

Занятие 15: Классические Machine Learning алгоритмы

Пришло время посмотреть истоки

Talk About...

- Функция ошибок
- Градиентный спуск
- Линейная регрессия

Функции ошибок

Функции ошибок - определенные математические функции, которые позволяют оценить качество модели и являются важным фактором обучения.

Некоторые базовые функции ошибок:

- **MSE (*mean squared error*)**
- **MAE (*mean absolute error*)**
- **Cross-entropy Loss**
- **F1-score**
- **Log-cosh Loss**
- **Accuracy**

Метрика - система оценки качества модели, которая не используется для обучения. Может являться различными показателями, не всегда представляющими из себя строгую математическую функцию.

Главная задача метрик - наглядно продемонстрировать качество модели на том или ином этапе.

Поэтому при создании/выборе той или иной метрики делается упор на интерпретируемость результатов, а в достаточно большом количестве случаев еще и на бизнес-составляющую.

Допустим, нам нужно повысить качество изображения с отпечатками пальцев. В качестве функции ошибок мы можем использовать MAE по픽сельно между полученным изображением и эталонным. А в качестве метрики оценивать то, на сколько быстрее стала работать программа поиска отпечатка по изображению в базе данных.

Градиентный спуск

Обучение алгоритмов машинного обучения основывается на подборе различных параметров моделей, позволяющих алгоритму делать верные предсказания.

Во время обучения основная цель заключается в том, чтобы минимизировать функцию ошибок с использованием оптимальных параметров модели. Это называется **поиском глобального минимума функции ошибок**.

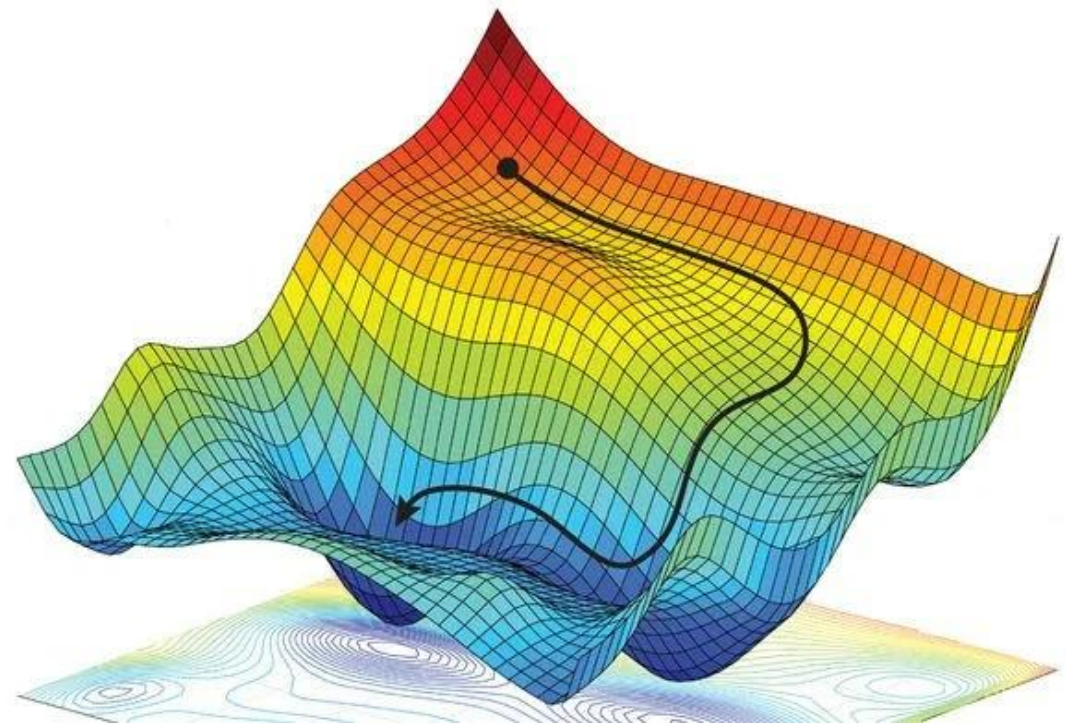
Одним из основных методов обучения алгоритмов машинного обучения является **градиентный спуск** и его разновидности.



Градиентный спуск

Градиентный спуск можно представить, как итеративный поиск маршрута с вершины горы до самой глубокой впадины.

В физическом смысле это напоминает шарик, который сталкивают с такой горы. Он катится с некоторым ускорением в сторону большего уклона до тех пор, пока не попадет в самую нижнюю точку.



Градиентный спуск

Для решения данной задачи используются **частные производные** (по направлению).

Поэтому важно понимать, что функция потерь (ошибок) **должна иметь производную**. При этом, оптимальным вариантом будет такая функция, производная которой будет довольно легко и наименее затратна по вычислениям, чтобы не нагружать систему каждый раз, когда нам необходимо провести итерацию обучения.

Просчитывая производные мы пытаемся определить направление с наклоном (то, куда должен скатиться шарик, чтобы оказаться ниже текущего значения).

Градиентный спуск

Но ведь шарик катится с какой-то скоростью?

Эта скорость зависит от параметров, которые мы будем задавать для алгоритма градиентного спуска.

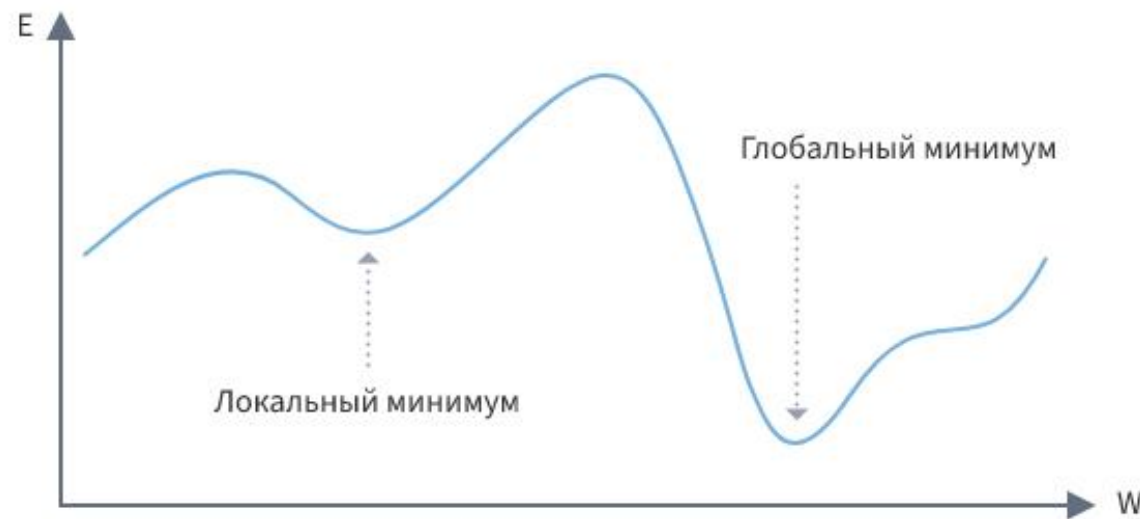
В различных вариантах алгоритма список таких параметров может несколько отличаться. Даже более того. В некоторых вариантах скорость будет динамически меняться в зависимости от "ландшафта" и текущей скорости.

Иногда для того, чтобы преодолеть какое-либо препятствие на пути шарика, нам придется придавать ему некоторый импульс или выбирать не самое оптимальное направление движения (допустим, случайное).

Градиентный спуск

Такие ситуации могут возникать, если шарик попал в низину, но при этом данная низина не будет являться самой глубокой.

Данное явление называется локальный минимум. То бишь для данной локации (местоположения) эта точка - самое низкое место в округе, но на глобальной карте есть места и поглубже.



Градиентный спуск

Одни из популярных вариаций градиентного спуска:

- **Adagrad** - адаптивный алгоритм, который уменьшает скорость изменения тех или иных параметров, чем ближе они к своему оптимуму.
- **RMSProp** - основная идея в том, чтобы не просто адаптивно менять скорость в зависимости от текущих условий, но и учитывать историю изменений.
- **Adam** - самый популярный метод, чаще всего используется первым и единственным методом. Работает сходим с rmsprop образом, за исключением математических оптимизаций (использование сглаживания).

Данные вариации призваны адаптировать скорость обучения (шарика) под текущие условия.

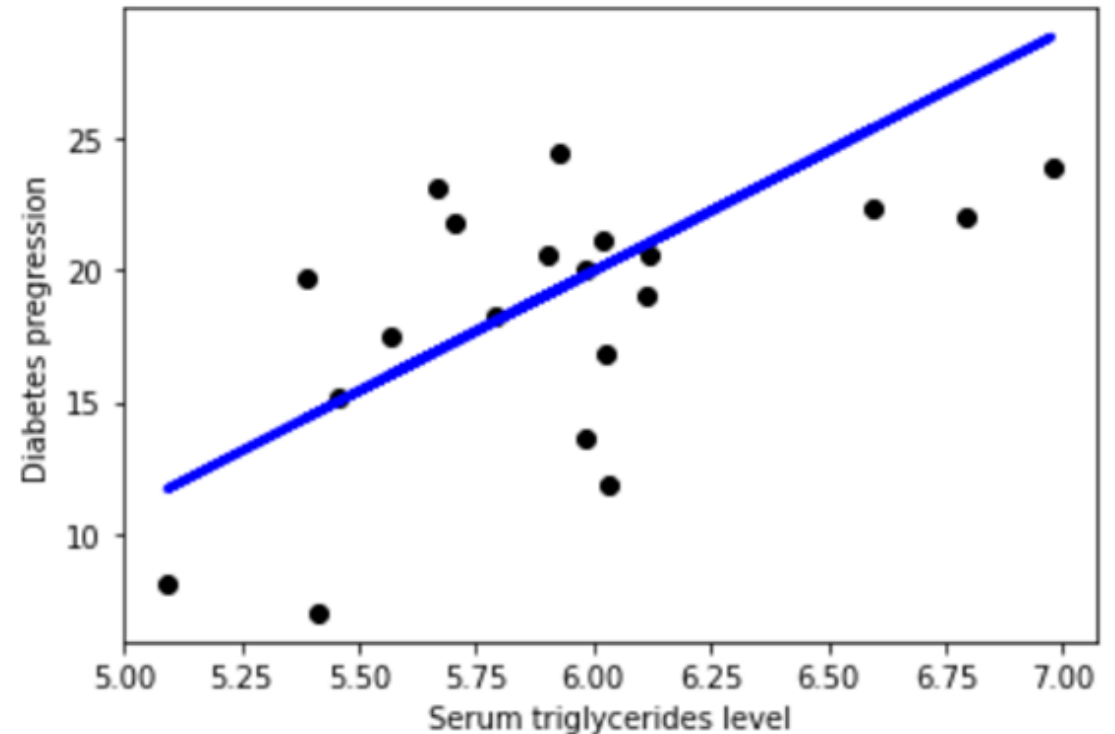
Линейная регрессия

Базовый алгоритм машинного обучения.

Линейная регрессия - это статистический метод, используемый для моделирования линейной зависимости между одной или несколькими независимыми переменными (признаками) и зависимой переменной (целевой переменной).

$$f(x) = w * x + b,$$

где w и b - параметры, которые необходимо вычислить, для осуществления предсказания.



Лекарство от скуки (домашка)

- Решить задачу с применением линейной регрессии. Объяснить выбор функции ошибок и метрики. Дать субъективную оценку качества, глядя на значения, полученные при помощи функции ошибок и метрик.
- *Придумать каким образом данная задача будет полезна для бизнеса. Почитать про бизнес-метрики. Связать целевую метрику и бизнес-метрики. **
- *Разработать собственную метрику качества, применимую к задаче. **

Ссылки

[Разработка R&D-проектов в сфере ИИ](#)