

УДК 004.032.24

С.И. Родзин

Таганрогский государственный радиотехнический университет, Россия

Параллелизм нейроэволюционных вычислений

Рассмотрен подход к реализации гибридной интеллектуальной системы, основанной на комбинировании нейросетей и эволюционных алгоритмов. Предлагаемый нейроэволюционный метод оптимизации процесса построения архитектуры сети ориентирован на параллельные вычислительные системы. Представлены три модели параллелизма: глобальный параллелизм, миграционная модель, диффузионная модель. Для оптимизации архитектуры и параметров нейросети используется комбинация прямого и непрямого способов кодирования архитектур многослойного перцептрона.

Введение

Нейросети доказали свою эффективность как при решении трудноформализуемых задач искусственного интеллекта (распознавание образов, кластеризация, ассоциативный поиск), так и при решении трудоемких задач оптимизации и аппроксимации [1]. К преимуществам нейросетей относятся способность к обучению, параллелизм и отказоустойчивость. Эволюционные алгоритмы (ЭА), включающие генетические алгоритмы, генетическое программирование, эволюционные стратегии и эволюционное программирование, также представляют собой перспективное и динамично развивающееся направление интеллектуальной обработки данных, связанное с решением задач поиска и оптимизации. Общепризнанной является возможность построения гибридных интеллектуальных систем на основе комбинирования парадигм нейросетей и ЭА, поскольку, по мнению многих исследователей, эти парадигмы являются аналогами процессов, происходящих в живой природе. Даже если это не так, и в природе происходит реэволюция, никто не может сказать, что эти алгоритмы неверны.

Проблемы оптимизации архитектуры нейросетей для решения конкретной задачи, а также значительные вычислительные затраты, опыт и интуиция исследователя, требуемые для обучения нейросети, остаются факторами, сдерживающими их массовое применение в качестве новых компьютерных технологий. Теория эволюции расширяет сферу применения классических методов искусственного интеллекта, основанных на комбинаторно-логических алгоритмах обработки символической информации. Данная работа посвящена анализу возможных путей решения указанных проблем на основе параллелизма и гибридизации эволюционных алгоритмов и нейросетей.

Параллелизм эволюционных алгоритмов

К известным достоинствам ЭА традиционно относятся: широкая область их применения; возможность проблемно-ориентированного кодирования решений; пригодность для поиска в сложном пространстве решений большой размерности; отсутствие ограничений на вид целевой функции; ясность схемы и базовых принципов.

В то же время эвристический характер ЭА не гарантирует оптимальности полученного решения (правда, на практике зачастую важно за заданное время получить одно или несколько субоптимальных альтернативных решений, тем более что исходные данные в задаче могут динамически меняться, быть неточными или неполными). ЭА имеют относительно высокую вычислительную трудоемкость, причиной которой является то обстоятельство, что в ходе моделирования эволюции многие решения отбрасываются как неперспективные (хотя многочисленные экспериментальные данные указывают на то, что временная сложность ЭА в среднем ниже, чем у лучших конкурирующих алгоритмов, но не более чем на один порядок). Невысокая эффективность ЭА на заключительных фазах моделирования эволюции (в особенности это касается генетических алгоритмов) объясняется тем, что операторы поиска в ЭА не ориентированы на быстрое попадание в локальный оптимум. Нерешенными остаются вопросы самоадаптации ЭА.

Преодоление перечисленных выше трудностей тесно связано с понятием параллелизма ЭА, который позволяет реалистично моделировать эволюцию с помощью параллельных вычислительных систем и использовать для эволюции популяции большой размерности.

Различают следующие типы параллелизма ЭА [2]:

- 1) глобальный параллелизм;
- 2) миграционная модель;
- 3) диффузионная модель.

В первом случае речь идет о параллельном вычислении целевой функции и параллельном выполнении операторов ЭА. Целевая функция отдельного индивидуума в популяции, как правило, не зависит от других индивидуумов, поэтому вычисление целевых функций допускает распараллеливание. При этом популяция хранится в общей памяти, а отдельный процессор считывает «хромосому» из памяти и возвращает результат после вычисления целевой функции в виде *fitness*-значения. Процедура синхронизации здесь проводится лишь при формировании новой популяции. В случае если вычислительная система имеет распределенную память, то популяция и *fitness*-значения отдельных индивидуумов хранятся в «мастер»-процессоре, а расчеты производятся на вспомогательных процессорах. Некоторую трудность для распараллеливания представляет процедура селекции. Здесь преимущество в смысле распараллеливания получают те формы селекции, которые не используют глобальную статистику о популяции, например «соревновательные» способы селекции. Легко распараллеливаемыми являются операторы мутации и кроссинговера, так как они выполняются независимо друг от друга и в основном случайным образом.

Второй тип параллелизма – миграционная модель – наиболее эффективно реализуется в вычислительных системах с ММД-архитектурой путем разделения общей популяции на отдельные подпопуляции, эволюционное моделирование которых осуществляется на отдельных процессорах. Если между отдельными подпопуляциями установить правила миграции отдельных индивидуумов, то получится миграционная модель (рис. 1). Для построения миграционной модели требуется задать количество и размеры подпопуляций, топологию связей между ними, частоту и интервал миграции, а также стратегию эмиграции и иммиграции. Размеры и число подпопуляций определяются числом и быстродействием используемых процессоров или транзисторов, а также сложностью решаемой оптимизационной проблемы. Что

касается топологии связей в миграционной модели, то здесь различаются два варианта: «островная» модель, предусматривающая миграцию между любыми подпопуляциями, и миграционная модель, предусматривающая обмен индивидами между соседними подпопуляциями. Применение того или иного варианта миграционной модели зависит от архитектуры параллельного вычислителя (сетевая или кольцевая). Частота миграции определяется размерами подпопуляций. При высокой частоте подпопуляция может быстро потерять гетерогенность. Наоборот, при слабой интенсивности миграции и большом числе подпопуляций возрастает вероятность получения нескольких хороших решений, однако сходимость ЭА замедляется.

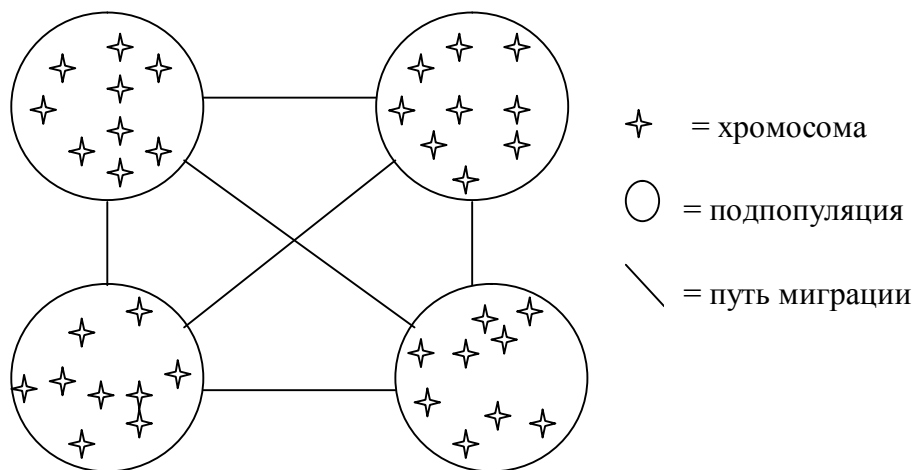


Рисунок 1 – Миграционная модель

Наконец, третья разновидность параллелизма ЭА – диффузионная модель – наиболее эффективно реализуется в вычислительных системах с SIMD-архитектурой. В этом случае общая популяция разделяется на большое число немногочисленных подпопуляций, а операторы кроссинговера и селекции применяются лишь в ограниченной области, определяемой отношением соседства (рис. 2). Это приводит к тому, что хорошие решения очень медленно распространяются в популяции (диффузия) и проблема попадания в «локальную яму» становится менее острой.

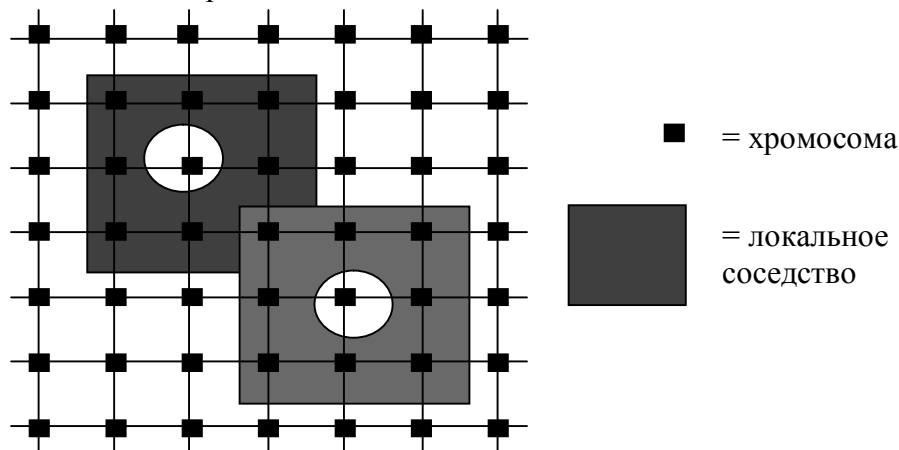


Рисунок 2 – Диффузионная модель

Популяция в природе, как правило, состоит из отдельных, более или менее изолированных локальных подпопуляций, допускающих миграцию или диффузию отдельных индивидуумов. Большинство генетических алгоритмов это обстоятельство игнорируется, алгоритмы работают с одной большой неструктурированной популяцией. Другое преимущество рассмотренных моделей состоит в том, что они допускают эффективную реализацию на массиве параллельных вычислителей с различной коммутационной структурой.

Ответ на вопрос о том, какая из представленных моделей параллелизма является более эффективной, зависит от архитектуры вычислительной системы. Эффективность при решении конкретной задачи принято оценивать двумя факторами: скоростью (время достижения заданного качества популяции) и устойчивостью (способность алгоритма к самостоятельному выходу из точки локального экстремума и способность постоянно увеличивать качество популяции от поколения к поколению). Эффективность представленных моделей параллелизма ЭА оценивалась путем тестирования. В качестве тестового примера была взята функция Клауса-Хотца [3]. Она является примером комбинаторной задачи определения перестановок. В частности, пусть задано множество $N = \{1, 2, \dots, n\}$. Если ψ – некоторая перестановка на множестве N , то требуется максимизировать функцию следующего вида:

$$F(\psi) = \sum_{i \neq j} \left| \frac{\psi_i - \psi_j}{i - j} \right| = 2 \cdot \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{|\psi_j - \psi_i|}{j - i}, i, j = 1, 2, \dots, n.$$

Функция $F(\psi)$ имеет несколько глобальных экстремумов. Ниже (табл. 1) для различных значений n приведены лучшие результаты.

Таблица 1

n	20	30	40	50	60	70	80	90	100
$F_{\max}(\psi)$	829.8	2034.7	4039.5	6680.4	10121.8	14343.1	20146.7	25538.6	32006.2

Другим примером применения моделей параллелизма ЭА является задача оптимизации архитектуры нейросети. Рассмотрим ее более подробно.

Оптимизация архитектуры нейросети

Процесс обучения нейросети заключается в подборе значений всех ее характеристик таким образом, чтобы отличия выходных векторов от эталонных не превышали заранее установленной величины. Поиск оптимального решения превращается в поиск лучшего решения, что позволяет говорить о применении ЭА в качестве основного или вспомогательного средства для улучшения нейросетей, уже созданных или обученных с помощью какого-либо алгоритма.

Популяция в ЭА состоит из хромосом (строинг, последовательность символов, с которыми работает алгоритм). Каждая хромосома представляется либо в

бинарном виде, либо в виде вектора действительных чисел, которые соответствуют весам входов нейронов и параметрам функций активации. Функционально связанные вход/выходные веса рекомендуется размещать в строке друг возле друга, поскольку они редко разрываются, например, одноточечным оператором кроссинговера. Недостатками бинарного кодирования являются ограниченная точность представления действительных чисел и размеры строки. Для больших сетей (сотни весовых значений) рекомендуется кодировать веса в виде вектора действительных чисел. Применение ЭА на этапе обучения нейросети имеет преимущество перед локально-оптимальными алгоритмами градиентного типа, поскольку ЭА позволяет быстрее и точнее вычислить конфигурацию весов сети, в отличие, например, от алгоритма обратного распространения [4]. Этот вывод справедлив не только для многослойного перцептрона, но и для большинства других архитектур нейросетей.

При обучении нейросети предполагается, что архитектура сети, включая число скрытых слоев, число нейронов и связи между ними, является известной. Поэтому оптимизация касается лишь значений весов. Предположим, что весовые значения нейросети уже инициализированы случайно или каким-либо другим методом. Целевая функция качества архитектуры нейросети требует вначале ее проверки на обучающей выборке по выбранным правилам. Оптимизация архитектуры даже для простейших нейросетей – это сложная комбинаторная проблема с большим пространством поиска среди альтернативных вариантов и недифференцируемой вероятностно-распределенной целевой функцией. Логичным выглядит решение о применении в качестве подходящего метода оптимизации ЭА, который в отличие от других методов позволяет учесть сложность сети через фитнес-функцию и улучшить качество сети. Требуется найти такой механизм конструирования нейросети, который бы позволял генерировать все потенциально интересные сетевые архитектуры и исключал бы бессмысленные структуры. Поиск такого механизма тесно связан с кодированием архитектурных решений. На практике обычно применяют прямой и непрямой способы кодирования.

Прямое кодирование предусматривает непосредственное задание связей между отдельными нейронами сети, например, с помощью матрицы смежности, состоящей из 0 и 1 (рис. 3) [5].

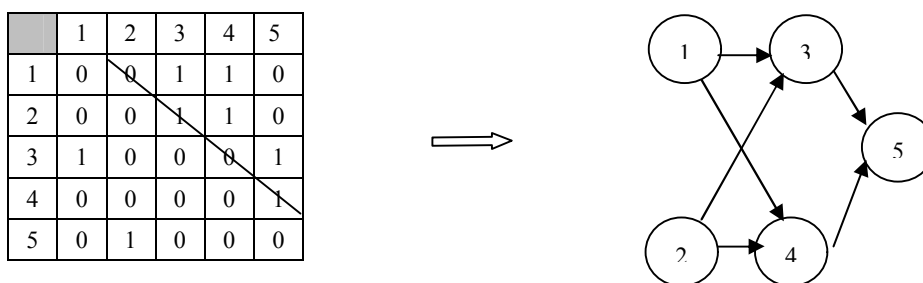


Рисунок 3 – Прямое кодирование архитектуры нейросети

Элементы треугольной части матрицы образуют строку вида **0110 110 01 1**. В случае прямого кодирования архитектуры нейросети необходимо считаться с ограничениями, например в виде запрета связей между нейронами внутри слоя. Другим не-

достатком является необходимость задания числа нейронов в сети и квадратичный рост длины стринга в зависимости от числа нейронов.

Непрямое кодирование позволяет избежать указанных недостатков [6]. Предлагается способ непрямого кодирования архитектуры многослойного перцептрона с помощью бинарного стринга, состоящего из нескольких сегментов, соответствующих слоям нейросети. Начало и конец каждого сегмента маркируются с целью упрощения последующего процесса декодирования информации об архитектуре сети. Кроме того, маркирование позволяет в ходе кроссинговера предотвратить появление бессмысленных стрингов. В свою очередь, параметры каждого сегмента состоят из двух подмножеств: подмножества параметров, определяющих число и структуру нейронов внутри слоя, и подмножества параметров, указывающих на связи данного слоя с другими слоями.

Другой способ непрямого кодирования предполагает использование грамматических правил. Идея этого способа заключается в генерации матриц смежности нейросети путем рекурсивного применения аксиоматических правил для контекстно-свободных грамматик. При этом грамматические правила представляют собой конструкции вида «Если (условия), то (действия)». Условная часть правил состоит из нетерминальных символов, а результирующая часть – из матриц размера (2x2), элементами которых могут быть терминальные и нетерминальные символы. Терминальные символы указывают на наличие или отсутствие связи между нейронами (1 или 0 соответственно), а нетерминальные символы позволяют рекурсивно применять грамматические правила. Таким образом, грамматика представляет собой стринг переменной длины, который кодирует множество правил. ЭА оптимизирует грамматические правила, используя стринги для декодирования по ним архитектуры нейросети. Приведем пример грамматического кодирования. Пусть дано правило вида

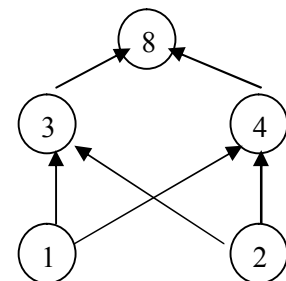
$$S \rightarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix},$$

где $A \rightarrow \begin{bmatrix} c & p \\ a & c \end{bmatrix}$, $B \rightarrow \begin{bmatrix} a & a \\ a & e \end{bmatrix}$, $C \rightarrow \begin{bmatrix} a & a \\ a & a \end{bmatrix}$, $D \rightarrow \begin{bmatrix} a & a \\ a & b \end{bmatrix}$, причем $a \rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$,

$b \rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$, $c \rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, ..., $p \rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$.

Тогда

$$S \rightarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix} \rightarrow \begin{matrix} c & p & a & a \\ a & c & a & e \\ a & a & a & a \\ a & a & a & b \end{matrix} \rightarrow \begin{matrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{matrix} \rightarrow$$



Стринг, кодирующий правило S, имеет вид: (S, A, B, C, D).

В частности, используя параллелизм ЭА, оптимизировался процесс построения эффективных сетевых архитектур. Размер популяции равнялся 100, ЭА использовал элитную селекцию, оператор мутации применялся с вероятностью от 0.02 до 0.3, оператор кроссинговера – с вероятностью 0.5. Весовые значения инициализировались случайно на интервале $[-1.0, 1.0]$, первоначальное обучение сети проводилось по алгоритму обратного распространения. Далее по обучающей выборке определялась сумма среднеквадратичной ошибки, а в качестве фитнес-меры качества архитектуры сети выбиралась величина, обратная полученной сумме.

Заключение

Теория эволюции как одна из базовых концепций для многих интеллектуальных процессов и систем может значительно расширить сферу применения классической компьютерной парадигмы искусственного интеллекта, основанной на символьном представлении информации и комбинаторно-логических процедурах ее обработки. Примером может служить идея соединения принципов эволюции с методами искусственных нейросетей.

Гибридизация различных эволюционных методов с нейросетями, а также с нечеткими множествами при использовании параллелизма на базе SIMD- и MIMD-архитектур, на наш взгляд, обеспечит их более широкое применение в таких сферах, как распознавание образов, таксономия, прогнозирование и т.п.

Одновременная оптимизация архитектуры и параметров нейросети с использованием комбинации прямого и непрямого кодирования и моделей параллелизма является перспективным нейроэволюционным методом. Этого можно достигнуть путем разделения каждого стринга на два сегмента, первый из которых представляет элементы списка матрицы смежности, а второй – значения весов и некоторых параметров сети. Далее, если в первом сегменте связь между нейронами является неактивной (0 в матрице смежности), то это необходимо учесть в ЭА в ходе обработки второго сегмента.

Решающим обстоятельством для оценки практической пригодности нейроэволюционных вычислителей является прежде всего то, насколько успешно с их помощью может быть решена та или иная оптимизационная задача с точки зрения качества решения и вычислительной сложности. Эксперименты подтвердили справедливость этого вывода не только для многослойного перцептрона, но и для других архитектур нейросетей.

Литература

1. Ковалев С.М., Родзин С.И. Информационные технологии: интеллектуализация обучения, моделирование эволюции, распознавание речи. – Ростов-на-Дону: Изд-во СКНЦ ВШ, 2002. – 224 с.
2. Курейчик В.М., Родзин С.И. Эволюционные алгоритмы: генетическое программирование. Обзор // Известия РАН. Теория и системы управления. – 2002. – № 1. – С. 127-137.
3. Whitley D. et al. Building Better Test Functions // Proc. of 6th International Conf. on GA. – San Francisco. – 1995.
4. Корнеев В.В. и др. Базы данных. Интеллектуальная обработка информации. – М.: Нолидж, 2001.

5. Harp S.A., Samad T. Genetic Synthesis of Neural Network Architecture. Handbook of GA. – NY: Van Nostrand Reinhold, 1991.
6. Родзин С.И. Гибридные интеллектуальные системы на основе алгоритмов эволюционного программирования // Новости искусственного интеллекта. – 2000. – № 3. – С. 159-170.

Neuroevolving Computation Parallelism. The article deals with approach to the realization of hybrid intelligent system, based on combinations of neural networks and evolutionary algorithms. Proposed neuroevolving optimization method of the network architecture construction process automation is oriented towards parallel computing systems. Three models of parallelism are presented: global parallelism, migration model and diffusion model. The combination of direct and indirect methods of multilayer perceptron architecture coding used for neural networks architecture and characteristics optimization.

Розглянуто підхід до реалізації гібридної інтелектуальної системи, заснованої на комбінуванні нейромереж і еволюційних алгоритмів. Пропонований нейроеволюційний метод оптимізації процесу побудови архітектури мережі орієнтований на паралельні обчислювальні системи. Подані три моделі паралелізму: глобальний паралелізм, міграційна модель, дифузійна модель. Для оптимізації архітектури та параметрів нейромережі використовується комбінація прямого й непрямого способів кодування архітектур багатoshарового перцептрона.

Статья поступила в редакцию 20.05.03.