



**3D МОДЕЛИРОВАНИЕ И
ПРОГРАММИРОВАНИЕ**



приоритет2030[^]
лидерами становятся

ТЕМА:

Типы входных данных при обучении

КОНСПЕКТ ЛЕКЦИИ

Преподаватель:

Шарипов Ильдар Курбангалиевич
к.т.н., доцент, заведующий кафедрой
электроснабжения и эксплуатации
электрооборудования.





Тема 1. Типы входных данных при обучении

Вопрос 1. Входные данные и функционал качества.

Вопрос 2. Входные и выходные данные.

Вопрос 3. Процесс тренировки.



Вопрос 1. Входные данные и функционал качества.

Классические задачи решаемые с помощью машинного обучения:

- Классификация как правило, выполняется с помощью обучения с учителем на этапе собственно обучения.
- Кластеризация как правило, выполняется с помощью обучения без учителя.
- Регрессия как правило, выполняется с помощью обучения с учителем на этапе тестирования, является частным случаем задач прогнозирования.
- Понижение размерности данных и их визуализация выполняется с помощью обучения без учителя.
- Восстановление плотности распределения вероятности по набору данных.
- Одноклассовая классификация и выявление новизны.
- Построение ранговых зависимостей.

Типы входных данных при обучении:

- Признаковое описание объектов - наиболее распространённый случай.
- Описание взаимоотношений между объектами, чаще всего отношения попарного сходства, выражаемые при помощи матрицы расстояний, ядер либо графа данных.
- Временной ряд или сигнал.
- Изображение или видеоряд.

Типы функционалов качества.

При обучении с учителем - функционал качества может определяться как средняя ошибка ответов. Предполагается, что искомый алгоритм должен его минимизировать. Для предотвращения переобучения в минимизируемый функционал качества часто в явном или неявном виде добавляют регуляризатор.



При обучении без учителя - функционалы качества могут определяться по-разному, например, как отношение средних межкластерных и внутрикластерных расстояний.

При обучении с подкреплением - функционалы качества определяются физической средой, показывающей качество приспособления агента.

Практические сферы применения

Целью машинного обучения является частичная или полная автоматизация решения сложных профессиональных задач в самых разных областях человеческой деятельности.

Машинное обучение имеет широкий спектр приложений:

- Распознавание речи
- Распознавание изображений
- Распознавание рукописного ввода
- Техническая диагностика
- Медицинская диагностика
- Прогнозирование временных рядов
- Биоинформатика
- Обнаружение мошенничества
- Обнаружение спама
- Категоризация документов
- Биржевой технический анализ
- Финансовый надзор
- Кредитный скоринг
- Предсказание ухода клиентов
- Хемоинформатика

Сфера применений машинного обучения постоянно расширяется. Повсеместная информатизация приводит к накоплению огромных объёмов данных в науке, производстве, бизнесе, транспорте, здравоохранении.



Возникающие при этом задачи прогнозирования, управления и принятия решений часто сводятся к обучению по прецедентам. Раньше, когда таких данных не было, эти задачи либо вообще не ставились, либо решались совершенно другими методами.

Вопрос 2. Входные и выходные данные.

Начнём со следующей задачи — даны входные и выходные значения.

Входные: 0, 8, 15, 22

Выходные: 32, 46.4, 59, 71.6

Когда в качестве входного значения у вас значение 0, то в качестве выходного значения — 32. Когда в качестве входного значения у вас 8, то в качестве выходного значения — 46.4. Когда в качестве входного значения у вас 15, то в качестве выходного значения — 59 и так далее.

Присмотримся к этим значениям и зададим вопрос. Можем ли мы определить, каким будет выходное значение, если на входе мы получим 38?

Входные: 0, 8, 15, 22, 38

Выходные: 32, 46.4, 59, 71.6, ?

Если вы ответили 100.4, то оказались правы!



Итак, как мы могли решить эту задачу? Если присмотреться внимательнее к значениям, то можно заметить, что они связаны выражением:

$$F = C \cdot 1,8 + 32$$

C – Цельсий, F – Фаренгейт

Входные: 0, 8, 15, 22, 38

Выходные: 32, 46.4, 59, 71.6, 100.4

где *C* — градусы Цельсия (входные значения), *F* — Фаренгейта (выходные значения).

То, что сейчас сделал ваш мозг — сопоставил входные значения и выходные значения и нашел общую модель (связь, зависимость) между ними, — именно это и делает машинное обучение.

По входным и выходным значениям алгоритмы машинного обучения найдут подходящий алгоритм преобразования входных значений в выходные. Это можно представить следующим образом:

Традиционная разработка программного обеспечения

Входные данные и алгоритм известны заранее, пишется функция для получения нужного выходного значения

- Входные данные
- Применяется логика обработки
- Производится нужный результат

Машинное обучение

Известны входные и выходные значения, но неизвестен алгоритм производящий выходные значения согласно входным

- Взять соответствующие пары входных и выходных значений
- Определить алгоритм преобразования

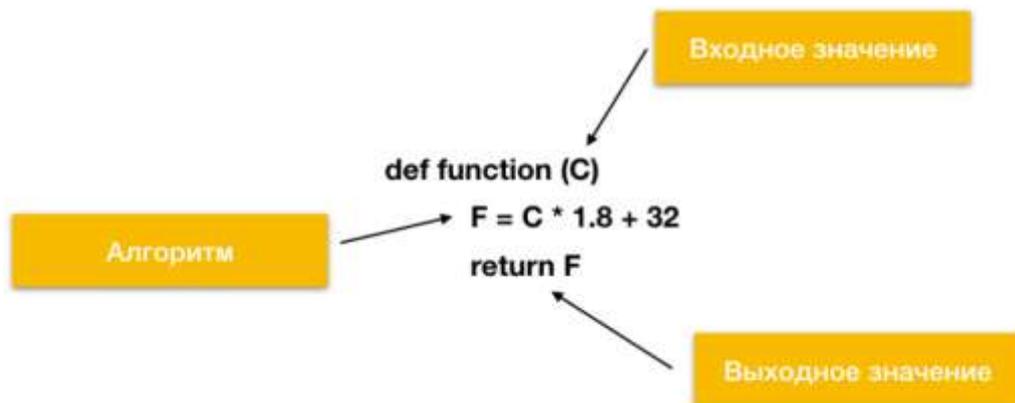


Представим себе, что мы хотим разработать программу, которая будет преобразовывать градусы Цельсия в градусы Фаренгейта используя формулу $F = C * 1,8 + 32$.

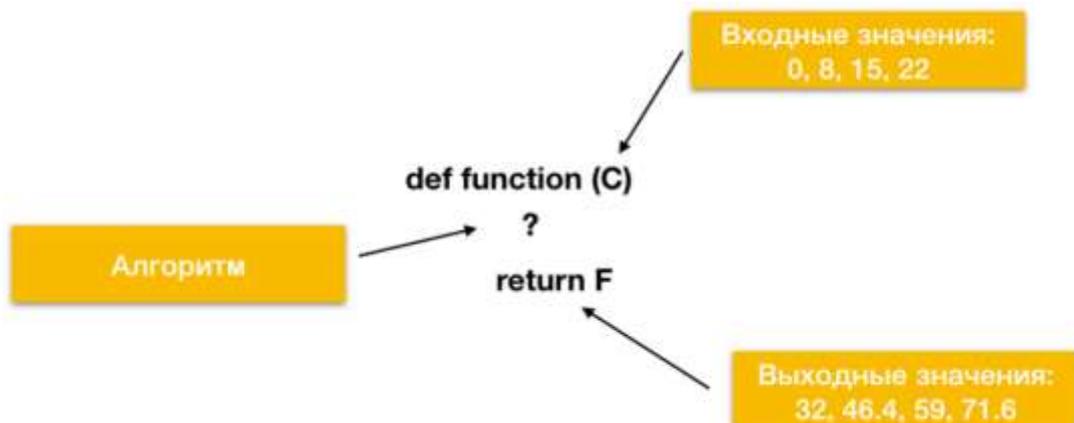
Решение, при подходе с точки зрения традиционной разработки программного обеспечения, может быть реализовано на любом языке программирования с использованием функции:

```
def function (C)  
    F = C * 1.8 + 32  
    return F
```

Функция принимает входное значение C, затем вычисляет выходное значение F используя явно заданный алгоритм, а затем возвращает вычисленное значение.

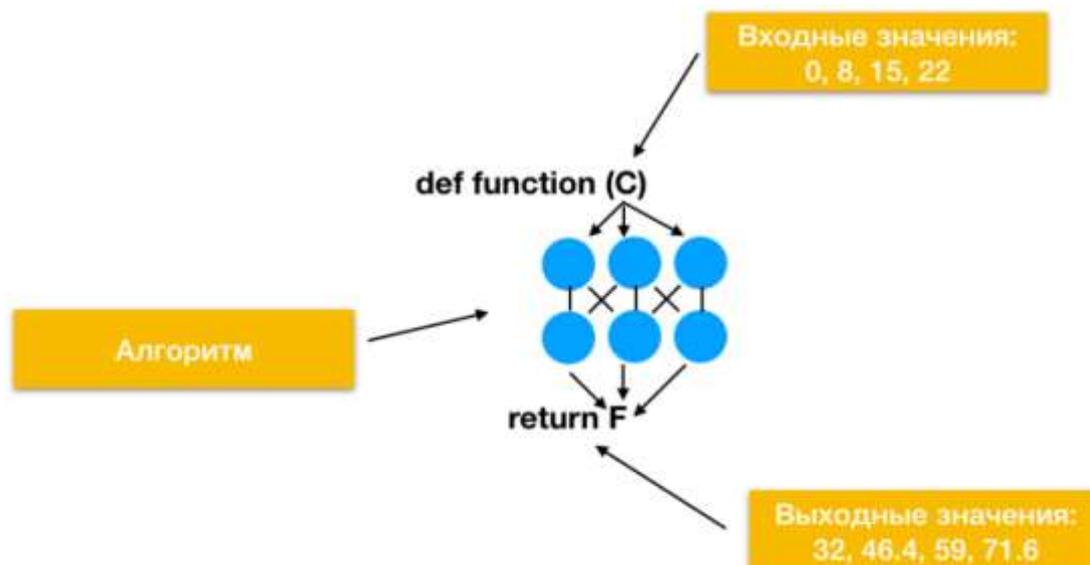


С другой стороны, в подходе с машинным обучением, у нас есть только входные и выходные значения, но не сам алгоритм:





Подход с машинным обучением основывается на использовании нейронных сетей для нахождения отношений между входными и выходными значениями.



Вы можете думать о нейронных сетях, как о стопке слоёв, каждый из которых состоит из заранее известной математики (формул) и внутренних переменных. Входное значение поступает в нейронную сеть и проходит сквозь стопку слоёв нейронов. Во время прохождения через слои, входное значение преобразовывается согласно математике (заданным формулам) и значениям внутренних переменных слоёв, производя выходное значение.

Для того, чтобы нейронная сеть смогла обучиться и определить правильные отношения между входными и выходными значениями, нам необходимо её обучить — натренировать.

Мы тренируем нейронную сеть через повторяющиеся попытки сопоставить входные значения выходным.

В процессе тренировки происходит «подгонка» (подбор) значений внутренних переменных в слоях нейронной сети до тех пор, пока сеть не



научится генерировать соответствующие выходные значения соответствующим входным значениям.

Как мы увидим в последующем, для того чтобы обучить нейронную сеть и позволить ей подобрать наиболее подходящие значения внутренних переменных, производят тысячи или десятки тысяч итераций (тренировок).

В качестве упрощенного варианта понимания машинного обучения вы можете представить себе алгоритмы машинного обучения как функции, которые подбирают значения внутренних переменных таким образом, чтобы соответствующим входным значениям соответствовали корректные выходные значения.

Существует множество типов архитектур нейронных сетей. Однако, вне зависимости от того, какую архитектуру вы выберете, математика внутри (какие вычисления выполняются и в каком порядке) останется неизменной в процессе тренировки. Вместо изменения математики, меняются внутренние переменные (веса и смещения) во время тренировки.

Например, в задаче конвертации из градусов Цельсия в Фаренгейты, модель начинает с умножения входного значения на некоторое число (вес) и добавления другого значения (смещения). Обучение модели заключается в нахождении подходящих значений для этих переменных, без изменения выполняемых операций умножения и сложения.

А вот одна крутая вещь над которой стоит задуматься! Если вы решили задачу преобразования градусов Цельсия в Фаренгейты, которая обозначена в видео и в тексте ниже, вы, вероятно, решили её потому, что обладали неким предыдущим опытом или знанием, как производить подобного рода преобразования из градусов Цельсия в Фаренгейты. Например, вы могли просто знать, что 0 градусов Цельсия соответствуют 32 градусам по Фаренгейту. С другой стороны, системы основанные на машинном обучении не обладают предыдущими вспомогательными знаниями для решения



Системы искусственного
интеллекта



приоритет2030⁺
лидерами становятся

поставленной задачи. Они учатся решать подобного рода задачи не основываясь на предыдущих знаниях и при их полном отсутствии.



Вопрос 3. Процесс тренировки.

Процесс тренировки (происходящий в методе `model.fit(...)`) состоит из весьма простой последовательности действий, результатом которых должны стать значения внутренних переменных дающих максимально близкий к исходному результаты. Процесс оптимизации, благодаря которому достигаются такие результаты, называется градиентным спуском, использует численный анализ для поиска максимально подходящих значений для внутренних переменных модели.

Чтобы заниматься машинным обучением вам, в принципе, нет необходимости разбираться в этих деталях. Но для тех, кому всё-таки интересно узнать больше: градиентный спуск посредством итераций изменяет значения параметров по-немногу, «вытягивая» их в нужном направлении, до тех пор пока не будут получены наилучшие результаты. В данном случае «лучшие результаты» (лучшие значения) означают, что любое последующее изменение параметра только ухудшит результат модели. Функция, которая измеряет насколько хороша или плоха модель на каждой итерации называется «функцией потерь», и цель каждого «вытягивания» (корректировки внутренних значений) — уменьшить значение функции потерь.

Тренировка модели осуществляется методом `fit`.

Во время тренировки модель получает на вход значения градусов Цельсия, выполняет преобразования используя значения внутренних переменных (называемые «весами») и возвращает значения, которые должны соответствовать градусами по Фаренгейту. Так как первоначальные значения весов установлены произвольными, то и результирующие значения будут далеки от корректных значений. Разница между



необходимым результатом и фактическим вычисляется с использованием функции потерь, а функция оптимизации определяет каким образом должны быть подкорректированы веса.

Этот цикл вычислений, сравнений и корректировки контролируется внутри метода `fit`. Первый аргумент — входные значения, второй аргумент — желаемые выходные значения. Аргумент `epochs` определяет какое количество раз этот обучающий цикл должен быть выполнен. Аргумент `verbose` контролирует уровень логгирования.

```
history = model.fit(celsius_q, fahrenheit_a, epochs=500, verbose=False)
print("Завершили тренировку модели")
```

Процесс тренировки начинается с блока «прямое распространение», при котором входные параметры поступают на вход нейронной сети, следуют к скрытым нейронам и затем идут к выходным. Затем модель применяет внутренние преобразования над входными значениями и внутренними переменными для предсказания ответа.

В нашем примере, входным значением является температура в градусах Цельсия и модель предсказывала соответствующее значение в градусах Фаренгейта.