

ПРИМЕНЕНИЕ АЛГОРИТМА Q-ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ В ЗАДАЧЕ РАСПОЗНАВАНИЯ ОБРАЗОВ

¹Лясковский В.Л., ²Куликова М.А.

¹МИРЭА - Российский технологический университет, 119454, Россия, г. Москва, проспект Вернадского, 78, e-mail: Rita.kul98@yandex.ru

²Федеральное государственное унитарное предприятие «Всероссийский научно-исследовательский институт «Центр», 123242, г. Москва, ул. Садовая — Кудринская, д. 11, строение 1, e-mail: Dop_big@mail.ru

В данной статье рассматривается алгоритм Q-обучения, основанный на использовании нейронной сети с архитектурой свёрточной нейронной сети. Помимо этого, также рассматривается метод градиентного спуска с оптимизатором Adam и функцией потерь Sparse Categorical Crossentropy. Описана методика обучения агента на примерах из обучающего набора данных MNIST и вычисления градиента с помощью алгоритма обратного распространения ошибки. Также описана стратегия ϵ -greedy, которая необходима для помощи агенту в исследовании различных действий и выбора оптимального в зависимости от текущего состояния среды. Кроме того, статья рассматривает параметры обучения, такие как количество эпох и размер пакета.

Ключевые слова: Q-обучение, свёрточная нейронная сеть, распознавание изображений, агент, высокоуровневая абстракция, метод градиентного спуска, оптимизатор Adam, функция потерь Sparse Categorical Crossentropy, набор данных MNIST, алгоритм обратного распространения ошибки, стратегия ϵ -greedy.

APPLICATION OF Q-LEARNING IN IMAGE RECOGNITION

¹Lyaskovskiy V.L., ²Kulikova M.A.

¹MIREA - Russian Technological University, 119454, Moscow, 78 Vernadsky Avenue, Russia, e-mail: Rita.kul98@yandex.ru

²Federal State Unitary Enterprise "All-Russian Scientific Research Institute "Center", 123242, Moscow, Sadovaya — Kudrinskaya str., 11, building 1, e-mail: Dop_big@mail.ru.

This article discusses the Q-learning algorithm based on the use of a neural network with a convolutional neural network architecture. In addition to that, the gradient descent method with the Adam optimizer and the Sparse Categorical Crossentropy loss function is also considered. The training methodology of the agent on examples from the MNIST training dataset and the computation of gradients using the backpropagation algorithm are described. Furthermore, the ϵ -greedy strategy is described, which is necessary to assist the agent in exploring different actions and choosing the optimal one based on the current state of the environment. Additionally, the article examines training parameters such as the number of epochs and batch size.

Keywords: Q-learning, convolutional neural network, image recognition, agent, high-level abstraction, gradient descent method, Adam optimizer, Sparse Categorical Crossentropy loss function, MNIST dataset, backpropagation algorithm, ϵ -greedy strategy.

Введение

В современном мире задача распознавания образов имеет огромное практическое значение в различных областях, таких как компьютерное зрение, робототехника, медицина и многие другие. Разработка эффективных методов автоматического распознавания образов является одной из главных целей исследователей в области машинного обучения.

В данной статье рассматривается применение Q-обучения с подкреплением в задаче распознавания образов. Q-обучение является одним из методов машинного обучения с подкреплением и широко применяется в задачах, требующих принятия последовательных решений в неопределенной среде. Оно позволяет агенту, основываясь на своем предыдущем опыте, выбирать действия, максимизирующие получаемую награду в задаче.

Основная цель статьи заключается в описании методов и алгоритмов, которые помогут в создании агента, способного классифицировать изображения на основе высокого уровня абстракции и предыдущего опыта для решения задачи на распознавание образов. Для достижения этой цели представлен алгоритм Q-обучения, основанный на использовании свёрточной нейронной сети.

Свёрточные нейронные сети известны своей способностью обрабатывать информацию о структуре и пространственных взаимосвязях образов, что делает их мощным инструментом для задач обработки изображений.

Стратегия ϵ -greedy применяется для повышения разнообразия выбора агента. Это позволяет балансировать исследование новых возможностей и использовать уже известные эффективные действия в процессе обучения.

Для обучения агента рассматривается метод градиентного спуска с оптимизатором Adam и функцией потерь Sparse Categorical Crossentropy. Обучение проводится на обучающем наборе данных, состоящем из изображений, каждому из которых сопоставлена определенная метка класса. Градиенты для корректировки весов нейронной сети вычисляются с помощью алгоритма обратного распространения ошибки.

Также описана методика обучения агента на наборе данных MNIST, содержащем рукописные цифровые изображения.

Кроме того, в статье рассматриваются и анализируются различные параметры обучения, такие как количество эпох (итераций обучения) и размер пакета (batch size). Эти параметры влияют на скорость и качество обучения агента, а их оптимальный выбор может быть зависимым от задач.

Таким образом, данная статья представляет исследование применения Q-обучения с подкреплением в задаче распознавания образов с использованием свёрточной нейронной сети.

Алгоритм Q-обучение

Q-обучение основано на понятии Q-функции, которая оценивает ожидаемую суммарную награду, получаемую агентом при выполнении определенных действий в конкретных состояниях среды. Идея заключается в том, что агент обновляет значения Q-функции на основе получаемых наград и пытается выбирать действия, которые максимизируют эту награду. Таким образом, Q-обучение позволяет агенту обучаться, испытывая различные действия и оценивая их эффективность в разных ситуациях [1].

$$Q_{st,at} = Q_{st,at} + \alpha * (r_t + \gamma * \max_a Q(st+1, a) - Q_{st,at})$$

The diagram shows the Q-learning update equation with labels pointing to its components:

- $Q_{st,at}$ (left): Новое значение (New value)
- $Q_{st,at}$ (middle): Текущее значение (Current value)
- α : Скорость обучения (Learning rate)
- r_t : Вознаграждение (Reward)
- γ : Коэффициент дисконтирования (Discount factor)
- $\max_a Q(st+1, a)$: Расчёт будущего значения (Calculation of future value)

Рисунок 1. Алгоритм Q-обучение

Далее рассмотрим подробнее архитектуру свёрточной нейронной сети (Convolutional Neural Network, CNN) и стратегию ϵ -greedy.

Архитектура свёрточной нейронной сети (CNN)

Свёрточные нейронные сети широко применяются в области компьютерного зрения и обработки изображений. Они специально разработаны для эффективной обработки данных с пространственной структурой, таких как изображения [2].

Архитектура CNN состоит из нескольких слоев, каждый из которых выполняет определенные функции (рис. 2). Свёрточные слои являются основным строительным блоком CNN. Они применяют фильтры (ядра) к входным данным для выделения локальных пространственных признаков. Фильтры перемещаются по изображению, выполняя операцию свертки, которая позволяет выявлять различные шаблоны и особенности. Каждый слой может иметь несколько фильтров, что позволяет извлекать различные признаки на разных уровнях абстракции [2].

Слои объединения выполняют операцию уплотнения (downsampling) данных, уменьшая их размерность и избавляясь от избыточной информации. Типичная операция объединения – Max Pooling, при которой выбирается максимальное значение из заданной области. Это позволяет уменьшить пространственный объем данных и обеспечивает инвариантность относительно небольших переверотов, сдвигов и масштабирования входного изображения.

Полносвязные слои соединяют выходы сверточных слоев для выполнения классификации или регрессии. Они

состоят из нейронов, каждый из которых связан со всеми нейронами предыдущего слоя. Полносвязные слои обрабатывают предварительно извлеченные признаки и вычисляют итоговые предсказания, соответствующие задаче.

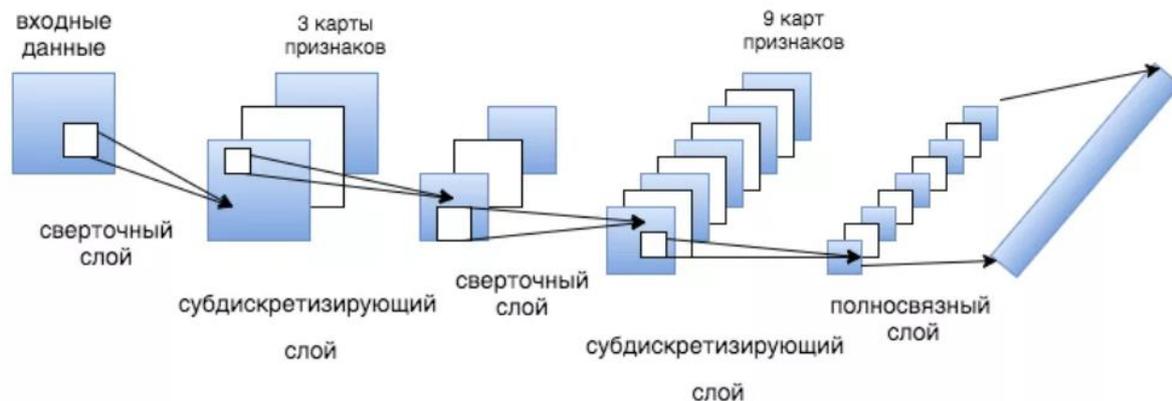


Рисунок 2. Архитектура свёрточной нейронной сети (CNN)

Архитектура CNN может содержать несколько свёрточных слоев, имеющих различное число фильтров и размеры ядер свертки. Затем следуют слои объединения для уплотнения данных, и один или несколько полносвязных слоев, которые производят предсказания классов образов [3].

Стратегия ϵ -greedy является одной из ключевых методик выбора действия в алгоритме Q-обучения с подкреплением. Она используется для балансировки исследования и использования оптимальной стратегии. В этой стратегии агент с вероятностью ϵ выбирает случайное действие (исследование), в то время как с вероятностью $(1-\epsilon)$ выбирает действие с наивысшей оценкой Q-функции (использование). Значение ϵ может быть настроено таким образом, чтобы постепенно снижать вероятность исследования по мере обучения, чтобы агент мог сосредоточиться на использовании оптимальной стратегии [4].

В нашем случае стратегия ϵ -greedy применяется для выбора действий агента, основываясь на значениях Q-функции для каждого действия. Она помогает агенту исследовать различные варианты действий и находить оптимальные стратегии для распознавания образов, это позволяет улучшить качество обучения и достичь более эффективной работы

Применение метода градиентного спуска, оптимизатора Adam и функции потерь Sparse Categorical Crossentropy является важным аспектом в обучении нейросети по распознаванию образов. Опишем каждый из этих элементов подробнее.

Метод градиентного спуска

Метод градиентного спуска является одним из наиболее популярных алгоритмов оптимизации в машинном обучении. Он используется для обновления параметров модели с целью минимизации функции потерь (рис. 3). В контексте Q-обучения метод градиентного спуска применяется для обновления параметров нейронной сети, которая выполняет аппроксимацию Q-функции. Он вычисляет градиент функции потерь по параметрам модели и обновляет их в направлении, противоположном градиенту, с учетом скорости обучения [5].

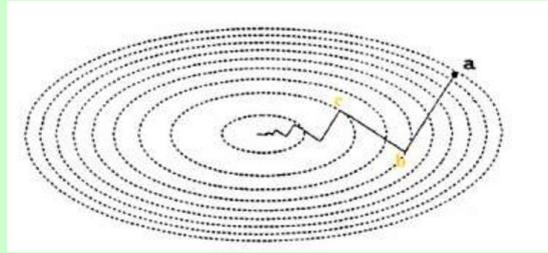
Оптимизатор Adam

Adam (Adaptive Moment Estimation) — это эффективный алгоритм оптимизации, который активно используется в области глубокого обучения. Он комбинирует в себе преимущества метода градиентного спуска и метода Momentum, позволяя более быстро и стабильно сходиться к оптимальным значениям параметров модели. Оптимизатор Adam автоматически адаптирует скорость обучения для каждого параметра, основываясь на опыте предыдущих итераций, что дает ему преимущество в сложных задачах оптимизации [5].

Функция потерь Sparse Categorical Crossentropy

Sparse Categorical Crossentropy (разреженная категориальная перекрестная энтропия) является одной из распространенных функций потерь, применяемых в задачах многоклассовой классификации. В данном контексте эта функция потерь используется для оценки разницы между предсказанными значениями Q-функции и желаемыми значениями. Она измеряет потери, возникающие при неправильной классификации образов, и помогает оптимизировать параметры нейронной сети таким образом, чтобы достичь наилучшей точности классификации.

Метод градиентного спуска



$$\nabla E|_a \Rightarrow b \Rightarrow \nabla E|_b \Rightarrow c \Rightarrow \nabla E|_c \Rightarrow \dots \Rightarrow \nabla E|_{MIN} \cong 0 \Rightarrow fin$$

На каждом шаге осуществляется смещение в направлении, противоположном градиенту функции, на величину, пропорциональную модулю градиента

Рисунок 3. Метод градиентного спуска

Метод градиентного спуска, оптимизатор Adam и функция потерь Sparse Categorical Crossentropy – совместно используются для обучения нейронной сети, выполняющей аппроксимацию Q-функции. Они помогают алгоритму Q-обучения привести к наилучшим решениям в задаче распознавания образов [5].

Обсудим подробнее методику обучения по набору данных MNIST и вычислению градиента с помощью алгоритма обратного распространения ошибки.

Набор данных MNIST (Modified National Institute of Standards and Technology) — это один из наиболее известных и широко используемых наборов данных в области машинного обучения. Он представляет собой набор изображений рукописных цифр от 0 до 9, написанных различными людьми [6].

В наборе данных MNIST содержится обучающая выборка из 60 000 изображений и тестовая выборка из 10000 изображений (рис. 4). Каждое изображение имеет размерность 28x28 пикселей и представлено в оттенках серого. Каждый пиксель хранит значение интенсивности от 0 до 255, где 0 — это черный цвет, а 255 — белый цвет. Изображения были нормализованы и центрированы, чтобы упростить их обработку [6].



Рисунок 4. Набор данных MNIST

Целью набора данных MNIST является обучение моделей машинного обучения распознавать и классифицировать рукописные цифры на изображениях. Это позволяет исследователям и разработчикам применять различные алгоритмы и модели для создания систем распознавания образов и демонстрации их работоспособности. В нашем случае MNIST используется для улучшения распознавания образов, то есть для разработки моделей или алгоритмов, которые научатся классифицировать и распознавать образы на основе набора данных MNIST. На наборе данных MNIST можно обучать различные модели машинного обучения для классификации изображений цифр. Этот набор данных стал основой для множества исследований и сравнений различных алгоритмов машинного обучения.

Алгоритм обратного распространения ошибки является ключевым компонентом многослойных нейронных сетей и используется для вычисления градиента с целью обновления весов сети в процессе обучения. Рассмотрим подробные шаги алгоритма обратного распространения ошибки (рис. 5).

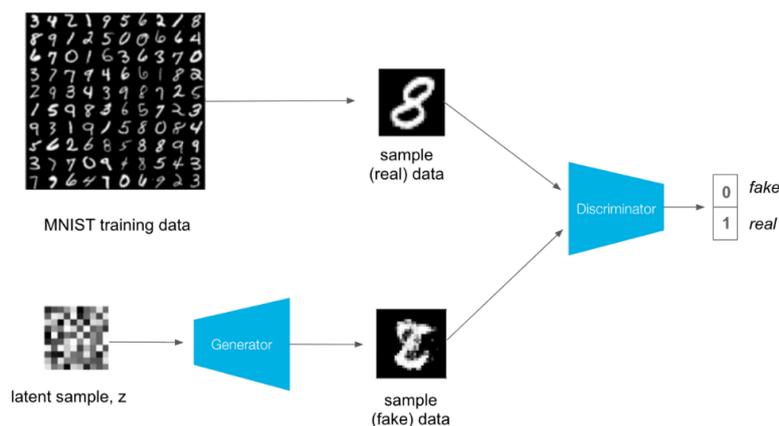


Рисунок 5. Алгоритм распознавания на наборе данных MNIST

Инициализация

Сначала инициализируются веса нейронной сети случайными значениями, а также определяются функция потерь, которая измеряет разницу между предсказанными и ожидаемыми значениями [7].

Прямое распространение

Входные данные передаются через каждый слой нейронной сети, и активации нейронов вычисляются последовательно. Каждый нейрон комбинирует свои входы с весами и применяет некую нелинейную функцию активации для вычисления своего выхода. Прямое распространение выполняется до получения выхода сети [7].

Вычисление ошибки

Разница между предсказанным выходом сети и ожидаемым выходом рассчитывается с помощью выбранной функции потерь. Ошибка обычно измеряется как среднеквадратичное отклонение или перекрестная энтропия [7].

Обратное распространение

На этом этапе начинается вычисление градиента ошибки по отношению к весам сети. Градиент ошибки вычисляется с использованием метода частных производных и цепного правила. Алгоритм движется назад от выходного слоя к входному, вычисляя градиенты для каждого нейрона на каждом слое [8].

Обновление весов

После вычисления градиентов ошибки для всех весов сети происходит их обновление. Обычно используется метод градиентного спуска, который обновляет веса в направлении, противоположном градиенту, с определенным шагом обучения (скоростью обучения).

Повторение процесса

Шаги 2-5 повторяются для каждого обучающего примера в наборе данных до достижения критерия останова, например, определенного числа эпох или достижения необходимой точности модели.

Алгоритм обратного распространения ошибки позволяет нейронной сети корректировать свои веса в процессе обучения, минимизируя ошибку и улучшая свою способность к предсказанию.

В контексте применения Q-обучения с подкреплением для задачи распознавания образов этот алгоритм может использоваться для обучения нейронной сети, которая будет обновлять свои веса на основе ожидаемых выходов и полученных наград. В итоге это может помочь улучшить способность модели к распознаванию образов и улучшить ее общую производительность [9].

Помимо всего прочего, также важно знать про параметры обучения, такие как количество эпох и размер пакета.

Количество эпох

Количество эпох — это количество полных проходов через обучающий набор данных. В каждой эпохе модель проходит через все обучающие примеры один раз. Количество эпох является важным гиперпараметром, который нужно настроить при обучении модели. Оптимальное количество эпох зависит от конкретной задачи и размера обучающего набора данных.

В контексте данного исследования количество эпох будет выбрано в соответствии с требуемым уровнем обучения модели. Обычно начинают с небольшого количества эпох и увеличивают его по мере необходимости, чтобы достичь оптимальной точности модели. Рекомендуется следить за значениями ошибки на валидационном наборе или использовать методы ранней остановки, чтобы определить оптимальное количество эпох и предотвратить переобучение [10].

Размер пакета (batch size)

Размер пакета определяет количество обучающих примеров, которые используются для обновления весов модели в процессе обучения. Вместо того, чтобы обновлять веса после каждого примера, модель обновляет их после каждого пакета. Размер пакета имеет влияние на скорость обучения и потребление памяти.

Размер пакета может быть выбран в соответствии с размером обучающего набора и вычислительными ресурсами. Обычно меньший размер пакета предоставляет большую гибкость для обновления весов и может привести к более стабильному обучению модели. Однако маленькие размеры пакетов могут замедлить процесс обучения из-за частых обновлений весов. Большой размер пакета может ускорить обучение, но может потребовать большего объема памяти. Оптимальный размер пакета зависит от конкретной задачи и аппаратных возможностей [10].

Важно отметить, что подбор оптимальных значений для количества эпох и размера пакета часто требует экспериментирования и настройки. Эти параметры должны быть выбраны таким образом, чтобы достигнуть хорошего баланса между точностью модели, эффективностью обучения и использованием вычислительных ресурсов.

Заключение

Итак, в статье были рассмотрены применение Q-обучения с подкреплением в задаче распознавания образов с использованием свёрточной нейронной сети. Помимо этого, также рассматривался метод градиентного спуска с оптимизатором Adam и функцией потерь Sparse Categorical Crossentropy. Была описана методика обучения агента на примерах из обучающего набора данных MNIST и вычисления градиента с помощью алгоритма обратного распространения ошибки. Также была описана стратегия ϵ -greedy, которая необходима для помощи агенту в исследовании различных действий и выборе оптимального в зависимости от текущего состояния среды. Кроме того, были рассмотрены параметры обучения, такие как количество эпох и размер пакета.

Список литературы

1. Эртель В. Введение в искусственный интеллект. – Москва: Эксмо, 2019. – 448 с.
2. Саттон Р.С., Барто Э.Г. Обучение с подкреплением: введение. – Москва: ДМК Пресс, 2020. – 552 с.
3. Траск Э. Грокаем. Глубокое обучение. – Санкт-Петербург: Питер, 2022. – 352 с.
4. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение. – Москва: ДМК Пресс, 2018. – 652 с.
5. Уатт Д. Машинное обучение: основы, алгоритмы и практика применения. – Санкт-Петербург: BHV-СПб, 2022. – 640 с.
6. Мэрфи К.П. Вероятностное машинное обучение. Введение. – Москва: ДМК Пресс, 2023. – 990 с.
7. Аггарвал Ч. Нейронные сети и глубокое обучение. Учебный курс. – Киев: Диалектика, 2020. – 752 с.
8. Николенко С., Архангельская Е., Кадуринов А. Глубокое обучение. Погружение в мир нейронных сетей. – Санкт-Петербург: Питер, 2020. – 480 с.
9. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. – Москва: ДМК Пресс, 2015. – 400 с.
10. Лекун Я. Как учится машина: Революция в области нейронных сетей и глубокого обучения. – Москва: Интеллектуальная Литература, 2020. – 351 с.

References

1. Ertel V. Introduction to Artificial Intelligence. – Moscow: Exmo, 2019. – 448 pages.
2. Sutton R.S., Barto A.G. Reinforcement Learning: An Introduction. – Moscow: DMK Press, 2020. – 552 pages.
3. Trask E. Grokking Deep Learning. – Saint Petersburg: Piter, 2022. – 352 pages.
4. Goodfellow I., Bengio Y., Courville A. Deep Learning. – Moscow: DMK Press, 2018. – 652 pages.
5. Watt D. Machine Learning: Foundations, Algorithms, and Applications. – Saint Petersburg: BHV-SPb, 2022. – 640 pages.
6. Murphy K.P. Probabilistic Machine Learning: An Introduction. – Moscow: DMK Press, 2023. – 990 pages.
7. Aggarwal C. Neural Networks and Deep Learning: A Textbook. – Kyiv: Dialektika, 2020. – 752 pages.
8. Nikolenko S., Archangel'skaya E., Kadurin A. Deep Learning: Dive into Neural Networks. – Saint Petersburg: Piter, 2020. – 480 pages.
9. Flach P. Machine Learning: The Art and Science of Algorithms that Extract Knowledge from Data. – Moscow: DMK Press, 2015. – 400 pages.
10. Lekun Y. How Machines Learn: A Revolution in Neural Networks and Deep Learning. – Moscow: Intellectual Literature, 2020. – 351 pages.