Visualizing Improvements of Reinforcement Learning Models

Allan Erlang Videbæk (s164197), Simon Tommerup (s164224) and Thomas Heshe (s164399)

Introduction

In recent years reinforcement learning (RL) models have shown promising results in scenarios where a series of decisions have to be made in an unknown environment, e.g. an Atari game, where the input consists of the raw pixels. For this reason, RL models are often explored in the setting of computer games.

However, RL currently relies on both sensitive parameter tuning and lots of computer power, and often the models do not generalize to other settings.

In this project different RL models are explored and compared in a 'low GPU' setting, to find a good model, while getting an intuition for why it is good. The starting point is a PPO algorithm [1] and used in Procgen environments [3].

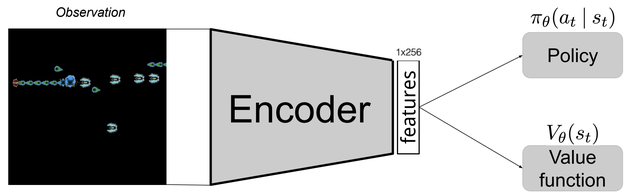
Approach and scope

Different models are explored by starting with a simple model and in turn fine tuning each parameter. Through this approach it is assumed that the parameters are independent of each other, which might lead to a suboptimal model but it makes the model exploration much cheaper.

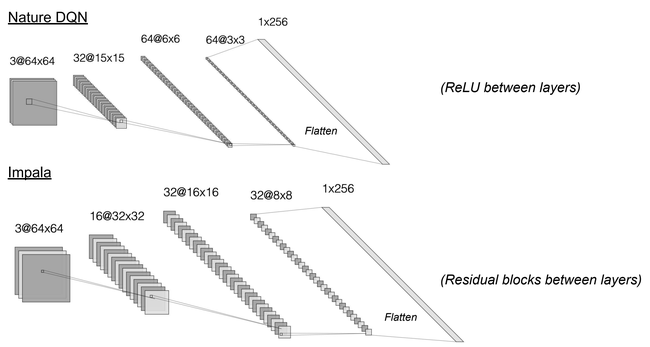
To aid this exploration, the gradients of the encoder network have been used to visualize which pixels play a role in the model.

Model

Takes an observation as input and produces a distribution of actions along with an estimate of the reward for the current state.



Encoders



PPO Objective

Policy gradient methods increase likelihood of 'good' actions based on $\hat{A}_t$

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t\left[\log \pi_\theta(a_t \mid s_t)\hat{A}_t\right]$$

which is changed slightly in proximal policy optimization (PPO) to

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

where $r_t(\theta)$ is now the change in $\pi_\theta(a_t \mid s_t)$ relative to before. This induces

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)],$$

where L_t^{VF} optimises the value estimate and S is an entropy bonus.

Visualizing with Saliency Maps

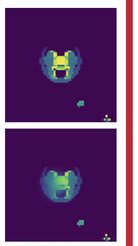
Which areas of a frame is important for deciding which actions to take? Blurring each pixel of a Procgen frame at time t in a loop

$$\Phi(F_t, i, j) = F_t \odot (1 - M(i,j)) + A(F_t, \sigma_A = 3) \odot M(i,j)$$

This can be interpreted as putting a visual uncertainty in a part of the image. We can then compute a saliency score for a pixel by

$$S(t,i,j) = \frac{1}{2} \parallel \pi_\theta(F_t) - \pi_\theta[\Phi(F_t, i, j)] \parallel^2$$

i.e we compute the mean squared difference in magnitude between the policy output (logits) on the original frame and the frame with a Gaussian blur centered at pixel (i,j) [2].

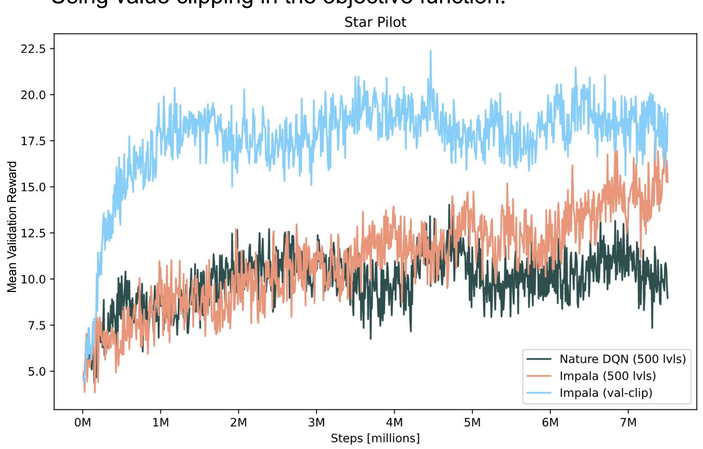


Experiments and Results

Parameters of Importance

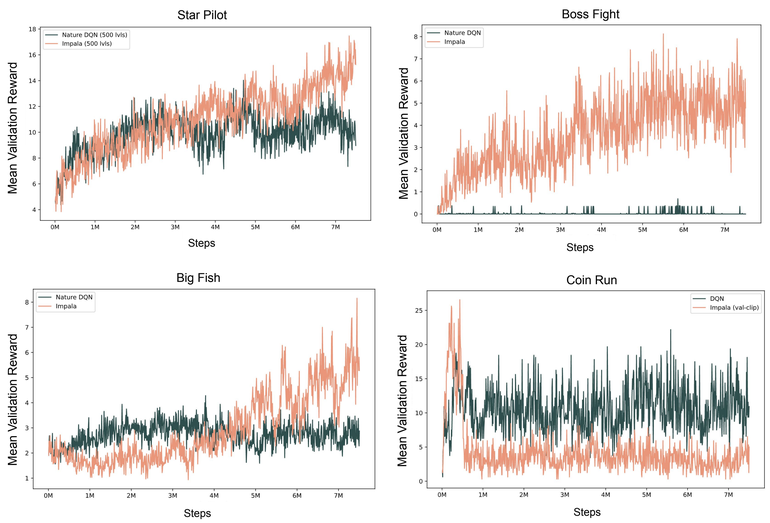
In line with the results in [3], the following parameter changes were found improve the performance of the model:

- Increasing the number of training levels.
- Changing the encoder network to Impala.
- Using value clipping in the objective function.



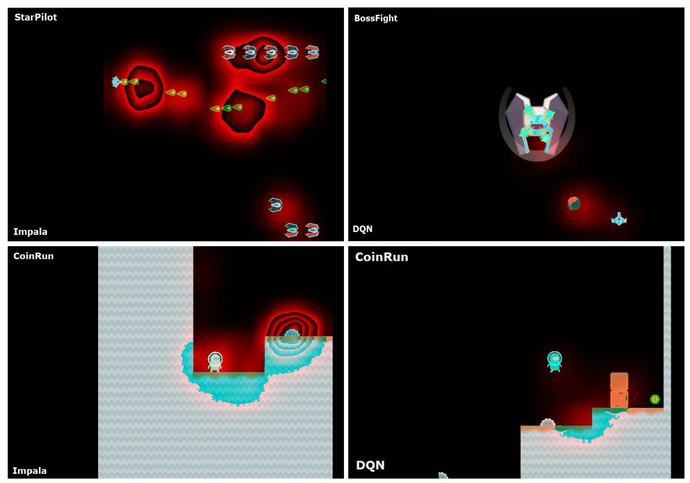
Generalizable Models

Impala model is mostly better at generalizing to other environments



Visualized Saliency Maps

The technique allows us to understand our network and choices in hyperparameters:



References: [1] John Schulman et al. (2017). "Proximal Policy Optimization Algorithms", [2] Sam Greydanus et al. (2018). "Visualizing and Understanding Atari Agents", [3] Karl Cobbe et al. (2017). "Leveraging Procedural Generation to Benchmark Reinforcement Learning"