

Отметим, что основной недостаток существующих методов разбиения – это получение за один прогон алгоритма одного локального оптимума.

С увеличением числа генераций в ГА время решения, конечно, повышается, но это повышение незначительное и компенсируется получением множества локально-оптимальных решений. Разбиение графов с применением ГА позволяет всегда получать локальные оптимумы, иметь возможность выхода из них и приближаться к получению оптимальных и квазиоптимальных решений.

ЛИТЕРАТУРА

1. Курейчик В.М. Математическое обеспечение конструкторского и технологического проектирования с применением САПР. М.: Радио и связь. 1990.
2. Охматов Н.А. Зарождение жизни на планете Земля. М.: Мир, 1994.
3. Айала Ф. Введение в популярную и эволюционную генетику. М.: Мир, 1984.

УДК 658.512

В.И. Божич, О.Б. Лебедев, Ю.Л. Шницер¹

РАЗРАБОТКА ГЕНЕТИЧЕСКОГО АЛГОРИТМА ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

Одной из областей применения генетических алгоритмов является обучение нейронной сети. В данной работе рассматривается генетический алгоритм синтеза архитектуры многослойной сети прямого распространения, определяемой числом слоев и числом нейронов в слое [1]. Процесс обучения нейронной сети с помощью генетического алгоритма разбивается на два этапа:

на первом этапе с помощью генетического алгоритма осуществляется поиск общих параметров нейронной сети, а именно: количества скрытых слоев и нейронов в каждом слое;

на втором этапе генетический алгоритм используется для поиска значений связей между нейронами и функций активации.

Целевая функция задается для множества примеров и представляет собой максимальное относительное отклонение от эталонного значения, выраженное в процентах. Цель применения эволюционной парадигмы – минимизация функции

$$C = \max_{j=1}^k \frac{(y'_{ij} - y_{ij})}{y_{ij}} \cdot 100\%,$$

где k – количество примеров; y'_{ij} – значение i -го выхода нейронной сети для j – того примера.

В данной работе предлагается два способа кодирования хромосом. В первом случае разработка структуры хромосомы производится так, чтобы гены в одних и тех же локусах хромосом являлись гомологичными, так как это упрощает выполнение генетических операторов кроссинговера и мутации.

¹ Работа выполнена при поддержке РФФИ, грант №00-01-00125

Число генов в хромосоме постоянно и равно максимальному числу слоев нейросети. Ген g_1 соответствует входному слою, а ген g_n – выходному слою. Гены g_2, \dots, g_{n-1} соответствуют скрытым слоям. Структура связей между двумя соседними слоями соответствует полному двудольному графу. Значением гена g_i является число нейронов в i -том слое. Если $g_i = 0$, то этот слой исключается из рассмотрения.

При втором способе кодирования длина хромосомы (число генов) равна фактическому числу слоев нейронной сети плюс один. При этом в хромосоме отсутствуют гены с нулевыми значениями. Значением первого гена g_1 является число слоев, значениями генов g_2, \dots, g_n является число нейронов в слое, причем g_2 соответствует входному слою, g_n – выходному, а g_3, \dots, g_{n-1} – скрытым слоям нейронной сети.

Пространственная сложность и трудоемкость декодирования хромосом имеет оценку $O(n)$, где для первого способа кодирования параметр n – это максимально возможное число слоев нейронной сети, а для второго способа – параметр $(n - 1)$ определяет число слоев нейронной сети.

Основными генетическими операторами являются операторы кроссинговера и мутации [2]. Для *хромосом первого вида* оператор кроссинговера выполняется следующим образом. Пусть имеются две родительские хромосомы H_1 и H_2 . Отметим, что гены g_1 и g_2 в обеих хромосомах имеют одинаковое значение. Последовательно просматриваются локусы хромосом, начиная со второго, и с вероятностью P_k осуществляется обмен генами в текущем локусе. Обмен завершается после просмотра всех локусов в интервале $2 \dots (n - 1)$. В результате получаются две хромосомы H'_1 и H'_2 .

Особенности применения оператора кроссинговера для *хромосом второго типа* связаны с тем фактом, что родительские хромосомы могут иметь различную длину. Если родительские пары имеют одинаковую длину, то действие оператора кроссинговера аналогично рассмотренному выше, с той лишь разницей, что локусы просматриваются в интервале $\{3 \dots (n - 1)\}$. Для пары хромосом различной длины процедура кроссинговера выполняется иначе. Пусть имеется пара хромосом H_1 и H_2 и пусть $g_1 > g_2$. Первоначально хромосома меньшей длины H_2 модифицируется путем добавления числа $(n_1 - n_2)$ генов с нулевым значением. Новые гены могут вставляться между генами $(g_2 - g_3), (g_3 - g_4), \dots, (g_{n-2} - g_{n-1})$. Места вставки выбираются случайным образом. После этого осуществляется процедура кроссинговера между хромосомами H_1 и модифицированной хромосомой H_{2M} . В результате образуется новая пара хромосом H'_{1M} и H'_{2M} . Затем из хромосом H'_{1M} и H'_{2M} удаляются гены с нулевым значением и обновляются значения генов g_1 .

Процедура мутации осуществляется следующим образом: последовательно просматриваются локусы, начиная со второго локуса, для хромосом первого класса и, начиная с третьего локуса, для хромосом второго класса, и, с вероятностью P_m ген $g_i \in H_i$ приобретает новое значение в текущем диапазоне $0 \leq g_i \leq m$, где m – максимально допустимое число нейронов в слое.

Из приведенных пояснений применения операторов кроссинговера и мутации следует, что временная сложность операторов кроссинговера t_k и мутации t_m применительно к одной хромосоме имеет линейную зависимость. Другими словами,

оценки временной сложности имеют вид: $t_k = O(n)$; $t_m = O(n)$, где n – длина хромосомы.

Для организации генетического поиска формируется исходная популяция особей: $P_H = \{H_k \mid k = 1, 2, \dots, M\}$, где M – размер популяции. Популяция P_H представляет собой репродукционную группу индивидуальностей, любые из которых могут размножаться, выступая в роли родителей. В алгоритме используется принцип случайного формирования исходной популяции. Формирование популяции хромосом первого типа осуществляется следующим образом. Генам g_1 и g_2 присваиваются значения $g_1 = input$, $g_2 = output$, где $input$ и $output$ – задаваемые в качестве исходных данных числа нейронов входного и выходного слоев нейронной сети.

Гены g_2, \dots, g_{n-1} принимают случайные значения на интервале $(0 - m)$, где m – задаваемое исходными данными максимально допустимое число нейронов в сети. Если используются хромосомы второго типа, то популяция получается из хромосом первого типа путем модификации. Для этого из хромосомы удаляются гены с нулевым значением, а в начало хромосомы помещается ген со значением, равным числу слоев в нейронной сети, кодируемой данной хромосомой. Объем ОЗУ, необходимый для хранения популяции, имеет оценку пространственной сложности, равную $O(n, m)$, где n – длина хромосомы, m – размер популяции.

При синтезе архитектуры нейронной сети решается оптимизационная задача на двух уровнях:

на первом уровне решается оптимизационная задача, связанная с обучением нейронных сетей;

на втором уровне решается задача поиска архитектуры такой нейронной сети, которая после обучения имела бы лучшие показатели качества.

Для решения указанных двух задач используется один и тот же показатель качества, причем: на первом уровне преследуется цель достижения минимального значения показателя C , на втором уровне при поиске нейронной сети – минимального значения показателя качества C . Для реализации указанных процедур формируются две выборки примеров: V_O – обучающая выборка и V_T – тестирующая выборка. На выборке V_O производится обучение нейронной сети с оптимизацией показателя C_O , при этом осуществляется поиск значений весов для всех скрытых и выходных нейронов и параметров функций активации. На выборке V_T (после обучения нейронной сети) осуществляется расчет показателя C_T , который и является оценкой качества обучения сети.

Таким образом, условно при поиске архитектуры нейронной сети процесс декодировки можно рассматривать как переход от хромосомы H_i к нейронной сети, потом производится ее обучение на выборке V_O с помощью некоторого алгоритма и определяется оценка его качества с помощью выборки V_T . Другими словами, множеством фенотипов популяции является множество обученных нейронных сетей с оценками качества C_T .

При таком подходе возникает задача выбора качественного и количественного состава образов из числа имеющихся с тем, чтобы обеспечить высокий процент верных классификаций после обучения нейронной сети, а также сократить машинные затраты на построение системы, исключив из выборки малозначимые образы, с точки зрения процедуры обучения нейронных сетей. Имеются различные подхо-

ды к решению данной задачи. Предлагаемый в данной работе подход ориентирован на случай, когда число образов в выборке достаточно велико, чтобы позволить алгоритму обучения нейронной сети эффективно «выбраться» из локальных минимумов функции ошибки, манипулируя малой частью выборки. Критерием отбора элементов в обучающую выборку служит отношение вида

$$V_{\text{верно}}(V_O, V_T) / |V_O|,$$

где $V_{\text{верно}}$ – число верно классифицированных образов в исходной выборке, $|V_O|$ – мощность (число образов) выборки. Данный критерий определяет число верно классифицируемых образов в тестирующей выборке. Минимально допустимой обучающей выборкой является число образов, для которого достигается максимум установленного критерия. При поиске решения с помощью генетического алгоритма используется глобальная информация о значении критерия, в то время как процедура поиска случайным образом зависит от задаваемых параметров алгоритма. Процедура поиска повторяется до выполнения некоторого заданного условия, например, когда значение критерия в среднем для отобранного поколения составит некоторый высокий процент от максимального значения.

В данной работе предлагается способ увеличения скорости обучения нейронных сетей, основанный на процедуре «выбивания» сети из локального минимума и процедуре «расщепления» нейронов в скрытых слоях сети. Как известно, существенным недостатком известных алгоритмов расщепления является экспоненциальный рост времени вычислений при увеличении размерности сети. Для преодоления указанного недостатка в данной работе в качестве критерия выбора нейрона для «расщепления» используется отношение суммы длин векторов изменений синаптических весов нейрона, соответствующих различным обучающим примерам, к длине суммы этих векторов, т.е. выбирается нейрон с наибольшим значением функционала

$$F = \frac{\sum_{e=1}^P |\delta w_i^e|}{\left| \sum_{e=1}^P \delta w_i^e \right|},$$

где δw – вектор изменений синаптических весов нейрона; i – номер нейрона ($i = 1, 2, \dots, N$); e – номер обучающего примера; P – число примеров в обучающей выборке; $|\cdot|$ – длина вектора.

В результате расщепления вместо исходного нейрона в сеть вводятся два новых нейрона. Значение каждого синаптического веса нового нейрона есть значение соответствующего веса старого нейрона плюс некоторый, очень небольшой, шум. Величины весов связей выходов новых нейронов и нейронов следующего слоя равны половине весов связей исходного нейрона с соответствующими весами следующего слоя.

Предлагаемый способ применим для задач, когда параметр времени обучения нейронной сети является существенным фактором, и гарантирует, что функция ошибки после «расщепления» нейронов увеличиваться не будет. Однако при этом общее число нейронов в сети, построенной с помощью генетического алгоритма по заданной обучающей выборке, может быть несколько больше, чем у сети, построенной с помощью известных алгоритмов.

ЛИТЕРАТУРА

1. Уоссермен Ф. Нейрокомпьютерная техника. М.: Мир, 1992
2. Курейчик В.М. Генетические алгоритмы: Монография. Таганрог: Изд-во ТРТУ, 1998. 242с.

УДК 681.518:339.13

О.В. Коновалов¹

КООПЕРАТИВНО-СОРЕВНОВАТЕЛЬНЫЙ АЛГОРИТМ ОБУЧЕНИЯ СЕТЕЙ РАДИАЛЬНОГО БАЗИСА

1. Сети радиального базиса. Сети радиального базиса, в отличие от обычных нейронных сетей, всегда имеют только один скрытый слой элементов, причем производимые вычислительные операции для скрытого и выходного слоев различаются. Первый (входной) и последний слой сети выполняют те же функции. Каждый элемент скрытого слоя характеризуется центром c_i – точкой во входном пространстве и отклонением d_i . При подаче на вход сети образа каждый элемент определяет евклидово расстояние от собственного центра до входного образа, соотносит это расстояние с отклонением и берет от полученной величины нелинейную функцию /1, 2/. Общее преобразование F_j , выполняемое сетью радиального базиса по выходу j , имеет следующий вид:

$$F(\vec{x})_j = \omega_{j0} + \sum_{i=1}^L v_{ij} \phi_i(\vec{x}), \quad (1)$$

где ω_{j0} – значение порога по j -му выходу.

L радиально-симметричных базисных функций осуществляют нелинейное преобразование, предварительно соотнеся входы с собственными значениями центра и отклонения.

$\phi_i(\mathbf{x}) = \phi(\|\mathbf{x} - \mathbf{c}_i\|/d_i)$, где $\mathbf{c}_i \in \mathcal{R}^n$ есть центр базисной функции ϕ , а d_i является отклонением или масштабирующим множителем для радиуса $\|\mathbf{x} - \mathbf{c}_i\|$, и $\|\cdot\|$ – обычно евклидова норма в \mathcal{R}^n . Теоретически и практически исследованы множество различных функций ϕ , однако наиболее популярны среди них следующие /1/:

– $\phi(r) = r^2 \log r$ – тонкопленочный сплайн (thin plate spline);

– $\phi(r) = e^{-r^2/2}$ – гауссоида;

– $\phi(r) = \sqrt{r^2 + 1}$ – мультиквадратичная функция;

– $\phi(r) = \frac{1}{\sqrt{r^2 + 1}}$ – инверсная мультиквадратичная функция,

где во всех случаях r – масштабированный радиус $\|\mathbf{x} - \mathbf{c}_i\|/d_i$.

Первоначально сети радиального базиса были предложены как интерполяционный метод и их свойства в решении соответствующих задач были достаточно

¹ Работа выполнена при поддержке РФФИ, грант №00-01-00125