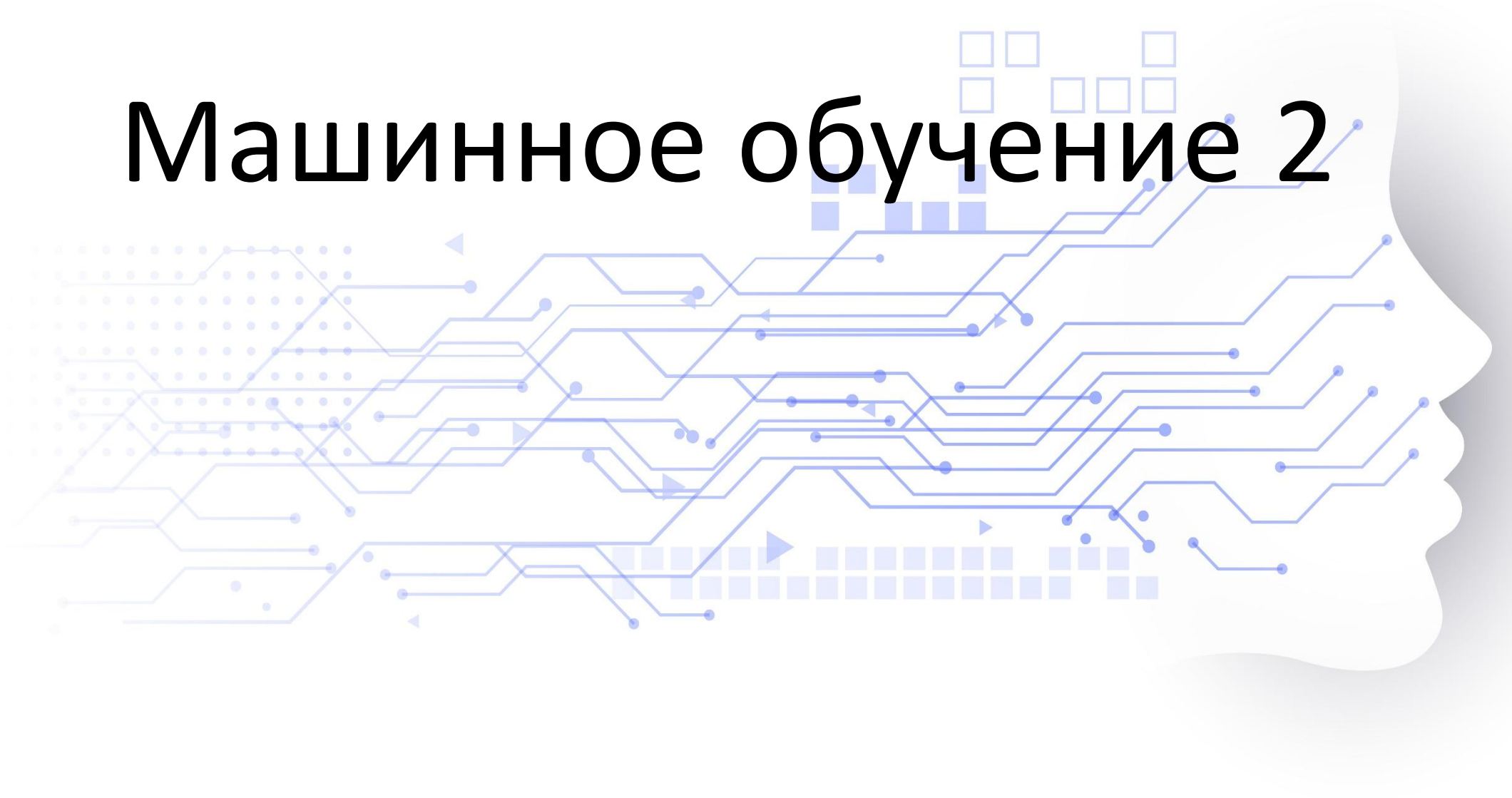
Машинное обучение 2



Резаиан Наим

E-mail: rezaian-n@rudn.ru

Telegram: @NaeimRezaeian

1. Заведующий лабораторией искусственного интеллекта

2. Руководитель направления разработок Центра развития цифровых технологий в образовательных процессах

3. Старший преподаватель факультета искусственного интеллекта

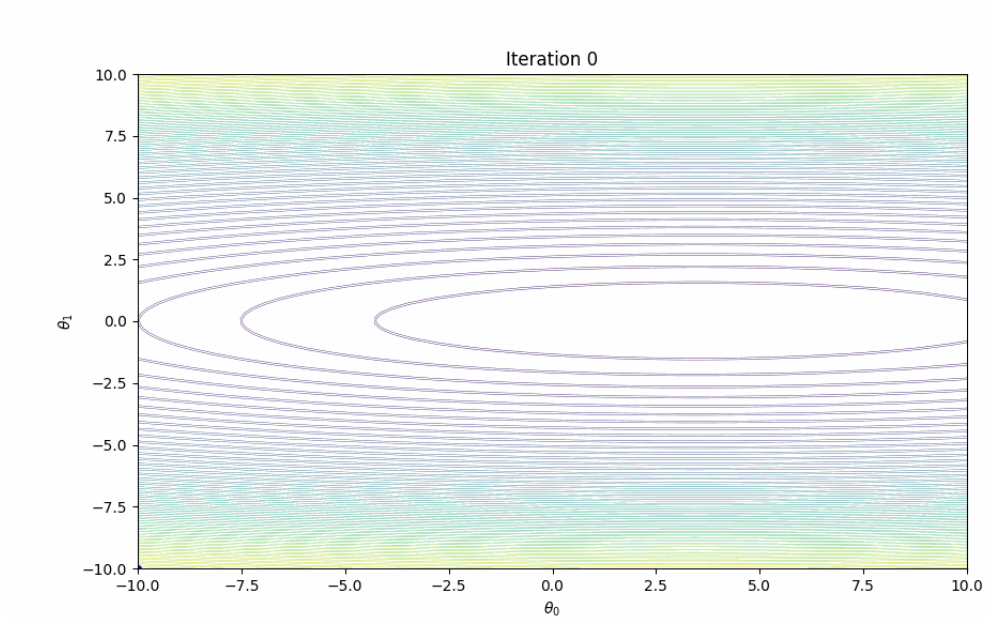
Продвинутые методы оптимизации

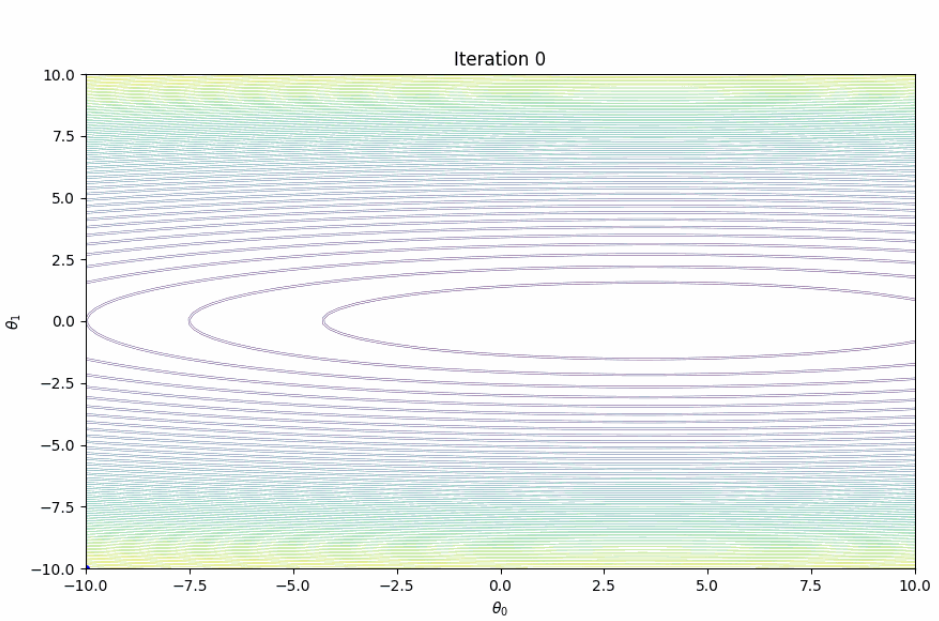
Оптимизаторы градиентных алгоритмов

Стохастический градиентный спуск (Stochastic gradient descent)

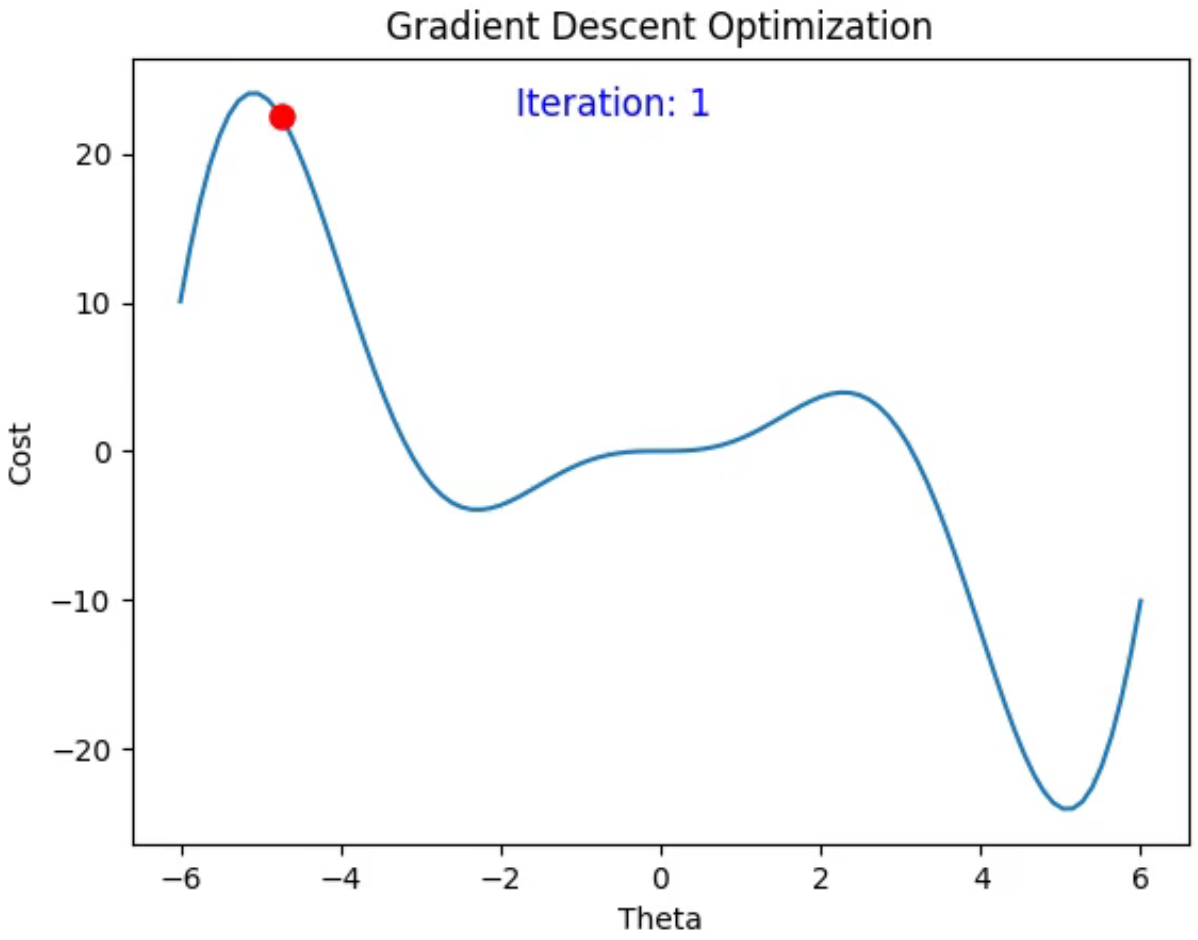
$$\theta^t = \theta^{t-1} - \alpha \nabla_{\boldsymbol{\theta}} L(\theta)$$

Оптимизаторы градиентных алгоритмов



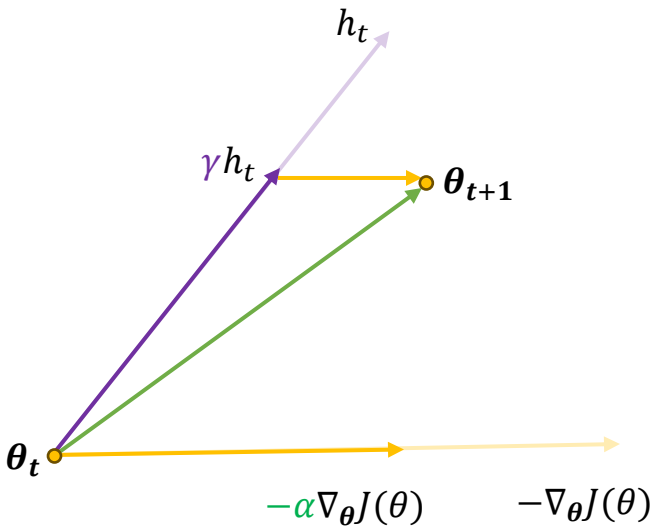


Оптимизаторы градиентных алгоритмов



Метод импульсов (momentum)

Борис Теодорович Поляк, 1964 год



$$h_{t+1} = \gamma h_t + \alpha\nabla_{\boldsymbol{\theta}}J(\theta)$$

$$\boldsymbol{\theta_{t+1}} = \boldsymbol{\theta_t} - h_{t+1}$$

h_t – инерция, усреднённое направление движения

γ – Коэффицент момента

α – скорость обучения

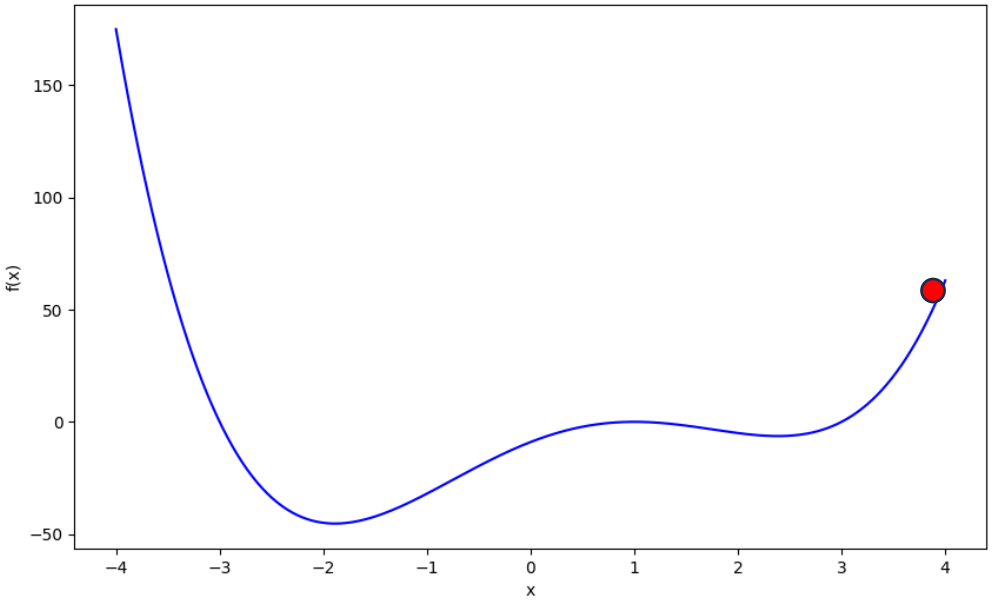
$\nabla_{\boldsymbol{\theta}}J(\theta)$ – градиент функции потерь

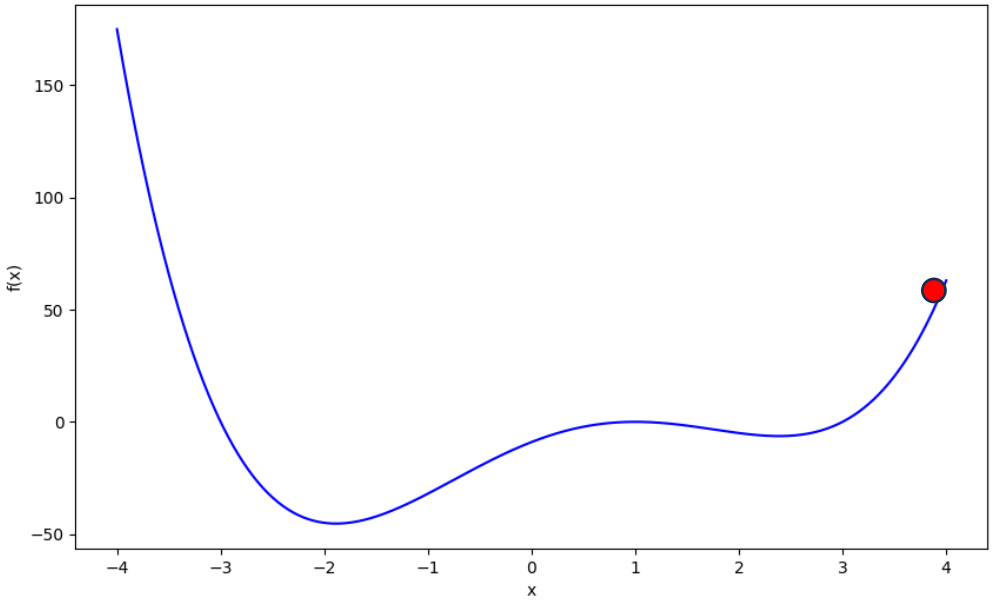
Метод импульсов (momentum)

Борис Теодорович Поляк, 1964 год

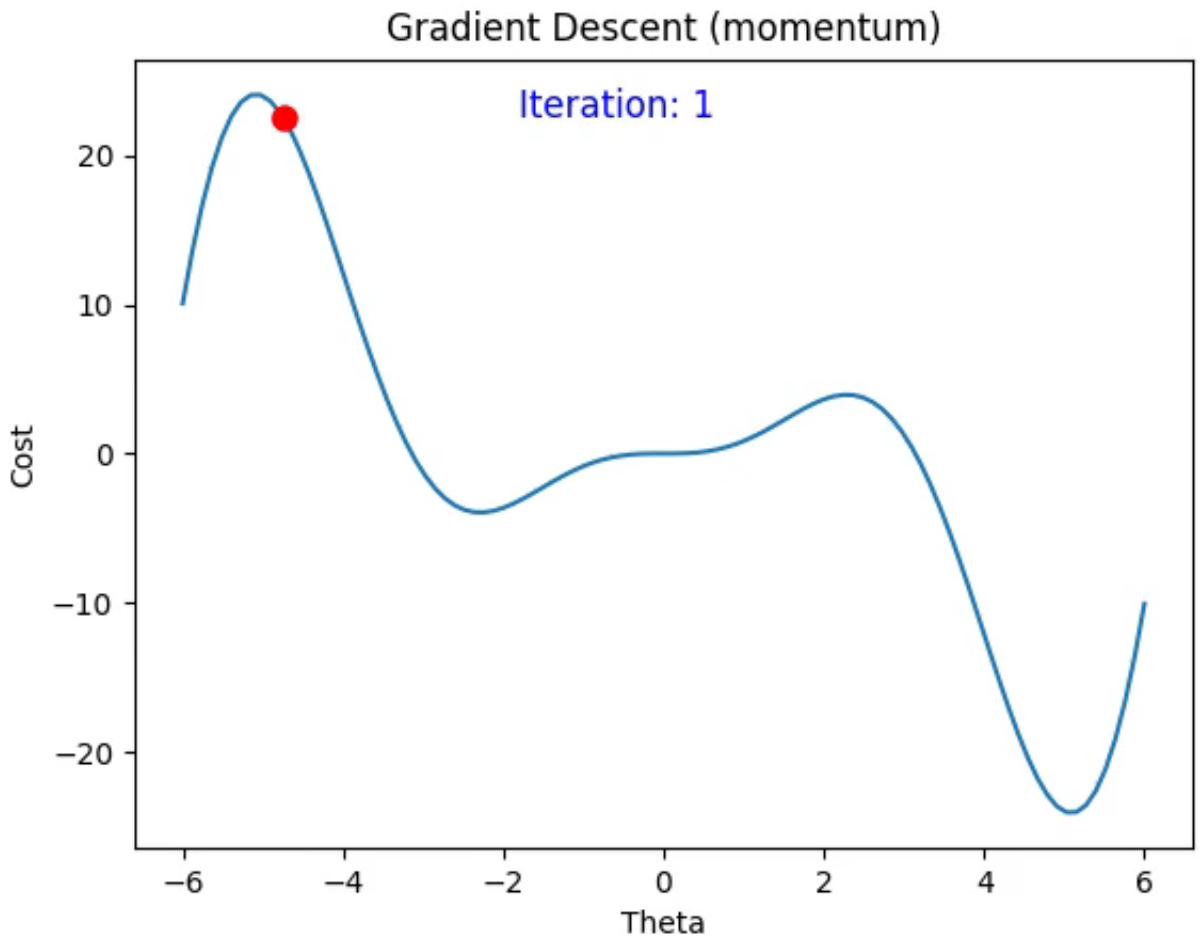
$$h_{t+1} = \gamma h_t + \alpha \nabla_{\boldsymbol{\theta}} J(\theta)$$

$$\boldsymbol{\theta_{t+1}} = \boldsymbol{\theta_t} - h_{t+1}$$

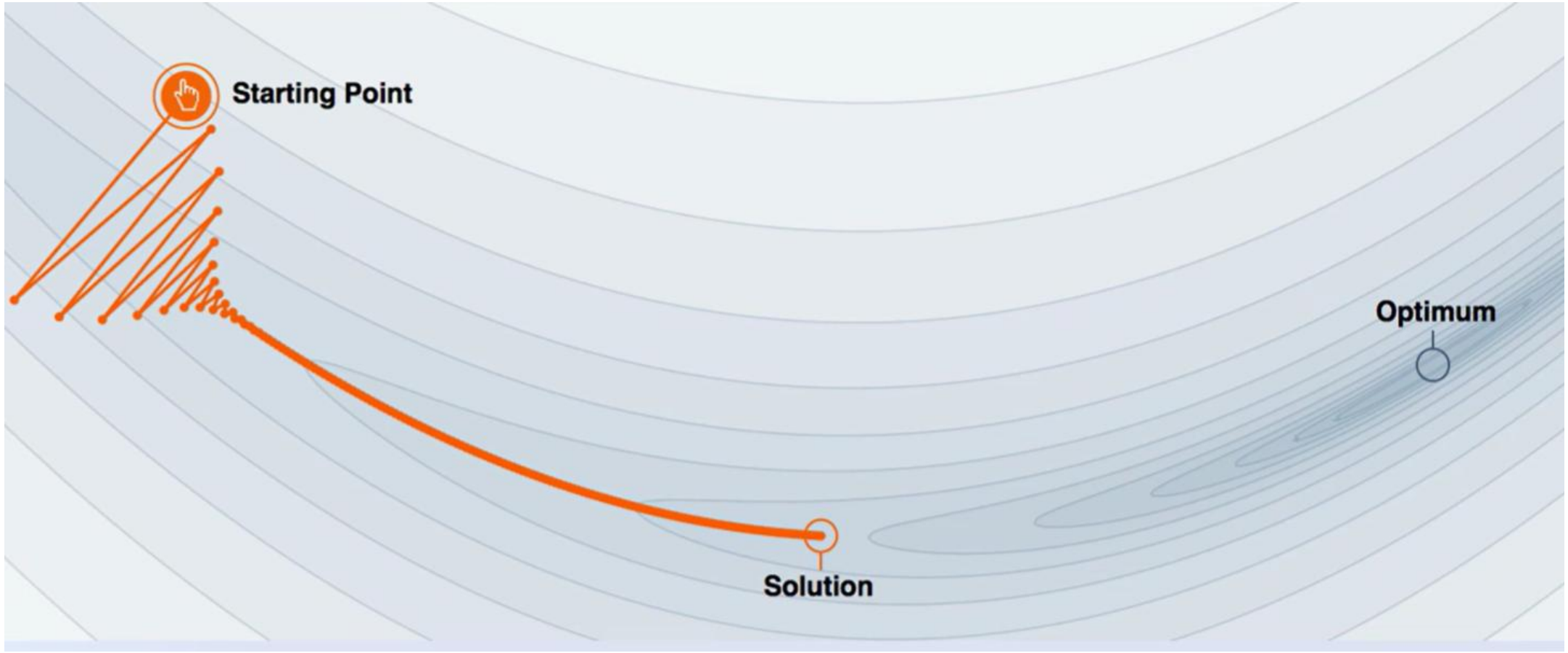




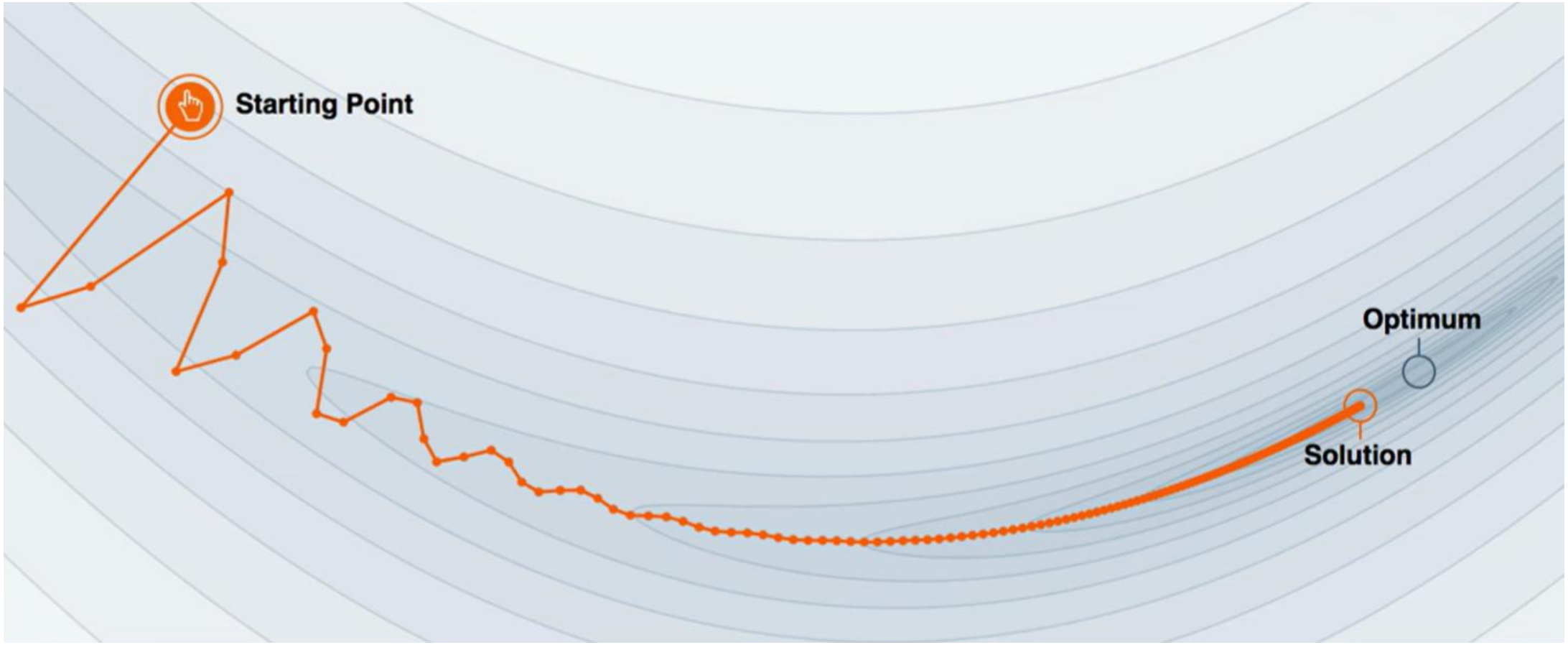
Оптимизаторы градиентных алгоритмов



Метод импульсов (momentum)



Метод импульсов (momentum)



Метод импульсов (momentum)

$$h_{t+1} = \gamma h_t + \alpha \nabla_{\boldsymbol{\theta}} J(\theta)$$

$$\boldsymbol{\theta_{t+1}} = \boldsymbol{\theta_t} - h_{t+1}$$

1. Благодаря "инерции", алгоритм может эффективнее выходить из мелких локальных минимумов, увеличивая шансы найти более оптимальное решение.

2. Момент сглаживает колебания в обновлениях параметров, делая процесс обучения более стабильным и предсказуемым.

Параметр γ определяет, какая доля предыдущего обновления будет добавлена к текущему градиенту. Обычно находится в диапазоне от 0 (без Momentum) до 1. Большие значения помогают ускорить SGD и сделать его более устойчивым к осцилляциям.

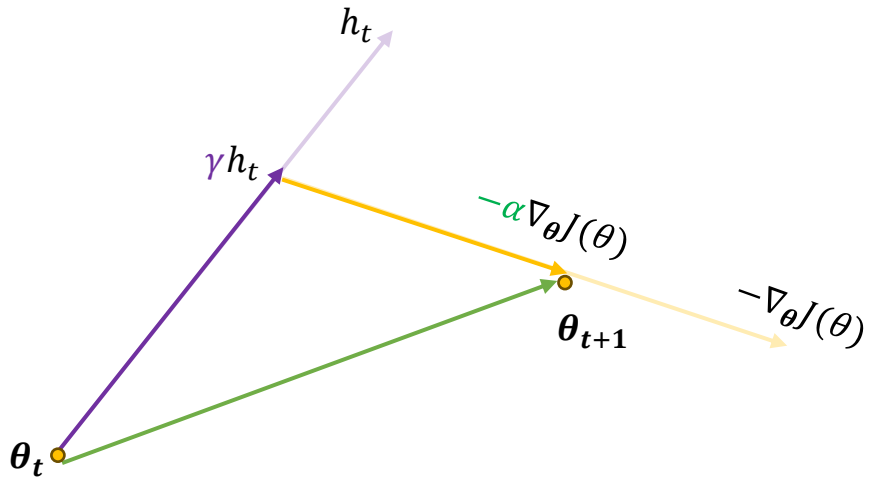
Стохастический градиент с импульсом Нестерова – NAG

Nesterov Accelerated Gradient

Нестеров, 1983 год

$$h_{t+1} = \gamma h_t + \alpha \nabla_{\boldsymbol{\theta}} J(\theta - \gamma h_t)$$

$$\boldsymbol{\theta_{t+1}} = \boldsymbol{\theta_t} - h_{t+1}$$

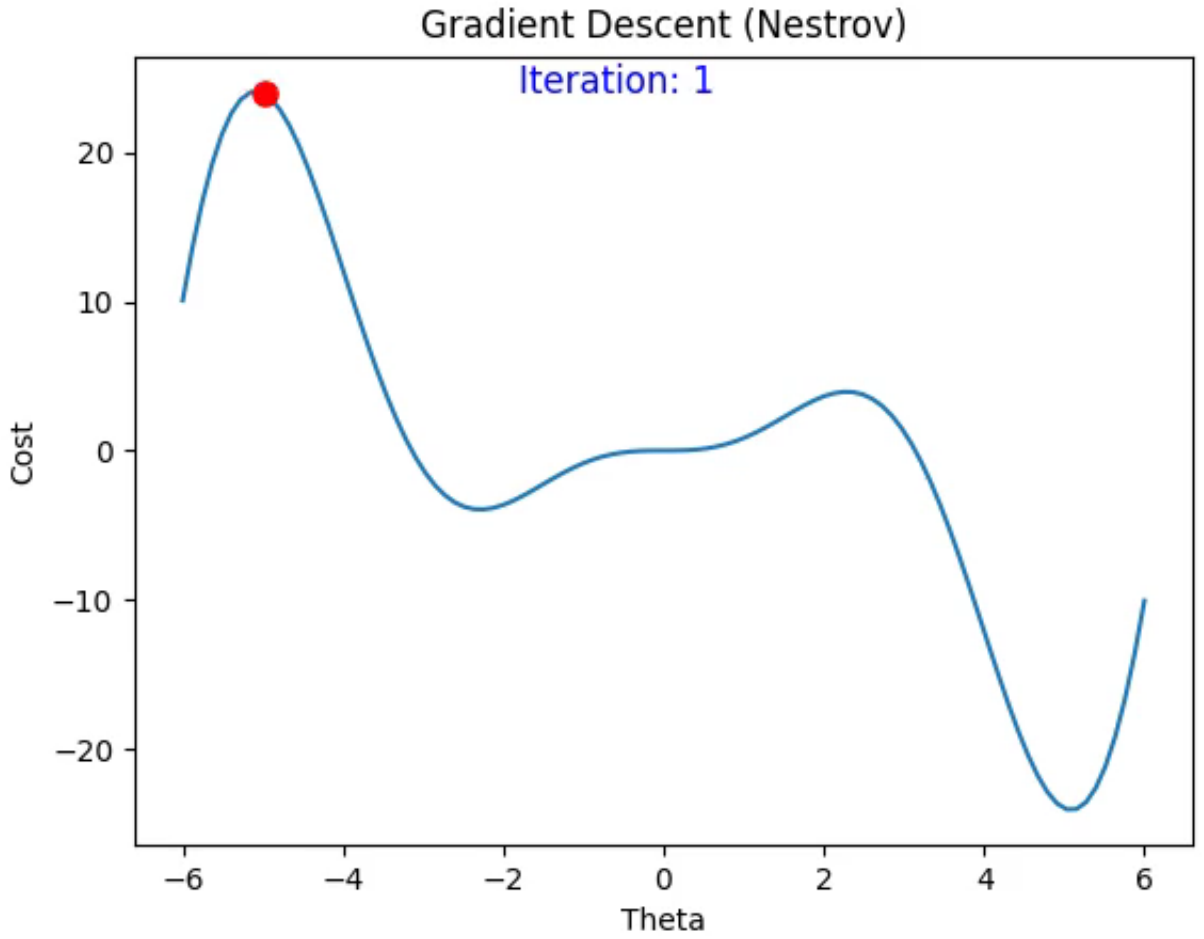


γ – Коэффицент момента

α – скорость обучения

$\nabla_{\boldsymbol{\theta}} J(\theta - \gamma h_t)$ – градиент, вычисленный в предсказанном положении

Стохастический градиент с импульсом Нестерова – NAG



Стохастический градиент с импульсом Нестерова – NAG

Nesterov Accelerated Gradient

$$h_{t+1} = \gamma h_t + \alpha \nabla_{\boldsymbol{\theta}} J(\theta - \gamma h_t)$$

$$\boldsymbol{\theta_{t+1}} = \boldsymbol{\theta_t} - h_{t+1}$$

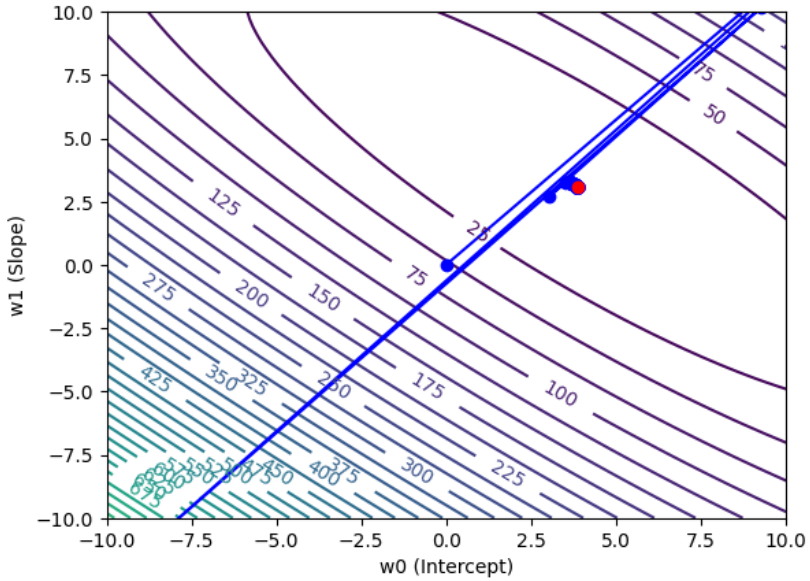
За счёт предвидения, NAG часто сходится быстрее, чем традиционный метод с моментом

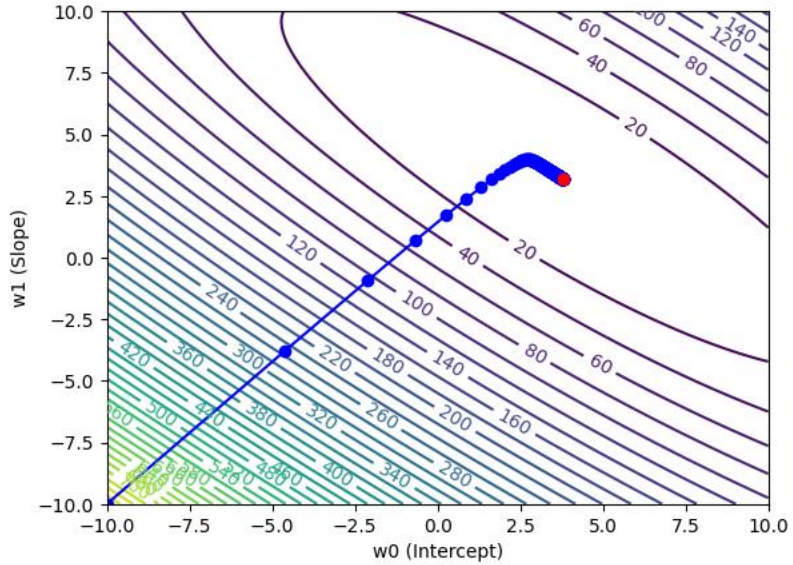
NAG эффективнее справляется с локальными минимумами

Адаптивный подбор размера шага

$$\theta^t = \theta^{t-1} - \alpha \nabla J(\theta^{t-1})$$

$$\alpha_t = \frac{1}{t} \ , \ \alpha_t = \frac{0.1}{t^\beta} \ , \ \ldots$$





AdaGrad (**Adaptive Gradient Algorithm**)

$Dense\ feature$

x_1	x_2
245	0
184	0
300	0
229	0
276	0
302	0
198	0
263	1
317	0
172	0

$Sparse\ feature$

AdaGrad (**Adaptive Gradient Algorithm**)

$$\nabla J = \begin{bmatrix} \frac{\partial J}{\partial \theta_0} \\ \vdots \\ \frac{\partial J}{\partial \theta_d} \end{bmatrix}$$

$$G_{t+1} = G_{t-1} + (\nabla_{\boldsymbol{\theta}} J(\theta_t))^2$$

$$\theta_{t+1} = \theta_{t-1} - \frac{\boldsymbol{\alpha}}{\sqrt{G_t} + \boldsymbol{\epsilon}} \nabla_{\boldsymbol{\theta}} J(\theta_t)$$

$\boldsymbol{\epsilon}$ – Констант для численной стабильности 1e-8

RMSProb (**Root Mean Square Propagation**)

$$\nabla Q = \begin{bmatrix} \frac{\partial Q}{\partial \theta_0} \\ \vdots \\ \frac{\partial Q}{\partial \theta_d} \end{bmatrix}$$

$$G_{t+1} = \beta G_{t-1} + (1-\beta)(\nabla_{\boldsymbol{\theta}} J(\theta_t))^2$$

$$\theta_{t+1} = \theta_{t-1} - \frac{\boldsymbol{\alpha}}{\sqrt{G_t} + \boldsymbol{\epsilon}} \nabla_{\boldsymbol{\theta}} J(\theta_t)$$

вместо суммы использует экспоненциальное скользящее среднее (Moving average)

Adam (Adaptive Momentum)

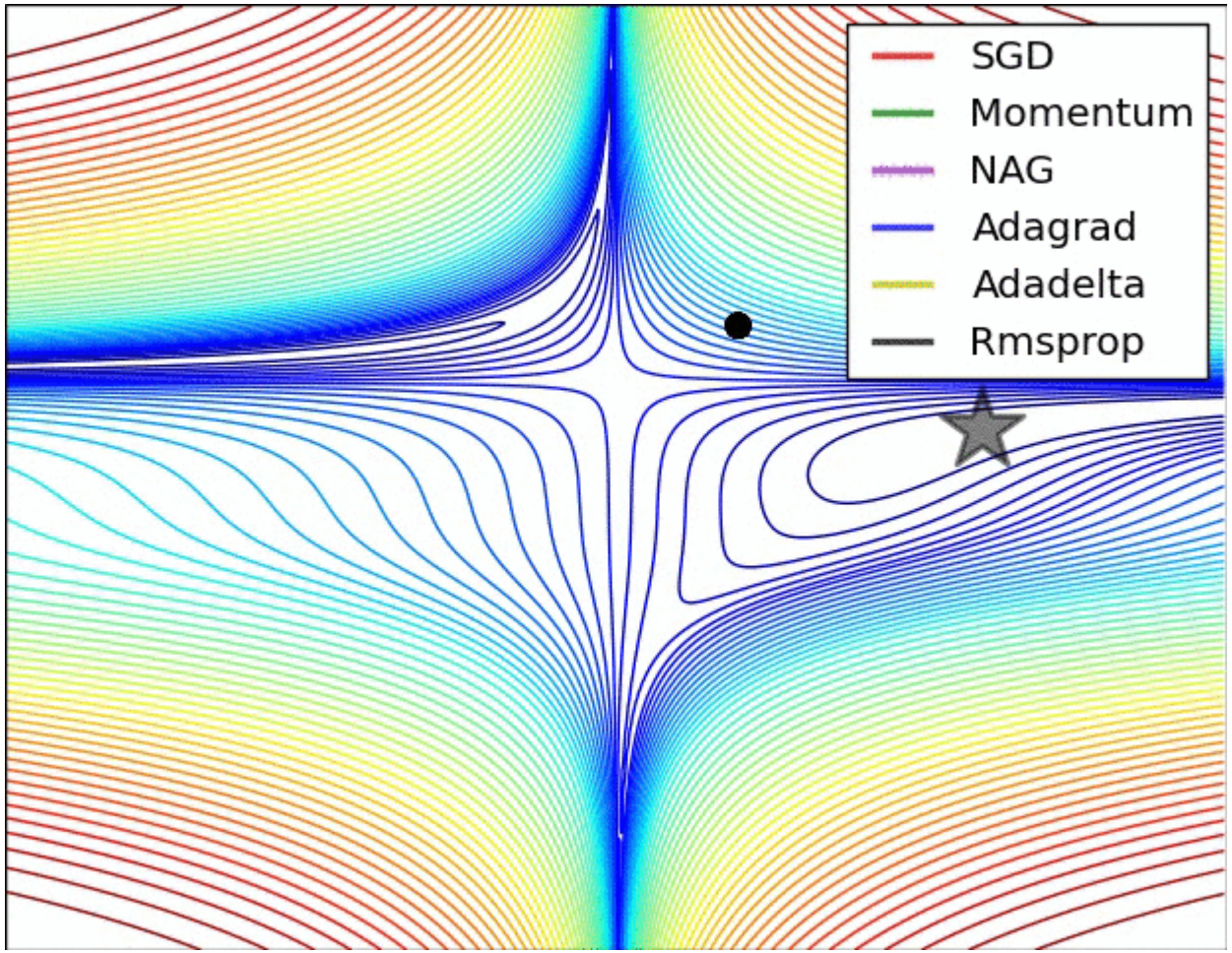
$$h_{t+1} = \gamma h_t + (1-\gamma)\nabla_{\boldsymbol{\theta}} J(\theta)$$

$$G_{t+1} = \beta G_{t-1} + (1-\beta)(\nabla_{\boldsymbol{\theta}} J(\theta_t))^2$$

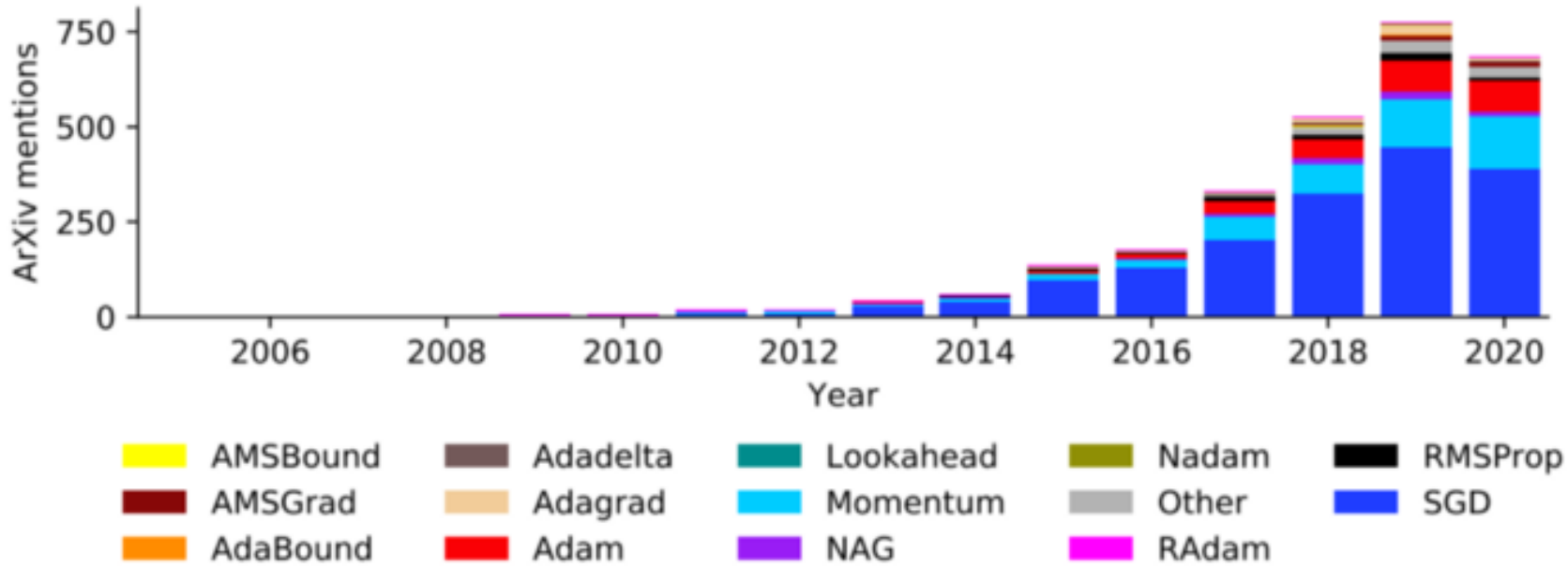
$$\theta_{t+1} = \theta_{t-1} - \frac{\boldsymbol{\alpha}}{\sqrt{G_{t+1}} + \boldsymbol{\epsilon}} h_{t+1}$$

Рекомендации: $\gamma = 0.9, \beta = 0.999$, $\epsilon = 10^{-8}$

Популярные оптимизаторы



Популярные оптимизаторы



Gradient Descent based Optimization Algorithms for Deep Learning Models Training

PyTorch optimizer

Tensorflow optimizer