Оптимизация — это о чём? Ну, о том, что мы ищем минимум (или максимум) какой-то функции на каком-то множестве. Причём минимум ищется обычно локальный, потому что глобальный искать очень сложно.
Рассмотрим непрерывную оптимизацию: наша функция непрерывна. В таких методах применяется принцип чёрного ящика: мы не анализируем то, что делает функция, нас интересуют только её значения. Иногда нас также интересует градиент функции и/или гессиан. В зависимости от того, производная какого порядка нам нужно в нашем методе, говорят о том, что наш метод — это метод соответствующего порядка (если не использовать ни градиент, ни гессиан, то это метод нулевого порядка, если использовать только градиент — первого,...). Методы второго порядка — это, например, методы Ньютона. Также есть методы квази-Ньютона, которые пытаются как-то аппроксимировать гессиан, но сейчас не о них.
Зачем нам вообще градиент и гессиан? Потому что они позволяют разложить функцию в ряд Тейлора, и выглядеть он будет так:

$$f(x+\Delta x) = f(x) + \Delta x^T \nabla f(x) + \frac{1}{2}\Delta x^T H_f(x)\Delta x + o(\|\Delta x\|^2)$$

Более высокие порядки не используются обычно, потому что там возникают такие операции как взятие обратной матрицы, которые вводят численную нестабильность.
Как это глобально работает: нам дают функцию, градиент (и гессиан, если для метода надо), и мы уже дальше интерпретируем функцию, градиент и гессиан как чёрные ящики.

Пример. Тернарный поиск. Все мы его знаем. Это метод нулевого порядка. Берём функцию, которая сначала убывает, потом возрастает и итеративно делим отрезок на три части, на каждом этапе отбрасывая одну треть.
Можно немного упростить жизнь себе, использовав метод золотого сечения. А именно отрезок $[a;b]$ мы делим на три части точками $x_1 = b - \frac{b-a}{\Phi}$, $x_2 = a + \frac{b-a}{\Phi}$, где $\Phi = \frac{1+\sqrt{5}}{2}$. Остальное как в обычном тернарном поиске. При таком разделении значение в одной из двух точек будет переиспользовано, ведь x_1 делит $[a;x_2]$ в соотношении золотого сечения и x_2 делит $[x_1;b]$ в соотношении золотого сечения.

Нулевого порядка в целом больше ничего не придумать.
Для методов первого порядка обычно итеративно выбирают последовательность точек, значения в которых должны уменьшаться. Как выбирать — рассмотрим позже. А сейчас подумаем, когда останавливаться.

Определение 1. Пусть $\{x_k\}$ — последовательность точек, которые выбирает метод, а x^* — локальный минимум, к которому метод стремится. Тогда если выполнено

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^\mu} < r$$

то r называют **скоростью сходимости**, а μ — **степенью сходимости**.

Определение 2. Контурными линиями называются множества

$$L_f(a) = \{x \mid f(x) = a\}$$

Утверждение. *Контурные линии ортогональны градиенту функции.*

Условия оптимальности. Хочется найти какие-нибудь критерии оптимальности точки в какой-то окрестности. У нас могут быть достаточные условия, могу быть необходимые, и все они следуют из квадратичной аппроксимации функции.
Ну, точка оптимальна, если при добавлении произвольного Δx функция возрастёт. Заметим, что $\nabla f(x)$ и $H_f(x)$ никак не зависят от x. Для методов первого порядка есть только необходимое условие:

Теорема 1. *Если точка оптимальна, то* $\nabla f(x) = 0$.

Доказательство. Ну, если он не ноль, то можно найти достаточно маленькое Δx, противонаправленное $\nabla f(x)$, и функция уменьшится. А значит точка не оптимальна была. □

Самое интересное, что для хороших функций можно решить уравнение $\nabla f(x) = 0$ и получить аналитическое решение. Например, так отлично решается линейная регрессия.
А вот если у нас есть гессиан, то мы и достаточное условие можем сформулировать. Это условие доказывается через спектральное разложение гессиана.

Теорема 2. *Если все собственные числа гессиана в точке положительные, то эта точка — минимум.*
Если точка — минимум, то все собственные числа гессиана неотрицательны.

Доказательство.

$$\begin{aligned} f(x+\Delta x) &\approx f(x) + \frac{1}{2}\Delta x^T H_f(x)\Delta x = \\ &= f(x) + \frac{1}{2}\Delta x^T Q \wedge Q^T \Delta x = \\ &= f(x) + \frac{1}{2}(Q^T\Delta x)^T \wedge Q^T\Delta x = \\ &= f(x) + \frac{1}{2}\sum_i \lambda_i \|(Q^T\Delta x)_i\| \end{aligned}$$

□

Линейные ограничения. Давайте добавим какие-нибудь ограничения. Например, линейные ограничения: минимизировать функцию f при условии $Ax = b$. Тогда как выглядит наша окрестность x? Ну, чтобы получить окрестность в подпространстве $Ax = b$, можно взять большую окрестность и взять оттуда только подходящие точки. То есть у нас окрестность их точек $x + \Delta x$ и мы берём только такие $x + \Delta x$, у которых $A(x + \Delta x) = b$. Тогда $A\Delta x = 0$. Соотвественно, проверяя условия необходимости и/или достаточности, нам надо проверять только те направления, для которых верно $A\Delta x = 0$. Как с этим работать? Ну, довольно легко: надо рассмотреть базис $\ker A$. Пусть $\Delta x \in \ker A$. И возьмём N — матрицу столбцов базиса $\ker A$. Тогда мы можем рассматривать $N\Delta x$ как отклонение от нашего x. Тогда наш $N\Delta x$ всегда будет правильным. Что будет с нашим рядом Тейлора?

$$f(x + N\Delta x) = f(x) + \Delta x^T N^T \nabla f(x) + \frac{1}{2}\Delta x^T N^T H_f(x) N\Delta x + o(\|\Delta x\|^2)$$

В итоге можно только домножать градиент и гессиан на N (с нужных сторон с нужным транспонированием), и необходимое и достаточное условия останутся.

1 Непрерывная оптимизация.

Пока без доказательств, потому что нужны практики.

Метод множителей Лагранжа. Минимизировать $f(x)$, при условии $g(x) = 0$. Чтобы это решить, введём функцию Лагранжа:

$$L(x, \lambda) = f(x) - \lambda g(x)$$

λ называется **множителем Лагранжа**.
Оказывается, что для поиска минимума $f(x)$, при условии $g(x) = 0$ можно рассмотреть градиент ∇L и приравнять нулю. Этот градиент равен нулю тогда и только тогда, когда $f'(x) = \lambda g'(x)$ и $g(x) = 0$.
Если ограничений у нас несколько, надо ввести несколько множителей, по одному на каждое ограничение.

Метод градиентного спуска. Обычный итеративный метод, в котором мы берём точки так: $\Delta x_{k+1} = -\alpha\nabla f(x_k)$. Основано на простом факте: функция быстрее убывает в направлении антиградиента.

При этом в общем случае градиентный спуск сходится как-нибудь рандомно, возможно, никак. Поэтому мы хотим узнавать какие-нибудь свойства функций. Одно из них: липшицевость градиента:

$$\exists M \ \|\nabla f(x) - \nabla f(y)\| \leqslant M\|x - y\|$$

Другое: строгая выпуклость с параметром m:

$$(\nabla f(y) - \nabla f(x))^T (y - x) \geqslant \|y - x\|^2$$

Так вот, если функция удовлетворяет обоим условиям, то наш градиентный спуск линейно сходится:

$$\|x_{k+1} - x^*\| \leqslant \left(1 - \frac{2\alpha m M}{M + m}\right) \|x_0 - x^*\|$$

Оптимальное α получается равным $\frac{2}{M+m}$.

Стохастический градиентный спуск. Этот чёрт используется для обучения нейронок. Пусть у нас функция представима как сумма каких-то более простых функций. Тогда, очевидно, градиент представим в виде суммы градиентов компонентов. И стохастический градиентный спуск заключается в том, что мы считаем не весь градиент сразу, а только у компонент (по очереди, в каком-нибудь рандомном порядке). Или у нескольких компонент сразу (эти несколько компонент называются batch). Экспериментально выяснено, что так будет лучше (с точки зрения количества вычислений). Как это практически работает? Ну, смотрите. В нейронках у нас есть какие-то образцы и обычный градиентный спуск должен взять ошибку для каждого из образцов и сложить. И исходя из градиента этого нечто сдвинуться. А поскольку образцов очень много, градиент получается весьма сложный. В чём тут заключается стохастический градиентный спуск? В том, чтобы считать за один шаг ошибку по одному образцу. Или по нескольким пачкой (если с batch'ами).

Метод Ньютона. Рассматриваем квадратичную аппроксимацию:

$$f(x + \Delta x) \approx f(x) + \Delta x^T \nabla f(x) + \frac{1}{2} \Delta x^T H_f(x) \Delta x$$

Возьмём квадратичную аппроксимацию, посмотрим на производную по Δx, получив $\nabla f(x) + H_f(x)t$. Соотвественно, берём

$$x_{k+1} = x_k - H_f^{-1}(x_k) \nabla f(x_k)$$

Очень весело, надо инвертировать гессиан. С этим можно просто жить и страдать (потому что любой способ инвертирования матриц численно нестабилен), а можно пользоваться методами квази-Ньютона, но о них позже.
Для выпуклых функций всё круто, оно сходится к минимуму. Иначе оно может искать седловые точки вместо минимума. Если мы этого хотим, то хорошо. Если нет, увы и ах.
Важный прикол: тут можно обойтись без α, потому что у квадратичной функции есть минимум, (а в методе Ньютона функция линейная и есть только направление). Но в целом, можно добавить, хуже не будет.

Симплекс-метод. Задача в том, стобы найти минимум линейной функции в выпуклом многограннике. То есть мы хотим минимизировать $Ax + b$ при условии $Cx \leqslant d$.
В симплекс-методе задачу сначала приводят к стандартной форме (**максимизировать** $A_2 x + b_1$ при условии $C_2 x = d_2$ и $x \geqslant 0$). Как это сделать? Сначала как добавить условие на $x \geqslant 0$? Если есть нижняя граница на x, смещаем переменную так, чтобы эта граница стала нулём. Если есть только верхняя граница, меняем x на $-x + b$. Если ограничений нет, заменим x на x^+ и x^-, которые обе неотрицательны, а $x = x^+ - x^-$. Чтобы преобразовать неравенства в равенства, мы вводим новые неотрицательные переменные в каждое уравнение, преобразуя $x_1 - 3x_2 \leqslant 7$ в $x_1 - 3x_2 + y_1 = 7$.

На первом этапе переносим нововведённые переменные вправо, в константу — влево. После этого получается что-то типа:

$$\begin{aligned} y_1 &= 3 - x_1 - x_2 \\ y_2 &= 1 + x_1 - 3x_2 \\ y_3 &= 3 - x_2 \\ z &= 0 + x_1 + x_2 \end{aligned}$$

Возьмём начальное значение: переменные слева (они называются базисными) равны константам, а справа все переменные равны нулю. Что мы делаем дальше? Дальше мы пытаемся увеличить входящие переменные. Например, x_2. Насколько можно её увеличить, если оставить остальные переменные как есть и неотрицательными?

$$3 - x_2 \geqslant 01 - 3x_2 \geqslant 03 - x_2 \geqslant 0$$

Самое строгое второе. А значит мы можем сделать x_2 равным $\frac{1}{3}$. Теперь мы преобразуем второе неравенство (т.е. неравенство на y_2):

$$y_2 = 1 + x_1 - 3x_2 \Longrightarrow x_2 = \frac{1}{3} + \frac{1}{3}x_1 - \frac{1}{3}y_2$$

И заменим во всех остальных x_2 на это значение (чтобы y_2 стало входящей переменной, в x_2 — базисной:)

$$\begin{aligned} y_1 &= \frac{8}{3} - \frac{4}{3}x_1 + \frac{1}{3}y_2 \\ x_2 = \frac{1}{3} + \frac{1}{3}x_1 - \frac{1}{3}y_2 y_3 &= \frac{8}{3} - \frac{1}{3}x_1 + \frac{1}{3}y_2 \\ z &= \frac{1}{3} + \frac{4}{3}x_1 - \frac{1}{3}y_2 \end{aligned}$$

Теперь y_2 называется выходящей переменной.
Сделаем так же с x_1.

$$\begin{aligned} x_1 &= 2 - \frac{3}{4}y_1 + \frac{1}{4}y_2 \\ x_2 &= 1 - \frac{1}{4}y_1 - \frac{1}{4}y_2 \\ y_3 &= 2 + \frac{1}{4}y_1 + \frac{1}{4}y_2 \\ z &= 3 - y_1 \end{aligned}$$

После этого получим, что больше мы ничего увеличить не можем. Такое случается, когда в целевой функции (которая $z =$) все коэффициенты отрицательны. Тогда симплекс-метод заканчивается.
Частные случаи: если ни одно ограничение не даёт верхней границы на входящую переменную, то решение не ограничено сверху. Если ограничение даёт увеличение на ноль, забиваем на это (т.е. увеличиваем на ноль и продолжаем).
Какую входящую переменную выбирать? Есть разные подходы. Можно случайно, можно для каждой считать, какое увеличит оптимально. Но всё это не важно, ведь есть Brand's rule: выбирать входящую переменную с наименьшим индексом. Такой выбор обеспечивает конечное время работы симплекс-метода.
И есть ещё одна проблема: если константа отрицательна. Типа такого:

$$\begin{aligned} &\ldots \\ y_2 = -1 + x_1 - 3x_2 \quad &\ldots \end{aligned}$$

Тогда строят ещё одну задачу линейного программирования, которая поможет найти начальные значения в исходной задаче. Как она выглядит? Мы вводим дополнительную переменную x_0, которую вычитаем из каждого неравенства. И максимизируем мы $-x_0$. Если в результате этой программы x_0 равно нулю, то мы нашли стартовое значение исходной задачи. Если не равно, значит указанные ограничения не умеют выполняться при $x_0 = 0$, а значит эти ограничения просто несовместны.

При этом у вспомогательной задачи можно взять x_0, равное самому минимальному значению в ограничениях, и сделать x_0 входящей переменной на первом шаге, то нам будет откуда начать вспомогательный симплекс-метод.