

УДК 004.41

## **ПРИМЕНЕНИЕ МАШИННОГО ОБУЧЕНИЯ ДЛЯ РЕАЛИЗАЦИИ ШАХМАТНОГО БОТА НА ОСНОВЕ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ**

*А. А. Киселёв, Д. В. Антоник*

*В статье приводятся общие подходы к созданию шахматного бота. Шахматный бот основан на машинном обучении, который имеет возможность самообучаться под воздействием новых данных. В результате исследований была получена нейронная сеть, с помощью которой можно находить оптимальный ход в шахматной партии.*

### **Введение**

Сложность шахмат делает невозможным для компьютеров исследовать все возможные варианты ходов во всей шахматной партии и выбрать лучший ход. Поэтому большинство шахматных ботов используют метод перебора для поиска оптимального хода из возможных ходов только до определенной глубины.

Основной недостаток такого подхода заключается в том, что компьютер находит оптимальную стратегию лишь до определенной глубины и, если противник будет анализировать игру глубже, чем это делает компьютер, то в этом случае компьютер не имеет никаких шансов на выигрыш.

Следующим недостатком такого подхода является не способность компьютера обучаться на сыгранных партиях, то есть, если компьютер, основываясь на оценочной функции, делает ход, который приводит его к поражению, то в следующей партии он так же его сделает.

### **Машинное обучение**

Машинное обучение – обширный подраздел искусственного интеллекта, математическая дисциплина, использующая разделы математической статистики, численных методов оптимизации, теории вероятностей, дискретного анализа, и извлекающая знания из данных [1].

Машинное обучение является одним из видов искусственного интеллекта, который позволяет компьютеру предсказать какое-либо действие, не будучи при этом явно запрограммированным на это. Машинное обучение фокусируется на разработке компьютерных программ, которые способны к обучению под воздействием новых данных. Машинное обучение играет ключевую роль в широком диапазоне критически важных приложений, таких, как интеллектуальный анализ данных, обработка естественного языка, распознавание образов.

Таким образом, если необходимо, чтобы программа предсказывала, например, интенсивность движения на оживленном перекрестке, то программу

можно запустить с обучающими данными о прошлой интенсивности движения на этом перекрестке.

### **Искусственная нейронная сеть**

Искусственная нейронная сеть – математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей – сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы [2].

Выделяют следующие этапы решения задач, связанных с построением искусственной нейронной сети:

- сбор данных для обучения;
- подготовка и нормализация данных;
- выбор топологии сети;
- экспериментальный подбор характеристик сети;
- экспериментальный подбор параметров обучения;
- обучение;
- проверка адекватности обучения;
- корректировка параметров, окончательное обучение;
- вербализация сети с целью дальнейшего использования.

### **Основные аспекты написания шахматного бота**

Шахматы – игра с конечным числом состояний [3], то есть, если существовало бы бесконечно много вычислительной мощности, то можно было решить шахматы. Решить шахматы означает найти оптимальную стратегию, при которой игрок всегда будет одерживать победу. Теорема Цермело гласит, что для любой игры с конечным числом состояний, в которой игроки делают ход по очереди и шанс не влияет на процесс принятия решения, может быть найдена оптимальная стратегия игры, которая принесёт игроку выигрыш [4]. Каждая позиция в шахматах либо победа для белых, либо победа для черных или ничья для обоих игроков.

Мы можем обозначить это как некоторую функцию  $f(p)$ . Если мы имели бы бесконечно быструю машину, то смогли бы вычислить это следующим образом:

1. Обозначить конечные позиции значениями -1, 0, 1 в зависимости от того, кто победит.

2. Использовать рекурсивную функцию  $f(p) = \max_{p \rightarrow p'} - f(p')$ , где  $p \rightarrow p'$  обозначает все доступные ходы из позиции  $p$ . Знак минус используется, потому что игроки чередуются между позициями, таким образом если позиция  $p$  является позицией для белых фигур, то позиция  $p'$  является позицией для черных фигур и наоборот. Это тоже самое, что и минимакс. Минимакс – правило принятия решений, используемое в теории игр, теории принятия решений, исследовании операций, статистике и философии для минимизации возможных потерь из тех,

которые лицу, принимающему решение, нельзя предотвратить при развитии событий по наихудшему для него сценарию [5].

Это приблизительно  $10^{43}$  позиций, поэтому мы не сможем вычислить их все. Нам нужно прибегнуть к аппроксимации функции  $f(p)$ .

### **Основной смысл использования машинного обучения для шахматного бота**

Машинное обучение сводится к аппроксимации функции с помощью заданных данных. Таким образом на основе полученных данных мы можем вычислить приближенное значение функции  $f(p)$ .

Функция  $f(p)$  обучена по истории сыгранных игр, используя при этом два принципа:

1. Игроки выбирают оптимальных ход или ход, который близок к оптимальному. Это означает, что для двух позиций в последовательности  $p \rightarrow q$  в анализируемой игре, мы имеем  $f(p) = -f(q)$ .

2. Переходя от  $p$  не к  $q$ , а к случайной позиции  $p \rightarrow r$ , мы должны иметь  $f(r) > f(q)$ , потому что случайное положение лучше для следующего игрока и хуже для игрока, который сделал этот ход.

### **Выбор случайного хода**

Для определения оценки позиции и выбора лучшего хода в программах, играющих в шахматы и другие игры, обычно используются переборные алгоритмы с отсечениями и различными эвристиками, позволяющими сокращать количество рассматриваемых позиций. В процессе перебора генерируются ходы из текущей позиции (рисунок 1), после чего рекурсивно обходятся доступные после каждого хода позиции и выбирается та, которая максимизирует выигрыш противника, взятый с противоположным знаком.

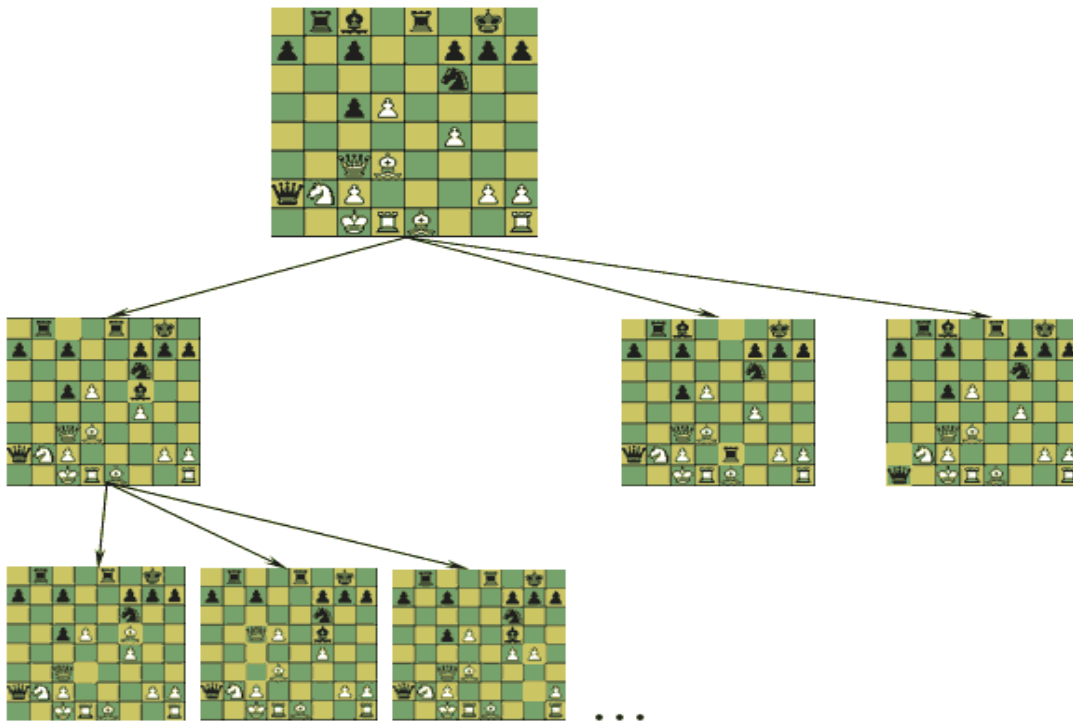


Рис. 1. Генерация ходов из текущей позиции

Для снижения времени, затрачиваемого на перебор, обычно используются альфа-бета отсечения, которые позволяют отсекать целые поддеревья, при этом не влияя на результат. В алгоритме с альфа-бета отсечениями при рекурсивном вызове передаются альфа и бета – нижняя и верхняя границы значений соответственно. Если возвращаемое значение превышает бета, то происходит отсечение, так как это значение уже не может повлиять на результирующую оценку родительского узла (рисунок 2).

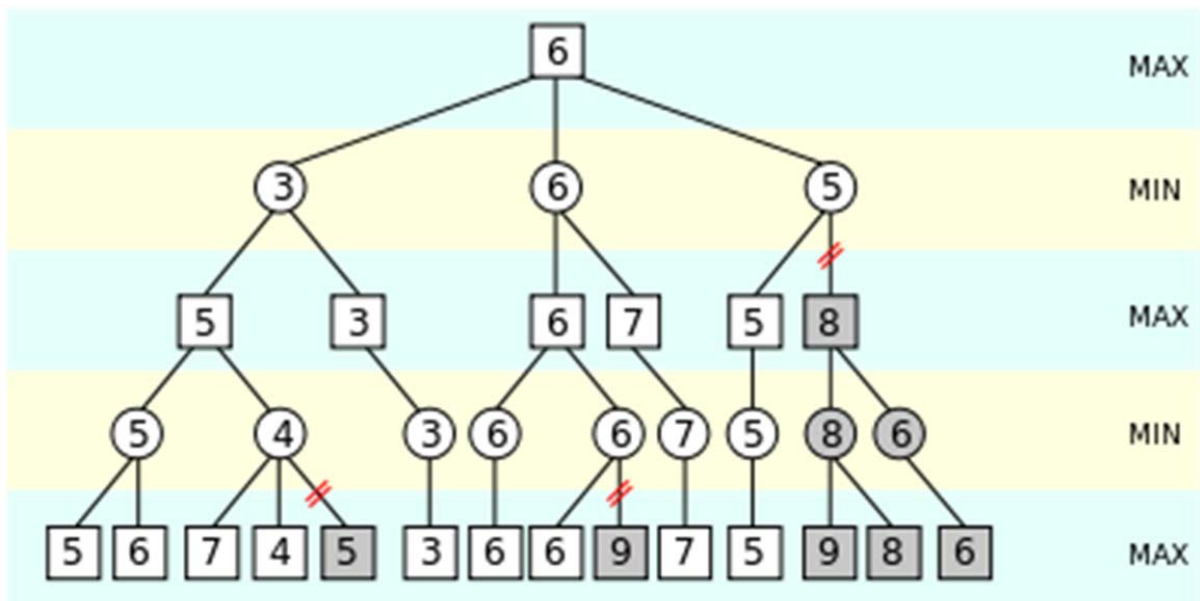


Рис. 2. Альфа-бета отсечение поддеревьев

При выборе лучшего хода компьютером необходимо минимизировать стоимость доски для оппонента и максимизировать стоимость доски для текущего игрока.

Исходя из вышесказанного был разработан алгоритм, который позволяет получить лучший ход. В листинге 1 представлен псевдокод этого алгоритма.

Листинг 1. Алгоритм выбора компьютером лучшего хода.

```
Функция выбора лучшего хода (глубина просмотра, цвет игрока, альфа, бета)
    Получить все ходы для текущего цвета
    Если глубина просмотра равна нулю
        Вернуть стоимость доски из текущей позиции
    Взять за лучшую оценку минимальное значение
    Если необходимо максимизировать результат
        Цикл для всех возможных ходов у текущего цвета
            Сделать ход
            Рекурсивный вызов функции выбора лучшего хода (глубина - 1,
противоположный цвет, альфа, бета)
            Отменить последний ход
            Если лучшая оценка доски меньше текущей оценки доски
                Взять за лучшую оценку текущий результат
            Если текущая оценка больше альфы
                Взять в качестве альфа текущий результат
            Если бета меньше или равна альфа
                Принять за лучшую оценку бета
                Вернуть лучшую оценку
        Вернуть лучшую оценку
    Иначе
        Цикл для всех возможных ходов у текущего цвета
            Сделать ход
            Рекурсивный вызов функции выбора лучшего хода (глубина - 1,
противоположный цвета, альфа, бета)
            Отменить последний ход
            Если лучшая оценка доски больше текущей оценки доски
                Взять за лучшую оценку текущий результат
            Если текущая оценка меньше бета
                Взять в качестве бета текущий результат
            Если бета меньше или равна альфа
                Принять за лучшую оценку альфа
                Вернуть лучшую оценку
        Вернуть лучшую оценку
```

## Модель искусственной нейронной сети

Построенная нейронная сеть имеет глубину 3 уровня и 2048 нейронов на каждом уровне. Входной слой состоит из 12 64-битных блоков. Каждый 64-битный блок представляет позицию одной из 12 возможных фигур на шахматной доске.

После трех матричных умножений получаем конечную точку с шириной вектора 2048, который необходимо упростить до одного значения (рисунок 3).

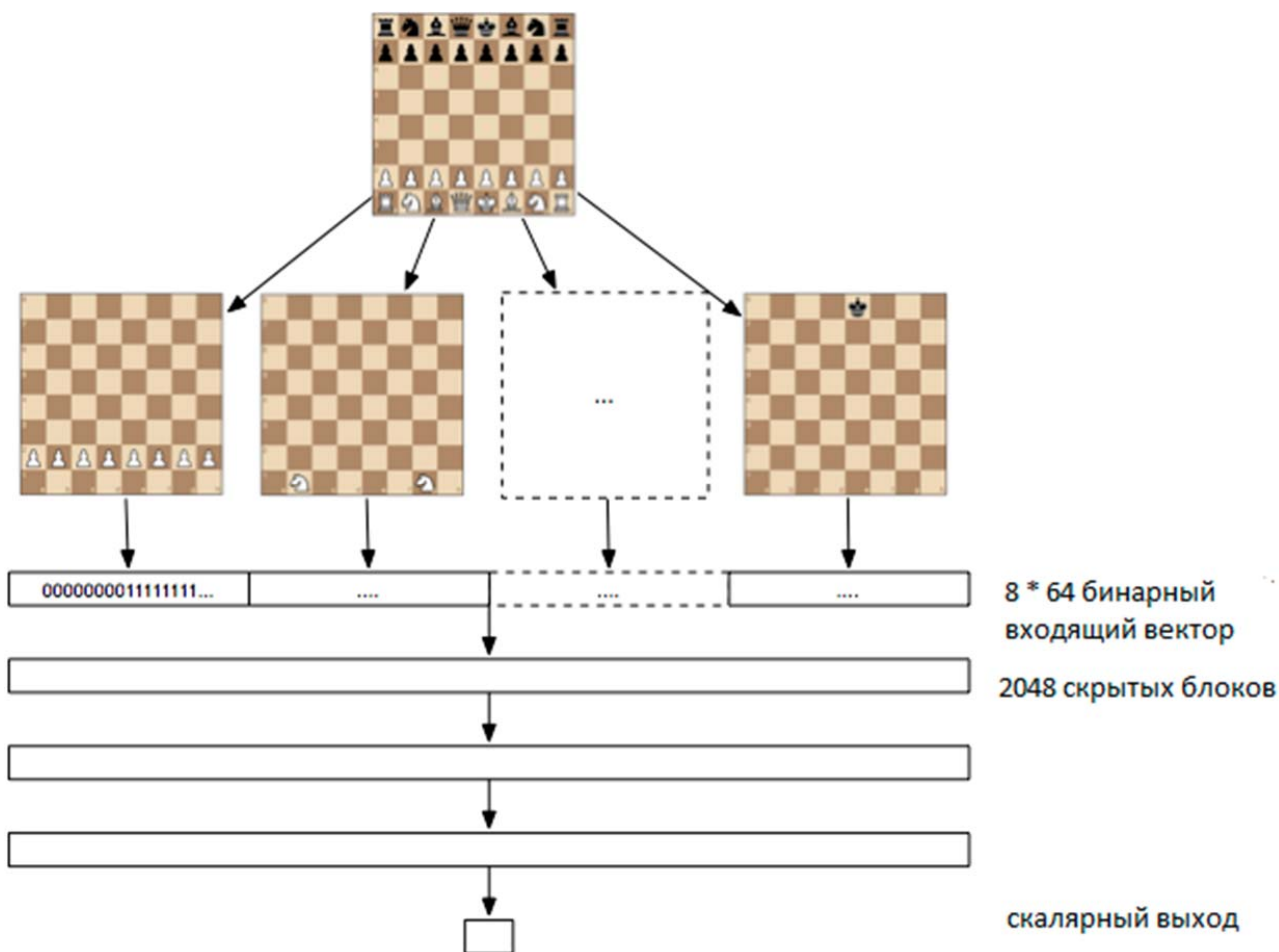


Рис. 3. Искусственная нейронная сеть для шахматного бота

Предполагается, что данные содержат информацию о функции оценки, которая представляет оценку хода, где равно 1 для возможного выигрыша, 0 для возможной ничьей и -1 для возможного поражения.

Искусственная нейронная сеть была обучена с тройкой значений, где — текущее положение, — ход выбранного игрока, — случайный ход. На рисунке 4 представлен основной принцип обучения искусственной нейронной сети.

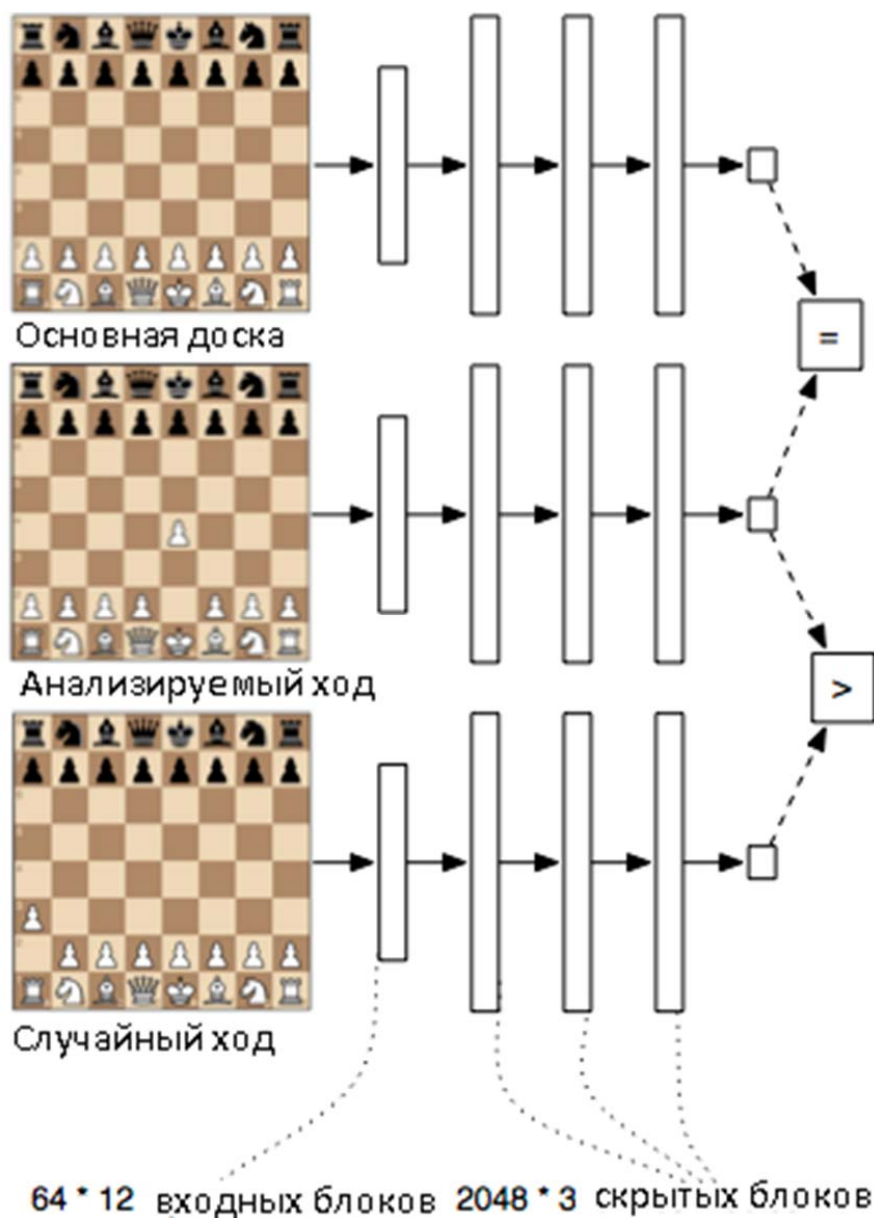


Рис. 4. Принцип обучения искусственной нейронной сети

Для обучения сети использовалась база данных с историей более миллиона шахматных партий.

### Заключение

Полученный шахматный бот тестировался против других популярных шахматных ботов, и он показал весьма удовлетворительные результаты. Так, например, игра против знаменитого шахматного бота sunfish заканчивалась в основном через 35-40 ходов победой sunfish, но в некоторых случаях были и ничьи.

Таким образом, машинное обучение прекрасно решает задачи написания искусственного интеллекта для различных логических игр не зависимо от их сложности.

## Список литературы

1. Машинное обучение // Википедия [Электронный ресурс]. – 2016. – Режим доступа: [https://ru.wikipedia.org/wiki/ Машинное\\_обучение](https://ru.wikipedia.org/wiki/Машинное_обучение). – Дата доступа: 31.03.2016.
2. Искусственная нейронная сеть // Википедия [Электронный ресурс]. – 2016. – Режим доступа: [https://ru.wikipedia.org/wiki/ Искусственная\\_нейронная\\_сеть](https://ru.wikipedia.org/wiki/Искусственная_нейронная_сеть). – Дата доступа: 01.04.2016.
3. Шахматы // Википедия [Электронный ресурс]. – 2016. – Режим доступа: <https://ru.wikipedia.org/wiki/Шахматы>. – Дата доступа: 31.03.2016.
4. Zermelo's theorem (game theory) // Википедия [Электронный ресурс]. – 2016. – Режим доступа: [https://en.wikipedia.org/wiki/ Zermelo%27s\\_theorem\\_\(game\\_theory\)](https://en.wikipedia.org/wiki/Zermelo%27s_theorem_(game_theory)). – Дата доступа: 01.04.2016.
5. Минимакс // Википедия [Электронный ресурс]. – 2016. – Режим доступа: <https://ru.wikipedia.org/wiki/Минимакс>. – Дата доступа: 01.04.2016.

*Киселев Андрей Александрович, студент 5 курса кафедры современных технологий программирования факультета математики и информатики Учреждения образования «Гродненский государственный университет имени Янки Купалы», [vvital@tut.by](mailto:vvital@tut.by)*

*Антоник Денис Владимирович, старший преподаватель кафедры современных технологий программирования Учреждения образования «Гродненский государственный университет имени Янки Купалы», [antonik@grsu.by](mailto:antonik@grsu.by)*