

Comparison of Frank-Wolfe Variants for White-Box Adversarial Attacks

Tanner Aaron Graves - 2073559 Alessandro Pala - 2107800

June 2024

1 Introduction

With deep neural networks becoming ubiquitous in application, adversarial attacks have received much attention, as it has proved remarkably easy to create adversarial examples- genuine data that undergoes a minimal and unobtrusive corruption process in order to maximally harm the performance of a model. This is typically stated as the following constrained optimization problem:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & \|x\|_p \leq \epsilon \end{aligned} \tag{1}$$

In the case of untargeted attacks on a classifier, where we perturb an example with the aim it be incorrectly predicted as any other class. Here $f(x)$ is the loss function of the attacked model $-\ell(x, \hat{y})$. In the case of targeted attacks we take $f(x) = \ell(x, \hat{y} \neq y)$. The L_p constraint $\|x\|_p \leq \epsilon$ directly restricts the size of perturbations made to the example. An inherent problem of DNNs is often attacks can be consistently successful even with very small, even imperceptible ϵ . Different choices of p may be made giving $\|x\|_p = (\sum_i x_i^p)^{1/p}$. Or commonly, as we use here $L_\infty(x) = \max_i |x_i|$. The constrained nature of this problem limits the applicability of a method like gradient descent, and requires integrating knowledge of the constraint space for effective optimization. Methods like Fast Signed Gradient attacks and projected gradient descent are popular choices, but are each limited either by providing maximal and simple adversarial examples or being computationally inefficient. We explore Frank-Wolfe variants which are well suited to this problem by ensuring feasibility within the constraint set at each iteration with the efficient solving of a Linear Minimization Oracle (LMO). Here we focus on white-box adversarial attacks which utilize the model structure for calculating $\nabla f(x)$. In scenarios where this is not accessible, attacks are black-box and generally require methods to estimate the gradient with zeroth order methods.

2 Algorithms

2.1 Frank-Wolfe

Algorithm 1 An algorithm with caption

Require: maximum iterations T , stepsizes $\{\gamma_t\}$, convergence tolerance δ

Ensure: $y = x^n$

```

1:  $x_0 = x_{\text{ori}}$ 
2: for  $t = 1, \dots, T$  do
3:    $s_t = \arg \min_{x \in \mathcal{M}} \langle x, \nabla f(x_t) \rangle$  ▷ LMO step
4:    $d_t = s_t - x_t$ 
5:    $x_{t+1} = x_t + \gamma_t d_t$ 
6:   if  $\langle d_t, -\nabla f(x) \rangle < \delta$  then return ▷ FW gap convergence criterion
7:   end if
8: end for
```

Observing oscilation in Frank Wolfe convergence is common and consequence of optimal points lying on a face of \mathcal{M} . Since at each iteration the method is moving twrods a vetex of \mathcal{M} , which we denote the set of as \mathcal{S} , the method "zigzags", moving twrods different points in effort to gradually approach the face on which the optimum lies. We implement varients that aim to address this problem to provide better convergece. The simplest of which is adding momentum to standerd frank wolf which replaces the gradient in the LMO calculation in line (4) with a momentum term $m_t = \beta m_{t-1} + (1 - \beta) \nabla f(x_t)$ and initialize $m_0 = \nabla f(x_0)$. By considering this exponentially weighted average of gradient information, momentum varients are emperically observed to have nicer convergence. The LMO finds an optimal direction on The LMO is also refered tp as the Frank-Wolfe sub problem and can be generally costly to compute, however, in the case of our constraint set \mathcal{M} which is the L_∞ ball, the LMO has a closed form solution:

$$s_t = -\epsilon \text{sign}(\nabla f(x)) + x_{\text{ori}}$$

The convergence criterion used here is the Frank

2.2 Away-Step Frank-Wolfe

2.3 Pairwise Frank-Wolfe

3 Results

Introduce Datasets

Algorithm 2 Away-Step FW for Adversarial Attacks

Require: maximum iterations T , stepsizes $\{\gamma_t\}$, convergence tolerance δ , $x_0 \in \mathcal{M}$

- 1: Define $S_0 := \{x_0\}$ with $\alpha_{x_0} = 1$
- 2: **for** $t = 1, \dots, T$ **do**
- 3: $s_t := \arg \min_{x \in \mathcal{M}} \langle x, \nabla f(x_t) \rangle$ ▷ LMO step
- 4: $d_t^{\text{FW}} := s_t - x_t$
- 5: $v_t := \arg \max_{v \in S_t} \langle v, \nabla f(x_t) \rangle$
- 6: $d_t^{\text{A}} := x_t - v_t$
- 7: **if** $\langle d_t^{\text{FW}}, -\nabla f(x) \rangle < \delta$ **then** return x_t ▷ FW gap convergence criterion
- 8: **end if**
- 9: **if then** $\langle d_t^{\text{FW}}, -\nabla f(x) \rangle < \langle d_t^{\text{A}}, -\nabla f(x) \rangle$
- 10: $d_t = d_t^{\text{FW}}, \gamma_{\max} := 1$
- 11: **else**
- 12: $d_t := d_t^{\text{A}}, \gamma_{\max} := \frac{\alpha_{v_t}}{1 - \alpha_{v_t}}$
- 13: **end if**
- 14: $x_{t+1} = x_t + \gamma_t d_t$
- 15: Update α, S_{t+1} s.t. $\langle \alpha, S_{t+1} \rangle = x_{t+1}$ (See below)
- 16: **end for**

3.1 Momentum

3.2 Early-Stopping (Convergece Criterion)

It is worth noting that Convergence Criterion For Frank-Wolfe is a somewhat imprecise surrogate for success in the context of adversarial attacks. For many examples, we find that Frank-Wolfe methods create successful attacks several iterations before convergence. We attribute this to an incorrect class probability being grater than the correct class being sufficient for success where convergence is reached when the new output class probability is maximized. We observe the convergence of the Frank-Wolfe gap

3.3 Stepsize

The methods for stepsize where implemented: fixed, where $\gamma_t = 1$ for all t , linesearching which solves $\arg \min_{\gamma} f(x + \gamma d_t)$ where $\gamma \in (0, 1]$, and decaying stepsize $\gamma_t = \frac{2}{t+2}$.

3.4 ϵ Choice

Create plot showing how accurate attacks are with different ϵ constraints.

4 Convergence Analysis

The constrained nature of the Adversarial Attack problem means that the norm of the gradient $\|\nabla_x f(x)\|$ is not a suitable convergence criterion as boundary points need not have 0 gradient. The Frank-Wolfe gap provides provides measure of both optimality and point feasibility. It is a measure of the maximum improvement over the current iteration x_t within the constraints C and defined

$$g(x_t) = \max_{x \in C} \langle x - x_t, -\nabla f(x_t) \rangle$$

We always have $g(x_t) \geq 0$ and its usefulness as a convergence criterion comes from $g(x_t) = 0$ iff x_t is a stationary point. For convex problems, we would have that the linear approximation $f(x_t) + \langle x_t - x, -\nabla f(x_t) \rangle \geq f(x)$. However, the loss of DNNs as commonly the subject of adversarial attacks, are highly non-convex, making this only true locally. This complicate the convergence of Frank-Wolfe in this application, but it is still gaurenteed.

4.1 Frank-Wolfe

4.2 Pairwise Frank-Wolfe

4.3 Away-Step Frank-Wolfe