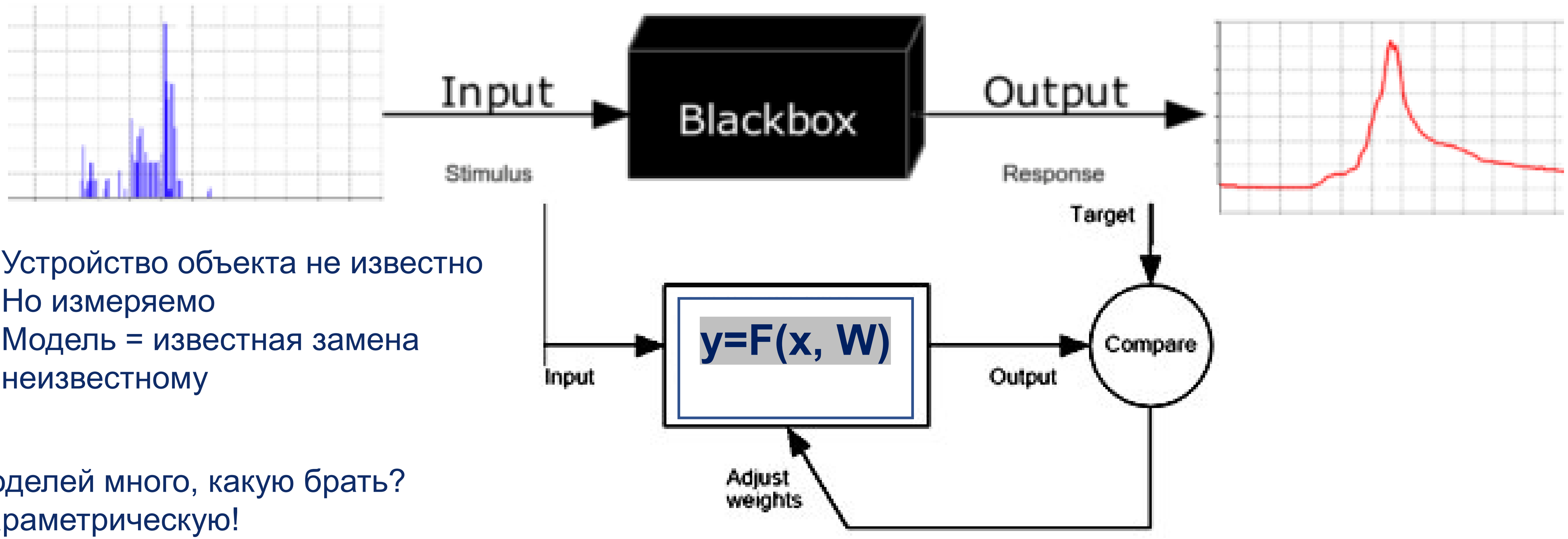
Лекция 03
Математический праймер для ML

А. Методы оптимизации
Б. Обратное распространение ошибки и автодифференцирование
В. Методы поиска и перебора

«Черный ящик»

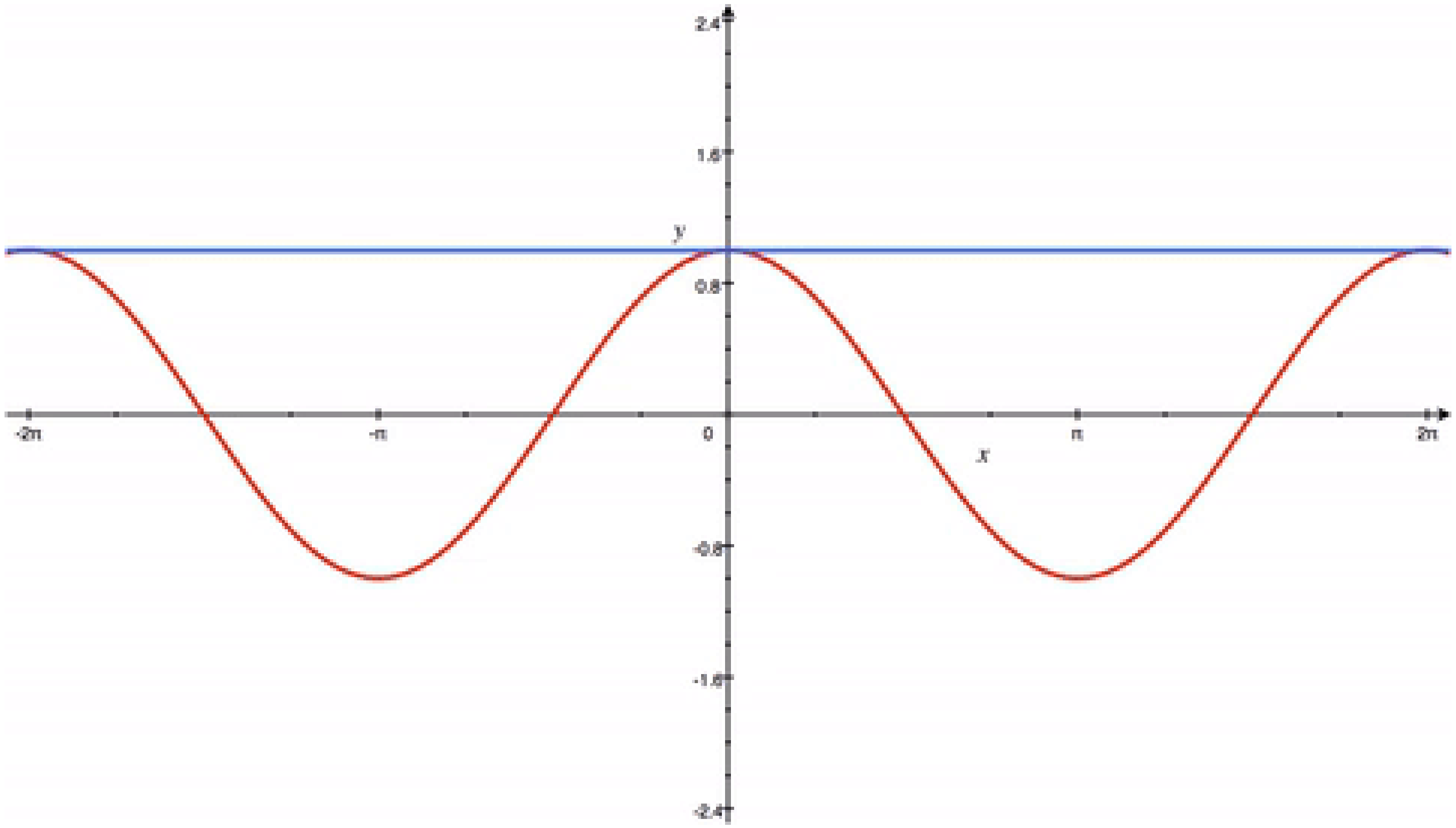


- Устройство объекта не известно
- Но измеряемо
- Модель = известная замена неизвестному

Моделей много, какую брать?
Параметрическую!

Обучение = Оптимизация

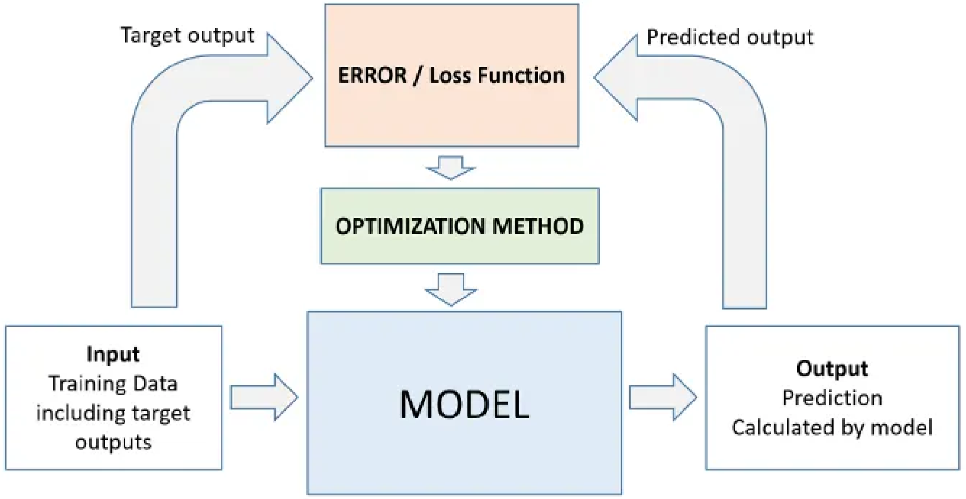
Найти такие параметры, чтобы при известных входах выходы модели совпадали как можно точнее с известными выходами объекта



Методы оптимизации

По методу:
Переборные
Градиентные
Поисковые

По охвату:
Локальные
Глобальные



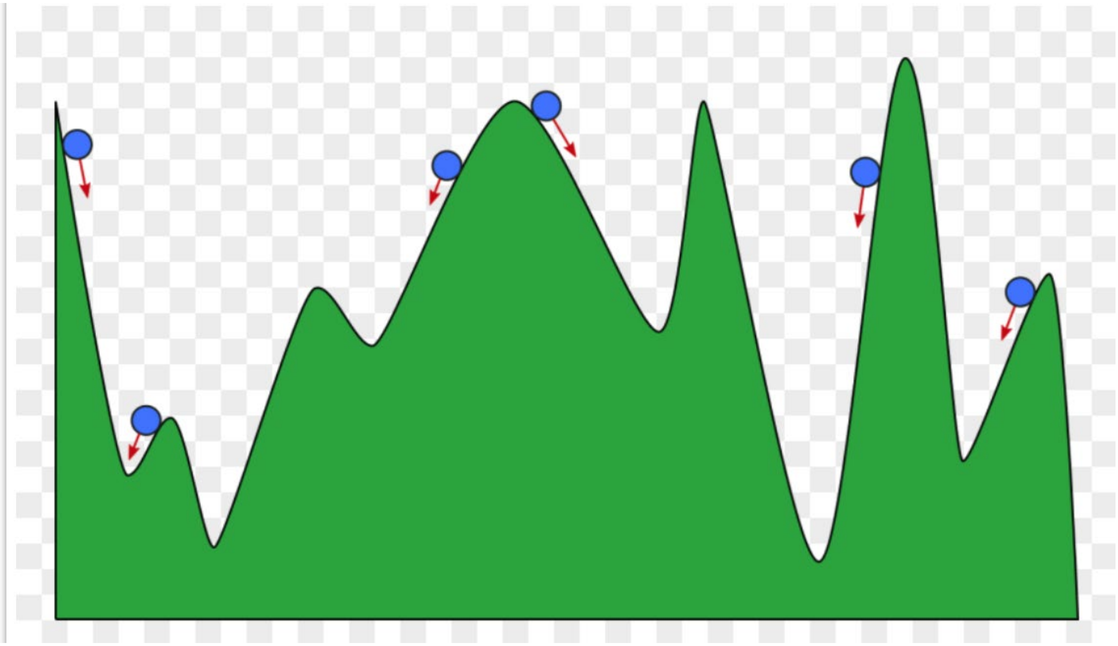
По охвату:
Локальные
Глобальные

По цикличности:
Цикличные (итерационные)
Ацикличные

По организации:

Онлайн	C	U	C	U	C	U	C	U	C	U	C	U	C	U
Оффлайн	C	C	C	C	C	C	C	C	C	C	C	C	C	U
Пакетные	C	C	C	C	U	C	C	C	C	U	C	C	C	U

Градиентные методы. Градиентный спуск

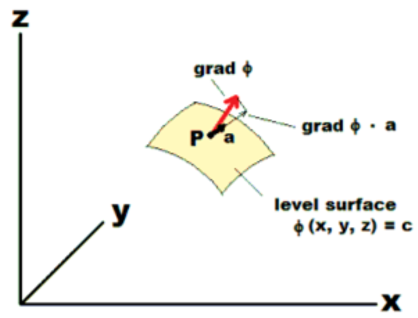


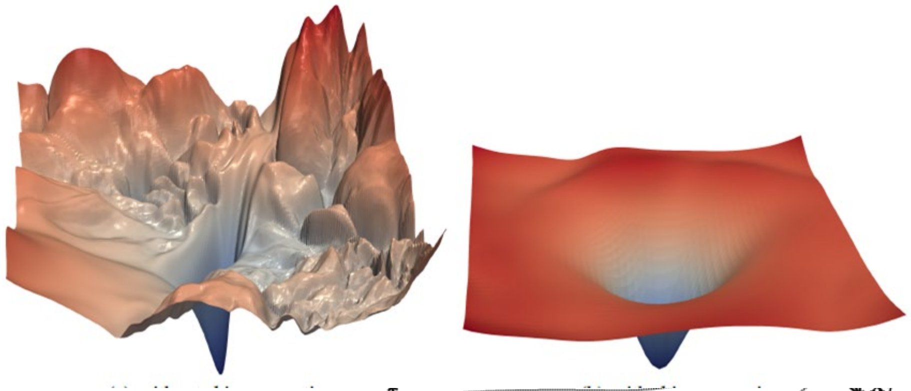
- Застревает в локальных минимумах
- Проблемы с узкими поверхностями
- Проблемы с широкими плато
- Не показывает направление на минимум
- Чувствителен к инициализации
- Чувствителен к величине шага learning rate
- Чувствителен к шуму

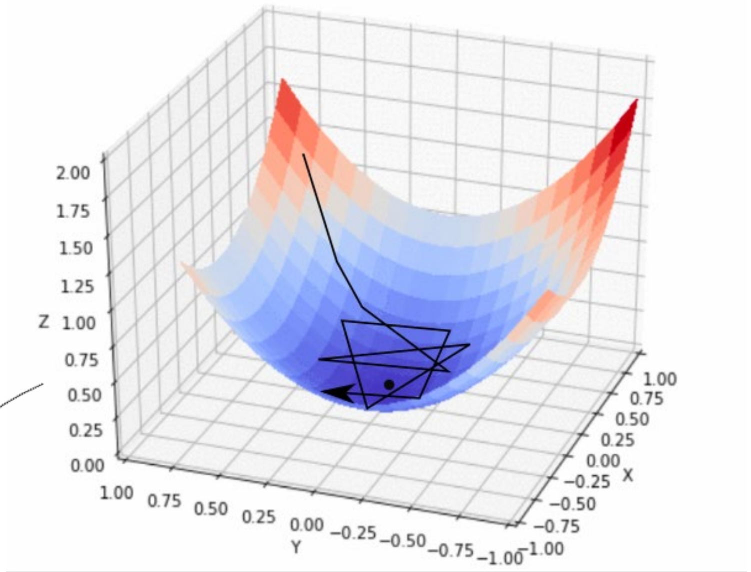
$$\text{for } \mathrm{n} = 1 : \mathrm{N}$$

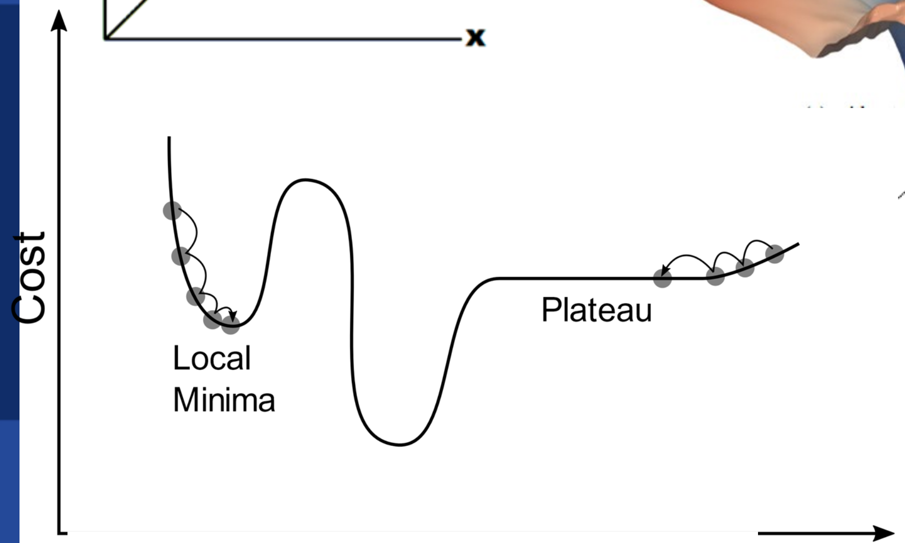
$$w_i^n := w_i^{n-1} - \eta \frac{\partial E_n(\mathrm{w})}{\partial w_i}$$

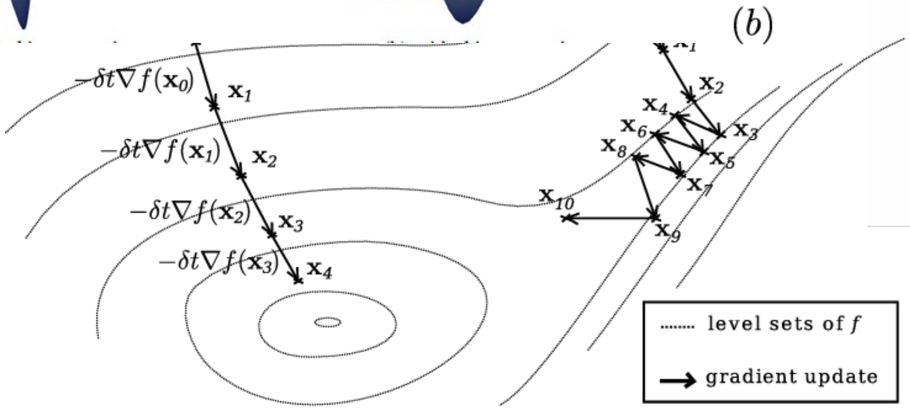
Проблемы градиентного спуска

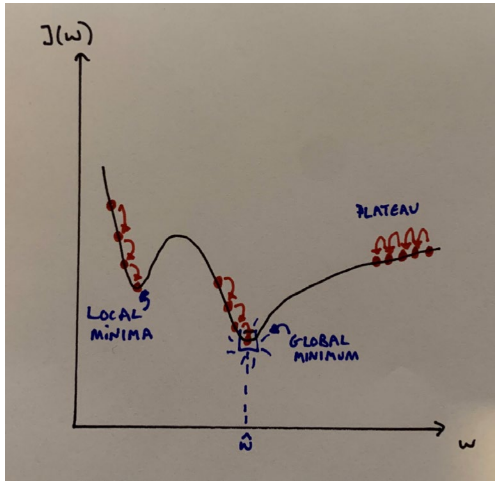


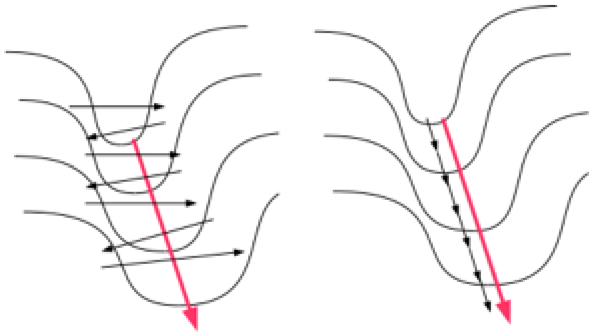


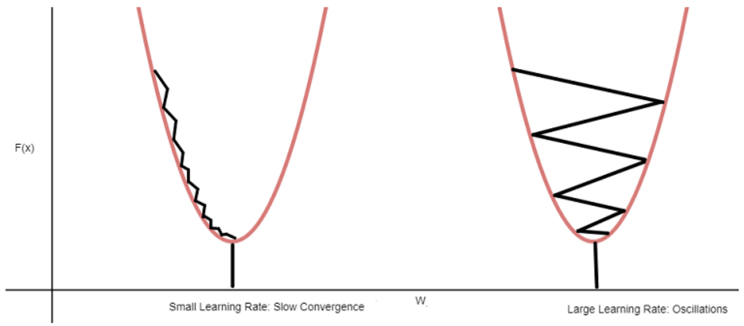




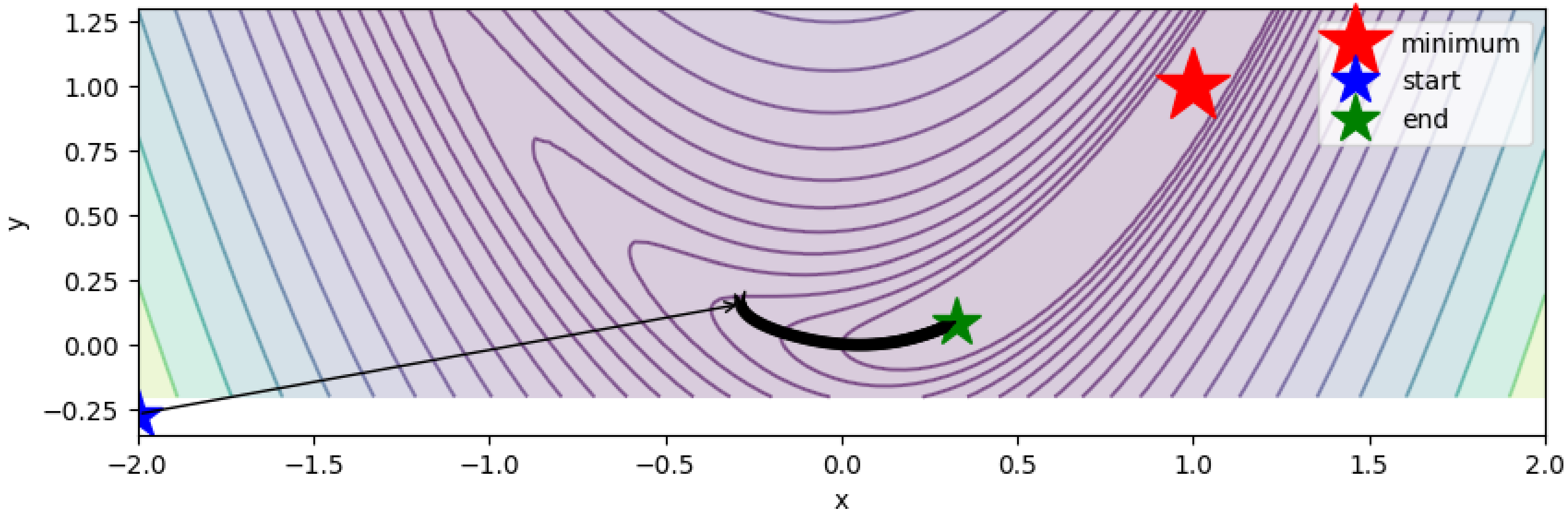






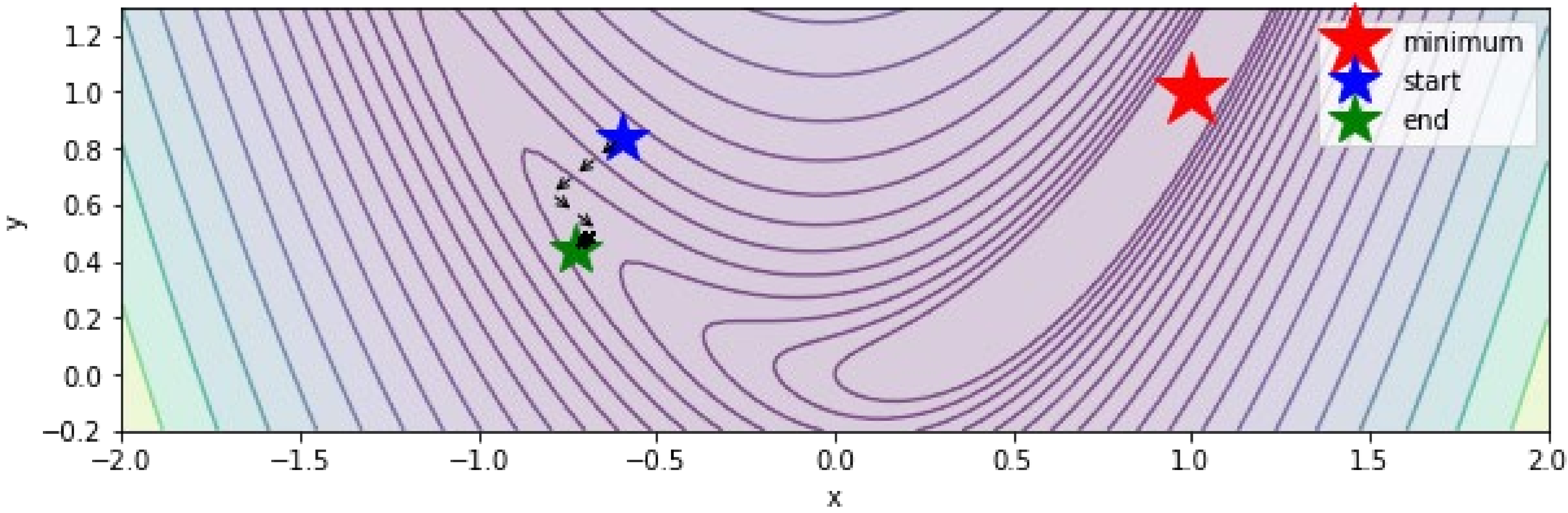


Градиентный спуск



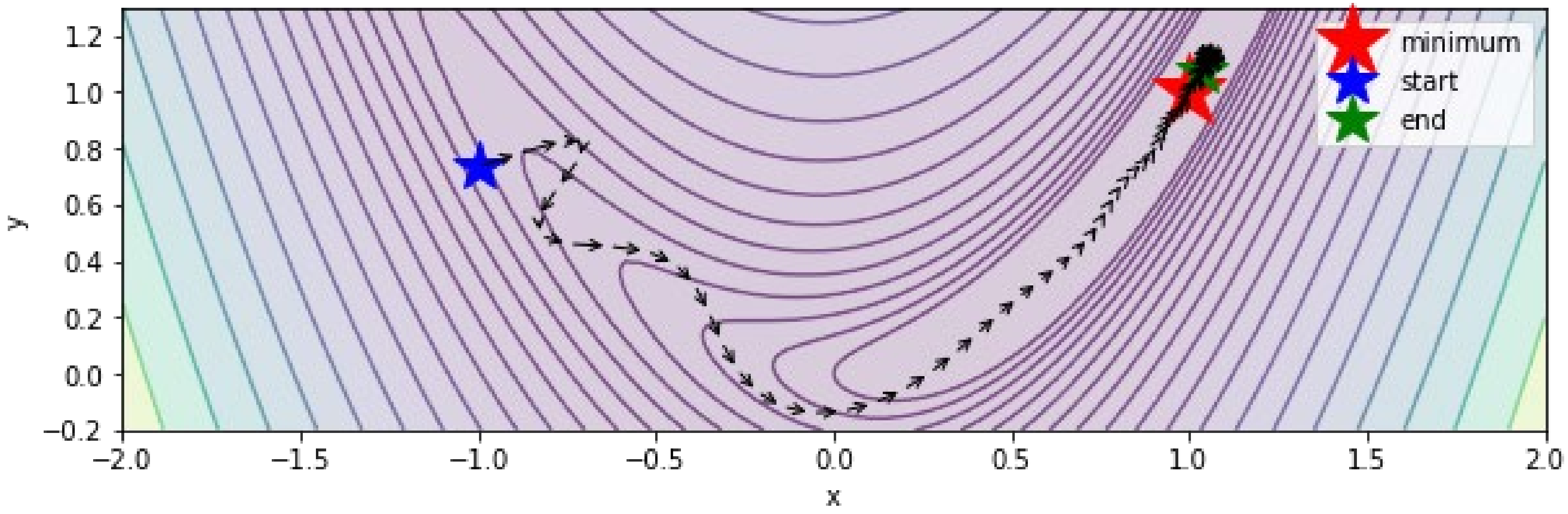
Знак градиента

$$\Delta\boldsymbol{\theta} = \eta * sign(\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}))$$
$$\boldsymbol{\theta} = \boldsymbol{\theta} - \Delta\boldsymbol{\theta}$$



Первый момент

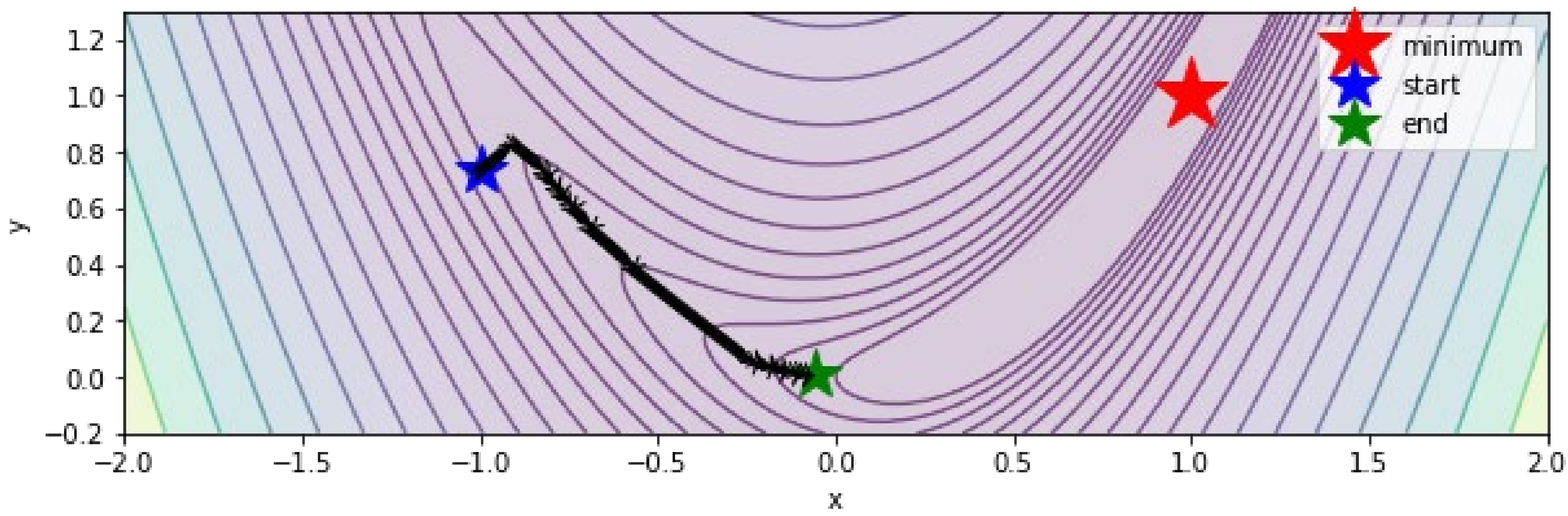
$$\Delta\boldsymbol{\theta} = \gamma\Delta\boldsymbol{\theta} + \eta\nabla_{\boldsymbol{\theta}}J(\boldsymbol{\theta})$$
$$\boldsymbol{\theta} = \boldsymbol{\theta} - \Delta\boldsymbol{\theta}$$



Усреднение. RMSProp

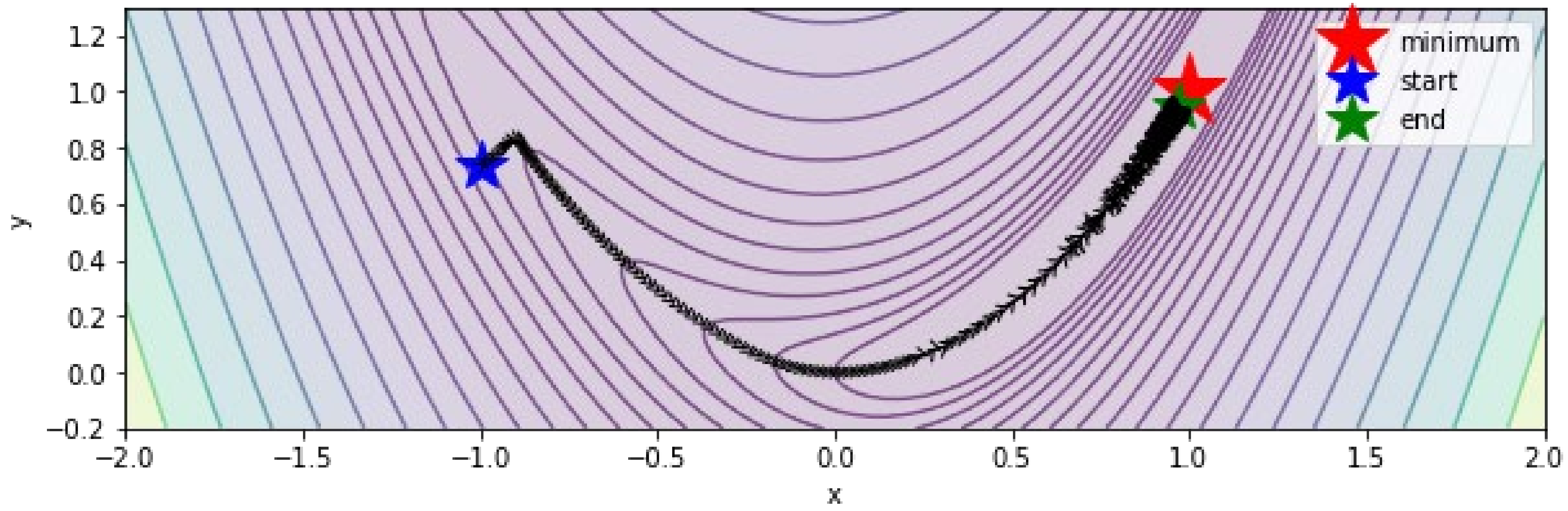
$$\mathbf{v} = \gamma \mathbf{v} + (1-\gamma)(\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}))^2$$

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \frac{\eta}{\sqrt{\mathbf{v}} + \epsilon} \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

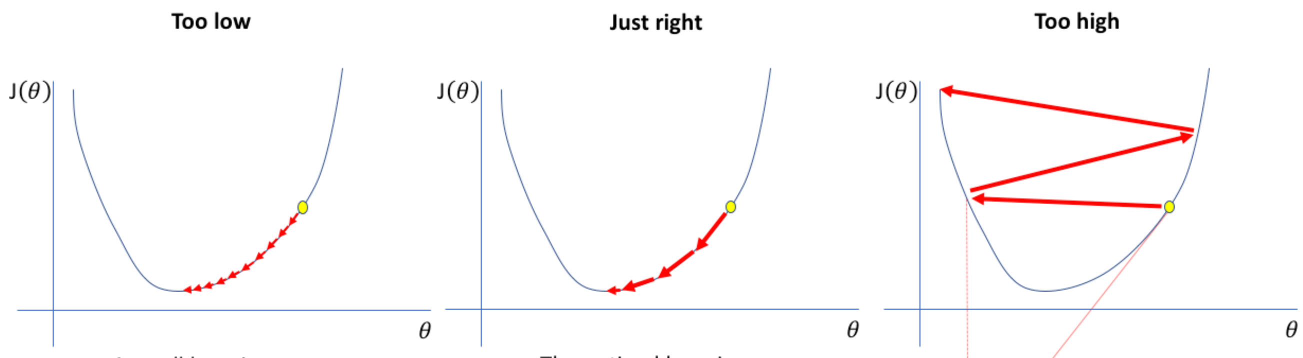


AdaM

$$\mathbf{m} = \beta_1 \mathbf{m} + (1-\beta_1)\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$
$$\mathbf{v} = \beta_2 \mathbf{v} + (1-\beta_2)(\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}))^2$$
$$\hat{\mathbf{m}} = \frac{\mathbf{m}}{1-\beta_1^t}$$
$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{1-\beta_2^t}$$
$$\boldsymbol{\theta} = \boldsymbol{\theta} - \frac{\eta}{\sqrt{\hat{\mathbf{v}}}+\epsilon}\hat{\mathbf{m}}$$



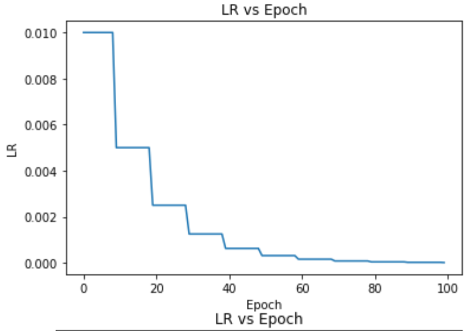
Изменение шага learning rate



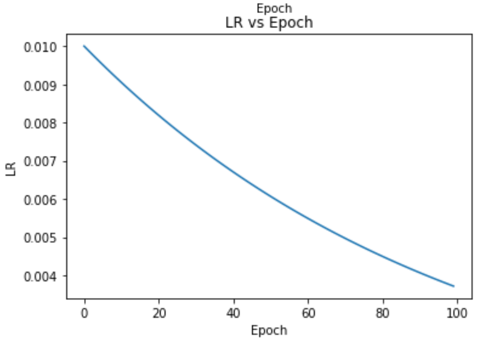
A small learning rate requires many updates before reaching the minimum point

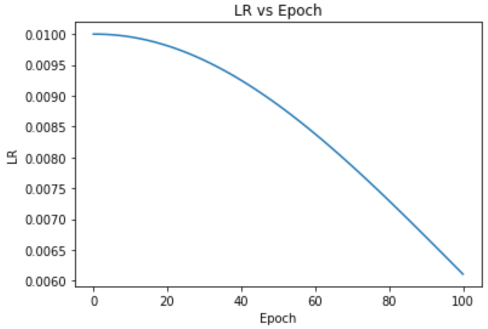
The optimal learning rate swiftly reaches the minimum point

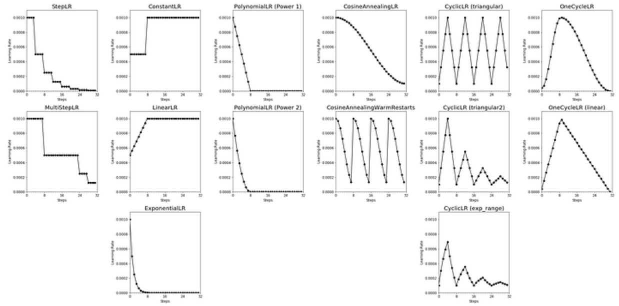
Too large of a learning rate causes drastic updates which lead to divergent behaviors



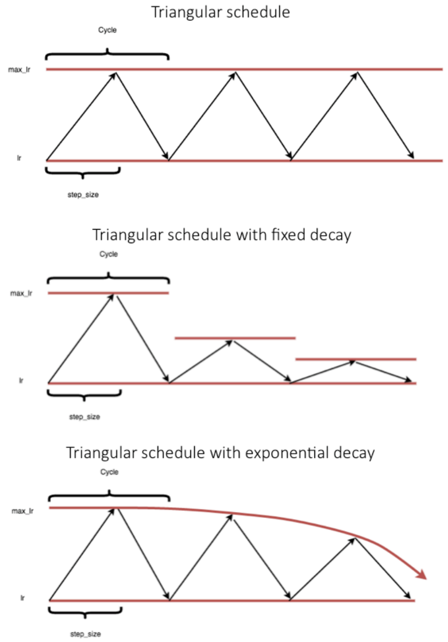
$$\eta = \eta \times \frac{1}{1 + \delta \times t}$$

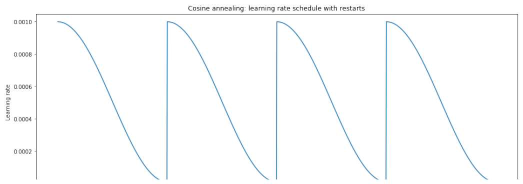






https://www.leoniemonigatti.com/blog/pytorch-learning-rate-schedulers.html





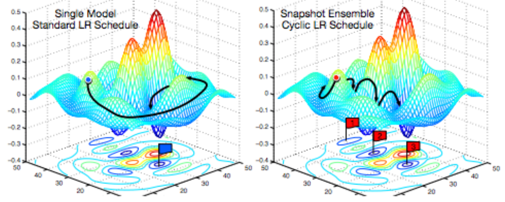


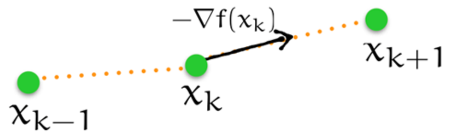
Figure 1: Left: Illustration of SGD optimization with a typical learning rate schedule. The model converges to a minimum at the end of training. **Right:** Illustration of Snapshot Ensembling. The model undergoes several learning rate annealing cycles, converging to and escaping from multiple local minima. We take a snapshot at each minimum for test-time ensembling.

https://www.jeremyjordan.me/nn-learning-rate/

Градиент Нестерова

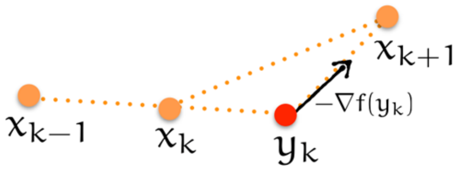
Gradient Descent

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



Accelerated GD

$$x_{k+1} = y_k - \epsilon \nabla f(y_k)$$
$$y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1})$$

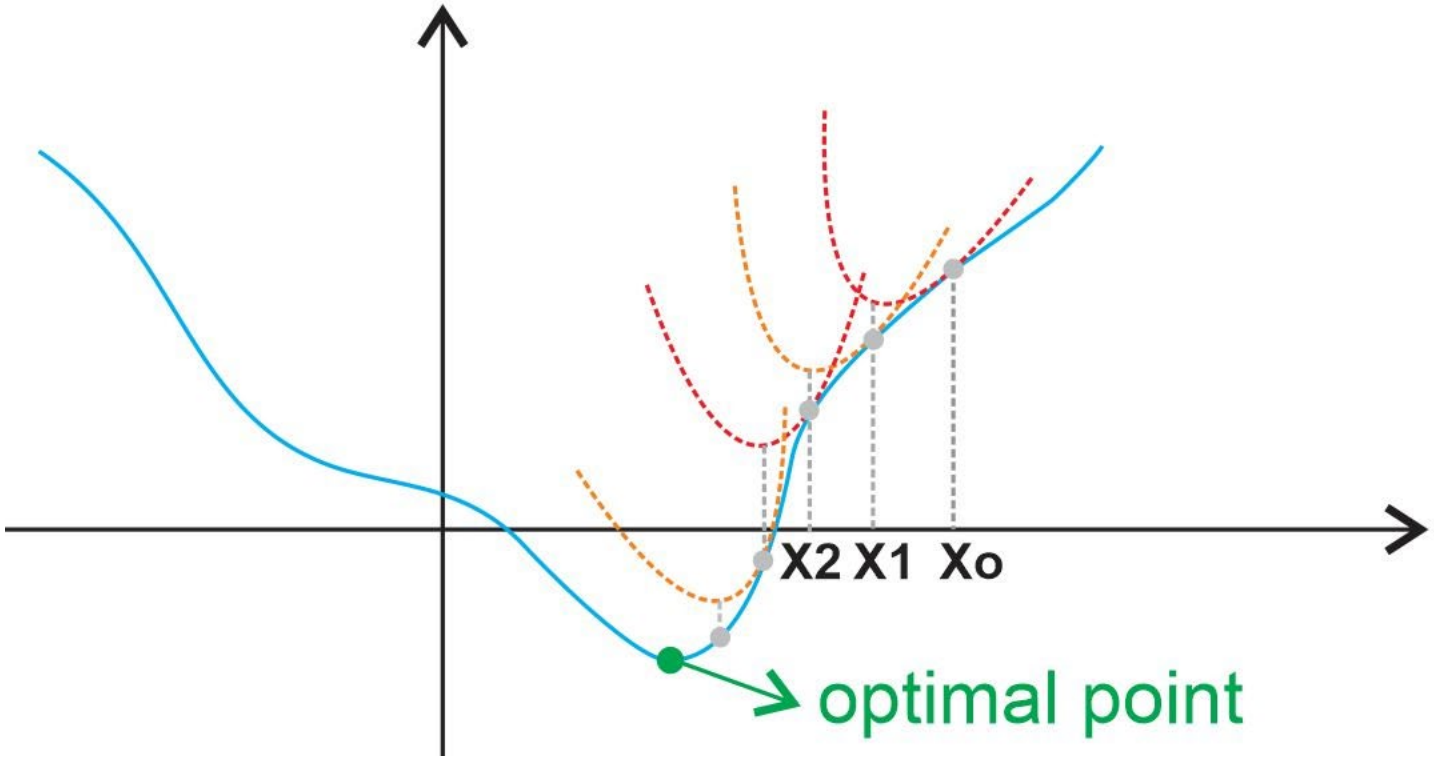


when $\|\nabla^2 f\| \leq \frac{1}{\epsilon}$

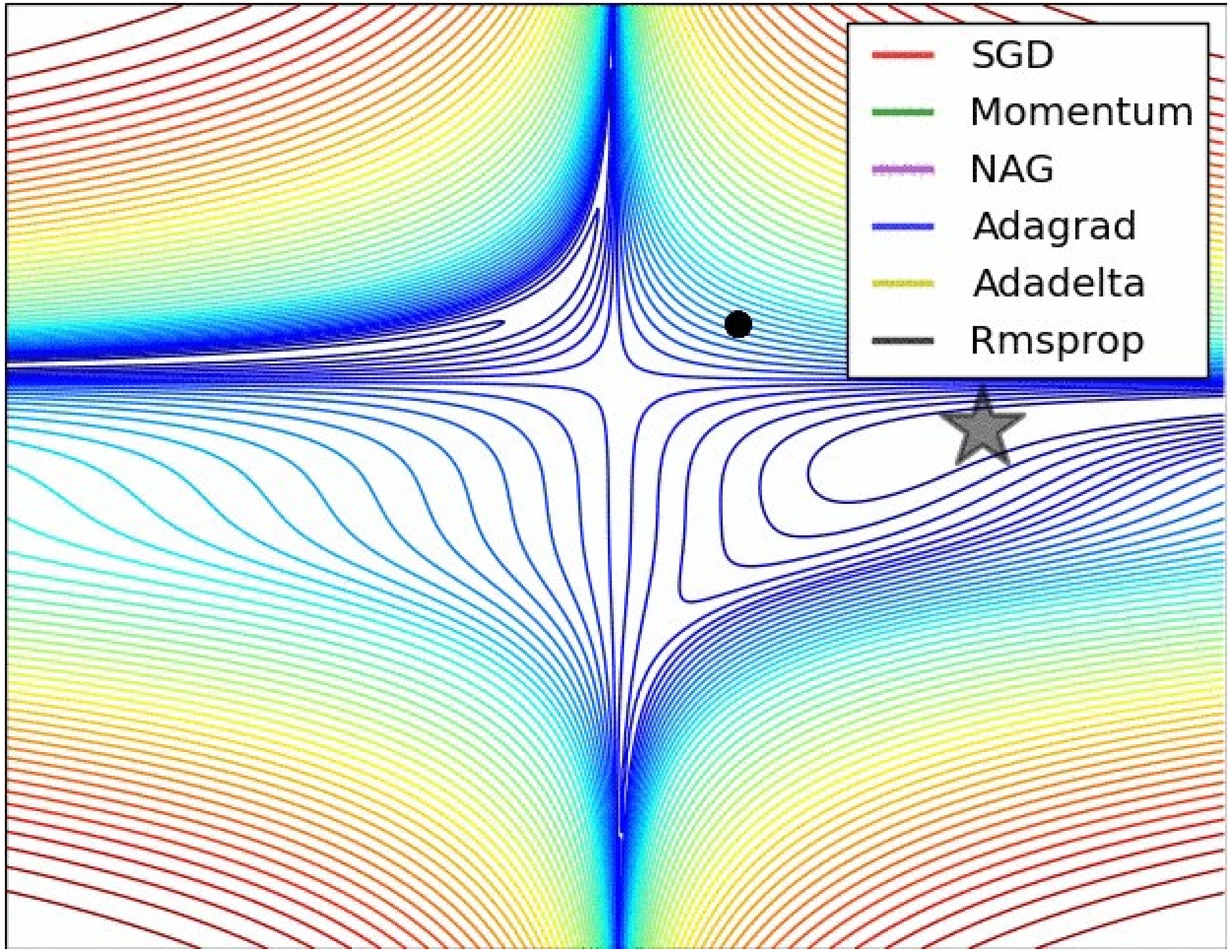
$$O\left(\frac{1}{\epsilon k}\right) \xrightarrow{\text{accelerated!}} O\left(\frac{1}{\epsilon k^2}\right)$$

optimal rate

Методы второго порядка



Другие идеи методов



И сотни других методов и их комбинаций:

- Увеличить размер истории (памяти)
- Неградиентные методы
- Комбинации всяческих методов, усреднением или последовательно
- Манипуляции с гиперпараметрами
- И др..

Цепное правило дифференцирования

Пусть: $z = f(u, v)$, $u = \varphi(x, y)$, $v = \psi(x, y)$, т.е.
$z = f(\varphi(x, y), \psi(x, y))$.

Производная сложной функции многих переменных: *производная внешней функции по каждому из промежуточных аргументов умножается на производную этого промежуточного аргумента по основному аргументу и все такие произведения суммируются.*

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial u} \cdot \frac{\partial u}{\partial x} + \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial x}, \quad \frac{\partial z}{\partial y} = \frac{\partial z}{\partial u} \cdot \frac{\partial u}{\partial y} + \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial y}$$

"Цепное правило":

$$z = f(u, v, \ldots, w)\ ; \quad u = u(x, y)\ , \quad v = v(x, y)\ , \ldots, \quad w = w(x, y)$$

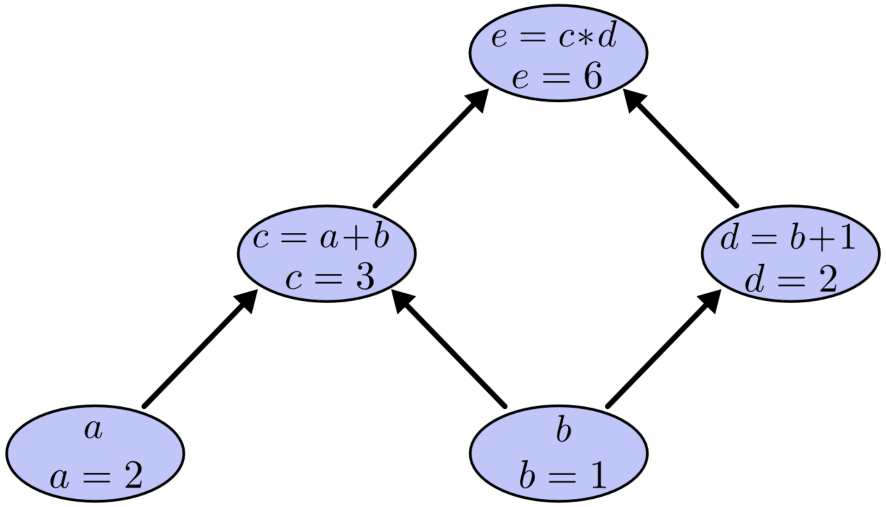
$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial u} \cdot \frac{\partial u}{\partial x} + \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial x} + \ldots + \frac{\partial z}{\partial w} \cdot \frac{\partial w}{\partial x}$$

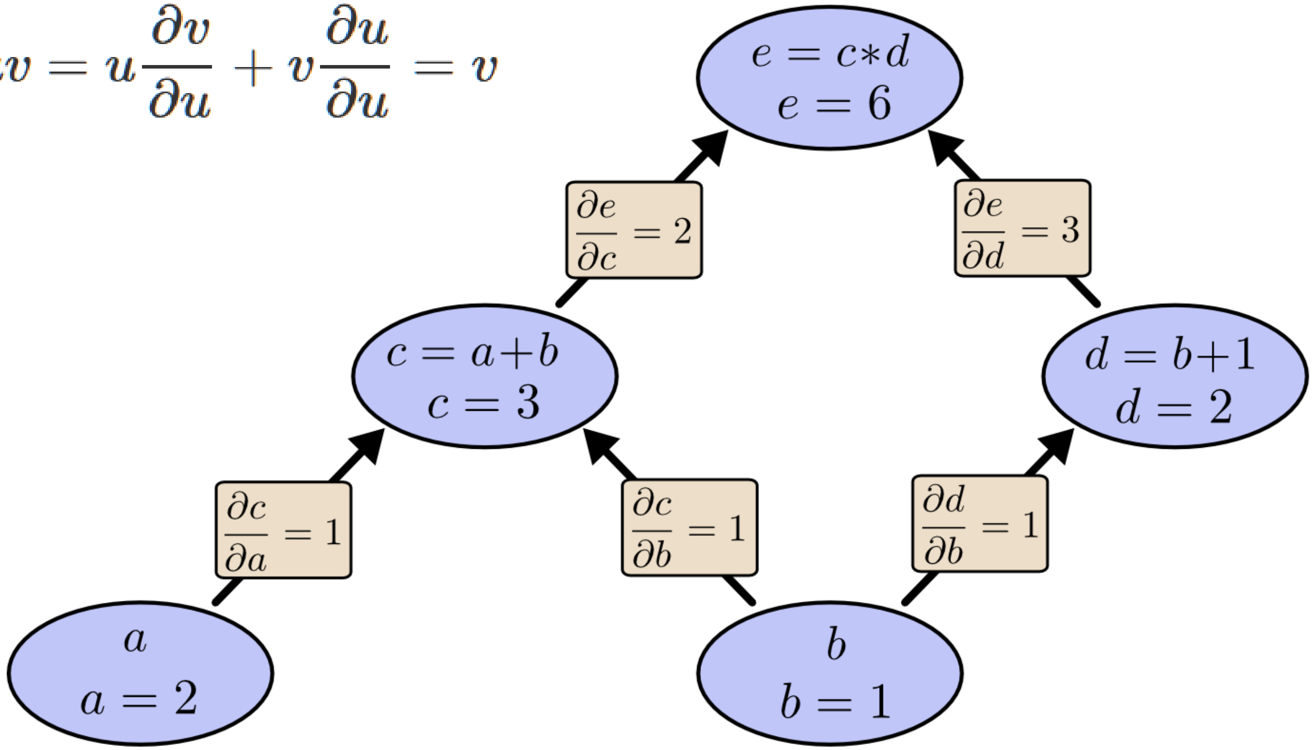
Автодифференцирование

c=a+b
d=b+1
e=c∗d

$$\frac{\partial}{\partial a}(a+b) = \frac{\partial a}{\partial a} + \frac{\partial b}{\partial a} = 1$$

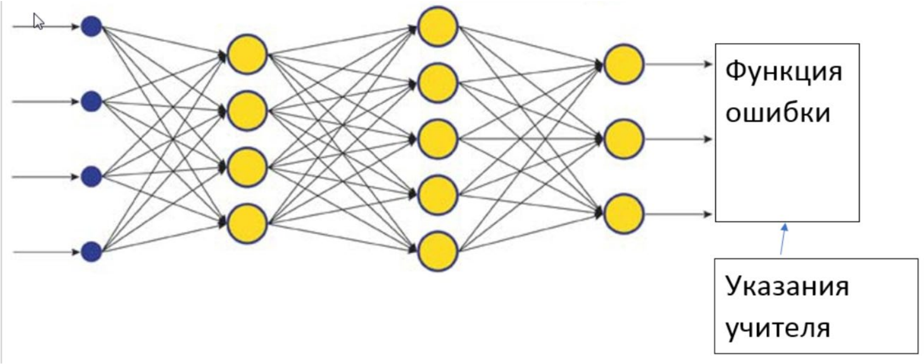
$$\frac{\partial}{\partial u}uv = u\frac{\partial v}{\partial u} + v\frac{\partial u}{\partial u} = v$$





https://colah.github.io/posts/2015-08-Backprop/

Обратное распространение ошибки



$$E(y_1^M, y_2^M, y_3^M, \dots, t)$$

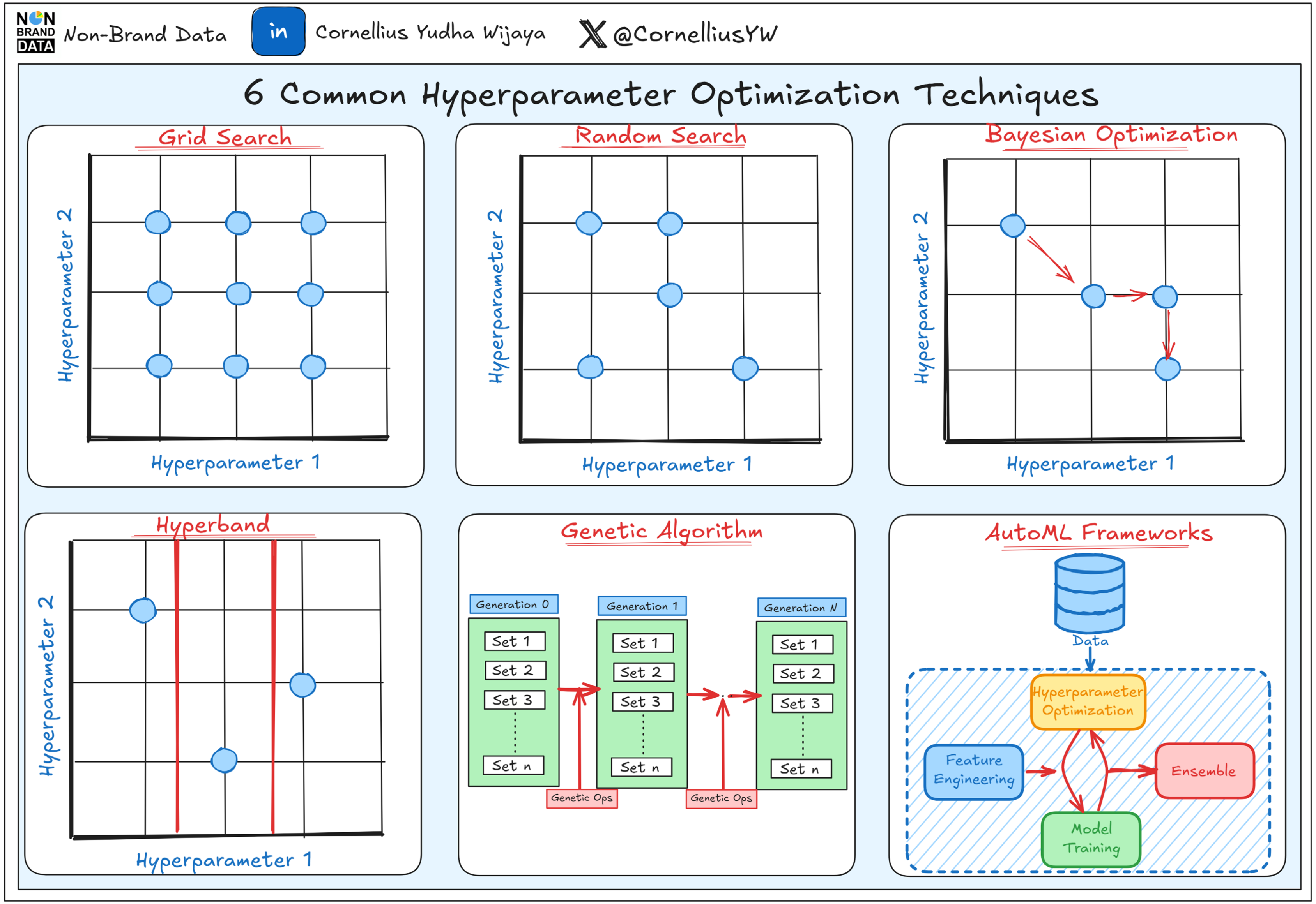
$$y_1^m(y_1^{m-1}, y_2^{m-1}, y_3^{m-1} \dots, W_1^m);$$
$$y_2^m(y_1^{m-1}, y_2^{m-1}, y_3^{m-1} \dots, W_2^m);$$
$$y_3^m = \dots$$

Обратное распространение ошибки

$$\frac{\partial E}{\partial y_i^m} = \sum_{j=1}^{N_{m+1}} \frac{\partial E}{\partial y_j^{m+1}} * \frac{\partial y_j^{m+1}}{\partial y_i^m}$$

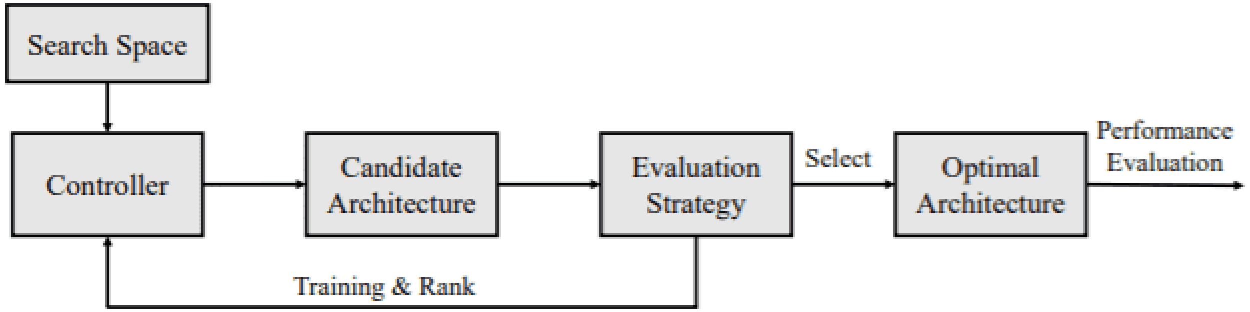
$$\frac{\partial E}{\partial w_{ij}^m} = \frac{\partial E}{\partial y_j^m} * \frac{\partial y_j^m}{\partial w_{ij}^m}$$

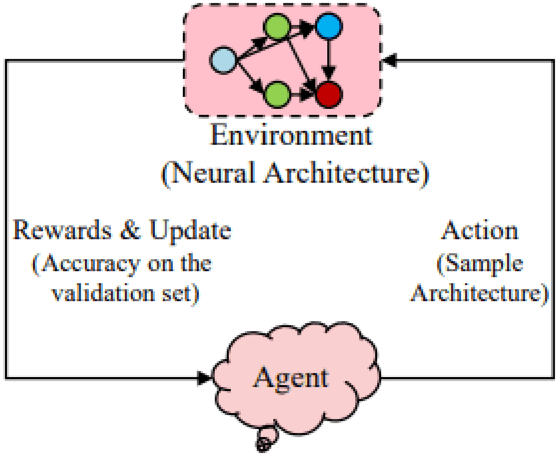
Настройка гиперпараметров

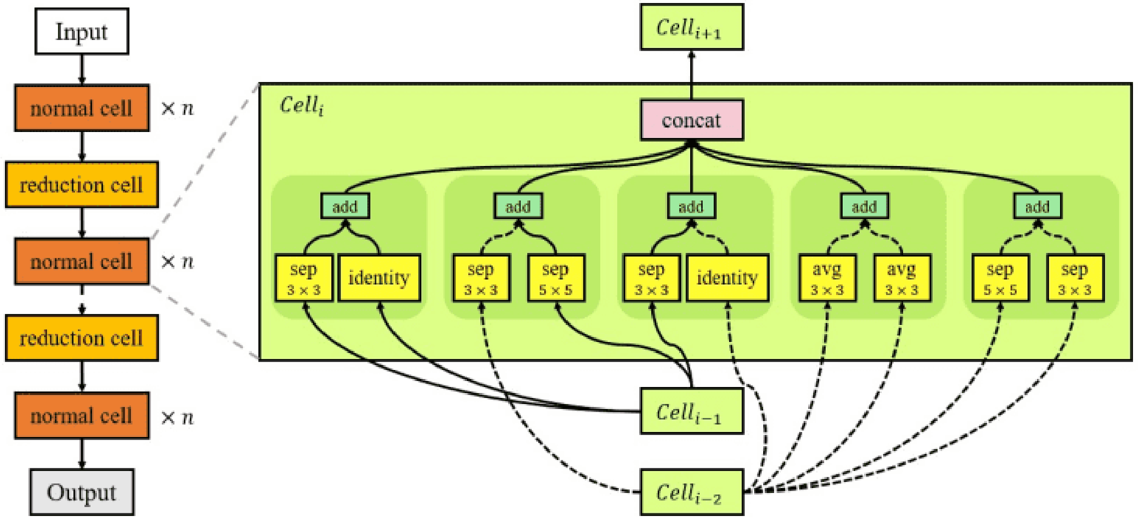


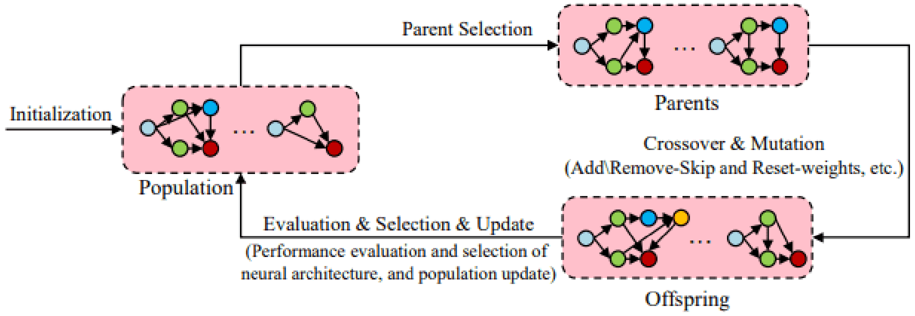
https://www.nb-data.com/p/6-common-hyperparameter-optimization

Поиск архитектуры модели



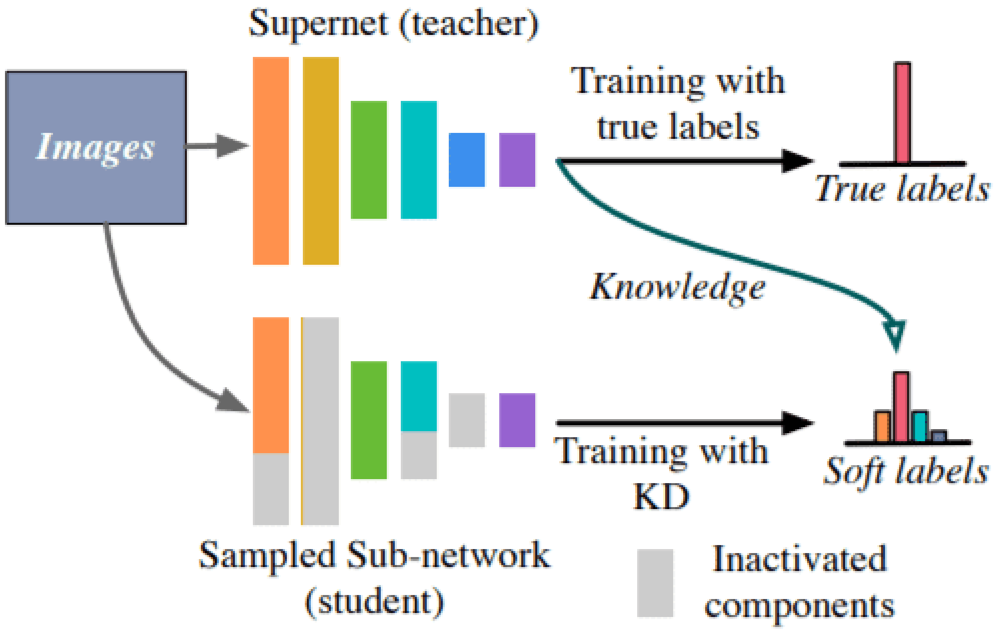


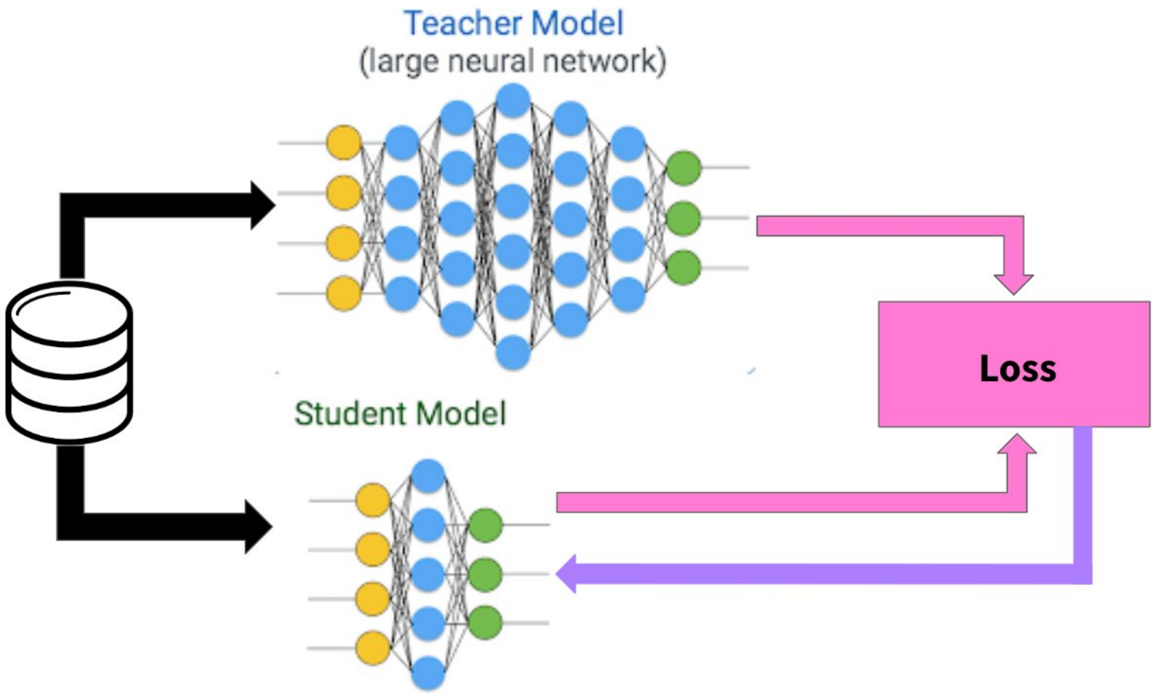




https://theaisummer.com/neural-architecture-search/

Поиск архитектуры модели





Q&A и контакты

Группа по дисциплине:

https://t.me/+8dShF1tFSDg0ZmJi