Evolutionary Algorithms in the Light of SGD: Limit Equivalence, Minima Flatness, and Transfer Learning

Andrei Kucharavy[1,*], Rachid Guerraoui[1] and Ljiljana Dolamic[2]

[1]IC department, EPFL, Lausanne, Switzerland
[2]CYD Campus, armasuisse S+T, Thun, Switzerland
[*]andrei.kucharavy@hevs.ch; Now at HES-SO Valais/Wallis, Sierre, Switzerland

Abstract

Whenever applicable, the Stochastic Gradient Descent (SGD) has shown itself to be unreasonably effective. Instead of underperforming and getting trapped in local minima due to the batch noise, SGD leverages it to learn to generalize better and find minima that are good enough for the entire dataset. This led to numerous theoretical and experimental investigations, especially in the context of Artificial Neural Networks (ANNs), leading to better machine learning algorithms. However, SGD is not applicable in a non-differentiable setting, leaving all that prior research off the table.

In this paper, we show that a class of evolutionary algorithms (EAs) inspired by the Gillespie-Orr Mutational Landscapes model for natural evolution is formally equivalent to SGD in certain settings and, in practice, is well adapted to large ANNs. We refer to such EAs as Gillespie-Orr EA class (GO-EAs) and empirically show how an insight transfer from SGD can work for them. We then show that for ANNs trained to near-optimality or in the transfer learning setting, the equivalence also allows transferring the insights from the Mutational Landscapes model to SGD.

We then leverage this equivalence to experimentally show how SGD and GO-EAs can provide mutual insight through examples of minima flatness, transfer learning, and mixing of individuals in EAs applied to large models.

Introduction

Over the last decade and a half, deep learning has achieved impressive progress(LeCun et al., 2015). From image recognition (Krizhevsky et al., 2012) to image and text synthesis (Karras et al., 2018; Brown et al., 2020), a progressive increase in the size of ANN models, combined with an increase in the size and variety of datasets and new approaches, such as GANs or Self-Attention (Goodfellow et al., 2014; Vaswani et al., 2017), unlocked new capabilities - both expected and unexpected (Ganguli et al., 2022). However, one of the most fundamental aspects of such models - their training process - is still far from fully understood.

The ability of ANN models to solve large classes of problems is not exactly surprising. ANNs have been proven to be universal approximators that can fit any function (Hornik et al., 1989). What is surprising is that good approximations can reliably be found by trivial gradient descent, even if it is calculated on small batches of dataset samples - a procedure known as Stochastic Gradient Descent (SGD) LeCun et al. (1998). SGD can teach ANNs robust, well-generalizing features (Bottou and Cun, 2003; Bottou and Bousquet, 2007), even when models are overparametrized enough to memorize images (Dinh et al., 2017; Zhang et al., 2021). While numerous other optimizers have been proposed and successfully applied - such as Adam (Kingma and Ba, 2015), few remain as well-explored as SGD.

Thanks to its simplicity and unreasonable effectiveness, SGD has been explored better than any other machine learning algorithm, both theoretically and experimentally. Among other things, we now understand how SGD optimization responds to learning rate adjustment (Jastrzebski et al., 2018; Hoffer et al., 2017), minibatch noise (Ziyin et al., 2021b; Wu et al., 2020), or additional noise injection in case of large batches (Xie et al., 2021; Zhu et al., 2018). Several experimental and theoretical works made clear the importance of model over-parametrization for both the loss landscape smoothing (Li et al., 2018) and increasing the connectedness of minima (Nguyen et al., 2021; Jacot et al., 2018), allowing the SGD to avoid getting trapped in local minima and to train ANNs to recognize robust and generalizing features instead.

However, SGD applies only in a fully differentiable setting. It requires a transformation of the learning problem into a continuous form and can only train fully differentiable models with layers that can support gradient back-propagation. However, a fully differentiable setting is highly limiting. For instance, a stronger, discrete version of the wildly successful soft attention architecture (Bahdanau et al., 2015; Vaswani et al., 2017) - hard attention (Mnih et al., 2014; Xu et al., 2015) - cannot be trained with SGD or other optimizers requiring a differentiable setting.

These limitations led to increasing attention to approaches that can work in a non-differentiable setting. A prime example is reinforcement learning (RL) (Sutton, 1991). While reinforcement learning achieved impressive results in some settings, notably strategy games (Silver et al., 2017; Vinyals et al., 2019), RL is more poorly understood and requires extensive hyper-parameter space sweeps for each new application (Salimans et al., 2017).

These shortcomings led to a resurgence of interest in evolutionary optimization algorithms (EA). Significantly more stable than RLs (Salimans et al., 2017), EAs were shown to scale well and, when supplied with sufficient computational power, to outperform reinforcement learning on a range of complex tasks (Salimans et al., 2017; Such et al., 2017). Unfortunately, the understanding of why EAs perform so well and how their performance could be improved has been limited.

Here, we show that a relatively simple class of EAs rooted in a formalization from population genetics - Gillespie-Orr Mutational Landscapes model - approximates well SGD in theory and practice. We call that class Gillespie-Orr Evolutionary Algorithms (GO-EA) and experimentally show how, thanks to them, insight can be transferred between SGD and the Mutational Landscapes model of evolution. We demonstrate it with a well-established MNIST digit recognition task (LeCun et al., 1998).

Interestingly, such an equivalence allows us a novel insight into the long-standing *Flat Minima* hypothesis in Machine Learning. The Flat Minima hypothesis postulates that suggesting that SGD trains better generalizing ANNs by finding flatter minima, given that those are the only ones to be robust to the inherent SGD minibatch noise (Hochreiter and Schmidhuber, 1994). While this theory has found some empirical support (Goodfellow and Vinyals, 2015; Keskar et al., 2017; Chaudhari et al., 2017; Li et al., 2018), it also has counter-examples (Dinh et al., 2017; Ziyin et al., 2021a). Here we leverage the equivalence between GO-EA and SGD to transfer results from the Mutational Landscapes model (Kucharavy et al., 2018), suggesting that Minima Flatness has more to do with the redundancy of feature recognition rather than the model generalizability. We show that this has implications for transfer learning, an increasingly prevalent paradigm, where existing models are adjusted for a new application rather than re-trained from scratch (Pan and Yang, 2010; Yosinski et al., 2014).

Specifically, our contributions are:

- **Establishing a limit equivalence** between SGD and GO-EA class in the low learning rate limit
- **Presenting a population size effect** as equivalent to uniform anisotropic noise in SGD and perturbed gradient descent (PGD), suggesting a new role for neutral drift
- **Empirically validating hypotheses** that arise from such equivalence, namely with regards to *minima flatness*, *transfer learning*, evolutionary algorithms *hyperparameters*, and sampled parameter *update vectors mixing*

Evolution in Algorithms and Genetics

Evolutionary Algorithms

Evolutionary algorithms for optimization and AI have been introduced by Fogel et al. (1966) and were directly inspired by evolution understood in a strictly Darwinian sense - as a pure mutation-selection loop looking for an optimum on a fitness landscape. As additional concepts were picked up from biological evolution, EA became progressively more complex, culminating in the Genetic Algorithm[1] (Goldberg, 1989; Holland, 1992) - perhaps the most widely known machine learning algorithm before the ubiquitous success SGD.

However, the Genetic Algorithm was far from the last evolutionary algorithm introduced. In the late 1990s and early 2000s, several new evolutionary algorithms were proposed. They were inspired less by biological evolution and more by heuristics to accelerate the search. Notable examples are Enforced Sub-Populations (ESP) (Gomez and Miikkulainen, 1997), Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen and Ostermeier, 2001), CooperativeSynapse Neuroevolution (CoSynNE) (Gomez et al., 2008) or Natural Evolution Strategies (NES) (Wierstra et al., 2008). However, their use remained limited, and they were rapidly eclipsed by the success of deep learning and reinforcement learning, especially when applied to ANNs, with neuroevolution remaining a relative niche domain (Floreano et al., 2008).

Population Genetics and Mutational Landscape

Where evolutionary algorithms drew inspiration from biological evolution, population genetics instead sought to formalize and precisely quantify it. Its first challenge was Darwin's theory of natural selection itself. The small, gradual changes that nature was supposed to select from were incompatible with observed patterns of inheritance in most organisms, as discovered by Mendel. It was not until Sir Ronald Fisher introduced his Geometric Model of Evolution, almost 70 years later, that the paradox was resolved from a theoretical standpoint (Fisher, 1930). By representing fitness relationship to traits as a scalar field, where gene variants encoded specific points, this model represented evolution as a random walk trying to ascend a fitness peak, with steps following Mendelian patterns and trait changes looking Darwinian (Tenaillon, 2014; Orr, 2005). Shortly after, Wright (1932) introduced a generalization of that model with numerous fitness peaks and valleys and coined the term of *fitness landscapes* to represent it, which in the context of

[1] Evolutionary algorithms often have multiple conflicting names. To avoid confusion, here we adopt the taxonomy presented in Hansen et al. (2015), and following the lead of Galván and Mooney (2021), use Evolutionary Algorithm as a general term and reserve Genetic Algorithm name strictly to the algorithm presented in Goldberg (1989); Holland (1992), aka including chromosomes and recombination.

machine learning saw their fitness inverted into the loss and became known as *loss landscapes*.

The Fitness Landscapes model did not stop there, however. Kimura (1968) realized that survival was not only determined by fitness but also by chance. A fire in the forest could easily exterminate a population of deer that acquired a mutation allowing them to better digest grass, eliminating the improved trait by pure chance. This observation led to the introduction of *neutral drift* as a counter-balance to natural selection and the formulation of the nearly neutral theory of molecular evolution (Ohta, 1992). By bringing in evidence from paleontology, Gould and Lewontin (1979) made it clear that evolution is not a steady process but rather occurs in rapid bursts shortly after the environment changes. If the environment does not change, neither do organisms inhabiting it, leading to living fossils, such as horseshoe crabs in the Delaware River delta, unchanged for the last 480 million years. This became known as the *adaptive bursts chain* theory of evolution (Lande, 1986).

However, despite its refinements, the geometric model of evolution still had one major issue - the biological reality. The discovery of DNA in the 1950s meant that genetic code was a long string, with mutations only affecting a letter or a word in it at a time. While theories of evolution representing it as such were developed - notably the string rewrite graph *NK theory* (Kauffman, 1969; Kauffman and Levin, 1987), they looked nothing like scalar vector fields of geometric models and lacked the quantitative explanatory capabilities of the latter. This conundrum was resolved by Gillespie (1983) and Orr (2002). The former noted that within the adaptive burst theory evolution, the adaptation would occur only after the environment would change and would start with an organism that already had a genetic code mapping to a fitness maximum within the accessible genetic code space prior to the environment change. Hence it was starting the search for mutations improving its fitness from a genetic code with an already extreme fitness compared to the ensemble of all possible genetic codes. In turn, it meant that the fitness change could be described by an extreme limit distribution, leveraging the Fisher-Tippet-Gnedenko theorem (Gillespie, 1984; Fisher and Tippett, 1928; Gnedenko, 1943). At this point, Orr (2002, 2006) showed that the adaptive walk in Fisher's Geometric Models belonged to the same limit distribution family; hence, both were formally equivalent. Finally, Joyce et al. (2008) showed that the underlying classes of fitness distributions across code strings mattered little - the Geometric Model still represented most heavy-tailed or truncated fitness distributions well enough.

The resulting model became known as *Mutational Landscapes models* (Orr, 2005), or Modern Fisher Geometric Model (Tenaillon, 2014). Remarkably, despite being explicitly developed for biological organisms, it is well-suited for any coding space search, with a strict equivalence to the biological setting when the adaptation occurs from an already well-performing code, such as in transfer learning or the final stages of model training.

Prior work

Unfortunately, to our knowledge, the equivalence between evolutionary algorithms and gradient descent algorithms has remained a relatively unexplored topic.

Closest to our approach, Salimans et al. (2017) established an informal equivalence between Q-Learning and Policy Gradient (Watkins and Dayan, 1992; Williams, 1992) and a type of evolutionary search algorithm (ES) - Scalable ES, in the context of reinforcement learning. The authors speculate that Q-Learning and Policy Gradient explore possible actions by perturbing the actions of a learning agent, ES perturbs the parameters of the ANN controlling the actor's action choice directly. However, the authors stop there and proceed to experimental investigations as to whether ES could solve the RL tasks they were interested in.

Similarly, there is a wealth of papers that establish in parallel an equivalence between a machine learning process and a physical process and between a biological evolutionary process and the same physical process. For instance, both Katsnelson et al. (2018) and Baity-Jesi et al. (2018) draw analogies with glassy systems for biological evolution and ANN training with SGD, respectively; while failing to acknowledge that the field of glassy systems has itself been inspired by NK models developed to explain natural evolution by Kauffman and Levin (1987).

However, to our knowledge, none formalized the direct equivalence between SGD and a class of EAs or connected it to population genetics.

Central Theoretical Results

Gillespie-Orr Evolutionary Algorithms Class

While the Mutational Landscape model of evolution is a general framework, here we will focus on a simplified version that is best suited for optimization tasks and theoretical analysis. Specifically, by noting N is the population size, s is a typical selection coefficient, and μ is the per-site mutation rate, we will be making the following assumptions:

1. Haploid populations (single code evaluated for fitness);
2. Under high selection ($Ns \gg 1$);
3. In the low mutation limit ($N\mu < 1$);

The main purpose of those assumptions is to ensure that a new code modification (mutation) is evaluated by itself and has the time to become universal in the evolving population on its own merits. To enforce it, we define the Gillespie-Orr Evolutionary Algorithms class (GO-EAs) as follows:

Definition 1 (Gillespie-Orr Evolutionary Algorithm class)**.** *Any parameter space search algorithm that evaluates the change in loss function $\mathcal{L}_\theta$ upon update of model parameters*

($\boldsymbol{\theta}$) with a random vector of perturbation ($\mathcal{L}_{\theta+\theta_{rand}}$), without aggregation with other loss function evaluations, and performing a greedy search based on such evaluation.

For simplicity, in the case when model parameters θ are all real numbers, for convenience, we decompose the angular component of update θ_{rand} from its scalar component ϵ and, by abuse of notation, write it as $\epsilon\theta_{rand}$. Given that ϵ indicates how much the model parameters (aka code) can change, we refer to it as the *rewrite capacity*. Conversely, given that θ_{rand} indicates the direction of a potential optimization step, we call it *update vector*. If $\mathcal{L}_{\theta+\epsilon\theta_{rand}} < \mathcal{L}_{\theta}$, we call θ_{rand} a *valid update vector*.

Similarly, the assumptions of the absence of aggregation between different loss function evaluations and haploidy are here for simplicity. In biological evolution, for polyploid sexually reproducing organisms, this assumption is relaxed by only considering mutations that spread throughout the entire population (sweep) and by counting the generations needed for that sweep as a single generation. Given that evolutionary algorithms are not constrained by molecular biology, we abstract this away through our definition.

This means that Scalabe ES (Salimans et al., 2017), CoSyNE (Gomez et al., 2008), NES (Wierstra et al., 2008), or the Genetic Algorithm (Goldberg, 1989) are not part of GO-EA class due to the aggregation of parameters coming from different fitness evaluations. Conversely, a simple greedy search algorithm described in Appendix Alg.1 is part of the GO-EA class, just as the algorithm proposed in Such et al. (2017).

In the Limit, GO-EA Converges to SGD in Mean

Using the standard notation, let $f_{\boldsymbol{\theta}}(\cdot)$ be an ANN parameterized by $\boldsymbol{\theta}$, that maps inputs $\mathbf{X} = \{\mathbf{x_i}\}_{\mathbf{i=1}}^{\mathbf{M}} \in \mathbb{Z}_{\mathbf{2}}^{\mathbf{n_x} \times \mathbf{d_x} \times \mathbf{M}}$ to outputs $\mathbf{Y} = \{\mathbf{y_i}\}_{\mathbf{i=1}}^{\mathbf{M}} \in \mathbb{Z}_{\mathbf{2}}^{\mathbf{n_y} \times \mathbf{d_y} \times \mathbf{M}}$, where $\mathbb{Z}_2 = \{0, 1\}$, d_x and d_y are dimensions of $\boldsymbol{x}$ and $\boldsymbol{y}$, n_x and n_y respectively the binary code length required to describe a single component of the vectors of $\boldsymbol{x}$ and $\boldsymbol{y}$ and M the maximum number of inputs the network can encounter, with potentially $M = \inf$.

Let $\mathcal{L}_{\boldsymbol{\theta}}$ be the fitness function associated to $f_{\boldsymbol{\theta}}$ on the $\mathbf{X}$ and $\mathbf{Y}$. A priori, $\mathcal{L}$ is inaccessible because it requires evaluating all the possible input-output pairs. However, it can be estimated with a finite sample of inputs and outputs $\mathbf{X_{samp}}$, $\mathbf{Y_{samp}}$, giving us an $\hat{\mathcal{L}}_{\boldsymbol{\theta}}|_{\mathbf{X}_{samp}, \mathbf{Y}_{samp}}$, that we will shorten to $\hat{\mathcal{L}}_{\boldsymbol{\theta}}$.

Let $\mathcal{O}$ be a greedy optimization process, such that $\mathcal{O}(\boldsymbol{\theta}) = \boldsymbol{\theta}'$, with a parameter change capacity d, such that $||\boldsymbol{\theta}' - \boldsymbol{\theta}||_p < d$, where $p \in \mathbb{N}$ and $\hat{\mathcal{L}}_{\boldsymbol{\theta}'}|_{\mathbf{X}_{samp}, \mathbf{Y}_{samp}} \geq \hat{\mathcal{L}}_{\boldsymbol{\theta}''}|_{\mathbf{X}_{samp}, \mathbf{Y}_{samp}}$ for any $\boldsymbol{\theta}''$ such that $||\boldsymbol{\theta}'' - \boldsymbol{\theta}||_p < d$.

Theorem 1 (Low learning rate, high population)**.** *GO-EA update converges in mean towards SGD gradient as sampling population increases towards infinity ($N \to \infty$), assuming a locally smooth surface ($\forall \boldsymbol{\theta}_1, \boldsymbol{\theta}_2$, such that $|\boldsymbol{\theta}_1 - \boldsymbol{\theta}'| < l$ and $|\boldsymbol{\theta}_2 - \boldsymbol{\theta}'| < l$, and $\frac{|\hat{\mathcal{L}}_{\boldsymbol{\theta}_1} - \hat{\mathcal{L}}_{\boldsymbol{\theta}_2}|}{|\theta_1 - \theta_2|} < k$ where k is the Lipschitz constant of the loss surface), a non-zero gradient and a low learning rate limit ($lk = d \to 0$).*

Proof. Let $\boldsymbol{\theta}_0$ be the starting parameters, $\boldsymbol{\theta}'$ be the parameter found by a step of SGD, and $\boldsymbol{\theta}''$ be the parameter found by a GO-EA process for a single minibatch of SGD. Given that the loss surface is locally smooth $\nabla\hat{\mathcal{L}}_{\theta_0}$ exists and $\boldsymbol{\theta}' = \boldsymbol{\theta}_0 + l\nabla\hat{\mathcal{L}}_{\theta_0}$, where $|\nabla\hat{\mathcal{L}}_{\theta_0}| = k$ (since $\mathcal{L}$ is fitness, we are looking to ascend the gradient, as opposed to the loss where we look to descend it, and $+$ becomes a $-$). Given the small learning rate approximation, we can ignore higher order terms and $\boldsymbol{\theta}'$ is the argmax of $\hat{\mathcal{L}}_{\theta}$, $\forall\theta$ such that $|\theta - \theta_0| < lk = d$. Because of linearity of the exploration space, $\forall\theta_1, \theta_2$ such that $\hat{\mathcal{L}}_{\theta_2} > \hat{\mathcal{L}}_{\theta_1}$, $|\theta_1 - \theta''| < |\theta_2 - \theta''|$.

Let $\boldsymbol{\theta}''_N$ be the best parameters found by $\mathcal{O}$ with a population N. Because of the above, we have $\forall N > 0$, $|\theta''_{N+1} - \theta'| \leq |\theta''_N - \theta'|$. Similarly, given that $kl = d$, $\forall\epsilon > 0$, $\exists N$ such that $|\theta''_N - \theta'| < \epsilon$. In turn, this implies that $lim_{N \to +\infty} E(|\theta''_N - \theta'|) = 0$, which is a convergence in mean. □

Relaxing Limit Constraints

The low learning limit used above is known as the continuous-time approximation and has been used to obtain several theoretical results regarding SGD learning. However, large learning rates have been shown to be critical for SGD generalization (Xing et al., 2018; Ziyin et al., 2021b) and the infinite population size is neither realistic nor necessary in practice.

The minibatch noise in SGD has been shown to play an essential role in its ability to teach well-generalizing models, even if the noise distribution matters little (Wu et al., 2020). One of the approaches that have been used to emulate the SGD in the case where the model and hardware allow for batches that are too large is artificial noise injection (Zhou et al., 2019; Orvieto et al., 2022), to the point where it is possible to recover generalization properties of SGD with large batches (Geiping et al., 2021).

Assuming that for large batch size, the SGD parameters update vector is close to the one of GD, and assuming that the random sampling process is anisotropic, we can easily calculate the probability of randomly sampling a vector θ_{rand} within an angle α of the GD update vector. If the sampling is uniform, the chance to sample such a vector would be equal to the ratio of the area of the cap of a hypersphere in dimension $n = dim(\theta)$ delimited by the colatitude angle α relative to the whole hypersphere surface area. Fortunately for us, this is a well-known function, mapping to the normalized incomplete beta functions $I_x(a, b) = \frac{B(x;a,b)}{B(a,b)} = \frac{\int_0^x t^{a-1}(1-t)^{b-1}\,dt}{\int_0^1 t^{a-1}(1-t)^{b-1}\,dt}$ (Li, 2011). The final closed form that can be used for the estimation is $f = \frac{1}{2} I_{sin^2\alpha}(\frac{n-1}{2}, \frac{1}{2})$.

Improvement Probability in Transfer Learning

In the context of fine-tuning, we expect to start with a model $f_{\boldsymbol{\theta}_0}(\cdot)$ parameterized so that it already performs well on all the sample tests drawn from the distribution it was used to train with - aka $\forall(\mathbf{X}_{samp}, \mathbf{Y}_{samp}) \subset \mathbf{X} \times \mathbf{Y}$, $\mathbb{P}(\hat{\mathcal{L}}_{\boldsymbol{\theta}_0}|_{\mathbf{X}_{samp},\mathbf{Y}_{samp}} \sim \max_{\boldsymbol{\theta}} \hat{\mathcal{L}}_{\boldsymbol{\theta}}|_{\mathbf{X}_{samp},\mathbf{Y}_{samp}}) \sim 1$. Formally, transfer learning consists in finding a new transfer parametrization $\boldsymbol{\theta}_T$, so that $\forall(\mathbf{X}_{samp}, \mathbf{Y}_{samp}) \subset \mathbf{X} \cup \mathbf{X}' \times \mathbf{Y} \cup \mathbf{Y}'$, $\mathbb{P}(\hat{\mathcal{L}}_{\boldsymbol{\theta}_T}|_{\mathbf{X}_{samp},\mathbf{Y}_{samp}} \sim \max_{\boldsymbol{\theta}} \hat{\mathcal{L}}_{\boldsymbol{\theta}}|_{\mathbf{X}_{samp},\mathbf{Y}_{samp}}) \sim 1$, where the $\mathbf{X}'$ and $\mathbf{Y}'$ are new domains application of the model.

Assuming $|\boldsymbol{X}| >> |\boldsymbol{X}'|$ and $|\boldsymbol{Y}| >> |\boldsymbol{Y}'|$, since otherwise, transfer learning would be equivalent to model retraining, the model is already performing well on the transfer model and the vast majority of the parameters within rewrite capacity d of $\boldsymbol{\theta}_0$ would be deleterious or neutral, meaning that the parametrizations offering improvement would be distributed according to the generalized Pareto distribution (Pickands, 1975; Joyce et al., 2008), which in the case of Gumbel domain of attraction would result in an exponential distribution of fitnesses $\boldsymbol{s} = (s_1, ..., s_{i-1})$ where $s_j = \hat{\mathcal{L}}_{\boldsymbol{\theta}_j}|_{\mathbf{X}_{samp},\mathbf{Y}_{samp}}$, the j^{th} best parametrization of better parametrizations and a probability to reach the better parametrization $\boldsymbol{\theta}_j$ of rank j in the neighborhood from a parametrization $\boldsymbol{\theta}_i$ of the rank i of $\mathbb{P}_{i,j}(\boldsymbol{s}) = \frac{s_j}{\sum_{k=1}^{i-1} s_k}$. In other terms, with finite populations, GO-EA sampling the parametrization neighborhood of the current optimum $\boldsymbol{\theta}_i$ will find advantageous model code rewrites with the probability that is inverse to the exponential probability of the difference between the loss associated to $\boldsymbol{\theta}_i$ and smallest possible loss within the edit distance budget.

Hypotheses based on central results

Overparameterized Setting

Previously, we demonstrated a formula to calculate the chance of random search finding a good approximation of the GD vector. However, if we visualize that function in different dimensions (Appendix Fig.6), we see that coming even within 30 degrees of the GD updates with GO-EA is unrealistic in any dimension above 100. Modern ANNs with thousands of parameters on the lower end would require sampling populations too large for any practical use.

However, most current ANN models are highly overparametrized to stabilize their learning. The overparametrization has been shown to smooth the loss landscape (Li et al., 2018) and connect minima (Nguyen, 2019; Jacot et al., 2018), allowing most strategies performing gradient descent to arrive at an acceptable minimum. Empirical investigations into the update vectors of SGD in this context (Xing et al., 2018) suggest that SGD updates are rarely aligned, can be orthogonal, and often point in opposite directions due to the fact that almost all of the points in the loss landscape model passes through are saddle points.

If this is indeed the case, then numerous vectors are ”valid” in the sense that they correspond to updates that could result from SGD minibatches from the training dataset that would still allow convergence. At this point, random sampling only needs to land close to a ”valid” update vector, effectively decreasing the required sampling population size. Informally, we expect random sample vectors that lead to a lower large batch loss to be close to SGD update components that do not cancel out and are not that rare. In fact, in the most extreme case, when SGD minibatch updates are orthogonal, the curse of the dimension is lifted, and the sampling size required to find a valid update is divided by the number of minibatches. Intuitively, large and diverse datasets applied to overparametrized models will lead to a reasonably fast convergence, even with small search populations.

Because of that, we hypothesize that the training hyperparameters from SGD are directly translatable to the hyperparameters of the GO-EA class algorithm and allow us to train a model with GO-EA while using a relatively small number of samples per step.

We validate this hypothesis by first verifying that for our model ANN training task, SGD update vectors are indeed highly dispersed and showing that a basic GO-EA algorithm (Appendix Alg.1) with a small population can efficiently train a model with hyperparameters copied over directly from SGD.

Minima Flatness as Error Correction Redundancy

The convergence of SGD training to a flat minimum for a model is believed to be one of the conditions for the model training stability (Li et al., 2018). The minima flatness was assumed to be connected to their generalization abilities through the minimal coding length of the model (Hochreiter and Schmidhuber, 1997; Goodfellow and Vinyals, 2015). However, recent evidence argues to the contrary (Dinh et al., 2017; Zhang et al., 2020; Mulayoff and Michaeli, 2020). Within the theory of evolution, the flatness of the fitness peak is commonly associated with the tolerance to the neutral drift - aka error correction capabilities. By using this analogy, we suggest that, just like in the context of the evolution, the flatness of the loss function minimum in ANNs optimized through SGD is determined by the redundancy of the features used by the trained ANN to recognize patterns in the target data.

This intuition seems to be consistent with empirical observations about the loss function minima flatness. Architectures that provide the model with the means to encode redundant features, such as wide hidden layers or skip-forwards connections in deep ConvNets, contribute to making the loss landscape minima flatter (Li et al., 2018). Similarly, drop-out regularization (Srivastava et al., 2014), forcing the ANNs to learn redundant, error-correcting codings, seems to flatten minima as well, along with the smaller batches, which can contain a larger proportion of samples that defy the heuristics that the ANN has learned before encountering them (Goodfellow and Vinyals, 2015). This hypothesis, in particular, goes against the suggestion that for EAs, the search population scales only with latent dimension, (Salimans et al., 2017). To allow robustness through redundant error correction, the actual parameter dimension matters. We validate this intuition experimentally.

Flat Minima and Transfer Learning

Building on the hypothesis presented above, if the minima flatness is indeed related to the classification robustness and error correction, we expect models that learned a variety of error-correcting representations of training data to not be able to transfer those representations without training on the new data.

Intuitively, they rely on parallel redundant subpaths traversing ANN layers to detect redundant relevant features present in the training dataset. With only some of those features present in the dataset on which the transfer task is performed, their error correction property could interfere with the corresponding output without an expected degree of redundant detection. Similarly, we do not expect flatter minima to accelerate transfer learning.

While some minima sharpening is observed experimentally during transfer learning, the results in the model we used are inconclusive, and more investigation is warranted. As for the speed of transfer learning, our experiments suggest flatter minima beyond base performance improvement do not accelerate that due to redundant coding. Experimental results supporting this hypothesis are provided further and in the Appendix.

Update Vector Mixing is Unnecessary

One of the prominent features of most recent Evolutionary Algorithms, ranging from the Genetic Algorithm and NES to the Scalable ES, is the mixing of different vectors that were used to sample the loss landscape to generate a new step. This approach is based on the intuition that vectors found by sampling are approximations of a ”true” empirical gradient descent vector, and by combining them, the empirical gradient can be better approximated.

Our formalization of GO-EA suggests that this is likely not the case. Valid update vectors are unlikely to be aligned and are potentially orthogonal, meaning that averaging them out is counter-productive, in the same way as increasing batch size is in SGD. There is a priori no reason why any interpolation between valid update vectors in EA would be a valid update vector itself, let alone result in a lower loss than either of the valid update vectors. In fact, empirical studies have shown that minima interpolations tend to perform poorly (Li et al., 2018). Similarly, the justification of the Genetic Algorithm’s chromosome cross-over - allowing the beneficial mutations to combine and eliminate deleterious ones - does not apply in the setting where the change of generations is not mandatory, and hence Muller’s Ratchet cannot occur (Lynch et al., 1993). The GO-EA class defined here goes around the problem altogether by evaluating only a single modification to the model parameters at a time. We observe & validate the hypothesis experimentally.

Experimental results

Appendix, supplementary figures, and the code used for experiments presented here are available from the GitHub repository of the project - https://github.com/chiffa/ALIFE2023_GOEA-SGD.

Model Used

In order to perform numerical experiments, we used a convolutional neural network (ConvNet) learning to recognize digits in the MNIST dataset (LeCun et al., 1998). It is a well-established model and a textbook use case for SGD, chosen to minimize the chance of unexpected edge cases interfering with our experimental results. The detailed ANN architecture and hyperparameters are available in the Appendix. To measure the flatness of minima and smoothness of loss landscapes consistently with prior work, we performed a spectral normalization of each layer so that a perturbation along a random axis would correspond to the local robustness of the model. This approach is strictly equivalent to the filter-wise normalized directions proposed in Li et al. (2018).

For clarity, we are using the following abbreviation for hyperparameters and architecture models: batch size as B, drop-out as DO, drop-out on inputs as DI, with two architectural parameters: latent maps in the first layer (LM) - all subsequent follow a predefined ration, and linear width (LW), determining the number of neurons in the hidden linear layer. For flatness experiments, we use three stereotypical settings: ”Robust Wide” (DO:0.25; DI:0.1; LM:16; LW:48; B:4) with high loss landscape smoothness/flat minima; ”Brittle Narrow” (-DO; -DI; LM:4; LW:12; B:128) with low loss landscape smoothness; and ”Brittle Wide” (-DO; -DI; LM:16; LW:48; B:128), added to make sure the instability of Brittle Narrow was not due to the lack of latent features/available feature encoding space. We confirmed the model validity by replicating results from Li et al. (2018), as described in the Appendix, notably the Appendix Fig.13. The ”Robust Wide” model indeed maps to flatter minima, whereas both ”Brittle” models map to sharper minima.

Modeling the transfer learning

We used the transfer learning setting to evaluate the trained model’s generalization ability. We expect that a model that generalizes better would be able to perform better when encountering new data or at least leverage more general features it learned to learn new data faster. We do not expect such abilities from models with simply a more redundant feature encoding, allowing us to differentiate the two in the context of minima flatness.

For that, we performed partial training dataset occlusion, where classes in the training dataset were occluded with respect to the model and loss calculation. In the transfer learning phase, the occluded categories were revealed and included in the loss computation.

SGD Minibatch Noise is High

To evaluate the minibatch noise for different batch sizes, we trained with occlusion the transfer model with default hyperparameters (-DO; -DI; LM:8; LW:12; B:32), froze its parameters and evaluated the angle between updates resulting from different batch sizes, ranging from 1024 to 4, shown on Fig.1. As we hypothesized, the batch noise increases as batch size decreases, with a modal angle of 80 degrees for minibatches of 4 samples. Similarly, large batches (1024) are collinear and approximate GD well. Given that the base model is approximately 10k parameters, 60k samples MNIST training dataset provides approximately 15k different sample batches of 4, suggesting that a small sampling population of 10-100 individuals would be sufficient.

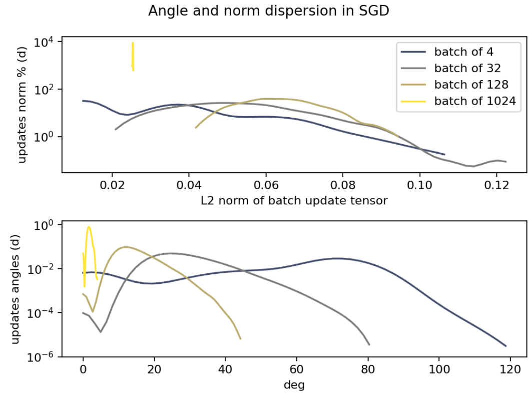

Figure 1: Empirical evaluation of angles between possible SGD update vectors for the same network state and data

GO-EA Trains Efficiently with SGD Hyperparameters

We ported the hyperparameters and model used with the SGD optimizer directly to a simple algorithm in the GO-EA class (Appendix Alg.1) and chose the sampling population size of 20, as per the previous section. The model could train rapidly, achieving an accuracy of over 60% after 800 sampling steps (Fig.2). Further hyperparameter tweaking had little effect on training speed, with the sampling population size having only a moderate effect (Appendix Fig.7), suggesting that SGD hyperparameters did indeed transfer well.

Flat Minima don't Generalize Better,

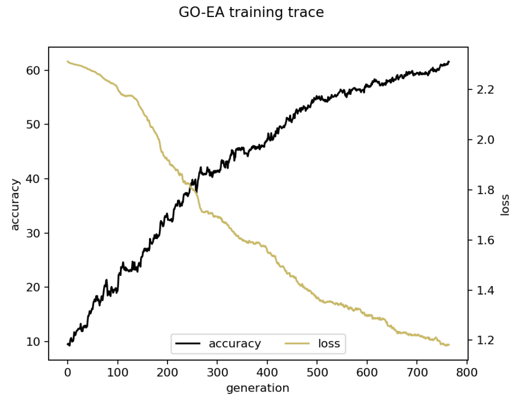

Figure 2: Loss and accuracy of the 9854 parameters ConvNet trained on MNIST dataset with a sampling population of 20 trained by a basic GO-EA class algorithm.

They are More Robust

After training the three minima flatness stereotype models according to the transfer learning model, we saw a significant difference in the flatness of their eight classes minima (Appendix Fig.8). Models with flatter minima do not seem to generalize better (Appendix Fig.15), given that they do not perform better on new data, nor do they undergo transfer learning faster beyond better performance on already known classes. We observe, however, significantly more redundant feature encoding, making them more resilient to noise in input images (Fig.3).

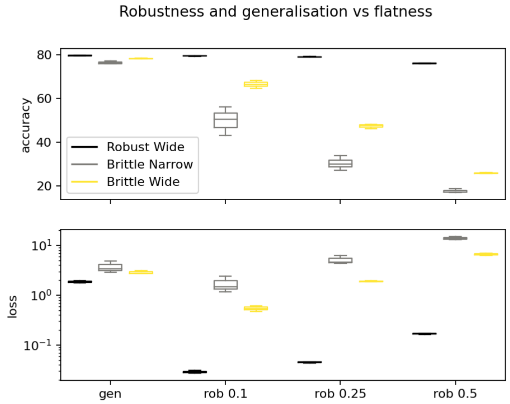

Figure 3: Flatter Minima models are more noise resistant but do not generalize better. Archetype models pre-trained on eight classes with occlusion generalization on the whole dataset (gen) as well as robustness to 10,25, and 50% source image and feature corruption (rob 0.1, 0.25, and 0.5).

.

Transfer Learning Conforms to

Mutational Landscapes Models

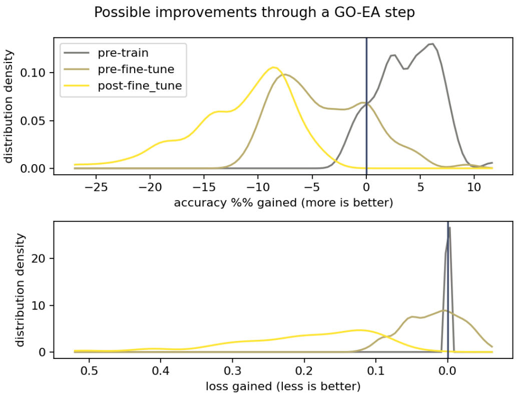

Figure 4: Probability of random update step being beneficial or deleterious as well as the magnitude of effect in accuracy (top panel, right is better) and in loss (bottom panel, left is better)

.

To verify that ANNs trained with SGD conformed to the Mutation Landscape Model during the transfer learning process, we performed a random axis sweep of mutations with an edit distance of the order of magnitude of the standard deviation of the norm of weights in each layer. As shown in Fig.4, it indeed does. We also observe that in conformity to Orr (2005), at random initialization, around 50% of directions result in improvement (assumption of NK models, unrealistic in population genetics). In contrast, at the optimum, all mutations are deleterious, and the start of adaptive burst (start of transfer learning) conforms perfectly to the Mutational Landscapes Model (cf Fig.3 in Orr (2005)).

Update Vector Mixing is Likely Undesirable

In conformity with our hypothesis, valid update vectors sampled by EA are orthogonal and indistinguishable from a pair of random sample vectors with a Kolmogorov-Smirnov test between the two. Fig.5 gives distributions of both in different settings and the p-value of the two-sample KS test for angles between random update vectors and between update vectors resulting in a better loss.

Discussion

In this paper, we define a new class of Evolutionary Algorithms - GO-EA, and establish a formal equivalence between them and SGD, both in the limit and in a more realistic setting. We then empirically test hypotheses arising from such equivalence for several well-established problems in machine learning, namely flat minimas, transfer learning, and applicability of EAs to modern highly overparametrized ML models, demonstrating the advantage of the simpler GO-EA class and partially explaining the past success of GO-EA class algorithms in neuroevolution on hard tasks (Such et al., 2017).

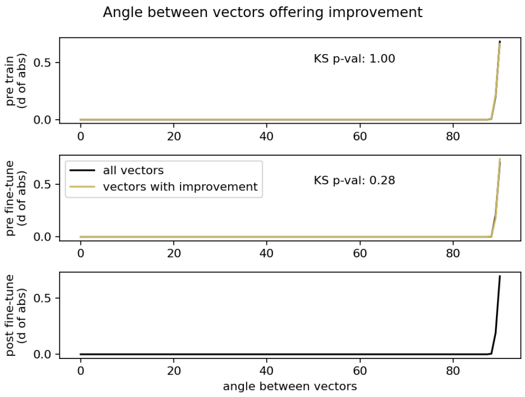

Figure 5: Distribution of angles between randomly sampled update vectors leading to better fitness (yellow) vs. all randomly sampled update vectors (black), from random initialization, transfer learning start point, and at optimum (panels 1-3, respectively)

.

However, this equivalence opens up a possibility to formalize and transfer other insights between machine learning and the theory of evolution. For instance, results by Kucharavy et al. (2018); Tenaillon et al. (2007) suggest that it is possible to evaluate the latent dimension of a problem directly. Conversely, insight into loss landscapes potentially can be translated back to the theory of evolution to explain adaptation, e.g., the feasibility of evolutionary traps to avoid drug resistance (Chen et al., 2015).

Finally, leveraging the existing knowledge regarding SGD in a non-differentiable setting opens up new possibilities for training large models with non-differentiable layers. Specifically, hard attention is generally considered superior to soft attention, which was popularized by the Transformer and is underlying the ubiquitous Large Language Models (LLMs) (Xu et al., 2015; Vaswani et al., 2017). Allowing hard attention in LLMs through GO-EAs has the potential to further the progress in that field.

Acknowledgements

We would like to thank the Cyber-Defence Campus, armasuisse W+T, VBS for the Distinguished CYD Post-Doctoral Fellowship to AK (ARAMIS CYD-F-2021004), as well as Fabien Salvi and France Faille (EFPL) for their technical and administrative support and the anonymous reviewers for their useful feedback.

References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference* *on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Baity-Jesi, M., Sagun, L., Geiger, M., Spigler, S., Arous, G. B., Cammarota, C., LeCun, Y., Wyart, M., and Biroli, G. (2018). Comparing dynamics: Deep neural networks versus glassy systems. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 324–333. PMLR.

Bottou, L. and Bousquet, O. (2007). The tradeoffs of large scale learning. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in Neural Information Processing Systems 20, NeurIPS 2007*, pages 161–168. Curran Associates, Inc.

Bottou, L. and Cun, Y. (2003). Large scale online learning. *Advances in neural information processing systems*, 16.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. cite arxiv:2005.14165Comment: 40+32 pages.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J. T., Sagun, L., and Zecchina, R. (2017). Entropy-sgd: Biasing gradient descent into wide valleys. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Chen, G., Mulla, W. A., Kucharavy, A., Tsai, H.-J., Rubinstein, B., Conkright, J., McCroskey, S., Bradford, W. D., Weems, L., Haug, J. S., et al. (2015). Targeting the adaptability of heterogeneous aneuploids. *Cell*, 160(4):771–784.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. (2017). Sharp minima can generalize for deep nets. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1019–1028. PMLR.

Fisher, R. A. (1930). *The genetical theory of natural selection*. Oxford University Press, Oxford.

Fisher, R. A. and Tippett, L. H. C. (1928). Limiting forms of the frequency distribution of the largest or smallest member of a sample. In *Mathematical proceedings of the Cambridge philosophical society*, volume 24, pages 180–190. Cambridge University Press.

Floreano, D., Dürr, P., and Mattiussi, C. (2008). Neuroevolution: from architectures to learning. *Evol. Intell.*, 1(1):47–62.

Fogel, L., Owens, A., and Walsh, M. (1966). *Artificial intelligence through simulated evolution*. Wiley, Chichester, WS, UK.

Galván, E. and Mooney, P. (2021). Neuroevolution in deep neural networks: Current trends and future challenges. *IEEE Transactions on Artificial Intelligence*, 2(6):476–493.

Ganguli, D., Hernandez, D., Lovitt, L., DasSarma, N., Henighan, T., Jones, A., Joseph, N., Kernion, J., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., Drain, D., Elhage, N., Showk, S. E., Fort, S., Hatfield-Dodds, Z., Johnston, S., Kravec, S., Nanda, N., Ndousse, K., Olsson, C., Amodei, D., Amodei, D., Brown, T. B., Kaplan, J., McCandlish, S., Olah, C., and Clark, J. (2022). Predictability and surprise in large generative models. *CoRR*, abs/2202.07785.

Geiping, J., Goldblum, M., Pope, P. E., Moeller, M., and Goldstein, T. (2021). Stochastic training is not necessary for generalization. *arXiv preprint arXiv:2109.14119*.

Gillespie, J. H. (1983). A simple stochastic gene substitution model. *Theoretical population biology*, 23(2):202–215.

Gillespie, J. H. (1984). Molecular evolution over the mutational landscape. *Evolution*, pages 1116–1129.

Gnedenko, B. (1943). Sur la distribution limite du terme maximum d'une serie aleatoire. *Annals of mathematics*, pages 423–453.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.

Gomez, F. J. and Miikkulainen, R. (1997). Incremental evolution of complex general behavior. *Adapt. Behav.*, 5(3-4):317–342.

Gomez, F. J., Schmidhuber, J., and Miikkulainen, R. (2008). Accelerated neural evolution through cooperatively coevolved synapses. *J. Mach. Learn. Res.*, 9:937–965.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27: NeurIPS 2014*, pages 2672–2680.

Goodfellow, I. J. and Vinyals, O. (2015). Qualitatively characterizing neural network optimization problems. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Gould, S. J. and Lewontin, R. C. (1979). The spandrels of san marco and the panglossian paradigm: a critique of the adaptationist programme. *Proceedings of the royal society of London. Series B. Biological Sciences*, 205(1161):581–598.

Hansen, N., Arnold, D. V., and Auger, A. (2015). Evolution strategies. In Kacprzyk, J. and Pedrycz, W., editors, *Springer Handbook of Computational Intelligence*, Springer Handbooks, pages 871–898. Springer.

Hansen, N. and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.*, 9(2):159–195.

Hochreiter, S. and Schmidhuber, J. (1994). Simplifying neural nets by discovering flat minima. In Tesauro, G., Touretzky, D. S., and Leen, T. K., editors, *Advances in Neural Information Processing Systems 7: [NIPS 1994]*, pages 529–536. MIT Press.

Hochreiter, S. and Schmidhuber, J. (1997). Flat minima. *Neural Comput.*, 9(1):1–42.

Hoffer, E., Hubara, I., and Soudry, D. (2017). Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *Advances in neural information processing systems*, 30.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359 – 366.

Jacot, A., Hongler, C., and Gabriel, F. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: NeurIPS 2018*, pages 8580–8589.

Jastrzebski, S., Kenton, Z., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. (2018). On the relation between the sharpest directions of dnn loss and the sgd step length. *arXiv preprint arXiv:1807.05031*.

Joyce, P., Rokyta, D. R., Beisel, C. J., and Orr, H. A. (2008). A general extreme value theory model for the adaptation of dna sequences under strong selection and weak mutation. *Genetics*, 180(3):1627–1643.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of gans for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Katsnelson, M. I., Wolf, Y. I., and Koonin, E. V. (2018). Towards physical principles of biological evolution. *Physica Scripta*, 93(4):043001.

Kauffman, S. and Levin, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *Journal of theoretical Biology*, 128(1):11–45.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2017). On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Kimura, M. (1968). Genetic variability maintained in a finite population due to mutational production of neutral and nearly neutral isoalleles. *Genetics research*, 11(3):247–270.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kucharavy, A., Rubinstein, B., Zhu, J., and Li, R. (2018). Robustness and evolvability of heterogeneous cell populations. *Molecular biology of the cell*, 29(11):1400–1409.

Lande, R. (1986). The dynamics of peak shifts and the pattern of morphological evolution. *Paleobiology*, 12(4):343–354.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018). Visualizing the loss landscape of neural nets. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: NeurIPS 2018*, pages 6391–6401.

Li, S. (2011). Concise formulas for the area and volume of a hyperspherical cap. *Asian Journal of Mathematics and Statistics*, 4(1):66–70.

Lynch, M., Bürger, R., Butcher, D., and Gabriel, W. (1993). The mutational meltdown in asexual populations. *Journal of Heredity*, 84(5):339–344.

Mnih, V., Heess, N., Graves, A., and Kavukcuoglu, K. (2014). Recurrent models of visual attention. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27: NeurIPS 2014*, pages 2204–2212.

Mulayoff, R. and Michaeli, T. (2020). Unique properties of flat minima in deep networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pages 7108–7118. PMLR.

Nguyen, Q. (2019). On connected sublevel sets in deep learning. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, pages 4790–4799. PMLR.

Nguyen, Q., Bréchet, P., and Mondelli, M. (2021). On connectivity of solutions in deep learning: The role of over-parameterization and feature quality. *CoRR*, abs/2102.09671.

Ohta, T. (1992). The nearly neutral theory of molecular evolution. *Annual review of ecology and systematics*, 23(1):263–286.

Orr, H. A. (2002). The population genetics of adaptation: the adaptation of dna sequences. *Evolution*, 56(7):1317–1330.

Orr, H. A. (2005). The genetic theory of adaptation: a brief history. *Nature Reviews Genetics*, 6(2):119–127.

Orr, H. A. (2006). The distribution of fitness effects among beneficial mutations in fisher's geometric model of adaptation. *Journal of theoretical biology*, 238(2):279–285.

Orvieto, A., Kersting, H., Proske, F., Bach, F. R., and Lucchi, A. (2022). Anticorrelated noise injection for improved generalization. *CoRR*, abs/2202.02831.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359.

Pickands, J. I. (1975). Statistical inference using extreme order statistics. *The Annals of Statistics*, 3(1):119–131.

Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I. (2017). Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T. P., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nat.*, 550(7676):354–359.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Such, F. P., Madhavan, V., Conti, E., Lehman, J., Stanley, K. O., and Clune, J. (2017). Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *CoRR*, abs/1712.06567.

Sutton, R. S. (1991). Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bull.*, 2(4):160–163.

Tenaillon, O. (2014). The utility of fisher's geometric model in evolutionary genetics. *Annual review of ecology, evolution, and systematics*, 45:179–201.

Tenaillon, O., Silander, O. K., Uzan, J.-P., and Chao, L. (2007). Quantifying organismal complexity using a population genetic approach. *PloS one*, 2(2):e217.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, page 5998–6008. Curran Associates, Inc.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gülçehre, Ç., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T. P., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D. (2019). Grandmaster level in starcraft II using multi-agent reinforcement learning. *Nat.*, 575(7782):350–354.

Watkins, C. J. C. H. and Dayan, P. (1992). Technical note q-learning. *Mach. Learn.*, 8:279–292.

Wierstra, D., Schaul, T., Peters, J., and Schmidhuber, J. (2008). Natural evolution strategies. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2008, June 1-6, 2008, Hong Kong, China*, pages 3381–3387. IEEE.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256.

Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding and selection in evolution. *Proceedings of the XI International Congress of Genetics*, 8:209–222.

Wu, J., Hu, W., Xiong, H., Huan, J., Braverman, V., and Zhu, Z. (2020). On the noisy gradient descent that generalizes as SGD. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pages 10367–10376. PMLR.

Xie, Z., Sato, I., and Sugiyama, M. (2021). A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Xing, C., Arpit, D., Tsirigotis, C., and Bengio, Y. (2018). A walk with SGD. *CoRR*, abs/1802.08770.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In Bach, F. R. and Blei, D. M., editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057. JMLR.org.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27: NeurIPS 2014*, pages 3320–3328.

Zhang, C., Bengio, S., Hardt, M., Mozer, M. C., and Singer, Y. (2020). Identity crisis: Memorization and generalization under extreme overparameterization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115.

Zhou, M., Liu, T., Li, Y., Lin, D., Zhou, E., and Zhao, T. (2019). Toward understanding the importance of noise in training neural networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, pages 7594–7602. PMLR.

Zhu, Z., Wu, J., Yu, B., Wu, L., and Ma, J. (2018). The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. *arXiv preprint arXiv:1803.00195*.

Ziyin, L., Li, B., Simon, J. B., and Ueda, M. (2021a). Sgd can converge to local maxima. In *International Conference on Learning Representations*.

Ziyin, L., Liu, K., Mori, T., and Ueda, M. (2021b). Strength of minibatch noise in sgd. *arXiv preprint arXiv:2102.05375*.

Appendix
Evolutionary Algorithms in the Light of SGD: Limit Equivalence, Minima Flatness, and Transfer Learning

This is the appendix to the ”Evolutionary Algorithms in the Light of SGD: Limit Equivalence, Minima Flatness, and Transfer Learning” paper currently in review.

Probability of Randomly Sampling a Vector Close to a Target Vector

Assuming the sampling is performed uniformly on a unit sphere, the probability to sample a vector within α degrees from a desired vector follows a regularized Beta-Incomplete function $\frac{1}{2}I_{sin^2\alpha}(\frac{n-1}{2}, \frac{1}{2})$. To visualize this function, we plotted the probability of a random sampling vector to land within α degrees from a reference vector as a function of angle α and dimension n on the Appendix Fig.6.

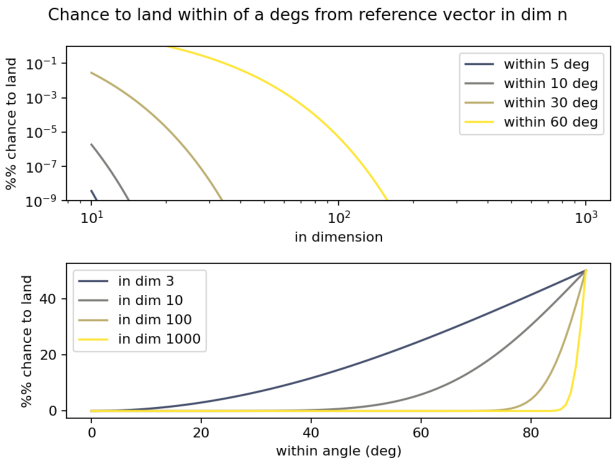

Figure 6: The probability (%) to land within α degrees from the reference vectors as a function of dimensions number for fixed angles and as a function of angles for fixed dimensions

A Trivial Algorithm in the GO-EA Class

A trivial example of an algorithm in GO-EA class is fixed population greedy random search, formally referred to as $(1, \lambda)$-Evolutionary Search in literature and also known as Evolutionary Strategies. The pseudo-code for the exact algorithm we used is presented in Appendix Alg.1. Namely, the version of GO-EA we use does not contain any drift or noise.

Algorithm 1: Fixed population greedy random search

Input: loss function $\mathcal{L}_{\boldsymbol{\theta}}$, starting parameter θ_0
Parameter: update distance ϵ, population size N, generations M
Output: optimal parameters found θ_{fin}

```
Initialize $\theta = \theta_0$
for m in [1, .., M] do
    for n in [1, ..., N] do
        choose random update direction $\theta_{rand}$
        if $\mathcal{L}_{\theta+\epsilon\theta_{rand}} < \mathcal{L}_\theta$ then
            $\theta = \theta + \epsilon\theta_{rand}$.
        end if
    end for
end for
return $\theta_{fin} = \theta$
```

Neural Network Architecture

Our ConvNet has 9854 trainable parameters with specific architecture of 784-**[8C5-P2]-[16C5-P2]-[8C3]-[4C3]-[196L24]-[24L10]**-10. **8C5** denotes a convolution layer with an 8 feature map using a 5x5 filter with stride 1 and padding 2, **P2** is max polling with 2x2 filter and stride 2, and **24L10** is a linear layer mapping from 24 to 10 channels. Every bloc ([]) is followed by a ReLU activation function, and a drop-out normalization is applied to it (10% by default, abbreviated to DO:0.1), except for the last output layer. The input image is partially occluded by using a 10% drop-out on input (DI:0.1). The width of the layers is controlled by two key variables: latent maps in the first layer (LM:8) with subsequent ones following a predefined ratio, and linear width (LW:24), determining the number of neurons in the hidden linear layer.

In SGD mode, the model is trained using cross-entropy loss and SGD optimizer with a learning rate of 0.002 and no momentum, with a batch size of 32 (B:32) and 10 epochs (E:10) unless specified otherwise. The abbreviations above are used in figures, with **x** or **/** denoting the multiplication or division of hyperparameters by a factor (e.g. LWx2) compared to the reference and **:** defining parameter being set to a value (e.g. DO/DI: .0/.0). Finally, **-** means the parameter was set to 0 (e.g. -DO). All figures can be re-generated from code and run records provided, including in greyscale-uniform colormap.

An Simple Algorithm in the GO-EA Is Able to Train a ConvNet On MNIST

Given that optimized, tested, and scalable Evolutionary Algorithms in the GO-EA class have already been described and made available, notably by (Such et al., 2017), here we limited ourselves only to a proof-of-concept implementation. Specifically, due to the experimental infrastructure limitation, forcing a sequential sampling, we are sampling a population of 20 individuals per generation, evaluating the fitness (cross-entropy loss) on the whole training part of the MNIST dataset. Given the lack of optimization of code used to implement an instance of GO-EA class, each population sampling round takes slightly over a minute, leading to a slow - comparatively to SGD - performance.

However, as Fig.2 demonstrates, in about 800 updates, our GO-EA instance (edit distance ED:0.01, population POP:20) is able to take our 9854 parameters ConvNet from 10% accuracy on MNIST to 60% accuracy. In a single epoch with base hyperparameters, SGD performed over 1800 updates with an accuracy of only about 15% on the training set and will need 1800 more to get to 85% in the following epoch, albeit at a lower learning rate. Unlike SGD, our instance of GO-EA did not perform back-propagation, and the population is fully and efficiently scalable through random seed sharing, as explained by (Such et al., 2017).

While increasing the population size is a possible way to increase the speed of convergence, the behavior of other hyperparameters seems to be negligible, as a rapid exploration of parameter

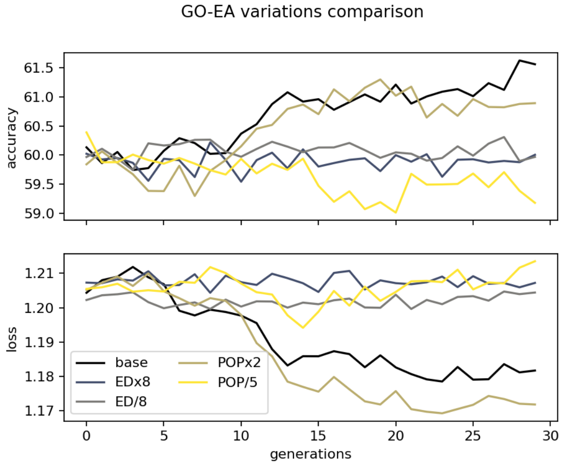

Figure 7: Variation of the speed of convergence of our base ConvNet on MNIST dataset based on the hyperparameters of our implementation of GO-EA class algorithm

space over 30 generations near 60% accuracy suggests in Appendix Fig.7.

Model Validity

The base model we used is able to achieve an error rate below 2% on the validation set after 20 epochs of training and about 2.5% in 10 epochs. Our approach to measuring the ruggedness of loss landscape and minima flatness was consistent with prior measures and recapitulated the results by Li et al. (2018), which we focus on due to its applicability in the non-differentiable setting. Specifically, flatter minima resulted from more overparametrized models, dropout normalization, and smaller batches (cf Appendix Figs. 10, 9, 11). We also confirmed that the total batch number in training was not affecting the minima flatness (cf. Appendix Fig.12).

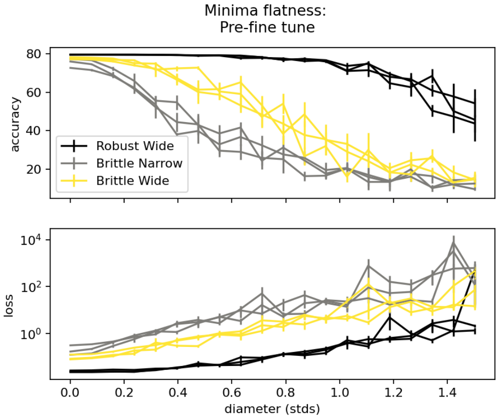

Figure 8: 8 class minima flatness for archetype models. Due to how accuracy is calculated, the best-expected model performance is 80%. Error bars derived from sampling replicas, lines are full independent replicas.

.

Minima Flatness Profiles

In order to obtain minima flatness profiles, several models with identical hyperparameters are trained independently. As described by Li et al. (2018), we measure the minima flatness or the local smoothness by selecting a random vector from an N-dimensional Gaussian distribution with covariance defined by the Identity matrix scaled for the standard deviation of activations in a given layer. Once a vector is selected at random, the model stability is recorded along the axis of its extension, performing a ”sweep” of the fitness landscape in the neighborhood of the model parameter vector θ.

In the diagrams of the minima flatness we present, the x axis is in standard deviations of activations. For completeness, we present the degradation of both loss and accuracy, and for precision, for each parameter vector, we sample at least 5 independent filter-wise normalized random vectors, resulting in error bars when the error at a given distance for them diverges (sampling replicas). Each line corresponds to a completely independent parameter vector from a model trained from scratch with the same hyperparameters (full replicas).

Dropout, Overparametrization, Small Batches, but Not Length of Training Contribute to Flatter Minima

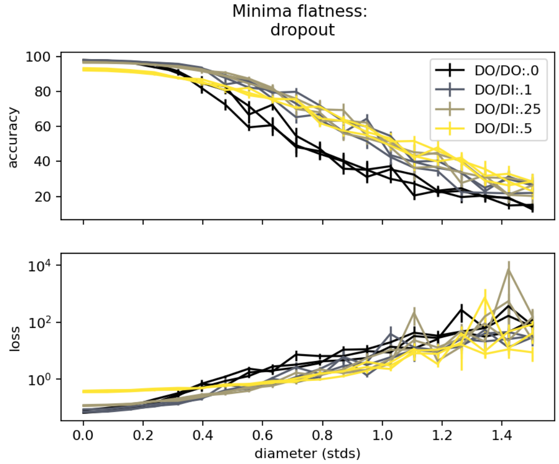

Figure 9: Effect of dropout and input dropout on minima flatness

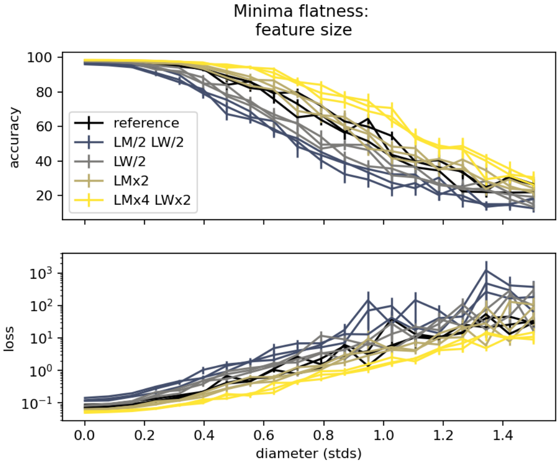

Figure 10: Effect of under- and over- parametrization of the ConvNet on the minima flatness

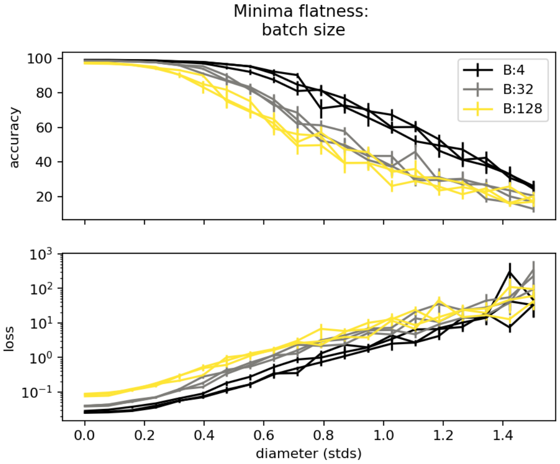

Figure 11: Effect of the small and large batch sizes on the minima flatness

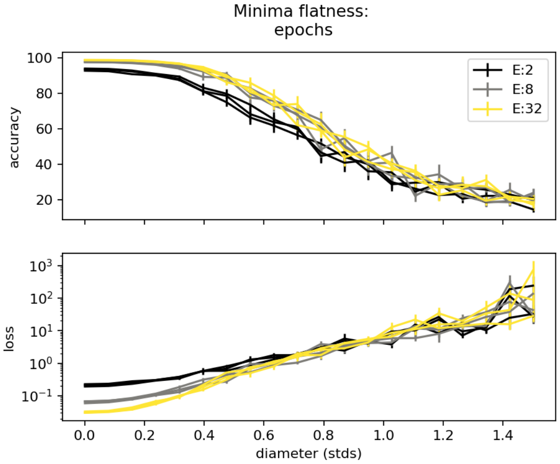

Figure 12: Larger number of updates with smaller batch size cannot explain flatter minima, given the length of training does not affect the minima flatness

Minima Flatness, Error Correction, Generalization, and Transfer Learning

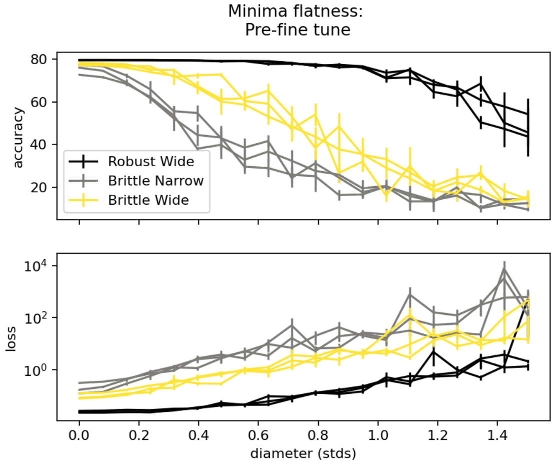

Figure 13: Minima flatness for 3 archetypes before transfer learning

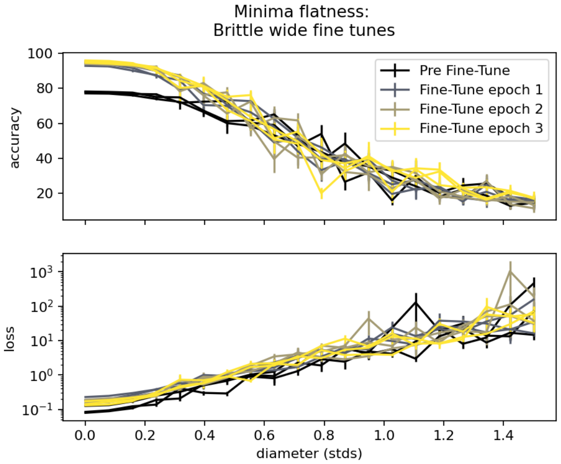

Figure 14: Change in minima flatness for the Brittle Wide archetype during transfer learning

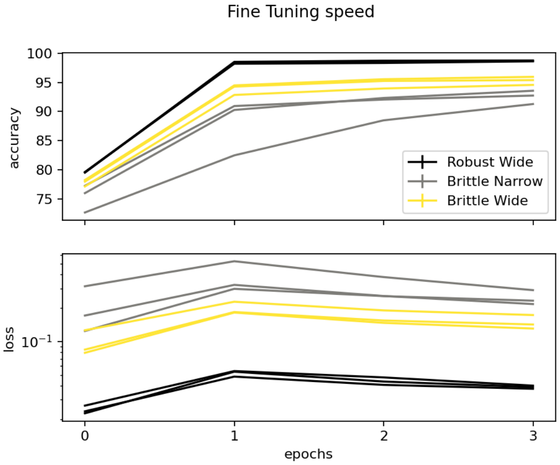

Figure 15: Change in accuracy and loss over epochs of transfer learning for the three archetype models. Minima flatness does not seem to affect transfer learning speed. Each line is a full independent replicate.

Experimental Setup and Computation Time

All the methods are evaluated on a workstation equipped with Intel Core i9-9900K (8 cores/16 threads CPU), 64 Gb of RAM clocked at 2666 MHz, 2 TB NVME M.2 SSD, and an RTX 3080 graphics cards, running an Ubuntu 20.04 LTS distribution. The evaluations were performed within a Docker container, Docker Community Edition, version 20.10.12. The code used Miniconda version 4.12.0 and Python 3.9 with CUDA version 11.3.1. More detailed information is available in the `requirements.txt` found in the code repository shared with this Appendix, https://github.com/chiffa/ALIFE2023_GOEA-SGD.

The total compute time for re-running experiments, except for the GO-EA training trace, is under 2 hours. GO-EA, due to non-optimized deterministic random update vector derivation from seed, shuttles data between CPU and GPU for every individual in every generation. Due to the overhead introduced, the GO-EA trace presented in Fig.2 took approximately 24 hours.

No hyperparameter optimization outside model versions presented here were presented. A common learning rate of 0.02 for training ConvNets on MNIST was divided by 10 to accommodate very small batches, resulting in 0.002. The same rate was also used for GO-EA training.