GANs for generating novel policies/rewards

06.07.2018

Pablo Campos Viana
MSc. Operational Research with Data Science
The University of Edinburgh

I - Goal of the project

In short, the main goal is to explore generative neural networks applied to Policy/Reward generation. In particular, since Generative Adversarial Networks (GANs) is a recently hot topic with proven success when applied to images and text, the general aim of the project is to explore the idea of applying GANs in the context of Reinforcement Learning (RL), namely on policy models and reward functions.

The completion criteria can be summarized in three stages:

S1) Train an RL agent on multiple tasks.

S2) Train a GAN on their policy/reward models.

S3) Apply GAN to accelerate new learning.

Besides, the idea is to complete each of the three stages using simple approaches and applied to a simple RL environment in order to eventually make incremental improvements.

II - Methods used

The methods used in each stage are the following.

S1) The environment used is Open AI Gym's **FrozenLake** with size 8x8. Standard **(tabular) Q-learning** algorithm to train the RL agent on multiple tasks. Standard **linear regression** to estimate the reward functions.

S2) GANs: Generator network with enough flexibility to capture well the distribution of policies/rewards, LeakyReLU activations and BatchNorm layers. Discriminator network with limited flexibility to avoid fast convergence. Sigmoid activation on the last layer of both networks. Binary cross entropy is used as objective function and the Adam learning rule is used for the optimization.

S3) GANs: Similar to **S2).**

Accomplishments

The accomplishments in each stage so far are as follows.

S1) Written Python scripts to: **create multiple tasks** (i.e. FrozenLake environments with pseudorandom location of start, holes and goal, where the task is to go from somewhere in north to somewhere in south, south to north, east to west and west to east) **perform Q-learning** to find optimal Q function and optimal Policy, **estimate the reward function** via linear regression (with one hot encoding on (S,A,S')) and **save results** of each created environment in order to eventually feed a GAN with this collected data.

S2) Once I have collected data for 800 (200 games per each possible direction) games and performed Q-learning for 2000 episodes (enough to facilitate both exploration and exploitation) and found optimal Q(s,a) function, optimal Policy π(s) and fitted linear regression to parametrize a reward function R(s,a,s) for each game, I have trained GANs:

1) over optimal Policies **π(s)** (vectors) to try to learn distribution of plausible optimal paths.
2) over learned reward functions **R(s,a,s')** to try to learn its distribution (i.e. distribution of parameters of linear regression models).

S3) Exploration regarding to the application setting where the agent is free to play in an environment, but what is costly is the human supervision of its reward for its RL. The pipeline is as follows:

Phase 1 training (unrewarded, target map): User gives no task (goal), but the agent make up new rewards via its generator, and learn to solve them with RL.

Phase 2 training (rewarded, target map): Using the same target map, the user now specifies a task/extrinsic goal, and the agent should learn to solve it as quickly as possible.

Partial results suggest that the agent would be indeed able to solve the target task more quickly rather than solving it from scratch.

Remaining work

The remaining work for now is concerned with **Stage 3**, i.e. accelerate new learning on both policies and rewards.

Regarding **Policies**, the following approaches will be tested:

1) **Initialisation:** Initialise the search of the target problem by sampling the generator
2) **Regularization:** Optimize the target problem cost by a) regularising the discriminator in a such a way to prefer policies that look "real", and not waste time on those that look "fake", and b) regularising the search to prefer policies that have high likelihood under the generator.
3) **Search in the latent space:** Rather than search policy parameters directly, search the noise parameter of GAN, each of which correspond to some kind of policy.

Besides, **the baselines** for these experiments will be:

(i) Learn the target problem from scratch

(ii) Learn the target problem regularised by the mean of the source problems

while **the metrics** considered so far are:

(i) Asymptotic performance at convergence of RL on the target problem,

(ii) Time (iterations/environmental rollouts) to achieve a given “solved” performance level.

(iii) Area under the reward vs iterations curve.

Regarding **Rewards**, the remaining work can be summarized as:

Continue with experiments and formalize results regarding the experiment stated about **S3** in the previous **Accomplishments** section.

Besides**, the baselines** for these experiments will be:

(i) Learn the target problem from scratch

while **the metric** considered so far is

(i) number of total interactions with environment.

Once completed **S3**, possible incremental approaches (and/or possibly more complex environments) will be considered and discussed with my first supervisor of the project.