

myRL Package User Manual

This user manual contains prerequisites, setup, and example use of the myRL package. All of the code shown in this section can be found and tested at <https://github.com/lg661/MSc-Computer-Science-Dissertation/tree/main/myRL%20Package/myRL%20Example%20Code>

Prerequisites:

There are several prerequisites to using the myRL package:

1. Use the google colab IDE.
2. Write the code for your algorithm in Python.
3. Use the pyTorchLightning module to build your RL algorithm.
4. Train your algorithm on an OpenAI Gym environment.
5. The class representing your RL algorithm must contain two attributes: 'episodic rewards' (that should contain a list of the rewards earned during each episode of training), and 'agent' (which should hold an exact copy of the actor/control neural network at the end of training).

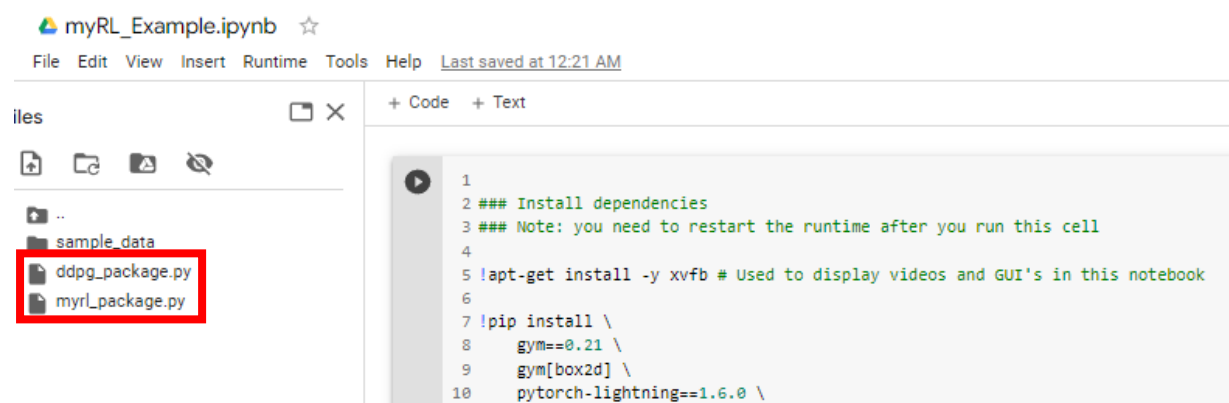
Setup:

Follow steps 1-5 below to setup the myRL package in your google colab environment.

1. Install dependencies to the system:

```
1
2 ### Install dependencies
3 ### Note: you need to restart the runtime after you run this cell
4
5 !apt-get install -y xvfb # Used to display videos and GUI's in this notebook
6
7 !pip install \
8     gym==0.21 \
9     gym[box2d] \
10    pytorch-lightning==1.6.0 \
11    pyvirtualdisplay
12
13 # gym -> Environment
14 # gym[box2d] -> Dependency for box2d category in the gym
15 # pytorch-lightning -> Reinforcement learning and neural networks
16 # pyvirtualdisplay -> Used to display videos in the environment
```

2. Drag and drop your RL algorithm python file and the myRL python file into the google colab files section. Your RL algorithm python file should contain your RL algorithm and all its dependencies e.g. Actor class, DQN class, etc. It should most notably contain the class representing your algorithm.



3. Run each package in google colab. It is important to note that you need to manually press the pause button after a few seconds to terminate the cell.

```
1 ### Run the ddpG and myRL package
2
3 !python ddpG_package.py
4 !python myrl_package.py
```

4. Import the RL algorithm class from the RL algorithm python file, and the myRL, TrainingPlan, TuningPlan, and TestingPlan classes from the myRL python file.

```
1 ### Import specific classes and functions from each package
2
3 from ddpG_package import DDPG
4 from myrl_package import myRL, TrainingPlan, TuningPlan, TestingPlan
```

You are now setup to use the myRL package with the RL algorithm contained in your RL algorithm python file.

Creating objects:

This section instructs developers how to create objects from the myRL, TrainingPlan, TuningPlan, and TestingPlan classes. Table 4, 5 and 6 (see below) contain details of the parameters involved with the initialization of the TrainingPlan, TuningPlan, and TestingPlan classes. Note, a myRL class initialization table is not included since it contains no parameters.

Table 4: A description of the parameters used to initialize a TrainingPlan object

TrainingPlan Object Initialization Details		
Parameters	Type	Description
name	str	The name of the training plan
algo	N/A	The RL algorithm class
eps	int	The number of episodes in a full training cycle
its	int	The number of iterations of full training cycles
env_name	str	The name of the OpenAI Gym environment used to train the agents

Table 5: A description of the parameters used to initialize a TuningPlan object

TuningPlan Object Initialization Details		
Parameters	Type	Description
name	str	The name of the tuning plan
trainingPlan	TrainingPlan	The TrainingPlan object used by the tuning plan (to run the training cycles for each parameter value)
param	str	The name of the parameter being tuned
vals	list[int/float]	The values of the arguments to be tested during the tuning process

Table 6: A description of the parameters used to initialize a TestingPlan object

TestingPlan Object Initialization Details		
Parameters	Type	Description
name	str	The name of the testing plan
agent	Agent Class	The agent to be tested
eps	str	The number of episodes used to test the agent
env_name	str	The name of the OpenAI Gym environment used to test the agent

myRL Methods:

Below is a table containing details of the methods available to objects of the myRL class.

Table 7: A description of the methods held by objects of the myRL class

myRL Method Details					
Method	Description	Parameter name	Param Type	Return description	Return Type
Training					
addTrainingPlan	Add a training plan	trainingPlan	TrainingPlan	None	None
removeTrainingPlan	Remove a training plan	name	str	None	None
runTrainingPlan	Run a training plan	name	str	Training plan results	list
getTrainingPlanResults	Get the results of a training plan	name	str	Training plan results	list
		it	int		
		download	bool		
showTrainingPlans	Show the details of all stored training plans	None	None	None	None
Tuning					
addTuningPlan	Add a tuning plan	tuningPlan	TuningPlan	None	None
removeTuningPlan	Remove a tuning plan	name	str	None	None
runTuningPlan	Run a tuning plan	name	str	Tuning plan results	dict
getTuningPlanResults	Get the results of a tuning plan	name	str	Tuning plan results	dict or list
		val	int or float		
		it	int		
		download	bool		
showTuningPlans	Show the details of all stored tuning plans	None	None	None	None
Testing					
addTestingPlan	Add a testing plan	testingPlan	TestingPlan	None	None
removeTestingPlan	Remove a testing plan	name	str	None	None

runTestingPlan	Run a testing plan	name	str	Testing plan results	list
getTestingPlanResults	Get the results of a testing plan	name	str	Testing plan results	list
		download	bool		
showTestingPlans	Show the details of all stored testing plans	None	None	None	None
Display					
displayTrainingResults	Graphically display the results of a training plan	trainingPlanName	str	None	None
		it	int or None		
		mean_its	bool		
		moving_average	int		
		figSize	(int, int)		
		x_label	str		
		y_label	str		
		legend	bool		
		legend_size	int		
		legend_loc	int		
		titleSize	int		
		labelSize	int		
displayTuningResults	Graphically display the results of a tuning plan	tuningPlanName	str	None	None
		val	int or float or None		
		it	int or None		
		mean_its	bool		
		moving_average	int		
		figSize	(int, int)		
		x_label	str		
		y_label	str		
		legend	bool		
		legend_size	int		
		legend_loc	int		
		titleSize	int		
		labelSize	int		
displayTestingResults	Graphically display the results of a testing plan	testingPlanName	str	Mean reward of the tests	float
		figSize	(int, int)		
		x_label	str		
		y_label	str		
		legend	bool		
		legend_size	int		
		legend_loc	int		
		titleSize	int		
		labelSize	int		
Agent Handling					
getAgent	Get an agent from the myRL object	trainingPlanName	str	Agent (neural network of the actor)	Actor neural network
		tuningPlanName	str		
		val	int or float		
		it	int		

Example code:

This section gives a demonstration of how the code may be used, and examples of what the outputs (to the console) of such code could look like. The code responsible for an output is connected to that output by a red line.

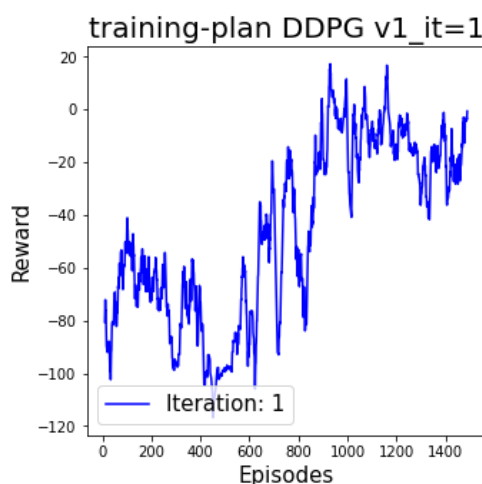
myRL object creation example:

```
3 # Create an object 'myRL_hub' from the myRL class
4 myRL_hub = myRL()
```

Training plan example:

```
7 ### TRAINING
8
9
10 # Create a training plan object from the TrainingPlan object
11 trainingPlan_v1 = TrainingPlan(algo=DDPG, eps=1500, its=2, name="training-plan DDPG v1", env_name="BipedalWalker-v3")
12
13 # Add the training plan to the myRL object
14 myRL_hub.addTrainingPlan(trainingPlan_v1)
15
16 # Show the training plans stored in the myRL object
17 myRL_hub.showTrainingPlans()
18
19 # Run the training plan
20 myRL_hub.runTrainingPlan("training-plan DDPG v1")
21
22 # Get the results for only iteration 1, and download them as an excel file
23 myRL_hub.getTrainingPlanResults(name="training-plan DDPG v1", it=1, download=True)
24
25 # Display the results of the training plan
26 myRL_hub.displayTrainingResults(trainingPlanName="training-plan DDPG v1", mean_its=False, it=1, legend=True, legend_loc=3, moving_average=20)
```

Name	Episodes	Iterations
training-plan DDPG v1	1500	2



Tuning plan example:

```

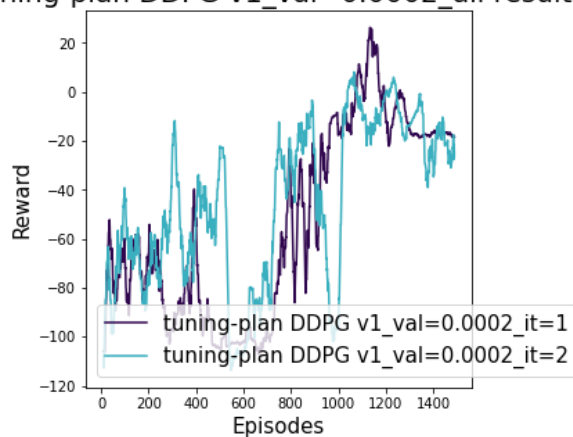
29 ### TUNING
30
31
32 # Create a tuning plan object from the TuningPlan object
33 tuningPlan_v1 = TuningPlan(trainingPlan=trainingPlan_v1, param="actor_lr", vals=[0.0005, 0.0002], name="tuning-plan DDPG v1")
34
35 # Add the tuning plan to the myRL object
36 myRL_hub.addTuningPlan(tuningPlan_v1)
37
38 # Show the tuning plans stored in the myRL object
39 myRL_hub.showTuningPlans()
40
41 # Run the tuning plan
42 myRL_hub.runTuningPlan("tuning-plan DDPG v1")
43
44 # Get the results for the specific value tested (0.0005), and don't download the results as an excel file
45 myRL_hub.getTuningPlanResults(name="tuning-plan DDPG v1", val=0.0005, download=False)
46
47 # Display the results of the training plan
48 myRL_hub.displayTuningResults(tuningPlanName="tuning-plan DDPG v1", mean_its=False, val=0.0002, legend=True, legend_loc=3, moving_average=20)

```

Name	Parameter	Training Plan	Value
tuning-plan DDPG v1	actor_lr	training-plan DDPG v1	[0.0005, 0.0002]

Name	Environment	Number of episodes
testing-plan 1	BipedalWalker-v3	10
testing-plan 2	BipedalWalker-v3	10

tuning-plan DDPG v1_val=0.0002_all results



testing-plan 1



Testing plan example:

```

51 ### TESTING
52
53
54 # Get the agents to be tested
55 Agent_1 = myRL_hub.getAgent(trainingPlanName = "training-plan DDPG v1", it=1)
56 Agent_2 = myRL_hub.getAgent(tuningPlanName = "tuning-plan DDPG v1", it=1, val=0.0005)
57
58 # Create the testing plans
59 testingPlan_1 = TestingPlan(name="testing-plan 1", env_name="BipedalWalker-v3", eps=10, agent=Agent_1)
60 testingPlan_2 = TestingPlan(name="testing-plan 2", env_name="BipedalWalker-v3", eps=10, agent=Agent_2)
61
62 # Add the testing plans to the myRL object
63 myRL_hub.addTestingPlan(testingPlan=testingPlan_1)
64 myRL_hub.addTestingPlan(testingPlan=testingPlan_2)
65
66 # Show the testing plans stored in the myRL object
67 myRL_hub.showTestingPlans()
68
69 # Run the test
70 myRL_hub.runTestingPlan(name="testing-plan 1")
71
72 # Get and download the results of the test
73 myRL_hub.getTestingPlanResults(name="testing-plan 1", download=True)
74
75 # Display the results of the testing plan
76 myRL_hub.displayTestingResults(testingPlanName="testing-plan 1", legend=True, legend_loc=3)

```