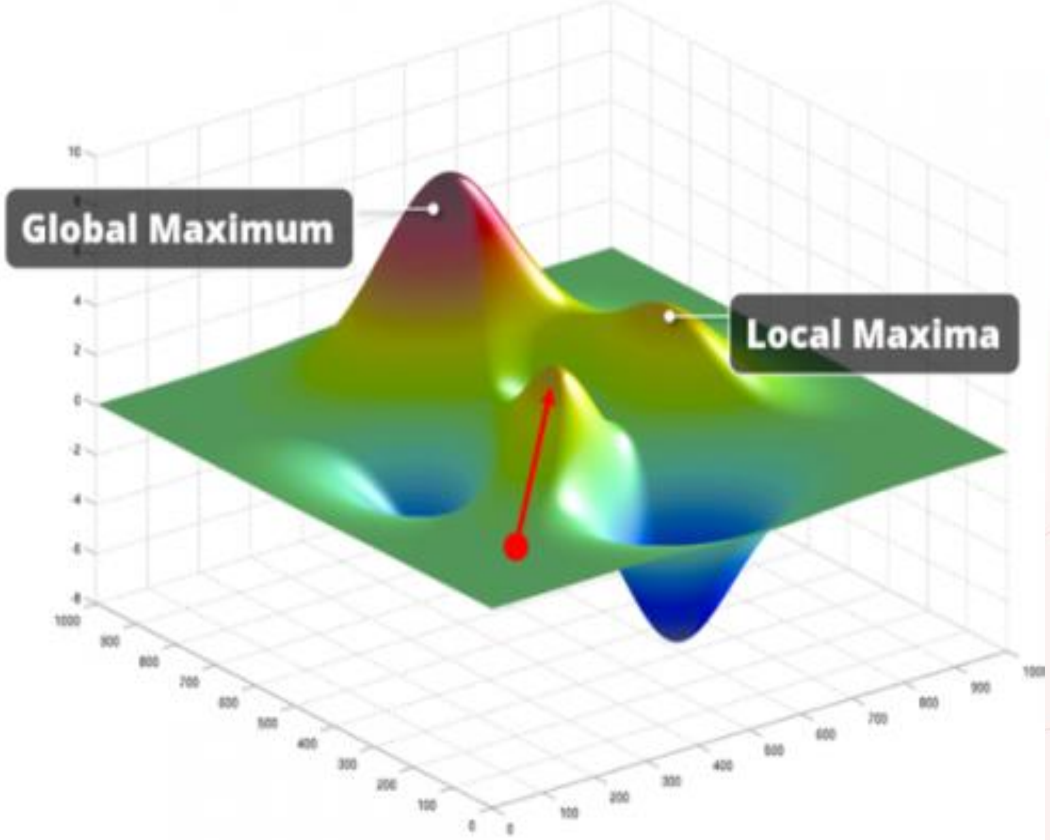
Основные понятия и определения градиентного спуска

1919
ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

Градиентный спуск в машинном обучении

Оптимизация - большая часть машинного обучения. В основе каждого алгоритма машинного обучения лежит алгоритм оптимизации.
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. Без градиентного спуска машинное обучение не было бы там, где сейчас.
Метод градиентного спуска с некоторой модификацией широко используется для обучения персептрона и глубоких нейронных сетей, используется и в методе градиентного бустинга – самом популярном метод решения задач классификации и регрессии для структурированных (табличных) данных



Частные производные

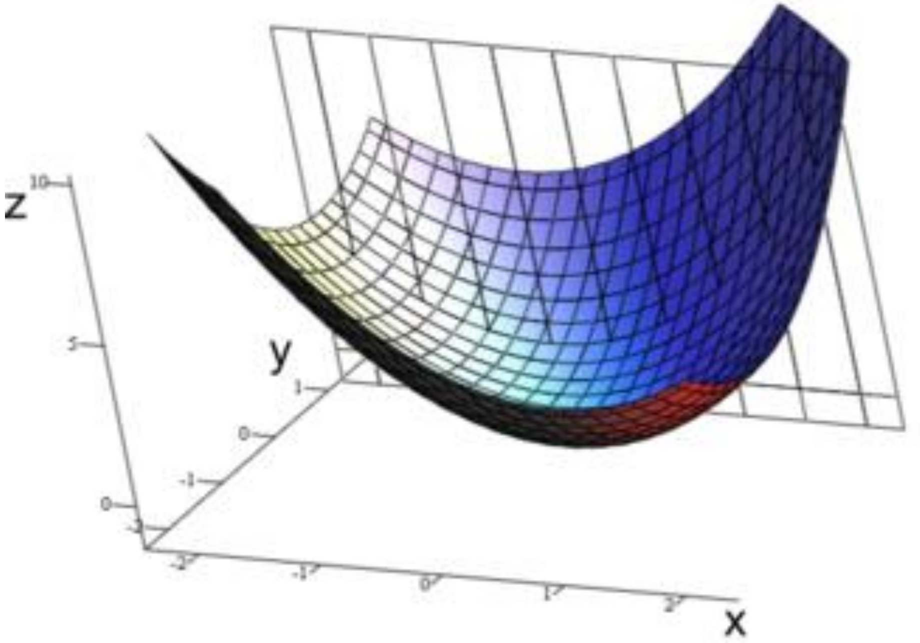
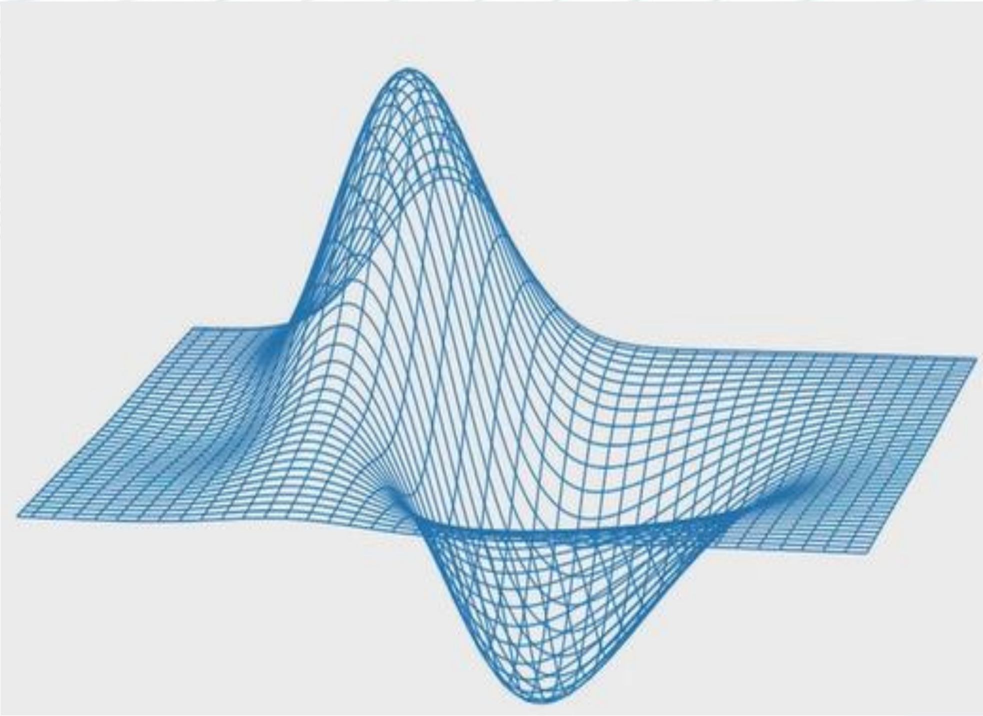


График функции z = x² + x·y + y²
Частная производная в точке (1, 1, 3) при постоянном y соответствует углу наклона касательной прямой, параллельной плоскости x·y.



$$f(x,y) = e^{x^2+\sqrt{y}}$$

$$\frac{\partial f}{\partial x} = e^{x^2+\sqrt{y}}.(2x)$$

$$\frac{\partial f}{\partial y} = e^{x^2+\sqrt{y}}\left(\frac{1}{2\sqrt{y}}\right)$$

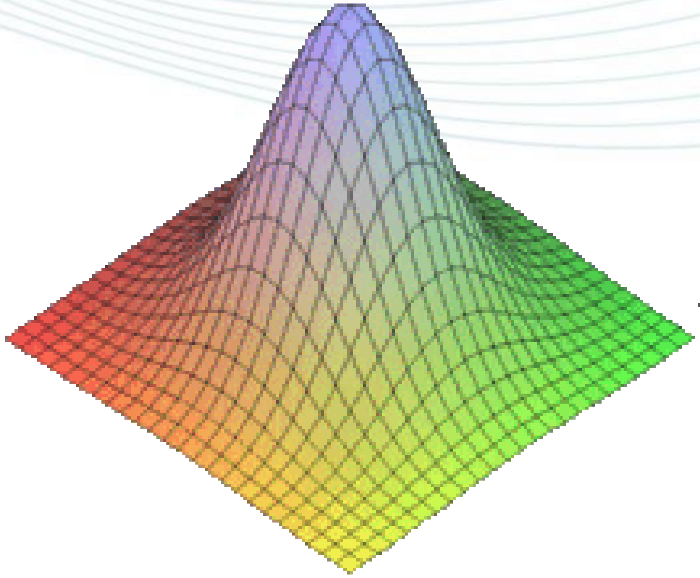
Градиент

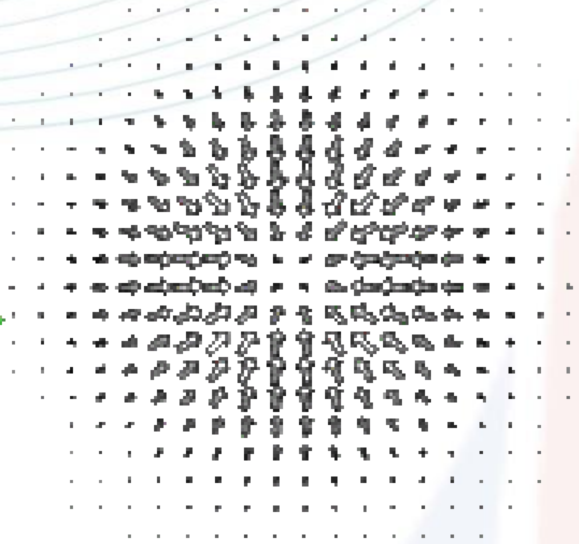
Графическое сопровождение

Градиент — **вектор**, своим направлением указывающий ***направление*** наибольшего возрастания некоторой величины φ, значение которой меняется от одной точки пространства к другой (скалярного поля), а по величине (модулю) равный ***скорости роста*** этой величины в этом направлении.
Другими словами, градиент — это производная по пространству, но в отличие от производной по одномерному времени, градиент является не скаляром, а векторной величиной.
Например, для функции

$$\varphi(x,\ y, z\ ) = 2x + 3y^2 - sinz.$$

$$\nabla\varphi = \left(\frac{\partial\varphi}{\partial x},\ \frac{\partial\varphi}{\partial y},\ \frac{\partial\varphi}{\partial z}\right) = (2,\ 6y,\ -\cos z)$$



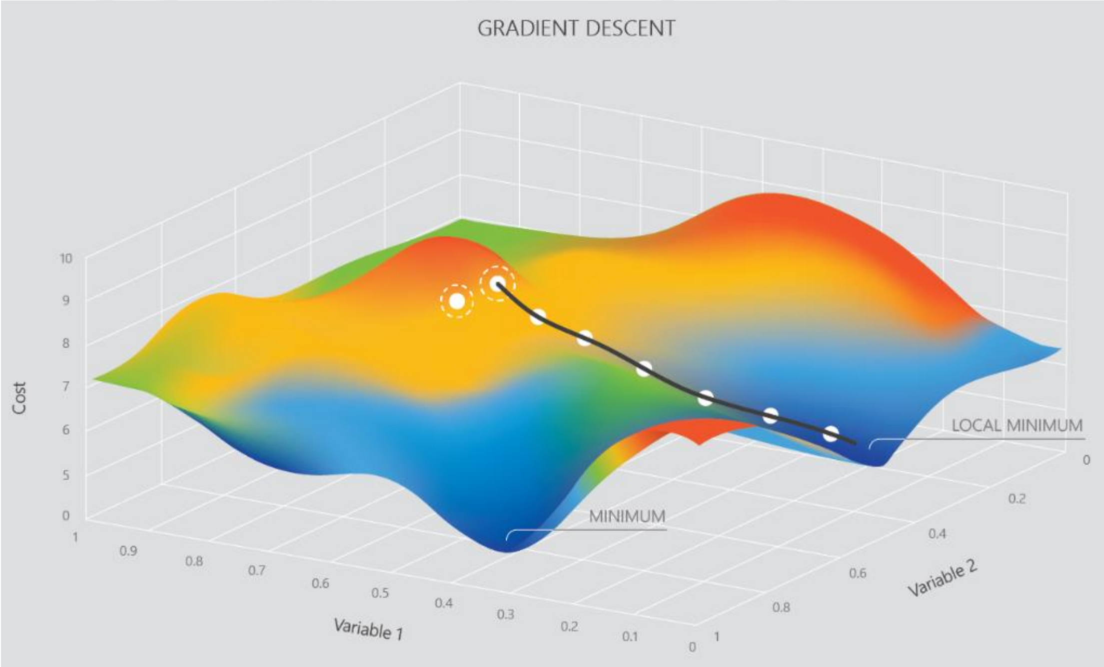


Операция градиента преобразует холм (слева), если смотреть на него сверху, в поле векторов (справа). Видно, что векторы направлены «в горку» и чем длиннее, тем круче наклон

Что такое градиентный спуск

Графическое сопровождение

Градиентный спуск — метод нахождения минимального значения функции потерь (существует множество видов этой функции). Минимизация любой функции означает поиск самой глубокой впадины в этой функции. Функция потерь (стоимости) или **Loss** 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 для нейронной сети)



Обучение и функция потерь (Loss function)
Обучение модели означает определение хороших значений для всех весов и смещений из отмеченных примеров. Функция потерь является мерой расхождения между истинным значением оцениваемого параметра и оценкой параметра.
В контролируемом обучении (обучении с учителем) алгоритм машинного обучения строит модель, анализируя множество примеров и пытаясь найти модель, минимизирующую loss; этот процесс называется минимизацией эмпирического риска.
Loss - это штраф за плохой прогноз. То есть loss - это число, указывающее, насколько плохое предсказание модели было на одном примере. Если прогноз модели идеален, потери (ошибки) равны нулю; в противном случае loss больше нуля. Цель обучения модели - найти набор весов и смещений, которые имеют низкие потери в среднем по всем примерам.
Наиболее часто используемой является **квадратичная функция потерь** (**L_2**-норма)

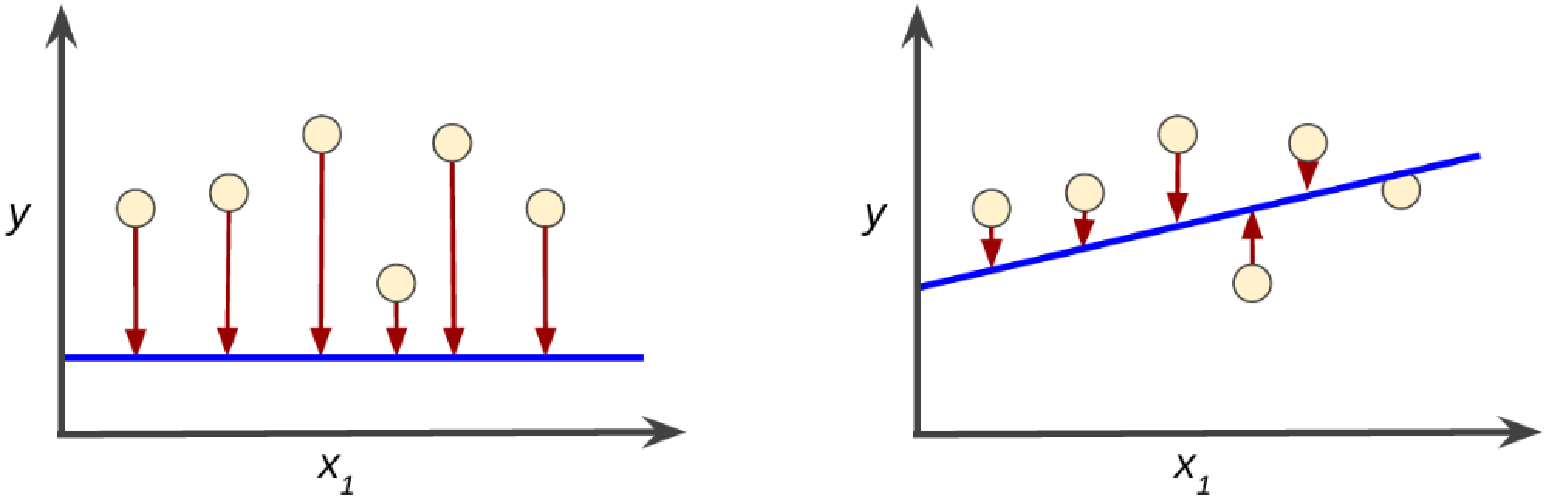
Обучение и функция потерь (Loss function)

Наиболее часто используемой является **квадратичная функция потерь** (**L_2**-норма)

$$= \text{квадрат разницы между меткой и предсказанием}$$
$$= (\text{observation} - \text{prediction}(x))^2$$
$$= (y - y')^2$$

Преимуществом квадратичной функции потерь являются инвариантность к знаку - значение функции всегда положительно. Т.е. независимо от знака ошибки результат будет один и тот же. Квадратичная функция потерь используется в моделях, параметры которых оцениваются на основе метода наименьших квадратов, например линейной регрессии.

Например, на рисунке показана модель с высокими loss слева и модель с низкими loss справа. Обратите внимание на следующее на рисунке:
Стрелки представляют loss. Синие линии представляют прогнозы.



Обратите внимание, что стрелки на левом графике намного длиннее, чем их аналоги на правом графике. Ясно, что линия на правом графике - гораздо лучшая прогностическая модель, чем линия на левом графике.
Вы можете задаться вопросом, можете ли вы создать математическую функцию - функцию потерь, которая бы содержательно объединяла отдельные потери

Среднеквадратичная ошибка (MSE) - это среднеквадратичная потеря на пример по всему набору данных. Чтобы вычислить MSE, суммируйте все возведенные в квадрат потери для отдельных примеров и затем разделите на число примеров:

$$MSE = \frac{1}{N} \sum_{(x,y)\in D} (y - prediction(x))^2$$

- где *(x,y)* это пример в котором:
  - *x* - набор функций (признаков, входов), который модель использует для прогнозирования
  - *y* - метка примера
- *prediction(x)* – предсказанное моделью значение
- *D* - набор данных, содержащий множество отмеченных примеров *(x,y)*
- *N* количество примеров в *D*

Квадратичную потерю (функцию стоимости) можно записать в следующем виде:

$$J\left(y, y^{/}\right) = C(y - y^{/})^2,$$

где де C - константа, y, - истинное значение выхода модели (которое должно быть получено в идеальном случае), $y^{/}$ - фактический выход модели.

Хотя MSE обычно используется в машинном обучении, это не единственная практическая функция потерь и не лучшая функция потерь при любых условиях.

В бинарной классификации используется двоичная функция потерь (0-1 *loss function*), которая определяется следующим образом:

$$J\left(y, y^{/}\right) = f(y \neq y^{/}).$$

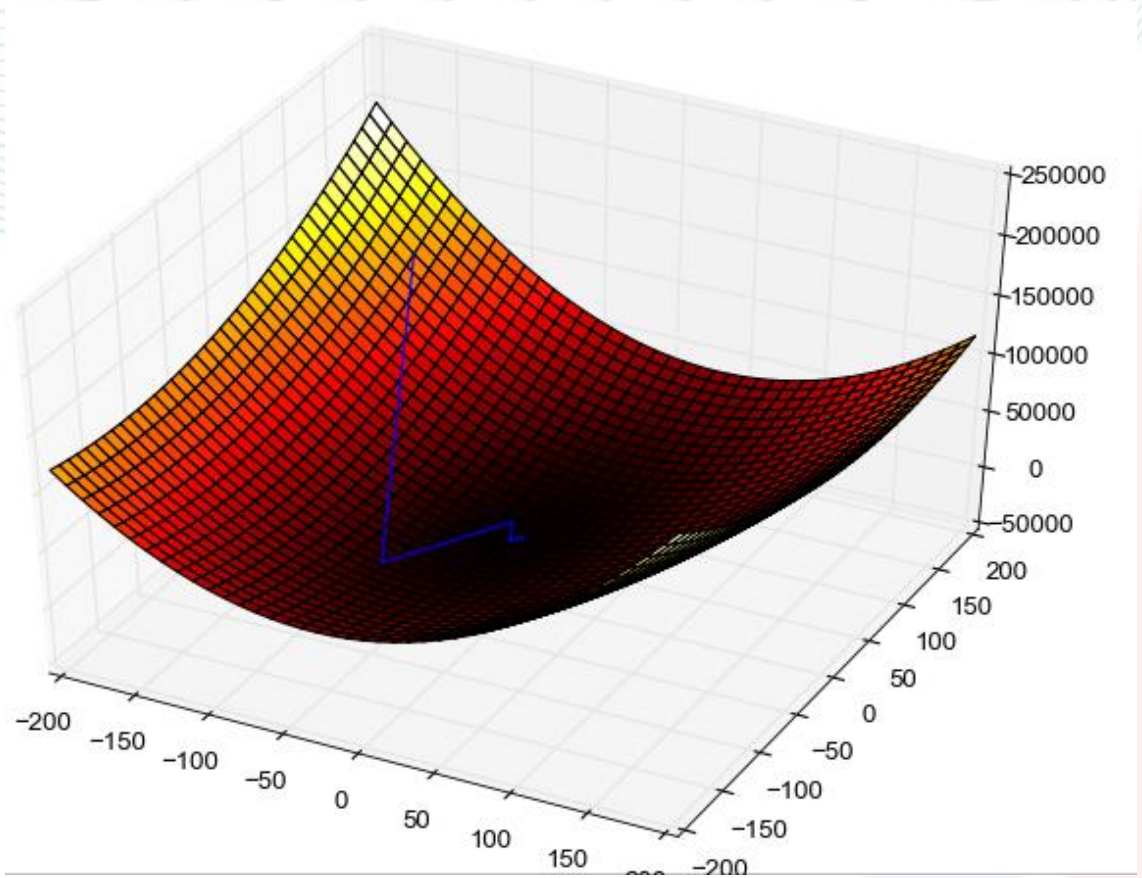
Как видно, потери определяются появлением двух взаимоисключающих состояний выхода модели.

Используется также и простая функция потерь, равная разности истинного и фактического выходов модели:

$$J\left(y, y^{/}\right) = f(y - y^{/}).$$

Она используется в тех случаях, где важен знак ошибки, например, при обучении нейронных сетей.

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. Вообще можно создавать модели даже без применения функции потерь. Но тогда придется использовать производную относительно каждого веса (dJ/dw).



Теперь, когда мы определили направление, в котором нужно подтолкнуть вес, нам нужно понять, как это сделать. Тут мы используем коэффициент скорости обучения, его называют гипер-параметром.

Гиперпараметры модели — параметры, значения которых задается до начала обучения модели и не изменяется в процессе обучения. У модели может не быть гиперпараметров.
Параметры модели — параметры, которые изменяются и оптимизируются в процессе обучения модели и итоговые значения этих параметров являются результатом обучения модели.
Примерами гиперпараметров могут служить количество слоев нейронной сети, а также количество нейронов на каждом слое.
Примерами параметров могут служить веса ребер нейронной сети.
Для нахождения оптимальных гиперпараметров модели могут применяться различные алгоритмы настройки гиперпараметров

Резюме

Вы узнали, что:

- Оптимизация - большая часть машинного обучения
- **Градиентный спуск** - это процедура оптимизации, которую вы можете использовать со многими алгоритмами машинного обучения