VPG Implementation using PyTorch

<https://github.com/openai/spinningup/blob/master/docs/algorithms/vpg.rst>

The algorithm function for a PyTorch implementation performs the following tasks in (roughly) this order:

1. Logger setup
2. Random seed setting
3. Environment instantiation
4. Constructing the actor-critic PyTorch module via the actor\_critic function passed to the algorithm function as an argument
5. Instantiating the experience buffer
6. Setting up callable loss functions that also provide diagnostics specific to the algorithm
7. Making PyTorch optimizers
8. Setting up model saving through the logger
9. Setting up an update function that runs one epoch of optimization or one step of descent
10. Running the main loop of the algorithm:
    1. Run the agent in the environment
    2. Periodically update the parameters of the agent according to the main equations of the algorithm
    3. Log key performance metrics and save agent

# Method

1. Test implementation on a simple environment (ex : cartpole) puis Humanoid

# To do

* Implement A2C and SAC using Pytorch
* Benchmark others methods with Stable Baselines3
* Try our algorithm on new tasks (backflip ?)s