Математика для Data Science. Линейная алгебра. Шпаргалка

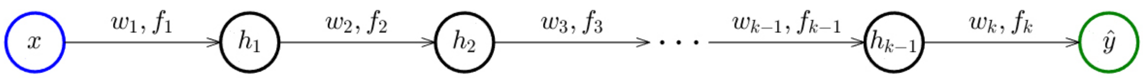
Содержание

Пятая неделя. Backpropagation

Одномерный backpropagation

Backpropagation — это метод вычисления градиентов функции потерь для нейросети путем обратного распространения ошибки.

Рассмотрим одномерную нейросеть в общем случае. Пусть она состоит из k линейных слоев с одним нейроном (без нейрона сдвига), веса этих слоев — $w_1, w_2, \dots, w_k$, функции активации $f_1, f_2, \dots, f_k$.



Вход обозначим за x, правильный ответ — за y, функцию потерь — за L. Обозначим также выходы линейных слоев с активацией:

$$h_1 = f_1(w_1 x), h_2 = f_2(w_2 h_1), \dots, h_{k-1} = f_{k-1}(w_{k-1} h_{k-2}), \hat{y} = f_k(w_k h_{k-1})$$

Частная производная по слою номер i равна

$$L(y, \hat{y})'_{w_i} = L(y, \hat{y})'_{\hat{y}} \cdot f'_k(w_k h_{k-1}) \cdot w_k \cdot f'_{k-1}(w_{k-1} h_{k-2}) \cdot w_{k-1} \cdot \dots \cdot f'_i(w_i h_{i-1}) h_{i-1}$$

Верны более компактные формулы:

$$L(y, \hat{y})'_{w_i} = L(y, \hat{y})'_{h_i} h_{i-1}$$

Чтобы эта формула работала и для $i = 1$, будем считать, что $h_0 = x$.

Матричное дифференцирование

Если функция отображает матрицу в число $f : \mathbb{R}^{n \times m} \to \mathbb{R}$, то будем записывать её градиент в матрицу:

$$\nabla_A f(A) = \begin{pmatrix} \frac{\partial f}{\partial a_{11}} & \frac{\partial f}{\partial a_{12}} & \cdots & \frac{\partial f}{\partial a_{1m}} \\ \frac{\partial f}{\partial a_{21}} & \frac{\partial f}{\partial a_{22}} & \cdots & \frac{\partial f}{\partial a_{2m}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial a_{n1}} & \frac{\partial f}{\partial a_{n2}} & \cdots & \frac{\partial f}{\partial a_{nm}} \end{pmatrix}$$

Верны следующие формулы:

- $\nabla_x x^T a = a$, или, что то же самое, $\nabla_x a^T x = a$
- $\nabla_x x^T A x = (A + A^T) x$
- $\nabla_A x^T A y = x y^T$

Точное решение для линейной регрессии с MSE

Пусть X — матрица, в которой по строкам записаны признаки объектов, y — вектор ответов для этих объектов, $w = (w_1, w_2, \dots, w_n)$ — веса линейной регрессии, $\hat{y} = Xw$ — вектор ответов модели для обучающей выборки. Тогда минимум среднеквадратичной функции потерь $L(y, \hat{y}) = \frac{1}{m}(y - \hat{y})^T (y - \hat{y})$ достигается при $w = (X^T X)^{-1} X^T y$ (если матрица $X^T X$ имеет полный ранг, в частности, если матрица X имеет ранг n).

Непрерывная всюду определенная функция $f : \mathbb{R}^n \to \mathbb{R}$ является *выпуклой*, если для любой пары точек $x, y \in \mathbb{R}^n$ выполнено $f(\frac{x}{2} + \frac{y}{2}) \leq f(\frac{x}{2}) + f(\frac{y}{2})$.

Среднеквадратичная функция потерь выпукла, поэтому у неё существует глобальный минимум.

Backpropagation для многомерного случая

Рассмотрим нейронную сеть с одним линейным слоем и функцией активации $f : \mathbb{R} \to \mathbb{R}$. Веса линейного слоя задаются матрицей $W \in \mathbb{R}^{n \times m}$, m — размер входа с учетом нейрона сдвига, n — размер выхода.

По функции f построим функцию $F : \mathbb{R}^n \to \mathbb{R}^n$, которая будет отвечать покоординатному применению функции f к вектору. То есть для $v \in \mathbb{R}^n$ выполнено $F(v) = (f(v_1), f(v_2), \ldots, f(v_n))$.

Для входного вектора $x \in \mathbb{R}^m$ и весов W значение функции потерь вычисляется как $L(y, \hat{y}) = L(y, F(Wx))$.

Утверждение. Пусть даны функции $f : \mathbb{R}^n \to \mathbb{R}^m$ и $g : \mathbb{R}^m \to \mathbb{R}$. Для $x \in \mathbb{R}^n$ пусть $y = f(x)$ и $z = g(y)$. Функцию f можно представлять себе, как набор функций $f_1, f_2, \ldots, f_m$ таких, что $f(x) = (y_1, y_2, \ldots, y_m) = (f_1(x), f_2(x), \ldots, f_m(x))$.

Градиент $\nabla_x z$ равен $J(y, x)^T \nabla_y z$, где $J_x(y)$ — матрица из частных производных (её ещё называют *матрицей Якоби*):

$$J_x(y) = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_n} \end{pmatrix}.$$

Правила дифференцирования многомерной сложной функции позволяет записать соотношения:

$$\frac{\partial L}{\partial w_i} = J_{w_i}(h_i)^T \frac{\partial L}{\partial h_i}$$

$$\frac{\partial L}{\partial x} = J_x(h_i)^T \frac{\partial L}{\partial h_i}$$