

Zeroth-Order Frank-Wolfe Optimization for Black-Box Adversarial Attacks

Marco Uderzo

marco.uderzo@studenti.unipd.it

Student ID: 2096998

Abstract

The goal of this project is to compare the behaviour and performance of two Zeroth-Order variants of the Frank-Wolfe Algorithm, aimed at solving constrained optimization problems with a better iteration complexity, especially with respect to oracle queries. We take into consideration: Faster Zeroth-Order Conditional Gradient Sliding (FZCGS) (Gao et al., 2018) and Stochastic Gradient Free Frank Wolfe (SGFFW) (Sahu et al., 2019). The latter algorithm branches off into three slightly different ones, depending on the Stochastic Approximation Technique used, namely: classical Kiefer-Wolfowitz Stochastic Approximation (KWSA) (Kiefer and Wolfowitz, 1952), Random Directions Stochastic Approximation (RDSA) (Nesterov and Spokoiny, 2011; Duchi et al., 2015), and an Improvised RDSA (IRDSA). The theory behind these algorithms is presented, with an emphasis on proving that the performance are guaranteed. Then, the aforementioned algorithms are tested on a black-box adversarial attack on the MNIST dataset.

1. Introduction

The Frank-Wolfe algorithm, also known as the conditional gradient method, is an iterative optimization technique used for constrained convex optimization problems. It was proposed by Marguerite Frank and Philip Wolfe in 1956, and nowadays finds various applications in the field of machine learning. It approximates the objective function by a first-order Taylor approximation. The algorithm iteratively selects a direction that minimizes the linear approximation of the objective function within the feasible set C . This direction is then combined with the current solution in a convex combination, and the process is repeated until convergence.

In particular, the Frank-Wolfe algorithm excels in constrained optimization problems with a closed convex set C :

$$\min_{x \in C} f(x)$$

The problem formulation can vary widely; for example, Gao et al. deal with a variant tailored for finite-sum minimization problems, in which the component functions $f_i(x)$ are summed up as follows:

$$\min_{x \in \Omega} F(x) = \frac{1}{n} \sum_{i=1}^n f_i(x)$$

Sahu et al., on the other hand, in order to estimate the loss function $f(x)$, deal with a variant that uses a stochastic zeroth-order oracle. The loss function is therefore defined as:

$$\min_{x \in C} f(x) = \min_{x \in C} \mathbb{E}_y [F(x, y)]$$

This paper addresses the use of the Frank-Wolfe algorithm for a particularly critical constrained optimization problem in Deep Learning, which is the problem of Adversarial Attacks. The objective of an adversarial attack is to find a small enough perturbation of the input able to make the neural network output the wrong prediction, while adhering to constraints inside of the convex set C . In our case, we use MNIST, so the goal is to find a non-trivial perturbation of the 28x28 black and white image of a hand-written digit. Moreover, as we will see later, we employ a zeroth-order variant of the Frank-Wolfe algorithm, since we don't have access to the full exact gradient.

1.1. Deterministic Frank-Wolfe Algorithm

In case first-order information is available in an optimization task, the deterministic version of the Frank-Wolfe algorithm can be a good choice.

When full, exact first-order information is available through an incremental first-order oracle (IFO), the Frank-Wolfe algorithm is basically described by the following two formulas:

$$\begin{aligned} v_t &= \arg \min_{v \in C} \langle h, \nabla f(x_t) \rangle \\ x_{t+1} &= (1 - \gamma_{t+1}) x_t + \gamma_{t+1} v_t, \end{aligned}$$

where

Method	Frobnability
Theirs	Frumpy
Yours	Frobbly
Ours	Makes one's heart Frob

Table 1. Results. Ours is better.

- $f(x_t)$ is the objective function we need to minimize;
- \mathcal{C} is the convex set;
- $\langle \cdot, \cdot \rangle$ is the inner/dot product;
- v_t is the direction we need to take in order to minimize the linear approximation;
- x_t is the current iteration result;
- h is a vector in the same space as x_t ;
- $\gamma_{t+1} = \frac{2}{t+2}$ is the step size.

2. References

List and number all bibliographical references in 9-point Times, single-spaced, at the end of your paper. When referenced in the text, enclose the citation number in square brackets, for example [1]. Where appropriate, include the name(s) of editors of referenced books.

2.1. Illustrations, graphs, and photographs

All graphics should be centered. Please ensure that any point you wish to make is resolvable in a printed copy of the paper. Resize fonts in figures to match the font in the body text, and choose line widths which render effectively in print. Many readers (and reviewers), even of an electronic copy, will choose to print your paper in order to read it. You cannot insist that they do otherwise, and therefore must not assume that they can zoom in to see tiny details on a graphic.

When placing figures in L^AT_EX, it's almost always best to use `\includegraphics`, and to specify the figure width as a multiple of the line width as in the example below

```
\usepackage[dvips]{graphicx} ...
\includegraphics[width=0.8\linewidth]
{myfile.eps}
```

References

- [1] Authors. The frobnicatable foo filter, 2014. Face and Gesture submission ID 324. Supplied as additional material fg324.pdf.