Member-only story

A Beginner's Guide to Reinforcement Learning Using Stable Baselines 3

FS Ndzomga Follow 3 min read · May 4, 2023

Photo by Ricardo Gomez Angel on Unsplash

Stable Baselines 3 (SB3) is a popular library for reinforcement learning in Python, providing a collection of high-quality implementations of RL algorithms such as Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC), and Deep Q-Network (DQN). In this tutorial, we will walk through the basics of using Stable Baselines 3 for training and evaluating RL agents.

Prerequisites

Before we begin, make sure you have Python 3.7 or later installed. You will also need to install the following libraries:

```
pip install stable-baselines3[extra] gym
```

Creating a Custom Gym Environment

To train an RL agent using Stable Baselines 3, we first need to create an environment that the agent can interact with. In this tutorial, we will use a simple example from the OpenAI Gym library called “CartPole-v1”:

```
import gym

env = gym.make("CartPole-v1")
```

Instantiating and Training an RL Agent

Now that we have our environment set up, we can instantiate an RL agent using an algorithm from the Stable Baselines 3 library. In this example, we will use the PPO algorithm:

```
from stable_baselines3 import PPO

model = PPO("MlpPolicy", env, verbose=1)
model.learn(total_timesteps=10000)
```

In the code above, we first import the `PPO` class from the Stable Baselines 3 library. We then create a PPO agent by passing the "MlpPolicy" (a feed-forward neural network policy), our environment, and a verbosity level to the PPO constructor. Finally, we train the agent using the `learn()` method, specifying the total number of time steps to train for.

Evaluating the Trained Agent

After training the agent, we can evaluate its performance by having it interact with the environment:

```
def evaluate_agent(model, num_episodes=100):
    success_count = 0
    total_reward = 0

    for _ in range(num_episodes):
        obs = env.reset()
        done = False
        episode_reward = 0

        while not done:
            action, _states = model.predict(obs)
            obs, reward, done, info = env.step(action)
            episode_reward += reward

        total_reward += episode_reward
        success_count += int(episode_reward >= 195)  # CartPole-v1 is considered

    avg_reward = total_reward / num_episodes
    success_rate = success_count / num_episodes

    return avg_reward, success_rate

# Evaluate the trained agent
avg_reward, success_rate = evaluate_agent(model)
print(f"Average reward: {avg_reward}, Success rate: {success_rate}")
```

Saving and Loading Trained Models

Stable Baselines 3 provides functionality to save and load trained models, which is useful for reusing agents across different sessions or sharing them with others. To save a trained model, use the `save()` method:

```
model.save("ppo_cartpole")
```

To load a saved model, use the `load()` method:

```
loaded_model = PPO.load("ppo_cartpole")
```

Conclusion

In this tutorial, we have covered the basics of using Stable Baselines 3 for reinforcement learning, including creating a custom environment, instantiating and training an RL agent, evaluating the trained agent, and

saving and loading models. The library provides a wide range of algorithms and tools for building and training more complex agents and environments, making it an excellent choice for both beginners and experienced researchers in reinforcement learning.

Next Steps

To further explore Stable Baselines 3 and its capabilities, consider the following resources:

1. Official Documentation: The official documentation for Stable Baselines 3 is an excellent resource for learning more about the available algorithms, policies, and features provided by the library. Visit the documentation at https://stable-baselines3.readthedocs.io/.
2. Custom Environments: While we used a pre-built Gym environment in this tutorial, you can create your own custom environments to solve specific problems. Learn more about creating custom environments in the OpenAI Gym documentation: https://gym.openai.com/docs/.
3. Hyperparameter Tuning: The performance of RL algorithms can often be significantly improved by tuning their hyperparameters. Read the Stable Baselines 3 guide on hyperparameter tuning for more information: https://stable-baselines3.readthedocs.io/en/master/guide/rl_tips.html.
4. Advanced Features: Stable Baselines 3 provides many advanced features, such as support for recurrent policies, parallel environments, and custom policies. Explore these features in the official documentation and examples provided by the library.
5. RL Algorithm Exploration: Stable Baselines 3 supports various RL algorithms, such as DQN, A2C, and TD3, in addition to PPO. Experiment with different algorithms to understand their strengths and weaknesses and identify the best algorithm for your specific problem.

By diving deeper into the capabilities of Stable Baselines 3, you can develop more sophisticated reinforcement learning agents and tackle a wide range of problems in various domains, such as robotics, finance, healthcare, and more.

BECOME a WRITER at MLearning.ai

Mlearning.ai Submission Suggestions

How to become a writer on Mlearning.ai

medium.com

Reinforcement Learning | Python | Gpt 3 | Machine Learning | MI So Good

Written by FS Ndzomga

2.3K followers · 425 following

Follow

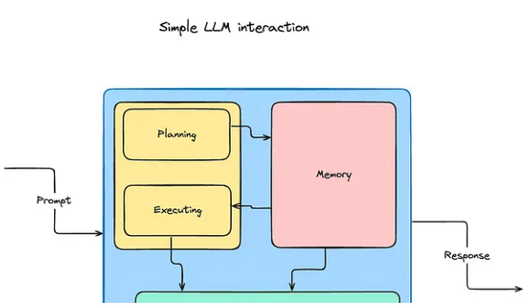
Engineer passionate about data science, startups, philosophy and French literature. Built lycee.ai, discute.co and rimbaud.ai . Open for consulting gigs

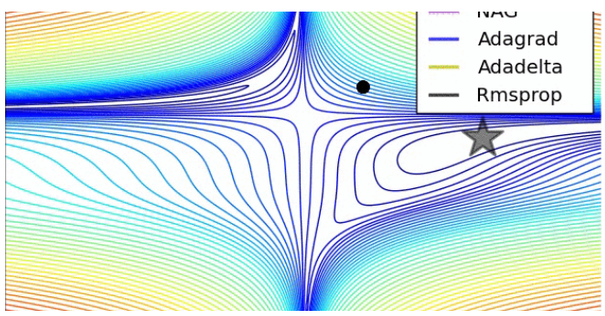
No responses yet

Jagannath Rao

What are your thoughts?

More from FS Ndzomga





In Thoughts on Machine Learning by FS Ndzomga

Building Powerful Open Source AI Agents with DSPy, LangGraph,...

Maximilian Vogel

The ChatGPT list of lists: A collection of 3000+ prompts, GPT...

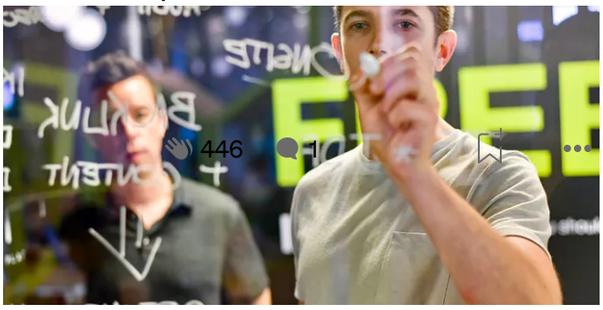
Updated Aug-8, 2025. Added New Models, Prompts, Lists and Tools

Feb 7, 2023 13.8K 197

Musstafa

Optimizers in Deep Learning

What is an optimizer?



In Thoughts on Machine Learning by FS Ndzomga

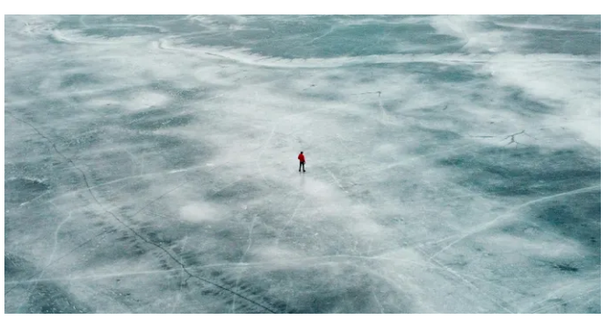
DSPy + LangChain: A Powerful Mix For Automatic Prompt...

DSPy is effective for automatic prompt optimization. The only constraint is that you...

Feb 29, 2024 300 1

See all from FS Ndzomga

Recommended from Medium

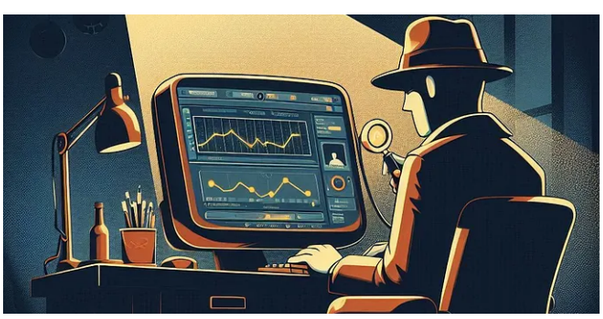


mohsin

Reinforcement Learning in Action: Surviving the FrozenLake with Q-...

“How does an AI learn not to fall into holes?” In this post, we’ll explore two classic...

Apr 8 1



The Scraper Guy

Amazon Price Tracker Telegram Bot using Python

Easiest method to track amazon products and get updates if they have dropped in price to ...

Jun 2 17

Bootcamp AI

Python Web Scraping: Step-By-Step

Web scraping with Python is one of the easiest and fastest ways to build large...

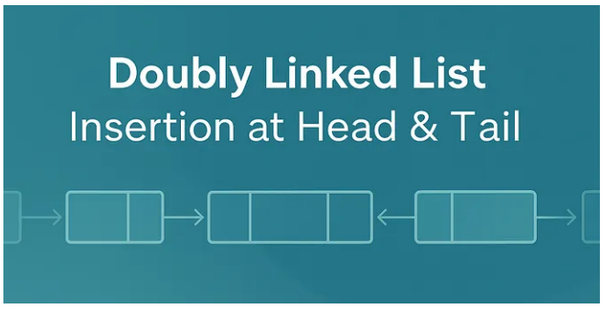
Sep 23 2 1

Mandeep Singh

Q-Learning: From Bellman Equations to Model-Free Learning

1) Introduction

Aug 13 26

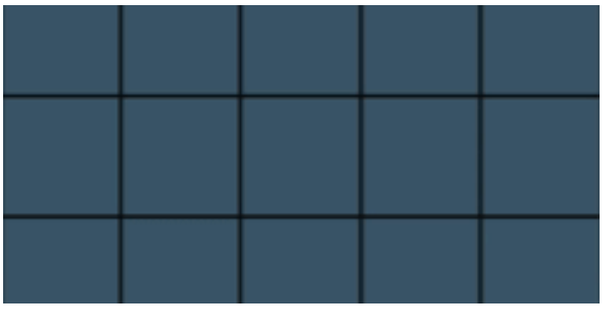


In Coding Nexus by YogiCode

DSA Lecture 45 : Insertion in Doubly Linked List For Head & Tail

Step by step guide with code and visuals.

Sep 13 87



In Artificial Intelligence in Plain English by Rem E

How RL Agents See the World: Gymnasium Basics

Tutorial 1 explores how an agent interacts with its environment using Gymnasium and...

Jul 4 51

See more recommendations