УДК 004.414.23

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ В МАШИННОМ ОБУЧЕНИИ. ЧАСТЬ 2

♦♦♦♦♦

GENETIC ALGORITHM IN MACHINE LEARNING. PART 2

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.

В предыдущей статье были рассмотрены понятие генетического алгоритма в машинном обучении, его основные положения, в каких случаях его следует применять для нахождения оптимального решения в поставленных задачах. В данной статье рассмотрим понятия отбора и воспроизводства в ГА, за что они отвечают и какую роль они играют в процессе обучения.

Принцип отбора включает в себя оценку популяции и определение того, какие элементы могут быть выбраны в качестве родителей для следующего поколения. Процесс выбора можно разделить на два этапа:

1. Вычисление оценки соответствия (*fitness*);
2. Создание выборки.

На первом этапе необходимо разработать функцию оценки соответствия – функцию, которая дает на выходе числовой показатель, описывающий степень соответствия данного элемента к совокупности. Конечно, реальный мир устроен не так. Существам не присваиваются баллы; скорее, они просто размножаются или не размножаются. Однако традиционный ГА направлен на поиск оптимального решения проблемы, поэтому требуется механизм численной оценки любого возможного решения [2, с. 397].

Рассмотрим сценарий из предыдущей статьи. Целевая фраза – *кошка*. Текущая популяция состоит из трех элементов: *ложка*, *булка и лимон*. В качестве функции вычисления оценки соответствия возьмем простейшее выражение:

$$fitness = N, \#(1.1)$$

где *N* – количество соответствующих символов (в данном случае те буквы, которые встречаются и исходной фразе).

Применив функцию к элементам текущей популяции, получим следующий результат:

– *fitness (ложка)* = 3;
– *fitness (булка)* = 2;
– *fitness (лимон)* = 1.

Полученные оценки используются для выбора элементов, способных стать родителями, и помещение их в выборку. Существует множество подходов, каким образом можно осуществить выборку. Одним из самых простых является выбор двух родителей, набравших наибольшую оценку соответствия, после чего они сделают потомков для дальнейшего обучения. Однако этот подход противоречит принципу вариативности, так как с большой долей вероятности следующее поколение будет обладать недостаточным разнообразием генов, и эволюционный процесс прекратится. Поэтому более подходящим в данном случае будет использование вероятностного похода [2, с. 398].

Предположим, что популяция состоит из пяти элементов, каждый из которых имеет свой показатель соответствия (табл. 1):

Таблица 1 – Элементы популяции и их значения оценки соответствия

Элемент	Оценка соответствия
A	2
B	4
C	1
D	2.5
E	0.5

Перед выборкой элементов необходимо нормализовать их оценки. Нормализация стандартизирует диапазон величин значений в пределах от 0 до 1. Для этого используем формулу:

$$normalized(x) = \frac{x}{sum(x)}, \#(1.2)$$

где x – оценка соответствия элемента популяции.

Используем нормализацию для всех элементов популяции и выразим их в процентах (табл. 2):

Таблица 2 – Нормализация оценок соответствия элементов популяции

Элемент выборки	Оценка соответствия	Нормализованные значения	В процентах
A	2	0.2	20 %
B	4	0.4	40 %
C	1	0.1	10 %
D	2.5	0.25	25 %
E	0.5	0.05	5 %

Из таблицы 2 видно, что элементы с наибольшей оценкой соответствия, будут отобраны с наибольшей вероятностью, но при этом не полностью устраняет любые вариации для популяции. Даже элемент с наименьшей оценкой (в данном случае E) имеет некоторый шанс передать свою генетическую информацию следующему поколению (попасть в выборку). Это важно, потому что вполне возможно (и так часто случается), что некоторые элементы с низкими показателями имеют крошечные кусочки генетического кода, которые действительно полезны и не должны быть удалены из популяции.

Воспроизводство

Последним шагом будет использование механизма воспроизводства для создания следующего поколения в популяции. Здесь ключевым является принцип наслед-

ственности – дети наследуют свойства своих родителей. Опять же, здесь можно использовать множество методик. Например, клонирование, довольно эффективная, и в то же время простая в реализации с точки зрения программирования, методика. Выбирается только один родительский элементов, и затем создается точная копия родительского элемента, которая в дальнейшем станет выступать в роли дочернего элемента. Однако это противоречит принципу вариации в ГА. Стандартный же подход заключается в выборе двух родительских элементов и создании дочернего элемента в процессе воспроизводства, который делится на два этапа:

1. Скрещивание
2. Мутация

На этапе скрещивания создается потомок из генетического кода двух его родителей [2, с. 43]. В качестве примера демонстрации возьмем две случайные родительские фразы Фраза потомка С получилась в результате конкатенации первой половины родительской фразы А со второй половиной родительской фразы В (рис. 1):

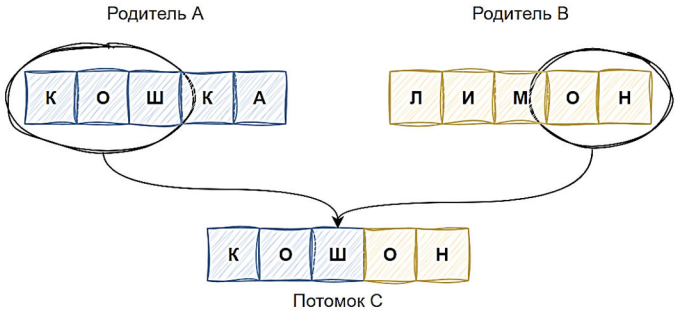

Рисунок 1 – Реализация скрещивания через метод средней точки

Другой способ реализации скрещивания – случайным образом выбрать родительский элемент для каждого символа фразы потомка, как показано на рисунке 2. Вы можете думать об этом как о подбрасывании монеты шесть раз: орел, возьмите символ от родителя А; решка – символ от родителя В:

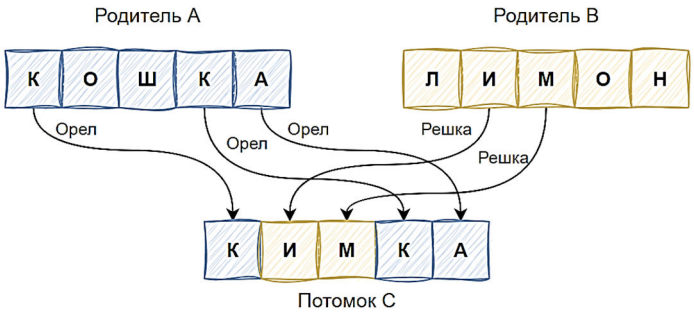

Рисунок 2 – Реализация скрещивания через метод подбрасывания монеты

Данная стратегия существенно не изменит результат при выборе случайной средней точки; однако, если порядок элементов играет роль для вычислений функции соответствия, лучше выбрать первую стратегию. В других задачах случайность, привносимая методом подбрасывания монеты, может принести больше пользы.

После того, как дочерняя ДНК получена через скрещивание, потомка необходимо поместить в следующее поколение. Однако, для улучшения эффективности ГА, можно воспользоваться вторым этапом (необязательным) процесса воспроизводства – *мутацией*. Мутация используется для подтверждения принципа изменчивости. Исходная популяция создавалась случайным образом, что изначально обеспечивало разнородность её элементов [2, с. 47]. Однако эта вариативность ограничена размером популяции, и со временем она сужается в результате отбора. Мутация позволяет внести дополнительное разнообразие в эволюционный процесс. На рисунке 3 показан процесс использования мутации в генетическом алгоритме:

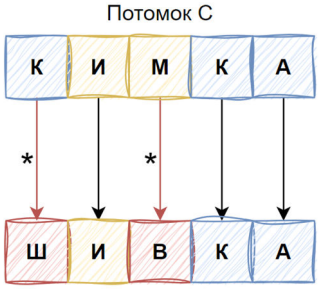

Рисунок 3 – Реализация мутации фразы потомка (частота мутации 20 %)

Существует понятие *частота мутации*. ГА может иметь частоту мутации, равной 5 %, или 1 %, или 0,1 % и так далее. Если уровень частоты мутаций равен 1 %, это означает, что каждый символ фразы-потомка С имеет шанс небольшой шанс мутировать, прежде чем «родиться» в следующем поколении. В данном примере под мутацией предполагается выбор нового случайного символа для итоговой фразы. Вероятность в 1% довольно мала, поэтому большую часть времени мутация вообще не происходить (символы практически не изменяются). Повышение частоты мутации может значительно ускорить процесс обучения за счет внесения разнообразия элементов для всех популяций в процессе обучения.

Как видно, мутация оказывает значительное влияние на поведение эволюционной системы. Очень высокий уровень частоты мутаций (например, 80 %) сведет на нет весь эволюционный процесс и будет больше походить на алгоритм грубой силы. Если большинство генов потомка генерируются случайным образом, нельзя гарантировать, что более подходящие гены будут встречаться с большей частотой в каждом последующем поколении.

Итеративный подход

Процесс отбора (выбора двух родителей) и воспроизводства (скрещивания и мутации) повторяется N раз, пока не появится новая популяция из N дочерних элементов. После этапа № 3 новая популяция потомков становится текущей популяцией (вместо исходной). Затем процесс работы ГА возвращается к шагу 2 и весь процесс обучения элементов популяции повторяется. Вычисляются оценки пригодность каждого элемента, выбираются родительски, которые воспроизводят новые поколение, каждое из которых может содержать в себе решение поставленной задачи. По мере того, как алгоритм будет обучать все больше и больше поколений, система будет развиваться, с каждым шагом приближая нас все ближе и ближе к желаемому решению проблемы.

Заключение

Важно отметить, что на практике, эффективность ГА зависит не только от вычислительной мощности компьютера, на котором будет выполнятся процесс обучения. Большое значение влияет то, какие исходные данные будут задействованы и в каком формате они будут поданы на вход программы. Настоятельно рекомендую вам ознакомится с программной реализацией построения модели машинного обучения на основе генетического алгоритма для решение прикладных задачи на любом современном языке программирования.

Литература

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

References

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