УДК 004.021
ББК 32.973
Л92

Лэнхэм М.
Л92 Эволюционное глубокое обучение / пер. с англ. А. В. Логунова. – М.: ДМК Пресс, 2023. – 440 с.: ил.

ISBN 978-5-93700-253-2

В книге проводится анализ методов усиления мощи нейронных сетей за счет биологических алгоритмов в решении задач поиска, оптимизации и управления. Книга знакомит с эволюционными вычислениями и предоставляет ряд методов, которые вы можете применять на протяжении всего процесса глубокого обучения. Вы откроете для себя генетические алгоритмы и эволюционно-вычислительные подходы к выведению нейросетевых топологий, генеративному моделированию, обучению с подкреплением и многому другому.

Издание предназначено для специалистов по анализу и обработке данных, хорошо знакомых с языком Python.

УДК 004.021
ББК 32.973

ISBN 978-1-6172-9952-0 (англ.)
ISBN 978-5-93700-253-2 (рус.)

Оглавление

Содержание

Предисловие

Когда более 25 лет назад я начинал свою карьеру в области машинного обучения и искусственного интеллекта, следующими крупными достижениями тогда считались две доминирующие технологии. Обе технологии показали многообещающие результаты в решении сложных задач, и обе были эквивалентны в вычислительном отношении. Этими двумя технологиями были эволюционные алгоритмы и нейронные сети (глубокое обучение).

В течение следующих двух десятилетий я был свидетелем резкого упадка эволюционных алгоритмов и взрывного роста глубокого обучения. Хотя эта битва велась и была выиграна глубоким обучением благодаря его вычислительной эффективности, оно также продемонстрировало множество новых приложений. С другой стороны, по большей части знания и использование эволюционных и генетических алгоритмов схлынули до сноски в публикациях.

Моя цель в этой книге – продемонстрировать способность эволюционных и генетических алгоритмов обеспечивать преимущества систем глубокого обучения. Эти преимущества особенно актуальны по мере того, как эпоха глубокого обучения переходит в эпоху автоматизированного машинного обучения (AutoML), в которой возможность автоматизировать крупномасштабную разработку моделей становится магистральной.

Помимо этого, считаю, что нашему поиску общего искусственного интеллекта может помочь обращение к эволюции. В конце концов, ведь именно эволюция является тем инструментом, который природа использовала для формирования нашего интеллекта. Так почему же она не может улучшить и искусственный интеллект? Полагаю, что мы слишком нетерпеливы и самонадеянны, чтобы думать, что человечество может решить эту задачу самостоятельно.

Написав эту книгу, я хотел продемонстрировать мощь эволюционных методов в дополнение к глубокому обучению как способу мышления, выходящему за рамки нормы. Надеюсь, что она продемонстрирует основы эволюционных методов увлекательным и инновационным способом, но в то же время позволит проникнуть на продвинутую территорию эволюционно выводимых нейросетей глубокого обучения (территорию так называемой NEAT) вплоть до инстинктивного (само)обучения. Инстинктивное (само)обучение – это мой взгляд на то, как нужно глядеть на процесс эволюции биологической жизни и отражать те же характеристики в наших поисках более интеллектуальных искусственных нейросетей.

Об этой книге

Эта книга знакомит читателей с эволюционными и генетическими алгоритмами, начиная с решения интересных задач машинного обучения и заканчивая сочетанием указанных концепций с глубоким обучением. Книга начинается с введения в симуляцию и концепций эволюции и генетических алгоритмов на языке Python. По мере изложения акцент смещается в сторону демонстрации их ценности применительно к глубокому обучению.

Кому следует прочитать эту книгу

Вы должны хорошо разбираться в языке Python и понимать стержневые концепции машинного обучения и науки о данных. Знания в области глубокого обучения будут необходимы для понимания концепций, описанных в последующих главах.

Как эта книга организована: дорожная карта

Эта книга состоит из трех частей: «*Начало работы*», «*Оптимизация глубокого обучения*» и «*Продвинутые применения*». В первой части мы начнем с рассмотрения основ симуляции, эволюции, генетических и других алгоритмов. Далее переходим к демонстрации различных применений эволюции и генетического поиска в рамках глубокого обучения. Затем завершаем книгу рассмотрением продвинутых приложений в генеративном моделировании, обучении с подкреплением и обобщенном интеллекте. Ниже приводится краткое изложение каждой главы.

Часть I «Начало работы»:

- *глава 1 «Введение в эволюционное глубокое обучение»* – в этой главе представлена концепция комбинирования эволюционных алгоритмов с глубоким обучением;
- *глава 2 «Введение в эволюционные вычисления»* – здесь дается базовое введение в вычислительную симуляцию и способы задействования эволюции;
- *глава 3 «Введение в генетические алгоритмы с помощью фреймворка DEAP»* – в данной главе рассматриваются концепции генетических алгоритмов и использование фреймворка DEAP;
- *глава 4 «Еще больше эволюционных вычислений с помощью фреймворка DEAP»* – тут разбираются интересные приложения генетических и эволюционных алгоритмов, от задачи о коммивояжере до генерирования изображений Моны Лизы.

Часть II «Оптимизация глубокого обучения»:

- *глава 5 «Автоматизация гиперпараметрической оптимизации»* – в этой главе демонстрируется несколько методов ги-

перпараметрической оптимизации в системах глубокого обучения с использованием генетических или эволюционных алгоритмов;

- *глава 6 «Нейроэволюционная оптимизация»* – в данной главе рассматривается оптимизация нейросетевой архитектуры систем глубокого обучения с использованием нейроэволюции;
- *глава 7 «Эволюционные сверточные нейронные сети»* – здесь рассматривается продвинутое применение оптимизации архитектуры сверточных нейронных сетей с использованием эволюции.

Часть III «Продвинутые применения»:

- *глава 8 «Эволюционное выведение автокодировщиков»* – в этой главе вводятся и рассматриваются основы генеративного моделирования с использованием автокодировщиков. Затем в ней демонстрируется развитие автокодировщиков в ходе эволюции;
- *глава 9 «Генеративное глубокое обучение и эволюция»* – эта глава подхватывает эстафету у предыдущей главы, продолжая тему рассмотрением генеративной состязательной сети и способов ее оптимизации с помощью эволюции;
- *глава 10 «NEAT: нейроэволюция расширяющих топологий»* – в данной главе рассказывается о методе NEAT и о способах его применения к различным приложениям базового уровня;
- *глава 11 «Эволюционное (само)обучение с помощью метода NEAT»* – здесь обсуждаются основы (само)обучения с подкреплением и глубокого (само)обучения с подкреплением, а затем демонстрируется использование метода NEAT для решения некоторых сложных задач из OpenAI Gym;
- *глава 12 «Эволюционное машинное обучение и за его пределами»* – в этой заключительной главе рассматривается будущее эволюционного машинного обучения и его способности дать представление об обобщенном искусственном интеллекте.

Хотя наша книга предназначена для прочтения от корки до корки, не у всех читателей, возможно, окажется достаточно времени, опыта или интереса, чтобы извлечь пользу из всего материала. Ниже приведено краткое руководство, которое поможет вам выбрать разделы или главы, на которых следует сосредоточиться:

- *часть I «Начало работы»* – обязательно прочтите всю эту часть, если вы – новичок в симуляциях и эволюционных или генетических вычислениях. Данный материал также может стать полезным обзором и продемонстрирует несколько интересных приложений;
- *часть II «Оптимизация глубокого обучения»* – прочтите эту часть либо отдельные главы из нее, если у вас есть реальная необходимость оптимизировать системы глубокого обучения с привлечением нейроэволюции или гиперпараметрической настройки;

- *часть III «Продвинутые приложения»* – главы здесь разбиты на три подчасти: эволюционное генеративное моделирование (главы 8 и 9), метод NEAT (главы 10 и 11) и инстинктивное (само) обучение (глава 12). Каждую из этих подчастей можно проштудировать независимо.

Об исходном коде

Весь исходный код этой книги был написан с использованием блокнотов Google Colab и находится в репозитории автора на GitHub: https://github.com/cxbxmxcx/EvolutionaryDeepLearning. Для выполнения исходного кода достаточно просто перейти в репозиторий GitHub в своем браузере и найти соответствующий образец исходного кода. Все примеры исходного кода были помечены префиксом номера главы, а затем номером примера, например `EDL_2_2_Simulating_Life.ipynb`. Находясь там, просто нажмите на значок Google Colab, чтобы запустить блокнот в Colab. Любые зависимости будут либо предустановлены в Colab, либо установлены как часть блокнота.

Эта книга содержит множество примеров исходного кода в виде отдельных нумерованных листингов и внутри обычного текста. В обоих случаях исходный код отформатирован `шрифтом фиксированной ширины, подобным этому`, чтобы отделять его от обычного текста. Иногда исходный код также выделяется **`жирным шрифтом`**, чтобы подчеркивать тот исходный код, который изменился по сравнению с предыдущими шагами в данной главе, например когда новая функциональная возможность добавляется в существующую строку исходного кода.

Во многих случаях изначальный исходный код был переформатирован; мы добавили переносы строк и переработали отступы, чтобы уместиться в доступное пространство страницы книги. В редких случаях даже этого было недостаточно, и листинги включали маркеры продолжения строки (➥). Вдобавок нередко комментарии в исходном коде из листингов удалялись, когда исходный код описывался в тексте. Многие листинги сопровождаются аннотациями к исходному коду, выделяющими важные концепции.

Вы можете получить исполняемые фрагменты исходного кода из онлайновой версии этой книги в liveBook по адресу https://livebook.manning.com/book/evolutionary-deep-learning. Полный исходный код примеров книги доступен для скачивания с веб-сайта издательства Manning по адресу https://www.manning.com/books/evolutionary-deep-learning и из репозитория на GitHub по адресу https://github.com/cxbxmxcx/EvolutionaryDeepLearning.

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не по-

нравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Manning Publications очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Об авторе

Майкл Лэнхэм – хорошо зарекомендовавший себя инноватор в области программного обеспечения и информационных технологий с 25-летним опытом работы. За это время он разработал широкий спектр программных приложений в таких областях, как игры, графика, веб, приложения для настольных компьютеров, инженерное дело, искусственный интеллект, ГИС и приложения машинного обучения для различных отраслей промышленности в качестве разработчика НИОКР. На рубеже тысячелетий Майкл начал работать с нейронными сетями и эволюционными алгоритмами в создании игр. Он использовал эти навыки и опыт работы в качестве архитектора ГИС и больших данных / предприятия для совершенствования инженерных и бизнес-приложений и применения в них различных походов из компьютерных игр. С конца 2016 года Майкл стал заядлым автором и ведущим, делясь своими знаниями с сообществом. В настоящее время он является автором многочисленных книг по дополненной реальности, звуковому дизайну, машинному обучению и искусственному интеллекту. Он известен своими познаниями во многих областях искусственного интеллекта и разработки программного обеспечения, но в настоящее время специализируется на генеративном моделировании, обучении с подкреплением и операциях машинного обучения. Майкл проживает со своей семьей в Калгари, Канада, и в настоящее время пишет, преподает и выступает с докладами об искусственном интеллекте, операционализации машинного обучения и конструировании программного обеспечения.

Часть I

Начало работы

Эволюционные и генетические алгоритмы существуют уже несколько десятилетий. С точки зрения вычислений эволюционные методы машинного обучения далеко не так эффективны, как глубокое обучение. Однако эволюционные методы способны предоставить нам уникальные инструменты, помогающие в широком спектре схем оптимизации, от гиперпараметрической настройки до нейросетевых архитектур. Но прежде чем мы обсудим эти схемы, необходимо познакомиться с эволюционными и генетическими алгоритмами.

В главе 1 мы знакомим с концепцией использования эволюционных методов для целей оптимизации систем глубокого обучения. Поскольку методы оптимизации глубокого обучения, которые мы рассматриваем в этой книге, относятся к автоматизированному машинному обучению, мы также вводим AutoML с эволюцией.

Затем в главе 2 мы знакомим с симуляцией жизни из книги Конвея «Игра в жизнь», используя простой сценарий, который позже эволюционно развивается с помощью генетических алгоритмов. Далее глава 3 знакомит с генетическими алгоритмами в различных формах с применением фреймворка распределенных генетических алгоритмов на Python (DEAP). Наконец, в главе 4 мы завершаем эту часть, представляя другие разнообразные формы эволюционных методов.

1 Введение в эволюционное глубокое обучение

Эта глава охватывает следующие ниже темы:

- понятие эволюционных вычислений и способы их интеграции в системы глубокого обучения;
- применение эволюционного глубокого обучения;
- устоявшиеся шаблоны оптимизации нейросетей глубокого обучения;
- роль автоматизированного машинного обучения в оптимизации нейросетей;
- применение эволюционных вычислительных методов для усиления разработки моделей глубокого обучения.

Глубокое обучение стало повсеместной технологией, которая больше всего ассоциируется с искусственным интеллектом и бурным развитием машинного обучения[1]. Из лженауки, которой его было принято считать (см. «Революция в области глубокого обучения» Теренса Дж.

[1] Термин *learning* означает процесс приобретения/усвоения знаний и выработки навыков в результате тренировки, где термин *train* (тренировать) подчеркивает итеративность процесса; термин *teach* (обучать) как передача знаний в спецлитературе по ИИ/МО практически не используется, т. к. речь идет именно о самообучении. Далее термины *обучение*, *(само) обучение* и *усвоение* используются в переводе взаимозаменяемо. – *Прим. перев.*

Сейновски, 2018, MIT Press)[1], оно превратилось в массовое применение для всего, что только можно, – от диагностики рака молочной железы до вождения автомобилей. В то время как многие считают его технологией будущего, другие придерживаются более прагматичного подхода к его растущей сложности и потребности в данных.

По мере того как глубокое обучение становится все сложнее, мы пичкаем его все большими объемами данных в надежде на какое-то грандиозное прозрение в определенной области. К сожалению, это случается редко, и слишком часто мы остаемся с плохими моделями, плохими результатами и сердитыми руководителями. Эта проблема будет продолжаться до тех пор, пока мы не разработаем эффективные процессы для наших систем глубокого обучения.

Процесс разработки эффективных и надежных систем глубокого обучения отражает – или должен отражать – процесс любого другого проекта машинного обучения или науки о данных. Хотя некоторые фазы могут отличаться по требуемым ресурсам и сложности, все шаги останутся прежними. Чего зачастую не хватает в относительно новом мире глубокого обучения, так это инструментария, который мог бы способствовать автоматизации некоторых из этих процессов.

И в этом как раз поможет *эволюционное глубокое обучение*[2]. Эволюционное глубокое обучение – это такой инструментарий, или набор шаблонов и практических приемов, который помогает автоматизировать разработку систем глубокого обучения. Используемый в этой книге термин *эволюционное глубокое обучение* охватывает широкий спектр эволюционных вычислительных методов и шаблонов, применяемых к различным аспектам систем глубокого обучения по всему процессу машинного обучения.

1.1 Что такое эволюционное глубокое обучение?

Термин *эволюционное глубокое обучение*, который впервые описан в этой книге, представляет собой общую систематизацию и группировку ряда методик, сочетающих эволюционные методы с глубоким обучением. Указанные методики могут использоваться для оптимизации систем глубокого обучения, начиная со сбора данных и заканчивая валидацией. Эволюционное глубокое обучение не ново; инструменты для комбинирования эволюционных методов с глубоким обучением получили множество названий, включая такие: глубокая нейронная эволюция (Deep Neural Evolution), эволюционно-нейронное автоматизированное машинное обучение (Evolution Neural AutoML), нейроэволюция, эволюционный искусственный интеллект и др.

---

[1] The Deep Learning Revolution, Terrence J. Sejnowski. – *Прим. перев.*

[2] Англ. evolutionary deep learning (EDL). – *Прим. перев.*

Эволюционное глубокое обучение возникло на стыке двух уникальных областей искусственного интеллекта – эволюционных вычислений[1] и применений глубокого обучения – с целью автоматизации и усовершенствования моделей. Сами по себе эволюционные вычисления представляют собой семейство методов, с помощью которых симулируются биологические или природные процессы, чтобы решать сложные задачи. Они, в свою очередь, могут применяться поверх глубокого обучения, чтобы автоматизировать и оптимизировать решения, при этом имея потенциал раскрывать новые стратегии и архитектуры.

Широкая категория методов, которые мы рассмотрим в рамках эволюционного глубокого обучения, ни в коем случае не нова и существует уже более 20 лет. Хотя бóльшая часть изысканий в данной области показала свою успешность в автоматической настройке моделей глубокого обучения, они получили второстепенное внимание из-за шумихи вокруг искусственного интеллекта и более передовых образцов ручной работы. Авторы многих статей обсуждают значительное время, затрачиваемое на конструирование данных или признаков и настройку гиперпараметров инновационной модели.

Однако для многих, кто сейчас осваивает глубокое обучение, задача разработки надежных, высокорезультативных моделей является пугающе непростой и сопряжена с трудностями. Многие из этих трудностей требуют продвинутого и изощренного знания всех опций и хитросплетений выбранной платформы глубокого обучения, чтобы понимать ситуации, в которых модель, возможно, просто неправильно подогнана. Представленное здесь эволюционное глубокое обучение как решение на основе автоматизированного машинного обучения (AutoML) способно решать большинство задач, с которыми столкнутся практики любого уровня, от начинающих до опытных.

Эволюционное глубокое обучение призвано предоставлять более совершенный механизм и набор инструментов оптимизации и автоматизации разработки решений с использованием глубокого обучения. Эволюционные методы представляют собой отличный и относительно простой механизм, обеспечивающий широкий набор инструментов оптимизации, могущих применяться в глубоком обучении. Хотя и существует неплохой потенциал автоматизации строительства более совершенного ИИ за счет эволюционных методов, это не входит в текущую цель ни эволюционного глубокого обучения, ни этой книги.

Вместо этого мы сосредоточимся на разработке более оптимизированных нейросетей с использованием эволюционных технологий. Однако прежде чем мы это сделаем, в следующем далее разделе мы рассмотрим операции и обсудим использование эволюционных вычислений и эволюционных алгоритмов, чтобы значительно углу-

[1] Англ. evolutionary computation (EC). – *Прим. перев.*

биться в базовые концепции, начиная с краткого введения в эволюцию и эволюционные процессы.

1.1.1 Введение в эволюционные вычисления

Эволюционные вычисления – это область искусственного интеллекта, в которой для решения сложных задач используются биологические и обусловленные природой процессы. Для описания этого семейства алгоритмов используется слово *эволюция*, поскольку теория естественного отбора применяется в качестве основы.

Теория естественного отбора, разработанная Чарльзом Дарвином в своей книге «Происхождение видов» (1859, Джон Мюррей)[1], дала определение эволюционному процессу жизни на Земле. Она описывает, как самые сильные и приспособленные представители жизни продолжат расти, в то время как слабые или плохо приспособленные будут умирать и вымрут. Он развил эту теорию на основе своего опыта натуралиста на борту корабля королевского флота Великобритании «Бигль», когда тот огибал Южную Америку около 1837 года. Прежде чем опубликовать свою знаменитую работу, Дарвин, будучи глубоко религиозным человеком, еще 22 года боролся со своими открытиями.

Основываясь на теории Дарвина, краеугольным камнем эволюционных вычислений является концепция симулирования особи или популяции особей в системе, чтобы отыскать наилучшую. Цель состоит в том, чтобы эволюционно вывести, или эволюционировать, особь, которая сможет выжить и процветать в такой искусственной среде, дав ей возможность изменяться. Этот механизм изменения особи будет варьироваться в зависимости от метода эволюционных вычислений, но во всех случаях требуется механизм, который оценивает выживаемость особи количественно.

Для количественного оценивания выживаемости или процветания особи используется специальный термин, который называется *приспособленностью*[2]. Этот универсальный термин применяется во всех эволюционных вычислениях и определяет степень выживаемости или живучести особи в некой среде. Приспособленность может измеряться множеством способов, но во всех случаях она является главным фактором, определяющим эффективность, с которой особь или группа особей решает ту либо иную задачу.

Концепции естественного отбора и приспособленности были использованы в качестве краеугольных камней нескольких вычислительных методов, разработанных для воспроизведения биологического процесса размножения, как в общих чертах, так и во всех деталях. Некоторые из этих методов даже симулируют генетический митоз в клетках, который происходит во время деления хромосом

---

[1] Origin of Species, Charles Darwin (1859, John Murray). – *Прим. перев.*

[2] Англ. fitness. – *Прим. перев.*

и обмена ДНК. Следующий ниже список представляет собой краткое изложение существующих известных методов эволюционных вычислений:

- *искусственная жизнь* – возвращаясь к игре Конвея[1] в жизнь и клеточному автомату фон Неймана, эти процессы симулируют искусственный процесс самой жизни, используя агентов. В данном алгоритме агенты перемещаются, перетекают, живут или умирают в зависимости от их близости к другим агентам или окружающей среде. Хотя симуляция агентов часто проводится для имитации реального мира, она также используется для оптимизации процессов;
- *дифференциальная эволюция* – это процесс, в котором поиск оптимизируется путем комбинирования дифференциального исчисления с эволюционными алгоритмами. Данная методика часто будет использоваться в наложении с еще одним методом эволюционных вычислений, в частности искусственной жизнью. В указанном алгоритме агенты эволюционируют, или изменяются, путем взятия векторных разностей и повторного их применения к популяции;
- *эволюционные алгоритмы*[2] – это более широкая категория методов эволюционных вычислений, в которых эволюция применяется к задаче в форме естественного отбора. Эти методы часто ориентированы на симулирование популяции особей;
- *эволюционное программирование*[3] – это специализированная форма эволюционных алгоритмов, которые генерируют алгоритмы, используя исходный код. В данном методе особь представлена блоком исходного кода, и соответствующая ему приспособленность измеряется до некоторого оптимального значения, генерируемого при исполнении исходного кода. Для генерации исходного кода в рамках эволюционного программирования реализовано несколько способов, и во многих случаях мы будем прибегать к более специфическим методам, таким как генные выражения;
- *генетический алгоритм*[4] – в этом методе используется низкоуровневый клеточный митоз, наблюдаемый в организмах, который допускает передачу генетических признаков потомку. Генетический алгоритм симулирует данный процесс путем кодирования характеристик особи в последовательность генов, где эта произвольная последовательность генов, могущая быть просто в виде последовательности из нулей и единиц, оценивается по некоторой метрике приспособленности. Указанная

---

[1] Game of Life, Conway. – *Прим. перев.*

[2] Англ. evolutionary algorithm (EA). – *Прим. перев.*

[3] Англ. evolutionary programming (EP). – *Прим. перев.*

[4] Англ. genetic algorithm (GA). – *Прим. перев.*

приспособленность используется для моделирования процесса биологического отбора и спаривания родительских особей с целью получения нового комбинированного потомства;

- *генетическое программирование*[1] – этот метод создает программный исходный код с использованием генетического алгоритма. В генетическом алгоритме черты особи имеют более обобщенный вид, а в генетическом программировании черта, или ген, может представлять любое число функций или другую логику исходного кода. Генетическое программирование представляет собой специализированный метод, который позволяет развивать новый алгоритмический исходный код. Его примеры использовались для написания исходного кода симуляции агента, который может разгадывать лабиринт или создавать картинки;
- *программирование генных выражений*[2] – этот метод является дальнейшим расширением генетического программирования, которое развивает исходный код или математические функции. В генетическом программировании исходный код абстрагируется до высокоуровневых функций, тогда как программирование генных выражений призвано развивать конкретные математические уравнения. Ключевая разница между программированием генных выражений и генетическим программированием лежит в использовании деревьев выражений для представления функций. В генетическом программировании деревья выражений представляют исходный код, тогда как в программировании генных выражений они представляют деревья математических выражений. При этом его преимущество заключается в том, что исходный код следует четко определенному порядку операций, основанному на расстановке;
- *оптимизация роя частиц*[3] – этот метод относится к подмножеству искусственной жизни и представляет собой симуляцию искусственных и в некоторой степени разумных частиц. В этом методе оценивается приспособленность каждой частицы, и самая лучшая частица становится центром роения остальных частиц;
- *роевой интеллект* – этот поисковый метод, который симулирует поведение роя насекомых или стаи птиц, чтобы отыскивать пиковые значения для задач оптимизации. Он очень похож на оптимизацию роя частиц, но отличается по реализации, в зависимости от процедуры оценивания приспособленности.

На рис. 1.1 показана иерархия методов эволюционных вычислений, используемых в этой книге для применения эволюционно-

---

[1] Англ. genetic programming (GP). – *Прим. перев.*

[2] Англ. gene expression programming (GEP). – *Прим. перев.*

[3] Англ. particle swarm optimization (PSO). – *Прим. перев.*

го глубокого обучения. Для совершенствования моделей глубокого обучения можно было бы использовать несколько других методов эволюционных вычислений, но в качестве введения на рисунке мы рассмотрим базовые методы, сосредоточив внимание на симуляции жизни и генетической симуляции.

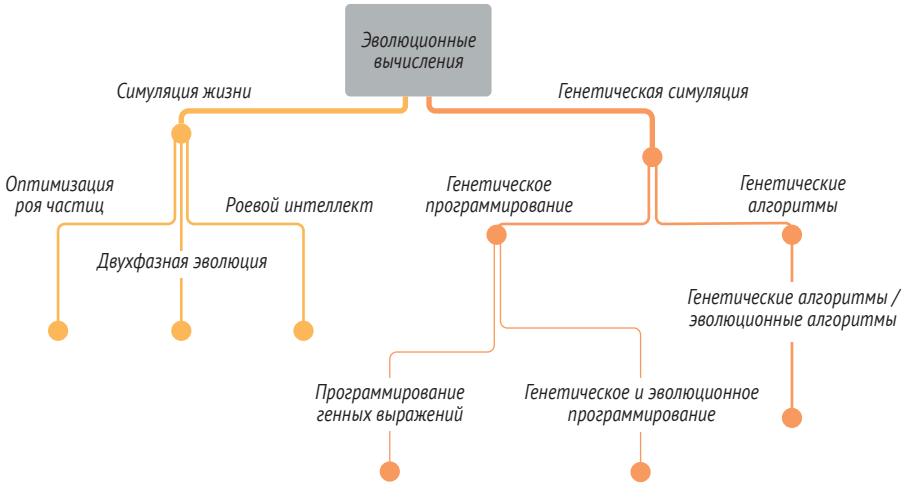

Рис. 1.1 Подмножество эволюционных вычислений, используемое для применения эволюционного глубокого обучения

Симуляция жизни – это специфическое подмножество эволюционных вычислений, в котором используется подход на основе симуляции наблюдаемых естественных процессов, встречаемых в природе, таких как роение частиц или перемещение стаи птиц. С другой стороны, генетическая симуляция имитирует процесс клеточного митоза, который наблюдается в биологической жизни. В частности, она симулирует генетическое перенесение генов и хромосом в процессе эволюции организма.

1.2 Зачем нужно эволюционное глубокое обучение и где оно применяется

Эволюционное глубокое обучение – это такая же концепция, как и ряд инструментов и техник, которые служат для оптимизации глубокого обучения. В концептуальном плане эволюционное глубокое обучение представляет собой шаблон и практику привлечения эволюционных вычислений для оптимизации нейросетей глубокого обучения. Вместе с тем оно также представляет собой набор инструментов, которые могут накладываться поверх глубокого обучения – или даже выступать в качестве его замены.

Ответ на вопросы, для чего и где применять эволюционное глубокое обучение, зависит не только от вашего уровня знаний в области глубокого обучения, но и от вашей потребности раздвигать границы дозволенного. Это не означает, что новички в глубоком обучении не могли бы извлекать выгоду из использования эволюционного глубокого обучения. На самом деле в этой книге проводится разведывательный анализ многих нюансов нейронных сетей, которые раскрываются с помощью эволюционного глубокого обучения и могут быть полезны любому практику.

Ответ на вопрос «*где* использовать эволюционное глубокое обучение» прост: где угодно. Его можно использовать для базовой гиперпараметрической оптимизации, поиска нейронных весов для прерывных (не непрерывных) решений, уравновешивания состязательных сетей в генеративных состязательных сетях и даже замены глубокого (само)обучения с подкреплением. Представленные в этой книге методы в действительности можно применять к любой системе глубокого обучения.

Ответ на вопрос «*зачем* его использовать» сводится к необходимости. Эволюционные методы предоставляют возможность дальнейшей оптимизации или усовершенствования решения на базе любой системы глубокого обучения. Впрочем, следует признать, что эволюционное глубокое обучение отличается вычислительной интенсивностью и, возможно, не будет подходить для применения в более простых системах. Однако в сложных или новых задачах эволюция предлагает любому практику глубокого обучения новый арсенал хитроумных приемов.

1.3 Потребность в оптимизации глубокого обучения

Глубокое обучение – это мощная, но относительно новая и нередко неправильно понимаемая технология, которая обладает множеством преимуществ, а также недостатков. Одним из таких недостатков является требование понимания и оптимизации модели. Этот процесс может отнимать несколько часов аннотирования данных или настройки модельных гиперпараметров.

Почти во всех случаях использовать модель прямо из коробки не получится, и зачастую приходится оптимизировать различные аспекты системы глубокого обучения, от настройки скорости усвоения до выбора функции активации. Оптимизация нейросетевой модели нередко становится первостепенной задачей, и если ее выполнять вручную, то на нее могут тратиться значительные усилия.

Оптимизация нейросети глубокого обучения может охватывать широкий спектр факторов. Помимо обычной гиперпараметриче-

ской настройки, также приходится обращать внимание на саму нейросетевую архитектуру.

1.3.1 Оптимизация нейросетевой архитектуры

По мере добавления слоев или различных типов узлов нейросеть становится сложнее, а это напрямую влияет на характер обратного распространения потери/ошибки по нейросети. На рис. 1.2 показаны наиболее распространенные проблемы, возникающие при выращивании более сложных и крупных систем глубокого обучения.

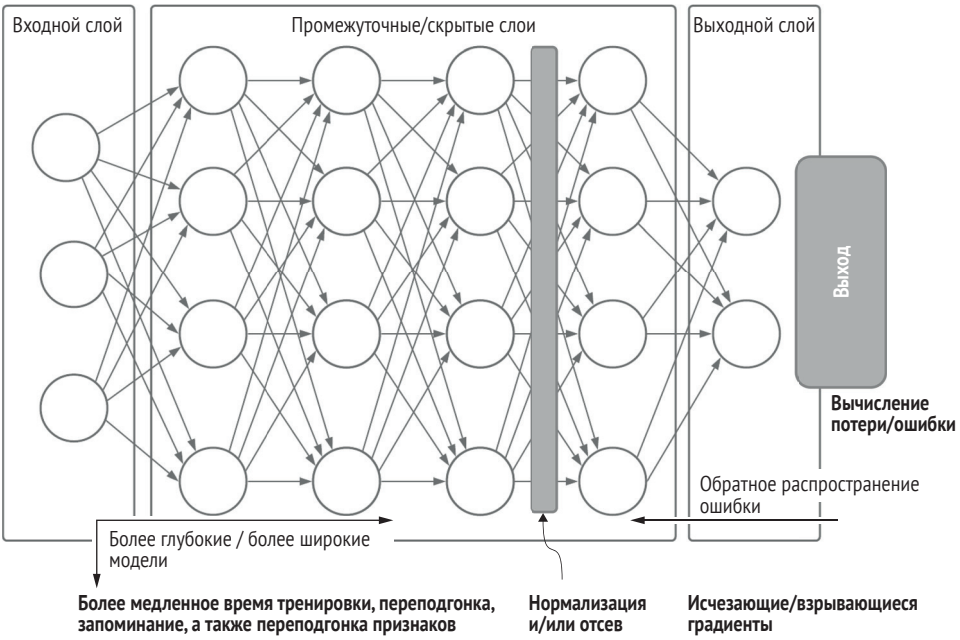

Рис. 1.2 Распространенные проблемы при выращивании систем глубокого обучения

В более крупных нейросетях величина потери должна делиться на все более мелкие составляющие, которые в конечном итоге приближаются к нулю. Когда эти компоненты потери, или градиенты, приближаются к нулю, такая ситуация называется проблемой *исчезающих градиентов*, которая часто ассоциируется с глубокими нейросетями. И наоборот, при прохождении через последовательные слои, которые увеличивают эти входные сигналы, компоненты также могут становиться исключительно большими. Это приводит к тому, что величина градиентных компонентов становится большой, и такая ситуация называется *взрывающимися градиентами*.

Обе проблемы с градиентами решаются с помощью различных методик, таких как нормализация входных значений, и, опять же, при прохождении слоев. На рис. 1.2 показаны специальные типы

слоевых методов, именуемых *нормализацией* и *отсевом*. Эти методы увеличивают вычислительную сложность и требования к нейросети, а также могут явно сглаживать важные и характерные признаки данных. И следовательно, для обеспечения хорошей результативности нейросети требуются более крупные и разнообразные тренировочные наборы данных.

Нормализация может решать проблемы с исчезающими и взрывающимися градиентами в глубоких сетях, но по мере роста моделей проявляются другие проблемы. Например, по мере роста увеличивается способность моделей переваривать большие наборы входных данных и изображений. Однако это может вызывать побочный эффект, так называемое *нейросетевое запоминание*, который может возникать, если входной тренировочный набор слишком мал. Это происходит из-за того, что сеть настолько велика, что она может начать запоминать блоки входных данных или даже целые изображения либо куски текста.

Передовые модели глубокого обучения, о которых вы, возможно, слышали, в частности разработанный в OpenAI процессор естественного языка GPT-3, частично страдают от запоминания. Эта проблема проявляется даже после ввода в такие модели миллиардов документов, представляющих многочисленные формы текста. Как было доказано, даже при использовании таких разнообразных и массивных тренировочных наборов GPT-3-подобные модели воспроизводят целые абзацы заученного текста. Эта «проблема» может быть результативным признаком базы данных, которая плохо вписывается в модель глубокого обучения.

Для решения проблемы запоминания были разработаны обходные пути, именуемые *отсевом*, – это процесс, с помощью которого при каждом прохождении тренировки в слоях нейросети деактивируется определенный процент узлов. В результате выключения и включения узлов внутри каждого прохождения создается более общая нейросеть. Однако это достигается за счет того, что теперь нейросеть должна быть на 100–200 % больше.

Вдобавок к этим проблемам добавление большего числа слоев в и так уже глубокие нейросети добавляет больше весов – которые необходимо тренировать индивидуально в течение миллиардов и триллионов итераций. Для тренировки таких моделей требуется экспоненциальный рост вычислительной мощности, и многие самые лучшие, ультрасовременные модели в настоящее время разрабатываются только в тех организациях, которые могут себе позволить подобные высокие затраты.

Многие видят, что тренд на более широкие и более глубокие нейросети вскоре достигнет плато для большинства практиков глубокого обучения, оставляя любые будущие передовые разработки гигантам искусственного интеллекта, таким как Google DeepMind. Поэтому простым решением было бы рассмотреть альтернативные подходы, которые могут рационализировать разработку таких круп-

ных сетей. Здесь мы возвращаемся к применению эволюционных вычислений к глубокому обучению в целях оптимизации нейросетевой архитектуры, весов либо того и другого вместе.

К счастью, эволюционное глубокое обучение предоставляет несколько перспективных методов, поскольку оно может автоматически оптимизировать размер и форму нейросети под решение самых разных задач, которые мы рассмотрим в этой книге. Автоматическая оптимизация является краеугольным камнем эволюционного глубокого обучения и будет в центре внимания многих упражнений, демонстрирующих эти методы в данной книге.

Поскольку эволюционные алгоритмы предусматривают несколько шаблонов оптимизации, которые могут решать множество задач, эволюционное глубокое обучение может работать с самыми разными аспектами процесса разработки моделей машинного обучения. К ним относятся настройка модельных гиперпараметров вплоть до конструирования данных или признаков, валидации моделей, отбора моделей и модельной архитектуры.

1.4 Автоматизация оптимизации средствами автоматизированного машинного обучения

Эволюционное глубокое обучение предоставляет набор инструментов, помогающих автоматизировать оптимизацию систем глубокого обучения с целью получения более надежных моделей. Как таковое его следует рассматривать как инструмент AutoML. Во многих коммерческих платформах AutoML, таких как Google AutoML, для разработки моделей используются различные эволюционные методы.

Прежде чем продолжить, нам также необходимо обсудить брендирование или ошибочное именование терминов *автоматизированное машинное обучение* и *AutoML*. В этой книге использование указанных терминов будет чередоваться; их часто считают одним и тем же, и для наших целей так оно и есть. Однако их можно считать разными в том смысле, что второй термин часто используется для описания «черноящичной» системы, которая производит оптимизированные модели.

Автоматизация оптимизации и разработки любой модели искусственного интеллекта / машинного обучения считается следующим шагом в процессе разработки любого научно-изыскательского проекта. Это эволюционный этап, заключающийся в выходе за рамки научных изысканий и разработок и формализации процесса строительства моделей, который позволяет практикам использовать модели для полномасштабной коммерциализации и продукционализации.

1.4.1 Что такое автоматизированное машинное обучение?

Автоматизированное машинное обучение, или AutoML, – это инструмент или набор инструментов, используемых для автоматизации и улучшения сборки модели искусственного интеллекта / машинного обучения. Это не конкретная технология, а коллекция методов и стратегий, в которых эволюционные алгоритмы или методы эволюционной оптимизации рассматриваются как подмножество. Как показано на рис. 1.3, этот инструмент можно использовать во всем рабочем процессе искусственного интеллекта / машинного обучения.

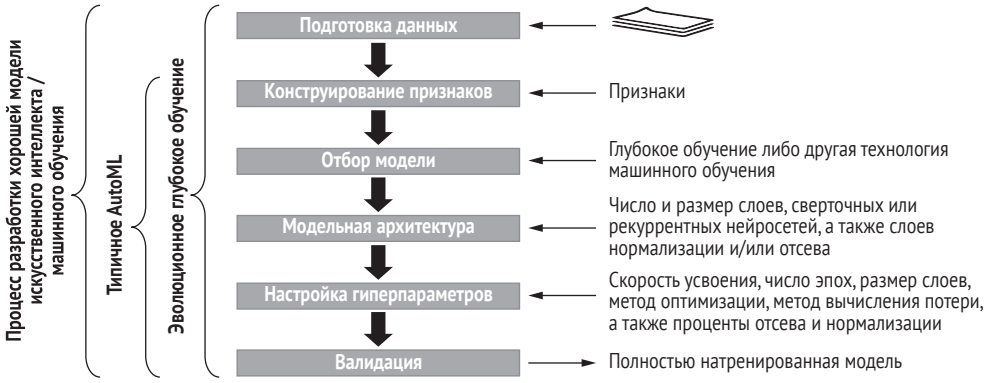

Рис. 1.3 Шаги по разработке хорошей модели искусственного интеллекта / машинного обучения с использованием AutoML и/или эволюционного глубокого обучения

На рис. 1.3 изображен типичный рабочий процесс сборки хорошей модели искусственного интеллекта / машинного обучения, применимой для уверенного модельного вывода на новых данных. Этот рабочий процесс часто выполняется вручную различными практиками искусственного интеллекта / машинного обучения. Вместе с тем неоднократно предпринимались различные попытки автоматизировать все шаги. Ниже приводится более подробная сводная информация по каждому шагу, в том числе приемы их автоматизации с помощью методов AutoML:

- *подготовка данных* – подготовка данных для тренировки модели искусственного интеллекта / машинного обучения требует много времени и затрат. В целом подготовка данных и автоматизация этого задания могут значительно повышать производительность рабочих процессов обработки данных, критически важных для точной настройки сложных моделей. В онлайновых службах AutoML часто берется за основу, что пользователь уже подготовил и очистил данные, как того требует большинство моделей машинного обучения. Существует ряд способов автоматизации подготовки данных с помощью эволюционных ме-

тодов, и хотя эта задача не является специфичной для эволюционного глубокого обучения, мы рассмотрим ее в последующих главах;

- *конструирование признаков* – это процесс извлечения релевантных признаков из данных при помощи предварительного знания предметной области, при этом эксперты отбирают релевантные признаки на основе своей интуиции и опыта. Поскольку эксперты в предметной области обходятся дорого и самоуверенны в своих суждениях, автоматизация этого задания снижает затраты и улучшает стандартизацию. В зависимости от инструмента AutoML в процесс может быть включено конструирование признаков;
- *отбор модели* – в ходе развития технологии искусственного интеллекта / машинного обучения были созданы сотни типов моделей, которые могут решать схожие задачи. Зачастую исследователи данных тратят дни или даже недели на то, чтобы лишь подобрать группу моделей для дальнейшего оценивания. Автоматизация этого процесса ускоряет разработку модели и помогает исследователю данных подтверждать, что он использует правильную модель. Хороший инструмент AutoML может выбирать из десятков или сотен моделей, включая вариации глубокого обучения или модельные ансамбли;
- *модельная архитектура* – в зависимости от области искусственного интеллекта / машинного и глубокого обучения определение правильной модельной архитектуры нередко имеет решающее значение. Выполнение этого требования в автоматическом режиме сокращает бесчисленные часы настройки архитектуры и повторного выполнения моделей. В зависимости от реализации некоторые системы AutoML отличаются модельной архитектурой, но обычно это отличие ограничивается хорошо известными вариациями;
- *гиперпараметрическая оптимизация* – процесс тонкой настройки модельных гиперпараметров может занимать много времени и приводить к ошибкам. В целях преодоления этой проблемы многие практики полагаются на интуицию и предыдущий опыт. Хотя в прошлом это работало успешно, в настоящее время возрастающая сложность моделей делает эту задачу невыполнимой. Автоматизируя гиперпараметрическую настройку, мы не только облегчаем работу разработчиков, но и выявляем потенциальные недостатки в отборе модели или модельной архитектуре;
- *валидационный отбор* – существует масса вариантов оценивания результативности модели, от принятия решения об объемах данных, которые следует использовать для тренировки и тестирования, вплоть до визуализации результативности на выходе из модели. Автоматизация процедуры валидации модели обеспечивает надежное средство переоценки резуль-

тативности модели при изменении данных и в долгосрочной перспективе делает модель более объяснимой. В онлайновых службах AutoML она является ключевым преимуществом и веской причиной для использования таких инструментов.

Инструменты AutoML

Ниже приведен список инструментов и платформ, которые обеспечивают работу AutoML:

- DataRobot – рассматривается как первая платформа и отправная точка AutoML, предоставляет разнообразный набор инструментов для автосборки моделей;
- Google Cloud AutoML – это популярная и надежная платформа от нынешнего главного игрока в области искусственного интеллекта. Указанная платформа оперирует разнообразными наборами данных, от изображений до структурированных данных;
- Amazon SageMaker AutoPilot – эта мощная платформа хорошо подходит для автоматизации разработки моделей, зависящих от структурированных данных;
- H2O AutoML – этот инструмент предоставляет различные функции, предназначенные для автоматизации рабочего процесса машинного обучения;
- Azure Machine Learning – эта платформа обеспечивает автоматизированные процессы настройки моделей на основе самых разных форм данных;
- AutoKeras – этот превосходный инструмент обеспечивает автоматизированную разработку нейросетевой архитектуры;
- AutoTorch – этот инструмент обеспечивает автоматический архитектурный поиск.

В приведенный выше список не вошел целый ряд других инструментов и платформ.

Типичный рабочий процесс автоматизированого машинного обучения / AutoML пытается охватить только шаг конструирования признаков и далее, где процесс часто выполняется итеративно, либо в течение одного шага, либо в сочетании с несколькими шагами. Некоторые шаги, такие как гиперпараметрическая настройка, специфичны для типа модели, и в случае глубокого обучения оптимизация модели может занимать значительное время.

В то время как эта новая волна коммерческой службы AutoML успешно справляется с обработкой широкого спектра типов и форм данных, производимые модели лишены инноваций и бывают довольно дорогими. Выполнение всех заданий, которые служба AutoML должна проделывать для сборки настроенной модели, требует зна-

чительного объема вычислительной мощности, а получаемые модели, по сути, являются реконструкцией тестовых эталонов предыдущего поколения и часто не содержат какого-либо нового понимания оптимизации.

Те практики искусственного интеллекта / машинного обучения, которые хотят получать более инновационные автоматизированные модели в рамках бюджета, часто обращаются к разработке своих решений в области AutoML, причем эволюционное глубокое обучение является первичным кандидатом. Как мы увидим в последующих главах этой книги, эволюционные методы могут предоставлять широкий спектр решений для автосборки и оптимизации моделей глубокого обучения, гиперпараметров, конструирования признаков и нейросетевой архитектуры.

1.5 Приложения эволюционного глубокого обучения

Теперь, когда мы знаем ответ на вопрос, *почему* необходимо комбинировать эволюционные вычисления и глубокое обучение в решении AutoML, можно перейти к вопросу о том, *как* это делать, то есть рассмотреть процедуры применения таких методов, как генетические алгоритмы, поверх глубокого обучения, чтобы совершенствовать работающие решения искусственного интеллекта. Вполне очевидно, что существует бесчисленное множество перспектив, создающих возможности для слияния эволюционных вычислений с глубоким обучением, но в этой книге мы будем придерживаться нескольких базовых практических стратегий.

Понимание этих стратегий позволит вам модифицировать существующие нейросети глубокого обучения или строить собственные новые комбинированные модели на основе эволюционных вычислений / глубокого обучения. Оно позволит создавать передовые оптимизированные нейросети за более короткий промежуток времени и с меньшими затратами ресурсов, предоставляя вам возможность выбирать стратегии или даже развивать новые по мере накопления опыта.

В целях достижения таких высоких целей мы проведем разведывательный анализ основ как глубокого обучения, так и конкретного подмножества эволюционных вычислений с самого начала. Мы построим базовые модели для решения задач с использованием обоих подобластей, а затем в последующих главах обратимся к их комбинированию в целях повышения результативности и автоматизации.

Эволюционные вычисления можно применять к глубокому обучению в нескольких формах, охватывая различные автоматизированные стратегии, обернутые в AutoML. На рис. 1.4 показаны различные подмножества эволюционных вычислений или эволюционного глу-

бокого обучения, которые могут применяться к глубокому обучению в рамках рабочего процесса разработки модели искусственного интеллекта / машинного обучения.

1.5.1 Отбор модели: поиск весовых коэффициентов

Как упоминалось ранее, отобранная базовая модель и типы слоев часто зависят от типа решаемой задачи. В большинстве случаев оптимизацию отбора модели можно выполнять быстро и вручную. Однако отбор модели не сводится только к отбору типа слоев; он также может включать форму оптимизации, начальные веса и потерю, используемые для тренировки модели.

Оптимизировав типы модельных слоев, механизм оптимизации и даже формы потери, нейросеть можно сделать надежнее и добиться того, чтобы она училась более эффективно. Мы рассмотрим примеры, в которых первоначальные веса, типы оптимизаций и меры потери настраиваются, чтобы укладываться в рамки различных задач.

1.5.2 Модельная архитектура: архитектурная оптимизация

Нередко при строительстве нейросетей глубокого обучения преувеличивают размер модели, или число узлов и слоев в модели. Затем, со временем, нейросеть снова масштабируют, чтобы сделать ее более оптимальной для решения задачи. Во многих случаях слишком большая сеть может приводить к запоминанию входных данных, что в свою очередь приводит к переподгонке. И наоборот, сеть, которая слишком мала для того, чтобы усвоить разнообразие и объем данных, как правило, страдает от недоподгонки.

В целях решения проблем с переподгонкой и недоподгонкой можно применять генетический алгоритм, чтобы автоматически подрезать нейросеть до ее самой низкой формы. Это не только повышает результативность модели и ограничивает переподгонку или недоподгонку, но и сокращает время тренировки за счет уменьшения размера нейросети. Указанная методика хорошо работает при попытке оптимизировать более крупные и глубокие сети.

1.5.3 Гиперпараметрическая настройка/оптимизация

Применяемый в искусственном интеллекте и машинном обучении процесс гиперпараметрической настройки предусматривает оптимизацию модели путем регулировки значений разных управляющих переменных, которые определяют эту модель. В глубоком обучении термин *параметры* используется для обозначения модельных весов; они отличаются от управляющих переменных, которые называются *гиперпараметрами*.

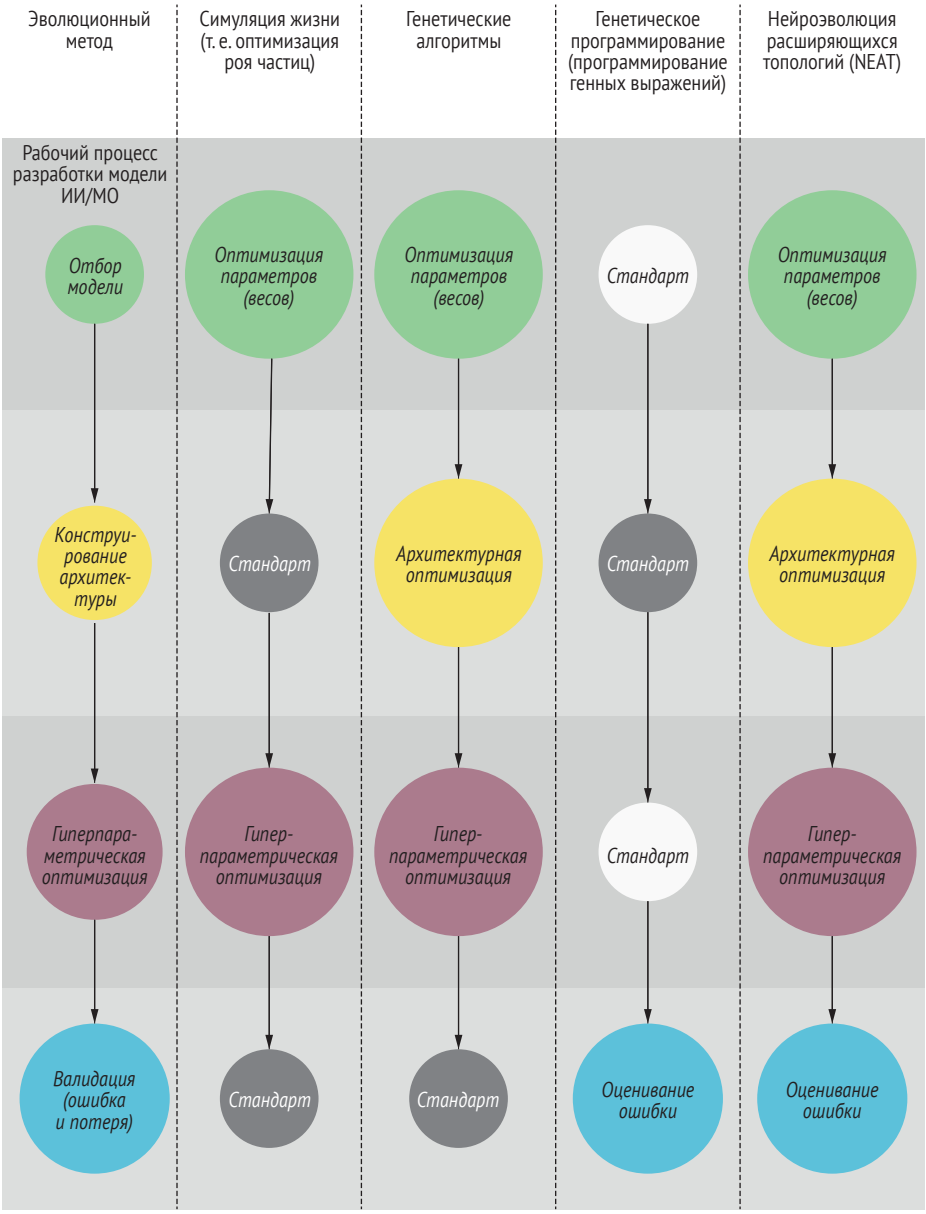

Рис. 1.4 Применение эволюционных вычислений (эволюционного глубокого обучения) к рабочему процессу развития модели искусственного интеллекта / машинного обучения для глубокого обучения

Эволюционные вычисления предлагают несколько альтернативных мер добавления автоматической гиперпараметрической оптимизации в широкий спектр моделей, включая модели глубокого обучения. Оптимизация роя частиц, дифференциальная эволюция и генетические алгоритмы – все они использовались довольно

успешно. Для того чтобы оценить результативность каждого из этих методов, будет проведен их разведывательный анализ в различных фреймворках.

1.5.4 Валидация и оптимизация функции потери

При разработке надежных моделей глубокого обучения нередко опираются на несколько устоявшихся шаблонов генерирования качественных нейросетей. Указанные шаблоны могут предусматривать валидацию тренировки и результативности модели путем итеративного наблюдения за потерей во время тренировки и тестирования. Это делается для подтверждения того, что обе меры потери не слишком сильно расходятся друг от друга.

В типичном сценарии контролируемого (само)обучения часто используются устоявшиеся меры, которые ориентированы на сравнения с метками. В сценариях с более продвинутыми генеративными моделями глубокого обучения становятся доступными возможности по оптимизации формы потери и даже мер валидации.

Нейросетевые архитектуры, такие как автокодировщики, слои векторных вложений и генеративные состязательные сети, предоставляют возможность использовать комбинаторные определения потери и валидации модели. Используя эволюционные вычисления, можно задействовать методы, позволяющие оптимизировать эти формы нейросетей в стиле AutoML.

1.5.5 Нейроэволюция расширяющихся топологий

Нейроэволюция расширяющих топологий (NEAT)[1] – это метод, сочетающий в себе гиперпараметрическую и архитектурную оптимизацию с весовым поиском с целью автоматической сборки новых моделей глубокого обучения, которые также могут развивать свой метод потери и валидации. Хотя метод NEAT был разработан почти 20 лет назад, только сравнительно недавно его стали применять к различным приложениям глубокого обучения, а также приложениям глубокого (само)обучения с подкреплением.

1.5.6 Цели

В этой книге мы обратимся к ранее упомянутому набору методик и способам их применения к глубокому обучению. Мы сосредоточимся на практических методах, которые можно применять ко множеству задач с использованием рабочих решений, также уделяя особое внимание применению разных форм автоматизированного машинного обучения / AutoML для оптимизации систем глубокого

[1] Англ. Neuroevolution of augmenting topologies (NEAT). – *Прим. перев.*

обучения и оценивания результативности в рамках разных методов. Наше внимание также включает в себя более широкий спектр методов, выходящих за рамки эволюционных методов.

В следующих главах мы рассмотрим разделы процесса AutoML, которые будут постепенно знакомить посвященных в глубокое обучение читателей с ключевыми понятиями. После рассмотрения основ эволюционных вычислений мы перейдем к демонстрации гиперпараметрической оптимизации, а затем к конструированию данных и признаков, отбору модельных параметров и модельной архитектуре. Наконец, обратимся к более сложным примерам, которые направлены на улучшение задач генеративного глубокого обучения и глубокого (само)обучения с подкреплением.

К концу этой книги вы должны чувствовать себя удобно, описывая и используя глубокое обучение и определенные подмножества эволюционных вычислений в отдельности или в комбинации в целях оптимизации нейросетей. Вы сможете строить модели для решения задач, используя обе подобласти, а также понимать, какие из них подходят для конкретных классов задач лучше всего, включая возможность применять эволюционные вычисления поверх моделей глубокого обучения для различных оптимизаций и приложений AutoML.

Резюме

- Глубокое обучение – это мощная технология, способная решать многие задачи искусственного интеллекта и машинного обучения, но она характерна своей сложностью, требует значительных объемов данных, а разработка, тренировка и оптимизация ее моделей обходятся дорого.
- Эволюционные вычисления – это подобласть искусственного интеллекта и машинного обучения, которая определяется теорией естественного отбора. Она созрела не так быстро, как глубокое обучение, но все же предоставляет методики для решения широкого спектра сложных задач.
- Эволюционное глубокое обучение – это широкий термин, охватывающий эволюционные методы и глубокое обучение. Примерами эволюционного глубокого обучения являются нейроэволюция, эволюционная гиперпараметрическая оптимизация и нейроэволюция расширяющихся топологий. Эволюционное глубокое обучение определяет подмножество методов эволюционных вычислений, которые могут использоваться для автоматизации и улучшения разработки моделей глубокого обучения на многих стадиях рабочего процесса машинного обучения.
- Автоматизированное машинное обучение и AutoML определяют набор инструментов и методик, направленных на автоматизацию

всего рабочего процесса разработки моделей искусственного интеллекта и машинного обучения. Многие формы эволюционных вычислений использовались и могут использоваться для автоматизации рабочего процесса разработки моделей. Google и другие компании вложили значительные средства в разработку технологии AutoML, чтобы помочь потребителям создавать надежные модели для своих собственных нужд. Несмотря на мощность этих служб, они нередко работают как черный ящик и ограничивают более маневренную адаптацию новых передовых моделей.