# Efficient Regret Minimization in Non-Convex Games

**Elad Hazan** [1] **Karan Singh** [1] **Cyril Zhang** [1]

## Abstract

We consider regret minimization in repeated games with non-convex loss functions. Minimizing the standard notion of regret is computationally intractable. Thus, we define a natural notion of regret which permits efficient optimization and generalizes offline guarantees for convergence to an approximate local optimum. We give gradient-based methods that achieve optimal regret, which in turn guarantee convergence to equilibrium in this framework.

## 1. Introduction

Repeated games with non-convex utility functions serve to model many natural settings, such as multiplayer games with risk-averse players and adversarial (e.g. GAN) training. However, standard regret minimization and equilibria computation with general non-convex losses are computationally hard. This paper studies *computationally tractable* notions of regret minimization and equilibria in non-convex repeated games.

Efficient online learning algorithms are intimately connected to convexity. This connection is natural, since in a very broad sense[1], convexity captures efficient computation in continuous mathematical optimization.

The recent success of non-convex learning models, notably deep neural networks, motivates the need for efficient non-convex optimization. Since the latter is NP-hard in general, efficient optimization methods for non-convex optimization are designed to find local minima of varied quality. Stochastic gradient-based methods for non-convex optimization are currently state-of-the-art in training non-convex machines.

[1]Computer Science, Princeton University. Correspondence to: Elad Hazan <ehazan@princeton.edu>, Karan Singh <karans@princeton.edu>, Cyril Zhang <cyril.zhang@princeton.edu>.

*Proceedings of the $34^{th}$ International Conference on Machine Learning*, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

[1]Though by no means exclusively, PCA and other spectral methods being a notable exception.

In this paper we investigate the generalization of the non-convex statistical, or batch, learning model to online learning. The main question we ask is what kind of guarantees can be obtained *efficiently* in an adversarial non-convex scenario.

After briefly discussing why standard regret is not a suitable metric of performance, we introduce and motivate *local regret*, a surrogate for regret to the non-convex world. We then proceed to give efficient algorithms for non-convex online learning with optimal guarantees for this new objective. In analogy with the convex setting, we discuss the way our framework captures the offline and stochastic cases. In the final section, we describe a game-theoretic solution concept which is intuitively appealing, and, in contrast to Nash equilibria, efficiently attainable by simple algorithms.

### 1.1. Related work

The field of online learning is by now rich with a diverse set of algorithms for extremely general scenarios, see e.g. (Cesa-Bianchi & Lugosi, 2006). For bounded cost functions over a bounded domain, it is well known that versions of the multiplicative weights method gives near-optimal regret bounds (Cover, 1991; Vovk, 1990; Arora et al., 2012).

Despite the tremendous generality in terms of prediction, the multiplicative weights method in its various forms yields only exponential-time algorithms for these general scenarios. This is inevitable, since regret minimization implies optimization, and general non-convex optimization is NP-hard. Convex forms of regret minimization have dominated the learning literature in recent years due to the fact that they allow for efficient optimization, see e.g. (Hazan, 2016; Shalev-Shwartz, 2011).

Non-convex mathematical optimization algorithms typically find a local optimum. For smooth optimization, gradient-based methods are known to find a point with gradient of squared norm at most $\varepsilon$ in $O(\frac{1}{\varepsilon})$ iterations (Nesterov, 2004).[2] A rate of $O(\frac{1}{\varepsilon^2})$ is known for stochastic

[2]We note here that we measure the squared norm of the gradient, since it is more compatible with convex optimization. The mathematical optimization literature sometimes measures the norm of the gradient without squaring it.

gradient descent (Ghadimi & Lan, 2013). Further accelerations in terms of the dimension are possible via adaptive regularization (Duchi et al., 2011).

Recently, stochastic second-order methods have been considered, which enable even better guarantees for non-convex optimization: not only is the gradient at the point returned small, but the Hessian is also guaranteed to be close to positive semidefinite (i.e. the objective function is locally almost-convex), see e.g. (Erdogdu & Montanari, 2015; Carmon et al., 2016; Agarwal et al., 2016a;b).

The relationship between regret minimization and learning in games has been considered in both the machine learning literature, starting with (Freund & Schapire, 1997), and the game theory literature by (Hart & Mas-Colell, 2000). Motivated by (Hart & Mas-Colell, 2000), (Blum & Mansour, 2005) study reductions from internal to external regret, and (Hazan & Kale, 2007) relate the computational efficiency of these reductions to fixed point computations.

## 2. Setting

We begin by introducing the setting of online non-convex optimization, which is modeled as a game between a learner and an adversary. During each iteration $t$, the learner is tasked with predicting $x_t$ from $\mathcal{K} \subseteq \mathbb{R}^n$, a convex decision set. Concurrently, the adversary chooses a loss function $f_t : \mathcal{K} \to \mathbb{R}$; the learner then observes $f_t(x)$ (via access to a first-order oracle) and suffers a loss of $f_t(x_t)$. This procedure of play is repeated across $T$ rounds.

The performance of the learner is measured through its regret, which is defined as a function of the loss sequence $f_1, \ldots, f_T$ and the sequence of online decisions $x_1, \ldots, x_T$ made by the learner. We discuss our choice of regret measure at length in Section 2.2.

Throughout this paper, we assume the following standard regularity conditions:

**Assumption 2.1.** *We assume the following is true for each loss function $f_t$:*

*(i)* $f_t$ *is bounded:* $|f_t(x)| \leq M$.

*(ii)* $f_t$ *is L-Lipschitz:* $|f_t(x) - f_t(y)| \leq L\|x - y\|$.

*(iii)* $f_t$ *is $\beta$-smooth (has a $\beta$-Lipschitz gradient):*

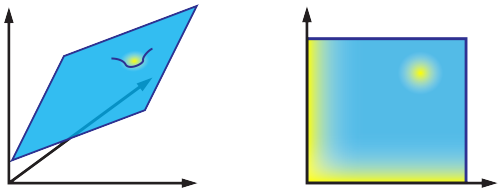
$$\|\nabla f_t(x) - \nabla f_t(y)\| \leq \beta\|x - y\|.$$

### 2.1. Projected Gradients and Constrained Non-Convex Optimization

In constrained non-convex optimization, minimizing the gradient presents difficult computational challenges. In general, even when objective functions are smooth and bounded, local information may provide no information about the location of a stationary point. This motivates us to refine our search criteria.

Consider, for example, the function sketched in Figure 1. In this construction, defined on the hypercube in $\mathbb{R}^n$, the unique point with a vanishing gradient is a hidden valley, and gradients outside this valley are all identical. Clearly, it is hopeless in an information-theoretic sense to find this point efficiently: the number of value or gradient evaluations of this function must be $\exp(\Omega(n))$ to discover the valley.

*Figure 1.* A difficult "needle in a haystack" case for constrained non-convex optimization. *Left:* A function with a hidden valley, with small gradients shown in yellow. *Right:* Regions with small *projected* gradient for the same function. For smaller $\eta$, only points near the valley and bottom-left corner have small projected gradient.



To circumvent such inherently difficult and degenerate cases, we relax our conditions, and try to find a vanishing *projected gradient*. In this section, we introduce this notion formally, and motivate it as a natural quantity of interest to capture the search for local minima in constrained non-convex optimization.

**Definition 2.2** (Projected gradient)**.** *Let $f : \mathcal{K} \to \mathbb{R}$ be a differentiable function on a closed (but not necessarily bounded) convex set $\mathcal{K} \subseteq \mathbb{R}^n$. Let $\eta > 0$. We define $\nabla_{\mathcal{K},\eta} f : \mathcal{K} \to \mathbb{R}^n$, the* $(\mathcal{K}, \eta)$-projected gradient *of $f$, by*

$$\nabla_{\mathcal{K},\eta} f(x) \stackrel{def}{=} \frac{1}{\eta}\left(x - \Pi_{\mathcal{K}}\left[x - \eta\nabla f(x)\right]\right),$$

*where $\Pi_{\mathcal{K}}[\cdot]$ denotes the orthogonal projection onto $\mathcal{K}$.*

This can be viewed as a surrogate for the gradient which ensures that the gradient descent step always lies within $\mathcal{K}$, by transforming it into a projected gradient descent step. Indeed, one can verify by definition that

$$x - \eta\nabla_{\mathcal{K},\eta}(x) = \Pi_{\mathcal{K}}\left[x - \eta\nabla f(x)\right].$$

In particular, when $\mathcal{K} = \mathbb{R}^n$,

$$\nabla_{\mathcal{K},\eta} f(x) = \frac{1}{\eta}(x - x + \eta\nabla f(x)) = \nabla f(x),$$

and we retrieve the usual gradient at all $x$.

We first note that there always exists a point with vanishing projected gradient.

**Proposition 2.3.** *Let $\mathcal{K}$ be a compact convex set, and suppose $f : \mathcal{K} \to \mathbb{R}$ satisfies Assumption 2.1. Then, there exists some point $x^* \in \mathcal{K}$ for which*

$$\nabla_{\mathcal{K},\eta} f(x^*) = 0.$$

*Proof.* Consider the map $g : \mathcal{K} \to \mathcal{K}$, defined by

$$g(x) \stackrel{\text{def}}{=} x - \eta \nabla_{\mathcal{K},\eta} f(x) = \Pi_{\mathcal{K}} \left[ x - \eta \nabla f(x) \right].$$

This is a composition of continuous functions (noting that the smoothness assumption implies that $\nabla f$ is continuous), and is therefore continuous. Thus $g$ satisfies the conditions for Brouwer's fixed point theorem, implying that there exists some $x^* \in \mathcal{K}$ for which $g(x^*) = x^*$. At this point, the projected gradient vanishes. □

In the limit where $\eta \|\nabla f(x)\|$ is infinitesimally small, the projected gradient is equal to the gradient in the interior of $\mathcal{K}$; on the boundary of $\mathcal{K}$, it is the gradient with its outward-facing component removed. This exactly captures the first-order condition for a local minimum.

The final property that we note here is that an approximate local minimum, as measured by a small projected gradient, is robust with respect to small perturbations.

**Proposition 2.4.** *Let $x$ be any point in $\mathcal{K} \subseteq \mathbb{R}^n$, and let $f, g$ be differentiable functions $\mathcal{K} \to \mathbb{R}$. Then, for any $\eta > 0$,*

$$\|\nabla_{\mathcal{K},\eta}[f+g](x)\| \leq \|\nabla_{\mathcal{K},\eta} f(x)\| + \|\nabla g(x)\|.$$

*Proof.* Let $u = x + \eta \nabla f(x)$, and $v = u + \eta \nabla g(x)$. Define their respective projections $u' = \Pi_{\mathcal{K}}[u], v' = \Pi_{\mathcal{K}}[v]$, so that $u' = x - \eta \nabla_{\mathcal{K},\eta} f(x)$ and $v' = x - \eta \nabla_{\mathcal{K},\eta}[f+g](x)$. We first show that $\|u' - v'\| \leq \|u - v\|$.

By the generalized Pythagorean theorem for convex sets, we have both $\langle u' - v', v - v' \rangle \leq 0$ and $\langle v' - u', u - u' \rangle \leq 0$. Summing these, we get

$$\begin{aligned} \langle u' - v', u' - v' - (u - v) \rangle &\leq 0 \\ \implies \|u' - v'\|^2 &\leq \langle u' - v', u - v \rangle \\ &\leq \|u' - v'\| \cdot \|u - v\|, \end{aligned}$$

as claimed. Finally, by the triangle inequality, we have

$$\begin{aligned} &\|\nabla_{\mathcal{K},\eta}[f+g](x)\| - \|\nabla_{\mathcal{K},\eta} f(x)\| \\ &\quad \leq \|\nabla_{\mathcal{K},\eta}[f+g](x) - \nabla_{\mathcal{K},\eta} f(x)\| \\ &\quad = \frac{1}{\eta}\|u' - v'\| \\ &\quad \leq \frac{1}{\eta}\|u - v\| = \|\nabla g(x)\|, \end{aligned}$$

as required.

□

In particular, this fact immediately implies that $\|\nabla_{\mathcal{K},\eta} f(x)\| \leq \|\nabla f(x)\|$.

As we demonstrate later, looking for a small projected gradient becomes a feasible task. In Figure 1 above, such a point exists on the boundary of $\mathcal{K}$, even when there is no "hidden valley" at all.

### 2.2. A Local Regret Measure

In the well-established framework of online convex optimization, numerous algorithms can efficiently achieve optimal regret, in the sense of converging in terms of average loss towards the best fixed decision in hindsight. That is, for any $u \in \mathcal{K}$, one can play iterates $x_1, \ldots, x_T$ such that

$$\frac{1}{T}\sum_{i=1}^{T} \left[ f_t(x_t) - f_t(u) \right] = o(1).$$

Unfortunately, even in the *offline* case, it is too ambitious to converge towards a global minimizer in hindsight. In the existing literature, it is usual to state convergence guarantees towards an $\varepsilon$-approximate stationary point – that is, there exists some iterate $x_t$ for which $\|\nabla f(x_t)\|^2 \leq \varepsilon$. As discussed in the previous section, the projected gradient is a natural analogue for the constrained case.

In light of the computational intractability of direct analogues of convex regret, we introduce *local regret*, a new notion of regret which quantifies the objective of predicting points with small gradients on average. The remainder of this paper discusses the motivating roles of this quantity.

Throughout this paper, for convenience, we will use the following notation to denote the sliding-window time average of functions $f$, parametrized by some window size $1 \leq w \leq T$:

$$F_{t,w}(x) \stackrel{\text{def}}{=} \frac{1}{w}\sum_{i=0}^{w-1} f_{t-i}(x).$$

For simplicity of notation, we define $f_t(x)$ to be identically zero for all $t \leq 0$. We define local regret below:

**Definition 2.5** (Local regret)**.** *Fix some $\eta > 0$. Define the* $w$-local regret *of an online algorithm as*

$$\mathfrak{R}_w(T) \stackrel{def}{=} \sum_{t=1}^{T} \|\nabla_{\mathcal{K},\eta} F_{t,w}(x_t)\|^2,$$

When the window size $w$ is understood by context, we omit the parameter, writing simply *local regret* as well as $F_t(x)$.

We turn to the first motivating perspective on local regret. When an algorithm incurs local regret sublinear in $T$, a randomly selected iterate has a small time-averaged gradient in expectation:

**Proposition 2.6.** *Let $x_1, \ldots, x_T$ be the iterates produced by an algorithm for online non-convex optimization which incurs a local regret of $\mathfrak{R}_w(T)$. Then,*

$$\mathbb{E}_{t\sim\mathrm{Unif}([T])}\left[\|\nabla_{\mathcal{K},\eta}F_{t,w}(x_t)\|^2\right] \le \frac{\mathfrak{R}_w(T)}{T}.$$

This generalizes typical convergence results for the gradient in offline non-convex optimization; we discuss concrete reductions in Section 4.

### 2.3. Why Smoothing is Necessary

In this section, we show that for any online algorithm, an adversarial sequence of loss functions can force the local regret incurred to scale with $T$ as $\Omega\left(\frac{T}{w^2}\right)$. This demonstrates the need for a time-smoothed performance measure in our setting, and justifies our choice of larger values of the window size $w$ in the sections that follow.

**Theorem 2.7.** *Define $\mathcal{K} = [-1, 1]$. For any $T \ge 1$, $1 \le w \le T$, and $\eta \le 1$, there exists a distribution $\mathcal{D}$ on 0-smooth, 1-bounded cost functions $f_1, \ldots, f_T$ on $\mathcal{K}$ such that for any online algorithm, when run on this sequence of functions,*

$$\mathbb{E}_{\mathcal{D}}\left[\mathfrak{R}_w(T)\right] \ge \frac{1}{4w}\left\lfloor\frac{T}{2w}\right\rfloor.$$

*Proof.* We begin by partitioning the $T$ rounds of play into $\lfloor\frac{T}{2w}\rfloor$ repeated segments, each of length $2w$.

For the first half of the first segment ($t = 1, \ldots, w$), the adversary declares that

- For odd $t$, select $f_t(x)$ i.i.d. as follows:

$$f_t(x) := \begin{cases} -x, & \text{with probability } \frac{1}{2} \\ x, & \text{with probability } \frac{1}{2} \end{cases}$$

- For even $t$, $f_t(x) := -f_{t-1}(x)$.

During the second half ($t = w+1, \ldots, 2w$), the adversary sets all $f_t(x) = 0$. This construction is repeated $\lfloor\frac{T}{2w}\rfloor$ times, padding the final $T \bmod 2w$ costs arbitrarily with $f_t(x) = 0$.

By this construction, at each round $t$ at which $f_t(x)$ is drawn randomly, we have $F_{t,w}(x) = f_t(x)/w$. Furthermore, for any $x_t$ played by the algorithm, $|f_t(x_t)| = 1$ with probability at least $\frac{1}{2}$. so that $\mathbb{E}\left[\|\nabla_{\mathcal{K},\eta}F_{t,w}(x_t)\|^2\right] \ge \frac{1}{2w^2}$. The claim now follows from the fact that there are at least $\frac{w}{2}$ of these rounds per segment, and exactly $\lfloor\frac{T}{2w}\rfloor$ segments in total. □

We further note that the notion of time-smoothing captures non-convex online optimization under limited *concept drift*: in online learning problems where $F_{t,w}(x) \approx f_t(x)$, a bound on local regret truly captures a guarantee of playing points with small gradients.

## 3. An Efficient Non-Convex Regret Minimization Algorithm

Our approach, as given in Algorithm 1, is to play follow-the-leader iterates, approximated to a suitable tolerance using projected gradient descent. We show that this method efficiently achieves an optimal local regret bound of $O\left(\frac{T}{w^2}\right)$, taking $O\left(Tw\right)$ iterations.

**Algorithm 1** Time-smoothed online gradient descent

```
Input: window size w ≥ 1, learning rate 0 < η < β/2,
  tolerance δ > 0, a convex body K ⊆ R^n.
Set x_1 ∈ K arbitrarily.
for t = 1, ..., T do
    Predict x_t. Observe the cost function f_t : K → R.
    Initialize x_{t+1} := x_t.
    while ‖∇_{K,η} F_{t,w}(x_{t+1})‖ > δ/w do
        Update x_{t+1} := x_{t+1} − η ∇_{K,η} F_{t,w}(x_{t+1}).
    end while
end for
```

**Theorem 3.1.** *Let $f_1, \ldots, f_T$ be the sequence of loss functions presented to Algorithm 1, satisfying Assumption 2.1. Then:*

*(i) The $w$-local regret incurred satisfies*

$$\mathfrak{R}_w(T) \le \left(\delta + 2L\right)^2 \frac{T}{w^2}.$$

*(ii) The total number of gradient steps $\tau$ taken by Algorithm 1 satisfies*

$$\tau \le \frac{M}{\delta^2\left(\eta - \frac{\beta\eta^2}{2}\right)} \cdot \left(2Tw + w^2\right).$$

*Proof of (i).* We note that Algorithm 1 will only play an iterate $x_t$ if $\|\nabla_{\mathcal{K},\eta}F_{t-1,w}\| \le \delta/w$. (Note that at $t = 1$, $F_{t-1,w}$ is zero.) Let $h_t(x) = \frac{1}{w}\left(f_t(x) - f_{t-w}(x)\right)$, which is $\frac{2L}{w}$-Lipschitz. Then, for each $1 \le t \le T$ we have a bound on each cost

$$\begin{aligned}\|\nabla_{\mathcal{K},\eta}F_{t,w}(x_t)\|^2 &= \|\nabla_{\mathcal{K},\eta}\left[F_{t,w-1} + h_t(x)\right](x_t)\|^2 \\ &\le \left(\|\nabla_{\mathcal{K},\eta}F_{t,w-1}\| + \|\nabla h_t(x_t)\|\right)^2 \\ &\le \left(\frac{\delta}{w} + \frac{2L}{w}\right)^2 = \frac{(\delta+2L)^2}{w^2},\end{aligned}$$

where the first inequality follows from Proposition 2.4. Summing over all $t$ gives the desired result. □

*Proof of (ii).* First, we require an additional property of the projected gradient.

**Lemma 3.2.** *Let $\mathcal{K} \in \mathbb{R}^n$ be a closed convex set, and let $\eta > 0$. Suppose $f : \mathcal{K} \to \mathbb{R}$ is differentiable. Then, for any $x \in \mathbb{R}$,*

$$\langle \nabla f(x), \nabla_{\mathcal{K},\eta} f(x) \rangle \geq \|\nabla_{\mathcal{K},\eta} f(x)\|^2.$$

*Proof.* Let $u = x - \eta \nabla f(x)$ and $u' = \Pi_{\mathcal{K}}[u]$. Then,

$$\begin{aligned} &\frac{\langle \nabla f(x), \nabla_{\mathcal{K},\eta} f(x) \rangle}{\eta^2} - \frac{\|\nabla_{\mathcal{K},\eta} f(x)\|^2}{\eta^2} \\ &= \langle u - x, u' - x \rangle - \langle u' - x, u' - x \rangle \\ &= \langle u - u', u' - x \rangle \geq 0, \end{aligned}$$

where the last inequality follows by the generalized Pythagorean theorem. □

For $2 \leq t \leq T$, let $\tau_t$ be the number of gradient steps taken in the outer loop at iteration $t-1$, in order to compute the iterate $x_t$. For convenience, define $\tau_1 = 0$. We establish a progress lemma during each gradient descent epoch:

**Lemma 3.3.** *For any $2 \leq t \leq T$,*

$$F_{t-1}(x_t) - F_{t-1}(x_{t-1}) \leq -\tau_t \left(\eta - \frac{\beta\eta^2}{2}\right) \frac{\delta^2}{w^2}.$$

*Proof.* Consider a single iterate $z$ of the inner loop, and the next iterate $z' := z - \eta \nabla_{\mathcal{K},\eta} F_{t-1}(z)$. We have, by $\beta$-smoothness of $F_{t-1}$,

$$\begin{aligned} &F_{t-1}(z') - F_{t-1}(z) \leq \langle \nabla F_{t-1}(z), z' - z \rangle + \frac{\beta}{2}\|z' - z\|^2 \\ &= -\eta \langle \nabla F_{t-1}(z), \nabla_{\mathcal{K},\eta} F_{t-1}(z) \rangle + \frac{\beta\eta^2}{2}\|\nabla_{\mathcal{K},\eta} F_{t-1}(z)\|^2. \end{aligned}$$

Thus, by Lemma 3.2,

$$F_{t-1}(z') - F_{t-1}(z) \leq -\left(\eta - \frac{\beta\eta^2}{2}\right)\|\nabla_{\mathcal{K},\eta} F_{t-1}(z)\|^2.$$

The algorithm only takes projected gradient steps when $\|\nabla_{\mathcal{K},\eta} F_{t-1}(z)\| \geq \delta/w$. Summing across all $\tau_t$ consecutive iterations in the epoch yields the claim. □

To complete the proof of the theorem, we write the telescopic sum (understanding $F_0(x_0) = 0$):

$$\begin{aligned} F_T(x_T) &= \sum_{t=1}^{T} F_t(x_t) - F_{t-1}(x_{t-1}) \\ &= \sum_{t=1}^{T} F_{t-1}(x_t) - F_{t-1}(x_{t-1}) + f_t(x_t) - f_{t-w}(x_t) \\ &\leq \sum_{t=2}^{T} [F_{t-1}(x_t) - F_{t-1}(x_{t-1})] + \frac{2MT}{w}. \end{aligned}$$

Using Lemma 3.3, we have

$$F_T(x_T) \leq \frac{2MT}{w} - \frac{\left(\eta - \frac{\beta\eta^2}{2}\right)\delta^2}{w^2} \cdot \sum_{t=1}^{T} \tau_t,$$

whence

$$\begin{aligned} \tau = \sum_{t=1}^{T} \tau_t &\leq \frac{w^2}{\delta^2\left(\eta - \frac{\beta\eta^2}{2}\right)} \cdot \left(\frac{2MT}{w} - F_T(x_T)\right) \\ &\leq \frac{M}{\delta^2\left(\eta - \frac{\beta\eta^2}{2}\right)} \cdot \left(2Tw + w^2\right), \end{aligned}$$

as claimed. □

Setting $\eta = 1/\beta$ and $\delta = L$ gives the asymptotically optimal local regret bound, with $O(Tw)$ time-averaged gradient steps (and thus $O(Tw^2)$ individual gradient oracle calls). We further note that in the case where $\mathcal{K} = \mathbb{R}^n$, one can replace the gradient descent subroutine (the inner loop) with non-convex SVRG (Allen-Zhu & Hazan, 2016), achieving a complexity of $O(Tw^{5/3})$ gradient oracle calls.

## 4. Implications for Offline and Stochastic Non-Convex Optimization

In this section, we discuss the ways in which our online framework generalizes the offline and stochastic versions of non-convex optimization – that any algorithm achieving a small value of $\mathfrak{R}_w(T)$ efficiently finds a point with small gradient in these settings. For convenience, for $1 \leq t \leq t' \leq T$, we denote by $\mathcal{D}_{[t,t']}$ the uniform distribution on time steps $t$ through $t'$ inclusive.

### 4.1. Offline non-convex optimization

For offline optimization on a fixed non-convex function $f : \mathcal{K} \to \mathbb{R}$, we demonstrate that a bound on local regret translates to convergence. In particular, using Algorithm 1 one finds a point $x \in \mathcal{K}$ with $\|\nabla_{\mathcal{K},\eta} f(x)\|^2 \leq \varepsilon$ while making $O\left(\frac{1}{\varepsilon}\right)$ calls to the gradient oracle, matching the best known result for the convergence of gradient-based methods.

**Corollary 4.1.** *Let $f : \mathcal{K} \to \mathbb{R}$ satisfy Assumption 2.1. When online algorithm $\mathcal{A}$ is run on a sequence of $T$ identical loss functions $f(x)$, it holds that for any $1 \leq w < T$,*

$$\mathbb{E}_{t \sim \mathcal{D}_{[w,T]}} \|\nabla_{\mathcal{K},\eta} f(x_t)\|^2 \leq \frac{\mathfrak{R}_w(\mathcal{A})}{T - w}.$$

*In particular, Algorithm 1, with parameter choices $T = 2w, \eta = \frac{1}{\beta}, \delta = L$, and $w = (\delta + 2L)\sqrt{\frac{2}{\varepsilon}}$, yields*

$$\mathbb{E}_{t \sim \mathcal{D}_{w,T}} \|\nabla_{\mathcal{K},\eta} f(x_t)\|^2 \leq \varepsilon.$$

*Furthermore, the algorithm makes $O\left(\frac{1}{\varepsilon}\right)$ calls to the gradient oracle in total.*

*Proof.* Since $f_t(x) = f(x)$ for all $t$, it follows that $F_{t,w}(x) = f(x)$ for all $t \geq w$. As a consequence, we have

$$\mathbb{E}_{t\sim\mathcal{D}_{w,t}}\|\nabla_{\mathcal{K},\eta}f(x_t)\|^2 \leq \frac{1}{T-w}\sum_{t=1}^{T}\|\nabla_{\mathcal{K},\eta}f(x_t)\|^2 \leq \frac{\mathfrak{R}_w(\mathcal{A})}{T-w}.$$

With the stated choice of parameters, Theorem 3.1 guarantees that

$$\mathbb{E}_{t\sim\mathcal{D}_{w,t}}\|\nabla_{\mathcal{K},\eta}f(x_t)\|^2 \leq \frac{\varepsilon}{2}\cdot\frac{T}{T-w} = \varepsilon.$$

Also, since the loss functions are identical, the execution of line 7 of Algorithm 1 requires exactly one call to the gradient oracle at each iteration. This entails that the total number of gradient oracle calls made in the execution is $O(Tw + w^2) = O(\frac{1}{\varepsilon})$. □

### 4.2. Stochastic non-convex optimization

We examine the way in which our online framework captures stochastic non-convex optimization of a fixed function $f : \mathbb{R}^n \to \mathbb{R}$, in which an algorithm has access to a noisy *stochastic gradient* oracle $\widetilde{\nabla} f(x)$. We note that the reduction here will only apply in the unconstrained case; it becomes challenging to reason about the projected gradient under noisy information. From a local regret bound, we recover a stochastic algorithm with oracle complexity $O\left(\frac{\sigma^4}{\varepsilon^2}\right)$. We note that this black-box reduction recovers an optimal convergence rate in terms of $\varepsilon$, but not $\sigma^2$.

In the setting, the algorithm must operate on the noisy estimates of the gradient as the feedback. In particular, for any $f_t$ that the adversary chooses, the learning algorithm is supplied with a stochastic gradient oracle for $f_t$. The discussion in the preceding sections may be viewed as a special case of this setting with $\sigma = 0$. We list the assumptions we make on the stochastic gradient oracle, which are standard:

**Assumption 4.2.** *We assume that each call to the stochastic gradient oracle yields an i.i.d. random vector $\widetilde{\nabla} f(x)$ with the following properties:*

*(i) Unbiased:* $\mathbb{E}\left[\widetilde{\nabla} f(x)\right] = \nabla f(x)$.

*(ii) Bounded variance:* $\mathbb{E}\left[\|\widetilde{\nabla} f(x) - \nabla f(x)\|^2\right] \leq \sigma^2$.

When an online algorithm incurs small local regret in expectation, it has a convergence guarantee in offline stochastic non-convex optimization:

**Proposition 4.3.** *Let $1 \leq w < T$. Suppose that online algorithm $\mathcal{A}$ is run on a sequence of $T$ identical loss functions $f(x)$ satisfying Assumption 2.1, with identical stochastic gradient oracles satisfying Assumption 4.2. Sample $t \sim \mathcal{D}_{[w,T]}$. Then, over the randomness of $t$ and the oracles,*

$$\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \leq \frac{\mathbb{E}\left[\mathfrak{R}_w(\mathcal{A})\right]}{T-w}.$$

*Proof.* Observe that

$$\mathbb{E}_{t\sim\mathcal{D}_{[w,T]}}\left[\|\nabla f(x_t)\|^2\right] \leq \frac{\sum_{t=1}^{T}\|\nabla f(x_t)\|^2}{T-w} \leq \frac{\mathfrak{R}_w(\mathcal{A})}{T-w}.$$

The claim follows by taking the expectation of both sides, over the randomness of the oracles. □

For a concrete online-to-stochastic reduction, we consider Algorithm 2, which exhibits such a bound on expected local regret.

**Algorithm 2** Time-smoothed online gradient descent with stochastic gradient oracles

1: Input: learning rate $\eta > 0$, window size $w \geq 1$.
2: Set $x_1 = 0 \in \mathbb{R}^n$ arbitrarily.
3: **for** $t = 1, \ldots, T$ **do**
4: Predict $x_t$. Observe the cost function $f_t : \mathbb{R}^n \to \mathbb{R}$.
5: Update $x_{t+1} := x_t - \frac{\eta}{w}\sum_{i=0}^{w-1}\widetilde{\nabla} f_{t-i}(x_t)$.
6: **end for**

**Theorem 4.4.** *Let $f_1, \ldots, f_t$ satisfy Assumption 2.1. Then, Algorithm 2, with access to stochastic gradient oracles $\{\widetilde{\nabla} f_t(x_t)\}$ satisfying Assumption 4.2, and a choice of $\eta = \frac{1}{\beta}$, guarantees*

$$\mathbb{E}\left[\mathfrak{R}_w(T)\right] \leq \left(8\beta M + \sigma^2\right)\frac{T}{w}.$$

*Furthermore, Algorithm 2 makes a total of $O(Tw)$ calls to the stochastic gradient oracles.*

Using this expected local regret bound in Proposition 4.3, we obtain the reduction claimed at the beginning of the section:

**Corollary 4.5.** *Algorithm 2, with parameter choices $w = \frac{12M\beta+2\sigma^2}{\varepsilon}$, $T = 2w$, and $\eta = \frac{1}{\beta}$, yields*

$$\mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \leq \varepsilon.$$

*Furthermore, the algorithm makes $O\left(\frac{\sigma^4}{\varepsilon^2}\right)$ stochastic gradient oracle calls in total.*

## 5. An Efficient Algorithm with Second-Order Guarantees

We note that by modifying Algorithm 1 to exploit second-order information, our online algorithm can be improved to play approximate first-order critical points which are also locally almost convex. This entails replacing the gradient descent epochs with a cubic-regularized Newton method (Nesterov & Polyak, 2006; Agarwal et al., 2016a).

In this setting, we assume that we have access to each $f_t$ through a value, gradient, and Hessian oracle. That is, once we have observed $f_t$, we can obtain $f_t(x)$, $\nabla f_t(x)$, and $\nabla^2 f_t(x)$ for any $x$. Let $\mathsf{MinEig}(A)$ be the minimum (eigenvalue, eigenvector) pair for matrix $A$. As is standard for offline second-order algorithms, we must add the following additional smoothness restriction:

**Assumption 5.1.** *$f_t$ is twice differentiable and has an $L_2$-Lipschitz Hessian:*

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \le L_2 \|x - y\|.$$

Additionally, we consider only the unconstrained case where $\mathcal{K} = \mathbb{R}^n$; the second-order optimality condition is irrelevant when the gradient does not vanish at the boundary of $\mathcal{K}$.

The second-order Algorithm 3 uses the same approach as in Algorithm 1, but terminates each epoch under a stronger approximate-optimality condition. We define

$$\Phi_t(x) := \max\left\{\|\nabla F_t(x)\|^2, -\frac{4\beta}{3L_2^2}\cdot\lambda_{\min}(\nabla^2 F_t(x))^3\right\},$$

so that the quantity $\sum_{t=1}^{T}\Phi_t(x_t)$ is termwise lower bounded by the costs in $\mathfrak{R}_w(T)$, but penalizes local concavity.

We characterize the convergence and oracle complexity properties of this algorithm:

**Theorem 5.2.** *Let $f_1, \dots, f_T$ be the sequence of loss functions presented to Algorithm 3, satisfying Assumptions 2.1 and 5.1. Choose $\delta = \beta$. Then, for some constants $C_1, C_2$ in terms of $M, L, \beta, L_2$:*

*(i) The iterates $\{x_t\}$ produced by Algorithm 3 satisfy*

$$\sum_{t=1}^{T}\Phi_t(x_t) \le C_1 \cdot \frac{T}{w^2}.$$

*(ii) The total number of iterations $\tau$ of the inner loop taken by Algorithm 3 satisfies*

$$\tau \le C_2 \cdot Tw^2.$$

**Algorithm 3** Time-smoothed online Newton method

Input: window size $w \ge 1$, tolerance $\delta > 0$.
Set $x_1 \in \mathcal{K}$ arbitrarily.
**for** $t = 1, \dots, T$ **do**
  Predict $x_t$. Observe the cost function $f_t : \mathbb{R}^n \to \mathbb{R}$.
  Initialize $x_{t+1} := x_t$.
  **while** $\Phi_t(x_{t+1}) > \delta^3/w^3$ **do**
    Update $x_{t+1} := x_{t+1} - \frac{1}{\beta}\nabla F_{t,w}(x_{t+1})$.
    Let $(\lambda, v) := \mathsf{MinEig}\left(\nabla^2 F_{t,w}(x_{t+1})\right)$.
    **if** $\lambda < 0$ **then**
      Flip the sign of $v$ so that $\langle v, \nabla F_{t,w}(x_{t+1})\rangle \le 0$.
      Compute $y_{t+1} := x_t + \frac{2\lambda}{L_2}v$.
      **if** $F_{t,w}(y_{t+1}) < F_{t,w}(x_{t+1})$ **then**
        Set $x_{t+1} := y_{t+1}$.
      **end if**
    **end if**
  **end while**
**end for**

*Proof of (i).* For each $1 \le t \le T$, we have

$$\Phi_{t-1}(x_t) \le \frac{\delta^3}{w^3}.$$

Let $h_t(x) := \frac{1}{w}\left(f_t(x) - f_{t-w}(x)\right)$. Then, since $h_t(x)$ is $\frac{2L}{w}$-Lipschitz and $\frac{2\beta}{w}$-smooth,

$$\begin{aligned}\Phi_t(x_t) &= \max\Big\{\|\nabla F_{t-1}(x_t) + \nabla h_t(x_t)\|^2, \\ &\qquad -\frac{4\beta}{3L_2^2}\cdot\lambda_{\min}(\nabla^2 F_t(x_t) + \nabla^2 h_t(x_t))^3\Big\} \\ &\le \max\left\{\left(\frac{\delta^{3/2}}{w^{3/2}} + \frac{2L}{w}\right)^2, \frac{4\beta}{3L_2^2}\cdot\left(\frac{\delta}{w} + \frac{2\beta}{w}\right)^3\right\},\end{aligned}$$

which is bounded by $C_1/w^2$, for some $C_1(M, L, \beta, L_2^2)$. The claim follows by summing this inequality across all $1 \le t \le T$. □

*Proof of (ii).* We first show the following progress lemma:

**Lemma 5.3.** *Let $z, z'$ be two consecutive iterates of the inner loop in Algorithm 3 during round $t$. Then,*

$$F_t(z') - F_t(z) \le -\frac{\Phi_t(z)}{2\beta}.$$

*Proof.* Let $u$ denote the step $z' - z$. Let $g := \nabla F_t(z)$, $H := \nabla^2 F_t(z)$, and $(\lambda, v) := \mathsf{MinEig}(H)$.

Suppose that at time $t$, the algorithm takes a gradient step, so that $u = g/\beta$. Then, by second-order smoothness of $F_t$, we have

$$F_t(z') - F_t(z) \le \langle g, u\rangle + \frac{\beta}{2}\|u\|^2 = -\frac{1}{2\beta}\|g\|^2.$$

Supposing instead that the algorithm takes a second-order step, so that $u = \pm\frac{2\lambda}{L_2}v$ (whichever sign makes $\langle g, u\rangle \le 0$), the third-order smoothness of $F_t$ implies

$$\begin{aligned} F_t(z') - F_t(z) &\le \langle g, u\rangle + \frac{1}{2}u^T H_t u + \frac{L_2}{6}\|u\|^3 \\ &= \langle g, u\rangle + \frac{\lambda}{2}\|u\|^2 + \frac{L_2}{6}\|u\|^3 \\ &\le \frac{2\lambda^3}{3L_2^2} = \frac{1}{2\beta}\cdot\frac{4\beta\lambda^3}{3L_2^2}. \end{aligned}$$

The lemma follows due to the fact that the algorithm takes the step that gives a smaller value of $F_t(z')$. □

The rest of the proof follows the same structure as that for part (ii) of Theorem 3.1: summing Lemma 5.3 implies a statement analogous to Lemma 3.3, which we telescope over all epochs. For sake of completeness, we give the proof in the appendix. □

## 6. A Solution Concept for Non-Convex Games

Finally, we discuss an application of our regret minimization framework to learning in $k$-player $T$-round iterated games with smooth, non-convex payoff functions. Suppose that each player $i \in [k]$ has a fixed decision set $\mathcal{K}_i \subset \mathbb{R}^n$, and a fixed payoff function $f_i : \mathcal{K} \to \mathbb{R}$ satisfies Assumption 2.1 as before. Here, $\mathcal{K}$ denotes the Cartesian product of the decision sets $\mathcal{K}_i$: each payoff function is defined in terms of the choices made by every player.

In such a game, it is natural to consider the setting where players will only consider small *local* deviations from their strategies. This is a natural setting, which models risk aversion. This setting lends itself to the notion of a local equilibrium, to replace the stronger condition of Nash equilibrium: a joint strategy in which no player encounters a large gradient on her utility. However, finding an approximate local equilibrium in this sense remains computationally intractable when the utility functions are non-convex.

Using the idea time-smoothing, we formulate a tractable relaxed notion of local equilibrium, defined over some time window $w$. Intuitively, this definition captures a state of an iterated game in which each player examines the past $w$ actions played, and no player can make small deviations to improve the average performance of her play against her opponents' historical play. We formulate this solution concept as follows:

**Definition 6.1** (Smoothed local equilibrium)**.** *Fix some $\eta > 0, w \ge 1$. Let $\left\{f_i(x^1, \ldots, x^k) : \mathcal{K} \to \mathbb{R}\right\}_{i=1}^k$ be the payoff functions for a $k$-player iterated game. A joint strategy $(x_t^1, \ldots, x_t^k)$ is an $\varepsilon$-approximate $(\eta, w)$-*smoothed local equilibrium *with respect to past iterates $\left\{(x_{t-j}^1, \ldots, x_{t-j}^k)\right\}_{j=0}^{w-1}$ if, for every player $i \in [k]$,*

$$\left\|\nabla_{\mathcal{K},\eta}\left[\frac{\sum_{j=0}^{w-1}\tilde{f}_{i,t-j}}{w}\right](x_t^i)\right\| \le \varepsilon,$$

*where*

$$\tilde{f}_{i,t'}(x) \stackrel{def}{=} f_i(x_{t'}^1, \ldots, x_{t'}^{i-1}, x, x_{t'}^{i+1}, \ldots, x_{t'}^k).$$

To achieve such an equilibrium efficiently, we use Algorithm 4, which runs $k$ copies of any online algorithm that achieves a $w$-local regret bound for some $\eta > 0$.

**Algorithm 4** Time-smoothed game simulation

1: Input: convex decision sets $\mathcal{K}_1, \ldots, \mathcal{K}_k \subseteq \mathbb{R}^n$, payoff functions $f_i : (\mathcal{K}_1, \ldots, \mathcal{K}_k) \to \mathbb{R}$, online algorithm $\mathcal{A}$, window size $1 \le w < T$.
2: Initialize $k$ copies $(\mathcal{A}_1, \ldots, \mathcal{A}_k)$ of $\mathcal{A}$ with window size $w$, where each $\mathcal{A}_i$ plays on decision set $\mathcal{K}_i$.
3: **for** $t = 1, \ldots, T$ **do**
4: Each $\mathcal{A}_i$ outputs $x_t^i$.
5: Show each $\mathcal{A}_i$ the online loss function

$$f_{i,t}(x) := -f_i(x_t^1, \ldots, x_t^{i-1}, x, x_t^{i+1}, \ldots, x_t^k).$$

6: **end for**

We show this meta-algorithm yields a subsequence of iterates that satisfy our solution concept, with error parameter dependent on the local regret guarantees of each player:

**Theorem 6.2.** *For some $t$ such that $w \le t \le T$, the joint strategy $(x_t^1, \ldots, x_t^k)$ produced by Algorithm 4 is an $\varepsilon$-approximate $(\eta, w)$-smoothed local equilibrium with respect to $\left\{(x_{t-j}^1, \ldots, x_{t-j}^k)\right\}_{j=0}^{t-1}$, where*

$$\varepsilon = \sqrt{\sum_{i=1}^k \frac{\mathfrak{R}_{w,\mathcal{A}_i}(T)}{T-w}}.$$

The proof, which we give in the appendix, follows from the same method as for the reductions in Section 4, after summing the regret bounds from each $\mathcal{A}_i$.

## 7. Concluding Remarks

We have described how to extend the theory of online learning to non-convex loss functions, while permitting *efficient* algorithms. Our definitions give rise to efficient online and stochastic non-convex optimization algorithms that converge to local optima of first and second order. We give a game theoretic solution concept which we call local equilibrium, which, in contrast to existing solution concepts such as Nash equilibrium, is efficiently attainable in any non-convex game.

## Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1523815. The authors gratefully acknowledge Naman Agarwal, Brian Bullins, Matt Weinberg, and Yi Zhang for helpful discussions.

## References

Agarwal, Naman, Allen-Zhu, Zeyuan, Bullins, Brian, Hazan, Elad, and Ma, Tengyu. Finding approximate local minima for nonconvex optimization in linear time. *arXiv preprint arXiv:1611.01146*, 2016a.

Agarwal, Naman, Bullins, Brian, and Hazan, Elad. Second order stochastic optimization for machine learning in linear time. *arXiv preprint arXiv:1602.03943*, 2016b.

Allen-Zhu, Zeyuan and Hazan, Elad. Variance reduction for faster non-convex optimization. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 699–707, 2016.

Arora, Sanjeev, Hazan, Elad, and Kale, Satyen. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(6): 121–164, 2012. doi: 10.4086/toc.2012.v008a006. URL http://www.theoryofcomputing.org/articles/v008a006.

Blum, A. and Mansour, Y. From external to internal regret. In *COLT*, pp. 621–636, 2005.

Carmon, Yair, Duchi, John C., Hinder, Oliver, and Sidford, Aaron. Accelerated methods for non-convex optimization. *arXiv preprint 1611.00756*, 2016.

Cesa-Bianchi, Nicolò and Lugosi, Gábor. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Cover, Thomas. Universal portfolios. *Math. Finance*, 1(1): 1–19, 1991.

Duchi, John, Hazan, Elad, and Singer, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

Erdogdu, Murat A and Montanari, Andrea. Convergence rates of sub-sampled newton methods. In *Advances in Neural Information Processing Systems*, pp. 3034–3042, 2015.

Freund, Yoav and Schapire, Robert E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997.

Ghadimi, Saeed and Lan, Guanghui. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Hart, Sergiu and Mas-Colell, Andreu. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

Hazan, Elad. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2 (3-4):157–325, 2016. ISSN 2167-3888. doi: 10.1561/2400000013. URL http://dx.doi.org/10.1561/2400000013.

Hazan, Elad and Kale, Satyen. Computational equivalence of fixed points and no regret algorithms, and convergence to equilibria. In Platt, J.c., Koller, D., Singer, Y., and Roweis, S. (eds.), *Advances in Neural Information Processing Systems 20*, pp. 625–632. MIT Press, Cambridge, MA, 2007. URL http://books.nips.cc/papers/files/nips20/NIPS2007_0695.pdf.

Nesterov, Yurii. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004.

Nesterov, Yurii and Polyak, Boris T. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Shalev-Shwartz, Shai. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

Vovk, Volodimir G. Aggregating strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, COLT '90, pp. 371–386, 1990.