

Configuration Manual

MSc Research Project

Programme Name

Forename Surname

Student ID: XXX

School of Computing

National College of Ireland

Supervisor: XXX



**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name:** | ……. …………………………………………………………………………………………………………… | | |
| **Student ID:** | ……………………………………………………………………………………………………………..…… | | |
| **Programme:** | ……………………………………………………………… | **Year:** | ………………………….. |
| **Module:** | ………………………………………………………………………………………………………….……… | | |
| **Lecturer:** | ………………………………………………………………………………………………………….……… | | |
| **Submission Due Date:** | ………………………………………………………………………………………………………….……… | | |
| **Project Title:** | ………………………………………………………………………………………………………….……… | | |
| **Word Count:** | ……………………………………… **Page Count:** ………………………………….…….……… | | |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

|  |  |
| --- | --- |
| **Signature:** | ……………………………………………………………………………………………………………… |
| **Date:** | ……………………………………………………………………………………………………………… |

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

|  |  |
| --- | --- |
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

|  |  |
| --- | --- |
| **Office Use Only** | |
| Signature: |  |
| Date: |  |
| Penalty Applied (if applicable): |  |

Configuration Manual

Johns Thomas

Student ID: x22203389

# Introduction

This manual provides detailed explanation and instructions for setting up, configuring, and running Q learning and Deep Q-learning agents to support serverless function configuration. It is intended to ensure the reproducibility of the work and for researchers and developers interested in similar applications. The manual covers the installation of necessary software, configuration of learning parameters, execution of the agents in AWS lambda, and evaluation of results.

# System Requirements

## Hardware Requirements

 CPU: AMD Ryzen 7 or equivalent.

 Memory: Minimum 16 GB RAM.

 Storage: At least 10 GB of available space.

## Software Requirements

|  |  |
| --- | --- |
| Operating System | Windows 11 or similar |
| Serverless Environment | AWS Lambda |
| Programming Language | Python 3.11 or later |
| Logging & Monitoring Service | AWS CloudWatch |
| Python libraries | Boto3, Numpy, Pandas, TensorFlow 2.x |
| Object Storage | AWS S3 |

Table 1: Components and corresponding software used in project

|  |  |
| --- | --- |
| qlearning\_agent.py | Python code for training the Q-learning agent |
| dqn.py | Python code for training the Deep Q-Learning agent |
| helper.py | Utility methods for invoking lambda functions and retrieving logs. |
| evaluate.py | Contains code for evaluating the results |
| image\_processing\_tasks/\* | Folder containing image processing tasks used for training the RL agents. |

Table 2: Details of artifacts

# Installation and Set Up

This section will guide you through the setting up the environment required for the training of Q-learning and Deep Q-learning agents. Also, instructions on how to install the software dependencies are included.

## AWS Resource Setup

This section assumes that the user has AWS account and appropriate permissions to create and manage AWS Lambda, S3, and CloudWatch resources.

### AWS S3 bucket

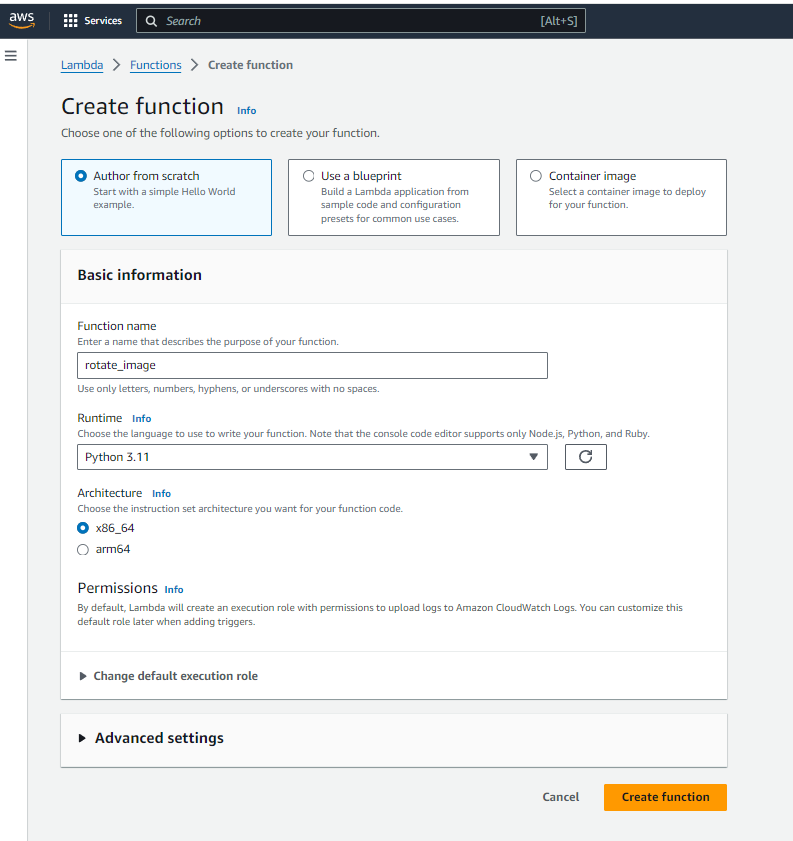
The purpose of the S3 bucket is to store the images required for processing by serverless functions.

1. Log in to the AWS Management Console.
2. Navigate to the S3 service and create a new bucket, say ‘my\_imagebucket’. The bucket name must be unique.
3. To my\_imagebucket’, upload images of different sizes from Flickr-Faces-HQ dataset.

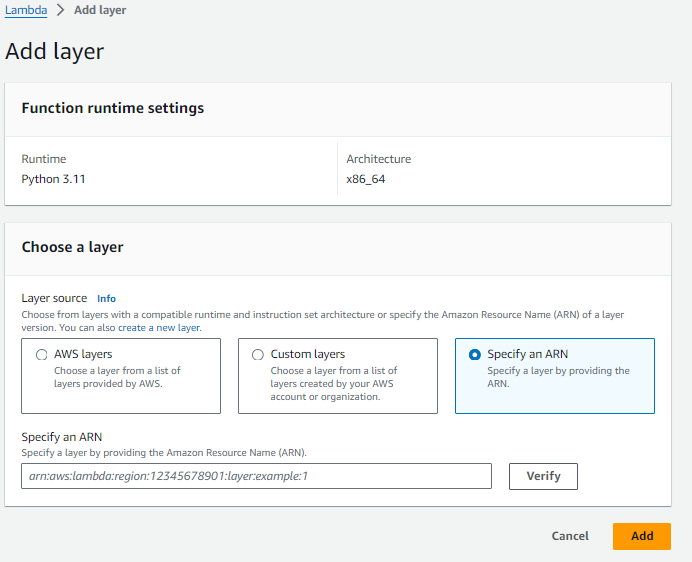
### AWS Lambda

In this research project, AWS lambda acts as the serverless environment for the Q learning and Deep Q-learning agent to interact with and learn. You must deploy the image processing functions given in the folder image\_processing\_tasks under github repository as AWS lambda functions. The following steps will walk you through the process.

1. Navigate to the AWS Lambda Console.
2. From console choose ‘Create function’:
   1. Select "Author from scratch".
   2. Enter a function name denoting the image processing function you are about to deploy.
   3. Choose Python 3.11 for the runtime.
   4. Set up the execution role with appropriate permissions (Lambda, S3, CloudWatch). You can choose ‘Create a new role with basic Lambda permissions’ and later add permissions for S3 and CloudWatch to this role.
   5. Click ‘Create function’
3. Copy the rotate\_image.py function from the folder image\_processing\_tasks under github repository and paste it into the Code source section of newly created lambda function.



Step 4: Now for the image processing tasks, python library has to be added as dependency. In AWS lambda, dependency libraries can be added using layers. Under layers section, choose ‘Add a Layer’.



Step 5: Choose option Specify an ARN and give the following ARN “arn:aws:lambda:ap-south-1:770693421928:layer:Klayers-p311-Pillow:4”

Step 6: Click verify. Once verified click Add. Now the lambda function is ready for execution and you view the layer in the layers section



Repeat the Steps for each of the image processing functions in the image\_processing\_tasks folder.

### AWS CloudWatch

The CloudWatch collects the logs of the execution of lambda function.

1. Navigate to CloudWatch console.
2. Go to Log groups
3. Create new log group with name /aws/lambda/<lambda funciton name>

Ensure that lambda function has proper rights to write to the CloudWatch log group created.

## Installation guide

1. Install Python 3.11 from [python.org](https://www.python.org/downloads/).
2. Optional, Create and activate python virtual environment
3. Install TensorFlow using pip : **pip install tensorflow**
4. Install numpy, pandas and matpltlib : **pip install numpy pandas matplotlib**
5. Verify that Tensorflow and other libraries are installed properly
6. Ensure that git is available in the system. Now clone the project artifact from Github repository by executing git clone <https://github.com/johns-thomas/ric_implementation.git>

# Configuration Settings

The Q-learning and Deep Q-learning agents can be configured using various parameters. These parameters affect the training time as well as the resultant models.

The following section discusses the parameters used during the training of the Q-learning and Deep Q learning agents.

## Q learning agent configuration

Various configurations available for Q learning agent present in qlearning\_agent.py is shown in Figure 1.

Variable q\_table\_file\_path is the name for the Q-table generated during the training, which is saved in .npy format.

Variable state\_data\_path saves the state data during the training of Q -learning agent for future reference in text format.

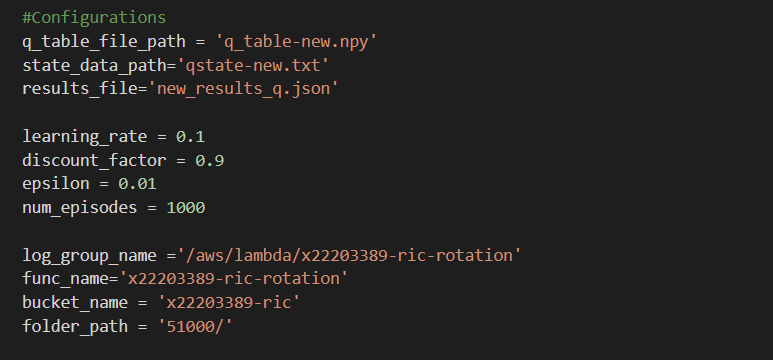


Figure 1: Configurations for Q learning agent

Variable results\_file is represents name of the results of training in a json file, which has data like episode number, reward, memory configurations, timeout, cost incurred.

The learning rate of the q learning can be configured by setting variable learning\_rate. In the research learning rate is taken as 0.1. Discount factor for Q learning agent is taken as 0.9 by setting the variable discount\_factor. Additionally, you can configure epsilon which controls the exploration rate of the q learning agent. In the project it is taken to be 0.01. Also, depending on the need, number of episodes can also be increased.

The name of the serverless function and its associated cloudwatch log group name can be configured by setting variable func\_name and log\_group\_name respectively. The S3 bucket containing the images can be configured by setting variable bucket\_name and folder\_path. folder\_path is just the name of the images folder.

## Deep Q-learning agent configuration

Various configurations available for Q learning agent present in qlearning\_agent.py is shown in Figure 2.

The learning rate of the dqn learning can be configured by setting variable learning\_rate and is taken as 0.1. Discount factor for Q learning agent is taken as 0.9 by setting the variable gamma. Additionally, you can configure epsilon which controls the exploration rate of the DQN learning agent. In the project it is taken to be 0.1. The exploration rate can be decayed over episodes. epsilon\_decay determines how of the exploration value to be reduced and epsilon\_min determines minimum possible exploration rate for the training.

The name of the serverless function and its associated cloudwatch log group name can be configured by setting variable func\_name and log\_group\_name respectively. The S3 bucket containing the images can be configured by setting variable bucket\_name and folder\_path. folder\_path is just the name of the images folder.

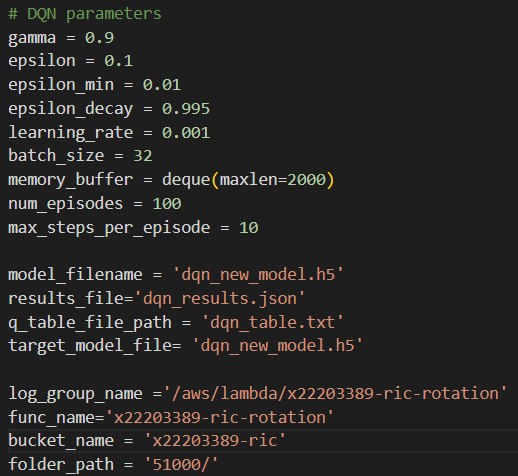


Figure 2: Configuration options for DQN learning agent

The model\_filename represents the deep neural network undergoing the training and it can be saved for running long training cycles. The target\_model\_file variable represents the actual deep neural network which holds the information about the Q-table. When training of DQN agent finishes it is stored as ‘.h5’ file. This deep neural network model file can be deployed to candidate serverless functions to optimize their configurations.

# Running the Software

To run the training of Q learning agent.

1. Ensure that you have set up AWS credentials to use AWS SDK in your system.
2. Go to command line

Step 3: Run **python qlearning\_agent.py**

Step 4: Wait until the training finishes, it may take up to 24 hours to complete 100 episodes

Step 5: After the training, use new\_results\_q.json for analysis, also you can find the qtable with name q\_table-new.py which can be used for configuring other serverless functions.

To run the training of DQN learning agent.

1. Ensure that you have set up AWS credentials to use AWS SDK in your system.
2. Go to command line

Step 3: Run **python dqn.py**

Step 4: Wait until the training finishes, it may take upto 28 hours to complete 100 episodes

Step 5: After the training, use dqn\_results.json for analysis.

Step 6: Use the dqn\_new\_model for configuring other serverless functions.

Note: The q learning and DQN learning agent take some iterations to optimally configure unseen serverless functions not used while training as in the case of any reinforcement learning algorithm.

# References

**References should be formatted using APA or Harvard style as detailed in NCI Library Referencing Guide available at** [**https://libguides.ncirl.ie/referencing**](https://libguides.ncirl.ie/referencing)

**You can use a reference management system such as Zotero or Mendeley to cite in MS Word.**

Beloglazov, A. and Buyya, R. (2015). Openstack neat: a framework for dynamic and energy-eﬃcient consolidation of virtual machines in openstack clouds, *Concurrency and Computation: Practice and Experience* 27(5): 1310–1333.

Feng, G. and Buyya, R. (2016). Maximum revenue-oriented resource allocation in cloud, *IJGUC* 7(1): 12–21.

Gomes, D. G., Calheiros, R. N. and Tolosana-Calasanz, R. (2015). Introduction to the special issue on cloud computing: Recent developments and challenging issues, *Computers & Electrical Engineering* 42: 31–32.

Kune, R., Konugurthi, P., Agarwal, A., Rao, C. R. and Buyya, R. (2016). The anatomy of big data computing, *Softw., Pract. Exper.* 46(1): 79–105.