Мотивация

Одна из функций математики — формализация привычных нам объектов и их свойств. Кроме того, математика минималистична: например, объекты казалось бы совершенно разной природы можно представлять в виде *векторов* или *матриц*. Их мы ещё упомянем чуть дальше и подробно обсудим в курсе программы, посвящённом линейной алгебре.

Обобщение и формализация позволяют сформулировать задачу максимально просто — например, найти минимум функции. А это в свою очередь позволяет придумать общие методы решения задач — например, *градиентный спуск*.

Задача машинного обучения в общем виде

В общем виде задача машинного обучения состоит в том, чтобы найти алгоритм, который хорошо приближает данную *целевую функцию* для объектов из некоторого множества. Поскольку объекты в задачах машинного обучения бывают устроены очень сложно, первый шаг — превратить сложный объект во входные данные для алгоритма, то есть отождествить объект с набором признаков (их ещё называют *фичи* от английского feature).

Алгоритмы в машинном обучении называют *моделями*, чуть позже в этом уроке мы поймём, почему.

Комментарий. Если термин выделен курсивом, значит, мы не предполагаем, что вы его знаете. В этом рассказе будут упоминаться математические объекты, которые впоследствии будут фигурировать в нашей программе.

Почему математический анализ?

Математический анализ состоит из двух направлений:

- **Дифференциальное исчисление** — свойства и применение *производных* функции. Основное применение дифференциального исчисления в машинном обучении – поиск минимумов и максимумов функции.
- **Интегральное исчисление** — свойства и применение *интегралов* функций. Заметная часть определений и вычислений в теории вероятностей использует интегралы. А на теорию вероятностей в свою очередь нередко опираются модели машинного обучения. Так же на теории вероятностей держится вся статистика.

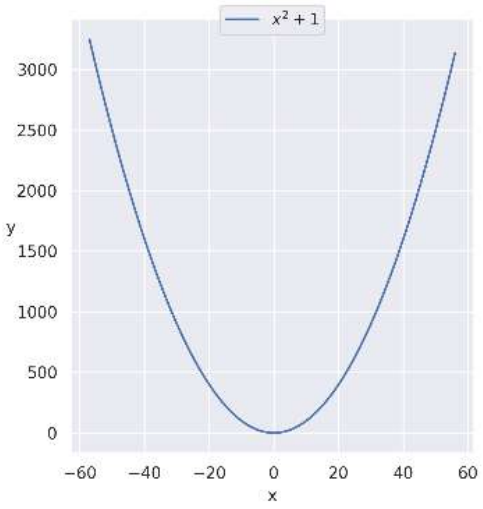
На ближайших трёх уроках мы поймём, почему в машинном обучении так важна задача поиска минимума.

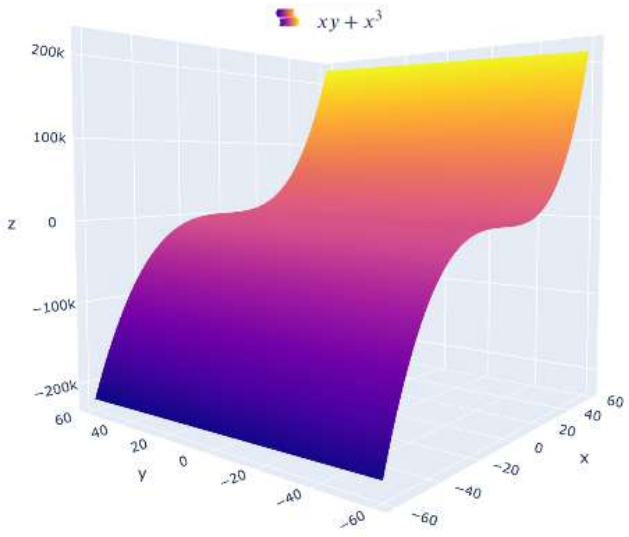
Задача поиска минимума при помощи *градиентного спуска* и является конечной целью первого курса нашей программы. Поэтому в этом курсе мы сконцентрируемся на дифференциальном исчислении.

Одномерный и многомерный математический анализ

Как мы узнали, математический анализ состоит из дифференциального исчисления (изучения производных) и интегрального исчисления (изучения интегралов). Также математический анализ можно условно разделить на одномерный и многомерный — изучение функций от одной и от многих переменных. На второй и третьей неделе курса мы будем изучать одномерный математический анализ, а на четвертой неделе обобщим наши знания на многомерный случай.

Пример. Функция от одной переменной $f(x) = x^2 + 1$ – слева на картинке. Функция от двух переменных $f(x, y) = xy + x^3$ – справа на картинке.





Примеры задач и целевой функции

Регрессия: оценка арендной платы за квартиру

Представим, что перед нами стоит задача предсказывать стоимость арендной платы за квартиру. В данном случае **объектами** будут квартиры. Задать какую-то конкретную квартиру можно многими разными способами: например, адресом. Однако, чтобы научиться хорошо предсказывать стоимость аренды по адресу, нужно побывать в очень и очень большом числе квартир. Более разумным будет выделить в качестве **признаков**, например, информацию из объявления об аренде квартиры на Циане – число комнат, наличие ремонта, этаж, расстояние до центра города и т.д.

Целевая функция – арендная плата в объявлении. Тип задач, где значение целевой функции может быть произвольным числом из некоторого промежутка, называется регрессией. Например, в данном случае целевая функция может быть любым положительным числом.

Бинарный классификатор: пёсики и кексики

Объекты – фотографии пёсиков и кексиков, задача – определить, что есть что. То есть наша **целевая функция** f определяется следующим образом: $f(\text{фотография пёсика}) = \text{dog}, f(\text{фотография кексика}) = \text{muffin}.$

Источник: гитхаб датасета

Такой тип задач называется *классификацией*, поскольку задача состоит в том, чтобы отнести объект к одному из классов. Иначе говоря, целевая функция принимает конечное число значений (значение – класс объекта). В данном пример класса два: пёсики и кексики, поэтому мы имеем дело с частным случаем задачи классификации – *бинарной классификацией*.

На предыдущем шаге вы познакомились с двумя типами задач: регрессией и классификацией. Попробуйте определить для приведенных ниже задач их тип:

Сопоставьте значения из двух списков

Игра съедобное-несъедобное	Регрессия
Определить по следам на снегу, кто их оставил	Классификация с более чем двумя классами
Вычислить расстояние от Москвы до Петушков	Бинарная классификация

Причём тут вообще математика?

Может показаться, что в предыдущих двух примерах математика ни при чём: ведь математика – это строгая наука, там должны быть какие-то числа, формулы. На самом деле любую задачу машинного обучения можно переписать в виде задачи *математической оптимизации*. Но давайте по порядку: для начала представим входные данные нашей задачи в виде математических объектов и представим целевую функцию в виде числовой функции.

Квартиры становятся векторами

Предположим для простоты, что в объявлении есть только такая информация:

Тогда каждое объявление задается тремя признаками – числами r, d и p:

- r – число комнат, целое положительное число
- d – расстояние до центра, произвольное положительное число
- p – разрешены ли в квартире питомцы (по смыслу возможные значения «да» и «нет», но мы можем договориться, что 1 – это «да», а 0 – это «нет»)

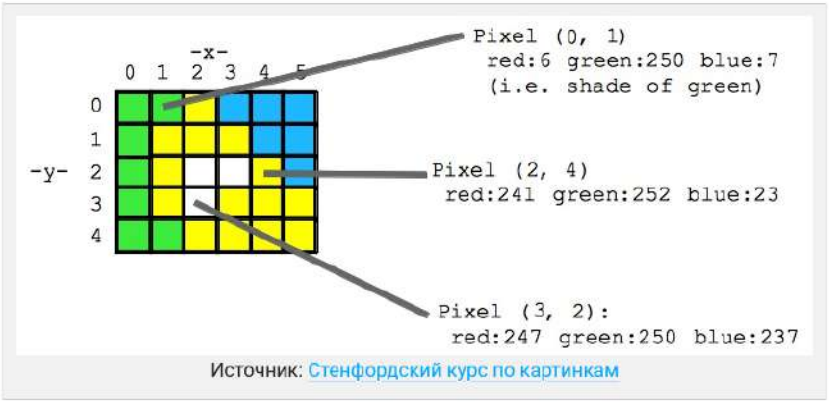
Таким образом, целевая функция f зависит от трёх чисел r, d, p, а её значение может быть любым положительным числом (вообще-то стоимость аренды обычно целая, но в принципе ничто не мешает хозяевам запросить за аренду $49\,999$ р. и 99 коп.). Набор чисел (r, d, p) еще называют *вектором* признаков.

Конечно, в реальности в объявлении куда больше информации, которая может влиять на его стоимость: например, фотографии. Про то, как представить фотографию в численном виде, вы узнаете на следующем шаге.

Причём тут вообще математика?

Пёсики и кексики превращаются в матрицы

Представление картинок в численном виде. Предположим, фотографии кексиков и пёсиков имеют разрешение 100×100 пикселей. Чтобы однозначно задать картинку, нужно указать цвет каждого пикселя. Пиксель в свою очередь задается парой чисел: (i, j), где i – номер строки, j – номер столбца, где этот пиксель находится. Существует множество способов представить цвет в численном виде, выберем, например RGB-формат. В RGB формате цвет – это тройка чисел от 0 до 255, каждое из которых обозначает интенсивность соответствующего базового цвета: красного, зелёного или синего. Отсюда и аббревиатура: Red, Green, Blue.



Источник: Стенфордский курс по картинкам

Для удобства введём обозначения: для пикселя, который находится в i-ой строке, в j-ом столбце обозначим за $r_{i,j}$ интенсивность красного цвета, $g_{i,j}$ – интенсивность зеленого и $b_{i,j}$ – интенсивность синего цвета. Таким образом, каждый пиксель входной картинки задается 3 числами $(r_{i,j}, g_{i,j}, b_{i,j})$. Так как всего пикселей $100 \cdot 100 = 10000$, картинка целиком задается упорядоченным набором из 30000 чисел, каждое из которых принимает значение от 0 до 255.

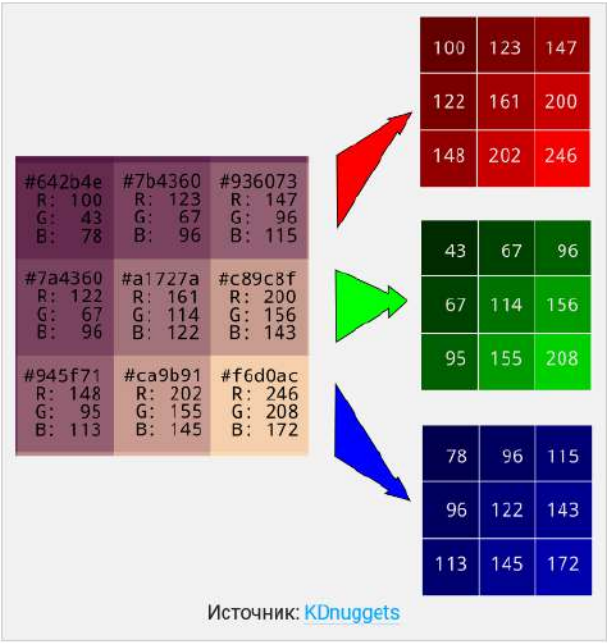
Численная целевая функция. Мы добились того, что входные данные целевой функции и нашего алгоритма машинного обучения – числа. Осталось только заменить значения целевой функции на числа: по аналогии с тем, как мы делали ранее, можно, например, сказать, что для фотографий собачек целевая функция f равна 0, а для кексиков – 1.

Картинка в виде матриц

Забегая вперёд, заметим, что удобный способ представлять себе набор из 30000 чисел, которым кодируется изображение, в виде трёх матриц 100×100. Пока можно считать, что матрица 100×100 – это таблица (или двумерный массив) со 100 строками и 100 столбцами:

$$R = \begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,100} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,100} \\ \vdots & \vdots & \ddots & \vdots \\ r_{100,1} & r_{100,2} & \cdots & r_{100,100} \end{pmatrix} \quad G = \begin{pmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,100} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,100} \\ \vdots & \vdots & \ddots & \vdots \\ g_{100,1} & g_{100,2} & \cdots & g_{100,100} \end{pmatrix} \quad B = \begin{pmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,100} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,100} \\ \vdots & \vdots & \ddots & \vdots \\ b_{100,1} & b_{100,2} & \cdots & b_{100,100} \end{pmatrix}$$

Возможно пока это кажется не очень понятным, подробно про матрицы мы поговорим в части программы посвящённой линейной алгебре :)



Источник: KDnuggets

Представьте, что перед вами стоит задача научиться предсказывать продолжительность жизни человека. Как и в случае с квартирами, каждый объект (человек) уникален. Подумайте, какие признаки могут быть значимыми для решения данной задачи?

Предлагаем вам записать ваш ответ: проверяться он никак не будет, но вам может быть любопытно посмотреть на свой ответ после прохождения первой части нашей программы.