University of Glasgow | School of Computing Science

Mini Project

Course Name	Programming Methodology (PM)			
Coursework Number	1			
Deadline	Time:	23:59	Date:	20 Nov 2022
% Contribution to final course mark	35%			
Solo or Group ✓	Solo		Group	✓
Anticipated Hours	15 per group member			
Submission Instructions	1. Submit via SIT_Xsite a Zip or compressed tar file containing your source code directory (UoG-PM.tgz or UoG-PM.zip), one PMReportSessionxGroupxx.pdf file plus group presentation video/YouTube video 2. Declaration Form			
Please Note: This Coursework cannot be Re-Done				

Code of Assessment Rules for Coursework Submission

Deadlines for the submission of coursework which is to be formally assessed will be published in course documentation, and work which is submitted later than the deadline will be subject to penalty as set out below.

The primary grade and marks awarded for coursework which is submitted after the published deadline will be calculated as follows:

(i) in respect of work submitted not more than four working days after the deadline

- the work will be assessed in the usual way;
- the primary grade and mark so determined will then be reduced by 15% for each working day (or part of a working day) the work was submitted late.

(ii) work submitted more than four working days after the deadline will be awarded Grade F.

Penalties for late submission of coursework will not be imposed if good cause is established for the late submission. You should submit documents supporting good cause to Admin-In-Charge

Penalty for non-adherence to Submission Instructions is 2 bands
You must complete an “Own Work” form

Programming Methodology(UoG-PM) 2022-23

Mini Project: Cognitive learning for Kids with IoT AI/ ML Tic Tac Toe game

Artificial Intelligence (AI) /Machine Learning (ML) in C programming

Introduction

The goal of this Mini-Project exercise is to familiarize yourselves with the design, implementation and performance testing of C programming on the Internet of Things (IoT) edge computing Artificial Intelligence (AI)/ Machine learning (ML)landscape. Given the problem statement, you will be required to go through the whole cycle of problem definition, problem analysis, and pseudocode design. You will implement with testing and demonstrate your AI/ML enabled Tic Tac Toe game algorithm in C programming.

Assessed task

Assuming you are the AI/ML engineer for a company and you are tasked by a particular nursery childcare organization to help to design a Tic Tac Toe (https://en.wikipedia.org/wiki/Tic-tac-toe) game on the memory and power limited IoT tablet for their kids to develop the motor skill, social skill and initial stage of left brain development of the kids (https://www.gamesver.com/benefits-and-advantages-of-playing-tic-tac-toe-importance/).

To fulfill above objectives, the main task of the team is to develop the 3x3 Tic Tac Toe game in C programming language with the following features

1. two player mode where two kids can play with each other (facilitate social skill development of kids and their motor skills)
2. A one-player mode where the sole kid will play with AI enabled computer (to develop initial stage of analytics thinking skill)

The design of the IoT game should come with user-friendly graphical user interface (GUI) such as https://playtictactoe.org

As such, the main task is multiple fold

1) A user-friendly interactive GUI that minimally display player 1 (having symbol X) and player 2 (having symbol O), and the winner declared. You can use any open library GUI tools such as http://www.gnuplot.info if desired.
2) For the two-player mode, each kid will take turn to interact to play their turn. Winner will be declared if any of the kid has 3 X or 3 O across any row, column, or diagonal.
   https://www.geeksforgeeks.org/implementation-of-tic-tac-toe-game/

http://www.cprogrammingnotes.com/question/tic-tac-toe-game.html

3) For one player mode, the AI engine should be kicked in to emulate computer player. As such, the AI engine should be based on popular recursive Minimax algorithm (https://en.wikipedia.org/wiki/Minimax) that is a depth first search (DFS) algorithm that will exhaust all possibilities to win the game without any training phase.(https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-3-tic-tac-toe-ai-finding-optimal-move/amp/)
4) However, a tic tac toe game that has AI wins every game will cause frustration, boredom, and demotivation on the kids as there is no possibility for her/him to win the game. This is not a good cognitive learning for kids. Furthermore, Minimax algorithm is a memory inefficient algorithm since it is an exhaustive search. Your team should also create some easiness through difficulty level by either one of the following
    a. Improve on the Minimax (perfect) algorithm that is memory efficient and not every time AI wins the game (imperfect)

    OR add the imperfectness with

    b. ML algorithms that need training data to train. Some of these algorithms are linear regression, navie bayes, neural network and reinforcement learning where accuracy is not perfect unless perfect training data. This hopefully give chances for kids to win the game. Dataset is given in the XSite mini project folder that has 958 legal move(row) that contributes to win (positive) or lose(negative)
        i. Calculate both training and test accuracy for the chosen ML algorithm with the dataset split into training and testing data ratio of 80:20.
        ii. Determine and plot the confusion matrix for the training and testing accuracy that reflects imperfection in learning.(https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62)
    c. Calculate the number of times the computer wins as a gauge of difficulty level.

Looking at the dataset and problem statement above, you should first perform problem definition, problem analysis followed by the pseudocode before proceeding to the C programming coding. Your C program should be of originality with modularity and functionality.

General step for ML algorithm (if adopt)

Below serves as a guideline but not limited to(Just a reference) for the ML algorithm for task 4(b) if chosen by your team.

1) Input
    a) Write a function to open the dataset file (Hint: File pointer) and extract the respective attributes/feature & outcome of each volunteer data $(x_{1,1}, x_{1,2}, x_{1,3},, x_{1,4}, x_{1,5}, x_{1,6}, x_{1,7}, x_{1,8}, x_{1,9}, y_{1,1}) \cdots$ $(x_{958,1}, x_{958,2}, x_{958,3},, x_{958,4}, x_{958,5}, x_{958,6}, x_{958,7}, x_{958,8}, x_{958,9}, y_{958,1})$ into multidimensional data array set. $x_{m,n}$ is the feature where m is the data set row from $1 \cdots 958$ and n is the square box number of the tic tac toe.

       $n = 1$: top-left-square: {x,o,b}
       $n = 2$: top-middle-square: {x,o,b}
       $n = 3$: top-right-square: {x,o,b}
       $n = 4$: middle-left-square: {x,o,b}
       $n = 5$: middle-middle-square: {x,o,b}
       $n = 6$: middle-right-square: {x,o,b}
       $n = 7$: bottom-left-square: {x,o,b}
       $n = 8$: bottom-middle-square: {x,o,b}
       $n = 9$: bottom-right-square: {x,o,b}

       and from $y_{m,1}$ is the classification output for each data row set, either positive (win) or negative (lose). x-player 1, o-player 2, b=blank square.
    b) Split into training set and testing set such as first 80% of the data is training set while the last 20% data as testing set
2) Write various **<u>modular functions</u>** to perform ML classifier training such as

    a) For Navie Bayes classifier (https://github.com/m-aminiz/NaiveBayes

Artificial Intelligence (AI) such as Naive Bayes classifier has been used in many applications such as Natural Language Processing (NLP) chatbot and fake new detection etc. It is a class of Bayesian estimator assuming features (each Tic Tac Toe move) are independent from each other. It is based on conditional probability and bayes theorem

- Conditional Probability:

$$\boxed{P(Y|X) = \frac{P(X,Y)}{P(X)}} \qquad \boxed{P(X|Y) = \frac{P(X,Y)}{P(Y)}}$$

- Bayes Theorem:

Conditional probability

Posterior probability

Prior probability

Marginal probability (the evidence)

$$\boxed{\begin{aligned} P(Y|X) &= \frac{P(X|Y)P(Y)}{P(X)} \\ &= \frac{P(X|Y)P(Y)}{\sum_{i=1}^{N} P(X|y_i)P(y_i)} \end{aligned}}$$

- Compute posterior probability $P(Y|X_1, X_2, X_3 \cdots X_d)$ using Bayes Theorem :

$$\boxed{\begin{aligned} P(Y|X_1, X_2, X_3 \cdots X_d) &= \frac{P(X_1, X_2, X_3 \cdots X_d|Y)P(Y)}{P(X_1, X_2, X_3 \cdots X_d)} \\ &\equiv P(X_1, X_2, X_3 \cdots X_d|Y)P(Y) \end{aligned}}$$

E.g.

ID	Home Owner	Marital Status	Defaulted Borrower
1	Yes	Single	No
2	No	Married	No
3	No	Single	No
4	Yes	Married	No
5	No	Divorced	Yes
6	No	Married	No
7	Yes	Divorced	No
8	No	Single	Yes
9	No	Married	No
10	No	Single	Yes

Determine whether the borrower will default payment for

$$\boldsymbol{X}(HomeOwner = No,\ Marital\ Status = Married)$$

$$P(\boldsymbol{X}\,|Yes) = P(HomeOwner = No|Yes) \times P(Maritial\ Status = Married|Yes)$$

$$P(\boldsymbol{X}\,|No) = P(HomeOwner = No|No) \times P(Maritial\ Status = Married|No)$$

$$\boxed{\begin{aligned} P(HomeOwner = No|Yes) \qquad &= \frac{3}{3} = 1 \\ P(Maritial\ Status = Married|Yes) &= \frac{0}{3} = 0 \\ P(HomeOwner = No|No) \qquad &= \frac{4}{7} \\ P(Maritial\ Status = Married|No) &= \frac{4}{7} \end{aligned}}$$

$$P(Yes|\boldsymbol{X}\,) = \frac{P(\boldsymbol{X}\,|Yes)P(Yes)}{P(\boldsymbol{X})} \propto P(\boldsymbol{X}\,|Yes)P(Yes)$$

$$\propto [\boldsymbol{P(HomeOwner = No|Yes)} \times \boldsymbol{P(Maritial\ Status = Married|Yes)}]P(Yes\boldsymbol{)}$$

$$= 1 \times 0 \times \frac{3}{10} = 0$$

since $P(Yes)$=fraction of instances of class Yes

$$P(No|\boldsymbol{X}\,) = \frac{P(X\,|No)P(No)}{P(X)} \propto P(\boldsymbol{X}\,|No)P(No)$$

$$\propto [P(HomeOwner = No|No) \times P(Maritial\ Status = Married|No)]P(No)$$

$= \frac{4}{7} \times \frac{4}{7} \times \frac{7}{10} = 0.8$
since $\mathrm{P(No)}$=fraction of instances of class No

Since $P(No|\boldsymbol{X}) > P(Yes|\boldsymbol{X})$, therefore class =No, borrower will not default payment

As such, amended the tic tac toe algorithm according to the predicted probability of every opponent move for all training dataset. This is followed by accuracy testing on the testing dataset.

b) For linear regression e.g.

- https://amit9oct.github.io/2020-08-12-LearnTicTacToe/

- https://medium.com/analytics-vidhya/building-a-tictactoe-bot-with-regression-ba79f2ae74bb

The objective is to build a linear equation such as a cost function to train a set of weights optimally over 958 rows of data to achieve minimum (smallest) mean square error (MMSE) that matches the data features and classification output

The linear regression is modelled as

$$y_{m,1} = w_{m,1}x_{m,1} + w_{m,2}x_{m,2} + w_{m,3}x_{m,3} + w_{m,4}x_{m,4} + w_{m,5}x_{m,5} + w_{m,6}x_{m,6} + w_{m,7}x_{m,7} + w_{m,8}x_{m,8} + w_{m,9}x_{m,9}$$

and the predicted regression is

$$\hat{y}_{m,1} = \widehat{w}_{m,1}x_{m,1} + \widehat{w}_{m,2}x_{m,2} + \widehat{w}_{m,3}x_{m,3} + \widehat{w}_{m,4}x_{m,4} + \widehat{w}_{m,5}x_{m,5} + \widehat{w}_{m,6}x_{m,6} + \widehat{w}_{m,7}x_{m,7} + \widehat{w}_{m,8}x_{m,8} + \widehat{w}_{m,9}x_{m,9}$$

where $\widehat{w}_{m,1} \cdots \widehat{w}_{m,9}$ is the predicted weight through is

$$MMSE = \frac{1}{M}\sum_{m=1}^{M}\left(y_{m,1} - \hat{y}_{m,1}\right)^2$$

Where M=958. From these trained weights, amended the tic tac toe algorithm according to the predicted weight for every opponent move. Repeat accuracy testing for testing dataset.

c) Neural network which consists of many linear regressions enabled neurons or any other ML algorithms (https://www.kaggle.com/code/dhanushkishore/a-self-learning-tic-tac-toe-program)

3) Calculate probability of error for each training data and testing data,i.e to check the predicted outcome $\hat{y}_{m,1}$ and the real outcome $y_{m,1}$ is the same. If differs, error in prediction occurs and set $error_i = 1 \;\; else \; 0$ if prediction is correct. Count the number of error for N dataset.

$$\boxed{probability\ of\ error = \frac{1}{N}\sum_{i=1}^{N} error_i}$$

.

For Confusion matrix, determine the four classes namely True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN)

True Positive : No of times predict correctly that classification is positive(wins)

True Negative: No of times predict correctly that classification is negative(lose)

False Positive: No of times predict wrongly that classification is positive(wins) but actual is negative (lose)

False Negative: No of times predict wrongly that classification is negative(lose) but actual is positive (wins)

https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

General Instruction

The ML dataset will be uploaded and be available in both SIT-Xsite Content and Dropbox Folder under the course assessment folder. The programming language is to be used is the same standard C programming language where Microsoft Visual Studio Code (VSC) is the baseline editor and GNU Compiler Collection (GCC) is the standard compiler. Your C Program is free to call/activate any plotting tool such as GNU plot at http://www.gnuplot.info Pls refer to the website on how to install the gnuplot for Windows and Mac (For Mac, it is better to use homebrew for ease of installation).

What to handin

According to your Group number, use

- SIT-Xsite Dropbox to submit a single zip or compressed tar file with the contents of the UoG-PM plus a separate PMReportSessionxGroupxx.pdf file and the scanned version of the signed declaration forms of all members in the group

To aid in testing and assessing of your code, please make sure that:

1) Your submission file is named UoG-PM.tgz or UoG-PM.zip.
2) When uncompressed, your files will be in a folder named UoG-PM_SessionxGroupxx. The folder should contain all the following
    a) all the C source codes (.c or even .h codes) and respective executable file (. exe file) that are necessary to take in the dataset and generate all the plots and the printed results. Please include all the printed results.
    b) PMReportSessionxGroupxx.pdf where x and xx indicate your lab session group and group number respectively according to the grouping list.
    c) A group power point presentation video of about 10 mins where all group members **(turn on camera to see your face)** must give their speech on their respective portion of work if physical presentation is not feasible at week 13.
    d) A YouTube video (about 5 mins) showing the demo and design capabilities of your team's Tic Tac Toe program **(no need to turn on camera, just articulate the demo and design)**
3) Your PMReportSessionxGroupxx.pdf file should outline and contain
    a) The name of the group member and the **individual contribution**.
    b) your design solution in terms of
        - problem definition,

- problem analysis,
- pseudocode.

c) all the source codes
d) all the plots and results
e) any interesting aspects of your solution (e.g., assumptions you've made, optimisations that you thought of, etc.),

4) Your submission will be tested for plagiarism with a specialised off-the-shelf tool. Plagiarism cases will be dealt on a case-by-case basis but suffice to say there will be little tolerance.

How this exercise will be marked

Following timely submission, the exercise will be given a numerical mark between 0 (no submission) and 100 (perfect in every way). The numerical marks will then be converted to a grade. The marking scheme as a group is as follows:

At scope level, the marking weightage is as follows:

1. 50% : GUI + manual two players mode (Task 1 + Task 2)
2. 30% : AI MiniMax algorithm covering one player mode (Task 3)
3. 20% : Imperfect MiniMax algorithm (Task 4a) or ML algorithm (Task 4b)

Overall, marking criteria will be as follows:

- 30 marks for the quality and structure of the code; make sure that you use appropriate names for variables/function and that your source code is properly structured.
- 20 marks for comments and documentation in the code and your PMReportGroupxx.pdf file; make sure that you comment and document in your source files, at the very least, the basic steps taken to comment on the function and various variables used. Do not make an essay of your code; use your pdf file to discuss further details.
- 30 marks for a solution that produces the correct results; partial marks will be awarded if your solution is on the right track but produces incorrect results due to minor errors. Marks will be awarded based on originality of the codes. It is of utmost importance to demonstrate the team competency in originality in producing the various program modules and shall be generated by the team.
- 20 marks for any optimization you have come up with, e.g., to increase accuracy, efficient coding, reusable functions, modularity, faster time to process and play, especially as a way to show the efficiency of implementation since it will be implemented in power and memory limited computing platform.

The final marking scheme for individual will be as follows:

Each group member grade = (weighted peer review score by group member with his/her presentation during group video presentation video) *Group score