

Обучение нейросети проходить фарватер в Unity

Радионов Сергей Владимирович

Приамурский государственный университет им. Шолом-Алейхема

студент

Баженов Руслан Иванович

Приамурский государственный университет им. Шолом-Алейхема

к.п.н., доцент, зав. кафедрой информационных систем, математика и методик обучения

Аннотация

В данной статье рассмотрена проблема обучения нейронной сети обходить препятствия, управляя виртуальной моделью корабля. Изучены известные решения проблемы и программное обеспечение, необходимое для её решения. Решение осуществлено с помощью игрового движка Unity.

Ключевые слова: нейросеть, искусственный интеллект, машинное обучение.

Training of neural networks to pass the channel in Unity

Radionov Sergei Vladimirovich

Sholom-Aleichem Priamursky State University

Student

Bazhenov Ruslan Ivanovich

Sholom-Aleichem Priamursky State University

Candidate of pedagogical sciences, associate professor, Head of the Department of Information Systems, Mathematics and teaching method

Abstract

This article discusses the problem of learning a neural network to bypass obstacles by controlling the virtual model of the ship. Known solutions of the problem and software needed to solve it were studied. The solution is implemented using the game engine Unity.

Keywords: neural network, artificial intelligence, machine learning.

В наше время нейросети получили очень широкое распространение, хоть мы этого и не замечаем. Например, нейросети уже используются в Google Translate для более качественного перевода текста. Также нейросети широко используются для распознавания разных объектов на изображениях. Уже существуют нейросети, способные управлять автомобилем. Такие сети разрабатывают и используют такие крупные компании как Google и Tesla.

На данный момент наиболее эффективные автопилоты используют нейросети, поэтому их необходимо изучать. С каждым годом транспортных средств с использованием автопилотов будет только расти, так как это значительно упростит жизнь людям.

Проблему использования нейросетей изучали многие российские и зарубежные ученые. В.М.Крыжановский и А.Л.Микаэлян изучали Векторную параметрическую сеть для распознавания образов [1]. Ю.В.Чернухин и А.А.Приемко исследовали трехмерную нейронную сеть адаптивного подводного робота в 2006 году [2]. В 2016 году Я.Бурьлин проводил исследование нейросетевых методов прогнозирования движения судна [3], а А.С.Барановская изучала нейропроцессорные технологии [4]. Можно выделить и другие исследования разных годов [5-7]. В англоязычном сегменте изучения нейросетей также существует множество исследований, так или иначе связанных с нейросетями [8-9].

Целью данного исследования является обучение нейросети обходить препятствия, управляя виртуальной моделью корабля, двигающегося с определенной скоростью.

Недавно разработчиками игрового движка Unity была предоставлена возможность использовать движок и уже готовый алгоритм обучения, для создания собственной нейронной сети Unity Machine Learning Agents. Все что необходимо, это запрограммировать сбор информации обучаемым виртуальным объектом, необходимой для выполнения поставленной задачи, которая будет подаваться в нейросеть как входные параметры, а также реализовать действия объекта при той или иной выходной информации нейросети. Помимо этого, необходимо обозначить какие действия являются ошибкой, давая за них отрицательную награду, а какие правильными с положительной наградой, чтобы алгоритм обучения мог понять в каком направлении двигаться.

Для начала в Unity было создано поле с помощью объекта Terrain для обучения нейросети и упрощенная модель корабля. Далее запрограммирован сбор данных. Набор входных данных представляет собой расстояние до препятствия в пяти направлениях, которые показаны зелеными линиями (рис. 1).

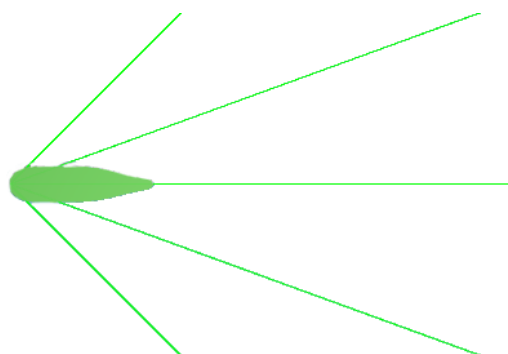


Рис. 1 Модель корабля

Данные о расстоянии нормализованы и лежат в диапазоне от 0 до 1. Количество выходных значений нейросети задано одним. Это значение ограничено диапазоном $[-1; 1]$ и является множителем для максимального угла поворота. Например, выходное значение -2 , будет приравнено к -1 , а значение 0.5 означает, что скорость поворота вправо должна равняться 50% от максимальной. Далее было задано, что при столкновении награда равна -1 , а за каждый шаг без поворота 0.1 , что обеспечит стабильное обучение и стремление нейросети сохранять прямой курс как можно дольше. На поле были расставлены точки, из которых случайным образом будет выбираться начальная позиция корабля после столкновения, чтобы нейронная сеть не обучилась одному правильному маршруту. Также было ограничено «время жизни» корабля, чтобы ускорить процесс обучения.

Скомпилировав проект и настроив среду Python, на которой написан алгоритм обучения и сама нейросеть, можно приступить к обучению кораблей.

Запускается оно с помощью скрипта, через консоль Windows. При запуске указываются параметры обучения, а также параметры нейросети, такие как количество нейронов. В нашем случае для стабильного обучения потребовалось указать название скомпилированной программы и изменить лишь параметр `learning-rate` на $3e-5$, что меньше стандартного $3e-4$. При запуске обучения (Рис.2) остальные параметры остались по умолчанию.

```
PS D:\work\ml-agents-master\python> python ppo.py ShipUnlearned --train --learning-rate 3e-5
{'--batch-size': '64',
 '--beta': '2.5e-3',
 '--buffer-size': '2048',
 '--curriculum': 'None',
 '--epsilon': '0.2',
 '--gamma': '0.99',
 '--help': False,
 '--hidden-units': '64',
 '--keep-checkpoints': '5',
 '--lambd': '0.95',
 '--learning-rate': '3e-5',
 '--load': False,
 '--max-steps': '1e6',
 '--normalize': False,
 '--num-epoch': '5',
 '--num-layers': '2',
 '--run-path': 'ppo',
 '--save-freq': '50000',
 '--summary-freq': '10000',
 '--time-horizon': '2048',
 '--train': True,
 '--worker-id': '0',
 '<env>': 'ShipUnlearned'}
INFO:unityagents:
'ShipAcademy' started successfully!
Unity Academy name: ShipAcademy
  Number of brains: 1
  Reset Parameters :

Unity brain name: Brain
  Number of observations (per agent): 0
  State space type: continuous
  State space size (per agent): 5
  Action space type: continuous
  Action space size (per agent): 1
  Memory space size (per agent): 0
```

Рис. 2 Запуск обучения из консоли

Алгоритм обучения тут же запустил программу (Рис.3) и приступил к обучению. Для ускорения обучения количество обучаемый агентов (кораблей) увеличено до шести, они невидимы друг для друга, а также ускорено время в 100 раз. Для наглядности на месте столкновения создается красная точка, которых может быть не более 5 шт. для каждого корабля.

В процессе обучения, каждые 10000 решений нейронной сети (Step) отображается статистика, где указывается средняя награда (Mean Reward). На начало обучения она не превышала единицы (Рис.4). А после конца обучения, которое было ограничено одним миллионом решений, средняя награда равнялась примерно 21 (Рис.5). Обучение заняло 25 минут, за это время корабли научились обходить часть препятствий, но все равно в некоторых местах испытывали трудности. Далее увеличил время обучения в два раза, средняя награда повысилась до 40, на практике же корабли практически перестали сталкиваться с препятствиями.

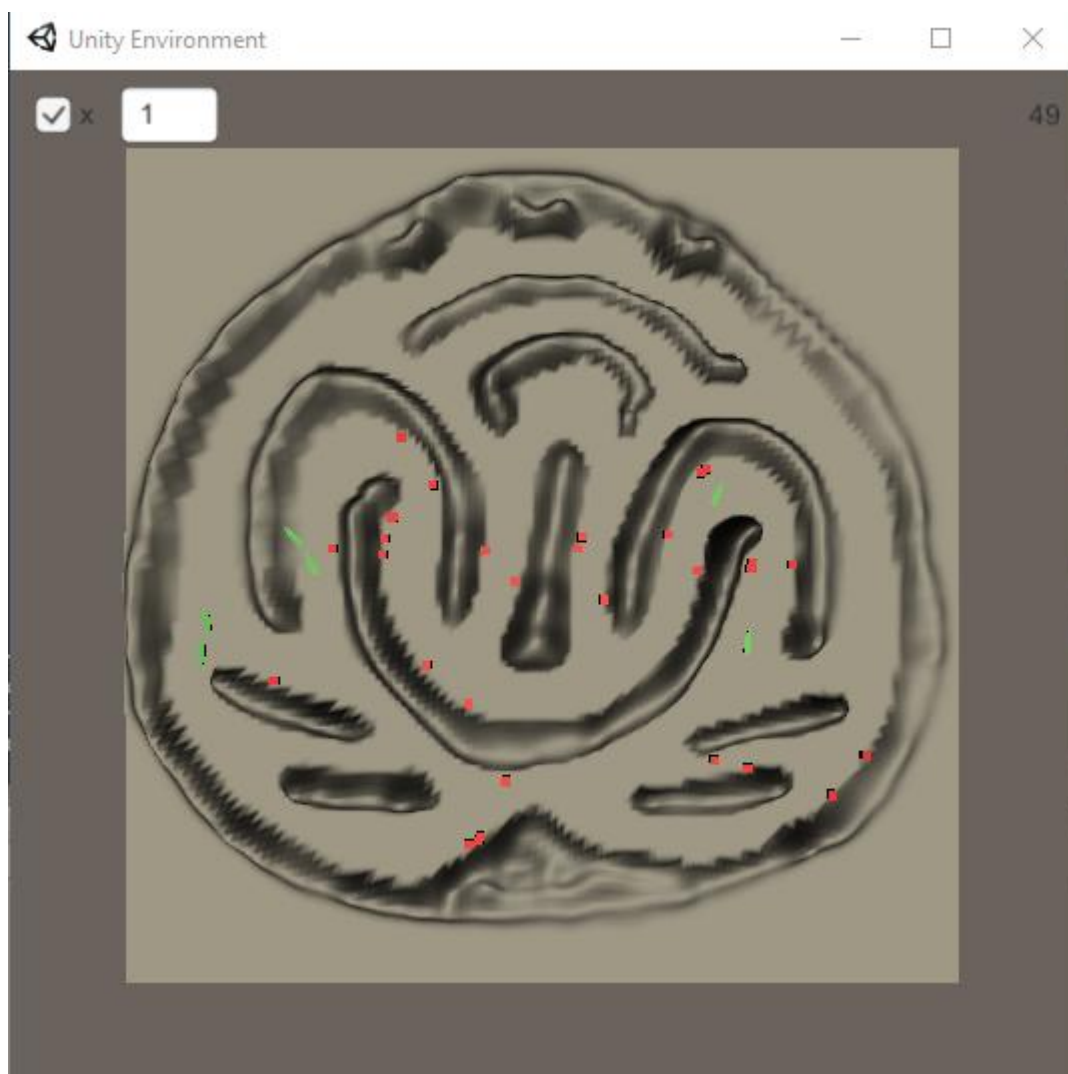


Рис. 3 Обучение нейросети

```
Step: 10000. Mean Reward: 0.11934426229508208. Std of Reward: 0.690772598705838.  
Step: 20000. Mean Reward: 0.7340937896070977. Std of Reward: 1.31735653281468.  
Step: 30000. Mean Reward: 0.7892037786774629. Std of Reward: 1.3521580842453433.  
Step: 40000. Mean Reward: 0.8716312056737588. Std of Reward: 1.4533661705478318.  
Step: 50000. Mean Reward: 0.9525147928994082. Std of Reward: 1.509920706435273.
```

Рис.4 Статистика на начало обучения

```
Step: 960000. Mean Reward: 20.288135593220467. Std of Reward: 18.03833098320313.  
Step: 970000. Mean Reward: 20.20769230769244. Std of Reward: 18.53413184890273.  
Step: 980000. Mean Reward: 22.955045871559786. Std of Reward: 18.607986281117814.  
Step: 990000. Mean Reward: 17.284210526315885. Std of Reward: 17.13772974368873.  
Step: 1000000. Mean Reward: 20.966956521739263. Std of Reward: 18.25853330810919.
```

Рис.5 Статистика на конец обучения

Таким образом, можно обучать нейронную сеть на виртуальных объектах для последующего использования на реальных, если полностью симитировать показания сенсоров и окружающую обстановку.

Проводя исследование, были сделаны следующие шаги:

1. Изучение документации;
2. Создание модели корабля и фарватера;
3. Настройка входных и выходных среды;
4. Обучение с помощью нейронной сети в Unity.

Библиографический список

1. Крыжановский В.М., Микаэлян А.Л. Векторная параметрическая нейросеть для распознавания бинарных образов // Известия ЮФУ. Технические науки. 2005. №10. С. 90-98.
2. Чернухин Ю.В., Приемко А.А. Трехмерная нейросеть адаптивного подводного робота // Известия ЮФУ. Технические науки. 2006. №5. С. 100-103.
3. Бурылин Я. Нейросетевые методы прогнозирования движения судна // Эксплуатация морского транспорта. 2016. №4. С. 62-67.
4. Барановская А.С. Немного о нейропроцессорных технологиях // Инновационные технологии в машиностроении, образовании и экономике. 2016. №1. С. 25-28.
5. Юдинцев Б.С., Даринцев О.В. Модификация нейросетевой системы планирования траекторий: методики и результаты // Фундаментальные исследования. 2014. №11-12. С. 2630-2635.
6. Кравец П.И., Шимкович В.М., Николин О.И. Нейросетевая система машинного зрения с аппаратно-программной реализацией на плис // Молодий вчений. 2015. №5-11. С. 47-50.
7. Акимов Е.Н., Балык В.М., Маленков А.А., Станченко А.С. Нейросетевое моделирование режимов движения высокоскоростного подводного аппарата // Наукоемкие технологии. 2016. №6. С. 48-52.
8. Huang G. B., Zhu Q. Y., Siew C. K. Extreme learning machine: a new learning scheme of feedforward neural networks //Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on. IEEE, 2004. T. 2. С. 985-990.

9. Sutskever I., Vinyals O., Le Q. V. Sequence to sequence learning with neural networks //Advances in neural information processing systems. 2014. С. 3104-3112.