МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Южно-Уральский государственный университет (национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

Применение глубокого обучения с подкреплением в играх двух лиц

09.04.04 «Программная инженерия»

Научный руководитель:
зав. каф. СП, д.ф.-м.н.,
профессор
Л.Б. Соколинский

Автор:
студент группы КЭ-228
Владислав Федорович Ушаков

Челябинск, 2025 г.

АКТУАЛЬНОСТЬ РАБОТЫ

- Глубокое обучение с подкреплением в играх двух лиц является одним из самых важных исследовательских направлений в области искусственного интеллекта
- Обучение с подкреплением позволяет нивелировать проблему создания обучающих наборов данных
- Выбор эффективных нейросетевых архитектур для глубокого обучения с подкреплением остается важной исследовательской задачей

ЦЕЛЬ РАБОТЫ

Разработать и исследовать нейронные сети с одним и многими выходами, аппроксимирующие Q-функцию для глубокого обучения с подкреплением в играх двух лиц

Исследовательская задача предложена научным руководителем

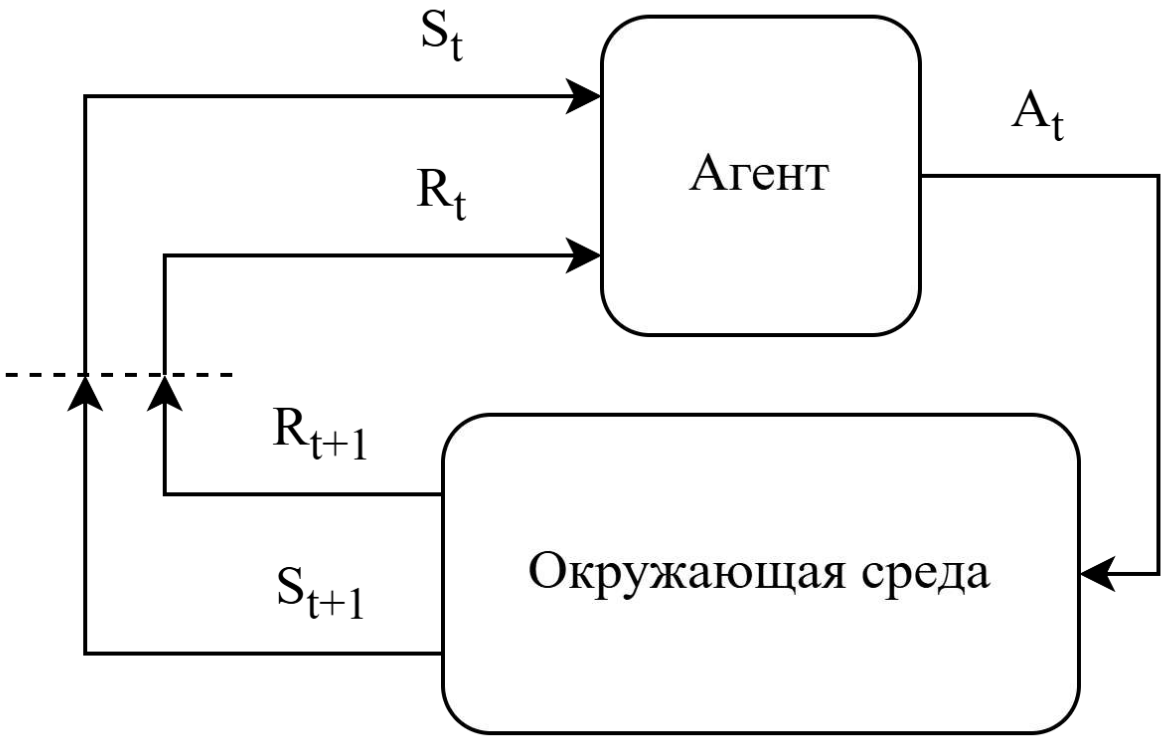
ЗАДАЧИ РАБОТЫ

1. Провести анализ предметной области
2. Реализовать систему глубокого обучения с подкреплением на основе нейросетевой модели со многими выходами для игры «Крестики-нолики»
3. Спроектировать и реализовать систему глубокого обучения с подкреплением на основе нейросетевой модели с одним выходом для игры «Крестики-нолики»
4. Провести сравнительный анализ двух архитектур для игры «Крестики-нолики»
5. Спроектировать и реализовать систему глубокого обучения с подкреплением на основе нейросетей с одним и многими выходами для игры «Шашки 6×6»
6. Провести сравнительный анализ двух архитектур для игры «Шашки 6×6»
7. Разработать графический интерфейс для игры «Крестики-нолики», предусматривающий игру человека с обученным агентом

ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ

МАРКОВСКИЙ ПРОЦЕСС ПРИНЯТИЯ РЕШЕНИЙ

- Агент
- Действие (A_t)
- Окружающая среда
- Состояние (S_t)
- Вознаграждение (R_t)



Эпизод:

$$(S_0, A_0, 0) \rightarrow (S_1, A_1, R_1) \rightarrow (S_2, A_2, R_2) \rightarrow \cdots \rightarrow (S_T, \blacksquare, R_T)$$

Q-ФУНКЦИЯ

- **Цель агента** – взаимодействовать со средой, выбирая действия таким образом, чтобы **максимизировать** будущие вознаграждения
- Чтобы определить **ожидаемое** суммарное вознаграждение в состоянии S_t при выборе действия A_t, используется **Q-функция** $Q(S_t, A_t)$
- Основная идея алгоритмов обучения с подкреплением – приближать Q-функцию к **оптимальной Q-функции** $Q_*(S_t, A_t)$, которая определяет **максимальное** суммарное вознаграждение
- Глубокое обучение с подкреплением используется для аппроксимации оптимальной Q-функции параметрами нейронной сети θ:

$$Q(S_t, A_t; \theta) \approx Q_*(S_t, A_t)$$

Источник

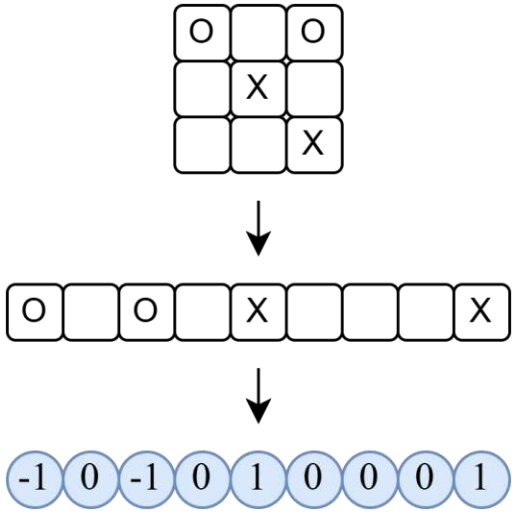
Playing Atari with Deep Reinforcement Learning

Mnih V. и др. // arXiv. 2013. URL: https://arxiv.org/abs/1312.5602

«КРЕСТИКИ-НОЛИКИ»

ЦИФРОВАЯ МОДЕЛЬ ИГРЫ «КРЕСТИКИ-НОЛИКИ»

Состояние (S_t)

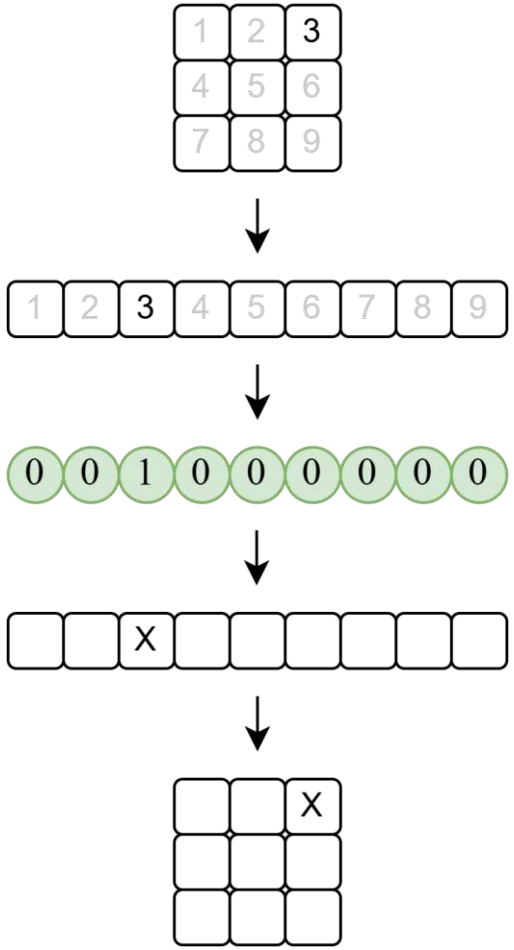


Размер вектора состояния: 9

Вознаграждение (R_t)

Победа	+1
Поражение	-1
Ничья	0

Действие (A_t)

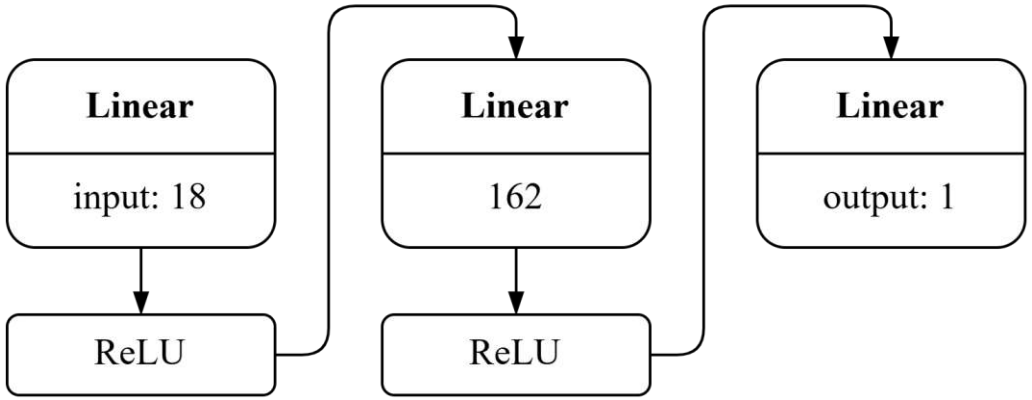


Размер вектора действия: 9

ВИДЫ НЕЙРОСЕТЕВЫХ АРХИТЕКТУР

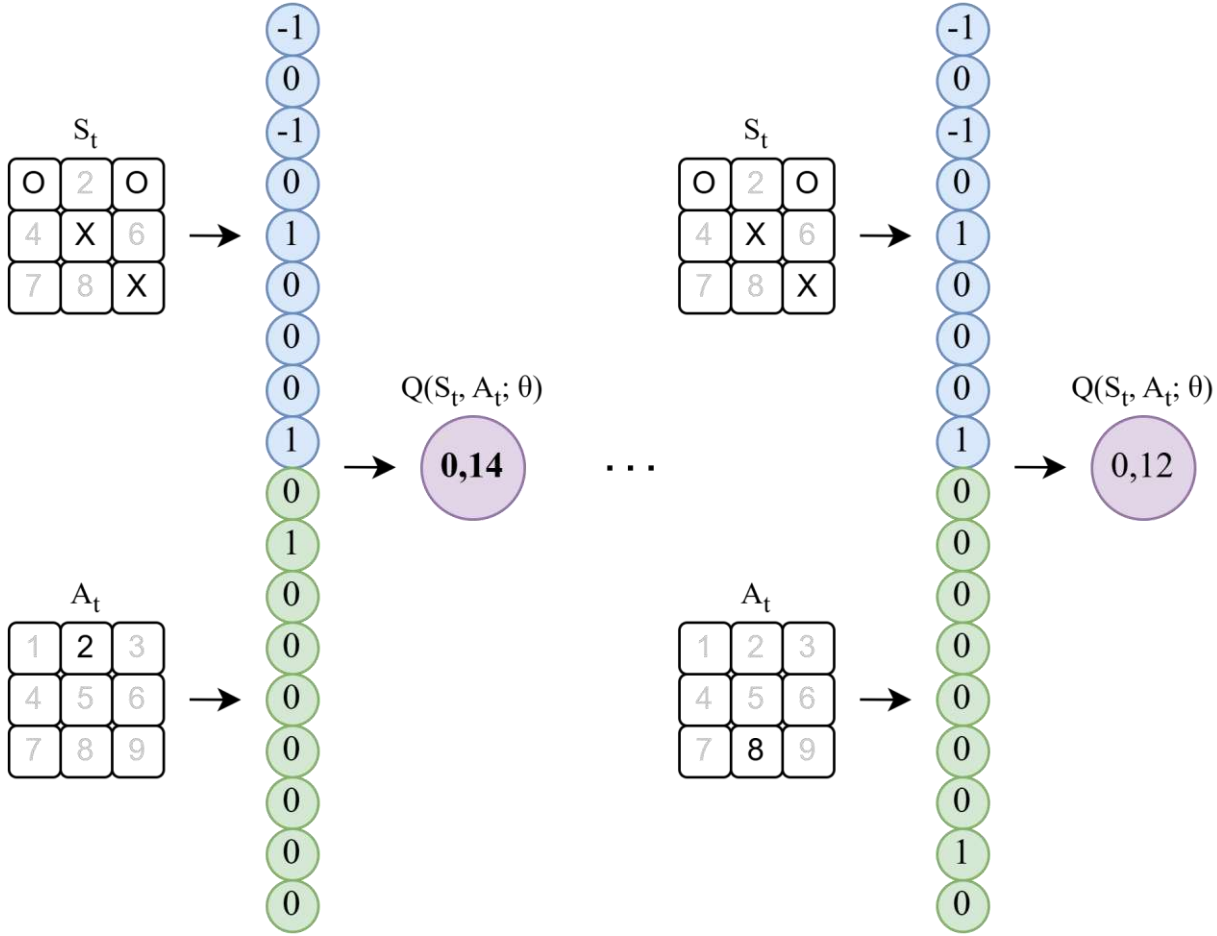
- Нейросетевая модель **с одним** выходным нейроном
- Нейросетевая модель **со многими** выходными нейронами

НЕЙРОННАЯ СЕТЬ С ОДНИМ ВЫХОДОМ

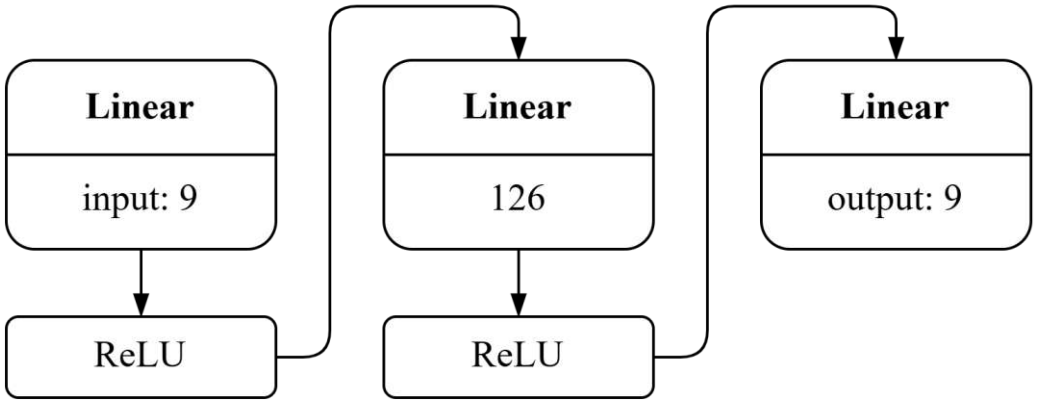


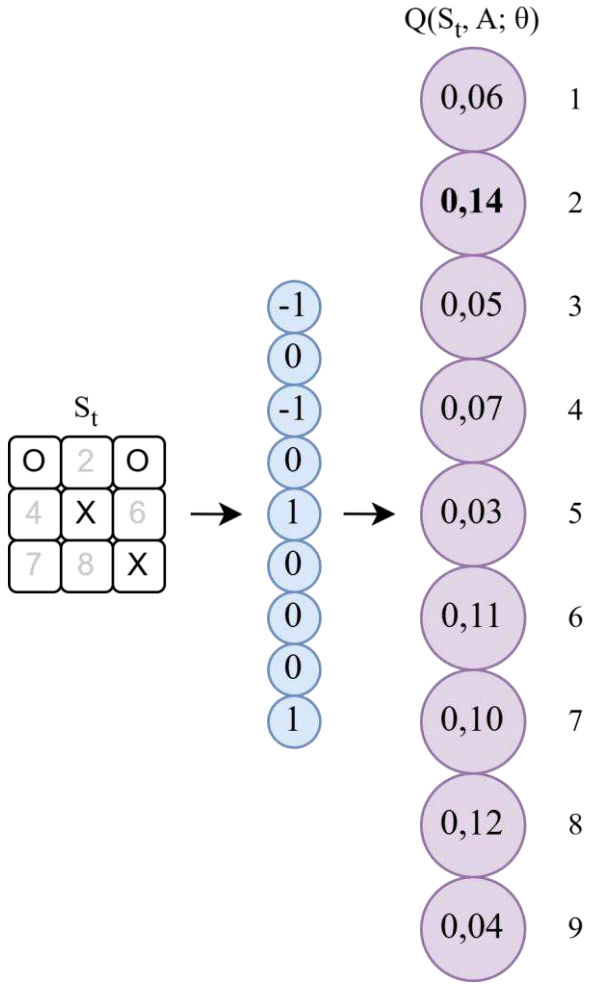
Вход: пара состояние-действие (S_t, A_t)

Выход: качество одного действия $Q(S_t, A_t; \theta)$



НЕЙРОННАЯ СЕТЬ СО МНОГИМИ ВЫХОДАМИ





Вход: состояние S_t

Выход: качество каждого действия $Q(S_t, A; \theta)$

Deep Q Learning for Tic Tac Toe - The Minimum Viable Model

https://the-mvm.github.io/deep-q-learning-tic-tac-toe.html

Источник

ОБУЧЕНИЕ МЕТОДОМ ИГРЫ С САМИМ СОБОЙ

Цель: Аппроксимировать оптимальную Q-функцию

Метод обучения:

- **Противник:** Сам агент
  (встроен в реализацию модели обучения с подкреплением)
- **Стратегия:** ε-жадная
- **Количество эпизодов* обучения:** 6 000
- **Набор данных:** Генерируется из сыгранных эпизодов*
- **Число экспериментов:** 70 (с различными гиперпараметрами)

*Эпизод – одна игровая партия

ТЕСТИРОВАНИЕ АГЕНТА

Цель: Оценить эффективность обучения

Условия тестирования:

- **Противник** делающий случайные ходы
  (встроен в реализацию модели обучения с подкреплением)
- **Стратегия:** Жадная
- **Частота тестирования:** каждые 100 эпизодов* обучения
- **Тестовый набор:** 100 эпизодов*
- **Метрика:** Процент побед

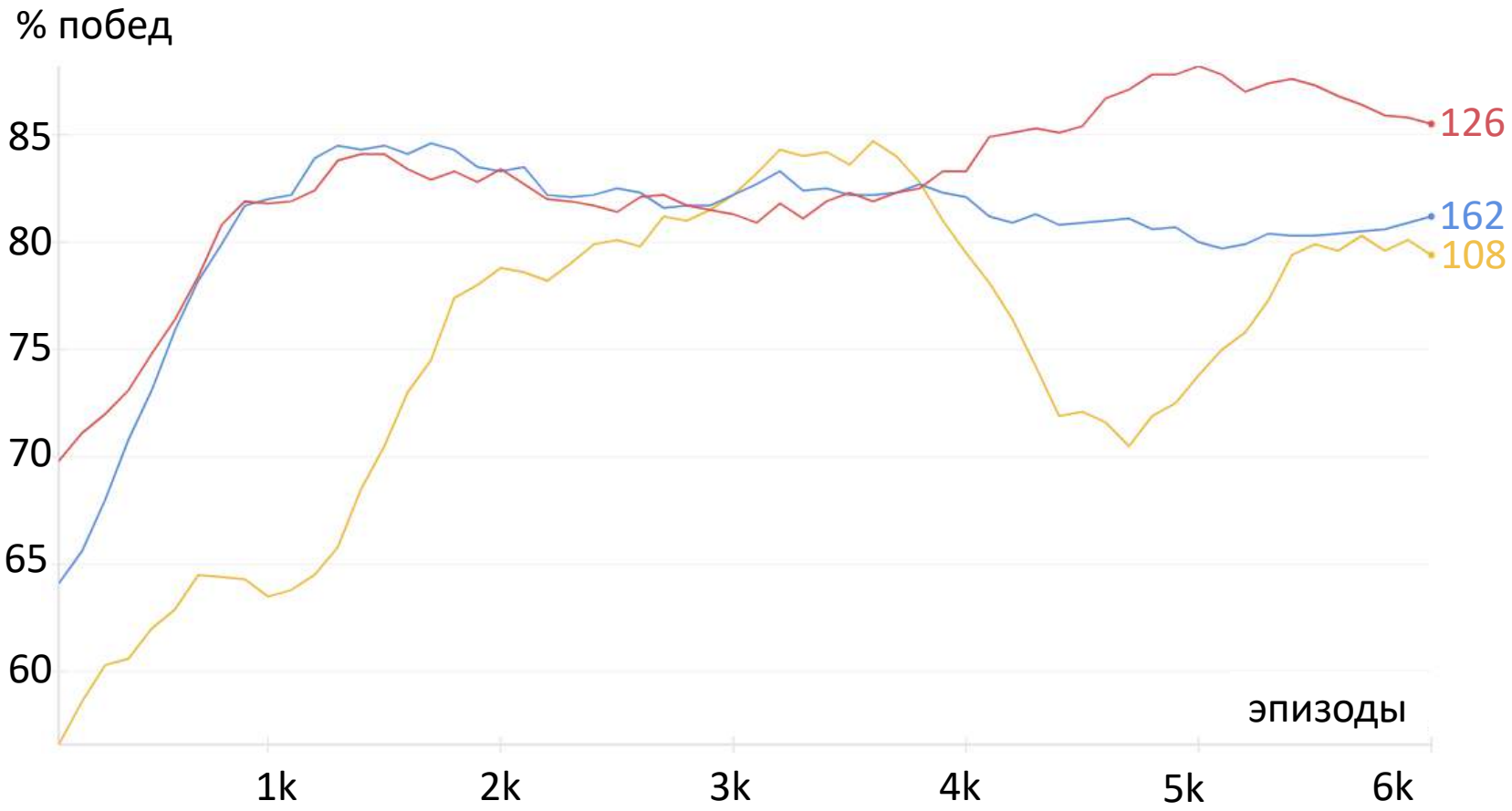
*Эпизод – одна игровая партия

СКРЫТЫЙ СЛОЙ В НЕЙРОСЕТИ С ОДНИМ ВЫХОДОМ

Сравнение эффективности моделей с различным размером скрытого слоя: 126, 162 и 180 нейронов

Гиперпараметр обучения	Значение
Доля случ. действий (ε)	0,1
Скорость обучения	0,001
Вознагр. за победу	0,5
Вознагр. за поражение	-0,5
Вознагр. за ничью	0,2

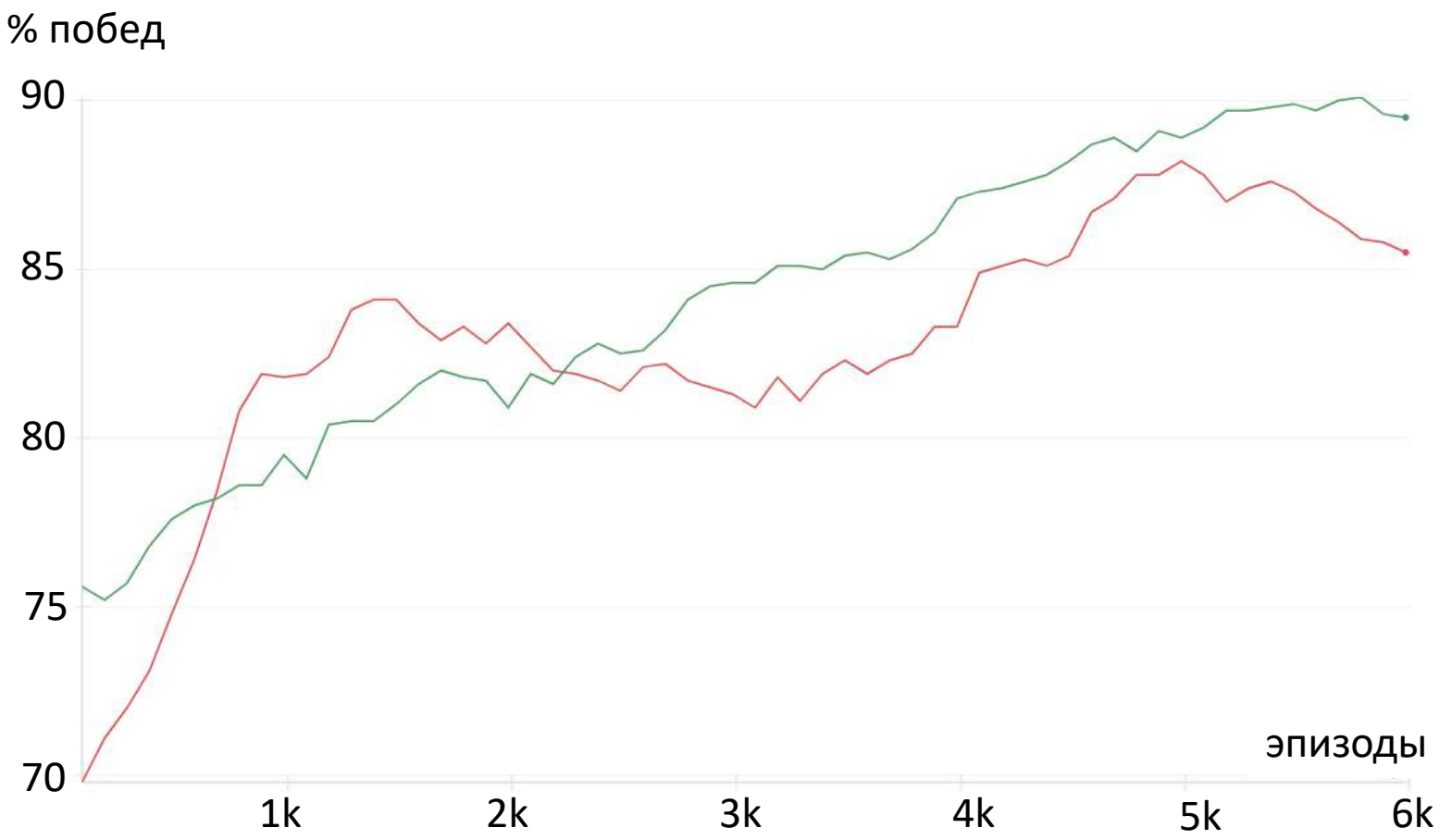
СКРЫТЫЙ СЛОЙ В НЕЙРОСЕТИ СО МНОГИМИ ВЫХОДАМИ



Сравнение эффективности моделей с различным размером скрытого слоя: 108, 126 и 162 нейронов

Гиперпараметр обучения	Значение
Доля случ. действий (ε)	0,1
Скорость обучения	0,001
Вознагр. за победу	0,5
Вознагр. за поражение	-0,5
Вознагр. за ничью	0,2

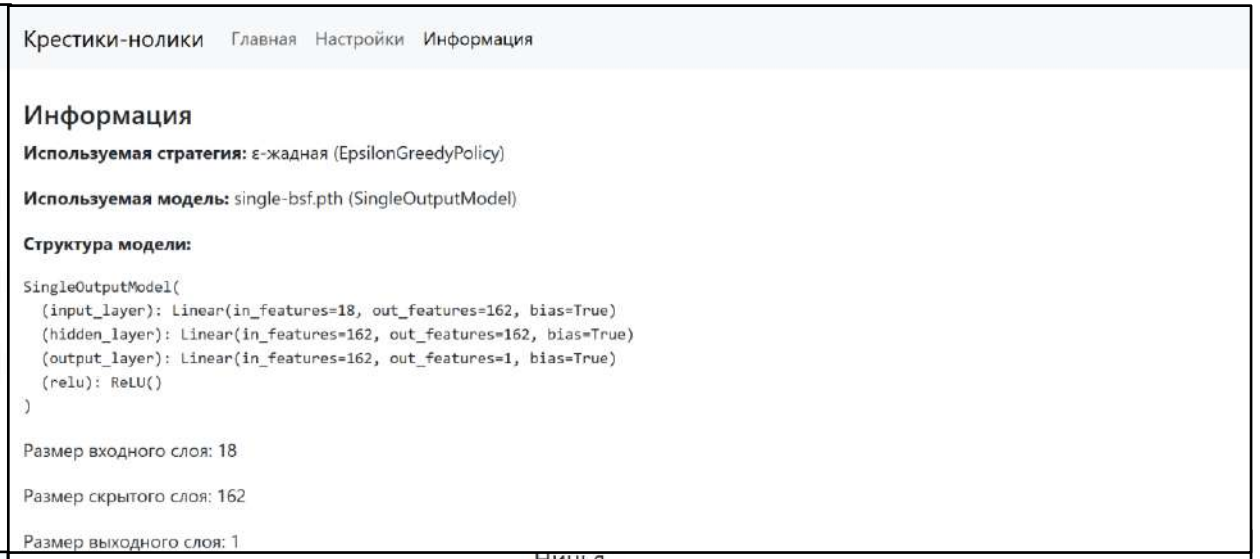
РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

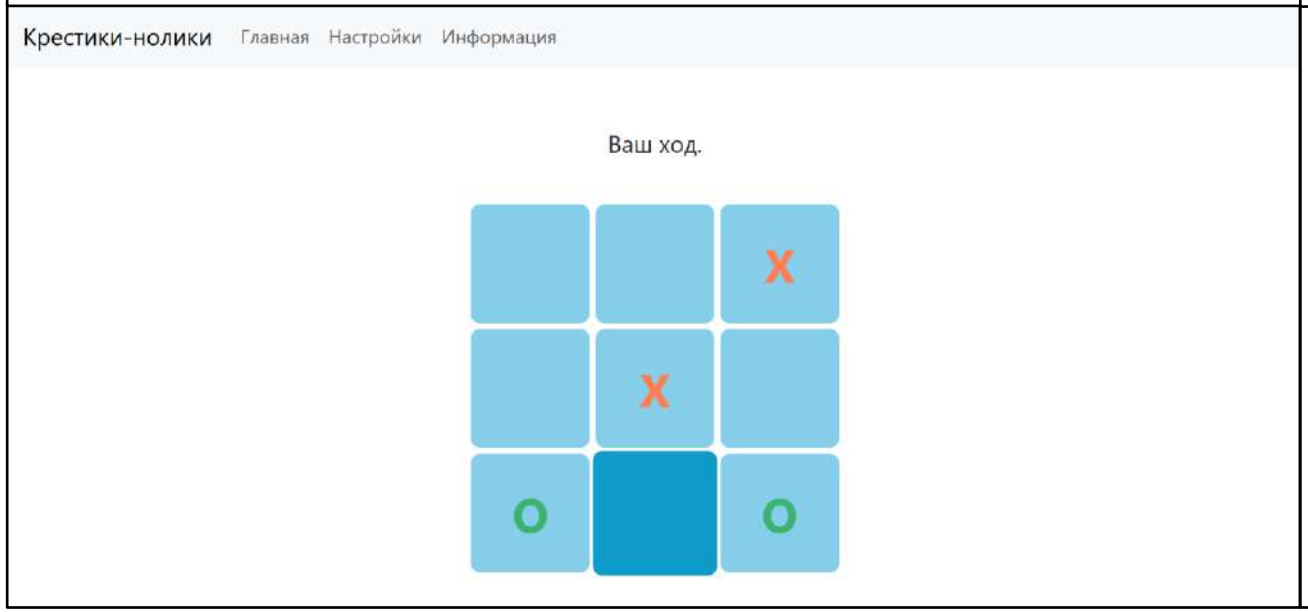


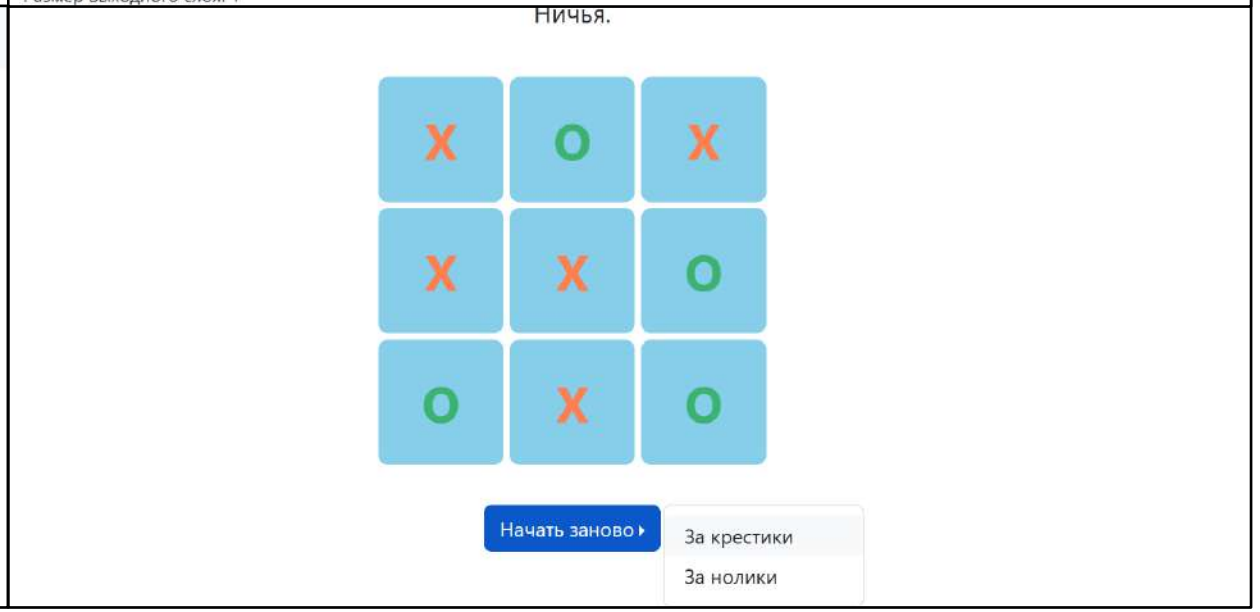
Сравнение эффективности моделей с одним и многими выходами

Гипер-параметр обучения	С одним выходом	Со многими выходами
Доля случ. действий (ε)	0,1	
Скорость обучения	0,001	
Вознагр. за победу	0,5	
Вознагр. за поражение	-0,5	
Вознагр. за ничью	0,2	
Размер скрытого слоя	162	126

СКРИНШОТЫ ПРИЛОЖЕНИЯ



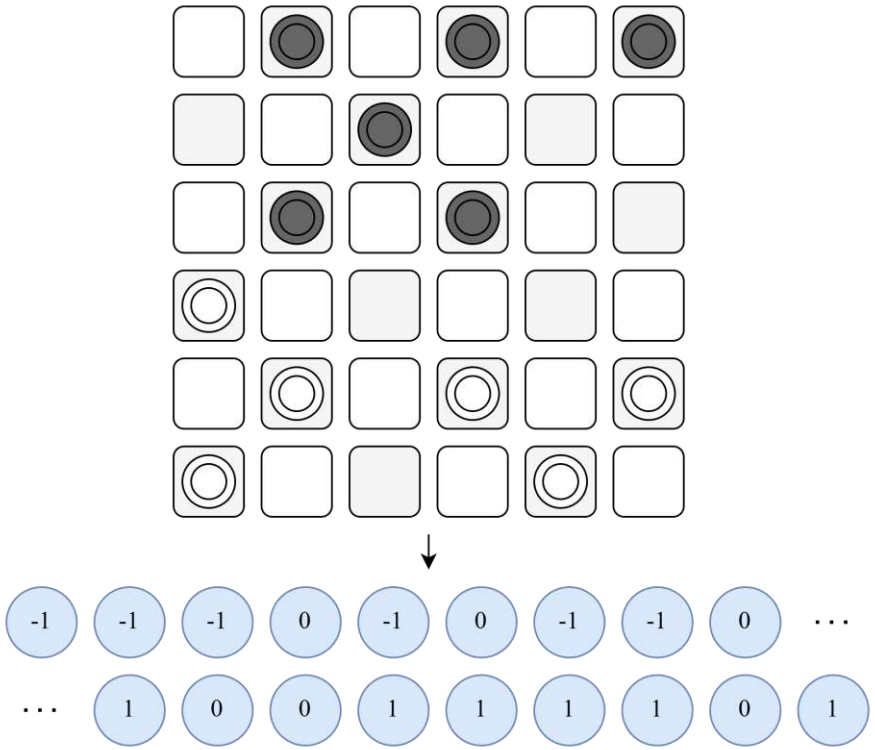




«ШАШКИ 6×6»

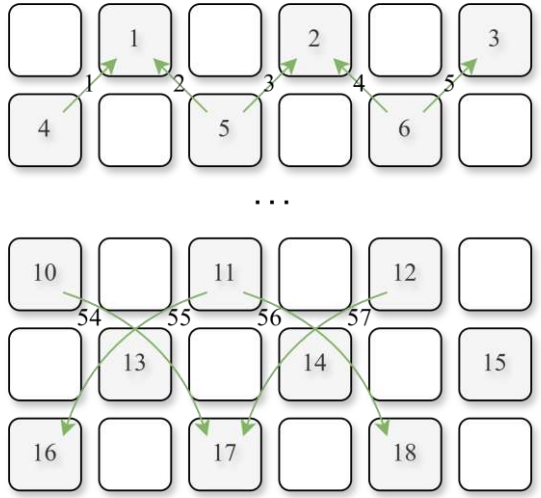
ЦИФРОВАЯ МОДЕЛЬ ИГРЫ «ШАШКИ 6×6»

Состояние (S_t)



Размер вектора состояния: 18

Действие (A_t)



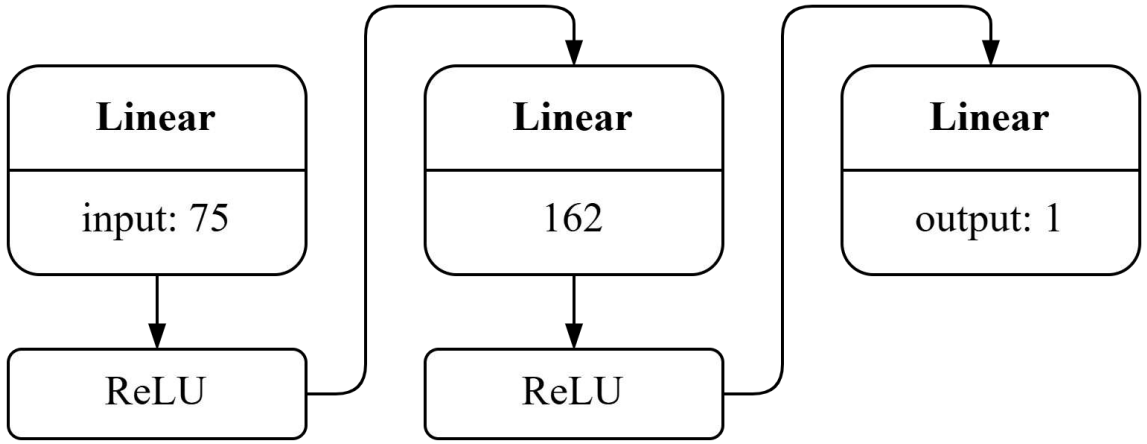
Размер вектора действия: 57

Вознаграждение (R_t)

Победа	+0,5
Взятие шашки	+0,1
Поражение	-1
Ничья	0

НЕЙРОННЫЕ СЕТИ ДЛЯ ИГРЫ «ШАШКИ 6x6»

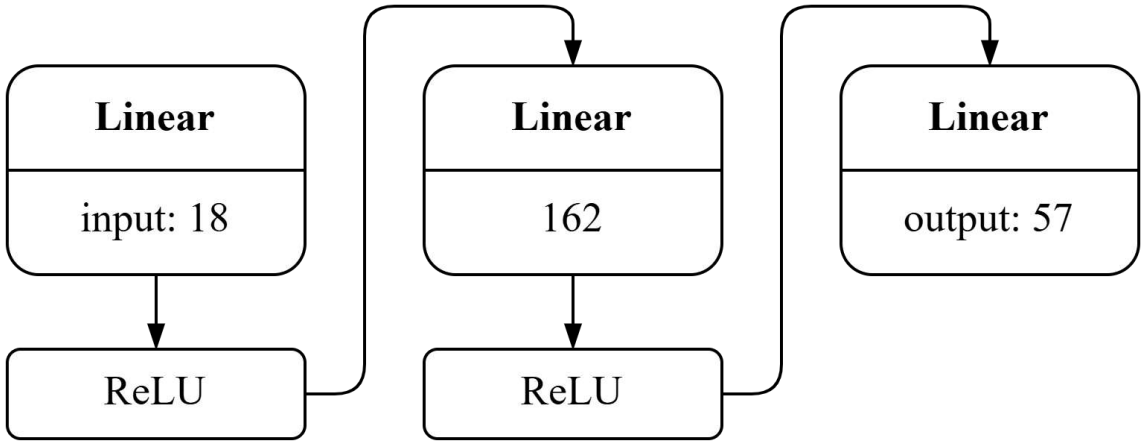
С одним выходом



Вход: пара состояние-действие (S_t, A_t)

Выход: качество одного действия $Q(S_t, A_t; \theta)$

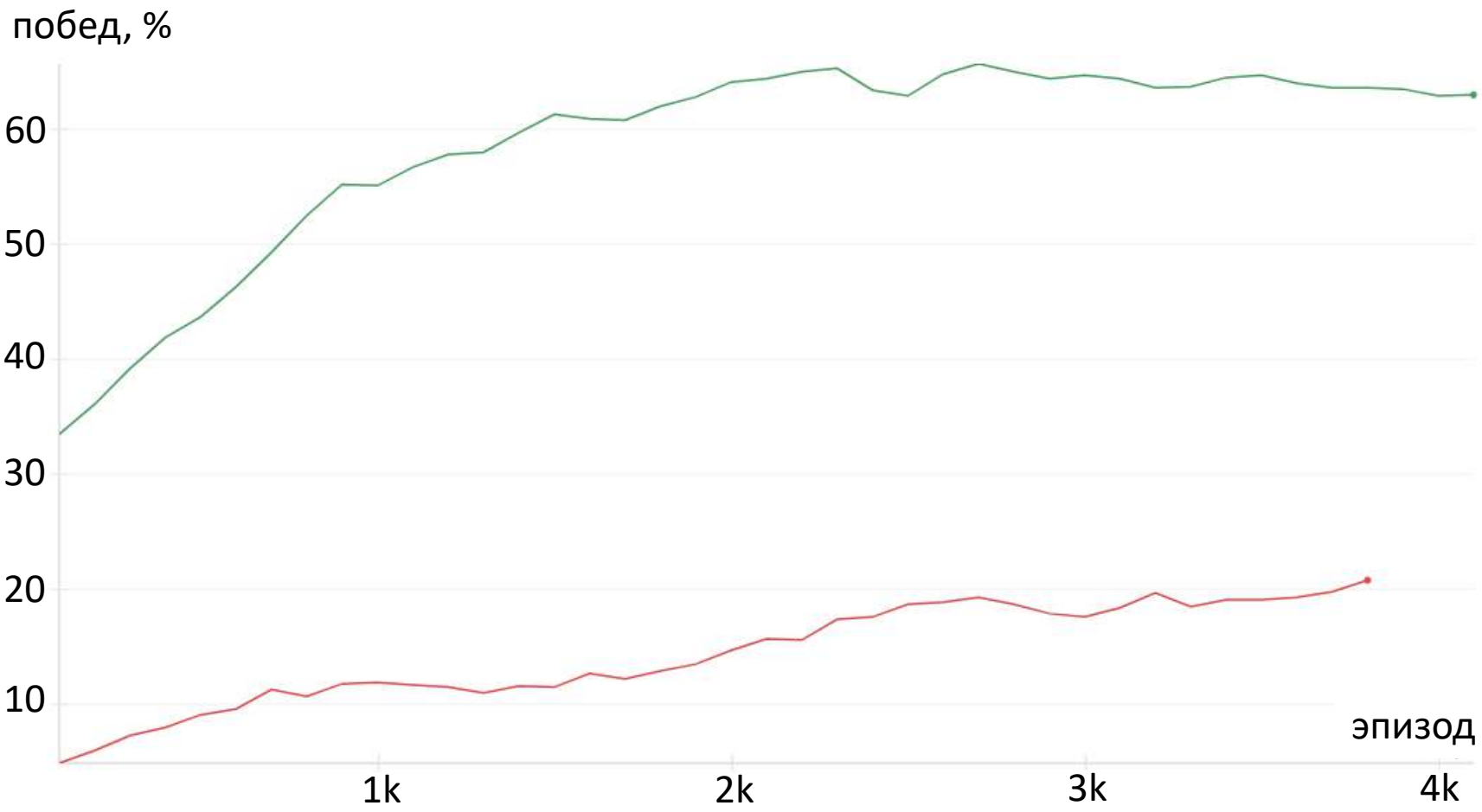
Со многими выходами



Вход: состояние S_t

Выход: качество каждого действия $Q(S_t, A; \theta)$

РЕЗУЛЬТАТЫ ОБУЧЕНИЯ СО СЛУЧАЙНЫМ ПРОТИВНИКОМ



Сравнение эффективности моделей с одним и многими выходами

Гипер-параметр обучения	Значение
Доля случ. действий (ε)	0,1
Скорость обучения	0,0025
Вознагр. за победу	0,1
Вознагр. за взятие шашки	0,1
Вознагр. за поражение	-0,7
Вознагр. за ничью	-0,5
Размер скрытого слоя	162

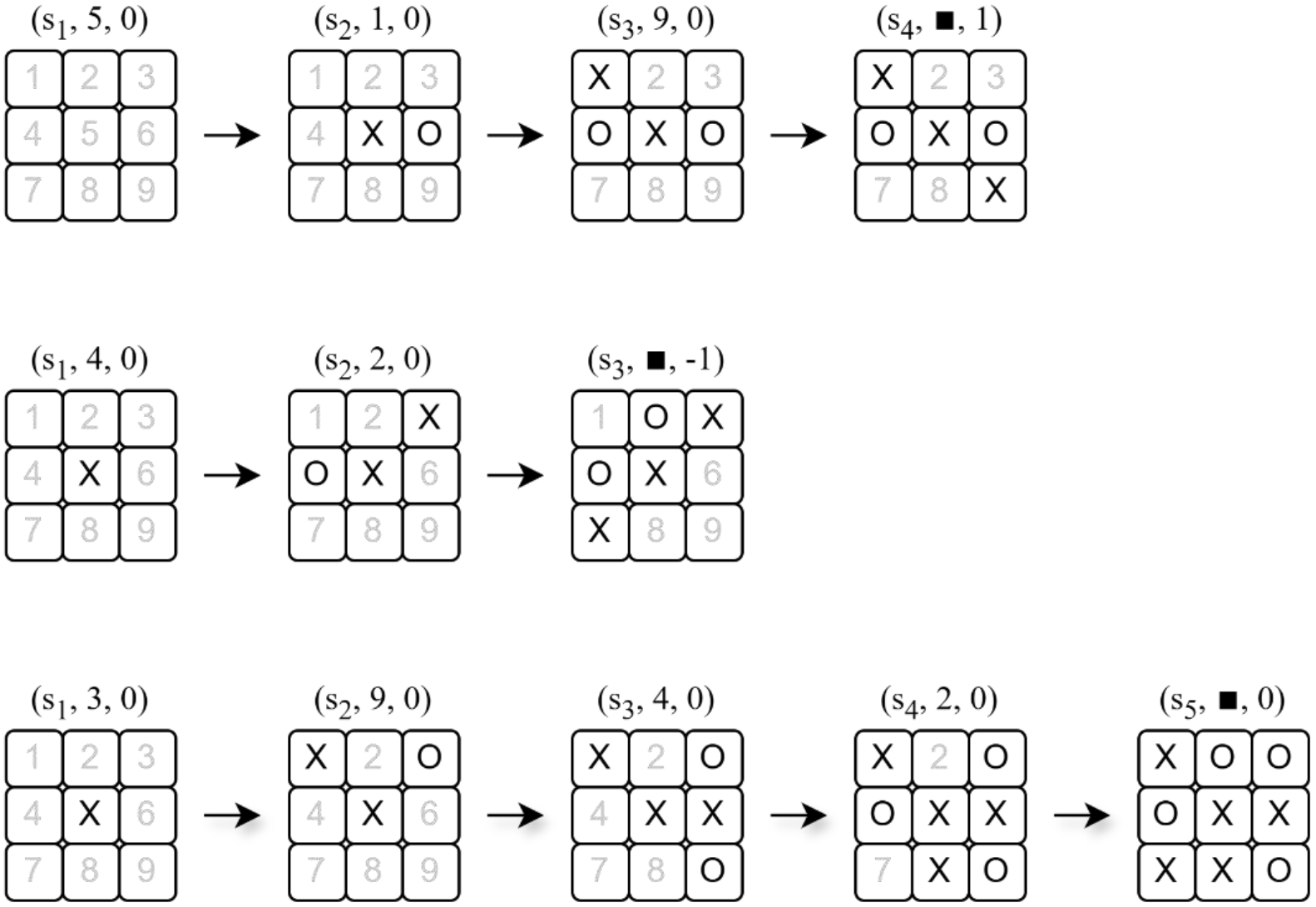
ОСНОВНЫЕ РЕЗУЛЬТАТЫ

1. Проведен анализ предметной области
2. Спроектирована и реализована система глубокого обучения с подкреплением на основе нейросетевой модели с одним выходом для игры «Крестики-нолики»
3. Реализована система глубокого обучения с подкреплением на основе нейросетевой модели со многими выходами для игры «Крестики-нолики»
4. Проведен сравнительный анализ двух архитектур для игры «Крестики-нолики»
5. Разработан графический интерфейс для игры «Крестики-нолики», предусматривающий игру человека с обученным агентом
6. Спроектирована и реализована система глубокого обучения с подкреплением на основе нейросетей с одним и многими выходами для игры «Шашки 6×6»
7. Проведен сравнительный анализ двух архитектур для игры «Шашки 6×6»

Исходный код на GitVerse

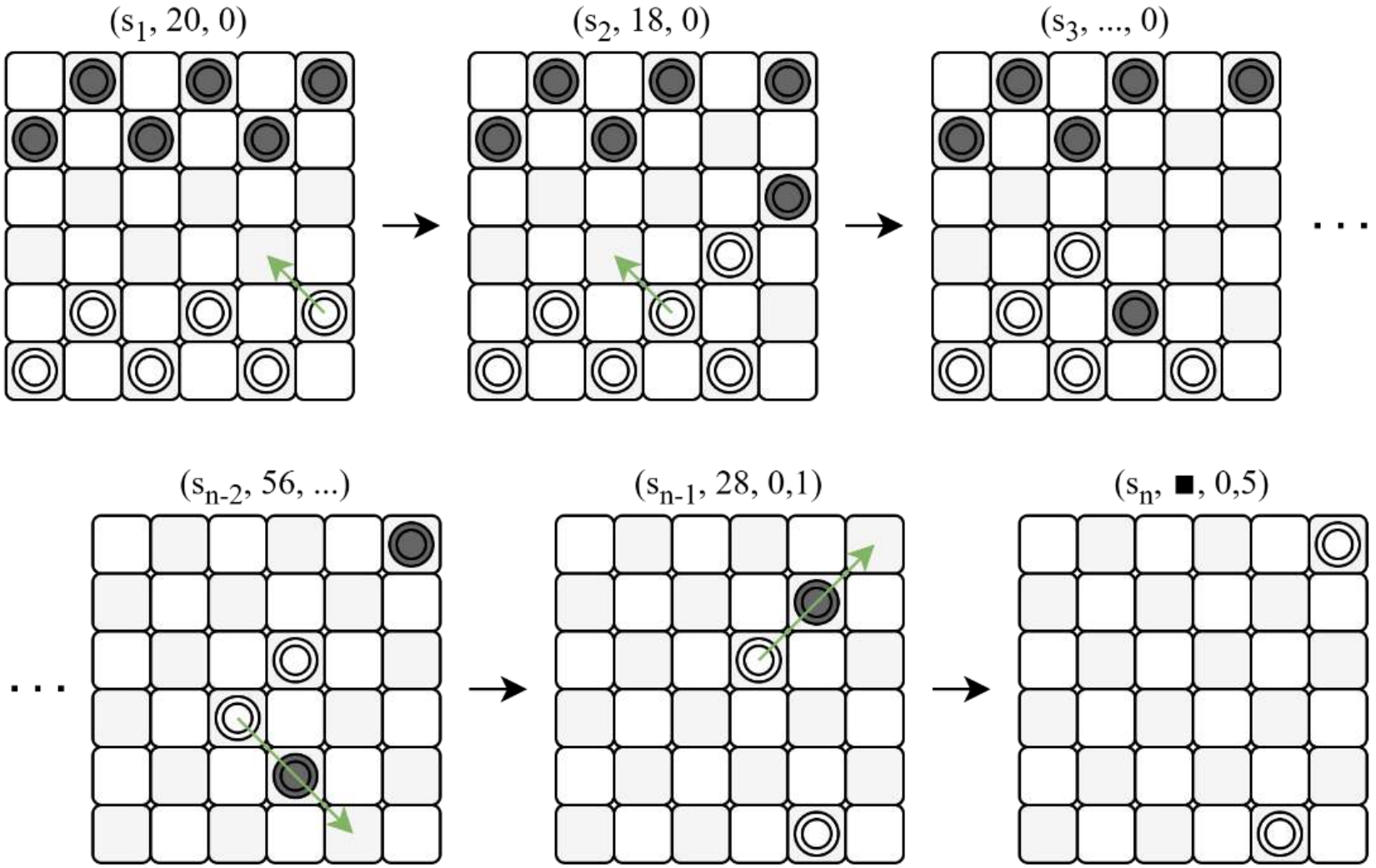
ДОПОЛНИТЕЛЬНЫЕ СЛАЙДЫ

ПРИМЕРЫ ЭПИЗОДОВ ИГРЫ «КРЕСТИКИ-НОЛИКИ»



*Каждый шаг представляется кортежем (s_t, a_t, r_t)

ПРИМЕР ЧАСТИ ЭПИЗОДА ИГРЫ «ШАШКИ 6×6»



*Каждый шаг представляется кортежем (s_t, a_t, r_t)

СРЕДСТВА РАЗРАБОТКИ

1. **Язык программирования**: Python 3.12.9
2. **Редактор исходного кода:** Visual Studio Code
3. **Библиотеки**:
   - torch 2.7.0
   - Flask 3.1.0
   - Weights & Biases 0.19.11

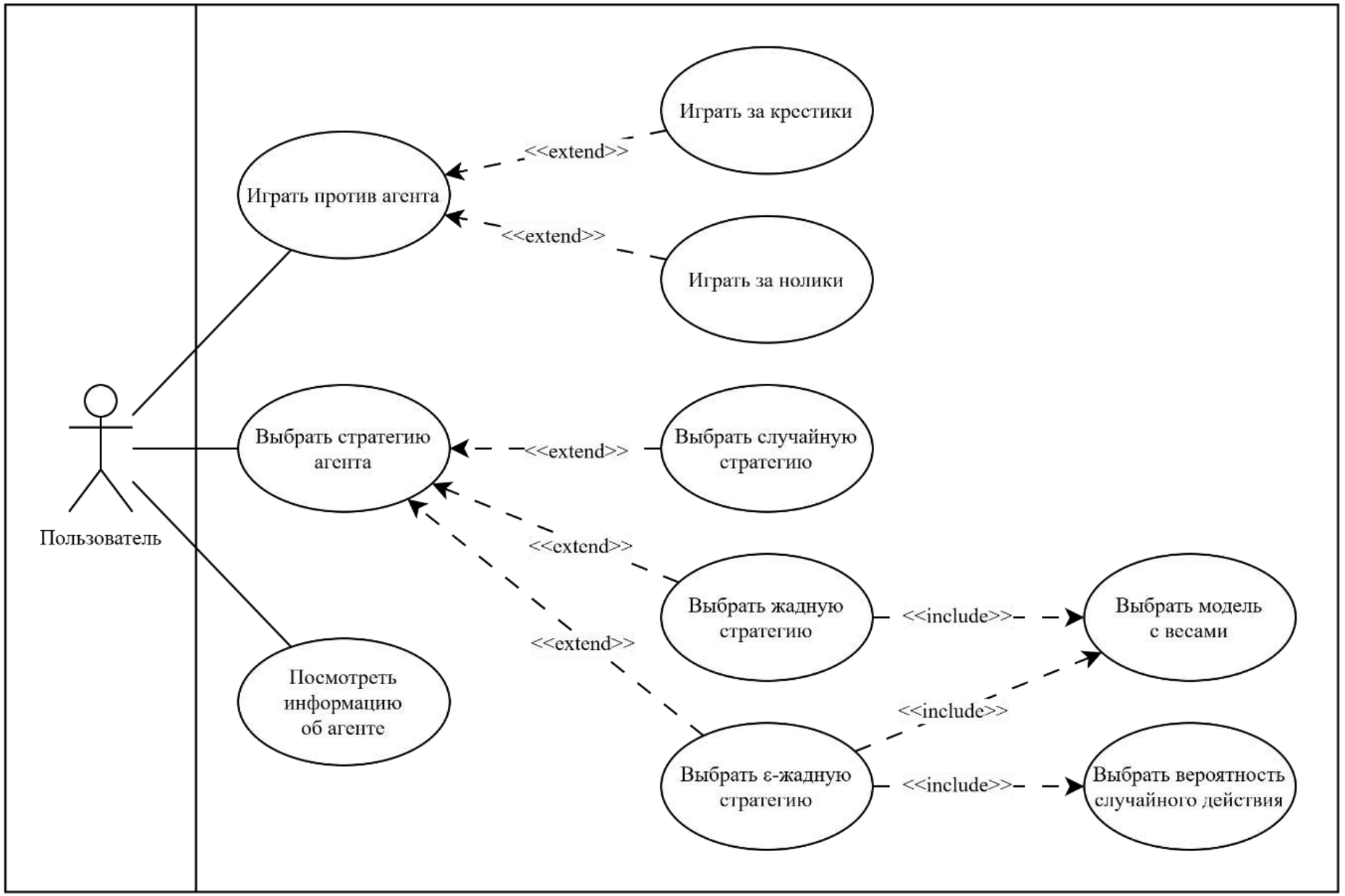



ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ