

Конспект по курсу Машинное обучение¹

Александра Лисицына²

11 января 2020 г.

¹Читаемый Андреем Фильченковым в 2019-2020 годах

²Студентка группы М3435

Оглавление

| | |
|--|-----------|
| 1 Введение | 2 |
| 1.1 Концепция машинного обучения | 2 |
| 1.1.1 Определения машинного обучения | 2 |
| 1.1.2 Родственные пространства | 2 |
| 1.1.3 Родственные сферы | 4 |
| 1.1.4 Требуемый бэкграунд | 4 |
| 1.1.5 Задачи машинного обучения | 4 |
| 1.2 Контролируемое обучение | 5 |
| 1.2.1 Задача | 5 |
| 1.2.2 Главные вопросы | 5 |
| 1.3 Линейные методы | 5 |
| 1.3.1 Показатели эффективности | 5 |
| 1.3.2 Линейная классификация | 7 |
| 1.3.3 Градиентный спуск | 8 |
| 1.4 Probabilistic classifiers (Вероятностные классификаторы) | 8 |
| 1.4.1 Байесская классификация | 8 |
| 1.4.2 непараметрическое восстановление плотности | 10 |
| 1.5 Lection6 | 11 |
| 1.5.1 Логистические правила | 11 |
| 2 Сверточные нейронные сети | 12 |
| 2.1 Беголый взгляд на ImageNet | 12 |
| 2.1.1 Соревнование ImageNet | 12 |
| 2.2 Более ранние приближения в компьютерном зрении | 12 |
| 2.2.1 Короткая история компьютерного зрения | 12 |
| 2.3 Сверточные нейронные сети | 12 |
| 2.4 Тензоры | 12 |
| 2.5 Развертывание и визуализация нейронов | 12 |
| 2.6 Обзор архитектуры | 12 |
| 2.7 Проблемы компьютерного зрения | 12 |

Глава 1

Введение

1.1 Концепция машинного обучения

1.1.1 Определения машинного обучения

Определение. *Машинное обучение* — процесс, позволяющий компьютерам учиться, не будучи явно запрограммированными.

Определение. Компьютерная программа называется *обучающейся* по некоторому опыту E , в отношении некоторой задачи T и некоторой меры производительности P , если его производительность на T , измеренная по P , улучшается с опытом E .

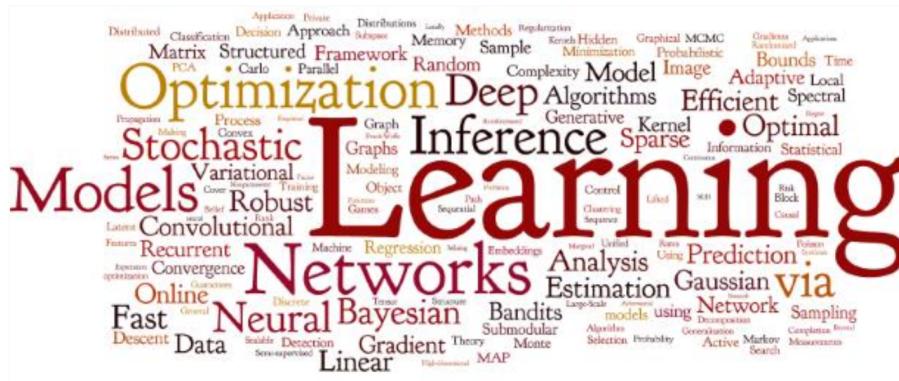


Рис. 1.1: Подходы машинного обучения

1.1.2 Родственные пространства

- Распознавание паттернов
 - Компьютерное зрение
 - Information Retrieval (IR)
 - Natural Language Processing (NLP)
 - Big Data
 - Data Mining (Интеллектуальный анализ данных)

Машинное обучения vs Интеллектуальный анализ данных

Формально, интеллектуальный анализ данных — шаг в открытии знаний в базах данных. Обычно эти два термина используются как синонимы.

1. Сбор данных
 2. Инженерные особенности
 3. Применение алгоритмов машинного обучения

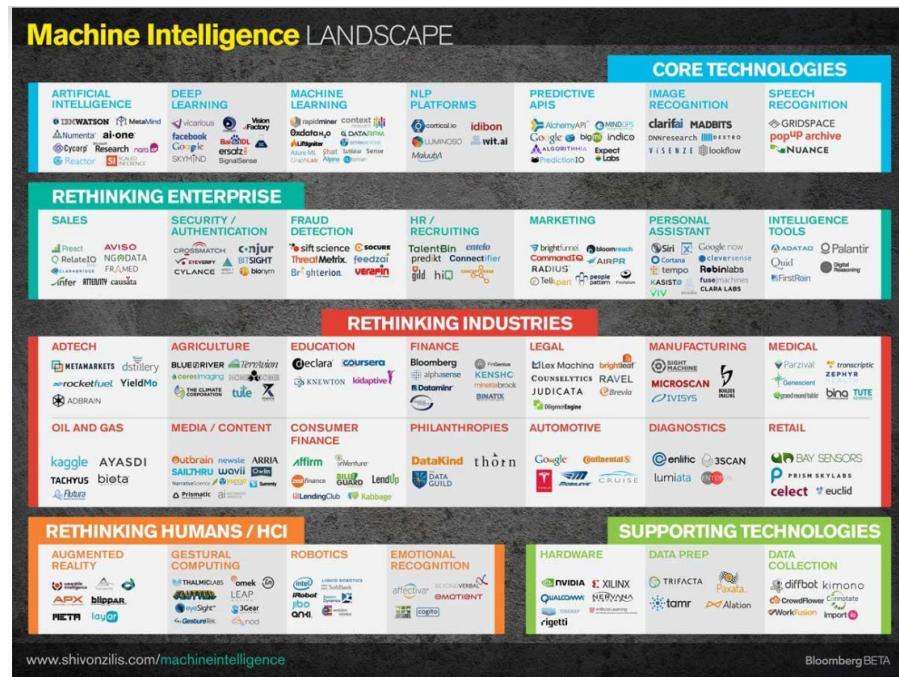


Рис. 1.2: Приложения машинного обучения

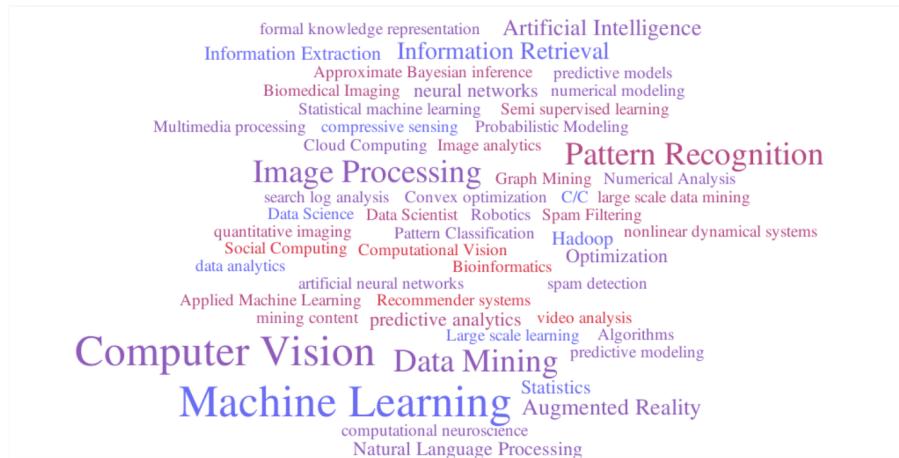


Рис. 1.3: Родственные концепции

Машинное обучение vs Наука о данных

- Сбор данных
- Объединение данных
- Сохранение данных
- Анализ данных
- Высоко производительные вычисления

Машинное обучение vs Анализ данных

Также известное как бизнес-аналитика

- Исследовательский анализ данных
- Подтверждающий анализ данных (статистическая проверка гипотезы)
- Прогностический анализ данных
- Визуализация данных

1.1.3 Родственные сферы

- Искусственный интеллект Сильный AI против слабого
- Интелектуальные системы Экспертные системы против ML систем
- Математическое моделирование
- Способы использования и представления знаний

Искусственный интеллект

- Сейчас у людей есть тенденция говорить AI обо всем, что родственно машинному обучению
- Машинное обучение — общая часть искусственного интеллекта
- Общий искусственный интеллект (раньше сильный AI) — более общая концепция, имеющая отношение к попытке достигнуть или превозмочь ментальные возможности человека

Знания vs данные

Знания *не* данные

Определение. *Знания* — это паттерны в определенной области знаний (принципы, регулярность, отношения, правила, законы), полученные с практикой и профессиональным опытом, которые помогают сформулировать и решить проблемы в определенной области.

1.1.4 Требуемый бэкграунд

- Теория вероятности и математическая статистика
- Оптимизация
- Вычислительная наука
- Линейная алгебра
- Дискретная математика
- Теория комплексных вычислений
- и так далее

1.1.5 Задачи машинного обучения

- Контролируемое обучение Дано множество примеров с ответами. Правило для получаемых ответов для всех возможных примеров требует:
 - классификации
 - регрессии
 - обучения оцениванию
 - прогнозирование
- Неконтролируемое обучение Дано множество примеров без ответов. Правило для нахождения ответов или периодичности требует:
 - кластеризации
 - обучения ассоциированным правилам
 - рекомендательных систем
 - уменьшения размеров
- Полуконтролируемое обучение
- Обучение с подкреплением
- Активное обучение
- Онлайн обучение
- Структурированный прогноз
- Выбор и валидация модели

1.2 Контролируемое обучение

Большую часть времени разговор будет именно о контролируемом обучении.



Рис. 1.4: Контролируемое обучение

1.2.1 Задача

X — множество объектов или входов

Y — множество меток, ответов или выходов

$y : X \rightarrow Y$ — неизвестная целевая функция (зависимость)

$\{x_1, \dots, x_l\} \subset X$ — тренировочное множество примеров

$y_i = y(x_i), i = 1, \dots, l$ — известные значения функции

Задача: найти $a : X \rightarrow Y$ — решающая функция, которая аппроксимирует y на X .

Мы собираемся говорить только о алгоритмах.

1.2.2 Главные вопросы

1. Как описаны объекты?
2. Как выглядят объекты?
3. Каково множество объектов, из которого мы выбираем a ?
4. Какова мера качества того, насколько хорошо a аппроксимирует y ?

1.3 Линейные методы

1.3.1 Показатели эффективности

Таблица сопряженности

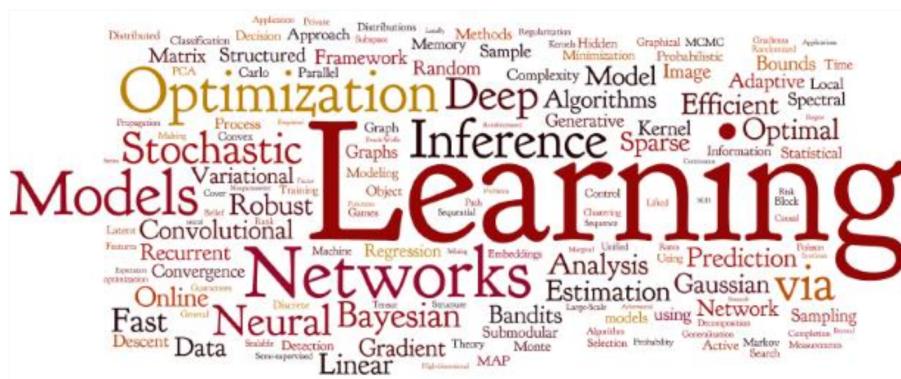


Рис. 1.5: Таблица сопряженности

В математической статистике:

- FN — ошибка первого рода
- FP — ошибка второго рода

$P = TP + FN$ — количество положительных примеров
 $N = FP + TN$ — количество негативных примеров

Некоторые определения

- Полнота (Recall/Sensitivity) $Recall = TPR = \frac{TP}{P}$
- Specificity $SPC = \frac{TN}{N}$
- Точность (Precision) $Precision = PPV = \frac{TP}{TP+FP}$
- Точность (Accuracy) $Accuracy = ACC = \frac{TP+TN}{P+N}$

F–мера

$$F_\beta = (1 + \beta^2) \cdot \frac{Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}$$

ROC–кривая

Определение. Кривая ошибок или ROC–кривая — графическая характеристика качества бинарного классификатора, зависимость доли верных положительных классификаций от доли ложных положительных классификаций при варьировании порога решающего правила. На графике будет не больше точек, чем объектов в обучающей выборке. По факту строится по числу различающихся точки значений порога (то есть если все значения равны, будут ровно две точки). Порог решающего правила в качестве аргумента принимает любым образом выраженную степень уверенности классификатора в ответе.

A — лучший алгоритм; B — типичный; C — худший.

AUC

Определение. AUC (Area Under Curve) — площадь под ROC–кривой. Используется для численной оценки алгоритма.

Случай multiclass

Разобраться с этим случаем.

- One vs One classification
- One vs all classification
- иерархическая классификация
- Confusion matrix

Штучные регрессионные ошибки

- Корневая средне–квадратичная ошибка (RMSE)
- Средняя абсолютная ошибка (MAE)
- Средне–квадратичная ошибка (MSE)
- Симметричная средняя абсолютная ошибка по процентам (SMAPE) $SMAPE = \frac{1}{l} \sum_{j=1}^l \frac{2 \cdot |a(x_j) - y_j|}{|a(x_j)| + |y_j|}$

1.3.2 Линейная классификация

Формуляция проблемы

Ограничение: $Y = \{-1, +1\}$

$T^l = \{(x_i, y_i)\}_{i=1}^l$ дан.

Найти классификатор $a_w(x, T^l)$ в виде $\text{sign}(f(x, w))$, где $f(x, w)$ — функция многообразия, w — вектор параметров

Ключевая гипотеза: все объекты (хорошо) разделимы

Основан идея: искать среди раздельных поверхностей определенных с $f(x, w) = 0$

Margin

Определение. Margin объекта x_i : $M_i(w) = y_i f(x_i, w)$.

$M_i(w) < 0$ — свидетельство неправильной классификации.

У нас есть предыдущее определение Margin: $M(x_i) = C_{y_i}(x_i) - \max_{y \in Y \setminus \{y_i\}} C_y(u)$, где $C_y(u) = \sum_{i=1}^l [y(u, i) = y] w(i, u)$, $w(i, u)$ — функция веса i го соседа u .

Сглаживание функции потерь

Эмпирический риск:

$$Q(a_w, T^l) = Q(w) = \sum_i^l [M_i(w) < 0]$$

просто количество ошибок демонстрируемых a_w .

Эта функция не гладкая, поэтому трудно найти ее оптимум.

Аппроксимация:

$$\tilde{Q}(w) = \sum_i^l L(M_i(w)),$$

где $L(M_i(w)) = L(a_w(x, T^l), x_i)$ — функция потерь

Мы хотим, чтобы L была неотрицательная, невозрастающая и гладкая.

Линейный классификатор

$f_j : X \rightarrow R, j = 1, \dots, n$ — числовые характеристики.

Линейный классификатор

$$a_w(x, T^l) = \text{sign}\left(\sum_{i=1}^n w_i f_i(x) - w_0\right)$$

w_1, w_2, \dots, w_n — весовая характеристика.

Эквивалентная нотация:

$$a_w(x, T^l) = \text{sign}(\langle w, x \rangle)$$

если характеристика $f_0(x) = -1$ добавлена.

Нейрон

McCulloch-Pitts нейрон:

$$a_w(x, T^l) = \sigma\left(\sum_{i=1}^n w_i f_i(x) - w_0\right)$$

где σ — функция активации

Множествов алгоритмов

Это предположение о том, как следует выглядеть классификатору, специфично к множеству A_{linear} , из которого мы и выбираем алгоритм.

$$A_{linear} = \{a_w(x) = sign(< w, x >) | w \in R^n\}$$

Эмпирический риск не является black-box функцией. И даже более того мы можем гарантировать, что она гладкая.

1.3.3 Градиентный спуск

Задача минимизации эмпирического риска:

$$\tilde{Q}(w) = \sum_i^l L(M_i(w)) = \sum_i^l L(< w, x > y_i) \rightarrow \min_w$$

Градиентный спуск:

$w^{[0]}$ — начальное предположение

$w^{[k+1]} = w^{[k]} - \mu \nabla Q(w^{[k]})$, где μ — шаг градиента.

$w^{[k+1]} = w^{[k]} - \mu \sum_i^l L'(< w, x > y_i) x_i y_i$

Останавливаемся, когда значение Q и/или значительно не изменяется.

Min-batch градиентный спуск

Проблема в том, что это слишком рандомно, так как зависит от единственного объекта.

$w^{[0]}$ — начальное предположение; b — размер батча;

$x_{(1)}, x_{(2)}, \dots, x_{(l)}$ — объектный порядок;

$w^{[K+1]} = w^{[K]} - \mu$

1.4 Probabilistic classifiers (Вероятностные классификаторы)

1.4.1 Байесская классификация

Задача

Заболевание распространено среди 1% популяции. Тест возвращает правдивый результат в 95% случаев. Некто получил положительный результат. Какова вероятность, что он действительно болен?

Ответ

$$Pr(d=1|t=1) = \frac{Pr(t=1|d=1)Pr(d=1)}{Pr(t=1|d=1)Pr(d=1) + Pr(t=1|d=0)Pr(d=0)} = \frac{0,95 \cdot 0,01}{0,95 \cdot 0,01 + 0,05 \cdot 0,99} = 0,16$$

Задача вероятностной классификации

Вместо неизвестной целевой функции $y^*(x)$, мы будем думать о неизвестном распределении на $X \times Y$ с плотностью $p(x, y)$.

Определение. Простой (независимо) одинаково распределенный пример — пример, состоящий из l случайных независимых наблюдений $T^l = \{(x_i, y_i)\}_{i=1}^l$.

Теперь у нас есть семейство распределений $\{\phi(x, y, \theta) | \theta \in \Theta\}$ вместо моделей алгоритма.

Задача: найти алгоритм, минимизирующий вероятность ошибки.

Утверждения

$a : X \rightarrow Y$ делит X на непересекающиеся области A_y :

$$A_y = \{x \in X | a(x) = y\}$$

Ошибка возникает, когда элемент x с меткой y классифицируется как принадлежащий $A_s, s \neq y$.

Вероятность ошибки: $Pr(A_s, y) = \int_{A_s} p(x, y) dx A_s$.

Ошибочная потеря: $\lambda_{sy} = \geq \forall (s, y) \in Y \times Y$.

Обычно $\lambda_{yy} = 0, \lambda_y = \lambda_{ys} = \lambda_{yt} \forall s, t \in Y, s \neq y, t \neq y$.

Средний риск a :

$$R(a) = \sum_{y \in Y} \sum_{s \in Y} \lambda_{ys} Pr(A_s, y)$$

Главное уравнение

$$p(X, Y) = p(x)Pr(y|x) = Pr(y)p(x|y)$$

- $Pr(y)$ — априорная вероятность класса y .
- $p(x|y)$ — функция правдоподобия класса y .
- $Pr(y|x)$ — апостериорная вероятность класса y .

2 проблемы

1. Восстановление плотности вероятности:

Дано: $T^l = \{(x_i, y_i)\}_{i=1}^l$.

Задача: найти эмпирическую оценку $\hat{Pr}(y)$ и $\hat{p}(x|y), y \in Y$.

2. Минимизация среднего риска:

Дано:

- априорная вероятность $Pr(y)$
- likelihood $p(x|y), y \in Y$

Задача: найти классификатор a , который минимизирует $R(a)$

Максимум апостериорной вероятности

Пусть $Pr(y), p(x|y)$ известны $\forall y \in Y$

$$p(x, y) = p(x)Pr(y|x) = Pr(y)p(x|y)$$

Главная идея: выбирать класс такой, что наблюдения принадлежат ему с наибольшей вероятностью.

Максимум апостериорной вероятности (MAP):

$$a(x) = \operatorname{argmax}_{y \in Y} Pr(y|x) = \operatorname{argmax}_{y \in Y} Pr(y)p(x|y)$$

Оптимальный байесовский классификатор

Теорема. Если $Pr(y)$ и $p(x|y)$ известны, тогда минимальный средний риск достигается Байесовским классификатором

$$a_O B(x) = \operatorname{argmin}_{s \in Y} \sum_{y \in Y} \lambda_{ys} Pr(y)p(x|y)$$

Если $\lambda_{yy} = 0, \lambda_y = \lambda_{ys} = \lambda_{yt} \forall s, t \in Y, s \neq y, t \neq y$

$$a_{OB}(x) = \operatorname{argmin}_{y \in Y} \lambda_y Pr(y)p(x|y)$$

Классификатор a_{OB} — **оптимальный Байесовский классификатор**.

Байесовский риск — минимальное значение $R(a)$.

Разделительная поверхность (separating surface)

Определение. **Разделительная поверхность (separating surface)** для классов a и b — геометрическое местоположение точек $x \in X$, таких что максимум Байесовского правила решения достигается и для $y = a$ и $y = b$:

$$\lambda_a Pr(a)p(x|a) = \lambda_b Pr(b)p(x|b)$$

1.4.2 непараметрическое восстановление плотности

2 подзадачи

Задача — оценить априорную и апостериорную(?) вероятности для каждого класса:

$$\hat{Pr}(y) = ? \hat{p}(x|y) = ?$$

Первая проблема может быть решена просто:

$$\hat{Pr}(y) = \frac{|X_y|}{l}, X - y = \{(x_i, y_i) \in T^l, y_i = y\}$$

Вторая проблема более сложна.

Проигравший класс

Мы можем решить:

$$\hat{p}(x|y) = ?$$

независимо для каждого класса.

Вместо восстановления $\hat{p}(x|y)$, мы будем восстанавливать $\hat{p}(x)$ используя $T^m = \{(x_{(1)}, s), \dots, (x_{(m)}, s)\}$, for all $s \in Y$.

Одномерный случай

Если $Pr([a, b])$ вероятностная мера на $[a, b]$, тогда

$$p(x) = \lim_{h \rightarrow 0} \frac{1}{2h} Pr([x - h, x + h])$$

Эмпирическая оценка плотности с окном ширины h :

$$\hat{p}_h(x) = \frac{1}{2mh} \sum_{i=1}^m [|x - x_i| < h]$$

Окно Парзена-Розенблатта

Оценка Парзена Розенблатта для окна ширины h :

$$\hat{p}_h(x) = \frac{1}{mh} \sum_{i=1}^m K\left(\frac{x - x_i}{h}\right),$$

где $K(r)$ — ядро.

$\hat{p}_h(x)$ сходится к $p(x)$.

Обобщение для многомерного случая

- Если объекты описаны n числовыми функциями: $f_j : X \rightarrow R, j = 1, \dots, n$

$$\hat{p}_h(x) = \frac{1}{m} \sum_{i=1}^m \prod_{j=1}^n \frac{1}{h_j} K\left(\frac{f_j(x) - f_j(x_i)}{h_j}\right).$$

- Если X метрическое пространство с расстоянием $\rho(x, x')$

$$\hat{p}_h(x) = \frac{1}{mV(h)} \sum_{i=1}^m K\left(\frac{\rho(x, x_i)}{h}\right),$$

где $V(h) = \int_X K\left(\frac{\rho(x, x_i)}{h}\right) dx$ — коэффициент нормализации.

Многомерное окно Парзена

Оценим $\hat{p}_h(x)$:

$$\hat{p}_h(x) = \frac{1}{mV(h)} \sum_{i=1}^m K\left(\frac{\rho(x, x_i)}{h}\right),$$

Окно Парзена:

$$a(x, T^k, h) = \operatorname{argmax}_{y \in Y} \lambda_y Pr(y) l_y^{-1} \sum_{i:y_i=y} K\left(\frac{\rho(x, x_i)}{h}\right),$$
$$\Gamma_y(x) = \lambda_y Pr(y) l_y^{-1} \sum_{i:y_i=y} K\left(\frac{\rho(x, x_i)}{h}\right) \text{близайший к классу}$$

1.5 Lection6

1.5.1 Логистические правила

Концепты и правила

Определение. *Концепт* — предикат над объектом класса X :

$$\phi$$

интерпретируемые концепты

Может быть интерпритируем, если:

- сформулированы на естественном языке
- не более 7 признаков

Информативные концепты

Концепт информативен, если он может покрыть наибольшее число объектов, который нужно покрыть и наименьшее тех, которых не нужно.

сигма, дельта

статистическое правило

$$H_{P,N}(p, n) = \frac{C_P^p C_N^n}{C_{P+N}^{p+n}}$$
$$I_c(\phi, T^l) = -\ln H$$

правило базирующееся на энтропии

$$\hat{H}(P, N) = H\left(\frac{P}{P+N}, \frac{N}{P+N}\right)$$

$$IGain_C(\phi, T^l) = \hat{H}(P, N) - \hat{H}_\phi(P, N, p, n)$$

Дерево решений

Дереворешений — классификатор и регрессия.

Общая схема Придумываем множество правил и меру качества разбиения.

1. Отправляем вершину в корень
2. Если вершина содержит объекты только из одного класса, то говорим, что это лист этого класса и останавливаемся.

Иначе выбираем разделяющее правило, наиболее точное по Φ

деревья регрессии

Глава 2

Сверточные нейронные сети

2.1 Беголый взгляд на ImageNet

2.1.1 Соревнование ImageNet

- 1000 изображений в классе
- 1000 классов
- на данный момент 14 млн изображений

2.2 Более ранние приближения в компьютерном зрении

2.2.1 Короткая история компьютерного зрения

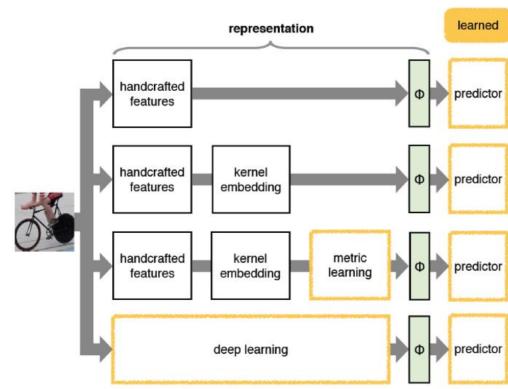


Рис. 2.1: История компьютерного зрения

2.3 Сверточные нейронные сети

2.4 Тензоры

2.5 Разворачивание и визуализация нейронов

2.6 Обзор архитектуры

2.7 Проблемы компьютерного зрения