Математика для Data Science. Линейная алгебра. Решения задач

Содержание

Замечание. Вот этим цветом отмечены ссылки на страницы внутри этого файла.

Матричное дифференцирование

Задача 1

Пусть $A \in \mathbb{R}^{n \times n}$. Найдите $\nabla_x \, x^T Ax$. Ответ постарайтесь записать в матричном виде.

Подсказка. Как и на предыдущем шаге, можно сначала представить $x^T Ax$ в виде суммы множителей и найти её частные производные.

Решение. $x^T Ax = \langle x, Ax \rangle = \sum\limits_{k=1}^{n} x_k (Ax)_k = \sum\limits_{k=1}^{n} x_k \sum\limits_{m=1}^{n} a_{km} x_m = \sum\limits_{k=1}^{n} \sum\limits_{m=1}^{n} x_k a_{km} x_m$.

Здесь и далее запись $(B)_{ij}$ означает элемент матрицы B на позиции (i, j).

Найдём частную производную по x_i:

$$\frac{\partial}{\partial x_i}(x^T Ax) = \frac{\partial}{\partial x_i}\left(\sum_{k=1}^{n}\sum_{m=1}^{n} x_k a_{km} x_m\right) = \sum_{k=1}^{n}\sum_{m=1}^{n} \frac{\partial}{\partial x_i}\left(x_k a_{km} x_m\right)$$

Производная по x_i от произведения $x_k a_{km} x_m$ равна $\frac{\partial x_k}{\partial x_i} a_{km} x_m + x_k a_{km} \frac{\partial x_m}{\partial x_i}$. При этом $\frac{\partial x_k}{\partial x_i} \neq 0$ только при $k = i$ и аналогично $\frac{\partial x_m}{\partial x_i} \neq 0$ только при $m = i$. Значит, эта производная равна $a_{im} x_m + x_k a_{ki}$.

Итак, продолжая цепочку, получаем

$$\frac{\partial}{\partial x_i}(x^T Ax) = \sum_{m=1}^{n} a_{im} x_m + \sum_{k=1}^{n} x_k a_{ki} = (Ax)_i + (x^T A)_i = (Ax)_i + (A^T x)_i = ((A + A^T)x)_i$$

Тогда в матричном виде

$$\nabla_x x^T Ax = (A + A^T)x$$

Задача 2

Определение. *Следом* матрицы $A \in \mathbb{R}^{n \times n}$ называется число $\mathrm{tr}A = \sum\limits_{i=1}^{n} a_{ii}$.

Пусть $A \in \mathbb{R}^{n \times n}$ найдите $\nabla_A \mathrm{tr}(AB)$. Ответ также попробуйте записать в матричном виде.

Подсказка. Как и в прошлой задаче, надо представить $\mathrm{tr}(AB)$ в виде суммы и посчитать её частные производные.

Решение. Найдём диагональные элементы: $(AB)_{mm} = \sum\limits_{k=1}^{n} a_{mk} b_{km}$.

Тогда $\mathrm{tr}(AB) = \sum\limits_{m=1}^{n} (AB)_{mm} = \sum\limits_{m=1}^{n} \sum\limits_{k=1}^{n} a_{mk} b_{km}$. Теперь найдём частные производные:

$$\frac{\partial}{\partial a_{ij}}(\mathrm{tr}(AB)) = \frac{\partial}{\partial a_{ij}}\left(\sum_{m=1}^{n}\sum_{k=1}^{n} a_{mk} b_{km}\right) = \sum_{m=1}^{n}\sum_{k=1}^{n} \frac{\partial}{\partial a_{ij}}\left(a_{mk} b_{km}\right)$$

Отметим, что $\frac{\partial}{\partial a_{ij}}\left(a_{mk} b_{km}\right) = b_{km}$ только при $m = i$ и $k = j$, во всех остальных случаях производная равна нулю. Значит, знаки суммы исчезнут, и в итоге получим

$$\frac{\partial}{\partial a_{ij}}(\mathrm{tr}(AB)) = b_{ji}$$

В матричном виде ответ выглядит так

$$\nabla_A \mathrm{tr}(AB) = B^T$$

.

Задача 3

Пусть $x \in \mathbb{R}^n, A \in \mathbb{R}^{n \times m}, y \in \mathbb{R}^m$. Найдите $\nabla_A x^T A y$.

Замечание. Тут будет полезно циклическое свойство следа матрицы: $\mathrm{tr}(ABC) = \mathrm{tr}(BCA) = \mathrm{tr}(CAB)$ для случаев, когда размеры матриц позволяют делать такие циклические сдвиги.

Подсказка. Воспользуйтесь предыдущей задачей.

Решение. Поскольку $x^T A y$ — это просто число, то $x^T A y = \mathrm{tr}(x^T A y)$. Согласно замечанию, $\mathrm{tr}(x^T A y) = \mathrm{tr}(A y x^T)$. Далее, по предыдущей задаче $\nabla_A \mathrm{tr}(A y x^T) = (y x^T)^T = x y^T$. Итак, $\nabla_A x^T A y = x y^T$.

Точное решение для линейной регрессии с MSE

Задача 1

Чтобы найти градиент $(y - Xw)^T (y - Xw)$ по w, для начала нужно раскрыть скобки в выражении. Эта задача подскажет, как.

Докажите, что для всех матриц $A, B \in \mathbb{R}^{n \times m}$ и $C \in \mathbb{R}^{m \times k}$:

1. $(A + B)^T = A^T + B^T$
2. $(AC)^T = C^T A^T$

Чему будет равняться $(ACD)^T$ для $D \in \mathbb{R}^{k \times r}$? $(A_1 A_2 \cdot \dots \cdot A_l)^T$? Размеры матриц $A_1, A_2, \cdot \dots, \cdot A_l$ согласованы.

Замечание. Запись вида "$A \in \mathbb{R}^{n \times m}$" означает, что матрица A составлена из действительных чисел и у неё n строк и m столбцов.

Подсказка. Для решения второго пункта надо будет вспомнить формулу умножения матриц.

Решение.

1. $(A + B)^T = A^T + B^T$, ведь и у той, и у той матрицы на позиции (i, j) стоит $a_{ji} + b_{ji}$.
2. Обозначим $X = (AC)^T$, $Y = C^T A^T$. Пусть также запись вида a'_{ij} означает, что надо поменять местами индексы, то есть $a'_{ij} := a_{ji}$.

   $X = (AC)^T$, следовательно, $X^T = AC$. По правилу умножения матриц $x'_{ij} = \sum\limits_{p=1}^{m} a_{ip} c_{pj}$, то есть $x_{ji} = \sum\limits_{p=1}^{m} a_{ip} c_{pj}$.

   Далее, $y_{ji} = \sum\limits_{p=1}^{m} c'_{jp} a'_{pi} = \sum\limits_{p=1}^{m} c_{pj} a_{ip}$, то есть $x_{ji} = y_{ji}$.

Дважды применив полученную формулу, получим $(ACD)^T = D^T (AC)^T = D^T C^T A^T$, то есть порядок матриц поменялся на противоположный. Аналогично $(A_1 A_2 \cdot \dots \cdot A_l)^T = A_l^T \cdot \dots \cdot A_2^T A_1^T$.

Задача 2

Пусть ранг матрицы $A \in \mathbb{R}^{m \times n}$ равен n. Докажите, что матрица $A^T A$ имеет полный ранг.

Подсказки.

1. Если ранг $A^T A$ меньше n, то существует ненулевой вектор v такой, что $A^T A v = 0$. Покажите, что из этого следует, что столбцы матрицы A будут линейно зависимы.
2. $A^T A v = 0 \implies v^T A A^T v = 0$

Решение. Из задачи этой недели следует, что $\text{rank}(AA^T) \leq \text{rank}(A) = n$.

Докажем утверждение от противного: пусть $\text{rank}(A^TA) < n$. Тогда столбцы матрицы линейно зависимы и, значит, найдётся ненулевой вектор $\vec{v}$, что $A^TA\vec{v} = \vec{0}$.

Транспонировав равенство $A^TA\vec{v} = \vec{0}$, получим $\vec{v}^TA^TA = \vec{0}$, а тогда и $\vec{v}^TA^TA\vec{v} = 0$. Другими словами, $\langle A\vec{v}, A\vec{v}\rangle = ||A\vec{v}||^2 = 0$. А это возможно только если $A\vec{v} = 0$. Но если для ненулевого вектора $\vec{v}$ выполнено $A\vec{v} = 0$, то столбцы матрицы A линейно зависимы. Тогда ранг матрицы A должен быть меньше n, что неверно по условию. Итак, мы получили противоречие, значит, $\text{rank}(A^TA) = n$.

Backpropagation в общем случае

Задача 1

Рассмотрим нейронную сеть с одним линейным слоем и без функции активации (другими словами, с тождественной функцией активации). Веса линейного слоя задаются матрицей $W \in \mathbb{R}^{n\times m}$, m — размер входа с учетом нейрона сдвига, n — размер выхода.

Пусть функция потерь $L(y, \hat{y})$ — это длина вектора $(y - \hat{y})$, то есть скалярное произведение $\langle y - \hat{y}, y - \hat{y}\rangle$. Найдите $\nabla_x L$ двумя способами:

1. Через матрицу Якоби для композиции сложных функций
2. И напрямую: раскрыв скобки и применив полученные нами ранее правила дифференцирования.

Сравните ответы.

Замечание. Нам интереснее искать в этом случае $\nabla_W L$, поскольку оптимизируются в нейросети именно веса. Чтобы найти $\nabla_W L$ нужно обобщить на матрицы ряд уже доказанных фактов про векторное дифференцирование и сделать шаги аналогичные поиску $\nabla_W L$. Чтобы не тратить много времени, ограничимся поиском $\nabla_x L$.

Подсказка. В первом пункте рассмотрите функции $u(x) := y - Wx$ и $g(u) := \langle u, u\rangle$.

Решение.

1. Вспомним, что $\hat{y} = Wx$. Обозначим $u := y - Wx$, $L(y, \hat{y}) := z = g(u)$, где $g(u) := \langle u, u\rangle$. Тогда $\nabla_x z = J_x(u)^T\nabla_u z$, где $J_x(u)$ — матрица из частных производных $\frac{\partial u_i}{\partial x_j}$.

   Так как $u = y - Wx$, то $u_i = y_i - \sum\limits_{k=1}^{n} w_{ik}x_k$.

   Далее,
   $$(J_x(u))_{ij} = \frac{\partial u_i}{\partial x_j} = \frac{\partial}{\partial x_j}\left(y_i - \sum_{k=1}^{n} w_{ik}x_k\right) = 0 - \sum_{k=1}^{n} w_{ik}\frac{\partial x_k}{\partial x_j}$$

   Заметим, что $\frac{\partial x_k}{\partial x_j} = 1$ только при $k = j$, а во всех остальных случаях производная равна нулю. Значит,
   $$(J_x(u))_{ij} = -w_{ij}$$

   То есть
   $$J_x(u) = -W$$

   Остаётся найти $\nabla_u z = \nabla_u\langle u, u\rangle$. Как мы помним, скалярное произведение $\langle\vec{a}, \vec{b}\rangle$ можно записать в виде $\vec{a}^T\vec{b}$. Значит, $\nabla_u\langle u, u\rangle = \nabla_u u^Tu = \nabla_u u^TEu = (E + E^T)u = 2u$.

   Итак,
   $$\nabla_x z = J_x(u)^T\nabla_u z = -W^T \cdot 2u = -2W^T(y - Wx) = 2W^T(Wx - y)$$

2. Решим теперь вторым способом
   $$L(y, \hat{y}) = \langle y - \hat{y}, y - \hat{y}\rangle = (y - Wx)^T(y - Wx) = y^Ty - 2y^TWx + x^TW^TWx$$

Здесь мы воспользовались свойствами транспонирования из первой устной задачи этого урока.

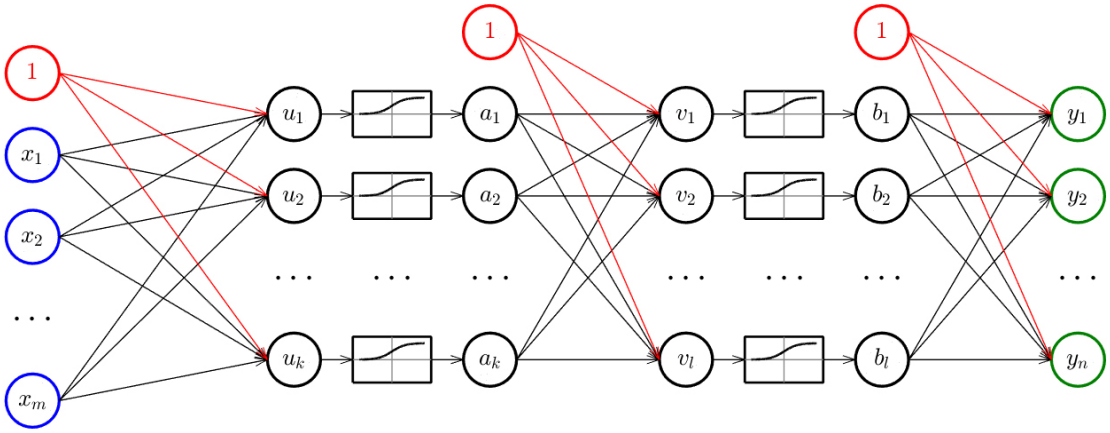
Теперь воспользуемся правилами дифференцирования:

$$\nabla_x(y^Ty - 2y^TWx + x^TW^TWx) = 0 - 2W^Ty + (W^TW + (W^TW)^T)x = 2(W^TWx - W^Ty) = 2W^T(Wx - y)$$

Ура, мы получили два одинаковых ответа!

Задача 2

Опишите, как будет работать backpropagation для сети ниже и функции потерь L:



Замечание. Далее будет написан очень подробный разбор того, какие формулы получаются для частных производных по весам нейросети. На сдаче задач такая степень подробности не требовалась. Самое главное в этой задаче — осознать, чему равна производная сигмоиды, а также как считать частные производные линейной функции многих переменных.

Решение. Функция активации в данном случае — сигмоида $\sigma(x) = \frac{1}{1+e^{-x}}$.

Для начала найдём её производную:

$$\sigma'(x) = \frac{-1}{(1+e^{-x})^2}\cdot(-e^{-x}) = \frac{e^{-x}}{(1+e^{-x})^2} = \sigma(x)\cdot\frac{1}{1+e^{-x}} = \sigma(x)\cdot\frac{1+e^{-x}-e^{-x}}{1+e^{-x}} = \sigma(x)\cdot\left(1 - \frac{e^{-x}}{1+e^{-x}}\right) = \sigma(x)(1-\sigma(x))$$

Частные производные по последнему слою

Обозначим веса, соединяющие b_i и y_j за $w_{ij}^{(3)}$, нейрон сдвига мы при этом считаем нулевым, то есть веса, соединяющие его и y_j обозначаем за $w_{0j}^{(3)}$. Значит, $y_t = \sum\limits_{p=0}^{l} w_{pt}^{(3)} b_p$.

Тогда

$$\frac{\partial}{\partial w_{ij}^{(3)}} y_t = \frac{\partial}{\partial w_{ij}^{(3)}}\left(\sum_{p=0}^{l} w_{pt}^{(3)} b_p\right) = \sum_{p=0}^{l} \frac{\partial}{\partial w_{ij}^{(3)}} w_{pt}^{(3)} b_p$$

При этом $\frac{\partial}{\partial w_{ij}^{(3)}} w_{pt}^{(3)} = 1$ только при $p = i$ и $t = j$, во всех остальных случаях производная равна нулю. Значит,

$$\frac{\partial}{\partial w_{ij}^{(3)}} y_j = b_i$$

а при $t \neq j$

$$\frac{\partial}{\partial w_{ij}^{(3)}} y_t = 0$$

Посчитаем частные производные по последнему (линейному) слою:

$$\frac{\partial L}{\partial w_{ij}^{(3)}} = \sum_{t=1}^{n} \frac{\partial L}{\partial y_t} \frac{\partial y_t}{\partial w_{ij}^{(3)}} = \frac{\partial L}{\partial y_j} \frac{\partial y_j}{\partial w_{ij}^{(3)}} = \frac{\partial L}{\partial y_j} b_i$$

Частные производные по предпоследнему слою

Теперь обозначим веса, соединяющие a_i и v_j за $w_{ij}^{(2)}$, нейрон сдвига мы при этом считаем нулевым, то есть веса, соединяющие его и v_j, обозначаем за $w_{0j}^{(2)}$. Значит, $v_p = \sum\limits_{t=0}^{k} w_{tp}^{(2)} a_t$. Проделав те же рассуждения, что для производной от y_j, получаем, что

$$\frac{\partial}{\partial w_{ij}^{(2)}} v_p = a_i \delta_{pj}$$

где $\delta_{pj} = 1$ только при $p = j$, а иначе $\delta_{pj} = 0$.

Замечание. Описанное выше δ_{pj} называется *символом Кронекера*.

Кроме того, $b_p = \sigma(v_p)$. А тогда

$$\frac{\partial}{\partial w_{ij}^{(2)}} b_p = \frac{\partial \sigma(v_p)}{\partial v_p} \frac{\partial v_p}{\partial w_{ij}^{(2)}} = \sigma(v_p)(1 - \sigma(v_p)) a_i \delta_{pj}$$

Посчитаем частные производные по предпоследнему слою:

$$\frac{\partial L}{\partial w_{ij}^{(2)}} = \sum_{p=1}^{l} \frac{\partial L}{\partial b_p} \frac{\partial b_p}{\partial w_{ij}^{(2)}} = \sum_{p=1}^{l} \frac{\partial L}{\partial b_p} \sigma(v_p)(1 - \sigma(v_p)) a_i \delta_{pj} = \frac{\partial L}{\partial b_j} \sigma(v_j)(1 - \sigma(v_j)) a_i$$

Далее,

$$\frac{\partial L}{\partial b_j} = \sum_{t=1}^{n} \frac{\partial L}{\partial y_t} \frac{\partial y_t}{\partial b_j} = \sum_{t=1}^{n} \frac{\partial L}{\partial y_t} \frac{\partial}{\partial b_j} \left(\sum_{p=0}^{l} w_{pt}^{(3)} b_p \right) = \sum_{t=1}^{n} \frac{\partial L}{\partial y_t} w_{jt}^{(3)}$$

Итого из последних двух цепочек равенств получаем

$$\frac{\partial L}{\partial w_{ij}^{(2)}} = \sum_{t=1}^{n} \frac{\partial L}{\partial y_t} w_{jt}^{(3)} \sigma(v_j)(1 - \sigma(v_j)) a_i$$

Или, так как $\sigma(v_j) = b_j$, можно переписать это так

$$\frac{\partial L}{\partial w_{ij}^{(2)}} = \sum_{t=1}^{n} \frac{\partial L}{\partial y_t} w_{jt}^{(3)} b_j(1 - b_j) a_i$$

Частные производные по первому слою

Наконец, обозначим веса, соединяющие x_i и u_j за $w_{ij}^{(1)}$, нейрон сдвига мы при этом считаем нулевым, то есть веса, соединяющие его и u_j, обозначаем за $w_{0j}^{(1)}$.

Проведя рассуждение, аналогичное написанному ранее, получим:

$$\frac{\partial}{\partial w_{ij}^{(1)}} a_p = \frac{\partial \sigma(u_p)}{\partial u_p} \frac{\partial u_p}{\partial w_{ij}^{(1)}} = \sigma(u_p)(1 - \sigma(u_p)) x_i \delta_{pj}$$

Посчитаем частные производные по первому слою:

$$\frac{\partial L}{\partial w_{ij}^{(1)}} = \sum_{p=1}^{k} \frac{\partial L}{\partial a_p} \frac{\partial a_p}{\partial w_{ij}^{(1)}} = \sum_{p=1}^{k} \frac{\partial L}{\partial a_p} \sigma(u_p)(1 - \sigma(u_p)) x_i \delta_{pj} = \frac{\partial L}{\partial a_j} \sigma(u_j)(1 - \sigma(u_j)) x_i$$

Далее,

$$\frac{\partial L}{\partial a_j} = \sum_{r=1}^{l} \frac{\partial L}{\partial b_r} \frac{\partial b_r}{\partial a_j}$$

Вспомним, что $\frac{\partial L}{\partial b_r}$ мы уже считали ранее. Значит, осталось разобраться с $\frac{\partial b_r}{\partial a_j}$:

$$\frac{\partial b_r}{\partial a_j} = \frac{\partial \sigma(v_r)}{\partial v_r}\frac{\partial v_r}{\partial a_j} = \sigma(v_r)(1-\sigma(v_r))\frac{\partial}{\partial a_j}\left(\sum_{q=0}^{k} w_{qr}^{(2)} a_q\right) = \sigma(v_r)(1-\sigma(v_r))w_{jr}^{(2)}$$

Собирая все формулы воедино, получаем

$$\frac{\partial L}{\partial w_{ij}^{(1)}} = \frac{\partial L}{\partial a_j}\sigma(u_j)(1-\sigma(u_j))x_i = \sum_{r=1}^{l}\frac{\partial L}{\partial b_r}\frac{\partial b_r}{\partial a_j}\sigma(u_j)(1-\sigma(u_j))x_i = \sum_{r=1}^{l}\frac{\partial L}{\partial b_r}\sigma(v_r)(1-\sigma(v_r))w_{jr}^{(2)}\sigma(u_j)(1-\sigma(u_j))x_i$$

Или, поскольку $\sigma(v_r) = b_r$ и $\sigma(u_j) = a_j$, получаем

$$\frac{\partial L}{\partial w_{ij}^{(1)}} = \sum_{r=1}^{l}\frac{\partial L}{\partial b_r} b_r(1-b_r)w_{jt}^{(2)} a_j(1-a_j)x_i$$

Наконец, чтобы получить окончательную формулу, подставим $\frac{\partial L}{\partial b_r}$, которые мы уже посчитали ранее:

$$\frac{\partial L}{\partial w_{ij}^{(1)}} = \sum_{r=1}^{l}\sum_{t=1}^{n}\frac{\partial L}{\partial y_t} w_{rt}^{(3)} b_r(1-b_r)w_{jt}^{(2)} a_j(1-a_j)x_i$$

Выпишем ещё раз все наши ответы:

$$\frac{\partial L}{\partial w_{ij}^{(3)}} = \frac{\partial L}{\partial y_j} b_i$$

$$\frac{\partial L}{\partial w_{ij}^{(2)}} = \sum_{t=1}^{n}\frac{\partial L}{\partial y_t} w_{jt}^{(3)} b_j(1-b_j)a_i$$

$$\frac{\partial L}{\partial w_{ij}^{(1)}} = \sum_{r=1}^{l}\sum_{t=1}^{n}\frac{\partial L}{\partial y_t} w_{rt}^{(3)} b_r(1-b_r)w_{jt}^{(2)} a_j(1-a_j)x_i$$

Как и в одномерном случае, мы получили, что при вычислении производной по более "ранним" слоям мы можем использовать значения, уже посчитанные для более "поздних" слоёв.