

*Моей жене Люси с благодарностью за то,  
что она была моим лучшим другом и любящей супругой  
на протяжении всей нашей совместной жизни*

# Оглавление

<b>Предисловие от издательства</b> .....	13
Отзывы и пожелания.....	13
Список опечаток.....	13
Нарушение авторских прав.....	13
<b>Об авторе</b> .....	14
<b>О рецензентах</b> .....	15
<b>Предисловие от автора</b> .....	16
Для кого написана эта книга.....	17
О чем эта книга.....	17
Как получить максимальную отдачу от этой книги.....	18
Скачивание исходного кода примеров.....	18
Скачивание цветных иллюстраций.....	18
Условные обозначения и соглашения, принятые в книге.....	18
<b>ЧАСТЬ I. Основы эволюционных вычислительных алгоритмов и методов нейроэволюции</b> .....	19
<b>Глава 1. Обзор методов нейроэволюции</b> .....	21
1.1 Эволюционные алгоритмы и нейроэволюционные методы.....	22
1.1.1 Генетические операторы.....	23
1.1.2 Схемы кодирования генома.....	25
1.1.3 Коэволюция.....	27
1.1.4 Модульность и иерархия.....	27
1.2 Обзор алгоритма NEAT.....	28
1.2.1 Схема кодирования NEAT.....	28
1.2.2 Структурные мутации.....	29
1.2.3 Кроссовер с номером обновления.....	30
1.2.4 Видообразование.....	32
1.3 NEAT на основе гиперкуба.....	33
1.3.1 Сети, производящие составные паттерны.....	34
1.3.2 Конфигурация субстрата.....	35
1.3.3 CPPN с развивающимися связями и алгоритм HyperNEAT.....	36
1.4 HyperNEAT с развиваемым субстратом.....	37
1.4.1 Плотность информации в гиперкубе.....	37

1.4.2 Квадродерево как эффективный экстрактор информации .....	38
1.4.3 Алгоритм ES-HyperNEAT .....	40
1.5 Метод оптимизации поиском новизны .....	42
1.5.1 Поиск новизны и естественная эволюция .....	43
1.5.2 Метрика новизны.....	43
1.6 Заключение .....	45
1.7 Дополнительное чтение .....	45

## **Глава 2. Библиотеки Python и настройка среды разработки .....**

2.1 Библиотеки Python для экспериментов с нейроэволюцией.....	47
2.1.1 Библиотека NEAT-Python.....	48
2.1.2 Библиотека PyTorch NEAT .....	49
2.1.3 Библиотека MultiNEAT.....	51
2.1.4 Библиотека Deep Neuroevolution .....	53
2.2 Настройка среды .....	56
2.2.1 Pipenv.....	56
2.2.2 Virtualenv .....	57
2.2.3 Anaconda Distribution.....	58
2.3 Заключение .....	59

## **ЧАСТЬ II. Применение методов нейроэволюции для решения классических задач информатики.....**

### **Глава 3. Использование NEAT для оптимизации решения задачи XOR .....**

3.1 Технические требования .....	63
3.2 Суть задачи XOR.....	64
3.3 Целевая функция для эксперимента XOR .....	65
3.4 Настройка гиперпараметров .....	66
3.4.1 Секция NEAT.....	67
3.4.2 Секция DefaultStagnation .....	67
3.4.3 Секция DefaultReproduction .....	68
3.4.4 Секция DefaultSpeciesSet .....	68
3.4.5 Секция DefaultGenome.....	68
3.4.6 Гиперпараметры эксперимента XOR .....	70
3.5 Выполнение эксперимента XOR .....	72
3.5.1 Настройка среды .....	72
3.5.2 Исходный код эксперимента XOR .....	73
3.5.3 Запуск эксперимента и анализ результатов .....	76
3.6 Упражнения .....	81
3.7 Заключение .....	83

### **Глава 4. Балансировка тележки с обратным маятником .....**

4.1 Технические требования .....	85
4.2 Задача балансировки обратного маятника.....	86

4.2.1 Уравнения движения балансировщика .....	86
4.2.2 Уравнения состояния и управляющие воздействия.....	87
4.2.3 Взаимодействие между решателем и симулятором.....	88
4.3 Целевая функция для эксперимента по балансировке одиночного маятника .....	90
4.3.1 Моделирование тележки .....	90
4.3.2 Цикл моделирования .....	91
4.3.3 Оценка приспособленности генома .....	93
4.4 Эксперимент по балансировке одиночного маятника .....	93
4.4.1 Выбор гиперпараметров .....	94
4.4.2 Настройка рабочей среды.....	95
4.4.3 Исходный код эксперимента .....	95
4.4.4 Запуск эксперимента по балансировке одиночного маятника .....	98
4.5 Упражнения .....	100
4.6 Задача балансировки двойного маятника .....	101
4.6.1 Переменные состояния системы и уравнения движения .....	101
4.6.2 Подкрепляющий сигнал .....	104
4.6.3 Начальные условия и обновление состояния .....	104
4.6.4 Управляющие действия .....	106
4.6.5 Взаимодействие между решателем и симулятором.....	106
4.7 Целевая функция для эксперимента по балансировке двойного маятника.....	107
4.8 Эксперимент по балансировке .....	108
4.8.1 Выбор гиперпараметров .....	108
4.8.2 Настройка рабочей среды.....	110
4.8.3 Реализация эксперимента.....	110
4.8.4 Запуск эксперимента с двумя маятниками .....	111
4.9 Упражнения.....	115
4.10 Заключение.....	116
<b>Глава 5. Автономное прохождение лабиринта .....</b>	<b>117</b>
5.1 Технические требования .....	117
5.2 Задача навигации в лабиринте .....	118
5.3 Среда моделирования лабиринта .....	119
5.3.1 Агент-решатель лабиринта .....	119
5.3.2 Реализация среды моделирования лабиринта .....	121
5.3.3 Хранение записей агента .....	125
5.3.4 Визуализация записей агента .....	127
5.4 Определение целевой функции с использованием показателя приспособленности .....	127
5.5 Проведение эксперимента с простой конфигурацией лабиринта.....	129
5.5.1 Выбор гиперпараметров .....	130
5.5.2 Файл конфигурации лабиринта .....	132
5.5.3 Настройка рабочей среды.....	132
5.5.4 Реализация движка эксперимента .....	132
5.5.5 Проведение эксперимента по навигации в простом лабиринте.....	135
5.6 Упражнения.....	140

5.7 Эксперимент со сложной конфигурацией лабиринта.....	140
5.7.1 Настройка гиперпараметров.....	141
5.7.2 Настройка рабочей среды и движок эксперимента.....	141
5.7.3 Выполнение эксперимента по прохождению сложного лабиринта .....	141
5.8 Упражнения .....	143
5.9 Заключение .....	144
<b>Глава 6. Метод оптимизации поиском новизны.....</b>	<b>145</b>
6.1 Технические требования .....	145
6.2 Метод оптимизации поиском новизны .....	146
6.3 Основы реализации алгоритма поиска новизны .....	147
6.3.1 NoveltyItem .....	147
6.3.2 NoveltyArchive .....	148
6.4 Функция приспособленности с оценкой новизны .....	149
6.4.1 Оценка новизны.....	150
6.4.2 Метрика новизны.....	152
6.4.3 Функция приспособленности.....	153
6.5 Эксперимент с простой конфигурацией лабиринта .....	158
6.5.1 Настройка гиперпараметров.....	159
6.5.2 Настройка рабочей среды.....	159
6.5.3 Реализация движка эксперимента .....	160
6.5.4 Простой эксперимент по навигации в лабиринте с поиском новизны.....	163
6.5.5 Упражнение 1 .....	168
6.6 Эксперимент со сложной конфигурацией лабиринта.....	169
6.6.1 Настройка гиперпараметров и рабочей среды .....	169
6.6.2 Выполнение эксперимента по прохождению труднодоступного лабиринта .....	169
6.6.3 Упражнение 2 .....	172
6.7 Заключение .....	173
<b>ЧАСТЬ III. Передовые методы нейроразвития.....</b>	<b>175</b>
<b>Глава 7. Зрительное различение с NEAT на основе гиперкуба ...</b>	<b>177</b>
7.1 Технические требования.....	177
7.2 Косвенное кодирование нейросетей с CPPN.....	178
7.2.1 Кодирование CPPN .....	178
7.2.2 Нейроразвития с развитием топологии на основе гиперкуба.....	179
7.3 Основы эксперимента по зрительному различению .....	180
7.3.1 Определение целевой функции .....	182
7.4 Подготовка эксперимента по зрительному различению .....	182
7.4.1 Тестовая среда зрительного дискриминатора .....	183
7.4.2 Движок эксперимента.....	190
7.5 Эксперимент по зрительному различению объектов .....	196
7.5.1 Выбор гиперпараметра.....	196
7.5.2 Настройка рабочей среды.....	197

7.5.3 Запуск эксперимента по зрительному различению .....	198
7.6 Упражнения .....	201
7.7 Заключение .....	202
<b>Глава 8. Метод ES-HyperNEAT и задача сетчатки .....</b>	<b>203</b>
8.1 Технические требования .....	204
8.2 Ручное и эволюционное формирование топографии узлов.....	204
8.3 Извлечение информации из квадродерева и основы ESHyperNEAT .....	205
8.4 Основы задачи модульной сетчатки .....	207
8.4.1 Определение целевой функции .....	209
8.5 Подготовка эксперимента с модульной сетчаткой .....	210
8.5.1 Начальная конфигурация субстрата.....	210
8.5.2 Тестовая среда для задачи модульной сетчатки.....	211
8.5.3 Движок эксперимента .....	215
8.6 Эксперимент с модульной сетчаткой.....	222
8.6.1 Настройка гиперпараметров.....	222
8.6.2 Настройка рабочей среды.....	223
8.6.3 Запуск эксперимента с модульной сетчаткой .....	223
8.7 Упражнения.....	227
8.8 Заключение .....	227
<b>Глава 9. Козволюция и метод SAFE.....</b>	<b>229</b>
9.1 Технические требования .....	229
9.2 Общие стратегии козволюции .....	229
9.3 Метод SAFE .....	230
9.4 Модифицированный эксперимент с лабиринтом.....	231
9.4.1 Агент-решатель задачи лабиринта .....	231
9.4.2 Среда лабиринта .....	232
9.4.3 Определение функции приспособленности .....	233
9.5 Модифицированный поиск новизны.....	234
9.5.1 Функция <code>_add_novelty_item</code> .....	235
9.5.2 Функция <code>evaluate_novelty_score</code> .....	235
9.6 Движок модифицированного эксперимента с лабиринтом.....	236
9.6.1 Создание совместно эволюционирующих популяций.....	237
9.6.2 Оценка приспособленности совместно развивающихся популяций .....	238
9.6.3 Выполнение эксперимента модифицированного лабиринта .....	243
9.7 Эксперимент с модифицированным лабиринтом .....	245
9.7.1 Гиперпараметры для популяции агентов-решателей.....	245
9.7.2 Гиперпараметры популяции кандидатов на целевую функцию .....	246
9.7.3 Настройка рабочей среды .....	246
9.7.4 Проведение модифицированного эксперимента с лабиринтом.....	247
9.8 Упражнения.....	250
9.9 Заключение .....	250

<b>Глава 10. Глубокая нейроразволюция</b> .....	251
10.1 Технические требования .....	251
10.2 Глубокая нейроразволюция для глубокого обучения с подкреплением.....	252
10.3 Обучение агента игре Atari Frostbite с использованием глубокой нейроразволюции .....	253
10.3.1 Игра Atari Frostbite .....	254
10.3.2 Отображение игрового экрана на действия.....	254
10.3.3 Обучение игрового агента.....	257
10.4 Обучение агента навыкам игры в Frostbite.....	260
10.4.1 Учебная среда Atari .....	260
10.4.2 Расчет RL на ядрах GPU.....	262
10.4.3 Движок эксперимента .....	267
10.5 Запуск эксперимента с игрой Atari Frostbite.....	271
10.5.1 Настройка рабочей среды.....	271
10.5.2 Запуск эксперимента .....	272
10.5.3 Визуализация прохождения игры Frostbite .....	273
10.6 Визуальный инспектор нейроразволюции.....	274
10.6.1 Настройка рабочей среды.....	274
10.6.2 Использование VINE для визуализации эксперимента .....	274
10.7 Упражнения .....	276
10.8 Заключение.....	276

## **ЧАСТЬ IV. Обсуждение результатов и заключительные замечания .....**

277

<b>Глава 11. Лучшие методы, советы и подсказки</b> .....	279
11.1 Первичный анализ задачи .....	279
11.1.1 Предварительная обработка данных.....	280
11.1.2 Исследование проблемной области.....	282
11.1.3 Написание хороших симуляторов .....	282
11.2 Выбор оптимального метода поисковой оптимизации.....	283
11.2.1 Целеориентированный поиск оптимального решения .....	283
11.2.2 Оптимизация поиском новизны .....	284
11.3 Качественная визуализация.....	285
11.4 Настройка гиперпараметров.....	285
11.5 Метрики качества модели .....	287
11.5.1 Точность.....	287
11.5.2 Отклик.....	287
11.5.3 Оценка F1 .....	287
11.5.4 ROC AUC.....	288
11.5.5 Достоверность .....	289
11.6 Python, кодирование, советы и рекомендации.....	289
11.6.1 Советы и рекомендации .....	290
11.6.2 Рабочая среда и инструменты программирования.....	291
11.7 Заключение.....	292

---

<b>Глава 12. Заключительные замечания</b> .....	293
12.1 Что вы узнали в этой книге .....	293
12.1.1 Обзор методов нейроэволюции .....	293
12.1.2 Библиотеки Python и настройка среды разработки .....	295
12.1.3 Использование NEAT для оптимизации решения задачи XOR .....	296
12.1.4 Балансировка тележки с обратным маятником .....	296
12.1.5 Автономное прохождение лабиринта .....	298
12.1.6 Метод оптимизации поиском новизны .....	299
12.1.7 Зрительное различение с NEAT на основе гиперкуба .....	300
12.1.8 Метод ES-HyperNEAT и задача сетчатки .....	301
12.1.9 Козволюция и метод SAFE .....	302
12.1.10 Глубокая нейроэволюция .....	302
12.2 Куда двигаться дальше .....	303
12.2.1 Uber AI Labs .....	304
12.2.2 alife.org .....	304
12.2.3 Открытая эволюция в Reddit .....	304
12.2.4 Каталог программного обеспечения NEAT .....	304
12.2.5 arXiv.org .....	305
12.2.6 Оригинальная публикация про алгоритм NEAT .....	305
12.3 Заключение .....	305
<b>Предметный указатель</b> .....	306



# Предисловие от издательства

## Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте [www.dmkpress.com](http://www.dmkpress.com), зайдя на страницу книги, и оставить комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com), при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу [http://dmkpress.com/authors/publish\\_book/](http://dmkpress.com/authors/publish_book/) или напишите в издательство по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com).

## Список опечаток

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в тексте или в коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com), и мы исправим это в следующих тиражах.

## Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли применить санкции.

Пожалуйста, свяжитесь с нами по адресу электронной почты [dmkpress@gmail.com](mailto:dmkpress@gmail.com) со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.

# Об авторе

**Ярослав Омеляненко** более десяти лет работает техническим директором и директором по исследованиям. Он является активным членом исследовательского сообщества и опубликовал несколько исследовательских работ на arXiv, ResearchGate, Preprints и др. Он начал заниматься прикладным машинным обучением более десяти лет назад, разрабатывая автономные агенты для мобильных игр. Последние 5 лет активно участвовал в исследованиях, связанных с применением методов глубокого машинного обучения для аутентификации, распознавания личных качеств, групповой робототехники, искусственного интеллекта и многого другого. Он активный разработчик программного обеспечения и создает реализации нейрорезолюционного алгоритма с открытым исходным кодом на языке Go.

*Я хочу поблагодарить всех исследователей и разработчиков за то,  
что они поделились результатами своей работы и вдохновляются  
идеалами открытого кода.  
Без сообщества open source наш мир был бы другим*

# О рецензентах

**Алан Макинтайр** (Alan McIntyre) – главный архитектор программного обеспечения в CodeReclaimers, LLC, где он оказывает услуги по индивидуальному проектированию и разработке программного обеспечения для технических вычислительных приложений, включая вычислительную геометрию, компьютерное зрение и машинное обучение. Ранее он работал инженером-программистом в General Electric, Microsoft и нескольких стартапах.

**Унсал Гокдаг** (Unsal Gokdag) с 2017 года работает старшим научным сотрудником в области логистики, а в 2013 году работал инженером по исследованиям и разработкам. В настоящее время он готовит докторскую диссертацию по сравнению алгоритмов машинного обучения для удаления пятен на изображениях и классификации поляриметрических изображений SAR. Его прошлый опыт включает работу в области машинного обучения, компьютерного зрения и биоинформатики. Он впервые применил алгоритм NEAT в своей дипломной работе бакалавра и с тех пор интересуется эволюционными алгоритмами. В настоящее время проживает в Германии.

*Я хочу поблагодарить мою семью за бескорыстную любовь, которую они мне подарили. Без этой поддержки я бы ничего не добился. Моей маме и сестре: спасибо за все, что вы сделали для меня в самые трудные времена. Папа, я скучаю по тебе*

# Предисловие от автора

Когда традиционные методы глубокого обучения приблизились к пределу своих возможностей, все больше и больше исследователей начали искать альтернативные подходы к обучению *искусственных нейронных сетей* (artificial neural network, ANN).

Глубокое машинное обучение чрезвычайно эффективно для распознавания образов, но не позволяет выполнять задачи, требующие понимания контекста или незнакомых данных. Многие исследователи, включая Джеффа Хинтона, отца современной концепции глубокого машинного обучения, согласны с тем, что существующий подход к проектированию систем искусственного интеллекта больше не в состоянии справиться с насущными проблемами.

В этой книге мы обсуждаем жизнеспособную альтернативу традиционным методам глубокого машинного обучения – нейроэволюционные алгоритмы. *Нейроэволюция* – это семейство методов машинного обучения, которые используют эволюционные алгоритмы для облегчения решения сложных задач, таких как игры, робототехника и моделирование естественных процессов. Нейроэволюционные алгоритмы имитируют процесс естественного отбора. Очень простые искусственные нейронные сети могут стать очень сложными. Конечным результатом нейроэволюции является оптимальная топология сети, которая делает модель более энергоэффективной и удобной для анализа.

В этой книге вы узнаете о различных нейроэволюционных алгоритмах и получите практические навыки по их использованию для решения различных типовых задач информатики – от классического обучения с подкреплением до создания агентов для автономной навигации по лабиринту. Кроме того, вы узнаете, как можно применить нейроэволюцию для обучения глубоких нейронных сетей при создании агента, способного играть в классические игры Atari.

Цель этой книги – дать вам ясное представление о методах нейроэволюции, проводя различные эксперименты с пошаговым руководством. Она содержит практические примеры в таких областях, как игры, робототехника и моделирование естественных процессов, с использованием реальных наборов данных, чтобы помочь вам лучше понять исследуемые методы. После прочтения этой книги у вас будет все необходимое для применения методов нейроэволюции к другим задачам, по аналогии с представленными экспериментами.

При написании этой книги я стремился дать вам знания о передовых технологиях, которые являются жизненно важной альтернативой традиционному глубокому обучению. Я надеюсь, что применение нейроэволюционных алгоритмов в ваших проектах позволит вам элегантно и эффективно решать проблемы, которые казались неразрешимыми.

## Для кого написана эта книга

Эта книга предназначена для практиков машинного обучения, исследователей глубокого обучения и энтузиастов искусственного интеллекта, которые стремятся реализовать нейроэволюционные алгоритмы с нуля. Вы узнаете, как применять эти алгоритмы к различным задачам реального мира, как методы нейроэволюции могут оптимизировать процесс обучения искусственных нейронных сетей. Вы познакомитесь с основными понятиями нейроэволюции и получите необходимые практические навыки, чтобы использовать ее в своей работе и экспериментах. Знание Python, а также основ глубокого обучения и нейронных сетей является обязательным.

## О чем эта книга

*Глава 1* знакомит с основными понятиями генетических алгоритмов, такими как генетические операторы и схемы кодирования генома.

В *главе 2* обсуждаются практические аспекты методов нейроэволюции. В этой главе рассказано о достоинствах и недостатках популярных библиотек Python, которые предоставляют реализации алгоритма NEAT и его расширений.

*Глава 3* – это место, где вы начинаете экспериментировать с алгоритмом NEAT, реализуя решение для классической проблемы информатики.

В *главе 4* продолжаются эксперименты, связанные с классическими проблемами информатики в области обучения с подкреплением.

В *главе 5* вы продолжаете эксперименты с нейроэволюцией, пытаетесь создать решатель, способный найти выход из лабиринта. Вы узнаете, как реализовать симуляцию робота с набором датчиков для обнаружения препятствий и контроля его положения в лабиринте.

В *главе 6* вы воспользуетесь практическим опытом, полученным при создании решателя лабиринта в предыдущей главе, чтобы встать на путь создания более совершенного решателя.

*Глава 7* знакомит вас с передовыми методами нейроэволюции. Вы узнаете о схеме косвенного кодирования генома, в которой используются *сети, производящие составные паттерны* (Compositional Pattern Producing Network, CPPN) для кодирования топологий крупномасштабных искусственных нейросетей.

В *главе 8* вы узнаете, как выбрать конфигурацию модульной сетчатки, которая лучше всего подходит для конкретной проблемной области.

В *главе 9* мы обсуждаем, как широко распространенная в природе стратегия коэволюции может быть перенесена в область нейроэволюции.

*Глава 10* представляет вам концепцию глубокой нейроэволюции, которую можно использовать для обучения *глубоких искусственных нейронных сетей* (Deep Artificial Neural Network, DNN).

В *главе 11* рассказано, как начать работу над прикладной задачей, как настроить гиперпараметры нейроэволюционного алгоритма, как использовать передовые инструменты визуализации и какие показатели можно использовать для анализа выполнения алгоритма.

*Глава 12* обобщает все, что вы узнали в этой книге, и дает рекомендации о том, как продолжить свое самообразование.

## КАК ПОЛУЧИТЬ МАКСИМАЛЬНУЮ ОТДАЧУ ОТ ЭТОЙ КНИГИ

Для работы с примерами, представленными в этой книге, необходимы практические навыки программирования на языке Python. Для лучшего понимания исходного кода предпочтительно использовать IDE, которая поддерживает подсветку синтаксиса Python и ссылки внутри кода. Если у вас нет специализированной среды разработки, вы можете использовать инструмент разработки Microsoft Visual Studio Code. Он бесплатный и кросс-платформенный, и вы можете свободно скачать его по адресу <https://code.visualstudio.com>.

Python и большинство библиотек, которые мы обсуждаем в этой книге, являются кросс-платформенными и совместимы с Windows, Linux и macOS. Все описанные в книге эксперименты выполняются из командной строки, поэтому ознакомьтесь с приложением консоли терминала, установленным в выбранной вами ОС.

Для выполнения эксперимента, описанного в главе 10, вам необходим доступ к современному ПК с графическим ускорителем Nvidia GeForce GTX 1080Ti или выше. Этот эксперимент также лучше проводить в среде Ubuntu Linux. Ubuntu – это современная и мощная ОС на базе Linux. Навык работы с ней вам очень пригодится.

## СКАЧИВАНИЕ ИСХОДНОГО КОДА ПРИМЕРОВ

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте [www.dmkpress.com](http://www.dmkpress.com) на странице с описанием соответствующей книги.

## СКАЧИВАНИЕ ЦВЕТНЫХ ИЛЛЮСТРАЦИЙ

Мы предоставляем файл PDF с цветными изображениями скриншотов и рисунков, используемых в этой книге. Вы можете скачать его по адресу: [https://static.packt-cdn.com/downloads/9781838824914\\_ColorImages.pdf](https://static.packt-cdn.com/downloads/9781838824914_ColorImages.pdf)

## УСЛОВНЫЕ ОБОЗНАЧЕНИЯ И СОГЛАШЕНИЯ, ПРИНЯТЫЕ В КНИГЕ

В книге используются следующие типографские соглашения.

*Курсив* используется для смыслового выделения новых терминов, адресов электронной почты, а также имен и расширений файлов.

Моноширинный шрифт применяется для листингов программ, а также в обычном тексте для обозначения имен переменных, функций, типов, объектов, баз данных, переменных среды, операторов и ключевых слов.

**Моноширинный полужирный шрифт** используется для обозначения команд или фрагментов текста, которые пользователь должен ввести дословно без изменений.

*Моноширинный курсив* – для обозначения в исходном коде или в командах шаблонных меток-заполнителей, которые должны быть заменены соответствующими контексту реальными значениями.

# Часть I

---

## ОСНОВЫ ЭВОЛЮЦИОННЫХ ВЫЧИСЛИТЕЛЬНЫХ АЛГОРИТМОВ И МЕТОДОВ НЕЙРОЭВОЛЮЦИИ

В первой части книги представлены основные понятия эволюционных вычислений и обсуждаются особенности алгоритмов на основе нейроэволюции и перечень библиотек Python, которые могут использоваться для их реализации. Вы познакомитесь с основами методов нейроэволюции и получите практические рекомендации о том, как начать свои эксперименты. Здесь также приведено краткое знакомство с менеджером пакетов Anaconda для Python в рамках настройки вашей среды разработки.

Эта часть книги состоит из следующих глав:

- главы 1 «Обзор методов нейроэволюции»;
- главы 2 «Библиотеки Python и настройка среды разработки».





# Глава 1

---

## Обзор методов нейроэволюции

Концепция *искусственной нейронной сети* (artificial neural networks, ANN), которую мы дальше для удобства будем называть просто *нейросетью*, основана на структуре человеческого мозга. Существует устойчивое убеждение, что если суметь достоверно скопировать эту сложнейшую структуру, то можно создать *искусственный интеллект* (artificial intellect, AI). Мы все еще на пути к достижению этой цели. Хотя нам по силам создание специализированных AI-агентов, мы еще далеки от создания универсального искусственного интеллекта.

В этой главе вы познакомитесь с понятием искусственных нейросетей и двумя методами, которые мы можем использовать для их обучения (градиентный спуск с обратным распространением ошибки и нейроэволюция), чтобы нейросеть научилась приближаться к целевой функции. Тем не менее мы сосредоточимся в основном на обсуждении семейства алгоритмов на основе нейроэволюции. Вы узнаете о реализации эволюционного процесса, основанного на естественной эволюции, и познакомитесь с наиболее популярными алгоритмами нейроэволюции: NEAT, HyperNEAT и ES-HyperNEAT. Мы также обсудим методы оптимизации, которые можно использовать для поиска окончательных решений, и проведем сравнение между алгоритмами приближения к цели и алгоритмом поиска новизны. К концу этой главы вы будете иметь полное представление о внутреннем устройстве алгоритмов нейроэволюции и будете готовы применить эти знания на практике.

В этой главе мы рассмотрим следующие темы:

- эволюционные алгоритмы и нейроэволюционные методы;
- обзор алгоритма NEAT;
- NEAT на основе гиперкуба;
- HyperNEAT с развивающимся субстратом;
- метод оптимизации на основе поиска новизны.

## 1.1 ЭВОЛЮЦИОННЫЕ АЛГОРИТМЫ И НЕЙРОЭВОЛЮЦИОННЫЕ МЕТОДЫ

Термин «искусственная нейронная сеть» обозначает граф *узлов*, соединенных *связями*, где каждая из связей имеет определенный *вес*. Узел нейросети является своего рода пороговым оператором, который позволяет сигналу проходить дальше только после срабатывания определенной функции активации. Это отдаленно напоминает принцип, по которому организованы нейроны головного мозга. Как правило, процесс обучения нейросети состоит из выбора подходящих значений веса для всех связей в сети. Таким образом, нейросеть способна аппроксимировать любую функцию и может рассматриваться как *универсальный аппроксиматор*, который определяется теоремой универсальной аппроксимации.

Чтобы познакомиться с доказательством теоремы универсальной аппроксимации, прочтите следующие статьи:

- *Cybenko G.* (1989) Approximations by Superpositions of Sigmoidal Functions, *Mathematics of Control, Signals, and Systems*, 2 (4), 303–314;
- *Leshno Moshe, Lin Vladimir Ya., Pinkus Allan, Schocken Shimon* (January 1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*. 6 (6): 861–867. doi:10.1016/S0893-6080(05)80131-5 (<https://www.sciencedirect.com/science/article/abs/pii/S0893608005801315?via%3Dihub>);
- *Kurt Hornik* (1991). Approximation Capabilities of Multilayer Feedforward Networks, *Neural Networks*, 4 (2), 251–257. doi:10.1016/0893-6080(91)90009-T (<https://www.sciencedirect.com/science/article/abs/pii/089360809190009T?via%3Dihub>);
- *Hanin B.* (2018). Approximating Continuous Functions by ReLU Nets of Minimal Width. arXiv preprint arXiv:1710.11278 (<https://arxiv.org/abs/1710.11278>).

За последние 70 лет было придумано много методов обучения нейросетей. Однако наиболее популярная техника, получившая известность в последнем десятилетии, была предложена Джеффри Хинтоном. Она основана на обратном распространении ошибки прогнозирования через сеть с различными методами оптимизации, построенными на основе градиентного спуска функции потерь по отношению к весам связей между узлами сети. Этот метод демонстрирует выдающуюся эффективность обучения глубоких нейронных сетей для задач, связанных в основном с распознаванием образов. Однако, несмотря на присущие ему достоинства, он имеет существенные недостатки. Один из этих недостатков заключается в том, что для усвоения чего-то полезного из определенного набора данных требуется огромное количество обучающих образцов. Другим существенным недостатком является фиксированная архитектура нейросети, созданная экспериментатором вручную, что приводит к неэффективному использованию вычислительных ресурсов. Это связано с тем, что значительное количество сетевых узлов не участвует

в процессе вывода. Кроме того, методы обратного распространения имеют проблемы с передачей полученных знаний в другие смежные области.

Наряду с методами обратного распространения применяются очень многообещающие эволюционные алгоритмы, которые могут решать вышеупомянутые проблемы. Эти основанные на биологии методы черпают вдохновение из теории эволюции Дарвина и используют принципы эволюции видов для создания искусственных нейронных сетей. Основная идея нейроэволюции состоит в том, чтобы создавать нейросеть с помощью стохастических методов поиска, основанных на популяции. Используя эволюционный подход, можно разработать оптимальные архитектуры нейронных сетей, которые точно решают конкретные задачи. В результате могут быть созданы компактные и энергоэффективные сети с умеренными требованиями к вычислительной мощности. Процесс эволюции реализуется путем применения генетических операторов (мутация, кроссовер) к популяции хромосом (генетически закодированные представления нейросетей или решений) на протяжении многих поколений. Большие надежды на этот метод основаны на том, что в природных биологических системах каждое последующее поколение становится все более приспособленным к внешним обстоятельствам, которые можно выразить целевой функцией, то есть они становятся лучшими приближениями целевой функции.

Далее мы обсудим основные понятия генетических алгоритмов. Вам достаточно будет иметь умеренный уровень понимания принципов работы генетических алгоритмов.

### 1.1.1 Генетические операторы

*Генетические операторы* находятся в самом сердце каждого эволюционного алгоритма, и от них зависит результативность любого нейроэволюционного алгоритма. Существует два основных генетических оператора: *мутация* и *кроссовер* (*рекомбинация*).

В этой главе вы узнаете об основах генетических алгоритмов и о том, как они отличаются от обычных алгоритмов, в которых для обучения нейросети используются методы обратного распространения ошибок.

#### Оператор мутации

Оператор мутации выполняет важную роль сохранения генетического разнообразия популяции в процессе эволюции и предотвращает остановку в локальных минимумах, когда хромосомы организмов в популяции становятся слишком похожими. Эта мутация изменяет один или несколько генов в хромосоме в соответствии с вероятностью мутации, определенной экспериментатором. Вводя случайные изменения в хромосому *решателя* (*solver*), мутация позволяет эволюционному процессу исследовать новые области в пространстве поиска возможных решений и находить все лучшие и лучшие решения на протяжении поколений.

На рис. 1.1 показаны распространенные типы операторов мутации.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

[e-Univers.ru](http://e-Univers.ru)