14.09.18 dl II

Методы оптимизации и регуляризации нейросетей

$$F(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x) \to \min_x, \quad n \gg 1$$

Величина	Стоимость
$f_i(x)$	$O(s)$
$\nabla f_i(x)$	$O(s)$
$F(x)$	$O(ns)$
$\nabla F(x)$	$O(ns)$

Необходимо выбирать методы оптимизации, не зависящие от числа слагаемых n.

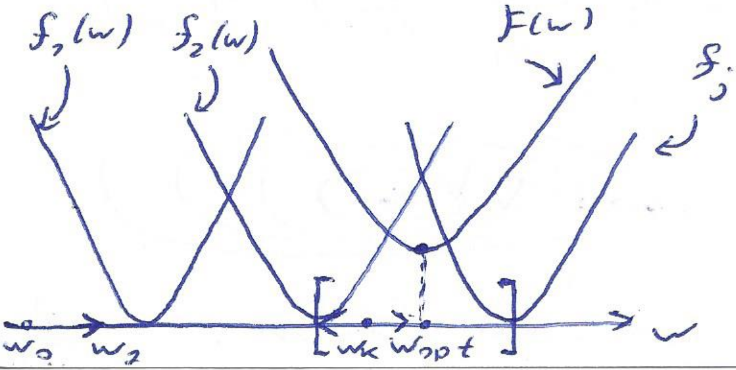
SGD: $\begin{cases} i_k \sim Unif(1 \ldots n) \\ g_k = \nabla f_{i_k}(x_k) \\ x_{k+1} = x_k - \alpha_k g_k \end{cases}$

$$\mathbb{E}_{i_k \sim U} g_k = \sum_{i=1}^{n} \frac{1}{n} \nabla f_i(x_k) = \nabla F(x_k)$$

SGD + mini-batches: $\begin{cases} I_k \sim Unif(1 \ldots n) \\ g_k = \frac{1}{|I_k|} \sum_{i \in I_k} \nabla f_i(x_k) \\ x_{k+1} = x_k - \alpha_k g_k \end{cases}$

Одномерная LR

$$F(w) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{(y_i - w x_i)^2}_{f_i(w)} \to \min_w$$

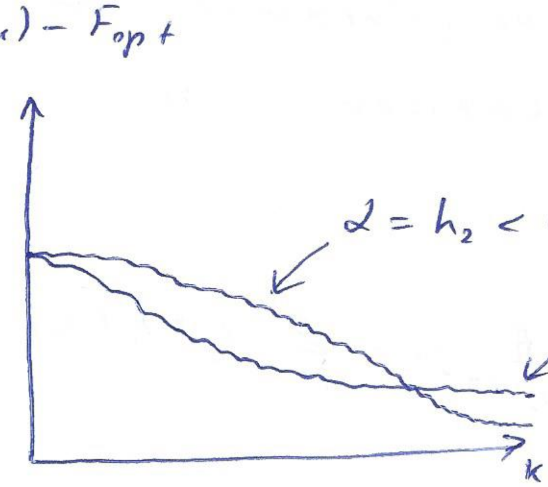


$\underline{\text{III}}$ F – выпуклая, $\in C^1$ $\Rightarrow$ для SGD верно

$$\mathbb{E} F(x_k) - F_{opt} \le \frac{\overbrace{\|x_0 - x_{opt}\|^2}^{\le R^2} + \sum_{i=0}^{k} \alpha_i^2 \overbrace{\mathbb{E}\|g_i\|^2}^{\le G^2}}{2\left(\sum_{i=0}^{k} \alpha_i\right)} \le \frac{R^2 + G^2 \sum_{i=0}^{k} \alpha_i^2}{2\left(\sum_{i=0}^{k} \alpha_i\right)}$$

$\underline{\alpha_i = h}$
$$\frac{R^2 + G^2 h^2 (k+1)}{2h(k+1)} = \frac{R^2}{2h(k+1)} + \frac{G^2 h}{2} \xrightarrow[k\to\infty]{} \frac{G^2 h}{2}$$

область блуждания метода пропорциональна шагу итерации и дисперсии стохастического град.



Достаточное условие сходимости

$$\begin{cases} \sum_i \alpha_i = \infty \\ \sum_i \alpha_i^2 < \infty \;\Big|\; \dfrac{\sum_i \alpha_i^2}{\sum_i \alpha_i} \xrightarrow[i\to\infty]{} 0 \end{cases}$$

$$\alpha_i = \frac{h}{(i+1)^\tau}, \quad \tau \in \left(\tfrac{1}{2}, 1\right]$$

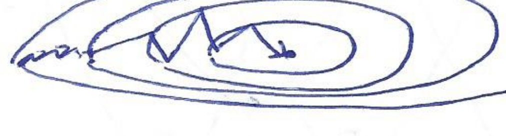
$\tau = 1$; сх-ть $\sim O\left(\frac{1}{\ln k}\right)$, $\frac{1}{\ln k} = \varepsilon$, $k = \exp(\varepsilon^{-1})$

очень медленно!

Для схем $\left(\dfrac{\sum_i \alpha_i^2}{\sum_i \alpha_i} \to 0\right)$ можно брать $\tau \in \left(0, \tfrac{1}{2}\right]$

Оптимальное $\tau_{opt} = \frac{1}{2}$ и сх-ть $\sim O\left(\frac{\ln k}{\sqrt{k}}\right) = \tilde{O}\left(\frac{1}{\sqrt{k}}\right)$

GD плохо справляется с вытянутыми ф-циями, что особенно важно для нейросетевых ф-ций с широкими плато

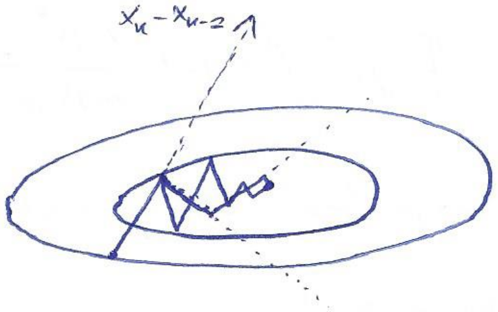


GD: $x_{k+1} = x_k - \alpha_k \nabla F(x_k)$

Newton: $x_{k+1} = x_k - \alpha_k \left[\nabla^2 F(x_k)\right]^{-1} \nabla F(x_k)$

SGD + momentum

$$x_{k+1} = x_k - \alpha_k g_k + \beta_k (x_k - x_{k-1}), \quad \beta_k \in (0, 1)$$



экспоненциальное сглаживание на историю шагов, инерция

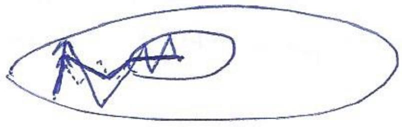
$-g_k \uparrow\uparrow (x_k - x_{k-1})$

Ada Grad

$$\begin{cases} x_{k+1,i} = x_{k,i} - \alpha_k \dfrac{g_{k,i}}{\sqrt{v_{k,i} + \varepsilon}} \\ v_{k,i} = \sum\limits_{j=0}^{k} g_{j,i}^2 \end{cases}$$

уменьшает шаг для больших град., увеличивает для мал.

шкалирование на квадратах



RMSprop

$$\begin{cases} x_{k+1,i} = x_{k,i} - \alpha_k \dfrac{g_{k,i}}{\sqrt{v_{k,i} + \varepsilon}} \\ v_{k,i} = \beta_k v_{k-1,i} + (1 - \beta_k) g_{k,i}^2 \end{cases}$$

ADAM

$$\begin{cases} x_{k+1,i} = x_{k,i} - \alpha_k \dfrac{\mu_{k,i}}{\sqrt{v_{k,i} + \varepsilon}} \\ \mu_{k,i} = \gamma_k \mu_{k-1,i} + (1 - \gamma_k) g_{k,i} \\ v_{k,i} = \beta_k v_{k-1,i} + (1 - \beta_k) g_{k,i}^2 \end{cases}$$

$v_{0,i} = 0$

$v_{1,i} = (1-\beta) g_{1,i}^2$

$v_{2,i} = \beta(1-\beta) g_{1,i}^2 + (1-\beta) g_{2,i}^2$

...

$v_{k,i} = \cancel{\beta}\sum_{j=1}^{k} (1-\beta)\beta^{k-j} g_{j,i}^2 \qquad \mathbb{E} g_{0,i}^2$

$$\mathbb{E} v_{k,i} = \sum_{j=1}^{k} (1-\beta)\beta^{k-j} \overbrace{\mathbb{E} g_{j,i}^2} \approx \mathbb{E} g_{0,i}^2 \cancel{(1-\beta)} \frac{1-\beta^k}{\cancel{1-\beta}} = (1-\beta^k)\mathbb{E} g_{0,i}^2$$

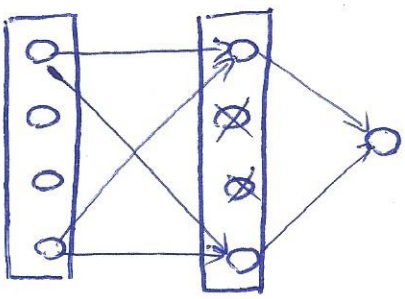
возникает смещение на первых итерациях метода, необходима коррекция!

$$\hat{\mu}_{k,i} = \frac{\mu_{k,i}}{1-\gamma^k}, \qquad \hat{v}_{k,i} = \frac{v_{k,i}}{1-\beta^k}$$

Регуляризация

Обычная l_1, l_2 не подходит, поскольку нормы градиента различны на разных уровнях сети.

DropOut

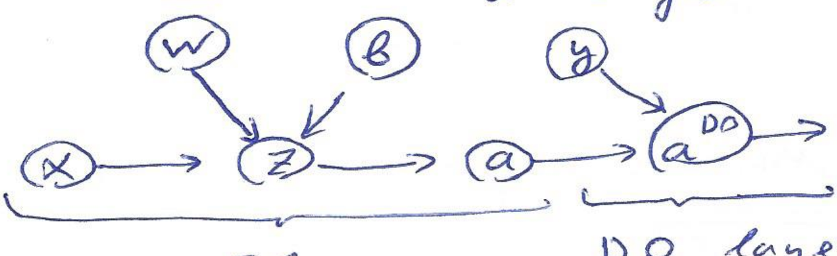


$z = Wx + b$

$a = g(z)$

$y_i \sim Bern(p): \begin{cases} 0 & 1 \\ p & 1-p \end{cases}$

$a^{DO} = y \odot a$



новый слой

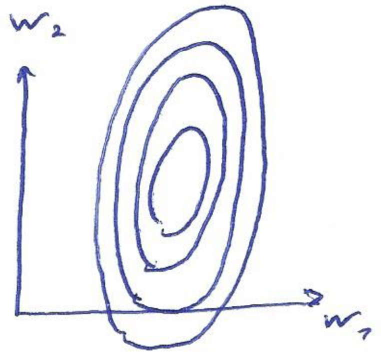
бернуллевский шум

разрушаем коадаптацию весов на внутренних слоях сети

$$\mathbb{E}_{y\sim Bern}\, a^{DO} = (1-p)a$$

необходимо корректировать выход сети на тесте, иначе $a^{DO} = \frac{1}{1-p}\, y \circ a \Rightarrow \mathbb{E}_{y\sim Bern}\, a^{DO} = a$

Регулирование нормы градиента на внутренних слоях сети



Batch Normalization

$$\{x_{ij}\}_{i=1}^{N_{batch}} \to \boxed{BN} \to \{y_{ij}\}_{i=1}^{N_{batch}}$$

$$\{\gamma_j, \beta_j\}$$

$$\mu_j = \sum_{i=1}^{N_{batch}} x_{ij}, \quad \sigma_j^2 = \frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} (x_{ij}-\mu_j)^2$$

$$y_{ij} = \gamma_j \frac{x_{ij}-\mu_j}{\sigma_j + \varepsilon} + \beta_j$$

вставляют после линейности и до нелинейности

для теста делается экспоненциальное сглаживание для μ и σ по всем мини-батчам, и эти посчитанные $\hat{\mu}$ и $\hat{\sigma}$ используются для прогноза, иначе можно посчитать $\hat{\mu}$, $\hat{\sigma}$ по всей выборке за дополнительную эпоху после обучения.