Tuesday, May 5

Last class!

Only one more OH
(on Monday.)
Please reach out to meet!

"Quiz" for specific feedback
↳ topics
↳ course components
(readings, activities, reviews,
self-grades, whiteboard lectures)

Student experience surveys
for more general feedback

Senior projects / talks due Thursday midnight

Presentations Wed., May 13 7-10pm

Motivation

Previously, treated gradient descent as vector operation

But most weights are naturally in matrices (e.g., fully connected layers)

$W \in \mathbb{R}^{n \times d}$ is a weight matrix

$G = \nabla_W \mathcal{L} \in \mathbb{R}^{n \times d}$ is its gradient, $n \geq d$ WLOG

Goal: $\underset{\Delta W \in \mathbb{R}^{n \times d}}{\operatorname{argmax}} \quad \mathcal{L}(W) - \mathcal{L}(W + \Delta W)$

Goal: $\underset{\Delta W \in \mathbb{R}^{n \times d}}{\text{argmax}} \; \mathcal{L}(W) - \mathcal{L}(W + \Delta W)$

Taylor approximation: $\mathcal{L}(W+\Delta W) \approx \mathcal{L}(W) + \langle \nabla_W \mathcal{L}, \Delta W \rangle_F$ (sum of entrywise product)

Rearranging,

$$\mathcal{L}(W) - \mathcal{L}(W+\Delta W) \approx - \langle \nabla_W \mathcal{L}, \Delta W \rangle_F$$

$$\langle A, B \rangle_F = \sum_{j=1}^{d} \sum_{i=1}^{n} [A]_{i,j} [B]_{i,j} = \sum_{j=1}^{d} \sum_{i=1}^{n} [A^T]_{j,i} [B]_{i,j}$$

$$= \sum_{j=1}^{d} [A^T B]_{j,j} = \text{tr}(A^T B)$$

$$\mathcal{L}(W) - \mathcal{L}(W+\Delta W) \approx - \text{tr}(G^T \Delta W)$$

$\leq$ max over ΔW s.t. ΔW not too large

$$\mathcal{L}(W) - \mathcal{L}(W + \Delta W) \approx -\operatorname{tr}(G^T \Delta W)$$

$\Leftarrow$ max over ΔW s.t. ΔW "not too large"

Frobenius norm: $\epsilon \geq \|\Delta W\|_F = \sqrt{\sum_{i,j} [\Delta W]_{ij}^2}$ $\Rightarrow$ gradient descent

Spectral norm $\epsilon \geq \|\Delta W\|_2 = \max_i \sigma_i(\Delta W)$ $\Rightarrow$ Muon

Let $G = \sum_{i=1}^{d} \sigma_i u_i v_i^T$.

Claim: $$\operatorname*{argmin}_{\Delta W : \|\Delta W\|_2 \leq \epsilon} \operatorname{tr}(G^T \Delta W) = -\epsilon \sum_{i=1}^{d} u_i v_i^T$$

Proof: since $\{u_i\}_{i=1}^{d}$, $\{v_j\}_{j=1}^{d}$ orthonormal, we can write

$$\Delta W = \sum_{i=1}^{n} \sum_{j=1}^{d} c_{ij} u_i v_j^T$$

$$tr(G^T \Delta W) =$$

$$= \sum_{k=1}^{d} c_{kk} \sigma_k$$

choose smallest c_{kk} possible

$\|\Delta W\|_2 \le \epsilon \iff \forall x, \quad \|\Delta W x\|_2 \le \epsilon$

Choose $x = v_k$, then

$$\|\Delta W v_k\|_2^2 = \Big\| \sum_{i=1}^{n} \sum_{j=1}^{d} c_{ij} u_i v_j^T v_k \Big\|_2^2 = \Big\| \sum_{i=1}^{n} c_{ik} u_i \Big\|_2^2 = \sum_{i=1}^{n} c_{ik}^2 \le \epsilon^2$$

$$c_{kk}^2 \le \sum_{i=1}^{n} c_{ik}^2 \le \epsilon^2 \qquad \Rightarrow |c_{kk}| < \epsilon$$

$$\Rightarrow \text{smallest} \quad c_{kk} = -\epsilon$$

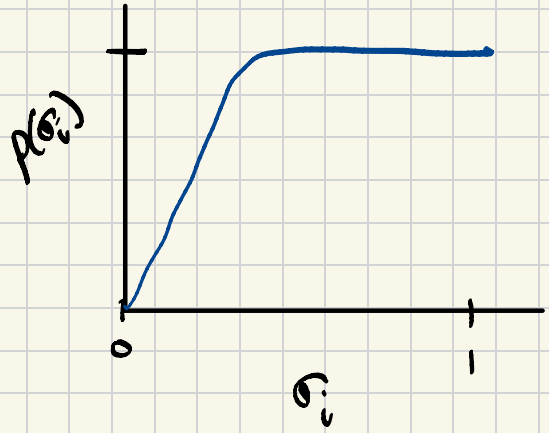
Then

$$c_{kk}^2 = \epsilon^2 \le \sum_{i=1}^{n} c_{ik}^2 \le \epsilon^2 \iff c_{ik} = 0 \quad \text{for } i \ne k$$

$$\Delta W = \sum_{i=1}^{d} -\epsilon \cdot u_i v_i^T$$

In practice, computing $\sum_{i=1}^{d} u_i v_i^T$ takes $O(nd^2)$ time.

Approximate via polynomial



Better polynomial => better optimizer

$\frac{3}{2}x - \frac{1}{2}x^3$

$3.4445x - 4.775x^3 + 2.0315x^5$

Polar Express