

УДК 519.233

**ОБНАРУЖЕНИЕ РОБОТОВ В КОМПЬЮТЕРНЫХ ИГРАХ С ИСПОЛЬЗОВАНИЕМ
НЕЙРОННЫХ СЕТЕЙ**Е.А. Задорожных

Научный руководитель: доцент, к.ф.-м.н. М.Е. Семенов

Национальный исследовательский Томский политехнический университет

Россия, г. Томск, пр. Ленина, 30, 634050

E-mail: caz28@tpu.ru**DETECTION OF ROBOTS IN COMPUTER GAMES USING NEURAL NETWORKS**

E.A. Zadorozhnykh

Scientific Supervisor: PhD, Associate Professor M.E. Semenov

Tomsk Polytechnic University, Russia, Tomsk, Lenin avenue, 30, 634050

E-mail: caz28@tpu.ru

***Abstract.** Computer games occupy a huge share in the field of entertainment, so it is important to improve methods to combat cheating. In this paper, we have used the own Targetman game to collect and analyze the data. The neural network model has been proposed and we solved the classification binary task "bot/not-bot" with one hot encoding of outcome variable.*

Введение. Развитие соревновательной составляющей в игровой индустрии привело к появлению игроков, которые стали создавать и использовать специальные программы - *роботы*, позволяющие получить преимущество, недостижимое при честной игре. Автоматизируя игровой процесс, роботы стали уменьшать сложность игры для пользователя. В зависимости от жанра игры роботы выполняют разные функции. В данной работе мы рассматриваем два типа роботов, которые чаще всего используются в играх жанра «стрельба от первого лица» (first-person shooter, FPS-игры): aimbot – предназначен для автоматического прицеливания и triggerbot – автоматического выстрела. Aimbot определяет местоположение других игроков и прочих целей вне зависимости от того, может ли реальный игрок видеть их. При этом робот прицеливается существенно точнее и быстрее, чем игрок. Triggerbot используется для автоматического выстрела при наведении на цель. Данные типы роботов часто используются вместе. При этом поведение игрока становится неестественным, поэтому игроки практикуют мимикрию, например, задержка между прицеливанием и выстрелом или случайное отклонение от центра мишени. Использование таких роботов дает несправедливое преимущество игроку и нарушает баланс игры.

Цель работы – разработка методов обнаружения роботов в компьютерных играх для дальнейшего выявления игроков, пользующихся роботами. Один из способов обнаружения роботов – использование моделей нейронных сетей для решения задачи бинарной классификации: 0 – игрок, 1 – робот. В статье [1] авторы приводят классификацию методов выявления использования роботов в компьютерных онлайн играх, в частности указаны: анализ поведения пользователей, анализ интернет трафика и анализ распознавания капча (captcha). Далее в работе [2] авторы продолжают исследования и описывают

подходы для обнаружения роботов в FPS-играх. Они выделяют методы, основанные на прямом анализе игровых действий пользователя (например, скорость наведения на цель, точность попадания), а также методы, основанные на косвенном анализе игрового процесса (например, время жизни цели, число фальшивых целей). В данной работе мы использовали авторскую игру TargetMan, разработанную в жанре «стрельба от первого лица».

Входные данные. Для обучения и тестирования модели нейронной сети мы использовали набор данных, собранных с помощью игры TargetMan [3]. Набор содержит девять признаков, а также целевой категориальный признак Running, идентифицирующий использование робота (True/False). Собранные данные были нормализованы и отображены на интервал [0,1] по правилу: $(x - \min(x))/(\max(x) - \min(x))$. Приведем описание признаков: Lives – количество оставшихся жизней игрока, Level – текущий уровень в игре, Points – количество очков, набранных игроком на текущий момент, Suspicion – уровень подозрительности игрока, TargetLifeTime – время жизни цели, Missed Shots – количество пропущенных выстрелов, TotalTargets – общее количество целей, TotalFakeTargets – общее количество фальшивых целей, $\exists \text{TotalTargets} \leq \text{TotalFakeTargets}$, TotalApprovedTargets – общее число пораженных целей. Матрица корреляций между независимыми признаками (категориальные признаки Lives, Level, представленные с помощью целых и вещественных значений, не учитывались) показала, переменные Total Targets, Total Fake Targets, Total Approved Targets имеют сильную положительную зависимость, в то время как Points и Target Life Time – сильную отрицательную. В игровом процессе на значения признаков наложены вероятностные условия: $P(\text{Suspicion}=0 \mid \text{Running}=\text{False})=1$, $P(\text{Missed Shots}>0 \mid \text{Running}=\text{False})>0$, $P(\text{TotalApprovedTargets}>0 \mid \text{TotalTargets}=\text{TotalFakeTargets}=0)>0$. Для вычисления значений Suspicion мы предложили использовать рекурсивную функцию от количества пораженных фальшивых целей:

$$f(x) = f(x - 1) + x^2 = x(x+1)(2x+1)/6 = \sum_{i=1}^x i, \quad f(0) = 0, \quad x \in \mathbb{N}$$

$f'(x) = x^2 + x + 1/6 > 0, \forall x \in \mathbb{N}$, следовательно, подозрительность игрока возрастает с ростом количества пораженных целей.

Набор данных для обучения нейронной сети содержит 5 249 записей, среди которых в 1 518 записях Running=False, в 3 731 – Running=True (отношение классов 0,4). В наборе для тестирования находится 515 записей, из которых в 139 записях Running=False, в 376 Running=True (отношение классов 0,36). На рис. 1 приведена структура нейронной сети, где параметры x_1, x_2, \dots, x_9 формируют входной слой и соответствуют девяти вышеописанным признакам. Сеть состоит из трёх скрытых слоёв, в каждом из которых находится три нейрона $N_{ij}, i, j = 1, 2, 3$. Выходной слой связан с признаком Running и содержит два нейрона: $N_{4,1}$ соответствует Running=True, а нейрон $N_{4,2}$ – Running=False.

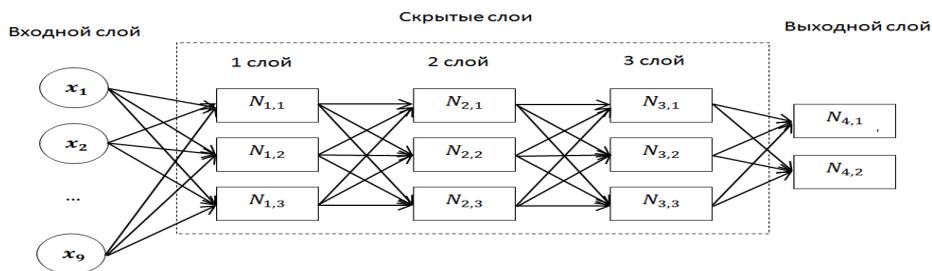


Рис. 1. Структура нейронной сети

Для создания модели нейронной сети мы использовали библиотеку Python Keras. Сначала был создан массив скрытых слоев `hidden_neurons`, где элементы массива – количество нейронов в данном слое. Затем создан входной слой с 9 нейронами по количеству независимых переменных, после чего написан цикл создания скрытых слоев, где задействован массив `hidden_neurons`. После цикла был добавлен выходной слой с двумя нейронами для прогнозирования целевой переменной `Running`.

Для обучения модели нейронной сети мы конвертировали (`one hot encoding`) категориальную целевую переменную `Running` в упорядоченные пары по правилу: `True := (1, 0)`, `False := (0, 1)`. Для всех слоев была выбрана функция активации – сигмоида

$$S(x)=1/(1+\exp(-x)),$$

в качестве оптимизатора выбран метод Adam, средняя квадратическая ошибка была использована в качестве функции потерь:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2,$$

здесь $y_i = \{a, b\}$: $a, b \in (0,1)$ – прогнозируемое значение, x_i – истинное значение, n – размер вектора x_i . Выходной массив y_i формируется нейронной сетью и представляет собой упорядоченные пары, принимающие вещественные значения. Для построения матрицы ошибок мы округляем полученные значения до ближайшего целого.

Результаты работы нейронной сети. Количественными критериями, позволяющими оценить работу сети, является параметр *MSE* и матрица ошибок. Для обучения нейронной сети было выполнено 40 эпох, после которых значение параметра ошибки вышло на стационарное поведение и достигло значения $MSE=0,0256$. По результатам вычислительных экспериментов на тестовом наборе мы провели симуляцию Монте-Карло с 500 итерациями и построили матрицу ошибок, усредненные значения которой равны: $\begin{bmatrix} 118,129 & 1,648 \\ 20,608 & 373,640 \end{bmatrix}$. Из матрицы видно, что в среднем почти все 376 случаев использования роботов были выявлены, при этом в среднем в 21 случае мы получили ошибку 1-го рода, в 2 случаях – ошибку 2-го рода.

Закключение. По результатам симуляции методом Монте-Карло видно, что матрица ошибок устойчива к случайному изменению входных значений тестовой выборки. Таким образом, предложенная модель нейронной сети показала свою работоспособность на использованном наборе данных. Дальнейшие шаги проведения исследований будут направлены на изучение способов мимикрии роботов и их идентификации с использованием методов машинного обучения и математической статистики. В частности стоит учесть факт корреляционной зависимости и рассмотреть категориальную корреляцию Крамера разных уровней признаков `Lives`, `Level` или применить использованный метод кодирования (`one hot encoding`).

СПИСОК ЛИТЕРАТУРЫ

1. Kang A. R., Woo J., Park J., Kim H.K. Online game bot detection based on party-play log analysis. *Computers and Mathematics with Applications*. – 2013. Vol. 65. – pp. 1384–1395.
2. Han M.L., Park J.K., Kim H.K. Online Game Bot Detection in FPS Game // *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems*. – 2018. Vol. 2, – pp. 479-491.