

ИСПОЛЬЗОВАНИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ В ОБУЧЕНИИ НЕЙРОННЫХ СЕТЕЙ

Мищенко В.А.*, Коробкин А.А.**

**Воронежский государственный педагогический университет, Воронеж, Россия, e-mail: Vadim-liski@mail.ru*

***Воронежский государственный университет, Воронеж, Россия,*

Данная статья описывает вариант построения структуры генетического алгоритма, а также возможность применения таких алгоритмов в задачах обучения нейронных сетей. В статье проведены анализ и выделены преимущества генетических алгоритмов в поиске оптимальных решений по сравнению с классическими методами. Детально описана схема построения каждого шага генетического алгоритма, в том числе рассмотрена работа различных операций мутации, таких как мутация на основе приращения и классический оператор мутации. Кроме того, предложен новый оператор мутации, включающий в себя алгоритмы работы описанных выше операторов мутации. Также приведена схема представления данных искусственной нейронной сети в виде генетического материала для возможности дальнейшей обработки данных с помощью генетических алгоритмов, в том числе приведена схема преобразования данных, хранящихся в матрице, в одну хромосому, что позволяет сократить количество итераций в процессе обучения сети.

Ключевые слова: генетический алгоритм, обучение нейронной сети.

USE OF GENETIC ALGORITHMS IN TRAINING ARTIFICIAL NEURAL NETWORKS.

Mishchenko V.A.*, Korobkin A.A.**

** Voronezh State Pedagogical University, Voronezh, Russia, e-mail: Vadim-liski@mail.ru*

*** Voronezh State University, Voronezh, Russia*

Given article describes a variant of construction of structure of genetic algorithm, and as an opportunity of application of such algorithms in problems of training of neural networks. In article are lead the analysis and advantages of genetic algorithms in search of optimum decisions are allocated in comparison with classical methods. The circuit of construction of each step of genetic algorithm is in details described, including, work of various operations of a mutation, such as, a mutation is considered on the basis of an increment and the classical operator of a mutation. Besides the new operator of the mutation including algorithms of work of operators described above of a mutation is offered. As the circuit of data presentation of an artificial neural network as a genetic material for an opportunity of the further data processing with the help of genetic algorithms is resulted, including, the circuit of transformation of the data stored in a matrix in one chromosome that allows to reduce quantity of iterations during training a network is resulted.

Key words: genetic algorithm, training of a neural network.

Наиболее приемлемым способом настройки весовых коэффициентов искусственных нейронных сетей можно считать генетические алгоритмы. Это связано с тем обстоятельством, что на начальной стадии нет абсолютно никакой информации о направлении движения в плане настройки весов матрицы [4; 5]. В условиях неопределенности эволюционные методы, в том числе и генетические алгоритмы, имеют наиболее высокие шансы для достижения требуемых результатов. Классический генетический алгоритм оперирует двоичной системой счисления, хотя в последнее время зачастую встречаются работы, в которых операторы генетических алгоритмов выполняют операции над множеством действительных чисел. Это позволяет существенно расширить возможности применения описываемых алгоритмов [1].

Рассмотрим математическую модель поставленной задачи. Дан вектор X с размерностью 256, который представляет собой закодированное изображение известного распознаваемого символа. Также есть вектор Y с размерностью 10, который отражает требуемый результат распознавания, указывающий принадлежность символа к

эталонному образцу. Необходимо найти весовую матрицу W , элементами которой являются вещественные числа в отрезке $[0; 1]$, чтобы выполнялось равенство:

$$X * W = Y \quad (1)$$

Необходимо осуществить настройку весовой матрицы W с помощью генетического алгоритма. В рамках решаемой задачи рассматриваемая матрица имеет следующую размерность: количество строк равняется 10, что представляет собой все символы, описывающие арабские цифры; количество столбцов определяется размером знакоместа, отводимого под каждый символ, 256 столбцов.

С точки зрения очевидности и простоты понимания следует каждый столбец матрицы весов рассматривать в качестве хромосомы, что приведет к наличию 256 хромосом, которые в совокупности будут отражать каждую особь. Однако при практической реализации гораздо удобнее использовать более простую структуру, хотя и менее наглядную. Это связано с тем, что довольно тяжело реализовать операции кроссинговера и мутации при наличии нескольких хромосом. Каждая особь в разрабатываемом генетическом алгоритме можно представить в виде только одной хромосомы, что значительно упростит программную реализацию на ЭВМ. Этого можно достичь следующим способом. Хромосома будет начинаться с первого столбца матрицы весов, и каждый следующий столбец будет просто добавляться в конец уже существующей «единой» хромосомы. Данный метод проиллюстрирован на рисунке 1.

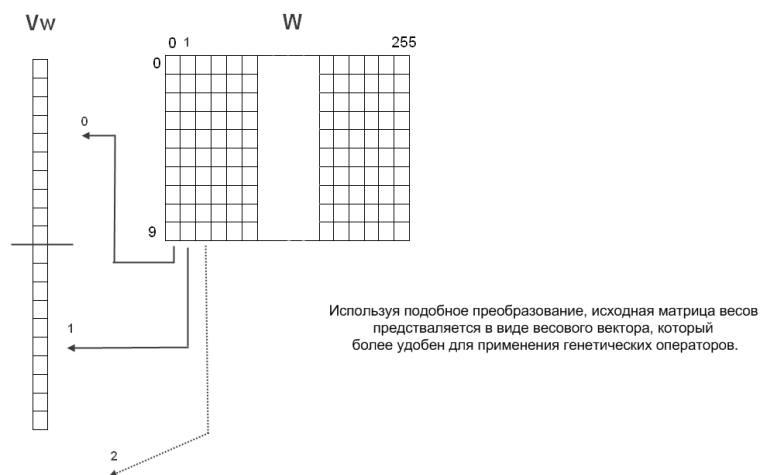


Рис. 1. Преобразование весовой матрицы W в вектор весов Vw .

Классический генетический алгоритм состоит из ряда наиболее важных этапов. Фактически данные этапы можно расположить в хронологическом порядке.

1. ИНИЦИАЛИЗАЦИЯ – формирование исходной популяции.
2. ОЦЕНКА ПРИСПОСОБЛЕННОСТИ – вычисление функции пригодности для каждой особи (в нашем случае хромосомы).
3. СЕЛЕКЦИЯ – выборка на основе оценки приспособленности наиболее приспособленных хромосом, которым будет предоставлено право участия в операциях кроссинговера.
4. КРОССИНГОВЕР – скрещивание двух особей.
5. МУТАЦИЯ – намеренное искусственное изменение определенных генов в хромосомах особи.
6. ФОРМИРОВАНИЕ НОВОЙ ПОПУЛЯЦИИ – снижение количества особей на основе оценки приспособленности вместе с выбором «наилучшей» особи.
7. ПРОВЕРКА КРИТЕРИЯ ОСТАНОВКИ АЛГОРИТМА – если требуемое условие поиска достигнуто – выход, в противном случае – переход к этапу № 3.

8. ИЗВЛЕЧЕНИЕ НАИЛУЧШЕГО РЕШЕНИЯ – наилучшим решением считается особь с максимальным значением функции пригодности.

Все описанные этапы являются абсолютно справедливыми в рамках решаемой задачи [1–3]. Практическая реализация первого шага любого генетического алгоритма в большинстве случаев представляет собой не что иное, как инициализацию случайным образом. Каждому гену любой хромосомы присваивается случайное значение из интервала допустимых значений. Соответственно, при настройке весовой матрицы W каждый ген получит генетическую информацию в виде случайного значения в отрезке $[0; 1]$. Программная реализация подобного метода инициализации является самой простой и имеет свои достоинства и недостатки. Достоинства заключаются в следующем.

- Не требуется применения дополнительных алгоритмов.
- Инициализация выполняется быстро и не загружает ЭВМ.
- Уменьшены шансы попадания в локальный оптимум.

В качестве недостатка можно отметить отсутствие в разрабатываемом алгоритме накопленных знаний о настраиваемых коэффициентах. Таким образом, в результате этапа инициализации имеется готовая популяция решений. Несмотря на абсурдность, можно говорить о наличии решения уже на первом этапе. После получения исходной популяции производится переход ко второму этапу – оценке приспособленности.

Оценка приспособленности особей в популяции заключается в вычислении значения функции пригодности для каждого члена популяции. И чем выше это значение, тем больше особь отвечает требованиям решаемой задачи. В рамках решаемой задачи весовая матрица есть часть нейронной сети, которая осуществляет распознавание образов. И соответственно генетический алгоритм выполняется на стадии обучения нейронной сети. Иными словами, необходимо привести весовую матрицу к такому виду, при котором ошибка распознавания эталонного образа будет минимальной. Таким образом, функция пригодности оценивает ошибку распознавания каждого эталонного образа, и чем меньше ошибка, тем выше значение функции пригодности. Задача состоит в минимизации ошибки распознавания. Для этого необходимо сравнить полученный вектор Y' с эталонным образцом Y .

Этап селекции предусматривает выбор тех особей, генетический материал которых будет участвовать в формировании следующей популяции решений, т.е. в создании очередного поколения. Описываемый выбор производится согласно принципу естественного отбора, благодаря которому максимальные шансы имеют особи с наибольшими значениями функции пригодности. Различают достаточно большое множество методов селекции. Одним из самых известных и часто применяемых является метод «рулетки». Такое название интуитивно помогает понять его принципы. Каждая особь получает определенный сектор на «колесе», размер которого напрямую зависит от значения функции пригодности.

Отсюда вытекает, что чем выше значение функции пригодности, тем больше будет размер сектора на «колесе рулетки». Очевидно, что чем больше сектор, тем выше вероятность «победы» соответствующей особи. И, как следствие, вероятность выбора определенной особи оказывается пропорциональной значению ее функции приспособленности. Использование метода «рулетки» часто приводит к преждевременной сходимости алгоритма, которая заключается в том, что в популяции начинают доминировать лучшие особи, но не оптимальные. Спустя несколько поколений популяция практически полностью будет состоять из копий наиболее лучших особей. Однако весьма маловероятно, что достигнутое решение будет оптимальным, так как исходная популяция генерируется случайным образом и представляет собой лишь малую часть пространства поиска. Чтобы предотвратить преждевременную сходимость генетического алгоритма, используется масштабирование функции приспособленности. Масштабирование функции пригодности позволяет исключить ситуацию, когда средние и лучшие особи начинают

формировать одинаковое число схожих потомков в следующих поколениях, что является крайне нежелательным явлением. Следует отметить, что масштабирование также предупреждает случаи, когда, несмотря на значительную неоднородность популяции, среднее значение функции приспособленности мало отличается от максимального. Итак, масштабирование функции приспособленности есть не что иное, как преобразование ее вида. Выделяют три основных преобразования: линейное, степенное и сигма-отсечение. В разработанном алгоритме используется преобразование сигма-отсечение.

$$F_{\delta} = F + (\bar{F} - a\delta), \quad (2)$$

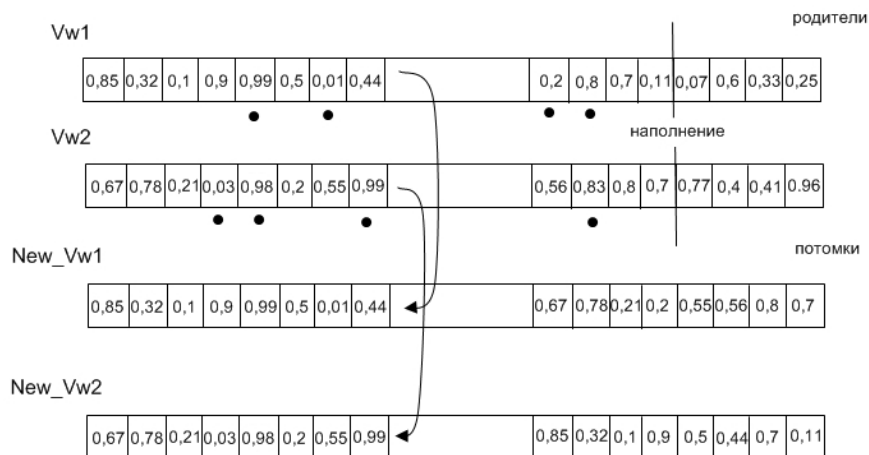
где a – малое число, как правило, от 1 до 5; \bar{F} – среднее значение функции приспособленности по популяции; δ – стандартное отклонение по популяции. В случае, когда полученные значения преобразованной функции будут отрицательны, их приравнивают к 0.

Также можно в определенной степени модифицировать данный метод селекции или строить селекцию на основе синтеза сразу нескольких методов.

Следующим этапом генетического алгоритма является рекомбинация или кроссинговер. Каждый участок хромосомы особи включает в себе определенную информационную нагрузку. Целью рекомбинации является получение такой комбинации промежутков хромосом, при которой особь будет представлять собой наилучшее из решений, возможное при текущем генетическом материале. В итоге основной задачей операции кроссинговера является получение в конечном итоге наиболее функциональных признаков, которые присутствовали в наборах исходных решений. Подобный механизм решения оптимизационных задач, в отличие от существующих методов, не осуществляет замену одного решения на другое, а получает новые возможные решения посредством обмена информацией между ними.

Скрещивание является наиболее важным оператором генетического алгоритма, так как именно с помощью оператора кроссинговера осуществляется обмен информацией между решениями. Потомки содержат в себе комбинацию специфических особенностей обоих родителей. Эффективность работы любого генетического алгоритма находится в прямой пропорциональной зависимости от эффективности операции кроссинговера. Кроме того, производительность генетического алгоритма зависит от успешности работы кроссинговера в первую очередь. В рамках решаемой задачи реализован упорядоченный оператор кроссинговера. Упорядоченный кроссинговер осуществляет поэтапное рутинное преобразование генетического материала, приближаясь к оптимальному решению.

На рис. 2 изображен процесс получения новых особей с использованием упорядоченного кроссинговера. Имеются две родительские хромосомы: $Vw1$ и $Vw2$. Генетическим материалом являются вещественные числа от 0 до 1. Упорядоченный кроссинговер работает следующим образом. Изначально случайным образом определяется «разрезающая точка». На следующем этапе первый потомок New_Vw1 наследует левую часть родительской хромосомы $Vw1$. Заполнение оставшихся генов новой хромосомы осуществляется за счет информации, хранящейся у второго родителя $Vw2$. Алгоритм просматривает хромосому $Vw2$ с самого начала и осуществляет извлечение генов, которые отличаются от генов, уже находящихся в потомке, более чем на величину $\epsilon = 0,02$. Малая величина ϵ задается в алгоритме для определения «родства» генов. С каждым последующим шагом, а в особенности на завершающих этапах алгоритма, имеет смысл уменьшать значение этой величины для достижения более точных результатов. Аналогичная процедура выполняется при получении второго потомка New_Vw2 . Второй потомок New_Vw2 наследует левую часть родительской хромосомы $Vw2$. Заполнение оставшихся генов получаемой хромосомы осуществляется за счет информации, находящейся у второго родителя $Vw1$.



Гены, отмеченные маркером не вошли в хромосомы-потомков New_Vw1 и New_Vw2, так как левая часть этих хромосом уже содержит гены, которые отличаются от них на величину $e \leq 0,02$. Также в каждой из родительских хромосом осталась часть нетронутых элементов, так как наполнение потомков уже завершилось.

Рис. 2. Принцип работы упорядоченного кроссинговера.

Алгоритм производит анализ хромосомы Vw1 с первого гена и осуществляет упорядоченное извлечение генов, которые отличаются от генов, уже находящихся в потомке, более чем на величину $e = 0,02$. В результате работы каждого оператора скрещивания в популяции появляются две новые особи. Чтобы контролировать количество операций кроссовера используется коэффициент скрещивания K_k , определяющий долю производимых на каждой итерации потомков. Количество потомков определяется следующей формулой:

$$Count_p = round(Size_p * K_k) * 2, \quad (3)$$

где $Size_p$ – размер популяции, $Count_p$ – количество получаемых потомков, $round$ – операция округления.

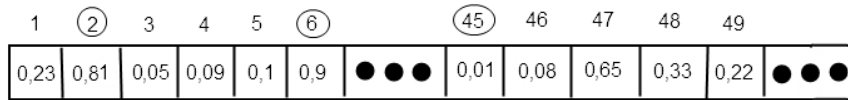
Высокое значение коэффициента скрещивания K_k позволяет увеличить количество областей пространства поиска и уменьшает опасность попадания в локальный оптимум, однако слишком большое значение указанного параметра приведет к увеличению времени работы алгоритма, а также к чрезмерному исследованию малоперспективных областей поискового пространства.

Следующим этапом генетического алгоритма является мутация. Мутация есть изменение, которое приводит к проявлению качественно новых свойств генетического материала. Мутации происходят случайным образом и вызывают скачкообразные изменения в структуре генотипа.

В рамках решения оптимизационных задач наибольшее значение имеют генные мутации, которые в большинстве случаев затрагивают один или несколько генов. Мутация может выглядеть любым образом, будь то обмен генов своими позициями или копирование значения другого гена и т.д. В каждом генетическом алгоритме необходимо определиться с выбором вида мутации. В рассматриваемом генетическом алгоритме гены содержат в себе вещественные числа от 0 до 1. Согласно этому оператор мутации должен вносить конкретные изменения в генетический материал, т.е. изменять значения определенных генов, не опираясь на уже существующие гены. Суть разработанного оператора мутации состоит в следующем. В исследуемой хромосоме случайным образом выделяют также случайное количество генов. Коэффициент мутации K_m определяет интенсивность мутаций. Он определяет долю генов, подвергнутых мутации на текущей итерации, в расчете на их общее количество. Если коэффициент мутации слишком мал, то получится ситуация, при которой множество полезных генов просто не будут

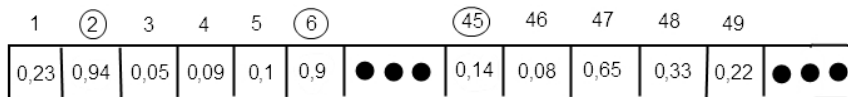
существовать в популяции. В то же время использование большого значения коэффициента мутации приведет к множеству случайных возмущений и значительно увеличит время поиска. Потомки перестанут походить на родителей, алгоритм больше не будет иметь возможности обучаться на основе сохранения наследственных признаков. К выбранным генам применяют преобразование, которое вызывает изменение значения текущего гена на некоторую небольшую величину δ . Величина δ выбирается с таким учетом, чтобы после изменения значения i -го гена он находился в отрезке $[0; 1]$.

Хромосома до мутации



Пусть, случайным образом выбрана интенсивность мутации, указывающая количество мутирующих генов равное 3. Также, случайным образом выбраны номера этих генов: 2,6,45.
 Приращение сигма получила случайное значение: 0,13.
 Тогда хромосома после мутации будет выглядеть следующим образом.

Хромосома после мутации



Ген №6 не изменил своего значения, так как его величина стала равна 1,12, что противоречит постановке задачи.
 В таком случае говорят о "ненужной мутации".

Рис. 3. Случайная мутация на основе приращения.

На рис. 3 показано, как выполняется мутация, используя приращение δ . Гены с номерами 2 и 45 успешно получили новое значение, что приведет к изменению показаний функции пригодности мутирующей особи. В то время как мутация гена номер 6 оказалась неприемлемой и соответственно была проигнорирована. Использование мутаций на основе δ -приращения позволяет вносить в популяцию новый генетический материал. Это приведет к увеличению пространства поиска, что необходимо для эффективного поиска оптимума. Разумеется, имеет смысл воспользоваться классическим оператором мутации, в основе которого лежит случайное изменение порядка следования генов. Это также приводит к получению достаточно хороших результатов уже на ранних этапах работы генетического алгоритма. При решении рассматриваемой задачи также используется многоточечный оператор мутации. Алгоритм случайным образом в соответствии с коэффициентом интенсивности мутации выбирает несколько генов, значения которых далее меняются со значениями соседних генов. Работа классического многоточечного оператора мутации изображена на рис. 4.

Использование в одном алгоритме сразу нескольких видов оператора мутации позволяет осуществить эффективный поиск оптимального решения. Что позволяет в течение короткого промежутка времени получить неплохие результаты, а также выделить наиболее пригодные для изучения «окрестности решений».

Оператор мутации применяется к потомкам особей, получаемых после операции скрещивания. Особи с мутациями остаются в популяции до начала этапа «формирование новой популяции». Количество мутирующих особей определяется формулой:

$$Count_m = round(Count_p * K_m^k) + round(Count_p * K_m^\delta), \quad (4)$$

$$K_m^k + K_m^\delta = K_m \leq 1,$$

где K_m^k – коэффициент классической мутации; K_m^δ – коэффициент мутации на основе δ -смещения; K_m – общий коэффициент мутации; $Count_m$ – количество особей, претерпевших мутацию; $Count_p$ – количество потомков; $round$ – операция округления.

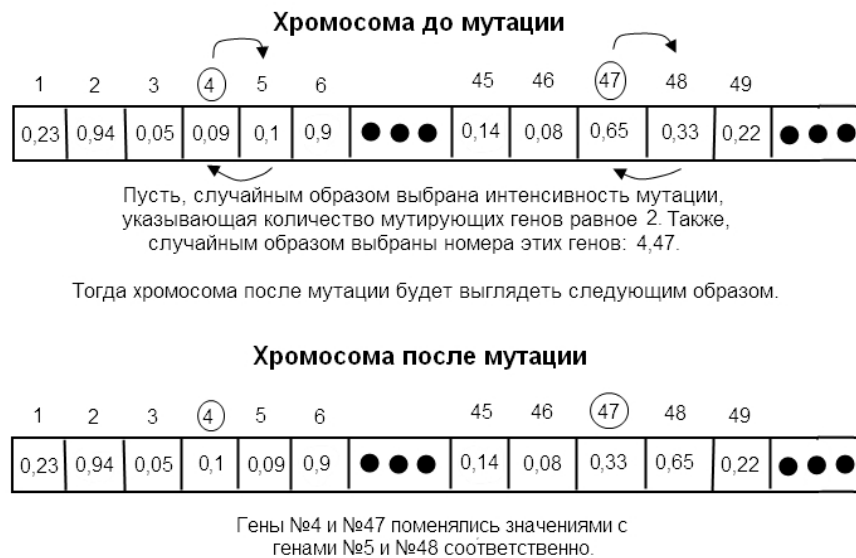


Рис. 4. Классический многоточечный оператор мутации.

Из формулы видно, что алгоритм позволяет в процентном соотношении регулировать баланс между классическим многоточечным оператором мутации и мутацией, основанной на δ -приращении. С помощью данной функции можно подстраивать алгоритм под условия конкретной задачи. При увеличении доли использования классического оператора мутации увеличивается тщательность поиска в окрестности лучших решений на основе текущего генетического материала, а с увеличением доли мутаций, основанных на δ -приращении соответственно растет общее пространство поиска и происходит обновление генного состава популяции.

Следующим этапом разработанного генетического алгоритма является формирование новой популяции. Насколько сильно выросла численность популяции, определяется общим коэффициентом мутации K_m и коэффициентом скрещивания K_k . В общем виде текущая численность популяции вычисляется по следующей формуле:

$$Size_p = Size_p + Count_p + Count_m, \quad (5)$$

где $Count_p$ – количество полученного потомства, $Count_m$ – количество особей, полученных в результате мутаций, а $Size_p$ – численность популяции.

Этап формирования новой популяции предназначен для того, чтобы вернуть численность популяции к ее исходному значению. Значение функций пригодности особей – родителей, которые уже находились в популяции, известно. Алгоритм осуществляет оценку пригодности особей – потомков, полученных в результате операции кроссинговера, а также оценку особей, которые были получены в результате работы оператора мутации. На основе полученных значений функции пригодности для каждой особи новой популяции производится удаление тех особей, у которых значения функции пригодности наименьшие. Алгоритм реализует данную задачу путем последовательного удаления особи с минимальным значением функции пригодности до тех пор, пока численность популяции не вернется к начальному значению. Количество «отмерших» особей вычисляется по формуле:

$$Count_d = Count_p + Count_m, \quad (6)$$

где $Count_d$ – количество «отмерших» особей, $Count_p$ – количество полученных потомков, $Count_m$ – количество особей, полученных с помощью оператора мутации.

На этапе формирования новой популяции также выполняется поиск наилучшего решения – той особи, у которой максимальное значение функции пригодности. Эта операция выполняется после того, как численность популяции установилась в исходную величину. После выбора наилучшей особи алгоритм передает управление процессом следующему этапу – проверке останковки алгоритма.

Определение критерия останковки генетического алгоритма напрямую зависит от специфики решаемой задачи и имеющихся сведений об объекте поиска. В большинстве оптимизационных задач, для которых известно оптимальное значение функции пригодности, останковку алгоритма можно организовать при достижении наилучшей особью этой величины, возможно с некоторой погрешностью. Решаемая задача на самом деле не располагает сведениями об оптимальном значении функции пригодности. Иными словами, алгоритм стремится максимизировать функцию пригодности с учетом того, что ошибка распознавания стремится к нулю. Поэтому в созданном алгоритме реализован механизм останковки поиска, основанный на отсутствии изменения функции пригодности наилучшей особи в течение определенного числа итераций, которое задается в качестве параметра генетического алгоритма. Кроме этого, алгоритм предусматривает останковку своей работы по истечении определенного числа итераций, которое также задается в качестве параметра. В том случае, если условие останковки выполнено, то алгоритм выдает в качестве оптимального решения то, которое представлено наилучшей особью, определенной на этапе формирования новой популяции. Если же условие не выполнено, то алгоритм передает управление этапу селекции.

Таким образом, в результате работы генетического алгоритма получен набор весовых коэффициентов, обеспечивающих корректную работу нейронной сети. Также стоит отметить тот факт, что использование специального оператора мутации позволило сократить время обучения сети. Для обучения было выбрано обучающее множество из 180 элементов. Время обучения на этом множестве с помощью описанного генетического алгоритма с использованием мутации на основе приращения, нечеткой сети Ванга-Менделя составило 2 мин. 50 сек., обучение с использованием алгоритма с применением классического многоточечного оператора мутации длилось 3 мин. 10 сек., а применение гибридного оператора мутации позволило сократить время обучения до 1 мин. 20 сек. Таким образом, полученный генетический алгоритм позволяет сократить время поиска наилучшего решения в рамках поставленной задачи.

Список литературы

1. Аксенов С.В., Новосельцев В.Б. Организация и использование нейронных сетей (методы и технологии) / под общ. ред. В.Б. Новосельцева. – Томск : Изд-во НТЛ, 2006. – 128 с.
2. Батыршин И.З. Нечеткие гибридные системы. Теория и практика / под ред. Н.Г. Ярушкиной. – М. : ФИЗМАТЛИТ, 2007. – 208 с.
3. Гладков Л.А. Генетические алгоритмы / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. – М. : ФИЗМАТЛИТ, 2006. – 320 с.
4. Осовский С. Нейронные сети для обработки информации / пер. с польского И.Д. Рудинского. – М. : Финансы и статистика, 2002. – 344 с.
5. Хайкин С. Нейронные сети: полный курс / пер. с англ. – 2-е изд. – М. : ИД «Вильямс», 2006. – 1104 с.

Рецензенты:

Кургалин С.Д., д.ф.-м.н., зав. кафедрой цифровых технологий ФГБОУ ВПО «Воронежский государственный университет», г. Воронеж.

Шашкин А.И., д.ф.-м.н., зав. кафедрой математического и прикладного анализа ФГБОУ ВПО «Воронежский государственный университет», г. Воронеж.

Сентябрев Н.Н. д.б.н., профессор кафедры физиологии ФГОУ ВПО «Волгоградская государственная академия физической культуры», г. Волгоград.