1MAI1 - MSc Artificial Intelligence

CT 5132 Programming and Tools for AI

Assignment 03

Submitted To	Dr James McDermott
Submitted By	Raksha Padamnoor Rao, Suruchi Devendra Gupta
Student ID	19230887, 19233027

Abstraction and Reasoning Corpus (ARC)

Abstraction and Reasoning Corpus(ARC) has a set of training and testing samples in form of JSON file. Each JSON file has a task involving a grid of colours (encoded in form of integers 0-9). The aim of ARC is to train the ML algorithm such that it identifies the pattern from the training set and accordingly produces output for the testing set. ARC as defined by François Chollet is "*a general artificial intelligence benchmark, as a program synthesis benchmark, or as a psychometric intelligence test*", as the tasks may seems trivial for the human mind but it is difficult to train an algorithm to identify the pattern in the task and to produce the output grid accordingly (sample task given below in Fig 1).

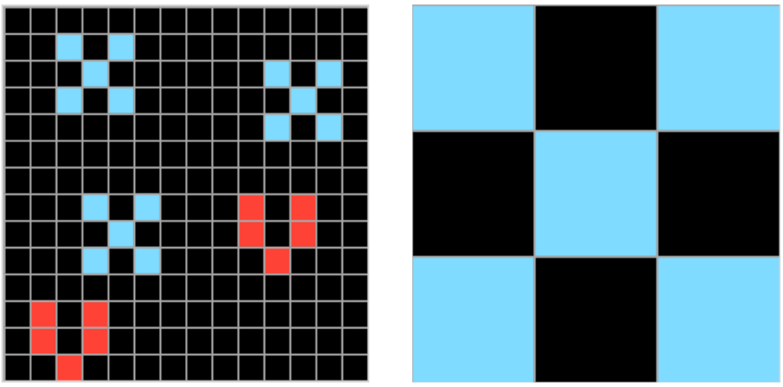

Fig 1. Sample Input and Output

Hence, a sample of four tasks was selected and hand coded in python. A python file has solution for one task and the details of the tasks are as given below:

Git Repository	https://github.com/RakshaPRao/ARC
Tasks Attempted	29c11459
	746b3537
	a699fb00
	aabf363d

Task Description and Implementation Details

1. Common Functionalities

As there are some functionalities which are common for all the tasks (Ex: reading the input file and printing the output grid on the console), the ioOps module contains all the methods required for these common processing procedures. The details of each method in the module are:

a) **get_file_path:** The method takes the command line arguments as input and returns the path to the input file which has the JSON details for the task. It returns a message if there are no command line arguments.

b) **read_file:** This method takes the file path as an input, reads the file and gives an equivalent numpy array for the input JSON.

c) **print_grid:** The method takes an numpy array as an input and prints it in a tabular format. This method is used for printing the output grid on the console.

2. Solution for task 29c11459

The task takes a black grid as an input with two colours in the extreme corners for some lines. Output grid has a line of the colour reaching out from left-most and right-most cells for the non-black rows and the middle cell as grey. A sample of input and output for this file is depicted in the Fig 1.

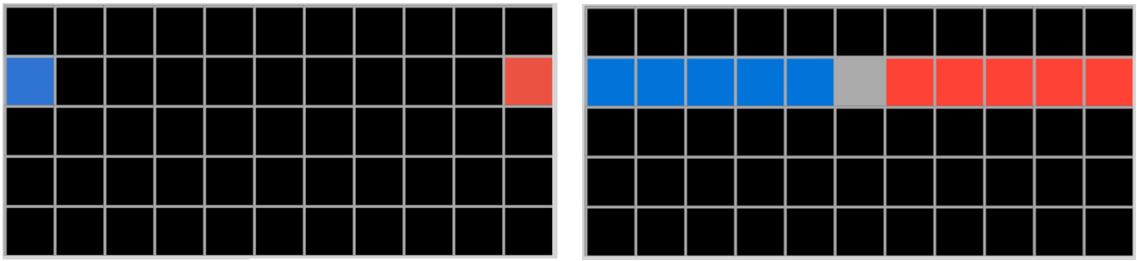

Fig 2. Input and Output for task 29c11459

The algorithm needs to find the rows which do not start with black colour. For these rows, it needs to store the colour in the beginning of the row and at the end the row. It then colours the left-most colour in the left to right direction of the row till the middle of row and colours the right-most colour from the middle till end of the row. The mid-point cell of the row (this task has only odd number of columns in the input output grid) is filled with grey colour in the end.

The solve method does the steps stated above and can successfully solve all the training and test cases in the JSON input file.

3. Solution for task 746b3537

The task aims to find the unique colours from the input grid in the manner that they appear in the input grid, i.e., row-wise or column-wise (sample in Fig. 3).

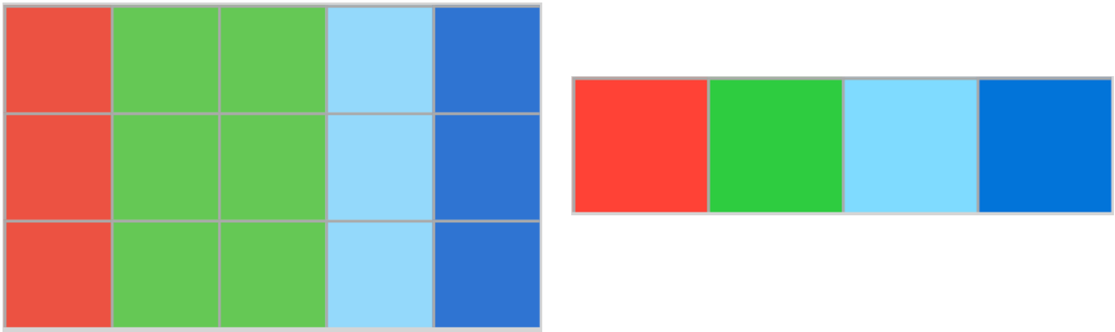

Fig 3. Input and Output for task 746b3537

The algorithm takes the grid as input as extracts a row and a column from it. It then compresses the row and the column array to find the unique colours in the array. The array with length greater than one is the output array. This output array is returned in the manner corresponding to as it was extracted from the input (row-wise or column-wise). All the given training and test cases are computed correctly in the program.

4. Solution for task a699fb00

The task a699fb00 takes a grid as input with randomly placed blue cells. The Output is a grid with cell filled with red colour where the adjacent cells are blue. The sample input and output for the task is as given in Fig 3.

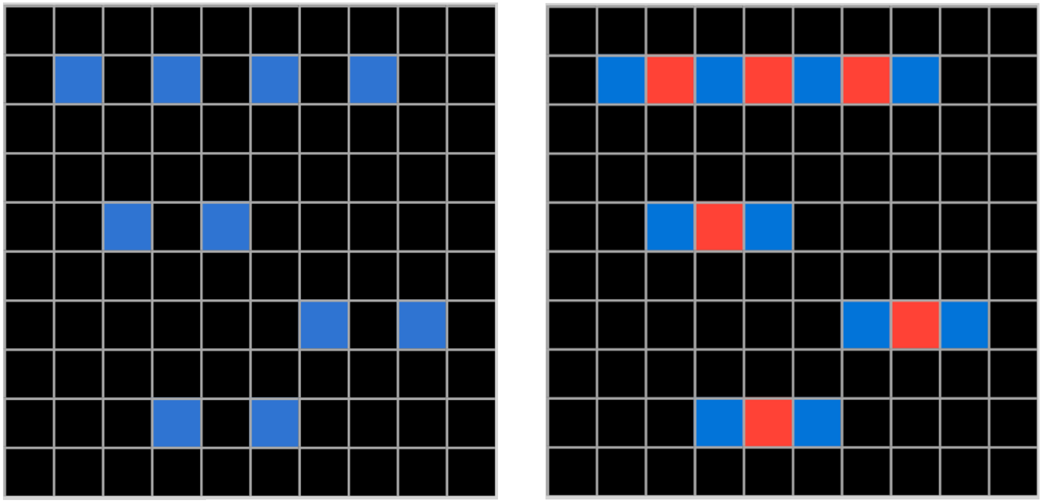

Fig 4. Input and Output for task a699fb00

The solve method in the module goes through each row and checks for blue coloured cells in the grid with one black cell in between. When this condition is satisfied, the black colour cell is changed to red colour. The Output is returned in form of an numpy array. All the training and test cases are successfully solved by the method.

5. Solution for task aabf363d

The input to this task is a grid with a pattern in it and a colour in the lower-most left cell. The aim of the task is to fill the pattern in the input grid with the colour in the lower-most left cell of the grid and once done, replace that cell with black colour (Fig 3).

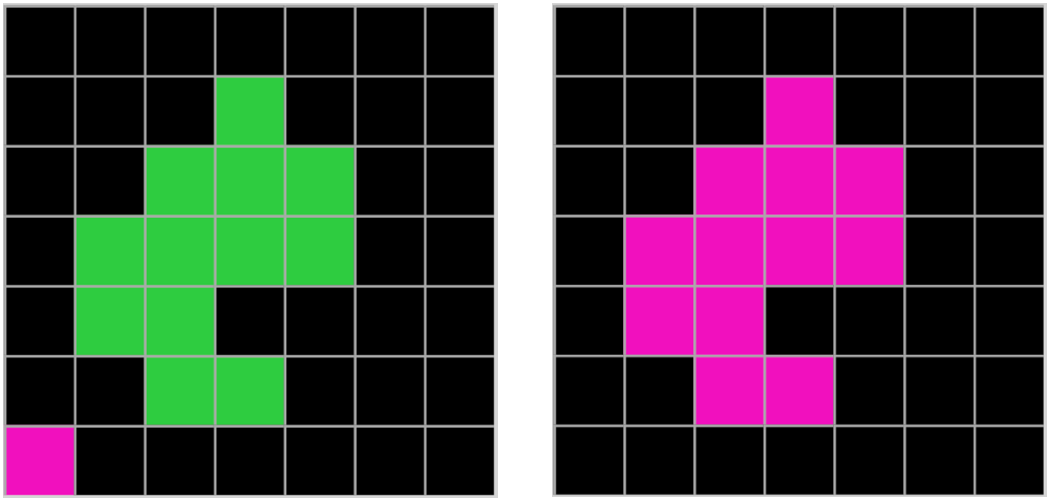

Fig 5. Input and Output for task a699fb00

The solve method fills all the non-black cells with the colour in the lower-most left cell. Then the cell which indicates the colour is changed to black. The operation is carried in bulk using the capabilities of Numpy package. The solve method completes all the training and test cases correctly.

Insights of Python Functionality in the Implementation

The data handling was simple and implementation of the tasks was easier in python due to the clean and well written libraries in python. Some of the libraries and functionalities used in this assignment are:

- **JSON:** This module in python makes JSON data handling easier, it provides functionalities to convert in-memory Python objects to a serialized representation known as JavaScript Object Notation (JSON) and vice-a-versa. The ioOps module uses these functionalities to read the input file as a JSON Object and then converts it to a numpy array for further processing.

- **Itertools:** This module implements a number of iterator building blocks for users to construct specialized tools succinctly. The chain method of this module is used to capture train and test inputs in a sequence after reading the data from the input file.

- **Sys:** This module provides information about constants and methods maintained by the python interpreter. The command line arguments that are passed to the python script can be accessed using argv attribute of this module.

- **Numpy:** This module has a variety of extremely powerful methods for homogeneous and multi-dimensional array objects and its equivalent operations. Flatten method is used to convert an n-dimensional array to one dimension and process the resulting array. Similarly, asarray method converts a list to a numpy array.

The tasks selected had some similar features like, input should be given in form of a JSON file which is passed as command line argument to the program. Each program takes 2-dimensional array as input which needs to be processed individually. After processing the input the output has to be displayed on the terminal in a tabular format.

The chosen tasks are similar even in the manner that the shape of the input and output is same, however for the task 746b3537, the shape of the output is different than that of the input data.

Contributions

The contributions of each person in the assignment are as given below:

Person 1: Raksha Padamnoor Rao

- The common functionalities methods to parse command line arguments, read file and print the output.
- Solution for the tasks a699fb00 and 29c11459.

Person 2: Suruchi Devendra Gupta

- Change of the solve method structure to handle data input for all tasks.
- Solution for the tasks 746b3537 and aabf363d.

Combined Contributions

- Design and architecture decisions of the solutions.
- Report documentation.

Bibliography

- Chollet, F. (2019, Nov 27). Retrieved from https://arxiv.org/pdf/1911.01547.pdf
- w3resource. (2019, November 30). *Python JSON Module tutorial*. Retrieved from w3resource: https://www.w3resource.com/JSON/python-json-module-tutorial.php