TRANSFERRING OPTIMALITY ACROSS DATA DISTRIBUTIONS VIA HOMOTOPY METHODS

Anonymous authors
Paper under double-blind review

ABSTRACT

Homotopy methods, also known as continuation methods, are a powerful mathematical tool to efficiently solve various problems in numerical analysis, including complex non-convex optimization problems where no or only little prior knowledge regarding the localization of the solutions is available. In this work, we propose a novel homotopy-based numerical method that can be used to transfer knowledge regarding the localization of an optimum across different task distributions in deep learning applications. We validate the proposed methodology with some empirical evaluations in the regression and classification scenarios, where it shows that superior numerical performance can be achieved in popular deep learning benchmarks, i.e. FashionMNIST, CIFAR-10, and draw connections with the widely used fine-tuning heuristic. In addition, we give more insights on the properties of a general homotopy method when used in combination with Stochastic Gradient Descent by conducting a theoretical analysis in a simplified setting.

1 INTRODUCTION

The use of deep neural networks has led to establish new state-of-the-arts in many applications, i.e. (Śmieja et al., 2018), (Ma et al., 2017), (Shen et al., 2019). Despite their great success and the many theoretical studies that have been published in the last years, i.e. (Balduzzi et al., 2017) and (Li et al., 2018),(Feizi et al., 2018), (Kunin et al., 2019), training these deep models remains a big challenge. Various stochastic optimization algorithms, (Duchi et al., 2011), (Kingma & Ba, 2015), (Reddi et al., 2018), and initialization heuristics, (Daniely et al., 2016), (Hanin & Rolnick, 2018), have been recently suggested in order to improve and speed up the training procedure. *Curriculum learning*, firstly introduced by Bengio et al. (2009) and then extended in different works, i.e. (Graves et al., 2017), (Weinshall et al., 2018), and (Hacohen & Weinshall, 2019), can also be listed among the optimization heuristics proposed to alleviate the complexity of solving such high dimensional and non-convex problems. In particular, taking inspiration from the fact that humans and animals learn "better" when exposed to progressively more complex situations in an organized manner, curriculum learning techniques guide the training by starting with "easy-to-learn" samples and progressively introducing more "complex-to-learn" ones. This guided learning process can also be rephrased as solving a sequence of optimization problems where the target training distribution changes gradually from considering only the "easy" examples to the full original training distribution.

Due to the massive amount of computational resources required by the development of modern deep learning applications, the community has started to explore the possibility of re-using learned weights across different tasks, leading to the development of many new *transfer-learning*, (Rohrbach et al., 2013), (Wang & Schneider, 2014), (Cui et al., 2018), and *meta-learning*, (Finn et al., 2017), (Zintgraf et al., 2019), algorithms. The simplest way to transfer knowledge across different tasks consists in using *warm-start* initialization. This heuristic is amply used in computer vision applications, where it is also known as *fine-tuning* technique (e.g. see (Krizhevsky et al., 2012), (Yosinski et al., 2014), (Reyes et al., 2015) and (Käding et al., 2017)). So far, there is no rigorous explanation on why and when fine-tuning works, but only empirical evaluations on different benchmarks which show that often warm-starting the weights of deep models instead of initializing them randomly leads to faster convergence in terms of number of epochs and better generalization.

The homotopy method (L. Allgower & Kurt, 1980), also known as the continuation method, is a powerful mathematical tool to efficiently solve various problems in numerical analysis (e.g.

see (Tran-Dinh et al., 2012), (Zanelli et al., 2019)). The core idea consists in sequentially solving a series of parametric problems, starting from an easy-to-solve problem and progressively deforming it, via a homotopy function, to the target one. Homotopy methods are suitable to solve complex non-convex optimization problems where no or only little prior knowledge regarding the localization of the solutions is available. In addition, in contrast to state-of-the-art algorithms in deep learning, (Bottou, 2010), (Duchi et al., 2011), (Kingma & Ba, 2015), these methods often achieve global convergence guarantees by only exploiting local structures of the problem. Concepts, such as curriculum-learning and warm-starting, that are related, though to different degrees, to homotopy methods have been proposed both in the deep learning, (Gulcehre et al., 2016), (Mobahi, 2016), (Gulcehre et al., 2017), and in the reinforcement learning, (Narvekar, 2017), communities.

In this work, we propose a novel homotopy-based numerical method to transfer knowledge regarding the localization of an optimum across different task distributions in deep learning. We demonstrate with some empirical evaluations their competitive numerical performance on popular deep learning benchmarks, i.e. FashionMNIST, CIFAR-10, and draw connections with the state-of-the-art fine-tuning heuristic. Finally, we give more rigorous insights on the properties of a general homotopy method by conducting a theoretical analysis in a simplified setting.

2 Background

In this work, we will focus on solving problems of the form

$$\theta^* := \underset{\theta \in \mathbb{R}^d}{\arg\min}\; L(\theta)\,, \tag{1}$$

where $L : \mathbb{R}^d \to \mathbb{R}$ is our target objective function. Problems as described in 1 arise, for instance, in classification and regression scenarios.

In the following section we briefly review the main concepts of *homotopy* and *continuation methods* which the proposed technique to solve problem 1 is based on.

2.1 Homotopic Functions and Continuation Methods for Optimization

Given two topological spaces Z and Y, a *homotopy* is a continuous deformation between two continuous functions $g, f : Z \to Y$ that fulfills certain properties. We can formalize this concept with the following definition

Definition 2.1. *Let $g, f : Z \to Y$ be continuous maps on the topological spaces Z, Y. A homotopy from g to f is a continuous function $H : Z \times [0, 1] \to Y$ such that*

$$H(z, 0) = g(z)\,, \qquad H(z, 1) = f(z)\,, \qquad \forall z \in Z\,. \tag{2}$$

If such H function exists, g is said to be homotopic of f, and this relation is denoted by $g \simeq f$.

It is straightforward to show that, $A \subseteq \mathbb{R}^n$ being a convex set, any two continuous maps $g, f : Z \to A$ are homotopic (see (Suciu, 2016) for a derivation). From this fact it follows that any two continuous and real functions are homotopic. See Figures 4a– 4b in the appendix for a graphical representation of two different homotopy maps between the probability density functions of two Gaussian distributions, where $\lambda \in [0, 1]$ denotes the homotopy parameter. See also Section B in the appendix for details on some of the main properties of homotopic functions.

Continuation methods (also known as homotopy methods) are a widely used mathematical tool to solve complex non-convex optimization problems where no or only very limited prior knowledge regarding the localization of optimal solutions is available (see (L. Allgower & Kurt, 1980) for a full characterization of continuation methods). The core idea of a homotopy approach consists in defining a homotopy function $H(\theta, \lambda)$ with $\lambda \in [0, 1]$ such that $H(\theta, 0) = L_0(\theta)$ is a trivial to optimize smooth map (or a smooth map of which a surrogate θ_0 of an optimal solution is available) and $H(\theta, 1) = L(\theta)$ is our target objective function. Instead of directly addressing problem 1, we approximately and sequentially solve $\gamma > 0$ parametric optimization problems of the form

$$\theta_i^* := \underset{\theta \in \mathbb{R}^d}{\arg\min}\; H(\theta, \lambda_i)\,, \tag{3}$$

for increasing values of the parameter λ for $i = 1, \ldots, \gamma$ and warm-starting each problem with the previously derived approximate solution. Conceptually, Algorithm 1 describes the basic steps of a general homotopy algorithm. Under appropriate assumptions, if the increment $\Delta\lambda$ is sufficiently small, then the iterative procedure in Algorithm 1 will converge to a neighborhood of an optimal solution of the target objective L that depends in some sense on the number of iterations $k > 0$ performed, (L. Allgower & Kurt, 1980). Many different variations of Algorithm 1 exist. In particular, different update schemes for the λ parameter can be adopted (e.g. geometric or sublinear rate of increase), various linear solvers can be used under distinct and specific assumptions, and, finally, also diverse level of approximations for the solutions θ_i^* can be considered, e.g. different k values.

Algorithm 1 A Conceptual Homotopy Algorithm

```
$\theta \leftarrow \theta_0$ such that $\theta_0 \approx \theta_0^* := \arg\min_\theta H(\theta, 0)$      ▷ Initialization of the iterate $\theta$
$\gamma > 0\,, \gamma \in \mathbb{Z}$      ▷ Definition of the number of homotopy steps
$\lambda = 0,\ \Delta\lambda = 1/\gamma$      ▷ Initialization of the homotopy parameters
$k > 0$      ▷ Number of iterations to be performed
for $i = 1, \ldots, \gamma$ do
    $\lambda \leftarrow \lambda + \Delta\lambda$
    procedure $\theta \leftarrow$ITERATIVESOLVER($\theta, k$)      ▷ Approximately solve of $H(\theta, \lambda)$
return $\theta$
```

Before going into the details of two concrete formulations of the conceptual homotopy method outlined in Algorithm 1 (see Section 4), we provide a general theoretical analysis in a simplified setting.

3 Theoretical Analysis

In this section, we provide a theoretical analysis of homotopy methods when Stochastic Gradient Descent (SGD) (Bottou, 2010) is used as iterative solver in Algorithm 1. In particular, we approximately and sequentially solve $\gamma > 0$ unconstrained optimization problems of the form

$$\theta_i^* := \underset{\theta \in \mathbb{R}^d}{\arg\min}\, H(\theta, \lambda_i)\,, \quad \forall i = 1, \ldots, \gamma\,, \tag{4}$$

where $H(\theta, \lambda_i)$ fulfills the assumptions described in Section 3.1 and $\lambda_i \in \Omega_\lambda$. Let θ_i be an approximate solution of the problem associated with parameter λ_i derived by applying $k > 0$ iterations of SGD (in the limit, $k = 1$) and also the starting point for the problem associated with parameter λ_{i+1}, $\forall i \in 1, \ldots, \gamma - 1$. In addition, let θ_0 denote the initial point for the problem associated with λ_1. In this section we derive error bounds for the numerical error $\mathbb{E}\left[||\theta_i - \theta_i^*||^2\right]$ when the changes in the parameter λ are "sufficiently" small.

3.1 Assumptions

We now expose the fundamental assumptions for our theoretical analysis on which all the derivations in Section 3.2 rely.

Assumption 3.1 (smoothness of the objective function)**.** *Assume that $H : \mathbb{R}^d \times \Omega_\lambda \to \mathbb{R}$ is a smooth function w.r.t. both arguments θ and λ.*

Assumption 3.2 (bounded "variance")**.** *Let θ_k denote the value of θ after k iterations of SGD, and $\tilde{g}_k$ denote an unbiased estimate of the gradient $\nabla H(\theta_k, \lambda_i)$ w.r.t. θ_k. Assume that there exists a constant $C > 0$ such that the following bound on the expected squared norm of the estimate of the gradient holds*

$$\mathbb{E}\left[||\tilde{g}_k||^2\right] \leq C^2\,, \quad \forall k\,,\ \forall i = 1, \ldots, \gamma. \tag{5}$$

Remark 3.3. *Assumption 3.2 is standard for proving error bounds on SGD iterates (see (Schmidt, 2014)). In addition, notice that, since*

$$\mathbb{E}\left[||\tilde{g}_k||^2\right] = \mathit{Var}\,(||\tilde{g}_k||) + \mathbb{E}\left[||\tilde{g}_k||\right]^2\,,$$

the C constant is proportional to the variance and the squared expected value of the norm of the gradient estimate. Therefore, it decreases when the iterates approach an optimum and by reducing *the noise in the estimate of the gradient. In the limit (i.e. exact gradient and convergence to an optimum), $C = 0$.*

Assumption 3.4 (strong convexity)**.** *Assume that there exists $\mu > 0$ such that*

$$H(\tilde{\theta}, \lambda) \geq H(\hat{\theta}, \lambda) + \nabla H(\hat{\theta}, \lambda)^\top (\tilde{\theta} - \hat{\theta}) + \frac{\mu}{2} ||\tilde{\theta} - \hat{\theta}||^2 , \quad \forall \tilde{\theta},\, \hat{\theta} \in \mathbb{R}^d,\ \forall \lambda \in \Omega_\lambda . \tag{6}$$

Remark 3.5. *Assumption 3.4 is rather strong when considered globally. Nevertheless, the proposed homotopy-based method enables the exploitation of (usually stronger) local properties of the problem unlike other state-of-the-art methods for optimization. Therefore, with some extra precautions, the analysis we conduct can also be applied locally in a more general context. In this case, the considered assumptions are only required to hold in a neighborhood of an optimal solution, leading to results applicable in the non-convex scenario too. We provide some considerations that can be used to derive more general results in Section G of the appendix, where we relax condition 6 to hold only in a neighborhood of an optimal solution.*

3.2 Derivations

Proposition 3.6. *Given Assumptions 3.1 and 3.4, there exists $\epsilon > 0$ such that $\theta^*(\lambda)$ is Lipschitz continuous over $\mathcal{B}_{\bar{\lambda}}(\epsilon) := \left\{\lambda \text{ s.t. } ||\bar{\lambda} - \lambda|| < \epsilon\right\}$ $\forall \bar{\lambda} \in \Omega_\lambda$, i.e. the following inequality holds*

$$||\theta^*(\lambda) - \theta^*(\bar{\lambda})|| \leq \delta ||\lambda - \bar{\lambda}|| , \quad \forall \lambda \in \mathcal{B}_{\bar{\lambda}}(\epsilon),\ \forall \bar{\lambda} \in \Omega_\lambda. \tag{7}$$

Proof. Proposition 3.6 follows directly from the application of the Implicit Function Theorem (see Lemma 2.1.8 in (L. Allgower & Kurt, 1980)). □

Proposition 3.7. *Let θ_i be the starting point for problem described in equation 1, and let θ_{i+1} denote the iterate after $k > 0$ SGD steps. Under Assumptions 3.2 and 3.4 and by setting the learning rate α to a constant value such that $0 < \alpha < \frac{1}{2\mu}$, the following error bound on the iterates holds*

$$\mathbb{E}\left[||\theta_{i+1} - \theta^*_{i+1}||^2\right] \leq (1 - 2\alpha\mu)^k \cdot \mathbb{E}\left[||\theta_i - \theta^*_{i+1}||^2\right] + \frac{\alpha\, C^2}{2\,\mu} . \tag{8}$$

Proof. See Section E in the appendix. □

Remark 3.8. *The expectation in equation 8 is taken w.r.t. all the random variables, i.e. gradients and initial point θ_0, involved in the optimization procedure up to the current $i+1$ iteration of the algorithm.*

Under the considered assumptions, we now derive the error bounds for the problem associated with parameter λ_{i+1} based on the iterates θ_i and θ_{i+1}. In particular, the goal is to show that, if the approximate solution θ_i for problem with parameter λ_i is sufficiently close to the optimum θ^*_i in expectation, i.e. $\mathbb{E}\left[||\theta_i - \theta^*_i||^2\right] \leq r_\theta^2$, then, for a “sufficiently” small change in the λ_i parameter, $\mathbb{E}\left[||\theta_{i+1} - \theta^*_{i+1}||^2\right] \leq r_\theta^2$. See Figure 9 in the appendix for a graphical representation.

Theorem 3.9. *Consider Algorithm 1 with Stochastic Gradient Descent as solver and let $k > 0$ be the number of iterations and $0 < \alpha < \frac{1}{2\mu}$ be the step size. For $r_\theta \in \mathbb{R}$ such that*

$$r_\theta^2 > \frac{C^2}{4\mu(1 - (1 - 2\alpha\mu)^k)} , \tag{9}$$

then, if $\mathbb{E}\left[||\theta_i - \theta^*_i||^2\right] \leq r_\theta^2$ and $||\lambda_i - \lambda_{i+1}|| \leq \tilde{\epsilon}$, where $\tilde{\epsilon} := \min\left\{\bar{\epsilon}, \epsilon\right\}$ with

$$\bar{\epsilon} = -\frac{r_\theta}{\delta} + \frac{1}{\delta}\sqrt{\frac{r_\theta^2 - \alpha C^2/2\mu}{(1 - 2\alpha\mu)^k}} , \tag{10}$$

the following inequality holds

$$\mathbb{E}\left[||\theta_{i+1} - \theta^*_{i+1}||^2\right] \leq r_\theta^2 . \tag{11}$$

Proof. See Section F is the appendix. □

Theorem 3.9 shows that, under the assumptions listed in Section 3.1, for small variations of the parameter λ_i whose feasible values are inversely proportional to the Lipschitz constant δ of $\theta^*(\lambda)$ (see Proposition 3.6), which describes how fast the optimal solution changes w.r.t. variations in the parameter, it is possible to define balls of radius r_θ around the optimal solutions θ_i^* and θ_{i+1}^* where the iterates θ_i and θ_{i+1} are respectively included in expectation (see Figure 9 in the appendix for a graphical representation). In addition, the lower bound on r_θ is proportional to the noise level C in the estimate of the gradient and inversely proportional to the number of iterations k performed. Consequently, the less noisy the gradient estimate is and the more iterations of SGD are performed, the closer the iterates θ_i and θ_{i+1} will be in expected value to the respective optimal solutions θ_i^* and θ_{i+1}^*. Finally, notice that these results can be applied recursively.

4 Transferring Optimality via Homotopy Methods

In this section we describe a possible application of homotopy methods to solve supervised regression and classification tasks. We address the case where deep neural networks are used as models. We start by introducing the problem framework of supervised learning and then we propose two different homotopy functions for the regression and classification scenarios, respectively.

4.1 Problem Formulation

Despite the generality of the proposed methodology, in this work we specifically address the supervised learning framework, and, in particular, when the predictive model is constituted by a deep neural network $f(x; \theta)$ parameterized by $\theta \in \mathbb{R}^d$.

In the supervised learning scenario, independently from the type of task τ, we typically dispose of a training set $\mathcal{D}_\tau$ consisting of N pairs of examples (x_j, y_j). The goal of the learning process is to find the values of θ that minimize an objective function which measures the discrepancy between the outputs produced by the network $\hat{y} = f(x; \theta)$ and the target outputs y. In particular, the learning process consists in minimizing the empirical objective function

$$L(\theta) := \frac{1}{N} \sum_{(x_j, y_j) \in \mathcal{D}_\tau} \ell(y_j, f(x_j; \theta)) \tag{12}$$

whose non-convexity originates from the high non-convexity of our model f.

In the classical setting, L is chosen based on the KL divergence between the target data distribution $Q_{x,y}$, with density $q_{x,y} = q(y|x)q(x)$, and the learned data distribution $P_{x,y}(\theta)$, with density $p_{x,y} = p(y|x; \theta)q(x)$, where $p(y|x; \theta)$ is modeled via a neural network. This leads to the following form for the objective function

$$L(\theta) = \frac{1}{N} \sum_{(x_j, y_j) \in \mathcal{D}_\tau} q(y|x) \log \frac{q(y|x)}{p(y|x; \theta)} \,. \tag{13}$$

4.2 Homotopy Functions Across Data Distributions

Finding a value of θ that attains a local minimum of the objective function in equation 12 is often a hard optimization task, given the high dimensionality and non-convexity of the problem. In addition, prior knowledge regarding the localization of the solutions is rarely available. The complexity of minimizing such functions also depends on the task distribution $Q_{x,y}$ that is addressed, (e.g. see (Ionescu et al., 2016), (Zendel et al., 2017)). For some tasks, convergence to a good approximate solution is achieved after a few epochs, while for other tasks, orders of magnitude more iterations are required to reach the neighborhood of a solution. In this perspective, different heuristics have been recently introduced in the attempt of re-using the prior knowledge gained from the approximate solution of hard tasks across different data distributions. The question whether we could exploit easy-to-solve or already-solved tasks to speed up and improve the learning of unsolved hard tasks arises. The method we propose in this paper addresses this question and attempts to do so by using a rigorous and well-established mathematical framework, with the goal of speeding up the learning processes in presence of hard-to-solve tasks.

In the perspective of homotopy methods, these goals can be achieved under some assumptions by defining a homotopy transformation between starting and target tasks and by following the procedure described by Algorithm 1. Despite the flexibility and generality of the method, with this work we only focus on homotopy deformations across different task distributions, but similar transformations can be applied in numerous different manners that are also worth exploring, e.g. progressively modifying the architecture of the network or the objective function terms.

Let s be the source task with training data $\mathcal{D}_s$ of pairs $(x_s, y_s) \sim Q_{x_s,y_s}$ whose good approximate solution θ_s^* for the minimization of the objective in 12 is available (or cheaply computable), and let τ denote the target task with training data $\mathcal{D}_\tau$ of pairs $(x_\tau, y_\tau) \sim Q_{x_\tau,y_\tau}$ whose conditional distribution we aim to learn. We propose two different homotopy deformations from task s to task τ for regression and classification, respectively.

4.2.1 Supervised Regression

In the supervised regression scenario, by modeling the density of the conditional learned distribution as $p(y|x;\theta) = \mathcal{N}\left(y; f(x,\theta), \sigma^2 I\right)$ and using the approximate KL divergence objective function described in equation 13, we recover the mean squared error as minimization criterion. The proposed homotopy deformation is based on the following equations

$$y_\lambda|x = \lambda\, y_s|x + (1-\lambda)\, y_\tau|x\,, \tag{14}$$

$$p(y_\lambda|x) = \mathcal{N}(y_\lambda\,;\, f(x;\theta), \sigma^2 I)\,. \tag{15}$$

Notice that the transformation described in equation 14 preserves the unimodality of the conditional distribution (see caption of Figures 4a and 4b), and, when used in combination with the objective function defined in equation 13, leads to the minimization w.r.t. θ of

$$H(\theta,\lambda) := E_{(x,y_\lambda)}\, ||\lambda\ (y_s - f(x;\theta)) + (1-\lambda)\ (y_\tau - f(x;\theta))\,||^2\,. \tag{16}$$

See Figure 5a in the appendix for a graphical representation of this homotopy deformation when applied to gradually transform a one-dimensional sine wave function with a frequency of 1 radian into a one-dimensional sine wave function with a frequency of 137 radians. A downside of this homotopy deformation is that the same support for x is required (the absence of the subscripts s and τ on x stands to indicate that the same realization for x_s and x_τ has to be considered). Alternatively, it is possible to approximate equation 14 by using a Gaussian filter (see Figure 5b and Section C in the appendix).

4.2.2 Supervised Classification

In the case of supervised classification, by modeling the density of the conditional learned distribution as $p(y|x;\theta) = Multinoulli(y; f(x;\theta))$, and using the approximate KL divergence objective function described in equation 13, we recover the cross-entropy loss function. A possible homotopy deformation for the classification case consists in applying the following transformations

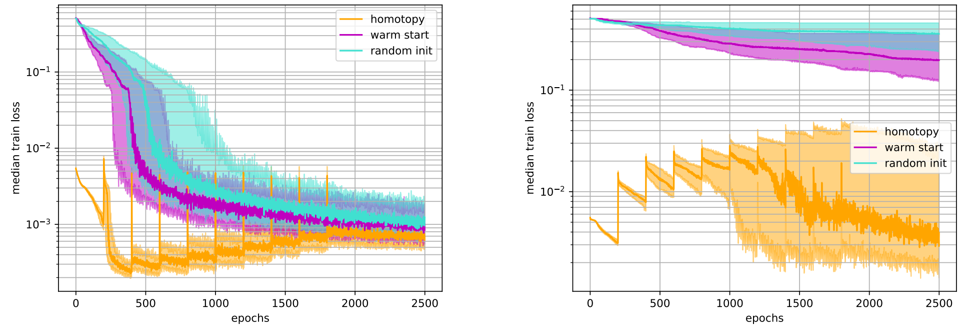
$$x_\lambda = (1-\lambda)\, x_s + \lambda\, x_\tau\,, \tag{17}$$

$$y_\lambda|x_\lambda = (1-\lambda)\, y_s|x_s + \lambda\, y_\tau|x_\tau\,, \tag{18}$$

which corresponds to the use of probabilistic labels. See Figure 6 in the appendix for a graphical representation of the proposed homotopy deformation. The corresponding label vector for the deformed image represented in Figure 6b is $y_{0.5} = [0,0,0.5,0,0,0.5,0,0,0,0]$, given that $\lambda = 0.5$ and that the sampled realizations of x_s and x_τ, represented in Figures 6a and 6c, belong to class 2 and 5 respectively.

5 Experimental Evaluation

In this section, we present some experimental evaluations of homotopy methods when applied to solve supervised regression and classification tasks. As homotopy functions we adopt the ones discussed in Section 4.2. We empirically show that homotopy methods outperform random and warm-start initialization schemes in terms of numerical performance. In particular, when the target task is complex and/or, in the transfer-learning scenario, when the data distributions are significantly different, continuation methods can achieve significant speed-up compared to random and



(a) $\omega = 74$ rad.

(b) $\omega = 137$ rad.

Figure 1: Median train loss across 100 runs versus epochs for sine wave regression tasks with different omega values.

warm-start initializations. We believe that their superior numerical performance relies on the use of homotopy functions that progressively deform the data distribution from an easy-to-solve or already-solved task to the target data distribution. In addition, consistently across all the benchmarks, our homotopy-based method shows faster convergence than random-initialization and faster or comparable convergence than warm-start initialization. When the source task is "similar" to the target one, there is indeed no need to gradually vary the λ parameter in Algorithm 1, but it suffices to directly set it to 1. In this extreme case, our homotopy method boils down to warm-start initialization.

5.1 Regression

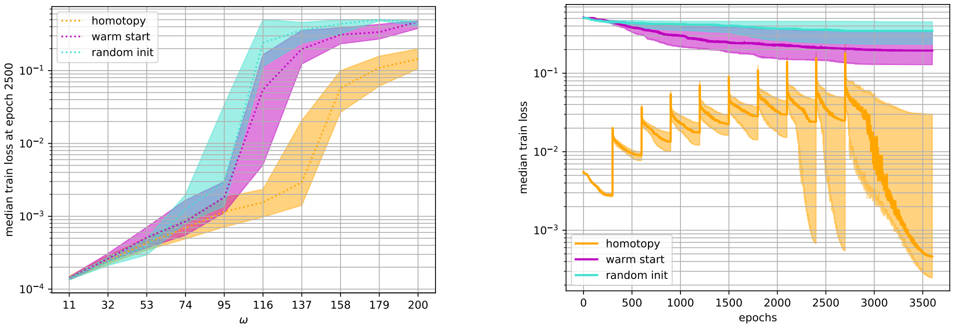
For the supervised regression scenario, the problem we address is how to transfer "optimality knowledge" across two tasks that involve regressing from the input to the output of two different sine wave functions with different values of phase ω. Each considered dataset has 10000 samples split across training and testing, where both x and y are defined as follows

$$x \sim \mathcal{U}(0,1)\,, \qquad y = \sin(\omega x) + \varepsilon\,, \quad \varepsilon \sim \mathcal{N}(0, 0.01)\,. \tag{19}$$

The goal is to start with an "easy-to-learn" task, i.e. $\omega \approx 1$ rad, whose optimum is available by performing only few epochs with a first-order optimizer, e.g. SGD, Adam, and progressively transfer the "optimality knowledge" to a more complex task, i.e $\omega >> 1$ rad, by approximately solving the homotopy problems for increasing values of λ as described in Algorithm 1. We set $\omega = 1$ rad for our source task distribution, and study the performance of the proposed approach with homotopy function as described in equation 14 for different target distributions with $\omega >> 1$ rad. See Figures 7a and 7b in the appendix for a visualization of the starting data distribution with $\omega = 1$ rad and the target data distribution when $\omega = 137$ rad, respectively. The regressor is a feedforward neural network with 6 hidden layers of 100 units each and *relu* as activation function. In order to make the experiments more robust with respect to the choice of the step size α, we use Adam as optimizer. For the experiments in Figures 1a–1b, Figures 8a–8b in the appendix, and Figure 2a, we set $\alpha = 0.001$, $\gamma = 10$, $k = 200$ and then performed an additional 500 epochs on the final target problem, while for the experiments in Figure 2b, we set $\gamma = 10$, $k = 300$ and performed an additional 600 epochs on the final target problem. In this last scenario we set $\alpha = 0.001$ and then decrease it with a cosine annealing schedule to observe convergence to an optimum. As shown in Figures 1a–1b, Figures 8a–8b in the appendix, and Figures 2a and 2b, the homotopy method leads to faster convergence than the considered baselines by preserving the vicinity to an optimal solution for problems $H(\theta, \lambda)$ across the different λ values. In particular, we achieve a training loss up to two orders of magnitude better than the considered baselines.

5.2 Classification

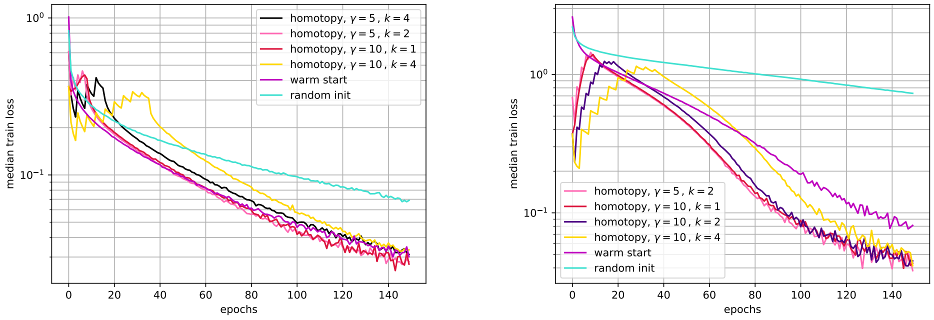
For the supervised classification scenario, we first apply the continuation method with the homotopy deformation described in equation 17 and equation 18 in order to transfer optimality from MNIST, a notoriously "easy-to-learn" task for neural networks, to the FashionMNIST task. Since the two



(a) Median train loss versus omega values.

(b) $\omega = 137$ rad.

Figure 2: Performances of homotopy method, warm start and random initialization on sine wave regression tasks with different omega values. On the left, the median train loss achieved by the different methods after 2500 epochs across 100 runs versus omega values is plotted. On the right, the median train loss across 100 runs versus epochs for final target distribution with $\omega = 137$ rad is plotted. With respect to Figure 1b, in Figure 2b a cosine decay schedule is used for the learning rate, and more epochs are performed to better observe the convergence properties of the different methods.



(a) FashionMNIST.

(b) CIFAR-10.

Figure 3: Median train loss across 10 runs versus epochs for different target task distributions. In both cases, the source task is MNIST.

datasets have the same input dimensionality and the same number of classes, no additional preprocessing of the data is required. As network architecture, we use a VGG-type of network, (Karen & Andrew, 2015), and Adam as optimizer with a step size of $\alpha = 0.001$.

Secondly, we consider CIFAR-10 as target data distribution. Differently from the previous scenario, padding of the MNIST samples is required in order to apply equation 17. Also in this case we adopt a VGG-type of network, (Karen & Andrew, 2015), and Adam as optimizer with a step size of $\alpha = 0.0001$.

As shown in Figures 3a and 3b, in both benchmarks the homotopy method leads to faster convergence than random initialization. While in the second benchmark our method reaches a lower value of training loss in fewer epochs than warm-start, in the MNIST-to-FashionMNIST case the performance is comparable to using warm-start initialization. A possible interpretation is that, when the source and target task distributions are “too similar”, as we hypothesize in the MNIST-to-FashionMNIST scenario, then there is no need for homotopy deformations to be applied, i.e. $0 < \lambda < 1$, but we can directly apply $\lambda = 1$ in our scheme, which corresponds to warm-start initialization.

6 Conclusions

In this paper we propose a new methodology based on homotopy methods in order to transfer knowledge across different task distributions. In particular, our homotopy-based method allows one to exploit easy-to-solve or already-solved learning problems to solve new and complex tasks, by approximately and sequentially solving a sequence of optimization problems where the task distribution is gradually deformed from the source to the target one. We conduct a theoretical analysis of a general homotopy method in a simplified setting, and then we test our method in popular deep learning benchmarks, where it shows superior numerical performance compared to random and warm-start initialization schemes. The proposed framework, in its limiting case, corresponds to the widely used fine-tuning heuristic, allowing for a new and more rigorous interpretation of the latter. Finally, the generality of homotopy methods also opens many novel and promising research directions in fundamental deep learning scenarios such as stochastic non-convex optimization, e.g. by definition of homotopy functions across architectures, objective function terms, and transfer-learning.

References

David Balduzzi, Brian McWilliams, and Tony Butler-Yeoman. Neural taylor approximations: Convergence and exploration in rectifier networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 351–360, 2017. URL `http://proceedings.mlr.press/v70/balduzzi17c.html`.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman (eds.), *ICML*. ACM, 2009. URL `http://dblp.uni-trier.de/db/conf/icml/icml2009.html#BengioLCW09`.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proc. 19th International Computational Statistics*, 2010.

Wanyun Cui, Guangyu Zheng, Zhigiang Shen, Jiang Sihang, and Weil Wang. Transfer learning for sequences via learning to collocate. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/pdf?id=ByldlhAqYQ`.

Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. 2016. URL `https://arxiv.org/abs/1602.05897`.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. 12:2121–2159, 2011.

Soheil Feizi, Hamid Javadi, Jesse Zhang, and David Tse. Porcupine neural networks: Approximating neural network landscapes. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 4831–4841. Curran Associates, Inc., 2018.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1126–1135. PMLR, 2017.

Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 1311–1320, 2017. URL `http://proceedings.mlr.press/v70/graves17a.html`.

Caglar Gulcehre, Marcin Moczulski, Misha Denil, and Y. Bengio. Noisy activation functions. In *International Conference on Learning Representations*, 2016.

Caglar Gulcehre, Marcin Moczulski, Francesco Visin, and Y. Bengio. Mollifying networks. In *International Conference on Learning Representations*, 2017.

Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pp. 2535–2544, 2019. URL `http://proceedings.mlr.press/v97/hacohen19a.html`.

Boris Hanin and David Rolnick. How to start training: The effect of initialization and architecture. In *NeurIPS*, 2018.

R. Tudor Ionescu, Bogdan Alexe, Marius Leordeanu, Marius Popescu, Dim P. Papadopoulos, and Vittorio Ferrari. How hard can it be? estimating the difficulty of visual search in an image. pp. 2157–2166, 2016.

Simonyan Karen and Zisserman Andrew. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *Proc. 3rd International Conference for Learning Representations*, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc., 2012.

Daniel Kunin, Jonathan M. Bloom, Aleksandrina Goeva, and Cotton Seed. Loss landscapes of regularized linear autoencoders. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pp. 3560–3569, 2019. URL `http://proceedings.mlr.press/v97/kunin19a.html`.

Christoph Käding, Erik Rodner, Alexander Freytag, and Joachim Denzler. Fine-tuning deep neural networks in continuous learning scenarios. pp. 588–605, 03 2017. doi: 10.1007/978-3-319-54526-4_43.

Eugene L. Allgower and Georg Kurt. *Numerical continuation methods. An introduction*. Springer–Verlag, 1980.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 6389–6399. Curran Associates, Inc., 2018.

Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 406–416. Curran Associates, Inc., 2017.

Hossein Mobahi. Training recurrent neural networks by diffusion, 2016. arXiv:1601.04114.

Sanmit Narvekar. Curriculum learning in reinforcement learning. In *International Joint Conference on Artificial Intelligence*, 2017.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=ryQu7f-RZ`.

Angie K. Reyes, Juan C. Caicedo, and Jorge E. Camargo. Fine-tuning deep convolutional networks for plant recognition. In *CLEF*, 2015.

Marcus Rohrbach, Sandra Ebert, and Bernt Schiele. Transfer learning in a transductive setting. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 26*, pp. 46–54. Curran Associates, Inc., 2013.

Mark Schmidt. Convergence rate of stochastic gradient with constant step size, July 2014. URL `https://www.cs.ubc.ca/~schmidtm/Documents/2014_Notes_ConstantStepSG.pdf`.

Guohua Shen, Kshitij Dwivedi, Kei Majima, Tomoyasu Horikawa, and Yukiyasu Kamitani. End-to-end deep image reconstruction from human brain activity. In *Front. Comput. Neurosci.*, 2019.

Marek Śmieja, Ł ukasz Struski, Jacek Tabor, Bartosz Zieliński, and Przemysł aw Spurek. Processing of missing data by neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 2719–2729. Curran Associates, Inc., 2018.

I. Alexandru Suciu. Lecture notes in topology, February 2016. URL `www.northeastern.edu/suciu/MATH4565/utop.sp16.html`.

Q. Tran-Dinh, C. Savorgnan, and M. Diehl. Adjoint-based predictor-corrector sequential convex programming for parametric nonlinear optimization. *SIAM J. Optimization*, 22(4):1258–1284, 2012.

Xuezhi Wang and Jeff Schneider. Flexible transfer learning under support and model shift. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 1898–1906. Curran Associates, Inc., 2014.

Daphna Weinshall, Gad Cohen, and Dan Amir. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 5235–5243, 2018. URL `http://proceedings.mlr.press/v80/weinshall18a.html`.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3320–3328. Curran Associates, Inc., 2014.

A. Zanelli, Q. Tran-Dinh, and M. Diehl. Contraction estimates for abstract real-time algorithms for nmpc. 2019.

Oliver Zendel, Katrin Honauer, Markus Murschitz, Martin Humenberger, and Gustavo Fernández Domínguez. Analyzing computer vision data — the good, the bad and the ugly. pp. 6670–6680, 2017.

Luisa M Zintgraf, Kyriacos Shiarlis, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. 2019.

A Appendix

B Properties of Homotopic Functions

Among the numerous properties of homotopic functions, for the formulation of our method (see Section 4.2) we make use of the following properties

Proposition B.1. *Suppose that there exists a homotopy $H : Z \times [0,1] \to Y$ from g to f, i.e. $g \simeq f$. Then*

- $g \simeq g$ *(reflexive property)*
- $g \simeq f \implies f \simeq g$ *(symmetric property)*
- $g \simeq f \quad \text{and} \quad f \simeq h \implies g \simeq h$ *(transitive property)*

Proof. See proof of Theorem 1.5 in (Suciu, 2016). □

Proposition B.2. *Let $g, g' : Z \to Y$ and $f, f' : Y \to W$ be continuous maps, and let $f \circ g,\ f' \circ g' : Z \to W$ be the respective composite maps. If $g \simeq g'$ and $f \simeq f'$, then $f \circ g \simeq f' \circ g'$.*

Proof. See proof of Proposition 1.7 in (Suciu, 2016). □

C Approximation via Gaussian Filter

For the supervised regression, we propose the following homotopy deformation

$$y_\lambda | x = \lambda\, y_s | x + (1 - \lambda)\, y_\tau | x\,. \tag{20}$$

A downside of this homotopy function is that the same support for x is required (the absence of the subscripts s and τ on x stands to indicate that the same realization for x_s and x_τ has to be considered). Alternatively, it is possible to approximate equation 20 by using a Gaussian filter, as depicted in Figure 5b.

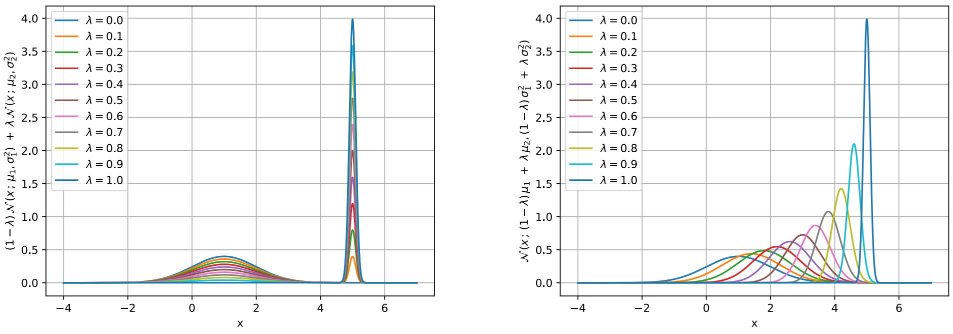
In particular, having sampled one realization j of the pair (x_s, y_s) from the training set $\mathcal{D}_s$, $M > 0$ realizations of the pair (x_τ, y_τ) are sampled from $\mathcal{D}_\tau$. Each $y_{\tau,i}$ realization is then weighted based on the vicinity of $x_{\tau,i}$ to the sampled $x_{s,j}$ realization. This leads to the following approximation of the j realization of y_λ

$$y_{\lambda,j} = (1 - \lambda)\, y_{s,j} + \frac{\lambda}{M} \sum_{i=1}^{M} w_i\, y_{\tau,i}\,, \tag{21}$$

$$w_i = \frac{1}{\sqrt{2\pi\xi^2}} \exp\left(-\frac{||x_{s,j} - x_{\tau,i}||^2}{2\xi^2}\right), \tag{22}$$

where $\xi > 0$ is the standard deviation of the Gaussian filter.

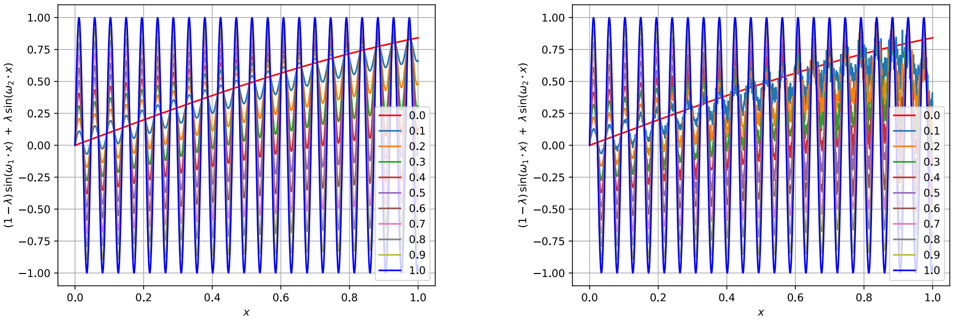
D Additional Figures



(a) Homotopy 1.

(b) Homotopy 2.

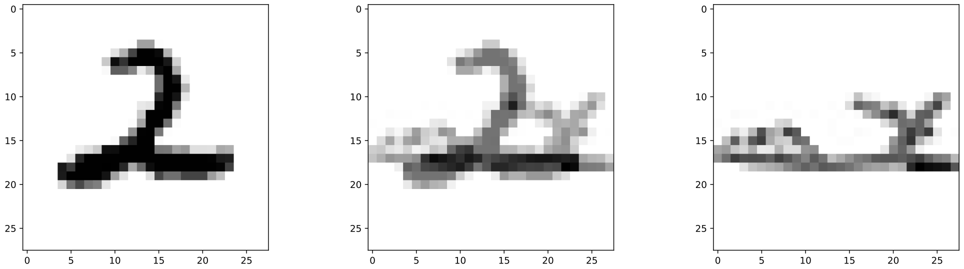
Figure 4: Two different homotopy deformations between the probability density functions of two Gaussian distributions with mean and standard deviation given by $\mu_1 = 1, \sigma_1 = 1$ and $\mu_2 = 5, \sigma_2 = 0.1$ respectively. The homotopy represented Figure 4a results in a mixture of Gaussian distributions, with mixture coefficient given by the homotopy parameter λ. In Figure 4b the deformation concerns instead the parameters μ and σ of the original distributions. Preserving unimodality is a desirable property when the homotopy function is used in combination with a continuation method since, as shown in Figure 4b, the location of the optimum moves together with the function deformation, allowing the optimizer to track it and gradually reach the optimum of the final target task. On the contrary, as shown in Figure 4a, deforming the function does not lead to a gradual shift of the optimal solutions. Consequently, approximately and sequentially solving the problems corresponding to intermediate values of the homotopy parameter λ, i.e. $0 < \lambda < 1$, will not allow the homotopy method to gradually approach the desired final optimal solution.



(a) Homotopy transformation described in equation 14.

(b) Approximation of the homotopy transformation in equation 14 (also equation 20) with a Gaussian filter as described in equation 21 and equation 22.

Figure 5: Graphic representation of the proposed homotopy transformation for the regression scenario when applied to progressively deform a sine wave function with frequency of 1 radian into a sine wave function with frequency of 137 radians.

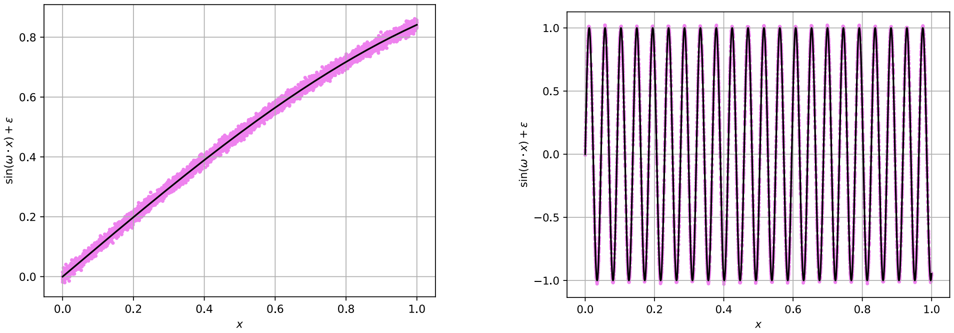


(a) Sampled image of an handwritten digit 2 (class 2) from MNIST dataset.

(b) Homotopy deformation of images 6a and 6c corresponding to $\lambda = 0.5$.

(c) Sampled image of a sandal (class 5) from FahionMNIST dataset.

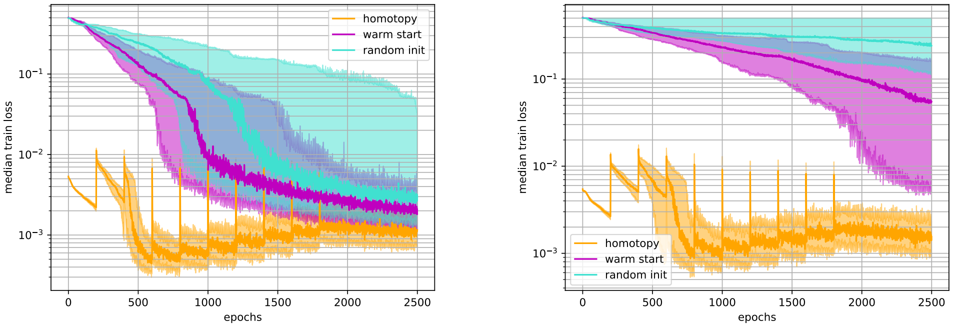
Figure 6: Graphic representation of the homotopy transformation from x_s to x_τ as described in Equation 17 for two sampled images from MNIST and FashionMNIST datasets.



(a) $\omega = 1$ rad.

(b) $\omega = 137$ rad.

Figure 7: Graphical representations of the source data distribution on the left side and the target data distribution corresponding to $\omega = 137$ radians on the right side.



(a) $\omega = 95$ rad.

(b) $\omega = 116$ rad.

Figure 8: Median train loss across 100 runs versus epochs for sine wave regression tasks with different omega values.

E Error Bounds on SGD Iterates

Considering problem 4 with fixed parameter λ_i, we will refer to $\theta_k = \theta_{i,k}$, where we drop the subscript $_i$ in order to simplify the notation. The analysis holds for all fixed parameters λ_i.

Let us use SGD to solve the following optimization problem

$$\arg\min_{\theta} H(\theta, \lambda_i)\,, \tag{23}$$

where the objective function H fulfills Assumptions 3.2 and 3.4. We now derive error bounds on the iterates of SGD

$$\theta_{k+1} = \theta_k - \alpha \tilde{g}_k \,, \tag{24}$$

where $\tilde{g}_k$ is an unbiased estimate of $\nabla H(\theta_k, \lambda_i)$ w.r.t. θ_k and $0 < \alpha < \frac{1}{2\mu}$ is the step size.

We start by applying the definition of SGD iterates

$$\begin{aligned} ||\theta_{k+1} - \theta^*||^2 &\overset{\text{SGD iterate}}{=} ||\theta_k - \alpha\tilde{g}_k - \theta^*||^2 \\ &= ||\theta_k - \theta^*||^2 - 2\alpha\tilde{g}_k^T(\theta_k - \theta^*) + \alpha^2||\tilde{g}_k||^2 \end{aligned} \tag{25}$$

We now take the expectation w.r.t. $\theta_0, \tilde{g}_0, \ldots, \tilde{g}_{k-1}, \tilde{g}_k$ and, considering Assumptions 3.2 and 3.4, we obtain

$$\begin{aligned} \mathbb{E}_{\theta_0,\tilde{g}_0,\ldots,\tilde{g}_{k-1},\tilde{g}_k}\left[||\theta_{k+1} - \theta^*||^2\right] &= \mathbb{E}_{\theta_0,\tilde{g}_0,\ldots,\tilde{g}_{k-1},\tilde{g}_k}\Big[||\theta_k - \theta^*||^2 - 2\alpha\tilde{g}_k^T(\theta_k - \theta^*) \\ &\qquad + \alpha^2||\tilde{g}_k||^2\Big] \\ &\overset{\text{bounded “variance”}}{\leq} \mathbb{E}_{\theta_0,\tilde{g}_0,\ldots,\tilde{g}_{k-1},\tilde{g}_k}\Big[||\theta_k - \theta^*||^2 - 2\alpha\tilde{g}_k^T(\theta_k - \theta^*) \\ &\qquad + \alpha^2C^2\Big] \\ &= \mathbb{E}_{\theta_0,\tilde{g}_0,\ldots,\tilde{g}_{k-1}}\mathbb{E}_{\tilde{g}_k}\Big[||\theta_k - \theta^*||^2 - 2\alpha\tilde{g}_k^T(\theta_k - \theta^*) \\ &\qquad + \alpha^2C^2\Big] \\ &\overset{\text{unbiased } \tilde{g}}{=} \mathbb{E}_{\theta_0,\tilde{g}_0,\ldots,\tilde{g}_{k-1}}\left[||\theta_k - \theta^*||^2 - 2\alpha\nabla H(\theta_k, \lambda_i)^T(\theta_k - \theta^*)\right] \\ &\qquad + \alpha^2C^2 \\ &\overset{\text{strong convexity}}{\leq} (1 - 2\alpha\mu)\cdot\mathbb{E}_{\theta_0,\tilde{g}_0,\ldots,\tilde{g}_{k-1}}\left[||\theta_k - \theta^*||^2\right] + \alpha^2C^2 \,. \end{aligned} \tag{26}$$

Applying this result recursively, we derive the following bound on the error associated with SGD iterates

$$\mathbb{E}_{\theta_0,\tilde{g}_0,\ldots,\tilde{g}_{k-1},\tilde{g}_k}\left[||\theta_{k+1} - \theta^*||^2\right] = (1 - 2\alpha\mu)^k\cdot\mathbb{E}_{\theta_0}\left[||\theta_0 - \theta^*||^2\right] + \frac{\alpha^2C^2}{2\mu} \,. \tag{27}$$

F Proof of Theorem 3.9

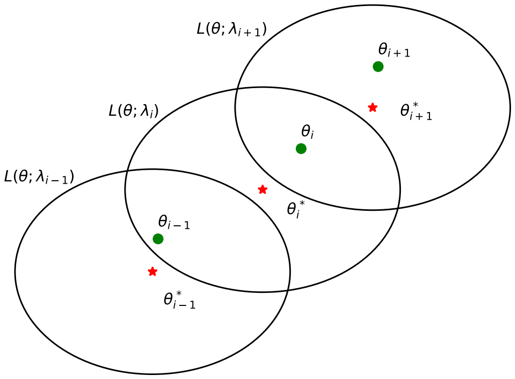


Figure 9: Graphic representation of the results derived in Theorem 3.9

Theorem F.1. *Consider Algorithm 1 with Stochastic Gradient Descent as solver and let $k > 0$ be the number of iterations and $0 < \alpha < \frac{1}{2\mu}$ be the step size. For $r_\theta \in \mathbb{R}$ such that*

$$r_\theta^2 > \frac{C^2}{4\mu(1 - (1 - 2\alpha\mu)^k)} \,, \tag{28}$$

then, if $\mathbb{E}\left[||\theta_i - \theta_i^*||^2\right] \leq r_\theta^2$ *and* $||\lambda_i - \lambda_{i+1}|| \leq \tilde{\epsilon}$*, where* $\tilde{\epsilon} := \min\{\bar{\epsilon}, \epsilon\}$ *with*

$$\bar{\epsilon} = -\frac{r_\theta}{\delta} + \frac{1}{\delta}\sqrt{\frac{r_\theta^2 - \alpha C^2/2\mu}{(1-2\alpha\mu)^k}}\,, \tag{29}$$

the following inequality holds

$$\mathbb{E}\left[||\theta_{i+1} - \theta_{i+1}^*||^2\right] \leq r_\theta^2\,. \tag{30}$$

Proof.

$$\begin{aligned}
\mathbb{E}\left[||\theta_{i+1} - \theta_{i+1}^*||^2\right] &\overset{\text{Ineq. 8}}{\leq} (1-2\alpha\mu)^k \mathbb{E}\left[||\theta_i - \theta_{i+1}^*||^2\right] + \frac{\alpha C^2}{2\mu} \\
&= (1-2\alpha\mu)^k \mathbb{E}\left[||\theta_i - \theta_i^* + \theta_i^* - \theta_{i+1}^*||^2\right] + \frac{\alpha C^2}{2\mu} \\
&\overset{\text{Triangle Ineq.}}{\leq} (1-2\alpha\mu)^k \mathbb{E}\left[\left(||\theta_i - \theta_i^*|| + ||\theta_i^* - \theta_{i+1}^*||\right)^2\right] + \frac{\alpha C^2}{2\mu} \\
&= (1-2\alpha\mu)^k \mathbb{E}\left[\left(||\theta_i - \theta_i^*||^2 + ||\theta_i^* - \theta_{i+1}^*||^2\right.\right. \\
&\qquad \left.\left. +2||\theta_i - \theta_i^*||||\theta_i^* - \theta_{i+1}^*||\right)\right] + \frac{\alpha C^2}{2\mu} \\
&\overset{\text{Proposition 3.6}}{\leq} (1-2\alpha\mu)^k \mathbb{E}\left[\left(||\theta_i - \theta_i^*||^2 + \delta^2||\lambda_i - \lambda_{i+1}||^2\right.\right. \\
&\qquad \left.\left. +2\delta||\theta_i - \theta_i^*||||\lambda_i - \lambda_{i+1}||\right)\right] + \frac{\alpha C^2}{2\mu} \\
&\leq (1-2\alpha\mu)^k \left(\delta^2\tilde{\epsilon}^2 + 2\delta r_\theta \tilde{\epsilon} + r_\theta^2\right) + \frac{\alpha C^2}{2\mu}\,.
\end{aligned}$$

We now solve in $\tilde{\epsilon}$ the following second degree inequality

$$(1-2\alpha\mu)^k \left(\delta^2\tilde{\epsilon}^2 + 2\delta r_\theta \tilde{\epsilon} + r_\theta^2\right) + \frac{\alpha C^2}{2\mu} \leq r_\theta^2\,. \tag{31}$$

The inequality 31 admits solutions if and only if $r_\theta^2 > \frac{\alpha C^2}{2\mu\,(1-(1-2\alpha\mu)^k)}$. Considering the conditions on the learning rate α, then the inequality admits solutions for values of r_θ such that $r_\theta^2 > \frac{C^2}{4\mu^2(1-(1-2\alpha\mu)^k)}$. In particular, inequality 31 holds $\forall\tilde{\epsilon} \in [0, \bar{\epsilon}]$, where $\bar{\epsilon} = -\frac{r_\theta}{\delta} + \frac{1}{\delta}\sqrt{\frac{r_\theta^2 - \alpha C^2/2\mu}{(1-2\alpha\mu)^k}}$. □

G Theoretical Analysis in the Non-Convex Setting

G.1 Realization of the Iterates in the Strongly Convex Region

Considering problem 4 with fixed parameter λ_i, we will refer to $\theta_k = \theta_{i,k}$, where we drop the subscript $_i$ in order to simplify the notation. The analysis holds for all fixed parameters λ_i.

In particular, we address the following optimization problem

$$\arg\min_\theta H(\theta, \lambda_i) := \arg\min_\theta \sum_{j=1}^{N} \ell_j(\theta, \lambda_i)\,. \tag{32}$$

Let assume that strong convexity only holds in a ball of radius B around the optimal solution θ^*, i.e. local strong convexity.

Let also assume that deterministic gradient descent converges linearly with contraction rate $0 < k_d < 1$

$$||\theta_{k+1}^D - \theta^*|| \leq k_d \cdot ||\theta_k - \theta^*||\,, \tag{33}$$

for any θ_k such that $||\theta_k - \theta^*|| \leq B$, and superscript ^{D} denotes iterates obtained by applying the full gradient $g_k := \nabla H(\theta_k, \lambda_i)$

$$\theta^D_{k+1} = \theta_k - \alpha g_k \,. \tag{34}$$

We also introduce the assumption that the norm of each gradient component $\nabla \ell_j$ is upper bounded by a constant ν

$$||\nabla \ell_j(\theta, \lambda_i)|| \leq \nu \,, \qquad \forall j = 1, \ldots, N \,, \tag{35}$$

and $\forall \theta$ s.t. $||\theta - \theta^*|| \leq B$.

Let θ_{k+1} denote the iterate obtained by applying one iteration of stochastic gradient descent

$$\theta_{k+1} = \theta_k - \alpha \tilde{g}_k \,, \tag{36}$$

where $\tilde{g}_k := \frac{1}{M} \sum_{j \in \mathcal{M}} \nabla \ell_j(\theta_k, \lambda_i)$ and $\mathcal{M}$ is a set of $0 < M < N$ indexes randomly sampled from $1, \ldots, N$.

Given any realization of θ_k s.t. $||\theta_k - \theta^*|| < B$ and any realization of $\tilde{g}_k$, we have that

$$\begin{aligned} ||\theta_{k+1} - \theta^*|| &= ||\theta_k - \alpha \tilde{g}_k - \theta^*|| \\ &= ||\theta_k - \alpha g_k + \alpha g_k - \alpha \tilde{g}_k - \theta^*|| \\ &\leq ||\theta_k - \alpha g_k - \theta^*|| + \alpha ||g_k - \tilde{g}_k|| \\ &\leq ||\theta^D_{k+1} - \theta^*|| + \alpha(N - M)\nu \\ &\leq k_d ||\theta_k - \theta^*|| + \alpha(N - M)\nu \,. \end{aligned} \tag{37}$$

Since we have assumed that the current realization of θ_k is in the ball of radius B around the optimal solution θ^*, it directly follows from equation 37 that whenever $\alpha(N - M) \leq \frac{(1-k_d)B}{\nu}$, the realization of θ_{k+1} will also lie in this ball.

These derivations show that when the realization of the current iterate θ_k lies in the ball of radius B around the optimum θ^*, and $\alpha(N - M) \leq \frac{(1-k_d)B}{\nu}$, then the next iterate θ_{k+1} will also lie in the ball. Consequently, if we assume that the initial point θ_0 is in the ball of radius B, then, by applying the derivations recursively, we can show that the iterates will remain in the set where strong convexity holds.

Consequently, we can now apply locally the error bounds on SGD iterates derived in equation 26.

G.2 Generalization of Theorem 3.9 for Locally Strongly Convex Problems

Let θ_i be an approximation of θ^*_i as defined in 4 and assume that the parameter λ_i is subject to a sufficiently small variation, i.e $||\lambda_i - \lambda_{i+1}|| \leq \epsilon$. In order to apply the results derived in Theorem 3.9, given a realization of θ_i, we are interested in showing what the conditions are on $||\theta_i - \theta^*_i||$ such that $||\theta_i - \theta^*_{i+1}|| \leq B$.

Proposition G.1. *Let λ_{i+1} be such that $||\lambda_i - \lambda_{i+1}|| \leq \epsilon$, with $0 \leq \epsilon \leq \frac{B}{\delta}$. If $||\theta_i - \theta^*_i|| \leq B - \delta\epsilon$, then $||\theta_i - \theta^*_{i+1}|| \leq B$.*

Proof.

$$\begin{aligned} ||\theta_i - \theta^*_{i+1}|| &\overset{\text{Triangle Ineq.}}{\leq} ||\theta_i - \theta^*_i|| + ||\theta^*_i - \theta^*_{i+1}|| \\ &\overset{\text{Proposition 3.6}}{\leq} ||\theta_i - \theta^*_i|| + \delta ||\lambda_i - \lambda_{i+1}|| \,. \end{aligned}$$

Finally, using the fact that $||\lambda_i - \lambda_{i+1}|| \leq \epsilon$, it follows that, if $||\theta_i - \theta^*_i|| \leq B - \delta\epsilon$, then $||\theta_i - \theta^*_{i+1}|| \leq B$. □

Given Proposition G.1 and the results from Section G.1, it is straightforward to show that any realization of the iterates will lie in the region around the optimal solution where strong convexity holds by assumption. In particular, the derivations in Theorem 3.9 are directly applicable and allow us to obtain the same results locally also in the non-convex scenario.