

Полный код ко всем заданиям (выполненным) есть на гитхабе
Важные кусочки кода

```
class ACArgs:
    def __getitem__(self, key):
        return getattr(self, key)
    def __setitem__(self, key, val):
        setattr(self, key, val)
    def __contains__(self, key):
        return hasattr(self, key)
    env_name = 'CartPole-v0' #@param ['CartPole-v0', 'InvertedPendulum-v2', 'HalfCheetah-v2']
    exp_name = 'q4_ac' #@param
    ## PDF will tell you how to set ep_len
    ## and discount for each environment
    ep_len = 200 #@param {type: "integer"}
    #@markdown batches and steps
    batch_size = 1000 #@param {type: "integer"}
    eval_batch_size = 400#@param {type: "integer"}
    n_iter = 100 #@param {type: "integer"}
    num_agent_train_steps_per_iter = 1 #@param {type: "integer"}
    num_actor_updates_per_agent_update = 1 #@param {type: "integer"}
    num_critic_updates_per_agent_update = 1 #@param {type: "integer"}
    #@markdown Actor-Critic parameters
    discount = 0.9#@param {type: "number"}
    learning_rate = 5e-3 #@param {type: "number"}
    dont_standardize_advantages = False #@param {type: "boolean"}
    num_target_updates = 10 #@param {type: "integer"}
    num_grad_steps_per_target_update = 10 #@param {type: "integer"}
    n_layers = 2 #@param {type: "integer"}
    size = 64 #@param {type: "integer"}
    #@markdown system
    save_params = False #@param {type: "boolean"}
    no_gpu = False #@param {type: "boolean"}
    which_gpu = 0 #@param {type: "integer"}
    seed = 1 #@param {type: "integer"}
    #@markdown logging
    ## default is to not log video so
    ## that logs are small enough to be
    ## uploaded to gradscope
    video_log_freq = -1#@param {type: "integer"}
    scalar_log_freq = 10 #@param {type: "integer"}
args = ACArgs()
if args['video_log_freq'] > 0:
    import warnings
    warnings.warn(
        "\nLogging videos will make eventfiles too"
        "\nlarge for the autograder. Set video_log_freq = -1"
        "\nfor the runs you intend to submit."
    )

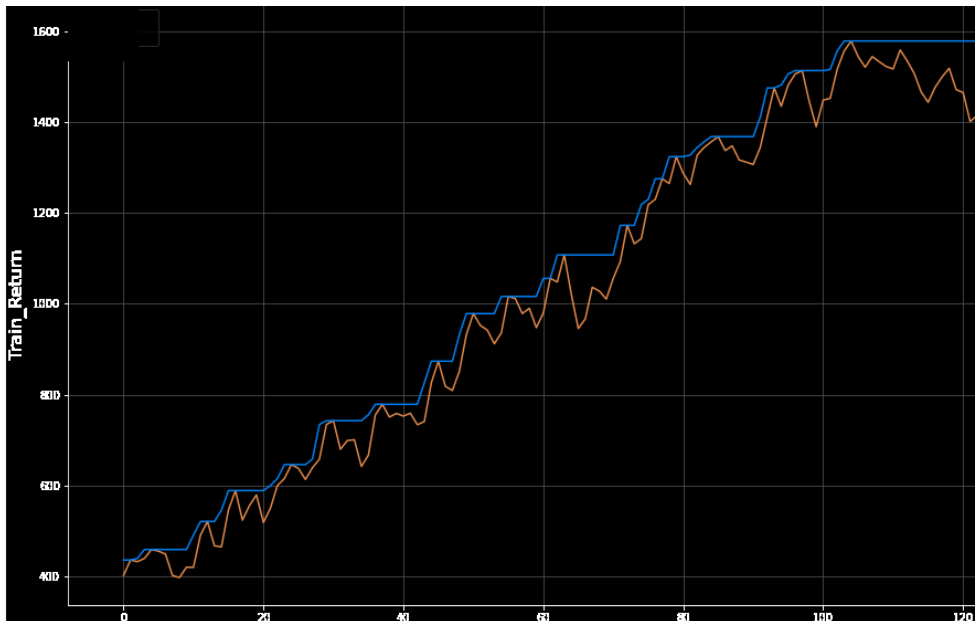
4 trainer = Q_Trainer(args)
5 trainer.run_training_loop()
```

Задание 1

Качество базовой версии Q-обучения.

*скриншоты могут немного съехать из-за моей версии редактора

Посмотрим на график

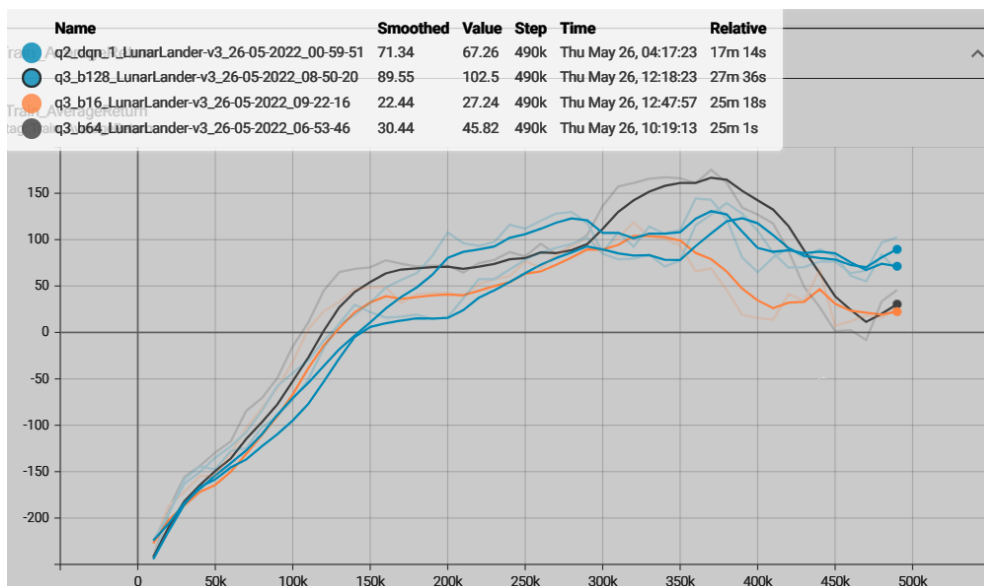


Где синий это “среднее”, а оранжевый это “лучший”

Задание 2

DQN VS DDQN

DDQN визуально лучше:



Времени на перебор много параметров не было, но несколько других случаев тоже было рассмотрено, например с num_target updates - steps 10 и 10. 20 и 20 . Также менялся размера батча в промежутке [16, 32, 64, 128]