УДК 004.414.23

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ В МАШИННОМ ОБУЧЕНИИ. ЧАСТЬ 1

♦♦♦♦♦

GENETIC ALGORITHM IN MACHINE LEARNING. PART 1

Мурлина Владислава Анатольевна
кандидат технических наук,
доцент кафедры
информационных систем и программирования,
Кубанский государственный технологический университет
murlina.v@yandex.ru

Савицкий Роман Владиславович
аспирант 2-го курса группы 22-АО-SA1,
Кубанский государственный технологический университет
savitskyrvdev@gmail.com

Аннотация. Данная статья посвящена обзору основных положений генетического алгоритма в машинном обучении, а также области ее применения. Подробно рассматриваются основные понятия и этапы реализации генетического алгоритма, а именно формирование исходной популяции, отбор кандидатов для дальнейшего обучения, воспроизводство следующих поколений с применением механизмов скрещивания и мутации.

Ключевые слова: генетический алгоритм, машинное обучение, наследственность, вариативность, популяция, выборка, отбор, воспроизводство, скрещивание, мутация.

Murlina Vladislava Anatolievna
Candidate of Technical Sciences,
Associate Professor of the Department
Information Systems and Programming,
Kuban State Technical University
murlina.v@yandex.ru

Savitsky Roman Vladislavovich
2nd year Graduate Student
of group 22-AO-SA1,
Kuban State Technical University
savitskyrvdev@gmail.com

Annotation. This article is devoted to an overview of the main provisions of the genetic algorithm in machine learning, as well as the scope of its application. The basic concepts and stages of implementing a genetic algorithm are discussed in detail, namely the formation of the initial population, the selection of candidates for further training, the reproduction of next generations using the mechanisms of crossing and mutation.

Keywords: genetic algorithm, machine learning, heredity, variability, population, sampling, selection, reproduction, crossing, mutation.

Генетический алгоритм (ГА) в машинном обучении представляет собой оптимизационный метод, вдохновленный основными принципами дарвиновской эволюционной теории. В основе алгоритма лежит понятие популяции. Популяцией является группа потенциальных решений поставленной проблемы. Популяциям свойственно развиваться на протяжении поколений посредством процессов, имитирующих естественный отбор [1, с. 24].

Первые упоминания о применении эволюционных процессов в компьютерном моделировании датируется еще 1950-ми годами. Однако, в большей степени современное представление генетического алгоритма основано на работе Джона Холланда, профессора Мичиганского университета, чья книга 1975 года выпуска «Адаптация в естественных и искусственных системах» положила начало исследованиям *ГА как одного из методов машинного обучения*. На данный момент генетические алгоритмы входят в состав более широкой области, которую часто называют **эволюционными вычислениями**.

Стоит уточнить, что генетический алгоритм в области компьютерных эволюционных вычислений основан лишь на общих положениях биологической теории эволюции. Данный метод не предназначен для взаимодействия с научными данными, лежащими в основе к применению теоретической базы на практике.

Несмотря на существование целой науки в области вычислительной биологии, которая решат проблемы эволюционных процессов в реальном мире на основе систематического моделирования и исследований этих процессов, в данной статье будут затронуты лишь основы общей теории и в подробности рассмотрена конкретная реализации генетического алгоритма для решения задач машинного обучения.

Целью генетического алгоритма является нахождение решения поставленной задачи, в которой пространство этих решений может быть настолько объемным, что попытка нахождения решения методом «грубой силы» (например, перебором возмож-

ных решений случайным образом) затратило бы слишком большие объемы вычислительных ресурсов. В этом сравнении главной отличительной особенностью генетического алгоритма лежит наличие способности *оценить*, насколько близко подобрано решение проблемы в конкретный момент времени. Эта оценка могла бы поспособствовать к выбору более верных предположений и, в конечном итоге, прийти к итоговому решению задачи с наименьшем количеством затраченного времени. Из этого определения можно сделать вывод, что предполагаемый ответ «развивается» в процессе выполнения алгоритма.

В качестве примера, который наглядно покажет принцип работы генетического алгоритма для решений практических задач, возьмем одну известную *теорему о бесконечных обезьянах*. Она может быть сформулирована следующим образом: абстрактная обезьяна, случайно нажимающая на клавиши пишущей печатной машинки, рано или поздно напечатает полное собрание сочинений Шекспира в течение неограниченно долгого времени. На практике же, количество возможных комбинаций букв и слов делает вероятность того, что обезьяна действительно напечатает все произведения Шекспира, ничтожно мала. Для сравнения, даже если бы обезьяна начала печатать текст с начала возникновения Вселенной, вероятность того, что к настоящему времени она воссоздала хотя бы пьесу «Гамлет», не говоря уже обо всех произведениях Шекспира, все еще абсурдно маловероятна.

Рассмотрим эту проблему в цифрах: чтобы воссоздать произведения Шекспира, необходима печатная машинка, содержащая всего 27 символов: 26 английских букв и пробел. Вероятность нажатия обезьяной на одну из клавиш, равна 1 к 27. Рассмотрим самую известную фразу из пьесы «Be, or not to be». Ее длина составляет 39 символов, включая пробелы. Вероятность правильного написания первой буквы этой фразы составляет 1 к 27. Вероятность того, что обезьяна правильно напишет второй символ, также равна 1 из 27, то есть шанс написать первые два символа фразы в правильном порядке уже равна 1 к 729. Следовательно, вероятность того, что обезьяна напечатает всю фразу правильно, равна $(1 / 27)^{39}$ [2, с. 393].

Данный пример показывает, насколько малоэффективным является метод перебора решений для получения ответа, ведь даже с учетом того, что замена обезьяны вычислительной машиной, способной набирать миллионы символов в секунду, никак не улучшит ситуацию на практике, так как машине все еще понадобиться немало лет, чтобы с вероятностью в 99 % воссоздать все произведения Шекспира (гораздо больше, чем предполагаемый возраст Вселенной).

Одним из вариантов решения данной проблемы является применение машинного обучения. Генетический алгоритм, который в данном случае начнет свое решение с абсолютно случайных фраз, способен за сравнительно небольшое количество времени найти точное решение посредством симуляции эволюционных вычислений.

Рассмотрим основные положения генетического алгоритма. Он базируется на трех основных принципах:

1. Наследственность – механизм, позволяющий родителям из одного поколения передавать свою генетическую информацию потомкам из следующего поколения.

2. Вариативность – эволюция возможна только в том случае, если все представители определенной популяции отличаются друг от друга по каким-либо характеристикам и свойствам. Без какого-либо разнообразия в популяции потомки всегда были бы идентичны родителям и, соответственно, друг другу. Отсутствие возможности наличия у потомков новых черт отрицало бы способность к развитию.

3. Отбор – механизм, с помощью которого представители одной популяции могут передавать свою генетическую информацию потомкам, а представители другой популяции не имеют такой возможности. Это и называют естественным отбором. Суть естественного отбора заключается в том, какие черты лучше всего подходят для окружающей среды организма. Данные черты повышают вероятность выживания этого организма, и, в конечном итоге, организм размножается. Поэтому превосходством обладают не самые приспособленные, а те, кто «способны воспроизводить потомство». Для печатающих обезьян более подходящая обезьяна та, которая напечатала больше всего символов, присутствующих в фразах из произведений Шекспира.

Определив данные концепции, пошагово рассмотрим реализацию генетического алгоритма на примере теоремы о бесконечных обезьянах (ее упрощенной версии). Все эти шаги можно условно разделить на два этапа:

1. Определение набора условий для инициализации исходной популяции.
2. Последовательное выполнение шагов алгоритма, пока не будет найдена итоговая популяция, содержащее верное решение поставленной задачи.

Создание исходной популяции

Исходной популяцией является совокупности фраз. Под термином «фраза» понимается любая строка символов. При создании популяции фраз применяется принцип вариации. Допустим, для простоты демонстрации, в результате должна получится фраза кошка. Исходная популяция состоит из следующих фраз: ложка, булка и лимон.

Все эти фразы различаются друг от друга количеством и качеством символов, и, если смешивать и сочетать их символы в разном порядке, в итоге никогда не получится исходная фраза «кошка». Данные фразы обладают недостаточным разнообразием символов для нахождения оптимального решения. Однако, если бы популяция состояла из тысячи фраз, сгенерированных случайным образом, есть вероятность, что по крайней мере одна фраза будет иметь в качестве первого символа букву «к», в качестве второго символа букву «о» и так далее. Большая популяция, скорее всего, обеспечит достаточное разнообразие для создания желаемой фразы. Таким образом, первый этап можно описать следующим образом: Создание популяции из N элементов, каждый из которых имеет случайно сгенерированную ДНК.

Здесь ДНК – это набор свойств (геномов), которые описывают, как данный элемент популяции (фраза) выглядит, какую он несет информацию, полезную для создания следующих поколений.

Для организации структуры данных, которая будет хранить значений свойств в каждом объекте, необходимо также определить различия между двумя понятиями в области генетики: генотипом и фенотипом. Генотип любого организма представляет последовательность молекул в ДНК, т. е. набор данных, которые хранятся в объекте. Фенотип, напротив, является выражением этих данных [2, с. 395].

Например, возьмем структуру хранения информации черно-белого изображения. Изображение является набором пикселей, каждый пиксель имеет определенное цифровое значение, представляющее один из оттенков серого – число в пределах от 0 до 255. Генотипом здесь является само значение числа (0, 127, 255…). А то, какой из оттеков серого представляет данное значение – его фенотип.

Весь смысл в том, что эти же цифровые значения мы можем использовать не для представления оттенков серого, но и для представления оттеков красного, зеленого и синего (если объектом исследования является не черно-белое, а цветное изображение). Это даже не обязательно должен быть цвет: можно использовать те же значения для описания длины линии (например, в метрах), веса (в граммах) и так далее.

В рассматриваемом примере ДНК представляют собой список символов, а выражением этих данных является самая строка:

[«к», «о», «ш», «к», «а»] => «кошка».

На данном этапе мы узнали про основные положения ГА, понятие вариативности и наследственности в контексте данного алгоритма, в каких ситуациях он может быть использован в качестве одного из способов решения задач с применением машинного обучения. В следующей части статьи будет рассмотрены следующие этапы реализации ГА – отбор и воспроизводство, за что они отвечают и как с их помощью можно находить оптимальные решения поставленных задач.

Литература

1. Вирсански Эйял. Генетические алгоритмы на Python. Печатная книга. – Изд. ДМК Пресс, 2020. – 286 с.
2. Дэниел Шиффман. Природа кода. Печатная книга. – 2012. – 520 с.

References

1. Virsanski Eyal. Genetic algorithms in Python. Printed book – Ed. DMK Press. – 2020. – 286 p.
2. Daniel Shiffman. The nature of code. Printed book – 2012. – 520 p.