International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

Exploration of Non-Convex Optimization Challenges Across Diverse Data Sets Using Machine Learning and Deep Learning Methods

Harish Kunder [1], Manjunath Kotari [2]

[1] Department,Name of Artificial Intellegence and Machine Learning, Alva's Institute of Engineering and Technology and , VTU Belagavi, ,INDIA
[2]Department,Name of Computer Science and Engineering, Alva's Institute of Engineering and Technology, VTU Belagavi INDIA

E-mail address: researchkunder@gmail.com, aietcse08@aiet.org.in,

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

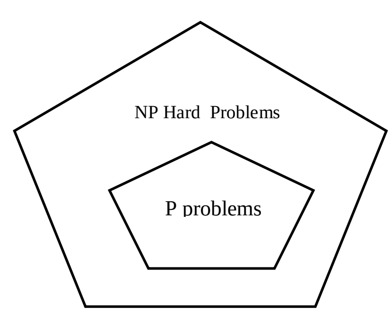
Abstract: In the modern era, experimenting with datasets to derive predictive insights has become both commonplace and highly effective. The success of experiments in machine learning and deep learning hinges on the availability of diverse datasets, which are important for achieving accurate outcomes across a spectrum of domains. Notably, primary datasets such as time series data often yield particularly efficient results. However, within this framework, the existence of NP-hard problems can present a significant challenge, potentially resulting in non-convex outputs. Addressing this challenge necessitates the transformation of NP-hard problems into P problems to optimize the outcomes. In instances where machine learning or deep learning analyses yield non-convex results, non-convex optimization methodologies come into play. These methodologies are designed to identify the global minimum amidst multiple local minima. This paper draws attention to datasets where suboptimal outcomes persist, underscoring the difficulty in achieving the global minimum in many scenarios. Furthermore, it provides insights into the prevalence of non-convex optimization challenges within these datasets, proposing avenues for future research aimed at making them more amenable to convex optimization techniques. By addressing these challenges, the field can enhance the efficiency and accuracy of predictive analytics, driving advancements in machine learning and deep learning applications.
Keywords: Non convex, Convex, Optimization, Global minimum, Local minimum.

1. Introduction

In today's data-focused research, digging into datasets to find useful predictions is not just common but really important. How well machine learning and deep learning experiments work depends a lot on the different and detailed datasets that are in use. These datasets are super important because they strongly impact how accurate, dependable, and useful the results are in many different areas.

Some datasets, like ones that track data over time or those considered primary sources, have their own built-in patterns [1]. These patterns helps to get really good and fitting results that make sense in different situations. This makes these kinds of datasets super important for making smart decisions in lots of industries.

But in this world of exploring data, there's a tough problem—the NP-hard problems. They're really tricky and can make things complicated by giving solutions that aren't straightforward. Exploring methods to transform NP-hard problems into more manageable P problems is key when addressing these formidable challenges. (Figure 1).



E-mail:author's email

Figure 1: Relationship of NP hard and P problems

Non-convex optimization is a fundamental aspect of machine learning and deep learning methods, addressing intricate problems across diverse datasets [2]. Unlike convex optimization, where the objective function has a single global minimum, non-convex optimization involves functions with multiple local minima, making the optimization process significantly challenging the visualization is shown in figure 2, figure 3 and figure 4.

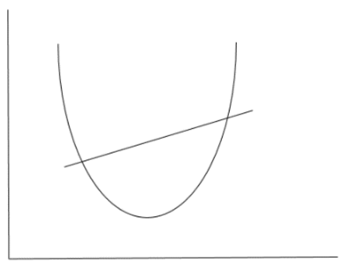


Figure 2: Convex optimization curve

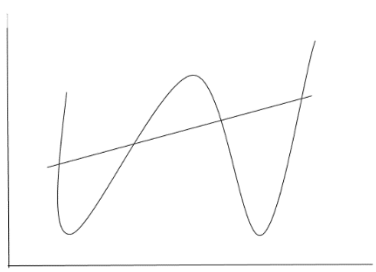


Figure 3. Non-Convex optimization curve

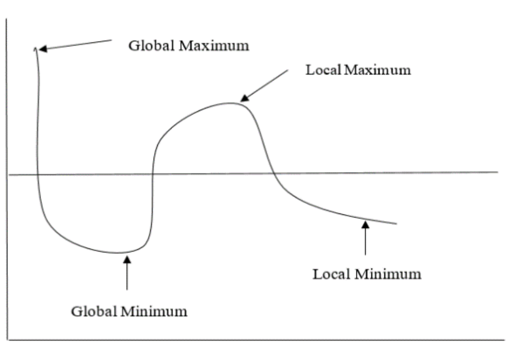


Fig. 4. minimum/maximum

In some scenarios, the learning task's natural objective is a non-convex function [3]. This notably occurs in training deep neural networks or dealing with tensor decomposition problems. While non-convex objectives and constraints allow for accurate modeling of learning problems, they present a significant challenge to algorithm designers. Unlike convex optimization, solving non-convex problems lacks a readily available toolkit. Many non-convex optimization problems are recognized as NP-hard, posing a substantial hurdle. Moreover, several of these non-convex problems are not only NP-hard to solve optimally but also NP-hard to solve approximately further complicating the scenario [1].

In the realm of machine learning and deep learning, many real-world problems exhibit non-convex characteristics due to the presence of complex interactions and high-dimensional data [4]. The optimization of non-convex functions involves finding optimal solutions amidst multiple local optima, saddle points, and plateaus.

Mathematically, a non-convex optimization problem can be represented as follows:

Minimize $f(x)$, where x belongs to a set X and $f(x)$ is a non-convex function. The objective is to find x^*such that $f(x^*)$ is minimized.

A. Diverse Data Sets and Challenges

Across various datasets, the non-convex nature of optimization poses unique challenges. These challenges arise in fields like computer vision, natural language processing, and reinforcement learning, among others [1].

For instance, in computer vision tasks, training deep neural networks involves optimizing non-convex loss functions. The presence of multiple local minima can impact the convergence and generalization of models. Similarly, in natural language processing, optimizing complex models like transformers on diverse text corpora encounters non-convexity challenges due to the high-dimensional nature of language representations.

B. Machine Learning and Deep Learning Methods

Various optimization algorithms are employed to tackle non-convex optimization challenges. Stochastic gradient descent (SGD) and its variants, such as Adam, RMSprop, and momentum-based methods, are commonly used in training deep neural networks. These algorithms navigate through the non-convex landscape by iteratively updating model parameters to find promising optima [1].

Moreover, techniques like random restarts, initialization strategies, and adaptive learning rates are employed to mitigate the impact of local minima and improve convergence rates.

2. LITERATURE SURVEY

A novel approach to identifying intrusions within social media networks is presented in [8]. This method utilizes soft computing techniques by integrating fuzzy clustering, particle swarm optimization (PSO), and a multi-layer perceptron (MLP) neural network to address non-convex optimization problems. The study underscores the practicality of this approach, particularly in financial applications.

In [9], the Water Cycle Optimization (WCO) algorithm is introduced as a means to address the economic dispatch problem in power systems. WCO, a stochastic search algorithm inspired by the natural water cycle, combines global and local search characteristics. Through simulations on a standard test system, its efficacy is demonstrated and compared against existing optimization methods. Results show WCO's ability to generate high-quality solutions for non-convex economic dispatch problems, boasting faster convergence rates and superior performance. Furthermore, researchers have improved the particle swarm optimization (PSO) algorithm to better handle non-convex optimization challenges.

In [10], an extensive review delves into present studies concerning distributed learning within non-convex optimization problems. It covers scenarios involving batch and streaming data, exploring diverse distributed optimization algorithms like stochastic gradient descent, coordinate descent, and proximal algorithms. Authors underscore the importance of handling issues such as communication efficiency, fault tolerance, and privacy protection within distributed learning. Moreover, they identify several unresolved challenges and potential research directions for the future.

In [11], an extensive overview explores the latest progress in non-convex optimization methods applied to signal processing and machine learning applications. It encompasses a range of optimization algorithms like gradient descent, alternating direction methods of multipliers, and proximal gradient methods. The paper also delves into the obstacles related to non-convex optimization, such as navigating multiple local optima and the challenge of ensuring global convergence. Additionally, the authors showcase the effectiveness of non-convex optimization methods in diverse signal processing and machine learning tasks, including matrix factorization, compressed sensing, and deep learning.

In [12], a novel approach utilizing second-order optimization is introduced to address non-convex optimization problems within machine learning. Through empirical studies across diverse machine learning applications like matrix factorization and deep learning, the authors assess the efficiency of this method against commonly used first-order optimization techniques. Results indicate that the proposed approach consistently surpasses first-order methods in terms of both convergence speed and final accuracy, especially in scenarios involving non-convex problems with highly complex and non-linear objective functions.

In [13], a ray-tracing-based approach is introduced for determining the subsequent optimal solution in multi-objective optimization challenges. This algorithm initiates by generating a set of random solutions, followed by their arrangement via a Pareto-dominance criterion. Subsequently, a ray extends from the present solution toward the non-dominated solutions, selecting the next optimal solution based on the intersection point of the ray with the Pareto front. The method's performance is assessed across various benchmark problems, revealing its superiority over existing techniques concerning solution convergence and diversity. The authors propose its potential applicability to a wide array of multi-objective optimization problems.

The paper [14] introduces a two-stage algorithm designed to address nonlinear non-convex minimum cost flow problems by combining a genetic algorithm with a local search approach. Initially, the genetic algorithm generates an initial population, followed by a refinement stage using a quasi-Newton-based local search algorithm. The algorithm's efficacy is assessed across benchmark problems and benchmarked against several state-of-the-art methods. The findings illustrate the superiority of this proposed algorithm in terms of solution quality and computational efficiency compared to other approaches. Consequently, the authors propose this method as a promising solution for analogous nonlinear non-convex optimization problems.

In their work [15], the authors address the complexities of optimizing wellbore trajectories and propose the utilization of Particle Swarm Optimization (PSO) to tackle this non-convex optimization challenge. They meticulously explain the PSO algorithm and its diverse adaptations, encompassing hybrid PSO methodologies. The authors delve into the strengths and constraints of employing PSO specifically in wellbore trajectory optimization, offering insights and suggestions for future investigations in this domain. Their article serves as a significant resource for researchers and industry professionals seeking to leverage PSO for solving non-convex optimization issues within petroleum engineering.

The paper in [16] introduces a novel method for anomaly detection in cyber-physical production systems (CPPS) by constructing a non-convex hull that captures the underlying geometric structure of the data. Tackling the complexities posed by high-dimensional data and intricate distributions in CPPS, the authors devise an iterative optimization algorithm to obtain this hull. The proposed method's evaluation on both simulated and real-world data demonstrates its superior performance in accuracy and computational efficiency compared to existing techniques. The authors anticipate its potential application in detecting anomalous events across various industrial processes, offering utility for researchers and practitioners in this domain.

In [17], the focus lies on non-convex optimization within control systems. Specifically addressing scenarios where a non-convex objective function faces inequality constraints, the authors propose a gradient-based method. This method involves projecting the gradient onto the feasible set, leading to non-smoothness and non-differentiability of the objective function. Conditions for the differentiability of the projected trajectory are provided, demonstrating the method's robust convergence to a stationary point under certain assumptions, elucidated through numerical examples.

Another paper, [18], presents a new method for addressing sparse multiple instance learning (MIL) problems using a non-convex penalty function. Employing the alternating direction method of multipliers (ADMM) algorithm, the authors optimize this non-convex penalty function, showcasing its superior performance in sparse MIL problems through experiments on various benchmark datasets, excelling in both classification accuracy and sparsity compared to existing methods.

The work in [19] proposes a real-time algorithm tailored for powered descent guidance capable of handling non-convex problems. Designed to navigate spacecraft to safe landing sites using limited computational resources and onboard sensors, the paper introduces a modified branch-and-bound algorithm. Extensive numerical simulations demonstrate the algorithm's success in guiding spacecraft to safe landing sites with high accuracy and efficiency, highlighting its potential for real-world missions in spacecraft guidance and control.

Additionally, [20] presents a deterministic algorithmic framework addressing the non-convex phase retrieval problem. Ensuring solution uniqueness, the authors introduce a sequence of non-convex optimization problems with convex constraints, guaranteeing exact signal recovery with high probability and requiring fewer measurements compared to existing methods, validated through numerical simulations.

The authors in [21] propose a novel method for detecting and segmenting salient image regions using non-convex non-local reactive flows. Providing a comprehensive overview of the approach, including mathematical formulations and implementation details, the paper demonstrates superior performance in accuracy and computational efficiency compared to existing methods. Through compelling case studies, it offers evidence of the proposed approach's effectiveness in saliency detection and segmentation, outlining the challenges and limitations of current methodologies.

Table 1: Summary of the review

Algorithm	Implementation	Research Focus
Hybrid Genetic Algorithm [22]	GA operators such as selection, crossover, and mutation can lead to convergence on a set of solutions.	Achieving the global optimum within the optimal timeframe is unattainable.
Non-Convex Gradient Descent [23]	It proves beneficial in discovering optimal solutions for non-convex problems characterized by a best-fit structure within a feasible time frame.	A rapid learning pace may lead the system to overlook the global optimum, whereas a slower pace could prolong the journey towards reaching it.
Alternating Minimization Principal [24]	When tackling an optimization problem involving numerous sets of variables	Its practical application primarily emerges in cases where the marginal optimization problem is straightforward.
EM Algorithm [25]	Its core function lies in complementing datasets lacking specific information. Within clustering, it forms the bedrock for unsupervised learning processes.	Attaining convergence solely to a local optimum is feasible, necessitating the utilization of both forward and backward probabilities.
Gradient Descent and Langevin Dynamics [26]	The strategy that appears most effective when employed together is replica exchange.	The most optimal method to exchange replicas depends on the specifics of the particular physical system under consideration.
Stochastic Optimization Techniques [27]	An invaluable approach for tackling the non-convex optimization challenges arising from diverse applications like deep learning, neural networks, top modeling, etc. It stands out as the preferred method for addressing uncertainties and ambiguities.	In high-dimensional scenarios, the presence of saddle points can be circumvented when the objective function is differentiable.

3. Data for Analysis

The utilization of three datasets aims to demonstrate the existence of non-optimal solutions and underscores the necessity for employing optimal methods to address the problem effectively.

Dataset 1 [24]: The "Kinematic Insights" dataset offers a comprehensive collection of velocity and position attributes pertaining to various entities in motion. It encompasses a broad range of scenarios, from vehicle movements to object trajectories, aiming to provide researchers and analysts with a detailed understanding of motion dynamics. This dataset serves as a valuable resource for studying patterns, developing predictive models, and testing algorithms across fields such as transportation, robotics, sports analytics, biomechanics, and physics simulations. With structured data formats and accompanying documentation, it facilitates analysis and exploration while offering insights into motion behavior across different environments and conditions.

Dataset 2 [25]: The "Polynomial Dataset: Exploring Curves and Trends" provides a collection of data points generated from polynomial functions, allowing researchers and analysts to explore various curves and trends. This dataset offers valuable insights into the behavior of polynomial equations across different degrees and coefficients, enabling the study of curve fitting, interpolation, and extrapolation techniques. With structured data formats and accompanying documentation, it facilitates the analysis of polynomial relationships and the development of mathematical models for predictive purposes. Researchers can utilize this dataset to gain a deeper understanding of polynomial functions and their applications in fields such as mathematics, statistics, engineering, and data science.

Dataset 3 [26]: The MNIST dataset is renowned in machine learning for its collection of 70,000 grayscale images, each measuring 28x28 pixels, depicting handwritten digits from 0 to 9, accompanied by corresponding labels. It stands as a cornerstone for evaluating machine learning algorithms, particularly in image classification. Researchers and practitioners rely on MNIST to develop and assess models for tasks such as digit and pattern recognition, as well as handwriting analysis. Its widespread use stems from its accessibility, straightforwardness, and clear labeling, making it invaluable for both newcomers and seasoned professionals in the field of machine learning, providing a standardized benchmark for evaluating algorithmic performance across various methodologies and techniques.

4. Methodology

Figure 4 illustrates the utilization of various datasets that were modeled to assess the attainment of local minimum results. The unsupervised dataset was subjected to k-means clustering, while the supervised dataset underwent optimization via Particle Swarm Optimization (PSO) in alignment with machine learning principles. Additionally, deep learning concepts were employed to further investigate and validate the absence of optimization results. The dataset and algorithm selection are summarized in Table 2.

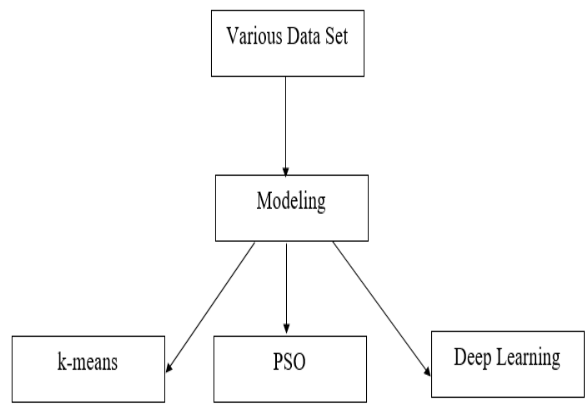


Fig. 5. Block diagram for Analysis

Table 2: Dataset Summary

Sl. No.	Dataset name	Algorithm
1	Kinematic Insights	K-means
2	Polynomial Dataset: Exploring Curves and Trends	PSO
3	MNIST dataset	Deep learning

A. *Kmeans*

K-means clustering is a popular unsupervised machine learning algorithm used for partitioning a dataset into K distinct, non-overlapping clusters. The algorithm begins by randomly initializing K cluster centroids, typically chosen from the dataset itself. Subsequently, it iteratively assigns each data point to the nearest centroid based on a distance metric, often Euclidean distance.

Algorithm 1: Kmeans

1. Initialization
   - Choose the number of clusters, K.
   - Randomly initialize K cluster centroids, $\mu_1, \mu_2, \ldots, \mu_k$
2. Assign Data Points to Nearest Centroids:
   - For each data point, x_i, calculate its distance to each centroid using Euclidean distance:

   $$dist(x_i, B_j) = \sqrt{\sum (x_{ii} - B_{jj})^2}$$

   - Assign the data point, x_i, to the cluster with the nearest centroid:

   $$argmin_j dist(x_i, B_j) = C_i$$

   where c_i represents the cluster assignment of data point x_i.
3. Update Centroids:
   - Recalculate the centroids of the clusters by taking the mean of all data points assigned to each cluster:

   $$B_j = \left(\frac{1}{|C_j|}\right) \sum (x_i \in C_j) x_i$$

   where $|C_j|$ represents the number of data points assigned to cluster j.
4. Convergence Check:
   - Check if the centroids have changed significantly.
   - If centroids have changed, repeat steps 2 and 3.
   - If centroids have not changed significantly, the algorithm has converged.
5. Output:
   - Final cluster assignments.
   - Centroid coordinates.

After the data points have been assigned, the centroids are recalculated by taking the mean of all the data points assigned to each cluster. This process of reassigning data points and updating centroids continues until a convergence criterion is met, such as when the centroids no longer change significantly or when a maximum number of iterations is reached. K-means aims to minimize the within-cluster variance, resulting in compact and well-separated clusters. However, the algorithm's effectiveness can be influenced by the initial placement of centroids and the choice of K, requiring careful consideration and sometimes multiple runs with different initializations. Despite its simplicity, K-means is widely used in various applications, including image segmentation, customer segmentation, and anomaly detection, due to its efficiency and scalability (Algorithm 1).

B. PSO

Algorithm 2: PSO

1. Initialization
   - Initialize population of particles with random positions p_i and velocities v_i
2. Evaluation:
   - Evaluate fitness for each particle using the objective function $f(p_i)$
3. Update Personal Best:
   - Update personal best position (pbest) for each particle

   $$pbest_i = \{ p_i, if\ f(p_i) < f(pbest_i) \quad \{pbest_i, otherwise$$

4. Update Global Best:
   - Update global best position (gbest) based on the personal best positions of all particles

   $$gbest_i = argmini_i(f(pbest_i))$$

5. Update Velocity and Position:
   - Update velocity for each particle

   $$v_i = wv_i + c_1 r_1 (pbest_i - p_i) + c_2 r_2 (gbest_i - p_i)$$

   - Update position for each particle:

   $$p_i = p_i + v_i$$

   where:
   - w is the inertia weight.
   - c_1 and c_2 are acceleration coefficients.
   - r_1 and r_2 are random numbers between 0 and 1.
6. Convergence Check:
   - Repeat steps 2 to 5 until a termination condition is met.
7. Output:
   - Best solution found represents the optimal solution.

Particle Swarm Optimization (PSO) is an optimization algorithm inspired by the social behavior of organisms like birds and fish. In PSO, a population of particles navigates through a search space to find the optimal solution to a given problem. Each particle represents a potential solution and adjusts its position based on its own experience (personal best) and the collective knowledge of the swarm (global best). By iteratively updating their positions and velocities, particles explore the search space in search of the optimal solution. PSO is favored for its simplicity and ability to efficiently explore high-dimensional search spaces, making it widely applicable in various optimization tasks (Algorithm 2).

C. Deep Learning

Deep learning is a subset of machine learning that focuses on the development and training of artificial neural networks with multiple layers (hence the term "deep"). These neural networks are designed to learn and extract hierarchical representations of data, enabling them to automatically discover intricate patterns and features from raw input data. Deep learning has revolutionized various fields such as computer vision, natural language processing, speech recognition, and more, by achieving state-of-the-art performance in tasks like image classification, object detection, language translation, and speech synthesis. Key components of deep learning include convolutional neural networks (CNNs) for processing visual data, recurrent neural networks (RNNs) for sequential data, and transformers for processing sequential data in parallel. Deep learning techniques are often used in conjunction with large datasets and powerful computing resources to train complex models with millions or even billions of parameters. Algorithm 3 explains the step by step procedure applying CNN on MNIST dataset. Principal Component Analysis (PCA) is applied to visualize the multiple local minimum.

Algorithm 3 : Visualizing Loss Landscape with CNN

---

1. Load MNIST dataset and normalize pixel values.
2. Define a CNN model architecture.
3. Train the CNN model multiple times, collecting loss values.
4. Plot loss curves for each training run.
5. Apply PCA to reduce loss values to 2 dimensions.
6. Visualize the 2D PCA projection of the loss landscape.

The algorithm 4 explains detail how the PCA projection is applied to view the loss projection on MNIST.

Algorithm 4: Principal Component Analysis (PCA)

Input: Dataset X

Output: Reduced dimensionality representation of X

1. Standardize X by subtracting mean and dividing by standard deviation.
2. Compute the covariance matrix of standardized X.
3. Perform eigendecomposition on the covariance matrix.
4. Select the top k eigenvectors corresponding to the largest eigenvalues.
5. Project the standardized dataset onto the new subspace formed by selected eigenvectors.
6. Output the reduced dimensionality representation of the dataset.

---

5. RESULTS

The results are divided into three parts, each focusing on different datasets. Part I examines an unsupervised dataset using K-means clustering, Part II explores a supervised dataset using Particle Swarm Optimization (PSO), and Part III employs deep learning techniques. The findings reveal the presence of non-optimized data points within the dataset. These results underscore the need for developing new methods to achieve optimal solutions.

A. Part 1: K-means on Kinematic Insights

At first, we start with a goal in mind: achieving a certain level of accuracy using regression analysis. To reach this goal, we employ two different models: K-means and quantile regression. These models help us work towards the desired accuracy.

Now, when it comes to unsupervised machine learning, where data points are not given label to train on, systems often face challenges in finding the best solution. The two mathematical equations are used to prove that the dataset provides a non-optimal solution.

The following are the terms used in the result,

- ACC_GLR: Establishes the target accuracy level.
- ACC_SLR_Kmeans: Evaluates the accuracy of a supervised learning regression model using K-means clustering for comparison.
- ACC_SLR_Quantile: Evaluates the accuracy of a supervised learning regression model using quantile regression for comparison.

Both K-means and quantile regression fail to yield the anticipated outcomes, exhibiting numerous valleys in the graph, indicating persistent challenges with local minima in machine learning (Figure 6).

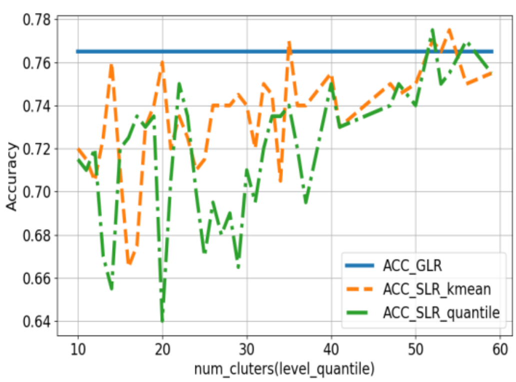


Fig. 6. Kmeans and Quantile regression on Kinematic Insights

B. Part 2: PSO on Polynomial Dataset: Exploring Curves and Trends

Particle Swarm Optimization (PSO) is a heuristic optimization algorithm inspired by the social behavior of bird flocks and fish schools. In the context of polynomial data fitting, PSO aims to find the coefficients of the polynomial that minimize the error between the polynomial and the actual data points. However, due to its stochastic nature and reliance on local information exchange among particles, PSO may struggle to escape from local minima in the search space. Each particle in the swarm explores the solution space by adjusting its position based on its own experience and the best positions found by neighboring particles. While this collaborative approach facilitates rapid exploration, it also increases the likelihood of particles converging towards local minima instead of the global minimum, particularly in complex and rugged search spaces. Consequently, PSO may provide non-optimal solutions for polynomial data fitting tasks when trapped in local minima, hindering its ability to identify the best-fitting polynomial coefficients across the entire search space.

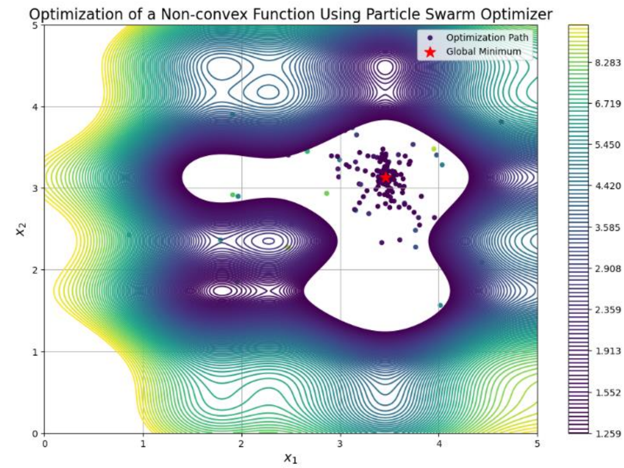


Fig. 7. PSO on Polynomial Dataset: Exploring Curves and Trends

C. Part 3: Deep learning on MNIST

Deep learning techniques were employed on the MNIST dataset to assess suboptimal solutions. Utilizing a sequential model architecture, three layers were introduced, as summarized in Table 3 Activation functions were employed to address the complexities inherent in non-linear data. Particularly, softmax activation was utilized for multi-class classification, ensuring that output values represent probabilities that collectively sum to 1, thereby indicating the likelihood of each class.

The model was compiled using the Adam optimizer, a widely-used optimization algorithm in neural network training. Adam dynamically adjusts the learning rate during training, facilitating efficient convergence. The loss function employed was sparse_categorical_crossentropy, tailored for multi-class classification tasks wherein labels are integers rather than one-hot encoded. This loss function effectively guides the model's training process by quantifying prediction errors.

Evaluation of model performance during training was conducted based on accuracy metrics. Accuracy measures the proportion of correctly classified images, providing insight into the model's effectiveness in classifying unseen data samples.

Table 3: Deep learning layers with Adam optimizer

Layer	Layer Name	Activation function	Mathematical derivation
1	Flatten	Input Layer	-
2	Dense	Relu	$f(x) = (0, x)$
3	Dense	Softmax	$y_i = e^{z_i} / \Sigma\, e^{z_j}$
Adam optimizer			

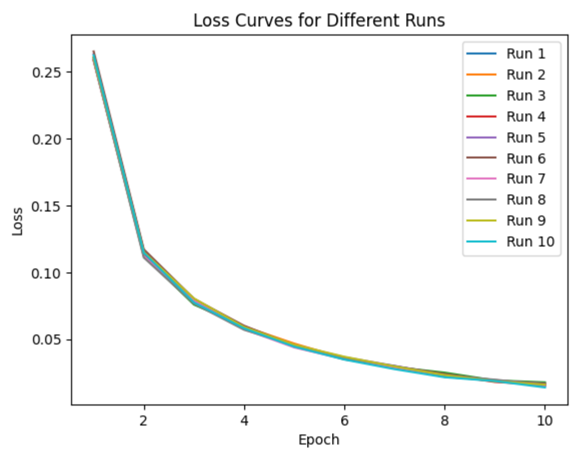


Fig. 8. Loss function MNIST using Adam optimizer

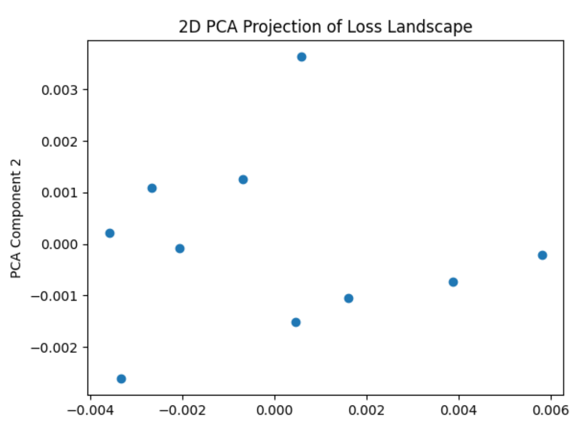


Fig. 9. PCA components to visualize a local minimum using Adam Optimizer

The loss graph depicted in Figure 8 illustrates the absence of local minimum points, attributed to the high dimensionality of the problem. However, upon conducting dimensionality reduction using PCA, the loss graph in Figure 9 reveals the presence of multiple local minima. Consequently, the primary challenge lies in distinguishing the global minimum to optimize the transition from non-convex results to a convex solution. The loss function is calculated using the formula (1)

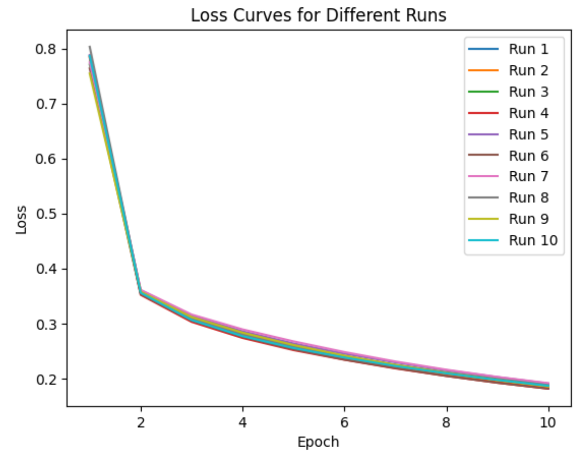


Fig. 10. Loss function MNIST using SGD optimizer

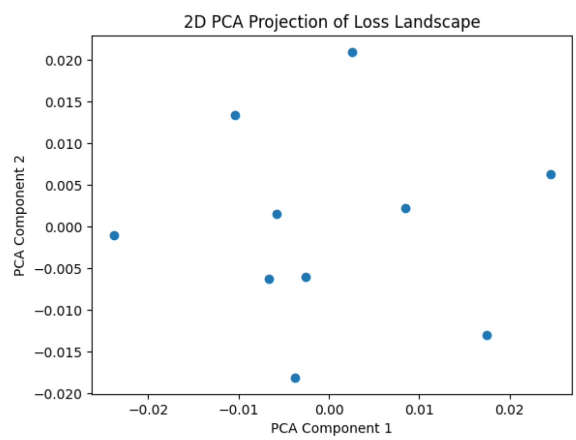


Fig. 11. PCA components to visualize a local minimum using SGD Optimizer

$$L(z, \hat{y}) = -1/M * \Sigma\left(log\ log\ \left(\hat{y}_i^{z_i}\right)\right) \text{ --------------- } (1)$$

where

$L(z, \hat{y})$ is a loss function

M is a number of samples

z_i is the true class label for sample i

$\hat{y}_i$ is the predicted probability distribution for sample i.

$\hat{y}_i^{z_i}$ is the predicted probability assigned to the true class label z_i for sample i.

Table 4 provides a summary of the deep learning layers applied to the MNIST dataset using the SGD optimizer. Additionally, Figures 10 and 11 illustrate the loss curve and the local minimum points obtained through PCA application.

Table 4: Deep learning layers with SGD optimizer

Layer	Layer Name	Activation function	Mathematical derivation
1	Flatten	Input Layer	-
2	Dense	Relu	$f(x) = (0, x)$
3	Dense	Softmax	$y_i = e^{z_i} / \Sigma\, e^{z_j}$
SGD optimizer			

Table 5 provides a summary of the deep learning layers applied to the MNIST dataset using the Adadelta optimizer. Additionally, Figures 12 and 13 illustrate the loss curve and the local minimum points obtained through PCA application.

Table 5: Deep learning layers with Adadelta optimizer

Layer	Layer Name	Activation function	Mathematical derivation
1	Flatten	Input Layer	-
2	Dense	Relu	$f(x) = (0, x)$
3	Dense	Softmax	$y_i = e^{z_i} / \Sigma\, e^{z_j}$
Adadelta optimizer			

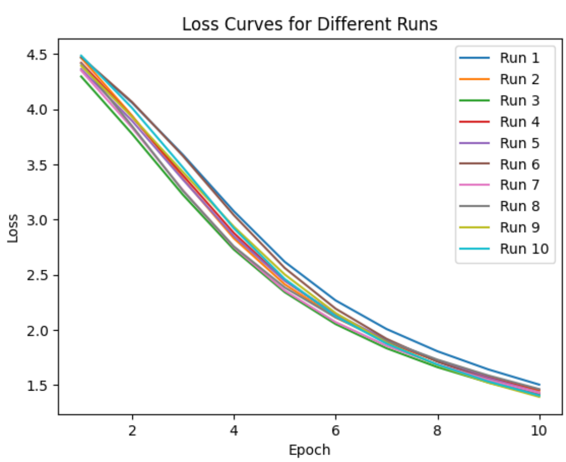


Fig. 12. Loss function MNIST using Adadelta optimizer

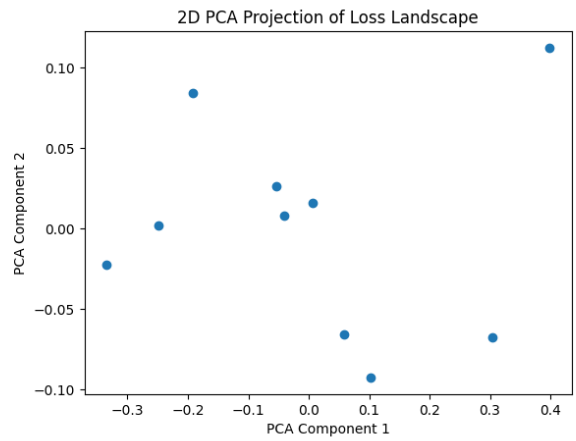


Fig.13. PCA components to visualize a local minimum using Adadelta Optimizer

6. Conclusion

Real-time data sets are integral to numerous applications, offering valuable insights and solutions to complex problems. However, they often pose non-convex optimization challenges, characterized by the presence of multiple local minima necessitating the selection of a global minimum for optimal results. This paper undertakes the task of substantiating the existence of non-convex solution spaces within real-time datasets. To achieve this, a diverse range of datasets from unsupervised, supervised, and deep learning domains is examined, collectively illustrating the pervasive nature of non-convexity across various data types and learning scenarios.

Identifying these non-convex solutions presents a formidable challenge, particularly given the limitations of traditional optimization techniques. Even renowned optimizers like ADAM struggle to effectively navigate the complexities associated with local minima, underscoring the inadequacies of conventional methods in addressing real-time dataset optimization. Consequently, specialized methodologies tailored to handle non-convex landscapes are deemed necessary to optimize real-time datasets effectively.

In conclusion, we have to find the way to minimize the loss function and get the global minimum. The optimization of any method gives the best solution to the problem, and it depends upon the dataset and the hyper parameter of the optimizers.

ACKNOWLEDGMENT

We express our gratitude to the principal and management of Alva's Institute of Engineering & Technology for their unwavering support throughout our research. We would also like to acknowledge the encouragement and support provided by our family members during the research process. Additionally. We extend our sincere thanks to Dr. Gopalakrishna N Kini, Professor, Department of CSE, MIT Manipal, Dr. Jyothi Shetty, Professor & Head, Department of CSE, NMAMIT Nitte, and Dr. Sarika Hgde, Professor, Department of CSE, NMAMIT Nitte for their timely suggestions and guidance.

REFERENCES

[1]. Jain, Prateek, and Purushottam Kar. "Non-convex optimization for machine learning." arXiv preprint arXiv:1712.07897 (2017).

[2]. Alomoush, Muwaffaq I. "Optimal Combined Heat and Power Economic Dispatch Using Stochastic Fractal Search Algorithm." Journal of Modern Power Systems and Clean Energy 8, no. 2 (2020): 276-286.

[3]. Yonel, Bariscan, and Birsen Yazici. "A Deterministic Convergence Framework for Exact Non-Convex Phase Retrieval." arXiv preprint arXiv:2001.02855 (2020).

[4]. Galiano, G., I. Ramírez, and E. Schiavi. "Non-convex non-local reactive flows for saliency detection and segmentation." Journal of Computational and Applied Mathematics (2020): 112873.

[5]. Hauswirth, Adrian, Florian Dörfler, and Andrew Teel. "On the Differentiability of Projected Trajectories and the Robust Convergence of Non-convex Anti-Windup Gradient Flows." IEEE Control Systems Letters (2020).

[6]. Zhang, Yuqi, Haibin Zhang, and Yingjie Tian. "Sparse multiple instance learning with non-convex penalty." Neurocomputing (2020).

[7]. Reynolds, Taylor, Danylo Malyuta, Mehran Mesbahi, Behcet Acikmese, and John M. Carson. "A Real-Time Algorithm for NonConvex Powered Descent Guidance." In AIAA Scitech 2020 Forum, p. 0844. 2020.

[8]. Sathesh, A. "Enhanced Soft Computing Approaches for Intrusion Detection Schemes in Social Media Networks." Journal of Soft Computing Paradigm (JSCP) 1, no. 02 (2019): 69-79.

[9]. Chaudhary, Vishal, Hari Mohan Dubey, and Manjaree Pandit. "Solution of Non-convex Economic Dispatch Problems by Water Cycle Optimization." In Advanced Engineering Optimization Through Intelligent Techniques, pp. 819-829. Springer, Singapore, 2020.

[10]. Chang, Tsung-Hui, Mingyi Hong, Hoi-To Wai, Xinwei Zhang, and Songtao Lu. "Distributed Learning in the Non-Convex World: From Batch to Streaming Data, and Beyond." arXiv preprint arXiv:2001.04786 (2020).

[11]. So, Anthony Man-Cho, Prateek Jain, Wing-Kin Ma, and Gesualdo Scutari. "Non-Convex Optimization for Signal Processing and Machine Learning." (2020).

[12]. Xu, Peng, Fred Roosta, and Michael W. Mahoney. "Second-order optimization for non-convex machine learning: An empirical study." In Proceedings of the 2020 SIAM International Conference on Data Mining, pp. 199-207. Society for Industrial and Applied Mathematics, 2020.

[13]. Nowak, Dimitri, and Karl-Heinz Küfer. "A Ray Tracing Technique for the Navigation on a Non-convex Pareto Front." arXiv preprint arXiv:2001.03634 (2020).

[14]. Ghasemishabankareh, Behrooz, Melih Ozlen, Xiaodong Li, and Kalyanmoy Deb. "A genetic algorithm with local search for solving single-source single-sink nonlinear non-convex minimum cost flow problems." Soft Computing 24, no. 2 (2020): 1153-1169.

[15]. Biswas, Kallol, Pandian M. Vasant, Moacyr Batholomeu Laruccia, José Antonio Gámez Vintaned, and Myo M. Myint. "Review on Particle Swarm Optimization Approach for Optimizing Wellbore Trajectory." In Deep Learning Techniques and Optimization Strategies in Big Data Analytics, pp. 290-307. IGI Global, 2020.

[16]. Li, Peng, and Oliver Niggemann. "Non-convex hull based anomaly detection in CPPS." Engineering Applications of Artificial Intelligence 87 (2020): 103301

[17]. Hauswirth, Adrian, Florian Dörfler, and Andrew Teel. "On the Differentiability of Projected Trajectories and the Robust Convergence of Non-convex Anti-Windup Gradient Flows." IEEE Control Systems Letters (2020).

[18]. Zhang, Yuqi, Haibin Zhang, and Yingjie Tian. "Sparse multiple instance learning with non-convex penalty." Neurocomputing (2020).

[19]. Reynolds, Taylor, Danylo Malyuta, Mehran Mesbahi, Behcet Acikmese, and John M. Carson. "A Real-Time Algorithm for NonConvex Powered Descent Guidance." In AIAA Scitech 2020 Forum, p. 0844. 2020.

[20]. Yonel, Bariscan, and Birsen Yazici. "A Deterministic Convergence Framework for Exact Non-Convex Phase Retrieval." arXiv preprint arXiv:2001.02855 (2020).

[21]. Galiano, G., I. Ramírez, and E. Schiavi. "Non-convex non-local reactive flows for saliency detection and segmentation." Journal of Computational and Applied Mathematics (2020): 112873.

[22]. Tianyi Lin, Chi Jin and Michael I Jordan, "On gradient descent ascent for nonconvex-concave minimax problems", 2019.

[23]. Robert Mansel Gower, Nicolas Loizou, Xun Qian, AlibekSailanbayev, Egor Shulgin and Peter Richtárik, "Sgd: General analysis and improved rates", International Conference on Machine Learning, pp. 5200-5209, 2019.

[24]. Online available at https://data.mendeley.com/drafts/tmtsxh828p

[25]. Online available at https://data.mendeley.com/datasets/bk8s9dfsys/1

[26]. Online available at https://www.kaggle.com/datasets/hojjatk/mnist-dataset

Prof. Harish Kunder is an esteemed researcher specializing in Machine Learning and Data Science. Presently, he holds the position of Associate Professor and serves as the Head of the Department of Artificial Intelligence & Machine Learning and Department of Data Science at Alva's Institute of Engineering and Technology. His expertise spans various domains including Data Science, Artificial Intelligence, Machine Learning, Deep Learning, Transfer Learning, and Computer Vision. With a prolific career, he has contributed significantly to both national and international scholarly publications. He can be reached at researchkunder@gmail.com

Dr. Manjunath Kotari is currently working as a Professor and Head, of the Department of Computer Science and Engineering at Alva's Institute of Engineering and Technology. He received his B.E., M.Tech, and Ph.D. in Computer Science and Engineering from Visvesvaraya Technological University Belagavi. He holds more than 23 years of academic and Industry experience. His area of interest includes Computer Networks, Security, Cloud Computing, Artificial Intelligence & Machine Learning, and Deep Learning. He can be reached at mkotari@gmail.com.