

УДК 004.41 + 004.02

КЛАССИФИКАЦИЯ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

А. А. Елизаров¹, Е. В. Разинков²

*Институт вычислительной математики и информационных технологий
Казанского (Приволжского) федерального университета*

¹artelizar@gmail.com, ²evgeny@razinkov.ai

Аннотация

В последнее время активно развивается такое направление машинного обучения, как обучение с подкреплением. Как следствие предпринимаются попытки использования обучения с подкреплением для решения задач компьютерного зрения, в частности для решения задачи классификации изображений. Задачи компьютерного зрения являются на сегодняшний день одними из наиболее актуальных задач искусственного интеллекта.

В статье предложен метод классификации изображений в виде глубокой нейронной сети с использованием обучения с подкреплением. Идея разработанного метода сводится к решению задачи о контекстном многоруком бандите с помощью различных стратегий достижения компромисса между эксплуатацией и исследованием и алгоритмов обучения с подкреплением. Рассмотрены такие стратегии, как ϵ -жадная, ϵ -softmax, ϵ -decay-softmax и метод UCB1, и такие алгоритмы обучения с подкреплением, как DQN, REINFORCE и A2C. Проведен анализ влияния различных параметров на эффективность работы метода.

Ключевые слова: машинное обучение, классификация изображений, обучение с подкреплением, задача о контекстном многоруком бандите.

ВВЕДЕНИЕ

Задача классификации изображений представляет собой одну из важнейших задач компьютерного зрения и заключается в разделении объектов на изображениях на группы, называемые классами [1]. С развитием информационных технологий для решения этой задачи стали успешно применяться методы машинного обучения, в особенности сверточные нейронные сети (англ. convolutional neural network, CNN), которые показывают лучшие результаты по распознаванию объектов на изображениях, начиная с соревнования ILSVRC 2012 (англ. ImageNet Large Scale Visual Recognition Challenge) [2], а в 2015 году превзошли человеческий уровень классификации изображений на наборе данных ImageNet [3].

Обучение глубоких нейронных сетей, в том числе глубоких сверточных сетей, происходит обычно с помощью методов обучения с учителем. Однако в последнее время одним из наиболее развивающихся направлений машинного обучения стало обучение с подкреплением (англ. reinforcement learning). В отличие от обучения с учителем, в обучении с подкреплением модель тренируется, взаимодействуя с окружающей средой и получая от нее отклик [4]. Обучение с подкреплением применяется в различных областях науки и также может использоваться при решении задачи классификации изображений, например, в работах FCAN [5] и DT-RAM [6].

В настоящей работе представлен метод классификации изображений в виде глубокой нейронной сети с использованием обучения с подкреплением, в основу которого также положены интуитивные принципы работы зрительной системы человека. Проведен ряд экспериментов по исследованию влияния различных параметров на эффективность работы метода.

ОПИСАНИЕ РАЗРАБОТАННОГО МЕТОДА

Пусть в качестве базовой модели для решения задачи классификации изображений используется нейронная сеть ResNet [7]. Изображение в виде карт признаков последовательно проходит через остаточные блоки каждой группы ResNet для извлечения от низкоуровневых до высокоуровневых признаков, а далее, на основе полученных признаков, классифицируется полносвязным сло-

ем. В таблице 1 приведены размеры карт признаков изображений на выходе каждой группы остаточных блоков ResNet.

Таблица 1. Размеры карт признаков изображений на выходе групп блоков ResNet

Номер группы блоков	Размер на выходе
1	56x56x64
2	28x28x128
3	14x14x256
4	7x7x512

Идея разработанного нами метода заключается в использовании помимо сети ResNet дополнительной нейронной сети, которую будем называть агентом. Также будем считать, что сеть ResNet является уже предобученной для используемого набора данных. Задача агента – получив на вход выход группы остаточных блоков ResNet, предсказать, какую область нужно вырезать из полученных карт признаков изображения, чтобы в процессе классификации увеличилась уверенность сети ResNet в принадлежности объекта на исходном изображении правильному классу. Далее карты признаков изображения обрезаются согласно предсказанию агента и подаются на вход следующей группе остаточных блоков ResNet для последующего извлечения признаков и классификации. Такая задача агента ставится из предположения, что при правильном вырезании сеть ResNet сможет лучше классифицировать объект на изображении.

Поставленная задача агента является задачей о контекстном многоруком бандите [4], так как эпизод состоит только из одного действия, и во время обучения агент получает m различных контекстов, где m – количество изображений в обучающей выборке.

Действиями агента являются варианты вырезания части карт признаков изображения. Пусть имеются три возможных типа области вырезания: ограничивающие прямоугольники с размерами сторон, равными соответственно 0.5, 0.75 и 1.0 от размера сторон входных карт признаков, которые являются квадратными. Выбирается параметр шага λ , а затем входные карты признаков замощаются слева–направо и сверху–вниз каждым из типов прямоугольников с ука-

занным шагом. Замоещение ограничивающим прямоугольником с размером сторон 1.0 от размера сторон карт признаков является действием агента «ничего не вырезать».

Вознаграждением агента является разность между уверенностью ResNet в принадлежности объекта на изображении правильному классу, если обрезать карты признаков изображения согласно выбору агента, и уверенностью ResNet в принадлежности объекта на изображении правильному классу, если ничего не вырезать (базовая модель ResNet), или, более формально, вознаграждение $r_{i,t}$ на временном шаге t для i -го изображения из набора данных можно записать как

$$r_{i,t} = outputs_{i,t}(c) - baseline_outputs_{i,t}(c),$$

где $outputs_{i,t}$, $baseline_outputs_{i,t}$ – векторы уверенностей ResNet в принадлежности объекта на i -м изображении различным классам, полученные соответственно после вырезания агентом и при отсутствии вырезания, а c – индекс правильного класса. Таким образом, вознаграждение агента является вещественным числом, изменяющимся в пределах от -1 до 1 .

Агент (нейронная сеть) будет обучаться с помощью выбранного алгоритма обучения с подкреплением, применяя при обучении выбранную стратегию достижения компромисса между эксплуатацией и исследованием (англ. exploitation–exploration) для решения задачи о контекстном многоруком бандите. При обучении агента веса ResNet будут заморожены. Стоит отметить, что для обучения агента в разработанном методе, как и в методах FCAN и DT-RAM, необходимы только метки правильных классов изображений. Место расположения агента в архитектуре ResNet, архитектура агента и другие параметры будут подробно рассмотрены далее.

ЭКСПЕРИМЕНТЫ

Особенности реализации

Для реализации разработанного метода применялся язык программирования Python 3.8 с библиотекой для глубокого обучения PyTorch 1.4 [8]. Для обучения нейронных сетей использовалась среда Google Colaboratory [9].

Описание набора данных

В качестве набора данных для обучения, тестирования нейронных сетей и проведения экспериментов было использовано подмножество набора данных ImageNet [10], который содержит почти 1.4 млн. изображений большого размера, принадлежащих 1000 различным классам от категории животных до категорий мебели и одежды.

Выбор набора данных ImageNet обусловлен наличием большого количества изображений для снижения влияния переобучения и наличием изображений большого размера для проверки эффективности вырезания. Использование только подмножества данных ImageNet обусловлено тем, что весь этот набор данных достаточно большой и для решения задачи не требовался полностью. В подмножество были включены только классы животных, что позволило приблизить исходную задачу классификации к задаче детального распознавания (англ. fine-grained recognition) [11] и увеличить влияние вырезания на эффективность классификации.

Приведем основные характеристики использованного подмножества:

- 60 классов различных животных,
- 14 тыс. изображений для обучения ResNet,
- 55 тыс. изображений для обучения агента,
- 8 тыс. изображений в валидационной выборке.

Основные параметры

Для изображений из использованного набора данных применялись следующие преобразования (указаны соответствующие классы преобразований в библиотеке PyTorch): `transforms.Resize(256)`, `transforms.RandomCrop((224, 224))` при обучении ResNet, `transforms.CenterCrop((224, 224))` при обучении агента.

В качестве базовой модели для классификации изображений использовалась модифицированная нейронная сеть ResNet-50, которая была инициализирована весами, предобученными на ImageNet, и дообучалась на используемом подмножестве. Модификация состояла в использовании перед полносвязным слоем ResNet слоя субдискретизации MaxPool вместо слоя AveragePool. Также в модифицированной архитектуре применялся слой ROI Pool.

Для обеспечения стабильности при обучении агента аугментация данных не применялась. В качестве метода оптимизации использовался метод Adam. Так как скорость обучения в Adam отдельно адаптируется для каждого обучаемого параметра, было принято решение не применять снижение скорости обучения, а использовать Adam с постоянным коэффициентом обучения. В качестве метода регуляризации применялся метод L2 weight decay.

Важным моментом является использование изображений из валидационной выборки агента так же, как и изображений из обучающей выборки агента для обучения ResNet. Этот выбор основан на предположении, что для эффективной работы агента нужно обучать и тестировать его на выходах группы блоков ResNet из одного распределения, то есть, другими словами, и обучать, и тестировать агента необходимо на изображениях, которые либо использовались при обучении ResNet, либо нет (то есть на изображениях, на которые ResNet реагирует схожим образом). Если для обучения и тестирования агента использовать изображения, на которых ResNet не обучалась, то в этом случае может быть сложным исследование влияния гиперпараметров на эффективность работы агента. Поэтому сначала были проведены эксперименты с агентом, который обучался и тестировался на изображениях, использованных при обучении ResNet.

Проверка предположения об эффективности вырезания

Задача агента строится на предположении об увеличении точности классификации за счет вырезания в процессе классификации правильной области на изображении. Для проверки этого предположения применялось моделирование работы «идеального» агента. Принцип работы такого агента заключался в поиске для каждого изображения из набора данных действия, приносящего наибольшее вознаграждение, и последующем использовании этого действия для вырезания в процессе классификации. Такой поиск осуществлялся с помощью метода полного перебора и был возможен благодаря относительно небольшому числу допустимых действий агента. По результатам классификации с использованием «идеального» агента из таблицы 2 можно сделать вывод, что исходное предположение действительно является верным.

Таблица 2. Результаты классификации ResNet с использованием «идеального» агента

Тип выборки	Точность классификации без учета вырезания	Среднее лучшее вознаграждение	Средняя доля действий с положительным вознаграждением	Точность классификации с учетом вырезания
Обучающая	0.8609	0.0865	0.192	0.9539
Валидационная	0.7992	0.0647	0.227	0.8687

Выбор места расположения агента в архитектуре ResNet

В процессе классификации сетью ResNet карты признаков изображений в мини-батче после вырезания агентом могут иметь разный размер, так как у агента есть возможность вырезать области различных размеров. Для вырезания и приведения вырезанных карт признаков к одному размеру применяется слой ROI Pool [12], который часто применяется при решении задачи локализации изображений.

После вырезания карты признаков изображений вероятнее всего будут уменьшены (если только агентом не будет выбрано действие «ничего не вырезать»), поэтому было бы разумным использовать агента и слой ROI Pool между соседними группами остаточных блоков ResNet, так как там, в архитектуре ResNet, происходит уменьшение размера карт признаков с помощью сверточного слоя с шагом свертки, равным 2. При этом шаг свертки при использовании агента в этом месте станет равным 1. На рисунке 1 показана визуализация изображения из набора данных и его карт признаков, которые являются выходами 1-й, 2-й и 3-й групп остаточных блоков ResNet соответственно.

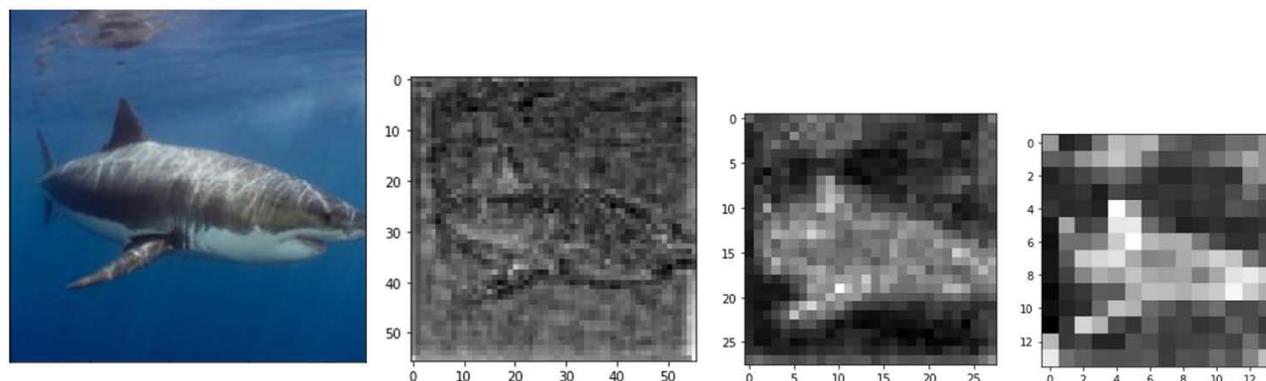


Рисунок 1. Визуализация изображения и выходов 1-й, 2-й и 3-й групп остаточных блоков ResNet.

Из рисунка видно, что после 3-й группы блоков ResNet определяет части объекта, наиболее отличающие его от других, например, у акулы это могут быть плавники. Поэтому использование агента после 3-й группы блоков может позволить в дальнейшем больше сосредоточиться на этих частях. Для упрощения обучения агента будем в дальнейших экспериментах использовать размер шага замощения $\lambda=2$, что дает агенту 26 возможных действий.

Архитектура агента

Базовым прототипом для архитектуры агента была архитектура, состоящая из двух полносвязных слоев с нелинейной функцией активации ReLU (таблица 3). Также были разработаны другие версии архитектуры, уменьшающие размерность входа. В таблице 4 приведена архитектура, уменьшающая размер карт признаков изображения с помощью слоя субдискретизации MaxPool, в таблице 5 – архитектура, уменьшающая число каналов карт признаков изображения с помощью сверточного слоя с размером ядра свертки 1×1 и слоем нормализации BatchNorm с функцией активации ReLU, а в таблице 6 – архитектура, уменьшающая как размер, так и число каналов входных карт признаков изображения.

Для архитектуры из таблицы 4 рассматривались варианты уменьшения размера карт признаков изображения до 7×7 (4.1) и 3×3 (4.2), для архитектуры из таблицы 5 – уменьшение числа каналов до 512 (5.1) и 256 (5.2). В архитектуре из таблицы 6 размер карт признаков изображения уменьшается до 7×7 , а число каналов – до 512. При этом во всех рассматриваемых архитектурах число нейронов в скрытом полносвязном слое равно 512.

Таблица 3. Полносвязная архитектура

Номер слоя	Тип слоя	Функция активации	Размер ядра и число фильтров либо количество нейронов	Размер выхода
0	Вход	–	–	14x14x1024
1	Полносвязный	ReLU	512	512
2	Полносвязный	–	26	26

Таблица 4. Архитектура с уменьшением размера карт признаков

Номер слоя	Тип слоя	Функция активации	Размер ядра и число фильтров либо количество нейронов	Размер выхода
0	Вход	–	–	14x14x1024
1	Слой субдискретизации (MaxPool)	–	–	7x7x1024 / 3x3x1024
2	Полносвязный	ReLU	512	512
3	Полносвязный	–	26	26

Таблица 5. Архитектура с уменьшением числа каналов карт признаков

Номер слоя	Тип слоя	Функция активации	Размер ядра и число фильтров либо количество нейронов	Размер выхода
0	Вход	–	–	14x14x1024
1	Сверточный	–	1x1, 512 / 1x1, 256	14x14x512 / 14x14x256
2	Слой нормализации (BatchNorm)	ReLU	512 / 256	14x14x512 / 14x14x256
3	Полносвязный	ReLU	512	512
4	Полносвязный	–	26	26

Таблица 6. Архитектура с уменьшением размера и числа каналов карт признаков

Номер слоя	Тип слоя	Функция активации	Размер ядра и число фильтров либо количество нейронов	Размер выхода
0	Вход	–	–	14x14x1024
1	Слой субдискретизации (MaxPool)	–	–	7x7x1024
2	Сверточный	–	1x1, 512	7x7x512
3	Слой нормализации (BatchNorm)	ReLU	512	7x7x512
4	Полносвязный	ReLU	512	512
5	Полносвязный	–	26	26

Графики точности классификации моделей на валидационной выборке в зависимости от используемой архитектуры приведены на рисунке 2. Черной пунктирной линией на рисунке обозначена точность классификации базовой модели ResNet. Так как веса ResNet в процессе обучения агента не обновляются, точность классификации базовой модели не изменяется на каждой эпохе.

По графикам на рисунке 2 можно сделать вывод, что лучшие результаты среди разработанных архитектур показали полносвязная архитектура из таблицы 3 и архитектура с уменьшением числа каналов карт признаков до 512 из таблицы 5 (5.1). Также было исследовано влияние числа нейронов в скрытом полносвязном слое на точность классификации для архитектур из таблиц 3 и 5 (5.1). Были рассмотрены варианты значений этого параметра, равные 256, 512 и 1024 нейронов. Результаты эксперимента приведены на рисунке 3.

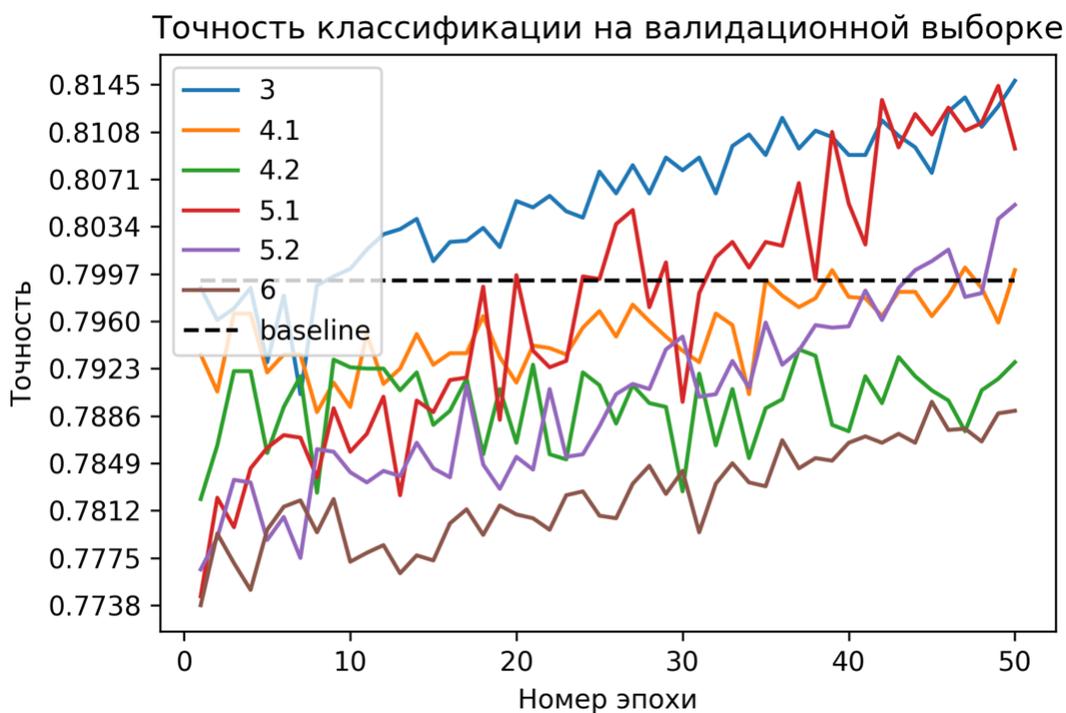


Рисунок 2. Точность классификации моделей в зависимости от используемой архитектуры.

Среди моделей на рисунке 3 наилучший результат показала архитектура 5.1 с 1024 нейронами в скрытом полносвязном слое. Эта архитектура и была выбрана в качестве основной архитектуры агента.

Алгоритм обучения и стратегия выбора действий

В качестве алгоритмов для обучения агента рассматривались алгоритмы обучения с подкреплением DQN [13], REINFORCE [4] и A2C [4]. При этом алгоритм DQN использовался с модификацией воспроизведения опыта [13] с максимально возможным размером буфера, а в функциях потерь алгоритмов REINFORCE и A2C использовался энтропийный бонус [14]. В качестве стратегий выбора действий рассматривались следующие стратегии достижения компромисса между эксплуатацией и исследованием: ϵ -жадная (англ. ϵ -greedy) [4], ϵ -softmax (модификация softmax [4] по аналогии с ϵ -жадной стратегией), ϵ -decay-softmax (по аналогии с ϵ -decay-жадной стратегией [4]) и метод UCB1 [15]. При этом в стратегии ϵ -decay-softmax параметр ϵ уменьшался со временем от значения 1.0 до значения 0.5 в процессе обучения.

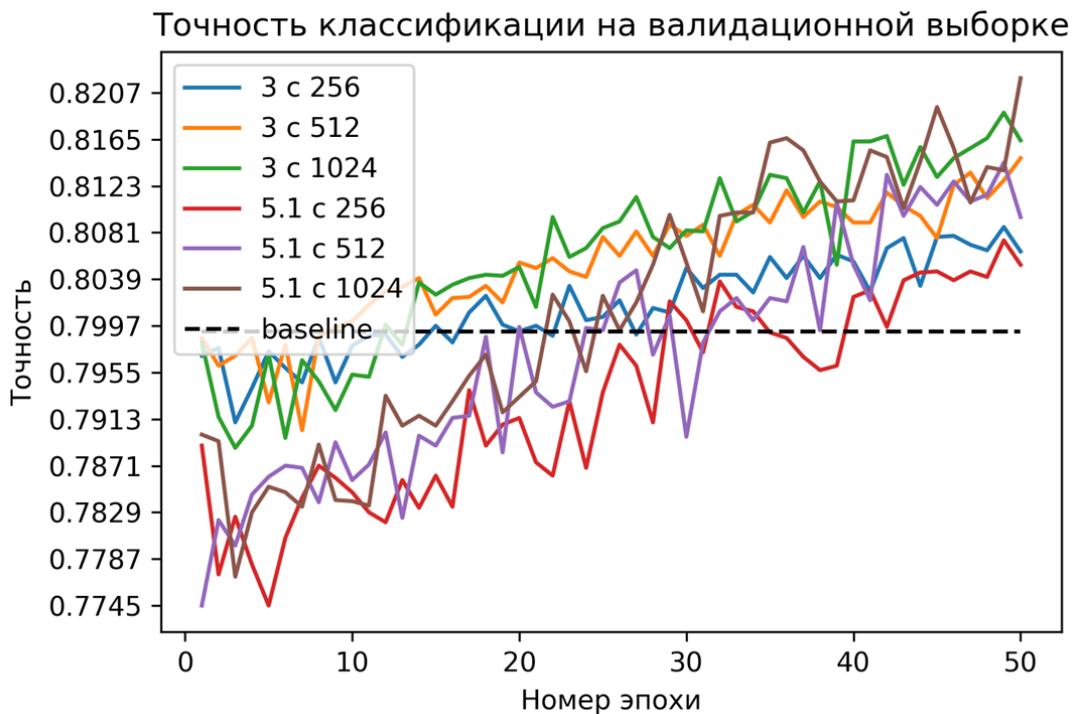


Рисунок 3. Точность классификации лучших моделей в зависимости от числа нейронов в скрытом полносвязном слое.

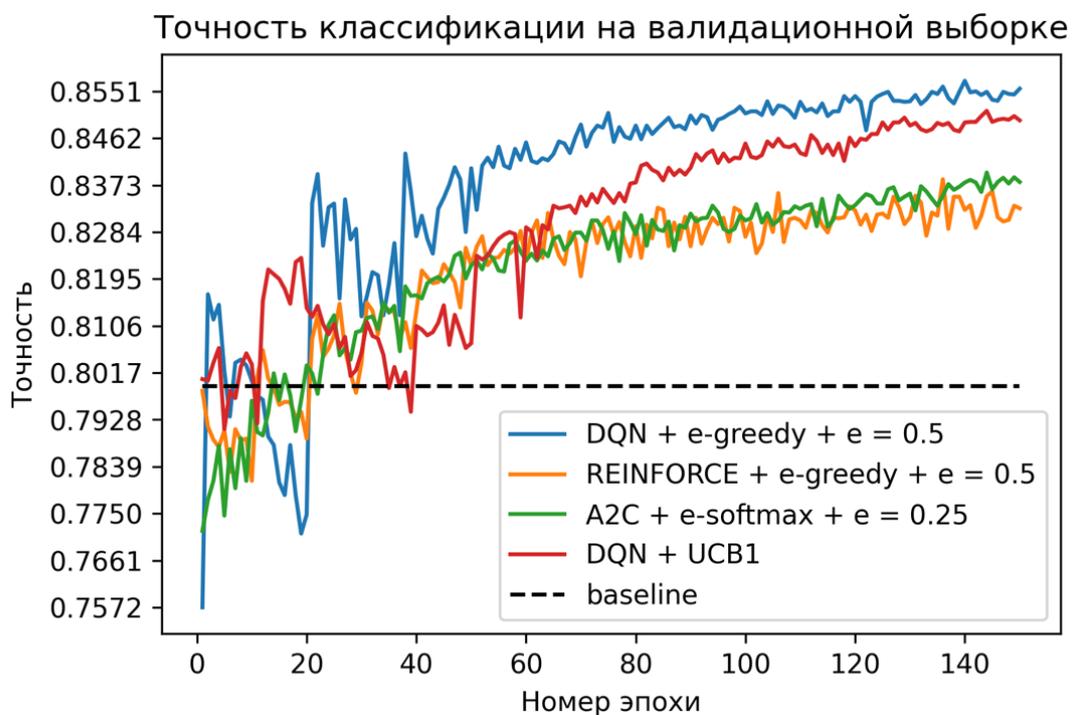


Рисунок 4. Точность классификации лучших моделей.

Для каждого из алгоритмов DQN, REINFORCE и A2C были выбраны наиболее оптимальные стратегии выбора действий, а затем был выбран лучший алгоритм из полученных. Результаты этого эксперимента приведены на рисунке 4 и в таблице 7.

Таблица 7. Результаты классификации лучших моделей

Лучшие модели	Лучшая точность классификации на валидационной выборке
DQN с ϵ -жадной стратегией при $\epsilon = 0.5$	0.8571
DQN с методом UCB1	0.8514
A2C со стратегией ϵ -softmax при $\epsilon = 0.5$	0.8397
REINFORCE с ϵ -жадной стратегией при $\epsilon = 0.5$	0.8392

Таким образом, лучший результат показала модель, обученная с помощью алгоритма DQN и ϵ -жадной стратегии выбора действий с $\epsilon = 0.5$. Она достигла лучшей точности в 0.857 на валидационной выборке, превысив точность базовой модели почти на 0.06, что является почти таким же высоким результатом, как результат классификации «идеального» агента. Результаты работы лучшей модели показаны на рисунках 5 и 6.

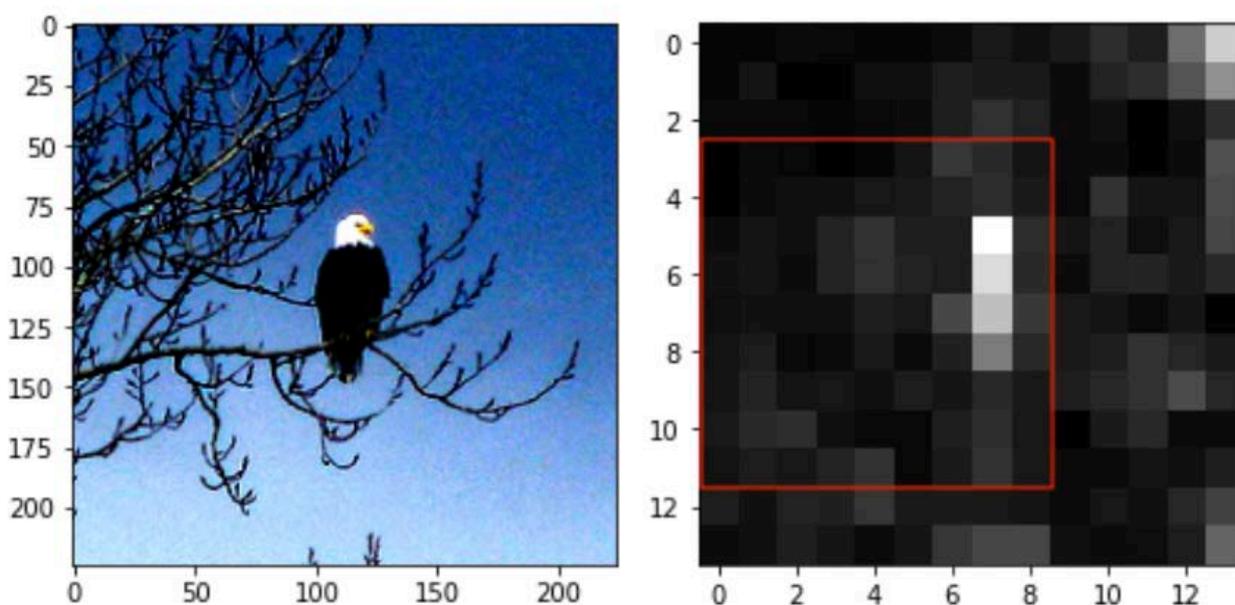


Рисунок 5. Результаты работы лучшей модели на примере.

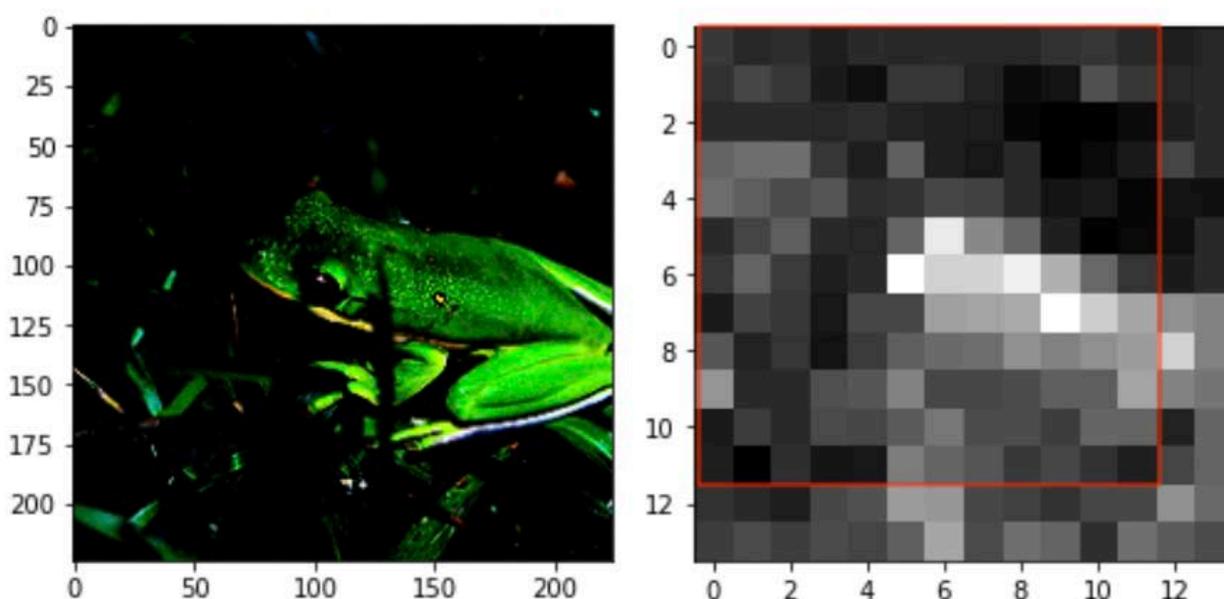


Рисунок 6. Результаты работы лучшей модели на примере.

Использование агента на данных, которые не использовались при обучении ResNet

На практике обучение ResNet на валидационной выборке агента неприменимо. Но можно предложить подход, в котором валидационная выборка используемого набора данных делится на две части: на одной части будет проходить обучение агента, а на другой – его тестирование. При этом сеть ResNet, как и полагается, не будет обучаться на изображениях из валидационной выборки агента и также на изображениях из обучающей выборки агента. Таким образом, агент будет обучаться и тестироваться на изображениях, на которые ResNet будет реагировать похожим образом, так как при обучении сети ResNet эти изображения использоваться не будут.

В качестве валидационной выборки для этого эксперимента использовалась валидационная выборка полного набора данных ImageNet, которая содержит 50 тыс. изображений 1000 различных классов. Эта выборка была поделена на две части: 45 тыс. изображений для обучения агента и 5 тыс. – для тестирования агента. В качестве параметров агента использовались параметры, полученные в ходе предыдущих экспериментов при использовании агента на изображениях, на которых проходило обучение сети ResNet.

По итогам эксперимента точность классификации базовой модели на тестовой части валидационной выборки составила 0.6934, тогда как точность классификации с учетом вырезания агента составила 0.6942. Можно сделать вывод, что предположение о необходимости использования в качестве входа агента выходов группы блоков ResNet из одного распределения является верным, однако обучение агента на изображениях, которые не используются при обучении ResNet, может потребовать большего объема тренировочных данных, а также детального подбора гиперпараметров.

ЗАКЛЮЧЕНИЕ

В статье предложен метод классификации изображений в виде глубокой нейронной сети с использованием обучения с подкреплением. Основная идея метода строится на предположении об увеличении точности классификации за счет вырезания правильной области на изображении в процессе классификации, что приводит к задаче о контекстном многоруком бандите. Это предположение было успешно подтверждено при помощи моделирования работы «идеального» агента.

Были проведены анализ по выбору места расположения агента в архитектуре ResNet, а также эксперименты по выбору архитектуры агента, алгоритма обучения с подкреплением и стратегии выбора действий при обучении. Помимо основных экспериментов также была осуществлена валидация других гиперпараметров нейронной сети. При проведении экспериментов изображения из валидационной выборки агента использовались при обучении сети ResNet, чтобы оценить эффективность работы агента для входа из выходов группы блоков ResNet из одного распределения. При таком подходе лучшая разработанная модель показала точность классификации, значительно превосходящую точность классификации базовой модели, что дает задел для будущих исследований.

Однако на практике использование изображений из валидационной выборки агента для обучения ResNet неприменимо. Поэтому был предложен подход, в котором изображения из обучающей и валидационной выборок агента не использовались при обучении ResNet. В результате этого эксперимента точность классификации с учетом вырезания агента оказалась немного выше точности базовой модели.

Результаты работы агента сильно отличаются в зависимости от того, на каких данных агент обучается. Это может быть связано со сложностью обучения агента на изображениях, на которых не проходило обучение сети ResNet. Дальнейшая работа по развитию разработанного метода видится в более детальном исследовании эффективности работы агента на изображениях, которые не использовались для обучения сети ResNet.

СПИСОК ЛИТЕРАТУРЫ

1. *Goodfellow I., Bengio Y., Courville A.* Deep learning // С.: The MIT Press, 2016, URL: <https://www.deeplearningbook.org/>.
2. *Krizhevsky A., Sutskever I., Hinton G. E.* ImageNet Classification with Deep Convolutional Neural Networks // Advances in neural information processing systems, 2012. Vol. 25, No. 2. P. 1097–1105, DOI: 10.1145/3065386.
3. *Russakovsky O., Deng J., Su H. et al.* ImageNet Large Scale Visual Recognition Challenge // International Journal of Computer Vision. 2015. Vol. 115, No. 3. P. 211–252, DOI: 10.1007/s11263-015-0816-y.
4. *Sutton R. S., Barto A. G.* Reinforcement learning: An introduction // С.: The MIT Press, 2018. URL: <http://www.incompleteideas.net/book/RLbook2020.pdf/>.
5. *Liu X., Xia T., Wang J. et al.* Fully Convolutional Attention Networks for Fine-Grained Recognition // arXiv:1603.06765, 2017.
6. *Li Z., Yang Y., Liu X. et al.* Dynamic Computational Time for Visual Attention // arXiv:1703.10332, 2017.
7. *He K., Zhang X., Ren S., Sun J.* Deep Residual Learning for Image Recognition // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016. P. 770–778, DOI: 10.1109/CVPR.2016.90.
8. PyTorch, 2016, URL: <https://pytorch.org/>.
9. Google Colaboratory, 2017, URL: <https://colab.research.google.com/>.
10. ImageNet Dataset, 2016, URL: <http://image-net.org/>.
11. Fine-Grained Image Classification, 2019, URL: <https://paperswithcode.com/task/fine-grained-image-classification/>.
12. *Girshick R.* Fast R-CNN // Proceedings of the IEEE International Conference on Computer Vision, 2015. P. 1440–1448, DOI: 10.1109/ICCV.2015.169.

13. *Mnih V., Kavukcuoglu K., Silver D. et al.* Playing Atari with Deep Reinforcement Learning // arXiv:1312.5602, 2013.
 14. *Abdolmaleki A., Springenberg J. T., Degraeve J. et al.* Relative Entropy Regularized Policy Iteration // arXiv:1812.02256, 2018.
 15. *Auer P., Cesa-Bianchi N., Fischer P.* Finite-time Analysis of the Multiarmed Bandit Problem // Machine Learning, 2002. Vol. 47, No. 2-3. P. 235–256, DOI: 10.1023/A:1013689704352.
-

IMAGE CLASSIFICATION USING REINFORCEMENT LEARNING

Artem Elizarov¹, Evgenii Razinkov²

Institute of Computational Mathematics and Information Technology, Kazan Federal University

¹artelizar@gmail.com, ²evgeny@razinkov.ai

Abstract

Recently, such a direction of machine learning as reinforcement learning has been actively developing. As a consequence, attempts are being made to use reinforcement learning for solving computer vision problems, in particular for solving the problem of image classification. The tasks of computer vision are currently one of the most urgent tasks of artificial intelligence.

The article proposes a method for image classification in the form of a deep neural network using reinforcement learning. The idea of the developed method comes down to solving the problem of a contextual multi-armed bandit using various strategies for achieving a compromise between exploitation and research and reinforcement learning algorithms. Strategies such as ϵ -greedy, ϵ -softmax, ϵ -decay-softmax, and the UCB1 method, and reinforcement learning algorithms such as DQN, REINFORCE, and A2C are considered. The analysis of the influence of various parameters on the efficiency of the method is carried out, and options for further development of the method are proposed.

Keywords: machine learning, image classification, reinforcement learning, contextual multi-armed bandit problem.

REFERENCES

1. Goodfellow I., Bengio Y., Courville A. Deep learning // C.: The MIT Press, 2016. URL: <https://www.deeplearningbook.org/>.
2. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet Classification with Deep Convolutional Neural Networks // Advances in neural information processing systems, 2012. Vol. 25, No. 2. P. 1097–1105, DOI: 10.1145/3065386.

3. *Russakovsky O., Deng J., Su H. at all.* ImageNet Large Scale Visual Recognition Challenge // *International Journal of Computer Vision*. 2015. Vol. 115, No. 3. P. 211–252, DOI: 10.1007/s11263-015-0816-y.
4. *Sutton R. S., Barto A. G.* Reinforcement learning: An introduction // C.: The MIT Press, 2018. URL: <http://www.incompleteideas.net/book/RLbook2020.pdf/>.
5. *Liu X., Xia T., Wang J. at all.* Fully Convolutional Attention Networks for Fine-Grained Recognition // arXiv:1603.06765, 2017.
6. *Li Z., Yang Y., Liu X. at all.* Dynamic Computational Time for Visual Attention // arXiv:1703.10332, 2017.
7. *He K., Zhang X., Ren S., Sun J.* Deep Residual Learning for Image Recognition // *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. P. 770–778, DOI: 10.1109/CVPR.2016.90.
8. PyTorch, 2016. URL: <https://pytorch.org/>.
9. Google Colaboratory, 2017. URL: <https://colab.research.google.com/>.
10. ImageNet Dataset, 2016. URL: <http://image-net.org/>.
11. Fine-Grained Image Classification, 2019.
URL: <https://paperswithcode.com/task/fine-grained-image-classification/>.
12. *Girshick R.* Fast R-CNN // *Proceedings of the IEEE International Conference on Computer Vision*, 2015. P. 1440–1448, DOI: 10.1109/ICCV.2015.169.
13. *Mnih V., Kavukcuoglu K., Silver D. at all.* Playing Atari with Deep Reinforcement Learning // arXiv:1312.5602, 2013.
14. *Abdolmaleki A., Springenberg J. T., Degraeve J. at all.* Relative Entropy Regularized Policy Iteration // arXiv:1812.02256, 2018.
15. *Auer P., Cesa-Bianchi N., Fischer P.* Finite-time Analysis of the Multiarmed Bandit Problem // *Machine Learning*, 2002. Vol. 47, No. 2-3. P. 235–256, DOI: 10.1023/A:1013689704352.

СВЕДЕНИЯ ОБ АВТОРАХ



ЕЛИЗАРОВ Артем Александрович – магистрант Института вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета.

Artem Aleksandrovich ELIZAROV – undergraduate of the Institute of Computational Mathematics and Information Technology at Kazan Federal University.

email: artelizar@gmail.com



РАЗИНКОВ Евгений Викторович – к. ф.-м. н., ассистент кафедры системного анализа и информационных технологий Института вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета.

Evgenii Viktorovich RAZINKOV – assistant professor at the Institute of Computational Mathematics and Information Technology at Kazan Federal University, PhD.

email: evgeny@razinkov.ai

Материал поступил в редакцию 12 мая 2020 года