DEEP LEARNING IV: О ГРАДИЕНТНОМ СПУСКЕ

Сергей Николенко

СПбГУ — Санкт-Петербург
22 сентября 2018 г.

Random facts:

- 22 сентября — Всемирный день слонов, Всемирный день без автомобиля и День американских бизнес-леди (American Business Women's Day)
- 22 сентября 1692 г. были повешены последние восемь салемских ведьм, а 22 сентября 1780 г. произошёл первый описанный случай линчевания
- 22 сентября 1862 г. Линкольн освободил рабов, 22 сентября 1944 г. Красная армия освободила Таллин, 22 сентября 1960 г. Мали освободилось от Франции, а 22 сентября 1991 г. впервые провозгласили независимую республику Косово
- 22 сентября 2000 г. Билл Гейтс потерял 22 миллиарда долларов (но всё равно оставался самым богатым человеком мира до 2008-го)

Варианты градиентного спуска

Метод моментов

- «Ванильный» стохастический градиентный спуск:

$$\theta_t = \theta_{t-1} - \eta \nabla E(x_t, \theta_{t-1}, y_t).$$

- Всё зависит от скорости обучения η.
- Первая мысль — пусть η уменьшается со временем:
  - линейно (linear decay):

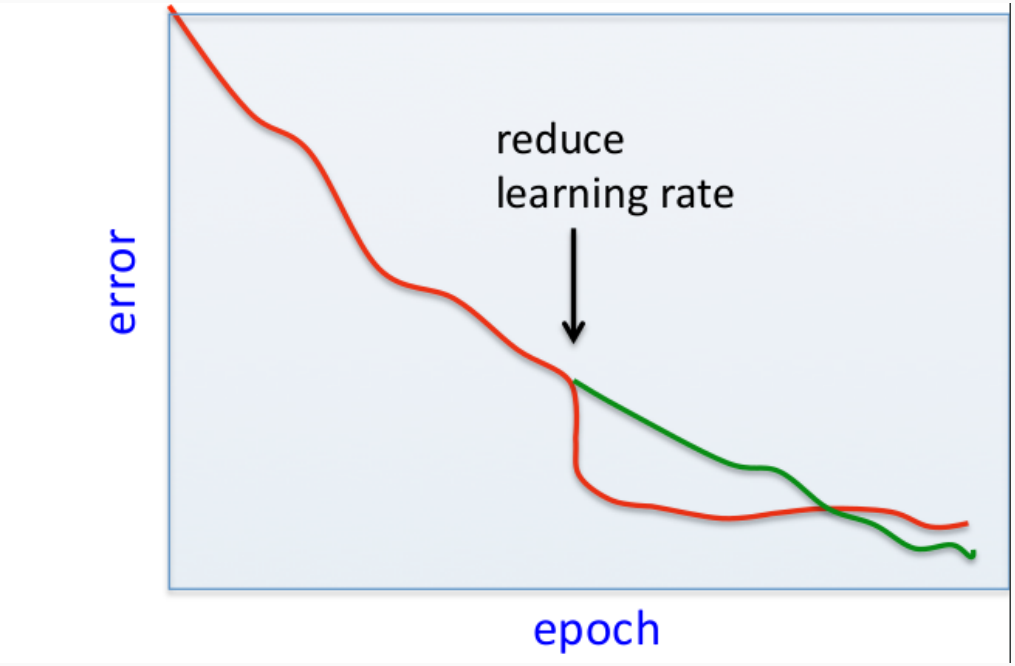
$$\eta = \eta_0 \left(1 - \frac{t}{T}\right);$$

  - или экспоненциально (exponential decay):

$$\eta = \eta_0 e^{-\frac{t}{T}}.$$

Метод моментов

- Скорость обучения лучше не уменьшать слишком быстро.



- Но это в любом случае никак не учитывает собственно E; лучше быть *адаптивным*.

Метод моментов

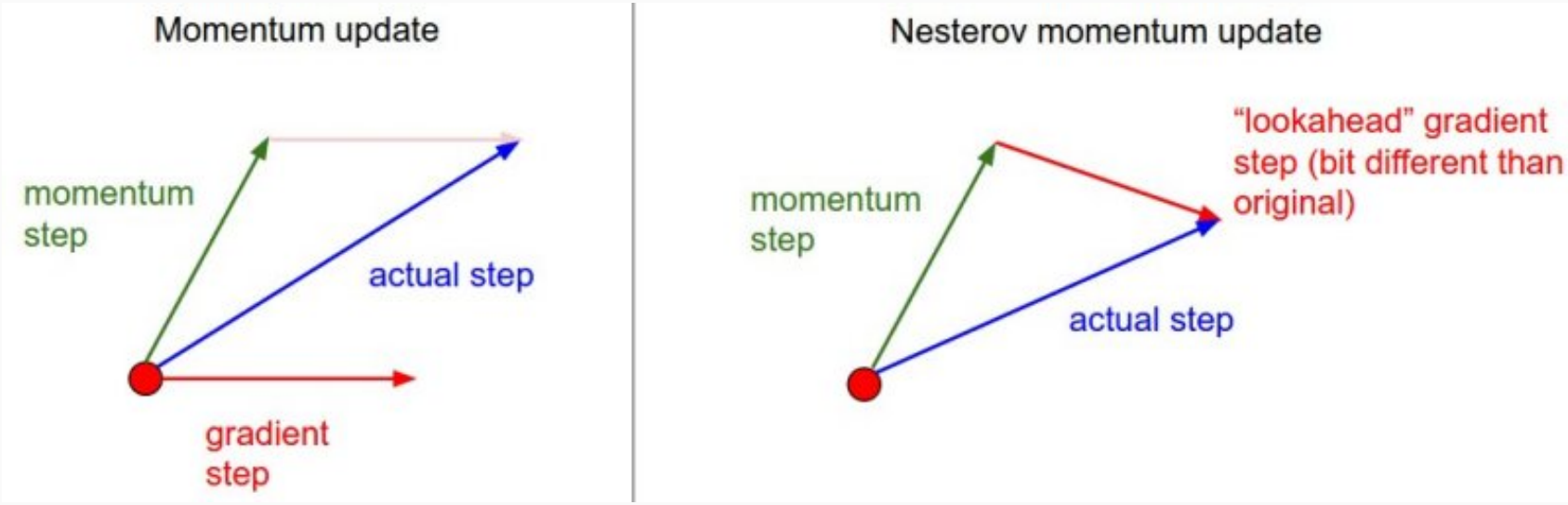
- *Метод моментов* (momentum): сохраним часть скорости, как у материальной точки.
- С инерцией получается

$$u_t = \gamma u_{t-1} + \eta \nabla_\theta E(\theta),$$
$$\theta = \theta - u_t.$$

- И теперь мы сохраняем γu_{t-1}.

Метод моментов

- Но на самом деле мы уже знаем, что попадём в γu_{t-1} на промежуточном шаге.
- Давайте прямо там, на полпути, и вычислим градиент!



Метод моментов

- *Метод Нестерова* (Nesterov's momentum):

$$u_t = \gamma u_{t-1} + \eta \nabla_\theta E(\theta - \gamma u_{t-1})$$

- Можно ли ещё лучше?..

Метод моментов

- ...ну, можно попробовать методы второго порядка.
- Метод Ньютона:

$$E(\theta) \approx E(\theta_0) + \nabla_\theta E(\theta_0)(\theta - \theta_0) + \frac{1}{2}(\theta - \theta_0)^\top H(E(\theta))(\theta - \theta_0).$$

- Когда работает, это обычно гораздо быстрее, и нет никакой η, ничего настраивать не надо.
- Но нужно считать гессиан $H(E(\theta))$, и это нереально.

Метод моментов

- Есть, правда, приближения.
- L-BFGS (limited memory Broyden–Fletcher–Goldfarb–Shanno):
  - строим аппроксимацию к H^{-1};
  - для этого сохраняем последовательно апдейты аргументов функции и градиентов и выражаем через них H^{-1}.
- Интересный открытый вопрос: можно ли заставить L-BFGS работать для deep learning?
- Но пока не получается («в лоб» было бы нужно считать градиент по всему датасету, а по мини-батчам непонятно как).

Адаптивные методы градиентного спуска

- Но дальше улучшить всё равно можно.
- Заметим, что до сих пор скорость обучения была одна во всех направлениях.
- Идея: давайте быстрее двигаться по тем параметрам, которые не сильно меняются, и медленнее по быстро меняющимся параметрам.

Адаптивные методы градиентного спуска

- *Adagrad*: давайте накапливать историю этой скорости изменений и учитывать её.
- Обозначая $g_{t,i} = \nabla_{\theta_i} L(\theta)$, получим

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i},$$

  где G_t – диагональная матрица с $G_{t,ii} = G_{t-1,ii} + g_{t,i}^2$, которая накапливает общее значение градиента по всей истории обучения.
- Так что скорость обучения всё время уменьшается, но с разной скоростью для разных θ_i.

Адаптивные методы градиентного спуска

- Проблема: G всё увеличивается и увеличивается, и скорость обучения иногда уменьшается слишком быстро.
- *Adadelta* (Zeiler, 2012) – та же идея, но две новых модификации.
- Во-первых, историю градиентов мы теперь считаем с затуханием:

$$G_{t,ii} = \rho G_{t-1,ii} + (1-\rho) g_{t,i}^2.$$

- А всё остальное здесь точно так же:

$$u_t = -\frac{\eta}{\sqrt{G_{t-1} + \epsilon}} g_{t-1}.$$

Адаптивные методы градиентного спуска

- Во-вторых, надо бы «единицы измерения» привести в соответствие.
- В предыдущих методах была проблема:
  - в обычном градиентном спуске или методе моментов «единицы измерения» обновления параметров $\Delta\theta$ — это единицы измерения градиента, т.е. если веса в секундах, а целевая функция в метрах, то градиент будет иметь размерность «метр в секунду», и мы вычитаем метры в секунду из секунд;
  - а в Adagrad получалось, что значения обновлений $\Delta\theta$ зависели от отношений градиентов, и величина обновлений вовсе безразмерная.

Адаптивные методы градиентного спуска

- Эта проблема решается в методе второго порядка: обновление параметров $\Delta\theta$ пропорционально $H^{-1}\nabla_\theta f$, то есть размерность будет

$$\Delta\theta \propto H^{-1}\nabla_\theta f \propto \frac{\frac{\partial f}{\partial \theta}}{\frac{\partial^2 f}{\partial \theta^2}} \propto \text{ размерность } \theta.$$

- Чтобы привести *Adadelta* в соответствие, нужно домножить на ещё одно экспоненциальное среднее, но теперь уже от квадратов обновлений параметров, а не от градиента.
- Настоящее среднее мы не знаем, аппроксимируем предыдущими шагами:

$$\mathbb{E}\left[\Delta\theta^2\right]_t = \rho\mathbb{E}\left[\Delta\theta^2\right]_{t-1} + (1-\rho)\Delta\theta^2, \text{ где}$$

$$u_t = -\frac{\sqrt{\mathbb{E}\left[\Delta\theta^2\right]_{t-1} + \epsilon}}{\sqrt{G_{t-1} + \epsilon}} \cdot g_{t-1}.$$

Адаптивные методы градиентного спуска

- Следующий вариант – *RMSprop* из курса Хинтона.
- Практически то же, что *Adadelta*, только *RMSprop* не делает вторую поправку с изменением единиц и хранением истории самих обновлений, а просто использует корень из среднего от квадратов (вот он где, RMS) от градиентов:

$$u_t = -\frac{\eta}{\sqrt{G_{t-1} + \epsilon}} \cdot g_{t-1}.$$

Адаптивные методы градиентного спуска

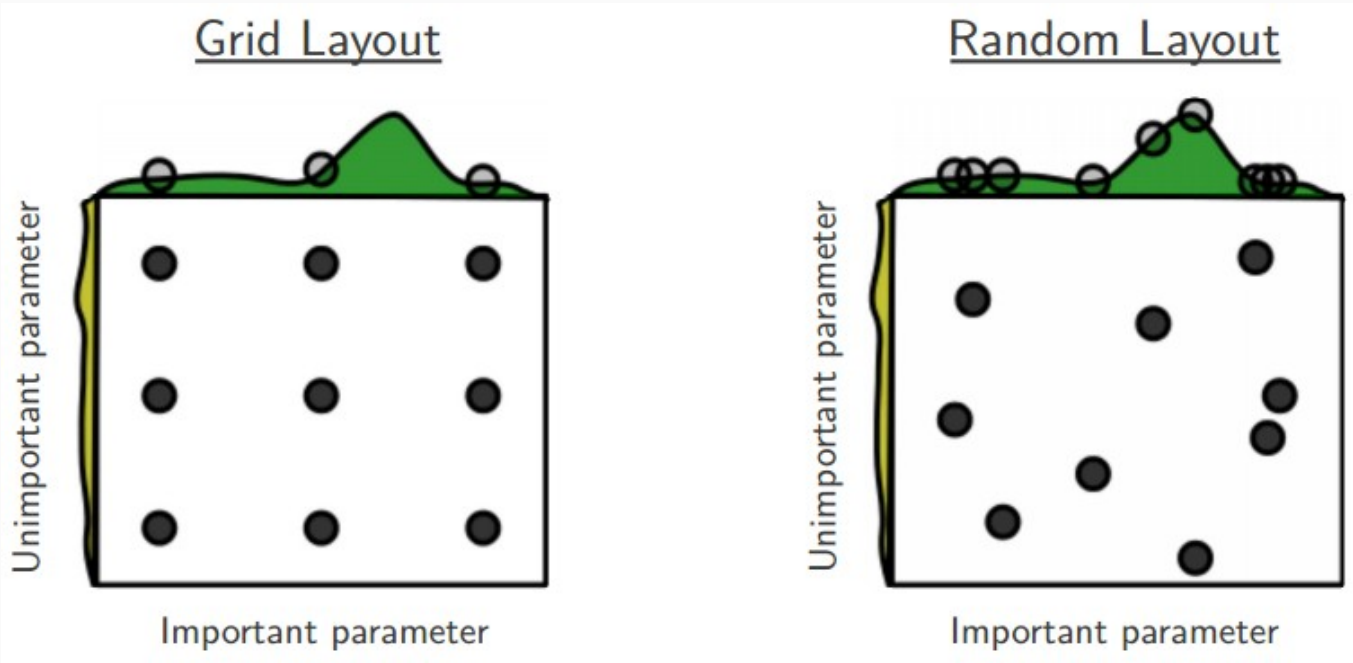
- И последний алгоритм – *Adam* (Kingma, Ba, 2014).
- Модификация *Adagrad* со сглаженными версиями среднего и среднеквадратичного градиентов:

$$
\begin{aligned}
m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t, \\
v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \\
u_t &= \frac{\eta}{\sqrt{v + \epsilon}} m_t.
\end{aligned}
$$

- (Kingma, Ba, 2014) рекомендуют $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$.
- *Adam* практически не требует настройки, используется на практике очень часто.
- ...[анимированные примеры]...

Практические замечания

- Ещё практические замечания об оптимизации гиперпараметров:
  - одного валидационного множества достаточно;
  - лучше гиперпараметры искать на логарифмической шкале;
  - и лучше случайным поиском, а не по сетке (Bergstra and Bengio).



Спасибо!

Спасибо за внимание!