

Домашнее задание 1: Выбор задачи и подготовка среды

Зимин Евгений Евгеньевич

1. Выбор задачи

Задача:

Мы выбрали задачу обучения агента игре **Breakout**. Эта игра является частью библиотеки Atari и подходит для методов обучения с подкреплением, поскольку обладает следующими особенностями:

- Набор состояний и действий достаточно прост для начального уровня реализации.
- Игра имеет четкую структуру наград, что упрощает формализацию целевой функции.
- Прогресс агента легко оценивать через счет, а также визуально наблюдать процесс обучения.

Цели:

- Обучить агента эффективно управлять платформой для максимального набора очков.
- Протестировать алгоритм **Double Deep Q-Network (DDQN)** для стабильного обучения агента.

Почему RL подходит:

- Breakout является задачей с частичной наблюдаемостью, где решения агента зависят от текущего состояния среды.
- RL позволяет обучить агента стратегическим действиям, таким как целенаправленный разрыв стен, с учетом долгосрочных выгод.
- Использование обучающих стратегий, таких как ϵ -жадная политика и глубокие нейронные сети, эффективно для обучения оптимальных действий.

2. Подготовка среды

Установка программного обеспечения:

Для подготовки среды использовались:

- **Gymnasium**: библиотека для работы с различными RL-средами.
- **PyTorch**: фреймворк для построения и обучения нейронных сетей.
- **ALE-Py**: библиотека для взаимодействия с Atari-средами.

Описание структуры среды ([ссылка](#)):

1. Состояния (Observations):

- Среда предоставляет изображение игры в формате rgb (размерность: (210, 160, 3)), которое агент использует для принятия решений.
 - Для упрощения и увеличения скорости обучения применяется предобработка: преобразование изображения в градации серого, обрезка и изменение размерности до (84, 84).
2. **Действия (Actions):**
- Пространство действий дискретно (Discrete(4)):
 - 0: Ничего не делать (NOOP).
 - 1: Запустить мяч (FIRE).
 - 2: Двигать платформу вправо (RIGHT).
 - 3: Двигать платформу влево (LEFT).
3. **Награды (Rewards):**
- Агент получает положительную награду за разрушение блоков. Размер награды зависит от цвета блока:
 - Верхние блоки дают больше очков, чем нижние.
 - Агент теряет жизнь при пропуске мяча.
4. **Заключительные состояния (Terminal states):**
- Игра завершается, когда агент теряет все жизни.

Тестирование:

Был подготовлен тестовый скрипт, который:

- Инициализирует среду Breakout.
- Выполняет серию случайных действий.
- Выводит наблюдения (изображения) для проверки корректности среды.
- Отслеживает основные метрики (наград, количество шагов, статус игры).

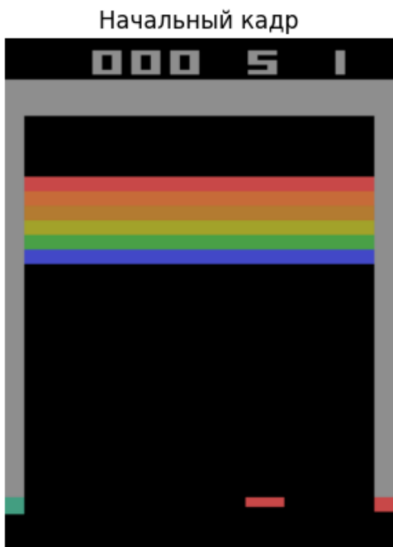
3. Результаты

Описание:

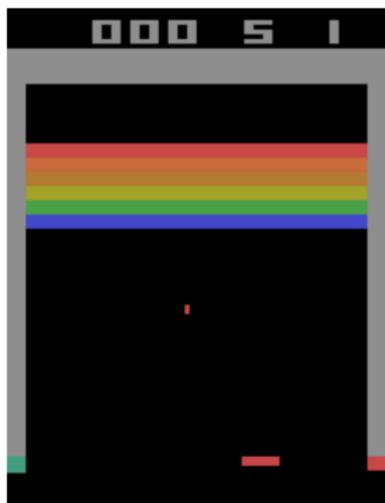
- Мы протестировали взаимодействие со средой, визуализировали наблюдения и убедились в корректной работе преобразования входных данных.
- Подготовили основу для запуска DDQN-агента.

Скриншот среды:

Начальное состояние: (210, 160, 3)
Информация о среде: {'lives': 5, 'episode_frame_number': 0, 'frame_number': 0}



Действие: 1, Награда: 0.0, Завершено: False



Приложение: Код настройки среды и тестовый скрипт

Установка и тестирование среды:

```
# Загрузка среды
env = gym.make('ALE/Breakout-v5', frameskip=4, repeat_action_probability=0.25,
render_mode='rgb_array')
state, info = env.reset()

# Проверка начального состояния
print("Начальное состояние:", state.shape)
print("Информация о среде:", info)

# Визуализация начального кадра
plt.imshow(env.render())
plt.title("Начальный кадр")
plt.axis("off")
plt.show()

# Тестирование выполнения нескольких действий
for _ in range(5):
    action = env.action_space.sample() # Случайное действие
    state, reward, terminated, truncated, info = env.step(action)
    print(f"Действие: {action}, Награда: {reward}, Завершено: {terminated or
truncated}")
    plt.imshow(env.render())
    plt.axis("off")
    plt.show()

env.close()
```