

Конструирование искусственных нейронных сетей с помощью меметических алгоритмов

А.А. Рыбаков¹, Т.Т. Сохибов², С.С. Шумилин³

^{1,3}МСЦ РАН – филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия;

²МГУ им. М.В. Ломоносова, Москва, Россия

e-mails: ¹rybakov@jsgcc.ru, ²dj.tolib.tj@mail.ru, ³shumilin@jsgcc.ru

Аннотация. В статье рассматривается семейство меметических алгоритмов конструирования искусственных нейронных сетей специального вида. В качестве целевых сетей рассматриваются сети, представленные ациклическими графами с фиксированным количеством входных и выходных нейронов. Меметический алгоритм представляет собой эволюционный алгоритм развития популяции нейронных сетей с итерационным частичным обучением, а также с выживанием, размножением и мутацией особей. В качестве завершения естественного отбора принимается момент, когда лидер популяции отвечает верно на все тестовые данные из целевого набора. Рассматриваемый алгоритм позволяет автоматически находить конфигурации нейронных сетей с меньшим количеством нейронов и связей, а также обладающие лучшими характеристиками по скорости обучения.

Ключевые слова. Искусственные нейронные сети, эволюционные алгоритмы, генетические алгоритмы, меметические алгоритмы, популяция, мутация.

Введение

В настоящее время искусственные нейронные сети находят широкое применение в различных областях жизни [1]. Распознавание и классификация изображений применяется в медицине (повышение осведомленности при диагностировании), безопасности (вопросы идентификации личности и определение потенциальных угроз), в геологоразведке, при анализе рукописных текстов. В экономике с помощью нейросетей производится предсказание состояния рынков, выявление коррупционных схем и оптимизация денежных потоков. Активно развивается рынок беспилотных объектов, начиная с летательных аппаратов и заканчивая автомобилями. Нейронные сети все глубже проникают с социальную сферу, они являются неотъемлемой частью маркетинга, адресной рекламы, автоматической рубрикации новостных лент.

Использование нейронных сетей позволяет существенно упростить процесс построения сложных нелинейных зависимостей, так как для этого требуется лишь задать структуру сети, а определение самих параметров (весов и смещений) выполняется автоматически в процессе обучения [2].

Однако само определение изначальной структуры сети может оказаться тонким местом построения сети. Так, при создании многослойной сети существует проблема как слишком простой структуры, так и слишком

сложной. В случае недостаточного количества слоев и нейронов в сети существует риск, что создаваемая сеть окажется неспособной обучиться для решения поставленной задачи. В случае же задания слишком большого количества слоев и нейронов существует опасность слишком медленного обучения или попадания сети в локальные экстремумы, из которых она будет неспособна выбраться в процессе дальнейшего обучения (неконтролируемое увеличение по модулю значений весов и смещений).

Для автоматического выбора подходящей начальной структуры нейронной сети предлагается использование эволюционного подхода, при котором в одной популяции одновременно существует несколько отличающихся друг от друга нейронных сетей, которые проходят параллельное частичное обучение и соревнуются друг с другом за право занимать место в популяции. При этом наиболее слабые особи вымирают, а более сильные получают право на размножение. Такие алгоритмы также называют меметическими, так как они являются объединением генетических алгоритмов и алгоритмов обучения нейросетей [3]. Модель эволюционной системы искусственных нейронных сетей рассматривается в данной статье.

Рассматриваемый тип нейронных сетей

В данной работе рассматриваются искус-

ственные нейронные сети, каждая из которых представлена ориентированным графом. В данном графе нейрон является вершиной, а связь между двумя нейронами – ребром. Ребра, входящие в вершину, называются входами соответствующего нейрона, по ним сигналы попадают в нейрон. Ребра, выходящие из вершины, аналогично, называются выходами соответствующего нейрона. В графе выделено фиксированное количество нейронов без входов, это начальные нейроны, или нейроны, относящиеся к сенсору сети (через них входной сигнал попадает в сеть). Также выделено фиксированное количество нейронов без выходов, это конечные нейроны, или нейроны, относящиеся к актуатору сети (сигналы на данных нейронах снимаются для получения ответа сети). Не рассматриваются сети, содержащие циклы. Таким образом сеть допускает нумерацию нейронов такую, что номер любого рассматриваемого нейрона больше номера любого его предшественника и меньше номера любого его последователя. При данной нумерации возможна последовательная обработка всех нейронов сети в порядке возрастания номеров.

Каждый нейрон, получив сигналы со всех своих входящих ребер, выполняет преобразование

$$z = b + \sum s_i w_i$$

$$a = \sigma(z)$$

и распространяет выходной сигнал по всем своим выходящим ребрам. Здесь s_i – входящий сигнал, w_i – вес входящего ребра, соответствующего данному сигналу, суммирование ведется по всем входящим ребрам, b – смещение, z – суммарный сигнал до применения активирующей функции, a – выходной сигнал после применения активирующей функции, в качестве активирующей функции берется сигмоидальная функция

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Будем рассматривать обучение нейронной сети с учителем, то есть когда предусмотрено наличие тестового набора, в котором каждому вектору входных значений x соответствует известный вектор $y(x)$. Если при этом нейронная сеть на поданный вектор x выдает ответ $a(x)$, то функцией ошибки (или функций стоимости) на этом элементе данных будем называть значение величины $0.5 \|y(x) - a(x)\|^2$. Значением функции полной стоимости при рассмотрении всех входных векторов будет являться следующее выражение (N обозначает конкретную конфигурацию

нейросети вместе с набором весов и смещений)

$$T(N) = \frac{1}{2|X|} \sum_{x \in X} \|y(x) - a(x)\|^2$$

где X – множество векторов входных данных из тестового набора [4].

Введем также другие обозначения. $R(N)$ – доля правильных ответов нейросети, при обучении с учителем идеальная верхняя граница данной величины равна единице. Порядком и размером сети будем называть порядок (количество вершин) и размер (количество ребер) соответствующего графа: $v(N)$, $\varepsilon(N)$.

Обучение рассматриваемых сетей будем проводить в помощь метода обратного распространения ошибки с фиксированной скоростью обучения. При этом количество итераций обучения будем запоминать в сети, данную величину обозначим $L(N)$ [5].

Описание семейства меметических алгоритмов

В основе генетических алгоритмов [6,7] лежит понятие популяции. Популяция – это множество особей, которые одновременно сосуществуют и соревнуются за право в этой популяции остаться. В процессе естественного отбора наиболее слабые особи популяции вымирают, а наиболее сильные дают потомство. В данной работе в качестве размножения особи будет пониматься просто ее клонирование. Таким образом восстановление популяции состоит всего лишь из клонирования ее лидеров и внесения с их структурой мутаций.

Одним из наиболее важных вопросов является критерий, по которому следует определять сильных и слабых особей. Конечно основным решающим показателем является параметр $R(N)$. Как только лидер популяции достигает значения 1.0 данного параметра, алгоритм может останавливаться. Однако, как показывают эксперименты, популяция способна довольно быстро сравняться по данному показателю и дальше должны учитываться другие параметры, такие как $T(N)$, $L(N)$, $v(N)$, $\varepsilon(N)$ и их комбинации. При этом следует отметить, что нейронная сеть не является статическим объектом во времени. При смене поколений каждая сеть частично обучается, поэтому только что созданные сети не следует оценивать наравне с более старыми. Поэтому в процессе тестирования алгоритмов была принята эвристика, моделирующая поведение защиты потомства:

никакая сеть не может быть уничтожена, пока она не достигла среднего возраста по-

пуляции (то есть каждая сеть получает шанс на начальное развитие).

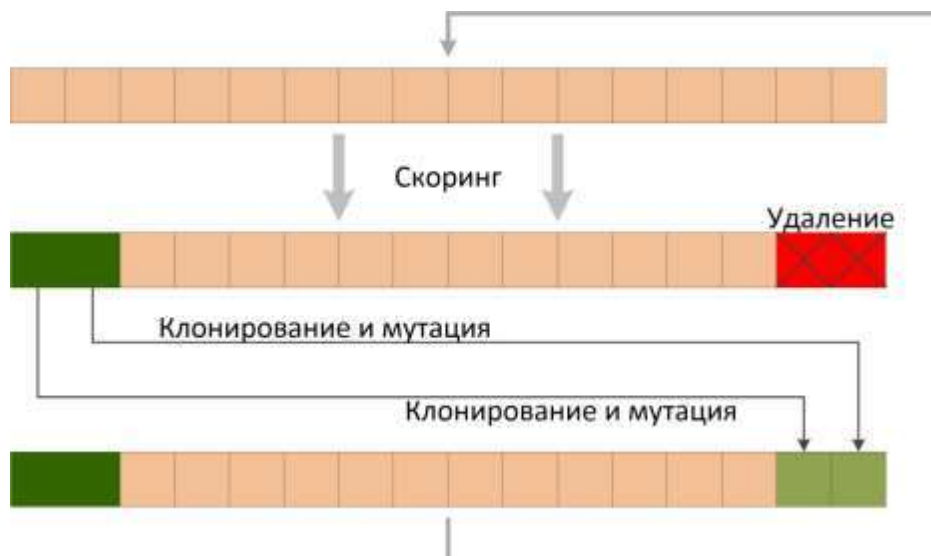


Рис. 1. Цикл простейшего генетического алгоритма: скоринг – удаление слабых особей – восстановление популяции – мутация.

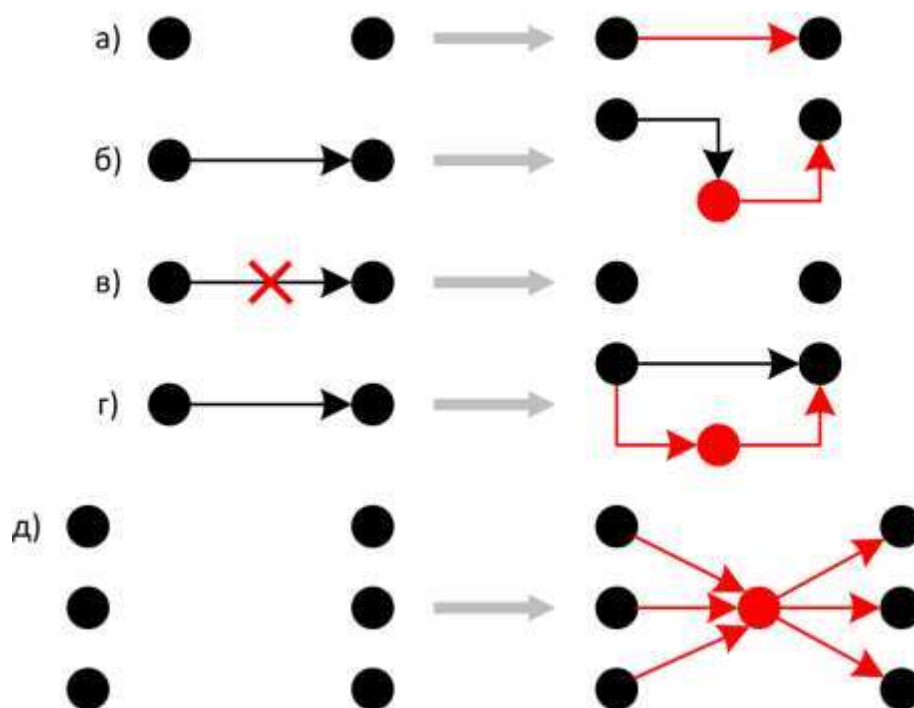


Рис. 2. Примеры используемых мутаций: а) добавление ребра между двумя нейронами; б) добавление нового нейрона на ребро (и еще одного ребра); в) удаление ребра (с возможным последующим удалением висячих нейронов); г) добавление дополнительной связи между двумя нейронами с помощью добавления нейрона и двух ребер; д) супермутация – связь двух подмножеств нейронов через новый нейрон.

С учетом замечания о защите недавно созданных сетей отработка поколения в генетическом алгоритме состоит из следующей

последовательности шагов: проведение скоринга (вычисление функций $R(N)$ и $T(N)$ для каждой сети); определение наиболее слабых

особей и их уничтожение; определение наиболее сильных особей и их клонирование для восстановления популяции (при этом в процессе клонирования производятся мутации с некоторой долей вероятности) (рис. 1).

Обычно в генетических алгоритмах структура и поведение особей кодируется в геноме, над которым и выполняются действия по клонированию, скрещиванию и мутациям. Такой подход можно наблюдать в системе DXNN [8]. Однако по сути это прямое кодирование нейросети, поэтому было принято решение в рамках данной работы вообще не использовать геном, а сразу выполнять мутации на готовом объекте, используя атомарные преобразования направленного графа по подобию преобразований из [9].

В качестве мутаций (рис. 2) были рассмотрены такие элементарные преобразования графа нейросети, как добавление и удаление ребра, добавление на ребро нового нейрона, добавление нового нейрона с одним входным и одним выходным ребром. Однако применение лишь элементарных преобразований не привело к достижению показателя $R(N) = 1.0$ даже на большом количестве поколений. Данный факт может свидетельствовать о том, что не всегда существует последовательность мелких атомарных преобразований, переводящих элементарный объект в процессе эволюционного развития в абсолютно оптимальный, который может достигнуть глобального максимума целевой функции.

В данном факте в принципе нет никакого противоречия, так как в процессе естественной эволюции в природе перед живым существом не ставится задача достижения глобального максимума, а стоит более приземленная задача — выживание в текущих локальных условиях в составе своей популяции [10]. В нашей же задаче конструирования оптимальной нейронной сети для данного тестового набора при обучении с учителем такой исход неприемлем.

Для преодоления данного препятствия была введена так называемая супермутация (рис. 2, д), в процессе которой добавляется один нейрон, связывающий два больших множества нейронов между собой. При этом для сохранения баланса была повышена ве-

роятность мутации, при которой происходит удаление ребра. Это помогло после возникновения супермутации быстро удалять избыточные ребра, оставляя только полезные связи, которые приводят к росту функции $R(N)$.

Применение к модельной задаче

В качестве модельной задачи была рассмотрена простая задача приближения логической функции с 5 входами и 2 выходами. Функция по сути принимает на вход число в двоичном виде (5 двоичных разрядов соответствуют числам от 0 до 31) и возвращает пару (1.0, 0.0) если число простое, и пару (0.0, 1.0) в противном случае. В процессе тестирования размер популяции брался равным 50, а количество вымирающих особей — не более двух (могло быть и менее, если в популяции слишком много молодых особей).

Также для борьбы с флуктуациями функции $T(N)$ лидера популяции было принято два дополнительных соглашения, которые в общем случае не гарантируют сходжение обучения. Во-первых, частичное обучение любой сети в рамках отработки текущего поколения не может завершиться, пока значение функции $T(N)$ для этой сети не уменьшится. Во-вторых, лидер популяции вообще не обучается. Если не придерживаться данного соглашения достаточно рано в популяции образуется компактная элита с одним показателем $R(N)$, и очень близкими значениями $T(N)$, и представители этой элиты начинают бесконечно поочередно становиться лидерами. После же принятия соглашения о монотонном убывании функции $T(N)$ данные проблемы исчезают, и неоптимальный лидер популяции рано или поздно оказывается смещенным своей копией, прошедшей через супермутацию.

Процесс эволюционного отбора на рассмотренной задаче сходится чуть более, чем за 1000 поколений. На рис. 3 представлена диаграмма продолжительности жизни всех особей популяции. На данной диаграмме видно, что практически в самом начале появились лидеры-долгожители, а остальные особи достаточно быстро вымирали (эпоха низких показателей количества верных ответов $R(N)$).

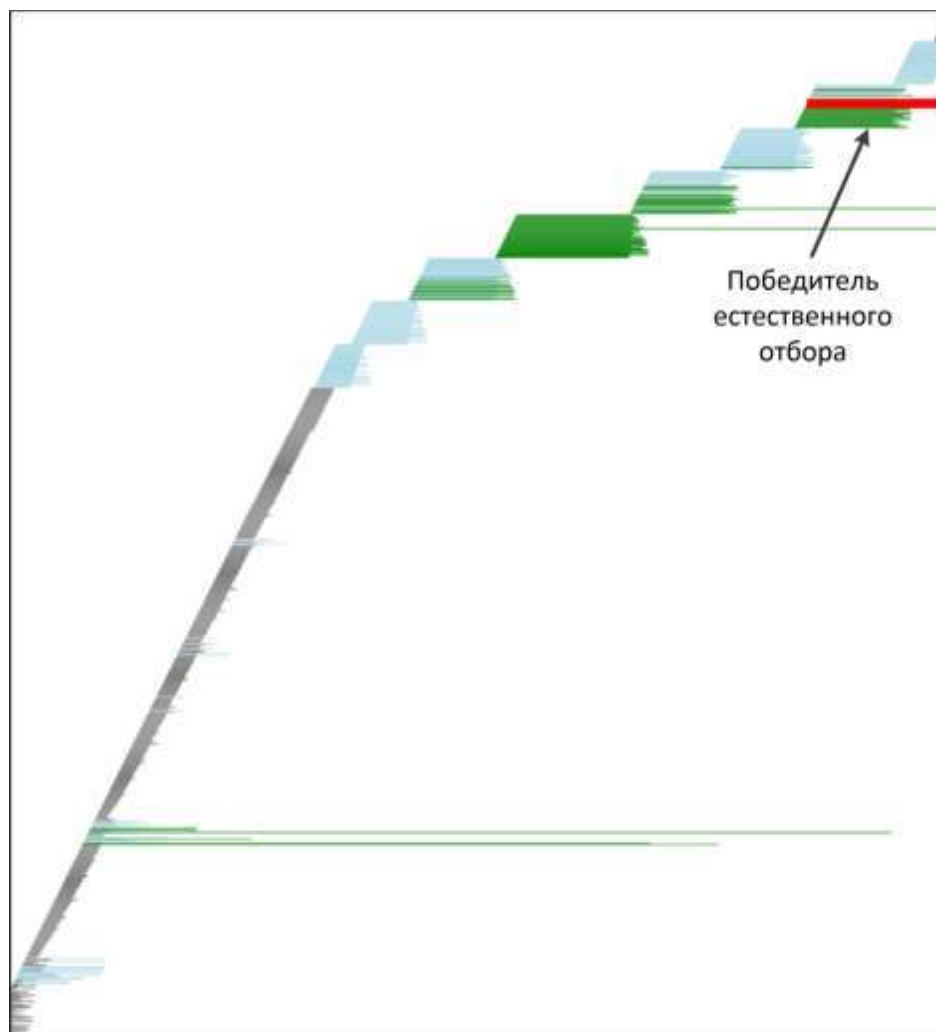


Рис. 3. Диаграмма, показывающая продолжительность жизни особей популяции. По вертикали показаны особи в порядке их рождения. Длина горизонтальной линии – время жизни особи (зеленые – время жизни более 100 поколений, голубые – от 30 до 100, серые – менее 30).

При возрастании показателя $R(N)$ лидера дальнейшее улучшение этой характеристики происходит только вследствие возникновения супермутаций, вероятность которых задана достаточно редкой. По сути каждая "ступенька" на диаграмме отражает реакцию на появление супермутации. В конце концов одна из особей, переживших супермутацию (отмечено на диаграмме красным), достигает показателя $R(N) = 1.0$, и алгоритм заканчивает работу. На рис. 4. представлены графики, на которых видно, как изменялись следующие показатели лидера популяции: возраст особи, а также $L(N)$, $R(N)$, $T(N)$, $v(N)$, $\varepsilon(N)$.

Для сравнения также было проведено обучение нескольких многослойных сетей. Результаты их обучения, а также результаты обучения победителя естественного отбора приведены в табл. 1. Видно, что первая многослойная нейросеть, достигшая показателя

$R(N) = 1.0$, больше по размеру и медленнее в плане обучения, чем лидер популяции.

Табл. 1. Сравнение показателей лидера популяции с обычными многослойными сетями.

Описание сети	$L(N)$	$R(N)$	$v(N)/\varepsilon(N)$
лидер	6700	1.0	11/25
[5, 2]	100 000	0.812 5	7/10
[5, 1, 2]	100 000	0.875	8/7
[5, 2, 2]	100 000	0.906 25	9/14
[5, 3, 2]	100 000	0.968 75	10/21
[5, 4, 2]	100 000	0.968 75	11/28
[5, 5, 2]	100	0.968	12/35

2]	000	75	
[5, 6, 2]	74 866	1.0	13/42
[5, 7, 2]	6 728	1.0	14/49

[5, 8, 2]	6 043	1.0	15/56
--------------	-------	-----	-------

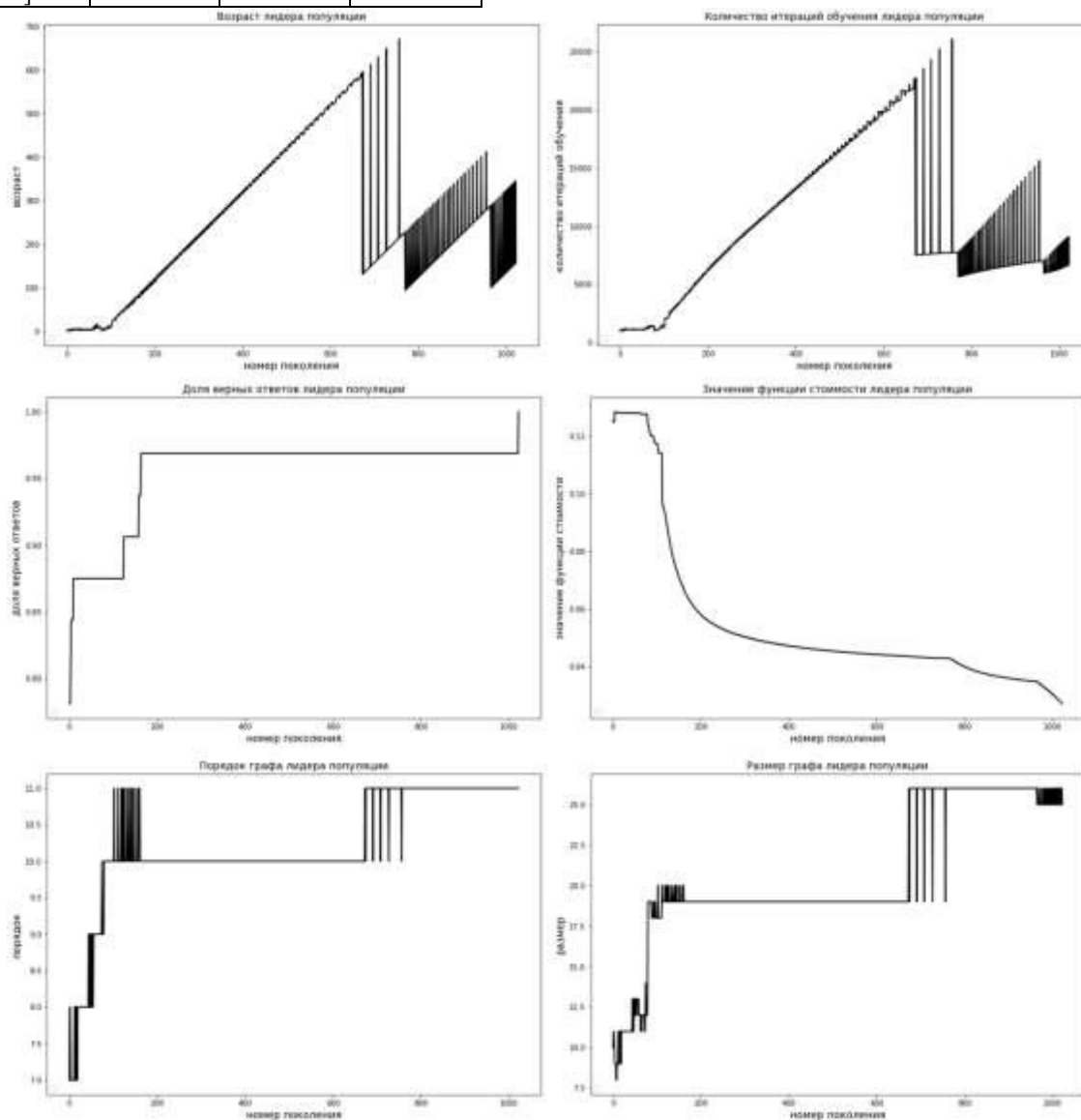


Рис. 4. Графики зависимостей лидера популяции от номера поколения (слева направо сверху вниз): возраст, количество итераций обучения, доля верных ответов, значение функции стоимости, количество нейронов, количество ребер.

Заключение

Было рассмотрено семейство меметических алгоритмов, с помощью которых возможно автоматическое конструирование искусственных нейронных сетей, обладающих высокой скоростью обучения и состоящих из меньшего количества нейронов и связей по сравнению со стандартными многослойными сетями (для сравнения из табл. 4 сеть "лидер" и сеть "[5, 6, 2]").

Механизмы эволюционного отбора явля-

ются довольно сложными объектами для моделирования, они обладают своей спецификой в зависимости от задачи и должны анализироваться отдельно. В частности ключевыми моментами для успеха эволюционного отбора оказались забота о молодых особях, разрушение стагнирующей элиты и наличие супермутаций, позволяющих преодолеть локальные экстремальные ямы.

Предложенный подход может быть использован для дальнейшего изучения меха-

низмов эволюционного отбора и для применения к практическим задачам.

Работа выполнена при поддержке гранта РФФИ № 18-29-03236.

Construction of artificial neural networks using memetic algorithms

A.A. Rybakov, T.T. Sokhibov, S.S. Shumilin

Abstract. The article considers a family of memetic algorithms for constructing artificial neural networks of a special kind. The networks represented by acyclic graphs with a fixed number of input and output neurons are considered as target networks. The memetic algorithm is an evolutionary algorithm for the development of neural networks population with iterative partial learning, as well as with the survival, reproduction and mutation of individuals. As the completion of natural selection, the moment is taken when the leader of the population answers correctly all test data from the target set. The considered algorithm allows to automatically find the configuration of neural networks with fewer neurons and connections, as well as having the best characteristics in terms of learning speed..

Keywords. Artificial neural networks, evolutionary algorithms, genetic algorithms, memetic algorithms, population, mutation..

Литература

1. С. Хайкин. Нейронные сети. Полный курс. // Диалектика/Вильямс, 2019.
2. Deep Learning. An MIT press book. // <https://www.deeplearningbook.org/>, дата обращения 10.02.2020.
3. G.I. Sher. Handbook of neuroevolution through Erlang. // Springer, 2013.
4. Neural networks and deep learning. // <http://neuralnetworksanddeeplearning.com/>, дата обращения 10.02.2020.
5. T. Harju. Lecture notes on graph theory. // <http://users.utu.fi/harju/graphtheory/graphtheory.pdf>, дата обращения 10.02.2020.
6. Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. Генетические алгоритмы. // М.: Физматлит, 2006.
7. Ю.А. Скобцов. Основы эволюционных вычислений. // Донецк: ДонНТУ, 2008.
8. G.I. Sher. DXNN platform: The shedding of biological inefficiencies. // arXiv:1011.6022, 2011.
9. А.А. Рыбаков. Алгоритм создания случайных графов потока управления для анализа глобальных оптимизаций в компиляторе. // Parallel and distributed computing systems PDCS 2013 collection of scientific papers, Kharkiv, Ukraine, 2013, p. 269-275.
10. Р. Докинз. Слепой часовщик. Как эволюция доказывает отсутствие замысла во вселенной. // Династия, 2014.

Подписано в печать 18.5.2020 г.

Формат 60х90/8

Печать цифровая. Печатных листов 8

Тираж 100 экз. Заказ №

Отпечатано в ППП «Типография «Наука»
121099, Москва, Шубинский пер., 6