09.01.20

14.00 - 16.00pm

CMPU 3051 Intro to Artificial Intell

Basement 3, Kevin Street

Programme Code: DT228
Module Code: CMPU 3051
CRN: 30102

TECHNOLOGICAL UNIVERSITY DUBLIN

KEVIN STREET CAMPUS

BSc. (Honours) Degree in Computer Science

Year 3

SEMESTER 1 EXAMINATIONS 2019/20

Introduction to Artificial Intelligence

Mr. Richard Lawlor
Dr. Deirdre Lillis
Dr. Martin Crane

Two Hours

Instructions to candidates

ANSWER **FOUR** QUESTIONS OUT OF **FIVE**.
ALL QUESTIONS CARRY EQUAL MARKS.

1 Financial Advisor Knowledge

The function of the advisor is to help a user decide whether to invest in a savings account or the stock market. Some investors may want to split their money between the two. The investment that will be recommended for individual investors depends on their income and the current amount they have saved according to the following criteria:

1. Individuals with an inadequate savings account should always make increasing the amount saved their first priority, regardless of their income.
2. Individuals with an adequate savings account and an adequate income should consider a riskier but potentially more profitable investment in the stock market.
3. Individuals with a lower income who already have an adequate savings account may want to consider splitting their surplus income between savings and stocks, to increase the cushion in savings while attempting to increase their income through stocks.

The adequacy of both savings and income is determined by the number of dependents an individual must support. The rule is to have at least €5,000 in the bank for each dependent. An adequate income must be a steady income and supply of at least €15,000 per year plus an additional €4,000 for each dependent.

Note that if the amount saved is greater than the minimum savings, then the savings account is deemed adequate, otherwise inadequate. Similarly with the adequacy/inadequacy of the income.

(a) Express the above knowledge in Predicate Calculus in terms of the following predicates and functions:

amount_saved/1, dependents/1, earnings/2
savings_account/1, income/1, investment/1
minsavings(Y) and minincome(Y) (functions).

(8 marks)

(b) Translate your knowledge described in part (a) into Prolog.

(8 marks)

(c) Describe the architecture of a rule-based expert system.

(9 marks)

2 (a) Write down Bayes' rule.

Then use it to calculate P(m | s) in the following situation where s is the proposition that a patient has a stiff neck and m is the proposition that a patient has meningitis. The doctor know some unconditional facts: prior probability of a patient having meningitis is 1/50,000, prior probability of any patient having a stiff neck is 1%. The doctor also knows that having meningitis causes a stiff neck 70% of the time.

(7 marks)

(b) Give three reasons why Bayesian reasoning is difficult to use in an expert system. Also, briefly describe two advantages in using certainty factors as opposed to Bayesian reasoning.

Briefly mention two well-known historical expert systems, one of which used Bayesian reasoning and the other which used certainty factors.

(10 marks)

(c) Given the forecasting rules below, show how they would fire to forecast tomorrow's weather when provided with the following information: there is rain today and the rainfall is low with a certainty factor of 0.8, and also it is cold with a certainty factor of 0.9. This formula may be useful:

$$cf(cf_1, cf_2) = cf_1 + cf_2 \times (1 - cf_1)$$

```
Rule: 1
        if today is rain
        then tomorrow is rain {cf 0.5}

Rule: 2
        if today is dry
        then tomorrow is dry {cf 0.5}

Rule: 3
        if today is rain
        and rainfall is low
        then tomorrow is dry {cf 0.6}

Rule: 4
        if today is rain
        and rainfall is low
        and temperature is cold
        then tomorrow is dry {cf 0.7}

Rule: 5
        if today is dry
        and temperature is warm
        then tomorrow is rain {cf 0.65}

Rule: 6
        if today is dry
        and temperature is warm
        and sky is overcast
        then tomorrow is rain {cf 0.55}
```

(8 marks)

3 (a) Given facts like the following:

```
bicycleLane(dublin, kilkenny, 120).
bicycleLane(dublin, carlow, 80).
bicycleLane(kilkenny, waterford, 85).
bicycleLane(mallow, carlow, 110).
bicycleLane(cork, mallow, 38).
bicycleLane(waterford, youghal, 110).
bicycleLane(youghal, cork, 30).
bicycleLane(dublin, athlone, 105).
bicycleLane(athlone, limerick, 135).
bicycleLane(limerick, cork, 75).
```

write a Prolog predicate **canGoFromTo(A, B, Dist)** which will tell you if you can ride a bicycle from town A to town B even if you have to pass thru other towns, and it will also compute the total distance. You can assume that bicycle lanes are one-way and no loops exist.

(5 marks)

(b) Write a Prolog predicate **f(X, N, Result)** to compute the function f(x,n) = x^n . You can assume n is an integer and n >= 0.

(5 marks)

(c) Write a Python function myDotProd(a,b) which computes the dot-product of 2 vectors of equal size. You are not to use the numpy.dot() operation.

(5 marks)

(d) Write Python code for a simple perceptron decision function **buyCocktail(x)**, which decides on whether to buy an expensive cocktail or not based on the following inputs where x= [x1, x2, x3] .

x1 = 1 means you have plenty of cash, x1 = 0 means you do not. x2 = 1 means you want to impress a new girlfriend, 0 means you don't. x3 = 1 means it's a busy exciting night on the town, 0 means it I not.

Choose the weights and threshold (or bias) so that if you want to impress someone, you will buy cocktails no matter what. However, if you are not out to impress but it's an exciting night and you have plenty of cash, you will buy a cocktail. Otherwise you will not buy a cocktail.

(5 marks)

(e) Suppose you were to train a perceptron to behave as described in part(d) instead of handcrafting the weights. Write down an appropriate list of input training vectors and the corresponding list of desired outputs which could be used to train such a perceptron.

(5 marks)

4 (a) What is a perceptron? Draw a diagram for a single two-input perceptron. Explain what is meant by linear separability. How many categories can it classify inputs into? Mention any issues involved in using a perceptron to compute logical AND, OR and XOR.

(7 marks)

(b) Illustrate the single perceptron training algorithm for an OR gate by computing the weight adjustments for the first three epochs in the table below.

(5 marks)

(c) Distinguish between a sigmoid neuron and a perceptron, and outline the difference in the training formula for Δw_i.

Why might it be preferable to use sigmoid neurons in a neural network rather than perceptrons?

(8 marks)

(d) Provide a Python code segment which implements the perceptron training algorithm.

(5 marks)

Perceptron Training for logical OR

Threshold $\theta = 0.2$, learning rate $\alpha = 0.1$.

Epoch	Inputs		Desired output	Weights		Weighted Sum	Actual Output	Error	Weight adjustments	
	x_1	x_2	Y_d	w_1	w_2	z	a	e	Δw_1	Δw_2
1	0	0	0	0.2	-0.1					
	0	1	1							
	1	0	1							
	1	1	1							
2	0	0	0							
	0	1	1							
	1	0	1							
	1	1	1							
3	0	0	0							

Table 4(b)

5 (a) Distinguish between classification and regression in machine learning.

Briefly describe the MNIST dataset and outline how it could be used to train a multilayer classification neural network.

(5 marks)

(b) Suggest a network architecture appropriate for MNIST and explain why the cross-entropy cost function might be preferable to the mean-squared-error when the sigmoid activation function is used.

(5 marks)

(c) Explain the problem of overtraining a neural network and how it could be alleviated. How does MNIST address this problem?

(5 marks)

(d) Briefly explain what gradient descent means and how it relates to the cost or loss function. Explain what each of the following symbols represents and the relationship between then in gradient descent.

$$\Delta \mathrm{w}^{l}{}_{\mathrm{ij}} \quad \text{and} \quad \frac{\partial c}{\partial w_{ij}^{l}}\,?$$

(5 marks)

(e) At the core of the backpropagation algorithm are the matrix formulae:

$$\frac{\partial C}{\partial w_{jk}^{l}} = a_k^{l-1}\delta_j^l$$

$$\delta^L = (a^L - y) \odot \sigma'(z^L) \qquad \delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l)$$

Briefly explain what each term means and which one in particular relates to backpropagation.

(5 marks)