ЗАСТОСУВАННЯ МІНІМІЗАЦІЇ

НАБЛИЖЕННЯ ФУНКЦІЇ МНОГОЧЛЕНОМ

Наблизимо неперервну функцію f на відрізку $[a, b]$ многочленом

$$P(x) = a_0 + a_1 x + ... + a_m x^m.$$

Для цього розіб'ємо відрізок $[a, b]$ на n рівних частин $(m < n)$ точками $x_0 < x_1 < ... < x_n$. Сума квадратів відхилень у цих точках – це функція

$$F(a_0, a_1, ...a_m) = \sum_{i=0}^{n} (f(x_i) - P(x_i))^2.$$

Праву частину можна розглядати, як квадрат довжини вектора

$$\begin{pmatrix} P(x_0) - f(x_0) \\ P(x_1) - f(x_1) \\ ... \\ P(x_n) - f(x_n) \end{pmatrix} =$$

$$= \begin{pmatrix} a_0 + a_1 x_0 + ... + a_m x_0^m - f(x_0) \\ a_0 + a_1 x_1 + ... + a_m x_1^m - f(x_1) \\ ... \\ a_0 + a_1 x_n + ... + a_m x_n^m - f(x_n) \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & x_0 & x_0^2 & ... & x_0^m \\ 1 & x_1 & x_1^2 & ... & x_1^m \\ ... & & & & \\ 1 & x_n & x_n^2 & ... & x_n^m \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ ... \\ a_m \end{pmatrix} - \begin{pmatrix} f(x_0) \\ f(x_1) \\ ... \\ f(x_n) \end{pmatrix} = Aa - b,$$

де

$$a = (a_0, a_1, ..., a_m)^T, \quad b = (f(x_0), f(x_1), ..., f(x_n))^T,$$

A – матриця, в якої на місці з індексами i, j стоїть x_i^{j-1}.

Отже,

$$F(a) = (Aa - b)^2 = (Aa - b, Aa - b) = a^T A^T Aa - 2(a, A^T b) + (b, b).$$

Звідси

$$F'(a) = 2A^T Aa - 2A^T b, \qquad F''(a) = 2A^T A.$$

Остання матриця додатно визначена. Тому розв'язок системи

$$A^T Aa = A^T b,$$

тобто

$$a = (A^T A)^{-1} A^T b$$

дасть локальний (і глобальний) мінімум.

ВІДСТЕЖЕННЯ ПОЛОЖЕННЯ РОБОТА

1. Розглянемо спрощену задачу визначення положення робота в просторі. Вважатимемо, що робот їздить по підлозі в кімнаті в напрямках вправо-вліво і вперед-назад (без поворотів). Він орієнтується в просторі за допомогою відеокамери, на зображенні з якої він розпізнає світильники на стелі, що розташовані в точках $(x_1, y_1), ..., (x_n, y_n)$. Своє положення відносно $i-$го світильника робот визначив, як (dx_i, dy_i) (при всіх i). Ці спостереження зроблено з помилками. Визначити найбільш ймовірне положення робота.

Якби спостереження щодо $i-$го світильника були точні, то робот знаходився б у точці $(x_i - dx_i, y_i - dy_i)$. Нехай насправді він знаходиться у точці (x, y). Сума квадратів помилок є

$$F(x, y) = \sum_{i=1}^{n}((x - x_i + dx_i)^2 + (y - y_i + dy_i)^2).$$

Мінімізуючи цю функцію, можна знайти шукане положення.

2. Ускладнимо задачу, дозволяючи роботу повертатися. Якщо робот повернутий на кут α. Тоді справжній напрямок на світильник визначається домноженням напрямку (dx_i, dy_i), який бачить робот, на матрицю повороту

$$\begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix},$$

тобто

$$(dx_i \cos\alpha + dy_i \sin\alpha, -dx_i \sin\alpha + dy_i \cos\alpha).$$

Сума квадратів помилок є

$$F(x, y) = \sum_{i=1}^{n}((x - x_i + dx_i \cos\alpha + dy_i \sin\alpha)^2 + (y - y_i - dx_i \sin\alpha + dy_i \cos\alpha)^2).$$

3. Узагальнення цієї задачі на тривимірний простір - задача визначення положення в просторі (роботи, комп'ютерний зір).

НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ

Нехай ваги вузлів нейронної мережі є $w_1, ..., w_n$, функція реакції на вхідний сигнал x буде $F(w_1, ..., w_n, x)$. Під час навчання мережі подається набір вхідних сигналів $x_1, ..., x_m$ з очікуваними відповідями $y_1, ..., y_m$. Сума квадратів помилок

$$E(w_1, ..., w_n) = \sum_{k=1}^{m} (F(w_1, ..., w_n, x_k) - y_k)^2$$

може бути мінімізована для знаходження оптимальних ваг $w_1^0, ..., w_n^0$. При цьому параметри зсуваються на вектор $-kE'$, де $k > 0$ – мала додатня стала.

На практиці мінімізація відбувається не по всій навчальній інформації, а порційно, поступово наближаючи параметри до оптимальних.

МЕТОД ГРАДІЄНТНОГО СПУСКУ

Цей і наведені далі методи призначені для знаходження локальних мінімумів функцій багатьох змінних. Для знаходження максимумів функцію слід взяти з мінусом.

Для функції багатьох змінних напрямок, в якому функція локально спадає, протилежний до напрямку градієнта (похідної). Тому, знаходячись в точці $x_n \in \mathbb{R}^m$, можна зменшити значення функції f, обравши

$$x_{n+1} = x_n - \gamma f'(x_n)$$

при малому $\gamma > 0$. Наскільки мале обирати γ, залежить від функції і точки. Один з варіантів методу:

1. Обрати $x_0 \in \mathbb{R}^m,\ \gamma_0 > 0$.
2. При $n \geq 1$ обрати $\gamma = \gamma_0$, обчислити $x_{n+1} = x_n - \gamma f'(x_n)$.

Якщо $f(x_{n+1}) < f(x_n)$, перейти до наступного n. Інакше повторити цей крок, зменшивши γ вдвічі.

Якщо $\gamma < \varepsilon$, то завершити алгоритм.

https://uk.wikipedia.org/wiki/Метод_зворотного_поширення_помилки - Використання методу при навчанні нейронної мережі.

МЕТОД НЬЮТОНА

Грунтується на наближенні функції за допомогою формули Тейлора:

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) + 0.5f''(x_0)(x - x_0)^2.$$

Мінімізуємо праву частину. Прирівнюємо її похідну до нуля:

$$f'(x_0) + f''(x_0)(x - x_0) = 0.$$

Звідси

$$x = x_0 - (f''(x_0))^{-1} f'(x_0)$$

(справа - множення матриці на вектор). Отже, маємо ітеративну формулу

$$x_{n+1} = x_n - (f''(x_n))^{-1} f'(x_n),\ n \geq 0.$$

Ітерації слід зупинити, коли функція перестає зменшуватися, або крок занадто малий.

МЕТОД ЛЕВЕНБЕРГА-МАРКУАРДТА

Нехай
$$F(x) = \sum_{k=1}^{n} f_k^2(x),\ x \in A \subset \mathbb{R}^m.$$
Розглянемо векторну функцію
$$f(x) = (f_1(x), f_2(x), ..., f_n(x)).$$
Тоді
$$F(x) = (f(x), f(x)) \approx (f(x_0) + f'(x_0)(x - x_0), f(x_0) + f'(x_0)(x - x_0)) =$$
$$= (f(x_0), f(x_0)) + 2f(x_0)^T f'(x_0)(x - x_0) + (x - x_0)^T f'(x_0)^T f'(x_0)(x - x_0) = H(x).$$
Прирівняємо похідну до нуля:
$$H'(x) = 2f(x_0)^T f'(x_0) + 2f'(x_0)^T f'(x_0)(x - x_0) = \vec{0}.$$
Якщо позначити $J = f'(x_0)$, отримаємо
$$x = x_0 - (J^T J)^{-1} J f(x_0).$$
З іншого боку, за методом градієнтного спуску
$$F'(x_0) = 2f'(x_0)f(x_0),$$
$$x = x_0 - kJf(x_0) = x_0 - (rI)^{-1} J f(x_0),\ r = 1/k > 0.$$
Левенберг та Маркуардт запропонували комплексний варіант
$$x = x_0 - (J^T J + rI)^{-1} J f(x_0),$$
який виявився дуже ефективним.

ПОСИЛАННЯ

CERES-SOLVER - Бібліотека для мінімізації (Google)
http://ceres-solver.org/

Список проектів, що використовують її
http://ceres-solver.org/users.html

Використання для знаходження позиції робота в 3-вимірному просторі (SLAM)
http://ceres-solver.org/nnls_tutorial.html#bundle-adjustment

Короткий популярний опис SLAM
https://www.vrfocus.com/2018/08/what-is-slam-and-why-its-one-of-the-biggest-challenges-in-development/

Опис SLAM
https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping

Типова стаття по SLAM, що використовується на практиці
https://jakobengel.github.io/pdf/engel14eccv.pdf

SLAM, що використовується у роботах на Марсі
https://www-robotics.jpl.nasa.gov/publications/Mark_Maimone/rob-06-0081.R4.pdf

ЗАВДАННЯ

0. Ознайомитися з інформацією за посиланнями.

1. Розв'язати задачу оптимального положення робота в загальному випадку - вивести формули для x, y. Застосувати формули, якщо положення світильників $(1.2, 1.3), (1.2, -1.3), (0, 0), (-1.5, 0.5), (-1.5, -0.5)$, робот їх бачить в напрямках $(0.19, 0.31), (0.205, -2.294), (-0.99, -0.985)$, $(-2.503, -0.51), (-2.497, -1.51)$ відповідно.

2. Користуючись улюбленою мовою програмування: Згенерувати випадково положення робота та положення світильників у кімнаті 4x5 метрів, згенерувати випадкові помилки вимірювань з проміжку $[-0.05, 0.05]$ по кожній координаті. Розв'язати задачу знаходження положення робота і порівняти зі згенерованим значенням.

3*. Завдання 2 для узагальненої задачі з поворотом (кут теж згенерувати в проміжку $[-\pi, \pi]$). Для мінімізації реалізувати один з наведених вище методів.