

# ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

*Г.А. Поллак*

Генетические алгоритмы применяются для решения оптимизационных задач с помощью метода эволюции, т.е. путем отбора из множества решений наиболее подходящего. Они отличаются от традиционных методов оптимизации следующими свойствами [1]:

1. Обрабатывают не значения параметров задачи, а их закодированную форму.
2. Осуществляют поиск решения исходя из некоторой популяции.
3. Используют только целевую функцию, а не ее производную.
4. Алгоритмы являются стохастическими.

Генетические алгоритмы для обучения нейросети применяются как альтернатива методу обратного распространения ошибки. Целью обучения является минимизация функции стоимости  $E(n) = \frac{1}{2} e_k^2(n)$ , где  $e_k(n) = d_k(n) - y_k(n)$  – ошибка,  $d_k(n)$  – желаемый выход нейросети,  $y_k(n)$  – реальный выход нейросети,  $n$  – номер итерации.

Обучение методом обратного распространения ошибки сводится к подбору значений весов прямонаправленной нейронной сети, основываясь на принципах наискорейшего спуска. Один из основных недостатков этого классического алгоритма состоит в возможном попадании в локальные минимумы функции стоимости.

Применение генетического алгоритма позволяет избежать этой опасности. Рассмотрим в качестве примера нейронную сеть, которая реализует операцию XOR (рис. 1).

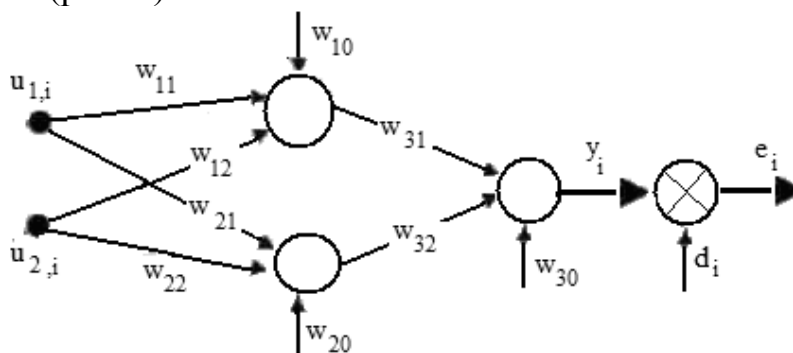


Рис. 1. Нейронная сеть, реализующая операцию XOR

Для сети требуется подобрать оптимальные веса  $w_{11}, w_{11}, w_{12}, w_{21}, w_{22}, w_{31}, w_{33}, w_{10}, w_{20}, w_{30}$ , минимизирующие значение целевой функции, определяемой среднеквадратичной погрешностью  $E = \frac{1}{2} \sum_{i=1}^4 (d_i - y_i)^2$ .

Параметрами задачи являются веса, которые определяют точку пространства поиска и, следовательно, представляет собой возможное решение.

Блок-схема, реализующая классический генетический алгоритм, показана на рис. 2.

Допустим, что веса принимают действительные значения из интервала  $[-1, +1]$ . Тогда каждая хромосома будет комбинацией из 9 двоичных последовательностей (генотипов), кодирующих конкретные веса. Соответствующие им фенотипы представлены значениями отдельных весов, т.е. множествами соответствующих действительных чисел из интервала  $[-1, +1]$ . Длина хромосом зависит от условия задачи. Допустим, что требуется найти решение с точностью до  $q = 2$  знаков после запятой для каждой переменной  $w_{ik}$ . Следовательно интервал  $[a, b]$  необходимо разбить на  $(b - a) \cdot 10^q$  одинаковых подынтервалов. Это означает применение дискрети-

зации с шагом  $r = 10^{-q}$ . Наименьшее натуральное число  $m_i$ , удовлетворяющее неравенству  $(b - a) \cdot 10^q \leq 2^m - 1$  определяет необходимую и достаточную длину двоичной последовательности, требуемой для кодирования числа из интервала  $[a, b]$  с шагом  $r$ . Результаты расчетов приведены в табл. 1.

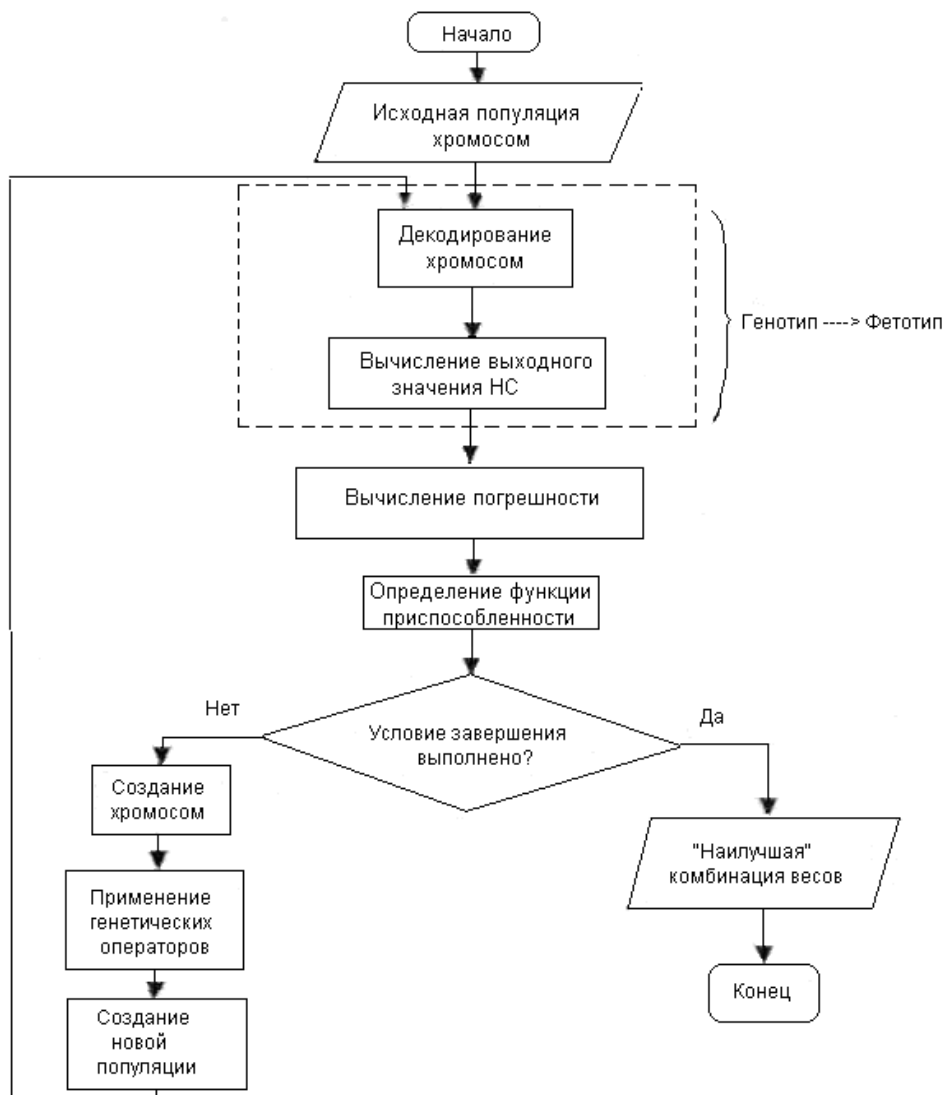


Рис. 2. Блок-схема классического генетического алгоритма

Таблица 1

a	b	n	r	q	m
-1	1	256	0,007843	2	7,65105

Как видно из табл. 1 длина двоичной кодирующей последовательности составляет 8 бит.

При кодировании действительных чисел в качестве значения гена берется целое число, определяющее номер интервала (используется код

Грея). В качестве значения фенотипа примем число, являющееся серединой этого интервала.

Исходная популяция хромосом задается случайным образом (табл. 2).

Таблица 2

Исходная популяция						
	Код Грея		Номер интервала	Левая граница	Правая граница	Среднее значение
w <sub>11</sub>	11001100G	CCh	204	0,5922	0,6000	0,5961
w <sub>12</sub>	01100010G	62h	98	0,2392	0,2314	0,2353
w <sub>21</sub>	00110011G	33h	51	0,6078	0,6000	0,6039
w <sub>22</sub>	01110111G	77h	119	0,0745	0,0667	0,0706
w <sub>31</sub>	00100010G	22h	34	0,7412	0,7333	0,7373
w <sub>32</sub>	01010101G	55h	85	0,3412	0,3333	0,3373
w <sub>10</sub>	01100010G	62h	99	0,2314	0,2235	0,2275
w <sub>20</sub>	10100010G	A2h	162	0,2627	0,2706	0,2667
w <sub>30</sub>	00101010G	2Ah	42	0,6784	0,6706	0,6745

Отклик сети на входные сигналы  $y_1 = 0,4592$ ,  $y_2 = 0,2631$ ,  $y_3 = 0,3463$ ,  $y_4 = 0,3551$ , среднеквадратичная ошибка  $E = 0,2704$ . При вычислении выходных значений нейросети использовалась логистическая функция активации с параметром скорости обучения  $\eta = 1$ .

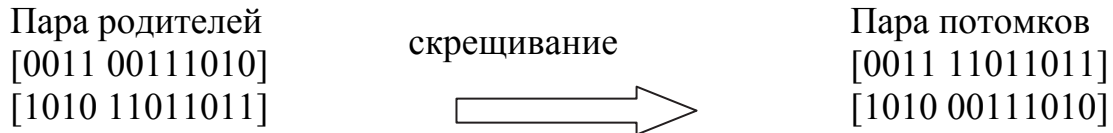
Наибольшую роль в успешном функционировании алгоритма играет этап отбора родительских хромосом для создания новой популяции. Наиболее часто используется метод отбора, называемый рулеткой. Однако вследствие некоторых недостатков [1] этот метод плохо приспособлен для решения задачи минимизации. Поэтому предлагается использовать турнирный метод. Суть метода заключается в следующем. Все особи популяции разбиваются на подгруппы по 2–3 особи в каждой. Выбор родителей осуществляется случайно с вероятностью, меньшей 1.

В классическом генетическом алгоритме применяются два основных генетических оператора: оператор скрещивания и оператор мутации. Вероятность скрещивания достаточно велика (обычно  $0,5 \leq p_c \leq 1$ ), вероятность мутации устанавливается весьма малой (чаще всего  $0 \leq p_m \leq 0,1$ ). Это означает, что скрещивание в классическом алгоритме производится практически всегда, тогда как мутация – достаточно редко.

Оператор скрещивания действует следующим образом.

- из популяции с вероятностью  $p_c$  выбираются две особи, которые включаются в состав временной родительской популяции;
- определяется (также случайным образом) точка скрещивания  $l_k$ ;
- выполняется конкатенация части первого и второго родителя.

Например для  $l_k = 4$



Оператор мутации с вероятностью  $p_m$  изменяет значение гена в хромосоме на противоположное. Вероятность мутации может вычисляться случайным выбором числа из интервала  $[0, 1]$  для каждого гена и отбором для выполнения этой операции тех генов, для которых разыгранное число оказывается меньше или равным значению  $p_m$ .

Хромосомы, полученные в результате применения генетических операторов к хромосомам временной родительской популяции, включаются в состав новой популяции. Она становится текущей популяцией для данной итерации генетического алгоритма. На каждой очередной итерации рассчитывается значение функции приспособленности для всех хромосом этой популяции, после чего проверяется условие остановки алгоритма. В качестве такого условия применяется либо ограничение на максимальное число эпох функционирования алгоритма, либо определяется сходимость алгоритма путем сравнения значений функции приспособленности популяции на нескольких эпохах. При стабилизации этого параметра алгоритм останавливается.

Следует подчеркнуть, что алгоритм обратного распространения ошибки, как правило, выполняется быстрее генетического, который просматривает все множество возможных решений. Однако градиентный метод не всегда приводит к ожидаемому результату, который зависит от выбора начальной точки. Кроме того, принципиальным недостатком метода обратного распространения ошибки является «застревание» в локальных оптимумах. Можно предложить гибридный метод обучения нейронных сетей, когда генетический алгоритм применяется для выбора начальной точки (исходного множества весов на каждой итерации), а затем применяется метод обратного распространения ошибки. Гибридный метод может позволить найти глобальное решение за приемлемое время.

#### Библиографический список

1. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский; пер. с пол. И.Д. Рудинского. – М.: Горячая линия–Телеком, 2004. – 383 с.