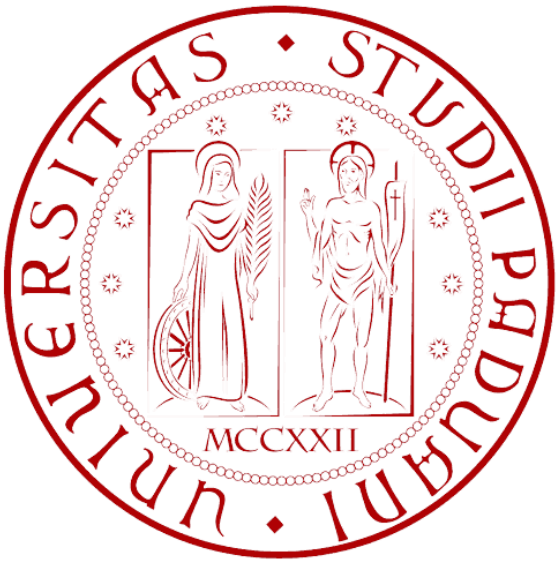

ANALYSIS OF ZEROTH ORDER FRANK WOLFE METHODS FOR ADVERSARIAL ATTACKS

September 22nd, 2023

OPTIMIZATION FOR DATA SCIENCE 2022/2023

Chelsie Adelle Renessa Romain
Matricola 2072067

Table of Contents

1 Abstract

This project presents an investigation and analysis of two gradient free modifications of the original Frank-Wolfe algorithm, namely Faster Zeroth Order Conditional Gradient Sliding Method (FZCGS) and Stochastic Gradient Free Frank-Wolfe (SGFFW).

This study examines the underlying principles and theoretical performance of these algorithms and then finally, tests their effectiveness on the practical problem of black-box attacks on neural networks.

2 Introduction

The Frank-Wolfe algorithm, also known as the Conditional gradient method, is based around approximating the objective function by a first order Taylor approximation. Its major applications are constrained optimization problems with a closed convex set.

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

2.1 Deterministic vs Stochastic Frank-Wolfe Algorithm

In cases where exact first order information is available, i.e., when one has access to an incremental first order oracle (IFO), a deterministic Frank-Wolfe method has the following steps:

$$\mathrm{v}_t = \arg\min_{v \in \mathcal{C}} \langle h, \nabla f(\mathrm{x}_t) \rangle$$

$$\mathrm{x}_{t+1} = (1 - \gamma_{t+1})\mathrm{x}_t + \gamma_{t+1}\mathrm{v}_t$$

Theoretically, both FZCGS and the KWSA implementation of SGFFW are deterministic. However, due to the large number of function queries in FZCGS which made the algorithm run for longer than 24 hours with little to no improvement to the loss value, in this project FZCGS was also implemented stochastically.

The stochastic variant of the Frank-Wolfe method is used when there is no access to an IFO i.e. first-order information is not available. A naïve replacement of $\nabla f(\mathrm{x}_k)$ by its stochastic counterpart $\nabla F(x_k, y_k)$ has the following steps

$$\mathrm{v}_t = \arg\min_{v \in \mathcal{C}} \langle h, \nabla F(\mathrm{x}_t, \mathrm{y}_t) \rangle$$

$$\mathrm{x}_{t+1} = (1 - \gamma_{t+1})\mathrm{x}_t + \gamma_{t+1}\mathrm{v}_t$$

However, this would make the algorithm potentially divergent due to non-vanishing variance of gradient approximations and the linear minimization constraint only holding in expectation. To counter this problem, in Sahu et al. the linear minimization was replaced with an averaging scheme that allows $\mathbb{E}[\|\mathrm{d}_t - \nabla f(\mathrm{x}_t)\|^2]$ to go to zero asymptotically. The subsequent steps are as follows:

$$\mathrm{d}_t = (1 - \rho_t)\mathrm{d}_{t-1} + \rho_t \mathrm{g}(\mathrm{x}_t, \mathrm{y}_t)$$

$$\mathrm{v}_t = \arg\min_{v \in \mathcal{C}} \langle \mathrm{d}_t, \mathrm{v} \rangle$$

$$\mathrm{x}_{t+1} = (1 - \gamma_{t+1})\mathrm{x}_t + \gamma_{t+1}\mathrm{v}_t$$

In this project both the RDSA and I-RDSA implementations of SGFFW are stochastic.

2.2 Zeroth-Order Optimization

The crux of zeroth order optimization consists of gradient approximation schemes from appropriately sampled values of the objective function. When the gradient of a function is not available, one can utilize the difference of the function value with respect to two random points to estimate it.

The paper by Gao et al. uses the coordinate wise gradient estimator which is defined as follows:

$$\hat{\nabla} f_i(x) = \sum_{j=1}^{d} \frac{f_i(\mathrm{x} + \mu_j \mathrm{e}_j) - f_i(\mathrm{x} - \mu_j \mathrm{e}_j)}{2\mu_j} \mathrm{e}_j$$

where $\mu_j > 0$ is the smoothing parameter and $\mathrm{e}_j \in \mathbb{R}^d$ denotes the basis vector where only the j-th element is 1 and all others are 0.

The paper by Sahu et al. uses multiple gradient approximation schemes. The first is the Kiefer-Wolfowitz stochastic approximation scheme (KWSA) which estimates the gradient by sampling the objective function along the canonical basis vectors. Formally, the gradient estimate is expressed as:

$$\mathrm{g}(\mathrm{x}_t; \mathrm{y}) = \sum_{i=1}^{d} \frac{F(\mathrm{x}_t + c_t \mathrm{e}_i\,; \mathrm{y}) - F(\mathrm{x}_t; \mathrm{y})}{c_t} \mathrm{e}_i$$

where c_t is a carefully chosen time decaying sequence. KWSA requires d samples at each step to evaluate the gradient.

The second gradient approximation scheme used by Sahu et al. is based on random directions. It avoids having to sample the objective function d times as in KWSA.

The random directions gradient estimator (RDSA) involves estimating the directional derivative along a randomly sampled direction from an appropriate probability distribution. Formally, the gradient estimator is given by:

$$\mathrm{g}(\mathrm{x}_t; \mathrm{y}, \mathrm{z}_t) = \frac{F(\mathrm{x}_t + c_t \mathrm{z}_t ; \mathrm{y}) - F(\mathrm{x}_t; \mathrm{y})}{c_t} \mathrm{z}_t$$

where $\mathrm{z}_t \in \mathbb{R}^d$ is a random vector sampled from a probability distribution such that $\mathbb{E}[\mathrm{z}_t \mathrm{z}_t^T] = \mathbb{I}_d$ and c_t is a carefully chosen time decaying sequence. With $c_t \to 0$ both estimators turn out to be unbiased estimators of the gradient $\nabla f(\mathrm{x}_t)$.

3 Algorithms

3.1 Faster Zeroth-Order Conditional Gradient Method (FZCGS)

The paper by Gao et. al (2020), considers the following constrained finite sum minimization problem:

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

where $\Omega \subset \mathbb{R}^d$ denotes a closed convex feasible set, each component function f_i is smooth and non-convex, and n represents the number of component functions.

The paper proposes the following:

- A faster stochastic zeroth-order Frank-Wolfe method with Function Queries Oracle (FQO) as $\mathcal{O}\left(\frac{n^{1/2}\, d}{\epsilon^2}\right)$.
- A faster stochastic zeroth-order conditional gradient sliding method and improved FQO to $\mathcal{O}\left(\frac{n^{1/2}\, d}{\epsilon}\right)$. This is the algorithm that is being explored in this report.
- A new stochastic first-order conditional gradient sliding method with Incremental First-Order Oracle (IFO) as $\mathcal{O}\left(\frac{n^{1/2}}{\epsilon}\right)$.

The paper has the following preliminaries:

Assumptions.

1. *The component function f_i $(i \in [n])$ is L-smooth as follows:*

$$\|\nabla f_i(\mathrm{x}) - \nabla f_i(\mathrm{y})\| \leq L\|\mathbf{x} - \mathrm{y}\|, \forall\, \mathrm{x}, \mathrm{y} \in \Omega \qquad (2)$$

2. *The diameter of the feasible set Ω is D.*
3. *The variance of the stochastic gradient $\nabla f_i(x)$ is bounded as follows:*

$$\frac{1}{n} \sum_{i=1}^{n} \|\nabla f_i(\mathrm{x}) - \nabla \mathrm{F}(\mathrm{x})\|^2 \leq \sigma^2 \qquad (3)$$

where $\sigma > 0$

Convergence Criterion.

The convergence criterion used for the standard Frank-Wolfe method is the Frank-Wolfe gap, which is defined as follows:

$$\mathcal{G}(\mathrm{x}) = \max_{\mathrm{u} \in \Omega} \langle \mathrm{u} - \mathrm{x}, \nabla F(\mathrm{x}) \rangle \qquad (4)$$

For the conditional gradient sliding method which incorporates the Nesterov's acceleration technique, the paper employs the following gradient mapping as the convergence criterion:

$$\mathcal{G}(\mathrm{x}, \nabla F(\mathrm{x}), \gamma) = \frac{1}{\gamma}\big(\mathrm{x} - \psi(\mathrm{x}, \nabla F(\mathrm{x}), \gamma)\big) \qquad (5)$$

where $\psi(\mathrm{x}, \nabla F(\mathrm{x}), \gamma)$ denotes a prox-mapping function which is defined as follows:

$$\psi(\mathrm{x}, \nabla F(\mathrm{x}), \gamma) = \arg\min_{\mathrm{y} \in \Omega} \langle \nabla F(\mathrm{x}), \mathrm{y} \rangle + \frac{1}{2\gamma} \|\mathrm{y} - \mathrm{x}\|^2 \qquad (6)$$

where $\gamma > 0$ is a hyper-parameter.

Oracle Model.

The paper uses the following oracle models to compare the iteration complexity of different algorithms.

- Function Query Oracle (FQO): FQO samples a component function and returns its function value $f_i(\mathrm{x})$
- Incremental First-Order Oracle (IFO): IFO samples a component function and returns its gradient $\nabla f_i(\mathrm{x})$
- Linear Minimization/ Maximization Oracle (LMO): LMO solves a linear programming problem and returns $\arg\max_{\mathrm{u} \in \Omega} \langle \mathrm{u}, \mathrm{v} \rangle$

For algorithm 1, if the parameters are chosen as $|S_1| = n$, $q = |S_2| = \sqrt{n}$, $\mu = \frac{1}{\sqrt{dK}}$, $\gamma_k = \gamma = \frac{1}{3L}$, and $\eta_k = \eta = \frac{1}{K}$ then theoretically the algorithm achieves the convergence rate:
$\mathbb{E}[\|\mathcal{G}(x_\alpha, \nabla F(x_\alpha), \gamma)\|^2] \leq \frac{(3(F(x_0)-F(x_*)+1)+7L)6L}{K}$.
With these same settings the amortized function queries oracle is $\mathcal{O}\left(\frac{n^{1/2}d}{\epsilon}\right)$ and the linear oracle is $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$.
The pseudocode for the algorithm and the gradient update is presented below.

Algorithm 1 Faster Zeroth-Order Conditional Gradient Method (FZCGS) (Gao et al. 2020)

Input: $x_0, q > 0, \mu > 0, K > 0, \eta > 0, \gamma > 0, n$
1: **for** $k = 0, \dots, K-1$ **do**
2: **if** $mod(k, q) = 0$ **then**
3: Sample S_1 without replacement to compute $\hat{v}_k = \hat{\nabla} f_{s_1}(x_k)$
4: **else**
5: Sample S_2 with replacement to compute $\hat{v}_k = \frac{1}{|S_2|}\sum_{i \in S_2}[\hat{\nabla} f_i(x_k) - \hat{\nabla} f_i(x_{k-1}) + \hat{v}_{k-1}]$
6: **end if**
7: $x_{k+1} = condg(\hat{v}_k, x_k, \gamma_k, \eta_k)$
8: **end for**
Output: Randomly choose x_α from $\{x_k\}$ and return it

Algorithm 2 $\mathrm{u}^+ = \mathrm{condg}(\mathrm{g}, \mathrm{u}, \gamma, \eta)$ (Qu et al. 2017)

1: $u_1 = u, t = 1$
2: v_t be an optimal solution for
$$V_{g,u,\gamma}(u_t) = \max_{x \in \Omega} \langle g + \tfrac{1}{\gamma}(u_t - u), u_t - x \rangle$$
3: If $V_{g,u,\gamma}(u_t) \leq \eta$, return $u^+ = u_t$.
4: Set $u_{t+1} = (1 - \alpha_t)u_t + \alpha_t v_t$ where,
$$\alpha_t = \min\left\{1, \frac{\langle \frac{1}{\gamma}(u - u_t) - g, v_t - u_t \rangle}{\frac{1}{\gamma}\|v_t - u_t\|^2}\right\}$$
5: Set $t \leftarrow t + 1$ and go to step 2

3.2 Stochastic Gradient Free Frank-Wolfe Method (SGFFW)

In the paper by Sahu et. al (2019), the objective is to solve the following optimization problem:

$$\min_{x \in \mathcal{C}} f(\mathrm{x}) = \min_{x \in \mathcal{C}} \mathbb{E}_{\mathrm{y} \sim \mathcal{P}}[F(\mathrm{x}; \mathrm{y})] \qquad (7)$$

where $\mathcal{C} \subset \mathbb{R}^d$ is a closed convex set, the loss functions and the expected loss functions, $F(\cdot\,; \mathrm{y})$ and $f(\cdot)$ respectively are possibly non-convex. In the context of (7), it is assumed that we have access to a stochastic zeroth-order oracle (SZO) and querying the SZO at iterate x_t yields an unbiased estimate of the loss function $f(\cdot)$ in the form of $F(\mathrm{x}_t; \mathrm{y}_t)$.
The paper has the following preliminaries:

Assumptions.

1. *In problem (7), the set $\mathcal{C}$ is bounded with finite diameter R.*
2. *F is convex and Lipschitz continuous with $\sqrt{\mathbb{E}[\|\nabla_x F(\mathrm{x}; \cdot)\|^2]} \leq L_1$ for all $x \in \mathcal{C}$.*
3. *The expected function $f(\cdot)$ is convex. Moreover, its gradient ∇f is L-Lipschitz continuous over the set $\mathcal{C}$, i.e. for all $x, y \in \mathcal{C}$,*
$$\|\nabla f(\mathrm{x}) - \nabla f(\mathrm{y})\| \leq L\|\mathbf{x} - \mathrm{y}\|$$
4. *The z_t's are drawn from a distribution μ such that $M(\mu) = \mathbb{E}[\|z_t\|^6]$ is finite, and for any vector $\mathrm{g} \in \mathbb{R}^d$, there exists a function $s(d)$: $\mathbb{N} \mapsto \mathbb{R}_+$ such that,*
$$\mathbb{E}[\|\langle \mathrm{g}, \mathrm{z}_t \rangle \mathrm{z}_t\|^2] \leq s(d)\|\mathrm{g}\|^2$$
5. *The unbiased gradient estimates, $F(\mathrm{x}\,; \mathrm{y})$ of $\nabla f(\mathrm{x})$, i.e., $\mathbb{E}_{\mathrm{y} \sim \mathcal{P}}[F(\mathrm{x}; \mathrm{y})] = \nabla f(\mathrm{x})$* satisfy
$$\mathbb{E}[\|\nabla F(\mathrm{x}, \mathrm{y}) - \nabla f(\mathrm{x})\|^2] \leq \sigma^2$$

The pseudocode for the algorithm is presented below.

Algorithm 3 Stochastic Gradient Free Frank Wolfe (SGFFW) (Sahu et al. 2019)

Require: Input, Loss Function $F(x)$, Convex Set C, number of directions m, sequences $\gamma_t = \frac{2}{t+8}$
$(\rho_t, c_t)_{KWSA} = \left(\frac{4}{(t+8)^{2/3}}, \frac{2}{d^{1/2}(t+8)^{1/3}}\right)$
$(\rho_t, c_t)_{RDSA} = \left(\frac{4}{d^{1/3}(t+8)^{2/3}}, \frac{2}{d^{3/2}(t+8)^{1/3}}\right)$
$(\rho_t, c_t)_{I-RDSA} = \left(\frac{4}{\left(1+\frac{d}{m}\right)^{1/3}(t+8)^{2/3}}, \frac{2\sqrt{m}}{d^{3/2}(t+8)^{1/3}}\right)$
Output: x_T or $\frac{1}{T}\sum_{t=0}^{T-1} x_t$
1: Initialize $x_0 \in C$
2: **for** $t = 0, \dots, T-1$ **do**

3: Compute

KWSA:

$\mathrm{g}(\mathbf{x}_t;\mathbf{y}) = \sum_{i=1}^{d} \frac{F(x_t+c_t e_i;y)-F(x_t;y)}{c_t} e_i$

RDSA: Sample $z_t \sim \mathcal{N}(0, \mathbb{I}_d)$

$\mathrm{g}(\mathbf{x}_t;\mathbf{y},\mathbf{z}_t) = \frac{F(x_t+c_t z_t;y)-F(x_t;y)}{c_t} z_t$

I-RDSA: Sample $\{z_{i,t}\}_{i=1}^{m} \sim \mathcal{N}(0, \mathbb{I}_d)$

$\mathrm{g}(\mathbf{x}_t;\mathbf{y},\mathbf{z}_t) = \frac{1}{m}\sum_{i=1}^{m} \frac{F(x_t+c_t z_{i,t};y)-F(x_t;y)}{c_t} z_{i,t}$

4: Compute $\mathrm{d}_t = (1-\rho_t)\mathrm{d}_{t-1} + \rho_t \mathrm{g}(\mathbf{x}_t, \mathrm{y}_t)$

5: Compute $\mathrm{v}_t = \mathrm{argmin}_{s \in C} \langle \mathrm{s}, \mathrm{d}_t \rangle$

6: Compute $\mathrm{x}_{t+1} = (1-\gamma_t)\mathrm{x}_t + \gamma_t \mathrm{v}_t$

7: **end for**

4 Experimental Results

4.1 Generation of Adversarial Examples

In this experiment, the performance of the proposed zeroth-order methods was verified on the task of an adversarial attack on black-box DNNs. In particular, given a black-box DNN $f: \mathbb{R}^d \to \mathbb{R}^c$ and a dataset $\left\{(x_i, y_i): x_i \in \mathbb{R}^d, y_i \in \{0, 1, \dots, c\}\right\}_{i=1}^{n}$, the task is to find the adversarial perturbation $\delta \in \mathbb{R}^d$ for sample x_i, such that the DNN model makes the incorrect prediction $\hat{y}_i \neq y_i$ (Figure 1). To accomplish this, the following problem was optimized:

$$\min_{\|\delta\|_\infty \leq s} \frac{1}{n}\sum_{i=1}^{n} \max\left\{f_{y_i}(x_i+\delta) - \max_{j \neq y_i} f_j(x_i+\delta), 0\right\}$$

where $f(x) = [f(x_1), f(x_2), \dots, f(x_c)]$ denotes the output of the last layer before conducting the softmax operation.

Figure 1 showing a sample from the MNIST dataset, the adversarial perturbation, and the resulting adversarial example.

Following both the Gao paper and its predecessors (Liu et al., 2018; Ji et al., 2019) the same pretrained DNN[1] for MNIST dataset was used as the black-box model.

This project employed some key changes from the papers studied. These changes include:

- For FZCGS, gradient estimation performed over random directions instead of over basis vectors. Since it is performed over random directions, we can choose the size of the subset of d-dimensions for best performance.
- In FZCGS, the conditional gradient algorithm only terminates when v_t is less than η. However, this may take a long time, so additional criteria were added for termination; when a certain number of iterations have been reached or when the update of α_t is no longer significant.
    - The size of the norm $\|\delta\|_\infty$ is no longer 0.01 as in the paper. Instead, s is set to ≥ 2. This made the algorithms faster as it is easier to find solutions in a larger solution space, however, the noise perturbation is consequently quite large and perceptible by the human eye (Figure 1) and not as covert as would be ideal in a true attack scenario.

4.2 Algorithm Comparison

The following graphs present the performance of each of the algorithms based on the number of iterations, time in hours, and query count. Figure 2 shows the same quantity twice, however the graph in b has been truncated to better visualize the number of iterations FZCGS, KWSA and I-RDSA as they are orders of magnitude less than the iterations of RDSA.

[1] https://github.com/IBM/ZOSVRG-BlackBox-Adv

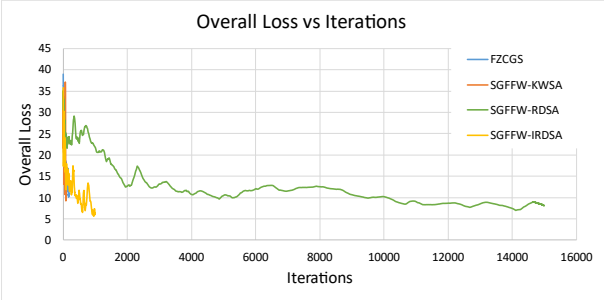

Figure 2.a. objective function value vs iterations

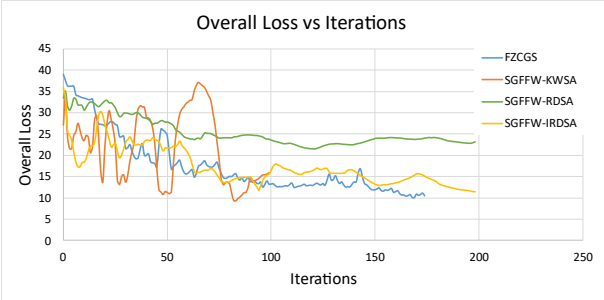

Figure 2.b. objective function value vs iterations (truncated graph)

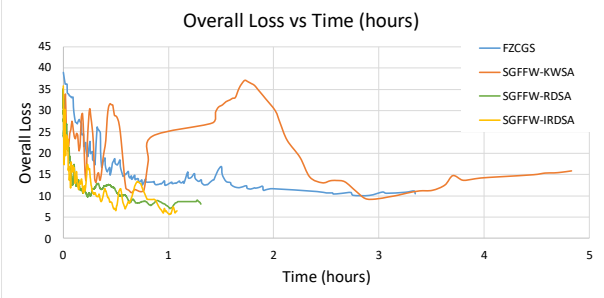

Figure 3 objective function value vs time in hours

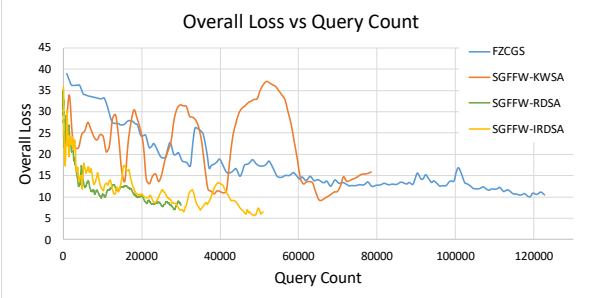

Figure 4 objective function value vs query count

5 Conclusion

Overall, no single algorithm is the best performer in all metrics. For this specific application, if performance is judged as “good” as soon as the algorithm reached an overall loss value near ten, then we can say that:

- FZCGS has the least number of iterations but the highest query count.
- KWSA took the longest time to reach a reasonable solution. Even though KWSA had the least number of iterations, the overall loss value is too variable and does not have a consistent decrease to be considered a good algorithm.
  It may benefit from some smoothing or early stopping condition once a “good” loss is reached.
- RDSA had the smallest query count but the largest number of iterations.
- I-RDSA was the fastest performing while not being the worst in any category and as such may be considered the “best” overall algorithm.

6 References

Gao, H., & Huang, H. (2020). Can stochastic zeroth-order Frank-Wolfe method converge faster for non-convex problems? *The 37th International Conference on Machine Learning* (pp. 3377-3386). Online: PMLR.

Sahu, A. K., Zaheer, M., & Kar, S. (2019). Towards gradient free and projection free stochastic optimization. *The 22nd International Conference on Artificial Intelligence and Statistics* (pp. 3468-3477). Naha, Okinawa, Japan: PMLR.