Runge Kutta Optimizers

Raghav Singhal

December 4, 2018

Abstract

We investigate a link between the most commonly used optimization scheme, Gradient-Descent, and a well known equation, the Gradient Flow. Using this analogy we establish a link between Optimization and Numerical Integration of the Gradient Flow Equation, and we then use this analogy to investigate Runge-Kutta Methods as optimization schemes.

1 Introduction

Gradient Descent or Steepest Descent Flows, is a well studied topic in Partial Differential Equations and Differential Geometry. Given a convex functional f on a space X, suppose we wish to minimize f, then one way to find points x^* such that $\nabla f(x^*) = 0$. Now, finding such points in directly is not feasible, hence we look for the shortest possible path from x_0 to x^*, which is provided by the equation $\dot{x}(t) = -\nabla f(x(t))$, with $x(0) = x_0$. This is fairly simple to work out in a finite-dimensional Hilbert Space but can also be implemented in Infinite-dimensional Space. For instance, the heat equation can be seen as gradient flow in the Hilbert Space $L_2(\mathbb{R}^n)$, $u_t = \frac{\partial}{\partial u} f(u)$,

where $f(u) = \frac{1}{2}\int |\nabla u|^2$.

Now, Gradient Flow is widely used as a continuous approximation of the Gradient Descent Method. The analogy can be made formal by noticing that Gradient Descent is Euler's Method to numerically solve Gradient Descent.

These two fields have different goals, in Optimization the goal is to seek the minimizer of a function or in other words, it focuses on the infinite time horizon. In numerical analysis, the focus is on finite-time horizon, $[0, T_{\max}]$ and to maintain consistency and not accumulate too much error.

This connection was also established by Scieur et al in [2], where they show that accelerated methods are instances of multi-step methods which integrate the Gradient Flow Equation. The idea of using differential equations to model optimization methods has gained prominence lately, recently [1] used a second-order differential equation to model Nesterov's Accelerated Gradient Descent. This approach however works backwards, that is it uses the optimization scheme to obtain a second order differential equation in the limit, [2] however analyze Nesterov's Accelerated Gradient Descent as a linear multi-step scheme.

Gradient Flow is the bridge between numerical integration and optimization, and in [2] they used concepts from numerical integration to show convergence and other properties of optimization schemes. They particularly use linear multi-step methods but in this work, we use intermediate step methods to integrate the gradient flow equation. The most famous family of intermediate step methods, or half-step methods as they are commonly known, is the Runge-Kutta Family. Here we

show the performace of half-step methods as optimization schemes.

2 Gradient Flow

Suppose f is a β-smooth and α-Strongly Convex function. Then gradient flow for f is defined as

$$\dot{x}(t) = -\nabla f(x(t)), \qquad x(0) = x_0 \tag{1}$$

Now, as f is β-smooth, we can guarantee that a solution exists and is unique, and as f is α-Strongly Convex, we can establish that all solutions converge to the same equilibrium point, x^*, this equilibrium point is the unique global minimum of f.

Now, we list the two propositions that motivated our study of numerical integration schemes for the gradient flow equation.

Proposition 1. *Let $x_1(t), x_2(t)$ be two solution trajectories of the gradient flow equation* (1)*, then the following holds:*

$$\|x_1(t) - x_2(t)\|^2 \leq e^{-2\mu t} \|x_1(0) - x_2(0)\|^2$$

Proof. Let $\mathcal{L}(t) = \|x_1(t) - x_2(t)\|^2$, then note that

$$\begin{aligned}
\frac{d}{dt}\mathcal{L}(t) &= 2\langle x_1(t) - x_2(t), \dot{x_1}(t) - \dot{x_2}(t)\rangle \\
&= 2\langle x_1(t) - x_2(t), -\nabla f(x_1(t)) + \nabla f(x_2(t))\rangle \\
&\leq -2\mu\mathcal{L}(t)
\end{aligned}$$

which implies $\mathcal{L}(t) \leq e^{-2\mu t}\mathcal{L}(0) = e^{-2\mu t} \|x_1(0) - x_2(0)\|^2$. □

This proposition states that any solution trajectory of the gradient flow equation (1) will converge to the same point over time, and we show that this point is the global minimum of f.

Proposition 2. *Let f be β-smooth and α-Strongly Convex and x^* is the global minimum of f, then the solution trajectory of gradient flow* (1) *satisfies:*

$$f(x(t)) - f(x^*) \leq e^{-2\mu t}\big(f(x(0)) - f(x^*)\big)$$

Proof. Note that

$$\begin{aligned}\frac{d}{dt}\big(f(x(t)) - f(x^*)\big) &= \langle \nabla f(x(t)), \dot{x(t)} \rangle \\ &= -\left\|\nabla f(x(t))\right\|^2\end{aligned}$$

Now, as f is strongly convex,

$$f(x) - f(x^*) \leq \frac{1}{2\mu}\left\|\nabla f(x)\right\|^2$$

which then implies

$$\frac{d}{dt}\big(f(x(t)) - f(x^*)\big) = -2\mu\big(f(x(t)) - f(x^*)\big)$$

Hence, $f(x(t)) - f(x^*) \leq e^{-2\mu t}\big(f(x(0)) - f(x^*)\big)$. □

These two propositions motivated us to consider numerical integration techniques for the gradient flow equation.

3 Integration Schemes

Schur et al [2] show that acceleration techniques like Nesterov's accelerated gradient descent can be seen as particular instances of multi-step methods, popular techniques for numerical integration.

Suppose we wish to solve (1) over the interval $[0, T]$, then we discretize the solution trajectory as $\{x_1, \dots x_K\} \sim \{x(t_1), \dots, x(t_k)\}$, where $t_0 = 0$ and $t_k = t_{k-1} + \eta$ and η is the step size. This step size is called the learning rate in the optimization community, and we choose it such that $x_k \sim x(t_k)$. Now, linear multi-step schemes are defined as those methods which use the previous points and the derivative values at the previous point to construct a new solution, more precisely a linear s-step method is defined as follows:

$$x_{k+s} = \sum_{i=0}^{s-1} a_i x_{k+i} + \eta \sum_{i=0}^{s} b_i g(x_{k+i})$$

where we are solving the equation $\dot{x}(t) = g(x(t))$. If $b_s \neq 0$ then the above method becomes an implicit method and if $b_s = 0$, then it is known as an explicit method. A lot of current optimization schemes and acceleration schemes rely on this technique. These techniques were invented in order to tackle several issues like stiffness, stability, error control, etc.

The focus of this work is analyzing the Runge-Kutta family, which is a family of intermediate-step methods, that is to get to point x_{k+1} from x_k, they use points which are not part of the solution trajectory we finally obtain. More formally, an s-step Runge Method is defined as

$$x_{k+1} = x_k + \eta \sum_{i=1}^{s} c_i k_i, \quad \text{where}$$

$$k_1 = g(x_k)$$

$$k_2 = g(x_k + \eta a_{2,1} k_1)$$

$$k_s = g(x_k + \eta \sum_{i=1}^{s} a_{s,i} k_i)$$

Now, Runge-Kutta methods have to satisfy $\sum_{j=1}^{i-1} a_{i,j} = c_i$ for all $i = 2, \dots, s$. These requirements are imposed so that the method has an error of order p, that is the local truncation error is $\mathcal{O}(\eta^{p+1})$

and a global error of $\mathcal{O}(\eta^p)$. The local truncation error, $\epsilon(x_k)$, is defined as

$$\epsilon(x_k) = x_k - x(t_k)$$

assuming the previous solutions $x_1, \ldots, x_{k-1}$ were exact so that $x_{k-1} = x(t_{k-1})$.

The main motivation behind explicit Runge-Kutta methods is to use a quadrature scheme for the following problem:

$$\begin{aligned} x(t_{k+1}) &= x(t_k) + \int_{t_k}^{t_{k+1}} g(x(s))ds \\ &= x(t_k) + \eta \int_0^1 g(x(t_k + \eta s)ds, \qquad \text{which can be approximated as} \\ x_{k+1} &= x_k + \eta \sum_{i=1}^{s} b_i g(x_k + \eta c_i) \end{aligned}$$

Now, as we don't have access to the intermediate points $x(t_k + \eta c_i)$, we approximate by using $x(t_k) + \eta c_i k_i$, where k_i is defined above.

Below, we analyze two popular 2nd-order Runge-Kutta Method, RK2 Heun's Method, and Ralston's Method. But first we show a preliminary analysis on a quadratic problem

Quadratic

The reason we focus on Runge-Kutta methods is that they perform better when dealing with stiff equations. For instance, let $f : \mathbb{R}^d \to \mathbb{R}$ be defined as

$$f(x) = \frac{1}{2} x^T H x$$

where H is positive definite, then note that $\nabla f(x) = Hx$, then gradient descent would yield

$$\begin{aligned} x_{k+1} &= x_k - \eta H x_k = (I - \eta H) x_k \\ &= (I - \eta H)^k x_0 \\ &= U(I - \eta \Lambda)^k U^T x_0 \end{aligned}$$

and RK4 would yield

$$\begin{aligned} x_{k+1} &= x_k - \frac{\eta}{6}\Big(k_1 + 2k_2 + 2k_3 + k_4\Big) \\ &= x_k - \frac{\eta}{6}\Big(Hx_k + 2H(x_k - \frac{\eta}{2}Hx_k) \\ &\qquad + 2H(x_k - \frac{\eta}{2}H(x_k - \frac{\eta}{2}Hx_k)) + H(x_k - \eta H(x_k - \frac{\eta}{2}H(x_k - \frac{\eta}{2}Hx_k)))\Big) \\ &= x_k - \frac{\eta}{6}\Big(6Hx_k - 3\eta H^2 x_k + \frac{3\eta^2}{4}H^3 x_k - \frac{\eta^3}{4}H^4 x_k\Big) \\ &= \Big(I - \frac{\eta}{6}\big(6H - 3\eta H^2 + \frac{3\eta^2}{4}H^3 - \frac{\eta^3}{4}H^4\big)\Big)x_k \\ &= U\Big(I - \eta\Lambda + \frac{\eta^2}{2}\Lambda^2 - \frac{3\eta^3}{8}\Lambda^3 + \frac{\eta^4}{24}\Lambda^4\Big)U^T x_k \end{aligned}$$

Now, note that when $\eta = \frac{1}{\Lambda_{max}}$ then RK4 converges faster than Gradient Descent, but we also note that RK4 is also a corrective method and integrates much more accurately than Euler's Method.

In this work, however we only focus on 2-nd order methods due to computational issues and in consideration of time.

4 RK2 Ralston

RK2 is a popular 2-nd order method. The scheme for solving $\dot{x}(t) = g(x(t))$ is defined as follows

$$k_1 = g(x_k)$$
$$k_2 = g(x_k + \eta\frac{3}{4}k_1)$$
$$x_{k+1} = x_k + \frac{\eta}{4}\big(k_1 + 3k_2\big)$$

Now, if we have optimize a function, f, then the RK2-Ralston Optimization Scheme is defined as follows:

$$x_{k+1} = x_k - \frac{\eta}{4}(\nabla f(x) + 3\nabla f(x - \frac{2}{3}\eta\nabla f(x)))$$

4.1 Strongly Convex

Theorem 1. *Let f be β-smooth and α-Strongly Convex function. Then let $\eta = \frac{2}{\alpha+\beta}$. Then RK-2 Ralston satisfies:*

$$f(x_k) - f(x^*) \leq \frac{\beta}{2}\exp(-\frac{4k}{\kappa+1})\,\|x_1 - x^*\|_2^2$$

where $\kappa = \frac{\beta}{\alpha}$ is the condition number.

Proof. Define $r_k = \|x_k - x^*\|$ and $F(x) = \nabla f(x) - 3\nabla f(x - \frac{2}{3}\eta \nabla f(x)) = k_1 + 3k_2$,

$$
\begin{aligned}
\|x_{k+1} - x^*\|^2 &= \left\| x_k - x^* - \frac{\eta}{4} F(x_k) \right\|^2 \\
&= \|x_k - x^*\|^2 + \frac{\eta^2}{16} \|F(x_k)\|^2 - \frac{\eta}{2}(x_k - x^*)^T F(x_k) \\
&= \|x_k - x^*\|^2 + \frac{\eta^2}{16} \|k_1 + 3k_2\|^2 - \frac{\eta}{2}(x_k - x^*)^T F(x_k) \\
&= \|x_k - x^*\|^2 + \frac{\eta^2}{16}(\|k_1\|^2 + 9\|k_2\|^2 + 6k_1^T k_2) - \frac{\eta}{2}(x_k - x^*)^T k_1 - \frac{3\eta}{2}(x_k - x^*)^T k_2 \\
&= \|x_k - x^*\|^2 + \frac{\eta^2}{16}(\|k_1\|^2 + 9\|k_2\|^2 + 6k_1^T k_2) - \frac{\eta}{2}(x_k - x^*)^T k_1 \\
&\qquad - \frac{3\eta}{2}(x_k - \frac{2\eta}{3} k_1 - x^*)^T k_2 - \eta^2 k_1^T k_2 \\
&\leq \|x_k - x^*\|^2 + \frac{\eta^2}{16}(\|k_1\|^2 + 9\|k_2\|^2) - \frac{\eta}{2}(x_k - x^*)^T k_1 - \frac{3\eta}{2}(x_k - \frac{2\eta}{3} k_1 - x^*)^T k_2
\end{aligned}
$$

Then again by equation (14), then the above equation becomes,

$$
\begin{aligned}
r_{k+1}^2 &\leq r_k^2 + + \frac{\eta^2}{16}(\|k_1\|^2 + 9\|k_2\|^2) - \frac{\eta}{2}(x_k - x^*)^T k_1 - \frac{3\eta}{2}(x_k - \frac{2\eta}{3} k_1 - x^*)^T k_2 \\
&\leq r_k^2 + \frac{\eta}{4}(\eta - \frac{2}{\alpha + \beta}) \left(\|k_1\|^2 + 3\|k_2\|^2 \right) - \frac{\eta}{2} \frac{\alpha\beta}{\alpha + \beta} \left(r_k^2 + 3 \left\| x_k - \frac{2\eta}{3} k_1 - x^* \right\|^2 \right) \\
&= r_k^2 - \frac{\eta}{2} \frac{\alpha\beta}{\alpha + \beta} \left(r_k^2 + 3 \left\| x_k - \frac{2\eta}{3} k_1 - x^* \right\|^2 \right) \\
&= r_k^2 - \frac{\eta}{2} \frac{\alpha\beta}{\alpha + \beta} \left(r_k^2 + 3r_k^2 + \frac{4\eta^2}{3} \|\nabla f(x_k)\|^2 - 4\eta(x_k - x^*)^T \nabla f(x_k) \right) \\
&= \left(1 - \frac{2\eta\alpha\beta}{\alpha + \beta} \right) r_k^2 - \frac{\eta}{2} \frac{\alpha\beta}{\alpha + \beta} \left(\frac{4\eta^2}{3} \|\nabla f(x_k)\|^2 - 4\eta(x_k - x^*)^T \nabla f(x_k) \right) \\
&\leq \left(\frac{\kappa - 1}{\kappa + 1} \right)^2 r_k^2
\end{aligned}
$$

□

4.2 Convex

Theorem 2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and β-smooth, then RK2-Ralston with $\eta = \frac{2}{\beta}$ satisfies the following:*

$$f(x_k) - f(x^*) \leq \frac{4}{3}\frac{\|x_1 - x^*\|^2}{k-1} \tag{2}$$

Lemma 1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and β-smooth, then RK2-Ralston satisfies the following:*

$$f(x_{k+1}) \leq f(x_k) - \frac{3}{4\beta(k+1)}\|\nabla f(x_k)\|^2$$

Proof. (Lemma 1) Let $\Delta x = \frac{1}{2\beta}(k_1 + 3k_2)$, where $\eta = \frac{2}{\beta}$ then note that

$$\begin{aligned}
f(x - \Delta x) - f(x) &\leq \nabla f(x)^T(x - \Delta x - x) + \frac{\beta}{2}||x - x - \Delta x||^2 \\
&= -\nabla f(x)^T(\Delta x) + \frac{1}{16\beta}||\Delta x||^2 \\
&= -\frac{1}{2\beta}\nabla f(x)^T(k_1 + 3k_2) + \frac{1}{16\beta}||\Delta x||^2 \\
&= -\frac{1}{2\beta}\nabla f(x)^T k_1 - \frac{3}{2\beta}\nabla f(x)^T k_2 + \frac{1}{16\beta}||k_1||_2^2 + \frac{9}{16\beta}||k_2||_2^2 + \frac{6}{16\beta}\langle k_1, k_2\rangle \\
&= -\frac{1}{2\beta}\nabla f(x)^T k_1 + \frac{1}{16\beta}||k_1||_2^2 - \frac{3}{2\beta}\nabla f(x)^T k_2 + \frac{9}{16\beta}||k_2||_2^2 + \frac{6}{16\beta}k_1^T k_2 \\
&= -\frac{7}{16\beta}||k_1||_2^2 - \frac{1}{16\beta}k_2^T(24\nabla f(x) - 9k_2) + \frac{6}{16\beta}k_1^T k_2 \\
&= -\frac{7}{16\beta}||k_1||_2^2 - \frac{24}{16\beta}k_2^T k_1 + \frac{6}{16\beta}k_1^T k_2 + \frac{9}{16\beta}||k_2||_2^2 \\
&= -\frac{7}{16\beta}||k_1||_2^2 - \frac{18}{16\beta}k_1^T k_2 + \frac{9}{16\beta}||k_2||_2^2
\end{aligned} \tag{3}$$

Now, using a Taylor Series approximation for $\nabla f\big(x - \frac{2\eta}{3}k_1\big)$, we get that,

$$\begin{aligned}
\nabla f\big(x - \frac{2\eta}{3}k_1\big) &= \nabla f(x) - \frac{2\eta}{3}\nabla^2 f(x)\nabla f(x) + \mathcal{O}(|\frac{2\eta}{3}|^2) \\
\implies k_2^T k_1 &= \nabla f\big(x - \frac{2\eta}{3}k_1\big)^T \nabla f(x) \\
&= \nabla f\big(x - \frac{2\eta}{3}\nabla f(x)\big)^T \nabla f(x) \\
&= ||\nabla f(x)||_2^2 - \frac{2\eta}{3}\nabla f(x)^T \nabla^2 f(x)\nabla f(x)
\end{aligned} \tag{4}$$

And, using (4),

$$\begin{aligned}
||k_2||_2^2 &= ||\nabla f(x) - \frac{2\eta}{3}\nabla^2 f(x)\nabla f(x)||_2^2 \\
&= ||\nabla f(x)||_2^2 + \frac{4\eta}{9}||\nabla^2 f(x)\nabla f(x)||_2^2 - \frac{4\eta}{3}\nabla f(x)^T \nabla^2 f(x)\nabla f(x)
\end{aligned} \tag{5}$$

Hence, using (4) and (5)

$$\begin{aligned}
f(x - \Delta x) - f(x) &\leq -\frac{7}{16\beta}||k_1||_2^2 - \frac{18}{16\beta}k_1^T k_2 + \frac{9}{16\beta}||k_2||_2^2 \\
&= -\frac{7}{16\beta}||\nabla f(x)||_2^2 - \frac{18}{16\beta}||\nabla f(x)||_2^2 + \frac{12}{16\beta^2}\nabla f(x)^T \nabla^2 f(x)\nabla f(x) \\
&+ \frac{9}{16\beta}(||\nabla f(x)||_2^2 + \frac{4}{9\beta}||\nabla^2 f(x)\nabla f(x)||_2^2 - \frac{4}{3\beta}\nabla f(x)^T \nabla^2 f(x)\nabla f(x)) \\
&= -\frac{16}{16\beta}||\nabla f(x)||_2^2 + \frac{1}{4\beta^2}||\nabla^2 f(x)\nabla f(x)||_2^2 \\
&= -\frac{1}{\beta}||\nabla f(x)||_2^2 + \frac{1}{4\beta^2}||\nabla^2 f(x)\nabla f(x)||_2^2
\end{aligned}$$

Using the lipschitz property of the Hessian of f, $||\nabla^2 f(x)u - \nabla^2 f(x)v||_2^2 \leq \beta||u - v||_2^2$, we get that,

$$\begin{aligned}
f(x - \Delta x) - f(x) &\leq -\frac{1}{\beta}||\nabla f(x)||_2^2 + \frac{1}{4\beta^2}||\nabla^2 f(x)\nabla f(x)||_2^2 \\
&\leq -\frac{4}{4\beta}||\nabla f(x)||_2^2 + \frac{\beta}{4\beta^2}||\nabla f(x)||_2^2 \\
&= -\big(\frac{4}{4\beta} - \frac{\beta}{8\beta^2}\big)||\nabla f(x)||_2^2 \\
&= -\frac{3}{4\beta}||\nabla f(x)||_2^2
\end{aligned} \tag{6}$$

□

Proof. (Theorem 2) Using Lemma 1, we have $f(x_{t+1}) - f(x_t) \leq -\frac{3}{4\beta}||\nabla f(x_t)||_2^2$. Now, let $\delta_t = f(x_t) - f(x^*)$, then note that:

$$\delta_{t+1} \leq \delta_t - \frac{3}{4\beta}||\nabla f(x)||_2^2$$

Now, by convexity of $f(x)$ we have:

$$\delta_t \leq \nabla f(x_t)^T(x_t - x^*) \tag{7}$$

$$\leq ||x_t - x^*||_2 * ||\nabla f(x_t)||_2 \tag{8}$$

$$\frac{1}{||x_t - x^*||}\delta_t^2 \leq ||\nabla f(x_t)||_2^2 \tag{9}$$

Now, note that $||x_t - x^*||_2^2$ is decreasing, using the following inequality

$$\big(\nabla f(x) - \nabla f(y)\big)^T(x - y) \geq \frac{1}{\beta}||\nabla f(x) - \nabla f(y)||_2^2$$

Using the above and the fact that $\nabla f(x^*) = 0$,

$$
\begin{aligned}
||x_{t+1} - x^*||_2^2 &= ||x_t - \Delta x_t - x^*||_2^2 \\
&= ||x_t - x^*||_2^2 + ||\Delta x_t||_2^2 - 2\Delta x_t^T (x_t - x^*) \\
&= ||x_t - x^*||_2^2 - \frac{1}{\beta}(k_1 + 3k_2)^T (x_t - x^*) + \frac{1}{4\beta^2}||k_1 + 3k_2||_2^2 \\
&= ||x_t - x^*||_2^2 - \frac{1}{\beta}k_1^T (x_t - x^*) + \frac{1}{4\beta^2}||k_1||_2^2 \\
&\qquad - \frac{3}{\beta}k_2^T (x_t - x^*) + \frac{9}{4\beta^2}||k_2||_2^2 + \frac{6}{4\beta^2}k_1^T k_2 \\
&= ||x_t - x^*||_2^2 - \frac{4}{4\beta^2}||k_1||_2^2 + \frac{1}{4\beta^2}||k_1||_2^2 \\
&\qquad - \frac{12}{4\beta^2}|k_2||_2^2 + \frac{9}{4\beta^2}||k_2||_2^2 + \frac{6}{4\beta^2}k_1^T k_2 \\
&= ||x_t - x^*||_2^2 - \frac{3}{4\beta^2}||k_1||_2^2 - \frac{3}{4\beta^2}||k_2||_2^2 + \frac{6}{4\beta^2}k_1^T k_2 \\
&= ||x_t - x^*||_2^2 - \frac{3}{4\beta^2}||k_1 - k_2||_2^2 \\
&\leq ||x_t - x^*||_2^2
\end{aligned}
$$

We will show that,

$$
\delta_{t+1} \leq \delta_t - \frac{3}{4\beta||x_1 - x^*||_2^2}\delta_t^2 \tag{10}
$$

Now, let $\omega = \frac{3}{4\beta||x_1 - x^*||_2^2}$,

$$
\begin{aligned}
&\frac{1}{\delta_t} \geq \omega(t-1) \\
\implies &f(x_t) - f(x^*) \leq \frac{4}{3\beta}\frac{||x_1 - x^*||_2^2}{t-1} \xrightarrow{t\to\infty} 0
\end{aligned}
$$

□

5 RK2 Heun

RK2 Heun is commonly known as a predictor-corrector algorithm. It is based on Euler's Method, where like Euler's Method it uses the tangent to the function at some point to obtain the next point, however even when the step-size or the learning rate is small, the error starts to accumulate over time, however the second step is meant to act as a corrector step. The scheme is defined as follows for the equation $\dot{x}(t) = g(x(t))$

$$k_1 = g(x_k)$$

$$k_2 = g(x_k + \eta k1)$$

$$x_{k+1} = x_k + \frac{\eta}{2}\big(k_1 + k_2\big)$$

In other words if we wish to minimize the function f, then RK2 heun's method scheme is defined as follows:

$$x_{k+1} = x_k - \frac{\eta}{2}(\nabla f(x_k) - \nabla f(x_k - \eta \nabla f(x_k)))$$

5.1 Strongly Convex

Theorem 3. *Let f be β-smooth and α-Strongly Convex function. Then let $\eta = \frac{4}{\alpha+\beta}$. Then RK-2 Heun's method satisfies:*

$$f(x_k) - f(x^*) \leq \frac{\beta}{2} \exp\left(- \frac{8k}{\kappa + 1} \right) \|x_1 - x^*\|^2$$

where $\kappa = \frac{\beta}{\alpha}$ is the condition number.

Proof. Define $r_k = \|x_k - x^*\|$ and $F(x) = \nabla f(x) + \nabla f(x - \eta \nabla f(x)) = k_1 + k_2$, then note that

$$\|x_{k+1} - x^*\|^2 = \left\|x_k - x^* - \frac{\eta}{2} F(x_k)\right\|^2 \tag{11}$$

$$= \|x_k - x^*\|^2 + \frac{\eta^2}{4} \|F(x_k)\|^2 - \eta (x_k - x^*)^T F(x_k) \tag{12}$$

$$\leq \|x_k - x^*\|^2 + \frac{\eta^2}{4} (\|k_1\|^2 + \|k_2\|^2 + 4 k_1^T k_2) - \eta (x_k - x^*)^T F(x_k) \tag{13}$$

Now, note that for $f \in \mathcal{S}_{\beta,\alpha}(\mathbb{R}^n)$, that is f is β-smooth and α-Strongly Convex function,

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq \frac{\alpha\beta}{\alpha + \beta} \|x - y\|^2 + \frac{1}{\alpha + \beta} \|\nabla f(x) - \nabla f(y)\|^2 \tag{14}$$

So using (14), the inequality (13) reduces to

$$r_{k+1}^2 \leq r_k^2 + \frac{\eta^2}{4} (\|k_1\|^2 + \|k_2\|^2) + \eta^2 k_1^T k_2 - \eta (x_k - x^*)^T (k_1 + k_2) \tag{15}$$

$$= r_k^2 + \frac{\eta^2}{4} (\|k_1\|^2 + \|k_2\|^2) - \eta (x_k - x^*)^T k_1 - \eta (x_k - \eta k_1 - x^*)^T k_2 \tag{16}$$

Now, using (14) on the last two terms yields

$$\begin{aligned}
r_{k+1}^2 &\leq r_k^2 + \frac{\eta^2}{4} (\|k_1\|^2 + \|k_2\|^2) - \eta \frac{\alpha\beta}{\alpha + \beta} (r_k^2 + \|x_k - \eta k_1 - x^*\|^2) \\
&\qquad - \eta \frac{1}{\alpha + \beta} (\|k_1\|^2 + \|k_2\|^2) \text{ , using (14)} \\
&= r_k^2 + \frac{\eta}{4} (\eta - \frac{4}{\alpha + \beta}) (\|k_1\|^2 + \|k_2\|^2) - \eta \frac{\alpha\beta}{\alpha + \beta} (r_k^2 + ||x_k - x^* - \eta k_1||^2) \\
&= r_k^2 - \eta \frac{\alpha\beta}{\alpha + \beta} (r_k^2 + \|x_k - x^* - \eta k_1\|^2) \text{ , since } \eta = \frac{4}{\beta + \alpha} \\
&\leq r_k^2 - \eta \frac{\alpha\beta}{\alpha + \beta} (r_k^2 + \|x_k - x^* - \eta k_1\|^2) \\
&\leq r_k^2 (1 - \eta \frac{\alpha\beta}{\alpha + \beta}) - \eta \frac{\alpha\beta}{\alpha + \beta} (\|x_k - x^* - \eta k_1\|^2)
\end{aligned}$$

where the last two inequality follows as $\|x-y\|^2 \geq (\|x\| - \|y\|)^2$ and as $\|\nabla f(x) - \nabla f(y)\| \leq \beta \|x-y\|$

$$r_k^2 = r_k^2\left(1 - 4\frac{\alpha\beta}{(\alpha+\beta)^2}\right) - 4\frac{\alpha\beta}{(\alpha+\beta)^2}(\|x_k - x^* - \eta k_1\|^2) \tag{17}$$

$$\leq r_k^2\left(\frac{\beta-\alpha}{\beta+\alpha}\right)^2 - 4\frac{\alpha\beta}{(\alpha+\beta)^2}(\|x_k - x^* - \eta k_1\|^2) \tag{18}$$

$$= \left(\frac{\kappa-1}{\kappa+1}\right)^2 r_k^2 - 4\frac{\alpha\beta}{(\alpha+\beta)^2}(\|x_k - x^* - \eta k_1\|^2) \tag{19}$$

$$r_{k+1}^2 \leq \exp\left(-\frac{8k}{\kappa+1}\right)\|x_1 - x^*\|^2 \tag{20}$$

where $\kappa = \frac{\beta}{\alpha}$ is known as the condition number of f. □

5.2 Convex

Theorem 4. *Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be convex and β-smooth, then with $\eta = \frac{1}{\beta}$, RK2-Heun satisfies:*

$$f(x_k) - f(x^*) \leq \frac{2}{3\beta}\frac{\|x_1 - x^*\|^2}{k-1}$$

Lemma 2. *Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be convex and β-smooth, then with $\eta = \frac{1}{\beta}$, RK2-Heun satisfies:*

$$f(x_{k+1}) \leq f(x_k) - \frac{3}{2\beta}\|\nabla f(x_k)\|^2 \tag{21}$$

The Proof for Theorem 4 follows the same steps as the proof for Rk2 Ralston convex case.

6 Optimization in Deep Leanrning

References

[1] Su, Weijie, Stephen Boyd, and Emmanuel Candes. *A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights*. Advances in Neural Information

Processing Systems. 2014.

[2] Scieur, Damien, et al. *Integration Methods and Accelerated Optimization Algorithms* arXiv preprint arXiv:1702.06751 (2017).

[3] Nesterov, Yurii. *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science and Business Media, 2013.

7 Experiments

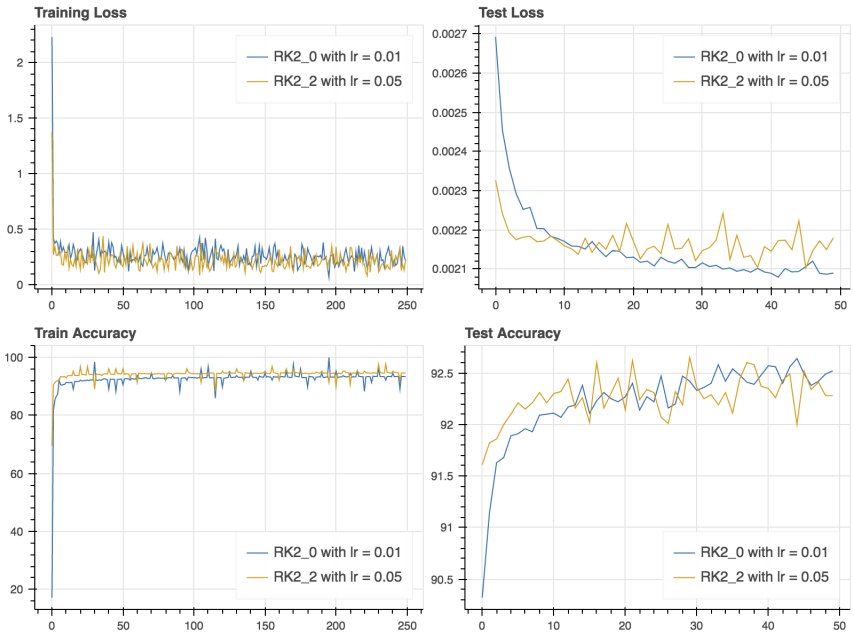
For the experimental section, we compared Stochastic RK2 methods with Stochastic Gradient Descent and its variants including SGD with momentum and SGD with nesterov's accelerated gradient descent. We conducted several experiments on different networks and with different datasets, across all network we noticed that RK2 methods decreased the training loss more and faster than any algorithm. However we saw that this did not always translate to better test performance, in other words the models trained on RK2 did not generalize as much as expected.

7.1 Convex Models

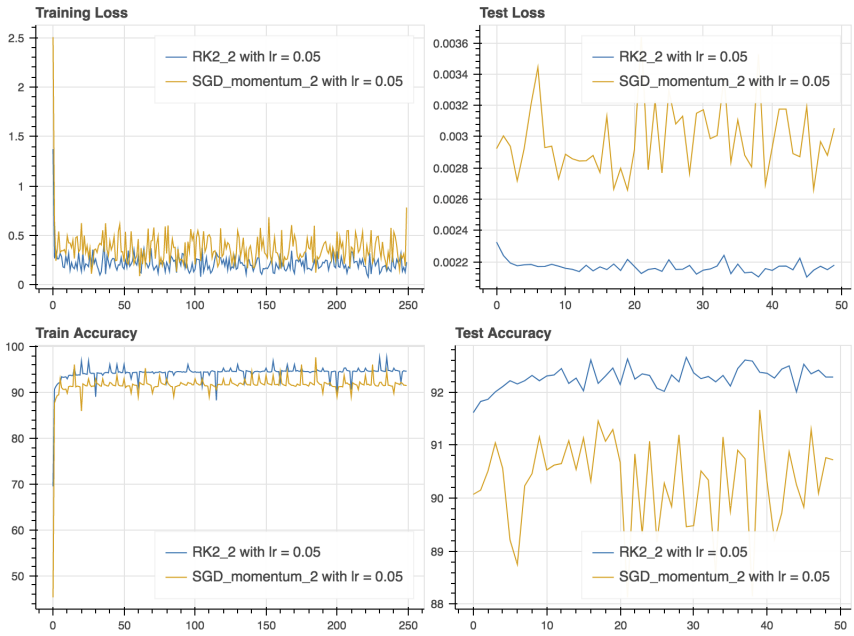
Here we do plain logistic regression and logistic regression with weight-decay on the mnist dataset. A big difference between the the previous experiments and these experiments is that the number of parameters in the model here are less than the number of training samples. So the chances of overfitting are much less. Here we present some plots for the experiments we did with these models.

Lasso

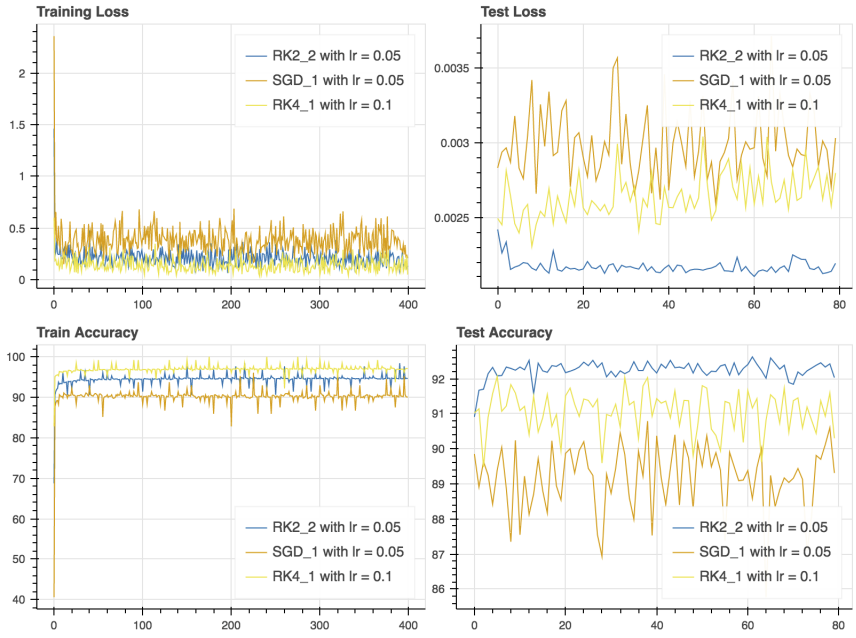
Here we show that our algorithm is not too sensitive to changes in the learning rate



Now, we show plots to compare our optimizer with simple stochastic gradient descent and its variants.



Below we show plots to compare our optimizer with Accelerated stochastic gradient descent.



7.2 Deep Learning

We experimented with the following setups

1. ResNet-18 on CIFAR-10 Dataset
2. ResNet-18 on Imagenet Dataset
3. WideResNet-16 on CIFAR-10
4. WideResNet-28 on CIFAR-10

The table below shows that RK2 methods consistently achieve the lowest training loss and highest training accuracy. That is a constant feature we see across different models. We make some observations in the next section

	best train loss	best train acc	best test loss	best test acc
resnet_cifar	RK2 0.00006	RK2_heun 100.00	RK2 0.06651	SGD_momentum 94.19
wideResnet_cifar_16	RK2 0.00302	RK2 99.92	SGD 0.24467	RK2 93.18
wideResnet_cifar	RK2 0.00157	RK2 99.96	SGD_nesterov 0.23740	SGD_nesterov 93.47
resnet_imagenet	RK2_heun 0.24764	SGD_nesterov 80.47	SGD_momentum 1.31651	SGD_nesterov 67.91

Figure 1: Overall Performance

ResNet-18 on CIFAR-10

Here we run the network with RK2 Ralston, RK2 Heun, SGD, SGD with momentum and SGD with nesterov's momentum. We ran the network with a wide variety of hyper-parameters and here we saw that lower train loss translated to a lower test loss.

RK2 Ralston and RK2 Heun consistently achieve a lower test loss than SGD and its variants. However, we notice that the disparity between training and test loss for RK2 is massive when compared to SGD.

	epochs done	best train loss	best train acc	best test loss	best test acc	optimizer	lr	epoch_step	wd
RK2_11	261/350	0.000938	100.0	0.066511	93.21	RK2	0.5	[60,90,120]	0.0005
RK2_3	350/350	0.000906	100.0	0.074918	92.83	RK2	0.5	[40,80]	0.0005
RK2_heun_5	350/350	0.000213	100.0	0.080567	92.68	RK2_heun	0.1	[30,60,90]	0.0001
RK2_0	350/350	0.000061	100.0	0.096568	92.62	RK2	0.1	[30,60,90,120]	0.0001
RK2_heun_4	273/350	0.002777	100.0	0.097050	93.45	RK2_heun	0.1	[30,60,90]	0.0010
RK2_heun_0	350/350	0.001521	100.0	0.102229	93.41	RK2_heun	0.1	[150, 250]	0.0005
RK2_9	350/350	0.000895	100.0	0.107186	93.64	RK2	0.5	[40,80]	0.0005
RK2_heun_2	350/350	0.000175	100.0	0.107224	92.93	RK2_heun	0.1	[50,150,250]	0.0001
RK2_13	282/350	0.001388	100.0	0.108330	92.89	RK2	0.1	[40,80,120]	0.0010
RK2_heun_3	350/350	0.002248	100.0	0.112350	93.85	RK2_heun	0.5	[100,200]	0.0005

Figure 2: CIFAR-10 Test Loss

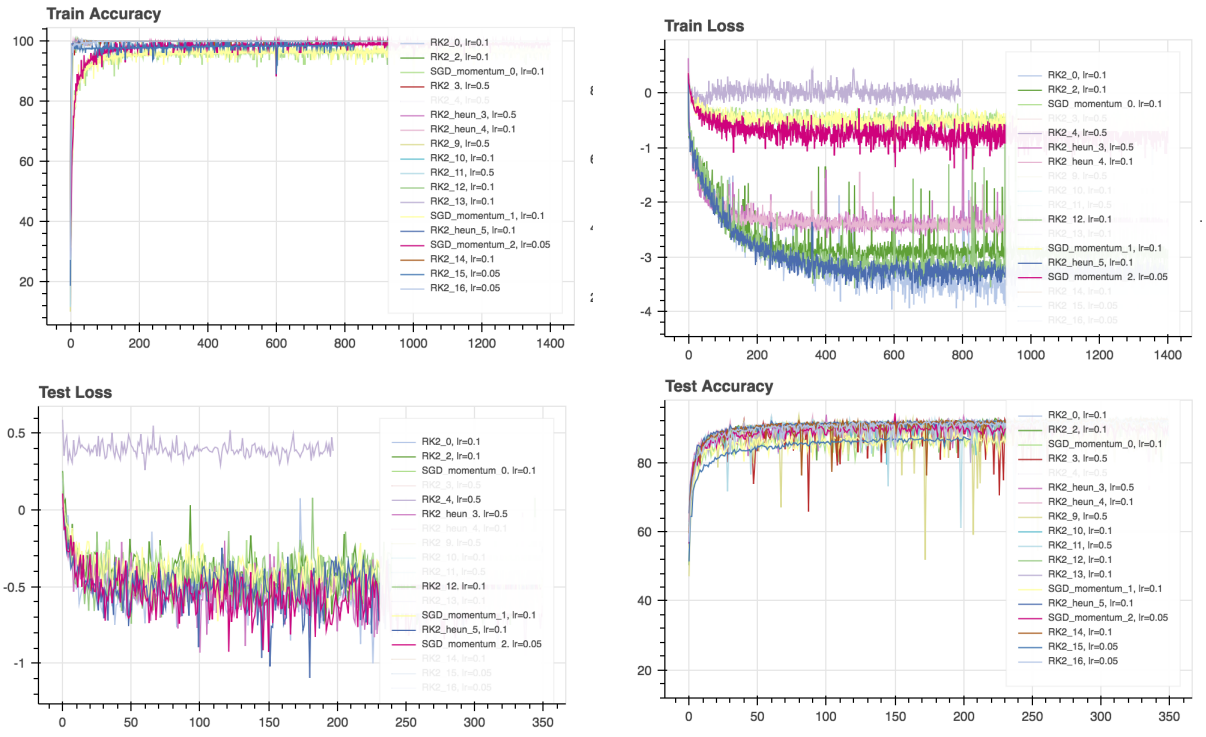


Figure 3: Log-Loss Plot & Accuracy : ResNet-18 on CIFAR-10

ResNet-18 on Imagenet

The results on ImageNet were also consistent with what is seen across all models, RK2 consistently decay the loss more than any other optimizer.

Here we experimented with a large number of Hyper-parameters and another experiment we did here was to nor use batch normalization. The exact effect of batch normalization on optimization is not well known and we can see that its presence can have a drastic effect on optimization.

Here we can see that with the same number of epochs, RK2 performs better but since it involves more computations, if one looks at the wall-clock time SGD performs better eventually.

	epochs done	best train loss	best train acc	best test loss	best test acc	optimizer	lr	epoch_step	weight_decay
SGD_nesterov_0	44/90	0.881294	80.46875	1.317926	67.912	SGD_nesterov	0.10	20	0.0001
SGD_momentum_0	44/90	0.921798	79.68750	1.316506	67.690	SGD_momentum	0.10	20	0.0001
SGD_nesterov_1	44/90	1.114107	78.90625	1.567616	62.196	SGD_nesterov	0.10	30	0.0001
SGD_momentum_1	43/90	1.125322	74.21875	1.581748	62.026	SGD_momentum	0.10	30	0.0001
RK2_heun_3	38/90	0.441446	74.21875	1.628603	60.792	RK2_heun	0.05	10	0.0005
RK2_heun_4	21/90	0.247642	71.87500	1.655458	60.540	RK2_heun	0.10	20	0.0001
RK2_heun_7	23/90	0.289915	68.75000	1.889192	55.558	RK2_heun	0.10	15	0.0005
RK2_heun_1	24/90	0.371729	67.96875	1.908143	55.284	RK2_heun	0.05	20	0.0009

Figure 4: Imagenet Test Accuracy

	epochs done	best train loss	best train acc	best test loss	best test acc	optimizer	lr	epoch_step	weight_decay
SGD_momentum_0	44/90	0.921798	79.68750	1.316506	67.690	SGD_momentum	0.10	20	0.0001
SGD_nesterov_0	44/90	0.881294	80.46875	1.317926	67.912	SGD_nesterov	0.10	20	0.0001
SGD_nesterov_1	44/90	1.114107	78.90625	1.567616	62.196	SGD_nesterov	0.10	30	0.0001
SGD_momentum_1	43/90	1.125322	74.21875	1.581748	62.026	SGD_momentum	0.10	30	0.0001
RK2_heun_3	38/90	0.441446	74.21875	1.628603	60.792	RK2_heun	0.05	10	0.0005

Figure 5: Imagenet Test Loss

8 Conclusion

We notice that Runge-Kutta methods when used as optimizers perform significantly better thqn Stochqstic Gradient Descent, however we do not see this perforance directly translated in the perofromance. This however is an open area of research, understanding generalization ability of neural networks.

	epochs done	best train loss	best train acc	best test loss	best test acc	optimizer	lr	epoch_step	weight_decay
SGD_nesterov_0	44/90	0.881294	80.46875	1.317926	67.912	SGD_nesterov	0.10	20	0.0001
SGD_momentum_0	44/90	0.921798	79.68750	1.316506	67.690	SGD_momentum	0.10	20	0.0001
SGD_nesterov_1	44/90	1.114107	78.90625	1.567616	62.196	SGD_nesterov	0.10	30	0.0001
SGD_momentum_1	43/90	1.125322	74.21875	1.581748	62.026	SGD_momentum	0.10	30	0.0001
RK2_heun_3	38/90	0.441446	74.21875	1.628603	60.792	RK2_heun	0.05	10	0.0005
RK2_heun_4	21/90	0.247642	71.87500	1.655458	60.540	RK2_heun	0.10	20	0.0001
RK2_heun_7	23/90	0.289915	68.75000	1.889192	55.558	RK2_heun	0.10	15	0.0005
RK2_heun_1	24/90	0.371729	67.96875	1.908143	55.284	RK2_heun	0.05	20	0.0009

Figure 6: Imagenet Train Accuracy

	epochs done	best train loss	best train acc	best test loss	best test acc	optimizer	lr	epoch_step	weight_decay
RK2_heun_4	21/90	0.247642	71.87500	1.655458	60.540	RK2_heun	0.1	20	0.0001
RK2_1	20/90	0.281883	57.03125	2.713761	41.952	RK2	0.1	30	0.0001
RK2_heun_7	23/90	0.289915	68.75000	1.889192	55.558	RK2_heun	0.1	15	0.0005
RK2_2	21/90	0.291872	57.03125	2.625814	43.072	RK2	0.1	20	0.0001
RK2_heun_2	25/90	0.333769	53.90625	3.543455	26.600	RK2_heun	0.1	30	0.0009

Figure 7: Imagenet Test Loss

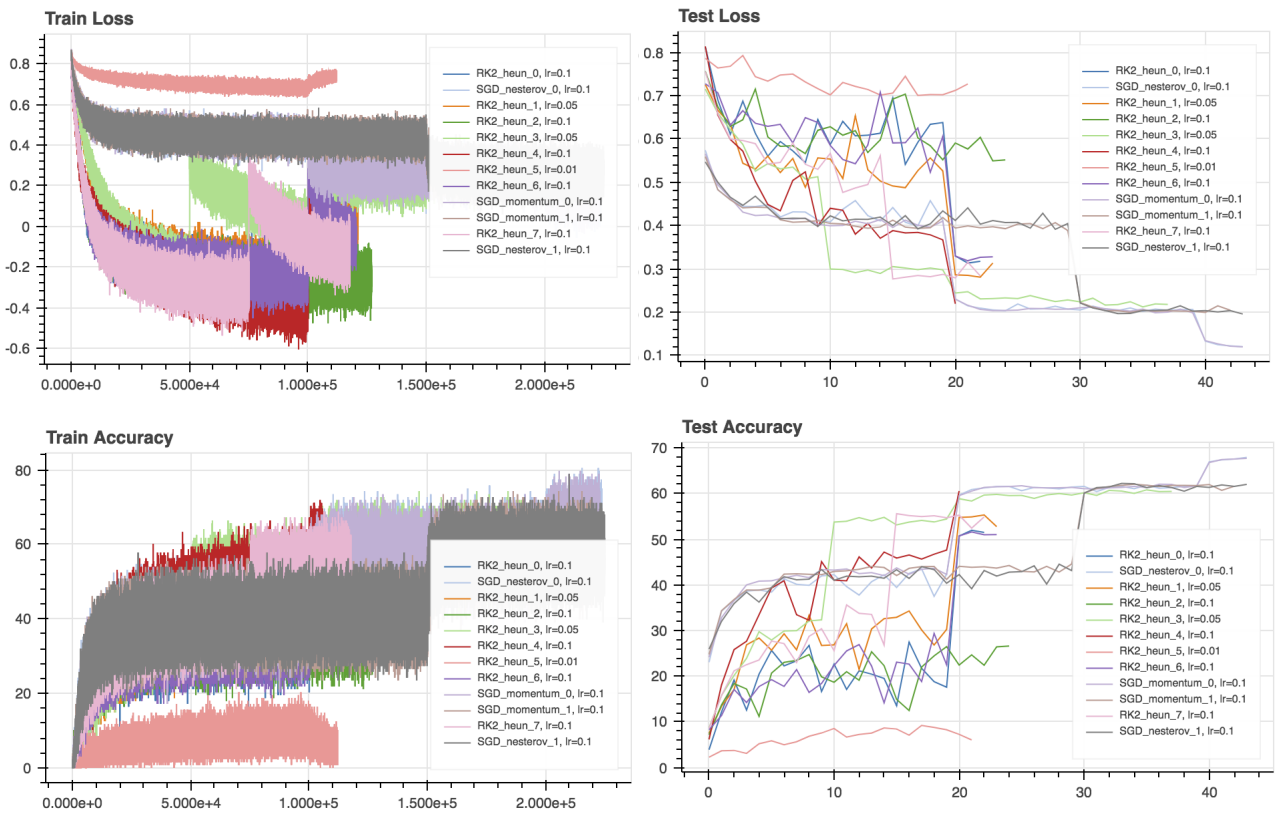


Figure 8: Log-Loss Plot & Accuracy : ResNet-18 on ImageNet

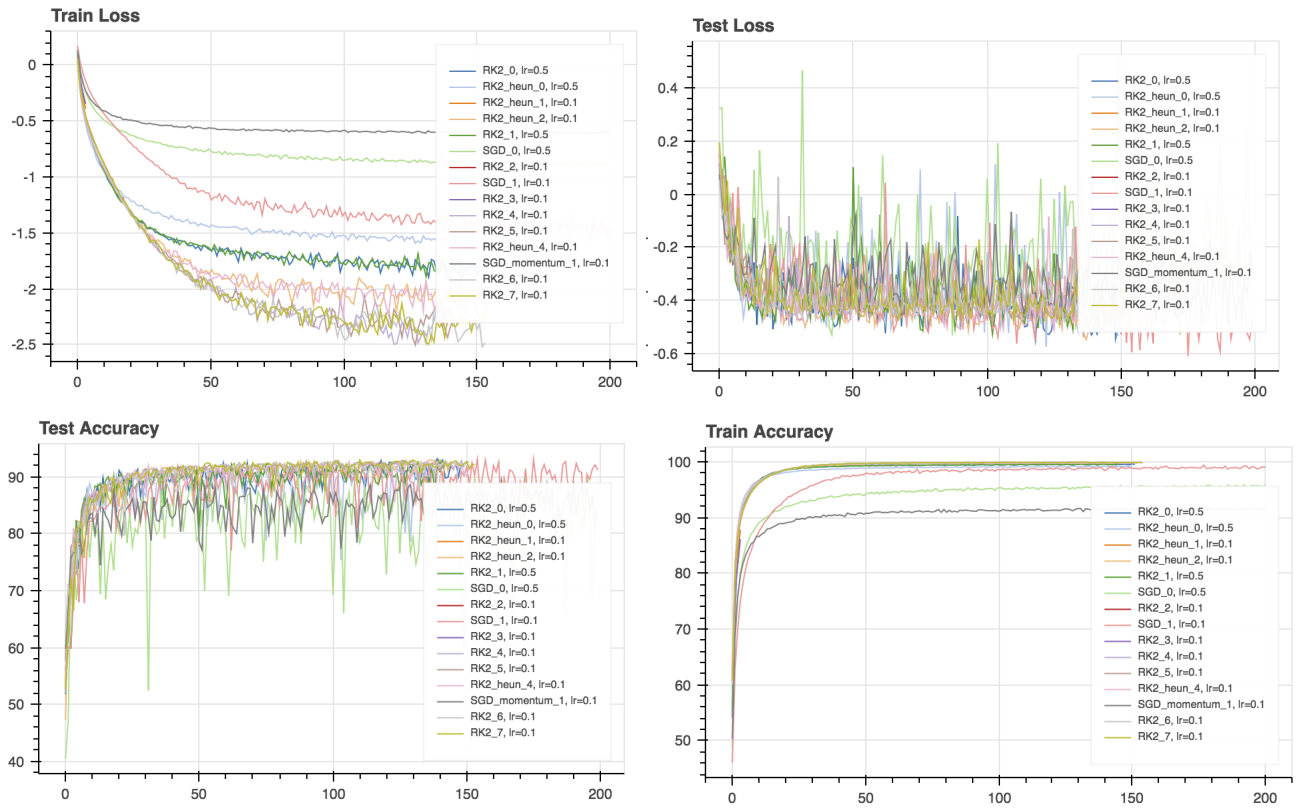


Figure 9: Log-Loss Plot & Accuracy : WideResNet-16 on CIFAR-10

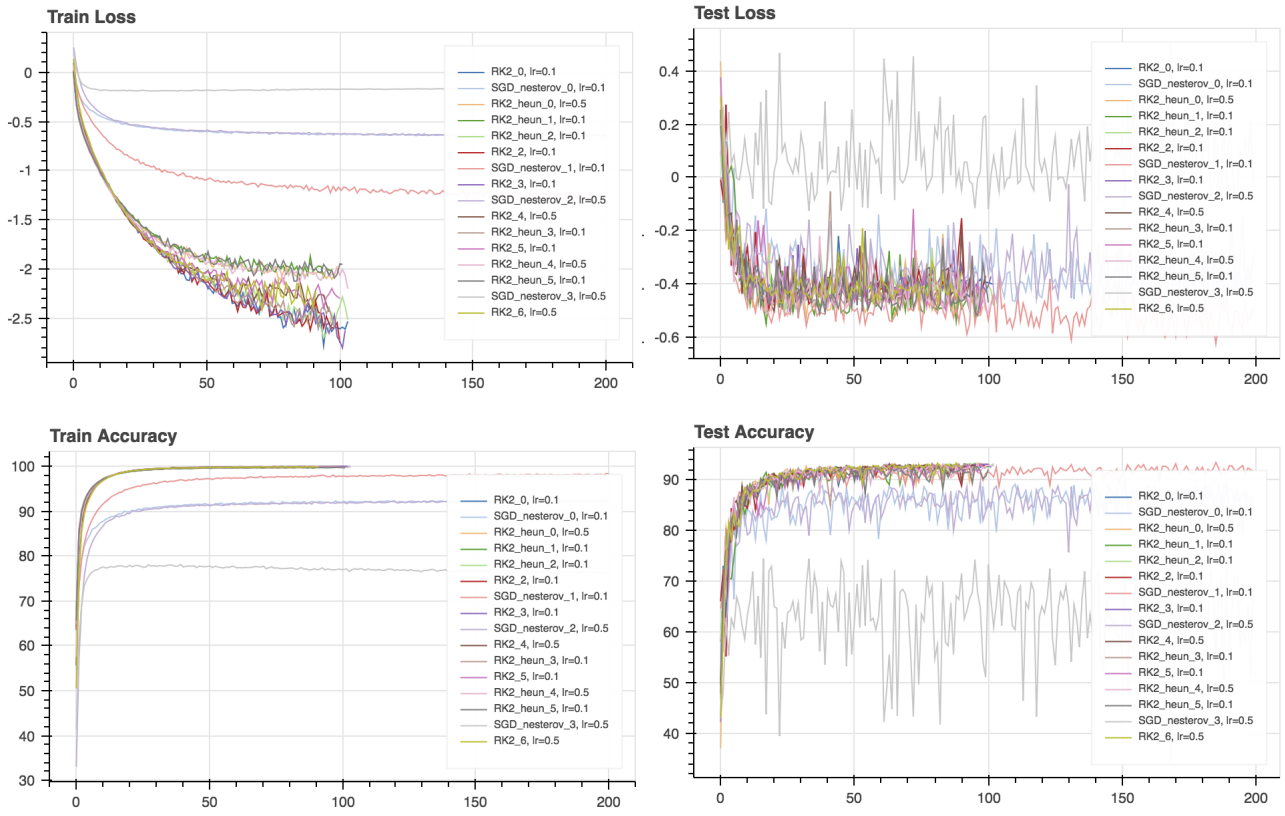


Figure 10: Log-Loss Plot & Accuracy : WideResNet-28 on CIFAR-10