Convex Optimization 10-725, Lecture 3: Momentum is all you need?

Yuanzhi Li

Assistant Professor, Carnegie Mellon University
Visiting Researcher, Microsoft

Today

Last lecture

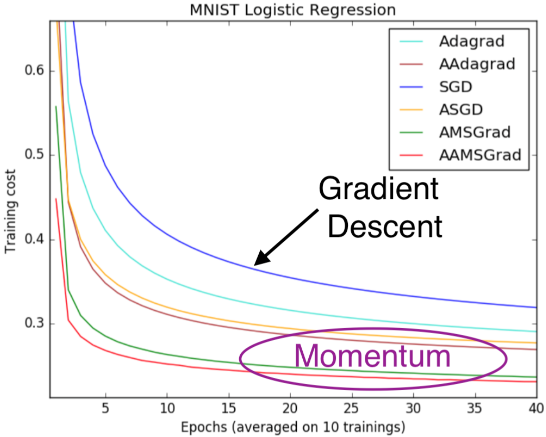
- We learn the smoothness of a function: the upper quadratic bound.
- We learn the Gradient Descent Algorithm: $x_{t+1} = x_t - \eta \nabla f(x_t)$.
- We learn the Gradient Descent Lemma.
- We learn the (Basic) Mirror Descent Lemma.

This lecture: Acceleration

- Key question: Is the Gradient Descent Algorithm: $x_{t+1} = x_t - \eta \nabla f(x_t)$ the best algorithm to use in practice?
- Best: Easy to code and fast to run.
  - Additional bonus: Easy to remember.
- No!
- This lecture we shall learn a new (fancy) algorithm called Accelerated Gradient Descent using a tool called Momentum.
- With Momentum: still easy to code and (much) faster to run.
  - *Harder* to remember.
  - No problem, in practice most of us (including myself) will just call something like $torch.optim.SGD(momentum = 0.9, nesterov = True)$.
  - The goal of this lecture is to understand the spirit of the acceleration: the Momentum.
- Momentum is one of the most powerful optimization tool and it is used almost everywhere in Machine learning, especially deep learning.

The power of momentum in practice

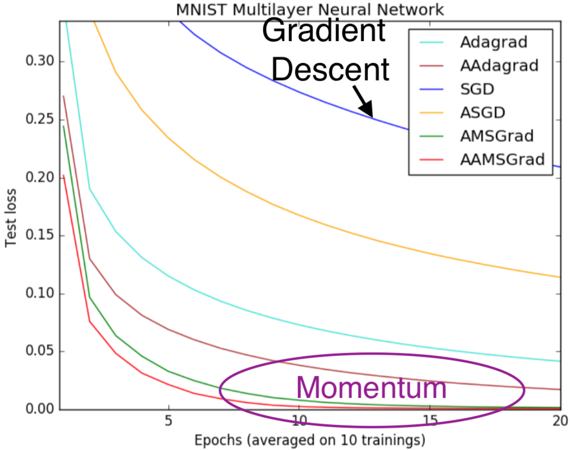
- Image Credit: "USING THE VARIATION OF THE GRADIENT TO ACCELERATE FIRST-ORDER OPTIMIZATION ALGORITHMS"
- Logistic Regression, MNIST data set (convex):



-

The power of momentum in practice

- Multi-layer neural network, MNIST data set (non-convex):



-

Warning

- Momentum is an extremely powerful optimization method, both in theory and in practice, to accelerate the gradient descent update.
- However, the rumor says that Accelerated Gradient Descent with Momentum is very difficult to learn.
- I can show you some proof (that it is very difficult to learn):

An Introduction to
the Conjugate Gradient Method
Without the Agonizing Pain
Edition $1\frac{1}{4}$

Jonathan Richard Shewchuk
August 4, 1994

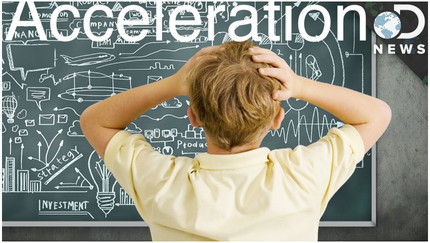
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

-
- Conjugate gradient is even a "simpler version" of Accelerated Gradient Descent, just on quadratic functions.

Warning

- From the author of the book "Convex Optimization: Algorithms and Complexity"

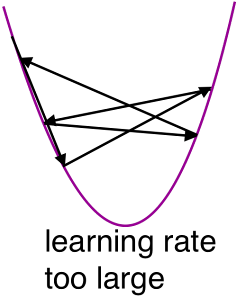
> In other words, Nesterov's Accelerated Gradient Descent performs a simple step of gradient descent to go from x_s to y_{s+1}, and then it 'slides' a little bit further than y_{s+1} in the direction given by the previous point y_s.
>
> The intuition behind the algorithm is quite difficult to grasp, and unfortunately the analysis will not be very enlightening either. Nonetheless Nesterov's Accelerated Gradient is an optimal method (in terms of oracle complexity) for smooth convex optimization, as shown by the following theorem.



Warming

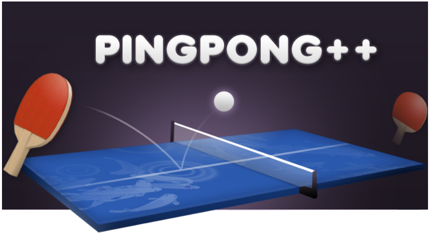
- But don't worry, I will teach you an Accelerated Gradient Descent that is actually easy to understand.
- The spirit of this lecture: Momentum is all you need.

Acceleration: The basic idea

- What can we do to outrun the update $x_{t+1} = x_t - \eta \nabla f(x_t)$ with $\eta \leq \frac{1}{L}$ on L-smooth function?
- Naive idea: Use the same update, with a larger learning rate $\eta \gg \frac{1}{L}$.
- Well, that sounds like a very dumb idea.
- 

- But that is how acceleration works.

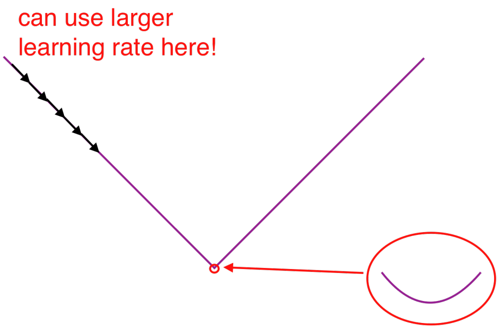
Gradient Descent with Large Learning Rate

- What if we use the update $x_{t+1} = x_t - \eta \nabla f(x_t)$ for $\eta \gg \frac{1}{L}$?
- Bad example: $f(x) = x^2$, then $\nabla f(x) = 2x$, $f(x)$ is 2-smooth.
- Theory predict that η should be no larger than 1/2.
- Using learning rate $\eta = 1$?
- $x_0 = 1$, $x_1 = x_0 - \eta \nabla f(x_0) = 1 - 2 = -1$, $x_2 = x_1 - \eta \nabla f(x_1) = -1 + 2 = 1$.
- $x_t = (-1)^t$, never converges.
- 

-

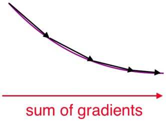
Gradient Descent with Large Learning Rate

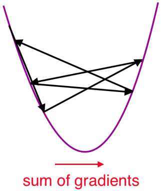
- On the other hand, for functions like this that is not very smooth just at some special places:



-
- On this function, we can indeed use larger learning rate most of the time.
- Key question: How do we adjust the learning rate automatically according to the shape of the function?

Acceleration and Momentum

- Key idea: always use large learning rate.
- Use the "weighted" sum of the gradients from the previous iterations to update the current point.
- When gradients point to the same direction:
- 

- When gradients bump back and forth:
- 

- "Weighted" sum/average of the past gradients is called the Momentum.

Momentum update

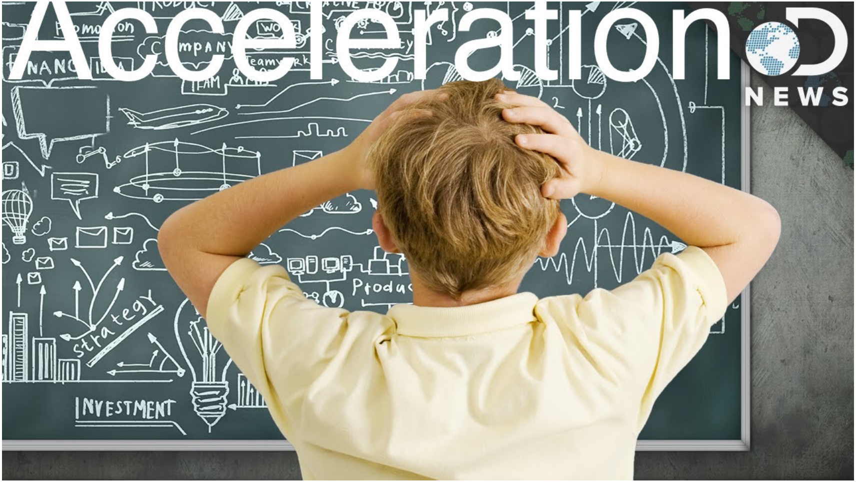
- Use a learning rate η (much) larger than $1/L$.
- Instead of update using $x_{t+1} = x_t - \eta \nabla f(x_t)$, we update using

$$x_{t+1} = x_t - \eta g_t$$

- Where g_t: "Weighted" average of the past gradients, the momentum.

Nesterov's Accelerated Gradient Descent

- Nesterov's Accelerated Gradient Descent Update:
- Warning:



-

Nesterov's Accelerated Gradient Descent

- For a L-smooth function:
- Gradient Descent step: $z_{t+1} = x_t - \eta \nabla f(x_t)$.
- Momentum step: $x_{t+1} = (1 - \gamma_t) z_{t+1} + \gamma_t z_t$.
- $\lambda_0 = 0, \lambda_t = \frac{1+\sqrt{1+4\lambda_{t-1}^2}}{2}$ and $\gamma_t = \frac{1-\lambda_t}{\lambda_{t+1}}$.
- This is the Nesterov's Accelerated Gradient Descent.
- Alternatively, one can compute that the update can also be approximately written as (for some sufficiently small value $\gamma > 0$): This is called heavy ball momentum.

$$x_{t+1} \approx x_t - \eta g_t, \quad g_t = \gamma \sum_{s \le t} (1 - \gamma)^{t-s} \nabla f(x_s) \tag{1}$$

- And we can choose $\eta = \frac{1}{\gamma L} \gg \frac{1}{L}$.

Nesterov's Accelerated Gradient Descent

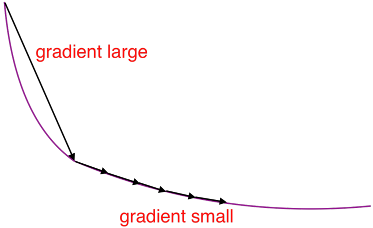
- Momentum: "Weighted" average of the past gradients.
- Intuitively, it makes Gradient Descent more stable when using large learning rate.
- But still, mathematically, why does it work?
- We are going to see a simpler way to perform acceleration with Momentum, and we see how it works mathematically.

Allen-Zhu and Orecchia's Accelerated Gradient Descent

- Recall: The Gradient Descent Lemma: for $\eta \leq \frac{1}{L}$,

$$f(x_{t+1}) \leq f(x_t) - \frac{\eta}{2}\|\nabla f(x_t)\|_2^2$$

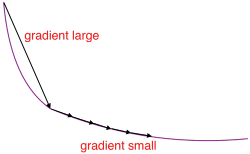
- Gradient is large, Gradient Descent converges fast.
- Key observation: Gradient is smaller than usual: We can now use a larger learning rate ($\eta \gg \frac{1}{L}$) without bumping back and forth.



-

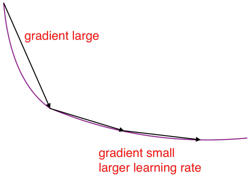
Thought Experiment: Function Value Decrease

- Uniform learning rate:



-

- Larger learning rate when we have smaller gradient:



-

- Momentum "tunes the learning rate" automatically according to this "local geometry".

Allen-Zhu and Orecchia's Accelerated Gradient Descent

- Key observation: Gradient is smaller than usual: We can use a larger learning rate without bumping back and forth!
- But (1). The function is L-smooth, how do we reason about the update with $\eta \gg \frac{1}{L}$?
- (2). How to we decide whether gradient is large or small? What is the threshold?

Allen-Zhu and Orecchia's Accelerated Gradient Descent

- Recall: The (basic) Mirror Descent Lemma: For every $\eta > 0$ and point y:

$$f(x_t) \le f(y) + \frac{1}{2\eta}\left(\|y - x_t\|_2^2 - \|y - x_{t+1}\|_2^2 + \|x_{t+1} - x_t\|_2^2\right)$$

- Recall: The telescoping sum using the (basic) Mirror Descent Lemma:

$$\frac{1}{T}\sum_{t=0}^{T-1} f(x_t) \le f(x^*) + \frac{1}{2\eta T}\|x^* - x_0\|_2^2 + \frac{\eta}{2T}\sum_{t=0}^{T-1}\|\nabla f(x_t)\|_2^2$$

Thought Experiment

- For simplicity, assuming $f(x^*) = 0$.
- We do the following thought experiment:
- For a fixed value $K > 0$, if $\|\nabla f(x_t)\|_2^2 \geq K$ holds for every t:
- The Gradient Descent Lemma: for $\eta \leq \frac{1}{L}$,

$$f(x_{t+1}) \leq f(x_t) - \frac{\eta}{2}\|\nabla f(x_t)\|_2^2$$

- Then, using $\eta = \frac{1}{L}$ and the Gradient Descent Lemma, $f(x_{t+1}) \leq f(x_t) - \frac{K}{2L}$: We need at most $\frac{Lf(x_0)}{K}$ iterations to find a point x_T with $f(x_T) \leq \frac{f(x_0)}{2}$

Thought Experiment

- On the other hand, if $\|\nabla f(x_t)\|_2^2 < K$ holds for every t:
- The telescoping sum after applying the (basic) Mirror Descent Lemma: For every η:

$$\frac{1}{T}\sum_{t=0}^{T-1} f(x_t) \le f(x^*) + \frac{1}{2\eta T}\|x^* - x_0\|_2^2 + \frac{\eta}{2T}\sum_{t=0}^{T-1}\|\nabla f(x_t)\|_2^2$$

- This implies that (by the assumption that $f(x^*) = 0$):

$$\frac{1}{T}\sum_{t=0}^{T-1} f(x_t) \le \frac{1}{2\eta T}\|x^* - x_0\|_2^2 + \frac{\eta K}{2}$$

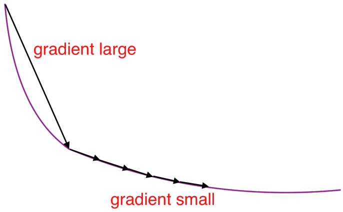
- With $\eta = \frac{f(x_0)}{2K}$: We need at most $\frac{4K\|x_0 - x^*\|_2^2}{f(x_0)^2}$ iterations to find a point x_T with $f(x_T) \le \frac{f(x_0)}{2}$

Thought Experiment: Function Value Decrease

- After the thought experiment:
- For a fixed value $K > 0$:
- If $\|\nabla f(x_t)\|_2^2 \geq K$ holds for every t: With $\eta = \frac{1}{L}$: We need at most $\frac{Lf(x_0)}{K}$ iterations to find a point x_T with $f(x_T) \leq \frac{f(x_0)}{2}$
- If $\|\nabla f(x_t)\|_2^2 < K$ holds for every t:
- With $\eta = \frac{f(x_0)}{2K}$: We need at most $\frac{4K\|x_0 - x^*\|_2^2}{f(x_0)^2}$ iterations to find a point x_T with $f(x_T) \leq \frac{f(x_0)}{2}$
- Picking K to be $\sqrt{\frac{Lf^3(x_0)}{4\|x_0 - x^*\|_2^2}}$, we know that in both cases: We need at most $\frac{2\|x_0 - x^*\|_2\sqrt{L}}{\sqrt{f(x_0)}}$ iterations to find a point x_T with $f(x_T) \leq \frac{f(x_0)}{2}$
- In the second case, when $f(x_0) \approx \|x_0 - x^*\|_2 \approx 1$, the learning rate is indeed much larger: $\eta = \frac{f(x_0)}{K} \approx \frac{1}{\sqrt{L}} \gg \frac{1}{L}$.

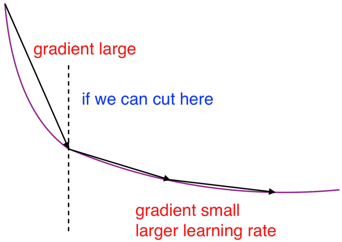
Thought Experiment: Function Value Decrease

- Uniform learning rate:



-

- Larger learning rate when we have smaller gradient:



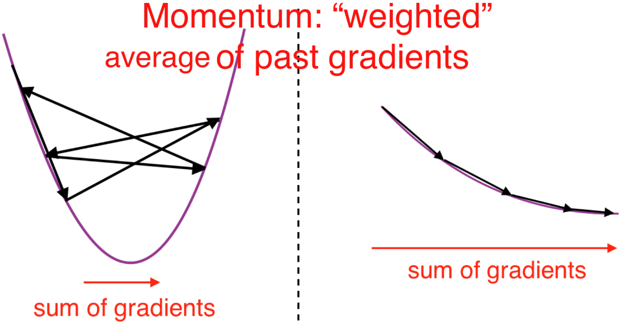
-

The "final convergence" of the though experiment

- We need at most $\frac{2\|x_0 - x^*\|_2\sqrt{L}}{\sqrt{f(x_0)}}$ iterations to find a point x_T with $f(x_T) \leq \frac{f(x_0)}{2}$.
- Say $f(x_0) = 1$ and $\|x_0 - x^*\|_2 = 1$:
- We need at most $\frac{2\sqrt{L}}{\sqrt{1}}$ iterations to a point x_T with $f(x_T) \leq \frac{1}{2}$.
- After that using x_T as the new initialization, we need at most $\frac{2\sqrt{L}}{\sqrt{1/2}}$ iterations to find a point $x_{T'}$ with $f(x_{T'}) \leq \frac{1}{4}$.
- After that using $x_{T'}$ as the new initialization, we need at most $\frac{2\sqrt{L}}{\sqrt{1/4}}$ iterations to find a point $x_{T''}$ with $f(x_{T''}) \leq \frac{1}{8}$.
- ......
- Eventually, for $\varepsilon > 0$, we need at most $\frac{\sqrt{2L}}{\sqrt{\varepsilon}}$ iterations to find a point x_{T_ε} with $f(T_\varepsilon) \leq \varepsilon$.
- Recall: Gradient Descent needs $\frac{2L}{\varepsilon}$ iterations to find that point.

The real acceleration

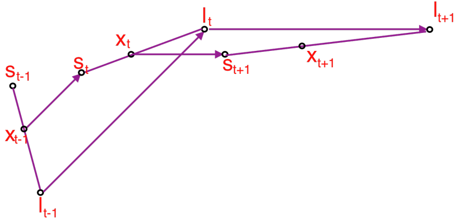
- How do we find this "magic value" K to tune the learning rate η?
- In particular, it might neither be the case of $\|\nabla f(x_t)\|_2^2 \geq K$ holds for every t nor $\|\nabla f(x_t)\|_2^2 < K$ holds for every t.
- Idea: Every iteration, we do both a step with $\eta = \frac{1}{L}$ (Gradient Descent) and a step with a larger $\eta \gg \frac{1}{L}$ (with momentum). In the end we combine them:



-

Linear Coupling

- At every iteration, for a fixed τ:
- Gradient Descent (proper learning rate): $s_{t+1} = x_t - \frac{1}{L}\nabla f(x_t)$.
- Update (large learning rate $\eta \gg \frac{1}{L}$): $l_{t+1} = l_t - \eta\nabla f(x_t)$.
- Linear coupling: for a $\tau \in [0,1]$: $x_{t+1} = (1-\tau)s_{t+1} + \tau l_{t+1}$.



-
- $l_t = l_0 - \eta\sum_{r=0}^{t-1}\nabla f(x_r)$ is the momentum term. The final update combines (small learning rate) gradient descent with this (large learning rate) momentum.

The summary

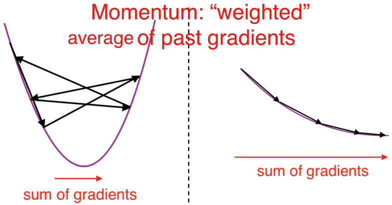
- The above algorithm is best for "mathematical thinking" purpose, where it combines "gradient descent" and "momentum".
- (1). Gradients bump back and forth using large learning rate: momentum becomes small, the algorithm updates the current point mostly using gradient descent with proper learning rate.
- (2). Gradients point to the same direction: momentum becomes large, the algorithm updates the current point mostly using momentum.
- In practice, we do not even need such a combination, momentum is all you need!
- Use the update:

$$x_{t+1} \approx x_t - \eta g_t, \quad g_t = \gamma \sum_{s \leq t} (1-\gamma)^{t-s} \nabla f(x_s) \tag{2}$$

- This is essentially what's inside $optim.SGD(momentum = 0.9, nesterov = True)$.

The summary

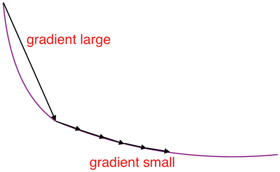
- Momentum is really really important in deep learning as well. We will see in the future lecture.
- In deep learning, we need large learning rate (to prevent memorization and to train faster).
- Use momentum: "weighted" average of the past gradients.



-

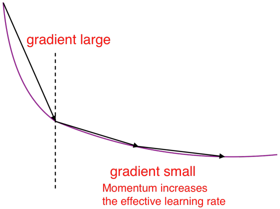
Thought Experiment: Function Value Decrease

- Gradient Descent:



-

- Momentum: When gradient is smaller than typical, “increases the effective learning rate”.



-

The summary

- Momentum is one of the most important tool in optimization, essentially all machine learning algorithms use momentum nowadays.
- However, in the later lectures, we will still mainly focus on gradient-based optimization algorithms, without using momentum.
- Since it is easier to grasp the spirit of these new algorithms.
- But you can always replace the gradient in those algorithms with momentum, and in practice, *momentum is all you need*.

The proof (not important)

- Update (small learning rate): $s_{t+1} = x_t - \frac{1}{L}\nabla f(x_t)$.
- Gradient Descent Lemma:

$$f(s_{t+1}) \leq f(x_t) - \frac{1}{2L}\|\nabla f(x_t)\|_2^2$$

The proof

- Update (large learning rate $\eta \gg \frac{1}{L}$): $l_{t+1} = l_t - \eta \nabla f(x_t)$.
- We will derive the ($\frac{\text{Basic + Real}}{2}$) Mirror Descent Lemma:
- To do that, notice:

$$\langle \nabla f(x_t), x^* - l_t \rangle = \frac{1}{\eta} \langle l_t - l_{t+1}, x^* - l_t \rangle$$

- Recall from last lecture, we have:

$$\frac{1}{\eta} \langle l_t - l_{t+1}, x^* - l_t \rangle = -\frac{1}{2\eta} \left(\|x^* - l_t\|_2^2 - \|x^* - l_{t+1}\|_2^2 + \|l_t - l_{t+1}\|_2^2 \right)$$

- Therefore we have:

$$\langle \nabla f(x_t), l_t - x^* \rangle = \frac{1}{2\eta} \left(\|x^* - l_t\|_2^2 - \|x^* - l_{t+1}\|_2^2 + \|l_t - l_{t+1}\|_2^2 \right)$$

The proof

- Now we have:

$$\langle \nabla f(x_t), l_t - x^* \rangle = \frac{1}{2\eta}\left(\|x^* - l_t\|_2^2 - \|x^* - l_{t+1}\|_2^2 + \|l_t - l_{t+1}\|_2^2 \right)$$

- On the other hand, using the lower linear bound:

$$\langle \nabla f(x_t), s_t - x_t \rangle \le f(s_t) - f(x_t)$$

- By linear coupling defintion: for a $\tau \in [0,1]$: $x_t = (1-\tau)s_t + \tau l_t$.
- Therefore,

$$l_t - x^* + \frac{1-\tau}{\tau}(s_t - x_t) = x_t - x^*$$

- Therefore,

$$\langle \nabla f(x_t), x_t - x^* \rangle = \langle \nabla f(x_t), l_t - x^* \rangle + \frac{1-\tau}{\tau}\langle \nabla f(x_t), s_t - x_t \rangle$$

$$\le \frac{1}{2\eta}\left(\|x^* - l_t\|_2^2 - \|x^* - l_{t+1}\|_2^2 + \|l_t - l_{t+1}\|_2^2 \right) + \frac{1-\tau}{\tau}(f(s_t) - f(x_t))$$

The proof

- Now we have:

$$\begin{aligned}&\langle \nabla f(x_t), x_t - x^* \rangle \\ &\le \frac{1}{2\eta}\left(\|x^* - l_t\|_2^2 - \|x^* - l_{t+1}\|_2^2 + \|l_t - l_{t+1}\|_2^2\right) + \frac{1-\tau}{\tau}(f(s_t) - f(x_t))\end{aligned}$$

- Using the lower linear bound: $f(x_t) - f(x^*) \le \langle \nabla f(x_t), x_t - x^* \rangle$, we have the ($\frac{\text{Basic + Real}}{2}$) Mirror Descent Lemma:

$$\begin{aligned}&f(x_t) - f(x^*) \\ &\le \frac{1}{2\eta}\left(\|x^* - l_t\|_2^2 - \|x^* - l_{t+1}\|_2^2 + \|l_t - l_{t+1}\|_2^2\right) + \frac{1-\tau}{\tau}(f(s_t) - f(x_t))\end{aligned}$$

The proof

- The Gradient Descent Lemma:

$$f(s_{t+1}) \le f(x_t) - \frac{1}{2L}\|\nabla f(x_t)\|_2^2$$

- The $\left(\frac{\text{Basic} + \text{Real}}{2}\right)$ Mirror Descent Lemma:

$$\begin{aligned} & f(x_t) - f(x^*) \\ & \le \frac{1}{2\eta}\left(\|x^* - l_t\|_2^2 - \|x^* - l_{t+1}\|_2^2 + \|l_t - l_{t+1}\|_2^2\right) + \frac{1-\tau}{\tau}(f(s_t) - f(x_t)) \end{aligned}$$

- Recall $l_{t+1} = l_t - \eta\nabla f(x_t)$, so we have:

$$\begin{aligned} & f(x_t) - f(x^*) \\ & \le \frac{1}{2\eta}\left(\|x^* - l_t\|_2^2 - \|x^* - l_{t+1}\|_2^2 + \eta^2\|\nabla f(x_t)\|_2^2\right) + \frac{1-\tau}{\tau}(f(s_t) - f(x_t)) \end{aligned}$$

The proof

- The Gradient Descent Lemma:

$$f(s_{t+1}) \leq f(x_t) - \frac{1}{2L}\|\nabla f(x_t)\|_2^2$$

- Using the $\left(\frac{\text{Basic} + \text{Real}}{2}\right)$ Mirror Descent Lemma, we have:

$$\begin{aligned} &f(x_t) - f(x^*) \\ &\leq \frac{1}{2\eta}\left(\|x^* - l_t\|_2^2 - \|x^* - l_{t+1}\|_2^2 + \eta^2\|\nabla f(x_t)\|_2^2\right) + \frac{1-\tau}{\tau}(f(s_t) - f(x_t)) \end{aligned}$$

- Combine these two, we have:

$$\begin{aligned} &f(x_t) - f(x^*) \\ &\leq \frac{1}{2\eta}\left(\|x^* - l_t\|_2^2 - \|x^* - l_{t+1}\|_2^2 + 2L\eta^2(f(x_t) - f(s_{t+1}))\right) \\ &+ \frac{1-\tau}{\tau}(f(s_t) - f(x_t)) \end{aligned}$$

The proof

- We have:

$$
\begin{aligned}
& f(x_t) - f(x^*) \\
& \leq \frac{1}{2\eta}\left(\|x^* - l_t\|_2^2 - \|x^* - l_{t+1}\|_2^2 + 2L\eta^2(f(x_t) - f(s_{t+1}))\right) \\
& + \frac{1-\tau}{\tau}(f(s_t) - f(x_t))
\end{aligned}
$$

- Pick τ such that $2L\eta^2 = \frac{1-\tau}{\tau}$, we have:

$$
\begin{aligned}
& f(x_t) - f(x^*) \\
& \leq \frac{1}{2\eta}\left(\|x^* - l_t\|_2^2 - \|x^* - l_{t+1}\|_2^2 + 2L\eta^2(f(s_t) - f(s_{t+1}))\right)
\end{aligned}
$$

The proof

- We have:

$$f(x_t) - f(x^*) \leq \frac{1}{2\eta}\left(\|x^* - l_t\|_2^2 - \|x^* - l_{t+1}\|_2^2 + 2L\eta^2(f(s_t) - f(s_{t+1}))\right)$$

- Averaging over $t = 0, 1, \cdots, T-1$ we have: (assuming $f(x^*) = 0$)

$$\frac{1}{T}\sum_{t=0}^{T-1} f(x_t) \leq \frac{1}{2T\eta}\left(\|x^* - l_0\|_2^2 + 2L\eta^2 f(s_0)\right)$$

- Picking $\eta = \frac{\|x^* - l_0\|_2}{2\sqrt{f(s_0)}}$, we can find a point x_T with $f(x_T) \leq \frac{f(s_0)}{2}$ in

$$T_{AGD} = \frac{\|x^* - l_0\|_2\sqrt{2L}}{\sqrt{f(s_0)}}$$

iterations. Recall: Gradient descent needs $T_{GD} \approx T_{AGD}^2$ iterations.