Нейронные сети.
Вводная лекция

Алексей Колесов

Белорусский государственный университет

3 октября 2018 г.

Содержание

Теория вероятностей

- дискретные случайные величины (бросок кубика)
- непрерывные случайные величины
  - функция распределения: $F(\alpha) = p(x < \alpha)$
  - плотность распределения: $p(x) = \dfrac{dF}{dx}$,

    $$\int\limits_{-\infty}^{\infty} p(x)dx = F(\infty) - F(-\infty) = 0$$

Теория вероятностей

- совместная вероятность: $p(x, y)$
- независимые величины: $p(x, y) = p(x)p(y)$
- маргинализация: $p(x) = \sum_y p(x, y)$ или $p(x) = \int_Y p(x, y)dy$
- условная вероятность: $p(x|y) = \dfrac{p(x, y)}{p(y)}$
- $p(x, y) = p(x|y)p(y) = p(y|x)p(x)$
- $p(y|x) = \dfrac{p(x|y)p(y)}{p(x)}$ — формула Байеса

Пример

- медицинский тест имеет вероятность успеха 95%
- болезнь распространена у 1% популяции
- какова вероятность, что человек болен, если получил положительный тест?

Пример

Пусть t — результат теста, d — наличие болезни

$$p(t=1) = p(t=1|d=1)p(d=1) + p(t=1|d=0)p(d=0)$$

По Байсу:

$$p(d=1|t=1) = \frac{p(t=1|d=1)p(d=1)}{p(t=1|d=1)p(d=1) + p(t=1|d=0)p(d=0)} \approx 16\%$$

Парадокс Монти Холла

- есть три шкатулки, одна из них с призом
- игрок выбирает одну шкатулку случайным образом
- ведущий открывает одну из двух оставшихся и показывает, что она пустая
- выгодно ли игроку изменить решение о выбранной шкатулке?

Теорема Байеса

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$$

- $p(\theta)$ — априорная вероятность (prior probability)
- $p(D|\theta)$ — правдоподобие (likelihood)
- $p(\theta|D)$ — апостериорная вероятность (posterior probability)
- $p(D)$ — вероятность данных (evidence)

Оценки:

- $\theta_{ML} = \operatorname{argmax}_{\theta} p(D|\theta)$
- $\theta_{MAP} = \operatorname{argmax}_{\theta} p(\theta|D) = \operatorname{argmax}_{\theta} p(D|\theta)p(\theta)$

Функции потерь и регуляризация

- нахождение θ — оптимизация $\log p(D|\theta) + \log p(\theta)$
- конкретный вид функции зависит от задачи
- если шум нормальный с нулевым средним одинаковым отклонением, то максимальное правдоподобие $\iff$ $\log p(D|\theta)$ — RSS!
- если к тому же параметры распределены нормально, то получаем ridge-регрессию!
- обычно оптимизируем с помощью градиентного спуска

Содержание

Расстояние Кульбака-Лейблера

- в задачах классификации целевая метрика, обычно, — accuracy (кусочно-постоянна)
- оптимизируют KL-divergence: $\mathrm{KL}(P||Q) = \sum_i p_i \log \frac{p_i}{q_i}$ или $\mathrm{KL}(P||Q) = \int_x p(x) \log \frac{p(x)}{q(x)}$
- $KL(P||Q) \geqslant 0$ и $KL(P||Q) = 0 \iff P = Q$ почти всюду
- у нас: P — распределение порождённое данными, Q — классификатором

Перекрёстная энтропия

$$KL(P||Q) = \sum_i p_i \log p_i - \sum_i p_i \log q_i = -H(p) - H(p, q)$$

Для бинарной классификации:

$$L(\theta) = H(p_D, q_\theta) = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log \hat{y}_i(\theta) + (1 - y_i)\log\left(1 - \hat{y}_i(\theta)\right)\right)$$

Такая же ошибка выходит при решении логистической регрессии с помощью максимизации правдоподобия

Содержание

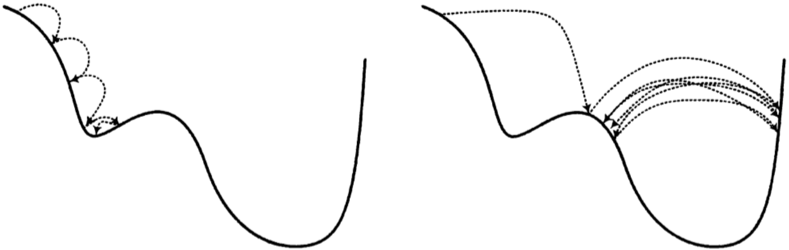
Градиентный спуск

- $x^{(t+1)} = x^{(t)} - \eta \nabla L(x^t)$, η — скорость обучения (learning rate)
- отлично работает для выпуклых функций
- (почему-то) хорошо работает для нейронных сетей
- нужно уметь брать производную в любой точке
- практически единственный алгоритм (не считая эвристик)

Если нет производной

- если умеем вычислять функцию: конечные разности
- если нет производной: суррогатная функция потерь
- если есть, но тривиальная: суррогатная функция потерь

Learning rate



Стохастический градиентный спуск

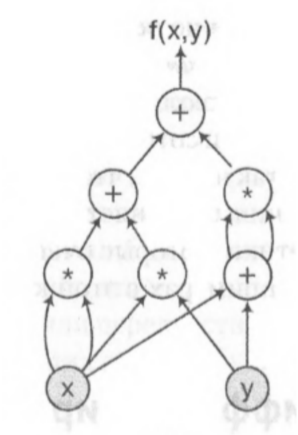
- если использовать напрямую, то надо вычислить производную «во всех драниках»
- можно использовать стохастический градиентный спуск — каждый раз брать новый случайный «драник»
- на практике используют минибатчи — несколько случайных «драников»

Содержание

Идея графа вычислений

- вычисление можно представить в виде графа: операции — вершины, потоки данных — дуги
- если умножение на матрицу Якоби — простая операция для каждой вершины, то легко применять градиентный спуск

Граф вычислений



$$f(x,y) = x^2 + xy + (x+y)^2$$

Chain rule

$$(f \circ g)'(x) = f(g(x))' = f'(g(x))g'(x)$$

Если x — вектор:

$$\nabla_x (f \circ g) = \begin{pmatrix} \frac{\partial f \circ g}{\partial x_1} \\ \vdots \\ \frac{\partial f \circ g}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial x_n} \end{pmatrix} = \frac{\partial f}{\partial g} \nabla_x g$$

Chain rule

Если g — векторная функция ($f = f(g_1(x), \ldots, g_k(x))$), то:

$$\nabla_x f = \frac{\partial f}{\partial g_1} \nabla_x g_1 + \ldots + \frac{\partial f}{\partial g_k} \nabla_x g_k$$

Т.е:

$$\nabla_x f = \nabla_x g \nabla_g f = \begin{pmatrix} \frac{\partial g_1}{x_1} & \ldots & \frac{\partial g_k}{\partial x_1} \\ \vdots & & \vdots \\ \frac{\partial g_1}{x_n} & \ldots & \frac{\partial g_k}{\partial x_n} \end{pmatrix} \nabla_g f$$

$\nabla_x g$ — матрица Якоби

Алгоритм обратного распространения

- проводим вычисления по графу (прямой проход)
- присвоим $\frac{\partial f}{\partial f} = 1$
- в обратном топологическом порядке, для каждой g:
  $\frac{\partial f}{\partial g} = \sum_v \frac{\partial f}{\partial v} \frac{\partial v}{\partial g}$, где есть дуги $g \to v$ (обратный проход)

Итоги

- вспомнили основы теории вероятностей
- вспомнили основы градиентного спуска
- рассмотрели граф вычислений

Литература

- С. Николенко, А. Кадурин, Е. Архангельская — Глубокое обучение (глава 2)