The Dynamic Distance Between Learning Tasks:[*]

From Kolmogorov Complexity to Transfer Learning via Quantum Physics and the Information Bottleneck of the Weights of Deep Networks

Alessandro Achille[*] **Glen Bigan Mbeng**[†] **Stefano Soatto**[*]
*University of California, Los Angeles; † SISSA and INFN

Abstract

We compute the transition probability between two learning tasks, and show that it decomposes into two factors. The first depends on the geometry of the loss landscape of a model trained on each task, independent of any particular model used. This is related to a distance function defined using Kolmogorov Complexity, but is insufficient to predict success in transfer learning, as nearby tasks can be unreachable via fine-tuning. The second factor depends on the ease of traversing the path between two tasks. With this dynamic component, we derive strict lower bounds on the complexity necessary to learn a task starting from the solution to another.

1 Introduction and related work

Understanding the geometry of the space of tasks can help predict how difficult it is to transfer a model from one task to another, or across different data domains. This has motivated recent interest in defining distances between classification tasks [13], but there are shortcomings. Architecture independent distances, such as lexicographic distances between label sets in a taxonomy, fail to capture the complex learning dynamics of deep neural networks (DNNs), which can fail in adapting to slight perturbations of the data distributions, even if the task variable remains identical [2]. On the other hand, distances between parametric representations of a task, for instance the weights of DNNs trained on them, fail to capture that very different parameters can represent the exact same posterior distribution. In order to relate to transfer learning, a distance function would have to be asymmetric [12] as it is typically easier to fine-tune a simple task from a complex one than vice-versa.

In this paper, we introduce the notion that, in addition to the geometry, the *dynamics* of the space of tasks is critical to understanding transfer learning.

To show that, we first define a proper (asymmetric) distance between tasks using ideas from Kolmogorov Complexity [8], which is independent of the particular learning algorithm. While such "static distance" gives qualitatively good results in many cases, it does not fully capture problems, particularly of domain adaption, where even nearby tasks may be *unreachable* with fine-tuning [2]. We therefore characterize the probability and expected training time of reaching one task from another, using tools from quantum physics, in particular Kramer's rate theory [7, 5] and the path-integral approach [6]. We then show that such a probability factorizes into two parts. Surprisingly, one turns out to be precisely the static distance we have defined based on Kolmogorov Complexity. The other, which we call *dynamic* distance, depends on the time a stochastic minimization procedure requires to *reach* a task from another. Finally, we verify empirically that the distance we define correlates with the ease of transfer learning.

[*] Published in the 2018 NeurIPS Workshop on Integration of Deep Learning Theories, Dec. 8, 2018. Extended version https://arxiv.org/abs/1810.02440, Oct. 4, 2018. Research supported by ONR N00014-17-1-2072 and ARO W911NF-17-1-0304.

2 Static distance between tasks

In general, we call a *task* any random variable y we want to infer given an observation x and a *training set* $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ of i.i.d. samples x_i and corresponding target values y_i. Given a task, or equivalently the representing dataset $\mathcal{D}$, we may define the Kolmogorov Structure Function (KSF) [8] of the task $\mathcal{D}$ as

$$S_{\mathcal{D}}(t) = \min_{L_{\mathcal{D}}(M) < t} K(M), \tag{1}$$

where the Kolmogorov complexity $K(M)$ denotes the minimum description length of a model M such that the cross-entropy loss $L_{\mathcal{D}}(M)$ obtained by the model M on the dataset $\mathcal{D}$ is less than t.[2] The Lagrangian associated to the minimization problem Equation (1) is

$$C_\beta(M; \mathcal{D}) = L_{\mathcal{D}}(M) + \beta K(M) \tag{2}$$

which measures the cost of encoding $\mathcal{D}$ using the model M when the cost $K(M)$ of encoding the model is discounted by a factor β. Because of this, we refer to $C_\beta(\mathcal{D}) = \min_M C_\beta(M; \mathcal{D})$ as the *complexity of the task $\mathcal{D}$ at level β*. Notice that this quantity is closely related to the Information Bottleneck of the weights studied in [4], to PAC-Bayes theory [11], and to the MDL principle. Moreover, for $\beta = 1$ it reduces to the Evidence Lower Bound (ELBO) of Variational Inference. While in general incomputable, we can upper-bound the term $K(M)$ using a simple encoding scheme for the weights [3]: For a fixed architecture, define the model class $\mathcal{M} = \{q(w) | q(w) \text{ is a normal distribution } N(\mu, \Sigma)\}$, and fix a prior $p(w) \sim N(0, \lambda^2 I)$. The cost of encoding the model $M = q(w) \sim \mathcal{N}(w_0, \Sigma)$ once the prior is fixed, is given by the Kullback-Liebler (KL) divergence

$$K(M) \leq \mathrm{KL}(\, q(w) \,\|\, p(w)\,) = \frac{1}{2}\left[\frac{w_0^2}{\lambda^2} + \frac{1}{\lambda^2}\operatorname{tr}\Sigma + k\log\lambda^2 + \log(|\Sigma|) - k\right],$$

for any choice of $\lambda > 0$. Using this model, we want to upper-bound the complexity $C_\beta(\mathcal{D})$ of the task by minimizing eq. (2). Of course, finding the optimal weights w_0 is far from trivial, as it involves training a deep network on the dataset. However, we can give a description of the optimal Σ for a weight configuration w_0: Assuming w_0 is a local minimum and approximating $C_\beta(q(w); \mathcal{D})$, we obtain the minimizer $\Sigma^* = \frac{\beta}{2}(H + \frac{\beta}{2\lambda^2} I)^{-1}$, which gives the following expression for $C_\beta(M; \mathcal{D})$ as a function of the local minimum w_0:

$$C_\beta(w_0; \mathcal{D}) \leq L_{\mathcal{D}}(w_0) + \frac{\beta}{2}\left[\frac{\|w_0\|^2}{\lambda^2} + \log\left|\frac{2\lambda^2}{\beta} H + I\right|\right], \tag{3}$$

where H is the Hessian of the cross-entropy loss $L_{\mathcal{D}}(w)$ computed in w_0. Notice that this also coincides with the Fisher Information Matrix [10], which gives a link between the KSF and the Fisher Information.

Now that we have a notion of complexity of a task, and a way of upper-bounding it using deep networks, we can use it to define an (asymmetric) distance between tasks: Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two tasks, we define the *reachability* at level β of $\mathcal{D}_1$ from $\mathcal{D}_2$, as

$$R(\mathcal{D}_2 | \mathcal{D}_1) = C_\beta(\mathcal{D}_1 \mathcal{D}_2) - C_\beta(\mathcal{D}_1).$$

where $\mathcal{D}_1\mathcal{D}_2$ denotes the concatenation of the two datasets. Intuitively, $R(\mathcal{D}_2|\mathcal{D}_1)$ measures the additional complexity that we need to learn in order to solve the task $\mathcal{D}_2$ at the required complexity level, assuming we have already learned a solution to the task $\mathcal{D}_1$.

3 The Dynamic Distance between tasks

But how difficult is for a DNN to find a solution to task $\mathcal{D}_2$ starting from task $\mathcal{D}_1$? Consider a network trained with the L_2 regularized loss $U(w) = L_{\mathcal{D}}(w) + \gamma/2\,\|w\|^2$: Sample paths, taken to the continuous limit, evolve according to the stochastic differential equation (SDE) $\dot{w} = f(w) + \sqrt{2D}n(t)$ where $f(w) = \nabla U(w)$, D is a constant and n is the derivative of a Wiener process [9]. Starting from

[2] Note that complexity depends on the level of desired accuracy t. This capture the fact that many tasks are easy to solve approximately, *e.g.*, using a set of simple yet informative features, but are hard to solve exactly.

an initial condition w_0 at time t_0, following the steps detailed in [1, 5], we show that the probability of following a path $w(t)$ starting from w_0 at time t_0 is given, using Stratonovich's convention, by

$$p(w(t)|w_0, t_0) = e^{-\frac{1}{2D}[U(w(t))-U(w(t_0))]} e^{-\frac{1}{2D}\int_{t_0}^{t} \frac{1}{2}[\dot{w}(\tau)^2+f(w)^2]+D \operatorname{div} f(w) d\tau}.$$

Defining the potential $V(w)$ as

$$V(w) = \frac{1}{2}f(w)^2 + D \operatorname{div} f(w) = \frac{1}{2}\nabla U(w)^2 - D\nabla^2 U(w), \tag{4}$$

we can integrate the above path density to obtain the total probability of reaching solution w_f in time $\Delta t = t_f - t_0$ starting from a weight configuration w_0 following any path:

$$p(w_f, t_f|w_0, t_0) = \underbrace{e^{-\frac{1}{2D}[U(w_f)-U(w_0)]}}_{\text{Static potential}} \int_{w_0}^{w_f} \underbrace{e^{-\frac{1}{2D}\int_{t_0}^{t_f} \frac{1}{2}\dot{w}(t)^2+V(w(t))dt}}_{\text{Reachability}} dw(t). \tag{5}$$

The first part is *static* in the sense that it depends only on the initial and final configurations and is independent of the path used to reach it. The second factor measures existence of likely paths $w(t)$ connecting the two points. It is called reachability because, regardless of how large the drop in static potential, the absence of probable paths makes transfer learning unlikely to succeed.

In principle, reachability depends both on the task (*i.e.,* the data), and the architecture. However, we will now show that to a first-order approximation it depends only on information theoretic quantities.

To start, we assume that most paths joining the two points cluster around a few sparse critical paths that are local maxima of the path density function [5]. Given a critical path (which without loss of generality we consider along a coordinate axis u to simplify the notation), we can expand the potential $U(w)$ to second-order around the path as $U(u, \mathbf{v}) = a(u) + \frac{1}{2}\mathbf{v} \cdot \mathcal{H}U(u)\mathbf{v}$, where $\mathcal{H}U(u)$ is the curvature (Hessian) of the loss landscape along the path. Under this approximation, following the derivation in [1], we have two main results: (1) the critical paths follow a deterministic dynamic along an "effective" potential $U_{\mathit{eff}}(w) = U(w) + D \log |\mathcal{H}U(w)|$, where H is the hessian in the point w, and (2) the total probability of reaching the solution following the critical path, or one of its perturbations, is

$$p(w_f, t_f|w_0, t_0) = e^{-\frac{1}{2D}\Delta U_{\mathit{eff}}(w)} \int_{w_0}^{w_f} e^{-\frac{1}{2D}\int_{t_0}^{t_f} \frac{1}{2}\dot{u}(t)^2+V(u(t))dt} du(t). \tag{6}$$

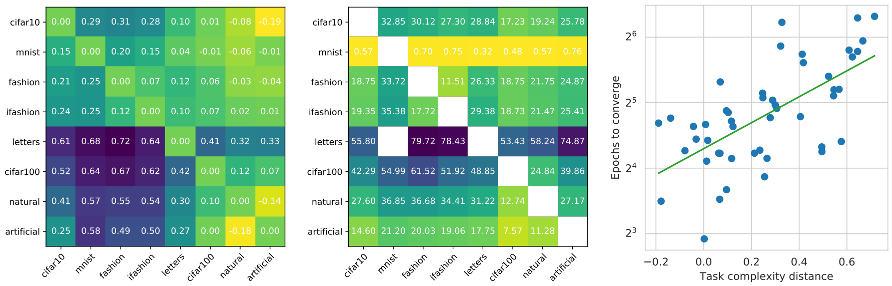
This is critical, as it shows that both the speed and probability of convergence are controlled by the effective potential $U_{\mathit{eff}} = U(w) - D \log |\mathcal{H}U(w)|$, which corrects the original potential by a term that depends on both the *diffusion coefficient* (which scales as $D = k/B$, where B is the batch-size and k is a constant that depends on the architecture), and the *curvature* (determinant of the Hessian) at that point. That is, *to account for reachability, the potential needs to be corrected with the local curvature, and the amount of correction depends on the temperature.* One consequence of this is the often observed fact that sharp minima may not be minima at all for this particular potential when the temperature is sufficiently high (recall that, for a fixed learning rate, the diffusion coefficient scales as $D = k/B$, where B is the batch-size and k is a constant that depends on the architecture). Moreover, this suggests that the dynamic part of the potential can create spurious local minima that can inhibit learning of new problems in a transfer learning scenario [2].

However, eq. (6) still depends on the geometry of the optimization landscape, rather than properties intrinsic to the task. We now connect the curvature to the amount of information needed to solve a task. Using this connection, we are able to characterize the *"learnability"* (reachability) of a task in terms of information-theoretic properties of the data. This completes our program of characterizing the geometry and topology of the space of tasks in a manner that, to first approximation, does not depend on how the task is actually learned.

To establish a link between the curvature $U_{\mathit{eff}}(w) = U(w) + D \log |\mathcal{H}U(w)|$ and the structure function of the task, note that when the network is trained with weight decay, with coefficient γ, the effective potential U_{eff} minimized by the network is given by:

$$U_{\mathit{eff}} = U + D \log |\mathcal{H}U(u)| = L_{\mathcal{D}}(w) + \frac{\gamma}{2}\|w\|^2 + D \log |\gamma I + H(w)|,$$

where $H(w)$ is the hessian of the cross entropy loss $L_{\mathcal{D}}(w)$. By letting $\beta = 2\lambda^2\gamma$ we obtain that the effective potential that affects the network while training with SGD is exactly the complexity



	cifar10	mnist	fashion	ifashion	letters	cifar100	natural	artificial
cifar10	0.00	0.29	0.31	0.28	0.10	0.01	-0.08	-0.19
mnist	0.15	0.00	0.20	0.15	0.04	-0.01	-0.06	-0.01
fashion	0.21	0.25	0.00	0.07	0.12	0.06	-0.03	-0.04
ifashion	0.24	0.25	0.12	0.00	0.10	0.07	0.02	0.01
letters	0.61	0.68	0.72	0.64	0.00	0.41	0.32	0.33
cifar100	0.52	0.64	0.67	0.62	0.42	0.00	0.12	0.07
natural	0.41	0.57	0.55	0.54	0.30	0.10	0.00	-0.14
artificial	0.25	0.58	0.49	0.50	0.27	0.00	-0.18	0.00

	cifar10	mnist	fashion	ifashion	letters	cifar100	natural	artificial
cifar10		32.85	30.12	27.30	28.84	17.23	19.24	25.78
mnist	0.57		0.70	0.75	0.32	0.48	0.57	0.76
fashion	18.75	33.72		11.51	26.33	18.75	21.75	24.87
ifashion	19.35	35.38	17.72		29.38	18.73	21.47	25.41
letters	55.80		79.72	78.43		53.43	58.24	74.87
cifar100	42.29	54.99	61.52	51.92	48.85		24.84	39.86
natural	27.60	36.85	36.68	34.41	31.22	12.74		27.17
artificial	14.60	21.20	20.03	19.06	17.75	7.57	11.28	

Figure 1: **(Left)** Reachability between tasks, based on the relative Kolmogorov complexity. Each element of the matrix shows the time to convergence when finetuning from a pretraining classification task (columns) to a target task (rows). Notice that semantically similar task are close to each other, and that it is easier to go from a complex task to a related simple task than vice-versa. **(Center)** Training epochs necessary to fine-tune from one task (row) to another (column). **(Right)** Scatter plot of the relation between number of steps necessary to converge and the reachability of two datasets.

$C_\beta(w;\mathcal{D})$ of the dataset at level β. Therefore, we may rewrite the first (static) term of the transition probability in eq. (6) as:

$$p(w_f, t_f|w_0, t_0)_{\text{static}} = e^{-\frac{1}{2D}\Delta C_\beta(w;\mathcal{D})}. \tag{7}$$

This has the important implication that the transition probability is upper-bounded by a static part that depends solely on the complexity of the task, or more generally on the difference in complexity between tasks when fine-tuning. To this, however, we must add a dynamic term that also depends on the architecture of the network and the geometry of the loss landscape, and may in general be non-trivial and further reduce the reachability of a task.

From eq. (6) and Equation (7), we can derive the Kramer's convergence rate $1/\tau_K$, which is the expected time of convergence to a minimum, as

$$1/\tau_K = Ce^{-\frac{1}{D}\Delta C_\beta(w;\mathcal{D})}. \tag{8}$$

That is, *the expected time of convergence scales with the difference in complexities between tasks.*

4 Empirical validation

In Figure 1 (Left) we shows for several popular datasets the reachability between tasks computed using the definition in Section 2 and approximated with a ResNet-18 using eq. (3). Notice that this matrix makes intuitive sense: semantically similar tasks are closer to each other, *e.g.,* CIFAR-100 is close to CIFAR-10 and to its two subsets of artificial and natural objects. Similarly, Fashion MNIST (fashion) is close to color inverted Fashion MNIST (ifashion) and to MNIST. Moreover the matrix captures the fact that it is generally easier to learn a task after training on a more complex, related, task (such as going from CIFAR-100 to CIFAR-10), rather than trying to learn a complex task starting from a simple one (*e.g.,* going from MNIST to CIFAR-100).

From eq. (8) and eq. (7) we know that the distance at level β may be compared with the matrix of the time necessary to fine-tune from one task to another (*i.e.*, the training time until we reach some loss threshold), which we show in Figure 1 (Center). In Figure 1 (Right) we show the relation between time to fine-tune and reachability for several pairs of datasets, which again follows the theoretical prediction between the two.

5 Discussion

The ability of deep networks to function for tasks other than those trained on is one of the reasons of their recent widespread diffusion. However, it is very difficult to predict whether such transfer learning will be successful other than just trying it. In this paper we have laid the foundations to enable quantifying the ease of transfer learning. This entails first defining and formally characterizing tasks, and then establishing some sort of topology in the space of tasks. To the best of our knowledge, we are the first to attempt this. We bring to bear tools from diverse fields, from Kolmogorov Complexity to quantum physics, to enable defining and computing sensible notions of distance that correlate with ease of transfer learning. In the process, we discover interesting connections between seemingly disparate concepts: The first is between the notion of task reachability, which we introduce, and the Kolmogorov Structure Function. This in turn is related to information-theoretic treatments of deep learning that have been recently developed [4]. Furthermore, our analysis points to the importance of analyzing the dynamics of learning, rather than just focusing on the asymptotics, which confirms recent empirical discoveries in critical periods and the notion of Information Plasticity [2].

Acknowledgments

Supported by ONR MURI, ARO. The work was done before the first author joined Amazon.

References

[1] A. Achille, G. Mbeng, and S. Soatto. The Dynamics of Differential Learning I: Information-Dynamics and Task Reachability. *ArXiv e-prints*, October 2018.

[2] A. Achille, M. Rovere, and S. Soatto. Critical Learning Periods in Deep Neural Networks. *ArXiv e-prints*, November 2017.

[3] Alessandro Achille, Glen Mbeng, Giovanni Paolini, and Stefano Soatto. Information complexity of tasks, their structure and their distance. Technical Report UCLA CSD: 180003, Department of Computer Science, University of California, Los Angeles, June 2018.

[4] Alessandro Achille and Stefano Soatto. On the Emergence of Invariance and Disentangling in Deep Representations. *ArXiv e-prints*, June 2017.

[5] B. Caroli, C. Caroli, and B. Roulet. Diffusion in a bistable potential: The functional integral approach. *Journal of Statistical Physics*, 26(1):83–111, Sep 1981.

[6] Katharine LC Hunt and John Ross. Path integral solutions of stochastic equations for nonlinear irreversible processes: the uniqueness of the thermodynamic lagrangian. *The Journal of Chemical Physics*, 75(2):976–984, 1981.

[7] H.A. Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(4):284 – 304, 1940.

[8] Ling Li. *Data complexity in machine learning and novel classification algorithms*. PhD thesis, California Institute of Technology, 2006.

[9] Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and adaptive stochastic gradient algorithms. *International Conference on Machine Learning*, page 2101âĂŞ2110, 2017.

[10] James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417, 2015.

[11] David McAllester. A pac-bayesian tutorial with a dropout bound. *arXiv preprint arXiv:1307.2118*, 2013.

[12] Andrea Mennucci. On asymmetric distances. *SNS Technical Report*, 2007.

[13] Amir R Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018.