



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего  
образования

«Дальневосточный федеральный университет»

---

**Институт математики и компьютерных технологий**

**Департамент математического и компьютерного моделирования**

**Проект**

по дисциплине «Введение в искусственный интеллект» на тему «Движок для  
китайской шахматы на основе CNN.»

Направление подготовки 02.03.01 «Математика и компьютерные науки»

Выполнил(а) студент(ка) гр. Б9121-02.03.01сцт

---

(подпись)

Москера Креспо А.Х.

Проверил

---

(подпись)

Литвин М.

5 февраля 2025г.

**г. Владивосток**

**2025**

# Введение

Сянци, или китайские шахматы, представляет собой древнюю стратегическую игру с уникальными правилами и полем размером  $10 \times 9$  клеток. Отличаясь от классических шахмат специфическими фигурами и ограничениями на их движения, сянци требует особого подхода при создании программ искусственного интеллекта.

В данном проекте разработан движок для сянци, основанный на свёрточных нейронных сетях (CNN). CNN особенно эффективны для анализа двумерных структур, таких как игровое поле сянци, позволяя выявлять важные позиционные и тактические паттерны. Благодаря этому технология помогает движку лучше понимать стратегические аспекты игры и принимать более качественные решения.

Проект демонстрирует, как современные методы машинного обучения могут быть успешно применены для создания конкурентоспособных компьютерных игр в традиционных видах настольных игр.

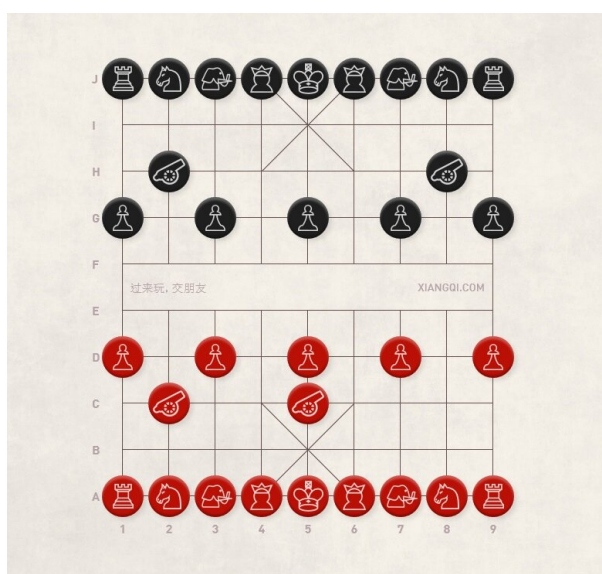


Рис. 1: Китайские шахматы (западная символизация)

## 1 Основная часть

### 1.1 Архитектура нейронной сети

Разработанная модель представляет собой свёрточную нейронную сеть (CNN) с двумя головками предсказания, спроектированную для анализа позиций в сянци. Архитектура оптимизирована для решения задачи предсказания оптимальных ходов в китайских шахматах.

#### 1.1.1 Входные данные

На вход сети подаётся тензор размерности:

$$(\text{None}, 10, 9, 7)$$

где:

- $10 \times 9$  — размер игровой доски в сянци
- 7 — количество каналов, кодирующих информацию о фигурах и их характеристиках (1 для красных, -1 для черных)
- None — размер батча

### 1.1.2 Выходные данные

Сеть имеет два независимых выхода:

- `output_from_move` (None, 90) — вероятность выбора начальной позиции
- `output_to_move` (None, 90) — вероятность выбора конечной позиции

Оба выхода преобразуются в матрицы размерности (None, 10, 9) через операцию Reshape.

### 1.1.3 Структура сети

Архитектура состоит из двух основных блоков:

#### 1. Блок предсказания начальной позиции:

- Свёрточный слой (128 фильтров, ядро  $3 \times 3$ , padding="same")
- Пакетная нормализация (BatchNormalization)
- Активация ReLU
- Свёрточный слой (256 фильтров, ядро  $5 \times 5$ , padding="same")
- Пакетная нормализация
- Активация ReLU
- Полносвязный слой с активацией softmax

#### 2. Блок предсказания конечной позиции:

- Объединение исходного входа с предсказанием начальной позиции
- Аналогичная свёрточная структура с дополнительной обработкой комбинированных признаков
- Итоговый полносвязный слой с активацией softmax

### 1.1.4 Особенности реализации

Архитектура сети использует операцию `Concatenate` для объединения исходного состояния доски с предсказанием начальной позиции, что позволяет учитывать контекст при выборе конечной позиции. После каждого свёрточного слоя применяется пакетная нормализация для стабилизации обучения, а в качестве функции активации используется ReLU, обеспечивающая нелинейность. Модель построена по двухэтапной схеме с отдельными головками предсказания, что позволяет независимо обрабатывать начальные и конечные позиции. Пространственные характеристики оптимизируются за счёт свёрточных слоёв, которые сохраняют локальные зависимости на доске.

Эксперименты с добавлением рекуррентных связей (например,  $x = F(x_{\text{prev}} + F(x_{\text{prev}}))$ ) показали снижение точности, поэтому от них было решено отказаться. Пулинг-слои не используются, так как они размывают важную позиционную информацию, а дропаут исключён для сохранения максимального количества данных, поскольку в сянги практически вся информация о состоянии доски является полезной для принятия решений.

```
inputs = tfk.Input(shape=BOARD_SHAPE)

from_head = layers.Conv2D(128, (3, 3), padding="same")(inputs)
from_head = layers.BatchNormalization()(from_head)
from_head = layers.ReLU()(from_head)
from_head = layers.Conv2D(256, (5, 5), padding="same")(from_head)
from_head = layers.BatchNormalization()(from_head)
from_head = layers.ReLU()(from_head)

from_head = layers.Flatten()(from_head)
from_head = layers.Dense(90, activation="softmax", name="output_from_move")(
    from_head
)

from_head_resaped = layers.Reshape((10, 9, 1))(from_head)

combined = layers.Concatenate(axis=-1)([inputs, from_head_resaped])

to_head = layers.Conv2D(128, (3, 3), padding="same")(combined)
to_head = layers.BatchNormalization()(to_head)
to_head = layers.ReLU()(to_head)
to_head = layers.Conv2D(256, (5, 5), padding="same")(to_head)
to_head = layers.BatchNormalization()(to_head)
to_head = layers.ReLU()(to_head)

to_head = layers.Flatten()(to_head)
to_head = layers.Dense(90, activation="softmax", name="output_to_move")(to_head)

model = models.Model(
    inputs=inputs, outputs=[from_head, to_head], name="xiangqi_model"
)
```

Рис. 2: Фрагмент реализации модели на TensorFlow Keras

### 1.1.5 Обучение модели

- Функция потерь: categorical crossentropy
- Оптимизатор: Adam с learning rate=0.001
- Метрики оценки: Accuracy

- Регуляризация: L2-регуляризация на свёрточных слоях

### 1.1.6 Пользовательский интерфейс

Для демонстрации работы модели был использован проект `xiangqi`, доступный на GitHub по ссылке: <https://github.com/techkang/xiangqi>. Этот проект представляет собой текстовый пользовательский интерфейс (TUI), который изначально был разработан для игры в сянци с использованием модели, основанной на оценке позиций. Исходная модель принимала решения, выбирая ходы, которые максимизируют позиционную оценку, а также учитывала матовые ситуации в один ход. Интерфейс использует китайские символы для отображения фигур на доске, что соответствует традиционному представлению сянци.

Для интеграции разработанной CNN-модели в интерфейс исходный код проекта был модифицирован. Вместо получения ходов от пользователя, программа теперь принимает входные данные от модели, что позволяет автоматизировать процесс игры. Это изменение открывает возможности для использования интерфейса в будущих задачах, таких как обучение с подкреплением (Reinforcement Learning), где автоматизированные игры могут служить основой для сбора данных и улучшения модели.

```

41 My move: 卒 7 进 1
40 Current evaluation: -118.
39
38 9 車 象 士 将 士 象 車
37 8 . . . . . . . .
36 7 . . 马 . 炮 . 马 炮 .
35 6 卒 . . . 卒 . . . 卒
34 5 兵 . . . 兵 . . . 兵
33 4 卒 . . . 卒 . . . 卒
32 3 兵 . . . 兵 . . . 兵
31 2 . 炮 马 . . . 马 炮 .
30 1 . . . 帅 . . . 车
29 0 车 相 仕 . 仕 相 . .
28 a b c d e f g h i
27
26 CNN's move: 车 八 平 九
25 Current evaluation: 130.
24
23 9 車 象 士 将 士 象 車
22 8 . . . . . . . .
21 7 . . 马 . 炮 . 马 炮 .
20 6 卒 . . . 卒 . . . 卒
19 5 . . . . . . . .
18 4 . 卒 . . . 卒 . . .
17 3 兵 . . . 兵 . . . 兵
16 2 . 炮 马 . . . 马 炮 .
15 1 . . . 帅 . . . 车
14 0 车 . 相 仕 . 仕 相 . .
13 a b c d e f g h i
12

```

Рис. 3: Пример работы TUI.

## 2 Результаты

### 2.1 Обучение модели

Обучение модели показало точность предсказания начальной и конечной позиции фигур на уровне 60%. Это значение было достигнуто на ограниченном наборе данных, что может свидетельствовать о недостаточном объёме обучающей выборки для полного охвата всех возможных игровых ситуаций в сянци.

### 2.2 Данные для обучения

Данные для обучения были извлечены из открытого набора данных, доступного на GitHub: <https://github.com/zuison/rs-chinese-chess>. Этот набор данных содержит информацию о лучших ходах, их весовых коэффициентах и FEN-строках, описывающих позиции на доске. Однако набор данных не охватывает все возможные ходы и позиции, что делает его ограниченным. Это может быть одной из причин относительно невысокой точности модели. В будущем планируется расширить набор данных с помощью методов обучения с подкреплением (Reinforcement Learning), что позволит улучшить производительность модели.

### 2.3 Сравнение с TUI-моделью

При тестировании модели против TUI-модели, используемой в интерфейсе, разработанная CNN-модель чаще проигрывает. Это может быть связано с несколькими факторами:

- TUI-модель основана на чисто позиционной оценке и не учитывает стоимость фигур или комплексную оценку позиции в виде единого числового значения.
- TUI-модель детерминирована: для одной и той же позиции она всегда возвращает одинаковый ход. Чтобы добавить вариативность, в разработанную CNN-модель был добавлен случайный множитель в диапазоне  $(0, 1]$ , который рандомизирует выходные данные. Это улучшило производительность модели, сделав её менее предсказуемой.

### 2.4 Перспективы улучшения

Для повышения точности модели планируется:

- Расширить набор данных за счёт автоматизированных игр и обучения с подкреплением.
- Добавить оценку стоимости фигур и комплексную оценку позиции.
- Улучшить архитектуру модели, например, за счёт добавления механизмов внимания (attention) или более глубоких свёрточных слоёв.

## **Список использованных источников**

1. <https://github.com/zuisong/rs-chinese-chess>
2. <https://github.com/techkang/xiangqi>