ML. Интерполяция и полиномы

1. Полиномы и интерполяция

Полиномы — это функции, формулы которых выглядят так:

коэффициенты (числа)

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_0$$

$a_0, a_1, a_2, a_3, \ldots, a_n$ — это числа, которые называются коэффициентами полинома.

Полиномы разделяют по величине старшей степени. Например, знакомая нам линейная функция — это полином первой степени, квадратичная — второй, а кубическая — третьей:

линейная функция: $f(x) = a_1 x + a_0$

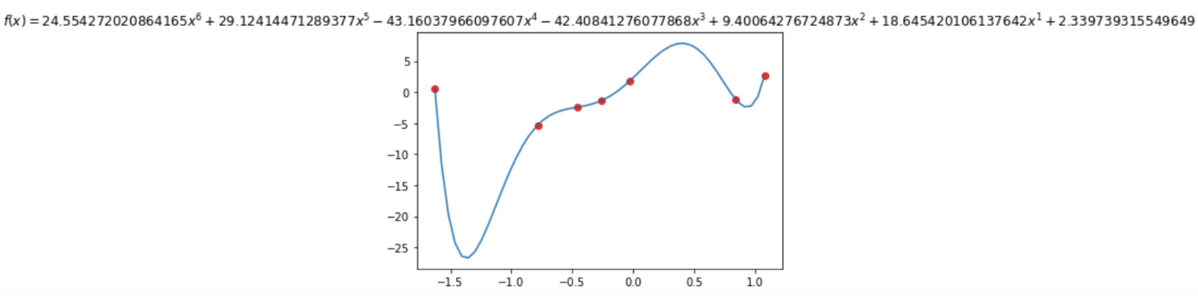
квадратичная функция: $f(x) = a_2 x^2 + a_1 x + a_0$

кубическая функция: $f(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0$

Полином может быть сколь угодно большой положительной степени, хоть миллионной. Тогда он гарантированно пройдёт через миллион и одну точку.

Нестрого можно представлять себе добавление степени к полиному как добавление в график возможности для ещё одного изгиба. Чем больше степень, тем больше можно согнуть график, а значит, тем лучше можно подстроить его под точки.

Посмотрим, как выглядит график полинома шестой степени, проведённый через семь точек:



Как видите, он может очень сильно изогнуться, подстраиваясь под точки. Для подстраивания используются коэффициенты. Поскольку точки сгенерированы случайно, то и коэффициенты получились некрасивыми. В реальных задачах они такими и бывают.

2. Свойства коэффициентов квадратичной функции

Квадратичная функция в обозначениях для коэффициентов полиномов, которые мы ввели, имеет следующий вид:

$$f(x) = a_2 x^2 + a_1 x + a_0$$

$$f(x) = a x^2 + b x + c$$

Коэффициенты a_0, a_1, a_2 определяют вид функции.

Когда мы увеличиваем коэффициент c (a_0), график поднимается, а когда уменьшаем — опускается.

При уменьшении коэффициента b (a_1) график сдвигается вправо и вниз, а при увеличении — влево и вверх. Но не совсем понятно, как именно он движется, — во всяком случае, не по прямой. Он движется по траектории параболы, перевёрнутой относительно оси OX.

Наконец, коэффициент a (a_2) определяет ширину раскрытия параболы: чем коэффициент больше, тем парабола уже, а чем коэффициент меньше, тем она шире, пока, наконец, не разгибается в другую сторону, изменяя направление ветвей. Такое изменение можно сравнить с тем, как гнётся железный прутик, если гнуть его, держась руками за концы.

3. Свойства коэффициентов кубической функции

Кубическая функция в обозначениях для коэффициентов полиномов, которые мы ввели, имеет следующий вид:

$$f(x) = a x^3 + b x + c x + d$$

$$f(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0$$

Давайте начнём разбираться с коэффициента a_0. Его роль точно такая же, как для линейной и квадратичной функций: он двигает кубическую параболу вверх и вниз.

Далее — коэффициент при первой степени a_1. Его уменьшение приводит к росту горба изгиба в левой части графика и углублению впадины в правой части. Соответственно, его увеличение, наоборот, выпрямляет график. При этом точка между ветвями остаётся неподвижной.

Теперь разберёмся с коэффициентом a_2. При увеличении растёт горб в левой части, а в правой, наоборот, ветвь графика прижимается к оси OY. При уменьшении эффект зеркальный: углубляется впадина в правой части, а левая ветвь прижимается к вертикальной оси.

Перейдём к коэффициенту при старшей степени, a_3. Его влияние на график похоже на влияние коэффициента при x^2 на параболу. При его увеличении график всё сильнее прижимается к оси OY, а при уменьшении начинает разгибаться, пока, наконец, пройдя ноль, не переворачивается.

4. Нахождение коэффициентов полиномов аналитически

Что же делать, если точек, например, тысяча? Значит ли это, что нужно подбирать коэффициенты полинома 999-й степени? Конечно, эту задачу невозможно решить вручную. Её решение можно свести к математической задаче, ответом в которой будут нужные значения коэффициентов.

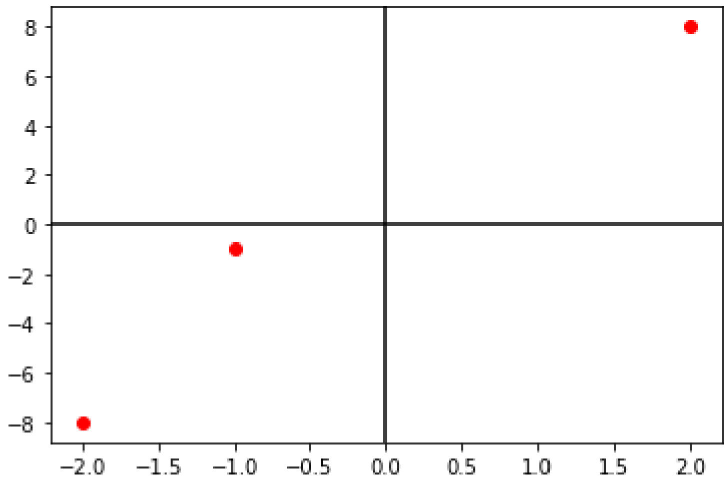
Например, если функция f проходит через точку (1; 2), это значит, что, если подставить единицу в f, то получится 2.

$(1; 2)$ – точка графика ф-ии $f \Rightarrow f(1) = 2$

В нашей задаче есть множество точек, через которые должна проходить функция f, которую мы подбираем для интерполяции. Давайте возьмём для примера три точки:

$$(-1; -1)$$
$$(2; 8)$$
$$(-2; -8)$$

Если отметить их на координатной плоскости, она будут выглядеть так:



Если f проходит через них, для неё должны выполняться такие соотношения:

$$(-1; -1) \qquad f(-1) = -1$$
$$(2; 8) \qquad f(2) = 8$$
$$(-2; -8) \qquad f(-2) = -8$$

Они должны выполняться все сразу, так что давайте поставим знак системы — он будет означать, что мы хотим одновременного выполнения всех этих соотношений:

$$\begin{cases} f(-1) = -1 \\ f(2) = 8 \\ f(-2) = -8 \end{cases}$$

Мы уже знаем, что для трёх точек степень полинома, который гарантированно сможет через них пройти, равна количеству точек минус один, то есть три минус один, то есть два.

степень полинома = количество точек $- 1 = 3 - 1 = 2$

Значит, мы ищем квадратичную функцию. Она имеет следующий вид:

$$f(x) = a_2 x^2 + a_1 x + a_0$$

Давайте подставим это выражение в систему, таким образом потребовав именно от квадратичной функции, а не от абстрактной, чтобы она проходила через данные точки:

$$\begin{cases} f(-1) = -1 \\ f(2) = 8 \\ f(-2) = -8 \end{cases} \Rightarrow \begin{cases} a_2(-1)^2 + a_1(-1) + a_0 = -1 \\ a_2(2)^2 + a_1(2) + a_0 = 8 \\ a_2(-2)^2 + a_1(-2) + a_0 = -8 \end{cases}$$

$$f(x) = a_2 x^2 + a_1 x + a_0$$

Упростим эти выражения, раскрыв скобки и перенеся всё в левую часть:

$$\begin{cases} a_2 - a_1 + a_0 + 1 = 0 \\ 4a_2 + 2a_1 + a_0 - 8 = 0 \\ 4a_2 - 2a_1 + a_0 + 8 = 0 \end{cases}$$

С одной стороны, это математическая запись требования к коэффициентам функции, чтобы она проходила через данный набор точек. А с другой стороны, это система из трёх уравнений с тремя неизвестными, вы можете помнить такие из школы. Методы решения таких систем лежат в области линейной алгебры, но мы можем воспользоваться языком Python и библиотекой SymPy:

In [9]:
```
a2, a1, a0 = symbols('a2,a1,a0')
```

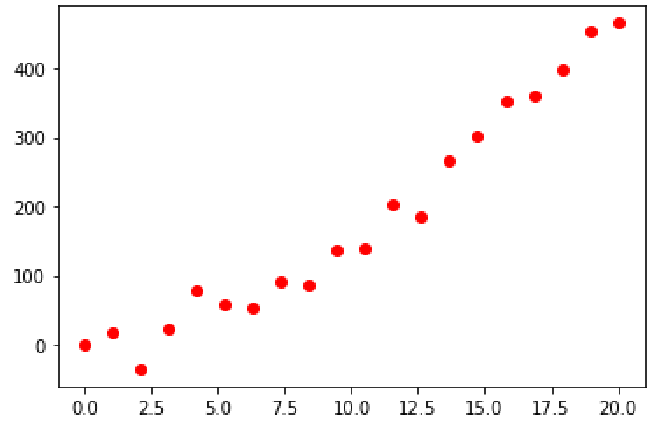
In [35]:
```
eq1_lp = a2 - a1 + a0 + 1
eq2_lp = 4*a2 + 2*a1 + a0 - 8
eq3_lp = 4*a2 - 2*a1 + a0 + 8
```

```
In [37]: nonlinsolve([eq1_lp, eq2_lp, eq3_lp], [a2, a1, a0])
Out[37]: {(−1, 4, 4)}
```

Это и есть наши коэффициенты. Они выведены в том порядке, в каком мы передали символьные переменные в nonlinsolve, то есть по убыванию, начиная от коэффициента старшей степени.

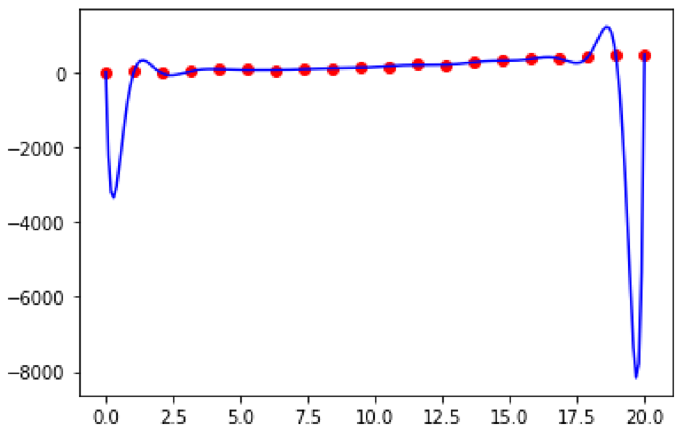
5. Недостатки интерполяции

К сожалению, у этого подхода есть проблемы, особенно для больших значений степени интерполяционного полинома. Давайте представим, что у нас есть двадцать точек, соответствующих замерам температуры в нагревающейся печи. Мы хотим подобрать функцию, чтобы вычислять значение температуры в промежуточные моменты времени или предсказывать, как будет идти нагревание в другой раз. Вот наши точки:

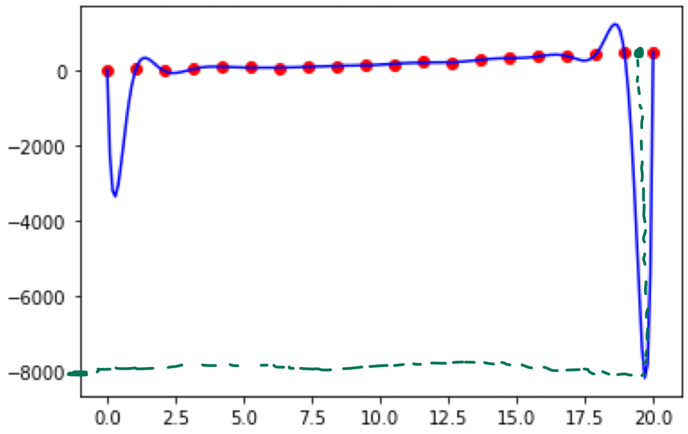


По оси X отложены секунды, а по оси Y — температура. Она немного скачет на коротких дистанциях, это связано с погрешностью измерений: у нас не очень точный градусник.

Попробуем приблизить их полиномом девятнадцатой степени и посмотрим, что у нас получится. Не будем вдаваться в вычисления, но после решения системы и построения графика у нас получится следующий результат:



Синий график принадлежит найденному нами полиному девятнадцатой степени. Можно обратить внимание на то, что масштаб графика с точками и с функцией сильно отличаются. Это так, потому что все точки по оси OY лежали в диапазоне от 0 до 500, а значения полинома лежат в диапазоне примерно от -8161 до 1200. Это уже довольно подозрительно: почему функция, которая будет предсказывать значения для точек, имеет масштаб, так сильно отличающийся от самих точек? Кроме того, если мы попробовали бы предсказать значения с помощью функции где-то между двумя последними точками, мы бы получили значение около −8000, хотя сами видим, что это большая ошибка: результат должен быть где-то в районе 500, у нас же получается, что температура опустилась ниже абсолютного нуля.



Такое поведение типично для полиномов высоких степеней при интерполяции. Похожие проблемы встречаются и при других способах интерполяции. Чрезмерное усложнение функции для того, чтобы она могла стать достаточно гибкой, приводит к её непредсказуемому поведению, и она перестаёт хорошо решать задачу приближения данных.