УНИВЕРСИТЕТ ИТМО

Оптимизаторы

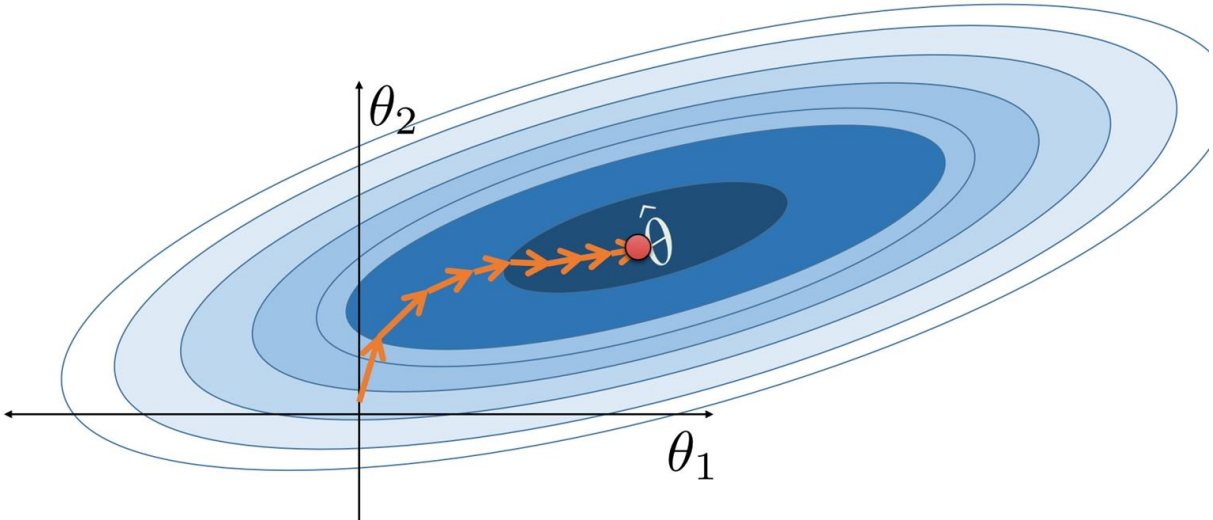
Тетерин Михаил Александрович

ОБРАЗОВАТЕЛЬНЫЕ ПРОГРАММЫ В ОБЛАСТИ
ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Стохастический градиентный спуск

- легко реализуется и приспособлен для динамического обучения, когда объекты поступают потоком
- способен обучаться на больших выборках
- может не сходиться
- может застрять в одном из локальных минимумов
- усреднение по маленьким батчам может приводить к шумным

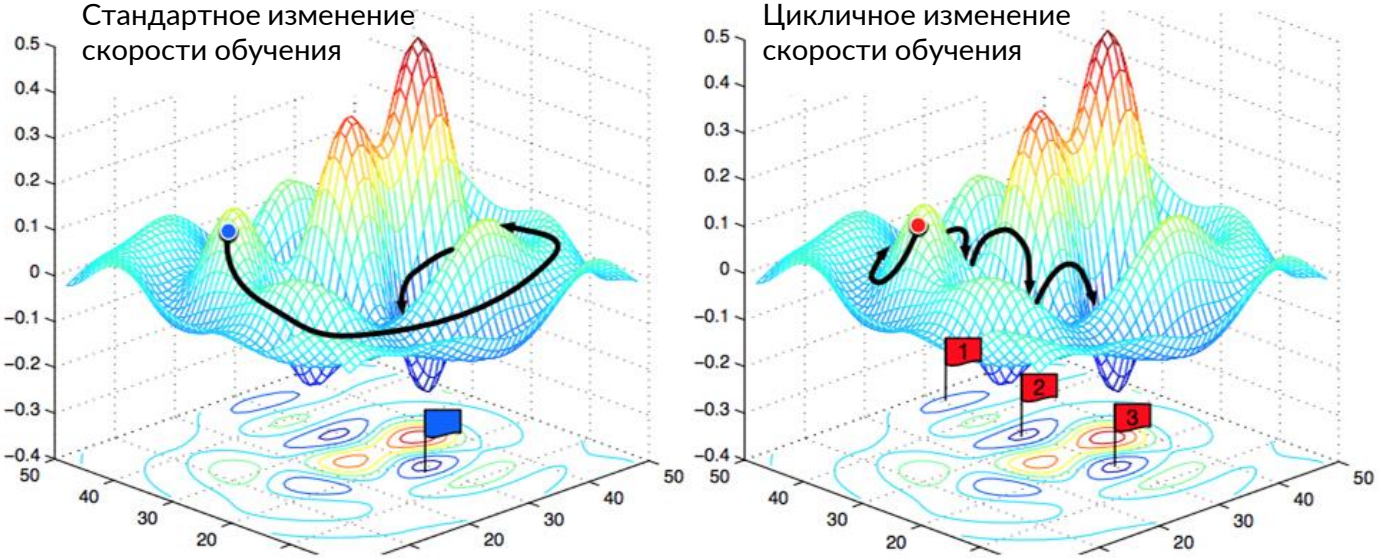
$$x_{t+1} = x_t - \alpha \nabla f(x_t)$$



Стохастический градиентный спуск

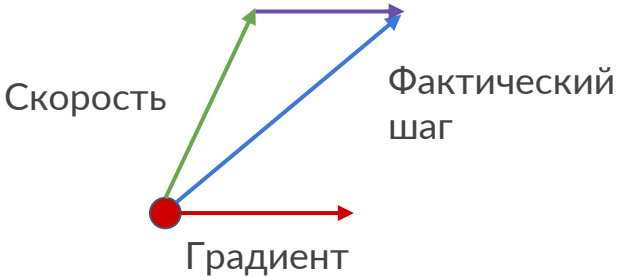
- легко реализуется и приспособлен для динамического обучения, когда объекты поступают потоком
- способен обучаться на больших выборках
- может не сходиться
- может застрять в одном из локальных минимумов
- усреднение по маленьким батчам может приводить к шумным

$$x_{t+1} = x_t - \alpha \nabla f(x_t)$$

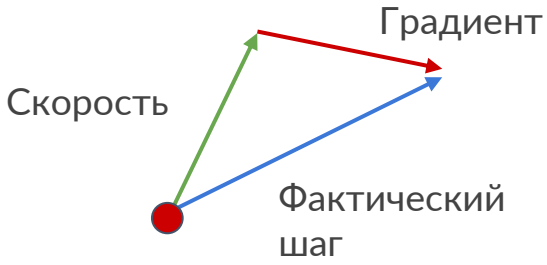


- легко реализуется и приспособлен для динамического обучения, когда объекты поступают потоком
- способен обучаться на больших выборках
- выше вероятность что сойдется
- может не сходиться
- может застрять в одном из локальных минимумов (седловые точки)

Обновление импульса:



Импульс Нестерова:



$$v_{t+1} = \rho v_t + \nabla f(x_t) \qquad v_{t+1} = \rho v_t - \alpha \nabla f(x_t + \rho v_t)$$
$$x_{t+1} = x_t - \alpha v_{t+1} \qquad x_{t+1} = x_t + v_{t+1}$$

Adagrad (Адаптивный градиент)

- стохастический градиент спуск + кеширование
- уменьшаем обновления для элементов, которые мы часто используем
- скорость обучения уменьшается слишком быстро
- глобальную скорость обучения надо подбирать, и она может быть хороша для одних размерностей, но плоха для других

$$cache_{t+1} = cache_t + (\nabla f(x_t))^2$$

$$x_{t+1} = x_t - \alpha \frac{\nabla f(x_t)}{cache_{t+1} + \epsilon}$$

Среднеквадратичное распространение (RMSProp)

- стохастический градиент спуск + кеширование + экспоненциальное сглаживание
- уменьшаем обновления для элементов, которые мы часто используем
- не нужно выбирать скорость обучения
- нет паралича алгоритма
- высокая вероятность преодоления локального оптимума

$$cache_{t+1} = \beta cache_t + (1 - \beta)(\nabla f(x_t))^2$$

$$x_{t+1} = x_t - \alpha \frac{\nabla f(x_t)}{cache_{t+1} + \epsilon}$$

Метод адаптивной оценки моментов (Adam)

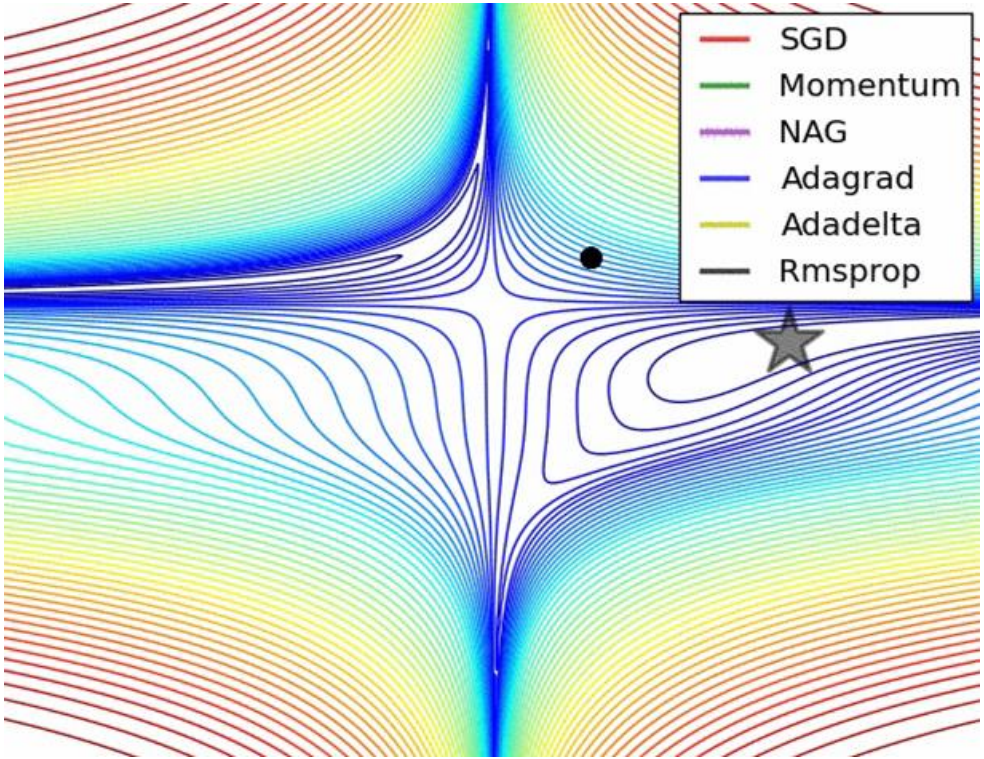
- такие же преимущества как и у RMSProp
- быстрее сходится

$$\nu_{t+1} = \gamma \upsilon_t + (1-\gamma)\nabla f(x_t)$$

$$cache_{t+1} = \beta cache_t + (1-\beta)(\nabla f(x_t))^2$$

$$x_{t+1} = x_t - \alpha \frac{\nu_{t+1}}{cache_{t+1} + \epsilon}$$

Сходимость оптимизаторов



УНИВЕРСИТЕТ ИТМО

Спасибо за внимание

ОБРАЗОВАТЕЛЬНЫЕ ПРОГРАММЫ В ОБЛАСТИ
ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА