

Глава 3

Лекция 5. Нейроэволюционные алгоритмы

Оглавление

3	Лекция 5. Нейроэволюционные алгоритмы	1
5.1	Задачи, решаемые с помощью НЭ подхода	2
5.2	Преимущества от НЭ алгоритмов	5
5.3	Недостатки НЭ алгоритмов	5
5.4	Прямое кодирование	6
5.4.1	Веса связей	6
5.4.2	Структура	8
5.4.3	Веса и структура	11

5.1 Задачи, решаемые с помощью НЭ подхода

Помимо решения «стандартных» нейросетевых задач классификации и аппроксимации с использованием обучающего множества данных НЭ алгоритмы представляют большой интерес для решения трудноформализуемых задач, где присутствует сложность формирования обучающего множества и/или затруднена оценка качества работы ИНС. Рассмотрим коротко примеры таких задач и некоторые особенности использования НЭ подхода:

Адаптивное управление При использовании НЭ алгоритмов для решения задач адаптивного управления можно перейти к безмодельному обучению, когда не нужно знать прямую либо обратную математическую модель объекта управления (ОУ), а ИНС оценивается в зависимости от используемого ОУ. Например, в [1] для обучения ИНС управлению инерционным колебательным звеном второго порядка с передаточной функцией вида $W = \frac{k}{T^2s^2 + 2T\zeta s + 1}$ использовалась следующая оценка:

$$F = \frac{1}{3} \sum_{i=1}^3 \frac{1}{t_{max}} \int_0^{t_{max}} (x(t) - u_r(t))^2 dt$$

где $x(t)$ и $u_r(t)$ – соответственно выход ОУ и сигнал задания в момент времени t ; t_{max} – время интегрирования переходного сигнала в системе. В [19, 25] при решении задачи балансирования на тележке 1 либо 2 шестов разной длины использовалась следующая оценка ИНС:

$$F = t,$$

где t – количество итераций, в течение которых удавалось удержать маятники от падения. В случае использования 2 маятников без информации о скорости их движения, когда появляется дополнительное требование подавления колебательных движений

тележки, в [11, 25] оценка ИНС определяется уже следующим образом (обозначения из [25]):

$$F = 0.1f_1 + 0.9f_2,$$

$$f_1 = t/1000,$$

$$f_2 = \begin{cases} 0, & t < 100 \\ \frac{0.75}{\sum_{i=t-100}^t (|x^i| + |\dot{x}^i| + |\theta_1^i| + |\dot{\theta}_1^i|)}, & t \geq 100 \end{cases},$$

где x^i и \dot{x}^i – положение и скорость тележки, θ^i и $\dot{\theta}^i$ – положение и угловая скорость первого маятника.

Адаптивное поведение и многоагентные системы Здесь большой интерес представляет возможность реализации интеллектуальных агентов [3] с использованием ИНС, при этом возможно наличие коллективного поведения [24]. В качестве примера рассмотрим проект NERO (Neuro-Evolution of Robotic Operatives¹ [24]. Ключевой технологией проекта является модифицированный алгоритм NEAT для одновременной настройки структуры и весов связей ИНС, разработанный Кеннетом Стэнли (*Kenneth Stanley*). Пользователь сначала тренирует на специальном тренировочном полигоне команду НС агентов, а затем его команда «соребнуется» с агентами компьютера или другого пользователя на своеобразном «поле боя». Поведение агентов 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 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, которая создается на тренировочном полигоне.

¹http://dmc.ic2.org/nero_public

Эволюционная робототехника (*Evolutionary robotics*) Представляет сравнительно молодое направление исследований, направленное на создание и исследование автономных роботов, которые рассматриваются как искусственные организмы, поведение которых развивается без человеческого вмешательства в результате взаимодействия с окружающей средой. Эволюционная робототехника во многом опирается на науки о природе, биологию и этологию, и использует такие концепции и подходы, как нейронные сети, генетические алгоритмы, динамические системы и бионика. Основное отличие исследований в области эволюционной робототехники от исследований адаптивного поведения интеллектуальных агентов заключается в использовании реальных роботов, а не их программных моделей. Это ограничивает условия проведения экспериментов (нельзя выйти за рамки действующих физических законов и технических и материальных ограничений), но добавляет значительно больше реализма за счет необходимости учета инерции, помех, характеристик датчиков, микроконтроллеров и материалов и т.д., что представляет интерес с практической точки зрения. Для управления роботами могут использоваться разнообразные модели ИНС и нейронов. Например, в [10] рассматривается обучение микроробота Alice [5] навигации с избеганием препятствий с использованием эволюционной настройки структуры ИНС с упрощенной моделью спайкового нейрона (далее используются оригинальные обозначения из [10]):

$$o_i^t \begin{cases} 1 \text{ and } v_i^t = v_i^{\min}, & v_i^t > v_i^{\max} + r^t, \\ 0, & v_i^t \leq v_i^{\max} + r^t, \end{cases}$$

где o_i^t – выходной сигнал; v_i^t, v_i^{\min} и v_i^{\max} – соответственно текущий, мини-мальный и максимальный мембранный потенциал нейрона; r^t – случайное целое, необходимое для предотвращения осцилляций и «блокировок» в ИНС с обратными связями. Значение мембранного потенциала v_i^t накапливается с течением времени с учетом входных сигналов нейрона и постоянной утечки. Оценка ИНС определяется в зависимости от скорости движения робота и расстояния до препятствий, которые необходимо максимизировать:

$$F = \sum_t V^t (1 - \Delta V^t) (1 - i),$$

где V^t – суммарная скорость вращения колес робота; ΔV^t – модуль разности скорости вращения левого и правого колеса; i – максимальная текущая активность одного из инфракрасных датчиков препятствий.

Поиск игровых стратегий Данное направление исследований рассматривает использование ИНС для принятия решений в настольных играх. Часто рассматриваются крестики-нолики [14] и го [17, 26]. Оценка работы ИНС затрудняется большим количеством различных вариантов развития событий и связанной с этим неопределенностью, что делает саму оценку неточной.

В [26] показано, что при использовании НЭ подхода возможно успешное обучение ИНС игре в го против программы с детерминированным алгоритмом, реализованным в программе Gnugo², на поле 5x5 на основе оценки

$$F = 100 - \left(\frac{2 \sum_{i=1}^n e_i}{n} = e_f \right),$$

²<http://www.gnu.org/software/gnugo.html>

где e_i – счет в партии после i -го хода, n – число ходов в партии, e_f – финальный счет партии. Также отметим вариант с «автообучением», когда настройка ИНС осуществляется путем соревнования одной эволюционирующей популяции ИНС с другой [17] (т.н. конкурентная коэволюция (*competitive coevolution*)). При этом оценка ИНС формируется в результате серии матчей с ИНС из «противоборствующей» популяции:

$$F = \sum_{i \in O} \frac{1}{L_i},$$

где O – множество оппонентов, побежденных данной ИНС, а L_i – число проигрышей i -й побежденной ИНС. НЭ обучение направлено на минимизацию оценки F .

Компьютерное творчество Применение ЭА алгоритмов для компьютерного синтеза изображений [23], звуков и мелодий [7], геометрических форм [20, 23] и т.д. возможно благодаря использованию концепции эстетической селекции (*aesthetic selection*, называемой также *aesthetic evolution*, *interactive evolution*) [18]. Особенностью этой концепции является интерактивное взаимодействие ЭА с пользователем, который «вручную» определяет какие решения (изображения, звуки и т.д.) в популяции являются более приемлемыми, и, следовательно, какие особи будут допущены к скрещиванию. Это существенно увеличивает время эволюционного обучения и накладывает значительные ограничения на параметры ЭА и получаемых решений (нельзя использовать популяцию большого размера; решения должны быть такими, чтобы их можно было оценить за сравнительно небольшой промежуток времени и др. [18]), однако позволяет решать поставленную задачу, избежав очевидных и неразрешенных в настоящее время трудностей с формализацией критериев оценки решений.

Например, в [23] описан способ генерации разнообразных геометрических образов и узоров, где ИНС реализует преобразование $L = g(x_1, x_2, d)$, здесь L – нормированная яркость пикселя с прямоугольными координатами $(x_1; x_2)$, d – расстояние до центра изображения. Таким образом, ИНС используется для «рисования», при этом яркость пикселей зависит от их координат. Отбор ИНС по всей популяции осуществляется пользователем путем выбора понравившихся ему соответствующих образов. Пример эволюции образа, названного «космический корабль» («spaceship»), сгенерированного ИНС в различных поколениях эволюционного поиска, показан на рис. 5.1.

5.2 Преимущества от НЭ алгоритмов

- Нет необходимости в обучающей выборке
- Гибкость
- Возможность интерактивной работы
- Очень широкая область применения

5.3 Недостатки НЭ алгоритмов

- Большое влияние случайного фактора

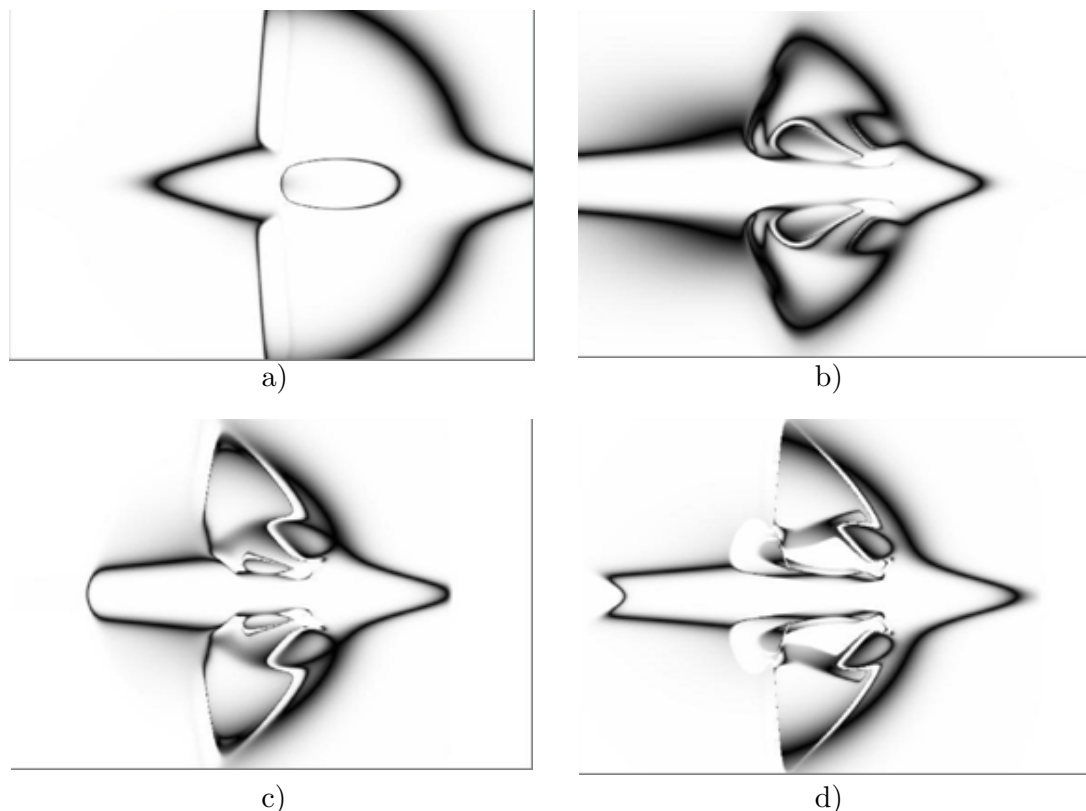


Рис. 5.1: Пример эволюции образа «космический корабль», сгенерированного ИНС в различных поколениях: а) начальный образ; б) появление узоров на «крыльях»; с) изменение формы «хвоста»; д) появление «выхлопов» за «крыльями»

- Сложность разработки и настройки
- Сложность анализа полученного результата
- Слабо проработанная теория

5.4 Прямое кодирование

5.4.1 Веса связей

Существенными преимуществами использования 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.

При использовании ЭА для настройки весов связей ИНС достаточно использовать оценку, отражающую качество нейросетевого решения в целом. Здесь важна *адекватность* оценки, то есть 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 такая 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. При этом появляется возможность использовать приблизительные оценки функционирования ИНС, и оцениваться может не каждый выходной сигнал ИНС, а последовательность сигналов. Это позволяет расширить круг практических приложений нейроинформатики, а также упростить процесс решения многих задач, связанных с адаптивным нейроуправлением, адаптивным поведением и моделированием.

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.

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. При использовании вещественного кодирования эта проблема не так актуальна, но не исчезает полностью из-за необходимости адаптивной настройки параметров операторов ЭА, так как характеристики эволюционного поиска изменяются с течением времени.

Проблема конкурирующих решений (*competing conventions problem*) [28] известная также как проблема перестановки (*permutation problem*) [21] связана с кодированием весов связей ИНС. Ее суть заключается в том, что генетическое представление допускает существование нескольких вариантов хромосом, кодирующих одну и ту же ИНС (см. пример на рис. 5.2).

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

1. Упорядочивание списка весов связей кодируемых ИНС (выравнивание хромосом) по значениям весов [27], либо с использованием специальных меток [25]. Считается, что такие меры позволят уменьшить вероятность появления особей, представляющих одинаковые ИНС. Также возможно раннее удаление особей-дубликатов на этапе формирования популяции следующего поколения.

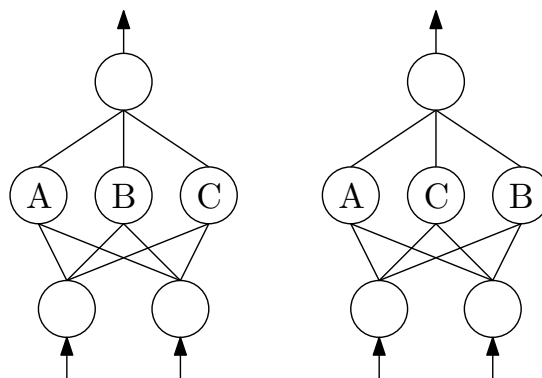


Рис. 5.2: Проблема перестановки. Две одинаковые ИНС могут быть закодированы различным образом «перестановкой» скрытых нейронов. Считается, что «скрещивание» таких сетей практически бесполезно [25]

2. Модификация компонент ЭА, включающая разработку адаптивных операторов скрещивания [21], уменьшение размера популяции, а также повышение вероятности мутации.
3. Уменьшение роли оператора кроссинговера, либо полный отказ от его использования [4, 28, 29]. При этом предполагается, что: (а) вероятность возникновения одинаковых ИНС в результате мутаций различных особей незначительна; (б) в случае, если такие особи все-таки появились, мутации приведут к меньшим «разрушениям», чем скрещивание.

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 [19].

Вектор весов связей

Пример приведен на рис. 5.3.

Коэволюционный подход

В алгоритме Enforced Sub-Population (ESP) [11] веса связей разбиваются на непересекающиеся группы (декомпозиция задачи). В i -ю группу попадают веса входных связей i -го нейрона. Каждая группа весов оптимизируется независимо в своей подпопуляции (рис. 5.4). Как правило в алгоритме ESP рассматриваются ИНС с обратными связями, хотя подобный подход можно применить и к ИНС прямого распространения.

5.4.2 Структура

Еще одной типичной задачей НЭ подхода является задача эволюционной настройки структуры ИНС [9, 15]. В хромосоме кодируется топология ИНС, а настройка весов осуществляется, например, с использованием градиентных алгоритмов. Каждая особь, представляющая сеть той или иной структуры, оценивается в зависимости от результатов обучения с использованием градиентных либо иных алгоритмов:

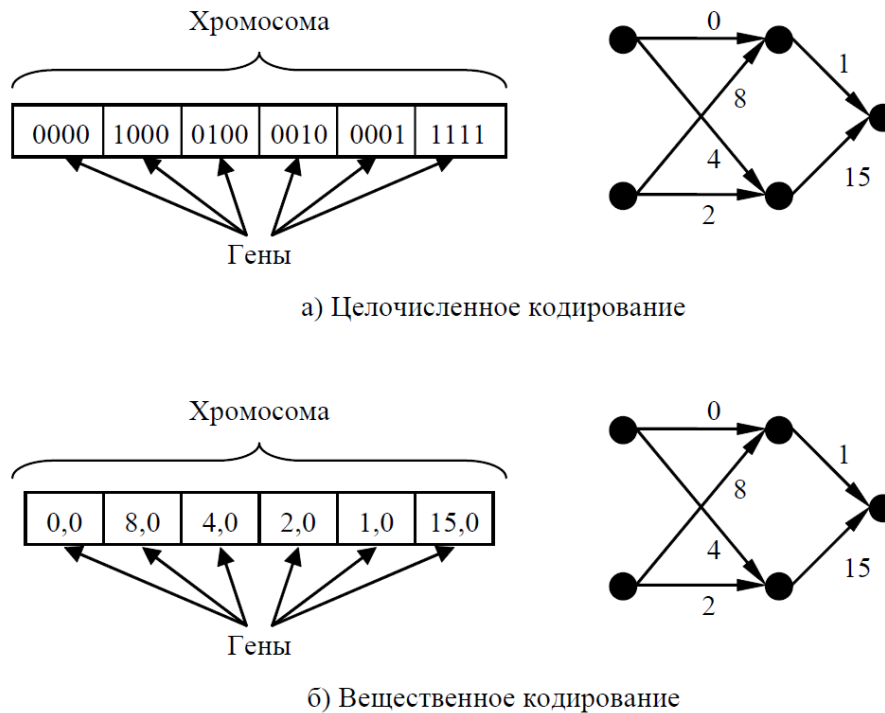


Рис. 5.3: Простые примеры целочисленного (а) и вещественного (б) кодирования весов связей для двух одинаковых ИНС

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. Из ранних работ в этом направлении отметим исследования А.Б. Глаза и Л.А. Растригина по эволюционной оптимизации структуры многослойного перцептрона [2].

Перечислим преимущества и недостатки эволюционной настройки структуры НС. 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 традиционных градиентных алгоритмов обучения к начальным условиям и значениям параметров алгоритма обучения. Таким образом, хромосоме, представляющей ИНС с некоторой

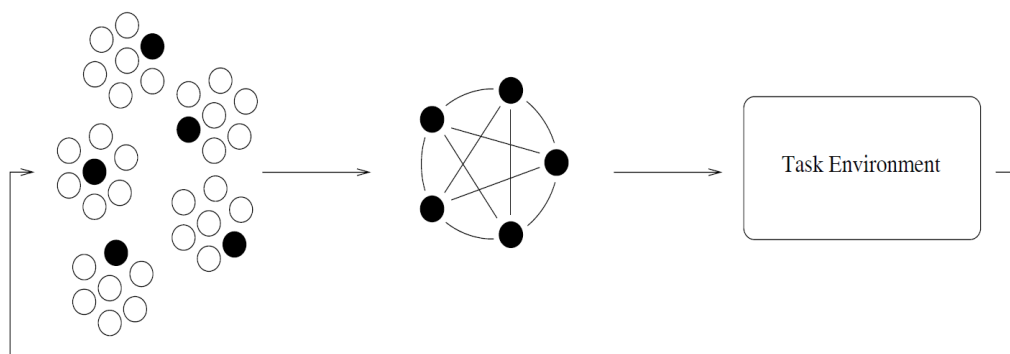


Рис. 5.4: Схема коэволюционного кодирования информации о весах ИНС.

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 [9], 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, так как многое зависит от результатов обучения [6].

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 весов связей ИНС (см. п. 5.4.3), поэтому многие дальнейшие замечания относятся в равной степени и к эволюционному поиску структуры ИНС, и к одновременной настройке структуры и весов связей. Для изменения структуры ИНС часто применяются следующие операции:

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

Встречаются также модификации этих операций, включающие, например, «расщепление» существующего нейрона, «перенаправление» связи [19, 29] 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. Известны следующие варианты решения проблемы по выбору операции модификации структуры ИНС:

- определение эффективности каждой операции преобразования топологии ИНС на основании оценки качества получаемых НС решений [13];
- использование дополнительной информации об ИНС для ее модификации (например, вычисление «значимости» связей [29], эвристическая оценка структуры ИНС [4]).

Заметим, что существуют неэволюционные подходы к настройке топологии ИНС либо во время ее обучения (алгоритм каскадной корреляции Фальмана [8] и его модификации и др.), либо по завершении процесса обучения (алгоритмы Optimal Brain

Damage (OBD) [16], Optimal Brain Surgeon (OBS) [12] и др.). Несмотря на то, что в [22] показана эффективность использования эволюционного подхода для настройки топологии ИНС по сравнению с алгоритмами OBD и OBS, однако детального исследования на эту тему, насколько известно автору, не проводилось.

Матрица смежности

Упрощенный вариант для ИНС без обратных связей

5.4.3 Веса и структура

Использование эволюционного подхода позволяет одновременно настраивать веса связей и структуру ИНС. 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 различных 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 [4, 13, 19];
- ограничение количества связей [19];
- ограничение изменения топологии ИНС (рассматриваются только растущие ИНС [?], либо только многослойные ИНС и др.);

Следствием первых двух ограничений часто является фиксация длины хромосомы, что существенно уменьшает пространство возможных решений и не способствует эффективному поиску структуры ИНС, но существенно упрощает задачу и позволяет использовать «стандартные» операторы скрещивания и мутации. Для повышения гибкости поиска НЭ алгоритма используют генетическое кодирование с переменной длиной хромосомы, которое позволяет добавлять и (в некоторых реализациях) удалять гены в процессе работы алгоритма. Это ведет к изменению общего порядка следования генов и требует реализации специализированных генетических операторов, учитывающих особенности используемого кодирования.

Одновременное решение двух отдельных задач: настройки весов связей и структуры ИНС – позволяет в некоторой степени скомпенсировать недостатки, 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, на основе только критерия оценки ИНС для осуществления эволюционного поиска. При этом, как и для эволюционного обучения ИНС (см. п. 5.4.1), 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, 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.

Коэволюционный подход

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 [19]. В хромосоме особи записывается информация о входных и выходных связях одного нейрона. Во время оценки популяции формируется к ИНС, включающих нейроны, соответствующих различным особям (рис. 5.6), и производится их оценка. Оценка каждой особи вычисляется как средняя оценка ИНС, в состав которых входил нейрон, соответствующий этой особи. Коэволюционный подход также применяется и для эволюционной настройки весов связей ИНС [26, 27].

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 [19] 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.

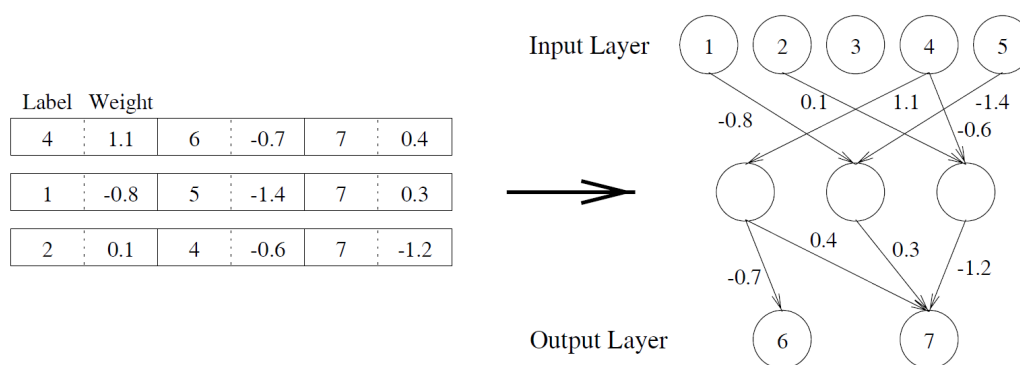


Рис. 5.5: Пример использования коэволюционного подхода к кодированию информации об ИНС для оценки особей. Черными кругами в популяции обозначены особи, соответствующие нейронам, включенным в структуру оцениваемой ИНС. Для определения приспособленности особей в популяции производится оценивание к различным ИНС. использован рисунок из [19].

Раздельное кодирование

NEAT & NEvA

AGE

Матрица смежности

Список ребер

Список узлов

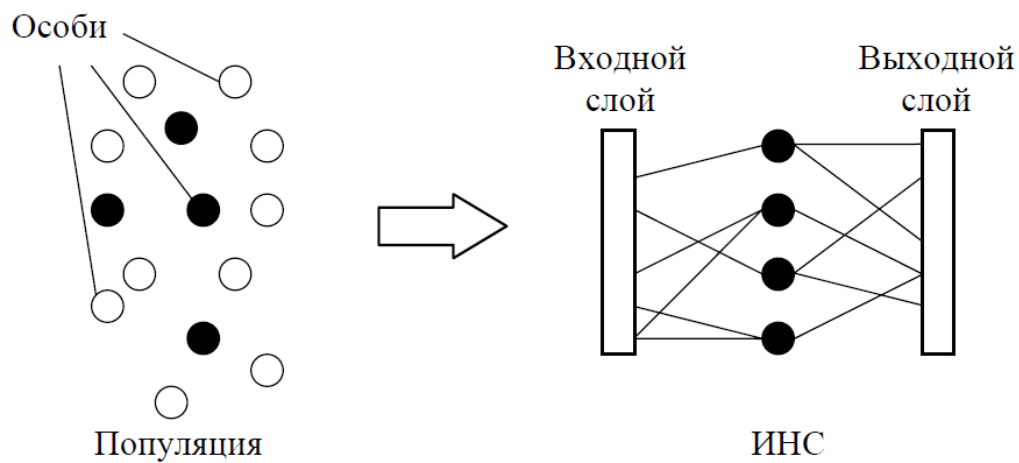


Рис. 5.6: Пример использования коэволюционного подхода к кодированию информации об ИНС для оценки особей. Черными кругами в популяции обозначены особи, соответствующие нейронам, включенным в структуру оцениваемой ИНС. Для определения приспособленности особей в популяции производится оценивание к различным ИНС. использован рисунок из [?].

Литература

- [1] Г.К. Вороновский, К.В. Махотило, С.Н. Петрашев, and Сергеев С.А. *Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности*. Харьков: Основа, 1997.
- [2] А.Б. Глаз and Л.А. Растрин. *Задачи статистической оптимизации*, chapter Оценка вероятности образования оптимальной структуры перцептрона при ее оптимизации методами случайного поиска, pages 131–142. Рига: Зинатне, 1971.
- [3] В.Б. Тарасов. *От многоагентных систем к интеллектуальным организациям: философия, психология, информатика*. М.: Эдиториал УРСС, 2002.
- [4] P.J. Angeline, G.M. Saunders, and J.B. Pollack. An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, (5):54–65, 1993.
- [5] G. Caprari and and Siegwart R. Estier, T. Fascination of downscaling – alice the sugarcube robot. *Journal of Micro-Mechatronics*, 3(1):177–189, 2002.
- [6] R.A. Caruana and V.R. de Sa. Benefiting from the variables that variable selection discards. *Journal of Machine Learning Research*, 3:1245–1264, 2003.
- [7] C.-C.J. Chen and R. Miikkulainen. Creating melodies with evolving recurrent neural networks. In *Proceedings of the 2001 International Joint Conference on Neural Networks (IJCNN-01)*. IEEE Press, 2001.
- [8] S.E. Fahlman and C. Lebiere. The cascade-correlation learning architecture. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 524–532. San Mateo, CA: Morgan Kaufman Publishers, 1990.
- [9] and Britos P. Fiszlelew, A., G. Perichisky, and R. Garcia-Martinez. Automatic generation of neural networks based on genetic algorithms. *Revista Eletronica de Sistemas de Informacao*, 2(1), 2003.
- [10] D. Floreano, Y. Epars, J.C. Zufferey, and C. Mattiussi. Evolution of spiking neural circuits in autonomous mobile robots. *International Journal of Intelligent Systems*, 21(9):1005–1024, 2006.
- [11] F. Gomez and R. Miikkulainen. Solving non-markovian control tasks with neuroevolution. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1356–1361. San Francisco, CA,, 1999.

- [12] B. Hassibi and D.G. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 164–171. San Mateo, CA: Morgan Kaufman Publishers, 1993.
- [13] C. Igel and M. Kreutz. Operator adaptation in evolutionary computation and its application to structure optimization of neural networks. *Neurocomputing*, 55(1-2):347–361, 2003.
- [14] D. James and P. Tucker. A comparative analysis of simplification and complexification in the evolution of neural network topologies. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2004)*. New York, NY: Springer-Verlag, 2004.
- [15] H. Kitano. Designing neural network using genetic algorithm with graph generation system. *Complex Systems*, (4):461–476, 1990.
- [16] Y. LeCun, J.S. Denker, and Solla S.A. Optimal brain damage. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 598–605. San Mateo, CA: Morgan Kaufman Publishers, 1990.
- [17] A. Lubberts and R. Miikkulainen. Co-evolving a go-playing neural network. In *Proceedings of Coevolution: Turning Adaptive Algorithms upon Themselves, Birds-of-a-Feather Workshop, Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 14–19. San Francisco, CA: Kaufmann, 2001.
- [18] J. McCormack. New challenges for evolutionary music and art. *SIGEVolution newsletter*, 1(1):5–11, 2006.
- [19] D.E. Moriarty and R. Miikkulainen. Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, (22):11–32, 1996.
- [20] T.C. Nguyen and T.S. Huang. *Advances in Genetic Programming*, chapter Evolvable 3D modeling for model-based object recognition systems, pages 459–475. Cambridge, MA: MIT Press, 1994.
- [21] N. Radcliffe. Genetic set recombination and its application to neural network topology optimization. *Neural Computing and Applications*, 1:67–90, 1993.
- [22] T. Ragg, H. Braun, and H. Landsberg. A comparative study of neural network optimization techniques. In *Proceedings of the ICANNGA 97*. Berlin Heidelberg: Springer-Verlag, 1997.
- [23] K.O. Stanley. Patterns without development, 2006. Technical report CS-TR-06-01, University of Central Florida, 2006.
- [24] K.O. Stanley, B.D. Bryant, and R. Miikkulainen. The nero real-time video game. Technical report AI-TR-04-312, The University of Texas at Austin, 2004.
- [25] K.O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

- [26] K.O. Stanley and R. Miikkulainen. Evolving a roving eye for go. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2004)*. New York, NY: Springer-Verlag, 2004.
- [27] D. Thierens. Non-redundant genetic coding for neural networks. Technical report UU-CS-1998-46, Utrecht University, Netherlands, 1998.
- [28] D. Whitley, T. Starkweather, and C. Bogart. Genetic algorithms and neural networks: Optimizing connections and connectivity. *Parallel Computing*, 14:341–361, 1990.
- [29] X. Yao and Y. Liu. Making use of population information in evolutionary artificial neural networks. *IEEE Transactions on systems, man, and cybernetics - part B: Cybernetics*, 28(3):417–425, 1998.