ОРИГИНАЛЬНАЯ СТАТЬЯ

УДК 004.8:339.178(045)

Торговые агенты на основе обучения с подкреплением

Павел Максимович Требоганов, *студент Факультета прикладной математики и информационных технологий,* ***Финансовый университет,*** *Москва, Россия* ***/ Pavel M. Treboganov,*** *student, Faculty of Applied Mathematics and Information Technology Financial University, Moscow, Russia*
paveltreboganov@yandex.ru

АННОТАЦИЯ

В статье рассматриваются различные подходы к созданию торговых агентов обучения с подкреплением на основе Q-обучения. Направление использования обучения с подкреплением не так распространено в задачах управления активами, как прогнозирование цены актива с использованием методов машинного обучения, однако их можно адаптировать под задачи управления активами. В рамках статьи рассматриваются реализации как дискретного Q-обучения, так и метода с использованием нейронных сетей. В рамках экспериментов агенты были обучены на реальных котировках некоторых акций из индекса S&P500 и сравнивались результаты как на обучающих, так и на тестовых периодах. Также была модифицирована функция наград в реализации с нейронными сетями, которая позволила сделать выводы об обучаемости агентов.
Ключевые слова: *искусственный интеллект; обучение с подкреплением; Q-обучение; Q-сети; нейронные сети; трейдинг; агент; машинное обучение; управление активами*

Для цитирования: **Требоганов П. М. Торговые агенты на основе обучения с подкреплением.** ***Научные записки молодых исследователей.*** **2020;8(6):46–57.**

ORIGINAL PAPER

Trading Agents with Reinforcement Learning

ABSTRACT

The article discusses various approaches to creating trading agents with reinforcement learning, based on Q-learning. The use of reinforcement learning is not as common in asset management tasks as a prediction the price of an asset using machine learning methods, but RL can be adapted to asset management tasks. The article deals with the implementations of discrete Q-learning and Q-learning with neural networks. As part of the experiments, the agents were trained on actual prices of some stocks from the S&P500 index. The results were evaluated during the training and test period. The reward function has also been modified in an implementation containing neural networks, which allowed to conclude the trainability of agents.

Научный руководитель: ***Коротеев М.В.,*** *кандидат экономических наук, доцент, доцент Департамента анализа данных, принятия решений и финансовых технологий, Финансовый университет, Москва, Россия / Scientific Supervisor:* ***Koroteev M.V.,*** *Cand. Sci. (Econ.), Associate Professor, Department of Data Analysis, Decision Making and Financial Technologies, Financial University, Moscow, Russia.*

Keywords: *artificial intelligence; reinforcement learning; Q-learning; Q-networks; neural networks; trading; agent; machine learning; asset management*

For citation: **Treboganov P. M. Trading agents with reinforcement learning.** ***Nauchnye zapiski molodykh issledovatelei = Scientific notes of young researchers*****. 2020;8(6):46–57.**

Введение

Все больше места в финансовом секторе занимает машинное обучение и наука о данных. Банки и инвестиционные компании борются между собой за максимальную прибыль в условиях постоянной «гонки вооружения». Для того чтобы выигрывать на рынке, необходимо либо быть быстрее всех, либо использовать стратегии, отличающиеся от стратегий конкурентов. Для достижения этой цели необходимо использовать различные методы, зачастую наименее исследованные и распространенные в мире. Именно такую функцию могут выполнять методы машинного обучения при грамотном использовании их в стратегиях.

Машинное обучение можно рассматривать как набор методов или алгоритмов, для которых характерно не прямое решение задачи, а обучение в ходе решения множества схожих задач.

Одним из наименее распространенных на данный момент направлением машинного обучения является обучение с подкреплением. Однако данное направление может дать преимущество на рынке тем, кто сумеет грамотно воспользоваться методами, ведь на рынке зачастую выигрывает тот, кто привнес что-то новое.

В данной работе описаны, рассмотрены и проанализированы несколько реализаций агентов обучения с подкреплением на основе метода под названием *Q*-обучение.

Q-обучение

Метод впервые был представлен Кристофером Уоткинсом в 1989 г.

Q-обучение может рассматриваться как одна из форм обучения на основе временных различий (TD, или temporal-difference, обучение) [1].

Обучение с временными различиями (TD) не зависит от модели и учится на основе опыта. Однако TD может учиться на неполных эпизодах, и поэтому, в отличие от метода Монте-Карло, нам не нужно отслеживать эпизод до конца.

Введем основные обозначения: $s \in S$ — состояния среды; $a \in A$ — действия; $r \in R$ — награды; S_t, A_t, R_t — состояния, действия, награды в момент времени t; $V(s)$ — функция полезности (Value-function); $Q(s,a)$ — полезность действия (action-value function) a, похожа на $V(s)$, но оценивает ожидаемую отдачу от пары состояние-действие (s,a) [2].

Ключевая идея TD-обучения состоит в обновлении функции полезности $V(S_t)$ в соответствии с оцененной наградой $R_{t+1} + \gamma V(S_{t+1})$:

$$V(S_t) \leftarrow V(S_t) + \alpha\left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right).$$

То же самое для функции полезности действия [3]:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \\ + \alpha\left(R_{t+1} + \gamma VQ(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\right).$$

Метод Q-обучения вычисляет ценность действия (Q-value) и обновляет его в соответствии с уравнением оптимальности Беллмана [4]:

$$Q_\pi(s,a) = R(s,a) + \gamma \sum_{s` \in S} P^a_{ss`} \max_{a` \in A} Q_*(s`,a`).$$

Ключевой момент заключается в том, что при оценке того, что является следующим действием, он не следует текущей политике, а скорее, принимает лучшее значение *Q*.

Таким образом, правило обновления Q-функции будет следующим:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left(r + \gamma \max_{a` \in A} Q(s',a')\right).$$

Глубокие Q-сети

Вместо того чтобы искать значение функции полезности действия напрямую, можно его аппроксимировать с помощью глубоких нейронных сетей, которые выступают в качестве нелинейного аппроксиматора функции. Обозначим аппроксимированную функцию как $Q(s,a,\theta_i)$, где θ_i — веса Q-нейронной сети на *i*-й итерации [5].

Алгоритмы обучения с подкреплением при использовании нейронных сетей могут оказаться нестабильными, например, из-за корреляции

в последовательностях наблюдений. Эту проблему решают «опытом переигровки»: в выборке сохраняются старые наблюдения и из них случайно выбираются данные.

Функция ошибки задается как:

$$L_i(\theta_i) = E_{(s,a,r,s')\sim U(M)}\left[\left(r + \gamma \max_{a'} Q\left(s', a', \theta_i^-\right) - Q\left(s, a, \theta_i\right)\right)^2\right],$$

где M — набор примеров $(s_t, a_t, r_{t+1}, s_{t+1})$; θ_i — веса «онлайн» Q-сети (обновление весов происходит на каждой итерации); θ_i^- — веса «офлайн» сети (обновление происходит раз в C шагов), эта сеть и является целевой.

Таким образом, необходимая для обновления Q-функции часть выглядит так:

$$Y_t = r_{t+1} + \gamma \max_a Q\left(S_{t+1}, a, \theta_i^-\right).$$

Агент с дискретным Q-обучением

Так как агент обучения с подкреплением учится на собственном опыте, самое важное — правильно обозначить действия, ограничения и настроить корректную логику взаимодействия агента со средой [6]. Рассматривая агентов, нацеленных на трейдинг, достаточно просто сформулировать изначальную цель: «необходимо увеличивать количество денег». Однако на практике зачастую нужно подходить более тонко к постановке и конструированию цели [7].

Рассмотрим обычное Q-обучение с дискретной таблицей Q-функции для обучения агента трейдингу на одном активе, в данном случае на дневных ценах закрытия.

Для начала должны быть обозначены действия: агент в каждый момент времени (т.е. каждый день) может принимать решение о покупке пакета акций фиксированного размера, продаже такого пакета или бездействии. Размер пакета акций управляется параметром BUY_SELL_NB, который по умолчанию установлен на 100 акций за действие.

С целью приближения к реальной ситуации, ограничения свободы действий агента и возможности работать с дискретной Q-функцией необходимо наложить ограничения:

1) агент не может купить акции, если у него не хватает денег или количество акций больше теоретического максимума, заданного исходя из желаемой размерности Q-функции и размера пакета акций;

2) агент не может продать акции, если у него нет купленных. Таким образом, вставать в короткую позицию запрещено.

Для возможности работы с таблицей Q-функции необходимо состояния среды сделать также дискретными. Поэтому вместо значения прироста цены акции за последний день используется индикатор:

1) $\text{рост} < -1\% : I = 0$;
2) $-1\% \le \text{рост} \le 0\% : I = 1$;
3) $0\% < \text{рост} \le 1\% : I = 2$;
4) $\text{рост} > 1\% : I = 3$.

Таким образом, в алгоритме используется информация только о последнем изменении цены актива.

Совместно с этим индикатором для определения состояния среды в момент t в данном агенте используется дискретное представление уровня денег и количества акций. Уровень денег определяется как

$$\left[\frac{\text{текущее количество денег} \times \text{количество уровней денег}}{\text{максимальное количество денег}}\right]$$

и если текущее количество превышает максимальное количество, берется максимальный уровень). Уровень количества акций определяется как

$$\left[\frac{\text{текущее количество акций} * \text{количество уровней акций}}{\text{максимальное количество акций}}\right].$$

Используя эти значения, можно задать индексы Q-таблицы, не слишком повышая ее размерность. Индекс высчитывается следующим образом:

$$I \times \text{количество уровней акций} \times$$
$$\times \text{количество уровней денег} +$$
$$+ \text{уровень денег} \times \text{количество уровней акций} +$$
$$+ \text{количество акций}.$$

В каждой ячейке Q-таблицы находятся значения Q-функции для трех обозначенных действий. Изначально Q-таблица заполнена нулями, и действия выбираются случайно.

Одна из самых важных частей агента — функция награды. От нее напрямую зависит то, к чему будет стремиться агент. В рассматриваемом агенте функция наград рассчитывается таким образом:

$$\text{текущее количество денег} +$$
$$+ \text{текущее количество акций} \times \text{стоимость акции}.$$

Также агент штрафуется, если не смог совершить сделку (вышел за ограничения).

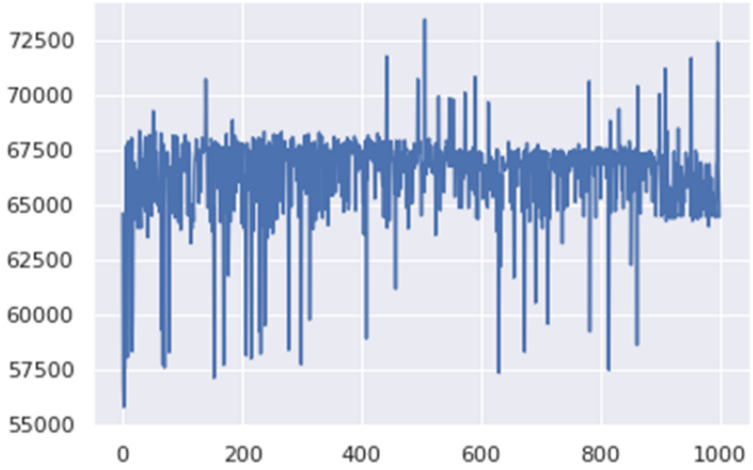

Рис. 1. **Результаты агента при итерациях обучения**

Источник: составлено автором.

Обновление Q-функции происходит согласно действию

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left(r + \gamma \max_{a` \in A} Q(s',a')\right).$$

В режиме тестирования агента Q-функция не обновляется, однако требуется логика выбора действия агента. При максимальном значении Q-функции у одного действия выбирается именно оно, однако если таких действий несколько, то случайным образом выбирается одно из них.

Рассмотрим результаты обучения агента, которое происходило на исторических ценах закрытия акций INTEL в 2013–2017 гг. с параметрами:

- начальное количество денег = 50 000 (BASE_CASH);
- максимальное количество денег = 100 000 (MAX_CASH);
- количество акций в одной сделке = 100 (BUY_SELL_NB);
- количество уровней денег = 100 (CASH_NB_VAL);
- количество уровней количества акций = 10 (SHARENB_NB_VAL);
- штраф за невыполненное действие = –10 (BAD_ACTION_REWARD);
- скорость обучения α = 0,001;
- коэффициент дисконтирования γ = 0,9;
- количество итераций = 1000.

На *рис. 1* отображены результаты агента по итерациям при начальных 50 тыс. в распоряжении агента.

Средний результат агента на тренировочной выборке по итерациям составил 66 399 долл. США, или +32,8%. Также, исходя из графика результатов агента по итерациям, можно увидеть, что поведение агента не стремится к одной стратегии при обучении. Причиной этому может быть наличие выбора случайных действий для исследования «окружающей среды». Также могут оставаться неизменными какие-то ячейки Q-таблицы, и из них также выбирается действие случайно. На *рис. 2* отображены активы агента на обучающей выборке по дням.

На *рис. 3* отображены сделки, совершенные агентом, наложенные на цену акции.

При рассмотрении графиков можно отметить, что агент на тренировочной выборке близок к графику цены актива, однако расхождения есть. На тренировочной выборке сделок совершено достаточно много, количество покупок и продаж совпадает. В данном алгоритме не учитывается стоимость транзакции, что может достаточно сильно исказить результаты в отличие от реальной торговли.

Алгоритм смог достичь положительной доходности на тренировочном периоде, но также необходимо оценить результаты на данных, не показанных агенту при обучении. На *рис. 4* отображены активы агента на тестовой выборке по дням, а на *рис. 5* показаны соответствующие сделки, наложенные на цену актива.

За два года получилось достигнуть результата +11 666 долл. США, или +23,33%. Из графиков видно, что агент пытался получить доходность выше, чем у самой акции, однако это происходило с переменным успехом. Если следовать стратегии BAH (купить акции и не совершать никаких больше действий) результат на тренировочной выборке составил бы +146,7%, а на тестовой +40,84%. Однако положительная доходность при торговле на дневных данных

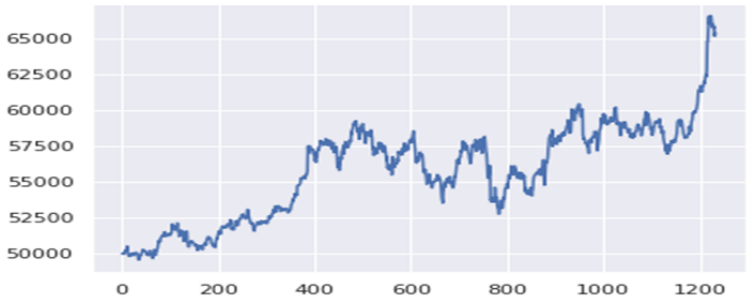

Рис. 2. **Суммарная стоимость активов при запуске агента на тренировочной выборке**

Источник: составлено автором.

Рис. 3. **Действия агента на тренировочной выборке, наложенные на цену акции**

Источник: составлено автором.

дается не каждому трейдеру, а простейший агент получил небольшой, но рост активов в своем распоряжении за два года.

Неустойчивость алгоритма по отношению к параметрам

Изначально, при покупке, условия про выход за максимальное количество не было, что приводило к постоянным покупкам растущей акции, с отсутствием сделок на продажу. Агент показывал хорошую доходность на тренировочной выборке, однако, если на тестовой выборке наблюдался медвежий тренд, то агент также только покупал и уходил в отрицательную доходность.

После попыток обучения агента на разных акциях стало ясно, что полученный алгоритм получился достаточно требовательным к выбору параметров, особенно таких, как изначальное количество денег и максимальное количество денег. Например, если запустить обучение на акции AAPL при таких же параметрах, алгоритм на тестовых данных не будет предпринимать никаких действий. Причиной этого может служить то, что акция выросла слишком сильно за несколько лет, из-за чего алгоритм штрафовал

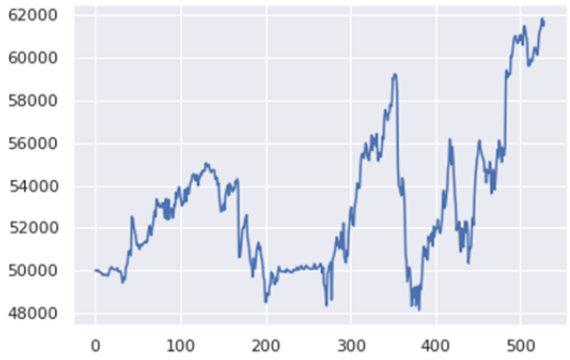

Рис. 4. **Суммарная стоимость активов при запуске агента на тестовой выборке**

Источник: составлено автором.

агента за неудачные попытки купить (потому что не хватает денег) и попытки продать (поскольку ничего не было куплено).

Так как размерность и индексы Q-таблицы зависят от значений параметров CASH_NB_VAL, SHARENB_NB_VAL и количества возможных значений индикатора I, то при увеличении изначального и максимального количества денег, до 500 000 и 1 000 000 соответственно, можно потерять необходимую точность алго-

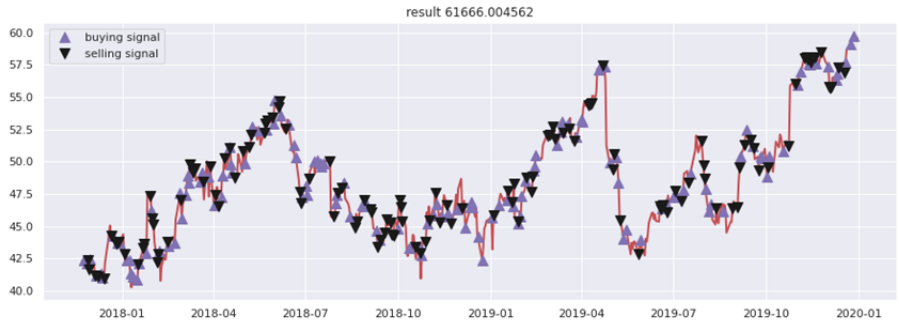

Рис. 5. **Действия агента на тестовой выборке, наложенные на цену акции**

Источник: составлено автором.

ритма, ведь он будет считать, что 509 999 и 500 000 дает один уровень денег. Таким образом, при увеличении доступных денег потребуется изменение других параметров, а при их изменении необходимо пристально следить за функцией получения индекса в q-таблице.

Агенты на основе глубокого Q-обучения

Все проблемы, описанные выше, связаны с дискретностью состояний и самой Q-функции. Поэтому имеет смысл перейти к реализации Q-обучения, позволяющей работать с бесконечным количеством состояний. В таком случае на помощь приходят глубокие Q-сети. Они позволяют рассматривать состояние среды без изощрений с использованием индикатора, используя действительное значение изменения цены. Также удается избавиться от проблем с выходом денег и количества акций за ограничения.

Рассмотрим реализацию агентов, полученных с использованием глубокого Q-обучения, двойного глубокого Q-обучения, представленную Дереком Сноу в рамках статьи Machine Learning in Asset Management [8].

Принцип стратегии напоминает агента, описанного выше. У агента есть начальная сумма денег в распоряжении, и он может принимать решение о выполнении одного из трех действий: купить акцию, продать акцию, ничего не делать. Схожесть агентов, можно сказать, заканчивается на этих принципах.

Дальнейшее описание агента можно продолжить с определения состояний среды. В оригинальной программе состояние среды определяется как изменение цены за последние несколько дней, а количество дней определяется параметром window_size, который имеет значение 30.

Автором [8] функция награды обозначена как

$$\frac{\text{текущее количество денег} - \text{изначальное количество денег}}{\text{изначальное количество денег}}.$$

После каждого действия информация о состоянии среды в момент *t*, выбранное действие, награда и состояние среды в момент *t* добавляется в память агента. Накопленная информация используется для обучения Q-функции, представленной полносвязной нейронной сетью из двух слоев, размерность которых составляет 256 и 3 соответственно.

В качестве функции ошибки для обучения нейронной сети используется среднеквадратичная ошибка между «реальными» и предсказанными значениями Q-функции.

«Реальные» значения Q-функции задаются в виде

$$r + \gamma * \max\left(Q\left(s', a\right)\right),$$

где Q — результат предсказания нейросети.

Накапливается 32 пары реальных и предсказанных значений. Количество пар задается параметром batch_size. После этого запускается итерация обучения сети. Автор [8] в алгоритме не добавляет элемент $\gamma * \max\left(Q\left(s', a\right)\right)$, если количество денег превысило изначальные инвестиции. Еще отметим, что в алгоритме не была предусмотрена возможность тестирования, получения результатов с учетом последней позиции по акциям, поэтому соответствующие изменения были внесены в код.

При запуске агента в тестовом режиме, как и в прошлом агенте, обучения не производится.

В данных агентах следующее состояние среды не зависит от действия: состояние среды — это последние 30 изменений цены, и мы считаем, что наши решения и действия не влияют на рынок.

Автором алгоритма не было представлено экспериментов, что потребовало проведения собственных исследований на реальных исторических данных.

Для оценки метода были обучены 58 агентов на соответствующих 58 лидирующих акциях, состоящих в индексе S&P500. Тренировочная выборка включала в себя период 2013–2017 гг., а тестирующая — 2018–2019 гг. Обучение производилось с параметрами:

- изначальное количество денег = 10000 (initial_money);
- количество изменений цены в состоянии среды = 30 (window_size);
- количество итераций — 200.

Для начала рассмотрим результат агентов, использующих Q-обучение. Для анализа были выбраны такие параметры, как:

- доход за тренировочный период (5 лет) — result train;
- количество сделок на покупку акции за тренировочный период — count buy train;
- количество сделок на продажу акции за тренировочный период — count sell train;
- позиция (в количестве акций) на конец тренировочного периода — last pos train;
- время обучения агента;
- доход за тестовый период (5 лет) — result test;
- количество сделок на покупку акции за тестовый период — count buy test;
- количество сделок на продажу акции за тестовый период — count sell test;
- позиция (в количестве акций) на конец тестового периода — last pos test.

Выведем первые 10 агентов по доходу за тренировочный период, результаты которых отображены в *табл. 1*.

Из таблицы видно, что положительный результат на обучающей выборке не гарантирует положительный результат на тестовой.

Можно заметить, что в показанных акциях существуют длинные позиции на окончание периодов. Это достаточно неожиданный результат, если учесть, что функцией наград агента является разница между текущим и изначальным количеством денег. С другой стороны, это выглядит логичным, ведь у таких акций наблюдался бычий тренд, как минимум на тренировочной выборке.

Результаты первой десятки достаточно обнадеживающие, однако были также получены агенты, которые даже на тренировочной выборке показывают отрицательную доходность.

Рассмотрим 10 худших по доходности агентов, результаты которых отображены в *табл. 2*.

Из представленных таблиц можно сделать предположение, что связь между доходностью на тренировочной и тестовой выборках достаточно слабая, что подтверждается коэффициентом корреляции доходностей, равным 0,4999.

Пять агентов из 58 показали отрицательную доходность. Еще 19 агентов на обучающей выборке не смогли достичь доходности в 1%. Средняя доходность на обучающей выборке составляет 1870 долл. США, или 18,7% со стандартным отклонением в 3769, что достаточно плохо для 5 лет. На тестовой выборке средняя доходность составляет 660 долл. США, или 6,6% за два года. Таким образом, полученные агенты в среднем не достигают приемлемых результатов.

Рассмотрим графики покупок доходного актива (AAPL) на тренировочном периоде (*рис. 6*) и тестовом периоде (*рис. 7*).

На конец тренировочного периода в деньгах находилось всего 1180 долл. США, или 11,8% от изначального количества денег. Выявленное означает: агент решил, что продавать этот актив не так выгодно, как покупать; однако так как функция наград не учитывает позицию в акциях, агенту требуется периодически продавать актив. На тестовом периоде лучше заметно, насколько неохотно агент продает активы. На уровне 200 дня торгов заметно, как агент перекупил активы, ожидая роста, и как неохотно торговал на медвежьем тренде.

Агенты на основе глубокого Q-обучения с измененной функцией наград

В рассмотренной реализации получились достаточно случайные результаты, которые не могут нам однозначно показать, обучается ли агент или действует случайно. В качестве инструмента, позволяющего выделить направление стремления агента, можно выделить функцию наград. Ранее функция наград содержала информацию только о количестве денег в данный момент, поэтому приблизим ее к знакомой нам функции из дискретной реализации. Теперь функция наград выглядит следующим образом:

$$\frac{\text{текущее количество денег}_t + \text{позиция}_t \times \text{цена акции}_t - \text{изначальное количество денег}}{\text{изначальное количество денег}}.$$

Таблица 1

10 лучших агентов по результатам на тренировочной выборке (Q-обучение)

	Result train	Count buy train	Count sell train	Last pos train	Time train	Result test	Count buy test	Count sell test	Last pos test
AVGO	18 553,8	473	362	111	209	3149,7	183	146	37
BA	14 539,7	471	378	93	213	172,0	182	159	23
GOOG	12 844,7	312	298	14	208	500,9	131	128	3
AAPL	10 766,3	409	290	119	208	4962,2	176	130	46
ACN	8228,7	390	270	120	216	3668,8	158	102	56
MMM	4279,5	265	212	53	215	–994,1	109	61	48
HD	4258,5	469	413	56	219	1091,5	172	158	14
GOOGL	3467,8	442	438	4	220	2249,2	181	178	3
PM	3058,7	515	376	139	215	791,1	247	142	105
JNJ	2589,2	446	385	61	222	1274,3	207	133	74

Источник: составлено автором.

Таблица 2

10 худших агентов по результатам на тренировочной выборке (Q-обучение)

	Result train	Count buy train	Count sell train	Last pos train	Time train	Result test	Count buy test	Count sell test	Last pos test
PEP	32,0	130	130	0	305	35,8	66	66	0
SLB	31,8	270	270	0	272	–4,2	135	135	0
MO	19,7	340	340	0	260	–21,7	150	150	0
KO	8,6	293	293	0	242	24,8	138	138	0
CMCSA	7,0	161	161	0	242	–2,4	43	43	0
GE	–1,4	317	317	0	276	–5,0	125	125	0
UTX	–15,1	351	351	0	231	185,8	166	164	2
WFC	–19,5	271	271	0	257	–0,6	120	120	0
AMZN	–167,5	205	205	0	259	1142,4	105	105	0
GILD	–573,6	456	319	137	218	–220,1	242	133	109

Источник: составлено автором.

Ожидается, что данная функция наград не будет принуждать агентов продавать активы, даже если это не выгодно.

Рассмотрим полученные результаты для алгоритма, использующего Q-обучение, отображенные в *табл. 3 и 4*.

Шесть агентов из 58 показали отрицательную доходность, 11 агентов на обучающей выборке не смогли достичь доходности в 1%. Еще 13 агентов показали доходность менее 10% на тренировочной выборке.

На тестовой выборке 12 агентов показали отрицательную доходность, 15 не достигли 1% доходности и 14 агентов показали доходность более 1%, но менее 10%.

Средняя доходность на тренировочной составляет 6496 долл. США, или 64,96% со стандартным отклонением в 16 928, а на тестирующей выборке —

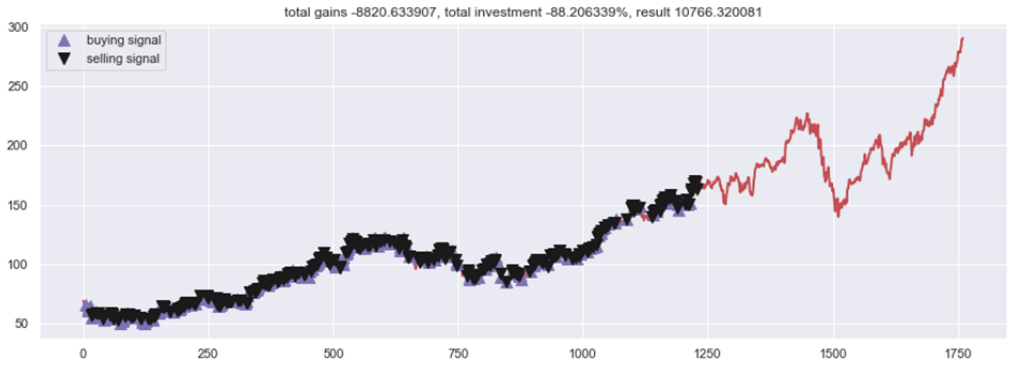

Рис. 6. **Действия агента на тренировочном периоде на акции AAPL**

Источник: составлено автором.

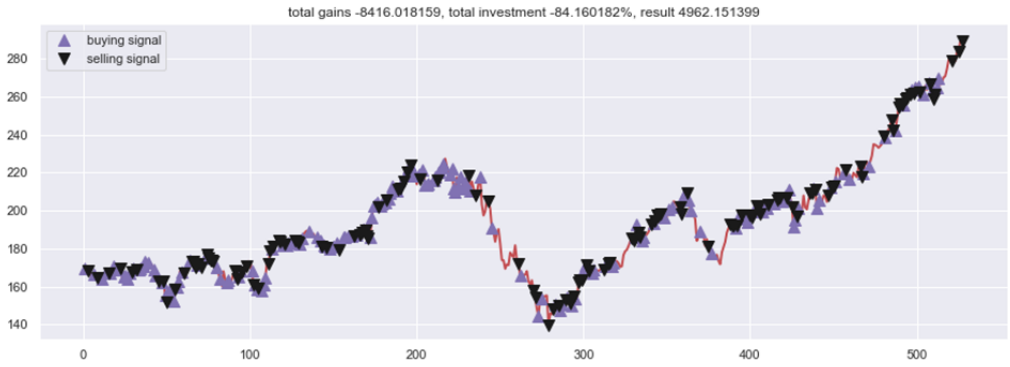

Рис. 7. **Действия агента на тестовом периоде на акции AAPL**

Источник: составлено автором.

996 долл. США, или 9,96% со стандартным отклонением в 1768. Количество агентов с доходностью на тренировочном периоде менее 1% уменьшилось по сравнению с агентами Q-обучения со старой функцией наград. Данные результаты немного превосходят результаты алгоритма со старой функцией наград, но проблема с разбросом доходностей не ушла.

Успех 10 лучших агентов кроется в том, что на сильно выросших акциях агенты научились скупать акции в самом начале и переставали действовать позже. Достаточно наглядно это видно на графиках действий агента, обученного на акции AAPL. Действия в тренировочном периоде отображены на *рис. 8*, а в тестовом — на *рис. 9*.

Такие агенты показывают хорошие цифры в результатах, но это обусловлено лишь на опыте агента, связанном с тем, что акция сильно растет.

От случайности результатов обучения уйти не удалось, однако положительные выводы все же можно сделать. Смотря на агентов, научившихся скупать активы в начале, можно однозначно сказать, что метод обучения работает, и проблема кроется не в методе, а в организации взаимодействия агента и среды.

Q-обучение: дискретный подход против Q-сетей

При сравнении дискретного подхода и глубоких Q-сетей можно выделить несколько важных особенностей, влияющих на результаты обучения и работы агентов.

В дискретном подходе необходимо тщательно подойти к построению архитектуры Q-функции. Для этого требуется привести состояние среды к дискретному виду и однозначно определить способ хранения информации в Q-таблице. При таком подходе достаточно сложно построить гибкую систему, на вход которой подается только ценовой ряд. Зачастую обучение может ломаться из-за несоответствия параметров цене актива, и это может оказаться большой проблемой в работе с увеличивающимися в несколько раз активами.

Таблица 3

10 лучших агентов по результатам на тренировочной выборке (Q-обучение с измененной функцией наград)

	Result train	Count buy train	Count sell train	Last pos train	Time train	Result test	Count buy test	Count sell test	Last pos test
NVDA	118 929	612	5	607	313	1149	49	1	48
NFLX	42 987	277	5	272	311	5219	51	4	47
UNH	26 072	182	4	178	314	3340	47	2	45
AAPL	18 805	178	3	175	301	7511	62	2	60
TXN	18 245	309	5	304	313	2725	102	2	100
V	16 380	246	3	243	307	6001	90	5	85
HD	15 673	162	2	160	311	2089	58	2	56
BKNG	13 599	354	346	8	303	1472	127	127	0
MO	11 299	379	2	377	306	–1081	184	2	182
MCD	9531	126	2	124	309	2540	67	3	64

Источник: составлено автором.

Таблица 4

10 худших агентов по результатам на тренировочной выборке (Q-обучение с измененной функцией наград)

	Result train	Count buy train	Count sell train	Last pos train	Time train	Result test	Count buy test	Count sell test	Last pos test
HON	16	148	148	0	313	38	60	60	0
PYPL	11	2	2	0	307	54	3	0	3
PG	4	246	246	0	308	21	86	86	0
CSCO	3	66	66	0	304	–4	18	18	0
PFE	–3	272	258	14	312	18	108	107	1
JNJ	–6	94	94	0	313	–19	49	49	0
IBM	–14	216	216	0	313	27	79	79	0
QCOM	–69	392	392	0	310	176	177	177	0
SLB	–102	327	327	0	312	–28	138	138	0
GE	–1238	677	483	194	307	–11	278	206	72

Источник: составлено автором.

Для обоих подходов необходимо формализовать взаимодействие со средой. Это могут быть, как представленные выше стратегии с изначальным количеством денег, на каждом шаге принимающие решения о покупке продаже или удержании позиции, так и абсолютно другой тип взаимодействия: например прогнозирование цены актива в будущем. Такую логику обычно легче всего проследить в алгоритме.

Подход с глубокими Q-сетями показал себя более универсальным подходом, не требующим подбора параметров под отдельный актив для запуска обучения. Однако при исследовании и тестировании реализации этого подхода были найдены слабые места.

Одним из самых больших осложнений при работе с глубокими Q-сетями-является невозможность интерпретировать работу сетей. Это сильно осложняет поиск ошибок в коде при подходе и определении адекватности работы алгоритма. Если алгоритм работает надлежащим образом,

Рис. 8. **Действия агента на тренировочной выборке на акции AAPL (Q-обучение с измененной функцией наград)**

Источник: составлено автором.

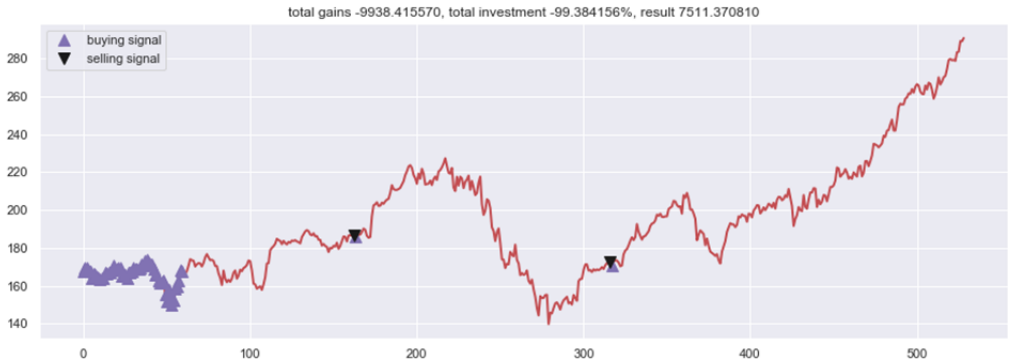

Рис. 9. **Действия агента на тестовой выборке на акции AAPL (Q-обучение с измененной функцией наград)**

Источник: составлено автором.

то необходимо подобрать функцию наград, которая будет приводить к ожидаемому результату. Другими словами, нужно очень аккуратно обозначать для агента, что такое хорошее действие (и насколько оно хорошее), а что такое плохое (и на сколько).

Как дискретная реализация, так и реализация с глубокими Q-сетями показывают себя не очень стабильно, и на одном активе могут давать совершенно разные результаты, поэтому при выборе агента очень важно проводить различные эксперименты и тесты для выявления слабых мест.

Возможные дальнейшие шаги

При рассмотренных архитектурах взаимодействия со средой и представленных параметрах алгоритма агенты показывают себя не слишком хорошо на задачах управления активами, однако сами алгоритмы обучения работают, что было продемонстрировано в работе выше.

Для создания более успешного алгоритма в качестве дальнейших исследований могут быть выделены несколько блоков:

- функция наград;
- состояние среды;
- логика разрешенных действий для агента;
- используемые данные.

Возможности для улучшения функции наград даже в существующей логике достаточно обширны. Одни из самых простых изменений могут нести в себе добавление штрафа за неудачные действия. Более сложной частью является ответ на вопрос: как можно учесть информацию о будущем результате действия в функции наград и не привести к переобучению агента? Возможно, проблема переобучения уйдет при увеличении количества данных, используемых при обучении, или при уменьшении количества нейронов в сети, аппроксимирующей Q-функцию. В приближении к реальным условиям необходимо учитывать

минимальное количество акций для покупки и комиссию за операции.

В качестве состояния среды нами были использованы последние 30 изменений цены. Самое очевидное направление для изучения — это поиск оптимального количества изменений. При этом в состояние среды можно включать достаточно много различной информации, например в дополнение к используемым данным брать изменения цены коррелирующей акции, количество сделок по дням, информацию о сезоне, дне недели. Однако чем больше параметров будет использоваться, тем сложнее будет отследить ошибки и утечки данных, которые могут помогать достигать очень хороших результатов, неповторимых на практике.

На данный момент логика взаимодействия очень ограничена: существует всего три возможных действия, и то с ограничениями. При рассмотрении результатов агентов было замечено, что некоторые из них научились избегать больших падений: у агентов тогда нет возможности встать в короткую позицию, и они просто избегают их, теряя возможность заработать. Таким образом, одним из самых первых изменений может быть добавление возможности вставать в короткую позицию. Также можно увеличить количество действий добавлением возможности выбирать количество покупаемых или продаваемых акций. Это позволит агенту быстро выходить из позиции и не терять на невозможности такого действия. Однако увеличение числа действий потребует большего количества данных для обучения.

Дневные данные достаточно давно считаются слишком непредсказуемыми, что осложняет трейдинг на них. В идеальном случае, чтобы пользоваться преимуществами очень быстрого принятия решения агентом, разумно обучить его на секундных, или вообще тиковых, данных. Однако это потребует усовершенствования реализации алгоритма для ускорения как самого обучения, так и работы в режиме выбора действия.

Выводы

В работе были рассмотрены реализации агентов на основе дискретного Q-обучения и Q-обучения с использованием нейронных сетей. Данные агенты на текущий момент показали не слишком хорошие результаты, однако это не означает, что нужно игнорировать это направление. Это указывает на то, что необходимо более тщательно подходить к построению взаимодействия агента и среды, следить за логикой обучения и используемых параметров.

Агенты с использованием нейронных сетей дают нам возможность использовать гораздо больше информации, чем агенты с дискретной реализацией. В качестве используемой информации могут выступать не только изменения цены, а множество других параметров, начиная от изменений цены другого актива, заканчивая результатами других алгоритмов. Это дает огромное пространство для исследований и возможность улучшать агентов практически бесконечно.

References

1. Sutton R.S., Barto A.G. Reinforcement Learning: An introduction. 2nd edition. April 2005. URL: http://www.incompleteideas.net/book/RLbook2020.pdf. (accessed on 22.02.2020).
2. Weng L. A (Long) Peek into Reinforcement Learning; February 2018. URL: https://lilianweng.github.io/lil-log/2018/02/19/a-long-peek-into-reinforcement-learning.html#key-concepts (accessed on 12.01.2020).
3. Maei H.R. Gradient temporal-difference learning algorithms. PhD thesis. University of Alberta; 2011. URL: https://era.library.ualberta.ca/items/fd55edcb-ce47–4f84–84e2-be281d27b16a (accessed on 15.03.2020).
4. Bellman R.E. Dynamic Programming. Princeton: Princeton University Press; 1957. URL: http://en.bookfi.net/book/677250 (accessed on 10.11.2019).
5. Jiang Z. Xu D., Liang J. A deep reinforcement learning framework for the financial portfolio management problem; July 2017. URL: https://arxiv.org/pdf/1706.10059.pdf. (accessed on 20.02.2020).
6. Doya K. Reinforcement learning in continuous time and space. *Neural computation*. 2000:219–245. URL: https://www.mitpressjournals.org/doi/10.1162/089976600300015961 (accessed on 15.02.2020).
7. Filos Angelos. Reinforcement Learning for Portfolio Management. 2018. June. URL: https://arxiv.org/pdf/1909.09571.pdf (accessed on 12.03.2020).
8. Snow D. Machine Learning in Asset Management. University of Auckland. 2019. June URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3420952. (accessed on 07.09.2019).