Алгоритм SAG (Stochastic Average Gradient)

```
Вход: выборка X^{\ell}, темп обучения h, темп забывания \lambda;
    Выход: вектор весов w;
 1 инициализировать веса w_i, j = 1, ..., n;
 2 инициализировать градиенты: G_i := \nabla \mathscr{L}_i(w), i = 1, ..., \ell;
 3 инициализировать оценку функционала: ar{Q}:=rac{1}{\ell}\sum_{i=1}^{\ell}\mathscr{L}_{i}(w);
   повторять
        выбрать объект x_i из X^\ell случайным образом;
 5
        вычислить потерю: \varepsilon_i := \mathscr{L}_i(w);
 6
        вычислить градиент: G_i := \nabla \mathscr{L}_i(w);
      сделать градиентный шаг: w := w - h \sum_{i=1}^{\ell} G_i;
 8
        оценить функционал: \bar{Q} := (1 - \lambda)\bar{Q} + \lambda \varepsilon_i;
10 пока значение \bar{Q} и/или веса w не сойдутся;
```

Schmidt M., Le Roux N., Bach F. Minimizing finite sums with the stochastic average gradient // arXiv.org, 2013.

9

Резюме

- Непрерывная аппроксимация пороговой функции потерь $[M < 0] \leqslant \mathcal{L}(M)$ позволяет использовать градиентную оптимизацию и повышает качество классификации благодаря увеличению зазора между классами.
- ullet SG легко обобщается для нелинейных моделей g(x,w)
- ullet и для любых функций потерь, $\mathscr{L}_i(w) = \mathscr{L}ig(g(x_i,w),y_iig)$.
- SG допускает онлайновое (потоковое) обучение.
- SG подходит для больших данных, т. к. даёт неплохие решения, даже не обработав всю выборку (x_i, y_i) .