

Universidade do Minho - Escola de Engenharia

Mestrado Integrado em Engenharia Informática

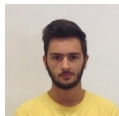
Sistemas de Representação de Conhecimento e Raciocínio

2017/2018 - 2º semestre

Exercício nº3 - Conhecimento não simbólico: Redes Neurais Artificiais

Autores :

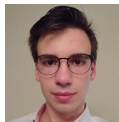
Carlos Campos (A74745)



Diana Costa (A78985)



Marcos Pereira (A79116)



Vitor Castro (A77870)



Braga, 20 de Maio de 2018

Resumo

Perante a proposta de realizar um sistema de representação de conhecimento e de desenvolvimento de mecanismos de raciocínio, nomeadamente, RNAs (Redes Neurais Artificiais), houve um impasse inicial devido à necessidade de uma boa estruturação do problema. Pretende-se abordar o problema de detenção de níveis de exaustão através de dados recolhidos da interação humano-computador. Desta forma, era necessária uma análise inicial ao *dataset* "Bank Marketing", por forma a fazer uma seleção de quais atributos seriam adequados ou não à representação do conhecimento, que influenciariam a escolha final das topologias de rede mais adequadas, e das regras de aprendizagem para treinar as mesmas. Tudo isto requereu a escolha e manipulação acertada de dados, e bom uso da linguagem de programação em R, de modo a que a resolução destes fosse a mais clara e simples possível.

Depois de algum tempo e trabalho, o resultado encontrado foi satisfatório, e os objetivos e respostas às questões do enunciado proposto cumpridos.

Conteúdo

| | | |
|----------|---|-----------|
| 1 | Introdução | 3 |
| 2 | Preliminares | 4 |
| 3 | Descrição do trabalho | 5 |
| 3.1 | Normalização de Dados | 6 |
| 3.1.1 | Conversão de tipos | 6 |
| 3.1.2 | Conversão em escala | 7 |
| 3.1.3 | Arredondamento dos valores | 7 |
| 3.2 | Atributos mais significativos | 9 |
| 3.3 | Fórmulas | 12 |
| 3.4 | Testes | 12 |
| 3.4.1 | Teste 1 | 12 |
| 3.4.2 | Teste 2 | 12 |
| 3.4.3 | Teste 3 | 13 |
| 3.5 | Análise dos resultados | 14 |
| 4 | Conclusões e Sugestões | 15 |

1 Introdução

Este terceiro e último projeto surge no âmbito da unidade curricular de Sistemas de Representação de Conhecimento e Raciocínio, do Mestrado Integrado em Engenharia Informática da Universidade do Minho. O principal objetivo deste exercício é aprofundar e refinar os conceitos do grupo sobre conhecimento não simbólico na representação de conhecimento e no desenvolvimento de mecanismos de raciocínio, utilizando Redes Neurais Artificiais (RNAs) para a resolução de problemas. Mais especificamente, o caso de estudo deste projeto está relacionado com campanhas de marketing baseadas em chamadas telefónicas, de uma instituição bancária portuguesa.

Em suma, a Secção 2 descreve os preliminares necessários ao projeto, enquanto que a Secção 3 descreve a resolução do trabalho, desde a normalização de dados (3.1), definição de atributos mais significativos (3.2), apresentação de fórmulas (3.3) e testes (3.4). Esta secção termina com a consequente análise de resultados, na Secção 3.5.

O relatório termina com uma breve conclusão na Secção 4, onde é feito um balanço do trabalho realizado, tendo em conta as dificuldades ao longo do desenvolvimento do mesmo.

2 Preliminares

Para o desenvolvimento deste exercício prático não foi necessário o estudo de matérias adicionais, apenas da matéria lecionada, até então, na presente unidade curricular. Ainda assim, o grupo considera pertinente declarar, aqui, o estudo inicial que permitiu fazer a transição para o terceiro trabalho prático, correspondente à definição de RNAs, conhecimento, e unidades de processamento da rede neuronal.

Assim sendo, uma RNA é um sistema computacional de base conexionista para resolução de problemas. Estas são concebidas com base num modelo simplificado do sistema nervoso central dos seres humanos, sendo definida por uma estrutura interligada de unidades computacionais, designadas neurónios, com capacidade de aprendizagem. Como tal, assemelha-se ao comportamento do cérebro nos seguintes aspetos:

- O conhecimento é adquirido a partir de um ambiente, através de um processo de aprendizagem;
- O conhecimento é armazenado nas conexões, também designadas por *ligações* ou *sinapses*, entre nodos.

Desta forma, distinguem-se as seguintes unidades de processamento de uma rede neuronal:

- Neurónio: unidades de processamento, responsáveis por coletar e combinar informação vinda de outros neurónios. Caracteriza-se pela sua posição na rede, e pelo valor do estado;
- Dendrite: tem como função receber os estímulos/sinais transmitidos pelos outros neurónios via sinapses;
- Axónio: transmite um estímulo, através das sinapses, para outras unidades de processamento.

A rede neuronal aceita diversos parâmetros de um acontecimento como input, e faz esta informação percorrer a sua rede até que são retornados um ou mais valores de output. A rede vai aprendendo a partir da sua regra de transferência, parâmetro que define o funcionamento da mesma. O cálculo do valor de ativação dos neurónios é afetado pela informação que lhes chega, pelos dados de input e pelo valor de ativação anterior.

Por fim, é importante comentar o poder computacional de uma RNA. Este alicerça-se em dois aspetos fundamentais: por um lado, uma topologia que premeia o paralelismo permite que tarefas complexas sejam realizadas num curto espaço de tempo; por outro lado, a sua capacidade de aprendizagem e generalização conseguem descrever o todo a partir de algumas partes, isto é, é capaz de responder adequadamente a novas situações com base em experiências passadas. São também benefícios de uma RNA a sua transparência, não linearidade, adaptatividade, resposta evidencial, robustez e flexibilidade. O peculiar é que, tal como os seres humanos, as redes neuronais são também suscetíveis ao esquecimento!

3 Descrição do trabalho

Para a realização deste trabalho foi-nos fornecido um conjunto de dados relacionados com campanhas de marketing baseadas em chamadas telefônicas, de uma instituição bancária portuguesa. Estes dados estão organizados numa tabela como a que apresentamos em seguida:

| age | job | marital | education | default | housing | loan | contact | month | day_of_w | duration | campaign | pdays | previous | poutcome | emp.var.r | cons.price | cons.conf | euribor3m | nr.employ | y |
|-----|------------|----------|------------|---------|---------|------|-----------|-------|----------|----------|----------|-------|----------|-----------|-----------|------------|-----------|-----------|-----------|----|
| 56 | housemai | married | basic.4y | no | no | no | telephone | may | mon | 261 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 57 | services | married | high.schoi | unknown | no | no | telephone | may | mon | 149 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 37 | services | married | high.schoi | no | yes | no | telephone | may | mon | 226 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 40 | admin. | married | basic.6y | no | no | no | telephone | may | mon | 151 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 56 | services | married | high.schoi | no | no | yes | telephone | may | mon | 307 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 45 | services | married | basic.9y | unknown | no | no | telephone | may | mon | 198 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 59 | admin. | married | professioi | no | no | no | telephone | may | mon | 139 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 41 | blue-colla | married | unknown | unknown | no | no | telephone | may | mon | 217 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 24 | techniciar | single | professioi | no | yes | no | telephone | may | mon | 380 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 25 | services | single | high.schoi | no | yes | no | telephone | may | mon | 50 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 41 | blue-colla | married | unknown | unknown | no | no | telephone | may | mon | 55 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 25 | services | single | high.schoi | no | yes | no | telephone | may | mon | 222 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 29 | blue-colla | single | high.schoi | no | no | yes | telephone | may | mon | 137 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 57 | housemai | divorced | basic.4y | no | yes | no | telephone | may | mon | 293 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 35 | blue-colla | married | basic.6y | no | yes | no | telephone | may | mon | 146 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 54 | retired | married | basic.9y | unknown | yes | yes | telephone | may | mon | 174 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 35 | blue-colla | married | basic.6y | no | yes | no | telephone | may | mon | 312 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 46 | blue-colla | married | basic.6y | unknown | yes | yes | telephone | may | mon | 440 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 50 | blue-colla | married | basic.9y | no | yes | yes | telephone | may | mon | 353 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 39 | managem | single | basic.9y | unknown | no | no | telephone | may | mon | 195 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 30 | unemploy | married | high.schoi | no | no | no | telephone | may | mon | 38 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |
| 55 | blue-colla | married | basic.4y | unknown | yes | no | telephone | may | mon | 262 | 1 | 999 | 0 | nonexiste | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | no |

Figura 1: Excerto de dados

Para a realização desta tarefa, utilizou-se dois *datasets*: o mais pequeno *dados* e o maior *dadosFull*. O primeiro, com cerca de 4000 entradas, servirá para treinar a rede neural, enquanto o segundo servirá de teste para a mesma. O seguinte comando faz o carregamento do *dataset* fornecido, para que possamos processá-lo.

```
> dados <- read.csv("C:\\Users\\vitor castro\\Desktop\\knowledge-representation-reasoning\\TP3\\bank-additional\\bank-additional.csv", header=TRUE, sep=";", dec=".")
```

Figura 2: Carregamento de dados

3.1 Normalização de Dados

3.1.1 Conversão de tipos

Para iniciar a normalização, é preciso converter todos os tipos de dados para *numeric*, para que depois possamos quantizá-los. Estes são os comandos utilizados.

```
>  
> dados$y <- as.numeric(dados$y)  
> dados$poutcome <- as.numeric(dados$poutcome)  
> dados$day_of_week <- as.numeric(dados$day_of_week)  
> dados$month <- as.numeric(dados$month)  
> dados$contact <- as.numeric(dados$contact)  
> dados$loan <- as.numeric(dados$loan)  
> dados$housing <- as.numeric(dados$housing)  
> dados$default <- as.numeric(dados$default)  
> dados$education <- as.numeric(dados$education)  
> dados$marital <- as.numeric(dados$marital)  
> dados$job <- as.numeric(dados$job)  
>
```

Figura 3: *Dataset dados* factor para numeric

```
> dados$age <- as.numeric(dados$age)  
> dados$duration <- as.numeric(dados$duration)  
> dados$campaign <- as.numeric(dados$campaign)  
> dados$pdays <- as.numeric(dados$pdays)  
> dados$previous <- as.numeric(dados$previous)
```

Figura 4: *Dataset dados* int para numeric

```

>
> dadosFull$age <- as.numeric(dadosFull$age)
> dadosFull$job <- as.numeric(dadosFull$job)
>
> dadosFull$marital <- as.numeric(dadosFull$marital)
> dadosFull$education <- as.numeric(dadosFull$education)
> dadosFull$default <- as.numeric(dadosFull$default)
> dadosFull$housing <- as.numeric(dadosFull$housing)
> dadosFull$loan <- as.numeric(dadosFull$loan)
> dadosFull$contact <- as.numeric(dadosFull$contact)
> dadosFull$month <- as.numeric(dadosFull$month)
> dadosFull$day_of_week <- as.numeric(dadosFull$day_of_week)
> dadosFull$duration <- as.numeric(dadosFull$duration)
> dadosFull$campaign <- as.numeric(dadosFull$campaign)
> dadosFull$pdays <- as.numeric(dadosFull$pdays)
> dadosFull$previous <- as.numeric(dadosFull$previous)
> dadosFull$poutcome <- as.numeric(dadosFull$poutcome)
> dadosFull$emp.var.rate <- as.numeric(dadosFull$emp.var.rate)
> dadosFull$cons.price.idx <- as.numeric(dadosFull$cons.price.idx)
> dadosFull$cons.conf.idx <- as.numeric(dadosFull$cons.conf.idx)
> dadosFull$euribor3m <- as.numeric(dadosFull$euribor3m)
> dadosFull$nr.employed <- as.numeric(dadosFull$nr.employed)
> dadosFull$y <- as.numeric(dadosFull$y)
>

```

Figura 5: *Dataset dadosFull* para numeric

3.1.2 Conversão em escala

Com vista à melhoria de resultados da rede, e ao menor impacto de valores grandes no resultado da mesma, fez-se a conversão dos valores nas colunas para um intervalo de 0 a 1.

A função seguinte mostra como é feito o cálculo de cada célula:

```

> range01 <- function(x){(x-min(x))/(max(x)-min(x))}

```

Figura 6: Função desenvolvida para o cálculo do valor de 0 a 1

Tendo a mesma sido aplicada a todas as colunas dos *datasets* usados. A seguinte imagem exemplifica a aplicação para *age*.

```

> dadosRounded$age <- range01(dadosRounded$age)

```

Figura 7: Convertendo os valores da coluna para um no intervalo definido

3.1.3 Arredondamento dos valores

Para conseguir reduzir o intervalo de valores existentes, procedeu-se ao arredondamento dos valores obtidos no tópico anterior.

```

> dadosRounded <- round(dadosRounded, digits = 2)

```

Figura 8: Arredondamento a duas casas decimais para *dados*


```
>  
> dadosFullRounded <- round(dadosFullRounded, digits = 2)
```

Figura 9: Arredondamento a duas casas decimais para *dadosFull*

3.2 Atributos mais significativos

De modo a identificar os atributos mais significativos para a representação do conhecimento do problema em análise, fizemos uma função que vai ser usada para ensinar a rede neural, com base em todas as colunas do dataset. De seguida fizemos quatro seleções, que usam os 4 métodos diferentes, onde recolhemos a informação consoante o numero de argumentos fornecidos.

```
> funcao <- y ~ age + job + marital + education + default + housing + loan + c
ontact + month + day_of_week + duration + campaign + pdays + previous + poutco
me + emp.var.rate + cons.price.idx + cons.conf.idx + euribor3m + nr.employed
>
```

Figura 10: Função usada para encontrar os atributos mais significativos

```
>
> rnacredito <- neuralnet(funcao, dados, hidden = c(4), lifesign = "full", lin
ear.output = FALSE, threshold = 0.1)
hidden: 4   thresh: 0.1   rep: 1/1   steps:      94   error: 225.59199
time: 0.24 secs
```

Figura 11: Ensinar a rede neural

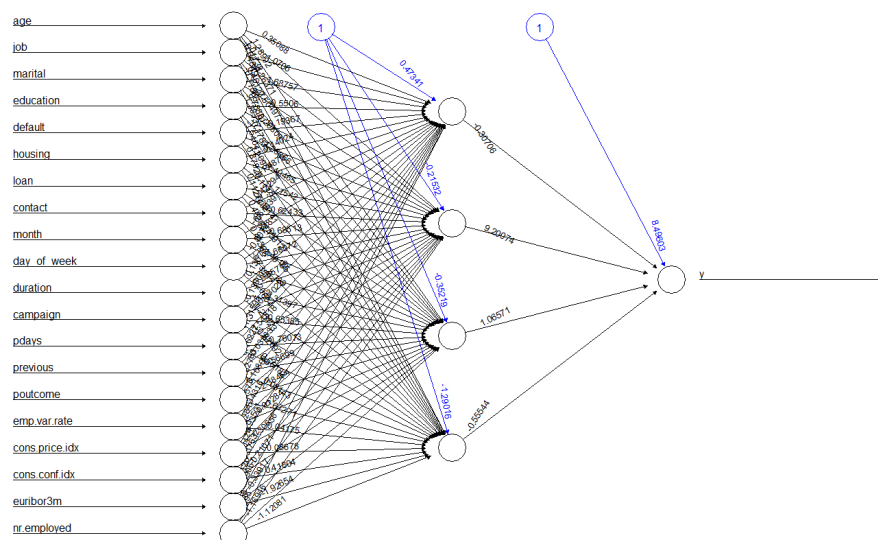


Figura 12: Rede neural gerada

```
> selecao <- regsubsets(funcao, dados)
> summary(selecao)
```

Figura 13: Seleção com o método default


```

selection Algorithm: 'sequential replacement'
1 (1) age job marital education default housing loan contact month day_of_week duration campaign pdays previous poutcome emp.var.rate cons.price.idx
2 (1)
3 (1)
4 (1)
5 (1)
6 (1)
7 (1)
8 (1)
cons.conf.idx euribor3m nr.employed
1 (1)
2 (1)
3 (1)
4 (1)
5 (1)
6 (1)
7 (1)
8 (1)
>

```

Figura 20: Summary da seleção seqreg

Das quatro seleções, escolhemos a que usa o método backward, visto que é a única que não dá importância ao nr.employed, e o grupo achou que esse atributo não tem a relevância dada nos outros métodos.

3.3 Fórmulas

Após identificados os atributos mais significativos (que se mantiveram até o fim dos testes) para cada um dos elementos que pretendemos analisar, prosseguiu-se com o desenvolvimento das fórmulas que serão passadas à rede neuronal. Para isso, com base nos resultados obtidos no subtópico descrito anteriormente, o grupo optou por utilizar uma fórmula. Esta fórmula agrega os atributos sugeridos como significativos no tópico anterior.

```
>
> formula.01 <- y ~ contact + month + duration + pdays + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx
>
```

Figura 21: Formula de cálculo com os atributos significativos

3.4 Testes

Para melhorar os resultados da rede neural, foram realizados alguns testes.

3.4.1 Teste 1

Primeiro foi selecionado o conjunto de atributos do *dadosFull* para teste da RNA.

```
>
> teste.01 <- subset(dadosFull, select = c("contact", "month", "duration", "pdays", "poutcome", "emp.var.rate", "cons.price.idx", "cons.conf.idx"))
>
```

Figura 22: Seleção dos atributos significativos

Foi criada esta RNA, com apenas um nível de profundidade:

```
> rna.01 <- neuralnet( formula.01, dados, hidden = c(4), lifesign = "full", linear.output = FALSE, threshold = 0.1)
hidden: 4   thresh: 0.1   rep: 1/1   steps:    26 error: 225.58138   time: 0.12 secs
>
```

Figura 23: RNA 1

E este foi o resultado obtido:

```
> resultados <- data.frame(atual = dadosFull$y, previsao = rna.01$resultados$net.result)
> resultados$previsao <- round(resultados$previsao, digits=0)
> rmse(c(dadosFull$y),c(resultados$previsao))
[1] 0.3356399427
>
```

Figura 24: Primeiro resultado

3.4.2 Teste 2

De seguida, foi feita a seleção a esses mesmos atributos, mas já normalizados.

```
> teste.02 <- subset(dadosFullRounded, select = c("contact", "month", "duration", "pdays", "poutcome", "emp.var.rate", "cons.price.idx", "cons.conf.idx"))
>
```

Figura 25: Teste aos atributos significativos normalizados

Esta foi a RNA criada:

```
> rna.02 <- neuralnet(formula.01, dadosRounded, hidden =  
c(4), lifesign = "full", linear.output = FALSE, threshold  
= 0.1)  
hidden: 4      thresh: 0.1    rep: 1/1    steps:      15 e  
rror: 225.56518 time: 0.05 secs
```

Figura 26: RNA 2

E este foi o resultado obtido:

```
>  
> resultadosNormalizados <- data.frame(atual = dadosFullR  
ounded$y, previsao = rna.02.resultados$net.result)  
> resultadosNormalizados$previsao <- round(resultadosNorm  
alizados$previsao, digits = 0)  
> rmse(c(dadosFullRounded$y), c(resultadosNormalizados$pr  
evisao))  
[1] 0.3356399427
```

Figura 27: Segundo resultado

3.4.3 Teste 3

Esta foi a RNA criada, tendo sido alterada a profundidade da rede.

```
> rna.03 <- neuralnet(formula.01, dadosRounded, hidden =  
c(4,3,2), lifesign = "full", linear.output = FALSE, thres  
hold = 0.1)  
hidden: 4, 3, 2    thresh: 0.1    rep: 1/1    steps:  
45      error: 225.56836      time: 0.22 secs
```

Figura 28: Teste com uma rede mais profunda

E este foi o resultado obtido:

```
> resultadosNorm <- data.frame(atual = dadosFullRounded$y  
, previsao = rna.03.resultados$net.result)  
> resultadosNorm$previsao <- round(resultadosNorm$previsa  
o, digits = 0)  
> rmse(c(dadosFullRounded$y), c(resultadosNorm$previsao))  
[1] 0.3356399427
```

Figura 29: Terceiro resultado

3.5 Análise dos resultados

A seguinte tabela, apresenta os resultados obtidos com as várias modificações feitas, levando o grupo a concluir que o resultado obtido está perto de ser o melhor possível para este caso.

Tabela 1: Resultados obtidos

| Rede Neural | Threshold | Normalizado | RMSE |
|--------------------|------------------|--------------------|--------------|
| (4) | 0.1 | Não | 0.3356399427 |
| (4) | 0.1 | Sim | 0.3356399427 |
| (4,3,2) | 0.1 | Sim | 0.3356399427 |
| (4,3,2) | 0.05 | Sim | 0.3356399427 |
| (4,3,2) | 0.01 | Sim | 0.3356399427 |
| (5,2) | 0.01 | Sim | 0.3356399427 |

4 Conclusões e Sugestões

A resolução deste último exercício prático foi bastante importante e enriquecedora, pois permitiu aos membros do grupo perceber e interiorizar melhor os conceitos abordados nas aulas práticas da Unidade Curricular de Sistemas de Representação de Raciocínio e Conhecimento. Deste modo, foram adquiridos conhecimentos acerca de programação em R e Redes Neurais Artificiais.

O grande esforço do grupo incidiu em tentar entender quais os mecanismos a usar para normalizar os dados e conseguir fornecer elementos de qualidade à RNA. A restante execução do exercício foi mais simples.

Em suma, é feita uma apreciação positiva relativamente ao trabalho realizado, visto que a implementação de todas as funcionalidades propostas foram conseguidas com sucesso. O grupo conseguiu tirar partido dos conhecimentos adquiridos neste projeto, sentido-se capaz de, num contexto futuro, aplicar os conceitos subjacentes de forma eficaz. É evidente que, num outro contexto (como um projeto de grandes dimensões), seria benéfico que fossem fornecidos mais exemplos práticos e justificação da sua aplicação.

Referências

- [1] [Analide, 2011] ANALIDE, César, NOVAIS, Paulo, NEVES, José,
"Sugestões para a Redação de Relatórios Técnicos",
Relatório Técnico, Departamento de Informática, Universidade do Minho, Portugal, 2011.
- [2] [Cortez, 2000] CORTEZ, Paulo, NEVES, José,
"Redes Neurais Artificiais",
Unidade de Ensino, Departamento de Informática, Universidade do Minho, Braga, Portugal,
2000.