Conditional gradient methods

Task 1

$\nabla f(X_k)$ - ?

$f(X) = (1/2)\|X - Y\|_F^2$

then $f(X) = (1/2)\sum_{i=1}^{n}\sum_{j=1}^{n} X_{ij}^2$, $\nabla f(X) = X - Y$

Решение задачи LMO

Линейная задача минимизации (LMO):

$\min\langle\nabla f(X_k), S\rangle_{S\in B_n} = \min\langle X_k - Y, S\rangle_{S\in B_n} = \min \operatorname{tr}((X_k - Y)^T S)_{S\in B_n}$

Задача сводится к:

$\min \operatorname{tr}((X_k^T - Y^T)S)_{S\in B_n}$

Минимизация $\operatorname{tr}((X_k^T - Y^T)S)$ при $S \in B_n$ — это задача линейного назначения (linear assignment problem).

Task 2

See implementation in `solvation.ipynb`

Task 3

See test code in `solvation.ipynb`

Subgradient method

Task 4

See test code in `solvation.ipynb`

Set loss function: $f(x) = \|A^{1/2}(x-y)\|_2 - 1 + \|\Sigma x\|_\infty - 1$

Gradient of loss function: $\nabla f(x) = 2(A^{1/2})^T A^{1/2}(x - y) + \nabla\|\Sigma x\|_\infty$

Where $\nabla f(x) = [0 \dots \sigma_{max} \dots]^T$

Proximal gradient method

Subgradient Method

For a non-smooth convex function f(W), the subgradient update at step k is:

$W_{k+1} = W_k - \alpha_k g_k$

where:

- $\alpha_k > 0$ is the step size
- $g_k \in \partial f(W_k)$ is any subgradient of f at W_k

Where: $\nabla\|W\|_1 = \operatorname{sign}(W)$

$(\nabla W)_{ij} = \operatorname{sign}(W)_{ij} = \{ +1 \text{ if } W_{ij} > 0, -1 \text{ if } W_{ij} < 0, \text{any value} \in [-1, 1] \text{ if } W_{ij} = 0 \text{ (typically 0) } \}$

Proximal Gradient Method

For a composite function $f(W) = g(W) + h(W)$, where g is convex and differentiable, and h is convex but non-smooth, the update is:

$W_{k+1} = \text{prox}_{\alpha_k h}(W_k - \alpha_k \nabla g(W_k))$

where:

- $\alpha_k > 0$ is the step size
- $\text{prox}_{\alpha h}(V) = \text{argmin}_W(h(W) + (1/2\alpha)\|W-V\|_2^2)$ is the proximal operator of h

See code in `solvation.ipynb`

Stochastic gradient methods

Общий вывод: На сильно выпуклых функциях (MSE с L2):

1. SAG и SVRG сходятся линейно, обгоняя SGD
2. SVRG предпочтительнее из-за экономии памяти

На выпуклых, но не сильно выпуклых (LogLoss без регуляризации):

1. SAG может быть нестабилен (если $\mu \approx 0$)
2. SVRG всё ещё хорош, но требует аккуратного выбора частоты обновления полного градиента
3. SGD сходится, но медленно и с большим разбросом

Neural network training

See code && report in `solvation.ipynb`

Big Models

Setup	# of parameters	GPU peak memory (MB)	Final eval loss	Batch Size	Time to run 5 epochs (s)	Generation example	Comment
Baseline (GPT2)	124M	10101	2.126	8	377.29	`A long time ago in a galaxy...`	
facebook/opt-125m	125M	6753	1.825	8	365.27	`A long time ago in a galaxy...`	
facebook/opt-125m	125M	4233	1.745	4	341.27	`A long time ago in a galaxy...`	

See code in `solvation.ipynb`