

Эволюционный подход к настройке и обучению искусственных нейронных сетей

Цой Ю.Р.¹⁾, Спицын В.Г.²⁾

Томский политехнический университет, Томск, ¹⁾ qai@mail.ru,
²⁾ spitsyn@ce.cctpu.edu.ru

Аннотация. В статье рассматривается применение эволюционных алгоритмов (ЭА) для настройки структуры и поиска весов связей искусственных нейронных сетей (ИНС) – нейроэволюционный (НЭ) подход. Дается общая характеристика НЭ подхода, формулируются его преимущества и недостатки. Рассматриваются проблемы разработки НЭ алгоритмов и возможные пути их решения. Обосновывается использование эволюционного поиска весов связей ИНС для решения ряда задач, использующих качественную оценку функционирования ИНС.

Введение

Выбор топологии и настройка весов связей искусственной нейронной сети (ИНС) [1] являются одними из важнейших этапов при использовании нейросетевых технологий для решения практических задач. От этих этапов напрямую зависит качество (адекватность) полученной нейросетевой модели, системы управления и т.д. В данной статье в общем виде рассматривается нейроэволюционный (НЭ) подход, как один из возможных вариантов решения проблем настройки структуры и обучения ИНС.

Статья организована следующим образом. В первом разделе дан краткий обзор эволюционных алгоритмов. Во втором разделе в общем виде описывается нейроэволюционный подход. Вопросам эволюционного поиска топологии ИНС, настройки весов связей и одновременному решению обеих задач посвящены соответственно разделы 3, 4 и 5. Проблема генерации новых решений-кандидатов в результате использования оператора скрещивания обсуждается в разделе 6. В разделе 7 приводится общая характеристика задач, использующих качественную оценку ИНС, рассматриваются случаи, когда введение такой оценки позволяет упростить решение прикладных задач, а также расширить область применения методов нейроинформатики.

1. Краткий обзор эволюционных алгоритмов

Прежде чем перейти к описанию нейроэволюционного (НЭ) подхода кратко рассмотрим эволюционные вычисления (ЭВ), методы которых лежат в его основе. Эволюционные принципы наследственности, изменчивости и естественного отбора используются для решения как задач моделирования (например, модели молекулярной эволюции [2] и макроэволюционные модели в социологии и истории [3]), так и прикладных задач оптимизации. Идея использования механизмов эволюции для машинного обучения прослеживается и в хрестоматийном труде Алана Тьюринга «Могут ли машины мыслить?». Множество ал-

Эволюционный подход к настройке и обучению искусственных нейронных сетей

горитмов и методов, использующие для поиска решения эволюционные принципы, объединяют под общим названием *эволюционные вычисления (ЭВ)* или *эволюционные алгоритмы (ЭА)* [4] (соответственно, *evolutionary computation* и *evolutionary algorithms*). Выделяют следующие основные виды ЭА:

- генетический алгоритм (ГА) [5, 6];
- эволюционное программирование (ЭП) [7, 8];
- эволюционные стратегии (ЭС) [9, 10];
- генетическое программирование (ГП) [11, 12].

Здесь мы более подробно рассмотрим генетический алгоритм, как один из самых распространенных ЭА.

Идея ГА предложена Джоном Холландом в 60-х годах, а результаты первых исследований обобщены в его монографии «Адаптация в природных и искусственных системах» [5], а также в диссертации его аспиранта Кеннета Де Йонга [13]. Общая схема канонического ГА представлена на рис. 1. Необходимо отметить, что в настоящее время существует множество модификаций канонического ГА, предложенного в [5], и не все из них могут быть описаны схемой на рис.1.

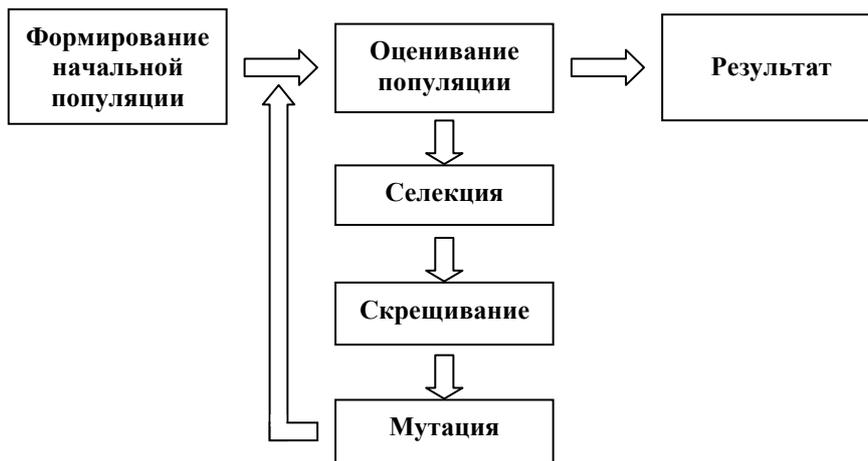


Рис. 1. Общая схема ГА. Комментарии к схеме в тексте.

ГА часто используются для решения оптимизационных задач, но, в отличие от большинства других алгоритмов (например, градиентные алгоритмы), на каждом этапе работы алгоритма рассматривается не одно потенциальное решение, а множество решений – *популяция*. Каждая *особь* популяции кодирует информацию о некотором решении в *хромосоме (генотипе)*, а декодированное из хромосомы решение иногда называют *фенотипом*.

В начале работы ГА популяция формируется, как правило, случайным образом (блок «**Формирование начальной популяции**» на рис. 1). Чтобы оценить качество закодированных решений (их *приспособленность*), используют *функцию приспособленности* (блок «**Оценивание популяции**»). По результатам оценивания особей наиболее приспособленные из них выбираются (блок «**Се-**

лекция») для скрещивания. В результате *скрещивания* выбранных особей посредством применения генетического оператора *кроссовера* создаются *потомки*, генетическая информация которых формируется в результате обмена хромосомной информацией между родительскими особями (блок «**Скрещивание**»). Созданные потомки формируют новую популяцию, причем часть потомков *мутирует* (с использованием генетического оператора *мутации*), что выражается в случайном изменении их генотипов (блок «**Мутация**»). Этап, включающий последовательность операций, обозначенных блоками «**Оценивание популяции**» – «**Селекция**» – «**Скрещивание**» – «**Мутация**», называется *поколением*. Эволюция популяции состоит из последовательности таких поколений.

Окончание эволюции может определяться следующими факторами:

- нахождение решения в результате эволюционного поиска;
- ограничение на число поколений, количество вычислений целевой функции, общее время работы ГА;
- вырождение популяции, когда степень разнородности хромосом в популяции становится меньше установленного значения¹.

Как правило, в хромосоме кодируются параметры решения. Для этого используют целочисленное, либо вещественное кодирование. Определение приспособленности *i*-й особи в популяции производится в соответствии с оценкой решения, определяемого набором параметров, закодированных в хромосоме *i*-й особи, и зависит от рассматриваемой задачи. Важным при использовании ГА для решения прикладных задач является предположение, что в результате повторяющегося отбора наиболее приспособленных особей, их скрещивания и мутации будет производиться отсев неудовлетворительных решений и постепенное повышение качества (приспособленности) существующих «хороших» решений.

Упомянутые выше другие виды ЭА несколько отличаются от ГА. Так, основным отличием ЭС от ГА является значительно меньшая роль оператора кроссовера, а в ЭП кроссовер не используется совсем. Также ЭС, и ЭП изначально разрабатывались не только для целочисленного, но и для вещественного кодирования. ГП, в отличие от ГА, ЭС и ЭП, которые часто используются для решения оптимизационных задач, предназначено для решения задач автоматизированного программирования и машинного обучения. Более подробное описание перечисленных видов ЭА, а также их особенностей приведено, например, в [15] и [16].

2. Нейроэволюционный подход

Первые работы, посвященные применению ЭА для обучения и настройки ИНС, появились около 20 лет назад. Исследования в этой области, как правило, рассматривают решение следующих задач:

- поиск значений весов связей ИНС при фиксированной структуре;

¹ Степень разнородности хромосом в популяции может определяться различными способами в зависимости от выбранного варианта кодирования информации. Примеры числовых характеристик разнообразия хромосом в популяции [14]: суммарное расстояние по Хеммингу для всех пар особей в популяции; использование аналога момента инерции; суммарная энтропия по всем разрядам в хромосомах; использование концепции лингвистической сложности (linguistic complexity) и др.

Эволюционный подход к настройке и обучению искусственных нейронных сетей

- настройка структуры ИНС без поиска весов связей;
- настройка параметров алгоритма обучения;
- настройка параметров активационных функций нейронов;
- фильтрация обучающих данных;
- различные комбинации вышеперечисленных задач.

Предлагались различные варианты названий для данного направления исследований. В настоящей статье будет использоваться термин «нейроэволюция» (НЭ) [17], а соответствующий подход будем называть нейроэволюционным. Отметим, что предлагаемый вариант не является устоявшимся и окончательным.

Комбинация ИНС и ЭА дает возможность совместить гибкость настройки ИНС и адаптивность ЭА, что позволяет реализовать во многом унифицированный подход к решению широкого спектра задач классификации, аппроксимации и моделирования (см. раздел 7). Тем не менее, у такого синтеза есть и недостатки. К числу важнейших из них необходимо отнести:

1. Отсутствие, в общем случае, гарантии сходимости НЭ алгоритма. Данный недостаток обусловлен, в первую очередь, универсальностью схемы эволюционного поиска. При этом в условиях конечного времени нельзя гарантировать ни получение решения, ни качество этого решения. Пользователь задает условия поиска в виде целевой функции и параметров эволюционного поиска, но стохастичность работы ЭА часто вносит свои коррективы. Частичным решением этой проблемы является использование гибридного подхода. Например, типичным является разделение этапов глобального и локального поиска, где ЭА используется для поиска приближенного, грубого решения, которое затем «доводится до ума» с использованием более «точных» методов, например, градиентных алгоритмов оптимизации [18-23].

2. Вероятностный характер ЭА ведет также и к потере контроля над процессом поиска решения. Результат отдельных операций, осуществляемых в процессе выполнения ЭА, часто не несет смысловой нагрузки, поэтому «отследить логику» работы ЭА, как правило, не представляется возможным. Данный недостаток существенен при использовании ЭА и НЭ алгоритмов для синтеза правил логического вывода, где помимо самих правил имеет значение и цепочка «рассуждений», приведшая к их появлению.

Однако «логика» поиска не всегда является важным условием, а по результатам нескольких запусков часто можно оценить качество решений, получаемых с помощью НЭ алгоритма. Полученные приближенные оценки позволяют сделать заключение о применимости НЭ алгоритма для решения поставленной задачи и, если необходимо, внести поправки в процесс эволюционного поиска.

3. Эволюционная настройка весов связей ИНС

Одной из типичных задач, решаемых в рамках НЭ подхода, является задача поиска весов связей ИНС при ее фиксированной структуре [3, 18-34]. При этом, как правило, рассматривается задача минимизации целевой функции (обычно функция ошибки выхода ИНС), а в качестве оптимизируемых параметров используются веса связей, значения которых подбираются с помощью ЭА. Одним из обоснований эволюционного обучения ИНС является «застывание» гради-

ентных алгоритмов в локальных экстремумах в процессе обучения ИНС. Кроме этого, как будет показано далее (см. раздел 7), эволюционный подход к поиску весов связей ИНС позволяет существенно упростить процесс решения ряда задач.

3.1. Плюсы и минусы эволюционной настройки весов связей ИНС. Существенными преимуществами эволюционной настройки весов связей ИНС являются:

1. Независимость от структуры ИНС и характеристик функций активации нейронов.
2. Отсутствие необходимости в обучающем множестве данных.

Первое преимущество дает возможность использовать единый подход к обучению ИНС с различной структурой.

Второе преимущество позволяет осуществлять обучение ИНС без информации об эталонных значениях выходных сигналов, а на основе оценки функционирования ИНС «в целом». Для градиентных алгоритмов оптимизации обучающее множество необходимо, чтобы подстраивать веса связей ИНС на основе расхождения значений реального и требуемого выходного сигнала ИНС. Подробнее этот вопрос обсуждается в разделе 7, здесь мы только отметим, что в случае эволюционной настройки весов связей ИНС появляется возможность использовать приближенные оценки функционирования ИНС, при этом оцениваться может не каждый выходной сигнал, а последовательность сигналов. Таким образом, упрощается поиск решения ряда задач.

К недостаткам использования ЭА для обучения ИНС следует отнести следующие:

1. Трудность «тонкой» настройки весов связей на поздних этапах эволюционного поиска.
2. Большие, по сравнению с градиентными алгоритмами, требования к объему оперативной памяти из-за использования популяции ИНС.
3. Проблема конкурирующих решений (см. подраздел 3.3).

Первая проблема во многом обусловлена использованием целочисленного кодирования весов связей ИНС, что может привести к значительным «скачкам» в пространстве поиска в результате применения операторов скрещивания и мутации. Данное свойство полезно на начальном этапе работы НЭ алгоритма, когда необходимо быстро исследовать пространство поиска, но затрудняет работу алгоритма на более поздних этапах.

Проблема конкурирующих решений заключается в том, что один и тот же набор весов связей может быть представлен в генетическом виде несколькими способами (см. подраздел 3.3).

3.2. Кодирование информации о весах связей ИНС. Как было сказано в начале текущего раздела, при эволюционном обучении ИНС, как правило, рассматривается задача минимизации целевой функции. Решение задачи часто заключается в отыскании такого набора значений весов связей ИНС, при которых функционирование ИНС оптимально с точки зрения выбранного критерия оценки.

Веса связей ИНС могут быть представлены с использованием как целочисленного, так и вещественного способов кодирования. Таким образом, хромосома каждой особи представляет вектор целочисленных, либо вещественных па-

раметров, соответствующих весам связей ИНС. Простые примеры кодирования весов связей показаны на рис. 2. В зависимости от выбранного варианта кодирования используются различные операторы скрещивания и мутации. Операторы для целочисленного кодирования достаточно подробно представлены в [6], для вещественного – в [35, 36]. Необходимо отметить, что использование нестандартных операторов, ориентированных на решение специфичных задач, способствует увеличению скорости работы и повышению качества результатов НЭ алгоритмов [37].

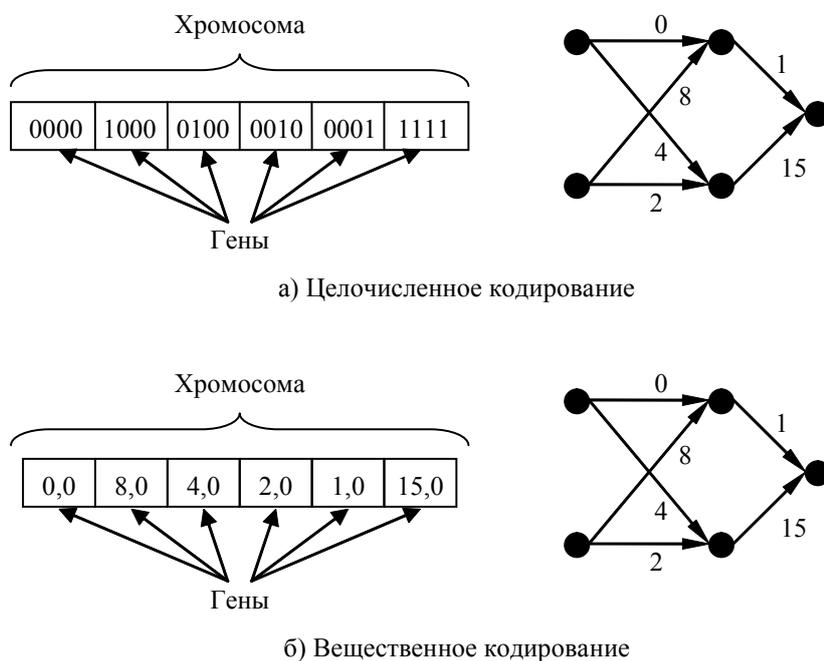


Рис. 2. Простые примеры целочисленного (а) и вещественного (б) кодирования весов связей для двух одинаковых ИНС. Здесь и на последующих рисунках кругами обозначены нейроны, стрелками показано направление связей, а числа возле стрелок соответствуют весам связей, если не указано иное.

3.3. Проблема конкурирующих решений. Одной из проблем, связанной с кодированием весов связей ИНС, является *проблема перестановки (permutation problem)* [28, 29], известная также как *проблема конкурирующих решений (competing conventions problem)* [38, 39]. Ее суть в том, что генетическое представление допускает существование нескольких вариантов хромосом, кодирующих одну и ту же ИНС (рис. 3).

Предложено несколько вариантов решения проблемы перестановки, среди которых необходимо отметить следующие:

1. Упорядочивание списка весов связей, кодируемых ИНС (выравнивание хромосом), по значениям весов [25], либо с использованием специальных меток [40]. Считается, что такие меры позволят уменьшить вероятность появления особей, представляющих одинаковые ИНС. Также возможно раннее удаление особей-дубликатов на этапе формирования популяции следующего поколения.

2. Разработка адаптивных операторов скрещивания [29, 41], уменьшение размера популяции, а также повышение вероятности мутации.

3. Уменьшение роли оператора кроссовера, либо полный отказ от его использования [42, 43]. При этом предполагается, что: (а) вероятность возникновения одинаковых ИНС в результате мутаций различных особей начальной популяции незначительна; (б) в случае, если такие особи все-таки появились, мутации приведут к меньшим «разрушениям», чем скрещивание.

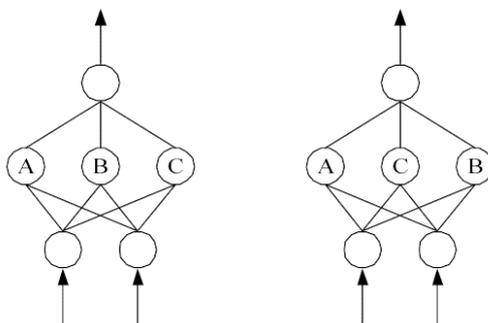


Рис. 3. Проблема перестановки. Две одинаковые ИНС могут быть закодированы различным образом «перестановкой» скрытых нейронов. Считается, что «скрещивание» таких сетей практически бесполезно (см. раздел 6) [40].

Отметим, что вопрос об актуальности проблемы конкурирующих решений остается открытым. Из-за того, что, в общем случае, вероятность появления в одной популяции двух одинаковых ИНС мала, проблема конкурирующих решений может игнорироваться либо вовсе не рассматриваться [17, 25, 44].

4. Эволюционная настройка структуры ИНС

Еще одной типичной задачей НЭ подхода является задача эволюционной настройки структуры ИНС. В хромосоме кодируется топология ИНС, а обучение осуществляется, например, с использованием градиентных алгоритмов [45-47]. Каждая особь, представляющая сеть той или иной структуры, оценивается в зависимости от результатов обучения: чем лучше результат, тем более приспособлена особь. Поскольку выбор топологии ИНС является, как правило, сложной задачей, решаемой методом проб и ошибок, то эволюционный поиск нейросетевой структуры способен облегчить и в определенной степени автоматизировать процесс решения задачи настройки и обучения ИНС.

4.1. Плюсы и минусы эволюционной настройки структуры ИНС. Перечислим преимущества и недостатки эволюционной настройки структуры ИНС.

Преимущества:

1. Возможность автоматического поиска топологии ИНС и получения более точной нейросетевой модели за счет рассмотрения «нестандартных», нерегулярных топологий.

2. Независимость от характеристик функций активации нейронов.

Для упрощения задачи и повышения качества результатов в процессе поиска топологии ИНС возможно использование дополнительных регулирующих ограничений, помогающих избежать чрезмерного «разрастания» сети, которое выражается в быстром увеличении количества скрытых нейронов и связей между ними.

Недостатки:

1. Сложность оценки структуры ИНС без информации о значениях весов связей.

2. Сложность организации поиска топологии ИНС.

Первый недостаток представляет основную проблему эволюционной настройки структуры ИНС. Он, в основном, обусловлен чувствительностью результатов обучения к начальным условиям и значениям параметров алгоритма обучения. Таким образом, хромосоме, представляющей ИНС с некоторой структурой, могут соответствовать совершенно разные нейросетевые модели с точки зрения их качества и свойств (в таких случаях говорят, что одному генотипу соответствует множество фенотипов). Для уменьшения влияния случайных факторов на оценку структуры ИНС проводится несколько независимых операций обучения [45, 47], усредненный результат которых и используется в качестве оценки топологии ИНС. Также, несмотря на то, что существуют эвристические правила, согласно которым ИНС с более простой структурой обладают лучшими способностями к обобщению, этот вопрос все еще нельзя считать до конца решенным, т.к. многое зависит от результатов обучения [48, 49].

Сложность объективной оценки топологии ИНС порождает также сложность оценки отдельных структурных частей сети и возможных структурных модификаций. Данная проблема характерна также и для алгоритмов одновременной настройки и обучения ИНС (см. раздел 5), поэтому многие дальнейшие замечания относятся в равной степени и к эволюционному поиску структуры ИНС, и к одновременной настройке структуры и обучению ИНС. Для изменения структуры ИНС часто применяются следующие операции:

- добавление/удаление нейронов;
- добавление/удаление связей.

Встречаются также модификации этих операций, включающие, например, «расщепление» существующего нейрона [43], «перенаправление» связи [17] и др.

Итак, выбор типа структурного изменения ИНС в общем случае неоднозначен из-за того, что оценка необходимости модификации топологии ИНС и варианта этой модификации не всегда возможна. Для решения проблемы выбора операции структурной модификации ИНС известны следующие варианты:

- определение эффективности каждой операции преобразования топологии ИНС на основании оценки качества получаемых НС решений [50, 51];

- использование дополнительной информации об ИНС для ее модификации (например, вычисление «значимости» связей [43], эвристическая оценка структуры ИНС [42, 52]).

4.2. Кодирование информации о структуре ИНС. Будем рассматривать два достаточно общих класса кодирования информации о топологии ИНС [37, 53]:

1. *Прямое кодирование (direct encoding)* – информация о структуре ИНС представлена в хромосоме в явном виде матрицей смежности [54, 55], списком связей [17, 40, 52, 56] и т.д. (рис. 4).

2. *Косвенное кодирование (indirect encoding)* – для описания структуры ИНС используются специальные грамматики [46, 57-59] (рис. 5 и 6).

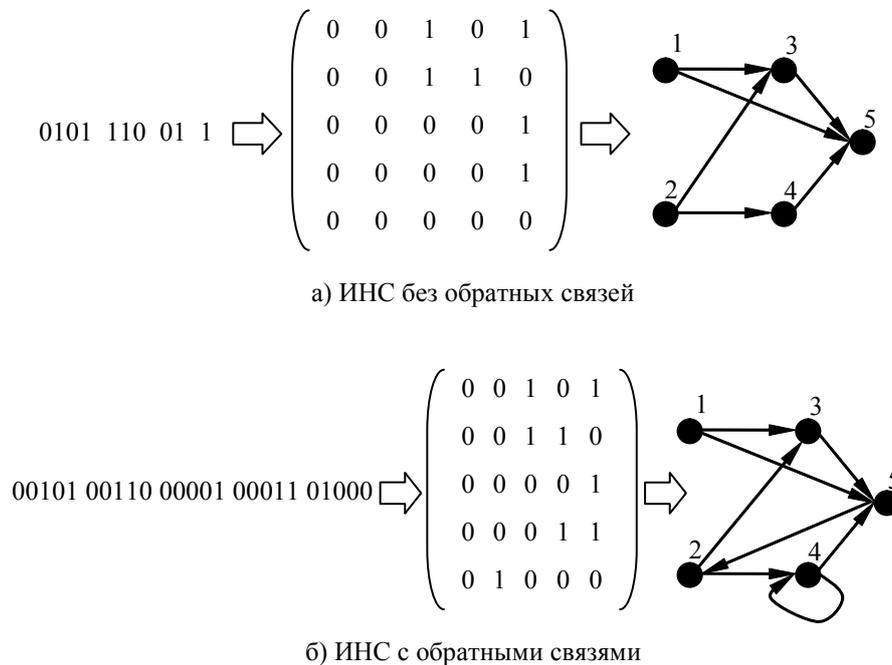


Рис. 4. Пример прямого кодирования структуры ИНС с использованием матрицы смежности для ИНС без обратных связей (а) и с обратными связями (б). Для каждого случая показаны (слева направо): хромосома, матрица смежности, структура ИНС с расставленными индексами нейронов.

Пример прямого кодирования структуры ИНС с использованием бинарной матрицы смежности $C = \{c_{ij}, i, j = 1, \dots, N\}$, где N – количество нейронов в ИНС, и соответствующие бинарные хромосомы представлены на рис. 4. Каждый элемент c_{ij} матрицы C принимает значения 1 или 0 и кодирует наличие связи (в случае $c_{ij} = 1$) от i -го нейрона к j -му. Если топология ИНС не имеет обратных связей (рис. 4а), то в хромосому достаточно записывать только части строк матрицы, находящиеся справа от главной диагонали.

1. Кодирование параметров структуры ИНС (количество слоев, число нейронов в каждом слое).

2. Введение специализированной грамматики, описывающей правила построения структуры ИНС. В данном варианте возможны два случая: (а) последовательность применения правил задана, но сами правила кодируются в хромосоме и отличаются у различных особей [46] (рис. 5); (б) в хромосоме кодируется последовательность применения правил, общих для всех особей, при этом набор правил и последовательность их применения для разных особей могут различаться [57, 59] (рис. 6).

Рассмотрим более подробно косвенное кодирование топологии ИНС. Иллюстративный пример эволюции правил построения структуры ИНС представлен на рис. 5 [37, 46]. Матрица смежности определяется в результате 3-шаговой итерационной процедуры, начиная с некоторого исходного состояния S (рис. 5в).

На первом шаге происходит декодирование символа S в матрицу четырех других символов $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$, каждый из которых затем также декодируется в соответствующую ему матрицу согласно правилам (рис. 5б), закодированным в хромосоме особи (рис. 5а) и т.д. Результатом является матрица смежности, определяющая структуру ИНС (рис. 5в). В хромосоме кодируются только правила, определяющие 1-й и 2-й шаги построения структуры ИНС, а 16 различных правил, применяемых на 3-м шаге, задающие все возможные бинарные матрицы 2×2 и обозначенные последовательно первыми 16-ю символами латинского алфавита (от «a» до «p»), определяются до начала работы алгоритма и не изменяются. Таким образом, какова бы ни была матрица смежности, сохраняется возможность закодировать ее содержимое в матрице 4×4 , содержащей символы от «a» до «p», которая затем может быть закодирована четырьмя символами $\{A, B, C, D\}$ в матрице 2×2 . Тем самым обеспечивается возможность представить в хромосоме любую структуру ИНС. Ограничениями описываемого способа эволюции правил является фиксированное количество шагов построения матрицы смежности, а также невозможность использования рекурсивных декодирующих правил (например, $A \rightarrow \begin{pmatrix} a & A \\ A & b \end{pmatrix}$).

Пример для эволюции последовательности применения правил для построения ИНС с бинарными весами связей¹, алгоритм «Клеточного кодирования» (Cellular encoding) Фредерика Груо [57], показан на рис. 6 [60]. Сами правила определяются до начала работы алгоритма, а в хромосоме кодируется дерево, соответствующее последовательности выполнения этих правил. Количество шагов декодирования и их последовательность зависят от структуры дерева правил. Способы генетического кодирования древообразных структур описаны, например, в [61, 62].

¹ Хотя, фактически, алгоритм Ф. Груо помимо структуры ИНС настраивает также и веса связей, он может быть адаптирован к решению задачи поиска топологии ИНС путем исключения правил, относящихся к изменению весов связей.

Эволюционный подход к настройке и обучению искусственных нейронных сетей

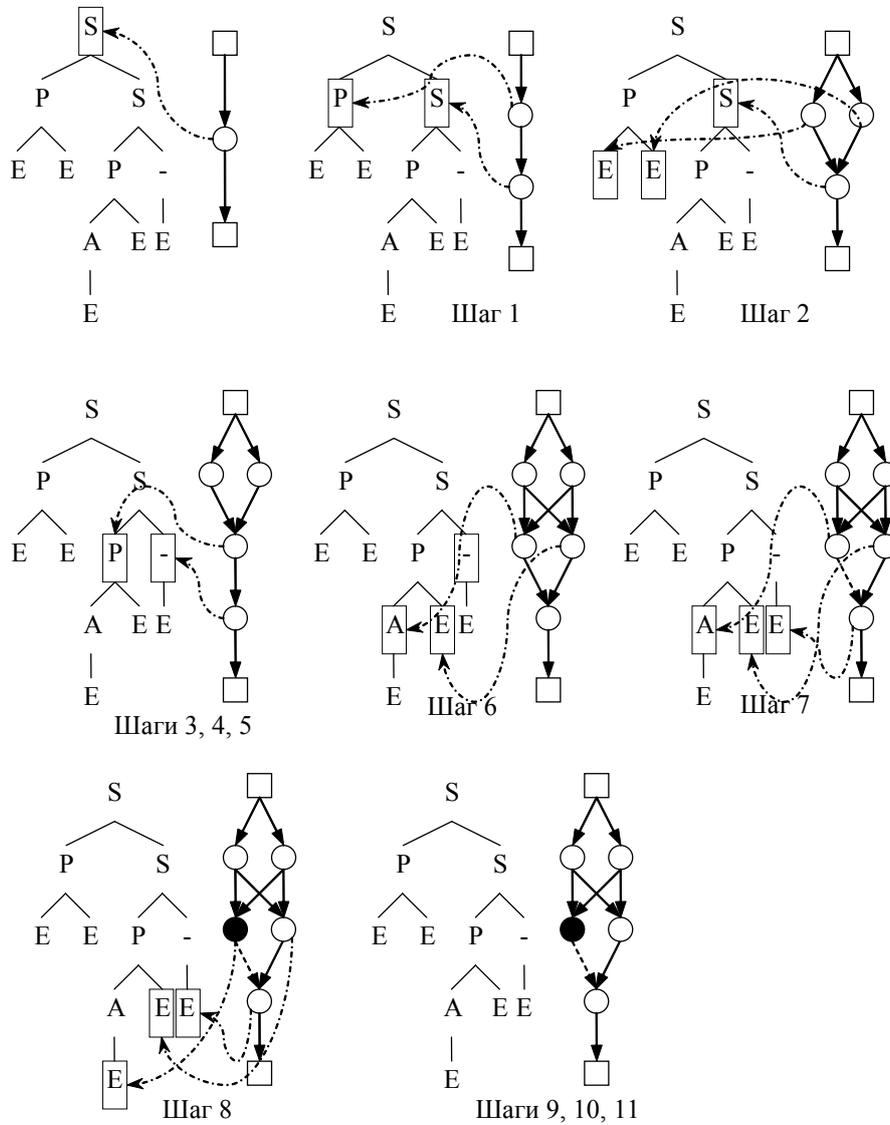


Рис. 6. Пример построения структуры ИНС с использованием специализированной грамматики и эволюцией последовательностей применения правил [60]. Описание операций, используемых при декодировании хромосомы, приведено в тексте.

Для примера, изображенного на рис. 6, используются следующие правила [60]:

- «S» – последовательное деление нейрона.
- «P» – параллельное деление нейрона.
- «A» – увеличение порога активационной функции нейрона.

«Е» – символ-терминатор, обозначающий окончание развития нейрона.

«-» – установка веса связи равным «-1».

На каждом шаге декодирования выполняется одно правило, записанное в узле дерева. Так, для примера на рис. 6, на первом шаге, в соответствии с правилом, обозначенным символом «S», нейрон делится последовательно, а на шаге 2 – параллельно, согласно правилу, обозначенному символом «P». После выполнения шагов 3, 4 и 5 два нейрона заканчивают развитие, а один делится последовательно и т.д.

Результатом является ИНС с 3-мя скрытыми слоями, в которой изменены значение порога функции активации одного скрытого нейрона (показано черным цветом) и величина одной связи (показано пунктиром). Кодирование в хромосоме дерева правил позволяет использовать рекурсивные операции для построения структуры ИНС [57]. Оригинальным и самостоятельным развитием описанного способа эволюции последовательности правил роста ИНС можно считать НЭ алгоритм, в котором учитываются пространственное расположение нейронов и длины связей между ними [63].

Заметим, что существуют неэволюционные подходы к настройке топологии сети либо во время ее обучения (алгоритм каскадной корреляции Фальмана [64] и его модификации [65] и др.), либо по завершении процесса обучения (алгоритмы Optimal Brain Damage (OBD) [66], Optimal Brain Surgeon (OBS) [67] и др.). Несмотря на то, что в [68] показана эффективность использования эволюционного подхода для настройки топологии ИНС по сравнению с алгоритмами OBD и OBS, однако детального исследования на эту тему, насколько известно авторам данной статьи, не проводилось.

5. Одновременная эволюционная настройка весов связей и структуры ИНС

Использование эволюционного подхода позволяет одновременно настраивать веса связей и структуру ИНС [17, 40, 42, 43, 50, 52, 54-59, 69-78]. При этом в хромосоме кодируется информация о весах и связях ИНС. Возможно использование как бинарного, так и вещественного кодирования для записи весов связей, а структура сети может быть представлена с использованием и прямого, и косвенного способов кодирования.

В силу того, что в случае одновременной настройки весов связей и структуры ИНС фактически комбинируются две различные задачи, пространство поиска многократно увеличивается и часто включает подпространства различной размерности (для ИНС с разным количеством связей). Для упрощения задачи поиска в таком сложном пространстве часто в явном или неявном виде вводятся ограничения на топологию ИНС и/или ее изменения:

- ограничение количества скрытых нейронов [17, 42, 50];
- ограничение количества связей [17, 59];
- ограничение изменения топологии ИНС (рассматриваются только растущие ИНС [40], либо только многослойные ИНС [73] и др.);

5.1. Плюсы и минусы одновременной настройки весов связей и структуры ИНС. Одновременное решение двух отдельных задач: настройки весов связей и структуры ИНС – позволяет в некоторой степени скомпенсировать недос-

татки, присущие каждой из них в отдельности и объединить их преимущества. С другой стороны, «платой» за это является огромное пространство поиска, а также объединение ряда недостатков, вызванных использованием эволюционного подхода. Суммируя, перечислим преимущества и недостатки.

Преимущества:

1. Независимость от структуры ИНС и характеристик функций активации нейронов.
2. Отсутствие необходимости в обучающей выборке.
3. Возможность автоматического поиска топологии ИНС и получения более точной нейросетевой модели.

Одним из основных преимуществ одновременной эволюционной настройки весов связей и структуры ИНС является возможность автоматизированного поиска ИНС, располагая только критерием оценки ИНС для осуществления эволюционного поиска. При этом, как и для эволюционного обучения ИНС (см. раздел 3), наличие обучающей выборки не является обязательным, а НЭ алгоритм может применяться для поиска ИНС с любыми активационными функциями нейронов.

По сравнению с отдельным эволюционным поиском структуры ИНС и настройкой весов связей, одновременное решение этих задач позволяет избежать некоторых недостатков. Так появление в популяции особей, которым соответствуют ИНС с различными топологиями, уменьшает значимость проблемы конкурирующих решений (см. подраздел 3.3), а наличие информации о весах связей позволяет обойти проблему субъективной оценки структуры ИНС, в силу того, что оценивается не структура нейросети, а вся ИНС «целиком».

Недостатки:

1. Сложность «тонкой» настройки весов связей на поздних этапах эволюционного поиска.
2. Большие, по сравнению с градиентными алгоритмами, требования к объему оперативной памяти из-за использования популяции ИНС.
3. Сложность организации поиска топологии ИНС.

5.2. Кодирование информации о структуре и весах ИНС. Поскольку веса связей кодируются вместе со структурой ИНС, то при записи информации о связи удобно записывать информацию о ее положении в сети и о величине веса вместе. Этим соображениям соответствует запись структуры ИНС в виде списка связей [17, 40, 52, 56, 74, 75]. При этом связи могут быть дополнительно сгруппированы по нейронам, для которых они являются входными [17, 56], что позволяет несколько сократить объем памяти, требуемый для записи информации об ИНС.

Для записи информации о структуре ИНС возможно также использование *кооперативного коэволюционного* подхода (*cooperative coevolution*) [79], когда решение записывается не в хромосоме каждой особи, а «разбивается на части» по хромосомам нескольких особей. Рассмотрим пример на основе алгоритма SANE Дэвида Мориарти [17]. В хромосоме особи записывается информация о входных и выходных связях одного нейрона. Во время оценки популяции формируется k ИНС, включающих нейроны, соответствующих различным особям (рис. 7), и производится их оценка. Оценка каждой особи вычисляется как средняя оценка ИНС, в состав которых входил нейрон, соответствующий этой осо-

би. Козволюционный подход также применяется и для эволюционной настройки весов связей ИНС [26, 27].

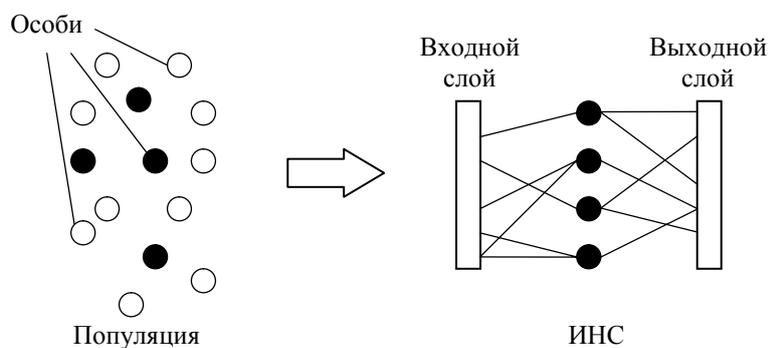


Рис. 7. Пример использования коэволюционного подхода к кодированию информации об ИНС для оценки особей. Черными кругами в популяции обозначены особи, соответствующие нейронам, включенным в структуру оцениваемой ИНС. Для определения приспособленности особей в популяции производится оценивание k различных ИНС. Рисунок представлен по [26].

Преимуществом использования коэволюционного подхода является уменьшение объема оперативной памяти необходимого для хранения информации о популяции. Также появляется возможность дополнительного распараллеливания эволюционного поиска за счет разбиения исходной задачи на отдельные подзадачи настройки связей скрытых нейронов. Недостатком является довольно сильное ограничение на структуру ИНС. Например, в [17] рассматриваются только сети прямого распространения с одним скрытым слоем, при этом количество скрытых нейронов, а также суммарное число входных и выходных связей для каждого скрытого нейрона фиксированы и задаются пользователем до начала работы алгоритма.

Информация о структуре ИНС и ее связях может также кодироваться отдельно. В работе [59] топология ИНС в хромосоме задается в виде дерева, а значения весов связей и величины порога функций активации нейронов записываются отдельно (рис. 8). Рассмотрим более подробно участок хромосомы, соответствующий верхней строке на рис. 8а. Рассматриваемый участок можно разбить на две части по выполняемым функциям:

1. Часть, кодирующая структуру ИНС («буквенная» часть участка хромосомы).
2. Часть, определяющая соответствие связей и нейронов с весами и порогами активационных функций соответственно («цифровая» часть участка хромосомы).

DTQaababaa (bbaabba0571745) 736284668 (2) 867

Веса: { -1,64; -1,834; -0,295; 1,205; -0,807; -0,856; 1,702; -1,026; -0,417; -1,061 }

Параметры: { -1,14; 1,177; -1,179; -0,74; 0,393; 1,135; -0,625; 1,643; -0,029; -1,639 }

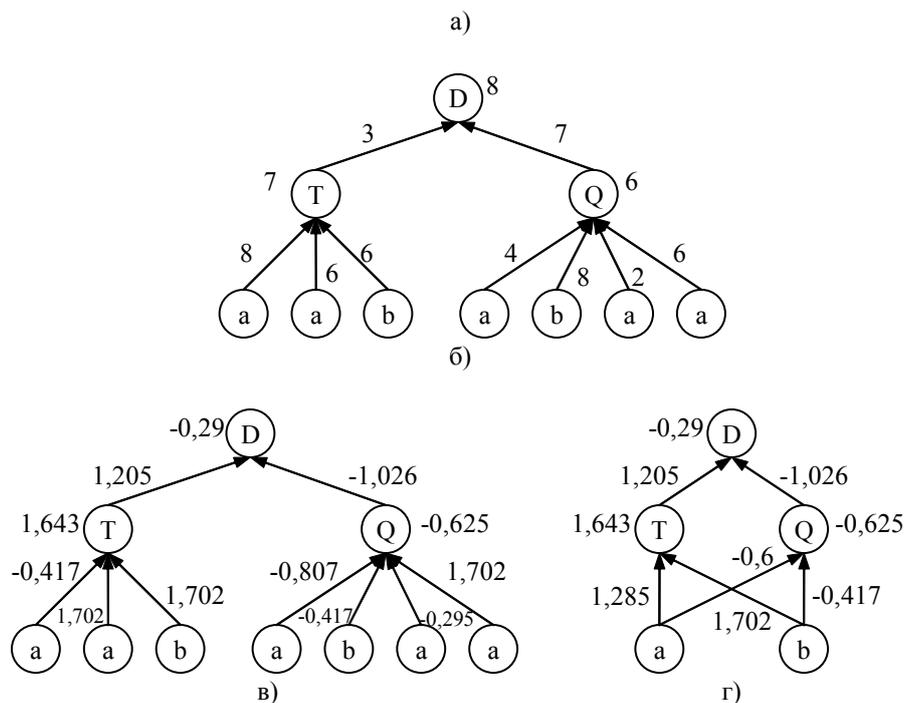


Рис. 8. Пример раздельного кодирования информации о структуре ИНС и значениях весов связей и параметрах функций активации [59]. Показаны: а) пример хромосомы; б) структура ИНС в виде дерева с индексами для весов связей и порогов функций активации нейронов; в) структура ИНС с «расставленными» весами связей и величинами параметров функций активации; г) структура ИНС в «стандартном» виде (связи от одинаковых входных нейронов объединены). Участки хромосомы, заключенные в круглые скобки, не учитываются при построении ИНС. Подробное описание правил декодирования хромосомы приведено в тексте.

В первой части рассматриваемого участка заглавные символы определяют количество входных связей нейронов сети («D», «T» и «Q» обозначают соответственно две, три и четыре входные связи), а строчные – входные нейроны ИНС («a» – первый входной нейрон, «b» – второй). Информация о связях «читается» в линейном порядке. Так, для части хромосомы «DTQaababaa» на рис. 8а, нейрон, соответствующий корню дерева (ИНС), обозначен символом «D» и поэтому имеет 2 входные связи от нейронов, обозначенных символами «T» и «Q». При этом «T-нейрон» имеет 3 входные связи от входных нейронов «a», «a» и «b»

(предполагается, что входные нейроны могут быть продублированы произвольное количество раз), а «Q-нейрон» имеет 4 входные связи от входных нейронов «a», «b», «a» и «a». Заметим, что оставшийся «отрезок» хромосомы «bbaabba» не используется. Таким образом, получаем структуру ИНС представленную в виде дерева на рис. 8б.

Во второй части участка хромосомы перечислены индексы для величин весов связей и порогов функций активации из отдельных одномерных массивов. Поскольку последние 7 символов из первой части пропускаются, то первые 7 символов из второй части «0571745» также не читаются. Следующие далее 9 индексов (по количеству связей) «расставляются» по соответствующим связям, начиная с корневого нейрона, при перечислении нейронов и связей справа налево, начиная от корня (рис. 8б). Для рассматриваемого примера индексы 7 и 3 определяют веса для входных связей к «корневому» нейрону («D-нейрон»), индексы 6, 2, 8 и 4 – веса связей для «Q-нейрона», а индексы 6, 6, 8 – веса связей для «T-нейрона».

Для определения величин порогов активационных функций следующий индекс в хромосоме (в данном случае «2», рис. 8а) пропускается, и записанные за ним индексы определяют величины порогов для нейронов в соответствии со значениями элементов отдельного массива пороговых величин. Порядок присвоения индексов – обратный. Результат расстановки индексов показан на рис. 8б, а соответствующая древовидная структура ИНС с определенными значениями весов связей – на рис. 8в.

Объединяя связи от одинаковых входных нейронов, получим «итоговую» ИНС на рис. 8г.

Для рассмотренного варианта раздельного кодирования структуры ИНС и весов связей характерна избыточность информации, записанной в хромосоме, что увеличивает требования к объему оперативной памяти, необходимого для хранения информации об одной ИНС, но, в то же время, оставляет «простор» для возможных дальнейших изменений структуры ИНС. Использование избыточности присутствует также в алгоритмах П. Ангелина [42] и К. Стенли [40, 73, 74].

Для одновременной настройки и обучения ИНС может использоваться косвенное кодирование информации [69]. В [69] рассматриваются бинарные ИНС, причем в хромосоме записывается информация только о росте, развитии ИНС, а величины весов связей по умолчанию считаются равными «1». В процессе декодирования хромосомы (см. рис. 6) значение веса связи может измениться и стать равным «-1».

Таким образом, существуют различные подходы к одновременному кодированию структуры и весов связей ИНС. Отметим, что чем сложнее запись информации об ИНС в хромосоме, тем, как правило, сложнее правила декодирования и больше вычислительные затраты на его реализацию. В [59, 70] показано, что подобные «сложные» способы кодирования могут использоваться для решения широкого круга задач, не только «нейросетевых», однако очень сложно сказать что-либо определенное об эффективности такого способа представления информации об ИНС.

6. Скрещивать или не скрещивать?

Оператор скрещивания (кроссовер, от англ. crossover) является основным оператором в генетическом алгоритме (ГА) и генетическом программировании (ГП). Одним из главных аргументов в пользу использования этого оператора является возможность рекомбинации параметров решений, представленных различными особями. Другими словами, в терминах *гипотезы о строительных блоках*¹ (*building blocks hypothesis*) [80], посредством кроссовера осуществляется поиск новых строительных блоков и смешивание уже существующих². Однако в эволюционных стратегиях (ЭС) кроссовер используется крайне редко, и совсем не используется в эволюционном программировании (ЭП).

Применительно к концепции эволюционирующих ИНС существует серьезный аргумент против использования оператора кроссовера [37, 42]. Его смысл может быть сформулирован следующим образом. Каждый нейрон в ИНС несет определенную функциональную нагрузку, например, реализуя разделяющую гиперплоскость (для нейронов с сигмоидной, линейной и ступенчатой функциями активации), либо выделяя центр некоторого кластера в пространстве входных параметров (для нейронов с радиально-базисными функциями активации и карт Кохонена). Таким образом, реализуемое ИНС отображение $X \rightarrow Y$, где X – пространство входных сигналов, Y – пространство выходных сигналов ИНС, существенно зависит от характеристик нейронов, их функциональной нагрузки и векторов весов их входных связей. Использование кроссовера для рекомбинации параметров двух ИНС может стать причиной существенного искажения преобразований, соответствующих ИНС родительских особей, и, тем самым, привести к потере существовавших решений. Другими словами, «скрещивание» ИНС без учета их свойств даст скорее неудовлетворительный результат, чем будет способствовать поиску решения³.

Если рассматривать проблему перестановки (см. подраздел 3.3, рис. 3), то скрещивание различных хромосом, кодирующих одинаковые ИНС, может привести к тому, что в структуре ИНС особей-потомков одни скрытые нейроны с одинаковой функциональной нагрузкой будут продублированы, а другие скрытые нейроны будут отсутствовать.

¹ Гипотеза о строительных блоках предполагает существование различных частей хромосом разной длины – строительных блоков (СБ) и их комбинаций. Согласно этой гипотезе, наличие СБ и комбинаций СБ в некоторой хромосоме явно и практически всегда однозначно (положительно или отрицательно) сказывается на приспособленности соответствующей ей особи. Тогда эволюционный поиск сводится к *выявлению (identification)* «хороших» строительных блоков и к поиску их комбинаций, *смешиванию (mixing)*, способствующих повышению приспособленности особей.

² Существует и «биологический» аргумент в пользу кроссовера [81]. Его суть заключается в том, что скрещивание, отсутствующее у простейших, «изобретено» эволюцией для развития сложных организмов насекомых, рыб, земноводных, птиц и млекопитающих как ответ на усложнение их генетического кода, морфологии и динамичные условия жизни. Исходя из этих соображений, делается вывод о необходимости использования оператора кроссовера.

³ Аналогичная проблема, когда использование кроссовера часто приводило к появлению «нежизнеспособных» решений существовала и при применении ГА для решения задач на графах (задача коммивояжера, поиск минимальных покрытий и др.), пока не были разработаны специализированные операторы, учитывающие специфику задачи.

Одним из естественных вариантов решения проблемы «скрещивания» ИНС является разработка адаптивных операторов скрещивания, которые учитывают различия в структуре скрещиваемых ИНС и создают «жизнеспособных» потомков, либо использование такого способа кодирования информации, который позволяет уменьшить ошибки, возникающие при использовании «стандартных» операторов скрещивания и мутации. Отметим, что разработка проблемно-ориентированных генетических операторов является одним из важнейших составляемых в разработке эффективных ЭА [82].

7. Нейроэволюционные алгоритмы и задачи с качественной оценкой ИНС

Обучение ИНС, с «классической» точки зрения, подразумевает, что существует обучающая выборка, использование которой дает возможность подстраивать веса связей ИНС с помощью градиентных алгоритмов, среди которых широко известны алгоритм обратного распространения ошибки, сопряженных градиентов, Левенберга-Марквардта и др. Однако, в ряде случаев формирование обучающей выборки сопряжено со следующими трудностями:

1. Необходимость переформулировки исходной задачи, в результате чего могут появиться дополнительные подзадачи, мало связанные с исходной проблемой, что делает решение громоздким. Например, один из вариантов решения задачи нейроуправления заключается в нейросетевой аппроксимации объекта управления с последующей оптимизацией состояния управляемого объекта посредством поиска соответствующих значений управляющих сигналов.

2. Сложность определения значений компонент вектора Y из обучающего множества. Данная проблема возникает при решении некорректных задач, а также, если необходимо оценить последовательность выходных сигналов ИНС. К таким задачам относятся задачи, связанные с адаптивным поведением, игровыми стратегиями, обработкой изображений и др.

Одним из возможных решений перечисленных проблем является обучение ИНС с использованием приближенной, интегральной оценки, отражающей качественные, внешние особенности ее функционирования. Будем называть оценку ИНС *качественной неполной* оценкой или, коротко, *качественной* оценкой¹, если эта оценка позволяет с требуемой точностью оценить качество функционирования ИНС, но при этом ее значения недостаточно для подстройки весов связей ИНС с использованием градиентных алгоритмов. Примерами качественной оценки являются: время поддержания стабильного состояния объекта управления для задачи нейроуправления [17, 26, 40, 52, 83]; процент выигранных игр, либо оценка позиции для задачи поиска игровых стратегий [28, 72, 74, 77]; качество изображения для задачи обработки изображений [84] и т.д. Такие оценки более естественны и «интуитивны», однако, как следствие, они не могут быть использованы градиентными алгоритмами обучения для подстройки весов

¹ Использование термина «качественная» оценка, очевидно, является не самым лучшим вариантом, т.к. означает существование не только «некачественной», но также и «количественной» оценок. При этом возникает закономерный вопрос о формальном разделении этих оценок, и оказывается, что провести четкие границы совсем непросто.

связей. Поэтому «подходящий» обучающий алгоритм должен изменять веса связей ИНС, не имея информации о величине ошибки для каждого выхода этой НС.

Отметим, что в случае «стандартного» обучения ИНС, для ее оценки и обучения, фактически, используются различные величины: среднеквадратичная ошибка выхода ИНС отражает качество ее функционирования, в то время как разность реального и требуемого выходных сигналов ИНС используется для подстройки весов связей. Т.е. качественная оценка присутствует и здесь, но она не используется напрямую для обучения ИНС, для осуществления которого необходимо репрезентативное с точки зрения решаемой задачи множество наборов входных и соответствующих им выходных данных.

При использовании ЭА для обучения ИНС достаточно одной качественной оценки. Здесь важна *адекватность* оценки, т.е. если согласно используемой оценке качество первой ИНС лучше качества второй ИНС, то из этого должно следовать, что первая ИНС лучше второй. Кроме адекватности оценки также важно ее *существование* для любого решения, которое может быть закодировано в хромосоме. При этом оценка может вычисляться независимо от расхождения выходного сигнала ИНС с эталонным значением. Таким образом, отсутствие обучающей выборки не является серьезным препятствием для эволюционного обучения ИНС, если существует альтернативный способ оценки ее функционирования. Это позволяет расширить круг практических приложений нейроинформатики, а также упростить процесс решения многих задач, связанных с адаптивным нейроуправлением, поведением и моделированием.

Рассмотрим коротко задачи, для решения которых использование качественной оценки ИНС помогает упростить процесс поиска решения и повысить качество результатов.

7.1. Адаптивное управление и поведение. При решении задач, связанных с управлением сложным динамическим объектом, эталонный управляющий сигнал в общем случае неизвестен. Сложность синтеза исполняющего устройства (ИУ) часто связана со сложностью сопоставления оценки ИУ и необходимых изменений структуры и параметров ИУ. Здесь одним из возможных решений является использование *обучения с подкреплением (reinforcement learning)* [85], однако для его применения необходимо оценивать каждое действие ИУ, что не всегда доступно. Использование ЭА для обучения ИНС позволяет использовать оценку нейросетевого ИУ по серии управляющих сигналов. Некоторые общие вопросы применения эволюционирующих ИНС для решения задач теории регулирования рассматриваются в [34].

В случае, когда ОУ является искусственным интеллектуальным агентом (виртуальным либо реальным, например, роботом), а внешняя по отношению к агенту среда является аналогом среды обитания, то задача управления таким агентом может рассматриваться как задача адаптивного поведения. При этом использование ИНС в качестве ИУ агента позволяет добиться большой гибкости действий агента вместе с его обучаемостью [2, 3, 75]. Во многих исследованиях поведение агентов определяется его НС [3, 22, 26, 31, 75, 86], при этом структура сети часто фиксирована. Возможность изменения топологии ИНС в ходе эволюции позволяет выделить и изучить интересные аспекты поведенческой адаптации и повысить, в целом, уровень «интеллектуальности» поведения

агентов. Таким образом, НЭ подход представляет интерес для таких направлений исследований как «Искусственная жизнь» (*Artificial Life, A-Life*) [2, 87] и «Адаптивное поведение» (*Adaptive Behavior*) [2, 88].

Отдельным направлением в исследовании адаптивного поведения является синтез оптимальных игровых стратегий. Часто рассматриваются настольные игры (например, крестики-нолики [72] и го [28, 74, 77]). При этом возможно использование «автообучения», когда настройка весов ИНС осуществляется путем соревнования одной популяции ИНС с другой [77] (т.н. *конкурентная коэволюция (competitive coevolution)*).

7.1.1. Проект NERO К. Стэнли. Ярким примером применения НЭ подхода к решению задач адаптивного поведения является проект NERO (Neuro-Evolution of Robotic Operatives, http://dmc.ic2.org/nero_public) [75]. Ключевой технологией проекта является модифицированный алгоритм NEAT для одно-временной настройки структуры и весов связей ИНС, разработанный Кеннетом Стэнли (Kenneth Stanley). Результатом проекта является интерактивная компьютерная игра, в которой пользователь сначала тренирует на специальном тренировочном полигоне команду агентов, а затем его команда «соревнуется» с агентами компьютера или другого пользователя на своеобразном «поле боя».

Поведение агентов игрока определяется соответствующими им ИНС, которые настраиваются в процессе обучения на полигоне. Действия агентов зависят от создающейся на тренировочном полигоне ситуации и определяются приоритетами поведения (расстояние до противника, подвижность агентов, активность стрельбы и др.), заданными пользователем. Более приспособленные с точки зрения выбранных приоритетов агенты используются для генерации агентов-потомков, которые замещают худших существующих агентов.

Для выбора действия (направление движения и стрельба) агентам доступна следующая информация:

- расположение агентов-противников, находящихся в поле зрения агента;
- находится ли агент-противник на линии огня;
- расположение препятствий (стены, различные объекты);
- направление стрельбы ближайшего агента-противника.

Использование для реализации проекта NERO НЭ алгоритма позволяет в реальном времени осуществлять поиск разнообразных стратегий поведения агентов. В процессе обучения агенты учатся решать такие задачи, как поиск маршрута движения в присутствии препятствий, преследование агентов-противников, использование стен для прикрытия от огня и избежания окружения и др. Отметим, что во время обучения агентов задачи ставятся в общем виде (дойти до определенной точки, атаковать цель и т.д.) и возникновение сложных стратегий поведения (например, укрытие за стенами) зависит от ситуации, которую создаст на тренировочном полигоне игрок.

7.2. Многоагентные системы. Поскольку во многих НЭ алгоритмах каждая особь представляет отдельную ИНС, то популяция может рассматриваться как эволюционирующая многоагентная система, в которой стратегия поведения каждого агента определяется соответствующей ему нейросетью [89]. Один из «классических» подходов к реализации стратегии поведения агента заключается в использовании набора правил формальной логики, состав которого формируется в результате действий агента. Среди недостатков такого подхода выделим

сложность синтеза правил, определяющих как приобретение агентом знаний, так и обмен знаниями между агентами, а также разрастание размера базы знаний агента в процессе накопления им «жизненного опыта». Применение ИНС, реализующей стратегию поведения агента, позволяет хранить знания в неявном виде в компактной форме, при этом сохраняется возможность дообучения и переобучения агента. Отметим, что взаимодействие НЭ-агентов в многоагентной системе возможно путем не только формального «общения» или противодействия, но и посредством применения операций селекции (своеобразное «соревнование» агентов) и скрещивания (обмен генетической информацией). При скрещивании агентов возможен обмен информацией, либо имевшейся у них в момент рождения (дарвиновская эволюция), либо накопленной ими в процессе жизни (эволюция ламарковского типа). В роли качественной оценки агента может использоваться показатель успешности его действий, определяемый в соответствии с поставленной задачей.

Заключение

Аппроксимирующие возможности ИНС очень высоки, однако волнующий вопрос «Как научить ИНС делать все?» поднимавшийся, например, в [90], остается открытым. С одной стороны, концепция ИНС является универсальной по отношению к решению широкого спектра задач, а с другой, существуют проблемы (например, задачи нейроуправления), сама формулировка которых вынуждает искать «обходные» пути. В то же время, известен эволюционный подход к решению оптимизационных задач самой разной природы и претендующий на универсальность применения, наравне со случайным поиском и последовательным перебором, поскольку может рассматриваться как формализованный метод проб и ошибок.

Совместное использование ЭА и ИНС позволяет решать задачи настройки и обучения ИНС как по отдельности, так и одновременно. Одним из достоинств такого синтезированного подхода является во многом унифицированный подход к решению разнообразных задач классификации, аппроксимации, управления и моделирования. Использование качественной оценки функционирования ИНС (подробнее см. раздел 7) позволяет применять НЭ алгоритмы для решения задач исследования адаптивного поведения интеллектуальных агентов, поиска игровых стратегий, обработки сигналов. Несмотря на то, что количество проблем и открытых вопросов, касающихся разработки и применения НЭ алгоритмов (способы кодирования, генетические операторы, методы анализа и др.) велико, часто для успешного решения задачи с использованием НЭ алгоритма достаточно адекватного понимания проблемы и НЭ подхода, свидетельством чего является большое число интересных и успешных работ в данном направлении.

Данная статья не претендует на полноту излагаемого материала, так, например, «за бортом» остались такие задачи как синтез правил формальной логики, создание компьютерной музыки, обработка сигналов и изображений и др. Однако авторы надеются, что содержащейся в статье информации достаточно для формирования общего представления об использовании эволюционных прин-

ципов для настройки и обучения ИНС при решении как «классических» задач классификации и аппроксимации, так и для задач с качественной оценкой ИНС.

Авторы выражают благодарность рецензентам за ценные замечания, способствовавшие улучшению статьи.

Литература

1. Осовский С. Нейронные сети для обработки информации. М.: Финансы и статистика, 2002.
2. Редько В.Г. Эволюционная кибернетика. М.: Наука, 2001.
3. Бурцев М.С. Исследование новых типов самоорганизации и возникновения поведенческих стратегий. Диссертация на соискание ученой степени кандидата физико-математических наук. Институт прикладной математики им. В.М. Келдыша РАН, Москва, 2005. См. также: <http://mbur.narod.ru/>
4. Beyer H.-G., Schwefel H.-P., Wegener I. How to analyse evolutionary algorithms. Technical Report No.CI-139/02. University of Dortmund, Germany, 2002. See also: <http://ls11-www.cs.uni-dortmund.de/people/schwefel/WelcomeE.html>
5. Holland J.H. Adaptation in Natural and Artificial Systems. The University of Michigan Press, 1975.
6. Емельянов В.В., Курейчик В.В., Курейчик В.М. Теория и практика эволюционного моделирования. М.: ФИЗМАТЛИТ, 2003.
7. Фогель Л., Оуэнс А., Уолш М. Искусственный интеллект и эволюционное моделирование. М.: Мир, 1969.
8. Родзин С.П. Гибридные интеллектуальные системы на основе алгоритмов эволюционного программирования // Новости искусственного интеллекта, 2000, № 3, с. 159-170.
9. Rechenberg I. Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution. Werkstatt Bionik und Evolutionstechnik, Stuttgart: Frommann-Holzboog, 1973.
10. Schwefel H.-P. Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie // Interdisciplinary Systems Research, 1977. Vol. 26, pp. 5-8.
11. Koza J. Genetic programming: a paradigm for genetically breeding computer population of computer programs to solve problems. Cambridge, MA: MIT Press, 1992.
12. Курейчик В.М., Родзин С.И. Эволюционные алгоритмы: генетическое программирование. Обзор // Известия РАН. ТИСУ, 2002, № 1, с. 127-137.
13. De Jong K. An analysis of the behavior of a class of genetic adaptive systems. Unpublished PhD thesis. University of Michigan, Ann Arbor, 1975. (University Microfilms No. 76-9381). See also: <http://www.cs.gmu.edu/~eclab/>
14. Mattiussi C., Waibel M., Floreano D. Measures of Diversity for Populations and Distances Between Individuals with Highly Reorganizable Genomes // Evolutionary Computation, 2004, no. 12(4), pp. 495-515. See also: <http://lis.epfl.ch/member.php?SCIPER=111729>
15. Whitley D. An Overview of Evolutionary Algorithms // Journal of Information and Software Technology, 2001, no. 43, pp. 817-831. See also: <http://www.cs.colostate.edu/~genitor>
16. Evolutionary Computation 1: Basic Algorithms and Operators / Back T., Fogel D.B., Michalewicz Z. (Eds.). Bristol and Philadelphia: Institute of Physics Publishing, 2000.
17. Moriarty D., Miikkulainen R. Efficient Reinforcement Learning through Symbiotic Evolution // Machine Learning, 1996, no. 22, pp. 11-33. See also: <http://www.nn.cs.utexas.edu/>

18. Montana D., Davis L. Training feedforward neural networks using genetic algorithms // Proceedings of the 11th International Joint Conference on Artificial Intelligence. San Mateo, CA: Morgan Kaufmann, 1989. P. 762–767.
19. Belew R. K., McInerney J., Schraudolph N. N. Evolving networks: Using the genetic algorithm with connectionist learning // Artificial Life II, 1992, pp. 511-547.
20. Heistermann J. Different learning algorithms for neural networks – a comparative study / In Y. Davidor, H.P. Schwefel, R. Manner (Eds.) Parallel Problems Solving from Nature, Workshop Proceedings. Springer-Verlag, 1994. P. 368-396.
21. Комарцова Л.Г. Исследование алгоритмов обучения многослойного персептрона // Нейрокомпьютеры: Разработка и применение. М.: Радиотехника, 2002, № 12.
22. Мосалов О.П., Редько В.Г. Модель эволюционной ассимиляции приобретенных навыков в нейросетевых системах управления адаптивных агентов // Научная сессия МИФИ-2005. VII Всероссийская научно-практическая конференция "Нейроинформатика-2005": Сборник научных трудов. В 2-х частях. Часть 1. М.: МИФИ, 2005. С.210-217. См. также: <http://www.iont.ru/projects/rfbr/90197/>
23. Valsalam V., Bednar J., Miikkulainen R. Constructing Good Learners using Evolved Pattern Generators // In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). Washington, DC, 2005. See also: <http://www.nn.cs.utexas.edu/>
24. Thierens D., Suykens J., Vanderwalle J., De Moor B. Genetic Weight Optimization of Feedforward Neural Network Controller // Proceedings of International Conference on Artificial Neural Networks and Genetic Algorithms. Springer-Verlag, 1993. P. 658-663.
25. Thierens D. Non-Redundant Genetic Coding for Neural Networks. Technical report No.UU-CS-1998-46. Utrecht University, Netherlands, 1998. See also: <http://www.cs.uu.nl/research/techreps/aut/dirk.html>
26. Gomez F., Miikkulainen R. Incremental Evolution of Complex General Behavior // Adaptive Behavior, 1997, no.5, pp. 317-342. See also: <http://www.nn.cs.utexas.edu/>
27. Gomez F., Miikkulainen R. Active Guidance for a Finless Rocket using Neuroevolution // Proceedings of Genetic and Evolutionary Computation Conference 2003 (GECCO 2003). Washington, USA, 2003. See also: <http://www.nn.cs.utexas.edu/>
28. Perez-Bergquist A. S. Applying ESP and Region Specialists to Neuro-Evolution for Go. Technical Report CSTR01-24. The University of Texas at Austin, 2001. See also: <http://www.nn.cs.utexas.edu/>
29. Saravanan N., Fogel D. B. Evolving neural control systems // IEEE Expert, 1995, pp. 23-27.
30. Korning P.G. Training of Neural Networks by means of Genetic Algorithms Working on very long Chromosomes // International Journal of Neural Systems, 1995, Vol. 6, No. 3, pp. 299-316. См. также: <http://citeseer.ist.psu.edu/>
31. Lund H.H., Mayoh B.H. Specialization in Populations of Artificial Neural Networks // Proceedings of the 5-th Scandinavian Conference on Artificial Intelligence (SCAI'95). Amsterdam: IOS Press, 1995. См. также: <http://citeseer.ist.psu.edu/>
32. Wieland A. Evolving neural network controllers for unstable systems // Proceedings of the International Joint Conference on Neural Networks. Piscataway, New Jersey, 1991. pp. 667–673.
33. Yamamichi T., Saito T., Taguchi K., Torikai H. Synthesis of Binary Cellular Automata based on Binary Neural Networks // Proceedings of International Joint Conference on Neural Networks. Montreal, Canada, 2005. P. 1361-1365.
34. Вороновский Г.К., Махотило К.В., Петрашев С.Н., Сергеев С.А. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. Харьков: Основа, 1997. См. также: <http://neuroschool.narod.ru/books/gannvirt.html>
35. Herrera F., Lozano M., Verdegay J.L. Tackling Real-Coded Genetic Algorithms: Operators and tools for the Behaviour Analysis // Artificial Intelligence Review, 1998, no. 12, pp. 265-319. See also: <http://decsai.ugr.es/~herrera/>

36. Deb K., Anand A., Joshi D. A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization. KanGAL Report No.2002003. Indian Institute of Technology, Kanpur, 2002. See also: <http://www.iitk.ac.in/kangal/>
37. Yao X. Evolving artificial neural networks // Proceedings of the IEEE, 1999, no. 9(87), pp. 1423-1447. See also: <http://www.cs.bham.ac.uk/~xin/>
38. Whitley D., Starkweather T., Bogart C. Genetic Algorithms and Neural Networks: Optimizing Connections and Connectivity // Parallel Computing, 1990, Vol. 14, pp. 341-361. See also: <http://www.cs.colostate.edu/~genitor/>
39. Shaffer J., Whitley D., Eshelmann L. Combination of Genetic Algorithms and Neural Networks: A Survey of the State of the Art // In Proceedings of the International Workshop on Combination of Genetic Algorithms and Neural Networks (COGANN-92). Los Alamos, CA: IEEE Computer Society Press, 1992. pp. 1-37. See also: <http://www.cs.colostate.edu/~genitor/>
40. Stanley K.O., Miikkulainen R. Evolving Neural Networks through Augmenting Topologies // Evolutionary Computation, 2002, no. 2(10), pp. 99-127. See also: <http://www.nn.cs.utexas.edu/>
41. Radcliffe N. Genetic set recombination and its application to neural network topology optimization // Neural Computing and Applications, 1993, Vol.1, pp. 67-90.
42. Angeline P.J., Saunders G.M., Pollack J.B. An Evolutionary Algorithm that Constructs Recurrent Neural Networks // IEEE Transactions on Neural Networks, 1994, no. 5(1), pp. 54-65. See also: <http://citeseer.ist.psu.edu/>
43. Yao X., Liu Y. Making Use of Population Information in Evolutionary Artificial Neural Networks // IEEE Transactions on systems, man, and cybernetics - part B: Cybernetics, 1998, no. 28(3), pp. 417-425. See also: <http://www.cs.bham.ac.uk/~xin/>
44. Hancock P. Recombination Operators for the Design of the Neural Nets by Genetic Algorithm // Parallel Problem Solving from Nature (PPSN-II). North-Holland, 1992. pp. 441-450.
45. Fiszlewe A., Britos P., Perichisky G., Garcia-Martinez R. Automatic Generation of Neural Networks based on Genetic Algorithms // Revista Eletronica de Sistemas de Informacao, 2003, Vol. 2, no. 1. См. также: <http://citeseer.ist.psu.edu/>
46. Kitano H. Designing neural network using genetic algorithm with graph generation system // Complex Systems, 1990, no. 4, pp. 461-476.
47. Хомич А.В., Жуков Л.А. Эволюционный метод оптимизации структуры нейронной сети с учителем // Научная сессия МИФИ-2005. VII Всероссийская научно-практическая конференция "Нейроинформатика-2005": Сборник научных трудов. В 2-х частях. Часть 1. М: МИФИ, 2005. С. 11-18.
48. Caruana R.A., de Sa V.R. Benefiting from the variables that variable selection discards // Journal of Machine Learning Research, 2003, Vol. 3, pp. 1245-1264.
49. Царегородцев В.Г. Редукция размеров нейросети не приводит к повышению обобщающих способностей // Материалы XII Всероссийского семинара «Нейроинформатика и ее приложения», Красноярск, 2004. С. 163-165. См. также: <http://www.neuropro.ru/>
50. Igel C., Kreutz M. Operator adaptation in evolutionary computation and its application to structure optimization of neural networks // Neurocomputing, 2003, no. 55(1-2), pp. 347-361. See also: <http://www.neuroinformatik.ruhr-uni-bochum.de/PEOPLE/igel/publications.html>
51. Хомич А.В., Степанян И.В., Карпищук А.В. Диагностика хронического пылевого бронхита по данным акустической спирометрии с применением блочных нейронных сетей // Информационные процессы, 2005, Том 5, № 5, с. 405-413.
52. Цой Ю.Р., Спицын В.Г. Применение генетического алгоритма для решения задачи адаптивного нейроуправления // Научная сессия МИФИ-2005. VII Всероссийская научно-практическая конференция "Нейроинформатика-2005": Сборник научных

- трудов. В 2-х частях. Часть 1. М.:МИФИ, 2005. С. 35-43. См. также: <http://qai.narod.ru/Publications/>
53. Balakrishnan K., Honavar V. Evolutionary Design of Neural Architectures – A Preliminary Taxonomy and Guide to Literature. Technical Report CS TR #95-01. Iowa State University, 1995. See also: <http://www.cs.iastate.edu/~gannadm/homepage.html>
 54. Law D., Miikkulainen R. Grounding Robotic Control with Genetic Neural Networks. Technical Report No. AI94-223. The University of Texas at Austin, 1994. See also: <http://www.nn.cs.utexas.edu/>
 55. Derakhshani R. GETnet: A General Framework for Evolutionary Temporal Neural Networks // Proceedings of International Joint Conference on Neural Networks. Montreal, Canada, 2005. pp. 3150-3155.
 56. Fullmer B., Miikkulainen R. Using Marker-Based Genetic Encoding of Neural Networks to Evolve Finite State Behavior // In Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life (ECAL-91). Paris, 1991. pp.255-262. See also: <http://www.nn.cs.utexas.edu/>
 57. Gruau F. Genetic synthesis of Boolean neural networks with a cell rewriting developmental process // In Proceedings of the International Workshop on Combination of Genetic Algorithms and Neural Networks (COGANN-92). Los Alamos, CA: IEEE Computer Society Press, 1992. pp. 55-74. See also: <http://www.cs.colostate.edu/~genitor/>
 58. Шукович Г. Применение генетических алгоритмов и систем генерирующих графов для создания модулярных нейросетей // Программирование, 2002, № 1, с. 13-20.
 59. Ferreira C. Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence. Angra do Heroismo, Portugal, 2002. See also: <http://gene-expression-programming.com/GepBook/Introduction.htm>
 60. Whitley D. Genetic Algorithms and Neural Networks / Winter, Periaux, Galan and Cuesta (Eds.) Genetic Algorithms in Engineering and Computer Science. John Wiley, 1995. pp. 203-216. See also: <http://www.cs.colostate.edu/~genitor/>
 61. Palmer C.C., Kershbaum A. Computer science: representing trees in genetic algorithms // Back T., Fogel D., Michalewicz Z. (Eds.) Handbook of evolutionary computation. Bristol, New York: Institute of Physics Publishing and Oxford University Press, 1997. pp. G1.3:1 – G1.3:8.
 62. Скобцов В.Ю., Скобцов Ю.А. Модификации генетического программирования // Труды Международных научно-технических конференций «Интеллектуальные системы (IEEE AIS'04)» и «Интеллектуальные САПР» (CAD-2004). Научное издание в 3-х томах. Т.1. М.: Физматлит, 2004. С. 78-86.
 63. Kodjabachian J., Meyer J.A. Evolution and development of modular control architectures for 1-D locomotion in six-legged animats // Connection Science, 1998, Vol. 18, pp. 211-237. See also: <http://animatlab.lip6.fr/index.en.html>
 64. Fahlman S.E., Lebiere C. The cascade-correlation learning architecture / Touretzky D. S. (Ed.) Advances in Neural Information Processing Systems 2. San Mateo, CA: Morgan Kaufman Publishers. P. 524-532.
 65. Littmann E., Ritter H. Cascade network architectures // Proceedings of International Joint Conference on Neural Networks, 1992, Vol. 2. pp. 398-404.
 66. LeCun Y., Denker J.S., Solla S.A. Optimal brain damage / In Touretzky D. S. (Ed.) Advances in Neural Information Processing Systems 2. San Mateo, CA: Morgan Kaufman Publishers, 1990. pp. 598-605. See also: <http://yann.lecun.com/>
 67. Hassibi B., Stork D.G. Second order derivatives for network pruning: Optimal brain surgeon / In Hanson S.J., Cowan J.D., Giles C.L. (Eds.) Advances in Neural Information Processing Systems 5. San Mateo, CA: Morgan Kaufman Publishers, 1993. P. 164-171.
 68. Ragg T., Braun H., Landsberg H. A Comparative Study of Neural Network Optimization Techniques. // In Proceedings of the ICANNGA 97. Springer-Verlag, 1997.

69. Yao X. Evolutionary artificial neural networks // *International Journal of Neural Systems*, 1993, vol. 4, no. 3, pp. 203–222. See also: <http://www.cs.bham.ac.uk/~xin/>
70. Gruau F. Neural network synthesis using cellular encoding and the genetic algorithm. Unpublished PhD thesis. Lyon: l'Universite Claude Bernard, 1994. See also: <http://citeseer.ist.psu.edu/>
71. Chen C.-C., Miikkulainen R. Creating Melodies with Evolving Recurrent Neural Networks // *Proceedings of the 2001 International Joint Conference on Neural Networks (IJCNN-01)*. Piscataway, NJ: IEEE, 2001. P. 2241-2246. See also: <http://www.nn.cs.utexas.edu/>
72. James D., Tucker P. A Comparative Analysis of Simplification and Complexification in the Evolution of Neural Network Topologies // *Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2004)*. New York, NY: Springer-Verlag, 2004. See also: <http://citeseer.ist.psu.edu/>
73. Паклин Н. Обучаем нейронную сеть генетическим алгоритмом. 2003. См. также: http://paklin.newmail.ru/mater/gene_net.html
74. Stanley K.O., Miikkulainen R. Evolving a roving eye for go // *In Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2004)*. New York, NY: Springer-Verlag, 2004. See also: <http://www.nn.cs.utexas.edu/>
75. Stanley K.O., Bryant B.D., Miikkulainen R. The NERO real-time video game. Technical report AI-TR-04-312. The University of Texas at Austin, 2004. See also: <http://www.nn.cs.utexas.edu/>
76. Pan Z., Kang L., Nie S. Evolving Both the Topology and Weights of Neural Networks // *Parallel Algorithms and Applications*, 1996, no. 9, pp. 299-307. See also: <http://citeseer.ist.psu.edu/>
77. Lubberts A., Miikkulainen R. Co-Evolving a Go-Playing Neural Network // *Proceedings of Coevolution: Turning Adaptive Algorithms upon Themselves, Birds-of-a-Feather Workshop, Genetic and Evolutionary Computation Conference (GECCO-2001)*. San Francisco, CA: Kaufmann, 2001. pp. 14-19. See also: <http://www.nn.cs.utexas.edu/>
78. Polani D., Miikkulainen R. Eugenic Neuro-Evolution for Reinforcement Learning // *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*. San Francisco, CA: Kaufmann, 2000. pp. 1041-1046. See also: <http://www.nn.cs.utexas.edu/>
79. Potter M. A., De Jong K. A. Cooperative coevolution: An architecture for evolving co-adapted subcomponents // *Evolutionary Computation*, 2000, no. 8, pp. 1–29. See also: <http://www.cs.gmu.edu/~eclab/>
80. Goldberg D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
81. Holland J.H. Building Blocks, Cohort Genetic Algorithms, and Hyperplane-Defined Functions // *Evolutionary Computation*, 2000, no. 4(8), pp. 373-391.
82. Barnett L. Evolutionary Search on Fitness Landscapes with Neutral Networks. Unpublished PhD thesis. The University of Sussex, England, 2003. See also: <http://www.cogs.susx.ac.uk/users/lionelb/>
83. Igel C. Neuroevolution for Reinforcement Learning Using Evolution Strategies // *Proceedings of Congress on Evolutionary Computation 2003 (CEC 2003)*, Vol. 4. IEEE Press, 2003. pp. 2588-2595. See also: <http://www.neuroinformatik.ruhr-uni-bochum.de/PEOPLE/igel/publications.html>
84. Цой Ю.Р., Спицын В.Г., Чернявский А.В. Нейроэволюционное улучшение качества изображений // *Научная сессия МИФИ-2006. VIII Всероссийская научно-практическая конференция "Нейроинформатика-2006": Сборник научных трудов*. В печати.
85. Sutton R.S., Barto A.G. *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press, 1998. See also: <http://www-anw.cs.umass.edu/~rich/book/the-book.html>

Эволюционный подход к настройке и обучению искусственных нейронных сетей

86. Ackley D., Littman M. Interactions between learning and evolution // Artificial Life II. Reading, MA: Addison-Wesley, 1992. pp. 487-509.
87. Artificial Life: The Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems / Langton C.G. (Ed.) Redwood City, CA: Addison-Wesley. 1992.
88. Meyer J.-A., Guillot A. Simulation of Adaptive Behavior in Animats: Review and Prospects / In Meyer J.-A., Wilson S. W. (Eds.) Proceedings of the First International Conference on Simulation of Adaptive Behavior: from animals to animats. Cambridge, MA: MIT Press, 1991. pp. 2-14.
See also: <http://www.poleia.lip6.fr/ANIMATLAB/#Publications>.
89. Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. М.: Эдиториал УРСС, 2002. 352 с.
90. Нейроинформатика / А.Н. Горбань, В.Л. Дунин-Барковский, А.Н. Кирдин и др. – Новосибирск: Наука. Сибирское предприятие РАН, 1998.
См. также: <http://www.neuroschool.narod.ru/books/neurinf.html>

Статья поступила 20 декабря 2005 г.

После доработки 11 января 2006 г.