### *Course: CSC14003 – Introduction to Artificial Intelligence*

### *Class 21CLC – Term II/2022-2023*

Homework 04

***Submission Notices:***

* *Conduct your homework by filling answers into the placeholders in this file (in Microsoft Word format).*

*Questions are shown in black color, instructions/hints are shown in italics and blue color, and your content should use any color that is different from those.*

* *After completing your homework, prepare the file for submission by exporting the Word file (filled with answers) to a PDF file, whose filename follows the following format,*

*<StudentID-1>\_<StudentID-2>\_HW01.pdf (Student IDs are sorted in ascending order)*

*E.g.,* ***2112001\_2112002\_HW02.pdf***

*and then submit the file to Moodle directly WITHOUT any kinds of compression (.zip, .rar, .tar, etc.).*

* *Note that you will get zero credit for any careless mistake, including, but not limited to, the following things.*
  1. *Wrong file/filename format, e.g., not a pdf file, use “-” instead of “\_” for separators, etc.*
  2. *Disorder format of problems and answers*
  3. *Conducted not in English*
  4. *Cheating, i.e., copying other students’ works or letting other students copy your work.*

**Problem 1. (2pts)** Identify each of the following activation functions.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **2-D representation** | **#** | **2-D representation** |
| a | Các hàm kích hoạt (activation function) trong neural network ... | b | Các hàm kích hoạt (activation function) trong neural network ... |
| c | Các hàm kích hoạt (activation function) trong neural network ... | d |  |
| e |  | f |  |
| g |  | h | Các hàm kích hoạt (activation function) trong neural network ... |

*Please fill your answer in the table below*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | a | b | c | d |
| **Function name** | Sigmoid | Tanh | ReLU | Leaky ReLU |
| **#** | e | f | g | h |
| **Function name** | Binary step function | Linear | Gaussian function | Parametric ReLu |

**Problem 2. (1pt)** Present two objective metrics that can be used to evaluate the attributes for a node on the decision tree. For each metric, you need to present the formula, identify its domain (i.e., range of values), and explain for every term in the formula.

*Please fill your answer in the table below*

|  |  |  |
| --- | --- | --- |
| **Metric name** | **Formula** | **Explanation** |
| Gini Index |  | * Pi : is the probability of the observation from splitting result for a distinct class * n : is the number of the class |
| Information Gain | Step 1 : Calculate Entropy of target attribute  Step 2 : Calculate Average Entropy of attribute Alternate  Step 3 : Calculate Information Gain | In step 1 :   * H(S) : Entropy of target attribute * vk is a class in S * P(vk) :is the proportion of the number elements in class vk to the number of elements in S   In step 2 :   * AEx : Average Entropy of attribute x * vi is a class in Si * P(vi) : is the proportion of the number elements in class vi to the number of elements in Si * H(Si) : Entropy of attribute x   In step 3 :   * IG ( x,S) : Information Gain of attribute x * H(S) : Entropy of target attribute * AEx : Average Entropy of attribute x |

**Problem 3. (2pts)** You are given the following tables, which represent the outcomes of some functions. The functions take two values and and output the outcomes of the operations. Please identify **at least two models** for each of the functions that are perfectly represent the functions for some choice of parameters. Justify your answer. Note: there are no constraints on the architecture (e.g, the number of neurons, activation function, or the best splitting criterion), and the depth of decision tree is 0-index.

1. (1pt)

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 0 | 0 | **0** |
| 0 | 1 | **1** |
| 1 | 0 | **1** |
| 1 | 1 | **0** |

* A neural network with no hidden layer
* A neural network with a single hidden layer
* A decision tree of depth one
* A decision tree of depth two

**Explanation:** In this case, since the function f(x,y) is non-linear, we need a model that can perform non-linear transformations of the input variables. Therefore, a neural network with at least one hidden layer would be the best choice.

Among the options given, the neural network with a single hidden layer would be the most suitable model for implementing f(x,y) = x ⨁ y. This model can perform non-linear transformations of the input variables and is able to capture the complexity of the function.

A decision tree of depth one would not be able to capture the full complexity of the function, as it is limited to a single threshold value. A decision tree of depth two could potentially capture the function, but a neural network with a single hidden layer would be a more efficient and effective model for this task.

1. (1pt)

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 0 | 0 | **1** |
| 0 | 1 | **0** |
| 1 | 0 | **0** |
| 1 | 1 | **0** |

* A neural network with no hidden layer
* A neural network with a single hidden layer
* A decision tree of depth one
* A decision tree of depth two

**Explanation**: In this case, the function f(x,y) is also non-linear, but it is a negation of the logical OR operation. One way to implement this function is to use a single hidden layer neural network with the logical OR operation and the negation operation as activation functions.

Among the options given, a neural network with a single hidden layer would be the most suitable model for implementing f(x,y) = ¬(x ∨ y). This model can perform non-linear transformations of the input variables and can implement the logical OR and negation operations as activation functions in the hidden layer.

A decision tree of depth one or two would not be able to capture the full complexity of the function, as it requires both the logical OR and negation operations to be performed on the inputs. A neural network with no hidden layer would also not be able to capture the complexity of the function, as it is not capable of performing any non-linear transformations. Therefore, a neural network with a single hidden layer would be the best choice for implementing f(x,y) = ¬(x ∨ y).

**Problem 4. (2pts)** Consider the following training dataset, in which **Transportation** is the target attribute. Show calculations to choose an attribute for the **root node** of the ID3 decision tree

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Gender** | **Car Ownership** | **Travel Cost** | **Income Level** | **Transportation** |
| Male | 0 | Cheap | Low | Bus |
| Male | 1 | Cheap | Medium | Bus |
| Female | 1 | Cheap | Medium | Train |
| Female | 0 | Cheap | Low | Bus |
| Male | 1 | Cheap | Medium | Bus |
| Male | 0 | Standard | Medium | Train |
| Female | 1 | Standard | Medium | Train |
| Female | 1 | Expensive | High | Car |
| Male | 2 | Expensive | Medium | Car |
| Female | 2 | Expensive | High | Car |

*Please fill your answer in the table below*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Counts | | | Metric values | | |
| Attribute values | Bus | Car | Train | H | AE | IG |
| **Whole** | 10 | 4 | 3 | 3 | 1.571 |  |  |
| Gender (0.5pt) | Female | 1 | 2 | 2 | 1.371 | 1.447 | 0.12 |
| Male | 3 | 1 | 1 | 1.522 |
| Car Ownership (0.5pt) | 0 | 2 | 0 | 1 | 0.918 | 1.0364 | 0.534 |
| 1 | 2 | 1 | 2 | 1.522 |
| 2 | 0 | 2 | 0 | 0 |
| Travel Cost  (0.5pt) | Cheap | 4 | 0 | 1 | 0.722 | 0.361 | 1.21 |
| Expensive | 0 | 3 | 0 | 0 |
| Standard | 0 | 0 | 2 | 0 |
| Income Level  (0.5pt) | Low | 2 | 0 | 0 | 0 | 0.8754 | 0.695 |
| Medium | 2 | 1 | 3 | 1.459 |
| High | 0 | 2 | 0 | 0 |

**Problem 5. (3pts)** Consider the following neuron network, which includes 3 input neurons, 2 hidden neurons and 1 output neurons.

Diagram

Description automatically generated

Initial input, weight and bias values are

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x1 | x2 | x3 | w14 | w15 | w24 | w25 | w34 | w35 | w46 | w56 | θ4 | θ5 | θ6 |
| 1 | 0 | 1 | 0.2 | –0.3 | 0.4 | 0.1 | –0.5 | 0.2 | –0.3 | –0.2 | –0.4 | 0.2 | 0.1 |

The expected output value is 1. The learning rate is 0.9.

Knowing that the actual output at some neuron *j* is calculated as follows.

where *n* is the number of inputs of neuron *j*, is the corresponding link from a neuron *i* in the previous layer to neuron *j*, and is the bias at neuron *j*.

Present all calculations required to perform the backpropagation once (i.e., one forward pass and one backward pass) on the given neural network in the following cases

1. Ignore all biases *(precision to 3 decimal places).*

*(0.25pt) Ignore all biases – Forward*

|  |  |  |  |
| --- | --- | --- | --- |
| Neuron | 4 | 5 | 6 |
| Output | Sigmoid(0,2\*1 + 0,4\*0 + (-0,5)\*1) = 0,426 | Sigmoid((-0,,3)\*1+ 0,1\*0 + 0,2\*1) = 0,475 | Sigmoid(0,426\*(-0,3) + 0,475\*(-0,2)) = 0,445 |

*(1pt) Ignore all biases – Backward*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Weight | w46 | w56 | w14 | w15 | w24 | w25 | w34 | w35 |
| Value | -0,247 | -0,142 | 0,191 | -0,306 | 0,4 | 0,1 | -0,509 | 0,194 |

1. Consider all biases such that each bias is treated as a neuron and thus it will be also updated *(precision to 3 decimal places).*

*(0.25pt) Consider all biases – Forward*

|  |  |  |  |
| --- | --- | --- | --- |
| Neuron | 4 | 5 | 6 |
| Output | Sigmoid(0,2\*1 + 0,4\*0 + (-0,5)\*1+(-0,4)) = 0,332 | Sigmoid((-0,,3)\*1+ 0,1\*0 + 0,2\*1+0,2) = 0,525 | Sigmoid(0,332\*(-0,3) + 0,525\*(-0,2) +0,1) = 0,474 |

*(1.5pt) Consider all biases – Backward*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Weight | w46 | w56 | w14 | w15 | w24 | w25 | w34 | w35 | θ4 | θ5 | θ6 |
| Value | -0,261 | -0,138 | 0,192 | -0,306 | 0,4 | 0,1 | -0,508 | 0,194 | -0,408 | 0,194 | 0,218 |