

# Введение в искусственный интеллект

Наумов Д.А., доц. каф. КТ

Экспертные системы и искусственный интеллект, 2019

# Содержание лекции

- 1 Введение в искусственный интеллект
- 2 История развития ИИ
- 3 Построение интеллектуальных машин
- 4 Терминология и система обозначений
- 5 Дорожная карта для построения систем обучения
- 6 Основные программно-аппаратные реализации

## Искусственный интеллект (ИИ, Artificial Intelligence, AI)

процесс создания машин, которые способны действовать таким образом, что будут восприниматься человеком как разумные.

- научиться лучше понимать нас самих
- научить искусственные системы вести себя разумно

## Феномен (эффект) ИИ

технологии, которые исследуются в его рамках, становятся обычными сразу после их внедрения

## Классификация ИИ:

- **сильный ИИ (Strong AI)** - программное обеспечение, благодаря которому компьютеры смогут думать так же, как люди;
- **слабый ИИ (Weak AI)** - широкий диапазон технологий ИИ, которые могут добавляться в существующие системы и придавать им различные «разумные» свойства.

## Рождение ИИ, 1950-е

- нейронные сети с обратной связью, Вальтер Питтс и Уоррен МакКуллочем, 1945;
- математическая теория обратной связи для биологических и инженерных систем, Норберт Винер;
- способ создания самообучающихся искусственных нейронных сетей («Обучение по Хеббсу»), Дональд Хеббс, 1949;
- «Тест Тьюринга» в качестве способа распознать разумность машины, Аллан Тьюринг, 1950;
- управление символьными данными Logic Theorist (Ньюэлл (Newell), Симон (Simon) и Шоу (Shaw)) и General Problem Solver (Ньюэлл и Симон));
- разработка языков ИИ: IPL (Ньюэллом, Симоном, Шоу), LISP (Джон МакКарти), 1950-е

## Подъем ИИ в 1960-е, спад в 1970-е

- скачок в развитии ИИ, вызванный прогрессом в компьютерных технологиях;
- критики ИИ «Компьютеры и здравый смысл: миф о мыслящих машинах» Мортимера Тауба (Mortimer Taub) и «Алхимия и ИИ» Хуберта и Стюарта Дрейфус (Hubert and Stuart Dreyfus)
- создание окружающей среды, в которой тестировались идеи по компьютерному зрению, роботехнике и обработке человеческого языка, 1960-е, МИТ;
- Мински, Паперт «Перцептроны: введение в вычислительную геометрию» (граничения по использованию простых одноуровневых перцептронов);
- спад исследований ИИ (1970-е) (не удалось выполнить нереальные обещания его успеха);
- применение нечеткой логики при управлении процессами;
- развитие ИИ для игр (первый случай победы компьютера в сложной игре);

## Подъем и спад ИИ, 1980

- продажи экспертных систем на LISP;
- ЭС для разработки ископаемых, прогнозирования инвестиций, диагностики электровозов
- возрождение нейросетей
- спад интереса - сбои ЭС;
- распознавание речи в реальном времени

## Постоянный прогресс ИИ, с 1990-х

Создание продукта, включающего элементы ИИ, является интересной задачей, поскольку позволяет добиться решения многих проблем быстрее и более эффективно, чем при использовании традиционных методов

- системы распознавания фальшивых кредитных карт;
- системы распознавания лиц;
- системы автоматического планирования;
- системы предсказания прибыли и потребности в персонале;
- конфигурируемые системы «добычи данных» из баз данных;
- системы персонализации;
- Deep Blue;
- Deep Space 1 - тестирование технологий с высокими степенями риска;

# Направления ИИ

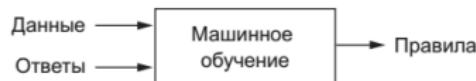
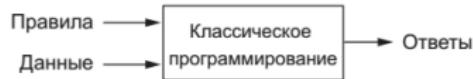
Проблемы	Пути решения
Автоматическое программирование	Определение поведения с тем, чтобы позволить системе ИИ написать программу
Сети Байезана (Bayesian)	Построение сетей на основании вероятностей
Решение проблемы ограничений	Решение переборных задач с помощью различных методик оптимизации поиска
Построение структуры знания	Модификация человеческих знаний в форму, которую сможет понять компьютер
Обучение машин	Создание программ, которые учатся на своем опыте
Нейронные сети	Моделирование программ, которые имеют структуру, схожую с человеческим мозгом
Планирование	Системы, которые способны идентифицировать наилучшую последовательность действий для достижения заданной цели
Поиск	Поиск пути от начальной точки к заданной цели

# Построение интеллектуальных машин

**Имеющийся ресурс:** большой объем структурированных и неструктурированных данных.

**Задача:** разработку самообучающихся алгоритмов для приобретения из этих данных знаний с целью выполнения прогнозов.

- **ручной режим:** выявлять правила и строить модели на основе анализа больших объемов данных;
- **машинное обучение (подобласть ИИ):** постепенное улучшение качества прогнозных моделей и принятие решений, управляемых данными.



# Три типа обучения



- обучение с учителем: мы знаем правильный ответ заранее;
- обучение с подкреплением: мы определяем меру вознаграждения отдельно взятые действия;
- обучение без учителя: выделение содержательной информации без контроля со стороны известной результирующей переменной или функции вознаграждения.

# Выполнение прогнозов о будущем на основе обучения с учителем

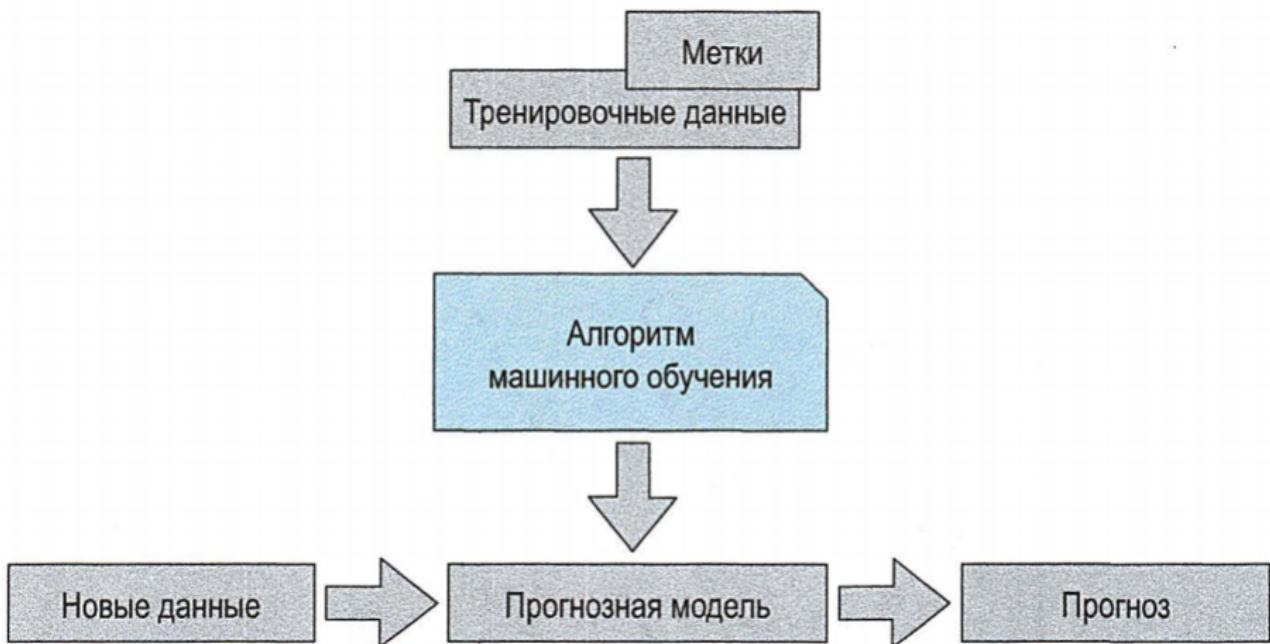
**Задача** обучения с учителем: на маркированных тренировочных данных извлечь модель, которая позволяет делать прогнозы о ранее не встречавшихся или будущих данных.

**Учитель**: подмножество образцов, в которых нужные выходные сигналы (метки) уже известны.

Пример: фильтрация "спама"

- методы классификации: имеются дискретные метки принадлежности к классу;
- регрессия: результирующий сигнал - непрерывная величина.

# Выполнение прогнозов о будущем на основе обучения с учителем

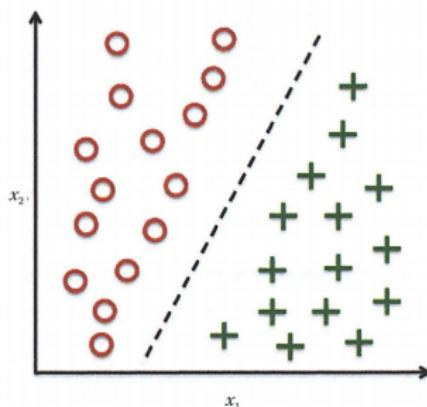


# Выполнение прогнозов о будущем на основе обучения с учителем

## Задача классификации

подкатегория методов машинного обучения с учителем, суть которой заключается в идентификации категориальных меток классов для новых экземпляров на основе предыдущих наблюдений.

Пример: распознавание рукописного текста.

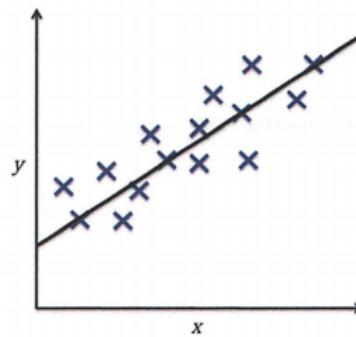


# Выполнение прогнозов о будущем на основе обучения с учителем

## Задача регрессии

предсказанием непрерывных результатов.

В регрессионном анализе нам даны несколько предикторных (объясняющих) переменных и непрерывная (результатирующая) переменная отклика, и мы пытаемся найти между этими переменными связь, которая позволит нам предсказывать результат.



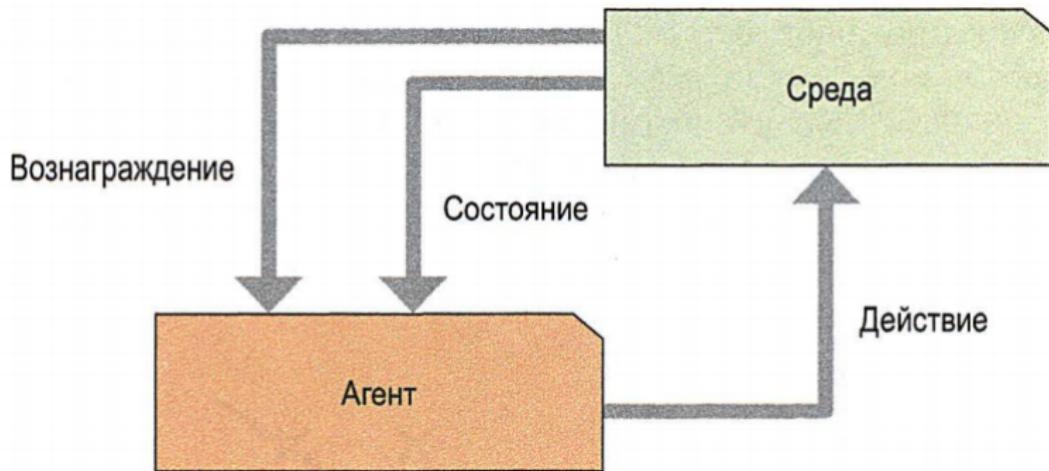
# Обучение с подкреплением

## Задача обучения с подкреплением

выработка системы (агента), которая улучшает свое качество на основе взаимодействий со средой.

- информация о текущем состоянии среды, как правило, содержит так называемый сигнал вознаграждения, обучение с подкреплением можно представить как область, имеющую отношение к обучению с учителем.
- обратная связь является не меткой или значением, раз и навсегда определенными в результате прямых наблюдений, а мерой того, насколько хорошо действие было оценено функцией вознаграждения.

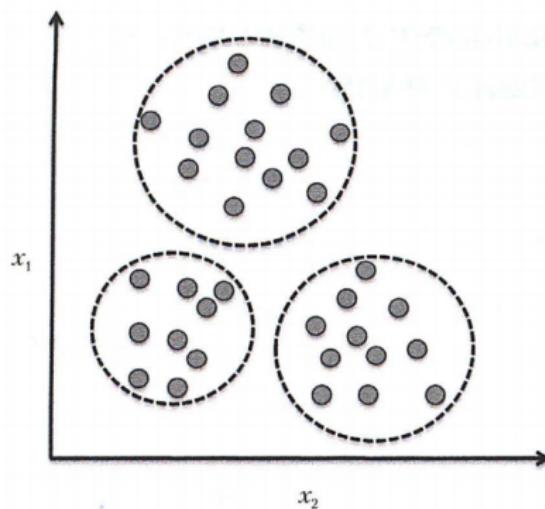
# Обучение с подкреплением



# Обучение без учителя

## Кластеризация

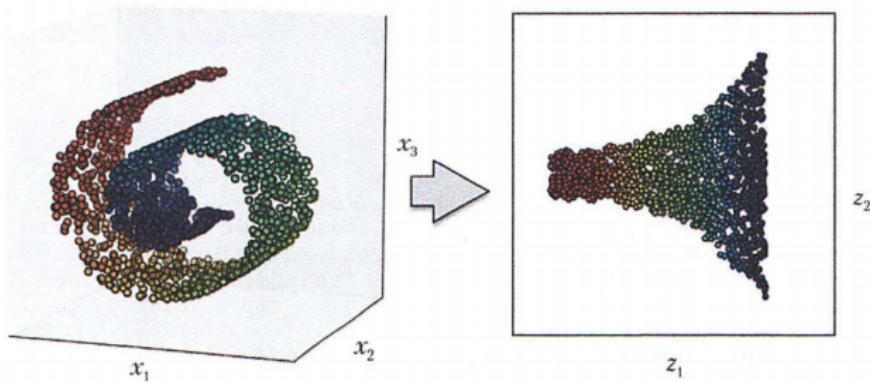
метод разведочного анализа данных, который позволяет организовать группу информации в содержательные подгруппы (кластеры), не имея никаких предварительных сведений о принадлежности группе.



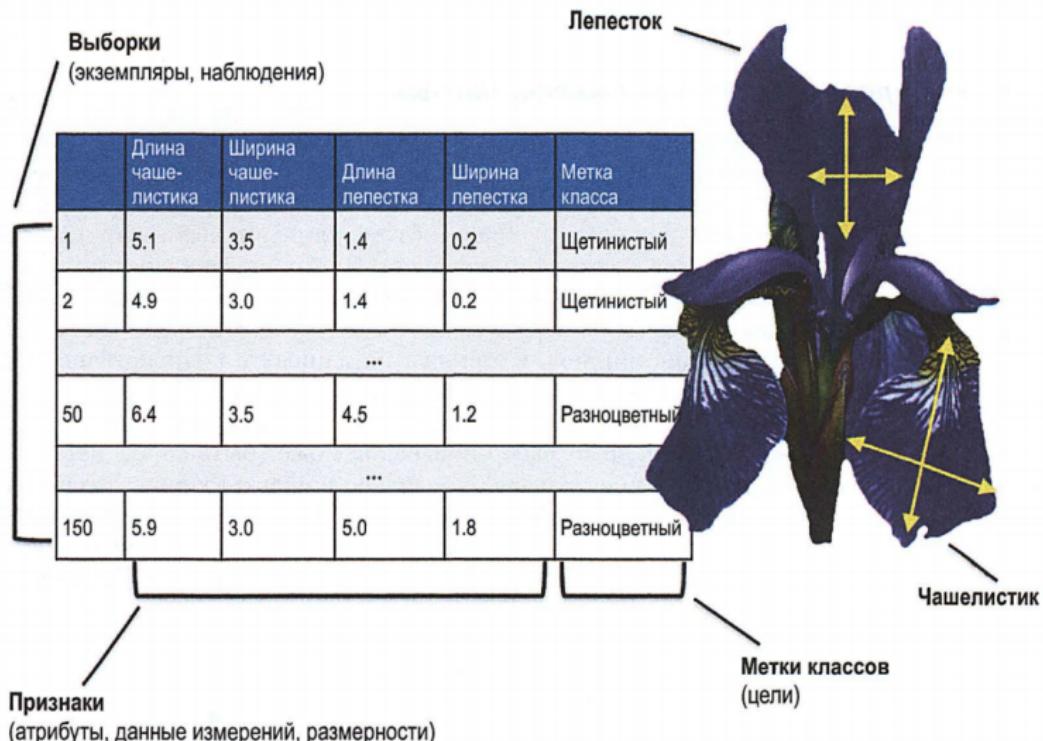
# Обучение без учителя

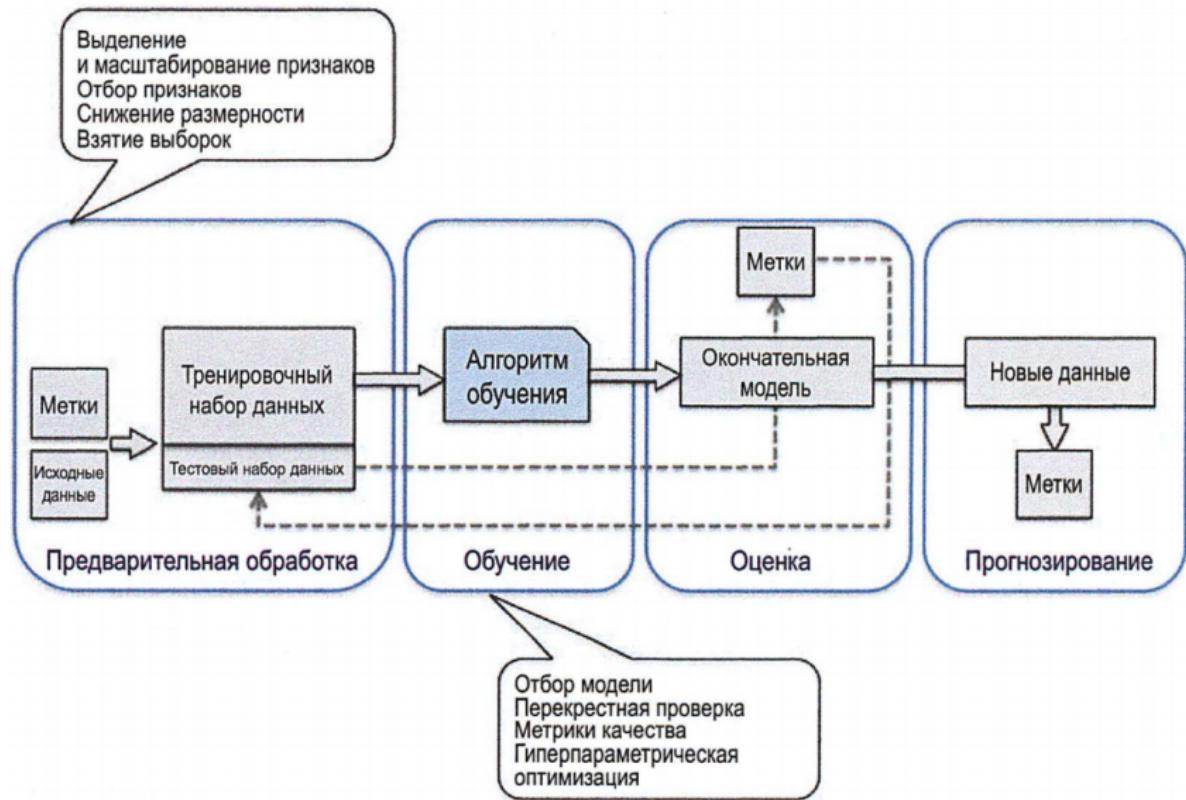
## Снижение размерности данных

подход, использующийся во время предобработки признаков с целью удаления из данных шума, который тоже может ухудшить предсказательную способность алгоритмов, а также для сжатия данных в подпространство меньшей размерности при сохранении большей части релевантной информации.



Ирисы Фишера (Iris) - набор данных для задачи классификации, на примере которого Рональд Фишер в 1936 году продемонстрировал разработанный им метод дискриминантного анализа.





# Основные программно-аппаратные реализации

## OpenCV - модуль ml

### Языки программирования

C, C++, Python; полуофициально - Java, Ruby и прочие



### Достоинства

- бесплатность (open source);
- модуль прекрасно отлажен и тщательно протестирован во многих бесплатных и коммерческих приложениях.

### Реализованные концепции

- многослойный перцептрон;
- машина опорных векторов;
- Байесовский классификатор;
- каскадный классификатор;
- ...

# Основные программно-аппаратные реализации

## ROOT



Программная библиотека, используемая в CERN'е - в частности, для расчетов в ходе экспериментов на БАК

### **Язык программирования**

C++

### **Достоинства**

- бесплатность (open source);
- библиотека прекрасно отлажена и тщательно протестирована;
- библиотека содержит огромное количество не-ml функционала.

# Основные программно-аппаратные реализации

## Системы численных расчетов

numpy / scipy / ipython, octave, sage, matlab, R



—предварительное проектирование алгоритмов;

—статистические расчеты (язык программирования R).

## Достоинства

—как правило, бесплатны (кроме matlab);

—низкий уровень вхождения и простота освоивания;

—реализованы практически все концепции ml.

## Недостатки

—возможно, проприетарны (matlab);

—низкая скорость работы;

—скрытие особенностей реализаций концепций.

# Основные программно-аппаратные реализации

## Нейропроцессоры

NeuroMatrix, процессоры Texas Instruments и прочие



### Достоинства

- высокая скорость обучения нейронных сетей;
- высокая скорость пакетного прогона нейронных сетей.

### Недостатки

- высокая цена;
- реализуют, как правило, некоторые подклассы нейронных сетей прямого распространения.