

Оглавление

| | |
|--|-----------|
| Предисловие | 17 |
| Введение. | 19 |
| Благодарности | 20 |
| О книге. | 22 |
| Кому адресована книга. | 24 |
| Структура издания | 24 |
| О примерах кода. | 26 |
| От издательства | 26 |
| Об авторах. | 27 |
| Иллюстрация на обложке | 28 |
| Глава 1. Обзор машинного и глубокого обучения | 29 |
| 1.1. Первое знакомство с машинным/глубоким обучением: смена парадигмы в вычислениях | 30 |
| 1.2. Взгляд на машинное обучение как на аппроксимацию функций: модели и их обучение. | 34 |
| 1.3. Простая модель машинного обучения «мозг кошки» | 35 |
| 1.3.1. Входные признаки | 36 |
| 1.3.2. Выходные решения. | 36 |
| 1.3.3. Аппроксимация модели | 36 |
| 1.3.4. Выбор архитектуры модели | 37 |
| 1.3.5. Обучение модели | 37 |
| 1.3.6. Прогнозирование | 39 |
| 1.4. Геометрический взгляд на машинное обучение. | 39 |
| 1.5. Регрессия и классификация в машинном обучении | 42 |
| 1.6. Линейные и нелинейные модели | 42 |

| | |
|---|-----------|
| 1.7. Увеличение выразительной способности за счет использования множества нелинейных слоев: глубокие нейронные сети | 43 |
| Резюме | 45 |
| Глава 2. Векторы, матрицы и тензоры в машинном обучении | 47 |
| 2.1. Векторы и их роль в машинном обучении | 48 |
| 2.1.1. Геометрическое представление векторов и его значение в машинном обучении. | 50 |
| 2.2. Манипуляции с векторами с помощью PyTorch | 52 |
| 2.2.1. Код PyTorch для знакомства с векторами. | 52 |
| 2.3. Матрицы и их роль в машинном обучении. | 53 |
| 2.3.1. Матричное представление цифровых изображений | 55 |
| 2.4. Код на Python: знакомство с матрицами, тензорами и изображениями с помощью PyTorch. | 55 |
| 2.5. Основные векторные и матричные операции в машинном обучении | 58 |
| 2.5.1. Транспонирование матриц и векторов. | 58 |
| 2.5.2. Скалярное произведение двух векторов и его роль в машинном обучении | 59 |
| 2.5.3. Произведение матриц и машинное обучение. | 61 |
| 2.5.4. Длина вектора (норма L2): ошибка модели. | 63 |
| 2.5.5. Геометрический смысл длины вектора | 66 |
| 2.5.6. Геометрический смысл скалярного произведения: сходство признаков | 66 |
| 2.6. Ортогональность векторов и ее физический смысл | 69 |
| 2.7. Код на Python: базовые векторные и матричные операции в PyTorch. | 70 |
| 2.7.1. Транспонирование матриц с помощью PyTorch. | 70 |
| 2.7.2. Скалярное произведение с помощью PyTorch | 71 |
| 2.7.3. Умножение матрицы на вектор с помощью PyTorch | 71 |
| 2.7.4. Умножение матрицы на матрицу с помощью PyTorch. | 72 |
| 2.7.5. Транспонирование произведения матриц с помощью PyTorch | 72 |
| 2.8. Уравнения многомерных линий и плоскостей и машинное обучение | 73 |
| 2.8.1. Уравнение многомерной линии | 73 |
| 2.8.2. Многомерные плоскости и их роль в машинном обучении | 74 |
| 2.9. Линейные комбинации, охватываемое пространство, базисные векторы и коллинеарность | 77 |
| 2.9.1. Линейная зависимость. | 78 |
| 2.9.2. Пространство, охватываемое векторами. | 79 |
| 2.9.3. Векторные пространства, базисные векторы и замыкания | 79 |
| 2.10. Линейные преобразования: геометрическая и алгебраическая интерпретации | 81 |
| 2.10.1. Обобщенное определение многомерных линейных преобразований. | 83 |

| | |
|--|-----|
| 2.10.2. Умножение матрицы на вектор является линейным преобразованием | 83 |
| 2.11. Многомерные массивы, полилинейные преобразования и тензоры | 84 |
| 2.11.1. Представление массивов: многомерные массивы чисел | 84 |
| 2.12. Линейные системы и обратные матрицы | 85 |
| 2.12.1. Линейные системы с нулевыми или близкими к нулю определителями и плохо обусловленные системы | 87 |
| 2.12.2. Вычисление обратной матрицы, детерминанта и проверка вырожденности с помощью PyTorch | 89 |
| 2.12.3. Переопределенные и недоопределенные линейные системы в машинном обучении. | 89 |
| 2.12.4. Псевдообращение матрицы Мура — Пенроуза | 91 |
| 2.12.5. Псевдообращение матриц: замечательная геометрическая интерпретация | 92 |
| 2.12.6. Решение переопределенных систем с помощью PyTorch | 94 |
| 2.13. Собственные значения и собственные векторы: универсальные инструменты машинного обучения | 95 |
| 2.13.1. Собственные векторы и линейная независимость | 98 |
| 2.13.2. Симметричные матрицы и ортогональные собственные векторы | 99 |
| 2.13.3. Вычисление собственных значений и собственных векторов с помощью PyTorch | 100 |
| 2.14. Ортогональные матрицы (вращения), их собственные значения и собственные векторы. | 100 |
| 2.14.1. Матрицы поворота | 100 |
| 2.14.2. Ортогональность матриц вращения. | 103 |
| 2.14.3. Проверка ортогональности матриц вращения с помощью PyTorch | 104 |
| 2.14.4. Собственные значения и собственные векторы матрицы вращения: определение оси вращения | 105 |
| 2.14.5. Определение собственных значений и собственных векторов матриц вращения с помощью PyTorch | 106 |
| 2.15. Диагонализация матрицы. | 106 |
| 2.15.1. Диагонализация матрицы с помощью PyTorch | 107 |
| 2.15.2. Решение линейных систем без обращения с помощью диагонализации. | 108 |
| 2.15.3. Решение линейных систем методом диагонализации с помощью PyTorch | 109 |
| 2.15.4. Возведение матрицы в степень с помощью диагонализации | 110 |
| 2.16. Спектральное разложение симметричной матрицы | 111 |
| 2.16.1. Спектральное разложение матрицы с помощью PyTorch | 112 |
| 2.17. Приложение, связанное с машинным обучением: нахождение осей гиперэллипса | 112 |
| 2.17.1. Решение задачи о гиперэллипсе с помощью PyTorch. | 113 |
| Резюме. | 115 |

| | |
|--|-----|
| Глава 3. Классификаторы и векторное исчисление | 117 |
| 3.1. Геометрическая интерпретация классификации изображений | 117 |
| 3.1.1. Представление входных данных. | 118 |
| 3.1.2. Классификаторы как решающие границы | 119 |
| 3.1.3. Коротко о моделировании | 120 |
| 3.1.4. Знак функции поверхности в бинарной классификации | 123 |
| 3.2. Ошибка, она же функция потерь | 124 |
| 3.3. Минимизация функций потерь: векторы градиента | 124 |
| 3.3.1. Градиенты: применение в машинном обучении | 126 |
| 3.3.2. Представление поверхности уровня и минимизация потерь | 132 |
| 3.4. Локальная аппроксимация функции потерь. | 136 |
| 3.4.1. Одномерный ряд Тейлора | 136 |
| 3.4.2. Многомерный ряд Тейлора и матрица Гессе | 137 |
| 3.5. Реализация градиентного спуска, минимизация ошибок и обучение модели с помощью PyTorch. | 138 |
| 3.5.1. Реализация линейных моделей с помощью PyTorch | 138 |
| 3.5.2. Автоградиент: автоматическое вычисление градиента с помощью PyTorch | 140 |
| 3.5.3. Реализация нелинейных моделей с помощью PyTorch | 141 |
| 3.5.4. Реализация линейной модели мозга кошки с помощью PyTorch | 144 |
| 3.6. Выпуклые и невыпуклые функции, а также глобальные и локальные минимумы. | 145 |
| 3.7. Выпуклые множества и функции. | 146 |
| 3.7.1. Выпуклые множества | 147 |
| 3.7.2. Выпуклые кривые и поверхности. | 147 |
| 3.7.3. Выпуклость и ряды Тейлора | 150 |
| 3.7.4. Примеры выпуклых функций | 150 |
| Резюме. | 151 |
| Глава 4. Инструменты линейной алгебры в машинном обучении | 153 |
| 4.1. Распределение точек данных и истинная размерность | 154 |
| 4.2. Квадратичные формы и их минимизация | 156 |
| 4.2.1. Минимизация квадратичных форм. | 158 |
| 4.2.2. Симметричные положительно (полу)определенные матрицы. | 160 |
| 4.3. Спектральная и фробениусова нормы матрицы | 160 |
| 4.3.1. Спектральная норма | 161 |
| 4.3.2. Фробениусова норма. | 161 |
| 4.4. Метод главных компонент | 162 |
| 4.4.1. Направление максимальной изменчивости. | 164 |
| 4.4.2. Метод главных компонент и уменьшение размерности | 165 |
| 4.4.3. Реализация МГК и уменьшение размерности с помощью PyTorch | 167 |

| | |
|---|------------|
| 4.4.4. Ограничения метода главных компонент | 169 |
| 4.4.5. Метод главных компонент и сжатие данных | 170 |
| 4.5. Сингулярное разложение матрицы | 170 |
| 4.5.1. Неформальное доказательство теоремы о сингулярном разложении | 171 |
| 4.5.2. Доказательство теоремы о сингулярном разложении | 173 |
| 4.5.3. Применение SVD: вычисление главных компонент | 175 |
| 4.5.4. Применение SVD: решение произвольных линейных систем | 176 |
| 4.5.5. Ранг матрицы | 177 |
| 4.5.6. Решение линейных систем методом SVD с помощью PyTorch | 178 |
| 4.5.7. Вычисление главных компонент методом SVD с помощью PyTorch | 179 |
| 4.5.8. Применение SVD: наилучшая аппроксимация матрицей более низкого ранга | 180 |
| 4.6. Применение машинного обучения: поиск документов | 181 |
| 4.6.1. Использование TF-IDF и косинусного сходства | 181 |
| 4.6.2. Латентно-семантический анализ | 183 |
| 4.6.3. Реализация LSA с помощью PyTorch | 186 |
| 4.6.4. Реализация LSA и SVD для большого набора данных с помощью PyTorch | 187 |
| Резюме | 188 |
| Глава 5. Распределение вероятностей в машинном обучении | 190 |
| 5.1. Вероятность: классический частотный взгляд | 191 |
| 5.1.1. Случайные величины | 192 |
| 5.1.2. Гистограммы численности | 193 |
| 5.2. Распределения вероятностей | 194 |
| 5.3. Основные понятия теории вероятностей | 196 |
| 5.3.1. Вероятности невозможных и определенных событий | 196 |
| 5.3.2. Исчерпывающие и взаимоисключающие события | 196 |
| 5.3.3. Независимые события | 197 |
| 5.4. Совместные вероятности и их распределения | 197 |
| 5.4.1. Безусловные вероятности | 199 |
| 5.4.2. Зависимые события и совместное распределение их вероятностей | 200 |
| 5.5. Геометрическое представление распределений зависимых и независимых величин в выборках | 201 |
| 5.6. Непрерывные случайные величины и плотность вероятности | 203 |
| 5.7. Свойства распределений: математическое ожидание, дисперсия и ковариация | 204 |
| 5.7.1. Математическое ожидание, или среднее | 205 |
| 5.7.2. Дисперсия, ковариация и стандартное отклонение | 207 |

| | |
|---|------------|
| 5.8. Выборка из распределения | 210 |
| 5.9. Некоторые известные распределения вероятностей | 213 |
| 5.9.1. Равномерные случайные распределения | 213 |
| 5.9.2. Нормальное (гауссово) распределение | 217 |
| 5.9.3. Биномиальное распределение | 227 |
| 5.9.4. Мультиномиальное распределение | 231 |
| 5.9.5. Распределение Бернулли | 234 |
| 5.9.6. Категориальное распределение и векторы one-hot-кодирования | 236 |
| Резюме | 238 |
| Глава 6. Байесовские инструменты в машинном обучении | 241 |
| 6.1. Условная вероятность и теорема Байеса | 242 |
| 6.1.1. Еще раз о совместной и безусловной вероятности | 243 |
| 6.1.2. Условная вероятность | 244 |
| 6.1.3. Теорема Байеса | 246 |
| 6.2. Энтропия | 247 |
| 6.2.1. Геометрический смысл энтропии | 251 |
| 6.2.2. Энтропия нормального распределения | 252 |
| 6.3. Кросс-энтропия | 253 |
| 6.4. Расстояние Кульбака — Лейблера | 257 |
| 6.4.1. KLD между нормальными распределениями | 258 |
| 6.5. Условная энтропия | 260 |
| 6.5.1. Цепное правило условной энтропии | 262 |
| 6.6. Оценка параметров модели | 262 |
| 6.6.1. Правдоподобие, доказательства, а также априорные и апостериорные вероятности | 262 |
| 6.6.2. Оценка параметра методом максимального правдоподобия | 264 |
| 6.6.3. Оценка параметра методом апостериорного максимума | 265 |
| 6.7. Скрытые переменные и доказательство максимизации | 266 |
| 6.8. Оценка параметра нормального распределения методом максимального правдоподобия | 267 |
| 6.8.1. Оценка методом максимального правдоподобия с помощью PyTorch | 269 |
| 6.8.2. Оценка максимального правдоподобия с помощью PyTorch и градиентного спуска | 270 |
| 6.9. Модели смесей нормальных распределений | 273 |
| 6.9.1. Функция плотности вероятности GMM | 275 |
| 6.9.2. Скрытые переменные для выбора класса | 279 |
| 6.9.3. Классификация с помощью GMM | 281 |
| 6.9.4. Подбор параметров GMM методом максимального правдоподобия (подгонка GMM) | 282 |
| Резюме | 289 |

| | |
|---|------------|
| Глава 7. Аппроксимация функций: как нейронные сети моделируют мир. | 291 |
| 7.1. Нейронные сети: общий взгляд | 292 |
| 7.2. Выражение реальных задач: целевые функции | 294 |
| 7.2.1. Логические функции в практических задачах | 294 |
| 7.2.2. Функции классификации в реальных задачах | 298 |
| 7.2.3. Общие функции в реальных задачах | 304 |
| 7.3. Основной строительный блок, или нейрон, — перцептрон. | 304 |
| 7.3.1. Ступенчатая функция Хевисайда. | 304 |
| 7.3.2. Гиперплоскости | 305 |
| 7.3.3. Перцептроны и классификация | 306 |
| 7.3.4. Моделирование обычных логических элементов с помощью перцептронов | 308 |
| 7.4. На пути к большей выразительности: многослойные перцептроны | 311 |
| 7.4.1. Многослойный перцептрон для моделирования исключающего ИЛИ | 311 |
| 7.5. Многослойные сети перцептронов: многослойные перцептроны, или нейронные сети. | 312 |
| 7.5.1. Многослойная организация | 312 |
| 7.5.2. Моделирование логических функций с помощью многослойных перцептронов | 313 |
| 7.5.3. Универсальная аппроксимационная теорема Цыбенко | 314 |
| 7.5.4. Многослойные перцептроны и многоугольные решающие границы | 322 |
| Резюме. | 323 |
| Глава 8. Обучение нейронных сетей: прямое и обратное распространение ошибки | 325 |
| 8.1. Дифференцируемые ступенчатые функции | 326 |
| 8.1.1. Сигмоидная функция | 326 |
| 8.1.2. Функция гиперболического тангенса | 329 |
| 8.2. Почему многослойная организация предпочтительнее | 330 |
| 8.3. Линейные слои | 330 |
| 8.3.1. Выражение линейных слоев как умножение матрицы на вектор | 332 |
| 8.3.2. Прямое распространение ошибки и выходные функции для многослойного перцептрона с линейными слоями | 333 |
| 8.4. Обучение и обратное распространение ошибки. | 334 |
| 8.4.1. Функция потерь и ее минимизация: цель обучения | 336 |
| 8.4.2. Поверхность функции потерь и градиентный спуск | 336 |
| 8.4.3. Почему градиент определяет наилучшее направление для спуска. | 338 |
| 8.4.4. Градиентный спуск и локальный минимум. | 338 |
| 8.4.5. Алгоритм обратного распространения ошибки | 339 |
| 8.4.6. Все вместе: общий алгоритм обучения. | 348 |
| 8.5. Обучение нейронной сети с помощью PyTorch. | 349 |
| Резюме. | 353 |

| | |
|--|------------|
| Глава 9. Функции потерь, оптимизация и регуляризация | 355 |
| 9.1. Функции потерь | 356 |
| 9.1.1. Количественная оценка и геометрическое представление функции потерь | 356 |
| 9.1.2. Регрессионная функция потерь | 358 |
| 9.1.3. Функция потерь кросс-энтропии | 359 |
| 9.1.4. Функция потерь бинарной кросс-энтропии при несовпадении изображений и векторов | 361 |
| 9.1.5. Функция softmax | 362 |
| 9.1.6. Вычисление функции потерь кросс-энтропии поверх softmax | 364 |
| 9.1.7. Фокальная функция потерь | 367 |
| 9.1.8. Кусочно-линейная функция потерь | 369 |
| 9.2. Оптимизация | 371 |
| 9.2.1. Геометрическое представление оптимизации | 372 |
| 9.2.2. Стохастический градиентный спуск и мини-пакеты | 372 |
| 9.2.3. Реализация SGD с помощью PyTorch | 373 |
| 9.2.4. Метод моментов. | 377 |
| 9.2.5. Геометрическое представление: линии уровня функции потерь, градиентный спуск и метод моментов. | 379 |
| 9.2.6. Метод Нестерова для ускорения градиентного спуска. | 381 |
| 9.2.7. AdaGrad | 383 |
| 9.2.8. Среднее квадратическое распространение ошибки | 384 |
| 9.2.9. Оптимизатор Adam | 386 |
| 9.3. Регуляризация | 387 |
| 9.3.1. Минимальная длина вектора параметров: взгляд на оптимизацию с точки зрения бритвы Оккама | 388 |
| 9.3.2. L2-регуляризация. | 389 |
| 9.3.3. L1-регуляризация. | 390 |
| 9.3.4. Разреженность: сравнение L1- и L2-регуляризации | 390 |
| 9.3.5. Теорема Байеса и стохастический взгляд на оптимизацию. | 392 |
| 9.3.6. Dropout | 394 |
| Резюме. | 398 |
| Глава 10. Свертки в нейронных сетях | 402 |
| 10.1. Одномерная свертка: геометрическое и алгебраическое представление | 404 |
| 10.1.1. Сглаживание кривой с помощью одномерной свертки. | 409 |
| 10.1.2. Обнаружение края кривой с помощью одномерной свертки | 410 |
| 10.1.3. Одномерная свертка как произведение матриц | 411 |
| 10.1.4. PyTorch: одномерная свертка с заранее заданными весами | 414 |
| 10.2. Размер выхода свертки | 415 |
| 10.3. Двумерная свертка: геометрическое и алгебраическое представление. | 416 |
| 10.3.1. Сглаживание изображения с помощью двумерной свертки. | 423 |
| 10.3.2. Обнаружение границ на изображении с помощью двумерной свертки. | 424 |

| | |
|--|------------|
| 10.3.3. PyTorch: двумерная свертка с заранее заданными весами | 426 |
| 10.3.4. Двумерная свертка как произведение матриц | 428 |
| 10.4. Трехмерная свертка | 430 |
| 10.4.1. Определение движения в видео с помощью трехмерной свертки | 433 |
| 10.4.2. PyTorch: трехмерная свертка с заранее заданными весами | 435 |
| 10.5. Транспонированная, или дробно-шаговая, свертка | 436 |
| 10.5.1. Применение транспонированной свертки: автокодировщики и векторные представления | 439 |
| 10.5.2. Размер выхода транспонированной свертки | 440 |
| 10.5.3. Увеличение разрешения с помощью транспонированной свертки | 441 |
| 10.6. Добавление сверточных слоев в нейронную сеть | 443 |
| 10.6.1. PyTorch: добавление сверточных слоев в нейронную сеть. | 443 |
| 10.7. Объединение, или пулинг. | 444 |
| Резюме | 446 |
| Глава 11. Нейронные сети для классификации изображений и обнаружения объектов | 448 |
| 11.1. Сверточные сети для классификации изображений: LeNet. | 449 |
| 11.1.1. PyTorch: реализация LeNet для классификации изображений в MNIST | 452 |
| 11.2. Вперед к еще более глубоким нейронным сетям | 453 |
| 11.2.1. Сеть VGG (Visual Geometry Group) | 454 |
| 11.2.2. Inception: парадигма «сеть в сети». | 460 |
| 11.2.3. ResNet: почему добавление слоев для увеличения глубины не масштабируется. | 464 |
| 11.2.4. PyTorch Lightning. | 470 |
| 11.3. Обнаружение объектов — краткая история. | 475 |
| 11.3.1. R-CNN | 475 |
| 11.3.2. Fast R-CNN | 476 |
| 11.3.3. Faster R-CNN | 477 |
| 11.4. Faster R-CNN: глубокое погружение | 477 |
| 11.4.1. Базовая сверточная сеть (backbone) | 478 |
| 11.4.2. Сеть выбора областей-кандидатов. | 479 |
| 11.4.3. Fast R-CNN | 491 |
| 11.4.4. Обучение Faster R-CNN | 499 |
| 11.4.5. Другие парадигмы обнаружения объектов | 499 |
| Резюме | 501 |
| Глава 12. Многообразия, гомеоморфизм и нейронные сети | 503 |
| 12.1. Многообразия | 503 |
| 12.1.1. Хаусдорфово свойство | 506 |
| 12.1.2. Второе свойство счетности | 507 |

| | |
|---|------------|
| 12.2. Гомеоморфизм. | 509 |
| 12.3. Нейронные сети и гомеоморфизм между многообразиями. | 509 |
| Резюме. | 511 |
| Глава 13. Полное байесовское оценивание параметров модели | 512 |
| 13.1. Полное байесовское оценивание. Неформальное введение. | 513 |
| 13.1.1. Оценка параметров и априорная информация | 513 |
| 13.2. Оценка значений параметров нормального распределения методом MLE (краткий обзор). | 514 |
| 13.3. Полное байесовское оценивание: нормальное распределение, неизвестное среднее, известная точность | 515 |
| 13.4. Большие и малые объемы обучающих данных, а также сильные и слабые априорные предположения | 518 |
| 13.5. Сопряженные априорные распределения. | 519 |
| 13.6. Полное байесовское оценивание: нормальное распределение, неизвестная точность, известное среднее | 520 |
| 13.6.1. Оценка параметра точности. | 521 |
| 13.7. Полное байесовское оценивание: нормальное распределение, среднее и точность неизвестны | 523 |
| 13.7.1. Нормальное гамма-распределение | 523 |
| 13.7.2. Оценка параметров распределения среднего и точности. | 523 |
| 13.8. Пример: полный байесовский вывод. | 526 |
| 13.8.1. Оценка методом максимального правдоподобия | 526 |
| 13.8.2. Байесовский вывод | 526 |
| 13.9. Полное байесовское оценивание: многомерное нормальное распределение, неизвестное среднее, известная точность | 528 |
| 13.10. Полное байесовское оценивание: многомерное распределение, неизвестная точность, известное среднее | 530 |
| 13.10.1. Распределение Уишарта | 530 |
| 13.10.2. Оценка точности | 531 |
| Резюме. | 533 |
| Глава 14. Латентное пространство, генеративное моделирование и автокодировщики | 535 |
| 14.1. Геометрическое представление латентного пространства. | 536 |
| 14.2. Генеративные классификаторы | 538 |
| 14.3. Преимущества и применение моделирования латентного пространства. | 540 |
| 14.4. Линейные латентные пространственные многообразия и метод главных компонент. | 541 |
| 14.4.1. Уменьшение размерности методом главных компонент с помощью PyTorch | 545 |
| 14.5. Автокодировщики | 546 |
| 14.5.1. Автокодировщики и метод главных компонент | 549 |
| 14.6. Гладкость, непрерывность и регуляризация латентных пространств | 550 |

| | |
|--|------------|
| 14.7. Вариационные автокодировщики. | 551 |
| 14.7.1. Геометрическое представление вариационных автокодировщиков | 551 |
| 14.7.2. Обучение VAE, потери и использование. | 553 |
| 14.7.3. Вариационные автокодировщики и теорема Байеса | 555 |
| 14.7.4. Стохастическое отображение приводит к гладкости латентного пространства | 556 |
| 14.7.5. Прямая максимизация апостериорной вероятности требует дорогостоящей нормализации | 556 |
| 14.7.6. Нижняя вариационная граница и вариационные автокодировщики | 556 |
| 14.7.7. Выбор априорного распределения: нормальное распределение с нулевым средним и единичной матрицей ковариации | 559 |
| 14.7.8. Трюк с перепараметризацией. | 561 |
| Резюме. | 565 |
| Приложение | 566 |
| П.1. Скалярное произведение и косинус угла между двумя векторами | 566 |
| П.2. Определители | 568 |
| П.3. Вычисление дисперсии нормального распределения. | 568 |
| П.4. Две теоремы из статистики | 570 |
| П.4.1. Неравенство Йенсена | 570 |
| П.4.2. Неравенство логарифмических сумм | 571 |
| П.5. Гамма-функции и гамма-распределение | 571 |
| П.5.1. Гамма-функция. | 571 |
| П.5.2. Гамма-распределение | 572 |
| Обозначения | 575 |