Optimization with large learning rate

Denis Grachev

supervisor: Yurii Malitsky

University of Vienna

June 20, 2023

Defenitions

$$f_* := \min_{x \in \mathbb{R}^d} f(x)$$

f is L-smooth and μ-one-point-strongly-convexity (OPSC) with respect to x_* over $M \subset \mathbb{R}^d$.

Definition ($f : \mathbb{R}^d \to \mathbb{R}$ is L-smoothness)

- f is differentiable.
- $\exists L : \|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$.

Definition ($f : \mathbb{R}^d \to \mathbb{R}$ is μ-one-point-strongly-convex (OPSC) with respect to x_* over M)

- f is differentiable
- $\exists \mu > 0 : \langle \nabla f(x), x - x_* \rangle \geq \mu \|x - x_*\|^2, \ \forall x \in M.$

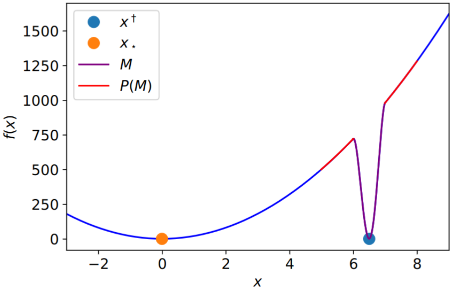
Motivation

- Standart threshold for learning rate in Gradient Descent is $\gamma < \frac{2}{L}$.
- For neural networks it has been widely observed that larger lerarning rates often obtain better models.
- In the article new theorems are presented about usage of larger learning rates in GD.
- Show the theorems and present performed experiments.

Lemma 1

- f is L_{global}-smooth.
- f has global minima x_* and local $x^\dagger$.
- f is $\mu^\dagger$-OPSC with respect to $x^\dagger$ over a set M with diameter r.
- $P(M) = B_{r_P}(x^\dagger) \setminus M$.
- f is $L < L_{\mathrm{global}}$ -smooth in $P(M)$ and μ_*-OPSC with respect to x_* and $\mu^\dagger > \frac{2L^2}{\mu_*}$.
- $x^\dagger$ is sufficiently far from $x_\star$.

Then using GD with $\frac{2}{\mu^\dagger} < \gamma < \frac{\mu_*}{L^2}$ if reach M GD will escape M and reach a point closer to x_* than $\|x^\dagger - x_*\| - r$ almost surely.



Theorem 1

- f is L-smooth.
- f is μ_*-OPSC with respect to the global minima x_* except the region M that contains local minima $x^\dagger$.
- $x^\dagger$ satisfies Lemma 1.

Then

- $\gamma < \frac{\mu^\dagger}{L^2_{\text{global}}}$
  GD initialized randomly inside M converges to $x^\dagger$.
- $\frac{2}{\mu^\dagger} < \gamma \leq \frac{\mu_*}{L^2}$
  GD initialized randomly inside $W : \mathcal{L}(W) > 0$ will converge to x^* almost surely.

Lemma 2

- GD initialized randomly in W with $\gamma \leq \frac{1}{2L}$
- $X \subset \mathbb{R}^d$ arbitrary set of points in the landscape, f is L-smooth over $\mathbb{R}^d \setminus X$

Then probabilty of encountering any point of X in first T steps is at most $2^{(T+1)d} \frac{\mathcal{L}(X)}{\mathcal{L}(W)}$

Theorem 2

- X be an arbitrary set of points.
- f is μ_*-OPSC with respect to a minima $x_* \notin X$ over $\mathbb{R}^d \setminus X$.
- $c_X := \inf \{\|x - x_*\| \mid x \in X\}$
- $r_W := \sup \{\|x - x_*\| \mid x \in W\}$

Then probability of not encountering any points of X during gradient descent with learning rate $\gamma \leq \frac{\mu_*}{L^2}$ is at least

- $c_X \leq r_W$

  $1 - \frac{r_W}{c_X}^{\frac{-d}{\log_2(1-\gamma\mu_*)}} \frac{\mathcal{L}(X)}{\mathcal{L}(W)} 2^d$
- otherwise

  1

Example 1D

$$f(x) := \begin{cases} -1600(x-2.5)^5 - 2000(x-2.5)^4 + & \\ \quad +800(x-2.5)^3 + 1020(x-2.5)^2 & 2 \leq x \leq 3 \\ 1411.2 \times \left(1 - 10^4(x-8.4)\right) & 8.4 \leq x \leq 8.40001 \\ 0 & 8.40001 \leq x \leq 8.59999 \\ 1479.2 \times \left(10^4(x-8.6)+1\right) & 8.59999 \leq x \leq 8.6 \\ 20x^2 & \textit{otherwise} \end{cases}$$

- Run GD with different start point and learning rate.
  $x_{\text{start}} \in \text{linespace}(8.5, 10, 20)$
  $lr \in \text{logspace}(-4, -0.5, 25)$
- Plot obtained minima and trajectories.
- Demo.

Example 2D

$$
\begin{aligned}
f(x,y) :=& x^2 + y^2 - \\
& 200\mathrm{ReLU}(|x|-1)\mathrm{ReLU}(|y|-1) \\
& \mathrm{ReLU}(2-|x|)\mathrm{ReLU}(2-|y|)
\end{aligned}
$$

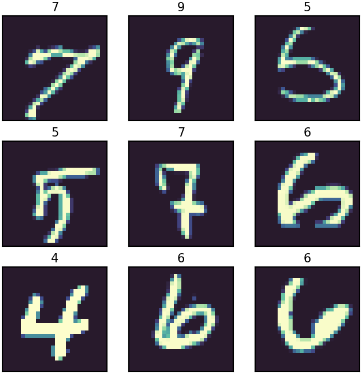
- Run GD with different start point and learning rate. x_{start} random in $[3,4] \times [3,4]$. 30 samples. $lr \in \text{logspace}(-1.75, -1.6, 25)$
- Plot trajectories.
- For each learning rate calculate share of each minima.
- Demo.

Brief introduction to ML

- Dataset
  $D := \{(x_1, y_1), (x_2, y_2) \dots (x_n, y_n) \parallel x_i \in \mathbb{R}^k, y_i \in C\}$
- NN with p parameters
  $f : \mathbb{R}^k \times \mathbb{R}^p \to \mathbb{R}^t$
- Loss function
  $\mathcal{L} : \mathbb{R}^t \times C \to \mathbb{R}$
- Total Loss
  $\mathcal{L}_{\text{total}}(D, \theta) = \sum_{i=1}^{n} l(f(x_i, \theta), y_i)$
- Training
  $\min_{\theta \in \mathbb{R}^p} \mathcal{L}(D, \theta)$
- Overfitting
  Dataset is splitted into training and testing

Dataset

- Mnist dataset.
- Pictures 28×28 pixels of numbers.
- 60000 training and 10000 validation sizes.
- $C = \{0, 1, 2, \ldots, 9\}$, $t = 10$

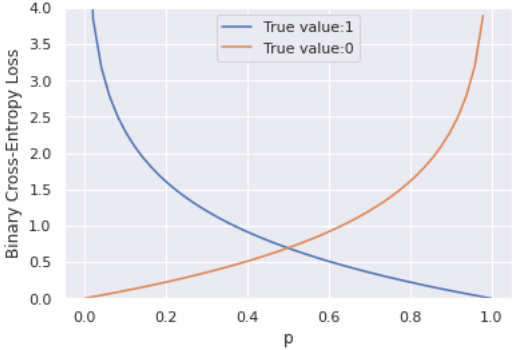


Loss function

Cross entropy loss.

$\mathcal{L} : \mathbb{R}^t \times C \to \mathbb{R}$

$\mathcal{L}(\hat{y}, y) := -\sum_{i=1}^{c} \mathbb{1}_{i==y} \log\left(\frac{\exp y_i}{\sum_{j=1}^{c} \exp y_j}\right) = -\log(p_{\text{true}})$



Structure of NN

Structure of NN is

$$f = f_k \circ \mathrm{Linear}_k \circ \ldots \circ f_1 \circ \mathrm{Linear}_1$$

where Linear_i is some linear function and f_i is a non linear elementwise function.
For f_i taken ReLU

$$\mathrm{ReLU}(x) := \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$

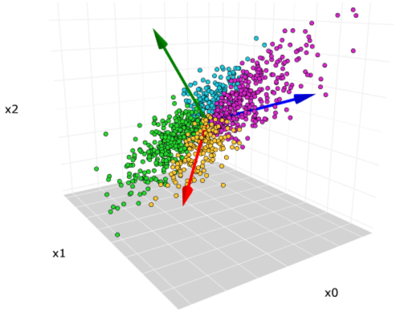
Total test structure

$$f = \mathrm{ReLU} \circ \mathrm{Linear}(16, 10) \circ \mathrm{ReLU} \circ \mathrm{Linear}(32, 16) \circ \mathrm{ReLU} \circ \mathrm{Linear}(784, 32)$$

Analyzis of NN

- 3 initial position were taken.
- From each position 3 GD with different learning rates started.
- Parameters were reduced to 2 dimensional space using PCA.
- Trajectories ploted.

Figure: PCA example



End

Thank you for your attention!