



Instituto Politécnico de Castelo Branco
Escola Superior de Tecnologia

Escola Superior de Tecnologia
Curso de 1.º Ciclo em Engenharia Informática
1.º Semestre do 2.º Ano
Unidade Curricular: Inteligência Artificial
Docentes: Ana Paula Silva
Hugo Santos
Trabalho Prático

Trabalho Prático

Treino de uma rede neuronal para identificar estados de depressão em alunos do ensino superior

1. Objetivo

O trabalho prático de Inteligência Artificial tem como objetivo a aplicação dos conceitos apresentados nas aulas teórico-práticas sobre agentes aprendizes usando a linguagem de programação Python. Para tal, os estudantes devem utilizar uma rede neuronal multicamada de maneira que esta aprenda a determinar se um estudante está num estado depressivo a partir de um conjunto de atributos. É fornecido código que permite treinar uma rede para imitar funções binárias simples, o qual deverá ser adaptado a este novo problema.

2. Funcionamento

- O trabalho deverá ser realizado em grupos de dois alunos.
- Deverá ser implementado utilizando a linguagem usada nas aulas práticas (Python).
- Apenas devem ser usadas classes introduzidas nas aulas práticas.
- **O trabalho está sujeito a defesa.**

3. O Problema

Neste trabalho pretendemos implementar um agente baseado em redes neuronais capaz de prever se um estudante está num estado depressivo com base num conjunto de atributos: É fornecido código que define uma rede neuronal multicamada e implementa o mecanismo de treino por retro-propagação do erro. Também é fornecido código exemplificativo de como utilizar uma rede neuronal para imitar funções booleanas simples como o AND, o OR e o XOR.

Pretende-se adaptar este código para treinar uma rede capaz de receber os atributos de um estudante e prever se o mesmo está num estado depressivo. Para o treino é fornecido um ficheiro adicional (Depression Student Dataset.csv) que contém exemplos de 512 estudantes, descritos por uma lista de 11 atributos:



Instituto Politécnico de Castelo Branco
Escola Superior de Tecnologia

Escola Superior de Tecnologia
Curso de 1º Ciclo em Engenharia Informática
1º Semestre do 2º Ano
Unidade Curricular: Inteligência Artificial
Docentes: Ana Paula Silva
Hugo Santos
Trabalho Prático

| Gender | Age | Academic Pressure | Study Satisfaction | Sleep Duration | Dietary Habits | Have you ever had suicidal thoughts ? | Study Hours | Financial Stress | Family History of Mental Illness | Depression |
|--------|-----|-------------------|--------------------|-------------------|----------------|---------------------------------------|-------------|------------------|----------------------------------|------------|
| Male | 28 | 2.0 | 4.0 | 7-8 hours | Moderate | Yes | 9 | 2 | Yes | No |
| Male | 28 | 4.0 | 5.0 | 5-6 hours | Healthy | Yes | 7 | 1 | Yes | No |
| Male | 25 | 1.0 | 3.0 | 5-6 hours | Unhealthy | Yes | 10 | 4 | No | Yes |
| Male | 23 | 1.0 | 4.0 | More than 8 hours | Unhealthy | Yes | 7 | 2 | Yes | No |
| Female | 31 | 1.0 | 5.0 | More than 8 hours | Healthy | Yes | 4 | 2 | Yes | No |
| Male | 19 | 4.0 | 4.0 | 5-6 hours | Unhealthy | Yes | 1 | 4 | Yes | Yes |
| Female | 34 | 4.0 | 2.0 | More than 8 hours | Moderate | Yes | 6 | 2 | No | Yes |
| Female | 20 | 4.0 | 1.0 | More than 8 hours | Healthy | Yes | 3 | 4 | Yes | Yes |
| Female | 33 | 1.0 | 4.0 | More than 8 hours | Moderate | No | 10 | 3 | No | No |
| Male | 33 | 4.0 | 3.0 | Less than 5 hours | Unhealthy | Yes | 10 | 1 | No | Yes |
| Female | 31 | 5.0 | 4.0 | 5-6 hours | Healthy | Yes | 6 | 4 | No | Yes |
| Male | 24 | 2.0 | 1.0 | 7-8 hours | Unhealthy | Yes | 11 | 5 | No | Yes |
| Female | 23 | 5.0 | 5.0 | Less than 5 hours | Unhealthy | Yes | 2 | 1 | Yes | Yes |
| Male | 25 | 1.0 | 1.0 | 5-6 hours | Moderate | Yes | 12 | 3 | Yes | Yes |

O último atributo é a classe do estudante (depressão ou não depressão) e não deverá ser utilizado no treino. Os anteriores 10 atributos são de dois tipos: numéricos e categóricos e deverão ser utilizados para criar as entradas para a rede. Os categóricos deverão ser transformados numa representação adequada, uma vez que as entradas de uma rede neuronal têm que ser valores numéricos. A rede deverá ter duas saídas, uma para cada possibilidade: Depressão ou Não Depressão.

Observação: A rede poderia ter apenas uma unidade na camada de saída, mas para este trabalho vamos considerar duas classes e por isso duas unidades na camada de saída.

4. Código disponibilizado

Antes de iniciar o trabalho propriamente dito, o estudante deverá analisar o código fornecido, o qual está comentado de forma a descrever o seu funcionamento. Deverá ainda testar o funcionamento das funções `train_and`, `train_or` e `train_xor`. Ao fazer 'Evaluate' do ficheiro fornecido já irá observar o treino de uma rede para aprender a função lógica AND e depois a resposta do modelo treinado quando lhe são fornecidas as combinações lógicas possíveis. É importante que nesta fase os estudantes testem o código para diferentes valores de número de épocas e para as diferentes funções booleanas.

5. Construção dos Conjuntos de Treino e Teste

Numa primeira fase vamos implementar a função `build_sets` e uma função auxiliar chamada `translate` que irá ser chamada pela anterior. A função `build_sets` cria os conjuntos de treino e teste que irão ser



utilizados no treino e avaliação da rede neuronal a partir dos dados armazenados no ficheiro **Depression Student Dataset.csv**.

Como já foi referido, este ficheiro apresenta informação sobre 512 estudantes. A função `build_sets` deve receber como argumento o número de exemplos que se pretende que sejam usados para os conjuntos de treino e de teste. A função deve ler cada linha e transformá-la numa lista de valores, tendo em atenção o tipo dos valores lidos. A lista é posteriormente passada como argumento à função `translate`, a qual construirá um padrão de treino no formato adequado, discutido mais abaixo. A lista resultante será armazenada numa lista de padrões, cuja ordem deve ser randomizada (consultar método `shuffle` disponível no módulo `random`). Finalmente, a função deverá devolver duas listas, o conjunto de treino e o conjunto de teste de tamanhos iguais aos valores recebidos como argumento.

A função `translate` recebe cada lista de valores e transforma-a num padrão de treino. Cada padrão é uma lista com o formato `[padrao_de_entrada, classe_do_estudante, padrao_de_saida]`, em que os elementos da lista são obtidos da seguinte forma:

- `padrao_de_entrada` é uma lista de valores numéricos. Os valores categóricos devem ser convertidos em sequências binárias, usando a codificação one-hot. Por exemplo, para o atributo **'Dietary Habits'** os valores possíveis são: Healthy, Moderate, Unhealthy. Considerando este alinhamento um estudante com:

'Dietary Habits' = Healthy

Este valor deve ser transformado em: **1 0 0**, considerando o alinhamento Healthy, Moderate, Unhealthy. Resumindo,

'Dietary Habits' = Healthy deve ser transformado em 1 0 0

'Dietary Habits' = Moderate deve ser transformado em 0 1 0

'Dietary Habits' = Unhealthy deve ser transformado em 0 0 1

Ou seja, para este atributo serão necessárias 3 unidades de entrada. O mesmo deve ser feito para todos os atributos categóricos.

- **`classe_do_estudante`** é uma de duas strings **'yes'** (tem depressão) ou **'no'** (não tem depressão) que corresponde ao último atributo do exemplo de treino.



Instituto Politécnico de Castelo Branco
Escola Superior de Tecnologia

Escola Superior de Tecnologia
Curso de 1.º Ciclo em Engenharia Informática
1.º Semestre do 2.º Ano
Unidade Curricular: Inteligência Artificial
Docentes: Ana Paula Silva
Hugo Santos
Trabalho Prático

- **padrao_de_saida** é uma lista de 0 e 1 que representa a classe do estudante ('yes' - tem depressão ou 'no' - não tem depressão), com apenas dois elementos: [1 0] -> 'yes' e [0,1] -> 'no'.

Em resumo, e para exemplificar, a primeira linha do ficheiro mushrooms.csv deve resultar no seguinte padrão:

| Gender | Age | Academic Pressure | Study Satisfaction | Sleep Duration | Dietary Habits | Have you ever had suicidal thoughts ? | Study Hours | Financial Stress | Family History of Mental Illness | Depression |
|--------|-----|-------------------|--------------------|----------------|----------------|---------------------------------------|-------------|------------------|----------------------------------|------------|
| Male | 28 | 2.0 | 4.0 | 7-8 hours | Moderate | Yes | 9 | 2 | Yes | No |

O qual deve ser transformado em:

[0, 1, 28, 2.0, 4.0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 9, 2, 1, 0], 'no', [0,1]]

Observações:

Sleep Duration tem 4 valores possíveis. Na representação em cima foi usado o alinhamento:

<5; 5-6; 7-8; >8 Assim, o valor '7-8 hours' foi transformado em 0 0 1 0

| Sleep Duration |
|-------------------|
| 7-8 hours |
| 5-6 hours |
| 5-6 hours |
| More than 8 hours |
| More than 8 hours |
| 5-6 hours |
| More than 8 hours |
| More than 8 hours |
| More than 8 hours |
| Less than 5 hours |

Dietary Habits tem 3 valores possíveis. Na representação em cima foi usado o alinhamento Healthy, Moderate, Unhealthy. Assim, o valor 'Moderate' foi transformado em 0 1 0.



Instituto Politécnico de Castelo Branco
Escola Superior de Tecnologia

Escola Superior de Tecnologia
Curso de 1.º Ciclo em Engenharia Informática
1.º Semestre do 2.º Ano
Unidade Curricular: Inteligência Artificial
Docentes: Ana Paula Silva
Hugo Santos
Trabalho Prático

| Dietary Habits |
|----------------|
| Moderate |
| Healthy |
| Unhealthy |

Os restantes atributos categóricos têm apenas dois valores possíveis: ‘yes’ ou ‘no’. Neste casos considerou-se o alinhamento ‘yes’, ‘no’. Assim o valor ‘yes’ foi transformado em **1, 0** e o ‘no’ em **0, 1**.

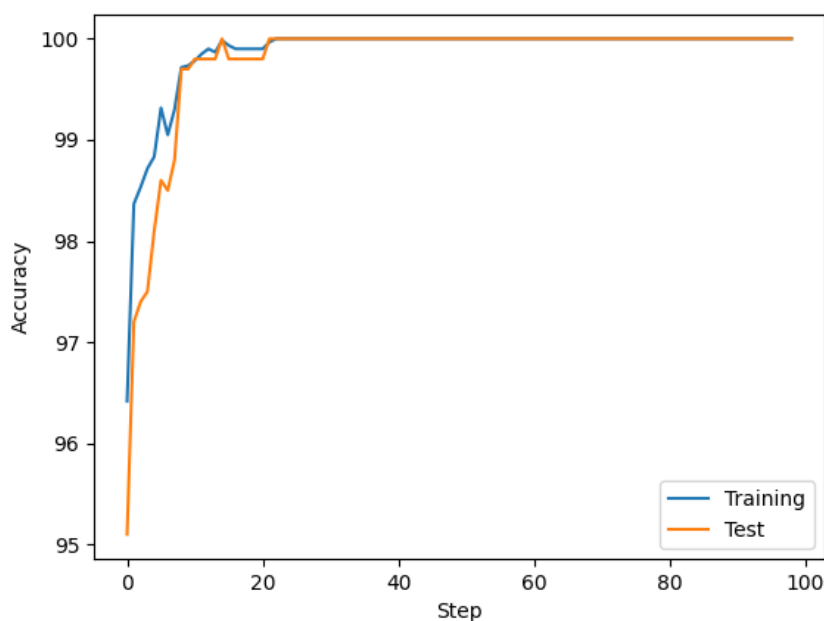
6. Treino da Rede Neural

Nesta segunda fase vamos implementar três funções. A primeira função chama-se `train_students`, recebe o conjunto de treino e o número de épocas (e outros parâmetros documentados no código) e deve criar a rede neuronal e chamar a função `iterate` para a treinar durante **o número de épocas indicado no argumento**. Esta função funciona de forma semelhante às funções de treino fornecidas no código, devendo, em cada iteração de treino, percorrer todos os padrões armazenados no conjunto de treino, chamando a função `iterate`, a qual recebe um padrão de entrada (lista dos atributos do estudante) e de saída (lista com a codificação binária da classe do estudante).



Instituto Politécnico de Castelo Branco
Escola Superior de Tecnologia

Escola Superior de Tecnologia
Curso de 1.º Ciclo em Engenharia Informática
1.º Semestre do 2.º Ano
Unidade Curricular: Inteligência Artificial
Docentes: Ana Paula Silva
Hugo Santos
Trabalho Prático



Esta função deve ainda testar, no final de cada época, a rede tanto com o conjunto de treino como de teste (ver função seguinte) e ir armazenando os valores em listas para no fim do treino, usando o package matplotlib, apresentar um gráfico da precisão da rede nos dois conjuntos semelhante ao da figura acima.

A segunda função, `test_students`, tem como objetivo testar o desempenho da rede treinada. Para tal deve percorrer o conjunto de exemplos recebido como argumento, e, para cada padrão, chamar a função `forward` passando como argumentos a rede treinada e a lista de atributos do estudante. De seguida, a função deve analisar a lista de saídas da rede (`net['y']`), chamando a terceira função a implementar (`retranslate`). Esta função deve devolver a classe do estudante correspondente à saída da rede com maior valor. A função `test_students` deve comparar o valor devolvido pela função auxiliar com a verdadeira classe do estudante, armazenada no padrão de treino, calculando a percentagem de respostas corretas. Para cada padrão deve ainda imprimir uma linha semelhante à seguinte:

```
The network thinks student number 3 has depression, it should have depression
```

Após a impressão das linhas com o resultado referente a cada padrão de treino, deve ainda imprimir o valor do desempenho final da rede (percentagem de respostas corretas), por exemplo:



Instituto Politécnico de Castelo Branco
Escola Superior de Tecnologia

Escola Superior de Tecnologia
Curso de 1.º Ciclo em Engenharia Informática
1.º Semestre do 2.º Ano
Unidade Curricular: Inteligência Artificial
Docentes: Ana Paula Silva
Hugo Santos
Trabalho Prático

Success rate: 94.12

Estas impressões só devem ocorrer se o parâmetro `printing for True`, se tal não acontecer a função deve devolver apenas o valor da precisão.

Por último, a função `run_students`, que será a função principal do nosso programa, deverá reunir as funções anteriormente implementadas: desde a construção dos conjuntos de treino e teste, até ao treino e teste da rede. Deve receber como argumento o *dataset*, o número de épocas que irá ser usado para o treino, o número de exemplos no conjunto de treino e o número de instâncias no conjunto de teste, além dos outros parâmetros presentes no código fornecido.

7. O que deve ser entregue

Os alunos devem submeter no moodle o ficheiro `.py` com o código desenvolvido (os nomes dos alunos devem ser colocados em comentário no início do ficheiro), assim como um ficheiro PDF com um pequeno relatório, onde apresentam os resultados obtidos no trabalho experimental realizado. O relatório deve ter as seguintes secções:

Trabalho Experimental – aqui devem apresentar os resultados obtidos no trabalho experimental, incluindo tabelas com os valores de precisão da rede e os gráficos produzidos por cada treino.

Discussão de Resultados – aqui devem discutir os resultados obtidos. Os alunos devem relacionar o impacto que a codificação dos dados, a configuração da rede, a dimensão dos conjuntos de teste e treino, assim como o número de épocas tem no desempenho da rede.

O trabalho experimental deve ser efetuado para 3 redes diferentes (com o número de unidades da camada escondida respetivamente igual a 4, 7 e 11) e valores de número de épocas de treino de 5, 50 e 100 para conjuntos de treino e teste com as seguintes dimensões:

- Conjunto de treino com 280 exemplos e conjunto de teste com 128 exemplos.
- Conjunto de treino com 384 exemplos e conjunto de teste com 128 exemplos.

Este trabalho experimental deve ser realizado usando a codificação descrita no enunciado, e depois usando a mesma codificação mas normalizando os valores numéricos correspondentes aos atributos: Age (considere a idade máxima 100), Academic Pressure, Study Satisfaction, Study Hours and Financial Stress.