Глубинное обучение. Семинар 1. Автоматическое дифференцирование

1 Автоматическое векторное дифференцирование

Напоминание: векторное дифференцирование

Пусть $f : \mathbb{R}^n \to \mathbb{R}$ — функция от n переменных. Градиентом функции f в точке x_0 называется вектор ее частных производных в этой точке:

$$\nabla_x f\Big|_{x_0} = \left\{ \frac{\partial f}{\partial x_i}\Big|_{x_0} \right\}_{i=1}^{n}.$$

Аналогично для векторной функции $f_2 : \mathbb{R}^n \to \mathbb{R}^m$ вводится понятие матрицы Якоби в точке x_0:

$$J(x_0) = \left\{ \frac{\partial f_i}{\partial x_j}\Big|_{x_0} \right\}_{i,j=1}^{m,n}.$$

Подобные конструкции можно определять для производных функций любой размерности от аргументов любой размерности.

Мы будем позволять себе вольность обозначений и использовать для всех таких конструкций нотацию частных производных: $\frac{\partial f}{\partial x_0} = \nabla_x f\Big|_{x_0}$, $\frac{\partial f}{\partial x_0} = J(x_0)$ и т. д.

Для вычисления всех таких производных можно выписать один элемент многомерного массива, а затем постараться записать выражение для всей производной в векторной записи.

Для композиции функций $f(x) = g(y), y = h(x)$ верно цепное правило:

$$\frac{\partial f}{\partial x} = \frac{\partial g}{\partial y}\frac{\partial h}{\partial x}.$$

Под произведением здесь понимается суммирование по всем размерностям y. Например, если $f : \mathbb{R}^n \to \mathbb{R}^m$, $g : \mathbb{R}^k \to \mathbb{R}^m$, $h : \mathbb{R}^n \to \mathbb{R}^k$, то

$$\frac{\partial f}{\partial x} \in \mathbb{R}^{m\times n}, \frac{\partial g}{\partial y} \in \mathbb{R}^{m\times k}, \frac{\partial g}{\partial y} \in \mathbb{R}^{k\times n}$$

и умножение — это обычное матричное умножение.

Для решения задач нам также понадобится тот факт, что производная сигмоиды $\sigma(z) = \frac{1}{1+e^{-z}}$ вычисляется по следующему правилу:

$$\sigma'(z) = \sigma(z)(1 - \sigma(z)).$$

Также мы будем использовать обозначения $\odot$ для поэлементного произведения массивов одинакового размера, $[\cdot]$ — для индикаторной величины, $\mathrm{diag}(x)$ — диагональной матрицы с диагональю x, $\bar{1}$ — вектора из всех единиц, e_i — вектора с 1 в i-й позиции и остальными нулями. Все векторы считаются вектор-столбцами.

Задача 1. Градиенты для логистической регрессии

Задача. Найти градиент функционала качества логистической регрессии по параметрам модели с помощью прохода назад по вычислительному графу.

Напоминание. В модели логистической регрессии вероятность принадлежности каждого из N объектов к положительному классу моделируют следующим образом:

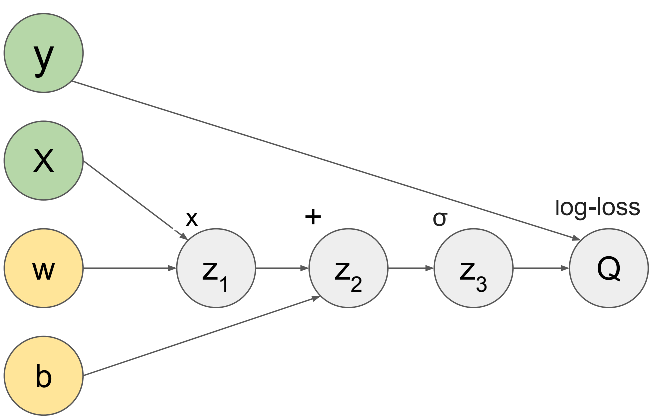
$$\hat{y} = \sigma(Xw + b\bar{1}), \quad y \in \mathbb{R}^N,\ X \in \mathbb{R}^{N\times D},\ w \in \mathbb{R}^D, b \in \mathbb{R},$$

D — число признаков. Для поиска оптимальных параметров w и b оптимизируют логарифм правдоподобия:

$$Q(w, b) = \sum_{i=1}^{N} L(y_i, \hat{y}_i) = -\sum_{i=1}^{N}(y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)) \to \min_{w,b}.$$

Обычно эту задачу с помощью градиентных методов, для чего нужно вычислять градиент функции Q по ее аргументам.

Решение. Изобразим функцию в виде вычислительного графа:



$$z_1 = Xw$$

$$z_2 = z_1 + b\bar{1}$$

$$z_3 = \sigma(z_2)$$

$$Q = -\sum_{i=1}^{N}(y_i \log z_{3,i} + (1 - y_i)\log(1 - z_{3,i}))$$

Зеленым отмечены входные данные, желтым - вершины, по которым нужно вычислить градиент. Во время прохода вперед мы сохраняем величины $z_1 - z_3$, которые могут фигурировать в выражениях для градиентов.

Чтобы найти градиенты, выполним проход назад:

1. $\dfrac{\partial Q}{\partial z_{3,i}} = -\dfrac{y_i}{z_{3,i}} + \dfrac{1 - y_i}{1 - z_{3,i}} \Rightarrow \dfrac{\partial Q}{\partial z_3} = -\dfrac{y}{z_3} + \dfrac{1 - y}{1 - z_3}$ (все операции поэлементные)

2. $\dfrac{\partial Q}{\partial z_2} = \dfrac{\partial Q}{\partial z_3}\dfrac{\partial z_3}{\partial z_2} = \dfrac{\partial Q}{\partial z_3}\,\mathrm{diag}(\sigma(z_2)(1 - \sigma(z_2))) = \dfrac{\partial Q}{\partial z_3} \odot \sigma(z_2) \odot (1 - \sigma(z_2)))$

3. $\dfrac{\partial Q}{\partial z_1} = \dfrac{\partial Q}{\partial z_2}\dfrac{\partial z_2}{\partial z_1} = \dfrac{\partial Q}{\partial z_2} I = \dfrac{\partial Q}{\partial z_2}$

4. $\dfrac{\partial Q}{\partial b} = \dfrac{\partial Q}{\partial z_2}\dfrac{\partial z_2}{\partial b} = \dfrac{\partial Q}{\partial z_2}\bar{1} \quad$ (сумма всех компонент $\dfrac{\partial Q}{\partial z_2}$)

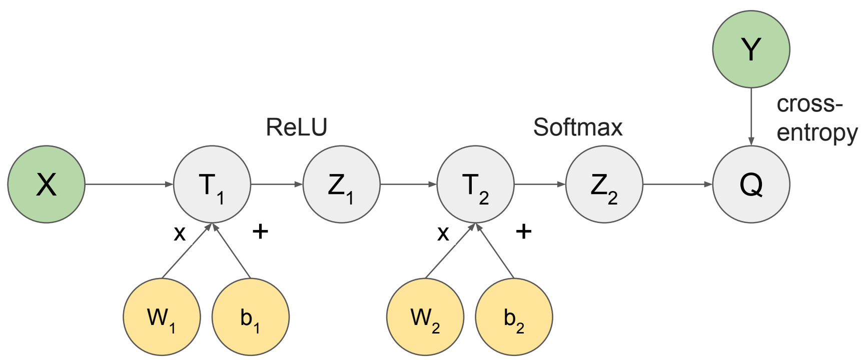
5. $\dfrac{\partial Q}{\partial w} = \dfrac{\partial Q}{\partial z_1}\dfrac{\partial z_1}{\partial w} = \dfrac{\partial Q}{\partial z_1} X$

Полученные формулы будут верны и в случае, если $X \in \mathbb{R}^D$ — единственный объект.

■

2 Градиенты для полносвязной нейросети в задаче многоклассовой классификации

В предыдущей задаче мы рассмотрели нейронную сеть с одним полносвязным для задачи бинарной классификации. Усложним модель — рассмотрим многослойную полносвязную нейронную сеть для задачи многокассовой классификации. Именно такую модель предлагается запрограммировать в практической части семинара и домашнем задании. Перерисуем граф для многослойной нейронной сети (для вывода формул достаточно рассмотреть двухслойную нейросеть):



Промежуточные величины теперь обозначаются большими буквами, потому что задают матрицы. В качестве нелинейности указана $\mathrm{ReLU}(t) = \max(t, 0)$, хотя, конечно, может использоваться и любая другая. Нашей ближайшей целью будет научиться пропускать градиент назад во всех компонентах этого вычислительного графа.

Ясно, что заодно мы покроем и случай многоклассовой логистической регрессии, что будет соответствовать однослойной нейронной сети.

Задача 2. Дифференцирование полносвязного слоя

Задача. Найдите производные функционала качества по параметрам полносвязного слоя (без нелинейности) и по входу этого слоя, считая известной производную по выходу слоя.

Решение. Полносвязный слой задается выражением $T = XW + b\bar{1}^T$ (в промежуточных слоях вместо X используется Z). Мы считаем, что нам известен градиент $\frac{\partial Q}{\partial T}$.

Запишем выражение для одной компоненты $\frac{\partial Q}{\partial W}$, используя цепное правило:

$$\frac{\partial Q}{\partial w_{ij}} = \sum_k \sum_\ell \frac{\partial Q}{\partial t_{k\ell}} \frac{\partial t_{k\ell}}{\partial w_{ij}}.$$

Найдем второй множитель, используя $t_{k\ell} = \sum_i x_{ki} w_{i\ell} + b_\ell$:

$$\frac{\partial t_{k\ell}}{\partial w_{ij}} = \begin{cases} x_{ki}, \ell = j \\ 0, \text{иначе} \end{cases}$$

Тогда получаем:

$$\frac{\partial Q}{\partial w_{ij}} = \sum_k \frac{\partial Q}{\partial t_{kj}} x_{ki} \Rightarrow \frac{\partial Q}{\partial W} = X^T \frac{\partial Q}{\partial T}$$

Для b:

$$\frac{\partial Q}{\partial b_i} = \sum_k \sum_\ell \frac{\partial Q}{\partial t_{k\ell}} \frac{\partial t_{k\ell}}{\partial b_i} = \sum_k \frac{\partial Q}{\partial t_{ki}} \cdot 1 \Rightarrow \frac{\partial Q}{\partial b} = \bar{1}^T \frac{\partial Q}{\partial T}$$

Производная по входу слоя:

$$\frac{\partial Q}{\partial x_{ij}} = \sum_k \sum_\ell \frac{\partial Q}{\partial t_{k\ell}} \frac{\partial t_{k\ell}}{\partial x_{ij}} = \sum_\ell \frac{\partial Q}{\partial t_{k\ell}} w_{j\ell} = \frac{\partial Q}{\partial T} W^T.$$

■

Задача 3. Дифференцирование нелинейности

Задача. Найдите производную функционала качества по входу слоя ReLU, считая известной производную по выходу слоя. Производную ReLU в нуле считаем равной 0.

Решение. Слой ReLU задается выражением $Z = ReLU(T) = \max(T, 0)$, операция max — покомпонентная. Вновь воспользуемся цепным правилом:

$$\frac{\partial Q}{\partial t_{ij}} = \sum_k \sum_\ell \frac{\partial Q}{\partial z_{k\ell}} \frac{\partial z_{k\ell}}{\partial t_{ij}}.$$

Считаем, что производную $\frac{\partial Q}{\partial Z}$ мы знаем.

$$\frac{\partial z_{k\ell}}{\partial t_{ij}} = \begin{cases} 1, (i = k) \,\&\, (j = \ell) \,\&\, (t_{ij} > 0) \\ 0, \text{иначе} \end{cases}$$

Тогда

$$\frac{\partial Q}{\partial t_{ij}} = \frac{\partial Q}{\partial z_{ij}} [t_{ij} > 0] \Rightarrow \frac{\partial Q}{\partial T} = \frac{\partial Q}{\partial Z} \odot [T > 0] \quad \text{(индикатор поэлементный).}$$

■

Для решения задачи многоклассовой классификации обычно используют softmax в качестве нелинейности на последнем слое, чтобы получить вероятности классов для каждого объекта:

$$z = softmax(t) = \left\{ \frac{exp(t_j)}{\sum_i exp(t_i)} \right\}_{j=1}^{K}, \quad K - \text{число классов}$$

В этом случае удобно оптимизировать логарифм правдоподобия:

$$L(y, z) = -\sum_{i=1}^{K} y_i \log z_i \to \min,$$

где $y_i = 1$, если объект принадлежит i-му классу, и 0 иначе. Записанная в таком виде, эта функция потерь совпадает с выражением для кросс-энтропии. Очевидно, что ее также можно переписать через индексацию, если через c обозначить класс данного объекта:

$$L(c, z) = -\log z_c \to \min$$

В таком виде ее удобно реализовывать.

Поскольку в функции потерь участвует только логарифм вероятности, то этот логарифм логично встроить в слой Softmax (получится слой log-softmax).

Задача 4. Дифференцирование кросс-энтропии

Задача. Найдите производную кросс-энтропии по входу — матрице логарифмов вероятностей.

Решение. В наших обозначениях матрица $Z \in \mathbb{R}^{N \times K}$ содержит вектора логарифмов вероятностей классов для объектов, записанные по строкам, матрица $Y \in \{0, 1\}^{N \times K}$ кодирует принадлежность классам ($y_{ij} = 0 \Leftrightarrow i$-й объект принадлежит j-му классу) . Тогда функционал качества записывается как

$$Q = -\sum_{i=1}^{N} \sum_{j=1}^{K} y_{ij} z_{ij}.$$

Значит,

$$\frac{\partial Q}{\partial Z} = -Y$$

Задача 5. Дифференцирование log-Softmax

Задача. Найдите производную функционала качества по входу слоя log-softmax.

Решение. Слой log-softmax можно записать следующим образом:

$$z_{ij} = t_{ij} - \log \sum_{k=1}^{K} \exp(t_{ik}).$$

Пользуемся цепным правилом:

$$\frac{\partial Q}{\partial t_{ij}} = \sum_m \sum_\ell \frac{\partial Q}{\partial z_{m\ell}} \frac{\partial z_{m\ell}}{\partial t_{ij}}$$

$$\frac{\partial z_{m\ell}}{\partial t_{ij}} = \begin{cases} 1 - \mathrm{softmax}_j(t_i), (m = i)\,\&\,(\ell = j) \\ -\mathrm{softmax}_j(t_i), (m = i)\,\&\,(\ell \neq j) \\ 0, \text{иначе} \end{cases}$$

$$\frac{\partial Q}{\partial t_{ij}} = \sum_{\ell} \frac{\partial Q}{\partial z_{i\ell}}([\ell = j] - \mathrm{softmax}_j(t_i)) = \frac{\partial Q}{\partial z_i}(e_j - \mathrm{softmax}_j(t_i)\bar{1})$$

Теперь вспомним (см. предыдущую задачу), что в векторе $\frac{\partial Q}{\partial z_i}$ только одно ненулевое значение (-1), стоящее в позиции c_i — номер класса, к которому принадлежит i-й объект. Получим:

$$\frac{\partial Q}{\partial t_{ij}} = -([j = c_i] - \mathrm{softmax}_j(t_i)) \Rightarrow \frac{\partial Q}{\partial t_i} = -(e_{c_i} - \mathrm{softmax}(t_i))$$

Это выражение можно записать и в матричном виде, если условиться, что операция softmax применяется к матрице построчно:

$$\frac{\partial Q}{\partial T} = -Y + \mathrm{softmax}(T)$$

■

Теперь предлагается перейти к практической части задания, где потребуются решения последних четырех задач.