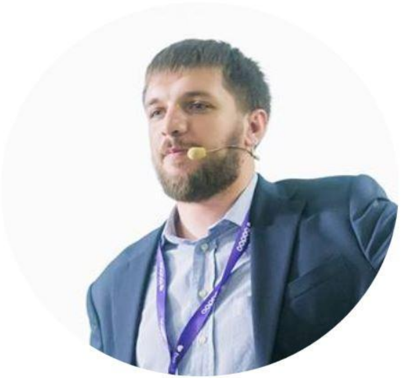
Машинное обучение

Лекция 3. Функции потерь и оптимизация

нетология



Алексей Кузьмин

Директор разработки; ДомКлик.ру

О спикере:

- Руковожу направлением работы с данными и Data Science
- Преподаю в Нетологии
- Работаю в IT с 2010 года (ABBYY, ДомКлик)
- Окончил МехМат МГУ в 2012 году

Я в Слаке:

@Alexey Kuzmin

Машинное обучение

С учителем

1

Алексей Кузьмин

Модель для задачи обучения с учителем

$$f(X,T) \to Y$$

- X - признаки объектов
- T - параметры модели
- Y - прогнозы модели

Задача обучения модели

$$f(X,T) \to Y$$

- Подобрать такие параметры T, чтобы получить максимально точный прогноз
- Два вопроса:
  - Что такое “максимально точный” прогноз
  - Как подбирать параметры?

Функция потерь

Что это такое и зачем нужно?

2

Алексей Кузьмин

Машинное обучение

нетология

Функция потерь

- Величина ошибки прогноза модели на объекте или выборке объектов.
- При обучении модели мы стараемся ее минимизировать

Зависит от типа решаемой задачи

Функции потерь для задачи регрессии

- Среднеквадратичная ошибка (mean squared error, MSE)
- Среднеабсолютная ошибка (mean absolute error, MAE)
- Более специфичные (например - функция потерь Хьюбера)

Mean Squared Error

$$J = \frac{1}{N} \sum_{i=1}^{N} (y_i - f(x_i))^2$$

Вывод

Рассмотрим самую простую модель:

$$h_\theta(x_i) = \theta_0 + \theta_1 x$$

Это простая линейная регрессия с двумя коэффициентами. Мы предполагаем, что целевое значение Y линейно зависит от входных данных X с добавлением шума.

$$Y = \theta_0 + \theta_1 x + \eta$$

Вывод

Будем считать, что шум имеет нормальное распределение с мат ожиданием 0 и дисперсией 1. Тогда

$$E[Y] = E[\theta_0 + \theta_1 x + \eta] = \theta_0 + \theta_1 x$$
$$Var[Y] = Var[\theta_0 + \theta_1 x + \eta] = 1$$

Распишем вероятность наблюдать значение yi для входящего xi

$$p(y_i|x_i) = e^{-\frac{(y_i-(\theta_0+\theta_1 x_i))^2}{2}}$$

Вывод

В предположении, что входные данные независимые и одинаково распределенные запишем правдоподобие нашей модели:

$$L(x, y) = \prod_{i=1}^{N} e^{-\frac{(y_i - (\theta_0 + \theta_1 x_i))^2}{2}}$$

Наша задача - подобрать такие параметры тета, что максимизировать правдоподобие (вероятность наблюдать такие y, при указанных значениях X)

Вывод

Перейдем от максимизации правдоподобия к максимизации логарифма правдоподобия (логарифм - монотонная функция)

$$l(x, y) = -\frac{1}{2} \sum_{i=1}^{N} (y_i - (\theta_0 + \theta_1 x_i))^2$$

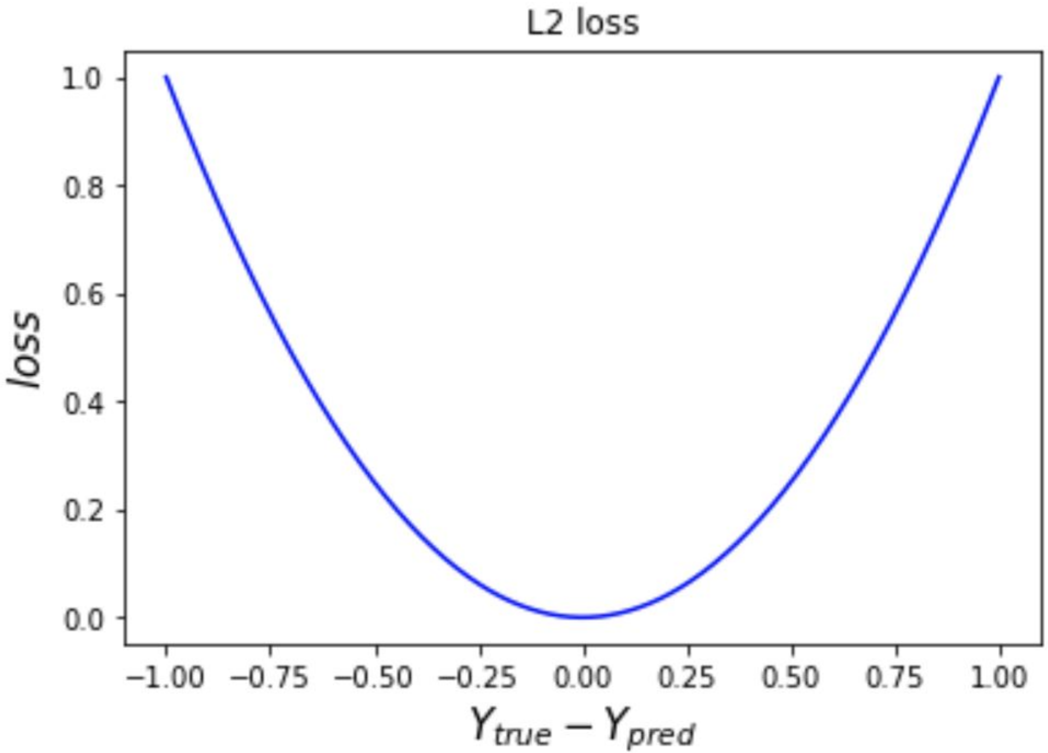
Вывод

Следующий (и последний) шаг - будем рассматривать $-l(x,y)$ и тогда наша цель будет ее минимизировать

$$-l(x, y) = \frac{1}{2} \sum_{i=1}^{N} (y_i - (\theta_0 + \theta_1 x_i))^2$$

Таким образом **оптимизация MSE-Loss - это просто максимизация правдоподобия**

Пример L2 loss



Mean Absolute Error

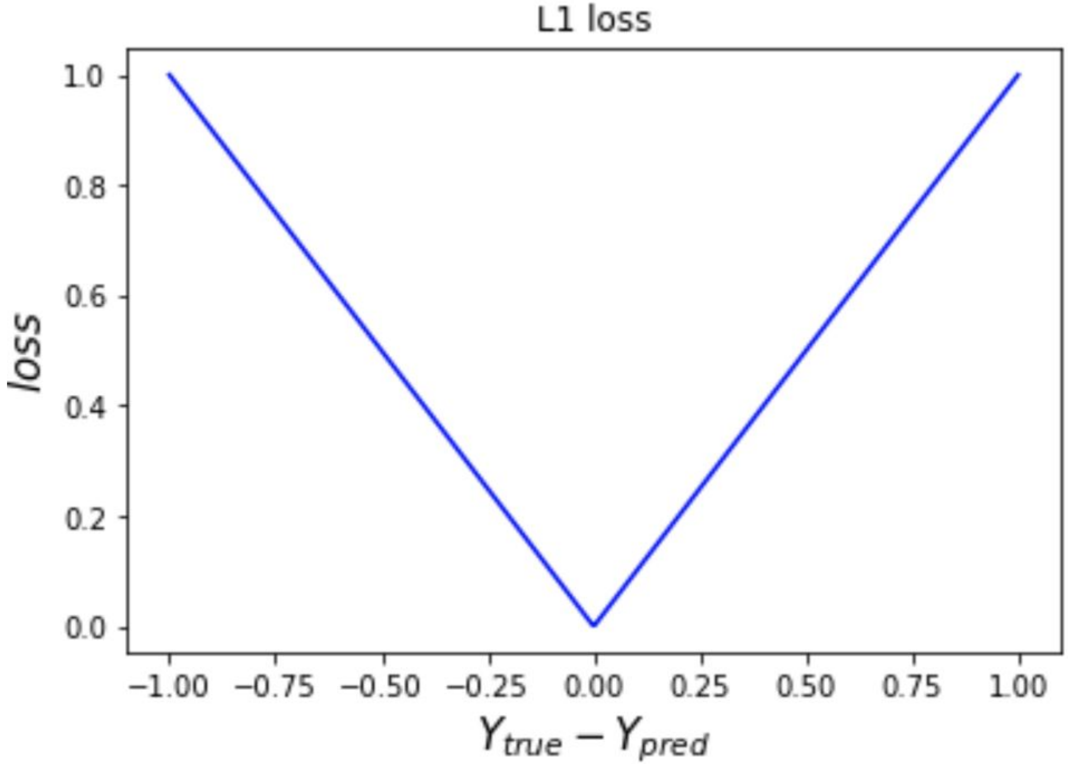
$$J = \frac{1}{N} \sum_{i=1}^{N} |y_i - f(x_i)|$$

Mean Absolute Error

L1-функция потерь - похожа на L2, но вместо того, чтобы брать квадрат расстояния -берется его модуль.

- L1 более устойчива к выбросам, потому что она не "взрывается" при больших значениях.
- У нуля она не такой гладкий как L2 и некоторые алгоритмы из-за этого могут хуже сходиться

Пример L1 loss



Huber Loss

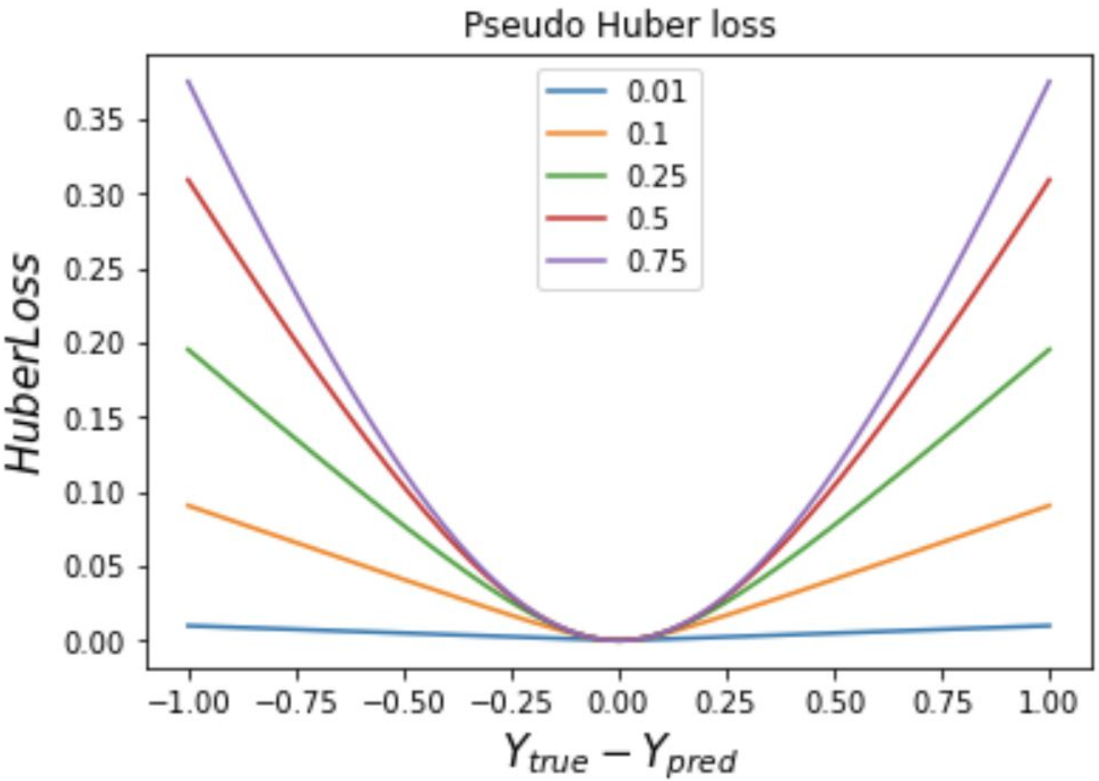
Функция потерь Хьюбера — это функция потерь, которая менее чувствительна к выбросам, чем квадратичная ошибка.

$$L_\delta(a) = \delta^2 \left(\sqrt{1 + (a/\delta)^2} - 1 \right)$$

a - в данной формуле - это y_true - y_pred

Дельта - гиперпараметр

Пример Huber loss



Функции потерь для задачи классификации

- Бинарная классификация (binary cross-entropy)
- Общая кросс-энтропия

Исходим из предположения, что наша модель возвращает *вероятность* принадлежности первому классу - h(xi)

Бинарная кросс-энтропия

$$J = -\sum_{i=1}^{N} y_i \log(h_\theta(x_i)) + (1 - y_i) \log(1 - h_\theta(x_i))$$

Вывод

Рассмотрим задачу классификации. Предположим, что ответ нашей модели $\hbar\theta$ (xi) получен на основе логистической регрессии $\sigma(W^{*}xi+b)$. Ее значения лежат в диапазоне от 0 до 1, что может быть интерпретировано, как вероятность, что xi принадлежит positive классу.

$$p(y_i = 1|x_i) = h_\theta(x_i)$$
$$p(y_i = 0|x_i) = 1 - h_\theta(x_i)$$

Вывод

Мы можем соединить это в одно уравнение:

$$p(y_i|x_i) = [h_\theta(x_i)]^{(y_i)}[1 - h_\theta(x_i)]^{(1-y_i)}$$

Вывод

Опять-таки из предположения, что наши данные независимы и одинаково распределены перейдем к правдоподобию:

$$L(x, y) = \prod_{i=1}^{N} [h_\theta(x_i)]^{(y_i)} [1 - h_\theta(x_i)]^{(1-y_i)}$$

Вывод

Точно так же, как в случае MSE, возьмьмем логарифм и инвертируем знак. Получим наш loss:

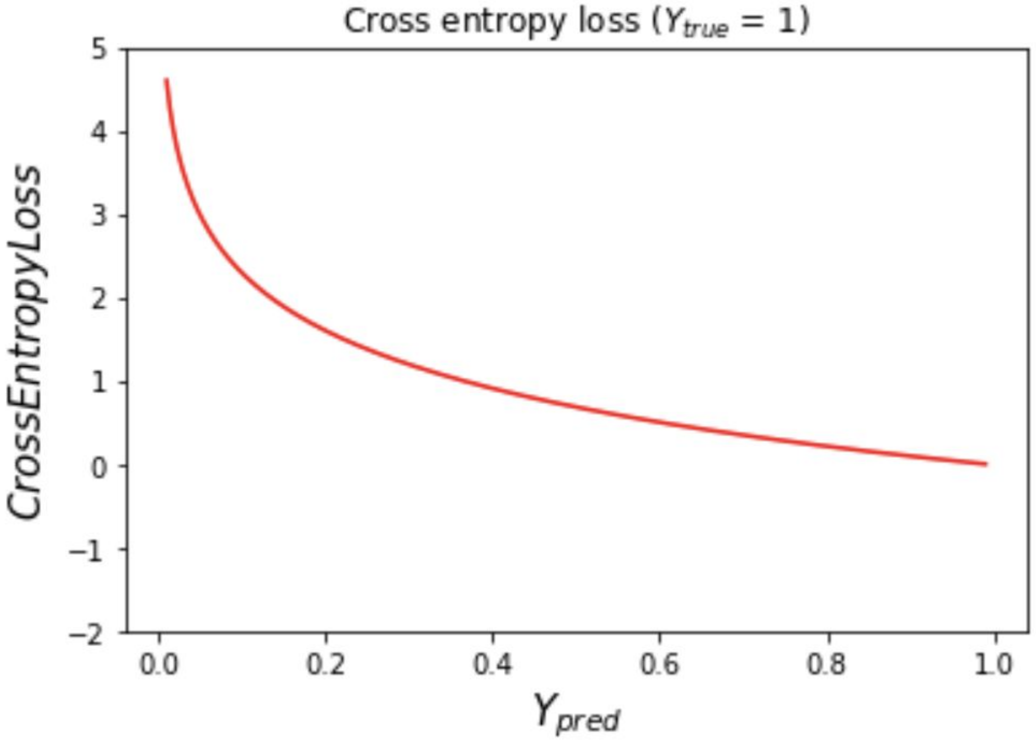
$$J = -\sum_{i=1}^{N} y_i \log(h_\theta(x_i)) + (1 - y_i) \log(1 - h_\theta(x_i))$$

Многоклассовая кросс-энтропия

- Основная идея - у нас ничего не меняется. Просто теперь вместо двух классов надо учитывать вероятности нескольких классов и теперь наш лосс примет такой вид:

$$-\sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij} \log(h_\theta(x_i)_j)$$

Пример кросс-энтропии



Выбор функции потерь:

- Модель и способ обучения - *персептрон vs градиентный спуск для логистической регрессии*
- Выборка - *L2 или L1 в задаче регрессии*

ВАЖНО! Функция потерь vs Метрика качества

Функция потерь - формальный функционал, который оптимизируется в процессе обучения модели

Метрика качества - способ оценить качество модели, возможно под другим углом.

Они не всегда совпадают

Оптимизация

Как найти минимум

3

Алексей Кузьмин

Машинное обучение

нетология

Оптимизация

Оптимизация — в математике, информатике и исследовании операций задача нахождения экстремума (минимума или максимума) целевой функции в некоторой области конечномерного векторного пространства, ограниченной набором линейных и/или нелинейных равенств и/или неравенств.

Локальные vs Глобальные

- Локальные методы: сходятся к какому-нибудь локальному экстремуму целевой функции. В случае унимодальной целевой функции, этот экстремум единственен, и будет глобальным максимумом/минимумом.
- Глобальные методы: имеют дело с многоэкстремальными целевыми функциями. При глобальном поиске основной задачей является выявление тенденций глобального поведения целевой функции.

Наличие производных у функции потерь

- прямые методы, требующие только вычислений целевой функции в точках приближений;
- методы первого порядка: требуют вычисления первых частных производных функции;
- методы второго порядка: требуют вычисления вторых частных производных, то есть гессиана целевой функции.

Градиентный спуск

Оптимизационный алгоритм для поиска локального минимума функции. Относится к методам первого порядка. Для поиска минимума делаем шаг в направлении, обратном градиенту функции.

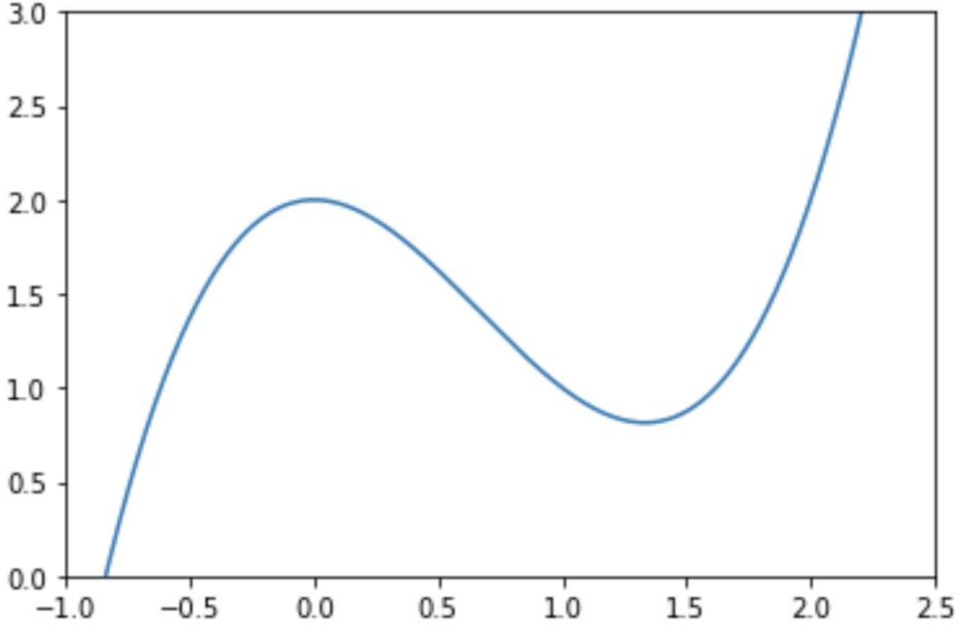
Основная рабочая “лошадка” поиска минимума во многих алгоритмах машинного обучения. Основной алгоритм оптимизации в нейронных сетях.

Алгоритм градиентного спуска

1. Задаем γ - "learning rate"
2. Выбираем начальное приближение x_0
3. for k = 0, 1, 2 ... do
   A. $s_k = -\nabla f(x_k)$
   B. $x_{k+1} = x_k + \gamma s_k$

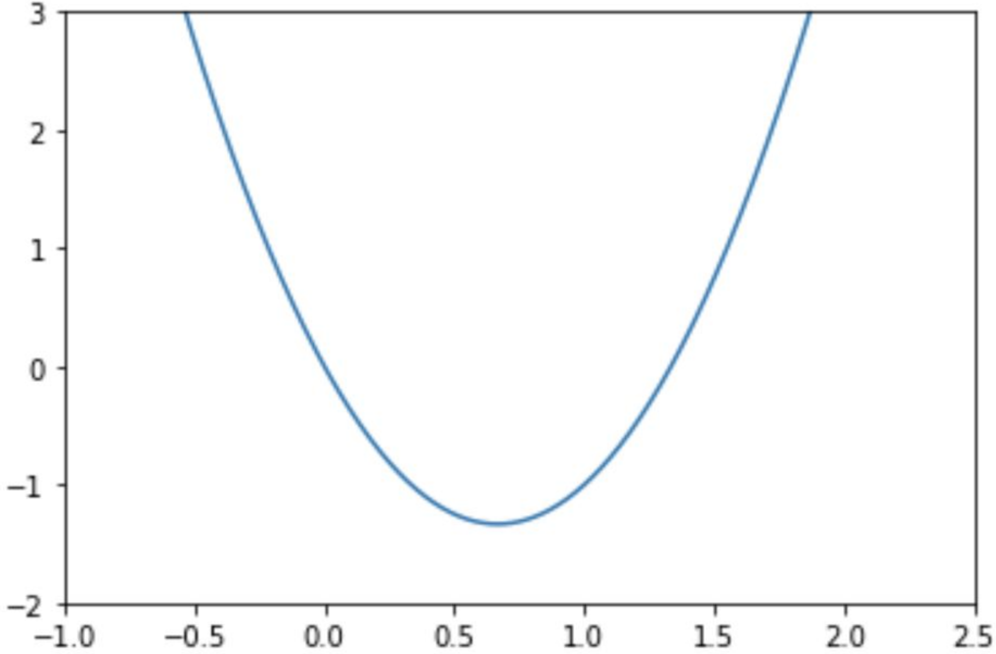
Пример

Функция x^3-2x^2+2



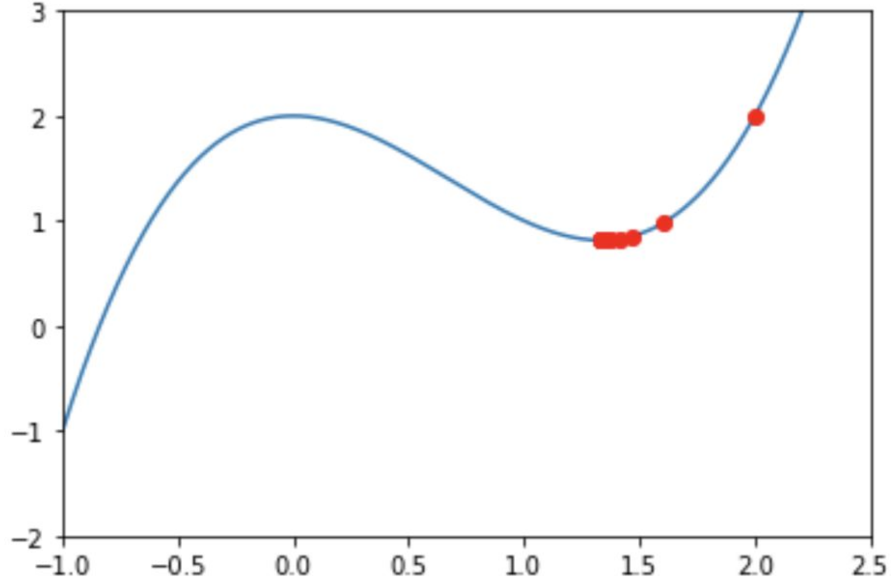
Пример

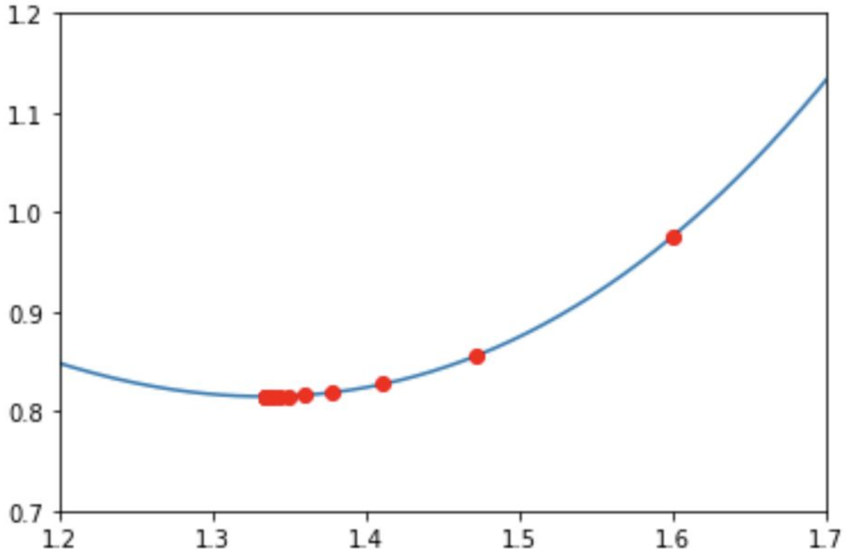
Ее производная 3x^2-4x



Пример

Оптимизация





Практика

Применим метод градиентного спуска к реальной задаче

Скорость градиентного спуска

1. Шаг градиентого спуска - проходим по всей выборке, прежде чем обновлять веса
2. Это достаточно долго для больших выборок
3. Нужны методы “ускорения”
4. Один из них - стохастический градиентный спуск

Стохастический градиентный спуск

Делаем маленький шаг для небольшой части датасета (мб даже одной точки)

Таким образом приближаем градиент по всей выборке градиентом по небольшому подмножеству

Практика

Применим метод стохастического градиентного спуска к реальной задаче

Другие методы ускорения градиентного спуска

1. Вокруг градиентного спуска есть еще много полуэврестических подходов по ускорению его сходимости
   a. Adam
   b. Rmsprop
   c. Nesterov Momentum
   d. И тп
2. https://habr.com/post/318970/

Оптимизация

Методы второго порядка

4

Алексей Кузьмин

Машинное обучение

нетология

Методы второго порядка

Рассмотренные нами выше методы - это методы первого порядка (требуется вычисление первой производной). Существуют методы оптимизации, требующие вычисления второй производной.

Они:

- Позволяют быстрее находить минимум (не линейная, а квадратичная аппроксимация функции в точки)
- Требует вычисления Гессиана - матрица всевозможных попарных производных
- Требует дважды дифференцируемости от функции

BFGS

Метод BFGS

- итерационный
- назван в честь его исследователей: Broyden, Fletcher, Goldfarb, Shanno.
- квазиньютоновский (гессиан вычисляется приближенно, исходя из сделанных до этого шагов)

Модификации:

- L-BFGS - ограниченное использование памяти (большое количество неизвестных).

BFGS

Метод эффективен и устойчив, поэтому зачастую применяется в функциях оптимизации. В SciPy в функции optimize по умолчанию применяется BFGS, L-BFGS-B.

https://habr.com/ru/post/333356/

Практика

Пример применения BFGS

Другие алгоритмы оптимизации

- Метод сопряжённых градиентов (Newton conjugate gradient method) http://www.machinelearning.ru/wiki/index.php?title=Метод_сопряжённых_градиентов
- Sequential Least SQuares Programming (SLSQP) Algorithm
- Симплекс метод
- ...

Итоги

5

Алексей Кузьмин

Что мы узнали сегодня

- Узнали что такое функции потерь и зачем они нужны. Рассмотрели типичные примеры таких функций для задачи регрессии и классификации
- Поговорили про методы оптимизации
- Рассмотрели подробно градиентный спуск и его модификацию - стохастический градиентный спуск

Домашнее задание

6

Алексей Кузьмин

Машинное обучение

нетология

ДЗ

- Прочитать про методы оптимизации для нейронных сетей https://habr.com/post/318970/. Взять код градиентного спуска для линейной регрессии (с занятия) и обучить ее
  - Методом nesterov momentum
  - Методом rmsprop
- Задание со звездочкой - доработать код логистической регрессии из первого занятия и обучить ее теми же методами для задачи классификации Ирисов (взять только два цветка - Iris Versicolor и Iris Virginica)

Спасибо за внимание

Алексей
Кузьмин

нетология