Open Engineering Inc. - engrXiv (Engineering Archive)

Survey of Loss Landscape Surfaces: Theory, Applications and Algorithms

Aldo Taranto[1], Ron Addie[2]

[1] *Aldo Taranto, ORCID: 0000-0001-6763-4997*
Aldo.Taranto@anu.edu.au
School of Computing
Australian National University
Canberra, 2601, Australian Capital Teritory, AUSTRALIA

[2] *Ron Addie, ORCID: 0000-0002-6664-8462*
Ron.Addie@unisq.edu.au
School of Mathematics, Physics and Computing
University of Southern Queensland
Toowoomba, 4350, Queensland, AUSTRALIA

Abstract

This paper presents a survey of loss landscape surface (LLS) research in machine learning optimization. A three-stage filtering methodology was applied to over 300 research papers to select mathematically rigorous studies, yielding 50 seminal papers with analysis of the top 10 key contributions. The survey provides chronological analysis spanning 1951-2024, showing evolution from 4 foundational papers (2007) to 21 active contributions (2018-2021). A taxonomical framework categorizes 32 stochastic gradient descent (SGD) algorithms from classical SGD (1951) to recent variants. Analysis reveals six research themes: visualization techniques, minima-generalization relationships, step size selection, SGD extensions, batch size effects, and adaptive learning rates. Research questions are categorized into resolved issues (flat minima correlation), ongoing debates (batch size selection), and emerging challenges (transformer landscapes). Results demonstrate that despite algorithmic proliferation, only limited fundamentally novel continuous LLS algorithms exist. The survey identifies theoretical-practical gaps and provides a standardized framework for key algorithms. Contributions include tables documenting research chronology and algorithm evolution, providing foundation for future high-dimensional optimization research.

Keywords and Phrases: Loss landscape surface, Gradient descent, Stochastic optimization, Machine learning.

ACM Classification: Primary: F. Secondary: F2, Analysis of algorithms and problem complexity.

1 Introduction

In the realm of computer science, the evolution from Artificial Intelligence (AI) to Machine Learning (ML) and subsequently to Deep Learning (DL) has been transformative. Despite their inherent complexity, these paradigms fundamentally boil down to solving optimization problems over a space of possible values -a domain known as the loss landscape surface (LLS) [101].

Definition 1.1. (Loss Landscape Surface - Informal): From the empirical risk minimization (ERM) [175] principle in statistical learning theory (SLT) [121], a model can have up to $n \in \mathbb{N}$ features or variables[1]. Let w denote the weights $w_1, \cdots, w_n$ for a model, and define $L(w)$ to be the total loss (across the entire training set). Then, the optimization problem is to find w that minimizes $L(w)$. The term "loss surface" or "loss landscape" refers to the graph of this function L and since L is n-Dimensional, we obtain a graph or surface in some n-Dimensional space $\mathbb{R}^n$. ■

This definition serves well to introduce key concepts of LLS, but does not quite provide a comprehensive understanding. Due to the complexity of LLS, a more detailed definition is given in full in Definition 1.2. Our survey paper emerges from a critical need: to review and consolidate the vast body of LLS research, of which we cite 180+ papers. While the first survey paper on LLS, authored by Sun *et al.* [166], laid essential groundwork, our investigation delves deeper into the mathematical rigor and comprehensiveness of modern LLS research. We address pivotal questions:

1. Which contemporary LLS research papers provide the necessary depth to support both theoretical and applied advancements? We seek mathematically robust contributions that transcend mere exploration.
2. How do LLSs emerge from AI, ML, and DL? We suggest a classification model for these intricate surfaces. Furthermore, we explore extensions to stochastic gradient descent (SGD), a cornerstone of optimization algorithms.
3. Why is LLS research so indispensable across diverse fields? Its impact reverberates far beyond the confines of computer science.

This paper presents a comprehensive survey of optimization algorithms for high-dimensional loss landscape surfaces (LLS), focusing on stochastic gradient descent (SGD) variants and non-gradient approaches. Rather than directly investigating optimization issues, this work systematically reviews existing research to identify key papers that can facilitate future algorithmic developments.

Our survey methodology employed a three-stage filtering process applied to over 300 research papers identified through Google Scholar: general screening for relevant optimization literature, LLS-specific research selection, and SGD-focused studies identification. These filters incorporated

[1] **(Gain Surface, Gain Landscape Surface)**. In mathematical optimization and decision theory, a loss function or cost function (sometimes also called an error function) [138] is a function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event. An optimization problem seeks to minimize a loss function. An objective function is either a loss function or its opposite (in specific domains, variously called a reward function, a profit function, a utility function, a fitness function, etc.), in which case it is to be maximized.

Survey Evaluation Criteria (SEC) and Survey Selection Criteria (SSC) designed to identify papers with the greatest potential to drive meaningful future research in high-dimensional optimization.

Papers were excluded not due to quality concerns, but because they failed to meet our specific objectives. For instance, studies focusing on narrow applications without sufficient depth in optimization theory were filtered out. The selected papers are analyzed through multiple analytical lenses to derive insights that inform the development of novel optimization algorithms for machine learning applications.

Before surveying existing approaches, we first establish the key challenges identified across the literature that make high-dimensional optimization a computationally challenging field.

Optimization, a sprawling field rooted in mathematics, spans convex and nonconvex problems, discrete and continuous domains, and smooth and nonsmooth functions. However, classical mathematical techniques often fall short when faced with NP-hard challenges. Enter the narrower domain of Data Science, Applied Probability, and Statistics, which grapple with these optimization problems using more stochastic techniques. Within this context, computer science plays a pivotal role. Algorithms designed to "learn" over LLSs are inherently heuristic and metaheuristic. While LLSs predominantly arise from Artificial Neural Networks (ANNs), other areas—such as Data Science—also traverse these challenging landscapes. LLSs are complex due to the following challenges.

1. **High Dimensional:** Computer science applications routinely deal with LLSs of staggering dimensions—often exceeding tens of thousands. These high-dimensional LLSs defy human visualization. For instance, ChatGPT operates with over 150 billion variables, resulting in a 150-billion-dimensional LLS: $\{(x_1, y_1), \cdots, (x_n, y_n)\} \in \mathbb{R}^d \times y$, where, d represents the dimension of each input sample, and n denotes the number of training data points.
2. **Nonconvex:** The high dimensionality implies that LLSs are inherently nonconvex. Even a simple 3-dimensional sphere becomes increasingly intricate as dimensions multiply (think of an n-dimensional sphere [158]). Expressing typical LLSs as straightforward functions—composed of polynomials, exponentials, or trigonometric terms—remains elusive. In fact, determining the convexity of an arbitrary polynomial function is an NP-hard problem [3].
3. **Nonsmooth:** Data points often form a discrete LLS or mesh. This nonsmooth surface rules out certain techniques like elementary differential calculus. Smoothing methods, such as skip connections, can mitigate this issue (see Figure 4).
4. **Missing Values:** Data from computer science applications may have missing or NULL data points. These 'holes' or 'singularities' in the LLS can hinder simple approaches like gradient descent (GD). GD may get stuck in these regions.
5. **Dynamic and Volatile:** LLSs are dynamic and can change significantly during optimization, such as with streaming or real-time data applications. The time taken to compute an optimization may render previous results obsolete. This dynamism reinforces the NP-hard nature of the problem.

Expressing LLSs mathematically is challenging as they cannot be captured by simple or even

complex formulas. Finding an appropriate expression is computationally expensive and can be futile due to the dynamic nature of LLS. Our survey approach considers theory and practice, shedding light on the profound implications of LLS exploration. As we navigate this complex terrain, we uncover new avenues for research and underscore the enduring value of studying LLSs.

Regarding optimization algorithms, we delve further into gradient descent algorithms. Classical Gradient Descent (CGD), (1.1), like Newtonian descent, calculates gradients across *all* dimensions in each iteration. However, they assume specific forms for the LLS function, which may not hold in practice. On the other hand, Stochastic Gradient Descent (SGD) is more efficient. It randomly chooses a *smaller subset* of the dimensions and computes the gradient, resulting in faster convergence (time complexity $\mathcal{O}(n)$) compared to classical gradient descent $\mathcal{O}(n^d)$).

These concepts can be summarised and captured visually in Figure 1.

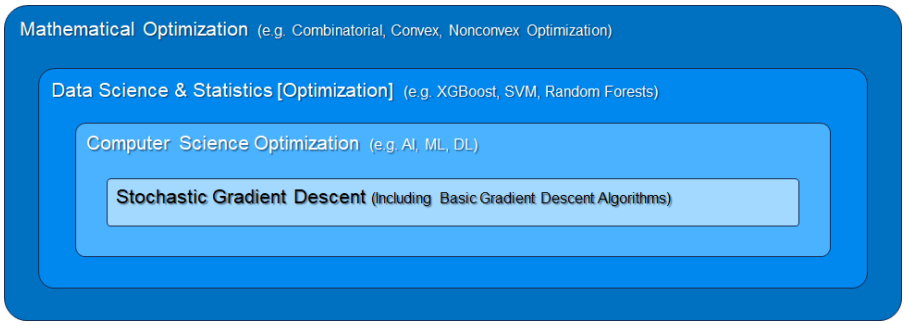


Figure 1: Simplified Levels of Abstraction in Optimization Algorithms
While computer science and mathematics share significant commonalities or overlaps, our survey focuses on the intricate interplay between loss landscape surfaces (LLS) and stochastic gradient descent (SGD). This diagram focusses on the fact that there is a hierarchy between these fields and how SGD is related to these.

1.1 What are Loss Landscape Surfaces?

The term 'loss landscape surfaces' is also known by, but not limited to, the following terms; '(static) fitness landscapes', 'global landscapes', 'loss surfaces', 'optimization landscapes', 'energy landscapes', 'potential-energy surfaces' and 'weight space' [53].

Definition 1.2. (Loss Landscape Surface - Formal) [102]: Given a model parameterized by weights $\theta \in \mathbb{R}^d$, and a loss function $\mathcal{L}(\theta)$, the loss landscape is defined as the mapping,

$$\mathbb{R}^d \to \mathbb{R}, \quad \mathcal{L}(\theta) \to \theta,$$

where,

- θ are the model parameters (e.g., weights in a neural network),
- $\mathcal{L}(\theta)$is typically the empirical risk or expected loss over the data distribution:

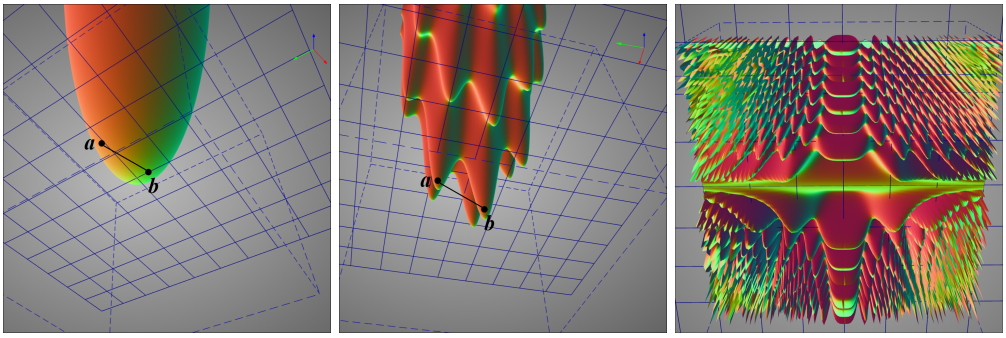
$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\ell\big(f(x;\theta), y\big)\Big]$$

or in practice,

$$\hat{\mathcal{L}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell\big(f(x_i;\theta), y_i\big).$$

The surfaces' geometry (e.g., local minima, saddle points, flatness, sharpness) directly impacts optimization and generalization in deep learning. ■

The function $\mathcal{L} : \mathbb{R} \to \mathbb{R}^n$ is 'embedded' in $\mathbb{R}^n$ and it suffices to say that the function may require additional dimensions $m \in \mathbb{N}$, to be ultimately embedded in $\mathbb{R}^{n\times m}$. To illustrate this definition, a convex proxy and some simple nonconvex proxies for LLSs are shown in Figure 2.

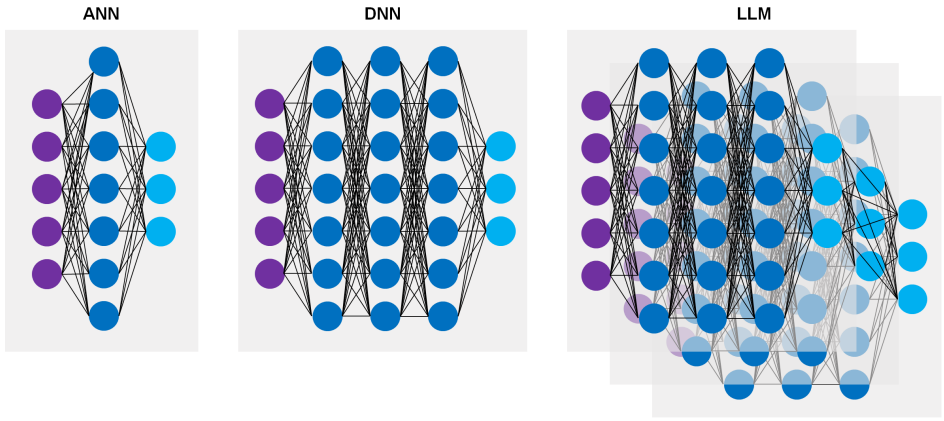


(a). **(b).** **(c).**

Figure 2: Optimisation Complexity of Convex and Nonconvex Surfaces
(a). $z = x^2 + y^2$: Only one global minimum. This is a convex surface.
(b). $z = x^2 + y^2 + \sin(5x) + \sin(5y)$: Multiple local minima, one global minimum. At a high level perspective (globally)This is either a weakly convex surface globally, or a nonconvex surface locally.
(c). $z = \sin(5x^2)\cos(5y^2)$: Multiple local minima, multiple global minima. This is a nonconvex surface locally and globally.

Figure 2(c) shows that if we were to have a nonconvex function, then there may be multiple local minima and either zero or multiple global minima. Under such a complex search space, an SGD algorithm would either never converge, run out of maximum allowed iterations, or converge onto one of many possible invalid local minima (or maxima) solutions.

Although these mathematical surfaces are provided solely for illustrative purposes to elucidate the complexity of LLS, such complexity may arise in a variety of contexts. These range from classical optimization problems -such as determining the optimal allocation of military resources across multiple interdependent dimensions (e.g., troop deployment, ammunition supply, reinforcement scheduling, and medical support) -to a broad spectrum of artificial ANN architectures, as depicted in Figure 3.



(a). **(b).** **(c).**

Figure 3: Increased Complexity Leads to Greater LLS Nonconvexity
(a). Simple ANN architecture with only one hidden (or middle) layer of neurons, leading to a relatively simple LLS.
(b). A more complex ANN, a dense neural network (DNN) having three hidden layers, leading to a more complex, nonconvex LLS.
(c). An even more complex ANN, a stylized depiction of a large language model (LLM), leading to very complex, nonconvex LLS.

Figure 3 shows that the main sources of the highly complex nonconvex LLS arise from LLM, which are essentially combinations of numerous DNN components, as seen for example in the transformer architecture. This not only involves the use of numerous hidden layers, but also numerous neurons in each layer. The seminal work of [102] supports this view, finding that,

"network architecture has a dramatic effect on the loss landscape. Shallow networks have smooth landscapes populated by wide, convex regions. However, as networks become deeper, landscapes spontaneously become 'chaotic' and highly non-convex, leading to poor training behavior".

Additional supporting evidence in the research literature consistently acknowledges that training neural networks requires minimizing a high-dimensional non-convex loss function and that DNNs are fundamentally solving extremely high dimensional problems with extremely non-convex LLS.

Having stated this, it is imparitive that we in this survey point out that there are some opposing edge cases that make this general rule of thumb more subbtle. Some work shows that "neural networks with a single hidden layer and non-decreasing positively homogeneous activation functions can lead to a convex optimization problem" even in high dimensions under specific conditions [8].

Further insights into the subtleties and nuances of LLS are articulated in the YouTube presentation, *The Misconception that Almost Stopped AI* [178], which provides both discussion and visualization

of key phenomena:

- The so-called "wormhole" effect is identified as a visualization artefact -when traversing high-dimensional space, a two-dimensional slice can change dramatically, giving the appearance that improvements emerge "out of nowhere".
- In high-dimensional settings, local minima become less problematic, as becoming trapped requires being confined in every dimension simultaneously -an increasingly improbable occurrence as dimensionality grows.
- As dimensionality increases, visualizations represent "an increasingly distant shadow of the model's true learning process", underscoring their limitations in accurately depicting underlying dynamics.

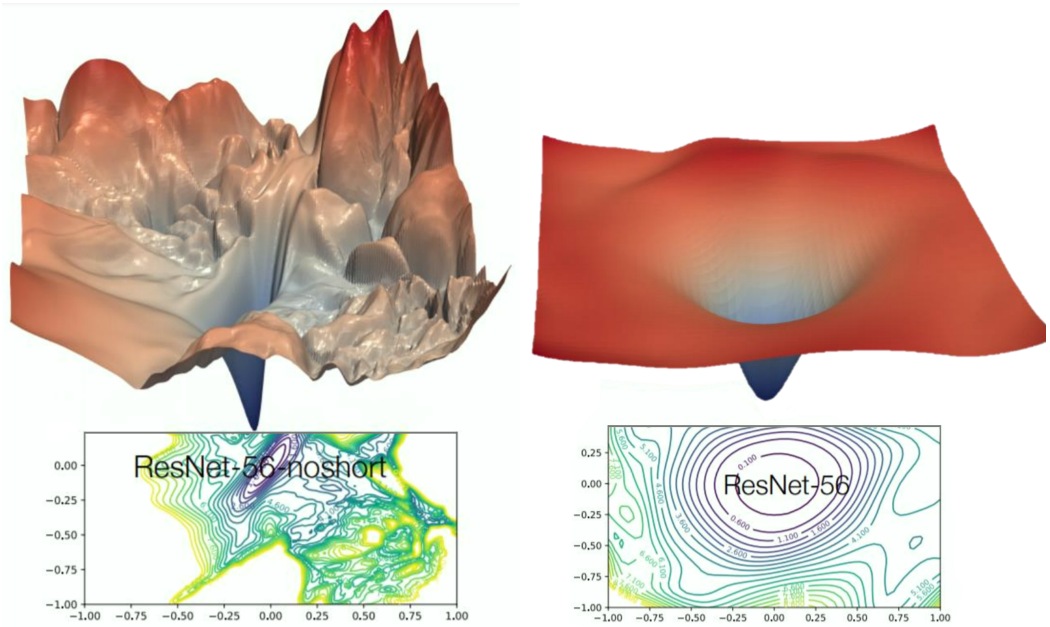
These observations are consistent with and supported by prior research, including [39] and [33]. Turning to a representative LLS arising from ANNs (in particular, ResNet-56 [101]), we can see the true non-stylized complexity and nature of LLS, as shown in Figure 4.

Figure 4 shows that a SGD algorithm's path towards the global minimum can either take an unacceptably high amount of time to reach the global minimum, or worse, can be blocked (i.e. never converge), due to the complexity imposed by the hills, valleys and nonsmoothness on the LLS. Such LLS can be smoothed out by the process of 'skip connections', to dramatically speed up the optimization process, even though there is a cost to undertake this smoothing process. This has been illustrated in Figure 4, which shows the LLS arising from the ResNet-56 algorithm [101].

Remark 1. As mentioned above, LLS are not exclusive to ANNs, despite the fact that LLSs appear predominantly in ANN-related research. In mathematical optimization and decision theory, a loss function -also known as a cost function or objective function -is precisely the scalar-valued function that an optimization problem seeks to minimize or maximize. The entirety of the optimization process can therefore be conceptualized as navigating the "surface" of this function across the decision-variable space, i.e. the LLS [74]. Non-neural examples that generate LLSs include Genetic algorithms (GAs), Support vector machines (SVMs), Decision trees, Linear regression, XGBoost, Random Forests, and many other modelling approaches [149]. ■

The concept of the "curse of dimensionality" was introduced in 1957 [16]. This phenomenon arises from the exponential increase in volume when adding extra dimensions to Euclidean space, and this entails:

- **Error Amplification:** As dimensions increase, so does the error. Algorithms become harder to design and often exhibit exponential running times. While higher dimensions theoretically allow more information storage (ChatGPT itself operates in a vast 150+ billion-dimensional space), there's a cost to optimization algorithms in such spaces. Noise and redundancy become more likely in real-world data.
- **Symmetry's Trap:** Symmetric nonconvex landscapes pose a challenge. Suboptimal solutions can trap algorithms, hindering generalization. Visualizing these high-dimensional spaces remains elusive for humans.



(a). Without skip connections **(b). With skip connections**

Figure 4: Examples of Loss Landscape Surface Arising from ResNet-56
Source: [101]. ResNet-56 is an ANN and the resulting LLS is shown in,
(a). without any smoothing, the resulting contour map is more chaotic and difficult to descend to global minimum,
(b). with smoothing via skip connections, the resulting contour map is more consistent and easier to descend to global minimum.

1.2 What are Gradient Descent Algorithms?

Gradient descent algorithms are optimization techniques commonly used in ML to minimize a cost function. They involve the following components.

- **Objective:** The goal of gradient descent algorithms is to find the optimal set of parameters for a model by iteratively adjusting these parameters.
- **Process:** Start with an initial guess for the parameters. Calculate the gradient (or approximate gradient) of the cost function with respect to these parameters. Update the parameters in the opposite direction of the gradient (i.e., toward the local minimum). Repeat the process until convergence.
- **Intuition:** Imagine walking down a hill. The steepest descent direction corresponds to the negative gradient, which guides us toward the lowest point (local minimum) of the cost function.
- **Applications:** Gradient descent is widely used for training ML models, including ANNs.

(Classical) Gradient Descent

Gradient descent, classical gradient descent (CGD) is based on the observation that if the multi-variable function $f(\mathbf{x})$ is defined, differentiable, and convex in a neighborhood of a point $\mathbf{a}$, as shown in Figure 2(a), then $f(\mathbf{x})$ decreases fastest if one moves from $\mathbf{a}$ in the direction of the negative gradient of f at $\mathbf{a}$, that is $-\nabla f(\mathbf{a})$, towards the global minimum at point $\mathbf{b}$. At iteration x_k, the next iteration x_{k+1} can be expressed as,

$$x_{k+1} = x_n - \eta_k \nabla f(x) = x_n - \eta_k \frac{1}{n} \nabla f_i(x_k), \tag{1.1}$$

where $f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x)$ and η_k is the k-th step size. A generic implementation of CGD is shown in Algorithm 1.

Algorithm 1 (Classical) Gradient Descent (CGD) Algorithm

INPUT:
$i = 1$, $j = 0$. Set ϵ as the limit of convergence
OUTPUT:
for $(i = 1, \cdots, m)$ **do**
  for $(j = 0, \cdots, n)$ **do**
    while $(\mid \omega_{j+1} - \omega_j \mid < \epsilon)$ **do**
      $\omega_{j+1} := \omega_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\omega(x^{(i)}) - y^{(i)}\right) x_j^{(i)}$

Stochastic Gradient Descent

CGD is expensive to evaluate because at each iteration k, all n^2 data points need to be calculated. SGD on the other hand is far more performant, because at each iteration k, only n data points need to be calculated. The term 'stochastic' does not mean that these points are chosen at random, creating a "random walk of descent", but rather each iteration of SGD computes the gradient on the basis of one randomly chosen partition of the dataset which was shuffled, instead of using the whole part of the observations. For SGD, at iteration x_k, the next iteration x_{k+1} can be expressed as,

$$x_{k+1} = x_k - \eta_k \nabla f_{i(k)}(x_k), \tag{1.2}$$

where $i(k) \in \{1, 2, \cdots, n\}$. A generic implementation of SGD is shown in Algorithm 2.

2 Methodology

A large sample of 300+ research papers were examined as part of this research paper's survey process. Google Scholar was used as the primary mechanism to identify popular academic research

Algorithm 2 Stochastic Gradient Descent (SGD) Algorithm

INPUT:
$i = 1$, $j = 0$. Set ϵ as the limit of convergence
OUTPUT:
for $(i = 1, \cdots, m)$ **do**
  for $(j = 0, \cdots, n)$ **do**
    while $(\mid \omega_{j+1} - \omega_j \mid < \epsilon)$ **do**
      $\omega_{j+1} := \omega_j - \alpha(h_\omega(x^{(i)}) - y^{(i)})x_j^{(i)}$

papers. Search terms included, but were not limited to (Neural Networks, Deep Learning, Optimization, Gradient/Descent, Loss Landscape, Stochastic, Training, Learning, Convex/Non-convex, Networks, Algorithms, Loss and Minimization). The main objective of this survey paper is to find key papers that facilitate new research into enhanced optimization algorithms on large LLS. To meet this objective, a three-stage filtering process was applied to the universe of potential research papers, as depicted in Figure 5.

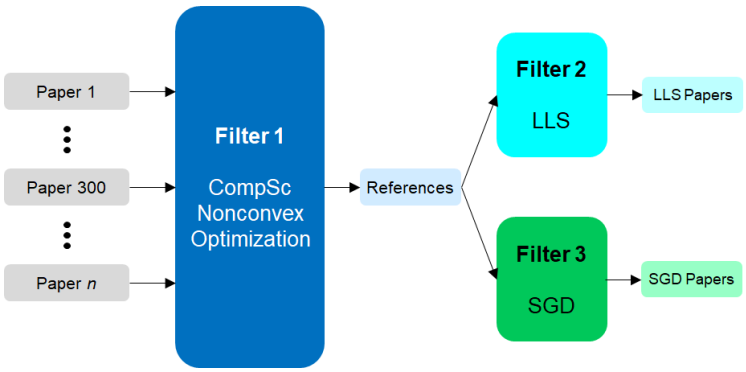


Figure 5: Research Paper Filtration Process

Our three-stage filtering process comprises,
Filter 1 (i.e. Table 5) which generates the list of candidates for the Literature Review section and are listed in the References section,
Filter 2 (i.e. Table 6) which lists the papers on SGD in Table 2,
Filter 3 (i.e. Table 7) which lists the papers on LLS in Table 3.
Note: There is an overlap between Filter 2 and Filter 3, because SGD techniques and algorithms traverse LLS.

Figure 5 shows that three filters were created in the form of,

- **Filter 1:** (i.e. Table 5) which generates the full list of References (Section 5) of research papers that met the high-level initial screening criteria.
- **Filter 2:** (i.e. Table 6) which generated the list of LLS research in Table 2.
- **Filter 3:** (i.e. Table 7) which generated the list of SGD research in Table 3.

In particular, these filters comprised a Survey evaluation criteria (SEC) and two Survey selection criteria (SSC). These criteria do not imply that the rejected papers were of low value, but that they

didn't meet our objectives, to find the key LLS and SGD papers that can drive new meaningful research in this field. Various SGD papers have been excluded from the analysis, even though they met the SEC but failed to meet the SSC. For example, if a paper did not delve enough into LLS nor SGD, and was focussed on specific subsets of ANNs such as Convolutional Neural Networks (CNN) with specific applications of computer vision of manufacturing assembly lines, then they were filtered out of our survey. In the Results & Discussion (Section 3), we dissect these research papers through different lenses to derive additional insights.

3 Results & Discussion

3.1 Historical Background

The papers are summarised here for the purposes of capturing the key historical developments of each decade of research.

1951–1960
Stochastic gradient descent used in the back propagation was first described by Robbins and Monro in 'A Stochastic Approximation Method' [141], and later Kiefer and Wolfowitz introduced the machine learning variant, Stochastic Gradient Descent, in their paper 'Stochastic Estimation of the Maximum of a Regression Function' [93], which is more recognizable to computer scientists.

1961–1970
In this decade, advances were indirectly made through various numerical methods research, including Runge-Kutta integration processes [26] and convergence of iteration methods [137], [25].

1971–1980
Further advances were made in ANN in relation to node complexity [12], [19] and improvements in convergence rates in convex programming [122].

1981–1990
This decade saw the transitioning of nonconvex optimization from purely a mathematics domain, to a more numerical computing domain, in sub-fields such as computer aided geometric design, mathematical software, mathematical programming, and operations research. The techniques derived included Genetic search [72], Proximal subgradients and augmented Lagrangians [142], matrix triangulations techniques [174], and nonlinear programming [49], [134]. This decade also had advances in LLS visualisation from parametric surface/surface intersection [14], together with the generation of continuous surfaces from discrete data points [170].

1991–2000

Stochastic gradient descent progressed as an improvement on gradient descent in an ANN context [21] together with generalization to infinite networks [179]. Further developments included enhancements to multiobjective optimization through the use of evolutionary algorithms [50], leveraging flat minima [79], showing how a simple weight decay can improve generalization [97].

2001-2010

LLS is then used to analysize improvements for solutions to constraint satisfaction (i.e. optimization) problems in [98]. Linkages with LLS and other areas are made including learning with kernels, SVMs and regularization [154]. The visualization of learning in multilayer perceptron networks using principal component analysis (PCA) helps characterise the resulting LLS [57]. Similarly, random matrix calculation of LLS complexity exposes the symmetry breaking conditions, which is also a form of LLS classification [56]. Enhancements to SGD were made by using distance-based classification via Lipschitz functions [177]. An interesting development is the reverse of conventional ANN research, where instead of LLSs arising from ANNs as is generally the case, [15] examines the generalized ANN representation of high-Dimensional LLSs. Further insights are obtained by examining the statistics of critical points of Gaussian fields on large-dimensional spaces [24], together with understanding the difficulty of training deep feedforward neural networks [66]. Additional insights into LLS are found via matrix analysis, in particular deep learning via Hessian-free optimization [115].

2011-2020

This decade has an explosion in SGD algorithm research [22], [94], RMSprop [173], Adaptive subgradient [44], Entropy-SGD [28], Phasemax [67], linear autoencoders [131]. Theoretical explorations were made on high dimensional LLS in [148], [30], [82], [108], [151], [45], [165], [2], [5], [59], together with the references therein. They show various similarities between LLSs with Boltzmann machines and spin systems (spin glasses) of statistical mechanics. A number of such papers were of a theoretical nature that generalise the characteristics of LLSs [13], [180], [88].

There are many papers published on deeper aspects of LLS in relation to ANNs [32], [152], [34], [68], [164], [112], [105], [181], [99], [166], [101]. Advances in overcomming the challenges of nonconvexity were made by using tensor methods [85] and by reducing internal covariate shift [83]. Mean field theory (and analysis) of ANNs also was advanced theoretically [156], [155], [127], [117], [9], [77], [118]. Progress was made in identifying and handling saddle points in high-dimensional nonconvex surfaces [38], [4], together with the importance of optimal momentum settings [167].

Numerous papers were derived that showed that the training of ANNs could be improved without necessarily over-training or over-fitting the data [81], [89], [168], [71], [62], [111], [60]. These papers on ANNs examine the relationship between the neural structure and the corresponding complexity of the LLS [162], [171]. These revolve around the depth and width of the LLS [116], [128], [136], [7]

where the depth of the LLS creates no bad local minima [113], [65]. Many other papers examined the matrix analysis that arises from various LLS configurations, including the use of Jacobians [132], Gaussian process behavior, tangent kernels [183], [51], Hessian eigenvalue density [64] and jamming transition [61]. Further progress was made by focusing on subsets of LLSs arising from ANNs; overparameterization [36], [147], [43], [124], [163], infinitely wide [6], dimensionality compression and expansion [139], [53] and finite versus infinite architectures [100], [42]. Mode connectivity, a recently introduced framework, empirically establishes the connectedness of minima by identifying a high-accuracy curve between two independently trained models [69], [58], [188].

Theoretical insights into the optimization landscape of over-parameterized shallow [160] and deep [52], [129] neural networks. [146] examined the quality of the initial basin in relation to the convergence rate on LLSs. Generalizations were obtained by examining the geometry [125], [182], [133] and topology [159], [54] of LLSs. Many other papers examined stochastic optinmization on convex [143] and nonconvex LLS [75], [135], [157], [186], [63]. Additional insights were obtained by examining sharp minima [40], bad local valleys [130] which is synonymous with bad/poor local minima [87], together with excluding bad local minima so that additional progress can be made in the simpler LLS [161], [73], [109].

2021-Present

The recent decade has continued it growth in interest of this important field, with an explosion of demand to find improved stochastic gradient descent algorithms. As expected, the vast majority of the research on LLS is ANN related [150], [169], [176], [110]. Computer vision and vision transformers are also related to ANNs, especially Convolutional neural networks (CNNs) and research helped visualise LLSs by examining local minima [55], [10] and leveraging flat local minima [27]. LLS smoothing was advanced by the use of techniques such as regularization [103], [47]. Further characterization of LLSs was achieved via non-negative matrix factorization [18]. This paper analyzes the loss landscape of non-negative matrix factorization (NMF) and shows that star-convexity properties—where gradients point toward the global minimum—hold with high probability, making optimization easier as model size increases.

3.2 Algorithm Papers Surveyed

After applying the Methodology to score numerous LLS research papers, a list of the viable research became apparent, as listed in the tables below. Table 1 lists the relatatively few survey papers on LLS and SGD in the last decade.

Table 1: Table of LLS and SGD Survey Papers

Year	Topic	Survey Paper Title	Authors
2021	LLS	Landscape and training regimes in deep learning	Geiger, Petrini & Wyart [60]
2021	LLS	LossPlot: A Better Way to Visualize Loss Landscapes	Bain, Tokarev *et al.* [11]
2021	LLS	Loss landscape analysis for deep learning: A survey	Liang, Bo & Sun [107]
2016	SGD	An overview of gradient descent optimization algorithms	Ruder [144]
2018	SGD	Stochastic gradient descent: Recent trends	Newton, Yousefian & Pasupathy [123]
2023	SGD	Recent advances in SGD in deep learning	Tian, Zhang & Zhang [172]

This survey paper compliments the existing surveys of Table 1 and then generates the chronological timeline of corresponding papers on LLS (Table 2) and on SGD (Table 3).

3.3 Filter 2: Identifying Key LLS Research Papers

Continuing our top down approach, the next set of results list the timeline of key LLS research papers (i.e. non-survey papers), as shown in Table 2.

Table 2: Timeline of Key LLS Research

This table focusses on the key research in LLS. By 'key', we mean here the research papers that satisfied Filter 3 of our three-stage survey scoring methodology. The research papers are listed chronologically.

Year	LLS References
2005	–
2006	–
2007	[98],[15],[24],[56]
2008	–
2009	–
2010	–
2011	–
2012	–
2013	–
2014	[148]
2015	[30],[34],[32]
2016	[82],[109]
2017	[54],[13],[180],[108],[128],[159],[160]
2018	[101],[69],[42],[58],[130],[36],[118],[143]
2019	[157],[52],[182],[61],[53]
2020	[131],[51],[166],[111]
2021	[10],[110],[18],[169],[176],[60]
2022	–
2023	[55],[103]
2024	–

Table 2 shows how LLS research is relatively new, has continued to gorow over time, and has become its own body of research distinct from similar surfaces studied formally in mathematical optimization.

3.4 Filter 3: Identifying Key SGD Research Papers

Continuing our top down approach, the next set of results is on SGD, which belong to the sub-class of Uncertainty / Probabilistic algorithms, as shown in Figure 6.

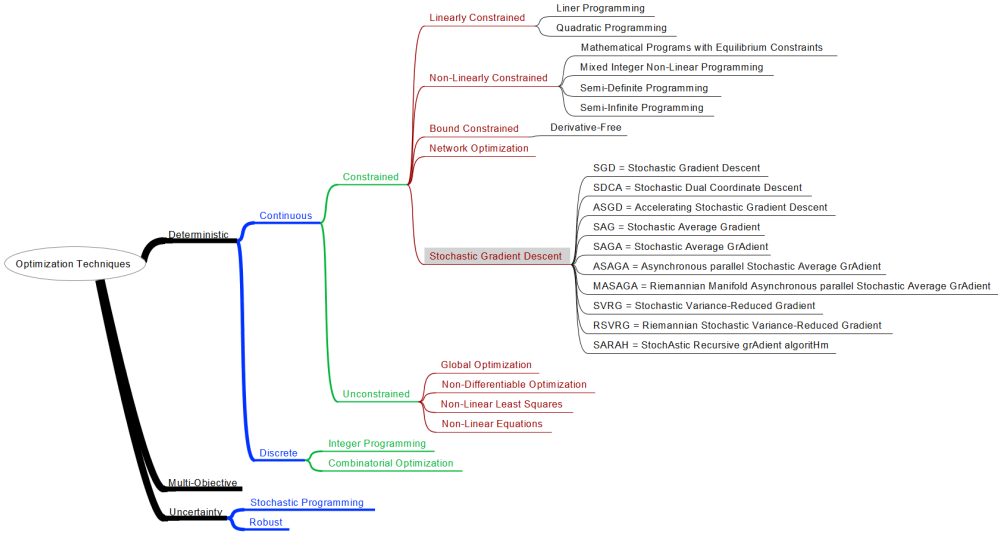


Figure 6: Taxonomy of SGD Algorithms

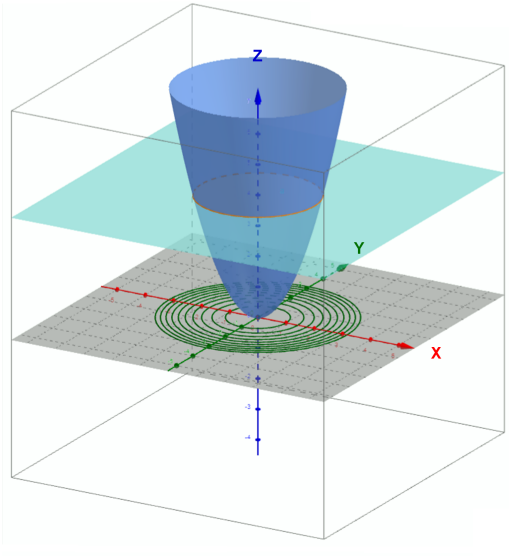
Figure 6 shows the overall taxonomy of various optimization approaches and lists 10 stochastic gradient descent techniques. A more comprehensive, although still not exhaustive, list of 32 key SGD algorithms is shown in Table 3.

Table 3 shows the evolution of SGD algorithm research over time. To examne ten of these most popular algorithms further, four 3D use-case proxies or surrogates of high-dimensional LLSs were identified of increasing complexity, to establish a benchmark of how well they traverse various surfaces. It is worthwhile noting that whilst LLSs arise from the data domain, such as an ANN classifying pictures of cats and dogs, and generating an 'illusive' and abstract LLS in the feature domain, in this paper we eliminate the data domain and go direct to the more simpler and concrete LLS proxies where we can better focus on and visualize the optimization algorithms' performance. These LLS are are shown in Figure 7.

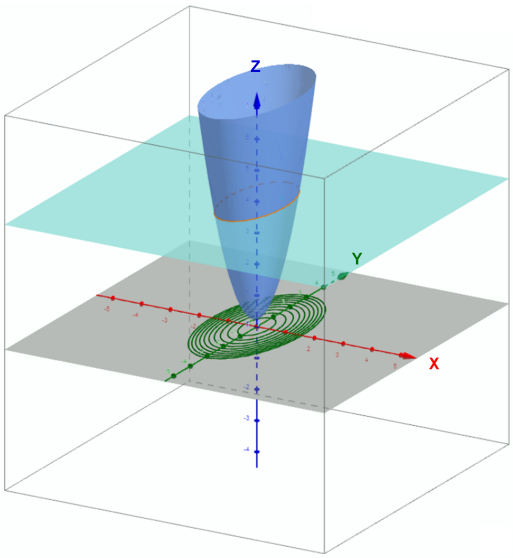
Table 3: Timeline of Key SGD Algorithms

This table focusses on the SGD algorithms and the timeline that they were introduced into the research community. Slight variations in the abbreviations for the algorithms can vary depending on the author, such as ISGD versus I-SGD.

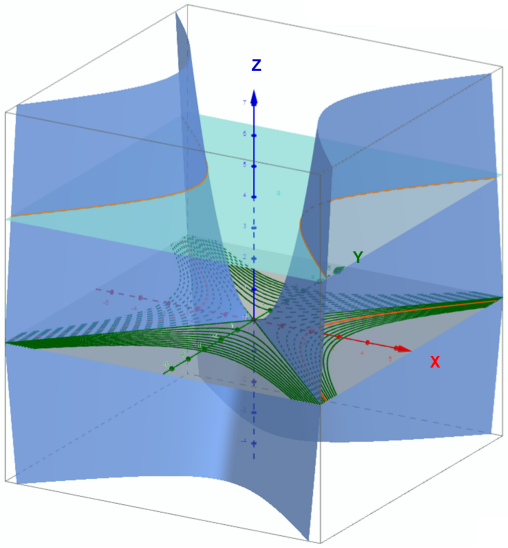
Index	Year	Algorithm	References	Contribution Summarized
1	1951	SGD	[141]	Stochastic Gradient Descent
2	1983	NAG	[122]	Nesterov Accelerated Gradient
3	2004	KSGD	[17]	Kalman-based Stochastic Gradient Descent
4	2007	AOGD	[76]	Adaptive Online Gradient Descent
5	2009	SGD-QN	[20]	Careful Quasi-Newton Stochastic Gradient Descent
6	2011	AdaGrad	[44]	AdaGrad
7	2012	AdaDelta	[184]	AdaDelta
8	2013	SVRG	[86]	Stochastic Variance-Reduced Gradient
9	2013	CSGD	[120]	Constrained Stochastic Gradient Descent
10	2014	SAGA	[1]	Stochastic Average GrAdient
11	2014	AdaMax	[94]	AdaMax
12	2014	MiniBatchSGD	[104]	Mini-Batch SGD
13	2015	SDCA	[70]	Stochastic Dual Coordinate Descent
14	2015	ADAM	[95]	ADAptive Moment estimation
15	2015	EASGD	[187]	Elastic Averaging SGD
16	2015	SDA	[70]	Stochastic Dual Ascent
17	2015	DSA	[119]	Decentralized Double Stochastic Averaging Gradient
18	2016	NADAM	[41]	Nesterov ADAM
19	2017	BGD	[92]	Batch Gradient Descent
20	2017	SAG	[153]	Stochastic Average Gradient
21	2017	SARAH	[126]	StochAstic Recursive grAdient algoritHm
22	2018	RMSprop	[78]	Root Mean Square Propagation
23	2018	FTRL	[114]	Follow the Regularized Leader
24	2018	ASGD	[84]	Accelerating Stochastic Gradient Descent
25	2018	AMSGrad	[140]	AMSGrad
26	2018	SPIDER	[48]	Stochastic Path-Integrated Differential EstimatoR
27	2019	ASAGA	[106]	Asynchronous parallel Stochastic Average GrAdient
28	2020	SGDM	[37]	SGD + Momentum
29	2023	SDCA	[185]	Stochastic Dual Coordinate Ascent
30	2023	SPSA	[91]	Simultaneous Perturbation Stochastic Approximation
31	2024	LLSGD	[35]	Locally Lipschitz Stochastic Gradient Descent
32	2024	GeoAdaLer	[46]	Geometric Insights into Adaptive SGD



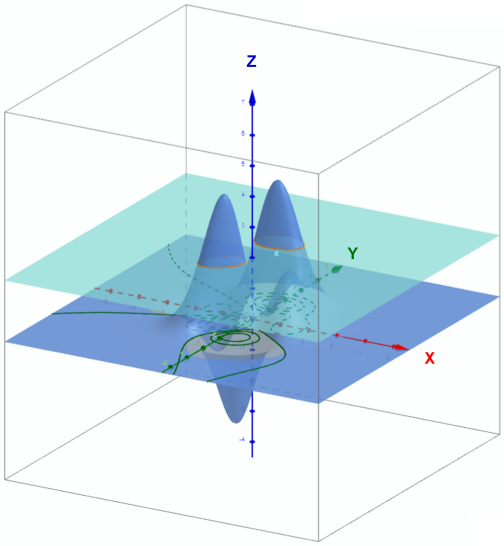
(a). Use Case I



(b). Use Case II



(c). Use Case III



(d). Use Case IV

Figure 7: Surface & Contour Plots of Four LLS

(a). Use Case I: $z = x^2 + y^2$.
(b). Use Case II: $z = 4x^2 + y^2$.
(c). Use Case III: $z = x^2 - y^2$.
(d). Use Case IV: $z{=}3(1-x)^2 \exp\left(-x^2-(y+1)^2\right) - 5(\frac{x}{5} - x^3 - y^5) \exp\left(-x^2-y^2\right) - \frac{1}{3}\exp\left(-(x+1)^2-y^2\right)$.

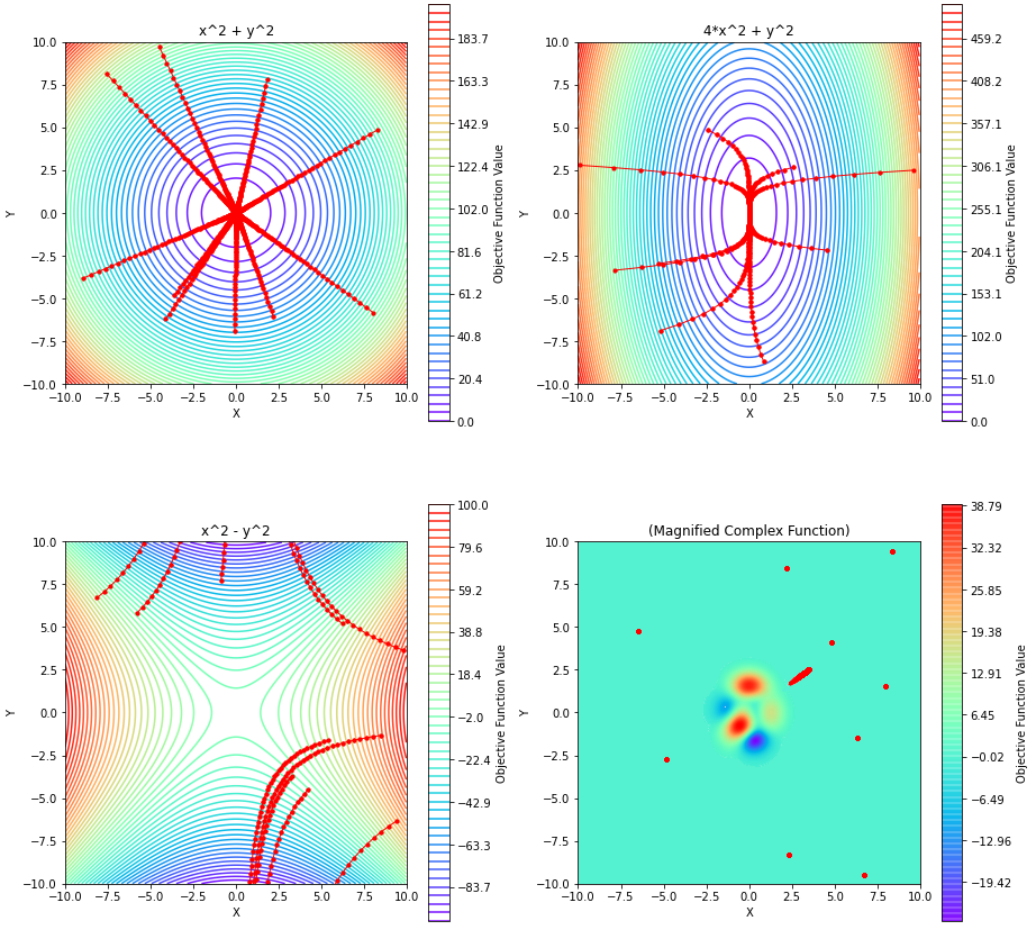


Figure 8: Contour Plots of Four LLS and SGD Simulations

Across our four chosen LLSs, we apply the default SGD algorithm with 1000 max iterations (or steps) and reinitialize it from 10 different starting points. The results of this experiment demonstrate how increasing levels of nonconvexity results in increasing challanges to SGD optimization algorithm.

(a). Use Case I: $z = x^2 + y^2$.

(b). Use Case II: $z = 4x^2 + y^2$.

(c). Use Case III: $z = x^2 - y^2$.

(d). Use Case IV: $z = 3(1-x)^2 \exp\left(-x^2-(y+1)^2\right) - 5\left(\frac{x}{5} - x^3 - y^5\right) \exp\left(-x^2-y^2\right) - \frac{1}{3}\exp\left(-(x+1)^2-y^2\right)$.

Figure 7 illustrates several key observations regarding LLS and their impact on optimization algorithms.

- **Use Case I (Full Symmetry): $z = x^2 + y^2$**
  When dealing with a convex surface that exhibits full symmetry, the starting point for the algorithm doesn't significantly affect convergence. As long as it begins at the same level plane, the algorithm will converge reliably.
- **Use Case II (Partial Symmetry): $z = 4x^2 + y^2$**
  In scenarios with lower symmetry, the initial starting point matters. Even though the surface itself is symmetrical, different starting points can lead to distinct convergence behavior.
- **Use Case III (Saddle Point): $z = x^2 - y^2$**
  Complexity increases when encountering saddle points. These points are not the true minima but can mislead the algorithm. If the algorithm veers toward or away from the saddle point, it may fail to converge.
- **Use Case IV (Multiple Peaks and Valleys):**
  $z=3(1-x)^2 \exp\left(-x^2-(y+1)^2\right)-5(\frac{x}{5}-x^3-y^5)\exp\left(-x^2-y^2\right)-\frac{1}{3}\exp\left(-(x+1)^2-y^2\right)$
  The complexity of Use Case IV, characterized by multiple peaks and valleys, challenges SGD algorithms. Less convex surfaces, particularly LLS, pose greater optimization difficulties.

Having selected the four use cases of Figure 7, it is instructive to visualise the four use case surfaces and also have the default SGD algorithm initiated at 10 different starting locations on the surface in question. This is because these SGD algorithms can miss either a local minimum and/or the global minimum, and so starting position is important. This forms a series of experiments comprising $10x4 = 40$ simulations of the SGD algorithm, which not only produces visual plots of the paths, but also statistical metrics for the paths, Table 4. The implementations of the various SGD algorithms and the corresponding numerous sample SGD paths are based on [145] and are shown in Figure 8.

Figure 8 shows how SGD descends on the various surfaces. We see that SGD easily veers off from finding the saddle point in Use Case III, because the surface descends infinitly, meaning that there is no such global minimum. SGD also fails the benchmark test in Cse Case IV, because of the flatness of the outer regions causing the algorithm to 'run out' of steps. That is to say, it stays in a linear trajectory, taking smaller and smaller steps, which does not adapt enough and terminates without having explored enough of the LLS.

The next step was to extend the use of just SGD so as to benchmark 10 popular SGD algorithms over these surfaces. Rather than have 1 algorithm starting at 10 points, we now consider 10 algorithms starting at 1000 points each. By now focussing less on displaying the surface and more on the learning rate (over the surface), and the results are shown in Figure 9.

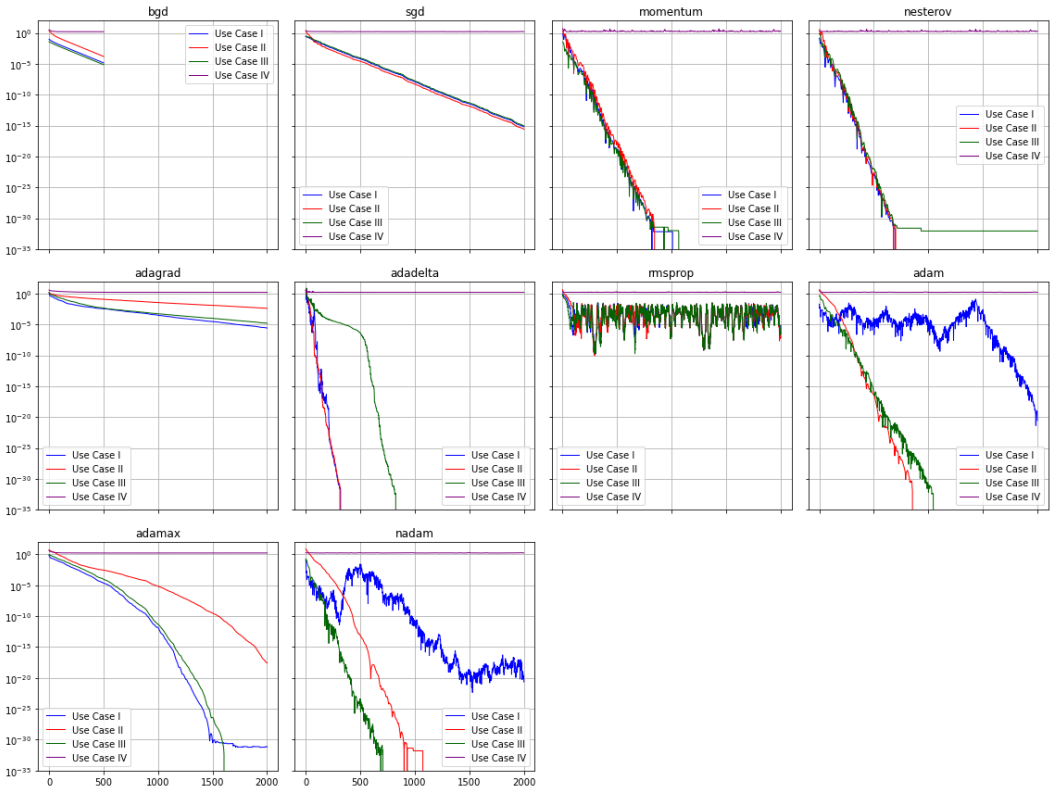


Figure 9: Benchmark Profile for 10 Popular SGD Algorithms

Using the SGD library from [145], we run the python code over our four use case surfaces. This generates the effective learning rates of the various algorithms over the given surfaces. It shows for example, that RMSprop did not perform very well in relation to the other algorithms, and also that the complex surface (Use case IV) was too complex for all SGD algorithms.

(a). Use Case I: $z = x^2 + y^2$.

(b). Use Case II: $z = 4x^2 + y^2$.

(c). Use Case III: $z = x^2 - y^2$.

(d). Use Case IV: $z{=}3(1{-}x)^2 \exp\left(-x^2-(y{+}1)^2\right) - 5(\frac{x}{5} - x^3 - y^5) \exp\left(-x^2-y^2\right) - \frac{1}{3}\exp\left(-(x{+}1)^2-y^2\right)$.

Figure 9 shows that as the complexity increases in each subsequent use case, the number of iterations for the SGD algorithm to converge on the global minimum or nearest local minimum increases dramatically. It also shows how quick an undergraduate text book example can quickly become highly complex and non-performant, the detailed statistics of which are shown in Table 4.

Table 4 shows how the SGD algorithms performed against each other over such use case LLSs. As the complexity grew, even the most robust SGD algorithms could not converge on the global minimum or even on the next nearest local minimum. The settings show that for each algorithm, its bespoke set of settings are required to make the algorithm perform at it's best. This aspect of model hyperparameter tuning is somewhat of an art, as it is dependent to a large extent on the LLS itself. The parameters are also sensitive to initial conditions, where if the learning rate η is either too large or too small, the algorithm will not converge within a reasonable time-frame.

Algorithm Performance Rankings

Top Performers (Most Robust):
Momentum and Nesterov: Consistently achieve perfect or near-perfect minima (-100.00) on convex surfaces (I, II, IV), with very similar statistical profiles AdaDelta: Excellent performance on all testable surfaces, particularly outstanding on Surface IV (-100.00 minimum).

Moderate Performers:
BGD and SGD: Identical performance profiles, reasonably good on most surfaces but don't reach global optima RMSProp: Good performance on Surface IV (-99.93) but weaker on simpler surfaces I and II.

Underperformers:
AdaGrad: Significantly poor performance across all surfaces (minimum only -6.05 to -27.11) Adam, Adamax, Nadam: Similar moderate performance, better than AdaGrad but substantially worse than momentum-based methods.

Surface-Specific Insights

Surface I & II (Simple Convex):
Momentum/Nesterov excel with perfect convergence. Most other algorithms show suboptimal performance. BGD/SGD perform identically as expected.

Surface III (Saddle Point):
All algorithms fail completely (all zeros) - this surface appears to trap all methods at the saddle point, highlighting a fundamental challenge in non-convex optimization.

Surface IV (Complex Multi-modal):
Shows greatest algorithm differentiation. AdaDelta and RMSProp perform exceptionally well. Reveals which algorithms can handle complex landscapes.

Key Observations

Table 4: Benchmark Statistics for 10 Popular SGD Algorithms

Benchmark statistics drill down deeper into the details of the relationship between the surfaces and the SGD algorithms. In column 3, S=Surface. Observation 1: Since SGD descends, the Maximum values are all 0. Observation 2: Since Surface III is curved downwards, the Minimum is never reached and so is 0.

ID	Algo	S	Min	Q1	Q3	Mode	Med	Mean	Var	SD	Skew	Kurt
1	BGD	I	-98.18	-95.04	-63.25	-98.18	-86.49	-75.45	649.60	25.49	1.27	0.58
2	BGD	II	-99.64	-99.01	-91.21	-99.64	-97.27	-90.08	314.71	17.74	2.87	8.33
3	BGD	III	0	0	0	0	0	0	0	0	0	0
4	BGD	IV	-47.86	-40.30	-17.72	-47.86	-30.59	-28.43	183.74	13.56	0.40	-1.01
5	SGD	I	-98.18	-95.04	-63.25	-98.18	-86.49	-75.45	649.60	25.49	1.27	0.58
6	SGD	II	-99.64	-99.01	-91.21	-99.64	-97.27	-90.08	314.71	17.74	2.87	8.33
7	SGD	III	0	0	0	0	0	0	0	0	0	0
8	SGD	IV	-47.86	-40.30	-17.72	-47.86	-30.59	-28.43	183.74	13.56	0.40	-1.01
9	Momentum	I	-100.00	-100.00	-100.00	-100.00	-100.00	-97.03	176.51	13.29	5.42	30.34
10	Momentum	II	-100.00	-100.00	-100.00	-100.00	-100.00	-98.53	85.08	9.22	8.26	72.37
11	Momentum	III	0	0	0	0	0	0	0	0	0	0
12	Momentum	IV	-93.57	-90.71	-71.77	-93.57	-85.34	-76.84	430.64	20.75	1.87	3.02
13	Nesterov	I	-100.00	-100.00	-100.00	-100.00	-100.00	-96.99	177.72	13.33	5.39	29.98
14	Nesterov	II	-100.00	-100.00	-100.00	-100.00	-100.00	-98.51	86.05	9.28	8.19	71.20
15	Nesterov	III	0	0	0	0	0	0	0	0	0	0
16	Nesterov	IV	-93.55	-90.69	-71.67	-93.55	-85.30	-76.79	431.56	20.77	1.87	3.00
17	AdaGrad	I	-6.05	-5.24	-2.99	-6.05	-4.27	-4.01	2.12	1.45	0.58	-0.57
18	AdaGrad	II	-6.05	-5.24	-2.99	-6.05	-4.27	-4.01	2.12	1.45	0.58	-0.57
19	AdaGrad	III	0	0	0	0	0	0	0	0	0	0
20	AdaGrad	IV	-27.11	-23.91	-14.38	-27.11	-19.92	-18.61	40.17	6.34	0.70	-0.36
21	RMSProp	I	-75.15	-61.25	-24.13	-75.15	-44.24	-42.16	464.03	21.54	0.23	-1.15
22	RMSProp	II	-75.15	-61.25	-24.13	-75.15	-44.24	-42.16	464.03	21.54	0.23	-1.15
23	RMSProp	III	0	0	0	0	0	0	0	0	0	0
24	RMSProp	IV	-99.93	-99.40	-76.86	-99.93	-95.77	-83.50	567.00	23.81	1.67	1.82
25	Adam	I	-68.68	-56.68	-22.85	-68.68	-41.57	-39.16	392.44	19.81	0.29	-1.11
26	Adam	II	-68.68	-56.68	-22.85	-68.68	-41.57	-39.16	392.44	19.81	0.29	-1.11
27	Adam	III	0	0	0	0	0	0	0	0	0	0
28	Adam	IV	-96.27	-93.63	-69.40	-96.27	-87.49	-77.25	534.04	23.11	1.56	1.58
29	AdaDelta	I	-100.00	-100.00	-75.97	-100.00	-97.98	-83.39	660.08	25.69	1.64	1.59
30	AdaDelta	II	-100.00	-100.00	-75.97	-100.00	-97.98	-83.39	660.08	25.69	1.64	1.59
31	AdaDelta	III	0	0	0	0	0	0	0	0	0	0
32	AdaDelta	IV	-100.00	-99.99	-97.72	-100.00	-99.87	-94.67	210.70	14.52	3.93	16.54
33	Adamax	I	-74.86	-60.63	-22.74	-74.86	-43.26	-41.17	482.20	21.96	0.23	-1.16
34	Adamax	II	-74.86	-60.63	-22.74	-74.86	-43.26	-41.17	482.20	21.96	0.23	-1.16
35	Adamax	III	0	0	0	0	0	0	0	0	0	0
36	Adamax	IV	-96.39	-93.33	-65.52	-96.39	-85.99	-75.38	594.53	24.38	1.43	1.15
37	Nadam	I	-68.66	-56.67	-22.84	-68.66	-41.55	-39.14	392.19	19.80	0.29	-1.11
38	Nadam	II	-68.66	-56.67	-22.84	-68.66	-41.55	-39.14	392.19	19.80	0.29	-1.11
39	Nadam	III	0	0	0	0	0	0	0	0	0	0
40	Nadam	IV	-96.25	-93.61	-69.32	-96.25	-87.44	-77.20	534.49	23.12	1.56	1.57

1. Momentum-based methods (Momentum, Nesterov) demonstrate superior robustness and convergence quality.
2. Adaptive learning rate methods show mixed results -AdaDelta excels while AdaGrad consistently underperforms.
3. Surface III exposes a critical limitation -all algorithms fail on saddle point landscapes.
4. Statistical distributions reveal algorithm behavior -momentum methods show high kurtosis, indicating consistent convergence to optima with occasional outliers.

For practical applications, Momentum or Nesterov variants appear most reliable, while AdaGrad should be avoided. The complete failure on Surface III suggests the need for saddle-point-aware optimization strategies.

4 Challenges and Future Directions

The main challenge facing continuous LLS research is the so-called 'exploitation versus exploration dilema'. Here, there is a trade off between determining the ideal/optimal solution (i.e. exploitation) versus ensuring that a sufficiently large search space is examined (i.e. exploration).

The study of loss landscapes and their relationship to optimization algorithms, particularly SGD algorithms, is an active area of research in machine learning. There are six key topics and recent findings.

1. **Visualizing Loss Landscapes:** Researchers have developed techniques to visualize the loss landscape of trained models. These visualizations help us understand the structure of the loss function in high-dimensional parameter spaces. The loss landscape provides insights into the model's generalizability and robustness. For instance, the shape of the landscape can predict whether a model will generalize well to unseen data. Visualization methods include dimensionality reduction techniques that map the parameter space to a two-dimensional surface. [102] develops visualization techniques for neural network loss landscapes using dimensionality reduction methods, showing how loss surface geometry affects generalization and trainability.
2. **Local Minima and Generalization:** Investigating the presence of local minima in the loss landscape is crucial. While deep learning models often have many local minima, not all of them lead to good generalization. Recent work has shown that the width of the minima at the top of the loss landscape correlates with generalization performance. Wider minima tend to lead to better generalization. [80] introduces the concept of "flat minima" in neural networks, showing that wider minima in the loss landscape correlate with better generalization performance.
3. **Optimal Step Size Selection:** Good choice in step size (also known as the learning rate) can have many downstream benefits including Convergence speed, Stability and robustness, Avoiding plateaus and saddle Points. [23] provides practical techniques for selecting optimal

step sizes in stochastic gradient descent, covering convergence speed, stability, and methods to avoid optimization plateaus.

4. **Extending SGD:** Understanding how SGD explores the loss landscape and converges to minima is essential. Some variants of SGD, such as Sharpness Aware Minimization (SAM), leverage local loss information to improve training. SAM smooths the loss landscape and leads to better test performance. [29] introduces Entropy-SGD, which biases gradient descent toward wide valleys in the loss landscape, improving generalization by seeking flatter minima similar to SAM approaches.
5. **Batch Size and Loss Landscape:** The choice of batch size during training affects the shape of the loss landscape. Small-batch training results in wider minima at the top of the landscape, which can lead to more generalizable models. Large-batch training may result in narrower minima, but it can also make optimization more challenging. [90] demonstrates that large-batch training leads to sharp minima with poor generalization, while small-batch training finds flatter minima that generalize better.
6. **Self-Adjusting Learning Rates:** Research explores self-adjusting learning rates for SGD variants. Techniques like Decentralized Parallel SGD (DPSGD) adaptively adjust learning rates based on the loss landscape. DPSGD often converges in cases where traditional SGD diverges. [96] demonstrates that large-batch training leads to sharp minima with poor generalization, while small-batch training finds flatter minima that generalize better.
7. **Smoothed Landscapes:** Some optimization algorithms operate on smoothed versions of the loss landscape. By designing specific loss landscape smoothing algorithms, researchers aim to improve optimization and generalization. [31] analyzes the energy landscape of deep networks and proposes methods for smoothing loss landscapes to improve optimization dynamics and achieve better generalization.

5 Conclusions

This paper presents the first comprehensive survey of continuous loss landscape surface (LLS) research, systematically analyzing over 300 research papers through a rigorous three-stage filtering methodology. Our investigation has identified the key research papers that are mathematically rigorous and sufficiently comprehensive to support future theoretical and applied developments in high-dimensional optimization. Our analysis reveals several critical themes that have emerged in the LLS literature: 1. Historical Evolution and Maturation: The chronological analysis (Tables 2-3) demonstrates that LLS research has evolved from mathematical curiosity in the 2000s to a distinct field with 32+ specialized SGD algorithms by 2024, indicating rapid algorithmic innovation but also potential fragmentation. 2. Visualization and Understanding: A central research question throughout the literature concerns how to effectively visualize and interpret high-dimensional loss landscapes. Our survey identifies key methodological advances from basic surface plotting to sophisticated dimensionality reduction techniques. 3. Generalization vs. Optimization Trade-offs: The literature consistently grapples with the fundamental question of whether optimization ease

correlates with generalization performance, particularly regarding flat vs. sharp minima. 4. Algorithmic Convergence in Non-convex Settings: Despite theoretical challenges, our survey reveals that practical SGD variants consistently find good solutions, raising important questions about why theory and practice diverge.

Resolved Questions: The literature has established that flat minima generally correlate with better generalization, and that loss landscape geometry significantly affects optimization dynamics. Ongoing Debates: Critical unresolved questions include optimal batch size selection, the relationship between landscape smoothness and robustness, and scalability of visualization techniques to modern deep architectures. Emerging Challenges: Recent work highlights new questions about loss landscape properties in transformer architectures and the role of overparameterization in landscape structure.

This survey addresses the fundamental question: "What are the key continuous LLS research papers that can guide future algorithmic development?" Our filtering methodology has identified a core set of influential papers while revealing that despite extensive research activity, only a limited number of truly novel continuous LLS algorithms exist. This finding suggests that the field may benefit from consolidation rather than further algorithmic proliferation. The taxonomical framework and chronological analysis provide researchers with a roadmap for understanding how LLS concepts interconnect across theoretical foundations, visualization techniques, and practical optimization methods. Our identification of research gaps, particularly in bridging theory-practice divides and addressing scalability challenges, offers clear directions for future investigation. Research Impact: This survey enables researchers to avoid redundant work, build upon established theoretical foundations, and focus efforts on the most promising algorithmic directions for advancing high-dimensional optimization in machine learning applications.

Data Availability

This is a review article that does not deal with any datasets. To access the datasets cited in this article, the readers are referred to the source articles' authors.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The first author was supported by an Australian Government Research Training Program (RTP) Scholarship.

References

[1] A. Defazio, F. B., and Lacoste-Julien, S. SAGA: A fast incremental gradient method with support for nonstrongly convex composite objectives. *Advances in neural information processing systems* (2014).

[2] Advani, M., Saxe, A., and Sompolinsky, H. 2020. *Neural Networks 132* (High-dimensional dynamics of generalization error in neural networks), 428–446. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7685244/pdf/main.pdf.

[3] Ahmadi, A., Olshevsky, A., Parrilo, P., and Tsitsiklis, J. Np-hardness of deciding convexity of quartic polynomials and related problems. *Mathematical programming 137* (2013), 453–476. https://arxiv.org/pdf/1012.1908.

[4] Anandkumar, A., and Ge, R. Efficient approaches for escaping higher order saddle points in non-convex optimization. *arXiv preprint arXiv:1602.05908* (2016), 1–21. https://arxiv.org/pdf/1602.05908.pdf.

[5] Ansuini, A., Laio, A., Macke, J., and Zoccolan, D. Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems* (2019), 6111–6122. arxiv.org/pdf/1905.12784.pdf.

[6] Arora, S., Du, S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. On exact computation with an infinitely wide neural net. *Advances in Neural Information Processing Systems 32* (2019), 8141–8150. https://proceedings.neurips.cc/paper_files/paper/2019/file/dbc4d84bfcfe2284ba11beffb853a8c4-Paper.pdf.

[7] Arora, S., Du, S., Li, Z., Salakhutdinov, R., Wang, R., and Yu, D. Harnessing the power of infinitely wide deep nets on small-data tasks. *arXiv preprint arXiv:1910.01663* (2019), 1–15. https://arxiv.org/pdf/1910.01663.pdf.

[8] Bach, F. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research 18*, 19 (2017), 1–53.

[9] Bach, F. Breaking the curse of dimensionality with convex neural networks. *J. Mach. Learn. Res. 18*, 1 (2017), 629–681. https://www.jmlr.org/papers/volume18/14-546/14-546.pdf.

[10] Bain, R. Visualizing the loss landscape of winning lottery tickets. *arXiv preprint arXiv:2112.08538* (2021), 1–7. https://arxiv.org/abs/2112.08538.

[11] Bain, R., Tokarev, M., Kothari, H., and Damineni, R. Lossplot: A better way to visualize loss landscapes. *arXiv preprint arXiv:2111.15133v1* (2021), 1–5. https://arxiv.org/pdf/2111.15133.

[12] Baldi, P., and Hornik, K. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks 2*, 1 (1989), 53–58. https://www.sciencedirect.com/science/article/pii/0893608089900142/pdf?md5=99ad18c4272a012f69dd63c51285c84a&pid=1-s2.0-0893608089900142-main.pdf.

[13] Ballard, A., Das, R., Martiniani, S., Mehta, D., Sagun, L., Stevenson, J., and Wales, D. Energy landscapes for machine learning. *Physical Chemistry Chemical Physics 19*, 20 (2017), 1–20. https://pubs.rsc.org/en/content/articlepdf/2017/cp/c7cp01108c/c7cp01108c.pdf.

[14] Barnhill, R., and Kersey, S. A marching method for parametric surface/surface intersection. *Computer aided geometric design 7*, 1-4 (1990), 257–280. https://pdf.sciencedirectassets.com/271514/1-s2.0-S0167839600X00854/1-s2.0-016783969090035P/main.pdf.

[15] Behler, J., and Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical Review Letters 98*, 14 (2007), 1–4. https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.98.146401.

[16] Bellman, R. Dynamic programming and the numerical solution of variational problems. *Operations Research* (1957), 277–288.

[17] Bittner, B., and Pronzato, L. Kalman filtering in stochastic gradient algorithms: Construction of a stopping rule. *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing* (2004), ii–709. https://ieeexplore.ieee.org/document/1326356.

[18] Bjorck, J., Kabra, A., Weinberger, K., and Gomes, C. Characterizing the loss landscape in non-negative matrix factorization. *Proceedings of the AAAI Conference on Artificial Intelligence 35*, 8 (2021), 6768–6776. https://ojs.aaai.org/index.php/AAAI/article/download/16836/16643.

[19] Blum, A., and Rivest, R. Training a 3-node neural network is NP-complete. *Advances in neural information processing systems* (1989), 494–501.

[20] Bordes, A., Bottou, L., and Gallinari, P. SGD-QN: Careful quasi-Newton stochastic gradient descent. *J. Mach. Learn. Res. 10* (2009), 1737–1754.

[21] Bottou, L. Stochastic gradient learning in neural networks. *Proceedings of Nuero-Nimes* (1991).

[22] Bottou, L. Stochastic gradient descent tricks. *Neural Networks: Tricks of the Trade* (2012), 421–436. https://www.microsoft.com/en-us/research/wp-content/uploads/2012/01/tricks-2012.pdf.

[23] Bottou, L. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 421–436.

[24] Bray, A., and Dean, D. The statistics of critical points of Gaussian fields on large-dimensional spaces. *Physics Review Letter* (2007), 1–5. https://arxiv.org/pdf/cond-mat/0611023.pdf.

[25] Broyden, C. The convergence of a class of double-rank minimization algorithms 1 - General considerations. *Journal of Applied Mathematics 6* (1970), 76–90.

[26] Butcher, J. Coefficients for the study of runge-kutta integration processes. *Society for Industrial and Applied Mathematics 3* (1963), 185–201.

[27] Caldarola, D., Caputo, B., and Ciccone, M. Improving generalization in federated learning by seeking flat minima. *CoRR* (2022).

[28] Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. *arXiv preprint arXiv:1611.01838* (2016), 1–19. https://arxiv.org/pdf/1611.01838.pdf.

[29] Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. *arXiv preprint arXiv:1611.01838* (2016).

[30] Chaudhari, P., and Soatto, S. On the energy landscape of deep networks. *arXiv preprint arXiv:1511.06485* (2015), 1–20. https://arxiv.org/pdf/1511.06485.pdf.

[31] Chaudhari, P., and Soatto, S. On the energy landscape of deep networks. *arXiv preprint arXiv:1511.06485* (2015).

[32] Choromanska, A., Henaff, M., Mathieu, M., Arous, G., and LeCun, Y. The loss surface of multilayer networks. *Proceedings of 18th International Conference on Artificial Intelligence and Statistics (AISTATS)* (2015), 1–13. https://proceedings.mlr.press/v38/choromanska15.pdf.

[33] Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The loss surfaces of multilayer networks. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics* (2015), 192–204.

[34] Choromanska, A., LeCun, Y., and Arous, G. Open problem: The landscape of the loss surfaces of multilayer networks. *Conference on Learning Theory 40* (2015), 1756–1760. http://proceedings.mlr.press/v40/Choromanska15.pdf.

[35] Chung, N., Hoai, P., and Chung, H. Some new accelerated and stochastic gradient descent algorithms based on locally Lipschitz gradient constants. *Optimization Online* (2024), 1–21. https://optimization-online.org/wp-content/uploads/2024/11/CHC1811.pdf.

[36] Cooper, Y. The loss landscape of overparameterized neural networks. *arXiv preprint arXiv:1804.10200* (2018), 1–9. https://arxiv.org/pdf/1804.10200.pdf.

[37] Cutkosky, A., and Mehta, H. Momentum improves normalized SGD. *International conference on machine learning* (2020), 2260–2268.

[38] Dauphin, Y., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *arXiv preprint arXiv:1406.2572* (2014), 1–14. https://arxiv.org/pdf/1406.2572.pdf.

[39] Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems* (2014), vol. 27, pp. 2933–2941.

[40] Dinh, L., Pascanu, R., Bengio, S., , and Bengio, Y. Sharp minima can generalize for deep nets. *International Conference on Learning Representations* (2017).

[41] Dozat, T. Incorporating Nesterov momentum into ADAM. *Proceedings of 4th International Conference on Learning Representations* (2016), 1–6. https://cs229.stanford.edu/proj2015/054_report.pdf.

[42] Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. Essentially no barriers in neural network energy landscape. *International conference on machine learning 35* (2018), 1309–1318. http://proceedings.mlr.press/v80/draxler18a/draxler18a.pdf, http://proceedings.mlr.press/v80/draxler18a/draxler18a-supp.pdf.

[43] Du, S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. *International Conference on Learning Representations arXiv preprint*, arXiv:1810.02054v2 (2019), 1–19. https://arxiv.org/pdf/1810.02054.pdf.

[44] Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research 12* (2011), 2121–2159.

[45] Dyer, E., and Gur-Ari, G. Asymptotics of wide networks from Feynman diagrams. *arXiv preprint arXiv:1909.11304* (2019), 1–37. https://arxiv.org/pdf/1909.11304.pdf.

[46] Eleh, C., Mwanza, M., Aguegboh, E., and van Wyk, H. GeoAdaLer: Geometric insights into adaptive stochastic gradient descent algorithms. *arXiv arXiv:2405.16255* (2024), 1–17. https://arxiv.org/pdf/2405.16255.

[47] Eustratiadis, P., Gouk, H., Li, D., and Hospedales, T. Attacking adversarial defences by smoothing the loss landscape. *arXiv preprint arXiv:2208.00862* (2022), 1–20. https://arxiv.org/pdf/2208.00862.pdf.

[48] Fang, C., Li, C., Lin, Z., and Zhang, T. SPIDER: near-optimal non-convex optimization via Stochastic Path Integrated Differential EstimatoR. *NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems* (2018), 687–697. https://dl.acm.org/doi/10.5555/3326943.3327007.

[49] Fiacco, A., and McCormick, G. *Nonlinear programming: Sequential unconstrained minimization techniques*. Society for Industrial and Applied Mathematics, 1990.

[50] Fonseca, C., and Fleming, P. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary computation 3*, 1 (1995), 1–16. https://eprints.whiterose.ac.uk/79797/1/acse%20research%20report%20527.pdf.

[51] Fort, S., Dziugaite, G., Paul, M., Kharaghani, S., Roy, D., and Ganguli, S. Deep learning versus kernel learning: An empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. *Advances in Neural Information Processing Systems 33* (2020), 5850–5861. https://proceedings.neurips.cc/paper/2020/file/405075699f065e43581f27d67bb68478-Paper.pdf.

[52] Fort, S., Hu, H., and Lakshminarayanan, B. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757* (2019), 1–15. arxiv.org/pdf/1912.02757.pdf.

[53] Fort, S., and Jastrzebski, S. Large scale structure of neural network loss landscapes. *Advances in Neural Information Processing Systems (NeurIPS 2019) 32* (2019), 1–9. https://proceedings.neurips.cc/paper_files/paper/2019/file/48042b1dae4950fef2bd2aafa0b971a1-Paper.pdf.

[54] Freeman, C., and Bruna, J. Topology and geometry of deep rectified network optimization landscapes. *International Conference on Learning Representations* (2017), 1–22. https://arxiv.org/pdf/1611.01540.pdf.

[55] Frumkin, N., Gope, D., and Marculescu, D. Jumping through local minima: Quantization in the loss landscape of vision transformers. *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), 16978–16988. arxiv.org/pdf/2308.10814.pdf.

[56] Fyodorov, Y., and Williams, I. Replica symmetry breaking condition exposed by random matrix calculation of landscape complexity. *Journal of Statistical Physics 129*, 5-6 (2007), 1081–1116. https://arxiv.org/pdf/cond-mat/0702601.pdf.

[57] Gallagher, M., and Downs, T. Visualization of learning in multilayer perceptron networks using principal component analysis. *IEEE Transactions on Systems, Man, and Cybernetics - Part B (Cybernetics) 33*, 1 (2003), 28–34.

[58] Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D., and Wilson, A. Loss surfaces, mode connectivity, and fast ensembling of DNNs. *Advances in neural information processing systems 31* (2018), 1–10. proceedings.neurips.cc/paper_files/paper/2018/file/be3087e74e9100d4bc4c6268cdbe8456-Paper.pdf.

[59] Geiger, M., Jacot, A., Spigler, S., Gabriel, F., Sagun, L., d'Ascoli, S., Biroli, G., Hongler, C., and Wyart, M. Scaling description of generalization with number of parameters in deep learning. *J. Stat. Mech. Theory Exp. 2* (2020), 1–13. https://arxiv.org/pdf/1901.01608.pdf.

[60] Geiger, M., Petrini, L., and Wyart, M. Landscape and training regimes in deep learning. *Physics Reports 924* (2021), 1–18. https://www.sciencedirect.com/science/article/pii/S0370157321001290.

[61] Geiger, M., Spigler, S., d'Ascoli, S., Sagun, L., Baity-Jesi, M., Biroli, G., and Wyart, M. Jamming transition as a paradigm to understand the loss landscape of deep neural networks. *Physical Review E 100*, 1 (2019), 2019. https://journals.aps.org/pre/pdf/10.1103/PhysRevE.100.012115.

[62] Geiger, M., Spigler, S., Jacot, A., and Wyart, M. Disentangling feature and lazy training in deep neural networks. *J. Stat. Mech. Theory Exp. 11* (2020), 1–28. https://iopscience.iop.org/article/10.1088/1742-5468/abc4de/pdf.

[63] Ghadimi, S., and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization 23*, 4 (2013), 2341–2368.

[64] Ghorbani, B., Krishnan, S., and Xiao, Y. An investigation into neural net optimization via Hessian eigenvalue density. *arXiv preprint arXiv:1901.10159* (2019).

[65] Ghorbani, B., Mei, S., Misiakiewicz, T., and Montanari, A. Linearized two-layers neural networks in high dimension. *arXiv preprint arXiv:1904.12191* (2019), 1–65. https://arxiv.org/pdf/1904.12191.pdf.

[66] Glorot, X., and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (2010), 249–256.

[67] Goldstein, T., and Studer, C. Phasemax: Convex phase retrieval via basis pursuit. *arXiv preprint arXiv:1610.07531* (2016).

[68] Goodfellow, I., Vinyals, O., and Saxe, A. Qualitatively characterizing neural network optimization problems. *Proceedings of the International Conference on Learning Representations* (2015).

[69] Gotmare, A., Keskar, N., Xiong, C., and Socher, R. Using mode connectivity for loss landscape analysis. *arXiv preprint arXiv:1806.06977* (2018), 1–9. https://arxiv.org/pdf/1806.06977.pdf.

[70] Gower, R., and Richtárik, P. Stochastic dual ascent for solving linear systems. *arXiv preprint arXiv:1512.06890* (2015), 1–29. https://arxiv.org/pdf/1512.06890.

[71] Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., , and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677* (2017).

[72] Hajela, P. Genetic search -An approach to the nonconvex optimization problem. *AIAA journal 28*, 7 (1990), 1205–1210.

[73] Hardt, M., and Ma, T. Identity matters in deep learning. *International Conference on Learning Representations* (2017).

[74] Hastie, T., Tibshirani, R., and Friedman, J. H. *The Elements of Statistical Learning*. 2001.

[75] Hazan, E., Levy, K., and Shalev-Shwartz, S. On graduated optimization for stochastic non-convex problems. *Proceedings of Machine Learning Research 48* (2016), 1833–1841. http://proceedings.mlr.press/v48/hazanb16.pdf, http://proceedings.mlr.press/v48/hazanb16-supp.pdf.

[76] Hazan, E., Rakhlin, A., and Bartlett, P. Adaptive online gradient descent. *Advances in neural information processing systems 20* (2007). https://proceedings.neurips.cc/paper/2007/file/afd4836712c5e77550897e25711e1d96-Paper.pdf.

[77] He, F., and Tao, D. Recent advances in deep learning theory. *arXiv preprint arXiv:2012.10931* (ZZZZZZZZZZ), 1–42. https://arxiv.org/pdf/2012.10931.

[78] Hinton, G. Coursera neural networks for machine learning - lecture 6. 1–31. https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

[79] Hochreiter, S., and Schmidhuber, J. Flat minima. *Neural Computation 9*, 1 (1997), 1–42.

[80] Hochreiter, S., and Schmidhuber, J. Flat minima. *Neural Computation 9*, 1 (1997), 1–42.

[81] Hoffer, E., Hubara, I., and Soudry, D. Train longer, generalize better: Closing the generalization gap in large batch training of neural networks. *Conference on Neural Information Processing Systems* (2017).

[82] Im, D., Tao, M., and Branson, K. An empirical analysis of deep network loss surfaces. *arXiv preprint arXiv:1612.04010* (2016).

[83] Ioffe, S., and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on International Conference on Machine Learning 37* (2015), 448–456.

[84] Jain, P., Kakade, S., Kidambi, R., Netrapalli, P., and Sidford, A. Accelerating stochastic gradient descent for least squares regression. *Proceedings of the 31st Conference On Learning Theory 75* (2018), 545–604. http://proceedings.mlr.press/v75/jain18a/jain18a.pdf.

[85] Janzamin, M., Sedghi, H., and Anandkumar, A. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *arXiv preprint arXiv:1506.08473* (2015), 1–52. https://arxiv.org/pdf/1506.08473.pdf.

[86] Johnson, R., and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing Systems 26* (2013).

[87] Kawaguchi, K. Deep learning without poor local minima. *NIPS* (2016).

[88] Kawaguchi, K., Kaelbling, L., and Bengio, Y. Generalization in deep learning. *arXiv preprint arXiv:1710.05468* (2017).

[89] Keskar, N., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. On large-batch training for deep learning: Generalization gap and sharp minima. *ICLR* (2017).

[90] Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations* (2017).

[91] Khaki-Sedigh, A. *An Introduction to Data-Driven Control Systems. Chapter 6: The Simultaneous Perturbation Stochastic Approximation-Based Data-Driven Control Design*. WILEY, 2023. https://onlinelibrary.wiley.com/doi/epdf/10.1002/9781394196432.ch6.

[92] Khirirat, S., Feyzmahdavian, H., and Johansson, M. Mini-batch gradient descent: Faster convergence under data sparsity. *IEEE 56th Annual Conference on Decision and Control* (2017), 2880–2887. https://ieeexplore.ieee.org/document/8264077.

[93] Kiefer, J., and Wolfowitz, J. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics 23*, 3 (1952), 462–466.

[94] Kingma, D., and Adam, J. A method for stochastic optimization. *Proceedings of the International Conference on Learning Representations* (2014).

[95] Kingma, D., and Ba, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations* (2015).

[96] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations* (2015).

[97] Krogh, A., and Hertz, J. A simple weight decay can improve generalization. *NIPS* (1992).

[98] Krzakala, F., and Kurchan, J. Landscape analysis of constraint satisfaction problems. *Physical Review E 76*, 2 (2007), 1–17. https://journals.aps.org/pre/pdf/10.1103/PhysRevE.76.021122.

[99] Lee, J., Bahri, Y., Novak, R., Schoenholz, S., Pennington, J., and Sohl-Dickstein, J. Deep neural networks as Gaussian processes. *arXiv preprint arXiv:1711.00165* (2018), 1–17. https://arxiv.org/pdf/1711.00165.pdf.

[100] Lee, J., Schoenholz, S., Pennington, J., Adlam, B., Xiao, L., Novak, R., and Sohl-Dickstein, J. Finite versus infinite neural networks: An empirical study. *arXiv preprint arXiv:2007.15801* (2020), 1–17. https://proceedings.neurips.cc/paper/2020/file/ad086f59924fffe0773f8d0ca22ea712-Paper.pdf.

[101] Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems 31* (2018), 1–11. https://proceedings.neurips.cc/paper_files/paper/2018/file/a41b3bb3e6b050b6c9067c67f663b915-Paper.pdf.

[102] Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In *Advances in neural information processing systems* (2018), vol. 31, pp. 6389–6399.

[103] Li, L., and Spratling, M. Understanding and combating robust overfitting via input loss landscape analysis and regularization. *Pattern Recognition 136* (2023).

[104] Li, M., Zhang, T., Chen, Y., and Smola, A. Efficient mini-batch training for stochastic optimization. *KDD '14: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (2014), 661–670. https://www.cs.cmu.edu/~muli/file/minibatch_sgd.pdf.

[105] Li, Y., and Yuan, Y. Convergence analysis of two-layer neural networks with relu activation. *arXiv preprint arXiv:1705.09886* (2017).

[106] Lian, X., Huang, Y., Li, Y., and Liu, J. Asynchronous parallel stochastic gradient for nonconvex optimization. *arXiv preprint arXiv:1506.08272* (2019), 1–33. https://arxiv.org/pdf/1506.08272.

[107] Liang, R., Bo, L., and Sun, Y. Loss landscape analysis for deep learning: A survey. https://api.semanticscholar.org/CorpusID:255786283.

[108] Liao, Q., and Poggio, T. Theory of deep learning ii: Landscape of the empirical risk in deep learning. *arXiv preprint arXiv:1703.09833* (2017).

[109] Lipton, Z. Stuck in a what? Adventures in weight space. *ICLR Workshop* (2016).

[110] Liu, B. Understanding the loss landscape of one-hidden-layer ReLU networks: Part 1 - Theory. *Knowledge-Based Systems 220* (2021), 1–10. https://arxiv.org/pdf/2002.04763.pdf.

[111] Liu, C., Salzmann, M., Lin, T., Tomioka, R., and S usstrunk, S. On the loss landscape of adversarial training: Identifying challenges and how to overcome them. *Advances in Neural Information Processing Systems 33* (2020). https://proceedings.neurips.cc/paper/2020/hash/f56d8183992b6c54c92c16a8519a6e2b-Abstract.html.

[112] Lorch, E. Visualizing deep network training trajectories with PCA. *ICML Workshop on Visualization for Deep Learning* (2016).

[113] Lu, H., and Kawaguchi, K. Depth creates no bad local minima. *arXiv preprint arXiv:1702.08580* (2017).

[114] Luo, L., Zhang, W., Zhang, Z., Zhu, W., Zhang, T., and Pei, J. Sketched follow-the-regularized-leader for online factorization machine. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining* (2018), 1900–1909. https://doi.org/10.1145/3219819.3220044.

[115] Martens, J. Deep learning via Hessian-free optimization. *Proceedings of the International Conference of Machine Learning* (2010).

[116] Matthews, A., Hron, J., Rowland, M., Turner, R., and Ghahramani, Z. Gaussian process behaviour in wide deep neural networks. *International Conference on Learning Representations arXiv preprint*, arXiv:1804.11271 (2018), 1–36. https://arxiv.org/pdf/1804.11271.pdf.

[117] Mei, S., Misiakiewicz, T., and Montanari, A. Mean-field theory of two-layers neural networks: Dimension-free bounds and kernel limit. *Proceedings of Machine Learning Research 99* (2019), 1–77. https://proceedings.mlr.press/v99/mei19a/mei19a.pdf.

[118] Mei, S., Montanari, A., and Nguyen, P. A mean field view of the landscape of two-layer neural networks. *Proc. Natl. Acad. Sci. 115*, 33 (2018), 1–103. https://arxiv.org/pdf/1804.06561.pdf.

[119] Mokhtari, A., and Ribeiro, A. Decentralized double stochastic averaging gradient. *49th Asilomar Conference on Signals, Systems and Computers* (2015), 406–410. https://ieeexplore.ieee.org/document/7421158.

[120] Mu, Y., Ding, W., Zhou, T., and Tao, D. Constrained stochastic gradient descent for large-scale least squares problem. *Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining* (2013). https://api.semanticscholar.org/CorpusID:9503956.

[121] Mukherjee, S., Niyogi, P., Poggio, T., and Rifkin, R. Learning theory: Stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Advances in Computational Mathematics 25* (2006), 161–193. https://www.cs.cmu.edu/afs/cs/project/link-3/lafferty/www/ml-stat-www/poggio.pdf.

[122] Nesterov, Y. A method for unconstrained convex minimization problem with the rate of convergence o(1/k2). *Doklady AN USSR 269* (1983), 543–547.

[123] Newton, D., Yousefian, F., and Pasupathy, R. Stochastic gradient descent: Recent trends. *Recent advances in optimization and modeling of contemporary problems* (2018), 193–220.

[124] Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076* (2018), 1–19. https://arxiv.org/pdf/1805.12076.pdf.

[125] Neyshabur, B., Tomioka, R., Salakhutdinov, R., and Srebro, N. Geometry of optimization and implicit regularization in deep learning. *arXiv preprint arXiv:1705.03071* (2017), 1–13. https://arxiv.org/pdf/1705.03071.pdf.

[126] Nguyen, L., Liu, J., Scheinberg, K., and Takáč, M. Stochastic recursive gradient algorithm for nonconvex optimization. *arXiv preprint arXiv:1705.07261v1* (2017), 1–15. https://arxiv.org/pdf/1705.07261.

[127] Nguyen, P., and Pham, H. A rigorous framework for the mean field limit of multilayer neural networks. *arXiv preprint arXiv:2001.11443* (2020), 1–125. https://arxiv.org/pdf/2001.11443.pdf.

[128] Nguyen, Q., and Hein, M. The loss surface of deep and wide neural networks. *International Conference on Machine Learning* (2017).

[129] Nguyen, Q., and Hein, M. Optimization landscape and expressivity of deep CNNs. *International conference on machine learning* (2018), 3730–3739. arxiv.org/pdf/1710.10928.pdf.

[130] Nguyen, Q., Mukkamala, M., and Hein, M. On the loss landscape of a class of deep neural networks with no bad local valleys. *arXiv preprint arXiv:1809.10749* (2018), 1–20. https://arxiv.org/pdf/1809.10749.pdf.

[131] Oftadeh, R., Shen, J., Wang, Z., and Shell, D. Eliminating the invariance on the loss landscape of linear autoencoders. *International Conference on Machine Learning 119* (2020), 7405–7413. http://proceedings.mlr.press/v119/oftadeh20a/oftadeh20a.pdf, http://proceedings.mlr.press/v119/oftadeh20a/oftadeh20a-supp.pdf.

[132] Oymak, S., Fabian, Z., Li, M., and Soltanolkotabi, M. Generalization guarantees for neural networks via harnessing the low-rank structure of the Jacobian. *arXiv preprint arXiv:1906.05392* (2019), 1–58. https://arxiv.org/pdf/1906.05392.pdf.

[133] Paccolat, J., Petrini, L., Geiger, M., Tyloo, K., and Wyart, M. Geometric compression of invariant manifolds in neural nets. *arXiv preprint arXiv:2007.11471* (2020), 1–26. https://arxiv.org/pdf/2007.11471.pdf.

[134] Pardalos, P. Generation of large-scale quadratic programs for use as global optimization test problems. *ACM Transactions on Mathematical Software 13*, 2 (1987), 133–137.

[135] Parisi, G. Probabilistic line searches for stochastic optimization. *arXiv preprint arXiv:0706.0094* (2016).

[136] Park, D., Sohl-Dickstein, J., Le, Q., and Smith, S. The effect of network width on stochastic gradient descent and generalization: An empirical study. *International Conference on Machine Learning* (2019), 5042–5051. https://proceedings.mlr.press/v97/park19b/park19b.pdf.

[137] Polyak, B. Some methods of speeding up the convergence of iteration methods. *Computational Mathematics and Mathematical Physics 4*, 5 (1964), 1–17.

[138] Raschka, S. *Python machine learning: Machine learning and deep learning with python, scikit-learn, and TensorFlow 2*. Birmingham: Packt Publishing, Limited, 2019.

[139] Recanatesi, S., Farrell, M., Advani, M., Moore, T., Lajoie, G., and Shea-Brown, E. Dimensionality compression and expansion in deep neural networks. *arXiv preprint arXiv:1906.00443* (2019), 1–12. https://arxiv.org/pdf/1906.00443.pdf.

[140] Reddi, S., Kale, S., and Kumar, S. On the convergence of ADAM and beyond. *arXiv preprint arXiv:1904.09237v1* (2018), 1–23. https://arxiv.org/pdf/1904.09237v1.

[141] Robbins, H., and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics 22*, 3 (1951), 400–407.

[142] Rockafellar, R. Proximal subgradients, marginal values, and augmented Lagrangians in nonconvex optimization. *Mathematics of Operations Research 6*, 3 (1981), 424–436.

[143] Rotskoff, G., and Vanden-Eijnden, E. Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error. *arXiv preprint arXiv:1805.00915* (2018), 1–36. https://www.researchgate.net/publication/324908439_Neural_networks_as_Interacting_Particle_Systems_Asymptotic_convexity_of_the_Loss_Landscape_and_Universal_Scaling_of_the_Approximation_Error.

[144] Ruder, S. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747* (2016).

[145] Ruder, S. An overview of gradient descent optimization algorithms. *arXiv arXiv:1609.04747v2* (2017), 1–14. https://arxiv.org/pdf/1609.04747.pdf.

[146] Safran, I., and Shamir, O. On the quality of the initial basin in overspecified neural networks. *International Conference on Machine Learning* (2016), 774–782.

[147] Sagun, L., Evci, U., G uney, V., Dauphin, Y., and Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454* (2018), 1–15. arxiv.org/pdf/1706.04454.pdf.

[148] Sagun, L., Guney, V., Arous, G., and LeCun, Y. Explorations on high dimensional landscapes. *arXiv preprint arXiv:1412.6615* (2014), 1–11. https://arxiv.org/pdf/1412.6615.pdf.

[149] Same, F., Chen, G., and van Deemter, K. Non-neural models matter: A re-evaluation of neural referring expression generation systems. *arXiv preprint arXiv:2203.08274v1* (2022), 1–14. https://arxiv.org/pdf/2203.08274v1.

[150] Sankar, A., Khasbage, Y., Vigneswaran, R., and Balasubramanian, V. A deeper look at the hessian eigenspectrum of deep neural networks and its applications to regularization. *Proceedings of the AAAI Conference on Artificial Intelligence 35*, 11 (2021), 9481–9488. https://ojs.aaai.org/index.php/AAAI/article/view/17142/16949.

[151] Saxe, A., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B., and Cox, D. On the information bottleneck theory of deep learning. *J. Stat. Mech. Theory Exp. 12* (2019).

[152] Saxe, A., McClelland, J., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *International Conference on Learning Representations* (2014).

[153] Schmidt, M., Roux, N. L., and Bach, F. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming 162* (2017), 83–112. https://doi.org/10.1007/s10107-016-1030-6.

[154] Scholkopf, B., and Smola, A. Learning with kernels: Support vector machines, regularization, optimization, and beyond. *MIT press* (2001).

[155] Sirignano, J., and Spiliopoulos, K. Mean field analysis of neural networks: A central limit theorem. *Stochastic Process Appl. 130*, 3 (2020), 1820–1852. arxiv.org/pdf/1808.09372.pdf.

[156] Sirignano, J., and Spiliopoulos, K. Mean field analysis of neural networks: A law of large numbers. *SIAM J. Appl. Math 80*, 2 (2020), 725–752. https://arxiv.org/pdf/1805.01053.pdf.

[157] Skorokhodov, I., and Burtsev, M. Loss landscape sightseeing with multi-point optimization. *arXiv preprint arXiv:1910.03867* (2019), 1–8. https://arxiv.org/pdf/1910.03867.pdf.

[158] Smith, D., and Vamanamurthy, M. How small is a unit ball? *Mathematics Magazine 62*, 2 (1989), 101–107. https://doi.org/10.1080/0025570X.1989.11977419.

[159] Smith, L., and Topin, N. Exploring loss function topology with cyclical learning rates. *arXiv preprint arXiv:1702.04283* (2017).

[160] Soltanolkotabi, M., Javanmard, A., and Lee, J. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *arXiv preprint arXiv:1707.04926* (2017).

[161] Soudry, D., and Carmon, Y. No bad local minima: Data independent training error guarantees for multilayer neural networks. *arXiv preprint arXiv:1605.08361* (2016), 1–12. https://arxiv.org/pdf/1605.08361.pdf.

[162] Soudry, D., and Hoffer, E. Exponentially vanishing sub-optimal local minima in multilayer neural networks. *arXiv preprint arXiv:1702.05777* (2017).

[163] Spigler, S., Geiger, M., d'Ascoli, S., Sagun, L., Biroli, G., and Wyart, M. A jamming transition from under- to over-parametrization affects generalization in deep learning. *Journal of Physics A: Mathematical and Theoretical 52*, 47 (2019), 1–18. https://iopscience.iop.org/article/10.1088/1751-8121/ab4c8b/pdf.

[164] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research 15*, 1 (2014), 1929–1958. https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf.

[165] Sun, R. Optimization for deep learning: An overview. *Journal of the Operations Research Society of China 8*, 2 (2020), 249–294. https://www.ise.ncsu.edu/fuzzy-neural/wp-content/uploads/sites/9/2020/08/Optimization-for-deep-learning.pdf.

[166] Sun, R., Li, D., Liang, S., Ding, T., and Srikant, R. The global landscape of neural networks: An overview. *IEEE Signal Processing Magazine 37*, 5 (2020), 95–108. https://arxiv.org/pdf/2007.01429.pdf.

[167] Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of momentum and initialization in deep learning. *Proceedings of the International Conference of Machine Learning* (2013).

[168] Swirszcz, G., Czarnecki, W., and Pascanu, R. Local minima in training of neural networks. *arXiv preprint arXiv:1611.06310* (2017), 1–12. https://arxiv.org/abs/1611.06310.

[169] Tartaglione, E., Grangetto, M., Cavagnino, D., and Botta, M. Delving in the loss landscape to embed robust watermarks into neural networks. *25th International Conference on Pattern Recognition (ICPR)* (2021), 1243–1250. https://iris.unito.it/bitstream/2318/1891440/1/output-5.pdf.

[170] Terzopoulos, D. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence 10*, 4 (1988), 417–438. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=85d6afc4cfbeba292626ed1867daa447b48309e5.

[171] Tian, Y. An analytical formula of population gradient for two-layered ReLU network and its applications in convergence and critical point analysis. *ICML* (2017).

[172] Tian, Y., Zhang, Y., and Zhang, H. Recent advances in stochastic gradient descent in deep learning. *Mathematics 11*, 3 (2023), 682.

[173] Tieleman, T., and Hinton, G. RMSprop gradient optimization. *Neural Networks for Machine Learning* (2012). http://www.cs.toronto.edu/œtijmen/csc321/slides/lecture_slides_lec6.pdf.

[174] Todd, M. 'fat'triangulations, or solving certain nonconvex matrix optimization problems. *Mathematical programming 31*, 2 (1985), 123–136.

[175] Vapnik, V. Principles of risk minimization for learning theory. *Annual Conference on Neural Information Processing Systems* (1992). https://koza.if.uj.edu.pl/~krzemien/radFarm2021/materials/vapnik-principles-of-risk-minimization-for-learning-theory.pdf.

[176] Vesseron, N., Redko, I., and Laclau, C. Deep neural networks are congestion games: From loss landscape to Wardrop equilibrium and beyond. *International Conference on Artificial Intelligence and Statistics* (2021), 1765–1773. https://proceedings.mlr.press/v130/vesseron21a/vesseron21a.pdf, https://proceedings.mlr.press/v130/vesseron21a/vesseron21a-supp.pdf.

[177] vonLuxburg, U., and Bousquet, O. Distance-based classification with lipschitz functions. *J. Mach. Learn. Res. 5* (2004), 669–695. https://www.jmlr.org/papers/volume5/luxburg04b/luxburg04b.pdf.

[178] Welch, S. C. YouTube Video: The Misconception that Almost Stopped AI. *Welch Labs* (2025). https://www.youtube.com/watch?v=NrO20Jb-hy0&t=206s.

[179] Williams, C. Computing with infinite networks. *Advances in Neural Information Processing Systems 9* (1997), 295–301. https://proceedings.neurips.cc/paper_files/paper/1996/file/ae5e3ce40e0404a45ecacaaf05e5f735-Paper.pdf.

[180] Wu, L., Zhu, Z., and E, W. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239* (2017), 1–12. https://arxiv.org/pdf/1706.10239.pdf.

[181] Xie, B., Liang, Y., , and Song, L. Diverse neural network learns true target functions. *Artificial Intelligence and Statistics* (2017), 1216–1224.

[182] Xu, J., Yap, D., and Prabhu, V. Understanding adversarial robustness through loss landscape geometries. *Proceedings of the International Conference on Machine Learning (ICML)* (2019), 18. https://arxiv.org/pdf/1907.09061.pdf.

[183] Yang, G. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760* (2019), 1–67. arxiv.org/pdf/1902.04760.pdf.

[184] Zeiler, M. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701* (2012), 1–6. https://arxiv.org/abs/1212.5701.

[185] Zeng, Y., Han, D., Su, Y., and Xie, J. Fast stochastic dual coordinate descent algorithms for linearly constrained convex optimization. *arXiv preprint arXiv:2307.16702* (2023). https://arxiv.org/pdf/2307.16702.

[186] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *ICLR* (2017).

[187] Zhang, S., Choromanska, A., and LeCun, Y. Deep learning with elastic averaging SGD. *Neural Information Processing Systems Conference* (2015), 1–24.

[188] Zhao, P., Chen, P., Das, P., Ramamurthy, K., and Lin, X. Bridging mode connectivity in loss landscapes and adversarial robustness. *International Conference on Learning Representations* (2020).

6 Appendix 1 - Survey Evaulation & Selection Criteria

Table 5: Survey Evaulation Criteria (SEC)

Index	Criterion	Ranking
E1	Abstract and keywords	Pass/Fail
E2	Introduction	Pass/Fail
E3	Literature review	Pass/Fail
E4	Methodology	Pass/Fail
E5	Results analysis & discussion	Pass/Fail
E6	Conclusion	Pass/Fail
E7	Originality	Pass/Fail
E8	Format	Pass/Fail
E9	Language (English)	Pass/Fail
E10	Number of Pages ≥ 5	Pass/Fail
E11	Paper did not reference a key/main algorithm(s)	Pass/Fail
E12	Not cited by ≥ 10 research papers after 10+ years since published	Pass/Fail
E13	Algorithm was relevant to only to a few applications	Pass/Fail
E14	Algorithm was hybrid of LLS and others, having minimal LLS details	Pass/Fail
E15	Paper was only a citation reference and can't be easily located online	Pass/Fail

Table 6: LLS Survey Selection Criteria (SSC)

Index	Criterion	Ranking
S1	Mentioned LLS or one of its other alternative names	Pass/Fail
S2	Algorithm based	Pass/Fail
S3	Did not require convixity or semi-convexity	Pass/Fail
S4	Algorithm was hybrid of LLS and others, having minimal LLS details	Pass/Fail
S5	Paper was only a citation reference and can't be easily located online	Pass/Fail

Table 7: SGD Survey Selection Criteria (SSC)

Index	Criterion	Ranking
S1	Mentioned LLS or one of its other alternative names	Pass/Fail
S2	Algorithm based	Pass/Fail
S3	Did not require convixity or semi-convexity	Pass/Fail
S4	Algorithm was hybrid of LLS and others, having minimal LLS details	Pass/Fail
S5	Paper was only a citation reference and can't be easily located online	Pass/Fail

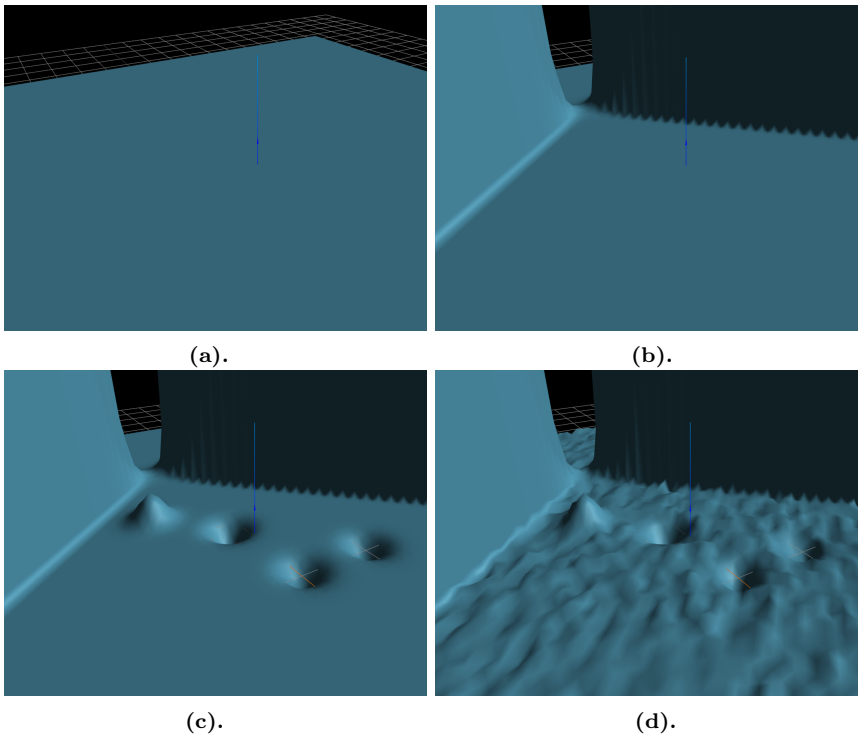
7 Appendix 2 - LLS and SGD Research Comparison

One way to help extrapolate our understanding to such higher dimensions is to take a flat plane in $\mathbb{R}^3$ and deform it in a series of steps to help it look more like an actual 3-dimensional LLS, or a 3-dimensional shadow/subset of an n-dimensional LLS, as shown in Figure 10.

Figure 10 was generated by building a custom simulation engine, coded in Javascript, Three.JS and other mathematics libraries. It helps visualise LLS in higher dimensions by layering various features in 3 dimensions to increase the complexity. Having introduced LLS, the next step is to introduce the conventional ways to search and traverse these LLSs.

8 Appendix 3 - Link Analysis of Stochastic Gradient Descent Research

The following link analysis in Figure 11 shows most of the subsequent research that has emerged has cited the original SGD algorithm.



(a). (b).

(c). (d).

Figure 10: Converting Simple Surface to Resemble a LLS

(a). A simple flat plane surface $\mathcal{F}(x, y) = 1$ is introduced.
(b). A sub-function that deforms the plane to have a saddle point is introduced by adding $f(x, y) = 2 \exp\Big(-\big((x+2)^2 - (y+2)^2\big)/0.2\Big)$.
(c). A sub-function that creates high mound is introduced by adding $f(x, y) = 4 \exp\Big(-\big((x+2)^2 + (y+2)^2\big)/0.2\Big)$, i.e. the global maximum, a sub-function that creates a deep well near the saddle point is introduced by adding $f(x, y) = -4 \exp\Big(-\big((x+0.5)^2 + (y+0.5)^2\big)/0.2\Big)$, i.e. the global minimum, together with a sub-function that creates two shallower wells by adding $f(x, y) = -2.85 \exp\Big(-\big((x-2)^2 + y^2\big)/0.2\Big) - 2.85 \exp\Big(-\big((x-2)^2 + (y-2)^2\big)/0.2\Big)$, i.e. local minima.
(d). Finally, a perturbation of random noise is introduced onto the surface, making it dynamically change over time.

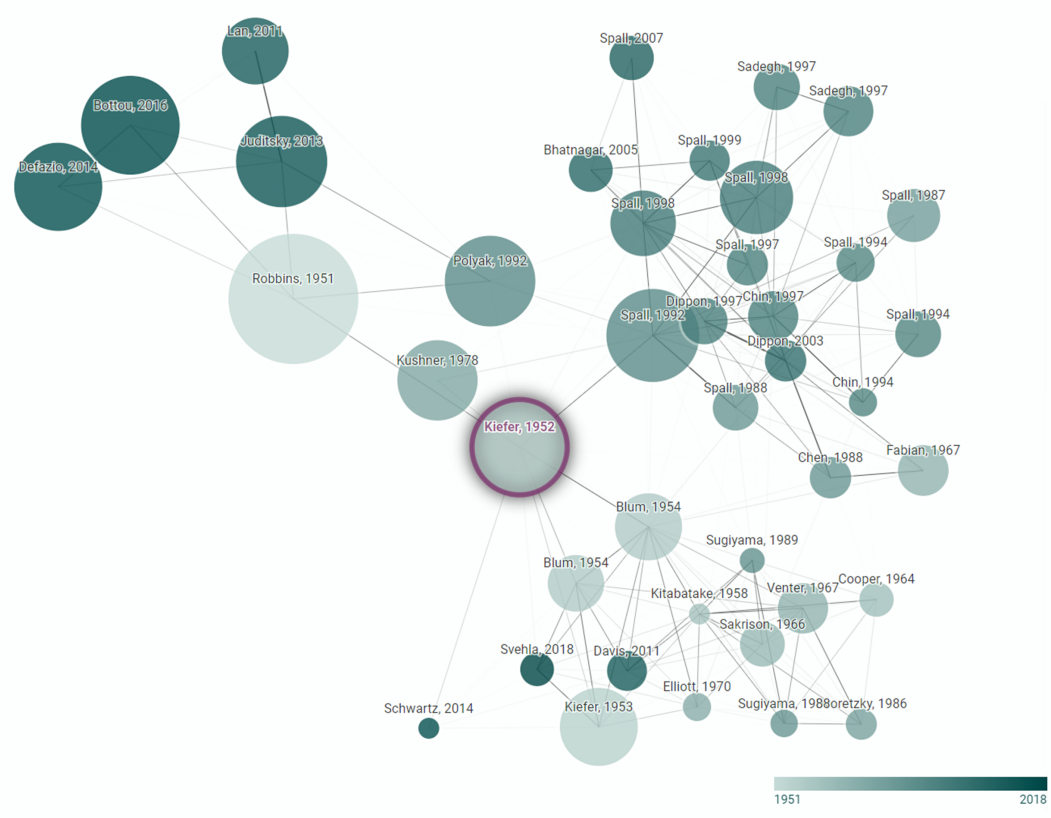


Figure 11: Link Analysis of SGD Related Research Papers

This link analysis visually shows the impact that the first research paper on SGD has had on subsequent researchers. The corresponding link analysis was obtained from the website https://www.connectedpapers.com on the paper [93]. Here we can see that the main subsequent researcher in SGD that quoted the seminal research paper [93] is James C. Spall.