Градиентный спуск

КУХАЛЬСКИЙ НИКОЛАЙ ГЕННАДЬЕВИЧ

ГРАДИЕНТНЫЙ СПУСК

Градиентный спуск — самый используемый алгоритм обучения, он применяется почти в каждой модели машинного обучения.

Градиентный спуск — это, по сути, и есть то, как обучаются модели.

Метод градиентного спуска с некоторой модификацией широко используется для обучения персептрона и глубоких нейронных сетей , и известен как метод обратного распространения ошибки.

ГРАДИЕНТНЫЙ СПУСК

Градиентный спуск — метод нахождения минимального значения функции потерь (существует множество видов этой функции).

Минимизация любой функции означает поиск самой глубокой впадины в этой функции. Имейте в виду, что функция используется, чтобы контролировать ошибку в прогнозах модели машинного обучения.

Поиск минимума означает получение наименьшей возможной ошибки или повышение точности модели. Мы увеличиваем точность, перебирая набор учебных данных при настройке параметров нашей модели (весов и смещений).

Экстремум функции

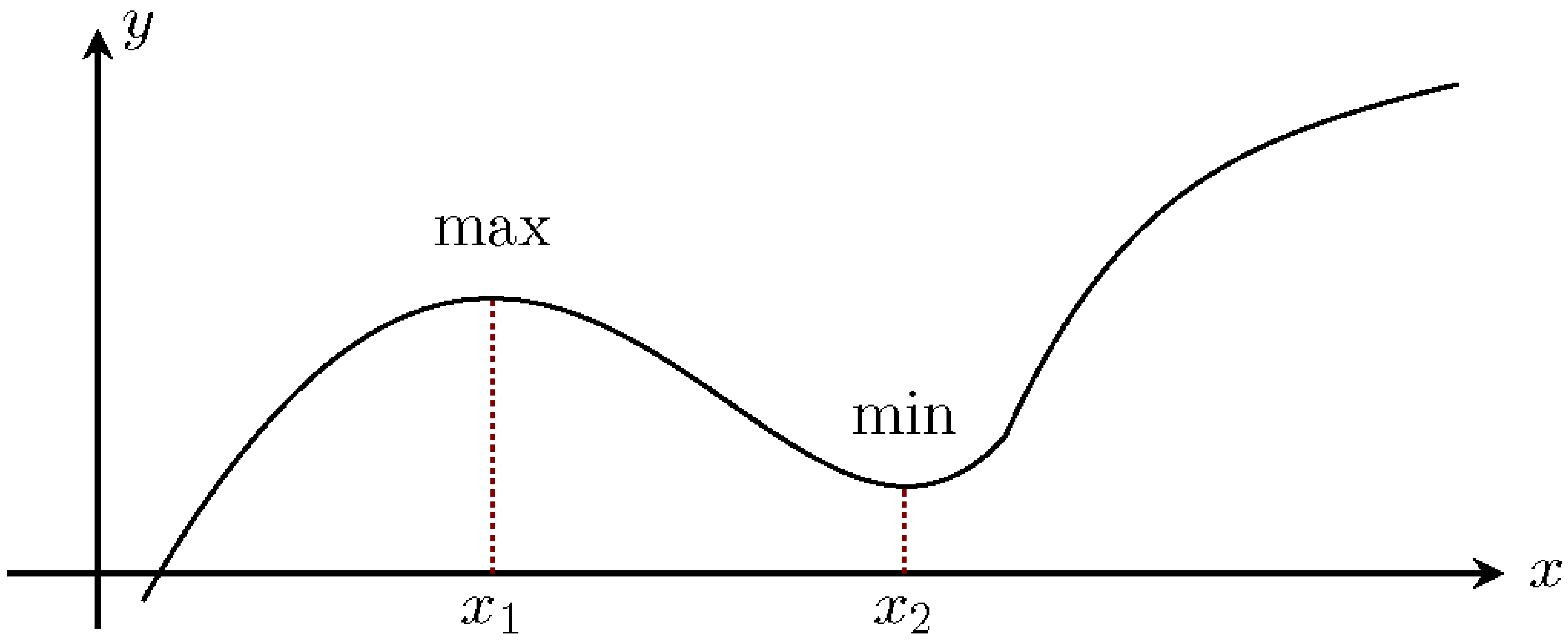
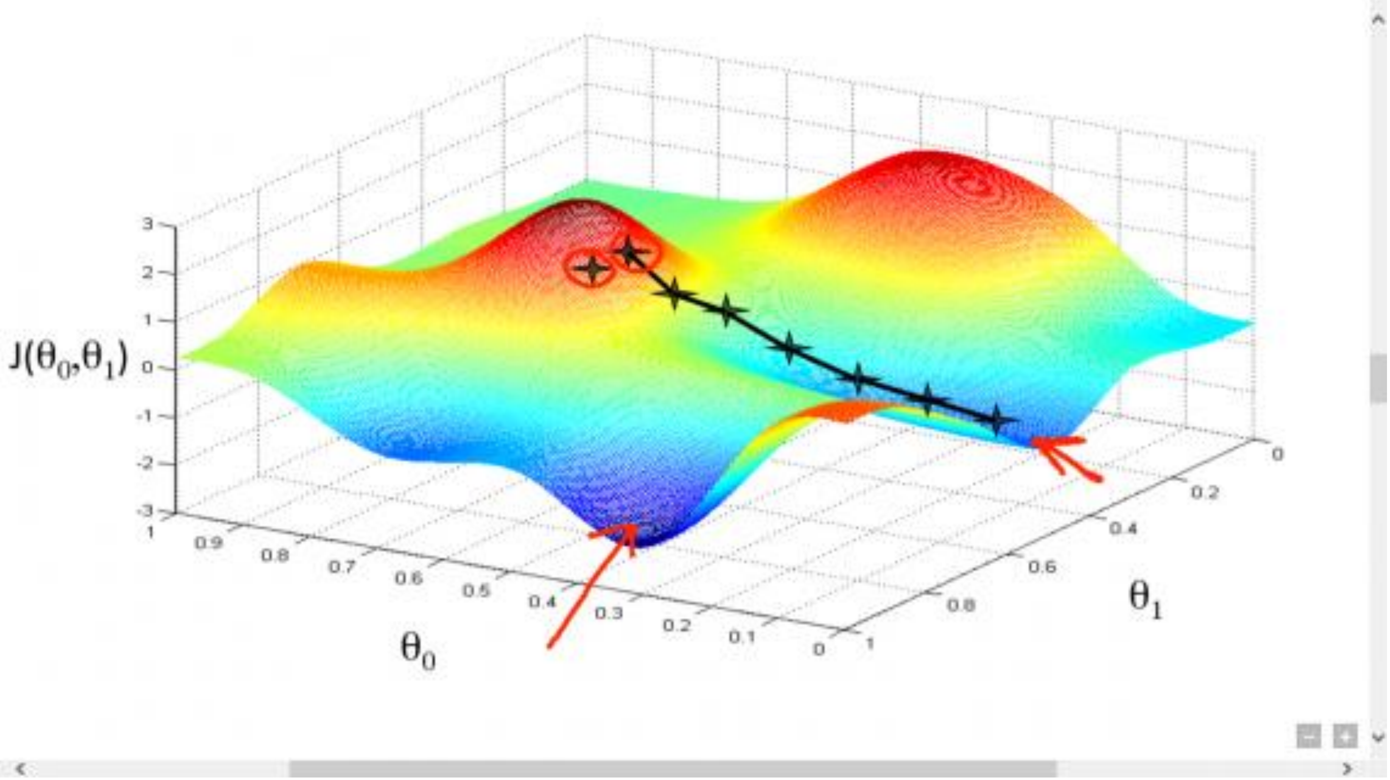


Рисунок – пример экстремумов функции

Экстремум функции



Суть алгоритма – процесс получения наименьшего значения ошибки.

ГРАДИЕНТНЫЙ СПУСК

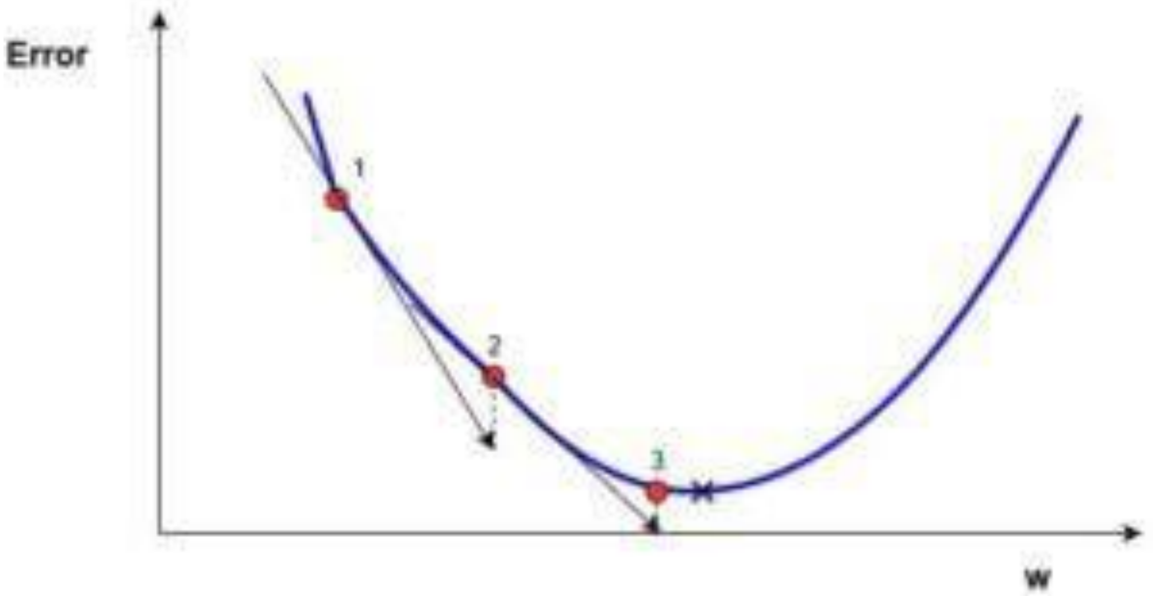


График функции потерь (названный «Ошибка» с символом «J») с одним весом.

Если вычислить наклон (производная dJ/dw) функции потерь относительно одного веса, то получим направление, в котором нужно двигаться, чтобы достичь локальных минимумов.

ГРАДИЕНТНЫЙ СПУСК

Функция потерь предназначена для отслеживания ошибки с каждым примером обучениям, в то время как производная функции относительного одного веса – это то, куда нужно сместить вес, чтобы минимизировать ее для этого примера обучения. Вы можете создавать модели даже без применения функции потерь. Но вам придется использовать производную относительно каждого веса (dJ/dw).

Когда определили направление, в котором нужно подтолкнуть вес, нужно понять, как это сделать. Используем коэффициент скорости обучения, его называют гиперпараметром. Гиперпараметр – значение, требуемое вашей моделью, о котором мы действительно имеем очень смутное представление. Обычно эти значения могут быть изучены методом проб и ошибок. Здесь не так: одно подходит для всех гипе-параметров. Коэффициент скорости обучения можно рассматривать как «шаг в правильном направлении», где направление происходит от dJ/dw.

Это была функция потерь построенная на один вес. В реальной модели мы делаем всё перечисленное выше для всех весов, перебирая все примеры обучения. Даже в относительно небольшой модели машинного обучения у вас будет более чем 1 или 2 веса. Это затрудняет визуализацию, поскольку график будет обладать размерами, которые разум не может себе представить.

ГРАДИЕНТНЫЙ СПУСК

dJ/dw зависит от вашего выбора функции потерь. Наиболее распространена функция потерь среднеквадратичной ошибки.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

Производная этой функции относительно любого веса (эта формула показывает вычисление градиента для линейной регрессии):

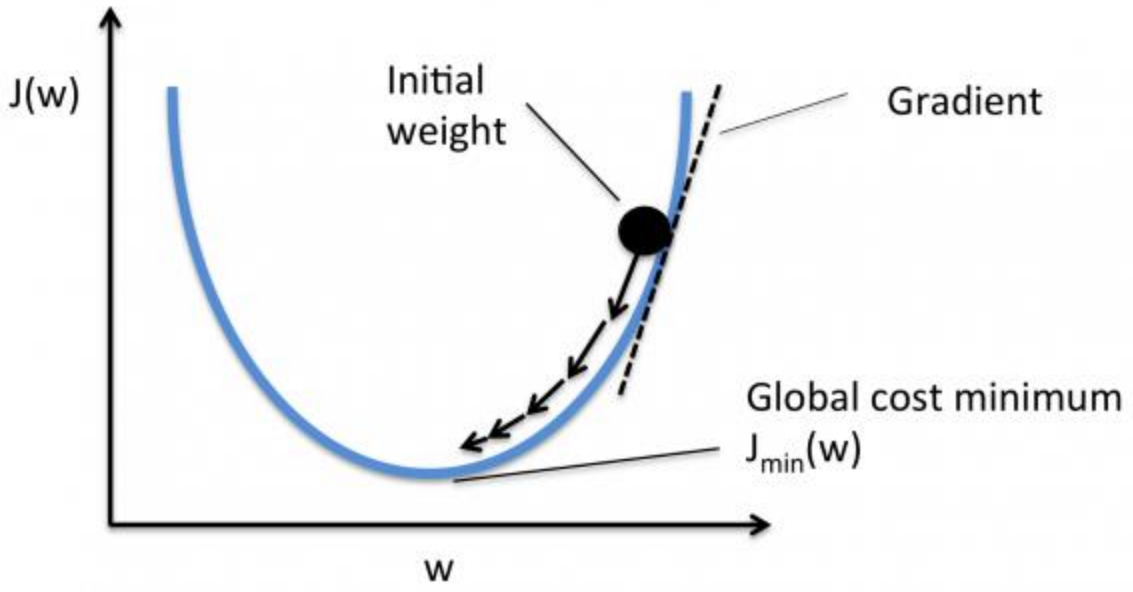
$$\frac{\partial}{\partial\theta_j}\mathrm{MSE}(\theta) = \frac{2}{m}\sum_{i=1}^{m}\left(\theta^T\cdot\mathbf{x}^{(i)} - y^{(i)}\right)x_j^{(i)}$$

ГРАДИЕНТНЫЙ СПУСК

Коэффициент скорости обучения

Если этот размера шага слишком велик, мы преодолеем минимум: то есть промахнемся мимо минимума. Если шаг слишком мал, мы используем слишком много итераций, чтобы добраться до минимума.

Итак, шаг должен быть подходящим.



Метод градиентного спуска

Пусть функция $f(x)$ такова, что можно вычислить ее градиент.

Необходимо найти $f(x) \to \min\limits_{x \in \mathbb{R}^n}$

Тогда можно применить метод градиентного спуска:

•Метод градиентного спуска с постоянным шагом
•Метод градиентного спуска с дроблением шага
•Метод наискорейшего спуска

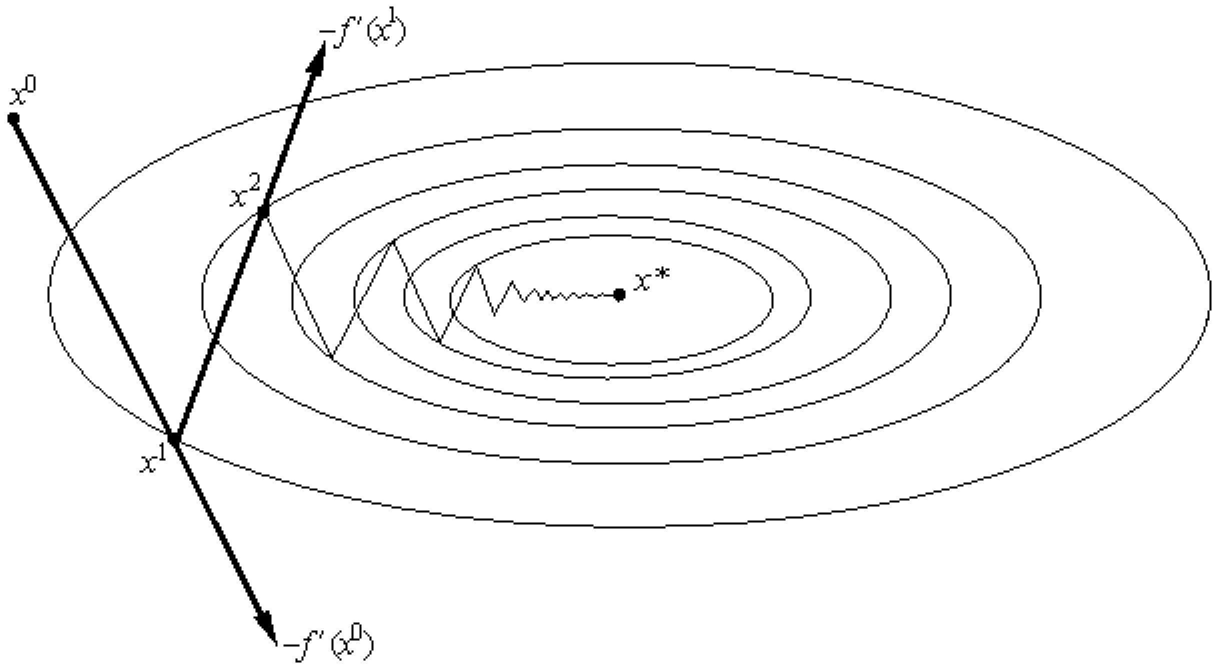
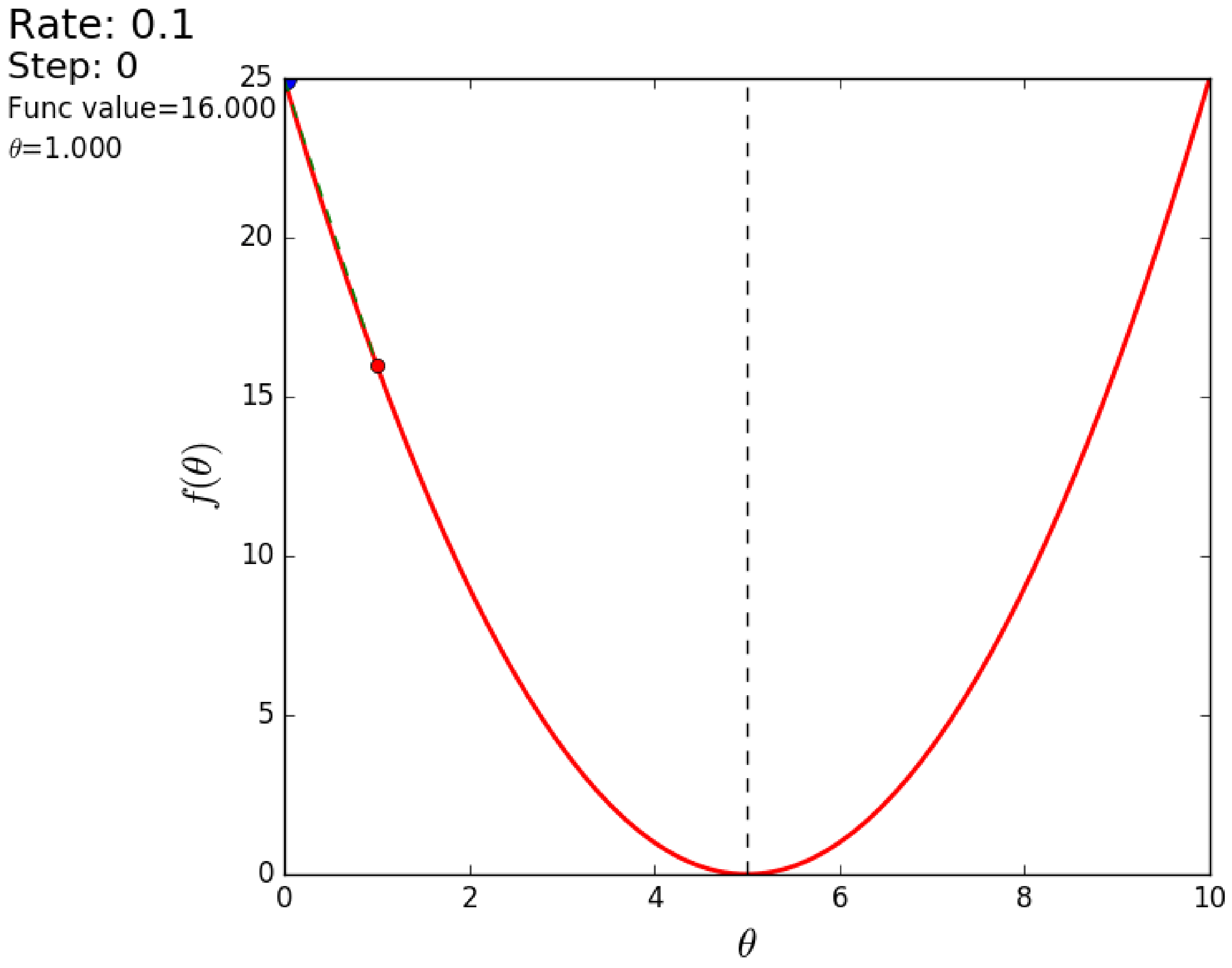


Рисунок – Иллюстрация последовательных приближений к точке экстремума в направлении наискорейшего спуска в случае дробного шага

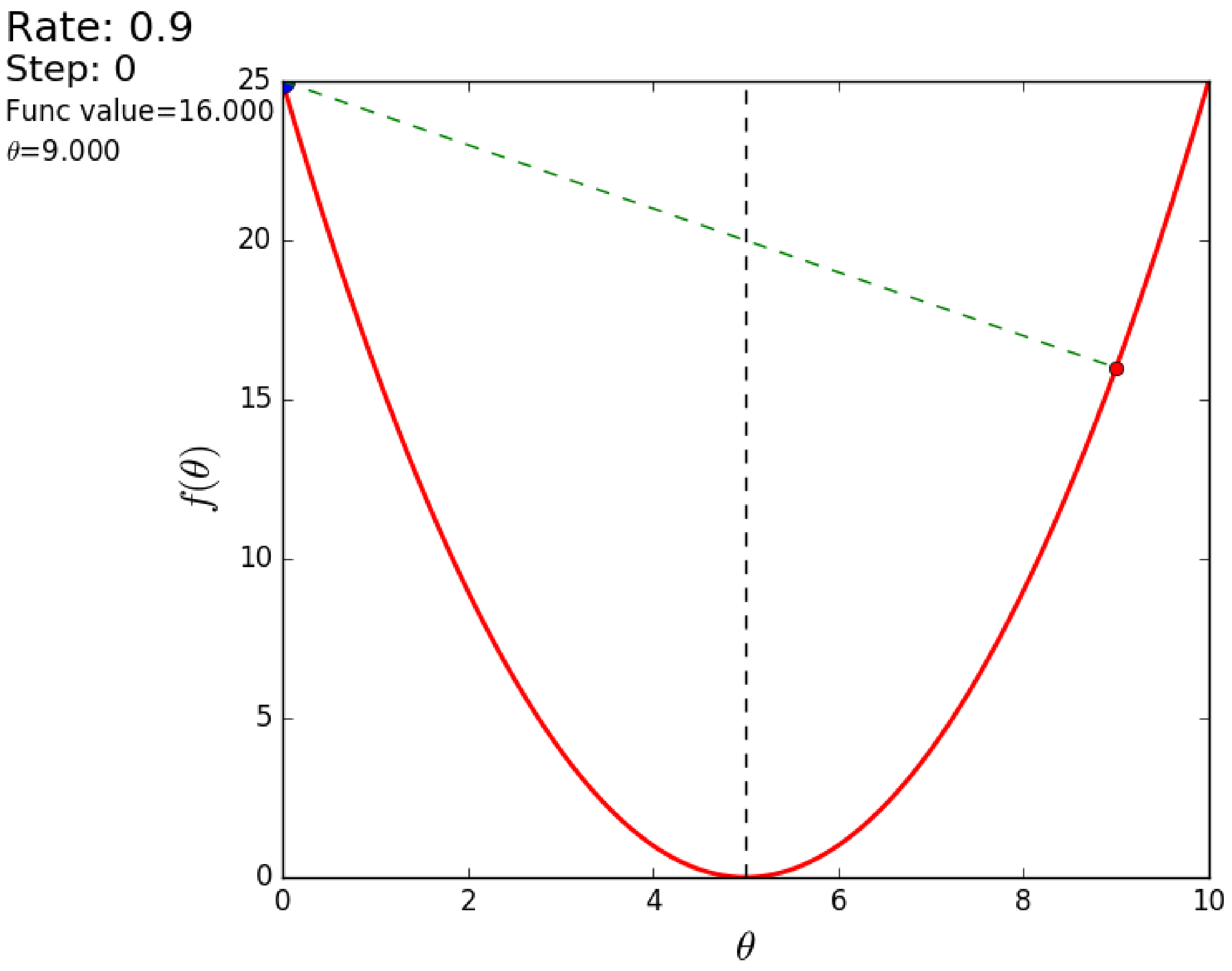
ГРАДИЕНТНЫЙ СПУСК

$$y = (\theta - 5)^2$$

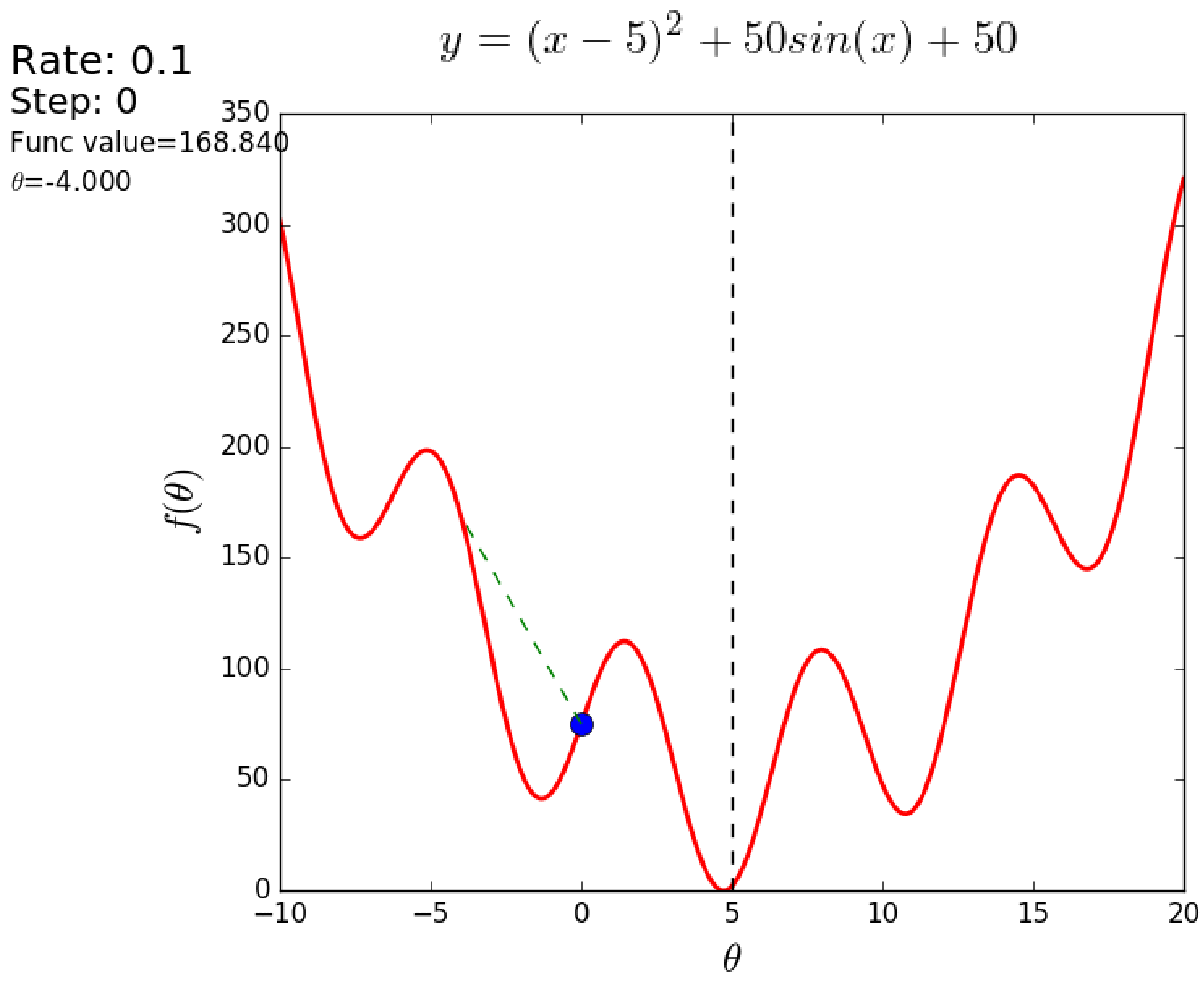


ГРАДИЕНТНЫЙ СПУСК

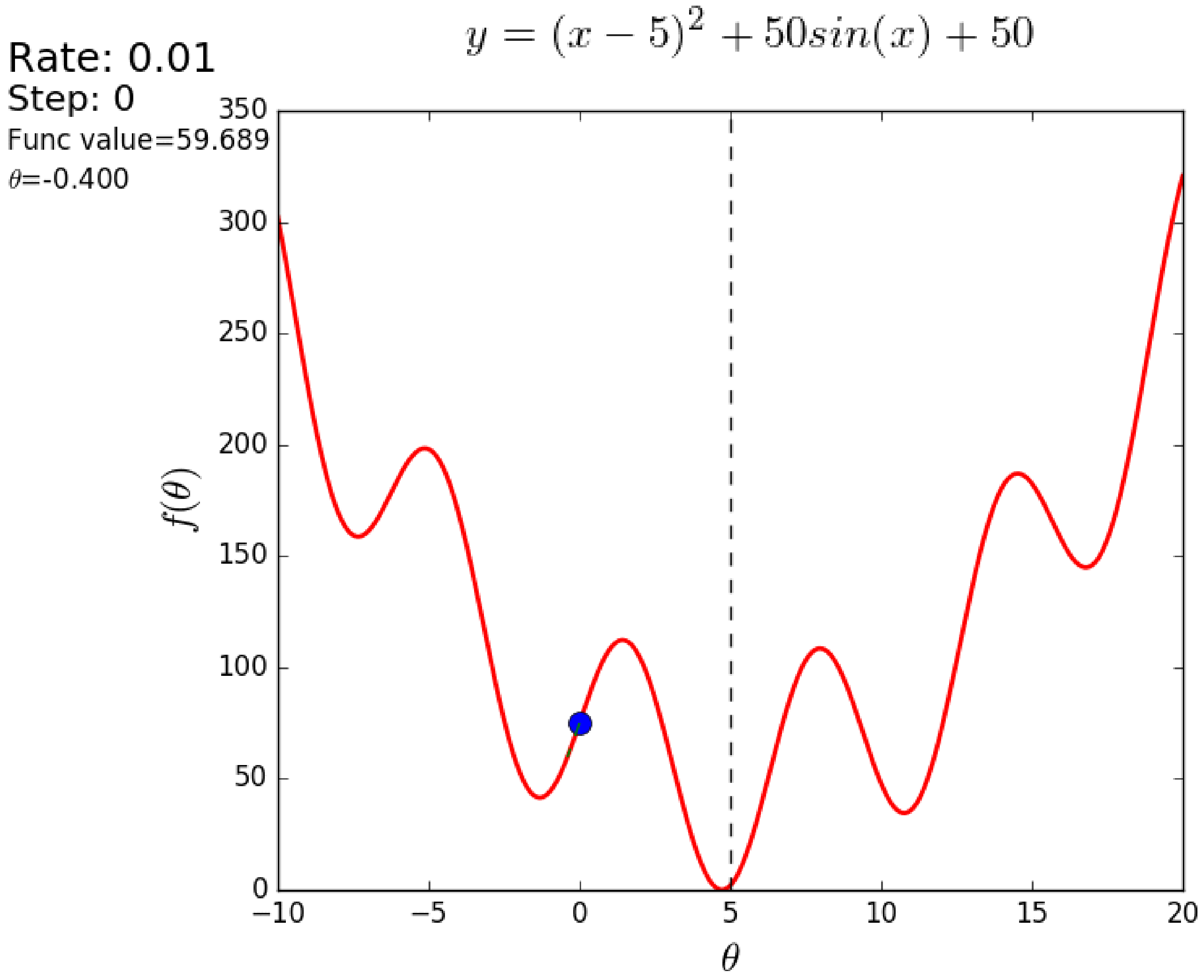
$$y = (\theta - 5)^2$$



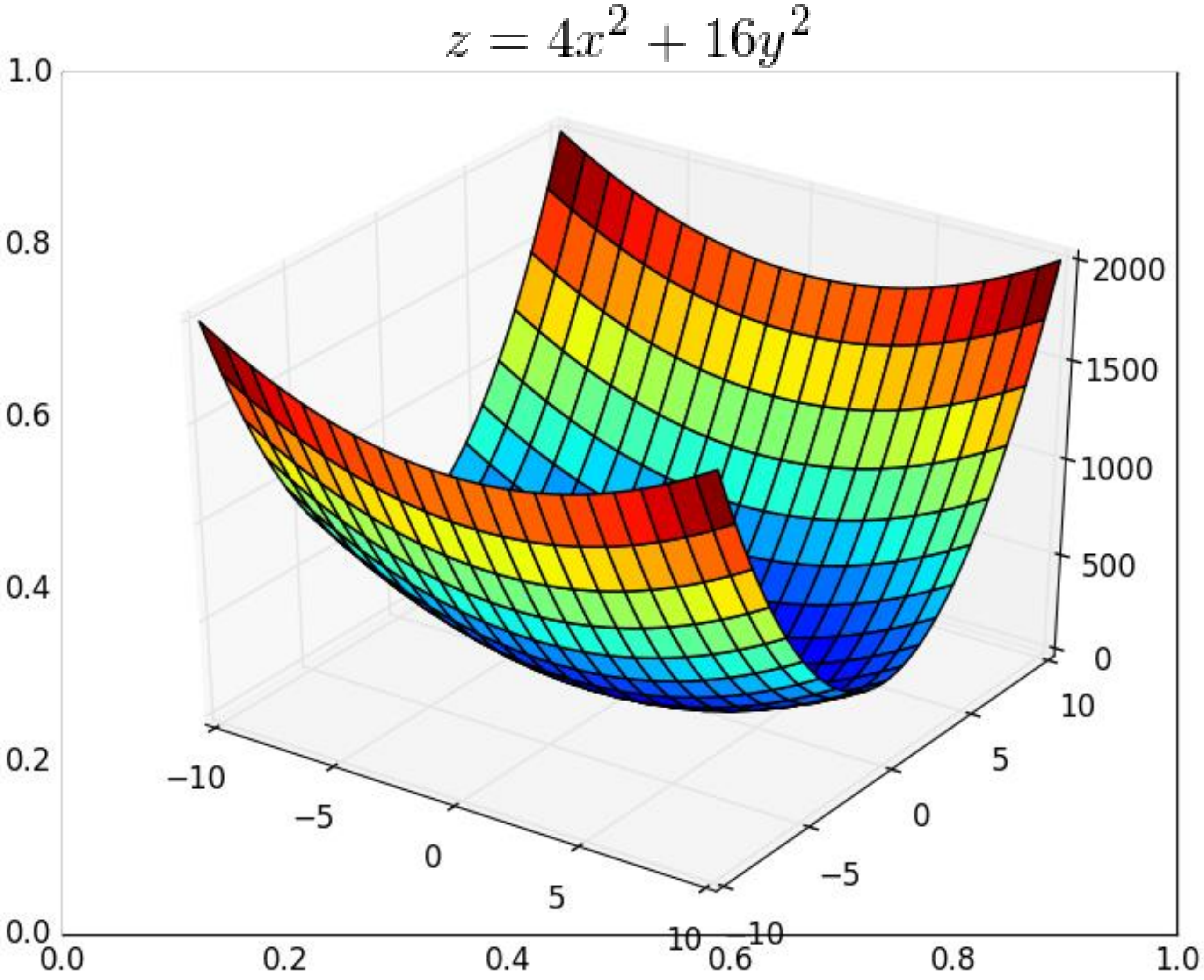
ГРАДИЕНТНЫЙ СПУСК



ГРАДИЕНТНЫЙ СПУСК

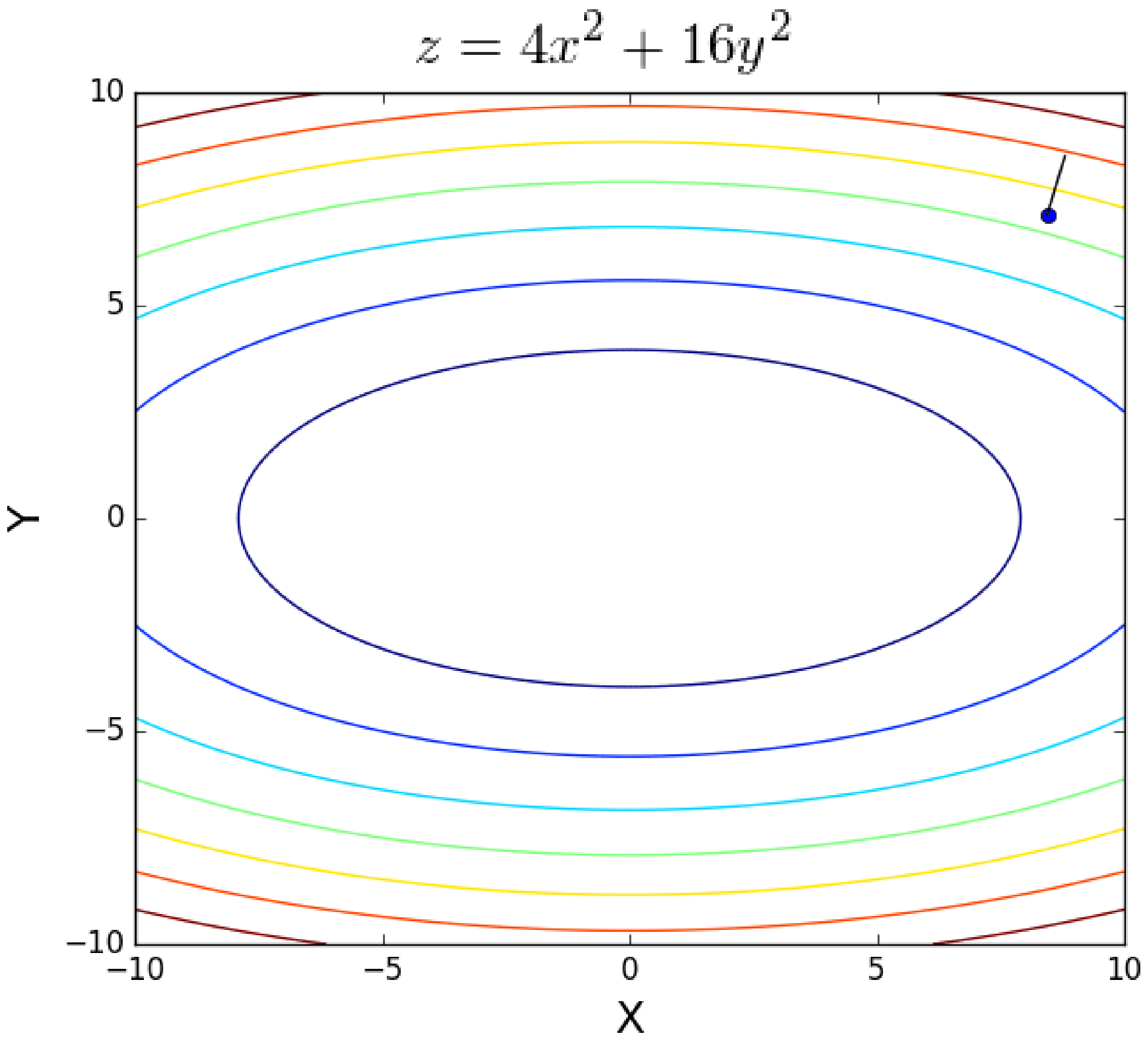


ГРАДИЕНТНЫЙ СПУСК



ГРАДИЕНТНЫЙ СПУСК

IT-Academy



Типы градиентного спуска:

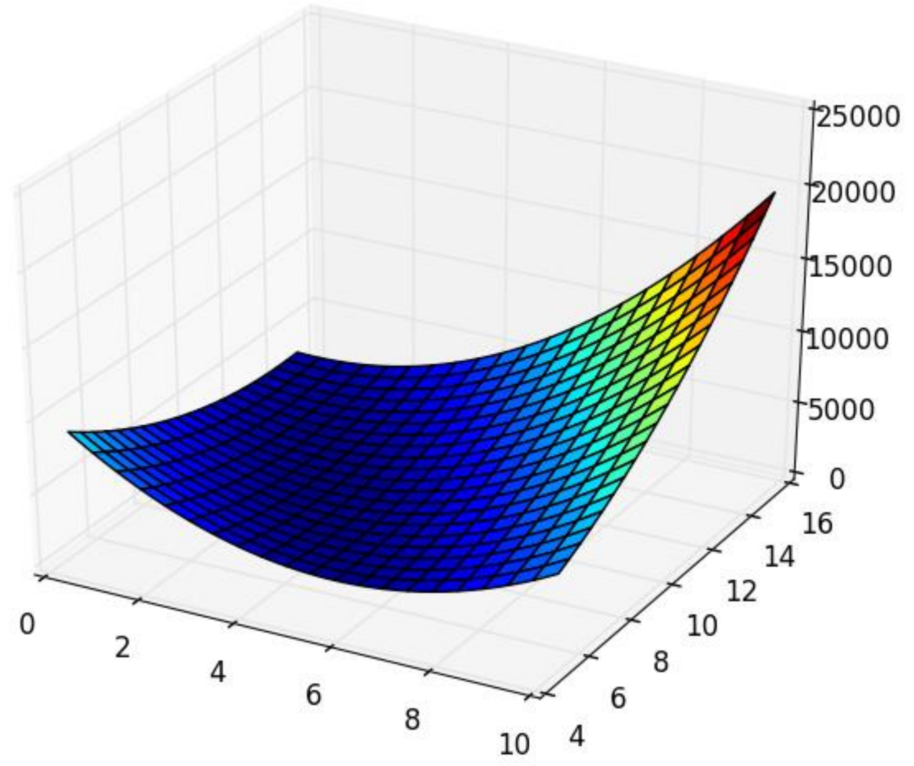
IT-Academy

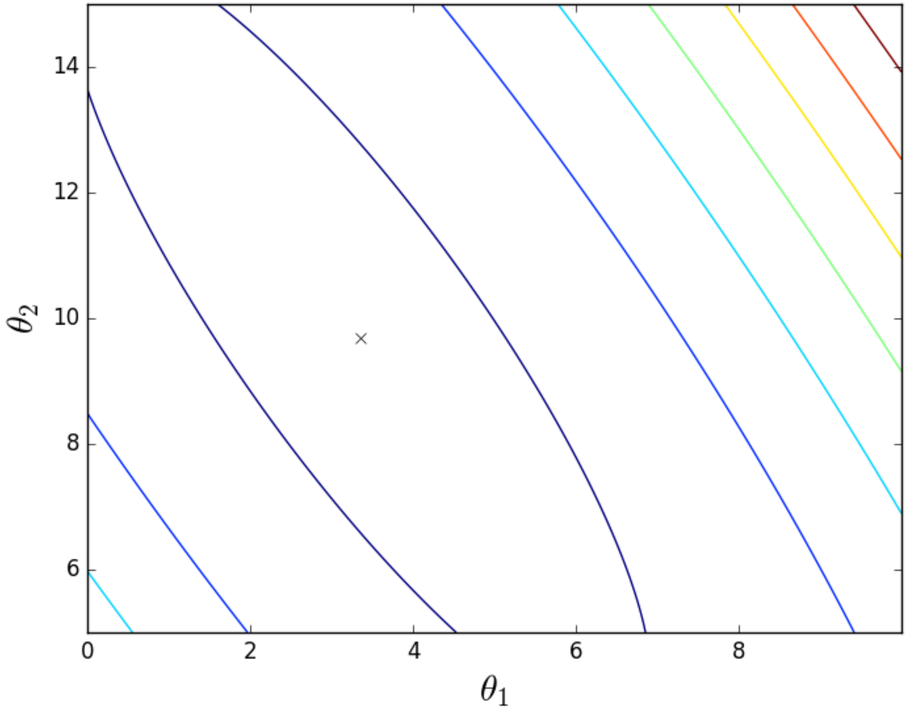
Пакетный градиентный спуск: это тип градиентного спуска, который обрабатывает все обучающие примеры для каждой итерации градиентного спуска.

Стохастический градиентный спуск: это тип градиентного спуска, который обрабатывает 1 обучающий пример за итерацию.

Мини-градиентный спуск: это тип градиентного спуска, который работает быстрее, чем как пакетный градиентный спуск, так и стохастический градиентный спуск. Здесь b примеры, где $b<m$ обрабатываются за одну итерацию. Таким образом, даже если количество обучающих примеров велико, оно обрабатывается партиями из b обучающих примеров за один раз.

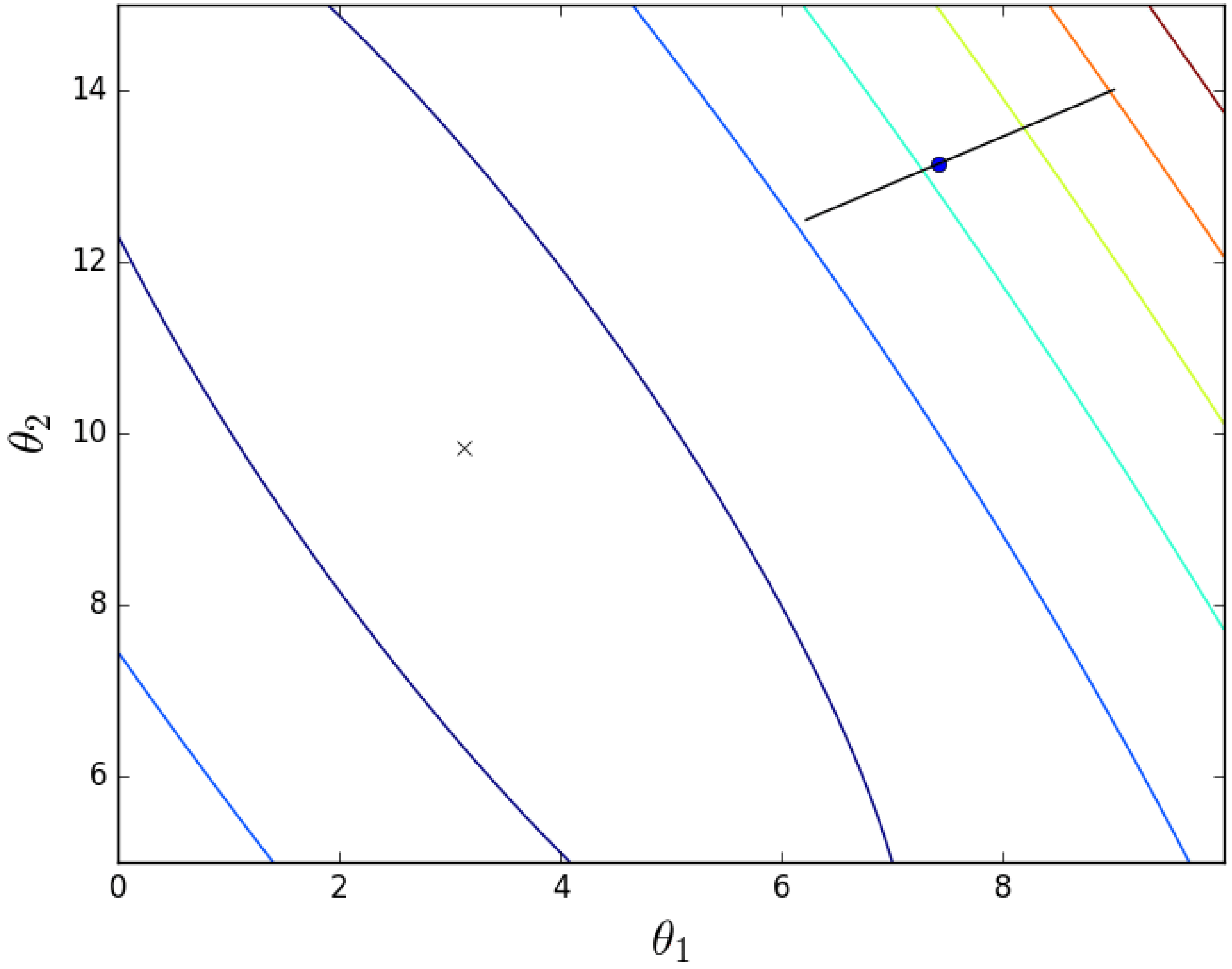
ГРАДИЕНТНЫЙ СПУСК





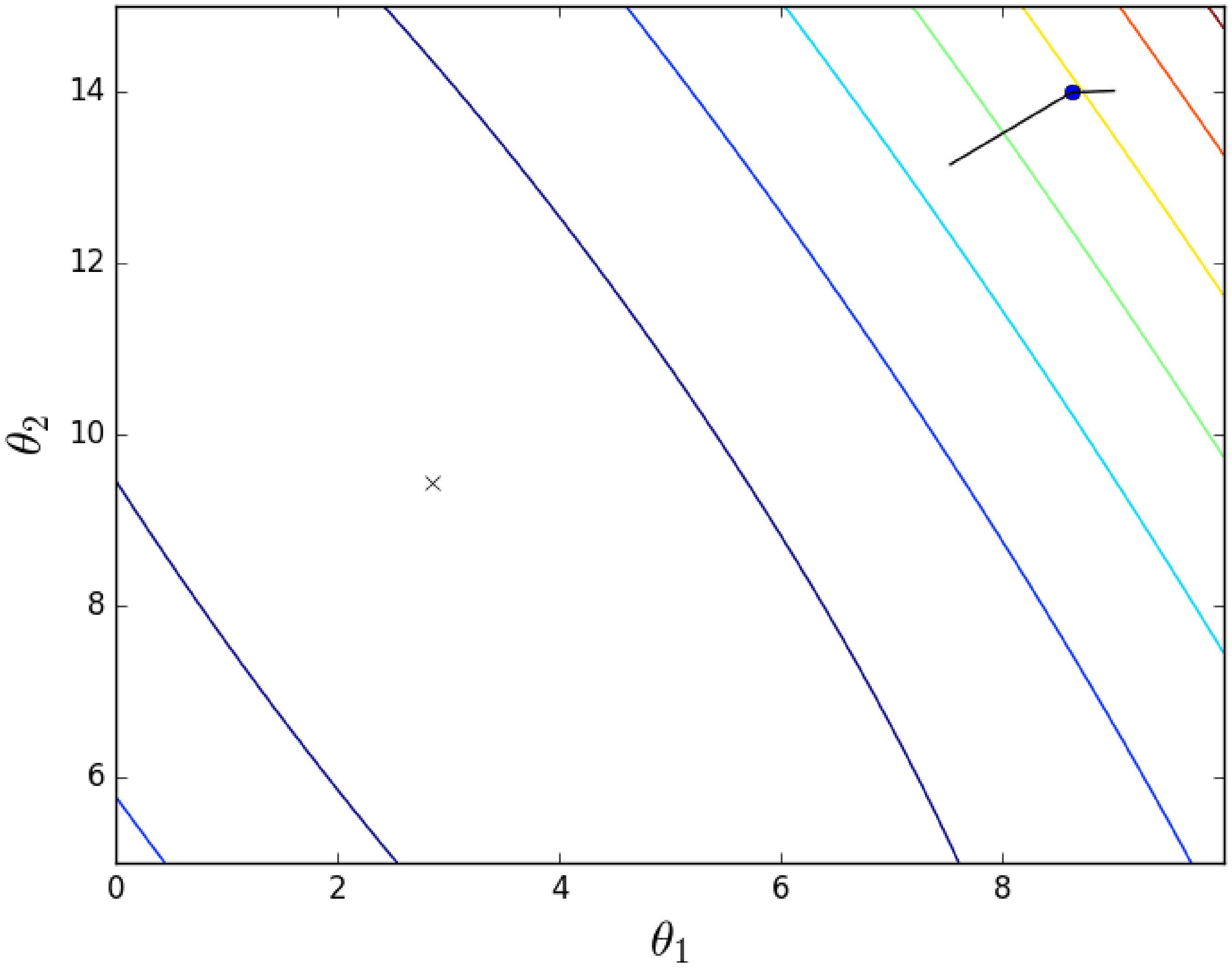
ГРАДИЕНТНЫЙ СПУСК

Пакетный спуск



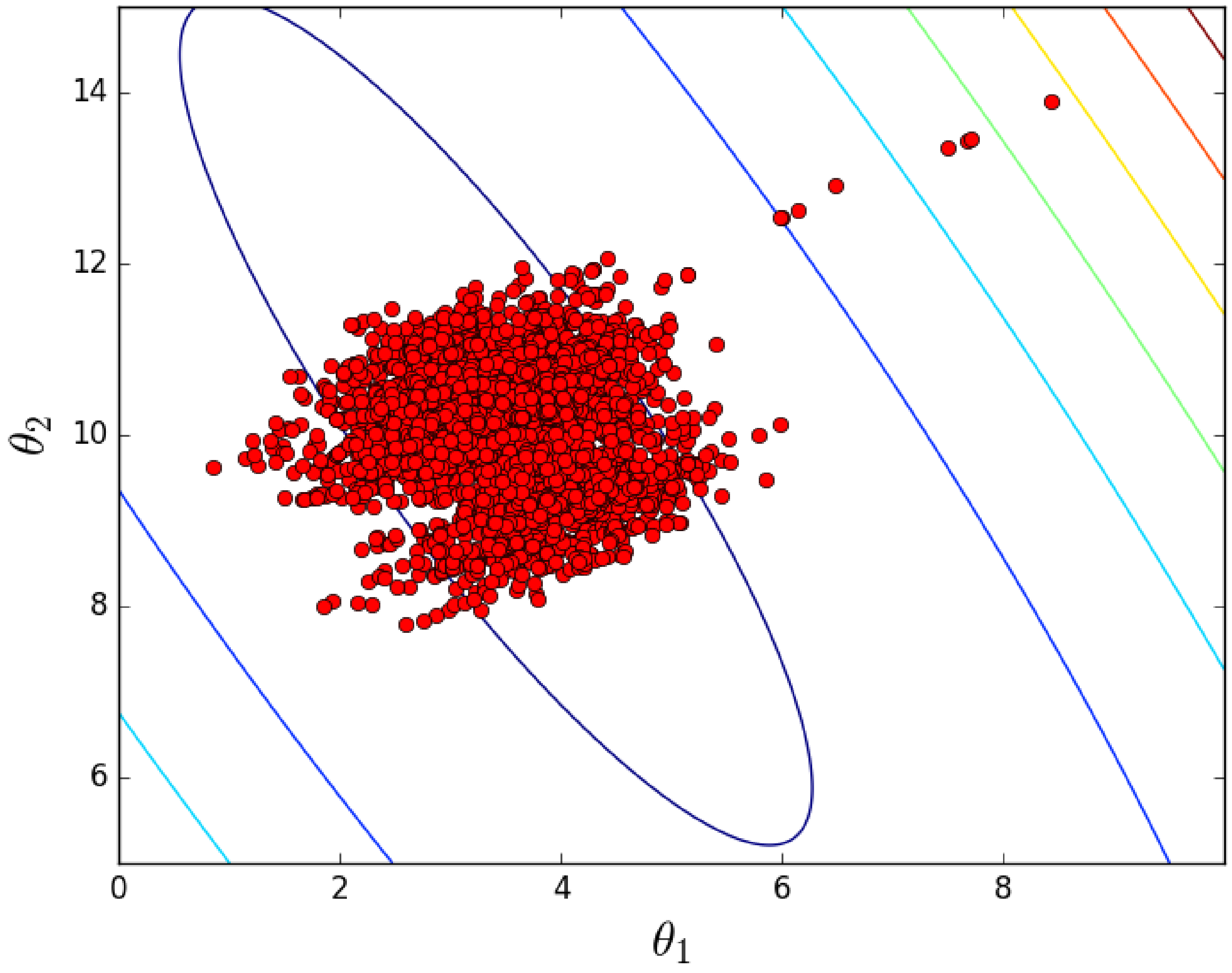
ГРАДИЕНТНЫЙ СПУСК

Стохастический спуск



ГРАДИЕНТНЫЙ СПУСК

Стохастический спуск



ПРАКТИКА

Pandas для больших данных

Pandas dtype	Python type	NumPy type	Usage
object	str	string_, unicode_	Text
int64	int	int_, int8, int16, int32, int64, uint8, uint16, uint32, uint64	Integer numbers
float64	float	float_, float16, float32, float64	Floating point numbers
bool	bool	bool_	True/False values
datetime64	NA	datetime64[ns]	Date and time values
timedelta[ns]	NA	NA	Differences between two datetimes
category	NA	NA	Finite list of text values

Градиентный спуск

КУХАЛЬСКИЙ НИКОЛАЙ ГЕННАДЬЕВИЧ