

Содержание

Автор	15
Технический рецензент	16
Благодарности	17
Вступление	18
Часть I. ОСНОВЫ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	23
Глава 1. Знакомство с интеллектуальными системами	24
Элементы интеллектуальной системы	25
Пример интеллектуальной системы	26
Умный тостер	26
Использование данных	27
Датчики и эвристический интеллект	29
Тостер с машинным обучением	30
Создание интеллектуальной системы	31
Итог главы	32
Темы для размышлений	32
Глава 2. Анализ применимости интеллектуальных систем	34
Типы задач, для которых нужны интеллектуальные системы	34
Масштабные задачи	35
Открытые задачи	35
Меняющиеся задачи	36
Сложные задачи	36
Ситуации, когда интеллектуальные системы успешны	37
Когда неполная система жизнеспособна и полезна	37
Обратная связь для улучшения интеллекта	38
Когда система способна влиять на результат	38
Когда интеллектуальная система окупается	39
Действительно ли нужна интеллектуальная система?	40
Итог главы	41
Темы для размышлений	41
Глава 3. Краткие основы работы с данными	42
Структурированные данные	42

Задавайте данным простые вопросы.....	44
Работа с моделями данных	45
Концепция машинного обучения.....	46
Распространенные ошибки при работе с данными	48
Нарушение доверительных интервалов	48
Зашумленность данных	48
Смещение данных	48
Устаревание данных	49
Необоснованное использование данных.....	49
Итог главы	49
Темы для размышлений.....	50
Глава 4. Определение целей интеллектуальной системы.....	51
Признаки хорошей цели	51
Пример затруднений при выборе цели	52
Типы целей.....	53
Организационные цели	54
Опережающие показатели.....	54
Результаты пользователя.....	56
Свойства модели.....	57
Расслоение целей.....	58
Способы измерить успех.....	58
Ожидание дополнительной информации	58
А/В-тестирование	59
Ручная маркировка	59
Опрос пользователей	60
Разделение задач.....	61
Сохраняйте актуальность целей	61
Итог главы	62
Темы для размышлений.....	62
Часть II. ИНТЕЛЛЕКТУАЛЬНЫЙ ОПЫТ.....	63
Глава 5. Компоненты интеллектуального опыта	64
Представление интеллекта пользователю.....	64
Пример представления интеллекта	66
Достижение целей системы	67
Пример достижения целей	68
Минимизация последствий ошибок интеллекта	69
Получение данных для расширения системы	70
Пример сбора данных	70
Итог главы	71
Темы для размышлений.....	72

Глава 6. Затруднения при разработке интеллектуального опыта	73
Интеллект делает ошибки	73
Безумные ошибки интеллекта	75
Интеллект совершает разные ошибки	76
Переменчивый интеллект	78
Человеческий фактор	79
Итог главы	80
Темы для размышлений	81
Глава 7. Разработка эффективного интеллектуального опыта	82
Действенность опыта	83
Частота взаимодействия	84
Выгода от взаимодействия	86
Цена взаимодействия	87
Обнаружение ошибки	87
Исправление ошибки	88
Качество интеллекта	89
Итог главы	90
Темы для размышлений	91
Глава 8. Режимы интеллектуального взаимодействия	92
Автоматизация действий	92
Запросы и подсказки	93
Организованная информация	94
Аннотации	96
Гибридный опыт	97
Итог главы	98
Темы для размышлений	99
Глава 9. Извлечение данных из опыта	100
Пример: TeamMaker	101
Прямое вмешательство	101
Увлекательное взаимодействие	102
Связь с результатами	102
Свойства хороших данных	103
Контекст, действия и результаты	103
Достоверный охват	104
Реальное применение	105
Отсутствие смещения	105
Отсутствие петель обратной связи	106
Масштаб	106
Правильное толкование данных	107
Скрытые наблюдения	108

Пользовательские рейтинги	108
Отчеты о проблемах	109
Эскалации	109
Пользовательские решения	109
Итог главы	110
Темы для размышлений	111
Глава 10. Проверка интеллектуального опыта	112
Достижение ожидаемого опыта	112
Работа с контекстом	113
Работа с интеллектом	114
Промежуточный итог	115
Достижение целей	115
Непрерывная проверка	116
Итоги главы	117
Темы для размышлений	118
Часть III. РЕАЛИЗАЦИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	119
Глава 11. Компоненты реализации интеллекта	120
Пример реализации интеллектуальной системы	120
Компоненты реализации интеллектуальной системы	123
Среда выполнения интеллекта	123
Распределение и доставка интеллекта	124
Канал интеллектуальной телеметрии	124
Среда разработки интеллекта	125
Оркестровка интеллекта	125
Итог главы	126
Темы для размышлений	126
Глава 12. Среда выполнения интеллекта	128
Сбор контекста	129
Извлечение признаков	130
Обновление моделей	131
Выполнение моделей	132
Результаты	132
Нестабильность в интеллекте	133
API интеллекта	133
Итог главы	134
Темы для размышлений	135
Глава 13. Где расположить интеллект?	136
Соображения по размещению интеллекта	136
Задержка при обновлении	137

Задержка выполнения.....	139
Стоимость эксплуатации	140
Автономная работа.....	142
Подходы к размещению интеллекта.....	142
Статический интеллект в составе продукта	142
Интеллект на стороне клиента	143
Интеллект на стороне сервера	144
Внутренний (кешируемый) интеллект.....	145
Гибридный интеллект	146
Итог главы	146
Темы для размышлений.....	147
Глава 14. Управление интеллектом	148
Механизм управления интеллектом.....	148
Сложная архитектура систем.....	149
Высокая частота обновления	149
Человеческий фактор	149
Проверка работоспособности интеллекта	150
Проверка на совместимость	150
Проверка ограничений на выполнение	151
Проверка на очевидные ошибки	151
Пробный запуск интеллекта	152
Однократное развертывание интеллекта.....	152
Тихий интеллект.....	153
Ограниченное развертывание.....	154
Флайтинг	154
Отмена обновления.....	155
Итог главы	156
Темы для размышлений.....	156
Глава 15. Интеллектуальная телеметрия	158
Зачем нужна телеметрия	158
Проверка текущей работоспособности.....	158
Проверка результатов пользователей.....	159
Сбор данных для развития интеллекта.....	160
Свойства эффективной телеметрии.....	161
Выборочное наблюдение	161
Резюмирование	162
Гибкий таргетинг.....	163
Общие проблемы	163
Смещение данных	163
Пропуск редких событий	164
Завышение значимости	165

Нарушение конфиденциальности.....	165
Итог главы	166
Темы для размышлений.....	167
Часть IV. СОЗДАНИЕ ИНТЕЛЛЕКТА.....	168
Глава 16. Общее представление об интеллекте	169
Пример интеллекта	169
Контексты.....	170
Реализация в среде выполнения	171
Доступность контекста для разработчика	172
Что может предсказать интеллект.....	173
Классификация	173
Оценка вероятности	174
Регрессия.....	175
Ранжирование.....	176
Составное предсказание	176
Итог главы	176
Темы для размышлений.....	177
Глава 17. Представление интеллекта	178
Критерии выбора представления интеллекта.....	178
Представление интеллекта в виде программы	179
Представление интеллекта в таблицах соответствий.....	180
Представление интеллекта в моделях	181
Линейные модели.....	182
Деревья решений.....	183
Нейронные сети.....	184
Итог главы	186
Темы для размышлений.....	186
Глава 18. Процесс создания интеллекта	187
Пример создания интеллекта	188
Понимание задачи и окружения	188
Определение критериев успеха.....	190
Получение данных.....	190
Данные для начального запуска.....	191
Данные из взаимодействий.....	192
Подготовка инструментов оценки	192
Простой эвристический интеллект	193
Машинное обучение.....	194
Поиск компромиссов.....	195
Оценка и повторение	195

Уровни зрелости интеллекта	196
Мастерство создания интеллекта	196
Анализ и отладка данных	197
Отладка на основе проверок и оценок	197
Интуитивное знание инструментария	197
Математика – нужна ли она?	198
Итог главы	198
Темы для размышлений	199
Глава 19. Оценка интеллекта	200
Оценка точности	200
Обобщение	201
Типы ошибок	201
Распределение ошибок	204
Оценка других типов прогнозов	204
Оценка регрессий	204
Оценка вероятностей	205
Оценка ранжирования	205
Использование данных для оценки	206
Независимые оценочные данные	206
Независимость на практике	207
Оценка для подгрупп населения	208
Приемлемый объем данных	210
Сравнение интеллектов	211
Рабочие точки	211
Кривые	212
Субъективные оценки	212
Изучение ошибок	213
Переосмысление опыта пользователя	214
Предсказание худшей ситуации	214
Итог главы	215
Темы для размышлений	216
Глава 20. Машинное обучение интеллекта	217
Как работает машинное обучение	217
Плюсы и минусы сложности	219
Недообучение	219
Переобучение	220
Поиск разумного компромисса	220
Конструирование признаков	221
Преобразование данных в удобную форму	221
Содействие модели в использовании данных	223
Нормализация значений признаков	224

Выявление скрытой информации.....	224
Расширение контекста.....	225
Устранение лишних признаков.....	225
Моделирование.....	226
Параметры сложности.....	226
Выявление переобучения.....	227
Итог главы.....	229
Темы для размышлений.....	229
Глава 21. Структурирование интеллекта.....	230
Причины структурирования интеллекта.....	230
Свойства правильно структурированного интеллекта.....	231
Способы структурирования интеллекта.....	232
Разделение признаков.....	232
Конкурентный поиск моделей.....	234
Распределение ошибок.....	235
Метамодель.....	235
Секвенированная модель.....	237
Разделение по контекстам.....	238
Замещение.....	239
Итог главы.....	240
Темы для размышлений.....	241
Часть V. ОРКЕСТРОВКА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ.....	242
Глава 22. Понятие оркестровки интеллекта.....	243
Что такое хорошая оркестровка.....	244
Зачем нужна оркестровка.....	244
Изменение цели.....	245
Изменение пользователей.....	245
Изменение проблем.....	246
Изменение интеллекта.....	247
Изменение затрат.....	247
Злоупотребления.....	248
Команда оркестровки.....	248
Итог главы.....	249
Темы для размышлений.....	249
Глава 23. Среда оркестровки интеллекта.....	251
Мониторинг критериев успеха.....	251
Изучение взаимодействий.....	253
Оптимизация опыта.....	254
Переопределение интеллекта.....	255
Создание интеллекта.....	256

Критерии инвестирования в создание интеллекта.....	257
Итог главы.....	257
Темы для размышлений.....	258
Глава 24. Работа над ошибками.....	259
Худшее, что могло случиться.....	259
Причины выхода системы из строя.....	260
Отказы системы.....	261
Отказы модели.....	261
Ошибки интеллекта.....	262
Деградация интеллекта.....	262
Снижение количества ошибок.....	263
Инвестиции в развитие интеллекта.....	263
Настройка значимости опыта.....	264
Установка ограничителей.....	264
Переопределение ошибок.....	265
Итог главы.....	265
Темы для размышлений.....	266
Глава 25. Злоумышленники и злоупотребления.....	267
Злоупотребления – это бизнес.....	267
Масштабы злоупотреблений.....	268
Оценка вашего риска.....	269
Признаки злоупотреблений.....	270
Способы борьбы со злоупотреблениями.....	270
Увеличение стоимости продукта.....	270
Снижение привлекательности для злоумышленников.....	271
Машинное обучение против злоумышленников.....	271
Отключение злоумышленника от системы.....	271
Итог главы.....	272
Темы для размышлений.....	272
Глава 26. В шаге от собственной интеллектуальной системы.....	273
Контрольный список разработчика.....	273
Подход к проекту интеллектуальной системы.....	274
Планирование интеллектуального опыта.....	274
Планирование внедрения интеллектуальной системы.....	276
Подготовка к созданию интеллекта.....	278
Управляйте вашей интеллектуальной системой.....	279
Итог главы.....	280
Темы для размышлений.....	280
Предметный указатель.....	281

Автор

Джефф Хултен (Geoff Hulten) – учёный, кандидат наук в области машинного обучения. Более десяти лет он руководил рабочими группами по прикладному машинному обучению, создавая десятки интеллектуальных систем в масштабе интернета, в которых происходят сотни миллионов взаимодействий с пользователями каждый день. Его исследования были представлены на ведущих международных конференциях, получили тысячи ссылок, выиграла награду **SIGKDD Test of Time** за значительный вклад в сообщество исследователей интеллектуального анализа данных и выдержали испытание временем.

Технический рецензент

Джеб Хейбер (Jeb Haber) имеет степень бакалавра информатики Университета Уилламетт (Willamette University). Он провел почти два десятилетия в Microsoft, работая над различными проектами в Windows, Internet Explorer, Office и MSN. За последние десять с лишним лет своей карьеры в Microsoft Джеб возглавлял команду управления проектами, отвечающую за службы безопасности и защиты, предоставляемые Microsoft SmartScreen (защита от фишинга, вредоносных программ и прочих киберугроз).

Команда Джеба разработала и управляла интеллектуальными системами глобального масштаба с сотнями миллионов пользователей. Его работа включала в себя разработку идеологии продукта, планирование, стратегию, управление проектами, определение метрик и развитие команды. Наряду с разработкой систем и процессов, необходимых для построения и запуска в глобальном масштабе интеллектуальных и оценочных систем в режиме 24×7, Джеб помог сформировать культуру целой отрасли. В настоящее время Джеб является президентом наблюдательных советов в двух некоммерческих организациях, помогающих отдельным лицам и семьям, страдающим редким генетическим расстройством – фенилкетонурией (ФКУ).

Благодарности

Меня окружает множество людей, ставших частью мира интеллектуальных систем, над которыми я работал на протяжении многих лет. Эти люди помогли мне многому научиться и многое понять. В частности, я хотел бы поблагодарить:

Джеба Хейбера и Джона Скарроу (John Scarrow) за то, что они были в числе ключевых умов при разработке концепций, описанных в этой книге, и за то, что они были отличными коллегами на протяжении многих лет. Ничего из этого не произошло бы без их руководства и преданности делу.

Благодарю вас, Энтони П. (Anthony P.), Томаш К. (Tomasz K.), Роб С. (Rob S.), Роб М. (Rob M.), Дейв Д. (Dave D.), Кайл К. (Kyle K.), Эрик Р. (Eric R.), Амейя Б. (Ameya B.), Крис И. (Kris I.), Джефф М. (Jeff M.), Майк С. (Mike C.), Шанкар С. (Shankar S.), Роберт Р. (Robert R.), Крис Дж. (Chris J.), Сьюзен Х. (Susan H.), Иван О. (Ivan O.), Чад М. (Chad M.), и многих других, кто не вошел в этот список.

Вступление

«Разработка интеллектуальных систем» – это книга о том, как использовать машинное обучение на практике.

Она охватывает все, что вам нужно для создания полностью функционирующей интеллектуальной системы, которая использует машинное обучение и данные взаимодействия с пользователем для непрерывного улучшения и достижения поставленной цели.

Прочитав эту книгу, вы сможете выполнить разработку интеллектуальной системы от начала и до конца.

Вы будете знать:

- когда следует использовать интеллектуальную систему и как сделать так, чтобы она достигла ваших целей;
- как разработать эффективные взаимодействия между пользователями и интеллектуальными системами;
- как внедрить интеллектуальную систему на стороне клиента и на сервере;
- как построить интеллект, который является сердцем интеллектуальной системы, и развивать его с течением времени;
- как управлять интеллектуальной системой в течение ее жизненного цикла.

Вы также поймете, как с наибольшей отдачей приложить свои усилия в области разработки программного обеспечения, обработки данных, машинного обучения и управления проектами.

Существует множество замечательных книг, которые учат работе с данными и навыкам машинного обучения. Эти книги похожи на книги по языкам программирования – они преподносят ценные навыки в мельчайших деталях. Но данная книга больше похожа на книгу по разработке программного обеспечения – она учит, как использовать эти базовые навыки и создавать прикладные системы.

Эта книга основана на более чем десятилетнем опыте создания интеллектуальных систем в масштабе интернета, где каждый день происходят сотни миллионов взаимодействий пользователей с крупнейшими и наиболее важными вычислительными системами в мире. Я надеюсь, что книга поможет ускорить распространение систем, которые превращают данные в результаты, и поможет читателям развить практические навыки в этой важной области.

Для кого эта книга

Эта книга предназначена для всех, кто имеет образование в области информатики и хочет разобраться, что нужно для создания эффективных интеллектуальных систем.

Представьте себе типичного инженера-программиста, который назначен на проект машинного обучения. Он хочет узнать больше о новом направлении, поэтому читает технические книги, полные статистики, математики и описаний методов моделирования. Это важные знания, но они плохо помогают разработчику программного обеспечения включиться в новую работу. Правильнее будет начать с книги «Разработка интеллектуальных систем».

Представьте себе специалиста по машинному обучению, который должен понимать, как законченная система будет взаимодействовать с его моделями, на что он может рассчитывать и на что следует обращать внимание на практике. «Разработка интеллектуальных систем» написана для него.

Представьте себе технического директора, который хочет успешно внедрить машинное обучение на своем предприятии. Возможно, он наймет доктора наук в области машинного обучения. Спустя некоторое время специалист по машинному обучению принесет директору различные диаграммы, кривые точности/отклика и выборки обучающих данных, но не добавит понимания того, как это все внедрить на практике. «Разработка интеллектуальных систем» – самая подходящая книга для технического директора.

СПЕЦИАЛИСТЫ ПО ДАННЫМ И МАШИННОМУ ОБУЧЕНИЮ

Данные и машинное обучение лежат в основе многих интеллектуальных систем, но между разработкой рабочей модели (созданной с помощью машинного обучения) и возможным устойчивым влиянием на пользователя предстоит проделать невероятную работу. Понимание особенностей этой вспомогательной работы открывает перед вами несколько путей для улучшения моделей.

Во-первых, важно **понимать ограничения**, которые эти системы накладывают на ваши модели. Например, где будет работать модель? К каким данным она будет иметь доступ? Как быстро это будет происходить? Каково влияние ложноположительных решений на бизнес? А ложноотрицательных? Как настроить модель, чтобы добиться максимальной отдачи для бизнеса?

Во-вторых, важно иметь возможность **влиять на других участников проекта**. Способность влиять на инженеров и владельцев бизнеса поможет вам найти лучшие решения и максимально увеличить собственные шансы на успех. Например, вы не можете получать достаточный объем обучающих данных из-за нехватки данных телеметрии. Должны ли вы удвоить усилия по доработке модели, или более разумным будет инженерное решение проблемы? Или, может быть, от вас требуют невероятно высокой точности, хотя ваши модели уже работают с очень хорошей (но немного меньшей) точностью? Стоит ли бороться за сверхвысокую точность или лучше подумать над тем, как изменить пользовательский опыт таким образом, чтобы уменьшить влияние ошибок на пользователя?

В-третьих, важно понять, **какие вспомогательные элементы могут принести вам пользу**. Эскалации, ручные перезагрузки, телеметрия, ограниче-

ния, которые предотвращают серьезные ошибки, – все это инструменты, которые вы можете использовать. Вы должны понимать, когда их использовать и как интегрировать их в процесс моделирования. Стоит ли отказываться от модели, которая работает приемлемо для 99 % пользователей, но чрезвычайно плохо для 1 % пользователей? Или, может быть, следует искать решение проблемы в других частях системы?

ИНЖЕНЕРЫ-ПРОГРАММИСТЫ

Создание программного обеспечения, которое радует пользователей, – это трудная работа. Обойти это невозможно, за каждым успешным программным продуктом и услугой стоит серьезная разработка. Интеллектуальные системы обладают некоторыми уникальными свойствами, которые предъявляют интересные вызовы. В этой книге описаны концепции, позволяющие разрабатывать эффективные и надежные интеллектуальные системы и лучше всего раскрывающие возможности машинного обучения и науки о данных.

Во-первых, эта книга дает определение **сущностей и абстракций**, которые должны присутствовать в успешной интеллектуальной системе. Вы изучите понятия, лежащие в основе среды выполнения интеллекта, контекста и функций, моделей, телеметрии, данных обучения, управления интеллектом, комбинирования и многого другого.

Во-вторых, книга даст вам **концептуальное понимание машинного обучения и науки о данных**. Она подготовит вас к продуктивному обсуждению компромиссов между трудозатратами на разработку и моделирование – когда пара часов вашей работы действительно помогает найти решение и когда вас просят о невозможном, чтобы сэкономить немного времени на моделировании.

В-третьих, в книге будут **рассмотрены шаблоны интеллектуальных систем**, которые мы с коллегами разработали и внедрили за минувшее десятилетие. Каковы плюсы и минусы выполнения интеллекта на стороне клиента и во внешнем сервисе? Как связывать и проверять вероятностные компоненты? Что нужно включить в телеметрию, чтобы система могла развиваться?

РУКОВОДИТЕЛИ ПРОГРАММ

Машинное обучение и наука о данных сегодня на пике популярности. Это фантастические инструменты, но все же это не более чем инструменты; они не являются решениями. Эта книга даст вам концептуальное понимание того, чем эти инструменты хороши и как их использовать для решения ваших бизнес-задач.



Первое, чему вы научитесь, – это **интуитивному чутью, когда уместны машинное обучение и наука о данных**. Для руководителя программы нет ничего хуже, чем пытаться вбить квадратный колышек в круглое отверстие. Вы

должны хорошо понимать, какие проблемы можно решить с помощью машинного обучения. Но, что не менее важно, вы должны понимать, какие проблемы *невозможно* или, по крайней мере, нелегко решить с его помощью. В этой области деятельности много участников, и они говорят на таких разных и сложных технических языках, что им трудно понять друг друга. Эта книга поможет вам научиться задавать правильные вопросы и понимать ответы.

Второй навык – чутье окупаемости инвестиций, чтобы определить, **насколько дорогостоящую интеллектуальную систему использовать**. Понимая реальные затраты на создание и обслуживание системы, которая превращает данные во влияние на пользователя, вы можете выбрать оптимальный момент для начала разработки и внедрения. Вы войдете в эту область с открытыми глазами и будете иметь навык оценки инвестиций, достаточный для успешной работы. Иногда востребованы все элементы интеллектуальной системы, описанные в этой книге, но иногда правильным выбором для бизнеса бывает менее сложная система. Эта книга поможет вам принимать правильные решения и обоснованно отстаивать их.

Наконец, третье, чему научится руководитель программы, – как **планировать, укомплектовывать персонал и управлять проектом интеллектуальной системы**. Воспользуйтесь нашим опытом построения множества масштабных интеллектуальных систем в следующих ключевых вопросах: жизненный цикл интеллектуальной системы; пошаговый процесс ее запуска; команда и навыки, необходимые для успеха.

ПРИНЯТЫЕ В ТЕКСТЕ СОГЛАШЕНИЯ

-  Так будут оформляться советы и рекомендации.
-  Так будут оформляться примечания.

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, 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 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/ или напишите в издательство по адресу www.dmkpress.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 данной книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу www.dmkpress.com, и мы исправим это в следующих тиражах.

НАРУШЕНИЕ АВТОРСКИХ ПРАВ

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Packt очень серьезно относятся 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 принять меры.

Пожалуйста, свяжитесь с нами по адресу электронной почты www.dmkpress.com со ссылкой на подозрительные материалы.

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

Часть I

ОСНОВЫ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Главы 1–4 закладывают основу для понимания интеллектуальных систем. Эта часть книги объясняет, что такое интеллектуальные системы и для чего они применяются. В ней рассказано, как удостовериться, что интеллектуальная система имеет полезную и достижимую цель, а также дается обзор некоторых проблем и методов реализации интеллектуальных систем.

Глава 1

Знакомство с интеллектуальными системами

Мы окружены *интеллектуальными системами* (intelligent systems). Они встроены в наши лампочки, автомобили, часы, термостаты и компьютеры. Как эти системы улучшают нашу жизнь? Нравится ли это нам?

Когда должна включиться лампочка? Когда на сайте магазина следует показать определенный товар? Когда поисковая система должна показать нам ссылку на сайт? Когда динамик должен воспроизвести звук?

В ответах на подобные вопросы (точнее, в достаточно хороших ответах) и содержится главная ценность интеллектуальных систем. Это трудная задача.

У некоторых крупнейших и наиболее дорогих компаний в мире весь бизнес, по сути, состоит из хороших ответов на простые вопросы:

- Какую страницу я должен показать в ответ на запрос?
- Какую рекламу показывать на странице сайта?
- Какой продукт я должен показать этому покупателю?
- Какой фильм этот пользователь захочет смотреть прямо сейчас?
- Какую книгу хотел бы прочитать этот человек?
- Какие новости вызовут наибольший интерес?
- Какие программы я должен заблокировать, чтобы защитить компьютер?

Хорошие ответы на эти вопросы принесли компаниям миллиарды, а в некоторых случаях сотни миллиардов долларов. При этом жизнь множества людей стала счастливее, продуктивнее и безопаснее. Но это лишь верхушка айсберга.

Есть десятки тысяч подобных вопросов, на которые нам приходится искать ответ: когда должна открыться моя входная дверь? Каким должно быть следующее упражнение в фитнес-приложении? Какой должна быть новая песня композитора? Как должен развиваться сюжет игры, чтобы максимально вовлечь игрока?

Эта книга о правильном и эффективном использовании возможностей интеллектуальных систем.

ЭЛЕМЕНТЫ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ

Интеллектуальные системы – это мостик, соединяющий пользователей с искусственным интеллектом (машинным обучением) для достижения значимых целей. В интеллектуальной системе интеллект развивается и совершенствуется с течением времени, особенно в процессе наблюдения за взаимодействием пользователя с системой.

Успешные интеллектуальные системы обязательно содержат следующие компоненты:

- **значимая цель** (meaningful objective). Интеллектуальную систему не создают просто так. Должен быть обоснованный повод создать систему, которая имеет значение для пользователей и соответствует вашей цели – такой, которая достижима именно при помощи интеллектуальной системы. Но выбрать значимую цель совсем не просто. Первая часть этой книги поможет вам понять, что делают интеллектуальные системы, поэтому вы будете знать, в каких случаях следует использовать одну из систем и какие цели вы должны для нее установить;

- **интеллектуальный опыт** (intelligent experience). Система предоставляет пользователям выходные данные системного интеллекта (например, прогнозы на основе машинного обучения) и получает от пользователей обратную связь.

Интеллектуальный опыт связан с пользовательским интерфейсом, который подстраивается под прогнозы и позволяет интеллектуальной системе сработать с максимальной отдачей, когда прогноз оправдался, и в то же время сводит к минимуму стоимость ошибки. Интеллектуальный опыт должен накапливать как скрытые, так и явные отзывы пользователей, чтобы система могла со временем улучшить свой интеллект. Во второй части этой книги будет рассмотрен интеллектуальный опыт, его возможности и подводные камни, подстерегающие нас при взаимодействии между интеллектом и пользователем;

- **реализация** (implementation). Внедрение интеллектуальной системы включает в себя создание среды для работы интеллекта, размещение среды в нужном месте, управление интеллектом, получение интеллектуального опыта, сбор телеметрии, чтобы убедиться, что система функционирует, сбор информации, а также обратную связь с пользователем. Третья часть этой книги описывает все компоненты реализации интеллекта. Она подготовит вас к разработке и внедрению собственной интеллектуальной системы;

- **создание интеллекта** (intelligence creation). Интеллектуальная система – это интеллект, настроенный на максимальный успех. Такой интеллект можно сформировать разными способами, от простой эвристики до сложного машинного обучения. Интеллект следует формировать так, чтобы подходящие типы интеллекта решали нужные задачи и чтобы он

мог эффективно совершенствоваться группой людей в течение длительного времени. Четвертая часть этой книги освещает процесс создания и развития интеллекта для интеллектуальных систем интернет-масштаба. Вы научитесь использовать все доступные подходы с максимальной отдачей;

- **оркестровка** (orchestration). Все элементы интеллектуальной системы должны работать согласованно, как оркестр. Оркестровка включает в себя управление изменениями в системе, синхронизацию опыта с интеллектом, принятие решений о том, какую телеметрию собирать для отслеживания и устранения проблем, а также управление денежными средствами, которые можно потратить на создание и развертывание нового интеллекта. Сюда также входят устранение ошибок, контроль рисков и предотвращение злоупотреблений. В пятой части этой книги говорится о том, что нужно для организации интеллектуальной системы и достижения целей на всех этапах ее жизненного цикла.

- ☑ Интеллектуальная система является одним из способов применения машинного обучения на практике. Интеллектуальная система включает в себя интеллект, созданный с помощью машинного обучения и других подходов, и применяет его для достижения ваших целей. Интеллект совершенствуется в течение всего срока жизни системы.

ПРИМЕР ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ

Интеллектуальные системы могут быть использованы для построения поисковых систем, сайтов электронной коммерции, беспилотных автомобилей и систем компьютерного зрения, которые знают, кем является человек и когда он улыбается. Но это большие и сложные системы.

Мы рассмотрим более простой пример, чтобы увидеть, как решение бытовой задачи может превратиться в интеллектуальную систему.

Умный тостер

Давайте обсудим подключенный к интернету тостер. Хорошая идея? Может быть, да, а может быть, нет – не узнаем, пока не попробуем. У нашего тостера есть два элемента управления: ползунок, управляющий интенсивностью обжаривания, и рычаг, включающий тостер.

Поначалу задача не выглядит сложной: умный тостер должен преобразовать положение ползунка в длительность обжаривания. При слабой обжарке тостер работает, скажем, 30 секунд. При сильной обжарке тостер работает в течение двух минут. Вот и все.

А теперь придумайте фиксированную настройку длительности обжарки и отправьте тостер пользователям. Что может пойти не так?

Если вы выберете максимальную длительность, тост может сгореть. Большинство пользователей, которые используют эту настройку, будут ужасно недовольны, выбросят обугленный тост и начнут снова.

Представьте себе, сколько случится сбоев во время приготовления тостов, когда раздраженные пользователи стоят над тостерами и держат руку на рычаге, чтобы вовремя прервать обжарку. Или когда пользователи многократно поджаривают один и тот же кусок хлеба, чтобы довести его до нужной кондиции.

Это плохо. Если вы собираетесь успешно продавать тостеры, вы должны разработать действительно хороший тостер.

Поэтому, возможно, вы проведете предварительное тестирование, подбирая длительность обжарки, пока не получите оптимально приготовленный тост. Не слишком хрустящий, не слишком бледный.

Отлично.

Правильно ли вы поняли задачу? Будет ли этот тостер делать то, что хотят пользователи? Трудно сказать однозначно. Независимо от того, сколько тостов вы съели в вашей жизни, невозможно доказать, что вы правильно предусмотрели все типы тостов, которые могут захотеть сделать все ваши пользователи.

Теперь вы понимаете, что необходимо учитывать мнения и опыт других людей в процессе разработки тостеров. Но как?

Может быть, вы начнете с фокус-группы? Пригласите десятки любителей тостов, поместите их в тостерную лабораторию и делайте заметки, наблюдая за процессом поджаривания.

Затем снова настройте длительность обжарки, чтобы отразить результаты лабораторных исследований. Ну как, теперь у вас есть идеальный тостер? Сможет ли этот тостер, настроенный для фокус-группы, сделать правильный тост, который понравится сотням тысяч людей по всему миру?

Что, если кто-то положит в тостер замороженный хлеб? Или другой продукт из холодильника? Или кто-то любит зажарить тост дочерна? А что, если кто-то изобрел новые тосты, не похожие на все прежние тосты человечества? Подходит ли ваш тостер для всех этих ситуаций? Наверняка нет.

Использование данных

Пожалуй, сделать идеальный тостер оказалось немного сложнее, чем просто спросить людей, что им нравится.

У вас получается слишком много вариантов настроек, если вы хотите получить идеальный тост в любой мыслимой ситуации. Вы можете исследовать фокус-группы каждый день всю оставшуюся жизнь, но так и не увидите все типы тостов, которые способен сделать тостер.

Вам нужен другой подход. Настало время для серьезной науки о данных.

Тостер подключен к интернету, поэтому вы можете запрограммировать его для отправки данных телеметрии на ваш сервер. Каждый раз, когда кто-то поджаривает тост, вы можете узнать, какую настройку он использовал и сколько времени длилась обжарка.

Вы отправляете первую версию тостера клиентам (возможно, ограниченному кругу пользователей), и тостерная телеметрия начинает отправлять данные к вам на сервер.

Теперь вы точно знаете, какие настройки тостеров используют люди в своей реальной жизни (а не в какой-то надуманной лабораторной обстановке). Вы знаете, сколько раз люди нажимают на рычаг, чтобы начать обжарку, и сколько раз они поднимают рычаг, чтобы срочно отключить тостер.

Можете ли вы использовать эти данные, чтобы сделать тостер лучше?

Конечно!

Вы могли бы для начала ограничить максимальную интенсивность обжарки тем значением, которое фактически использует лишь несколько пользователей. Затем вы можете подобрать метрики, чтобы убедиться, что эта настройка не смещена в сторону чрезмерного поджаривания. Например, вы можете отслеживать процент тостов, которые преждевременно извлечены пользователями (предположительно потому, что они начали подгорать), и настраивать верхнюю границу обжарки, пока не исключите подобные ситуации.

Аналогично вы могли бы определить минимальную интенсивность обжарки, исходя из предпочтений пользователей. Вы можете отследить случаи двойной обжарки (когда кто-то повторно включает тостер сразу после первого цикла обжарки) и устранить склонность тостера к недостаточному поджариванию.

Замечательно – теперь вы можете даже настроить обжарку по умолчанию, которая находится в середине диапазона, то есть установить наиболее востребованную длительность обжарки тоста.

Поскольку ваши тостеры подключены к интернету, в них можно загружать новые настройки с вашего сервера. Вообще-то, вы можете менять настройки тостера хоть каждый день или дважды за воскресенье – мы живем в эпоху невиданных чудес!

В вашем подходе есть несколько допущений. Например, вы могли предположить, что поджаривание несколько раз подряд является признаком недостаточной обжарки, когда пользователь снова поджаривает один и тот же тост, а не пытается быстро приготовить несколько разных тостов.

Вы также могли предположить, что срочное отключение – это признак того, что хлеб начинает гореть, а не признак того, что пользователь опаздывает на работу и выбегает из дома.

Кроме того, когда вы загружаете в тостеры новые настройки, как вы можете быть уверены, что они понравятся пользователям? Вероятно, вы полагаете (основываясь на данных), что новые настройки лучше соответствуют запросам наибольшего количества пользователей.

Но как насчет пользователя, который еще вчера получал свой собственный идеально поджаренный тост, а сегодня внезапно получает ... нечто другое?

Несмотря на эти проблемы, у вас получился довольно приличный тостер. У него есть телеметрия, которая сообщает, что тостер примерно выполняет свою работу. Тостер можно обслуживать и улучшать с течением времени. А теперь окончательно отпустим фантазию с поводомка и придумаем по-настоящему крутую штуку.

Датчики и эвристический интеллект

Если вы хотите сделать лучший в мире тостер, вам не хватит регулятора обжарки и рычага включения. Придется добавить в тостер несколько датчиков:

- датчик веса, чтобы узнать, сколько тостов в тостере, и определить, когда клиент поместил что-то в тостер и когда он вынул содержимое;
- датчик температуры, чтобы узнать температуру предмета, помещенного в тостер;
- датчик геолокации, позволяющий узнать, в каком регионе мира находится тостер, чтобы он мог адаптироваться к различным вкусам в разных регионах;
- датчик приближения, чтобы узнать, когда кто-то подошел к тостеру, и камера, чтобы определить, кто это;
- часы, чтобы узнать, является тост завтраком или ужином;
- небольшую память, для того чтобы запомнить, что было недавно поджарено, и для отслеживания изменений в настройках;
- датчик дыма, чтобы узнать, когда тостер допустил серьезную ошибку и собирается что-то сжечь.

Теперь, когда пользователь подходит к тостеру и что-то кладет в него, тостер может посмотреть, кто подошел, попытаться угадать, что он пытается поджарить, и автоматически предложить настройку.

И вообще, если тостер настолько хорош, больше нет необходимости в настройке интенсивности обжарки или рычаге включения. Вы могли бы отправлять потребителям тостеры без кнопок, ручек и прочих органов управления. Пользователь бросает в тостер кусок хлеба, уходит и возвращается к восхитительному тосту!

Эх, остается всего лишь найти способ превратить все эти показания датчиков в правильную длительность обжарки для любого тоста...

Для этого вам необходим *интеллект* – программа, набор правил или машинно-обучаемая модель, которая принимает решения.

Давайте начнем с ручного подбора правил для интеллекта.

Возьмите блокнот и запишите набор *эвристических правил*, которые учитывают показания датчиков и рекомендации по интенсивности обжарки. Например: если в тостер положили что-то холодное и тяжелое, то жарить 5 минут с высокой интенсивностью. Но для каждого градуса выше нуля уменьшить время обжарки на 2 секунды. Но в Британии к обжарке прибавляют 15 секунд (потому что им это нравится). Но если вес и размер тоста соответствуют известному фирменному продукту, то следует выбрать «достаточную продолжительность» в соответствии с руководством изготовителя тостов.

Ну и так далее.

Каждый раз, когда пользователь жалуется, что тостер неправильно поджарил хлеб, вы можете добавить новое правило.

Каждый раз, когда телеметрия показывает всплеск двойных обжарок, вы можете настроить правила для улучшения обжарки.

Каждый раз, когда вы добавляете новое правило или обновляете старое, вы можете обновить все тостеры, которые вы продали по всему миру, попросив их загрузить новые настройки с вашего сервера.

При таком подходе вам, вероятно, потребуется написать и поддерживать множество правил, которые соответствуют всем возможным ситуациям. Это очень большая работа. Вы можете нанять дюжину сотрудников и дать им несколько месяцев для написания правил, но все равно не будете довольны результатом.

Тостер с машинным обучением

В подобных ситуациях, когда процесс слишком трудно или слишком дорого оптимизировать вручную, люди обращаются к *машинному обучению*.

На высоком уровне машинное обучение может отслеживать поведение владельцев интернет-тостеров и автоматически вырабатывать набор правил для управления тостером. Эти правила не отличаются от эвристических правил, которые вы записали в блокнот, разве что созданы машиной.

Но машины могут создавать гораздо больше правил, чем люди. Машины способны собирать данные с десятков датчиков и оптимизировать работу тостеров для тысяч различных пользователей одновременно. Они могут подключать новые данные и повторно оптимизировать систему каждый день, каждый час – иногда даже быстрее. Машины могут составить личные правила для каждого пользователя. Для работы машинному обучению нужны сведения о ситуации, с которой столкнулся пользователь (показания датчиков), какое действие предпринял пользователь (как он настроил тостер) и какой результат он получил (понравился ли тост, или пришлось что-то исправить и попробовать снова). Машинному обучению нужны примеры, когда дела идут хорошо, а также примеры, когда дела идут плохо, – как можно больше разных примеров.

В качестве примеров успешных операций вы можете взять выборки данных для случаев, когда пользователь положил хлеб в тостер, нажал на рычаг запуска, дождался завершения, вынул тост и ушел. Это те случаи, когда пользователь, вероятно, получил то, что хотел. У вас есть записи показаний всех датчиков и длительности обжаривания. Этот набор данных можно использовать для системы машинного обучения.

В качестве примеров неудачных операций имеет смысл взять выборки данных для случаев, когда пользователям приходилось возиться с тостером, преждевременно отключать тостер или многократно повторять обжарку одного тоста. Теперь у вас есть записи показаний всех датчиков и длительности обжаривания для неудачных операций.

Таким образом, располагая наборами данных для удачных и неудачных операций плюс машинное обучение, вы обучаете алгоритм управления тостером. Вы вводите данные и получаете программу, которая просматривает показания датчиков и определяет оптимальные настройки длительности обжарки.

Затем вы рассылаете эту обученную программу на тостеры ваших клиентов. Вы можете делать это каждый день, каждую минуту.

Вы извлекаете уроки из сотен тысяч взаимодействий с клиентами – и даже из сотен миллионов взаимодействий.

Это великолепно!

СОЗДАНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ

Создание эффективной интеллектуальной системы не обходится без пяти основных элементов: цели, опыта, реализации, интеллекта и оркестровки.

Вам нужна проблема, которая стоит затраченных усилий и решается с помощью интеллектуальной системы. Например, вы должны управлять тостером, основываясь на том, что решил интеллект. Вам придется создать систему, которая работает так, как нравится пользователю, предоставить ему удобные элементы управления и обратную связь, которую вы будете использовать для обучения интеллекта. Вам нужно разработать все сервисы, инструменты и код, который собирает телеметрию, генерирует интеллект, перемещает его туда, где он должен быть, и связывает его с пользователями.

Вам придется совершенствовать интеллект каждый день, снова и снова, что вполне предсказуемо. И вам нужно, чтобы все работало в течение долгого времени, так как на рынке появляются новые продукты для поджаривания в тостере, а вкусы людей меняются.

Вам нужно решить, какой объем данных телеметрии следует собирать, чтобы потенциальная ценность данных компенсировала эксплуатационные расходы. Также нужно решить, насколько сильно менять интеллект при обновлении, чтобы тостер улучшался, но не сбивал с толку и не расстраивал пользователей.

Вы должны отслеживать ключевые показатели и реагировать, если они начинают ухудшаться. Может быть, надо точнее настроить опыт? Может быть, надо срочно вмешаться в настройки интеллекта? И конечно же, вам придется иметь дело с ошибками, ибо они непременно случаются.

Интеллект (особенно машинно-обучаемый интеллект) может совершать грубые ошибки – зрелищные, нелогичные, наносящие большой ущерб клиентам.

Например, ваш тостер может узнать, что люди в определенном районе любят зажаренный до угольков хлеб. Не важно, чего пользователь хочет от тостера, но если он живет в этом районе, то получит угли. Поэтому вам нужны способы выявления и устранения подобных ошибок. Иногда приходится ставить заграждения на пути машинного обучения.

И к сожалению, приходится признать, что... люди есть люди. Каждый раз, когда вы используете обратную связь от пользователя для оптимизации процесса, например используете машинное обучение для создания тостера, вы должны думать о том, как человек может извлечь выгоду из того, что что-то идет не так.

Представьте себе, что крупный производитель хлеба платит наемным работникам за то, чтобы они жарили хлеб конкурента, используя аномальные настройки. Наемники станут жарить миллионы тостов в неделю, наводняя вашу систему телеметрии ложными данными. Машинное обучение воспользуется этими данными и научит интеллект плохим вещам – ваши тостеры будут регулярно портить продукты конкурента, что приведет, как минимум, к проблемам с безопасностью пищевых продуктов.

Это очень плохо.

Успешная интеллектуальная система должна учитывать все эти проблемы и многое другое.

Итог главы

В этой главе были представлены интеллектуальные системы, а также пять концептуальных задач, которые необходимо решать при разработке интеллектуальной системы: цель, опыт, реализация, интеллект и оркестровка.

Теперь вам должно быть ясно, что существует множество способов решения этих задач и что они тесно взаимосвязаны. Чтобы получить успешную интеллектуальную систему, вы должны применять сбалансированный подход. Если в вашем контексте особенно сложна одна из концептуальных задач, придется пересмотреть другие задачи, чтобы компенсировать перекос.

Например, если вы пытаетесь внедрить интеллектуальную систему в существующую систему управления, в которой опыт уже определен (и не может быть изменен), возможно, имеет смысл установить менее сложную цель, чтобы вкладывать больше средств в интеллект, или иметь более продуманную стратегию смягчения ошибок.

Но можно взглянуть на ситуацию и с другой стороны: есть разные пути для достижения успеха.

Вы развернули интеллектуальную систему и обнаружили, что проблема с интеллектом сложнее, чем вы думали? Не надо паниковать. Существует множество способов компенсировать дисбаланс интеллекта и создать систему, которая радует клиентов и помогает вашему бизнесу прямо сейчас, в то время как для развития интеллекта требуется время.

Остальная часть этой книги даст вам инструменты для уверенной работы над проектами интеллектуальных систем.

Темы для размышлений

Прочитав эту главу, вы сможете:

- распознать интеллектуальные системы в окружающем мире;
- увидеть возможности, которые открывают интеллектуальные системы;
- понять разницу между интеллектом (который делает прогнозы) и интеллектуальной системой (которая сочетает в себе цель, опыт, реализацию, интеллект и управление для достижения результатов);

- сформулировать все концептуальные задачи, которые вам нужно будет решить, чтобы построить успешную интеллектуальную систему;
- понять, как взаимодействуют сложные элементы системы, включая компромиссы и способы, которыми они могут дополнять друг друга.

Постарайтесь ответить на следующие вопросы:

- Какие сервисы вы используете, которые (как вы думаете) созданы путем превращения массива данных клиентов в интеллект?
- Какой самый лучший опыт вы получили с одним из этих сервисов?
- Какой у вас был самый худший опыт?
- Можете ли вы сказать, как опыт пользователя помогает интеллекту?
- Можете ли вы найти какую-либо информацию о том, как формируется интеллект? Может быть, вы читали об этом в новостях или статьях?
- Можете ли вы назвать какие-либо способы выявления и смягчения ошибок интеллекта?