To use reinforcement learning algorithms

1. Install Python package of OpenVO library from github repository.

2. Next step, create project in PyCharm, and create file \*.py.

3. For example for SARSA algorithm, add next libs:

import torch  
import torch.optim as optim  
  
from Open\_VO import BaseAgentSARSA, BaseModelSARSA, BaseTrainerSARSA, MemoryCell

4. After, you should create code construction like this:

4.1. Initialize necessary variables:

maxMemory = 100000  
batchSize = 1000  
randCoef = 60  
randRange = 200  
inputSize = 4  
hiddenSizes = [32, 16]  
numClasses = 4  
lr = 0.001  
gamma = 0.9  
device = torch.device('cpu')

4.2. Get environment state:

oldState = [1, 0, 0, 1]  
newState = [0, 1, 0, 1]

\* For the environment, you can use class interface “BaseEnvironment”.

4.3. Predict next move for inference:

finalMove = agent.act(oldState)  
newMove = agent.act(newState)  
print(finalMove, newMove)

4.4. Or train model:

cell = MemoryCell(state=oldState, action=finalMove, next\_state=newState, next\_action=newMove, reward=2.3, done=False)  
  
agent.train\_short\_memory(cell)  
agent.remember(cell)  
agent.remember(cell)  
agent.remember(cell)  
agent.train\_long\_memory()

Keep in mind that different algorithms use different train pipeline and operates different variables. The RL module includes 5 algorithms:

* DQN
* SARSA
* PG
* A2C
* PPO

To understand math, read more about required algorithm.