

ИСПОЛЬЗОВАНИЕ НЕЙРОННОЙ СЕТИ ДЛЯ ОБУЧЕНИЯ ИГРОВОГО ПЕРСОНАЖА

Рассматривается реализация обучения игрового персонажа с помощью нейронной сети на основе экспериментов кибер-биолога Михаила Царькова.

ВВЕДЕНИЕ

Суть задачи по обучению игрового персонажа такова: по полю движется змейка, которая уменьшается, если не находит еду, либо увеличивается, если находит. Если змейка врежется в препятствие, то она погибает. Я буду обучать бота, который управляет змейкой. Если змейка разделится, то бот получит в управление ещё одну змейку, которая в свою очередь тоже может разделиться. За основу взят эксперимент со змейками кибер-биолога Михаила Царькова.

I. ОБУЧЕНИЕ БОТА

Сперва был создан бот-рандомайзер. Так я называю бота, который ходит случайным образом. Он необходим для проверки эффективности нейронной сети. При правильном обучении нейросеть должна легко его обыгрывать. Бот получает на вход поле размером 9x9 клеток, в середине которого находится голова змейки. Соответственно наша нейронная сеть будет иметь 81 вход. Порядок расположения клеток, подаваемых на вход, не имеет значения. Сеть при обучении «сама разберется», где что находится. Для обозначения препятствий и других змеек я использовал значения от -1 до 0 (не включительно). Пустые клетки обозначались значением 0.01, а еда 0.99. На выходе нейросети использовалось 5 нейронов для действий:

двигаться влево по оси X; вправо; вверх по оси Y; вниз; делиться пополам.

Движение бота определял нейрон, который имеет самое большое значение на выходе.

II. А/Б ТЕСТИРОВАНИЕ

Для выполнения этой задачи была создана программа, которая делит змеек на две части в зависимости от конфигурации нейронной сети. На поле выпускалось по 20 змеек каждой конфигурации.

Кобяк Валерий Сергеевич, магистрант кафедры информационных технологий автоматизированных систем БГУИР, valerakobyak@icloud.com.

Научный руководитель: Навроцкий Анатолий Александрович, заведующий кафедрой кандидат физико-математических наук, доцент, navrotsky@bsuir.by

Все змейки, находящиеся под управлением одного бота, имели одну и ту же нейронную сеть. Чем больше было змеек в его управлении и чем чаще они сталкивались с разными задачами, тем быстрее проходило обучение. Если, например, одна змейка научилась избегать тупиков или делиться пополам, попав в тупик, то автоматически все змейки данного бота, приобрели эти умения. Меняя конфигурацию нейронной сети можно получить хорошие результаты, но этого не достаточно. Для дальнейшего улучшения алгоритма я решил использовать память на несколько ходов.

Для каждого бота я создал память на 8 ходов. В память было записано состояние поля и ход, который предлагал бот. После этого я проводил корректировку весов для всех восьми состояний, которые предшествовали ходу. Для этого я использовал единый коэффициент корректировки, независимый от глубины хода. Таким образом, каждый ход приводил к корректировке весов не один раз, а восемь. Как и ожидалось, боты с памятью начали быстро обыгрывать ботов, которые обучались без использования памяти.

III. ВЫВОДЫ

Предлагаемая мною модификация нейронной сети позволяет быстрее обучить игрового персонажа. Чем больше подконтрольных элементов в сети и чем чаще они сталкивались с разными задачами, тем быстрее проходило обучение нейронной сети. Но есть добавить использование памяти, то бот будет более "умным чем бот, который существует без использования памяти.

1. Сереган, Т. Programming Collective Intelligence: Building Smart Web 2.0 Applications / Т. Сереган // 2011. – №2. – С. 126-2301.