A Study on Continuous Optimization Schemes and Guarantees for Machine Learning

Jian Vora, 170100026
Department of Electrical Engineering
Indian Institute of Technology, Bombay
Mumbai, MH 400076
jianvora@iitb.ac.in

Abstract

This document serves as a literature survey for three papers which are mainly related to continuous optimization for machine learning. Each paper is summarized as a different section of this report. The first paper proves that overparameterized deep neural networks can be trained using simple first-order methods like gradient descent with random intializations to achieve zero training error [1]. The second paper shows that Q-learning is provably almost efficient in terms of sample complexity as compared to model-based approaches [2]. The third paper couples gradient and mirror descent to design faster algorithms. It turns out that this class of algorithms can be used to get a clean proof for Nesterov's accelerated gradient method and also extend to a variety of different problems [3].

1 A Convergence Theory for Deep Learning via Over-Parameterization

1.1 Introduction

Most of machine learning essentially boils down to minimizing some loss function where predictions come from a parameterized function F_θ. To make things easier, we assume a specific structure for F_θ by stacking linear layers followed by point-wise non-linearities. Optimising these deep neural networks is a non-convex optimization problem with very little theoretical guarantees. Most simple gradient based algorithms get stuck at critical points(saddle points, local minimas) but it has been shown empirically that for a variety of tasks and network architectures, training deep neural networks never encounters from these spurious points. Most theoretical understanding has been limited to shallow single hidden layer networks whereas most models used in practice are many layers deep. This work shows that overparameterized deep networks can be trained using first-order methods like (stochastic) gradient descent to drive the training error to zero. This is not surprising from a capacity point of view as deep nets have been shown to achieve random labellings for any N input samples. However, this analysis is mainly from an optimization perspective where we can provably show that simple SGD can actually find the correct set of parameters (weights). In terms of network architectures, the theory applies to fully-connected neural networks, convolutional and residual neural networks. In the subsequent subsections, we shall review some prior work, state the main results of the paper and present some important lemmas which can be used to derive the results.

1.2 Prior Work

In this subsection, we outline major lines along which convergence in deep learning has progressed.

1. One line of work deals with studying linear networks without any activation function. Some similar area of work is on linear dynamical systems which can be viewed as the linear version of recurrent neural networks or reinforcement learning [4], [5].
2. There exist convergence guarantees on shallower networks like single hidden layer neural networks which achieve 100% classification accuracy on the training data. It is suggested that overparameterization as is a way to avoid spurious minima. This has also been extended to single hidden layers CNNs [6], [8].
3. For deeper networks, most of the work focusses on assuming some simple distribution from which inputs are drawn like a Gaussian. In general, learning a neural network without any assumptions is shown to be NP-complete [7].

1.3 Problem Statement and Important Takeaway Results

This work considers a supervised setting where data samples are in the form of tuples $(x_i, y_i), 1 \leq i \leq n$ and we assume a degenerate dataset which means that we cannot have the case where $x_i = x_j, y_i \neq y_j$. Denote δ as the minimum distance between two training data points, i.e., $\forall i, j, i \neq j, \|x_i - x_j\| \geq \delta$. Consider a L-layer neural network with each hidden layer consisting of m nodes followed by a ReLU activation. In such a situation, the authors show that -

1. For overparameterized networks where $m > \text{poly}(n, L, \delta^{-1})$, starting from random Gaussian initialized weights, GD and SGD achieve ϵ error on the training data for l_2 regression task in atmost $T = \text{poly}(n, L, \delta^{-1})\text{log}(1/\epsilon)$ iterations.
2. The above even holds for multi-class classification task where we achieve 100% classification accuracy in atmost $T = \text{poly}(n, L, \delta^{-1})$ iterations. The result also applied to other Lipschitz-smooth loss functions and different architechtures like CNNs and ResNets.

Consider a l_2 loss function $F(\mathbf{W}) = \sum_i F_i(\mathbf{W})$ and the let dimensionality of the output be denoted by d. More formally, the above theorems can be stated in the following manner -

1. Suppose $m > \Omega(\text{poly}(n, L, \delta^{-1}).d)$. Starting from random initialization, with probability atleast $1 - e^{\Omega(log^2 m)}$, gradient descent with learning rate $\eta = \Theta(\frac{d\delta}{\text{poly}(n,L).m})$ finds a point $F(\mathbf{W}) \leq \epsilon$ in $T = \Omega(\frac{\text{poly}(n,L)}{\delta^2}.\text{log}(\epsilon^{-1}))$ iterations.
2. For training using SGD, let b denote and batch size and as earlier let $m > \Omega(\text{poly}(n, L, \delta^{-1}).d)$. Then with probability atleast $1 - e^{\Omega(log^2 m)}$, and learning rate $\eta = \Theta(\frac{bd\delta}{\text{poly}(n,L).mlog^2 m})$, SGD finds a point $F(\mathbf{W}) \leq \epsilon$ in $T = \Omega(\frac{\text{poly}(n,L)log^2 m}{\delta^2 b}.log(\epsilon^{-1}))$ iterations. Both GD and SGD achieve a linear convergence rate.

1.4 Major Lemmas and Proof Sketch

The above convergence results can be proved if we prove two main theorems which help us answering the following important questions which will form the basis of the proof sketch -

1. For points near random initialization, are there any gradient bounds?
2. Even if the above holds true, what guarantee do we have that the objective function decreases even if we follow the negative gradient descent. Classically, this holds true for smooth functions but neural nets with ReLU activations aren't smooth. But is there still any similar notion of smoothness? Does overparameterization help?

We now state the theorem answering the following two questions which gives us some hope of finding the global minima of $F(\mathbf{W})$. These are the following -

1. **Theorem 1:** Let $\mathbf{W}^{(0)}$ be the initialization (drawn from a gaussian) of the weights to be estimated. Over this randomness of choosing the initial parameters with probability $\geq 1 - e^{-\Omega(m/\text{poly}(n,L,\delta^{-1}))}$ and for any general weight matrix $\mathbf{W}$ in the optimization satisfying $\|\mathbf{W} - \mathbf{W}^{(0)}\| \leq \frac{1}{\text{poly}(n,L,\delta^{-1})}$, we have the following two bounds:

$$\|\nabla F(\mathbf{W})\|_F^2 \leq \mathcal{O}(F(\mathbf{W}).\frac{Lnm}{d}), \|\nabla F(\mathbf{W})\|_F^2 \geq \Omega(F(\mathbf{W}).\frac{\delta m}{dn^2})$$

2. **Theorem 2:** With probability $\geq 1 - e^{-\Omega(m/\text{poly}(L, logm))}$ over the randomness of every $\mathbf{W}^{(0)}$, and for $\hat{\mathbf{W}}, \tilde{\mathbf{W}} \in (\mathbb{R}^{m \times m})^L$ (for now the weights of the first and last layers can be assumed to be fixed, the theory can be easily extended in that case as well), if

$$\|\hat{\mathbf{W}} - \mathbf{W}^{(0)}\|_2 \leq \frac{1}{\text{poly(L, log m)}} \text{ and } \|\tilde{\mathbf{W}}\|_2 \leq \frac{1}{\text{poly(L, log m)}} \text{ we have}$$

$$F(\hat{\mathbf{W}}+\tilde{\mathbf{W}}) \leq F(\hat{\mathbf{W}})+\langle \nabla F(\hat{\mathbf{W}}), \tilde{\mathbf{W}}\rangle+\mathcal{O}(\frac{nL^2m}{d})\|\tilde{\mathbf{W}}\|_2^2+\frac{\text{poly}(L)\sqrt{\text{nmlogm}}}{\sqrt{d}}\|\tilde{\mathbf{W}}\|_2 F(\hat{\mathbf{W}})^{1/2}$$

Given below are interpretations of the above theorems in words which shall help in the actual proof:

1. Theorem 1 suggests that at as long as the objective is large (weights far from actual), the norm of the gradient is also large. *This means if we are sufficiently close to random initialization, there is no critical point of any order*.
2. In Theorem 2, one quickly notices is that the first order term involving $\|\tilde{\mathbf{W}}\|_2$ is the additional term in the conventional smoothness definition. It turns out that as m is increased(network is overparameterized), then the contribution of the last term keeps on diminishing and then function F becomes *almost* smooth or semi-smooth.
3. One important thing to note is that the above theorems hold when the spectral norm of the difference in weights with the initialization is small. Although the bound of $1/\text{poly}(n, L, \delta^{-1})$ seems small, it is large enough to change the outputs and fit the training data. In practice, such strong bounds do not hold true but still the results of the theorems have been shown to hold true empirically on image classification tasks.

Proof Sketch:

1. **Properties at random initialization:** Define $h_{i,l} = \text{ReLU}(\mathbf{W}_l h_{i,l-1})$ as the activation after lth layer for ith input sample. If each element of $\mathbf{W}_l \sim \mathcal{N}(0, 2/m)$, then for any y, $\|\mathbf{W}y\|^2 \approx 2$ and thus $\|h_{i,l}\|^2 \approx 1$ as ReLU will kill off approximately half entries. *This implies that the forward propagation will neither vanish nor explode*. We can also bound the backward propagation. Define $\mathbf{D}_{i,l(k,k)} = \mathbf{I}_{(\mathbf{W}_l h_{i,l-1})_k > 0}$ and $\mathbf{B} \in \mathbb{R}^{m \times d}$ as the final layer weights, then $\|\mathbf{B}\mathbf{D}_{\mathbf{i},\mathbf{L}}\mathbf{W}_{\mathbf{L}}...\mathbf{D}_{\mathbf{i},\mathbf{a}}\mathbf{W}_{\mathbf{a}}\| \leq \mathcal{O}(\sqrt{m/d})$ for any $a \in \{1, 2, ..., L-1\}$. It can be further shown that if $\|x_i - x_j\| \geq \delta$, then $\|h_{i,l} - h_{j,l}\| \geq \Omega(\delta)$.
2. **Stability after adversarial perturbation:** This involves showing that any $\mathbf{W}$ stays close to $\mathbf{W}^{(0)}$, i.e. $\|\mathbf{W}_l - \mathbf{W}_l^{(0)}\| \leq \omega$ for every l and $\omega \leq 1/\text{poly}(L)$. This is known as the forward stability and the number of sign changes $\|\mathbf{D}_{i,l} - \mathbf{D}_{i,l}^{(0)}\|_0$ is atmost $\mathcal{O}(m\omega^{2/3}L)$ and the perturbation amount $\|h_{i,l} - h_{i,l}^{(0)}\| \leq \mathcal{O}(\omega L^{5/2})$.
3. **Gradient Bound:** The most crucial part is the show the gradient lower bounds. For each samples x_i, it is easy to show using the fact that due to the nature of initialization, $\|h_{i,l}^{(0)}\| \approx 1$. However, the sum over n samples may not lead to the final gradient norm to be bounded (as they are not independent). However, we can use the fact that $\|h_{i,l} - h_{j,l}\| \geq \Omega(\delta)$ and show that the gradient norm remains sufficiently large.
4. **Semi-Smoothness:** If we perturb the current weights $\mathbf{W}$ by $\mathbf{W}^{'}$, then the new activations computed at this new weight will not be far off from the original activations, i.e. $\|h_{i,l} - h_{i,l}^{'}\| \geq \mathcal{O}(L^{3/2})\|\mathbf{W}^{'}\|_2$. This is easy to see as ReLU is Lipschitz continuous but it is non-trivial to show that composotion of L such functions does not lead to an exponential blow up in the final lipschitz parameter. This is where overparameterization helps.

Using the above properties, we can show that gradient descent can find ϵ-additive optimal point from the true global minima. The proof follows along the same lines of showing convergence of gradient descent on convex and smooth functions. The theory can be easily extended to stochastic gradient descent as well given the second moment of the noise term is bounded.

1.5 Further Extensions and Conclusion

1. Although the proof has been shown for the case of l_2 regression, it can be extended to any Lipschitz-smooth loss function f. Theorem 5 in the original paper provides bounds for a variety of different loss function settings.
2. Regarding network architectures, it is shown that the analysis holds even for CNNs and ResNets with a more careful choice for weight initialization (although still random). Theorem 6 and 7 in the original paper given similar bounds to what are presented above for new network architectures.

In conclusion, this paper shows that overparameterized networks can be trained to get very close to the global optimal weights using first-order methods like gradient descent. The way they argue this is by showing that the neural network is not very far from a convex and smooth function where there is overparameterization. However, this is purely from an optimization perspective and does not deal with generalization. Although overparameterization helps in getting a better optimization landscape, it suffers from overfitting and generalization on unseen data.

2 Is Q-Learning Provably Efficient?

2.1 Introduction

Reinforcement Learning is the third broad paradigm of machine learning (supervised, unsupervised being the others) which involves an agent interacting with it's environment to maximise it's long term expected rewards. Two main approaches include model-based and model-free algorithms. Model based algorithms try to learn a model of the environment while model-free algorithms directly optimise the end-task - updating the value function or the policy directly from data observed during interaction with the environment. Model-free algorithms are generally simpler and more flexible to use but it has been suspected that these are not as sample efficient as model-based methods. This has even been seen empirically but there has not been any theoretical understanding of whether model-free algorithms are really sample inefficient. This particular paper tries to answer the precise question - are model-free RL algorithms sample efficient? It studies Q-learning which a popular model-free algorithm. We shall show that for an episodic MDP setting, Q-learning with UCB exploration achieves regret $\mathcal{O}(\sqrt{H^3SAT})$ where S is the number of states, A is the number of actions, H is the number of steps per episode and T is the total number of steps. This sample efficiency matches the optimal regret that can be achieved by any model-based approach, up to a single $\sqrt{H}$ factor. All of this is done in the true RL setting which requires exploration and does not have access to any simulator which when given a state action pair outputs a distribution over next states and reward.

2.2 Prior Work

All of this work only considers the tabular setting where number of states and actions is finite. In the presence of a simulator, model-free algorithms are known to be almost as sample efficient as the best model-based algorithms. However the simulator setting does not really need any "exploration" and hence is considered easier. There exist model-based algorithms that achieve a regret of $\mathcal{O}(\sqrt{H^2SAT})$ which exactly matches with the information-theoretic lower bound [9], [10]. The downside with these is that they suffer from worst time and space compexity (which typically involves storing an entire transition matrix taking space of the order $\mathcal{O}(S^2AH)$). It is the simulator free setting which mimics any real-life RL problem. In this, it has been shown that Q-learning with $\epsilon-$greedy exploration policy may take exponentially many episodes to learn the optimal policy [12]. This suggests that perhaps we need better exploration strategies if we hope to achieve near-optimal sample complexities. We formally define model-free algorithms as follows:

Definition: *A reinforcement learning algorithm is model-free if its space complexity is always sublinear (for any T) relative to the space required to store an MDP. In episodic setting of this paper, a model-free algorithm has space complexity $o(S^2AH)$ (independent of T).*

In the model-free setting Q-learning with $\epsilon-$greedy explorations suffers from a regret of $\Omega(\min(T, A^{H/2}))$ which as stated earlier is exponential in H. [11] introduced delayed Q-learning which replaces old Q values with the average of the most recent m experiences. It achieves a regret of $\mathcal{O}(T^{4/5})$ ignoring other problem parameters like H, S, A but this is still worse than model-based algorithms which achieve a regret of the $\sqrt{T}$.

2.3 Preliminaries and Problem Formulation

The case being considered is that of a tabular episodic MDP(S, A, H, P, r) where all notations are the same as described above. P stands for the transition matrix where $P_h(.|x, a)$ gives the distribution over next states if action a is taken at state x. $r_h : S \times A \to [0, 1]$ is a deterministic reward function at step h. H is the total number of steps in each episode. A policy π is a collection of H functions $\{\pi_h : S \to A\}$ and V_h^π is the value function at step h under policy π. The Q value function is also defined similarly $\mathcal{Q}_h^\pi : S \times A \to \mathbb{R}$. It is the expected reward starting at state x and taking an initial action a. The optimal value function is defined as $V_h^*(x) = \sup_\pi V_h^\pi$. All these follow the Bellman and Bellman optimality equations with $V_{H+1}^\pi(x) = 0,\ \forall x \in S$. Finally, an agent play a total of K episodes and every time an initial state can be thought of as chosen by an adversary. Then the total expected regret (which is what we want to minimise and want bounds for) is given by:

$$\text{Regret}(K) = \sum_{k=1}^{k=K} V_1^*(x_1^k) - V_1^{\pi_k}(x_1^k)$$

2.4 Takeaway Results

1. Q-learning is *almost* sample efficient and achieves a regret of $\mathcal{O}(\sqrt{H^3SAT})$ which is just $\sqrt{H}$ worse than the information-theoretic lower bound.
2. UCB algorithm for exploration is better than $\epsilon-$greedy because it allows for better treatment of uncertainties for different states and actions.
3. It is crucial to use a learning rate $\alpha_t = \mathcal{O}(H/t)$ and not just $1/t$ as otherwise the sample complexity will come out to be exponential in H. More details in the proof sketch.

2.5 Final Algorithm and Major Theorems

The authors propose Q-learning with UCB exploration in order to achieve better sample complexity. The primary update equation is given as follows:

$$Q_h(x, a) \leftarrow (1 - \alpha_t)Q_h(x, a) + \alpha_t(r_h(x, a) + V_{h+1}(x^{'}) + b_t)$$

where $x^{'}$ is the next state, t is the counter of how many times it has visited state-action pair (x, a) at step h, b_t is the confidence bonus indicating how certain the algorithm is about current state-action pair, and α_t is a learning rate given by $\frac{H+1}{H+t}$. Note that α_t is the form $\mathcal{O}(H/t)$ which is what was mentioned in the earlier subsection. The above algorithm will henceforth be referred to as Q-learning with UCB Hoeffding exploration. Another option we had was Q-learning with UCB Bernstein exploration. Now we look into various choices we have for setting b_t which shall lead to different bounds. Before proceeding, we formally state the information-theoretic lower bound for regret:

Theorem: *For the episodic MDP problem as described in the above subsections, the expected regret for any algorithm must be at least* $\Omega(\sqrt{H^2SAT})$ *.*

The first choice of b_t is $b_t = \mathcal{O}(\sqrt{H^3\eta/t})$ where $\eta := \log(SAT/p)$ which can be thought of as just a log factor. This specification of b_t is called UCB-Hoeffding as a confidence bound for the Q value scales as $1/\sqrt{t}$ which is similar to Hoeffding-type martingale concentration inequalities. With the above setup we have the following theorem:

Theorem: (UCB-Hoeffding) *There exists an absolute constant* $c > 0$ *such that, for any* $p \in (0, 1)$*, if we choose* $b_t = c\sqrt{H^3\eta/t}$*, then with probability* $1 - p$*, the total regret of Q-learning with UCB-Hoeffding is at most* $\mathcal{O}(\sqrt{H^4SAT\eta})$*, where* $\eta := log(SAT/p)$*.*

The above is the first model-free algorithm which achieves a $\sqrt{T}$ regret without the need of a simulator. However, it is not as efficient as model-based algorithms which achieve the lower bound but it is better in the sense of requiring lesser time and space complexity. It now becomes clear that we can tune the exploration bonus b_t in a better way which might lead to a lower regret bound. The use of a Bernstein-type martingale concentration result could be sharper than the Hoeffding-type bound by an additional factor of H. The new b_t is defined as follows:

$$\beta_t \leftarrow \min\{c_1\left(\sqrt{\left(\frac{H}{t}\frac{\sigma_h-\mu_h^2}{t}+H\right)\eta}+\frac{\sqrt{H^7SA\eta}}{t}\right), c_2\sqrt{\frac{H^3\eta}{t}}\}$$

$$b_t \leftarrow \frac{\beta_t+(1-\alpha_t)\beta_{t-1}}{2\alpha_t} \text{ with } \mu_h \leftarrow \mu_h + V_{h+1} \text{ and } \sigma_h \leftarrow \sigma_h + V_{h+1}^2$$

All the above operations are done for the current state-action pairs (x_h, a_h) but are not mentioned in the updates for the sake of clarity. With this new better choice for b_t, we get the following theorem:

Theorem: (UCB-Bernstein) *For any $p \in (0,1)$, one can specify b_t so that with probability $1-p$, the total regret of Q-learning with UCB-Bernstein is at most $\mathcal{O}(\sqrt{H^3SAT\eta}+\sqrt{H^9S^3A^3\eta^2})$.*

Thus a more sophisticated choice of exploration bonus leads to a $\sqrt{H}$ decrease in the regret over the UCB-Hoeffding algorithm. The second additional term will be dominant only when T is small as compared to the other parameters as then η^2 and η won't have much difference and terms like H^9 will start to dominate. Thus, the authors have provided an algorithm which is *almost* sample efficient upto a factor of $\sqrt{H}$ from the lower bound (even in the simulator free setting).

2.6 Proof Sketch

In this subsection, we shall outline some steps required to complete the proof of the Theorem which uses UCB-Hoeffding exploration. Recall the Q-learning update equation:

$$Q_h(x,a) \leftarrow (1-\alpha_t)Q_h(x,a) + \alpha_t(r_h(x,a) + V_{h+1}(x^{'}) + b_t)$$

$$V_h^k(x) \leftarrow \min\{H, \max_{a\in A} Q_h^k(x,a)\}$$

The important part is to note the dependence of α_t on H. For convenience, define $\alpha_t^i = \alpha_i \prod_{j=i+1}^{t}(1-\alpha_j)$. According to the update equation the Q value at episode k equals a weighted average of the V values of the next states with weights $\alpha_t^1, \alpha_t^2, .., \alpha_t^t$. The choice of $\alpha_t = \frac{H+1}{H+t}$ ensures that $1/H$ fraction of the i indices have non-negligible weight. Instead of the uniform weight $\alpha_t = 1/t$ we start putting more weight on the newer version of the value function (which is certainly more accurate and we are more confident about) which will lead to variance reduction. We now state a few lemmas which are useful for the proof.

Lemma: The following properties hold for α_t^i:

1. $\frac{1}{\sqrt{t}} \leq \sum_i \frac{\alpha_t^i}{\sqrt{i}} \leq \frac{2}{\sqrt{t}}$ for all $t \geq 1$
2. $\max_{i\in[t]} \alpha_t^i \leq 2H/t$ and $\sum_i \alpha_t^{2i} \leq 2H/t$ for all $t \geq 1$
3. $\sum_{t=i}^{\infty} \alpha_t^i = 1 + \frac{1}{H}$ for all $i \geq 1$

.The last property is particularly useful as later in the proof, across each step we need to multiply terms of the form $\sum_{t=i}^{\infty} \alpha_t^i$ which means the blowup $(1+1/H)^H$ which is a constant factor. For actually proving the regret, we go back to the definition and let $\delta_h^k = (V_h^k - V_h^{\pi_k})(x_h^k)$ and thus

$$\text{Regret}(K) = \sum_{k=1}^{k=K} V_1^*(x_1^k) - V_1^{\pi_k}(x_1^k) \leq \sum_{k=1}^{k=K} V_1^k(x_1^k) - V_1^{\pi_k}(x_1^k) = \sum_{k=1}^{k=K} \delta_1^k$$

The main idea of the proof involves bounding $\sum_{k=1}^{k=K} \delta_h^k$ with $\sum_{k=1}^{k=K} \delta_{h+1}^k$ using a recursive formulation. Some work leads to the following relation:

$$\sum_{k=1}^{k=K} \delta_h^k \leq SAH + \left(1+\frac{1}{H}\right)\sum_{k=1}^{k=K} \delta_{h+1}^k + \sum_{k=1}^{k=K}(\beta_{n_h^k} + \zeta_{h+1}^k)$$

The above uses the property 3 of α_t^i as described above. $\sum_{k=1}^{k=K} \beta_{n_h^k}$ can be upper bounded by $\mathcal{O}(H^2SAT\eta)$ using the pigeonhole principle and $\sum_{k=1}^{k=K} \zeta_{h+1}^k$ can be upper bounded by $cH\sqrt{T\eta}$ using the Azuma Hoeffding inequality. Combining all the above pieces we can state that

$$\sum_{k=1}^{k=K} \delta_h^k \leq \mathcal{O}(H^2SA + \sqrt{H^4SAT\eta}) \text{ with probability atleast } 1-2p$$

When $T \geq \sqrt{H^4SAT\eta}$, H^2SA can be ignored as it is a lower quantity and hence this concludes the proof. In conclusion, we say that Q-learning is in fact efficient in terms of sample complexity however it needs to be done using a UCB exploration and a carefully chosen learning rate.

3 Linear Coupling - An Ultimate Unification of Gradient & Mirror Descent

3.1 Introduction

In this paper, we discuss a new first-order method of optimization in a black box framework. Consider minimizing a function $f(x)$, where $x \in \mathcal{Q} \subseteq \mathbb{R}^N$. We treat the optimization method as a black box where for each query y, we have access to $(f(y), \nabla f(y))$ and the goal is to minimize the number of queries in order to achieve an additive ϵ-approximate minimizer. Two main first-order methods include *gradient descent* and *mirror descent*. Gradient descent focuses on the primal progress while mirror descent focuses on the dual progress. We shall see that both these procedures benefit from complementary conditions which motivates the formulation of a new algorithm which somehow combines both of the above. The hope being that we can exploit the best of both the worlds in order to derive a faster algorithm. It turns out that even Nesterov's accelerated gradient method [13] can be derived in the above framework which provides a much cleaner proof for the same. In the next subsections, we first discuss gradient and mirror descent in some detail followed by how we can couple both of them together to get a faster first-order black box optimization algorithm.

3.2 Gradient Descent

Consider the unconstrained optimization problem ($\mathcal{Q} = \mathbb{R}^N$) and a general norm $\|.\|$ and its dual norm $\|.\|_*$. Consider a L-smooth function f which had to minimized which for any generic norm satisfies the following - $\|\nabla f(x) - \nabla f(y)\|_* \leq L\|x-y\|$ for every x, y. We have a global quadratic upper bound for such a function given by -

$$\forall y, f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y-x\|^2$$

Gradient Descent focuses on primal progress, i.e, maximise the objective decrease at any iteration k, $f(x_{k+1}) - f(x_k)$. Hence the update equation comes out to be the following -

$$x_{k+1} \leftarrow \arg\min_{y \in \mathcal{Q}} \frac{L}{2}\|y - x_k\|^2 + \langle \nabla f(x_k), y - x_k \rangle$$

For the normal l_2 norm, the above update equation comes out to the well-known update in the direction of gradient of f, i.e., $x_{k+1} \leftarrow x_k - \nabla f(x_k)/L$. In case of other non-Euclidean norms, the update need not be in the direction of gradient. Under the smoothness assumption, it can be shown that the primal progress is atleast -

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2L}\|\nabla f(x_k)\|_*^2$$

In gradient descent, the progress is better when the norm of $\nabla f(x_k)$ is large. *Thus large gradient norms are favourable for faster convergence in gradient descent.* For L-smooth objectives, the final bound shows that gradient descent converges in $\mathcal{O}(\frac{L}{\epsilon})$ iterations to find an additive ϵ-approximate minimizer. The limitation is that gradient descent does not make any attempt to construct a good lower bound to the optimum value $f(x^*)$. This dual problem is handled by mirror descent.

3.3 Mirror Descent

Mirror-descent methods tackle the dual problem by constructing lower bounds to the optimum. Each queried gradient can be viewed as a hyperplane lower-bounding the objective function f, i.e.,

$\forall u, f(u) \geq f(x) + \langle \nabla f(x), u - x \rangle$. Mirror-descent methods attempt to carefully construct a convex combination of these hyperplanes in order to yield even a stronger lower bound. For k query points $x_0, x_1, .., x_{k-1}$ we have,

$$\forall u, f(u) \geq \frac{1}{k} \sum_{t=0}^{t=k-1} f(x_t) + \frac{1}{k} \sum_{t=0}^{t=k-1} \langle \nabla f(x_t), u - x_t \rangle$$

Let $\bar{x}$ be the empirical average of all the query points $\{x_t\}_{t=0}^{t=k-1}$. Then using the above property and some basic convexity results, we can write the following -

$$\forall u, f(\bar{x}) - f(u) \leq \frac{1}{k} \sum_{t=0}^{t=k-1} \langle \nabla f(x_t), u - x_t \rangle = R_k(u)$$

$R_k(u)$ is known as the regret of the sequence $\{x_t\}_{t=0}^{t=k-1}$ with respect to the point u. We shall now a regularized version of regret which shall help in defining the update equation for mirror descent,

$$\tilde{R}_k(u) = \frac{1}{k} \sum_{t=0}^{t=k-1} \left(\frac{-w(u)}{\alpha} + \langle \nabla f(x_t), u - x_t \rangle \right)$$

where $\alpha > 0$ is trade-off parameter and w is a regularizer which is generally strongly convex. Using this, the update equations for mirror descent can be written in the following term: $x_{k+1} \leftarrow$ argmax$_u \tilde{R}_k(u)$. It can be shown that mirror descent converges in $T = \mathcal{O}(\rho^2/\epsilon^2)$ iterations where ρ^2 is the average value of $\|\nabla f(x_k)\|_*^2$ across iterations. *Thus small gradient norms are favourable for faster convergence in mirror descent.*

3.4 Initial Intuition

From the above discussion of gradient and mirror descent, one quickly identifies the complementary nature of both of them, gradient descent needs larger gradient norms for faster convergence while mirror descent prefers the opposite. This raises a question of whether we can couple both of them. Nesterov had designed an accelerated gradient method for L-smooth functions which is faster than gradient descent and requires only $\Omega(\sqrt{\frac{L}{\epsilon}})$ iterations. This has been shown to be asymptotically optimal even for generic non-Euclidean norms. As a thought experiment, suppose that $\|\nabla f(x_k)\|_2$ is always less than or more than a constant K. If $\|\nabla f(x_k)\|_2 \geq K$, then we perform T steps of gradient descent else we perform T steps for mirror descent. Without any loss of generality, consider we start with an initialization x_0 such that $f(x_0) - f(x^*)$ ia atmost 2ϵ and we want to find an x such that $f(x) - f(x^*) \leq \epsilon$. At each gradient descent step, the objective decreases by atleast $K^2/2L$ and thus to decrease it by ϵ, we would need $T = \Omega(\frac{\epsilon L}{K^2})$ iterations. For mirror descent, we would need $T = \Omega(\frac{K^2}{\epsilon^2})$ iterations, so the total iterations needed would be $\max\{\Omega(\frac{K^2}{\epsilon^2}), \Omega(\frac{\epsilon L}{K^2})\}$ iterations. If we choose K carefully so that both these terms become equal, then we need $T \geq \Omega(\sqrt{\frac{L}{\epsilon}})$ iterations which is what we wanted. Hence the above thought experiment can help in achieving the optimal convergence rate by using both gradient and mirror descent.

The problem with the above experiment is the assumption is gradient value is always more than or less than K. We cannot alternate between gradient and mirror descent at every algorithm as then we loose all guarantees (mirror descent needs steps in continuation and a gradient descent step in between will destroy it's convergence guarantees). Thus, we need to perform both gradient and mirror descent at every iteration. Let's say if x_k is the current estimate at kth iteration. We perform gradient descent and mirror descent on x_k to get y_k and z_k respectively. Then we find the new estimate $x_{k+1} \leftarrow \tau z_k + (1 - \tau) y_k$ where τ is a parameter which needs to be adjusted similar to the K in the earlier thought experiment. In the next section we look how we formalise this intuition.

3.5 Final Algorithm

To formally state the algorithm and the proof, we first look into some other properties of gradient and mirror descent which shall be helpful in the convergence proof of the algorithm.

Recall that gradient descent does updates in the following manner -

$$x_{k+1} \leftarrow \arg\min_{y \in \mathcal{Q}} \frac{L}{2}\|y - x_k\|^2 + \langle \nabla f(x_k), y - x_k \rangle$$

We shall henceforth refer to the above update as $\text{Grad}(x_k)$, i.e., $x_{k+1} = \text{Grad}(x_k)$, which when applied T times, we obtain the following convergence guarantee:

$$f(x_T) - f(x^*) \leq \mathcal{O}\left(\frac{L\|x_0 - x*\|_2^2}{T}\right) \text{ or when } T \geq \Omega\left(\frac{L\|x_0 - x*\|_2^2}{\epsilon}\right), f(x_T) - f(x^*) \leq \epsilon$$

If a function is ρ-Lipschitz continuous, this means that the subgradients follow the property that $\|\partial f(x)\|_* \leq \rho$. Let w be a 1-strong convex function with respect to $\|.\| \forall x \in \mathcal{Q} \setminus \partial\mathcal{Q}$. Define:

$$V_x(y) = w(y) - \langle \nabla w(x), y - x \rangle - w(x)$$

This property ensures that $V_x(x) = 0$ and $V_x(y) \geq \|x - y\|^2/2 \geq 0$ (using the property that w is 1-strongly convex. Some choices of $w(.)$ include the following -

1. $w(y) = \frac{1}{2}\|y\|_2^2$ when $\|.\|$ is the l_2 norm
2. $w(y) = \sum_i y_i \log(y_i)$ when $\|.\|$ is the l_1 norm

Using the above introduced notation, we can define the mirror descent step in the following manner -

$$\tilde{x} \leftarrow \text{Mirr}_x(\alpha.\partial f(x)) \text{ where } \text{Mirr}_x(z) = \arg\min_{y \in \mathcal{Q}}\{V_x(y) + \langle z, y - x \rangle\}$$

Let Θ be the upper bound on $V_{x_0}(x^*)$ and the learning rate $\alpha = \frac{\sqrt{2\Theta}}{\rho\sqrt{T}}$, the the following result can be stated regarding the convergence of mirror descent with T being the number of iterations:

$$T \geq \frac{2\Theta\rho^2}{\epsilon^2} \implies f(\bar{x}) - f(x^*) \leq \epsilon$$

Using the above, the basic version of the algorithm is described as above. Let $x_0 = y_0 = z_0$ and take two (fixed for now) parameters α and τ. For each iteration, we do the following:

- $y_k \leftarrow \text{Grad}(x_k)$
- $z_k \leftarrow \text{Mirr}_{z_{k-1}}(\alpha \nabla f(x_k))$
- $x_{k+1} \leftarrow \tau z_k + (1 - \tau) y_k$

The above two parameters need to be adjusted - α similar to the one done for mirror descent analysis and τ is similar to the K which balance drop in objective errors making them equal for gradient and mirror descent. For the above algorithm, we can state the following:

$$\begin{aligned}\alpha\langle \nabla f(x_{k+1}), z_k - u \rangle &\leq \frac{\alpha^2}{2}\|\nabla f(x_{k+1})\|_*^2 + V_{z_k}(u) - V_{z_{k+1}}(u) \\ &\leq \alpha^2 L(f(x_{k+1}) - f(y_{k+1})) + V_{z_k}(u) - V_{z_{k+1}}(u)\end{aligned}$$

The first inequality is a result of using the property of mirror descent while the second inequality comes by bounding the gradient norm in gradient descent. We need to choose a $\tau \in (0, 1)$ which balances the objective decrease $f(x_{k+1}) - f(y_{k+1})$, and the objective increase $f(y_k) - f(x_{k+1})$. The following lemma helps in choosing this mixing constant τ -

Lemma: *Letting τ to satisfy $\frac{1-\tau}{\tau} = \alpha L$ we get the following:*

$$\alpha\langle \nabla f(x_{k+1}), x_{k+1} - u \rangle \leq \alpha^2 L(f(y_k) - f(y_{k+1})) + V_{z_k}(u) - V_{z_{k+1}}(u)$$

So now, once we have a learning rate α, then τ can be easily obtained using the above relation. Using the above lemma, we can telescope the above inequality where $\bar{x}$ is the empirical average of all estimates x_i and $0 \leq i \leq T - 1$:

$$\alpha T(f(\bar{x}) - f(x*)) \leq \alpha^2 L(f(y_0) - f(y_T)) + V_{x_0}(u) - V_{x_T}(u)$$

Suppose our initial estimate $x_0 = y_0 = z_0$ suffers from an error of atmost d, i.e., $f(y_0) - f(x^*) \leq d$, and if $V_{x_0}(x^*) \leq \Theta$, we get that $(f(\bar{x}) - f(x*)) \leq \frac{1}{T}(\alpha L d + \Theta/\alpha)$. Thus we choose $\alpha = \sqrt{\Theta/Ld}$ which would equate the two terms in addition and minimize that quantity.
The above thing can be equivalently written in the following manner - in $T = 4\sqrt{L\Theta/d}$ iterations, we can obtain some $\bar{x}$ satisfying $f(\bar{x}) - f(x^*) \leq d/2$, which halves the distance from the optimum which was d to start off with. We continue doing this halving the distance every T iterations and hence we obtain an ϵ-approximate solution in:

$$T = \mathcal{O}\left(\sqrt{L\Theta/\epsilon} + \sqrt{L\Theta/2\epsilon} + \sqrt{L\Theta/4\epsilon} + ...\right) = \mathcal{O}\left(\sqrt{L\Theta/\epsilon}\right)$$

This convergence rate matches with the one as shown by Nesterov. As the iterations keep on increasing, $\alpha = \sqrt{\Theta/Ld}$, keeps on increasing and hence τ descreaing putting more weight on gradient descent near convergence. Both these parameters are adapted with iterations in the following manner: $\alpha_k = (k+1)/2L$ and $\tau_k = 2/(k+2)$. With this framework, we call the above algorithm as AGM and state the final theorem:

Theorem: *If $f(x)$ is $L-$smooth with respect to $\|.\|$ on $\mathcal{Q}$ and $w(x)$ is $1-$strongly convex with respect to $\|.\|$ on $\mathcal{Q}$, then AGM outputs y_T satisfying $f(y_T) - f(x^*) \leq 4\Theta L/T^2$, where Θ is any upper bound on $V_{x_0}(x^*)$.*

3.6 Beyond Accelerated Gradient Methods

Linear coupling not only provides a reinterpretation of Nesterov's accelerated gradient method but also provides a recipe for designing new first-order optimization methods.

1. **Beyond Non-Smooth Functions:** f need not be always smooth such as in the problem of positive linear programming. To apply Nesterov's accelerated methods, one usually smoothens f to another function say $f^{'}$ over which the algorithm is applied. It has been shown that directly applying linear coupling on f (with slight change in bounds which assumed smoothness) performs faster than the Nesterov's method [14].
2. **Three-Point Coupling:** In the stochastic case where we do not have access to the true gradient, the update equation can be modified to couple 3 vectors instead of 2. The update becomes $x_{k+1} \leftarrow \tau_1 z_k + \tau_2 \tilde{x} + (1 - \tau_1 - \tau_2) y_k$ where $\tilde{x}$ is a snapshot point whose full gradient is computed exactly but very infrequently. This three way coupling can be shown to have faster run times [15].
3. **Optimal Sampling Probability:** In the co-ordinate descent setting, where f is L_i-smooth in each coordinate i. Conventional methods sample each coordinate with probability L_i [16], however it can be shown using linear coupling that each coordinate should be sampled with a probability proportional to $\sqrt{L_i}$ [17]. This leads to the fastest coordinate-descent method.
4. **Supporting Non-Convexity:** It turns out that linear coupling can be extended to non-convex but smooth functions by replacing convexity arguments with a weaker quadratic lower bound. This leads to a stochastic algorithm that converges to approximate saddle-points which is the best we can hope for first-order methods for optimising a general non-convex function [18].

References

[1] Zeyuan Allen-Zhu, Yuanzhi Li, Zhao Song. A Convergence Theory for Deep Learning via Over-Parameterization, *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97:242-252

[2] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pages 4863–4873, 2018.

[3] Z. Allen-Zhu and L. Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. *Arxiv preprint* arXiv:1407.1537, 2014

[4] Arora, S., Hazan, E., Lee, H., Singh, K., Zhang, C., and Zhang, Y. Towards provable control for unknown linear dynamical systems. 2018b.

[5] Bartlett, P., Helmbold, D., and Long, P. Gradient descent with identity initialization efficiently learns positive definite linear transformations. In *International Conference on Machine Learning* (ICML), pp. 520–529, 2018.

[6] Goel, S., Klivans, A., and Meka, R. Learning one convolutional layer with overlapping patches. In *International Conference on Machine Learning* (ICML). arXiv preprint arXiv:1802.02547, 2018.

[7] Blum, A. L. and Rivest, R. L. Training a 3-node neural network is np-complete. *In Machine learning: From theory to applications (A preliminary version of this paper was appeared in NIPS 1989)*, pp. 9–28. Springer, 1993.

[8] Du, S. S., Lee, J. D., Tian, Y., Poczos, B., and Singh, A. Gradient descent learns one-hidden-layer CNN: don't be afraid of spurious local minima. In *International Conference on Machine Learning* (ICML). http://arxiv.org/abs/1712.00779, 2018b.

[9] Mohammad Gheshlaghi Azar, Ian Osband, and Remi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning* ICML, pages 263–272, 2017.

[10] Sham Kakade, Mengdi Wang, and Lin F Yang. Variance reduction methods for sublinear reinforcement learning. *ArXiv e-prints*, abs/1802.09184, April 2018

[11] Alexander L Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L Littman. PAC model-free reinforcement learning. In Proceedings of the 23rd *International conference on Machine learning*, pages 881–888. ACM, 2006.

[12] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):209–232, 2002.

[13] Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, December 2005. doi:10.1007/s10107-004-0552-5.

[14] Zeyuan Allen-Zhu and Lorenzo Orecchia. Nearly-Linear Time Positive LP Solver with Faster Convergence Rate. In *STOC*, 2015.

[15] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, NIPS 2013, pages 315–323, 2013.

[16] Yin Tat Lee and Aaron Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In FOCS, pages 147–156. IEEE, 2013.

[17] Zeyuan Allen-Zhu, Peter Richtárik, Zheng Qu, and Yang Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *ICML*, 2016.

[18] Zeyuan Allen-Zhu and Elad Hazan. Variance Reduction for Faster Non-Convex Optimization. In *ICML*, 2016.

[19] Shai Shalev-Shwartz and Tong Zhang. Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization. In *ICML*, pages 64–72, 2014