

**Conhecimento sub-simbólico: Redes Neuronais Artificiais**

Bruno Pereira, 69303

João Mano, 69854

Patrícia Rocha, 69636

**Resumo**

Este relatório tem como objetivo explicar e documentar todo o processo de desenvolvimento do terceiro trabalho prático da unidade curricular de Sistemas de Representação de Conhecimento e Raciocínio no âmbito de conhecimento sub-simbólico, nomeadamente, o uso de Redes Neuronais Artificiais, RNAs.

As decisões e soluções implementadas serão aqui explicitadas e detalhadas por forma a apresentar todo o trabalho envolvido neste desafio.

Para a realização desta tarefa foram aplicados todos os conhecimentos, teóricos e práticos, adquiridos durantes as aulas desta unidade curricular.

Tabela de Conteúdos

**Extensão à Programação em Lógica e Conhecimento Imperfeito**

Introdução**6**

Objetivos**7**

Preliminares**7**

Descrição do Trabalho e Análise de Resultados**8**

Base de Conhecimento**8**

Implementação dos Predicados**9**

Predicado demo**9**

Evolução do Predicado demo**10**

Predicado evolEvol**11**

Predicados de Contexto**12**

Conhecimento Imperfeito**12**

Conhecimento Impreciso**12**

Conhecimento Incerto**13**

Conhecimento Interdito**13**

Invariantes **14**

Análise de Resultados**15**

Conclusões e Sugestões**25**

Tabela de Figuras

**­­Figura 1 –** Questão sobre a cor do automóvel a0002**15**

**Figura 2 –** Questão sobre a cor do automóvel a0002**15**

**Figura 3 –** Questão sobre a cor do automóvel a0002**15**

**Figura 4 –** Questão sobre qual o proprietário antigo do automóvel a0002**16**

**Figura 5 –** Questão sobre qual o novo proprietário do automóvel a0004 **16**

**Figura 6 –** Tentativa de inserção do registo do novo proprietário **16**

**Figura 7 –** Tentativa de inserção de matrícula existente **17**

**Figura 8 –** Tentativa de inserção de automóvel com código existente**17**

**Figura 9 –** demoE com verdadeiro e falso**17**

**Figura 10 –** demoE com verdadeiro e desconhecido**18**

**Figura 11 –** demoE com falso e desconhecido**18**

**Figura 12 –** demoE com verdadeiro e verdadeiro**18**

**Figura 13 –** demoOu com verdadeiro e falso**18**

**Figura 14 –** demoOu com verdadeiro e desconhecido**19**

**Figura 15 –** demoOu com falso e desconhecido **19**

**Figura 16 –** demoOu com falso e falso **19**

**Figura 17 –** Inserção de conhecimento verdadeiro repetido**20**

**Figura 18 –** Inserção de conhecimento positivo com negativo na base**20**

**Figura 19 –** Inserção de conhecimento positivo com desconhecido incerto na base **20**

**Figura 20 –** Inserção de conhecimento positivo com desconhecido impreciso na base**20**

**Figura 21 –**

Inserção de conhecimento negativo com positivo na base**21**

**Figura 22 –** Inserção de conhecimento negativo com negativo na base**21**

**Figura 23 –** Inserção de conhecimento negativo com desconhecido incerto na base**21**

**Figura 24 –**

Inserção de conhecimento negativo com desconhecido impreciso na base**21**

**Figura 25 –** Inserção de conhecimento incerto com positivo na base **22**

**Figura 26 –** Inserção de conhecimento incerto com negativo na base **22**

**Figura 27 –** Inserção de conhecimento incerto com conhecimento incerto na base**22**

**Figura 28 –** Inserção de conhecimento incerto com impreciso na base **23**

**Figura 29 –** Inserção de conhecimento impreciso com positivo na base **23**

**Figura 30 –** Inserção de conhecimento impreciso com negativo na base **23**

**Figura 31 –** Inserção de conhecimento impreciso com incerto na base **23**

**Figura 32 –** Inserção de conhecimento impreciso com impreciso na base**24**

**Conhecimento Sub-simbólico: Redes Neuronais Artificiais**

O objetivo deste relatório prende-se na documentação e explicitação do processo de resolução do terceiro exercício prático da unidade curricular se Sistemas de Representação de Conhecimento e Raciocínio.

A realização deste exercício te como objetivo incitar ao uso de sistemas sub-simbólicos face à representação de conhecimento e desenvolvimento de mecanismos de raciocínio.

Ao longo do documento expõem-se as decisões e conceitos definidos relevantes para a resolução do problema.

1. **Introdução**

Redes Neuronais Artificiais são técnicas computacionais, inspiradas no sistema nervoso, que adquirem conhecimento através de processamento de casos base e aprendizagem sobre estes. Estas técnicas permitem solucionar problemas de inteligência artificial através da construção de um processo baseado em circuitos imitadores do processamento e comportamento do sistema neuronal humano, adquirindo conhecimento através de experiências e erros.

As redes neuronais correspondem a grafos, com algumas restrições, onde a informação é distribuída para os nodos pelos caminhos.

O nível de cansaço aquando o uso de um computador pode ser medido pela análise de parâmetros como o movimento do rato ou *clicks*, assim sendo, é possível identificar através de uma recolha de dados o nível de fadiga de um utilizador. Neste contexto, através do uso de uma rede neuronal, teoricamente, será possível treinar a rede (com um erro de previsão associado) para que esta indique qual o nível de cansaço de um indivíduo face a um conjunto de dados sobre a utilização do computador.

Assim sendo, neste trabalho, serão usadas redes neuronais por forma a definir qual o nível de fadiga de um indivíduo.

1. **Objetivos**

Como já foi mencionado, o objetivo deste trabalho é utilizar redes neuronais para calcular níveis de fadiga de um utilizador, exercitando o conhecimento sobre este tema já abordado na unidade curricular.

O trabalho será realizado na linguagem de programação R, onde será criada e treinada a rede neuronal para utilização da mesma.

É então requerido que, face a um conjunto de casos base, se treine uma rede para que os seus valores de decisão retornados sigam, o mais próximo possível, os valores disponibilizados. É ainda pedido que, também utilizando os dados disponibilizados, a escala de valores retornados apenas indique se existe ou não fadiga e, por último, criar uma escala de valores de fadiga que melhor se aproxime do pretendido.

1. **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;
* **Dentrite:** 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.

1. **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 adeque ao problema.

* 1. **Desafio 1 – RNA com Decisão Mais Próxima**

Por forma a obter decisões o mais perto possível das que estão presentes na experiência fornecida foram variados parâmetros como o número de nodos, camadas, os dados teste e os dados de treino, etc.

As primeiras tentativas foram longe do pretendido e não foi possível sequer a rede convergir para um comportamento pois o erro a que esta tinha de obedecer era de uma grandeza muito baixa, aumentando este valor conseguiu-se que a rede fosse treinada e convergisse.

Os testes realizados foram obtidos pela variação de parâmetros, um pouco aleatórios dada a baixa experiência do grupo. Foi então obtida a seguinte tabela – tabela 1 – que identifica os parâmetros usados em cada teste.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Testes | Dados Treino | Dados Teste | Algoritmo | Camadas e Neurónios | *Threshold* | Atributos | Tempo |
| 1 | 1:700 | 701:844 | Default | 2 camadas =(20,16) | 0.01 | 9 (Todos) | 6min. |
| 2 | 1:700 | 701:844 | Default | 2 camadas =(8,6) | 0.1 | 9 (Todos) | 2min. |
| 3 | 1:700 | 701:844 | Default | 2 camadas =(16,8) | 0.1 | 4(KDT,MAM,MVM,TBC) | 3min. |
| 4 | 1:700 | 701:844 | Default | 3 camadas =(20,16,8) | 0.1 | 9 (Todos) | 5min. |
| 5 | 1:700 | 701:844 | Default | 2 camadas =(16,8) | 0.1 | 9 (Todos) | 1min. |
| 6 | 1:700 | 701:844 | Default | 2 camadas =(8,6) | 0.1 | 9 (Todos) | <1min. |
| 7 | 1:700 | 701:844 | sag | 2 camadas =(16,8) | 0.1 | 9 (Todos) | 3min. |
| 8 | 1:700 | 701:844 | Default | 2 camadas =(20,16) | 0.1 | 9 (Todos) | 3min. |

Tabela - Dados usados para testes

Com estes testes foram obtidos os resultados que se podem verificar na tabela 2, dado que o objetivo era “imitar” o comportamento da rede que gerou os dados disponibilizados, foi utilizado um método de escolha que apenas olha ao número de “unidades falhadas”.

|  |  |  |  |
| --- | --- | --- | --- |
| Testes | Error | *Threshold* | *Steps* |
| 1 | - | - | - |
| 2 | 168.7629203 | 0.0935584923 | 18325 |
| 3 | 177.6783951 | 0.08916475274 | 96230 |
| 4 | - | - | - |
| 5 | 62.99818819 | 0.09602170327 | 26713 |
| 6 | 181.0426652 | 0.09735742473 | 12358 |
| 7 | 77.38043285 | 0.09810262053 | 45187 |
| 8 | 15.47814457 | 0.09697254518 | 33153 |

Tabela -Resultados dos testes

Este número foi obtido, para cada teste, através do cálculo da soma das diferenças em módulo do valor previsto pela rede e do valor pretendido, obtendo assim a seguinte tabela – tabela 3.

|  |  |
| --- | --- |
| Testes | Unidades Falhadas |
| 1 | - |
| 2 | 186 |
| 3 | 186 |
| 4 | - |
| 5 | 186 |
| 6 | 182 |
| 7 | 220 |
| 8 | 194 |

Tabela - Unidades Falhadas de cada teste

De todas as variações construídas a mais próxima dos resultados disponibilizados, calculada pela quantidade de unidades falhadas, ou seja, fazendo a diferença entre a resposta da rede neuronal e a resposta pretendida e, somando o módulo dos valores, foi obtida com os parâmetros:

* Casos de treino: [1:700] dos dados disponibilizados;
* Casos de teste: [701:844] dos dados disponibilizados;
* Todos os atributos fornecidos;
* Duas camadas, 8 neurónios na primeira e 6 na segunda;
* *Threshold* de 0.1;
* Algoritmo por *default*;

Onde se conseguiram os seguintes valores:

* 182 unidades falhadas;
* 181.0426652 de erro;
* 0.09735742473 de *threshold*;
* 12358 *steps*.

Correspondente ao teste 6. É de notar que apesar de existirem testes com melhores valores para os restantes parâmetros (por exemplo erro), como o objetivo é atingir os valores de dados teste a decisão baseou-se no teste com menor número de unidades falhadas.

* 1. **Desafio 2 – Escala Identificadora de Fadiga**

Tal como já foi mencionado, neste segundo desafio proposto é pretendido que se altere a escala de sete níveis de fadiga para uma escala com dois níveis, que apenas identifica se existe ou não fadiga.

Seguindo o senso comum e decidindo que estar Ok é ainda não demonstrar qualquer fadiga a escala foi dividida de 1 a 3, inclusive, em não ter qualquer sinal de fadiga e de 4 a 7, inclusive, apresentar já sinais de fadiga.

O método utilizado para a construção das redes neuronais, tal como no desafio anterior, foi a variação de parâmetros como o número de nodos e camadas, a tabela 4 demonstra os valores de cada parâmetro utilizado em cada teste.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Testes | Dados Treino | Dados Teste | Algoritmo | Camadas e Neurónios | *Threshold* | Atributos | Tempo |
| 1 | 1:700 | 701:844 | Default | 2 camadas =(8,6) | 0.1 | 9 (Todos) | 1min. |
| 2 | 1:700 | 701:844 | Default | 2 camadas =(16,8) | 0.1 | 4(DDC,MAM,MVM,AED) | <1min. |
| 3 | 1:700 | 701:844 | Default | 2 camadas =(20,16) | 0.1 | 9 (Todos) | 1min. |
| 4 | 1:700 | 701:844 | Default | 2 camadas =(4,2) | 0.01 | 2 (MAM,MVM) | <1min. |
| 5 | 1:700 | 701:844 | Default | 2 camadas =(8,6) | 0.1 | 4(MAM,MVM,DDC,DMS) | <1min. |
| 6 | 1:700 | 701:844 | Default | 2 camadas =(10,8) | 0.1 | 8 (Todos exceto Task) | <1min. |
| 7 | 1:700 | 701:844 | Default | 2 camadas =(30,15) | 0.1 | 9 (Todos) | 1min. |
| 8 | 1:700 | 701:844 | Default | 2 camadas =(50,40) | 0.1 | 9 (Todos) | 2min. |

Tabela - Dados usados para os testes

Para estes testes os resultados obtidos foram os seguintes:

|  |  |  |  |
| --- | --- | --- | --- |
| Testes | Error | *Threshold* | *Steps* |
| 1 | 26.5667975 | 0.08258607406 | 6797 |
| 2 | 29.10708517 | 0.09630905422 | 18165 |
| 3 | 4.822427697 | 0.09987366954 | 16251 |
| 4 | 51.0215376 | 0.009913180645 | 24248 |
| 5 | 38.10694219 | 0.09929191445 | 11717 |
| 6 | 14.00029961 | 0.0990107609 | 15600 |
| 7 | 1.909048147 | 0.09990094724 | 16516 |
| 8 | 0.2699905278 | 0.09934648589 | 9161 |

Tabela -Resultados dos teste

Durante estas variações foi possível constatar que à medida que o número de neurónios aumenta tanto o erro como o número de passos diminui. Para além disto, ao diminuir o número de atributos de *input* o erro aumenta, o que nos leva a decidir utilizar os 10 atributos para obter um melhor resultado. Tendo isto em conta, e como o objetivo principal será reduzir o erro, o teste que consideramos mais indicado para a resolução deste problema é o teste que utiliza os seguintes parâmetros:

* Casos de treino: [1:700] dos dados disponibilizados;
* Casos de teste: [701:844] dos dados disponibilizados;
* Todos os atributos fornecidos;
* Duas camadas, 50 neurónios na primeira e 40 na segunda;
* *Threshold* de 0.1;
* Algoritmo por *default*;

Onde são obtidos os seguintes resultados:

* 0.2699905278 de erro;
* 0.09934648589 de *threshold*;
* 9161 *steps*.

Note-se que poderia aumentar-se ainda mais o número de nodos mas a variação na diminuição do erro será mínima…

Tal como no desafio anterior, os resultados dos testes encontram-se por completo em anexo na secção de desafio 2.

* 1. **Desafio 3 – Escala Mais Indicada**

Neste último desafio é pedido que se identifique qual a escala mais apropriada para o problema, para isto criaram-se redes neuronais para diferentes escalas e compararam-se os valores dos erros das mesmas.

As escalas criadas e os testes correspondentes foram:

* **Teste 1:** Escala de 7 níveis, idêntica à disponibilizada;
* **Teste 2:** Escala de 6 níveis, esta une o nível 6 e 7 da anterior, em termos de resultados será esperado que sejam similares aos da escala anterior pois nos casos de treino e teste não existe qualquer caso com resultado 7 de nível de fadiga;
* **Teste 3:** Escala de 5 níveis, unindo os níveis 5 e 4 da escala de 7 níveis;
* **Teste 4:** Escala de 5 níveis, unindo os níveis 2 e 3 da escala de 7 níveis;
* **Teste 5:** Escala de 4 níveis, unindo os níveis 1 a 2 da escala de 7 níveis, os níveis 4 e 5 da mesma e finalmente os níveis 6 e 7;
* **Teste 6:** Escala de 3 níveis, obtida pela união dos níveis 1 e 2 da escala com 7 níveis, e os níveis 4,5,6 e 7;
* **Teste 7:** Escala de 3 níveis, unindo os níveis 1,2 e 3, os níveis 4 e 5 e ainda os níveis 6 e 7 da escala de 7 níveis;
* **Teste 8:** Escala de 2 níveis, já utilizada no desafio número dois deste trabalho prático.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Primeira ronda de testes | | | Segunda ronda de testes | | |
| **Erro** | **Threshold** | **Steps** | **Erro** | **Threshold** | **Steps** |
| Teste 1 | 11.07531607 | 0.09863818107 | 38929 | 15.53881078 | 0.09963276669 | 36090 |
| Teste 2 | 10.48619602 | 0.09920660501 | 36535 | 12.28460929 | 0.09644114657 | 42955 |
| Teste 3 | 18.62542792 | 0.09829971592 | 52189 | 15.1568844 | 0.0987377886 | 34861 |
| Teste 4 | 9.140345767 | 0.0935282354 | 35452 | 6.434509689 | 0.09911087559 | 21924 |
| Teste 5 | 10.09699008 | 0.09981006275 | 25109 | 10.08447533 | 0.09911773449 | 37807 |
| Teste 6 | 12.31500399 | 0.08759471436 | 33231 | 5.870310109 | 0.09950673146 | 38582 |
| Teste 7 | 2.876720765 | 0.09674010573 | 25809 | 3.198930515 | 0.09799707195 | 17710 |
| Teste 8 | 3.825953218 | 0.09934976269 | 15608 | 5.392447048 | 0.0937653441 | 13658 |

Tabela - Resultados dos testes

A rede neuronal escolhida foi então a correspondente ao teste 7 da tabela 6, o que nos leva a concluir que a escala mais apropriada é de 3 níveis o que corresponde a:

* Nível 1: Bem sem fadiga visível;
* Nível 2: Estado um pouco fatigado.
* Nível 3: Fatigado, bastante cansado.

Note-se que a fórmula usada para a construção da rede neuronal usada corresponde ao teste 3 do segundo ponto, decidiu-se desta forma porque o erro é já bastante baixo e aceitável e o número de nodos não é exageradamente alto. Note-se também que foram realizadas 2 rondas de teste para que a certeza da escolha fosse estatisticamente mais fiável.

* 1. **Extras – Identificação da Tarefa** 
     1. **1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Testes | Dados Treino | Dados Teste | Algoritmo | Camadas e Neurónios | *Threshold* | Atributos |
| 1 | 1:845 | 600:700 | Default | 2 camadas =(20,16) | 0.01 | 9 (Todos) |
| 2 | 1:845 | 600:700 | Default | 2 camadas =(8,6) | 0.1 | 9 (Todos) |
| 3 | 1:845 | 600:700 | Default | 2 camadas =(16,8) | 0.1 | 4(KDT,MAM,MVM,TBC) |
| 4 | 1:845 | 600:700 | Default | 3 camadas =(20,16,8) | 0.01 | 9 (Todos) |
| 5 | 1:845 | 600:700 | Default | 2 camadas =(16,8) | 0.1 | 9 (Todos) |
| 6 | 1:845 | 600:700 | Default | 2 camadas =(20,16) | 0.1 | 9 (Todos) |

|  |  |  |  |
| --- | --- | --- | --- |
| Testes | Error | *Threshold* | *Steps* |
| 1 | - | - | - |
| 2 | 73.41696511 | 0.09120470748 | 45396 |
| 3 | 58.8425284 | 0.06966731668 | 14962 |
| 4 | - | - | - |
| 5 | 32.7643073 | 0.08971241999 | 23070 |
| 6 | 7.725810367 | 0.09827479757 | 37438 |

|  |  |
| --- | --- |
| Testes | Unidades Falhadas |
| 1 | - |
| 2 | 4 |
| 3 | 15 |
| 4 | - |
| 5 | 5 |
| 6 | 0 |

* + 1. **2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Testes | Dados Treino | Dados Teste | Algoritmo | Camadas e Neurónios | *Threshold* | Atributos |
| 1 | 1:845 | 600:700 | Default | 2 camadas =(8,6) | 0.1 | 9 (Todos) |
| 2 | 1:845 | 600:700 | Default | 2 camadas =(16,8) | 0.1 | 9 (Todos) |
| 3 | 1:845 | 600:700 | Default | 2 camadas =(30,15) | 0.1 | 4(KDT,MAM,MVM,TBC) |

|  |  |  |  |
| --- | --- | --- | --- |
| Testes | Error | *Threshold* | *Steps* |
| 1 | 70.35132917 | 0.09726379024 | 16641 |
| 2 | 62.73343497 | 0.09762023485 | 48475 |
| 3 | 2.600991211 | 0.09752877172 | 48475 |

|  |  |
| --- | --- |
| Testes | Unidades Falhadas |
| 1 | 15 |
| 2 | 12 |
| 3 | 0 |

1. **Conclusões e Sugestões**

Uma das conclusões a que chegamos na variação de parâmetros é que o número de nodos da primeira camada tem de ser igual ou maior ao número de *inputs*, e se for maior o erro é mais pequeno quanto maior(até um ponto) mas a rede demora mais tempo a aprender, uma outra conclusão é que o threshold demasiado baixo faz com que a rede não convirja para um comportamento, diminuindo o numero de atributos input apesar ser calculado muito mais rápido o erro aumenta bastante, ou seja quanto maior o número de atributos deste problema com que a rede é treinada mais eficaz é a sua decisão, menor é o erro da sua decisão