

ОБУЧЕНИЕ ИИ ИГРЕ FLAPPY BIRD С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМА НЕЙРОЭВОЛЮЦИИ НАРАСТАЮЩИХ ТОПОЛОГИЙ

Бруй Н.М., Чернышенко М.С.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Научный руководитель: Ролич О.Ч. – канд. техн. наук

Аннотация. Для демонстрации алгоритма нейроэволюции нарастающих топологий NEAT на языке программирования Python написана игра Flappy Bird. Рассмотрена технология обучения нейронных сетей с использованием данного алгоритма, а также преимущества внедрения её в машинное обучение.

Ключевые слова: NEAT, нейронная сеть, историческое происхождение, гены, виды, Python, Flappy Bird

Введение. Развитие искусственных нейронных сетей с помощью генетических алгоритмов широко используется в способе машинного обучения с подкреплением. Важным фактором в нейроэволюции является извлечение преимущества от развития топологий нейронных сетей наряду с весами [1]. В данной статье рассматривается алгоритм нейроэволюции нарастающих топологий, который является особенно эффективным вследствие 3 ключевых подходов:

1. Осознанное скрещивание генов в результате использования исторического происхождения.
2. Защита от исчезновения значимых генов в процессе отбора в результате разделения генов на отдельные виды.
3. Старт с минимального количества связей и увеличение их числа только в случае необходимости.

Алгоритм NEAT создан специально для реализации вышеперечисленных принципов. NEAT (алгоритм нейроэволюции нарастающих топологий) это эволюционный алгоритм, который создаёт искусственные нейронные сети [2]. В реализации технологии развивается популяция отдельных геномов, содержащих два вида генов, которые описывают построение искусственной нейронной сети:

1. Узловые гены, каждый из которых представляет отдельный нейрон;
2. Гены связи, каждый из которых определяет единственную связь между нейронами.

Чтобы разработать решение задачи, пользователь должен предусмотреть фитнес-функцию (функцию приспособленности), которая вычисляет вещественное число, указывающее на качество отдельного генома: лучшая способность решать задачу означает более высокий балл. Алгоритм выполняется с заданным пользователем количеством поколений путём скрещивания и мутации наиболее подходящих особей предыдущего поколения.

Операции скрещивания и мутации могут добавлять узлы и/или связи в геномы, поэтому геномы расширяются в процессе развития, а нейронная сеть становится более сложной. Когда образуется заданное количество поколений, или когда хотя бы одна особь достигает определённого пользователем показателя приспособленности, алгоритм прекращается.

Для осуществления механизма скрещивания система запрограммирована понимать, что гены с одним историческим предком представляют одинаковую структуру (хотя, возможно, с разными весами) и совпадают, так как они были получены из одного наследственного гена в прошлом [1].

Отслеживание исторического происхождения требует небольшого количества расчётов. NEAT справляется с этим, отслеживая происхождение узлов с помощью идентификационного номера, представленного вещественным числом. Когда в ходе мутации появляется новый ген, ему присваивается увеличенный на единицу идентификационный номер, который таким образом отражает хронологию возникновения каждого гена в системе. Те гены, которые происходят от

общего предка, сопоставляются для скрещивания, и происходит соединение узлов с общим происхождением [1].

Опасность мутации заключается в возможности разрушить перспективные в будущем гены, поэтому, основываясь на историческом происхождении геномов, NEAT разделяет их на отдельные виды. Количество лишних и непересекающихся генов является мерой сопоставимости. Чем более непересекаемы геномы, тем меньше эволюционной истории они делят, тем менее они сопоставимы. Если сопоставимость генов не достигает определённой границы, они относятся к разным видам и не конкурируют между собой. Самые успешные особи каждого вида в будущем скрещиваются, что в итоге приводит к замене всей популяции и получению нужного результата на выходном слое.

Основная часть. Целью проекта является использование алгоритма нейроэволюции нарастающих топологий для обучения искусственного интеллекта проходить игру Flappy Bird. Для осуществления поставленной задачи было выделено 2 подзадачи:

1. Написание игры Flappy Bird на языке программирования Python;
2. Внедрение алгоритма NEAT для обучения ИИ прохождению игры.

С помощью среды разработки PyCharm была создана игра Flappy Bird. Это игра жанра аркада, в которой птица должна пролетать через просвет между трубами, возникающими парами сверху и снизу экрана на случайно установленной высоте. Птица перемещается вдоль положительного направления оси X, а игрок задаёт её прыжок путём нажатия на экран. Игра заканчивается только в том случае, если птица касается трубы, верхнего или нижнего края экрана.

Алгоритм нейроэволюции нарастающих топологий развивает нейронную сеть. На входе алгоритм принимает начальные значения узлов, в процессе обучения случайным образом добавляет новые узлы и образует дополнительные связи между ними, в результате чего формирует на выходе наиболее подходящую для решения задачи топологию нейронной сети. Особенностью NEAT является то, что он начинает работу с упрощённой моделью и усложняет её только в случае необходимости.

Для обучения искусственного интеллекта в код программного модуля был внедрён файл конфигурации, задающий все основные настройки. Модель обучалась на основе нейронной сети прямого распространения. В качестве входных значений передавались 3 узла, определяющие позицию птицы по оси Y, дистанцию между птицей и верхней трубой и между птицей и нижней трубой. Альтернативно можно было передавать только данные о позиции птицы и дистанцию между одной из двух труб, однако алгоритму каждый раз нужно было бы высчитывать неизменное значение просвета, что замедлило бы процесс обучения. Для оценки полученного на выходе числа нужна функция активации, в роли которой была использована функция гиперболического тангенса, она приводила все выходные значения в промежуток от -1 до 1 и выдавала на входной узел решение «прыжок» в том случае, если значение было больше 0.5. Размер популяции устанавливался в 20 птиц для создания каждым поколением, что можно увидеть на рисунке 1.



Рисунок 1 – Запуск программы обучения нейросети с 20 особями

Так как фитнес-функция определяет пригодность особи, она приравнивалась к пройденной дистанции и добавляла 1 балл за каждую секунду игрового процесса, за получение успешно натренированной модели и достижение поставленной цели было принято пороговое значение баллов для остановки программы – 30 очков. В случае обучения до 30 поколений без достижения заданного числа очков игровой процесс останавливался без получения нужного результата.

```
flappy_bird

***** Running generation 0 *****

Population's average fitness: 3.25000 stdev: 1.17324
Best fitness: 6.60000 - size: (1, 3) - species 1 - id 6
Average adjusted fitness: 0.221
Mean genetic distance 0.894, standard deviation 0.399
Population of 20 members in 1 species:
  ID age size fitness adj fit stag
  --- -- --- -
  1  0  20   6.6  0.221  0
Total extinctions: 0
Generation time: 2.832 sec

***** Running generation 1 *****

Population's average fitness: 5.58000 stdev: 5.87313
Best fitness: 30.60000 - size: (1, 3) - species 1 - id 38
Average adjusted fitness: 0.116
Mean genetic distance 1.306, standard deviation 0.468
Population of 20 members in 1 species:
  ID age size fitness adj fit stag
  --- -- --- -
  1  1  20  30.6  0.116  0
Total extinctions: 0
Generation time: 9.168 sec (6.000 average)
```

Рисунок 2 – Статистические данные обучения нейронной сети по алгоритму NEAT

Все вышеперечисленные значения были подобраны в ходе 23 запусков, в итоге удалось достичь результата успешного обучения модели в 2 поколения за 9.168 секунд, что продемонстрировано на рисунке 2. В результате осуществим запуск бесконечного цикла прохождения птиц разрывов между трубами.

Заключение. Для демонстрации работы алгоритма NEAT на языке программирования Python была написана игра Flappy Bird, прохождения которой было реализовано с помощью обученной нейронной сети. Таким образом было доказано, что NEAT является эффективным методом развития нейронных сетей, который через одновременную оптимизацию и усложнение сети усиливает аналогию между генетическим алгоритмом и естественной эволюцией.

Список литературы

1. Kenneth O. Stanley and Risto Miikkulainen. *Efficient Evolution of Neural Network Topologies*
2. *Общее представление о NEAT [Электронный ресурс]. – Режим доступа: NEAT Overview – NEAT-Python 0.92 documentation (neat-python.readthedocs.io) – Дата доступа: 16.03.2021.*

UDC 004.8

TRAINING AI TO PLAY FLAPPY BIRD USING NEUROEVOLUTION OF AUGMENTING TOPOLOGIES

Bruj N.M., Tchernyshenko M.S.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus (style T-institution)

Rolich O.C. – PhD in Technology

Annotation. The game Flappy Bird is written using the Python programming language to demonstrate the NeuroEvolution of Augmenting Topologies algorithm NEAT. The technology of training neural networks using this algorithm is discussed, as well as the advantages of implementing it in machine learning.

Keywords. NEAT, neural network, historical markings, genes, speciation, Python, Flappy Bird