Оптический журнал
Нейроиконика

DOI: 10.17586/1023-5086-2022-89-08-08-23
УДК 004.93'12; 004.932.72'1; 004.832.2

Обучение динамически конфигурируемого классификатора с использованием глубокого Q-обучения

Роман Олегович Малашин[1], Арина Андреевна Бойко[2]✉

[1, 2]Институт физиологии им. И. П. Павлова РАН, Санкт-Петербург, Россия
[1, 2]Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург, Россия

[1]malashinroman@mail.ru https://orcid.org/0000-0002-2493-839X
[2]boikoaa@infran.ru https://orcid.org/0000-0001-7520-0056

Аннотация

Предмет исследования. Были рассмотрены динамические сети, которые позволяют осуществлять вычисления, обусловленные входными данными. **Цель работы.** Исследовать возможности использования методов глубокого Q-обучения для создания таких сетей в задачах компьютерного зрения. **Метод.** В современных динамически конфигурируемых системах анализа изображений принято использовать алгоритм градиентов по стратегиям. Нами предложен метод гибридного Q-обучения агента для классификации изображений с учётом ограничения на доступные вычислительные ресурсы. Мы обучаем агента воспринимать изображения через набор предварительно обученных классификаторов и хотим, чтобы результирующая динамически конфигурируемая система была способна построить вычислительный граф с учётом ограничения на количество операций и с такой траекторией, которая относится к максимальной ожидаемой точности. Агент получает вознаграждение только в том случае, если изображение распознано правильно, при этом количество возможных действий для него ограничено. Были проведены эксперименты с базой данных изображений CIFAR-10 и набором из шести внешних классификаторов, которыми учился управлять агент.

В соответствии с приведёнными экспериментами стандартный метод глубокого обучения по ценностям действий (DQN, Deep Q-Network) не позволяет агенту выучить стратегий, отличимых от случайных по точности распознавания. В связи с этим предложена архитектура Q-КНД (Q-классификатор наименьшего действия), которая аппроксимирует требуемую функцию выбора классификатора с помощью обучения с подкреплением, а функцию предсказания метки с помощью обучения с учителем. **Основные результаты.** Обученный агент значительно превзошёл случайные стратегии по точности распознавания (уменьшает ошибку на 9,65%). Показано, что такой агент явно может использовать информацию от нескольких классификаторов, т.к. точность при увеличении допустимого количества действий растёт. **Практическая значимость.** Наше исследование показывает, что модифицированный метод глубокого Q-обучения демонстрирует способность извлекать информацию из разреженных откликов классификаторов так же хорошо, как классификатор наименьшего действия, обученный методом градиента по стратегиям. При этом предложенный в этой работе метод не требовал создания специальных функций потерь.

Ключевые слова: динамически конфигурируемые вычисления, принцип наименьшего действия, обучение с подкреплением, ансамбль методов, детерминированное планирование, анализ изображений, классификация изображений

Благодарность: исследование выполнено при поддержке Российского научного фонда (проект № 19–71–00146).

Ссылка для цитирования: Малашин Р.О., Бойко А.А. Обучение динамически конфигурируемого классификатора с использованием глубокого Q-обучения // Оптический журнал. 2022. Т. 89. № 8. С. 8–23. DOI: 10.17586/1023-5086-2022-89-08-08-23

Коды OCIS: 150.1135, 100.4996.

ВВЕДЕНИЕ

Динамически конфигурируемые (динамические) нейронные сети — это новый тип нейронных сетей глубокого обучения, который позволяет осуществлять вычисления, обусловленные входными данными и состоянием среды. В этом случае для получения каждого конкретного решения активируется только подмножество из всех нейронов вместо того, чтобы проводить полный цикл преобразований, используя все доступные параметры сети. Такие сети, в частности, позволяют экономить вычислительные ресурсы при решении различных задач компьютерного зрения. Динамические сети привлекают значительное внимание различных исследователей. В работе [1] рассмотрена возможность построения «интеллектуального» дропаута, который обнуляет узлы нейронной сети не случайным образом, а в зависимости от данных. Для этого авторы используют обучение с подкреплением и накладывают ограничение на разреженность масок для обнуления. Обучение слоёв сети и дропаута чередуется, а в остальном парадигма остаётся классической. Подход [1] был модифицирован в [2], где вместо обучения с подкреплением используют дифференцируемые операции с использованием SoftMax активации, после выполнения которой выбирается только k модулей-экспертов, которые соответствуют наибольшим k значениям выхода SoftMax.

Также используется функция ошибки, которая балансирует нагрузку на каждого из экспертов. Архитектура была использована для моделирования языка на большом наборе данных и всего было использовано до 4096 экспертов, каждый из которых содержал около одного миллиона параметров. Балансировка поощряет равномерное использование всех элементов, и это не позволяет осуществлять адаптивную настройку сложности для конкретного экземпляра задачи.

В [3] описан подход к последовательному применению сетей различных известных архитектур к задаче классификации. Для обучения функции выбора сетей предложено использование «мягкого оракула», использующего знание всех ответов нейронных сетей при расчёте функции ошибки (в то время, как при тестировании системы эта информация недоступна). В работе описана в том числе каскадная модель с использованием раннего выхода из сети.

В [4] описана модель «свёрточной нейронной смеси». Основная идея заключается в том, что нейронная сеть с большим количеством свёрточных блоков описывается в терминах вероятностей по картам признаков, которые могут генерироваться последовательно посредством вызовов свёрточных блоков. Вероятность карт признаков на практике принимает вид дельта-функции, центрированной на выходе определенной нейронной сети. Таким образом, можно сформировать дерево, каждому листу которого соответствует уникальный путь из свёрточных блоков. Теоретически для работы с такой моделью в процессе получения ответа надо генерировать множество нейронных архитектур и считать матожидание по их ответам. Для уменьшения количества вычислений авторы предлагают использовать единственную архитектуру, состоящую из усреднённых весов в каждом блоке. Кроме того, авторы рассматривают варианты раннего выхода из вычислений, обучая полносвязный классификатор поверх признаков каждого из уровней сетей.

Динамические вычисления для задачи анализа изображений хорошо согласуются с механизмом визуального внимания. В работе [5] нейронная сеть учится последовательно выбирать фрагменты на изображениях базы данных ImageNet [6], что позволяет подавать в сеть в зависимости от уже просмотренного контекста различные части изображения.

В этом случае, однако, граф вычислений отличается только по длине, но каждый цикл вычислений повторяется, поскольку для анализа каждого из них используется один и тот же подмодуль сети.

В [7] рассматривается задача поиска архитектуры нейронной сети по единственному примеру. В этом случае для распознавания

каждого конкретного изображения производится поиск архитектуры нейронной сети с помощью агента, обучаемого с подкреплением. В систему поощрения вводятся штрафы за использование вычислительно сложных архитектур, но при этом агент учится предсказывать всю архитектуру нейронной сети сразу, а не адаптироваться к промежуточным ответам отдельных вызванных модулей.

В отличие от этих подходов в [8] предлагается обучать алгоритм автоматического анализа изображений согласно задаче динамической настройки алгоритма [9] — после каждого шага анализа требуется подбирать следующую ветку вычислений в соответствии с уже полученной информацией об анализируемом изображении (контекстом). Такие стратегии называются контекстно-зависимыми (уникальными для каждого конкретного изображения). При этом важным является использование поощрения за вычислительно эффективное решение. Можно провести аналогию с принципом наименьшего действия из физики [10], согласно которому реальные траектории движения объектов удовлетворяют максимуму потенциальной и минимуму кинетической энергии. В [11] проводится аналогия с мышлением человека — реально принятое решение обуславливается максимумом оценки его достоверности (и желанием получить максимально возможную информацию), а также минимумом энергетических и временны́х ресурсов, которые необходимо на это затратить. По аналогии с этим в динамически конфигурируемых нейронных системах граф вычислений следует разворачивать таким образом, чтобы он удовлетворял минимуму вычислений и максимальной ожидаемой точности решений [8]. В [12] были проведены эксперименты по обучению классификатора наименьшего действия (агента, который функционирует в среде простых свёрточных нейронных сетей) классификаторов изображений и взаимодействует опосредовано через них с изображением. Было показано, что обучение с подкреплением может быть использовано для того, чтобы выучивать стратегии, которые обеспечивают преимущество по сравнению с любыми контекстно-независимыми стратегиями последовательного выбора классификаторов. Для обучения использовалась база данных CIFAR-10, которая является сравнительно небольшой, и использованный алгоритм гибридного актора-критика был склонен к переобучению. Кроме того, само пространство классификаторов может быть очень большим, а поиск оптимальной стратегии поведения по результатам проведённых в [12] экспериментов даже в небольших пространствах классификаторов может занимать достаточно много времени. Обучение с подкреплением в очень больших пространствах действий сейчас активно изучается. В частности, для этого используют подход, основанный на ценности действий (Q-learning), например [13]. Известно, что этот подход требует меньше обучающих примеров.

Цель настоящей работы заключалась в проведении исследования о применимости методов обучения с использованием информации о ценности действий к задаче построения классификаторов наименьшего действия.

Статья организована следующим образом. В первом разделе рассматривается задача разреженного стекинга классификаторов, для решения которой обучается классификатор наименьшего действия. Во втором разделе описан известный метод глубокого обучения на основе ценности действий и его модификация, предложенная нами для решения конкретной задачи. В третьем разделе мы приводим результаты численных экспериментов, которые показывают преимущество предложенной гибридной модификации по сравнению со стандартным DQN [14], а также производим сравнение с алгоритмом градиента по стратегиям, использованного в [12].

Разреженный стекинг классификаторов

Рассмотрим задачу распознавания изображений опосредованно через набор (пул) заранее обученных свёрточных нейронных сетей-классификаторов [8]. В этом случае считаем, что задачей агента является правильное распознавание конкретного изображения путём наблюдения откликов свёрточных нейронных сетей. Свёрточные нейронные сети обучены на базе данных CIFAR-10, но при этом каждый классификатор в процессе обучения видел только часть обучающей выборки (изображения только определённых классов).

От агента требуется последовательно проанализировать ограниченное количество откликов наиболее информативных классификаторов для каждого конкретного случая, при

этом выбор классификаторов осуществляется непосредственно самим агентом. Важно, что агенту позволяется использовать только часть из всех доступных классификаторов.

Задача объединения откликов нейронных сетей соответствует задаче стекинга [15], а с учётом рассматриваемого в настоящей статье ограничения на количество доступных для агента анализируемых классификаторов она может быть сформулирована как разреженный стекинг [8].

1. МЕТОДЫ

1.1. Глубокое Q-обучение (DQN)

Обучение с подкреплением на основе ценности действия происходит за счёт того, что создаётся и итеративно заполняется таблица $Q(s, a)$, где s — это состояние и a — выбранное агентом действие. Обученный агент может в каждом состоянии использовать действие a^*, которое соответствует максимальному значению ценности действия для текущего состояния в этой таблице: $a^* = \arg\max_a Q(s,a)$.

Для рассматриваемой задачи классификации изображений посредством разреженного стекинга мощность множества состояний среды чрезвычайно велика, так как каждое состояние агрегирует вещественные векторы откликов вызываемых классификаторов.

Для преодоления комбинаторного взрыва используется глубокое Q-обучение [15], в этом случае таблица аппроксимируется с помощью нейронной сети. Далее в разделе приводится описание реализованного метода глубокого обучения, который был использован и модифицирован в данной работе для решения поставленных задач.

Перейдём к некоторой параметрической модели $Q(s, a, \theta)$, где θ — ею обучаемые параметры в сжатом виде и предоставляющие информацию о ценности действия. При использовании глубокого Q-обучения агент руководствуется оценкой ценности действий, выдаваемой нейронной сетью с параметрами θ, при этом на вход сети подаётся описание конкретного состояния системы.

По аналогии с [12] в рассматриваемой в данной работе задаче состоянием системы является таблица, в которой хранятся отклики вызванных классификаторов. На рис. 1 изображён процесс распознавания изображения с помощью последовательного заполнения таблицы-состояния.

Также, аналогично работе [12], таблицу будущих откликов мы конкатенируем с таблицей масок, которая будет содержать информацию о присутствии отклика. Маска представляет собой вектор с длиной, равной выходному вектору классификатора, значения которого сигнализируют о вызове классификатора: в случае получения отклика классификатора все значения в векторе-маске равны 1, в противном случае — 0. Согласно работе [12], такие маски позволяют ввести робастное представление о том, какие классификаторы уже были вызваны, тем самым помогают агенту избегать дублирования действий. Таким образом, размер вектора состояний составляет $N \times C \times 2$, где N — количество классификаторов, а C — это размер выходного вектора классификатора. В начале «эпизода» обе таблицы заполнены нулями.

Пространство действий A включает выбор классификатора для обновления состояния и выбор метки класса. Это позволяет продолжать эпизод, пока не закончатся разрешённые действия, или агент не решит закончить эпизод, выбрав определённую метку.

Применительно к рассматриваемой задаче (без излишних деталей) алгоритм обучения агента с помощью глубокого Q-обучения [16] можно представить следующим образом:

1. Инициализируем параметры сети $Q(s, a, \theta)$ случайными значениями параметров θ, вероятность выбора случайного действия $\varepsilon \to 1{,}0$, и пустой буфер опыта, куда впоследствии будут заноситься кортежи вида (s^t, a^t, r, s^{t+1}), где s^t, s^{t+1} — состояния до и после выполнения агентом действия a^t, r — полученное вознаграждение (если эпизод закончился), t — текущая итерация алгоритма.

2. С вероятностью ε выбираем случайное действие (классификатор) a^t, с вероятностью $(1 - \varepsilon)$ действие (классификатор) выбирается с учётом предсказания модели $Q(s, a; \theta)$ (1):

$$a^t = \arg\max_a Q(s, a; \theta). \quad (1)$$

3. Получаем отклик выбранного классификатора на изображения и обновляем состояние s^{t+1} среды и информацию о награде r в случае окончания эпизода.

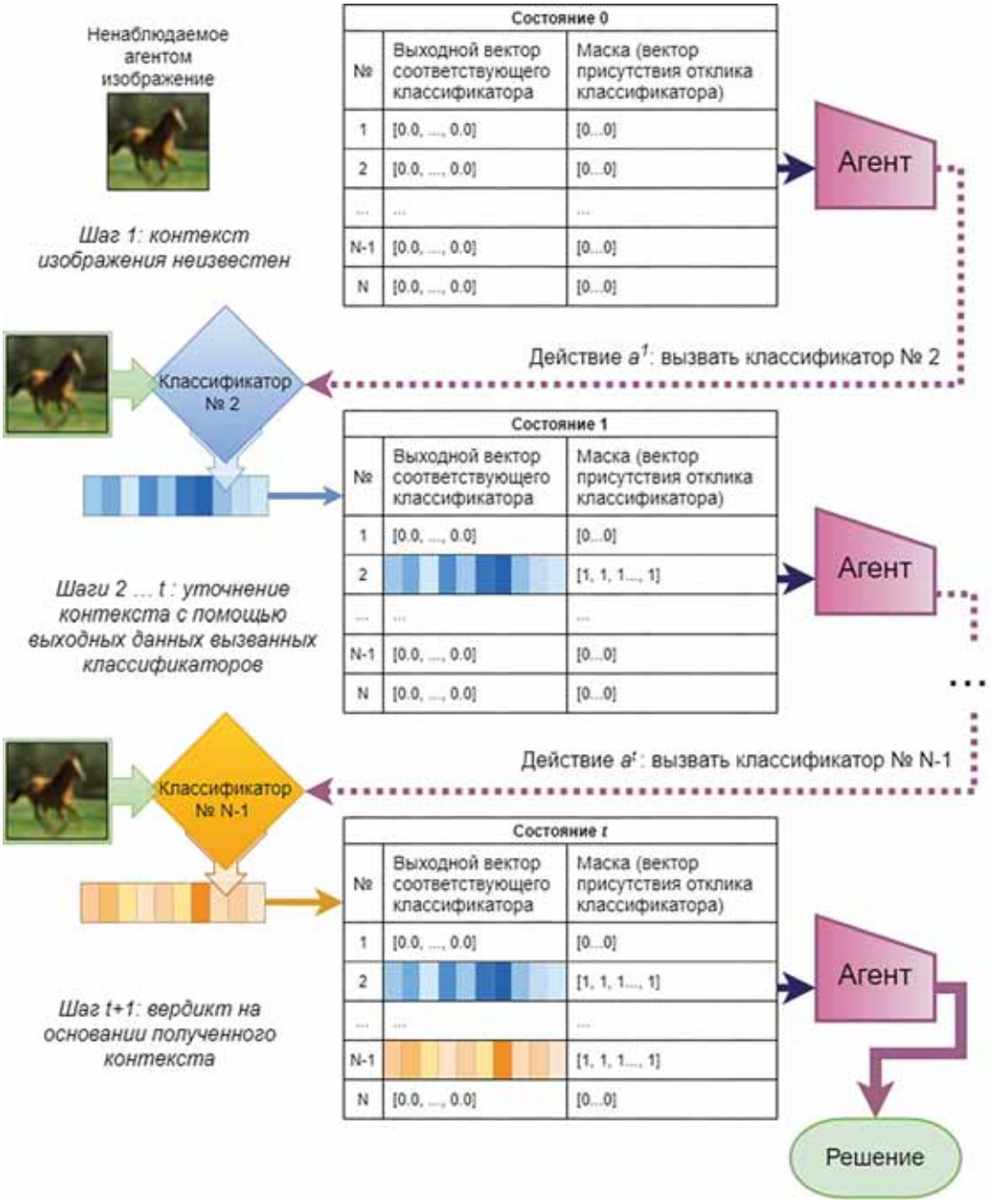

Рис. 1. Обновление состояния среды в задаче распознавания изображений опосредованно через набор заранее обученных классификаторов

4. Сохраняем полученный кортеж перехода (s^t, a^t, r, s^{t+1}) в буфере опыта. Буфер представляет собой массив, хранящий множество последних переходов. Случайные записи из буфера используются при расчёте целевых значений y_i, учитываемых в функции потерь из пункта 7.

5. Если буфер не заполнен (в нём меньше K записей, где число K определяется эмпирически), то надо вернуться на шаг 2. Если данных в буфере достаточно (его размер больше или равен K), то он используется для получения K случайных кортежей перехода.

6. Для каждого кортежа опыта рассчитываем целевое значение $y_i = r_i$, если эпизод окончен, в противном случае целевое значение будет рассчитано по формуле (2):

$$y_i = r_i + \gamma \max_{a^{t+1} \in A} Q(s^{t+1}, a^{t+1}; \theta), \qquad (2)$$

где $r_i = 1$, если эталонная метка изображения равняется предсказанной, и $r_i = 0$ — в противном случае.

7. Рассчитаем функцию потерь $L = (y_i - Q(s_i, a_i, \theta))^2$ и шаг её градиентного спуска для расчёта изменений параметров θ.

8. Повторяем алгоритм с пункта 2 до сходимости к оптимальным параметрам θ.

На рис. 2 представлена диаграмма процесса обучения $Q(s, a, \theta)$ агента в среде.

Ожидалось, что агент научится динамически на начальных итерациях выбирать вызов классификатора и только затем выбирать метку класса. На практике оказалось, что агенту не удаётся достаточно эффективно составить долгосрочную стратегию последовательного выбора классификатора и затем метки класса наблюдаемого опосредованно изображения.

1.2. Hybrid DQN — гибридное глубокое Q-обучение

В экспериментах оказалось, что классическое Q-обучение не может справиться с поставленной задачей, поэтому предлагаются модификации, описанные в этом разделе. Для этого выделяются два модуля — модуль выбора классификатора и модуль классификации, которые учатся независимо решению отдельных задач. Модуль классификации отвечает за предсказание метки изображения по состоянию s_t, а модуль выбора классификатора — за выбор классификаторов из пула для обновления состояния.

1.2.1. Функция выбора метки класса изображения (модуль классификации)

Для обучения модуля классификации, непосредственно позволяющего извлечь информацию о метке класса, используется кросс-

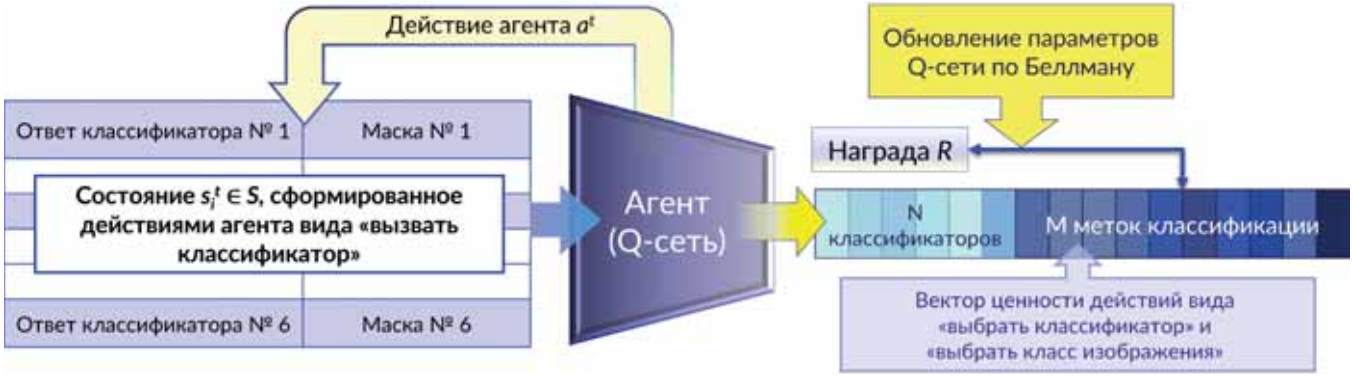

Рис. 2. Иллюстрация процесса обучения агента, для которого пространство действий A включает вызовы классификатора и выбор метки класса рассматриваемого изображения

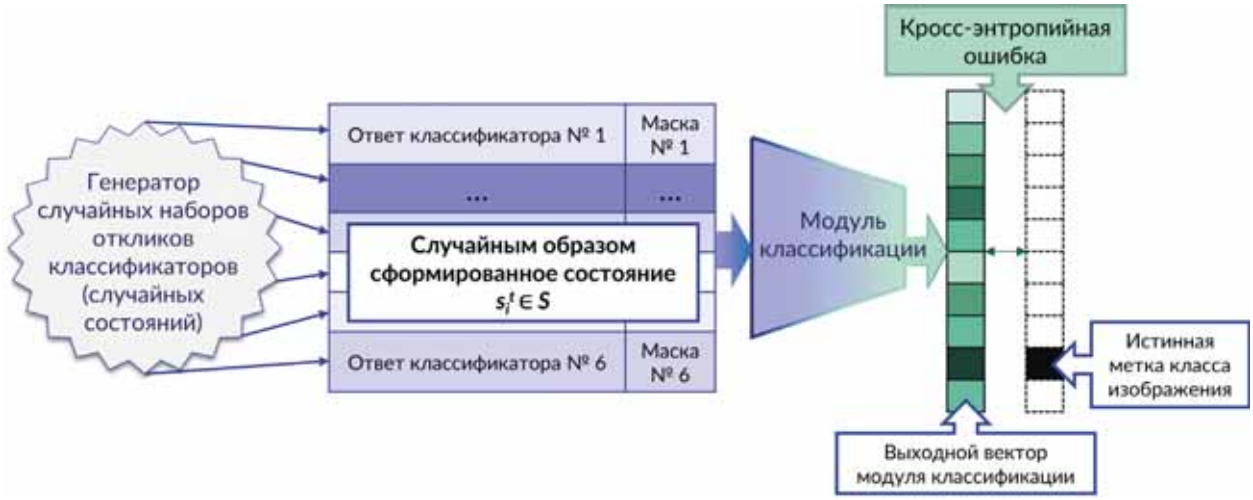

Рис. 3. Процесс обучения функции выбора класса изображения на основе состояний, сформированных случайным образом из пространства состояний среды S

энтропийная функция потерь, а предсказания формируются по случайным состояниям, которые потенциально могут возникать при функционировании агента. Такие состояния состоят из случайного набора L откликов в ответ на исследуемое изображение. Величина L также выбирается либо фиксировано, либо случайно в диапазоне от 1 до C, где C — количество классификаторов в пуле. Модуль обучается на парах вида (s_i^t, y_i), где s_i^t — экземпляр состояния (таблица из L откликов с соответствующими масками для изображения i), а y_i — метка этого изображения. Таким образом, модуль классификации точно также напрямую не наблюдает изображения, а только извлекает информацию из доступного пула экспертных заключений. На рис. 3 представлен процесс обучения модуля классификации.

Модуль классификации, обученный на случайном количестве откликов ($L \sim U\{1, C\}$), в дальнейшем будем называть универсальным, а обученный на фиксированном количестве (L) — специализированным на определённом количестве откликов.

1.2.2. Взаимодействие агента со средой

Агент награждается только раз — в момент окончания эпизода, когда модуль классификации выдаст метку класса. Предобученный модуль классификации теперь выполняет функцию среды, поскольку агент не может в процессе обучения влиять на него. Пространство действий A состоит только из вызовов классификаторов, и, следовательно, все действия являются однородными — любое действие $a^t \in A$ не останавливает эпизод до тех пор, пока t меньше разрешённого количества действий на эпизод. Соответственно, весь алгоритм обучения агента в такой среде можно представить как классический DQN, описанный в пункте 1.1 со следующими отличиями:

1. Длина одного эпизода не зависит от понижающего коэффициента γ, управляющего штрафом за отложенное решение. Эпизод завершается после фиксированного количества шагов. Агент принудительно учится выбирать оптимальное подмножество всех классификаторов для каждого конкретного изображения.

2. По окончании эпизода среда вызывает модуль классификации, формируя вознаграждение. Задача обучения сводится к тому, чтобы научить агента вызывать такие классификаторы, которые помогут модулю классификации лучше извлечь достаточно информации для выбора верного класса.

На рис. 4 представлена диаграмма обучения агента задаче эффективного подбора классификаторов для модуля классификации.

Рис. 4. Иллюстрация процесса обучения агента вызовам максимально выгодных классификаторов в зависимости от предыдущего состояния среды и последующей награды, связанной с ответом функции выбора класса изображения

Параметры модуля классификации не связаны с параметрами модели агента. На рисунке также указана кросс-энтропийная функция потерь, которая может опционально быть использована для подстройки (нюансировки) весов с учётом действий агента во время его обучения.

Благодаря появлению дополнительного модуля, задача, требующая от агента изучения длинных неоднородных по действиям стратегий, теперь сводится к задаче считывания содержания классифицируемых данных — задаче предоставления модулю классификации наиболее информативных классификаторов. Поскольку обучение осуществляется с помощью гибридной функции потерь (с учителем и с подкреплением), мы назвали эту модификацию Hybrid DQN, а архитектуру, используемую в нашей задаче классификации — Q-КНД (Q-классификатора наименьшего действия, Least Action Classifier, QLAC).

2. РЕЗУЛЬТАТЫ

В экспериментах мы использовали набор данных CIFAR10, который содержит 50000 обучающих и 10000 тестовых цветных изображений десяти классов объектов размером 32×32. Для выявления лучших гиперпараметров модели Hybrid DQN (в том числе количества итераций обучения) из набора данных CIFAR10, предназначавшихся для обучения, мы выделили 10000 изображений для валидационной выборки.

2.1. Пул классификаторов

В рамках обсуждаемой задачи для достижения желаемого поведения от агента необходимо множество классификаторов, которые будут выступать независимыми экспертами для агента, помогая последнему обособленно анализировать изображение. Аналогично [12], подобранные нами классификаторы ранее обучались на разных подмножествах классов в наборе данных CIFAR-10 и, следовательно, способны давать достаточно разнородные и независимые отклики. В наших экспериментах мы использовали две простые архитектуры свёрточных сетей из трёх и пяти слоёв. В табл. 1 приведены шесть классификаторов, которые мы обучили на случайно выбранных подмножествах из десяти оригинальных классов CIFAR-10; архитектура сети была выбрана также случайным образом.

Таблица 1. Множество свёрточных классификаторов, обученных на подмножестве классов изображений из набора данных CIFAR10 [12]

Номер классификатора	Классы изображений, на которых происходило обучение	Точность классификации на всех 10 классах, %
0	0, 1, 8, 4	35,6
1	1, 2, 3, 5, 6, 7, 9	57,1
2	3, 2, 4	24,6
3	7, 2	18,3
4	0, 1, 6, 7, 8, 9	51,0
5	0, 2, 3, 5	29,5

В проводимых нами экспериментах модулю выбора классификатора доступен весь пул из 6 классификаторов. Классификаторы в процессе обучения агента не изменяются (дообучение не происходит). Это позволяет полноценно анализировать стратегии, выученные агентом.

2.2. Deep Q-Network

Мы исследовали возможность обучить агента оперировать множеством действий A, состоящим из действия вида «выбрать классификатор» и «выбрать класс изображения» (рис. 2). При выборе действия «выбрать класс изображения» эпизод заканчивается, и среда принимает решение о награде, которое будет равно 1 в случае, если класс был выбран верно (в противном случае награда равна нулю). Агенту требовалось научиться оперировать одновременно действиями, позволяющими продолжить эпизод, тем самым отсрочив момент награды, и действиями, которые прерывают эпизод, и награда за которые связана с содержанием изображения (значит, невыгодно выбирать их случайным образом). Для выбора архитектуры Q-сети были исследованы два варианта: перцептрон с тремя слоями и сиамская сеть, состоящая из двух трёхслойных перцептронов. Сиамская сеть позволила разделить выходные векторы отдельно для выбора классификаторов и для выбора метки класса, однако, поиск параметров происходил

Таблица 2. Сравнение результатов тестирования агентов DQN и случайной стратегии выбора

Агент	Количество выполненных действий в среднем	Точность классификации на тестовой выборке, %
Случайная стратегия	2	66,5
Случайная стратегия	3	72,8
DQN: сиамская сеть	2,4	68,98
DQN: 3х-слойный перцептрон	2,5	66,72

одновременно для обеих частей. Агенту были доступны все классификаторы из табл. 1. В табл. 2 показаны результаты распознавания тестовой выборки и сравнение с результатами агента, использующего универсальный модуль классификации и случайную стратегию выбора классификаторов, независимую от контекста (без повторов действий).

По результатам можно видеть, что результаты, полученные с помощью классического глубокого Q-обучения, незначительно отличаются от результатов агента, использующего случайную стратегию выбора классификатора, что говорит о неспособности агента извлекать информацию о контексте действия и обобщать информацию от классификаторов. Стратегия такого агента не отличается по эффективности от случайной.

2.3. Обучение гибридного агента

Структура QLAC агента (рис. 4) помогает разделить процессы обучения модуля выбора классификатора и модуля классификации. Мы обучали модуль выбора классификатора в течение 2400000 итерации с оптимизатором Adam и с постоянным шагом обучения. В процессе обучения мы установили пороговое количество действий для агента. Универсальный модуль классификации предварительно обучался на различных количествах откликов, и в случае использования его со случайной стратегией выбора классификаторов (без повторов) его точность классификации тестовой выборки доходит до 76% для 4 действий. Для агента и для модуля классификации в качестве архитектуры используется многослойный перцептрон, состоящий из трёх полносвязных слоёв.

В табл. 3 показаны результаты обучения гибридного агента QLAC на тестовой выборке, и результаты агента, использующего случайную стратегию выбора классификаторов, независимую от контекста. Оба агента использовали фиксированный универсальный модуль классификации.

Из таблицы видно, что QLAC удаётся подобрать более эффективные наборы откликов для модуля классификации по сравнению со случайными.

Так как модуль классификации обучался на случайном количестве откликов ($L \sim U\{1, 6\}$), можно предположить, что ёмкость модуля используется неоптимальным образом для стратегии агента, который использует фиксированное количество шагов. Кроме того, распределение используемых агентом откликов также отличается от случайного, что может влиять на точность распознавания. Для проверки этого

Таблица 3. Сравнение результатов тестирования агентов DQN и случайной стратегии выбора

Разрешённое количество действий	Случайная стратегия (без повторов)	QLAC
2	66,50	76,15
3	72,80	77,74
4	76,08	78,01

Таблица 4. Точность классификации тестовой выборки с помощью QLAC с универсальным модулем классификации и после его дообучения, %

Разрешённое количество действий	QLAC с универсальным модулем классификации	QLAC с дообучением универсального модуля классификации
1	67,00	67,03
2	76,15	75,93
3	77,74	77,30
4	78,01	78,31
5	77,71	77,36
6	77,17	76,79

Таблица 5. Результаты обучения агента QLAC со специализированным (к разрешённому количеству откликов) модулем классификации к агенту со случайно инициализированными параметрами

Разрешённое количество действий	Случайная стратегия и специализированный модуль классификации	QLAC с фиксированным специализированным модулем классификации	QLAC с дообучением специализированного модуля классификации
1	59,14	67,65	67,67
2	58,36	76,12	75,41
3	47,29	77,59	77,34
4	70,09	77,50	77,10
5	40,44	77,37	76,27

предположения мы проводили дообучение параметров модуля классификации одновременно с обучением модуля выбора классификатора. Во время дообучения мы оптимизировали параметры модуля классификации, но теперь обучающая выборка состояла не из случайных состояний, а из подобранных модулем выбора классификаторов. В табл. 4 показаны результаты.

Из таблицы можно сделать вывод, что ёмкости универсальной модели уже было достаточно для исследуемого набора данных, и дообучение не повышает точность. Примерно к тем же наблюдениям пришли, исходя из результатов табл. 5, — специализированные модели классификации (по количеству откликов, содержащихся в анализируемом состоянии) не способствовали улучшению результатов QLAC по сравнению с QLAC с универсальной моделью.

Из таблицы видно, что агент может использовать информацию от нескольких классификаторов, т.к. точность при увеличении допустимого количества действий растёт. Однако при превышении порога 3–4-х действий во всех случаях точность не увеличивается, а начинает падать. У нас есть два предположения, почему это могло произойти. Первое предположение связано с переобучением, т.к. увеличение вектора входных данных позволяет легче подбирать правильные решения для обучающей выборки, которые не обобщаются на тестовое множество. Также параметры обучения в процессе экспериментов подбирались при использовании от 1 до 4-х действий, что также может давать некоторый вклад в получаемые результаты.

Из таблиц также можно видеть, что подстройка параметров не сказывается принципиальным образом на точность распознавания тестовой выборки, что говорит о том, что ёмкости модуля классификации хватает для работы с переменным числом. Также использование предобученного модуля в исходном виде снижает эффект переобучения.

2.4. Сравнение классификатора наименьшего действия при обучении методом градиентов по стратегиям

Интересно понять, позволяет ли QLAC выучивать контекстно-зависимые стратегии или осуществляет лишь поиск оптимальных классификаторов и выбирает их независимо от содержания тестового изображения. Для этого в работе [12] были проанализированы вычислительные графы, в которые разворачиваются вычисления, и было показано, что классификатор наименьшего действия (Least action classifier, LAC), обучающийся с помощью градиентов по стратегиям (гибридной функции ошибки и метода актора-критика) оперирует контекстно зависимыми стратегиями, и это позволяет значительно повысить точность классификации. В настоящей работе вместо того, чтобы анализировать графы вычислений мы сравниваем наши результаты с классификатором наименьшего действия [12]. В табл. 6 представлено сравнение точности классификации QLAC и LAC [12].

Можно видеть, что результаты сравнимы по точности. На основании этого можно предположить, что QLAC также оперирует контекстно-зависимыми стратегиями. При этом, для обучения QLAC (в отличие от [12]), мы использовали обучающую выборку только из 40000 образцов (10000 использовались для контроля переобучения (ранней остановки)), поэтому потенциально существует возможность поднять точность, используя всю выборку с найденными гиперапараметрами, однако, мы не проводили таких экспериментов. Также для обучения LAC в функции потерь было необходимо

Таблица 6. Точность классификации тестовой выборки CIFAR-10 с помощью QLAC и LAC [12]

Разрешённое количество действий	QLAC, %	LAC [12], %
2	76,15	75,81
3	77,74	77,81
4	78,01	78,62

использовать специально адаптированный энтропийный бонус за исследование новых стратегий на каждом шаге, в то время как для агента QLAC этой необходимости не было.

Из табл. 6 также видно, что при увеличении разрешённого количества действий точность LAC начинает превосходить QLAC. Эффект ограничения роста точности QLAC при увеличении количества разрешённых действий также наблюдался в табл. 4. Мы связываем это с эффектом переобучения. Поскольку в [12] обучение модуля классификации происходит одновременно с обучением стратегии агента с использованием энтропийного бонуса, то это может оказывать эффект регуляризации на ход обучения. Исходя из этих результатов, можно сделать вывод, что для создания динамически конфигурируемых систем целесообразно использовать методы обучения без учителя и самообучения (self-supervised learning), снимающие ограничения на требования к использованию маркированных данных. Мы планируем провести такие эксперименты в следующих работах.

2.5. Стратегии поведения агента

Рассмотрим подробно, как именно агент QLAC оперирует стратегиями на примере тестовой выборки в условиях ограничения количества

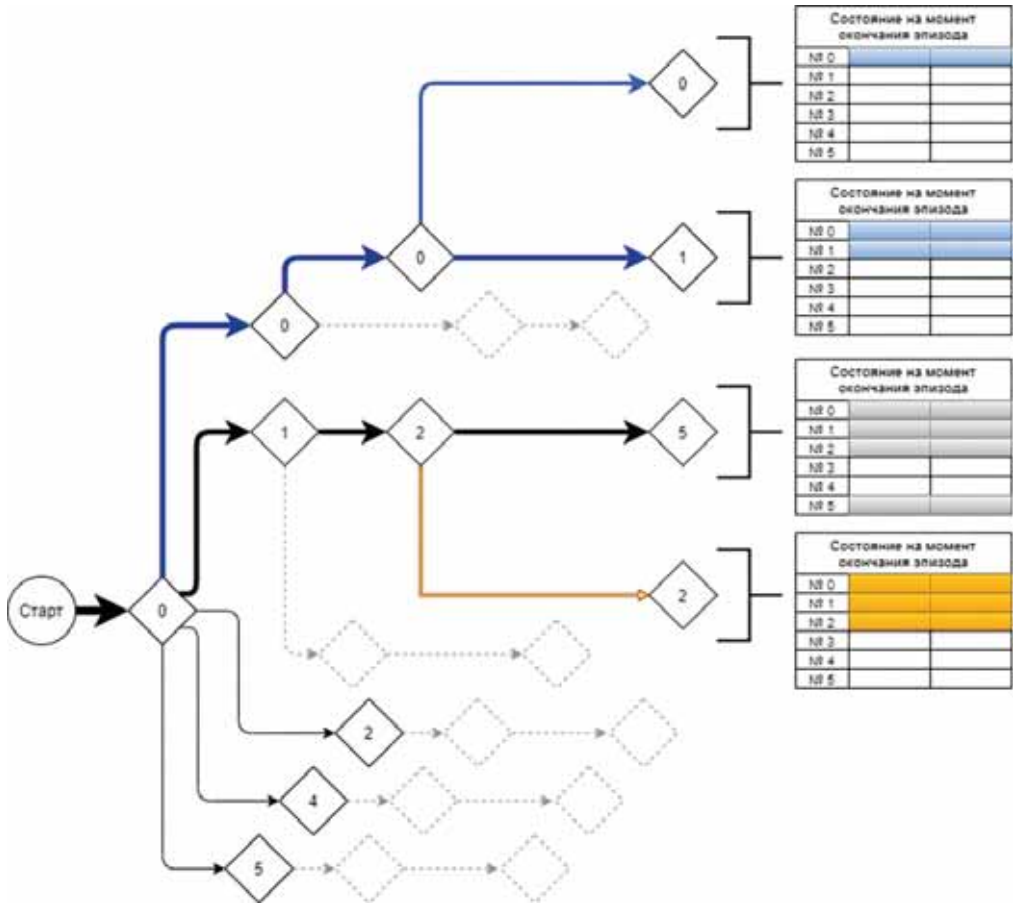

Рис. 5. Визуализация части графа вызовов классификаторов для тестовой выборки с ограничением в 4 возможных действия для агента QLAC. Вершины графа — вызванные классификаторы. Рёбрам графа соответствуют разные действия, выбираемые агентом. Там, где из узла выходит несколько ребер, действия являются контекстно-зависимыми

возможных действий. На основе последовательностей вызванных классификаторов был визуализирован вычислительный граф, частично представленный на рис. 5.

Мы наблюдаем, что агент при отсутствии контекста (первое действие, таблица откликов классификаторов пуста) всегда выбирает один и тот же классификатор в качестве первого — им оказался классификатор «0». Такое же поведение — выбор одного и того же классификатора в качестве первого при нулевом контексте — также демонстрировалось у агента из предыдущей работы [12]. Состояния в листьях дерева показывают распределение полученной информации от классификаторов внутри таблицы. Так как эксперимент проводился

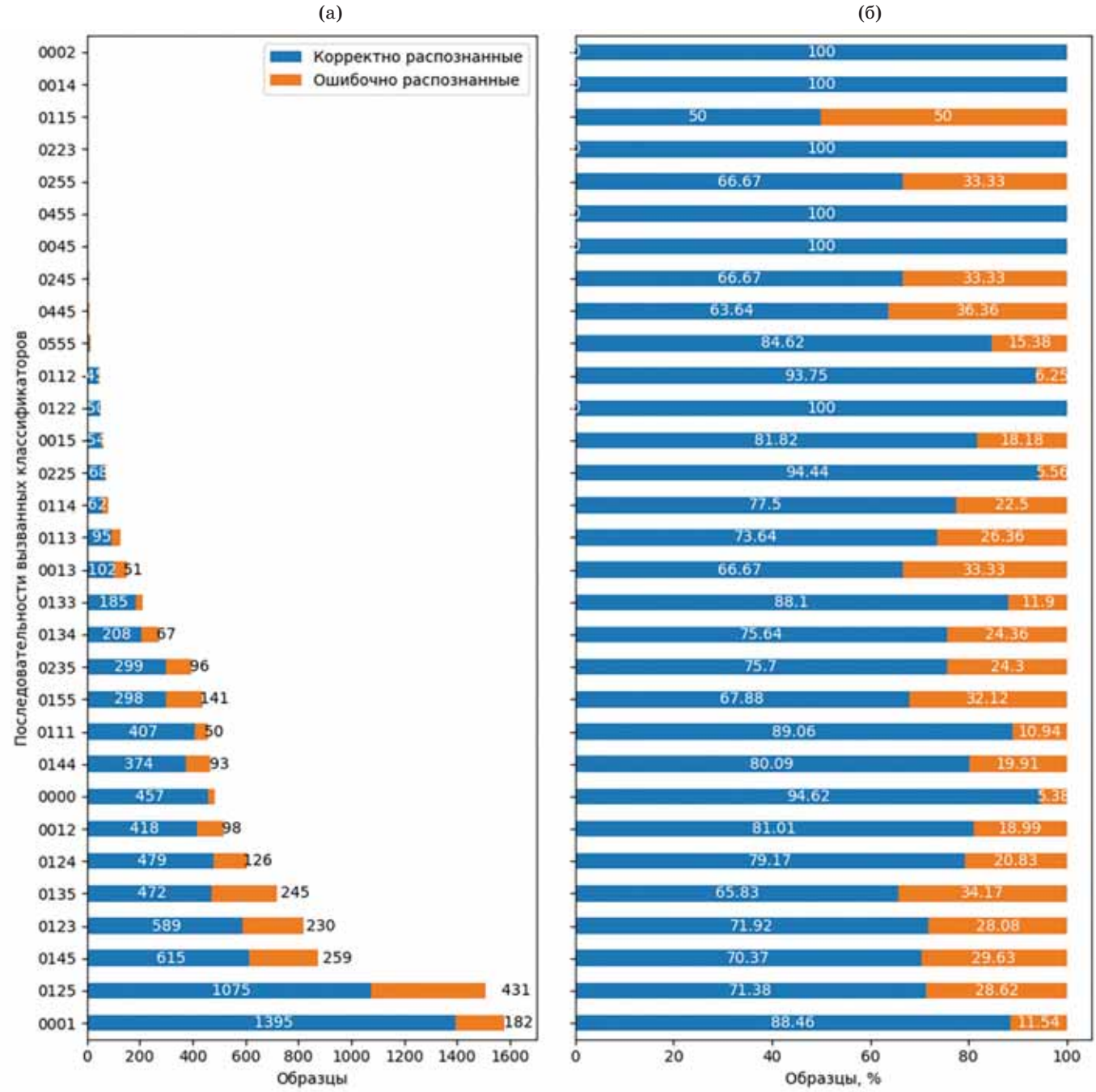

Рис. 6. Распределение результатов распознавания тестовой выборки по последовательностям выбранных классификаторов. а) В абсолютном значении; б) в процентном соотношении от количества образцов, для которых были зарегистрированы соответствующие последовательности

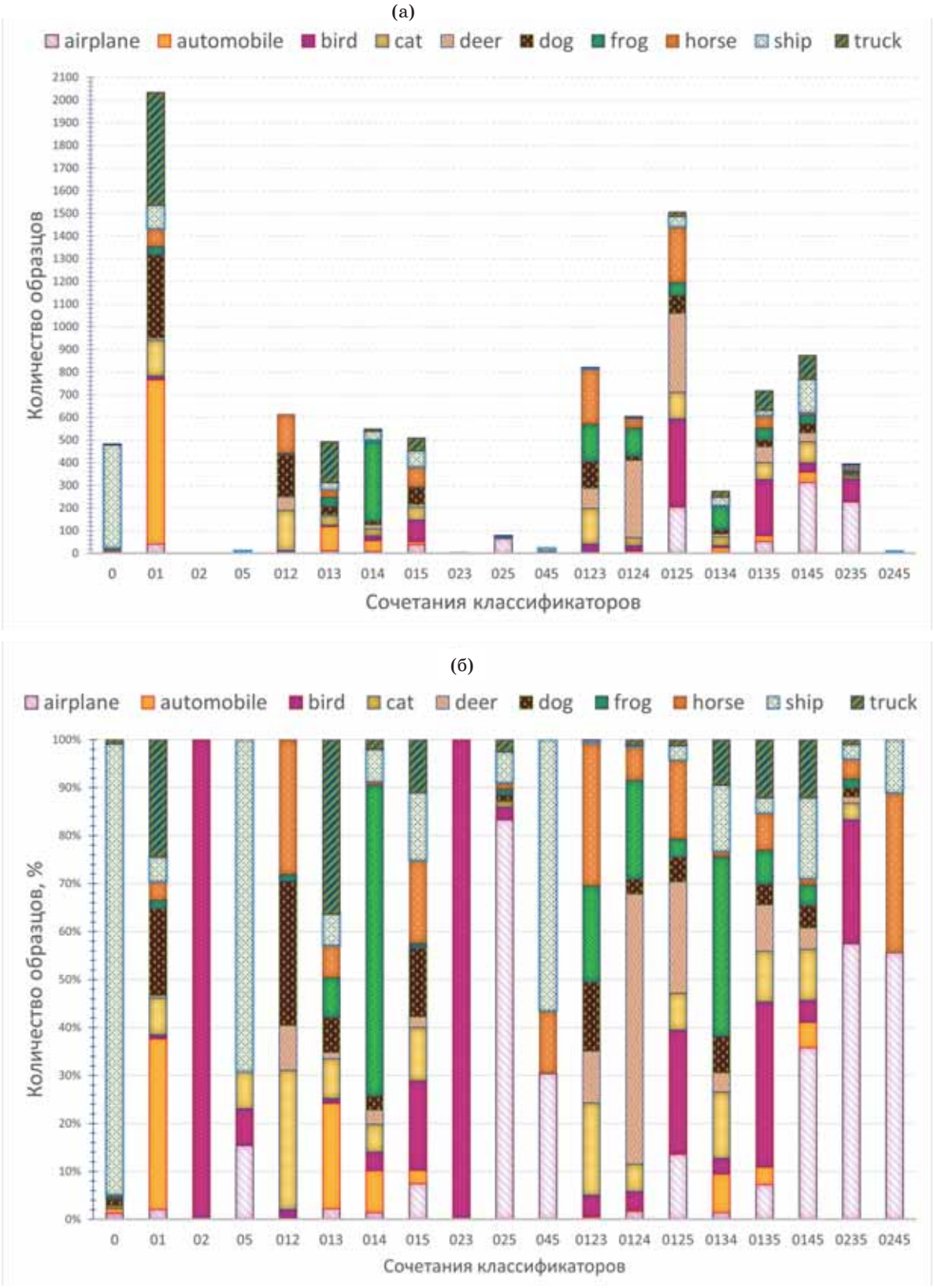

Рис. 7. Распределение истинных классов образцов тестируемой выборки по наборам вызванных классификаторов. а) По количеству всех зарегистрированных вызовов соответствующего набора классификаторов; б) в процентном соотношении от количества образцов, для которых были вызваны соответствующие сочетания классификаторов. Дублирующие вызовы не показаны, сочетания, отличающиеся только порядком вызовов, объединены вместе

без запрета на повторные вызовы классификаторов, можно видеть, что иногда таблица состояний по окончанию эпизода может включать меньшее количество активированных классификаторов, чем количество доступных агенту действий. Оказалось, что агент может специально ограничивать доступную информацию для модуля классификации — так, набор классификаторов самой «популярной» последовательности выборов состоит только из двух классификаторов: «0» и «1». Убедиться в этом можно на распределении результатов классификации по зарегистрированным последовательностям на рис. 6.

Можно видеть, что в общем случае (кроме, ранее упомянутого популярного набора из двух классификаторов и некоторых других, более редких наборов) цепочки с уникальными вызовами классификаторов оказались более популярны, то есть уточнение контекста от разных источников для агента оказалось более ценным, чем намеренное сокрытие информации от модуля классификации. При этом существуют цепочки с высоким процентным соотношением наград к количеству вызовов, и цепочки с повторяющимися вызовами входят в число таковых. Так цепочка, состоящая исключительно из вызовов классификатора «0», оказалась одной из наиболее ценных с точки зрения распределения наград.

Для того, чтобы исследовать связь между выбираемой цепочкой действий и семантическим содержанием изображений было проанализировано отношение истинных меток изображений к выбираемой агентом цепочке действий. На рис. 7 представлено распределение классов, для которых были вызваны соответствующие определённые сочетания классификаторов (без учёта их последовательности и частоты вызова).

По графикам можно видеть, что для некоторых классов сочетания классификаторов более специфичны, чем для других, при этом нет такого класса, для распознавания которого применялся бы только один набор классификаторов, и никакие другие. Например, набор, в котором присутствует единственный классификатор «0» (что соответствует последовательности из повторных вызовов классификатора «0»), оказался полезным для классификации образцов класса «ship».

Таким образом, так как наборы классификаторов подобраны агентом QLAC, исходя исключительно из контекстной информации от вызванных классификаторов, и так как каждый такой набор оказался специфичен к определённому набору классов, можно утверждать, что агент QLAC оперирует контекстно-зависимыми стратегиями, которые связаны с семантикой изображения.

ВЫВОДЫ

В данной работе был предложен метод гибридного обучения по ценностям действий Hybrid DQN в задаче классификации по принципу наименьшего действия. Нам было важно, чтобы агент был способен обучиться контекстно-зависимой стратегии для разворачивания вычислительного графа таким образом, чтобы получить максимальную ожидаемую точность при условии ограниченного числа действий. Благодаря модификации, задача, требующая от агента изучения неоднородных по действиям стратегий, сводится к поиску наиболее полезных (для модуля классификации) классификаторов. Соответствующую архитектуру агента мы назвали QLAC.

В наших экспериментах QLAC оказался способен использовать информацию от нескольких классификаторов, т.к. точность при увеличении допустимого количества действий росла. Также было проведено сравнение с классификатором наименьшего действия (Least action classifier, LAC). Результаты оказались сравнимы по точности даже при меньшем наборе данных для обучения агента QLAC и без использования модифицированной функции ошибки за действия без учёта контекста. Таким образом, разработанный метод может быть хорошей альтернативой методу градиента по стратегиям (использованного в классификаторе наименьшего действия) для обучения агента поиску решения в задачах, требующих эффективного подбора инструментов, уникального для каждого исследуемого образца.

В современных динамических конфигурируемых системах анализа изображений принято использовать алгоритм градиентов по стратегиям [5, 7] (в том числе из-за меньших требований к памяти), однако результаты, полученные в этой работе (а также

в работах [5, 7]), свидетельствуют о склонности к переобучению агента, поэтому предложенный нами алгоритм может быть востребован (например, при использовании его для последовательного обучения отдельных элементов агента).

СПИСОК ИСТОЧНИКОВ

1. Bengio E., Bacon P.L., Pineau J., Precup D. Conditional computation in neural networks for faster models // URL: https://arxiv.org/abs/1511.06297 (accessed 01.04.2022).
2. Shazeer N., Mirhoseini A., Maziarz K., Davis A., Le Q., Hinton G., Dean J. Outrageously large neural networks: The sparsely gated mixture-of-experts layer // URL: https://arxiv.org/abs/1701.06538 (accessed: 01.04.2022).
3. Bolukbasi T., Wang J., Dekel O., Saligrama V. Adaptive neural networks for efficient inference // International Conference on Machine Learning. 2017. Sydney, Australia. August 6–11. P. 527–536.
4. Ruiz A., Verbeek J. Adaptative inference cost with convolutional neural mixture models // International Conference on Computer Vision. 2019. Seoul, Republic of Korea. October 27 to November 2. P. 1872–1881.
5. Wang Y., Lv K., Huang R., Song S., Yang L., Huang G. Glance and focus: a dynamic approach to reducing spatial redundancy in image classification // Advances in Neural Information Processing Systems. 2020. V. 33. P. 2432–2444.
6. Deng J., Dong W., Socher R., Li L.-J., Li K., Fei-Fei L. ImageNet: A large-scale hierarchical image database // IEEE Conference on Computer Vision and Pattern Recognition. 2009. Miami, Florida, USA. June 20–25. P. 248–255. DOI: 10.1109/CVPR.2009.5206848.
7. Cheng A.C., Lin C.H., Juan D.C., Wei W., Sun M. InstaNAS: Instance-aware neural architecture search // Proceedings of the AAAI Conference on Artificial Intelligence. 2020. New York, USA. February 7–12. V. 34. № 4. P. 3577–3584.
8. Malashin R.O. Principle of least action in dynamically configured image analysis systems // Journal of Optical Technology. 2019. V. 86. № 11. P. 678–685.
9. Biedenkapp A., Bozkurt H.F., Eimer T., Hutter F., Lindauer M. Dynamic algorithm configuration: Foundation of a new meta-algorithmic framework // Proceedings of the Twenty-fourth European Conference on Artificial Intelligence. 2020. Santiago de Compostela, Spain. 29 August — 8 September. P. 427–434. DOI: 10.3233/FAIA200122
10. Полак Л.С. Вариационные принципы механики. М.: Физматлит, 1959. 930 с.
11. Шелепин Ю.Е., Красильников Н.Н. Принцип наименьшего действия, физиология зрения и условнорефлекторная теория // Российский физиологический журнал им. И.М. Сеченова. 2003. Т. 89. № 6. С. 725–730.
12. Malashin R.O. Sparsely ensembled convolutional neural network classifiers via reinforcement learning // The 6th International Conference on Machine Learning Technologies. 2021. April 23–25. Jeju Island, Republic of Korea. P. 102–110.
13. Van de Wiele T., Warde-Farley D., Mnih A., Mnih V. Q-learning in enormous action spaces via amortized approximate maximization // URL: https://arxiv.org/abs/2001.08116 (accessed 01.04.2022)
14. Mnih V., Kavukcuoglu K., Silver D. et al. Human-level control through deep reinforcement learning // Nature. 2015. V. 518. № 7540. P. 529–533.
15. Wolpert D.H. Stacked generalization // Neural Networks. 1992. V. 5. № 2. P. 241–259. DOI:10.1016/S0893-6080(05)80023-1
16. Lapan M. Deep reinforcement learning hands-on: Apply modern RL methods to practical problems of chatbots, robotics, discrete optimization, web automation, and more. 2nd ed. Birmingham: Packt Publishing Ltd, 2020. 799 p.

АВТОРЫ

Роман Олегович Малашин — канд. техн. наук, старший научный сотрудник, Институт физиологии им. И.П. Павлова РАН, 199034, Санкт-Петербург, Россия; доцент,

AUTHORS

Roman O. Malashin — PhD, senior research fellow, Pavlov Institute of Physiology, Russian Academy of Sciences, 199034, Saint-Petersburg, Russia, associate professor,

Санкт-Петербургский государственный университет аэрокосмического приборостроения, 1900031, Санкт-Петербург, Россия, Scopus ID: 55789083900, https://orcid.org/0000-0002-2493-839X, malashinroman@mail.ru

Арина Андреевна Бойко — младший научный сотрудник, Институт физиологии им. И.П. Павлова РАН, 199034, Санкт-Петербург, Россия; ассистент, Санкт-Петербургский государственный университет аэрокосмического приборостроения, 1900031, Санкт-Петербург, Россия, Scopus ID: 57225010175, https://orcid.org/0000-0001-7520-0056, boikoaa@infran.ru

Saint-Petersburg State University of Aerospace Instrumentation, 190031, Saint-Petersburg, Russia, Scopus ID: 55789083900, https://orcid.org/0000-0002-2493-839X, malashinroman@mail.ru

Arina A.Boiko — junior research fellow, Pavlov Institute of Physiology, Russian Academy of Sciences, 199034, Saint-Petersburg, Russia, assistant, Saint-Petersburg State University of Aerospace Instrumentation, 190031, Saint-Petersburg, Russia, Scopus ID: 57225010175, https://orcid.org/0000-0001-7520-0056, boikoaa@infran.ru

Статья поступила в редакцию 29.10.2021, одобрена после рецензирования 15.04.2022, принята к печати 04.07.2022