Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Национальный исследовательский университет

“Высшая школа экономики”»

Жалкова Наталия Евгеньевна

**Разработка интеллектуальных систем на основе нейросетевых конструкций: интеллектуальная система искусственного интеллекта в настольной игре**

Отчет по нейронной сети студента

2 курса бакалавриата группы НТ-6

Москва 2017

Постановка задачи

В качестве настольной игры выбрать игру “шашки” или “проверку характеристик” в играх типа Dungeons & Dragons. В ходе выполнения работы необходимо проанализировать алгоритмы выбранной настольной игры, определить и создать выборку, требуемую для обучения нейронной сети, разработать систему, способную производить решения по правилам настольной игры, имея входные параметры пользователя.

В случае игры “шашки” входным параметром будет матрица, содержащая пустые позиции и позиции шашек обеих сторон, по которой нейронная сеть будет принимать решения о своем следующем ходе.

При выборе темы “проверка характеристик”, нейросеть должна, опираясь на полученные значения броска кубика, типа кубика и характеристик игрока, определить, был ли бросок выигрышным или нет.

Содержание работы: постановка задачи, введение, теоретический материал, анализ решения, разработка ПО, тестирование ПО, заключение, список литературы.

Введение

Для выполнения своего домашнего задания я выбрала нейропакет Нейросимулятор v 1.0, так как считаю, что он позволяет легко и точно моделировать необходимую сеть, получая графики процесса обучения, а так же имея возможность быстро проводить проверку нейросети на новой выборке, округлять выходные значения. Нейросимулятор v 1.0 – это программа конструирования нейронных сетей с последующим обучением и тестированием. Одним из плюсов данной программы является также и то, что кроме сигмоидного преобразователя можно выбирать и другие: линейный, ступенчатый и логарифмический, а также варьировать коэффициенты этих функций. Все это позволяет построить наиболее оптимальную сеть для требуемого задания.

Я выбрала тему создания интеллектуальной системы искусственного интеллекта в “броске атаки” игр типа D&D из-за того, что данная настольная игра очень популярна во многих странах, соответственно разработка нейронной сети является довольно востребованной. С помощью внедрения нейросети станет намного проще проводить проверки, заменяя проведение ее игровым мастером. Кроме того, на данный момент отсутствуют программные обеспечения, производящие “проверку характеристик”.

Теоретический материал

**Dungeons & Dragons** (D&D, DnD; Подземелья и Драконы) — настольная ролевая игра в жанре фэнтези, разработанная Гэри Гайгэксом и Дэйвом Арнесоном. Она впервые была издана в 1974 году компанией “Tactical Studies Rules, Inc”. С 1997 года D&D издаётся компанией “Wizards of the Coast” [1].

Ведущий (**Dungeon Master**) является арбитром и рассказчиком истории. Игроки создают группу из нескольких персонажей, которая взаимодействует с окружающим миром, разрешает различные конфликты, путешествует по землям придуманной вселенной, участвует в сражениях, получает награды и опыт.

В игре участвуют один ведущий (так называемый “мастер”) и несколько игроков, число которых варьируется в зависимости от редакции и пожеланий участников. Как правило, один игрок руководит в игровом мире действиями только одного, своего, персонажа. Мастер действует от лица всех неигровых персонажей, описывает окружающую среду и происходящие в ней события, давая всю необходимую информацию игрокам.

В течение игры каждый игрок задаёт действия для своего персонажа, ведет разговоры от его лица, принимает решения, а результаты действий определяются мастером в соответствии с правилами, описанными в специальных книгах. Случайные события моделируются броском кубика (для разных событий и в зависимости от сложности, кидаются разные типы кубиков). Традиционно руководство игры или свод правил включает в себя три книги: «Player’s Handbook», «Dungeon Master’s Guide» и «Monster Manual». Также существуют различные дополнения, которые мастер может использовать по своему усмотрению. На данный момент существует 5 редакций правил D&D.

**Проверка (чек)** – это тест, позволяющий определить успешность заявленного действия, с использованием или без талантов игрока. В большинстве случаев необходимо делать проверки, если Мастер решает, что заявленное игроком действие имеет определенный шанс провала. Результат является неопределенным, и Мастер обращается к кубикам, чтобы определить судьбу персонажа игрока [2].

Чтобы сделать проверку, первым делом нужно найти соответствующую характеристику в Листе Персонажа. Каждая характеристика имеет свой модификатор.

1. **Бросок кубика**: Бросок d20 (20-гранного кубика) и добавление модификатора соответствующей характеристики.
2. **Бонусы и штрафы**: Если класс, навык, заклинание или что-то ещё дает игроку бонусы или штрафы к текущей проверке, необходимо применить их к полученному результату.
3. **Оглашение**: Сообщение Мастеру полученного в итоге результата проверки. Определение успеха/неуспеха.

Когда совершается проверка, Мастер определяет Коэффициент\Уровень\Класс Сложности (КС\УС) для заявленного действия. Мастер руководствуется некоторой информацией для определения Класса Сложности. Чем труднее задача, тем выше её КС.

Если результат проверки (значение на костях) выше КС, то действие выполнено успешно. В ином случае, действие провалено.

В своей работе, я выбрала “Бросок атаки” в качестве проверяемой характеристики, так как данная проверка происходит наиболее часто во время игровой сессии.

Бросок атаки

**Бросок атаки (Attack roll)** - применяется для проверки успешности какого-либо атакующего действия (а иногда и заклинания) в игровой системе D&D. [3]

Сначала нужно определить, сколько нужно выкинуть на кубике, чтобы поразить цель. Происходит вычитание из THAC0 (числа, которое нужно пробросить на 20-стороннем кубике, чтобы поразить врага с АС = 0) атакующего AC (броню) цели. Чтобы нанести удар по цели, атакующему нужно выкинуть на двадцатиграннике (d20) число больше полученного. Существуют также некоторые модификаторы (модификатор силы, модификаторы от используемых магических предметов, модификатор оружия), изменяющие окончательные результаты броска. Они применяются к выпавшему на кубике числу.

Модификатор броска атаки по силе необходимо смотреть в соответствующей таблице:

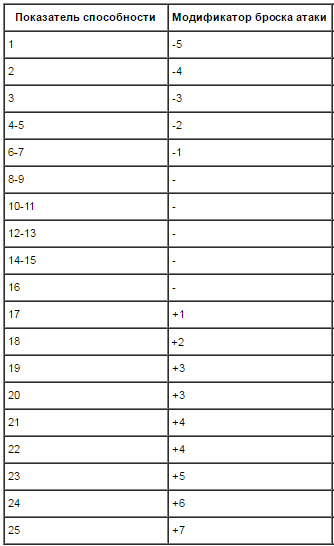


Рисунок 1. Таблица модификатора броска атаки.

**Пример 1**:

Атакующий: воин, THAC0 = 14, сила = 18 (для этого значения модификатор броска атаки = +2), Длинный меч +1.

Защищающийся: гоблин, АС = 5.

Результат: THAC0 (14) - модификатор силы (2) - модификатор от меча (1) - AC (5) = 6.

**Пример 2**:

Атакующий: маг, THAC0 = 18, сила = 5 (для этого значения модификатор броска атаки = -2).

Защищающийся: гоблин, АС = 5.

Результат: THAC0 (18) - модификатор силы (-2) - AC (5) = 15.

Анализ решения

**Формула для вычисления**:

D20 >? THAC0 – МС – М – AC

**Входные данные**:

Результат броска двадцатигранника D20 (от 1 до 20), значение THAC0 (от 1 до 20), значение силы С (от 1 до 25), по которому рассчитывается МС из таблицы, значение модификатора М (от -5 до +5), АС противника (от -15 до 15) – 5 выходных нейронов.

**Выходные данные**:

Успех (1) / неуспех (0) броска – 1 выходной нейрон.

Всего в процессе подготовки к работе было создано 100 выборок для обучения, среди которых 45 неуспешных бросков костей и 55 – успешных, и 10 дополнительных выборок для проверки работы нейросети, среди которых одинаковое число успешных и неуспешных бросков.

Все данные содержатся в табличке “данные.xls”. Имеющиеся параметры будут передаваться в нейросеть как обучающие (страница “Обучение”) и как проверочные (страница “Проверка”). Таким образом, анализ завершен, подходящая для решения задачи нейронная сеть выбрана (части ее структуры). Теперь необходимо описать ее с помощью программы Нейросимулятор v 1.0, получить результаты обучения, проверки, определить возникающие (при наличии) ошибки.

Разработка ПО

Приведу ниже выполненный мной порядок действий:

Прежде всего, необходимо определится со структурой нейронной сети. Число нейронов входного слоя = 5, число скрытых слоев = 1, число нейронов в скрытом слое = 3, число нейронов выходного слоя = 1. Данная структура нейронной сети выбрана после проведения нескольких тестов, после которых вышеописанный вариант оказался наиболее “умным”, а получаемая средняя ошибка - минимальной. При большом количестве скрытых слоев начинается переобучение.

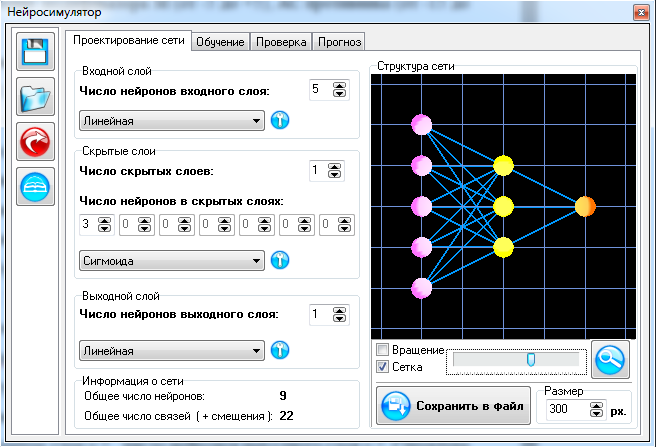


Рисунок 2. Структура нейронной сети.

Далее необходимо загрузить выборку и провести обучение сети. Для этого я нажму на кнопку “Загрузить из EXCEL-файла” и выберу ранее созданный файл выборки. Загрузятся данные о входных и выходных параметрах, необходимых для обучения.

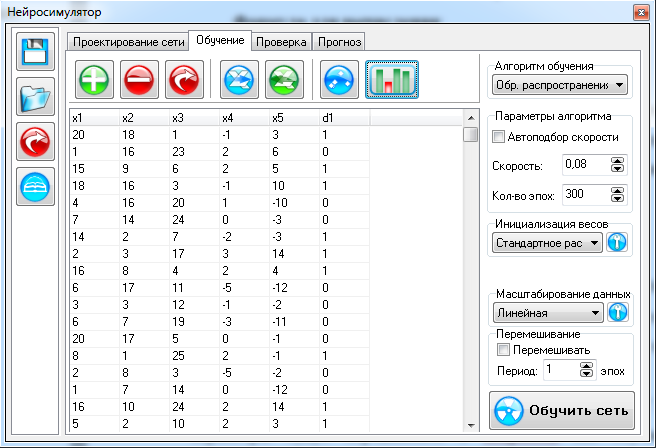


Рисунок 3. Окно обучения нейронной сети.

После обучения нейронной сети, программа демонстрирует окно с двумя графиками. На графиках ниже можно видеть изменения максимальной и средней ошибок с течением эпох, вплоть до 300 эпохи (кол-во эпох, выбранное ранее).

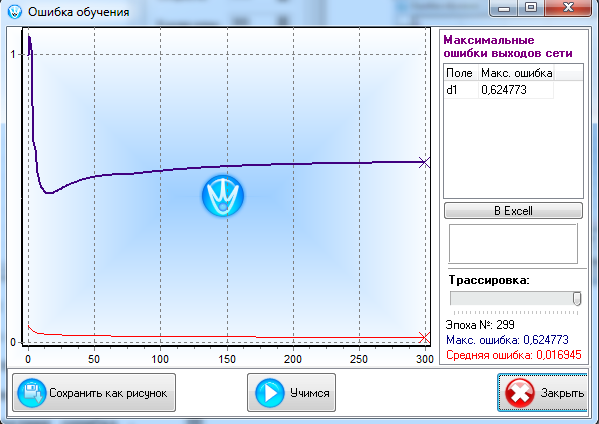


Рисунок 4. Ошибка обучения.

Тестирование ПО

Теперь полученную нейросеть можно протестировать новыми наборами данных, которые не были использованы во время обучения, с помощью функции “Проверка”. Данную операцию стоит проводить для того, чтобы узнать, насколько хорошо нейросеть может определять сама успех броска. Для этого загружу данные из EXCEL-файла и произведу проверку.

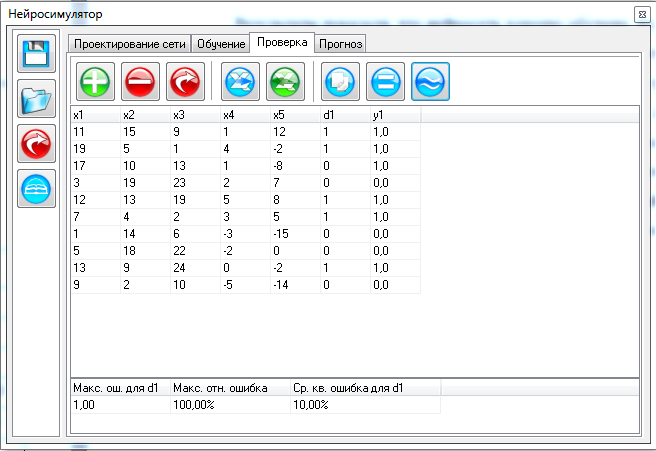


Рисунок 5. Проведение проверки.

Привожу таблицу с проведенными тестами:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Вводимые данные** | | | | | **Ответ нейросети** |
| D20 | THAC0 | C | M | AC | Успех/неуспех |
| 11 | 15 | 9 | 1 | 12 | 1 |
| 19 | 5 | 1 | 4 | -2 | 1 |
| 17 | 10 | 13 | 1 | -8 | 1 |
| 3 | 19 | 23 | 2 | 7 | 0 |
| 12 | 13 | 19 | 5 | 8 | 1 |
| 7 | 4 | 2 | 3 | 5 | 1 |
| 1 | 14 | 6 | -3 | -15 | 0 |
| 5 | 18 | 22 | -2 | 0 | 0 |
| 13 | 9 | 24 | 0 | -2 | 1 |
| 9 | 2 | 10 | -5 | -14 | 0 |

Заключение

В процессе выполнения работы был проанализирован алгоритм “броска силы”, создана нейронная сеть, способная определять успешность броска по введенным данным о состоянии персонажа игрока и характеристике его противника. Результаты последующего тестирования показали, что нейросеть хорошо обучена, хоть и встретилась единственная ошибка. Ее возникновение можно легко объяснить, решив:

THAC0 = 10, C = 13, M = 1, AC = -8.

Для C = 13, МС = 0 (по рис. 1). Следовательно,

THAC0 (10) - модификатор силы (0) - модификатор от меча (1) - AC (-8) = 17. Следовательно, бросок должен быть D20 > 17

Таким образом, ошибка возникла при непосредственной близости значения броска к границе (равенство ей). Подобные ошибки в работе нейросети можно убрать, увеличив объем выборки, добавив туда больше примеров, когда, аналогично появившемуся несоответствию, граница успеха равна значению броска игрока.

Список литературы

Dungeons & Dragons – Википедия [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Dungeons_%26_Dragons> (Дата обращения: 2.06.2017).

Игромеханика, Основы, Dungeon wiki [Электронный ресурс]. – URL: <http://ru.dungeon.wikia.com/wiki/Игромеханика,_Основы> (Дата обращения: 3.06.2017).

Бросок атаки – AERIE wiki [Электронный ресурс]. – URL: <http://wiki.aerie.ru/wiki/Бросок_атаки> (Дата обращения: 2.06.2017).