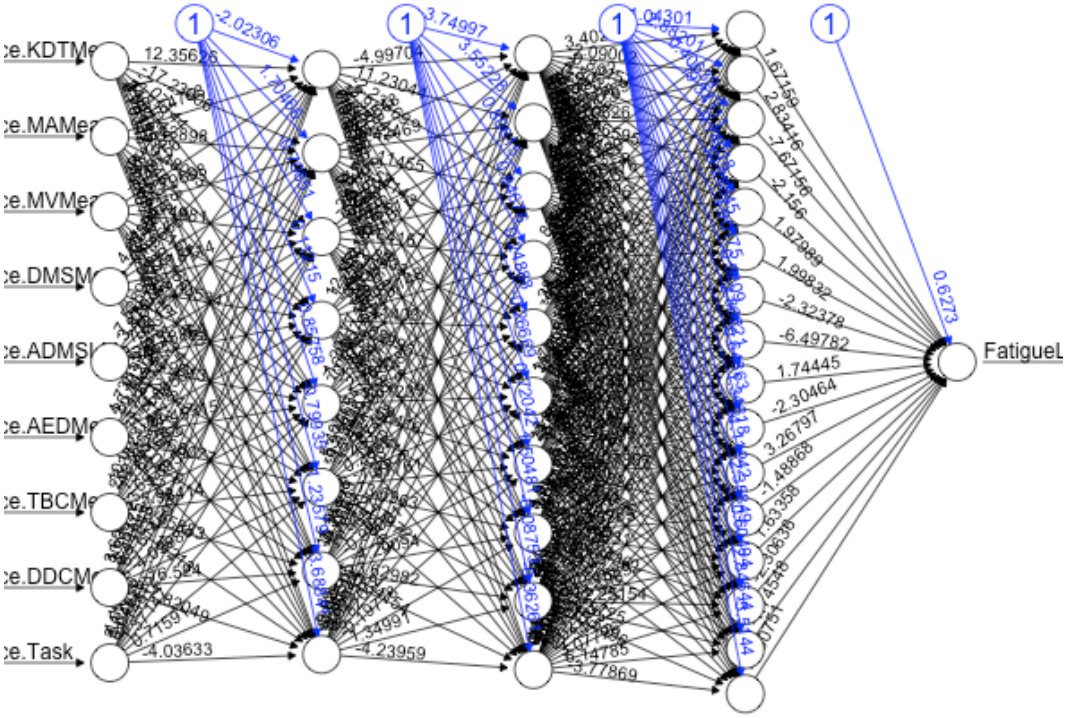

Figure 2 Rede Neuronal FatigueLevel com 9 Biométricas

2.3. Determinar tipo de Tarefa

Para conseguir determinar o tipo da tarefa realizada foi necessário cria uma nova rede neuronal, sendo que a criação desta sofre o mesmo processo das redes anteriores.

```
tasknet <- neuralnet(Performance.Task ~ Performance.KDTMean + Performance.MAMean +
                                        Performance.MVMean  + Performance.TBCMean +
                                        Performance.DDCMean + Performance.DMSMean +
                                        Performance.AEDMean + Performance.ADMSLMean
                                       , trainset, hidden = c(10,16), threshold = 0.01)
```

```
print(tasknet)
        Error Reached          Threshold                Steps
        0.0410406908    0.00974262026    20148
```

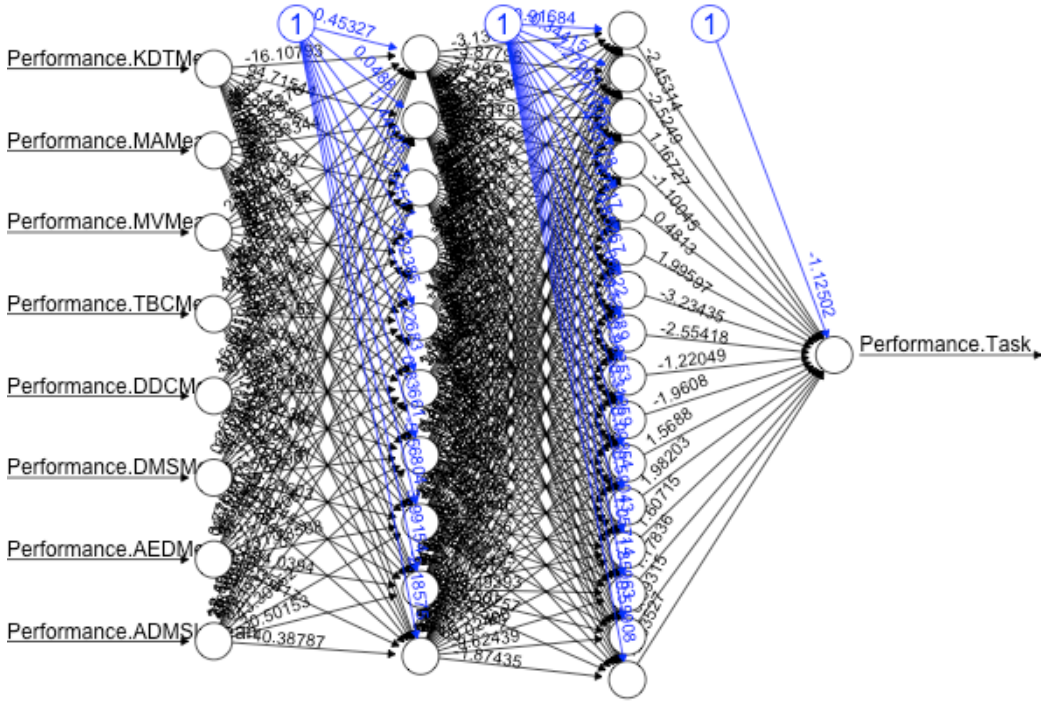

Figure 3 Rede Neuronal Performance.Task

A representação dos resultados obtidos sege a mesma forma que as rede anteriors.

```
tasknet.results <-compute(tasknet, subset_test)


taskResults <- data.frame(Performance.Task = testset$Performance.Task,
                          TaskOutput = round(tasknet.results$net.result))
```

2.4. Manipulação dos Dados

Após criar as diferentes redes, de forma a testar outras variantes do projeto, retiramos algumas das biométricas que achamos serem menus relevantes, ou seja, como as tarefas aqui descritas são maioritariamente de interação através de teclado, retiramos algumas biométricas correspondentes a movimentação do rato, retirando:

- Performance.TBCMean;
- Performance.DDCMean;
- Performance.AEDMean;

Partindo do testeset e o trainset original, criamos para cada um deles um subset contendo apenas as biométricas a utilizar, sendo posteriormente aplicado o mesmo processo das redes anteriores.

```
# TRANSET MODIFICADO
modif_trainset <- subset(trainset, select = c("Performance.KDTMean","Performance.MAMean",
                                              "Performance.MVMean","Performance.DMSMean",
                                              "Performance.ADMSLMean","FatigueLevel",
                                              "Performance.Task"))


# TESTSET MODIFICADO
modif_testset <- subset(testset, select = c("Performance.KDTMean","Performance.MAMean",
                                            "Performance.MVMean","Performance.DMSMean",
                                            "Performance.ADMSLMean"))


# CONSTRUCAO DA REDE
modif_fadigaMental <- neuralnet(FatigueLevel ~ Performance.KDTMean + Performance.MAMean +
                                               Performance.MVMean + Performance.DMSMean +
                                               Performance.ADMSLMean, trainset, hidden=c(15,10),
                                               threshold = 0.05)

print(modif_fadigaMental)
        Error Reached   Threshold       Steps
        7.600003586     0.04688715257   25229
```

Após vários testes não foi possível identificar uma rede com o erro inferior a um. Ou seja, ao contrario do que se pensava inicialmente estas biométricas são importantes para uma correta e precisa representação da rede neuronal.

3. Conclusão

Com a realização deste terceiro exercício foi possível utilizar uma linguagem de programação ainda não conhecida, sendo esta a principal dificuldade na realização deste mesmo trabalho.

O sistema implementado apresenta todas as funcionalidades inicialmente pretendidas, sendo ainda adicionadas novas funcionalidades ao sistema de forma a obter um resultado final mais completo.

Anexos

I. Código

```
library(neuralnet)

# READ FILE
fadigaData <- read.csv("exercicio3.csv")


# ---------------------------------------------------------------------
# 300 CASOS PARA TARAINSET
trainset <- fadigaData[1:300, ]


# O RESTANTE PARA TESTSET
testset <- fadigaData[301:844, ]
# ---------------------------------------------------------------------

# CRIAR SUBSET PARA TESTE
subset_test <- subset(testset, select =
                          c("Performance.KDTMean","Performance.MAMean",
                          "Performance.MVMean","Performance.DMSMean",
                          "Performance.DDCMean","Performance.ADMSLMean",
                          "Performance.AEDMean","Performance.TBCMean"))


# ---------------------------------------------------------------------
#                 PARA PARA DETERMINAR O NIVEL DE FADIGA
# ---------------------------------------------------------------------

# CONSTRUCAO DA REDE
fadigaMental <- neuralnet(FatigueLevel ~ Performance.KDTMean +
                          Performance.MAMean + Performance.MVMean+
                          Performance.DMSMean + Performance.ADMSLMean +
                          Performance.AEDMean + Performance.TBCMean +
                          Performance.DDCMean, trainset,
                          hidden=c(8,10,16), threshold = 0.05)


# IMPRIME RESULTADOS
print(fadigaMental)


# DESENHAR REDE NEURONAL
plot(fadigaMental)
gwplot(fadigaMental)


# TESTAR O TESTSET NA REDE
fadigaMental.results <-compute(fadigaMental, subset_test)
```

```
# RESULTADOS PARA O TESTSET
results <- data.frame(FatigueLevel= testset$FatigueLevel,
                  NetOutput = round(fadigaMental.results$net.result))


# ACRESCENTAR UMA COLUNA (Fadiga) APENAS COM A INFORMACAO True/False,
#                                              SE ESTA CONSADO OU NAO
results["Fadiga "] <- results$Fadiga
                           <- ifelse(results$NetOutput>3,"True","False")


# IMPRIME RESULTADO OBTIDO
print(results)

# ------------------------------------------------------------------------
#             DETERMINAR A FADIGA TENDO EM CONTA A TAREFA
# ------------------------------------------------------------------------
# --------------------
# Work        -> -1  |
# programming ->  0  |
# office      ->  1  |
# --------------------


# CRIAR SUBSET PARA TESTE
subset_test_2 <- subset(testset, select =
                           c("Performance.KDTMean","Performance.MAMean",
                             "Performance.MVMean","Performance.DMSMean",
                             "Performance.DDCMean","Performance.ADMSLMean",
                             "Performance.AEDMean","Performance.TBCMean",
                              "Performance.Task"))


# CONSTRUCAO DA REDE
fadigaMental_2 <- neuralnet(FatigueLevel~Performance.KDTMean +
                                  Performance.MAMean + Performance.MVMean +
                                  Performance.DMSMean + Performance.ADMSLMean +
                                  Performance.AEDMean + Performance.TBCMean +
                                  Performance.DDCMean + Performance.Task,
                                  trainset, hidden=c(8,10,16),
                                  threshold = 0.05)

# IMPRIME RESULTADOS
print(fadigaMental_2)


# DESENHAR REDE NEURONAL
plot(fadigaMental_2)
gwplot(fadigaMental_2)


# TESTAR O TESTSET NA REDE
fadigaMental_2.results <-compute(fadigaMental_2, subset_test_2)


# RESULTADOS PARA O TESTSET
results_2 <- data.frame(FatigueLevel = testset$FatigueLevel,
                      NetOutput = round(fadigaMental_2.results$net.result))


# ACRESCENTAR UMA COLUNA (Fadiga) APENAS COM A INFORMACAO True/False,
#                                              SE ESTA CONSADO OU NAO
results_2["Fadiga"] <- results$Fadiga <-
                           ifelse(results$NetOutput>3,"True","False")
```

```
# IMPRIME RESULTADO OBTIDO
print(results_2)



# ------------------------------------------------------------------
#                 PARA PARA DETERMINAR O TIPO DE TRABALHO
# ------------------------------------------------------------------


# --------------------
# Work        -> -1  |
# programming ->  0  |
# office      ->  1  |
# --------------------


# CONSTRUCAO DA REDE
tasknet <- neuralnet(Performance.Task ~ Performance.KDTMean +
                           Performance.MAMean + Performance.MVMean +
                           Performance.TBCMean + Performance.DDCMean +
                           Performance.DMSMean + Performance.AEDMean +
                           Performance.ADMSLMean, trainset,
                           hidden = c(10,16), threshold = 0.01)


# IMPRIMIR OS RESULTADOS
print(tasknet)


# DESENHAR REDE NEURONAL
plot(tasknet)
gwplot(tasknet)


# TESTAR O TESTSET NA REDE
tasknet.results <-compute(tasknet, subset_test)


# RESULTADOS PARA O TESTSET
taskResults <- data.frame(Performance.Task = testset$Performance.Task,
                      TaskOutput = round(tasknet.results$net.result))


# IMPRIME RESULTADO OBTIDO
print(taskResults)
```

```
# ------------------------------------------------------------------------
# ------------------------------------------------------------------------
#   MIDIFICACOES NOS DADOS FORNECIDOS
# ------------------------------------------------------------------------
# ------------------------------------------------------------------------
# Para testar variantes do projeto, retiramos algumas das biometricas
# que achamos menus relevantes, para ritirar mais conclusoes
# Biometricas retiradas: Performance.TBCMean;
#                        Performance.DDCMean;
#                        Performance.AEDMean;
# Como as tarefas descritas sao principalmente de interacao atraves de
# teclado, nao retiramos todas em relacao a movimentacao do rato,
# mas algumas delas.


# TRANSET MODIFICADO
modif_trainset <- subset(trainset, select =
                           c("Performance.KDTMean","Performance.MAMean",
                           "Performance.MVMean","Performance.DMSMean",
                           "Performance.ADMSLMean","FatigueLevel",
                           "Performance.Task"))


# TESTSET MODIFICADO

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


# DETERMINAR O NIVEL DE FADIGA
# CONSTRUCAO DA REDE
modif_fadigaMental <- neuralnet(FatigueLevel ~ Performance.KDTMean +
                           Performance.MAMean + Performance.MVMean +
                           Performance.DMSMean + Performance.ADMSLMean,
                           modif_trainset, hidden=c(15,10), threshold = 0.01)


# IMPRIME RESULTADOS
print(modif_fadigaMental)


# DESENHAR REDE NEURONAL
plot(modif_fadigaMental)


# TESTAR O TESTSET NA REDE
modif_fadigaMental.results <-compute(modif_fadigaMental, modif_testset)


# RESULTADOS PARA O TESTSET
modif_results <- data.frame(FatigueLevel = testset$FatigueLevel,
       modif_NetOutput = round(modif_fadigaMental.results$net.result))
```

```
# ACRESCENTAR UMA COLUNA (Fadiga) APENAS COM A INFORMACAO True/False,
#                                                SE ESTA CONSADO OU NAO
modif_results["Fadiga "] <- modif_results$Fadiga
             <- ifelse(modif_results$modif_NetOutput>3,"True","False")


# IMPRIME RESULTADO OBTIDO
print(modif_results)


# ------------------------------------------------------------------------
# ------------------------------------------------------------------------
```