Современные методы оптимизации второго порядка

Руднев Виктор

МГУ имени М.В. Ломоносова, факультет ВМК, кафедра ММП

24 мая 2018 г.

Мотивация

SGD и его модификации

- сходятся достаточно медленно,
- имеют много гиперпараметров,
- не учитывают структуры того, что оптимизируется.

Постановка задачи

Поставлена задача регрессии с выборкой (X, Y)

Постановка задачи

Поставлена задача регрессии с выборкой (X, Y)

$$p(y_i|x_i, \theta) = \mathcal{N}(y_i|f(x_i, \theta), I)$$

Постановка задачи

Поставлена задача регрессии с выборкой (X, Y)

$$p(y_i|x_i, \theta) = \mathcal{N}(y_i|f(x_i, \theta), I)$$

$$p(Y|X, \theta) = \prod_{i=1}^{N} \mathcal{N}(y_i|f(x_i, \theta), I)$$

Постановка задачи

Поставлена задача регрессии с выборкой (X, Y)

$$p(y_i|x_i,\theta) = \mathcal{N}(y_i|f(x_i,\theta), I)$$

$$p(Y|X,\theta) = \prod_{i=1}^{N} \mathcal{N}(y_i|f(x_i,\theta), I)$$

$$L(\theta) = -\log p(Y|X,\theta) = -\sum_{i=1}^{N} \log \mathcal{N}(y_i|f(x_i,\theta), I)$$

Постановка задачи

Поставлена задача регрессии с выборкой (X, Y)

$$p(y_i|x_i, \theta) = \mathcal{N}(y_i|f(x_i, \theta), I)$$

$$p(Y|X, \theta) = \prod_{i=1}^{N} \mathcal{N}(y_i|f(x_i, \theta), I)$$

$$L(\theta) = -\log p(Y|X, \theta) = -\sum_{i=1}^{N} \log \mathcal{N}(y_i|f(x_i, \theta), I)$$

$$L(\theta) = \sum_{i=1}^{N} \frac{1}{2} ||y_i - f(x_i, \theta)||^2 \to \min_{\theta}$$

Метод Ньютона

Разложим $L(\theta)$ до второй степени:

Метод Ньютона

Разложим $L(\theta)$ до второй степени:

$$L(\theta + h) = L(\theta) + (\nabla_\theta L(\theta))^T h + \frac{1}{2} h^T H(\theta) h + o(||h||^2),$$

$$\text{где } H(\theta) = \frac{\partial^2}{\partial\theta\partial\theta} L(\theta) \text{ – гессиан.}$$

Метод Ньютона

Разложим $L(\theta)$ до второй степени:

$$L(\theta + h) = L(\theta) + (\nabla_\theta L(\theta))^T h + \frac{1}{2} h^T H(\theta) h + o(||h||^2),$$

где $H(\theta) = \dfrac{\partial^2}{\partial\theta\partial\theta} L(\theta)$ – гессиан.

$$\arg\min_h \hat{L}(\theta + h) = -H(\theta)^{-1} \nabla_\theta L(\theta), \text{ если } H(\theta) \succ 0$$

Метод Ньютона

Разложим $L(\theta)$ до второй степени:

$$L(\theta + h) = L(\theta) + (\nabla_\theta L(\theta))^T h + \frac{1}{2} h^T H(\theta) h + o(||h||^2),$$

$$\text{где } H(\theta) = \frac{\partial^2}{\partial\theta\partial\theta} L(\theta) \text{ – гессиан.}$$

$$\arg\min_h \hat{L}(\theta + h) = -H(\theta)^{-1} \nabla_\theta L(\theta), \text{ если } H(\theta) \succ 0$$

Тогда параметры лучше всего обновить так:

$$\theta_{t+1} = \theta_t - H(\theta)^{-1} \nabla_\theta L(\theta)$$

Метод Ньютона

- сходится очень быстро по шагам, медленно по времени,
- не имеет гиперпараметров,
- не работает, если гессиан не положительно определен,
- требует очень много памяти, так как нужно хранить гессиан,
- не работает с минибатчами.

Метод Гаусса-Ньютона

Распишем гессиан $L(\theta)$:

Метод Гаусса-Ньютона

Распишем гессиан $L(\theta)$:

$$H(\theta)_{kl} = \frac{\partial^2}{\partial\theta_k\partial\theta_l}L(\theta) = \frac{\partial^2}{\partial_k\theta\partial\theta_l}\frac{1}{2}\sum_{i=1}^{N}||y_i - f(x_i,\theta)||^2 =$$

Метод Гаусса-Ньютона

Распишем гессиан $L(\theta)$:

$$H(\theta)_{kl} = \frac{\partial^2}{\partial\theta_k \partial\theta_l} L(\theta) = \frac{\partial^2}{\partial_k \theta \partial\theta_l} \frac{1}{2} \sum_{i=1}^{N} ||y_i - f(x_i, \theta)||^2 =$$

$$\sum_{i=1}^{N} \left\langle \frac{\partial f(x_i, \theta)}{\partial\theta_k}, \frac{\partial f(x_i, \theta)}{\partial\theta_l} \right\rangle + \left\langle f(x_i, \theta) - y_i, \frac{\partial^2 f(x_i, \theta)}{\partial\theta_k \partial\theta_l} \right\rangle$$

Распишем гессиан $L(\theta)$:

$$H(\theta)_{kl} = \frac{\partial^2}{\partial\theta_k \partial\theta_l} L(\theta) = \frac{\partial^2}{\partial_k\theta\partial\theta_l} \frac{1}{2} \sum_{i=1}^{N} ||y_i - f(x_i, \theta)||^2 =$$

$$\sum_{i=1}^{N} \left\langle \frac{\partial f(x_i, \theta)}{\partial\theta_k}, \frac{\partial f(x_i, \theta)}{\partial\theta_l} \right\rangle + \left\langle f(x_i, \theta) - y_i, \frac{\partial^2 f(x_i, \theta)}{\partial\theta_k \partial\theta_l} \right\rangle$$

Без второй части — матрица Гаусса-Ньютона:

$$G(\theta) = \sum_{i=1}^{N} \left(\frac{\partial f(x_i, \theta)}{\partial\theta} \right) \left(\frac{\partial f(x_i, \theta)}{\partial\theta} \right)^T \succeq 0$$

Метод Гаусса-Ньютона

Теперь $L(\theta)$ можно расписать так:

Метод Гаусса-Ньютона

Теперь $L(\theta)$ можно расписать так:

$$L(\theta + h) \approx \hat{L}(\theta_h) = L(\theta) + (\nabla_\theta L(\theta))^T h + \frac{1}{2} h^T (G(\theta) + \alpha I) h,$$

где α — коэффициент регуляризации

Метод Гаусса-Ньютона

Теперь $L(\theta)$ можно расписать так:

$$L(\theta + h) \approx \hat{L}(\theta_h) = L(\theta) + (\nabla_\theta L(\theta))^T h + \frac{1}{2} h^T (G(\theta) + \alpha I) h,$$

где α — коэффициент регуляризации

$$\arg\min_h \hat{L}(\theta + h) = -(G(\theta) + \alpha I)^{-1} \nabla_\theta L(\theta)$$

Метод Гаусса-Ньютона

Теперь $L(\theta)$ можно расписать так:

$$L(\theta + h) \approx \hat{L}(\theta_h) = L(\theta) + (\nabla_\theta L(\theta))^T h + \frac{1}{2} h^T (G(\theta) + \alpha I) h,$$

где α — коэффициент регуляризации

$$\arg\min_h \hat{L}(\theta + h) = -(G(\theta) + \alpha I)^{-1} \nabla_\theta L(\theta)$$

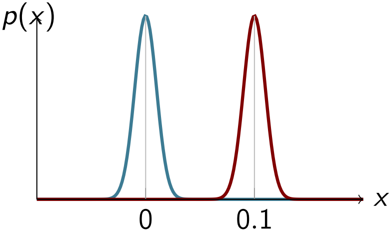
Тогда параметры лучше всего обновить так:

$$\theta_{t+1} = \theta_t - (G(\theta) + \alpha I)^{-1} \nabla_\theta L(\theta)$$

Метод Гаусса-Ньютона

- сходится очень быстро по шагам, медленно по времени,
- почти не имеет гиперпараметров,
- всё ещё требует очень много памяти, так как нужно хранить матрицу G,
- не работает с минибатчами.

Проблемы с обычной оптимизацией



Две нормальных распределений – пара узких, пара широких. Если сдвинуть широкое на 1, то оно особо не изменится, но если сдвинуть узкое даже на 0.1, то оно будет совершенно другим.

Матрица Фишера

Возьмем расстояние $\rho(\theta, \theta + h) =$

$$\frac{1}{2}\Big(\mathrm{KL}\big[p(y|x,\theta)||p(y|x,\theta+h)\big] + \mathrm{KL}\big[p(y|x,\theta+h)||p(y|x,\theta)\big]\Big) =$$

Матрица Фишера

Возьмем расстояние $\rho(\theta, \theta + h) =$

$$\frac{1}{2}\Big(\mathrm{KL}\big[p(y|x,\theta)||p(y|x,\theta+h)\big] + \mathrm{KL}\big[p(y|x,\theta+h)||p(y|x,\theta)\big]\Big) =$$

$\frac{1}{2}h^T F(\theta)h + o(||h||^2)$, где

$$F(\theta) = \mathbb{E}_{p(y|x,\theta)}(\nabla_\theta \log p(y|x,\theta))(\nabla_\theta \log p(y|x,\theta))^T \text{ —}$$

матрица Фишера

Натуральный градиент

Мы знаем, что с обычным евклидовым расстоянием направление наискорейшего спуска есть просто градиент:

$$\arg \min_{||h||_{\mathrm{normal}}=1} L(\theta + h) = -\frac{\nabla_\theta L(\theta)}{\sqrt{(\nabla_\theta L(\theta))^T \nabla_\theta L(\theta)}}$$

Натуральный градиент

Мы знаем, что с обычным евклидовым расстоянием направление наискорейшего спуска есть просто градиент:

$$\arg \min_{||h||_{\mathrm{normal}}=1} L(\theta + h) = -\frac{\nabla_\theta L(\theta)}{\sqrt{(\nabla_\theta L(\theta))^T \nabla_\theta L(\theta)}}$$

С расстоянием, заданным матрицей Фишера, направление наискорейшего спуска чуть отличается:

$$\arg \min_{||h||_{\mathrm{natural}}=1} L(\theta + h) = -\frac{F(\theta)^{-1} \nabla_\theta L(\theta)}{\sqrt{(\nabla_\theta L(\theta))^T F(\theta)^{-1} \nabla_\theta L(\theta)}}$$

Натуральный градиент

Мы знаем, что с обычным евклидовым расстоянием направление наискорейшего спуска есть просто градиент:

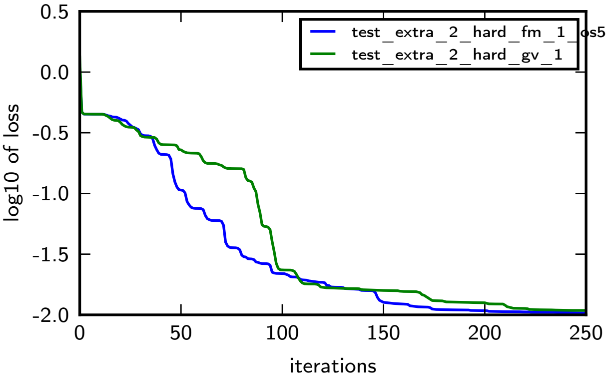
$$\arg \min_{||h||_{\mathrm{normal}}=1} L(\theta + h) = -\frac{\nabla_\theta L(\theta)}{\sqrt{(\nabla_\theta L(\theta))^T \nabla_\theta L(\theta)}}$$

С расстоянием, заданным матрицей Фишера, направление наискорейшего спуска чуть отличается:

$$\arg \min_{||h||_{\mathrm{natural}}=1} L(\theta + h) = -\frac{F(\theta)^{-1} \nabla_\theta L(\theta)}{\sqrt{(\nabla_\theta L(\theta))^T F(\theta)^{-1} \nabla_\theta L(\theta)}}$$

$\hat{g}(\theta) = (F(\theta_t) + \alpha I)^{-1} \nabla_{\theta t} L(\theta_t)$ назовем натуральным градиентом, а метод $\theta_{t+1} = \theta_t - \hat{g}(\theta_t)$ назовем методом натурального градиента.

Натуральный градиент



Сравнение оптимизаторов на автокодировщике с полносвязной архитектурой 64-15-15-15-15-10 для DIGITS (1000 рукописных цифр размера 8x8) (синим — натуральный градиент, зелёным Гаусс-Ньютон)

Метод натурального градиента

- сходится очень быстро по шагам, медленно по времени,
- почти не имеет гиперпараметров,
- всё ещё требует очень много памяти, так как нужно хранить матрицу F,
- всё ещё не работает с минибатчами,
- для нейросетей есть приближения, которые не требуют памяти, не хранят полную F, работают значительно быстрее Adam по времени (и работают с минибатчами).

Кронекерово произведение

$$A = \begin{bmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nm} \end{bmatrix} \quad B = \begin{bmatrix} b_{11} & \dots & b_{1q} \\ \vdots & \ddots & \vdots \\ b_{p1} & \dots & b_{pq} \end{bmatrix}$$

Кронекерово произведение

$$A = \begin{bmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nm} \end{bmatrix} \quad B = \begin{bmatrix} b_{11} & \dots & b_{1q} \\ \vdots & \ddots & \vdots \\ b_{p1} & \dots & b_{pq} \end{bmatrix}$$

$$\mathrm{vec}\ A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} & a_{21} & \dots & a_{nm} \end{bmatrix}^T$$

Кронекерово произведение

$$A = \begin{bmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nm} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & \dots & b_{1q} \\ \vdots & \ddots & \vdots \\ b_{p1} & \dots & b_{pq} \end{bmatrix}$$

$$\mathrm{vec}\ A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} & a_{21} & \dots & a_{nm} \end{bmatrix}^T$$

$$A \otimes B = \begin{bmatrix} a_{11}\boldsymbol{B} & \dots & a_{1m}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{n1}\boldsymbol{B} & \dots & a_{nm}\boldsymbol{B} \end{bmatrix}$$

Кронекерово произведение

$$A \otimes B = \begin{bmatrix} a_{11}b_{11} & \dots & a_{11}b_{1q} & a_{12}b_{11} & \dots & a_{1m}b_{1q} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & \dots & a_{11}b_{pq} & a_{12}b_{p1} & \dots & a_{1m}b_{pq} \\ a_{21}b_{11} & \dots & a_{21}b_{1q} & a_{22}b_{11} & \dots & a_{2m}b_{1q} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{21}b_{p1} & \dots & a_{21}b_{pq} & a_{22}b_{p1} & \dots & a_{2m}b_{pq} \\ a_{n1}b_{11} & \dots & a_{n1}b_{1q} & a_{n2}b_{11} & \dots & a_{nm}b_{1q} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{n1}b_{p1} & \dots & a_{n1}b_{pq} & a_{n2}b_{p1} & \dots & a_{nm}b_{pq} \end{bmatrix}$$

K-FAC

Зададим нейросеть $f(x, \theta)$ с L слоями:

$$s_k = W_k a_{k-1}, \quad a_k = \sigma(s_k), \text{ где}$$

$$a_0 = x, \quad f(x, \theta) = a_L,$$

$$\theta = \left[(\mathrm{vec}\ W_1)^T ; \dots ; (\mathrm{vec}\ W_L)^T \right]^T$$

K-FAC

Зададим нейросеть $f(x, \theta)$ с L слоями:

$$s_k = W_k a_{k-1}, \quad a_k = \sigma(s_k), \text{ где}$$

$$a_0 = x, \quad f(x, \theta) = a_L,$$

$$\theta = \left[(\mathrm{vec}\ W_1)^T; \ldots; (\mathrm{vec}\ W_L)^T \right]^T$$

Введем обозначение $\mathcal{D}z = -\dfrac{\partial \log p(y|x,\theta))}{\partial z} = \dfrac{\partial \frac{1}{2}||y - f(x,\theta)||^2}{\partial z}$

K-FAC

Зададим нейросеть $f(x,\theta)$ с L слоями:

$$s_k = W_k a_{k-1}, \quad a_k = \sigma(s_k), \text{ где}$$

$$a_0 = x, \quad f(x,\theta) = a_L,$$

$$\theta = \left[(\mathrm{vec}\ W_1)^T; \ldots; (\mathrm{vec}\ W_L)^T \right]^T$$

Введем обозначение $\mathcal{D}z = -\dfrac{\partial \log p(y|x,\theta))}{\partial z} = \dfrac{\partial \frac{1}{2}||y - f(x,\theta)||^2}{\partial z}$

Тогда $F(\theta) = \mathbb{E}_{p(y|x,\theta)}\left[\mathcal{D}\theta \mathcal{D}\theta^T \right]$

K-FAC

Зададим нейросеть $f(x, \theta)$ с L слоями:

$$s_k = W_k a_{k-1}, \quad a_k = \sigma(s_k), \text{ где}$$

$$a_0 = x, \quad f(x, \theta) = a_L,$$

$$\theta = \left[(\mathrm{vec}\ W_1)^T; \dots; (\mathrm{vec}\ W_L)^T \right]^T$$

Введем обозначение $\mathcal{D}z = -\dfrac{\partial \log p(y|x, \theta))}{\partial z} = \dfrac{\partial \frac{1}{2} ||y - f(x, \theta)||^2}{\partial z}$

Тогда $F(\theta) = \mathbb{E}_{p(y|x,\theta)} \left[\mathcal{D}\theta \mathcal{D}\theta^T \right]$

$$F(\theta)_{ij} = \mathbb{E}_{p(y|x,\theta)} \left[(\mathrm{vec}\ \mathcal{D}W_i)(\mathrm{vec}\ \mathcal{D}W_j)^T \right]$$

K-FAC

Зададим нейросеть $f(x, \theta)$ с L слоями:

$$s_k = W_k a_{k-1}, \quad a_k = \sigma(s_k), \text{ где}$$

$$a_0 = x, \quad f(x, \theta) = a_L,$$

$$\theta = \left[(\mathrm{vec}\ W_1)^T; \ldots; (\mathrm{vec}\ W_L)^T\right]^T$$

Введем обозначение $\mathcal{D}z = -\dfrac{\partial \log p(y|x,\theta))}{\partial z} = \dfrac{\partial \frac{1}{2}||y - f(x,\theta)||^2}{\partial z}$

$$\text{Тогда } F(\theta) = \mathbb{E}_{p(y|x,\theta)}\left[\mathcal{D}\theta \mathcal{D}\theta^T\right]$$

$$F(\theta)_{ij} = \mathbb{E}_{p(y|x,\theta)}\left[(\mathrm{vec}\ \mathcal{D}W_i)(\mathrm{vec}\ \mathcal{D}W_j)^T\right]$$

$$\mathcal{D}W_k = (\mathcal{D}s_k)a_{k-1}^T = g_k a_{k-1}^T$$

K-FAC

Зададим нейросеть $f(x, \theta)$ с L слоями:

$$s_k = W_k a_{k-1}, \quad a_k = \sigma(s_k), \text{ где}$$

$$a_0 = x, \quad f(x, \theta) = a_L,$$

$$\theta = \left[(\mathrm{vec}\ W_1)^T; \dots; (\mathrm{vec}\ W_L)^T \right]^T$$

Введем обозначение $\mathcal{D}z = -\dfrac{\partial \log p(y|x, \theta))}{\partial z} = \dfrac{\partial \frac{1}{2}||y - f(x, \theta)||^2}{\partial z}$

$$\text{Тогда } F(\theta) = \mathbb{E}_{p(y|x,\theta)} \left[\mathcal{D}\theta \mathcal{D}\theta^T \right]$$

$$F(\theta)_{ij} = \mathbb{E}_{p(y|x,\theta)} \left[(\mathrm{vec}\ \mathcal{D}W_i)(\mathrm{vec}\ \mathcal{D}W_j)^T \right]$$

$$\mathcal{D}W_k = (\mathcal{D}s_k) a_{k-1}^T = g_k a_{k-1}^T$$

$$\mathrm{vec}\ \mathcal{D}W_k = a_{k-1} \otimes g_k$$

K-FAC

$$\mathrm{vec}\ \mathcal{D}W_k = \mathrm{vec}\ a_{k-1} \otimes \mathrm{vec}\ g_k$$

$$F(\theta)_{ij} = \mathbb{E}_{p(y|x,\theta)}\left[(\mathrm{vec}\ \mathcal{D}W_i)(\mathrm{vec}\ \mathcal{D}W_j)^T\right] =$$

K-FAC

$$\mathrm{vec}\ \mathcal{D}W_k = \mathrm{vec}\ a_{k-1} \otimes \mathrm{vec}\ g_k$$

$$F(\theta)_{ij} = \mathbb{E}_{p(y|x,\theta)}\left[(\mathrm{vec}\ \mathcal{D}W_i)(\mathrm{vec}\ \mathcal{D}W_j)^T\right] =$$

$$\mathbb{E}_{p(y|x,\theta)}\left[(a_{i-1} \otimes g_i)(a_{j-1} \otimes g_j)^T\right] =$$

K-FAC

$$\text{vec } \mathcal{D}W_k = \text{vec } a_{k-1} \otimes \text{vec } g_k$$

$$F(\theta)_{ij} = \mathbb{E}_{p(y|x,\theta)} \left[(\text{vec } \mathcal{D}W_i)(\text{vec } \mathcal{D}W_j)^T \right] =$$

$$\mathbb{E}_{p(y|x,\theta)} \left[(a_{i-1} \otimes g_i)(a_{j-1} \otimes g_j)^T \right] =$$

$$\mathbb{E}_{p(y|x,\theta)} \left[(a_{i-1} a_{j-1}^T) \otimes (g_i g_j^T) \right] \approx$$

K-FAC

$$\text{vec } \mathcal{D}W_k = \text{vec } a_{k-1} \otimes \text{vec } g_k$$

$$F(\theta)_{ij} = \mathbb{E}_{p(y|x,\theta)}\left[(\text{vec } \mathcal{D}W_i)(\text{vec } \mathcal{D}W_j)^T\right] =$$

$$\mathbb{E}_{p(y|x,\theta)}\left[(a_{i-1} \otimes g_i)(a_{j-1} \otimes g_j)^T\right] =$$

$$\mathbb{E}_{p(y|x,\theta)}\left[(a_{i-1}a_{j-1}^T) \otimes (g_i g_j^T)\right] \approx$$

$$\mathbb{E}_{p(y|x,\theta)}\left[a_{i-1}a_{j-1}^T\right] \otimes \mathbb{E}_{p(y|x,\theta)}\left[g_i g_j^T\right] = A_{i-1,j-1} \otimes G_{ij}$$

K-FAC

А если считать только блочнодиагональные элементы:

$$F_i = A_{i-1} \otimes G_i, \text{ где } A_{i-1} = \mathbb{E} a_{i-1} a_{i-1}^T, G_i = \mathbb{E} g_i g_i^T$$

K-FAC

А если считать только блочнодиагональные элементы:

$$F_i = A_{i-1} \otimes G_i, \text{ где } A_{i-1} = \mathbb{E} a_{i-1} a_{i-1}^T, \; G_i = \mathbb{E} g_i g_i^T$$

Более того, мы умеем легко обращать эту матрицу:

$$(F_i)^{-1} \operatorname{vec} \mathcal{D}W_i = G_i^{-1} W_i A_{i-1}^{-1}$$

K-FAC

А если считать только блочнодиагональные элементы:

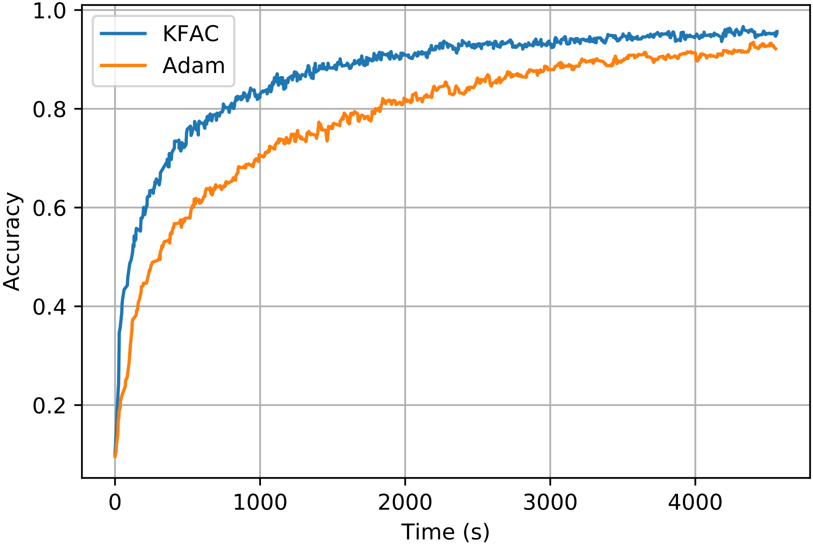
$$F_i = A_{i-1} \otimes G_i, \text{ где } A_{i-1} = \mathbb{E} a_{i-1} a_{i-1}^T, G_i = \mathbb{E} g_i g_i^T$$

Более того, мы умеем легко обращать эту матрицу:

$$(F_i)^{-1} \operatorname{vec} \mathcal{D} W_i = G_i^{-1} W_i A_{i-1}^{-1}$$

Ура, мы умеем считать натуральный градиент по K-FAC

K-FAC против Adam на CIFAR-10



Метод K-FAC

- сходится очень быстро по шагам, быстро по времени,
- почти не имеет гиперпараметров,
- не требует много памяти,
- работает с минибатчами,
- обобщается на сверточные, рекуррентные слои.
- обобщается на задачи классификации, регрессии, автокодировщика, VAE, GAN, сэмплирования.

Итоги

- Когда датасет достаточно маленький, а параметров немного, методы Ньютона, Гаусса-Ньютона, нат. градиентов сходятся значительно быстрей, и есть смысл их использовать.
- Для нейросетей есть метод K-FAC, который пока обгоняет Adam и его друзей. Заключается он просто в “отбеливании” градиентов и активаций, участвующих при подсчете градиента, для каждого слоя в отдельности.
- Более того, K-FAC доступен в TensorFlow как встроенный модуль, и его можно подключить в пару строчек.