МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа № 7
по дисциплине «Методы машинного обучения»

Тема: « Алгоритмы Actor-Critic.»

ИСПОЛНИТЕЛЬ: Ли Яцзинь
ФИО
группа ИУ5И-21М
подпись"

ПРЕПОДАВАТЕЛЬ: Гапанюк Ю .Е

Москва - 2024

Задание:

- Реализуйте любой алгоритм семейства Actor-Critic для произвольной среды.

```
import matplotlib.pyplot as plt

# Actor Critic Network
class ActorCritic(nn.Module):
        def __init__(self, num_inputs, num_actions, hidden_size=128):
                super(ActorCritic, self).__init__()
                self.common = nn.Sequential(
                        nn.Linear(num_inputs, hidden_size),
                        nn.ReLU()
                )
                self.actor = nn.Sequential(
                        nn.Linear(hidden_size, num_actions),
                        nn.Softmax(dim=-1)
                )
                self.critic = nn.Linear(hidden_size, 1)

        def forward(self, x):
                x = self.common(x)
                return self.actor(x), self.critic(x)


# Training function
def train(env_name='CartPole-v1', num_episodes=300, gamma=0.99, lr=0.001):
        env = gym.make(env_name)
        num_inputs = env.observation_space.shape[0]
        num_actions = env.action_space.n

        model = ActorCritic(num_inputs, num_actions)
        optimizer = optim.Adam(model.parameters(), lr=lr)

        all_rewards = []
```

✓ 8 秒 完成时间：18:46

Рисунок 1- код алгоритма

```
Episode 9930, Reward: 10.0
Episode 9940, Reward: 10.0
Episode 9950, Reward: 9.0
Episode 9960, Reward: 9.0
Episode 9970, Reward: 9.0
Episode 9980, Reward: 10.0
Episode 9990, Reward: 10.0
```

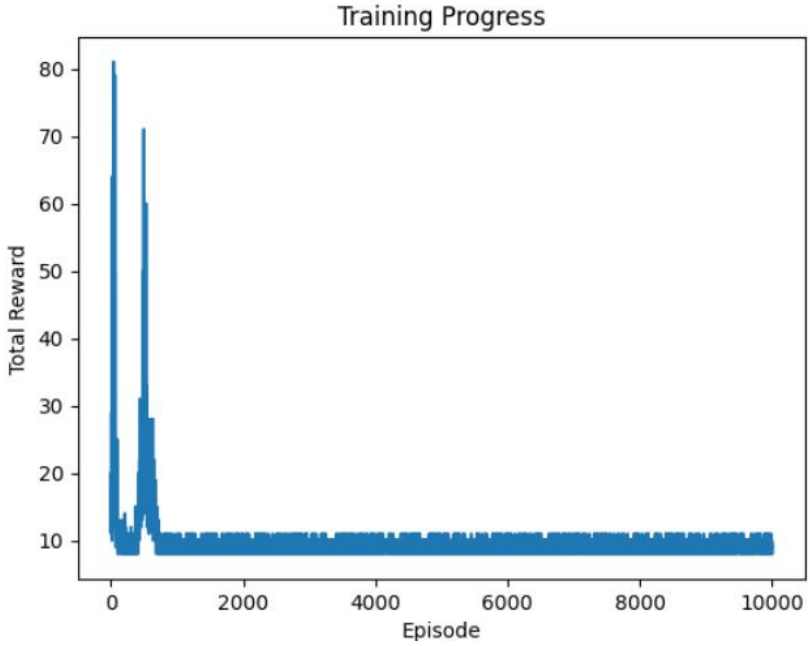

Рисунок 2- Визуализация результатов

Этот код реализует простой алгоритм «Актор-критик» и используется для решения проблемы среды в OpenAI Gym. Алгоритм Актер-Критик — это метод глубокого обучения с подкреплением, основанный на ценностных функциях и стратегиях. Актер отвечает за изучение стратегии, которая поможет агенту совершать действия в окружающей среде, а Критик отвечает за оценку качества этой стратегии, т.е. , функция значения.

В коде мы сначала определяем класс ActorCritic, который содержит общий полносвязный уровень для извлечения функций состояния, а также полносвязные слои, подключенные к Actor и

Critic соответственно. Сеть актеров выводит вероятность действия, а сеть критиков — значение состояния.

Затем мы определяем поезд функции обучения, который принимает в качестве параметров имя среды, общее количество эпизодов обучения, коэффициент дисконтирования гамма и скорость обучения lr. В цикле обучения мы используем состояние окружающей среды в качестве входных данных, а затем используем сети актеров и критиков для выбора действий и оценки значения состояния. На основе распределения вероятностей действия мы используем категориальное распределение для выборки действия. Затем мы выполняем это действие и наблюдаем обратную связь от окружающей среды, включая следующее состояние и награду. Затем мы вычисляем потери Актера и Критика и обновляем параметры модели с помощью обратного распространения ошибки. Во время обучения мы записываем общую награду за каждый эпизод и печатаем общую награду за каждые 10 эпизодов.

Наконец, мы визуализируем ход обучения, строя график общей кривой вознаграждения для всех эпизодов. Это может помочь нам наблюдать эффект обучения агента в окружающей среде, а также оценивать и корректировать процесс обучения.