Exploring Accelerated Gradient Descent Methods for RNNs

Rohan Gulati, Timothy Park, Sam Skolnick

April 30, 2024

Abstract

We discuss Nesterov's original gradient descent paper and then discuss a paper linking mirror descent and Nesterov's gradient descent and explain the motivations behind each algorithm. We then prove the convergence of mirror descent ($O(\frac{1}{\sqrt{n}})$) and of the convergence of Nesterov's gradient descent ($O(\frac{1}{n^2})$). In our extension, we examine the convergence and optimality of various gradient descent algorithms for training recurrent neural networks over different amounts of training time. Nesterov's gradient descent converges the fastest with the least training loss, but only by a small margin. We discuss the implications of this and how perhaps, collecting more data might show stochastic gradient descent as a better model for RNNs in general because over time it tends towards superior minima.

1 Introduction

Training neural networks for machine learning is becoming increasingly important as research into deep learning and artificial intelligence grows more complex. Indeed, training neural networks as efficiently as possible while achieving the optimum is the goal of many of these areas of research. To train a neural network, one must choose an optimization algorithm that can find the appropriate weights to support any input. In this paper, we derive the convergence of Mirror gradient descent and Nesterov's Accelerated gradient descent. We then evaluate various gradient descent methods in training recurrent neural networks on time-series weather data at each epoch while training. The six algorithms evaluated are Stochastic Gradient Descent (SGD), Momentum Gradient Descent, Adagrad, Nesterov's Accelerated Gradient Descent (NAG), RMS Prop, and Adam. Based off Weinan E et. al's analysis of the application of SGD and Adam to deep learning we expand this in the context of the evaluation of the various gradient descent's performance context of finding optimum solutions.

2 Related Work

2.1 Nesterov's Gradient Descent Paper

Nesterov's gradient descent algorithm is a specific type of gradient descent algorithm designed to solve convex optimization problems with a greatly increased convergence rate. This method of gradient descent requires that the functions are L-smooth and convex, as well as unconstrained. Initially, we start with values $x_0, y_0, k, a_0, \alpha_{-1}$ where k is the iteration, x is the location along the convex function, y is an intermediate value representing the look ahead value, α is the step size, and a is the look ahead ratio.

The update step involves multiple parts. At the kth time step, an index i is determined that solves

$$min \quad i$$

$$s.t. f(y_k) - f(y_k - 2^{-i}\alpha_{k-1}\nabla f(y_k) \geq 2^{-i-1}\alpha_{k-1}\|\nabla f(y_k)\|_2^2$$

This alpha value allows us to determine the step size of this specific time step. y_k is looking ahead past the current position x_k in the direction of the last update step. In other words, y_k is the result of projecting the line between x_k and x_{k-1} further past x_k. The step size is then determined based on a point that is

further than the current point. Because y_k is overshooting the current position, it should catch changes in the gradient at a much faster rate than if only x_k was used. Here is how everything is derived.

$$\alpha_k = 2^{-i}\alpha_{k-1}$$

$$x_k = y_k - \alpha_k \nabla f(y_k)$$

A new a "look ahead ratio" is calculated for the sole purpose of finding a new y_k value, which will look forward along the line drawn from x_{k-1} to x_k

$$a_{k+1} = (1 + \sqrt{4a_k^2 + 1})/2$$

$$y_{k+1} = x_k + (a_k - 1(x_k - x_{k-1})/a_{k+1}$$

To repeat, the general flow of the algorithm is as follows 1.) Determine the step size based on the calculated i, which was based upon the previous look ahead point. 2.) Update x_k based upon the gradient of the function at the look ahead point y_k and the new step size α_k. 3.) Find a suitable look ahead size a_k. This determines how far past x_k along the previous path that y_k lies) 4.) Determine the next look ahead point

After T iterations of Nesterov's Accelerated Descent, the error $f(\vec{x}_T) - min_{\vec{x}\in R^n} f(\vec{x})$ is upper-bounded by

$$\frac{4L\|y_0 - x^*\|_2^2}{(k+2)^2}$$

In other words,

$$f(\vec{x}_T) - min_{\vec{x}\in R^n} f(\vec{x}) \leq \frac{4L\|y_0 - x^*\|_2^2}{(k+2)^2}$$

This allows Nesterov's gradient descent to converge at a rate of $O(1/k^2)$, much faster than normal gradient descent.

2.1.1 Linear Coupling of Gradient and Mirror Descent Paper

This paper discusses the strengths and weaknesses of gradient descent as well as mirror descent. Then, it moves towards discussing a linear coupling method that performs a gradient descent step and a mirror descent step in tandem. This allows for a very strong convergence because it leverages both methods' strengths. The authors of this paper claim that gradient descent and mirror descent can be thought of as primal and dual to each other. In other words, gradient descent makes primal progress, and mirror descent makes dual progress. A few assumptions are made regarding the function f and the distance generating function (DGF) w. The algorithm proposed by Allen-Zhu and Oreccia is shown on an unconstrained convex and differentiable function f which is L-smooth with respect to a norm. In addition, the DGF w must be 1-strongly convex with respect to the same norm. The algorithm at the kth iteration is as follows: 1a. We determine the step size for the next mirror descent step as follows:

$$\alpha_{k+1} = \frac{k+2}{2L}$$

1b. We also determine the ratio between mirror descent and gradient descent results that will be combined to get our final result:

$$\tau_k = \frac{1}{\alpha_{k+1}L} = \frac{2}{k+2}$$

2. We then compile our values to find the location on the function using the ratio we just derived, where y_k is the result of a gradient descent step, and z_k is the result of a mirror descent step:

$$x_{k+1} = \tau_k z_k + (1 - \tau_k) y_k$$

The paper found the gradient to be much more useful than the mirror descent step when we are close to the optimal point, which is why that ratio moves towards the gradient descent step over time.

3. Then we find the gradient descent and mirror descent values for the next update step:

$$y_{k+1} = Grad(x_{k+1})$$

$$z_{k+1} = Mirr_{z_k}(\alpha_{k+1}\nabla f(x_{k+1}))$$

We repeat these iterations until we get a desirable y_T After T iterations, we can upper bound the error $f(\vec{x}_T) - min_{\vec{x}\in R^n} f(\vec{x})$ as follows:

$$f(\vec{x}_T) - min_{\vec{x}\in R^n} f(\vec{x}) \leq \frac{4\theta L}{(T+1)^2}$$

where θ is the upper bound of the Bregman Divergence $D_w(z_0; x^*)$ These two papers both outline methods to accelerate gradient descent. The first paper looks to introduce Nesterov's accelerated gradient (NAG) descent in an isolated manner. The second linear coupling paper analyzes mirror descent, and then it provides motivation behind the derivation NAG, which is opaque in the Nesterov's original paper. Following this, it proves Nesterov's convergence rate. In doing so, the linear coupling paper provides an alternate, clearer explanation for NAG. NAG is utilizes a combination of multiple points to look forward and find a more efficient step to take. This runs in parallel with how a convex combination of mirror descent and gradient descent results create the next linear coupling step. In addition, the paper on linear coupling discusses other settings where linear coupling may be a stronger option than Nesterov's gradient descent, showing that it is not just an alternate interpretation but an improvement to some capacity.

2.1.2 Accelerated Gradient Descent on Saddle Points

This paper studies a variant of Nesterov's Accelerated Gradient Descent (AGD) that is able to find a stationary point more quickly than normal or accelerated gradient descent. It can find a stationary point in $O(1/\epsilon(7/4))$, faster than the current $O(1/\epsilon^2)$ steps necessary for gradient descent to converge. This is done by utilizing a Hamiltonian function and using a different framework that is able to track long term behavior of the algorithm. Nesterov's is proven to converge more quickly for convex problems, not non convex ones. However, it is still being used to perform non convex minimizations, such as the training of a neural network. It is extremely important to find gradient descent methods that are faster because this enables us to train models with higher efficiency. By finding an algorithm that functions better than the previous best, this can allow us to better understand how acceleration algorithms and nonconvex optimizations can be best leveraged. One open ended question that remains is which accelerated gradient descent methods work the best for which types of functions, as well as which types of settings each can be leveraged to their fullest expense.

2.1.3 Towards Theoretically Understanding Why SGD Generalizes Better Than ADAM in Deep Learning by Wienan E et. al.

Wienan E et. al. analyze the escape performance of Adam and SGD from local minima. They strive to address model performance over time. More specifically, that Adam progresses faster in the initial training phase but plateaus while SGD is initially slower but could have higher test performance. Initially, the authors propose that SGD is more "locally unstable", leading it converge to minima at flatter or more asymmetric valleys, which happen to have better performance than minima in steeper valleys. The authors examine SGD and Adam in the context of their Levy-driven differential equations and systemic α-stable distribution tails. The author conclude that the faster convergence rate of Adam but better ultimate solution of SGD are theoretically and experimentally explainable i.e. SGD's preference of flatter-basin minima yields better minimas.

3 Problems

3.1 Mirror Gradient Descent

3.1.1 Mirror Gradient Descent a

Prove that for any convex set X and α-strongly convex function $h : X ß R$, for any fixed $\vec{x} \in X$, the Bregman divergence $D_h(\vec{y}; \vec{x})$ is a α-strongly convex function of $\vec{y}$.

$$D_h(\vec{y}, \vec{x}) = h(\vec{y}) - h(\vec{x}) - [\nabla h(\vec{x})]^T(\vec{y} - \vec{x})$$

Show: $D_h(\vec{y}_1, \vec{x}) \geq D_h(\vec{y}_2, \vec{x}) + \nabla D_h(\vec{y}_2, \vec{x})^T(\vec{y}_1 - \vec{y}_2) + \frac{\alpha}{2}\|\vec{y}_1 - \vec{y}_2\|_2^2$
We can expand and simplify like terms to get

$$D_h(\vec{y}_1, \vec{x}) - D_h(\vec{y}_2, \vec{x}) = h(\vec{y}_1) - h(\vec{y}_2) - [\nabla h(\vec{x})]^T(\vec{y}_1 - \vec{y}_2)$$

Then we calculate $\nabla D_h(\vec{y}_2, \vec{x})$

$$\nabla D_h(\vec{y}_2, \vec{x}) = \nabla h(\vec{y}_2) - \nabla h(\vec{x})$$

We see we need $\nabla D_h(\vec{y}_2, \vec{x})^T(\vec{y}_1 - \vec{y}_2)$ in our final inequality, so we calculate it here:

$$\nabla D_h(\vec{y}_2, \vec{x})^T(\vec{y}_1 - \vec{y}_2) = \nabla h(\vec{y}_2)^T(\vec{y}_1 - \vec{y}_2) - \nabla h(\vec{x})^T(\vec{y}_1 - \vec{y}_2)$$

We now observe that:

$$\begin{aligned} D_h(\vec{y}_1, \vec{x}) - D_h(\vec{y}_2, \vec{x}) - \nabla D_h(\vec{y}_2, \vec{x})^T(\vec{y}_1 - \vec{y}_2) &= h(\vec{y}_1) - h(\vec{y}_2) - [\nabla h(\vec{x})]^T(\vec{y}_1 - \vec{y}_2) - \nabla h(\vec{y}_2)^T(\vec{y}_1 - \vec{y}_2) + \nabla h(\vec{x})^T(\vec{y}_1 - \vec{y}_2) \\ &= h(\vec{y}_1) - h(\vec{y}_2) - \nabla h(\vec{y}_2)^T(\vec{y}_1 - \vec{y}_2) \end{aligned}$$

h($\vec{x}$) is α-strongly convex, so:

$$h(\vec{y}_1) \geq h(\vec{y}_2) + \nabla h(\vec{y}_2)^T(\vec{y}_1 - \vec{y}_2) + \frac{\alpha}{2}\|\vec{y}_1 - \vec{y}_2\|_2^2$$

So we can plug this into the equation above:

$$D_h(\vec{y}_1, \vec{x}) - D_h(\vec{y}_2, \vec{x}) - \nabla D_h(\vec{y}_2, \vec{x})^T(\vec{y}_1 - \vec{y}_2) \geq h(\vec{y}_2) + \nabla h(\vec{y}_2)^T(\vec{y}_1 - \vec{y}_2) + \frac{\alpha}{2}\|\vec{y}_1 - \vec{y}_2\|_2^2 - h(\vec{y}_2) - [\nabla h(\vec{x})]^T(\vec{y}_1 - \vec{y}_2) - \nabla h(\vec{y}_2)^T(\vec{y}_1 - \vec{y}$$

Simplifying like terms, we get:

$$D_h(\vec{y}_1, \vec{x}) - D_h(\vec{y}_2, \vec{x}) - \nabla D_h(\vec{y}_2, \vec{x})^T(\vec{y}_1 - \vec{y}_2) \geq \frac{\alpha}{2}\|\vec{y}_1 - \vec{y}_2\|_2^2$$

$$D_h(\vec{y}_1, \vec{x}) \geq D_h(\vec{y}_2, \vec{x}) + \nabla D_h(\vec{y}_2, \vec{x})^T(\vec{y}_1 - \vec{y}_2) + \frac{\alpha}{2}\|\vec{y}_1 - \vec{y}_2\|_2^2$$

As such, we have arrived at the fact that $D_h(\vec{y}, \vec{x})$ is α-strongly convex when h is α-strongly convex. So when h is α-strongly convex, so is its Bregman Divergence.

3.1.2 Mirror Gradient Descent - b

$$Prove : \langle \nabla h(\vec{x}) - \nabla h(\vec{y}), \vec{y} - \vec{u} \rangle = D_h(\vec{u}; \vec{y}) - D_h(\vec{u}; \vec{y}) - D_h(\vec{y}; \vec{x})$$

We will expand out the LHS and demonstrate that the RHS and LHS are equal.

$$[\nabla h(\vec{x})]^\top \vec{y} - [\nabla h(\vec{y})]^\top \vec{y} - [\nabla h(\vec{x})]^\top \vec{u} + [\nabla h(\vec{y})]^\top \vec{u}$$

We will then add and subtract $h(\vec{u}), h(\vec{x}), h(\vec{y})$ to be put into the respective Bregman Divergence while maintaining the equality's value as for each term added the equivalent term is subtracted.

$$h(\vec{u}) - h(\vec{u}) - h(\vec{x}) + h(\vec{x}) - h(\vec{y}) + h(\vec{y}) + [\nabla h(\vec{x})]^\top \vec{y} - [h(\vec{y})]^\top \vec{y} - [h(\vec{x})]^\top \vec{u} + [h(\vec{y})]^\top \vec{u} + [h(\vec{x})]^\top \vec{x} - [h(\vec{x})]^\top \vec{x}$$

Grouping each term as a Bregman Divergence

$$(h(\vec{u}) - h(\vec{x}) - [\nabla h(\vec{x})]^\top (\vec{u} - \vec{x})) - (h(\vec{u}) - h(\vec{y}) - [\nabla h(\vec{y})]^\top (\vec{u} - \vec{y})) - (h(\vec{y}) - h(\vec{x}) - [\nabla h(\vec{x})]^\top (\vec{y} - \vec{x}))$$

Finally we achieve

$$\langle \nabla h(\vec{x}) - \nabla h(\vec{y}), \vec{y} - \vec{u} \rangle = D_h(\vec{u}; \vec{x}) - D_h(\vec{u}, \vec{y}) - D_h(\vec{y}, \vec{x})$$

3.1.3 Mirror Gradient Descent - c

Given $\vec{x} \in X$ and given some $\vec{g} \in \mathbb{R}^n$ and $\eta > 0$, compute Mirr$(\eta\vec{g}; \vec{x})$ where $h(\vec{x}) = \sum(x_i \log(x_i) - x_i)$ such that $\sum x_i = 1$.

$$argmin_z(\langle \nabla f(\vec{x_k}), \vec{z} \rangle + D_h(\vec{z}; \vec{x}))$$

$$argmin_z(\nabla f(\vec{x_k})^T \vec{z} + \sum(z_i log(z_i) - z_i) - \sum(x_i log(x_i) - x_i)) - \sum log(x_i)(z_i - x_i)$$

$$argmin_z(\eta \vec{g_k}^T \vec{z} + \sum(z_i log(z_i) - y_i) - \sum(x_i log(x_i) - x_i)) - \sum log(x_i)(z_i - x_i)$$

Constructing the Lagrangian of this problem subject to $\sum x_i = 1$ to determine $Mirr(\eta\vec{g}; \vec{x})$ as Mirror Descent goes to the optimalvalue of the function. As the function is convex we can solve the Lagrangian to find the optimal solution that satisfies it returns the optimal solution satisfying the constraints.

$$\eta \vec{g_k}^T \vec{z} + \sum(z_i log(z_i) - z_i) - \sum(x_i log(x_i) - x_i)) - \sum log(x_i)(z_i - x_i) + \nu(\sum x_i - 1) - \lambda^T z$$

We take the gradient with respect to $\vec{z}$

$$\eta \vec{g}^T \vec{1} + \sum log(z_i) - \sum log(x_i) + \nu \vec{1} - \vec{\lambda} = \vec{0}$$

We now solve for where the gradient is zero component by component.

$$\eta g_i + log(z_i) - log(x_i) + \nu - \lambda_i = 0$$

$$log(z_i) = log(x_i) - \eta g_i - \nu + \lambda_i$$

$$z_i = x_i e^{-\eta g_i - \nu + \lambda_i}$$

$$z = x e^{-\eta g - \nu \vec{1} + \lambda}$$

Note: We can create a point x st $x_i = \frac{1}{n}$ for all i from 1 to n. This point is strictly in the n-dimensional probability simplex or in the relative interior of this problem, since each element is greater than 0 and it satisfies the equality constraint. Thus, by Slater's condition, strong duality holds. As a result, we can use the necessary conditions of KKT to say that the optimal solution above must obey the KKT Conditions. By primal feasibility, the following is true.

$$\sum_{i=1}^{n} z_i = 1$$

$$z_i > 0 \forall i$$

As a result, λ and ν can be set to 0 as their constraints are proven to be met at the optimum, so we can simplify the solution to the following.

$$z = x e^{-\eta g}$$

3.1.4 Mirror Gradient Descent - d

We shall now compute Mirr$(\eta\vec{g};\vec{x})$ with $h(\vec{x}) = 1/2\|\vec{x}\|_2^2$

$$Mirr(\eta\vec{g};\vec{x}) - \eta\vec{g}^\top\vec{z} + \frac{1}{2}\|\vec{z}\|_2^2 - \vec{x}^\top\vec{z} - \vec{x}$$

If we take the gradient with respect to $\vec{z}$ and set it equal to zero we will find the solution to Mirr$(\eta\vec{g};\vec{x})$, which is the optimal solution.

$$\eta\vec{g} - \vec{z} - \vec{x} = \vec{0}$$
$$\vec{z} = \vec{x} - \eta\vec{g}$$

Our optimal solution is therefore

$$\vec{z} = \vec{x} - \eta\vec{g}$$

3.1.5 Mirror Gradient Descent - e

The regret is defined as $Reg_k(\vec{u}) = \langle\eta_k\vec{g}_k, \vec{x}_k - \vec{u}_k\rangle$. We will show that

$$Reg_k(\vec{u}) = \langle\eta_k g_k, \vec{x}_k - \vec{u}\rangle = \langle\eta_k g_k, \vec{x}_k - \vec{x}_{k+1}\rangle + D_h(\vec{u};\vec{x}_k) - D_h(\vec{u};\vec{x}_{k+1}) - D_h(\vec{x}_{k+1};\vec{x}_k)$$
$$= \frac{\eta_k^2\|g_k\|^2}{2} + D_h(\vec{u};\vec{x}_k) - D_h(\vec{u};\vec{x}_{k+1})$$

$$Reg_k(\vec{u}) = \langle\eta_k g_k, \vec{x}_k - \vec{u}\rangle = \langle\eta_k g_k, \vec{x}_k - \vec{x}_{k+1}\rangle + \langle\eta_k g_k, \vec{x}_{k+1} - \vec{u}\rangle$$

From the Mirror descent algorithm we derive

$$\eta_k\vec{g}_k = \vec{x}_k - \vec{x}_{k+1}$$

$$\langle\eta_k g_k, \vec{x}_k - \vec{x}_{k+1}\rangle + \langle\vec{x}_k - \vec{x}_{k+1}, \vec{x}_{k+1} - \vec{u}\rangle = \langle\eta_k g_k, \vec{x}_k - \vec{x}_{k+1}\rangle + \langle\nabla h(\vec{x}_k) - \nabla h(\vec{x}_{k+1}), \vec{x}_{k+1} - \vec{u}\rangle$$

From Bregman's three point inequality we derive:

$$\langle\eta_k g_k, \vec{x}_k - \vec{u}\rangle = \langle\eta_k g_k, \vec{x}_k - \vec{x}_{k+1}\rangle + D_h(\vec{u};\vec{x}_k) - D_h(\vec{u};\vec{x}_{k+1}) - D_h(\vec{x}_{k+1};\vec{x}_k)$$

We will now show that

$$\langle\eta_k g_k, \vec{x}_k - \vec{x}_{k+1}\rangle + D_h(\vec{u};\vec{x}_k) - D_h(\vec{u};\vec{x}_{k+1}) - D_h(\vec{x}_{k+1};\vec{x}_k) = \frac{\eta_k^2\|g_k\|^2}{2} + D_h(\vec{u};\vec{x}_k) - D_h(\vec{u};\vec{x}_{k+1})$$

First let us rearrange and then expand the product:

$$\langle\eta_k g_k, \vec{x}_k - \vec{x}_{k+1}\rangle + D_h(\vec{u};\vec{x}_k) - D_h(\vec{u};\vec{x}_{k+1}) - D_h(\vec{x}_{k+1};\vec{x}_k)$$
$$= \eta_k\vec{g}_k^T\vec{x}_k - \eta_k\vec{g}_k^T\vec{x}_{k+1} - \frac{1}{2}\|\vec{x}_{k+1}\|_2^2 + \frac{1}{2}\|\vec{x}_k\|_2^2 + \vec{g}_k^T\vec{x}_{k+1} - \vec{g}_k^T\vec{x}_k + D_h(\vec{u};\vec{x}_k) - D_h(\vec{u};\vec{x}_{k+1})$$

Substituting $\vec{x}_{k+1} = \vec{x}_k - \eta_k\vec{g}_k$ from Mirror Descent:

$$\eta_k\vec{g}_k^T\vec{x}_k - \eta_k\vec{g}_k^T(\vec{x}_k - \eta_k\vec{g}_k) - \frac{1}{2}\|\vec{x}_k - \eta_k\vec{g}_k\|_2^2 + \frac{1}{2}\|\vec{x}_k\|_2^2 + \vec{g}_k^T(\vec{x}_k - \eta_k\vec{g}_k) - \vec{g}_k^T\vec{x}_k + D_h(\vec{u};\vec{x}_k) - D_h(\vec{u};\vec{x}_{k+1})$$

$$\begin{aligned}
&\eta_k\vec{g}_k^T\vec{x}_k - \eta_k\vec{g}_k^T\vec{x}_k + \eta_k^2\|\vec{g}_k\|_2^2 - \frac{1}{2}(\|\vec{x}_k\|_2^2 - 2\eta_k\vec{x}_k^T\vec{g}_k + \eta_k^2\|\vec{g}_k\|_2^2) \\
&- \frac{1}{2}\|\vec{x}_k\|_2^2 + \|\vec{x}_k\|_2^2 - \eta_k\vec{x}_k^T\vec{g}_k + D_h(\vec{u};\vec{x}_k) - D_h(\vec{u};\vec{x}_{k+1}) \\
&= \eta_k^2\|\vec{g}_k\|_2^2 - \frac{1}{2}\eta_k^2\|\vec{g}_k\|_2^2 + D_h(\vec{u};\vec{x}_k) - D_h(\vec{u};\vec{x}_{k+1}) \\
&= \frac{1}{2}\eta_k^2\|\vec{g}_k\|_2^2 + D_h(\vec{u};\vec{x}_k) - D_h(\vec{u};\vec{x}_{k+1}).
\end{aligned} \tag{1}$$

Thus we have proven what we need to show.

3.1.6 Mirror Gradient Descent - f

Prove $\text{TotalReg}_T(u) \leq \sum_{k=0}^{T} \eta_k^2 \|\vec{g}_k\|_2^2 + D_h(\vec{u}; \vec{x}_0)$

$$\text{TotalReg}_K(u) = \sum_{k=0}^{K} \langle \eta_k \vec{g}_k, \vec{x}_k - \vec{u} \rangle$$

$$= \sum_{k=0}^{K} \frac{\eta_k^2 \|\vec{g}_k\|_2^2}{2} + D_h(\vec{u}; \vec{x}_k) - D_h(\vec{u}; \vec{x}_{k+1})$$

K = 0:

$$\frac{\eta_0^2 \|\vec{g}_0\|_2^2}{2} + D_h(\vec{u}; \vec{x}_0) - D_h(\vec{u}; \vec{x}_1)$$

K = 1:

$$\frac{\eta_0^2 \|\vec{g}_0\|_2^2}{2} + D_h(\vec{u}; \vec{x}_0) - D_h(\vec{u}; \vec{x}_1) + \frac{\eta_1^2 \|\vec{g}_1\|_2^2}{2} + D_h(\vec{u}; \vec{x}_1) - D_h(\vec{u}; \vec{x}_2)$$

$$= \frac{\eta_0^2 \|\vec{g}_0\|_2^2}{2} + \frac{\eta_1^2 \|\vec{g}_1\|_2^2}{2} + D_h(\vec{u}; \vec{x}_0) - D_h(\vec{u}; \vec{x}_2)$$

K = 2:

$$\frac{\eta_0^2 \|\vec{g}_0\|_2^2}{2} + D_h(\vec{u}; \vec{x}_0) - D_h(\vec{u}; \vec{x}_1) + \frac{\eta_1^2 \|\vec{g}_1\|_2^2}{2} + D_h(\vec{u}; \vec{x}_1) - D_h(\vec{u}; \vec{x}_2)$$

$$+ \frac{\eta_1^2 \|\vec{g}_2\|_2^2}{2} + D_h(\vec{u}; \vec{x}_2) - D_h(\vec{u}; \vec{x}_3)$$

$$= \frac{\eta_0^2 \|\vec{g}_0\|_2^2}{2} + \frac{\eta_1^2 \|\vec{g}_1\|_2^2}{2} + \frac{\eta_1^2 \|\vec{g}_2\|_2^2}{2} + D_h(\vec{u}; \vec{x}_0) - D_h(\vec{u}; \vec{x}_2)$$

k = T:

$$= \sum_{k=0}^{T} \frac{\eta_k^2 \|\vec{g}_k\|_2^2}{2} + D_h(\vec{u}; \vec{x}_0) - D_h(\vec{u}; \vec{x}_{T+1})$$

Note : $D_h(\vec{u}; \vec{x}_{T+1}) = \frac{\|\vec{u} - \vec{x}_0\|_2^2}{2} \geq 0$ as proved in part e and by positive definiteness of norms

$$\text{TotalReg}_T(u) = \sum_{k=0}^{T} \frac{\eta_k^2 \|\vec{g}_k\|_2^2}{2} + D_h(\vec{u}; \vec{x}_0) - D_h(\vec{u}; \vec{x}_{T+1})$$

$$\leq \sum_{k=0}^{T} \frac{\eta_k^2 \|\vec{g}_k\|_2^2}{2} + D_h(\vec{u}; \vec{x}_0)$$

$$\leq \sum_{k=0}^{T} \eta_k^2 \|\vec{g}_k\|_2^2 + D_h(\vec{u}; \vec{x}_0) \qquad \text{Since we know } \eta_k^2 \|\vec{g}_k\|_2^2 \geq 0 \quad \forall k$$

3.1.7 Mirror Gradient Descent - g

Prove $TotalRegret_T(\vec{u}) \geq T\eta(f(\vec{x}_T) - f(\vec{u})) \quad where \quad \eta_k = \eta \quad \forall k$

$$\eta \sum_{k=0}^{T} \vec{g}_k^T (\vec{x}_k - \vec{u}) \geq T\eta(f(\vec{x}_T) - f(\vec{u}))$$

At every step of mirror descent, we can think of the queried gradient as a hyperplane lower bounding the objective function (from lit review research paper 2). This is because we know a convex function is lower bounded by its first order Taylor's approximation at every point

$$\forall \vec{u} \quad f(\vec{u}) \geq f(\vec{x}) + \nabla f(\vec{x})^T (\vec{u} - \vec{x})$$

We can extend this to the convex combination of all gradients from our T steps:

$$\forall \vec{u} \quad f(\vec{u}) \geq \frac{1}{T}\sum_{k=0}^{T} f(\vec{x}_k) + \frac{1}{T}\sum_{k=0}^{T} \nabla f(\vec{x}_k)^T(\vec{u} - \vec{x}_k)$$

We know $f(\bar{x}_T) \leq \frac{1}{T}\sum_{k=0}^{T} f(\vec{x}_k)$ from Jensen's Inequality

$$\forall \vec{u} \quad f(\vec{u}) - f(\bar{x}_T) \geq \frac{1}{T}\sum_{k=0}^{T} \nabla f(\vec{x}_k)^T(\vec{u} - \vec{x}_k)$$

$$f(\bar{x}_T) - f(\vec{u}) \leq \frac{1}{T}\sum_{k=0}^{T} \nabla f(\vec{x}_k)^T(\vec{u} - \vec{x}_k)$$

$$\sum_{k=0}^{T} \nabla f(\vec{x}_k)^T(\vec{u} - \vec{x}_k) \geq T(f(\bar{x}_T) - f(\vec{u}))$$

$$\eta \sum_{k=0}^{T} \nabla f(\vec{x}_k)^T(\vec{u} - \vec{x}_k) \geq T\eta(f(\bar{x}_T) - f(\vec{u}))$$

$$\eta_k = \eta, \quad \vec{g}_k = \nabla f(\vec{x}_k) \quad \forall k$$

$$\sum_{k=0}^{T} \eta_k \vec{g}_k{}^T(\vec{x}_k - \vec{u}) \geq T\eta(f(\bar{x}_T) - f(\vec{u}))$$

The left side of this inequality is $TotalReg_T(\vec{u})$, so we conclude:

$$TotalReg_T(\vec{u}) \geq T\eta(f(\bar{x}_T) - f(\vec{u}))$$

For $\eta_k = \eta \quad \forall k, \quad Let \quad \vec{u} = \vec{x}^*$, we can combine this with the upper bound found in f:

$$T\eta(f(\bar{x}_T) - f(\vec{x}^*)) \leq TotalReg_T(\vec{x}^*) \leq \sum_{k=0}^{T} \eta_k^2 \|\vec{g}_k\|_2^2 + D_h(\vec{x}^*; \vec{x}_0)$$

$$T\eta(f(\bar{x}_T) - f(\vec{x}^*)) \leq \eta^2 \sum_{k=0}^{T} \|\vec{g}_k\|_2^2 + D_h(\vec{x}^*; \vec{x}_0)$$

$$\eta(f(\bar{x}_T) - f(\vec{x}^*)) \leq \frac{1}{T}\eta^2 \sum_{k=0}^{T} \|\vec{g}_k\|_2^2 + D_h(\vec{x}^*; \vec{x}_0)$$

$$\eta f(\bar{x}_T) \leq \eta f(\vec{x}^*) + \frac{1}{T}\eta^2 \sum_{k=0}^{T} \|\vec{g}_k\|_2^2 + D_h(\vec{x}^*; \vec{x}_0)$$

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \frac{1}{T}\eta \sum_{k=0}^{T} \|\vec{g}_k\|_2^2 + \frac{D_h(\vec{x}^*; \vec{x}_0)}{\eta}$$

So, we can conclude that $f(\bar{x}_T) \leq f(\vec{x}^*) + \frac{1}{T}[\eta \sum_{k=0}^{T} \|\vec{g}_k\|_2^2 + \frac{D_h(\vec{x}^*; \vec{x}_0)}{\eta}]$

3.1.8 Mirror Gradient Descent - h

Let f be L-Lipschitz such that $\|\nabla f(\vec{x})\|_2 \leq L \forall \vec{x} \epsilon X$

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \eta L^2 + \frac{1}{2\eta T}\|\vec{x}_0 - \vec{x}^*\|_2^2$$

From part g)

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \frac{1}{T}[\eta \sum_{i=0}^{T} \|\vec{g}_k\|_2^2 + \frac{1}{\eta} D_H(\vec{x}^*; \vec{x}_0)$$

Substituting $\vec{g}_k = \nabla f(\vec{x}_k)$

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \frac{1}{T}[\eta \sum_{i=0}^{T} \|\nabla f(\vec{x}_k)\|_2^2 + \frac{1}{\eta} D_H(\vec{x}^*; \vec{x}_0)]$$

Taking advantage of the fact that $\nabla f(\vec{x}_k) \leq L$:

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \frac{1}{T}[\eta \sum_{i=0}^{T} L^2 + \frac{1}{\eta} D_H(\vec{x}^*; \vec{x}_0)]$$

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \frac{1}{T} T \eta L^2 + \frac{1}{\eta T} D_H(\vec{x}^*; \vec{x}_0)$$

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \eta L^2 + \frac{1}{\eta T} D_H(\vec{x}^*; \vec{x}_0)$$

We will now expand $D_H(\vec{x}^*; \vec{x}_0)$

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \eta L^2 + \frac{1}{2\eta T}(\|\vec{x}^*\|_2^2 - \|\vec{x}_0\|_2^2 - 2\vec{x}_0^T \vec{x}^* + 2\|\vec{x}_0\|_2^2)$$

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \eta L^2 + \frac{1}{2\eta T}(\|\vec{x}^*\|_2^2 + \|\vec{x}_0\|_2^2 - 2\vec{x}_0^T \vec{x}^*)$$

Rewriting the last the expression as one norm:

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \eta L^2 + \frac{1}{2\eta T}\|\vec{x}^* - \vec{x}_0\|_2^2$$

As $\|\vec{x}^* - \vec{x}_0\|_2^2 = \|\vec{x}_0 - \vec{x}^*\|_2^2$, we can conclude

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \eta L^2 + \frac{1}{2\eta T}(\|\vec{x}_0 - \vec{x}^*\|_2^2$$

Now consider $\eta > 0$, we will show that

$$f(\bar{x}) \leq f(\vec{x}^*) + \frac{\sqrt{2} L \|\vec{x}_0 - \vec{x}^*\|_2}{\sqrt{T}}$$

To achieve this, we will the gradient of the inequality derived about to find a value of η to show that this inequality can be satisfied.

$$\nabla(f(\vec{x}^*) + \eta L^2 + \frac{1}{2\eta T}(\|\vec{x}_0 - \vec{x}^*\|_2^2) = L^2 - \frac{1}{2\eta T}\|\vec{x}_0 - \vec{x}^*\|_2^2 = 0$$

Solving for η that satisfies this equality, we derive

$$\eta^* = \frac{\|\vec{x}_0 - \vec{x}^*\|_2}{\sqrt{2}\sqrt{T} L}$$

We plug this back into our original inequality:

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \eta L^2 + \frac{1}{2\eta T}\|\vec{x}_0 - \vec{x}^*\|_2^2$$

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \frac{\|\vec{x}_0 - \vec{x}^*\|_2}{\sqrt{2}\sqrt{T}L}L^2 + \frac{1}{2\frac{\|\vec{x}_0 - \vec{x}^*\|_2}{\sqrt{2}\sqrt{T}L}T}\|\vec{x}_0 - \vec{x}^*\|_2^2$$

Rearranging this we attain:

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \frac{\|\vec{x}_0 - \vec{x}^*\|_2}{\sqrt{2}\sqrt{T}}L + \frac{L}{\sqrt{2}\sqrt{T}}\|\vec{x}_0 - \vec{x}^*\|_2$$

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \frac{2L}{\sqrt{2}\sqrt{T}}\|\vec{x}_0 - \vec{x}^*\|_2$$

$$f(\bar{x}_T) \leq f(\vec{x}^*) + \frac{\sqrt{2}L}{\sqrt{T}}\|\vec{x}_0 - \vec{x}^*\|_2$$

3.2 Accelerated Gradient Descent

3.2.1 Accelerated Gradient Descent - a

Prove $\langle \eta_{k+1}\nabla f(\vec{x}_{k+1}), z_k - \vec{u}\rangle = \frac{\eta_{k+1}^2}{2}\|\nabla f(\vec{x}_{k+1})\|_2^2 + D_h(\vec{u}; z_k) - D_h(\vec{u}; z_{k+1})$

From mirror descent part (e) we know

$$\langle \eta_k \vec{g_k}, \vec{x_k} - \vec{u}\rangle = \frac{\eta_k^2\|\vec{g_k}\|_2^2}{2} + D(\vec{u}; \vec{x_k}) - D(\vec{u}; \vec{x_{k+1}})$$

We know that these follow a very similar form as (e). In addition, other than notation changes, we see we utilize no assumptions about $\vec{g}$ in order to prove. So, this will hold when $\vec{g}$ is a form other than $\nabla f(\vec{z_k})$
To formally prove, we observe from (e)

$$\langle \eta_k \vec{g_k}, \vec{x_k} - \vec{u}\rangle = \langle \eta_k \vec{g_k}, \vec{x_k} - \vec{x_{k+1}}\rangle + D_h(\vec{u}; x_k) - D_h(\vec{u}; x_{k+1}) - D_h(\vec{x_{k+1}}; \vec{x_k})$$

$$= \eta_k^2\|\vec{g_k}\|_2^2 2 + D_h(\vec{u}; \vec{x_k}) - D_h(\vec{u}; \vec{x_{k+1}})$$

Through simplifications, we have

$$\langle \eta_{k+1}\nabla f(\vec{x}_{k+1}), z_k - \vec{u}\rangle = \langle \eta_{k+1}\nabla f(\vec{x}_{k+1}), \vec{z_k} - \vec{z_{k+1}}\rangle + D_h(\vec{u}; z_k) - D_h(\vec{u}; z_{k+1}) - D_h(\vec{u}; \vec{z_{k+1}})$$

$$\langle \eta_{k+1}\nabla f(\vec{x}_{k+1}), \vec{z_k} - \vec{z_{k+1}}\rangle - D_h(\vec{z_k}; \vec{z_{k+1}}) = \frac{\eta_{k+1}^2\|\nabla f(\vec{x}_{k+1})\|_2^2}{2}$$

$$\eta_{k+1}\nabla f(\vec{x}_{k+1})^T\vec{z_k} - \eta_{k+1}\nabla f(\vec{x}_{k+1})^T\vec{z_{k+1}} - \frac{1}{2}\|\vec{z_{k+1}}\|_2^2 + \frac{1}{2}\|\vec{z_k}\|_2^2 + \vec{z_k}^T\vec{z_{k+1}} - \|\vec{z_k}\|_2^2 = \frac{\eta_{k+1}^2\|\nabla f(\vec{x}_{k+1})\|_2^2}{2}$$

From Mirror Descent we know

$$\vec{z_{k+1}} = \vec{z_k} - \eta_{k+1}\nabla f(\vec{x}_{k+1})$$

So, we plug that into our previous equation and simplify to get:

$$\eta_{k+1}^2\|\nabla f(\vec{x}_{k+1})\|_2^2 - \frac{1}{2}\eta_{k+1}^2\|\nabla f(\vec{x}_{k+1})\|_2^2 = \frac{1}{2}\eta_{k+1}^2\|\nabla f(\vec{x}_{k+1})\|_2^2$$

$$\frac{1}{2}\eta_{k+1}^2\|\nabla f(\vec{x}_{k+1})\|_2^2 = \frac{1}{2}\eta_{k+1}^2\|\nabla f(\vec{x}_{k+1})\|_2^2$$

So equality holds

3.2.2 Accelerated Gradient Descent - b

Prove $\langle \eta_{k+1}\nabla f(\vec{x}_{k+1}), \vec{x}_{k+1} - \vec{u}\rangle = \frac{(1-\tau_k)\eta_{k+1}}{\tau_k}\langle \nabla f(\vec{x}_{k+1}), \vec{y_k} - \vec{x}_{k+1}\rangle + \frac{1}{2}\eta_{k+1}^2\|\nabla f(\vec{x}_{k+1})\|_2^2 + D_h(\vec{u};\vec{z}_k) - D_h(\vec{u};\vec{z}_{k+1})$

We know from (a) we have $\langle \eta_{k+1}\nabla f(\vec{x}_{k+1}), z_k - \vec{u}\rangle = \frac{\eta_{k+1}^2}{2}\|\nabla f(\vec{x}_{k+1})\|_2^2 + D_h(\vec{u};\vec{z}_k) - D_h(\vec{u};\vec{z}_{k+1})$
We also know $\vec{x}_{k+1} = \tau_k\vec{z_k} + (1-\tau_k)\vec{y_k}$
So we can instead write:

$$\langle \eta_{k+1}\nabla f(\vec{x}_{k+1}), \vec{x}_{k+1} - \vec{u}\rangle = \frac{(1-\tau_k)\eta_{k+1}}{\tau_k}\langle \nabla f(\vec{x}_{k+1}), \vec{y_k}\rangle + \langle \eta_{k+1}\nabla f(\vec{x}_{k+1}), z_k - \vec{u}\rangle$$

$$\langle \nabla f(\vec{x}_{k+1}), \vec{x}_{k+1}\rangle = \frac{(1-\tau_k)}{\tau_k}\langle \nabla f(\vec{x}_{k+1}), \vec{y_k}\rangle + \langle \nabla f(\vec{x}_{k+1}), z_k\rangle$$

We can plug in our value for $\vec{x}_{k+1}$ to get:

$$\tau_k\nabla f(\vec{x}_{k+1})^T\vec{z}_k + (1-\tau_k)\nabla f(\vec{x}_{k+1})^T\vec{y}_k - \nabla f(\vec{x}_{k+1})^T\vec{z}_k = \frac{(1-\tau_k)}{\tau_k}\nabla f(\vec{x}_{k+1})^T(\vec{y}_k - \tau_k\vec{z}_k - \vec{y}_k + \tau_k\vec{y}_k)$$

$$(1-\tau_k)\nabla f(\vec{x}_{k+1})^T(\vec{y}_k - \vec{z}_k) = \frac{(1-\tau_k)}{\tau_k}\nabla f(\vec{x}_{k+1})^T(\tau_k\vec{y}_k - \tau_k\vec{z}_k)$$

$$(1-\tau_k)\nabla f(\vec{x}_{k+1})^T(\vec{y}_k - \vec{z}_k) = (1-\tau_k)\nabla f(\vec{x}_{k+1})^T(\vec{y}_k - \vec{z}_k)$$

So, equality holds

Prove the right hand side can be upper bounded by

$$\frac{(1-\tau_k)\eta_{k+1}}{\tau_k}(f(\vec{y}_k) - f(\vec{x}_{k+1}) + \eta_{k+1}^2 L((f(\vec{y}_k) - f(\vec{x}_{k+1})) + D_h(\vec{u};\vec{z}_k) - D_h(\vec{u};\vec{z}_{k+1})$$

$$\frac{(1-\tau_k)\eta_{k+1}}{\tau_k}\langle \nabla f(\vec{x}_{k+1}), \vec{y_k} - \vec{x}_{k+1}\rangle + \frac{1}{2}\eta_{k+1}^2\|\nabla f(\vec{x}_{k+1})\|_2^2 \leq \frac{(1-\tau_k)\eta_{k+1}}{\tau_k}(f(\vec{y}_k) - f(\vec{x}_{k+1})) + \eta_{k+1}^2 L(f(\vec{x}_{k+1}) - f(\vec{y}_{k+1}))$$

We can prove this term-wise by individually proving the following:

$$1.)\frac{(1-\tau_k)\eta_{k+1}}{\tau_k}\langle \nabla f(\vec{x}_{k+1}), \vec{y_k} - \vec{x}_{k+1}\rangle \leq \frac{(1-\tau_k)\eta_{k+1}}{\tau_k}(f(\vec{y}_k) - f(\vec{x}_{k+1}))$$

$$2.)\frac{1}{2}\eta_{k+1}^2\|\nabla f(\vec{x}_{k+1})\|_2^2 \leq \eta_{k+1}^2 L(f(\vec{x}_{k+1}) - f(\vec{y}_{k+1}))$$

We start with the first inequality:

$$\frac{(1-\tau_k)\eta_{k+1}}{\tau_k}\langle \nabla f(\vec{x}_{k+1}), \vec{y_k} - \vec{x}_{k+1}\rangle \leq \frac{(1-\tau_k)\eta_{k+1}}{\tau_k}(f(\vec{y}_k) - f(\vec{x}_{k+1}))$$

$$\langle \nabla f(\vec{x}_{k+1}), \vec{y_k} - \vec{x}_{k+1}\rangle \leq f(\vec{y}_k) - f(\vec{x}_{k+1})$$

$$-f(\vec{y}_k) \leq -f(\vec{x}_{k+1}) - \langle \nabla f(\vec{x}_{k+1}), \vec{y_k} - \vec{x}_{k+1}\rangle$$

$$f(\vec{y}_k) \geq f(\vec{x}_{k+1}) + \langle \nabla f(\vec{x}_{k+1}), \vec{y_k} - \vec{x}_{k+1}\rangle$$

This last term is true because $f(\vec{x})$ is a convex function. This means it follows the first-order of convexity, which states exactly

$$f(\vec{y}_k) \geq f(\vec{x}_{k+1}) + \langle \nabla f(\vec{x}_{k+1}), \vec{y_k} - \vec{x}_{k+1}\rangle$$

Because it follows exactly, we have proven that inequality 1.) holds.

For the second inequality:

$$\frac{1}{2}\eta_{k+1}^2\|\nabla f(\vec{x}_{k+1})\|_2^2 \leq \eta_{k+1}^2 L(f(\vec{x}_{k+1}) - f(\vec{y}_{k+1}))$$

$$\frac{1}{2}\|\nabla f(\vec{x}_{k+1})\|_2^2 \leq L(f(\vec{x}_{k+1}) - f(\vec{y}_{k+1}))$$

$$f(\vec{x}_{k+1}) - f(\vec{y}_{k+1}) \geq \frac{1}{2L}\|\nabla f(\vec{x}_{k+1})\|_2^2$$

$$f(\vec{y}_{k+1}) \leq f(\vec{x}_{k+1}) - \frac{1}{2L}\|\nabla f(\vec{x}_{k+1})\|_2^2$$

We proved in lecture for L-smooth functions:

$$f(\vec{y}) \leq f(\vec{x}) - \frac{1}{2L}\|\nabla f(\vec{x})\|_2^2$$

where $\vec{y} = \vec{x} - \frac{1}{L}\nabla f(\vec{x})$ is the gradient descent step
$\vec{y}_{k+1} = \vec{x}_{k+1} - \frac{1}{L}\nabla f(\vec{x}_{k+1})$ is the gradient descent step here, so it holds that

$$f(\vec{y}_{k+1}) \leq f(\vec{x}_{k+1}) - \frac{1}{2L}\|\nabla f(\vec{x}_{k+1})\|_2^2$$

So, inequality 2.) holds.
Both 1.) and 2.) hold, so we can conclude that the right hand side can in fact be upper bounded.

$$\frac{(1-\tau_k)\eta_{k+1}}{\tau_k}\langle \nabla f(\vec{x}_{k+1}), \vec{y}_k - \vec{x}_{k+1}\rangle + \frac{1}{2}\eta_{k+1}^2\|\nabla f(\vec{x}_{k+1})\|_2^2 \leq \frac{(1-\tau_k)\eta_{k+1}}{\tau_k}(f(\vec{y}_k) - f(\vec{x}_{k+1})) + \eta_{k+1}^2 L(f(\vec{x}_{k+1}) - f(\vec{y}_{k+1}))$$

3.2.3 Accelerated Gradient Descent - c

The intuition behind this proof was derived from Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent. We are trying to prove:

$$\eta_{k+1}^2 L f(\vec{y}_{k+1}) - (\eta_{k+1}^2 L - \eta_{k+1}) f(\vec{y}_k) - D(\vec{u}; \vec{z}_k) + D(\vec{u}; \vec{z}_{k+1}) \leq \eta_{k+1} f(\vec{u})$$

We begin with the follow inequalities

$$\begin{aligned}
\eta_{k+1}(f(\vec{x}_{k+1}) - f(\vec{u})) &\leq \eta_{k+1}\langle \nabla f(\vec{x}_{k+1}), \vec{x}_{k+1} - \vec{u}\rangle \\
&= \eta_{k+1}\langle \nabla f(\vec{x}_{k+1}), \vec{x}_{k+1} - \vec{z}_k\rangle + \eta_{k+1}\langle \nabla f(\vec{x}_{k+1}), \vec{z}_k - \vec{u}\rangle \\
&\textit{Let } x_{k+1} \textit{ satisfy } \tau_k(\vec{x}_{k+1} - \vec{z}_k) = (1 - \vec{t_k})(\vec{y}_k - \vec{x}_{k+1}) \\
&\leq \frac{(1-\tau_k)\eta_{k+1}}{\tau_k}\langle \nabla f(\vec{x}_{k+1}), \vec{y}_k - \vec{x}_{k+1}\rangle + \eta_{k+1}\langle \nabla f(\vec{x}_{k+1}), \vec{z}_k - \vec{u}\rangle \\
&\textit{As } f \textit{ is convex and } 1 - \tau_k \geq 0 \\
&\leq \frac{(1-\tau_k)\eta_{k+1}}{\tau_k}(f(\vec{y}_k) - f(\vec{x}_{k+1})) + \eta_{k+1}\langle \nabla f(\vec{x}_{k+1}), \vec{z}_k - \vec{u}\rangle \\
&\textit{By part 2b,} \\
&\leq \frac{(1-\tau_k)\eta_{k+1}}{\tau_k}(f(\vec{y}_k) - f(\vec{x}_{k+1})) + \eta_{k+1} L(f(\vec{x}_{k+1}) - f(\vec{y}_{k+1})) + V_{\vec{z}_k}(\vec{u}) - V_{\vec{z}_{k+1}}(\vec{u}) \\
&\textit{As } \tau = \frac{1}{\eta_{k+1} L} \\
&= (\eta_{k+1}^2 L - \eta_{k+1}) f(\vec{y}_k) - (\eta_{k+1}^2 L) f(\vec{y}_{k+1}) + \eta_{k+1} f(\vec{x}_{k+1}) + (D_{\vec{z}_k}(\vec{u}) - D_{\vec{z}_{k+1}}(\vec{u}))
\end{aligned}$$

Multiplying both side of the equation by -1 and subtracting $\eta_{k+1} f(\vec{x}_{k+1})$, we get

$$\eta_{k+1}^2 L f(\vec{y}_{k+1}) - (\eta_{k+1}^2 L - \eta_{k+1}) f(\vec{y}_k) - D(\vec{u}; \vec{z}_k) + D(\vec{u}; \vec{z}_{k+1}) \leq \eta_{k+1} f(\vec{u})$$

3.2.4 Accelerated Gradient Descent - d

Prove

$$f(y_T) \leq f(x^*) + \frac{2 \cdot L||\vec{x^*}\vec{x}_0||_2^2}{(T+1)^2}$$

We know:

$$\eta_{k+1}^2 L f(\vec{y}_{k+1}) - (\eta_{k+1}^2 \cdot L - \eta_{k+1}) f(\vec{y}_k) - D(\vec{u}, \vec{z}_k) + D(\vec{u}, \vec{z}_{k+1}) \leq \eta_{k+1} f(\vec{u})$$

$$\eta_{k+1} = \frac{k+2}{2L}$$

$$\eta_k = \frac{k+1}{2L}$$

$$\eta_k^2 L = \eta_{k+1}^2 L - \eta_{k+1} + \frac{1}{4L}$$

We use induction to prove the telescoping sum of this sequence over T iterations is equal to:

$$\eta_T^2 L f(y_T) + \sum_{k=1}^{T-1} \frac{1}{4L} f(\vec{y}_k) + D(\vec{u}, \vec{z_T}) - D(u, z_0) \leq \sum_{k=1}^{T} \eta_k f(\vec{u})$$

1. **Base Case ($T = 0$):**

$$\eta_1^2 L f(\vec{y}_1) - (\eta_1^2 \cdot L - \eta_1) f(\vec{y}_0) - D(\vec{u}, \vec{z}_0) + D(\vec{u}, \vec{z}_1) \leq \eta_1 f(\vec{u})$$

2. **Inductive Step (T to T+1):**

$$\eta_T^2 L f(y_T) + \sum_{k=1}^{T-1} \frac{1}{4L} f(\vec{y}_k) + D(\vec{u}, \vec{z_T}) - D(u, z_0) \leq \sum_{k=1}^{T} \eta_k f(\vec{u})$$

$$+\eta_{T+1}^2 L f(\vec{y}_{T+1}) - (\eta_{T+1}^2 \cdot L - \eta_{T+1}) f(\vec{y}_T) - D(\vec{u}, \vec{z}_T) + D(\vec{u}, \vec{z}_{T+1}) \leq \eta_{T+1} f(\vec{u})$$

$$= \eta_T^2 L f(y_T) + \sum_{k=1}^{T-1} \frac{1}{4L} f(\vec{y}_k) + D(\vec{u}, \vec{z_T}) - D(\vec{u}, \vec{z_0}) \leq \sum_{k=1}^{T} \eta_k f(\vec{u})$$

$$+\eta_{T+1}^2 L f(\vec{y}_{T+1}) - (\eta_T^2 L + \eta_{T+1} - \frac{1}{4L} - \eta_{T+1}) f(\vec{y}_T) - D(\vec{u}, \vec{z}_T) + D(\vec{u}, \vec{z}_{T+1}) \leq \eta_{T+1} f(\vec{u})$$

$$= \sum_{k=1}^{T-1} \frac{1}{4L} f(\vec{y}_k) - D(\vec{u}, \vec{z_0}) \leq \sum_{k=1}^{T} \eta_k f(\vec{u})$$

$$+\eta_{T+1}^2 L f(\vec{y}_{T+1}) - (-\frac{1}{4L}) f(\vec{y}_T) + D(\vec{u}, \vec{z}_{T+1}) \leq \eta_{T+1} f(\vec{u})$$

$$= \eta_{T+1}^2 L f(\vec{y}_{T+1}) + \sum_{k=1}^{T} \frac{1}{4L} f(\vec{y}_k) + D(u, z_{T+1}) - D(u, z_0) \leq \sum_{k=1}^{T+1} \eta_k f(u)$$

Since this summation over $T+1$ iterations is equivalent to the formula when $T+1$ is substituted for T, this summation holds from $T = 0$ to all positive integers.

Letting $u = x^*$:

$$\eta_T^2 L f(y_T) + \sum_{k=1}^{T-1} \frac{1}{4L} f(\vec{y}_k) + D(\vec{x^*}, \vec{z_T}) - D(x^*, \vec{z_0}) \leq \sum_{k=1}^{T} \eta_k f(\vec{x^*})$$

$$= \frac{(T+1)^2}{4L^2} L f(y_T) + \sum_{k=1}^{T-1} \frac{1}{4L} f(\vec{y}_k) + D(\vec{x^*}, \vec{z_T}) - D(x^*, z_0) \leq \sum_{k=1}^{T} \eta_k f(\vec{x^*})$$

$$\frac{(T+1)^2}{4L^2}Lf(y_T) - \frac{1}{2}\|\vec{z_T} - \vec{z_0}\|_2^2 \leq \sum_{k=1}^{T} \eta_k f(\vec{x^*}) - \sum_{k=1}^{T-1} \frac{1}{4L} f(\vec{y_k})$$

Since we know $\vec{y_k} \geq \vec{x^*}$:

$$\frac{(T+1)^2}{4L^2}Lf(y_T) - \frac{1}{2}\|\vec{z_T} - \vec{z_0}\|_2^2 \leq \sum_{k=1}^{T} \eta_k f(\vec{x^*}) - \sum_{k=1}^{T-1} \frac{1}{4L} f(\vec{x^*})$$

$$\frac{(T+1)^2}{4L^2}Lf(y_T) - \frac{1}{2}\|\vec{z_T} - \vec{z_0}\|_2^2 \leq \sum_{k=1}^{T} \eta_k f(\vec{x^*}) - \frac{T-1}{4L} f(\vec{x^*})$$

We know $\sum_{k=1}^{T} \eta_k = \left(\frac{2}{2L} + \frac{T+1}{2L}\right) \cdot \frac{T}{2} = \frac{T(T+3)}{4L}$:

$$\frac{(T+1)^2}{4L^2}Lf(y_T) - \frac{1}{2}\|\vec{z_T} - \vec{z_0}\|_2^2 \leq \frac{T(T+3)}{4L} f(\vec{x^*}) - \frac{T-1}{4L} f(\vec{x^*})$$

$$\frac{(T+1)^2}{4L^2}Lf(y_T) - \frac{1}{2}\|\vec{z_T} - \vec{z_0}\|_2^2 \leq \left(\frac{T(T+3)}{4L} - \frac{T-1}{4L}\right) f(\vec{x^*})$$

$$\frac{(T+1)^2}{4L}f(y_T) - \frac{1}{2}\|\vec{z_T} - \vec{z_0}\|_2^2 \leq \left(\frac{(T+1)^2}{4L}\right) f(\vec{x^*})$$

$$f(y_T) \leq f(\vec{x^*}) + \frac{2L\|\vec{z_T} - \vec{z_0}\|_2^2}{(T+1)^2}$$

Please complete the guided problem set in the project assignment. Feel free to add more sections/subsections as necessary.

4 Extensions

Please describe your project extension as per the project assignment. Feel free to add more sections/subsections as necessary.

4.1 Main Idea

To add upon our research on Accelerated Gradient Descent using Nesterov's momentum or the linear coupling between gradient and mirror descent, we wanted to analyze the performance of Nesterov's Accelerated Gradient Descent, Stochastic Gradient Descent, Adam, AdaGrad, Momentum Gradient Descent, and RMS Prop in the context of convergence to the optimum for recurrent neural networks (RNN). Earlier, we explored the performance of these various methods for classification using a logistic loss function and its gradient. We wanted to extend these methods to more complex applications, and after reading Sutskever's paper on training recurrent neural networks (RNNs), we noticed he used Nesterov's gradient descent for improved performance and wanted to explore this behavior on our own. As opposed to logistic regression, neural networks use backpropagation, calculating the loss at the end of the network and the gradient of this loss function layer by layer of the neural network till the beginning. Recurrent neural networks in particular use backpropagation *through time*, which also propagates the error through time as RNN's are built for time-series applications. Weinein E et. al. explored the convergence rate and solution of Adam and SGD, finding that SGD converged to better solutions, often minimas in flatter valleys, while Adam converged faster but often to less optimal solutions. We would like to expand upon this paper by exploring convergence and optimality of more gradient descent methods.

4.2 Methodology

To analyze the impact of various gradients descent models we tested each model's performance on predicting future temperature based on past temperature from this data set. We used NumPy, TensorFlow, and Pandas to create various sample RNN. We benchmarked each version of gradient descent based on validation loss over each epoch. First, we cleaned the data. This involved filling NaNs with 0s and processing all the string entries in the dataframe. We converted them to numerical values through a direct replacement, not adding any extra columns to the dataframe. This was done because the dimensionality of the data would have become much too difficult to work with. After the data was cleaned, we took a subset of 10,000 entries to work with out of the 96,000 total in the dataset. Then, we created sequences of length 72. data entries were time stamped by the hour, so this was the equivalent of knowing the temperature and other weather measurements every hour for the last 3 full days. After generating these sequences, we split it into train and test, with 80% becoming train data and 20% test data. Then, we trained a series of RNN models with different optimizer functions. We trained 8,000 entries over 20 epochs, recording loss and MAE. We recorded the loss of each gradient descent method in both train data and validation data for every epoch.

4.3 Overview of Each Gradient Descent Method

4.3.1 Normal Gradient Descent

The original method of gradient descent calculates the gradient of the function at each time step and determines the next position with that and a learning rate η. Although computationally expensive, traditionally gradient descent is guaranteed to to converge to a local minimum but may get stuck there without reaching a global minimum. Below is the algorithm for this gradient descent where $\vec{x}_k$ is the parameter vector, η is the learning rate, and $\nabla f(\vec{x}_k)$ is the gradient of the objective function at time k.

$$\vec{x}_{k+1} = \vec{x}_k - \eta \nabla f(\vec{x}_k) \tag{2}$$

4.3.2 Stochastic Gradient Descent

Stochastic Gradient Descent is a variant of the standard Gradient Descent algorithm that incorporates randomness into the optimization by sampling a single data point per step. By only testing one training example per iteration, Stochastic Gradient Descent performs much more quickly than standard Gradient Descent and avoids bias through the random sampling. However, this also leads to a volatile cost function that can change significantly and unpredictably per iteration due to the randomness. Below, we can see that instead of taking the gradient of the loss of the entire dataset, we only compute it for the gradient of the loss of a single training example.

$$\vec{x}_{k+1} = \vec{x}_k - \eta \nabla f_i(\vec{x}_k) \tag{3}$$

4.3.3 Momentum Gradient Descent

This type of gradient descent adds a momentum term that increases the gradient's value as the number of steps increase, reducing variation in the cost function's affect on the gradient. This serves to increase the convergence rate and can more directly approach minima of noisier functions or functions with high degrees of curvature. The implementation of momentum gradient descent is below with $\vec{x}_k$ as the parameter for the cost function, η as the learning rate, and μ as the hyper parameter denoting the momentum constant :

$$\vec{b}_i = \mu * \vec{b}_{i-1} + \nabla f(\vec{x}_{i-1}) \tag{4}$$

$$\vec{x}_i = \vec{x}_{i-1} - \eta \vec{b}_i \tag{5}$$

4.3.4 Adagrad

Adagrad is a form of gradient descent that ensures a stable descent pattern by adjusting the learning rates according to recent gradient step sizes. For a large or steep gradient, Adagrad will set the learning rate

to be smaller, and for shallower gradient steps, Adagrad will amplify the learning rate in order to ensure the updates are roughly similar across all iterations and to converge quickly. For our update rule here, we constantly update a diagonal matrix G with the squared gradients per iteration and then use it to normalize the learning rate. To do this, we divide the learning rate by G plus a small epsilon to eliminate any dividing-by-zero errors, as shown below.

$$\vec{g}_k = \nabla f(\vec{x}_k) \tag{6}$$

$$\vec{G}_k = \vec{G}_{k-1} + \vec{g}_k^2 \tag{7}$$

$$\vec{x}_{k+1} = \vec{x}_k - \frac{\eta}{\sqrt{\vec{G}_k + \epsilon}} \vec{g}_k \tag{8}$$

4.3.5 Nesterov's Accelerated Gradient Descent

Explored earlier in this paper, Nesterov's Accelerated Gradient Descent combines the advantages of mirror gradient descent over traditional gradient descent and essentially looks a step ahead when calculating the next gradient to more quickly converge. As we have shown earlier in this paper, the convergence of Nesterov's accelerated gradient descent is $O(\frac{1}{n^2})$. The algorithm is defined as follows for time T from Allen-Zhu and Orecchia: Let f be a differentiable and convex function that L-smooth,

Define $D_h(\vec{y}) = h(\vec{y}) - h(\vec{x}) - \langle \nabla w(\vec{x}), \vec{y} - \vec{x} \rangle$.
$\vec{y}_0 \leftarrow \vec{x}_0$, $\vec{z}_0 \leftarrow \vec{x}_0$.
for $k \leftarrow 0$ to $T-1$ **do**
  $\alpha_{k+1} \leftarrow \frac{k+2}{2L}$, and $\tau_k \leftarrow \frac{1}{\alpha_{k+1} L} = \frac{2}{k+2}$.
  $\vec{x}_{k+1} \leftarrow \tau_k \vec{z}_k + (1-\tau_k)\vec{y}_k$.
  $\vec{y}_{k+1} \leftarrow \text{Grad}(\vec{x}_{k+1})$ ▷ $\vec{y}_{k+1} = \arg\min_{\vec{y} \in Q} \left\{ \frac{L}{2} \|\vec{y} - \vec{x}_{k+1}\|^2 + \langle \nabla f(\vec{x}_{k+1}), \vec{y} - \vec{x}_{k+1} \rangle \right\}$
  $\vec{z}_{k+1} \leftarrow \text{Mirr}_{\vec{z}_k}(\alpha_{k+1} \nabla f(\vec{x}_{k+1}))$ ▷ $\vec{z}_{k+1} = \arg\min_{\vec{z} \in Q} \{ V_{\vec{z}_k}(\vec{z}) + (\alpha_{k+1} L) \langle \nabla f(\vec{x}_{k+1}), \vec{z} - \vec{z}_k \rangle \}$
end for
return $\vec{y}_T$.

4.3.6 RMS Prop

Root mean squared propagation was first proposed by Geoff Hinton and is similar to Adagard. RMS Prop keeps track of a moving average of the squared gradients for each weight. The learning rate is then divided by this moving average. RMS prop takes advantage of the mini-batch idea from SGD and using previous gradients from Adagrad. The method of RMS prop is outlined below with C as the cost function, β as the moving average parameter, η as the learning rate, and $E[\vec{g}^2]_t$ as the moving average of the squared gradients:

$$E[\vec{g}^2]_t = \beta E[\vec{g}^2]_{t-1} + (1-\beta) \left(\frac{\partial C}{\partial \vec{w}} \right)^2 \tag{9}$$

$$\vec{w}_t = \vec{w}_{t-1} - \frac{\eta}{\sqrt{E[\vec{g}^2]_t}} \frac{\partial C}{\partial \vec{w}} \tag{10}$$

4.3.7 Adam

Adam is a common optimization algorithm that keeps an average of the decaying standard and squared gradients and uses it to normalize the learning rate and speed up gradient descent. We use two hyperparameters to control the influence of each the standard and squared past gradients: β_1 and β_2, as shown below.

$$\vec{m}_k = \beta_1 \vec{m}_{k-1} + (1-\beta_1)\vec{g}_k \tag{11}$$

$$\vec{v}_k = \beta_2 \vec{v}_{k-1} + (1-\beta_2)\vec{g}_k^2 \tag{12}$$

$$\hat{\vec{m}}_k = \frac{\vec{m}_k}{1-\beta_1^k} \tag{13}$$

$$\hat{\vec{v}}_k = \frac{\vec{v}_k}{1-\beta_2^k} \tag{14}$$

$$\vec{x}_{k+1} = \vec{x}_k - \frac{\eta}{\sqrt{\hat{\vec{v}}_k + \epsilon}} \hat{\vec{m}}_k \tag{15}$$

4.4 Results

Below we provided graphs evaluating the performance of each gradient descent method at each epoch of training. We additionally compare the performance of each model on testing and training data for each epoch.

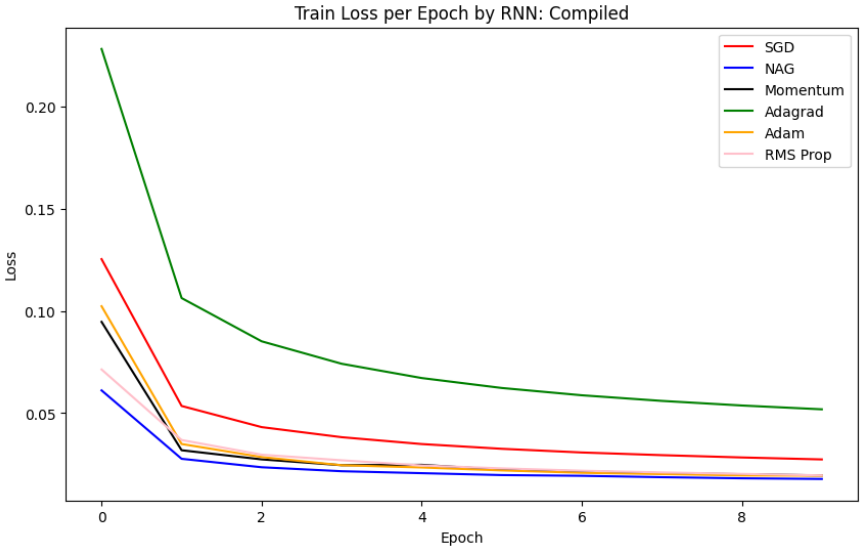

Figure 1: Training loss per epoch comparison of the gradient descent algorithms over 10 epochs

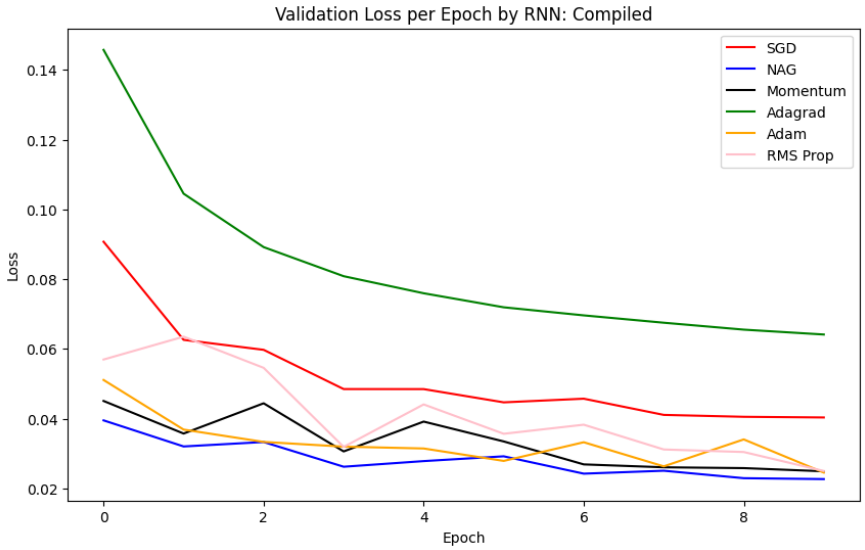

Figure 2: Validation loss per epoch comparison of the gradient descent algorithms over 10 epochs

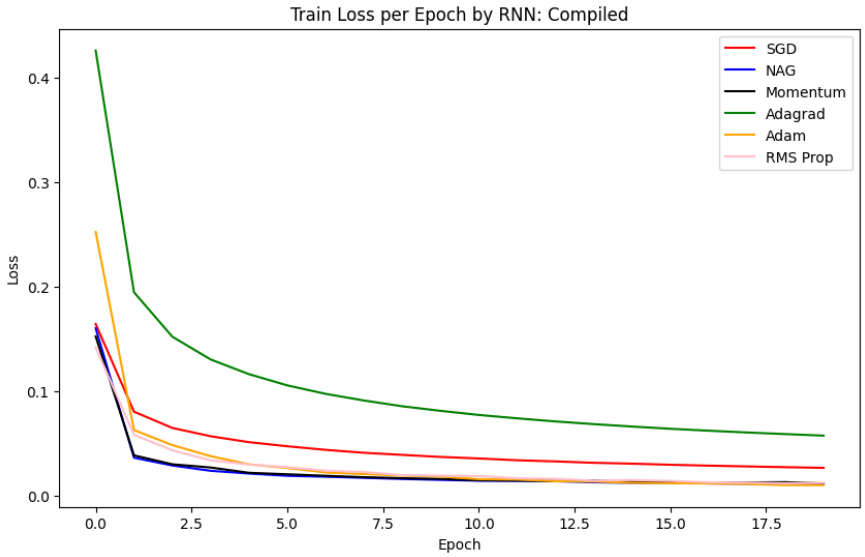

Figure 3: Training loss per epoch comparison of the gradient descent algorithms over 20 epochs

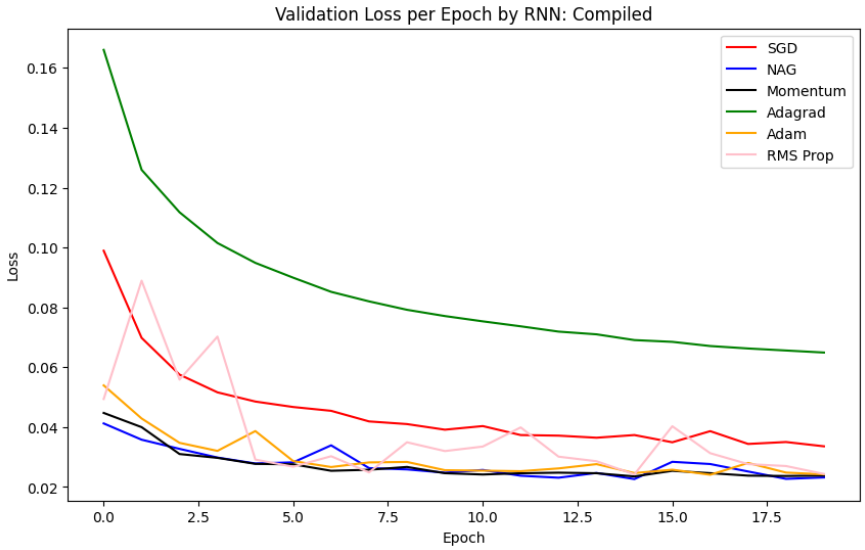

Figure 4: Validation loss per epoch comparison of the gradient descent algorithms over 20 epochs

Out of the six models examined, Adagrad performed the worst in both testing and training, finding a significantly less optimal solution compared to every other method. NAG has the best performance, often reaching the most optimal solution, and doing so faster than the other methods. In the initial epochs of training, Adagrad seems to start with the most loss even though every model starts with the same exact initial state. NAG starts at the lowest loss, and it continues to have low loss throughout. Training loss is consistently decreasing, but validation loss seems to fluctuate more, likely due to the algorithms finding local minima. In addition, moving down the gradient regarding the train data may be very different than that of the validation data.

4.5 Discussion

The convergence rates of the algorithms behaved as expected, with NAG converging the fastest. This could be caused by NAG's systematically faster convergence time. However, what is not taken into account is if more training epochs would lead to another algorithim finding a more optimal solution than NAG. What could further be expanded upon is training on more data sets to avoid one data set unfairly advantaging one descent method. A very close second place was Adam, which is one of the industry favorites at the moment. Both of these algorithms have a common strategy, which is coupling other algorithms together. Nesterov's Gradient Descent can be taken as a combination of gradient descent. Adam aggregates more gradient data in a coupling fashion. Both of these algorithms were found to be the most robust and accurate after the alotted epochs.

Previous research suggests that SGD should eclipse Adam and NAG in terms of overall model performance, despite Adam and NAG having much better training speed [10]. Zhou argues that SGD is able to escape local minima more easily, allowing it to outperform the faster methods. However, we did not necessarily find this to be the case in our further research and testing. There are multiple possible reasons for this. First, we may not have had a large enough number of epochs in order for SGD to converge to a degree that surpassed the accelerated methods. Additionally, the loss landscape that resulted from training this dataset may also not contain deep local minima, which allows accelerated gradient descent methods to move in the correct direction more consistently. Overall, this means SGD isn't moving as consistently in the right direction, and accelerated methods may have the advantage in lower epoch counts such as this one.

5 Contributions

Rohan, Timmy and Sam read through and discusses each of the papers and then did half of the problems independently and half of them together. Sam wrote 1b, 1d, 1f, 1h, and 2c. Timmy wrote problems 1a, 1e, 1g, 2a, and 2b. Rohan did problems 1c, 2d, 1i, and 2e. Rohan presented the initially paper for the extensions which Sam combined with another paper to come up with the final idea. Timmy wrote the code for the extensions while Sam and Rohan wrote about the various gradient descent methods. Sam and Rohan wrote the abstract and introduction. Timmy proof read the entire document.

5.1 Works Cited

1. Y. E. Nesterov, "A method of solving a convex programming problem with convergence rate o\bigl(k2\bigr)," in *Doklady Akademii Nauk*, Russian Academy of Sciences, vol. 269, 1983, pp. 543–547.

2. Y. Nesterov et al., *Lectures on Convex Optimization*. Springer, 2018, vol. 137.

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

4. Cornell University, "Momentum - Optimization for Machine Learning," [Online]. Available: https://optimization.cbe.co [Accessed 2024].

5. Towards Data Science, "Gradient Descent with Momentum," [Online]. Available: https://towardsdatascience.com/gradi descent-with-momentum-59420f626c8f. [Accessed 2024].

6. Towards Data Science, "Understanding RMSprop - Faster Neural Network Learning," [Online]. Available: https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a. [Accessed 2024].

7. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

8. D. Sutskever et al., "On the importance of initialization and momentum in deep learning," arXiv preprint arXiv:2010.0562, 2020.

9. C. Jin, P. Netrapalli, and M. I. Jordan, "Accelerated Gradient Descent Escapes Saddle Points Faster than Gradient Descent," in *Proceedings of the 31st Conference On Learning Theory*, vol. 75, pp. 1042–1085, PMLR, July 2018. Link to article.

10. P. Zhou, J. Feng, C. Ma, C. Xiong, S. Hoi, and W. E., "Towards Theoretically Understanding Why SGD Generalizes Better Than ADAM in Deep Learning," arXiv preprint arXiv:2010.05627, 2020. Link to article.

11. M. J., "Weather Dataset," Kaggle, 2020. Available Online.

12. I. Sutskever, "Training Recurrent Neural Networks," PhD Thesis, University of Toronto, 2013. Available Online.