Curriculum Learning by Optimizing Learning Dynamics

Tianyi Zhou[*]
University of Washington

Shengjie Wang[*]
University of Washington

Jeff A. Bilmes
University of Washington

Abstract

We study a novel curriculum learning scheme where in each round, samples are selected to achieve the greatest progress and fastest learning speed towards the ground-truth on all available samples. Inspired by an analysis of optimization dynamics under gradient flow for both regression and classification, the problem reduces to selecting training samples by a score computed from samples' residual and linear temporal dynamics. It encourages the model to focus on the samples at the learning frontier, i.e., those with large loss but fast learning speed. The scores in discrete time can be estimated via already-available byproducts of training, and thus require negligible extra compute. We discuss the properties and potential advantages of the proposed dynamics optimization via current deep learning theory and empirical studies. By integrating it with cyclical training of neural networks, we introduce "*dynamics-optimized curriculum learning* (**DoCL**)", which selects the training set at each step by a weighted sampling based on the scores. On nine different datasets, DoCL significantly outperforms random mini-batch SGD and recent curriculum learning methods both in terms of efficiency and final performance.

1 Introduction

Effective human learning requires dynamically adjusting one's training contents based on past learning experience and future learning expectations and desires. Most widely-deployed machine learning schemes, on the other hand, use the same static training set identically and repeatedly over numerous optimization epochs, thus lacking any nuanced adjustment to what best should be learnt at any moment. Such adjustments, however, should be highly beneficial for machine learning (ML) since the amount of new information each sample carries can vary drastically at different learning stages. For example, samples with losses close to zero often contribute nearly nothing to the gradients in back-propagation. Although the idea of data subset selection has been studied for a variety of classical machine learning problems, e.g., active learning [45, 3, 12], boosting [43, 16] curriculum learning [5, 31, 59] and machine teaching [28, 63], selection criteria in such cases are often heuristic [18] (e.g., select samples with small losses) and they are designed for specific settings (e.g., convex objectives) and may not be universally applicable.

In this work, we propose to select training sample subsets that most quickly help the predictions for all samples in the training set get close to their targets (Eq. (1)). Unlike previous data selection methods, we directly relate our selection criteria to the changes in the training objective at every step. Specifically, we select samples to maximize the linear dynamics of the model's output along the direction from the current output to its ground truth target at time-t, i.e., the learning speed, in expectation over the data distribution. This provides **a principle formulation of curriculum learning** from which we can derive data selection criteria:

$$\max_{S\subseteq[n],|S|\leq k} \mathbb{E}_{x\sim\mathcal{D}} \left\langle y - f(x), \left.\frac{\partial f(x)}{\partial t}\right|_S \right\rangle. \qquad (1)$$

where $\mathcal{D}$ is the empirical training data distribution, $[n] = \{1, 2, \ldots, n\}$ is the training index set, $f(x)$ is the model's prediction for x, y is the target for $f(x)$, and the linear dynamics $\frac{\partial f(x)}{\partial t}|_S$ can be thought of as the prediction change for x when training on subset S. Following a simple analysis for regression and classification objectives, we reduce the problem to selecting samples with the largest scores in each step, with the score on each sample calculated from its current residual and its linear dynamics under the gradient flow computed on the data distribution $\mathcal{D}$. These two quantities can be estimated directly via byproducts of normal training, and therefore, computing the curriculum of training sets above incurs negligible additional costs.

Our score matches the intuition to always select data at the learning frontier, i.e, hard samples that the model is making the greatest progress on. The first quantity of our score is the

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

sample's prediction residual, and encourages the selection of samples that predict far away from their targets. Similar criteria have been studied in active learning, boosting, and curriculum learning [59, 24]. By focusing on data with large residuals, we do not waste computation on samples that are already learnt. The second quantity is the sample's linear dynamics under the gradient flow over the data distribution, i.e., the learning speed. Empirical deep neural network (DNN) studies [52, 60, 62] show that predictions for some samples remain fixed and correct (i.e., memorized) once learnt, while some samples' predictions frequently change during training and are easier to be forgotten. Moreover, they show that training on the latter minimally impacts the former's predictions, so we can focus training only on the latter for better efficiency. Our score is mathematically derived to be a combination of the two selection criteria formerly motivated by empirical studies and heuristics, and hence bridges the gap between theoretical principles of curriculum design and empirical observations of training dynamics.

We further discuss a natural interpretation of our score achieved when relating it to the neural tangent kernel (NTK) [23, 1, 14]. Its properties in the NTK regime also suggest the feasibility of a lazy update and moving average of the scores. We show that linear dynamics capture the gradient similarity between samples. Intuitively, applying gradient descent on samples with large linear dynamics can effectively reduce losses on many similar samples and may further stabilize their dynamics to reach flat minima. By selecting samples with higher scores, we focus on unlearnt data whose gradients are most consistent with gradients of other data. Hence, the selected samples have significant impacts on the optimization process and by focusing on them we potentially shorten the optimization trajectory.

Based on the selection criterion, we propose a cyclical curriculum learning algorithm, "*Dynamics-optimized curriculum learning* (**DoCL**)", which seamlessly incorporates several other techniques for better performance and efficiency. To evaluate the improvement solely brought by the selection criterion, we present an empirical study without using these techniques. In experiments over nine datasets, DoCL significantly improves the training efficiency and model's generalization performance on test sets compared with random mini-batch SGD and recent curriculum learning methods.

1.1 Related Work

Active learning [45, 53, 3] allows there to be an interaction between machines and (often human) annotators, where the former can iteratively select samples and query their labels from the latter. It aims to reduce the labeled sample complexity (or annotation cost), and therefore it usually prefers the most uncertain/noisy samples [10, 44, 11, 12] — this also, however, can make the learning susceptible to adversarially chosen noise on a small number of samples. Boosting [43, 16], as an ensemble method, aims to compose a strong learner from sequentially trained weak learners, each trained on a weighted dataset that emphasizes the samples found difficult by the predecessors. Machine teaching (MT) [28, 63, 41, 32] focuses on having the learner train only on an extracted "teaching" subset of training data. A recent line of work [33, 34] studies iterative machine teaching (IMT) that allows iterative interactions between the teacher and student via sequential selection of subsets. MT and IMT are different from our problem because: (1) they assume that the teacher knows the optimal model — we do not make this assumption; (2) their objective is to minimize the distance a student model to the optimal one but ours is to speedup the learning dynamics.

Curriculum learning (CL) was first introduced as a method that relied on human experts to determine a training sample order [5, 28, 4, 47] in order to avoid local minima. CL was later extended to strategies that automatically select samples over the course of training using various criteria [50, 49, 51, 18, 19], e.g., hardness [31] or representativeness [25, 59] of samples. However, these criteria might not necessarily be directly related to the original training objective. Some of them suffer from hyperparameter sensitivity, e.g., a threshold on loss values. Although the ultimate goal of CL (i.e., finding an optimal sequence of training samples) is more general than other data selection methods, CL strategies are usually built upon relatively simple heuristics without having a complete mathematical analysis.

In addition, the selection criteria of these methods were developed for various learning settings and hypothesis class assumptions, and thus can sometimes be contradictory. For example, active learning and boosting both favor difficult-to-learn samples, while many CL methods prefer easy-to-learn samples [26, 31]. Although selection criteria are often partially adaptive to per-sample feedback during training, they are not designed to directly accelerate the learning process, as we do in this paper. Some recent work [27, 15] resorts to an additional model to directly generate selection results but they require training another model using non-stationary feedback from the ongoing training process, e.g., via reinforcement learning, which might be more challenging and costly to solve than the original problem.

A line of recent research [48, 54] has studied accelerated optimization dynamics derived from discretized Lagrangian/Hamiltonian dynamics of a model, showing optimal convergence rates. By doing so, they recover a class of accelerated optimization schemes and even generate new ones. These approaches mainly focus on convex optimization. The major difference with our work is that we optimize the dynamics of a model's output on individual samples (vs. on model parameters) by changing the training set (vs. by choosing kinetic energy function, scaling conditions and discretization). In addition, they optimize the total energy along the optimization trajectory, which might be an objective worth studying for our problem in the future.

2 Optimizing Training Dynamics

The ultimate goal of curriculum learning is to find an optimal sequence of training samples that will lead to faster training progress, lower training computation, and better generalization performance. It is, however, prohibitively expensive to directly search for the optimal sequence since the set of candidates grows exponentially with nT, where n is the training set size and T is the number of training epochs. In this section, we reduce curriculum learning to optimizing the per-step training dynamics for samples drawn from the data distribution $\mathcal{D}$. Specifically, at step t, we show how to select a subset $S_t \subseteq [n]$ leading to f making the greatest progress towards the ground truth y in expectation for $x \sim \mathcal{D}$ as per Eq. (1). By relating it to a simple analysis of training dynamics, we will show that the problem can be efficiently solved using only pre-existing byproducts of training, as mentioned above. For simplicity, we remove all subscripts denoting the time step, for example, we use S for S_t.

2.1 Problem Formulation

We first consider a regression task that aims to learn a prediction model $f(x;\theta)$ by minimizing the expected ℓ_2 loss $\ell(y, f(x;\theta))$ for x drawn from the data distribution $\mathcal{D}$, i.e.,

$$\min_{\theta} \mathbb{E}_{x\sim\mathcal{D}} \ \ell(y, f(x;\theta)) \triangleq \frac{1}{2}\|y - f(x;\theta)\|_2^2. \quad (2)$$

In the following, we will use simplified notations: we will use $f(x)$ and $\ell(x)$ to denote $f(x;\theta)$ and $\ell(y, f(x;\theta))$, respectively. Under the gradient flow (continuous-time gradient descent) computed on a subset $S \subseteq [n]$, we have $\frac{\partial\theta}{\partial t}\big|_S = -\sum_{i\in S}\frac{\partial \ell(x_i)}{\partial \theta} = \sum_{i\in S} -\frac{\partial \ell(x_i)}{\partial f(x_i)} \cdot \frac{\partial f(x_i)}{\partial \theta}$. The linear dynamics of the model's output $f(x)$ for any sample x can be represented as

$$\frac{\partial f(x)}{\partial t}\Big|_S = \frac{\partial f(x)}{\partial \theta}\cdot\frac{\partial \theta}{\partial t}\Big|_S = \frac{\partial f(x)}{\partial \theta}\cdot\sum_{i\in S} -\frac{\partial \ell(x_i)}{\partial f(x_i)}\cdot\frac{\partial f(x_i)}{\partial \theta}. \quad (3)$$

For this section, we always assume that the optimization is performed in the continuous time domain, so the gradients, chain-rules and integration are all well-defined, and the derivation holds rigorously. We will discuss the discretization in Sec. 4 when we need to estimate the continuous-time quantities in an algorithmic implementation.

At step t, we aim to find a subset $S \subseteq [n]$ of size $|S| \leq k$ whose gradient flow maximizes the projection of residual $y - f(x)$ on the dynamics $\frac{\partial f(x)}{\partial t}\big|_S$ for all $x \sim \mathcal{D}$, i.e.,

$$\max_{S\subseteq[n],|S|\leq k} \mathbb{E}_{x\sim\mathcal{D}} \left\langle y - f(x), \frac{\partial f(x)}{\partial t}\Big|_S \right\rangle. \quad (4)$$

Intuitively, the goal is to maximize the momentum of each sample's prediction $f(x)$ moving towards its target y. The dynamics $\partial f(x)/\partial t$ are weighted by their residuals $y - f(x)$ so we achieve faster decreasing loss for samples with larger residual. Thereby, predictions of different samples ideally can reach their targets at the same time without overshooting. The objective maximizes the dynamics of decreasing the objective in Eq. (2), i.e., $\frac{\partial \mathbb{E}_{x\sim\mathcal{D}}\, -\ell(y,f(x;\theta_t))}{\partial t} = \mathbb{E}_{x\sim\mathcal{D}}\left\langle y - f(x;\theta_t), \frac{\partial f(x;\theta_t)}{\partial t}\Big|_S\right\rangle$.

One can think that we break down the original problem in Eq. (2) into a sequence of sub-problems in the form of Eq. (4) over time steps. To verify this, we can integrate the objective in Eq. (4) over time from $t = 0$ to T, which recovers the objective in Eq. (2) (negated, up to a constant):

$$\begin{aligned}&\int_0^T \left\langle y - f(x;\theta_t), \frac{\partial f(x;\theta_t)}{\partial t}\right\rangle dt \\ =&\frac{1}{2}\left(\|y - f(x;\theta_T)\|_2^2 - \|y - f(x;\theta_0)\|_2^2\right). \end{aligned} \quad (5)$$

Since the gradients $\partial \ell(x_i)/\partial\theta$ computed on different samples might have conflicts and cancel out with each other if selected in the same training batch, compared to uniform sampling S, maximizing the dynamics encourages selecting samples with consistent gradients that decrease the expected risk/loss over the data distribution.

2.2 Regression

To optimize Eq. (4), we approximate the expected momentum w.r.t. $x \sim \mathcal{D}$ in Eq. (4) by averaging over a finite number of samples D drawn from the data distribution $\mathcal{D}$, i.e.,

$$\begin{aligned}&\mathbb{E}_{x\sim\mathcal{D}}\left\langle y - f(x), \frac{\partial f(x)}{\partial t}\Big|_S\right\rangle \\ \approx&\frac{1}{|D|}\sum_{x\in D}[y - f(x)]^T\frac{\partial f(x)}{\partial \theta}\cdot\sum_{i\in S} -\frac{\partial \ell(x_i)}{\partial f(x_i)}\cdot\frac{\partial f(x_i)}{\partial \theta} \\ =&\frac{1}{|D|}\sum_{i\in S} -\left[\frac{\partial \ell(x_i)}{\partial f(x_i)}\right]^T\cdot\sum_{x\in D}\frac{\partial f(x_i)}{\partial \theta}\left[\frac{\partial f(x)}{\partial \theta}\right]^T[y - f(x)] \\ =&\frac{1}{|D|}\sum_{i\in S}\left[\frac{\partial \ell(x_i)}{\partial f(x_i)}\right]^T\frac{\partial f(x_i)}{\partial \theta}\cdot\sum_{x\in D}\left[\frac{\partial f(x)}{\partial \theta}\right]^T\frac{\partial \ell(x)}{\partial f(x)} \\ =&\frac{1}{|D|}\sum_{i\in S}\left[\frac{\partial \ell(x_i)}{\partial f(x_i)}\right]^T\frac{\partial f(x_i)}{\partial \theta}\cdot -\frac{\partial \theta}{\partial t}\Big|_D \\ =&\frac{1}{|D|}\sum_{i\in S}\left\langle y_i - f(x_i), \frac{\partial f(x_i)}{\partial t}\Big|_D\right\rangle. \end{aligned} \quad (6)$$

This introduces a per-sample score $a_t(i)$ as the inner product of two vectors at step t, i.e., the residual $y_i - f(x_i)$ and its dynamics under the gradient flow computed on D:

$$a_t(i) \triangleq \left\langle y_i - f(x_i;\theta_t), \frac{\partial f(x_i;\theta_t)}{\partial t}\Big|_D\right\rangle. \quad (7)$$

Hence, the expectation in Eq. (4) can be approximated by a function that sums up the scores of all selected samples $i \in S$. Note the two vectors can be directly obtained from the byproduct of training on S and D, so estimating

the score for all the candidate samples does not require any additional computation. However, in each step of curriculum learning, we only train the model on a subset S and only update the score for $i \in S$. In practice, this problem can be mitigated by maintaining an exponential moving average $\hat{a}_{t+1}(i)$ of $a_t(i)$ over time:

$$\hat{a}_{t+1}(i) = \begin{cases} \gamma \times \hat{a}_t(i) + (1-\gamma) \times a_t(i) & \text{if } i \in S_t \\ \hat{a}_t(i) & \text{otherwise,} \end{cases} \quad (8)$$

As we will discuss later, with sufficient exploration over all samples and in the regime of DNN training, $\hat{a}_{t+1}(i)$ is a high-quality and more stable alternative to $a_t(i)$ that is almost free to compute. According to Eq. (4), the optimal S_t simply selects the top-k samples with the largest scores. However, S_t cannot replace D in estimating the linear dynamics $\partial f(x_i)/\partial t|_D$ because S_t can be biased from the data distribution $\mathcal{D}$. Therefore, in our algorithm presented later, instead of selecting the top-k, we sample S_t based on their scores, and cyclically employ a large-batch training epoch over uniform samples from the training set after every episode of mini-batch training on the selected subsets. These strategies encourage more exploration for better estimate to the scores in practice.

Remarks: Take a closer look at the induced score at the end of Eq. (6): the residual $y_i - f(x_i)$ measures the gap between the current prediction $f(x_i)$ and the ground truth y_i (i.e., how hard the sample is), while the linear dynamics delineates how $f(x_i)$ changes (i.e., speed and direction) when training the model using samples drawn from the data distribution. Together, their inner product reflects the momentum of $f(x_i)$ moving towards y_i under the gradient flow on D. Intuitively, we tend to select (1) harder samples that the model can make more progress on and (2) samples that are consistent with most other samples drawn from the same distribution (indicating that reducing the losses on D helps to also move $f(x_i)$ towards y_i). The former intends to select the most informative ones (compared to the ones already learned) and is consistent with the selection criteria proved to be effective in previous curriculum learning [59, 60] and boosting methods [43, 16], while the latter tends to select the most representative ones that are consistent with other data, which is another criterion whose success has been demonstrated in recent curriculum learning methods [61, 62]. However, unlike many previous criteria that are built upon empirical observations or human heuristics, Eq. (6) is derived from a well-formulated and motivated optimization problem.

2.3 Classification

We can extend the above analysis of dynamics for regression to the general multi-class classification task, which learns a model $f(x;\theta)$ to minimize the cross entropy loss $\ell_{xe}(x;\theta)$:

$$\min_\theta \mathbb{E}_{x\sim\mathcal{D}} \ \ell_{xe}(x) \triangleq -\log p(x)[y], \quad (9)$$

$$p(x)[y] = \frac{\exp(f(x)[y])}{\sum_{j=1}^c \exp f(x)[j]},$$

where c is the number of classes and y is the class label of x. We denote the one-hot encoding of y as $\mathbf{y}$. Similarly, at step t, we aim to find a subset $S \subseteq [n]$ of size $|S| \leq k$ whose resulting gradient flow maximizes the projection of the residual $\mathbf{y} - p(x)$ onto the dynamics $\frac{\partial p(x)}{\partial t}\Big|_S$ for all $x \sim \mathcal{D}$, i.e.,

$$\max_{S\subseteq[n],|S|\leq k} \mathbb{E}_{x\sim\mathcal{D}} \left\langle \mathbf{y} - p(x), \frac{\partial p(x)}{\partial t}\Big|_S \right\rangle. \quad (10)$$

Similar to the regression case, we approximate the expectation with samples D drawn from $\mathcal{D}$, i.e.,

$$\begin{aligned}
&\mathbb{E}_{x\sim\mathcal{D}} \left\langle \mathbf{y} - p(x), \frac{\partial p(x)}{\partial t}\Big|_S \right\rangle \\
\approx &\frac{1}{|D|} \sum_{x\in D} [\mathbf{y} - p(x)]^T \cdot \frac{\partial p(x)}{\partial f(x)} \cdot \frac{\partial f(x)}{\partial \theta} \\
&\cdot \sum_{i\in S} -\frac{\partial \ell_{xe}(x_i)}{\partial f(x_i)} \cdot \frac{\partial f(x_i)}{\partial \theta} \\
= &\frac{1}{|D|} \sum_{i\in S} -\left[\frac{\partial \ell_{xe}(x_i)}{\partial f(x_i)}\right]^T \\
&\cdot \sum_{x\in D} \frac{\partial f(x_i)}{\partial \theta} \left[\frac{\partial f(x)}{\partial \theta}\right]^T \left[\frac{\partial p(x)}{\partial f(x)}\right]^T [\mathbf{y} - p(x)] \\
= &\frac{1}{|D|} \sum_{i\in S} \left[\frac{\partial \ell_{xe}(x_i)}{\partial f(x_i)}\right]^T \frac{\partial f(x_i)}{\partial \theta} \\
&\cdot \sum_{x\in D} \left[\frac{\partial f(x)}{\partial \theta}\right]^T \left[\frac{\partial p(x)}{\partial f(x)}\right]^T \frac{\partial \ell(x)}{\partial p(x)} \\
= &\frac{1}{|D|} \sum_{i\in S} \left[\frac{\partial \ell_{xe}(x_i)}{\partial f(x_i)}\right]^T \frac{\partial f(x_i)}{\partial \theta} \cdot \sum_{x\in D} \left[\frac{\partial f(x)}{\partial \theta}\right]^T \frac{\partial \ell(x)}{\partial f(x)} \\
= &\frac{1}{|D|} \sum_{i\in S} \left[\frac{\partial \ell_{xe}(x_i)}{\partial f(x_i)}\right]^T \frac{\partial f(x_i)}{\partial \theta} \cdot -\frac{\partial \theta}{\partial t}\Big|_D \\
= &\frac{1}{|D|} \sum_{i\in S} \left\langle \mathbf{y}_i - p(x_i), \frac{\partial f(x_i)}{\partial t}\Big|_D \right\rangle. \quad (11)
\end{aligned}$$

Hence, we compute the per-sample score $a_t(i)$ by

$$a_t(i) \triangleq \left\langle \mathbf{y}_i - p(x_i;\theta_t), \frac{\partial f(x_i;\theta_t)}{\partial t}\Big|_D \right\rangle, \quad (12)$$

Which has a form similar to Eq. (7) except that the residual is $\mathbf{y}_i - p(x_i;\theta_t)$ for classification. The linear dynamics term in Eq. (12) is associated with the gradient flow minimizing the L2 loss $\ell(\cdot)$ on D instead of the cross-entropy loss $\ell_{xe}(\cdot)$ on S. This is the major difference between Eq. (12) and Eq. (7), which uses the same loss $\ell(\cdot)$ for both the model training and dynamics estimation. This difference requires the training steps to switch between the two types of losses, i.e., we minimize the cross-entropy loss $\ell_{xe}(\cdot)$ during mini-batch training on S_t and switch to the square loss $\ell(\cdot)$ in the large-batch training epoch on $D \sim \mathcal{D}$ at the end of each episode/cycle.

2.4 Learning Dynamics with Neural Tangent Kernel

We can obtain an intuitive explanation of the score in Eq. (6) under the context of neural tangent kernel (NTK) [23, 14]. For simplicity, we focus on the regression task (Section 2.2) in the single-output case (the result can be extended to every dimension in the multiple-output case). The second row of Eq. (6) can be written as

$$\begin{aligned}
&\mathbb{E}_{x\sim\mathcal{D}}\left\langle y-f(x), \frac{\partial f(x)}{\partial t}\Big|_S\right\rangle \\
\approx&\frac{1}{|D|}\sum_{i\in S}-\frac{\partial \ell(x_i)}{\partial f(x_i)}\cdot\sum_{x\in D}\left\langle\frac{\partial f(x_i)}{\partial\theta},\frac{\partial f(x)}{\partial\theta}\right\rangle\cdot[y-f(x)] \\
=&\frac{1}{|D|}\sum_{i\in S}\frac{\partial \ell(x_i)}{\partial f(x_i)}\cdot\sum_{x\in D}\left\langle\frac{\partial f(x_i)}{\partial\theta},\frac{\partial f(x)}{\partial\theta}\right\rangle\cdot\frac{\partial\ell(x)}{\partial f(x)} \\
=&\frac{1}{|D|}r_S^T H_{S,D} r_D=\frac{1}{|D|}\sum_{i\in S,j\in D}H_{i,j}r_i r_j, \qquad (13)\\
r_i\triangleq&\frac{\partial\ell(x_i)}{\partial f(x_i)}=f(x_i)-y_i,\ \ H_{i,j}\triangleq\left\langle\frac{\partial f(x_i)}{\partial\theta},\frac{\partial f(x_j)}{\partial\theta}\right\rangle.
\end{aligned}$$

One can think that H is a dynamic kernel matrix describing the pairwise relationship between sample-i and sample-j in terms of their model gradients at step t. Note both r and H depend on θ_t so they are time-variant. In recent work [23, 1], it is shown that when $f(\cdot)$ is a neural network with enough neurons per layer (i.e., with adequate but still finite width), with high probability, H converges to a deterministic kernel matrix H^* so-called the "neural tangent kernel (NTK)" computed on random initialization. In this case, our objective becomes a weighted sum of the pairwise product of residuals $r_i r_j$ over all $i\in S, j\in D$, where the weights are time-invariant and determined by H^*, i.e.,

$$\begin{aligned}
&\mathbb{E}_{x\sim\mathcal{D}}\left\langle y-f(x), \frac{\partial f(x)}{\partial t}\Big|_S\right\rangle\to\frac{1}{|D|}\sum_{i\in S,j\in D}H^*_{i,j}r_i r_j \\
=&\frac{1}{|D|}\sum_{i\in S}\left[\sum_{j\in D}H^*_{i,j}r_j\right]\cdot r_i. \qquad (14)
\end{aligned}$$

Given the NTK H^*, which is a static matrix describing the pairwise correlation between samples, we can obtain more insights about dynamics optimization in Eq. (4). First, setting S to be all the training samples, i.e., $S=[n]$, is not guaranteed to maximize the objective in Eq. (14). Instead, it prefers samples with both large (i.e., large in magnitude) residuals r_i and strong correlations to other samples with large residual r_j. Specifically, the objective tends to select difficult samples (i.e., large $|r_i|$) that are representative of (i.e., $\text{sign}(H_{i,j})=\text{sign}(r_i r_j)$) and strongly related to (i.e., large $|H_{i,j}|$) other difficult samples $j\in D$ (i.e., large $|r_j|$). Such criteria rule out the following two types of samples, which might be selected by previous curricula: (1) difficult samples with large residuals but weakly related to other samples, which can possibly be outliers (or adversarially chosen) that fail on training; (2) easy samples with small residuals that can only contribute very weak gradients to improve the predictions on difficult samples.

Furthermore, in the NTK regime, H^* does not change over time, so the score of each sample x_i solely depends on its own residual r_i and the residual r_j of its strongly related samples from D. Hence, when applied to training over-parameterized (wide-enough) neural nets, the objective tends to keep selecting the same x_i until most of the strongly-related-samples to x_i have sufficiently small residuals or r_i itself becomes nearly zero. If samples can be well structured by H^*, e.g., H^* has a block diagonal structure after certain symmetric row/column permutation where each block forms a cluster, the dynamics optimization will keep reducing the errors on some clusters until their errors become sufficiently small before switching to other clusters. This property allows us, in practice, to lazily update the scores (which requires large-batch training on i.i.d. samples $D\sim\mathcal{D}$ and might degenerate performance), and for most other steps we can still train the model via mini-batch SGD on the selected subset S_t. That being said, a static H^* is not required by DoCL: the lazy update should work well if the block diagonal structure of H does not change too quickly. In addition, **the score computation in DoCL does not require explicitly computing** H^*. In fact, we avoid additional heavy computation by using only already-computed byproducts of the training process to estimate the linear dynamics in Eq. (6).

3 Empirical Studies of Training Dynamics under Three Data Selection Curricula

The above analysis of dynamics-optimization suggests that we should select samples with larger scores $\hat{a}_t(i)$ (Eq. (8)) for training in each step. In this section, we present an empirical study of the training dynamics with different data selection curricula in a more primitive framework (Algorithm 2 in Appendix) that in each step computes the score for all samples and then trains the model on the top-k samples with the largest/smallest scores. **The aim is to solely evaluate the effectiveness of the proposed scores and rule out influences of any additional heuristics, techniques, or hyperparameters** that we will introduce later for building a more practical algorithm (DoCL in Algorithm 1).

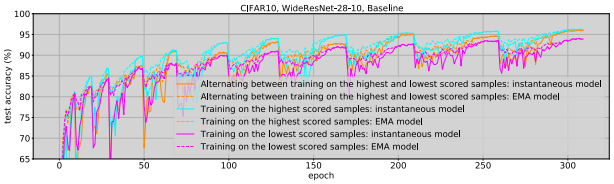

Figure 1: Test set accuracy of WideResNet-28-10 (instantaneous model) and its exponential moving average (EMA model) during the course of training when using the three data selection curricula.

In particular, we train a WideResNet-28-10 on CIFAR10 for multiple episodes/cycles each applying SGD with cosine

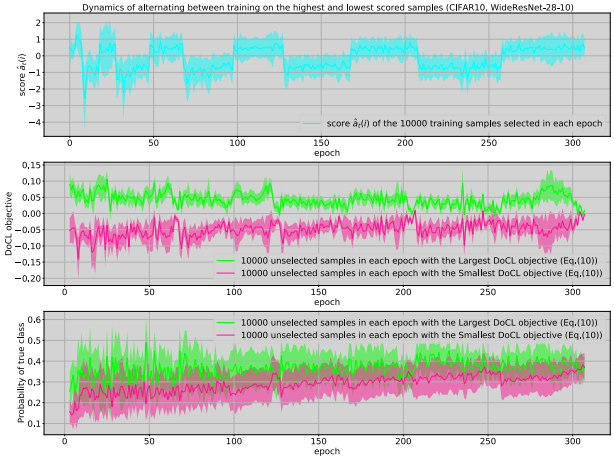

Figure 2: *Baseline1 alternates between the highest and lowest scored samples*: Dynamics (mean±std) for (**Top**) the score $\hat{a}(i)$ (Eq. (8)) of the selected samples, (**Middle**) DoCL objective (Eq. (10)) values of unselected samples, and (**Bottom**) output true-class probabilities for unselected samples. We split the unselected samples in each epoch into two groups with the largest/smallest DoCL objective values.

annealing learning rates. In every epoch of an episode, we select $k = 10000$ samples to train the model and we compare three data selection curricula: (1) *Baseline1* selects the k highest-scored samples in oddly-numbered episodes and k lowest-scored samples in evenly-numbered episodes; (2) *Baseline2* always selects the k highest-scored samples; and (3) *Baseline3* always selects the k lowest-scored samples. To update the scores for all the n samples, in each epoch, we uniformly draw 2048 samples as D to estimate the linear dynamics $\frac{\partial f(x_i;\theta_t)}{\partial t}\Big|_D$ in Eq. (12), and we apply inference on all samples to obtain $f(x_i)$ (costly for practice usage).

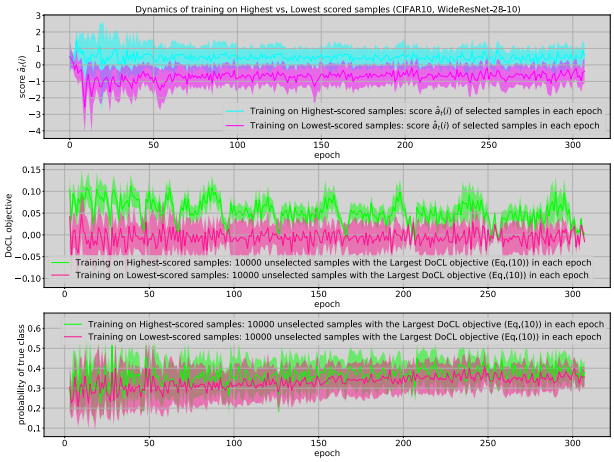

Figure 3: *Training with highest-scored (Baseline2) vs. lowest-scored (Baseline3) samples*: Dynamics (mean±std) for (**Top**) the score $\hat{a}(i)$ (Eq. (8)) of the selected samples, (**Middle**) DoCL objective (Eq. (10)) values and (**Bottom**) output true-class probabilities for unselected samples with the largest DoCL objective values.

We first compare the test set accuracy of the three curricula in Figure 1. Baseline2 keeps achieving the highest test accuracy among the three since very early episodes. In addition, Baseline1 and Baseline2 outperform Baseline3 by a large margin. This indicates that the samples with higher scores bring more improvement to the generalization performance than the ones with lower scores.

Next, we take a closer look at the proposed score $\hat{a}_t(i)$ on selected samples, the DoCL objective (Eq. (10)), the prediction quality (measured by true class probabilities) of unselected data, and their correlations during the course of training. The results verify that optimizing the learning dynamics indeed improves the generalization performance and training on high-scored samples. In Figure 2, we report the dynamics observed on Baseline1. In the middle and bottom plots, we split the unselected samples in each epoch into two groups, i.e., the 10000 samples with the largest DoCL objective values and the 10000 samples with the smallest DoCL objective values. They together show that the model performs better (i.e., producing higher true-class probabilities) on samples with larger DoCL objective values. Hence, optimizing the learning dynamics of all samples (not only the selected training samples) is consistent with the learning goal of reducing the classification error over the data distribution. Moreover, in the top and middle plots, we observe that the DoCL objective (for both groups) degrades when training on the lowest-scored samples, while it increases when training on the highest-scored ones. It indicates that the highest-scored samples improve the DoCL objectives more effectively and result in better prediction qualities.

In Figure 3, we compare the dynamic patterns of Baseline2 and Baseline3. The top plot shows that the samples are distinguishable based on their scores, indicating that they are not equal in accelerating the learning process and thus a data selection curriculum can be better than uniform sampling. In the middle and bottom plots, we compare the two baselines on their 10000 unselected samples with the largest DoCL objective values, which are the better-predicted samples as implied by Figure 2 and the objective formulation in Eq. (10). The middle and bottom plots show that by selecting the highest-scored samples as in Baseline2, we can make greater learning progress and achieve better prediction accuracy on these better-predicted samples.

Therefore, the training dynamics observed on the three baseline curricula justify the proposed DoCL objective for dynamics optimization and motivate us to develop a curriculum learning method based on selecting samples with higher score $\hat{a}_t(i)$. We also provide similar empirical results on CIFAR100 and larger version of the plots in Appendix.

4 Dynamics-optimized Curriculum Learning (DoCL)

In this section, we will develop a new practical curriculum learning algorithm based mainly on the above dynamics-optimization strategy. It also integrates other techniques to make it more efficient and compatible with current deep

learning schemes. We provide its detailed procedures in Algorithm 1 and subsequently elaborate on its major steps.

Algorithm 1 Dynamics-optimized Curriculum Learning

1: **input:** $\{(x_i, y_i)\}_{i=1}^n$, $\ell(\cdot,\cdot)$, $f(\cdot;\theta)$, $\{\eta_t\}_{t=0}^{T_\kappa}$, $\{T_i\}_{i=0}^{\kappa}$, $\gamma_k \in [0,1]$, $k_{\min}$
2: **initialize:** $T_{-1}=0$, $k=n$, $\rho_i=0$, $g_i=f(x_i)$
3: **for** $j \in \{0,\cdots,\kappa\}$ **do**
4: **for** $t \in \{T_{j-1},\cdots,T_j\}$ **do**
5: **if** $t < T_0$ or $t = T_j$ **then**
6: Uniform sampling $S_t \subseteq [n]$ up to size n;
7: Update θ by large-batch SGD with learning rate η_t to minimize L2 loss on S_t;
8: **else**
9: $S_t \leftarrow$ Draw k samples with probability $\propto \hat{a}_t(i)$;
10: *Optional*: prune S_t to a diverse subset by submodular maximization in Eq. (15);
11: Update θ by mini-batch SGD with learning rate η_t to minimize the task's loss $\ell(\cdot)$ on S_t;
12: **end if**
13: **for** $i \in S_t$ **do**
14: Estimate linear dynamics of $f(x_i)$: $\rho_i \leftarrow \rho_i + \eta_t$, $\frac{\partial f(x_i)}{\partial t} = \frac{f(x_i)-g_i}{\rho_i}$;
15: Restore $\rho_i \leftarrow 0$ and $g_i \leftarrow f(x_i)$;
16: Compute $a_t(i)$ by Eq. (7) (regression) or Eq. (12) (classification);
17: Update $\hat{a}_{t+1}(i)$ using Eq. (8);
18: **end for**
19: **end for**
20: Reduce training set size: $k \leftarrow \max\{k_{\min}, \gamma_k \times k\}$;
21: **end for**

Warm starting. To initialize the scores, at the beginning we run T_0 epochs of large-batch SGD (line 5-7) to minimize the L2 loss on the whole training set. These warm-start epochs provide accurate estimates of the scores in Eq. (7) (regression) or Eq. (12) (classification), in which the linear dynamics should be estimated under the full gradient flow (rather than stochastic gradient flow) that minimizes the L2 loss on a training set D drawn from the data distribution $\mathcal{D}$.

Cyclical curriculum learning. We train the model for multiple (κ) episodes/cycles with an increasing number of steps (i.e., $T_{j+1}-T_j > T_j - T_{j-1}$ for $\{T_j\}_{j=1}^{\kappa}$ in Algorithm 1), where each episode starts with a large or rapidly increasing learning rate, which gradually decays towards zero by a predefined function (e.g., cosine or exponent). The learning rate decay results in a fast convergence to local minima, while its surge at the beginning of each episode helps to quickly jump out from the previous local minima. Hence, cyclical learning rates [46] such as the cosine annealing schedule [35] can quickly jump between different local minima on the loss landscape and explore more regions without being trapped in challenging local minima. It is a perfect match to our strategy since it leads to more exploration of the training dynamics under different learning rates, which improves the estimates of the scores. Moreover, at the end of each episode (line 20), we reduce the training set size k because more samples have their predictions converging to the ground truth as the training proceeds. In addition, we apply a large-batch training epoch (similar to the ones during warm starting) to update the scores (line 5-7).

Estimate the linear dynamics under varying learning rates. For computing $a_t(i)$ (line 16), we need to estimate the linear dynamics $\partial f(x)/\partial t$ in continuous time from the observations of $f(x)$ at discrete time steps. Since the learning rate η_t can change over time, and a larger learning rate leads to greater changes in $f(x)$, we estimate $\partial f(x)/\partial t$ at step t by $(f_t(x)-f_{t'}(x))/\sum_{q=t'}^{t}\eta_q$, where t' is the last step before t when x is selected for training. In line 14-15 of Algorithm 1, we update ρ_i and g_i to keep a record of $\sum_{q=t'}^{t}\eta_q$ and $f_{t'}(x)$ for sample-i, which is used to estimate the linear dynamics.

Update the scores by using dynamics computed on S_t. Theoretically, the scores can only be updated during the warm start epochs at the beginning of the algorithm and the update epoch at the end of each episode. In other steps (line 9-11), since we instead apply mini-batch training on a possibly biased subset S_t (i.e., not guaranteed to be i.i.d. drawn from $\mathcal{D}$) and minimize a loss determined by the task (i.e., not always to be the L2 loss), the resulting training dynamics $\partial f(x)/\partial t$ can be different from the one required in Eq. (7) and Eq. (12). However, in practice, by encouraging more exploration on samples with small $\hat{a}_t(i)$ when sampling S_t (line 9), we find that the byproducts of those training steps can also be leveraged to update $\hat{a}_t(i)$ (line 13-18) and produce compelling performance.

Weighted sampling. The problem formulations in Eq. (4) and Eq. (10) suggest directly selecting samples with the largest scores $\hat{a}_t(i)$. For better exploration, however, we instead apply a weighted sampling of S_t based on the scores (line 9). We can also trade off exploration vs. exploitation using strategies from online learning methods. For example, we can sample S_t from a Boltzmann distribution, i.e., $\Pr(i \in S_t) = \exp(\hat{a}_t(i)/\tau)$, where τ is a temperature parameter. We can additionally apply exponential weights similar to Exp3 [2] if we assume the feedback $a_t(i)$ is more adversarial than entirely stochastic. The momentum $a_t(i)$ can either increase or decrease during different training stages so it is not entirely stochastic. It is neither purely adversarial since SGD on a complicated loss landscape does not play against the curriculum. In this case, we can additionally re-scale $a_t(i) \leftarrow a_t(i)/\Pr(i \in S_t)$ after line 16. It encourages more exploration since x with a smaller probability is more likely to be selected in the future.

Further prune S_t to a diverse subset. We can further reduce training time in early stages when k is large by extracting a small and diverse/representative subset of S_t. Inspired by MCL [59], at line 10, we reduce S_t to a subset of size $k'_t = \gamma_{k'} k_t$ ($0 < \gamma_{k'} \leq 1$) by (approximately)

solving the following submodular maximization problem:

$$\max_{S\subseteq S_t, |S|\leq k'} \sum_{i\in S} \hat{a}_t(i) + \lambda_t F(S), \tag{15}$$

where $F : 2^{S_t} \rightarrow \mathbb{R}_+$ is a submodular function [17] so we can exploit fast greedy algorithms [39, 37, 38] to solve Eq. (15) with an approximation guarantee. We gradually reduce preference for diversity as training proceeds via reducing λ_t by a factor $0 \leq \gamma_\lambda \leq 1$ at each step.

5 Experiments

We compare DoCL with the widely-used random mini-batch SGD, two recent curriculum learning methods, i.e., self-paced learning (SPL) [31] and minimax curriculum learning, and one variant of DoCL (DoCL-NR) by using them to train different DNNs architectures on 9 image classification datasets (without pre-training), i.e., **(A)** WideResNet-28-10 [57] on CIFAR10 and CIFAR100 [30]; **(B)** ResNeXt50-32x4d [56] on Food-101 [7], FGVC Aircraft (Aircraft) [36], Stanford Cars [29], and Birdsnap [6]; **(C)** ResNet50 [20] on ImageNet [13]; **(D)** WideResNet-16-8 on Fashion-MNIST (FMNIST) [55]; **(E)** PreActResNet34 [20] on SVHN [40]. The major difference across different curriculum learning methods lies in the criteria to select samples. Random mini-batch SGD adopts the criterion of uniform sampling over training set. SPL and MCL rely on the instantaneous loss of each sample: SPL tends to select easier samples while MCL prefers harder ones and applies an additional diversity criterion as in Eq. (15). As defined in Eq. (7) and Eq. (12), DoCL's criteria are built upon the inner product of the residuals and linear dynamics. Hence, MCL can also be seen as a variant of DoCL that only considers the instantaneous feedback on the residual part. To complete this ablation study, we consider another variant DoCL-NR (NR stands for "no residual") that only relies on the linear dynamics part, i.e., it instead uses $a_t(i) = \|\partial f(x_i;\theta_t)/\partial t\|_2$ to compute the running mean in Eq. (8) and follows the schedule of increasing samples over epochs as MCL (vs. decreasing samples in DoCL).

Hyperparameters. For all these methods, we update the model on their selected/sampled data using mini-batch SGD with momentum of 0.9. We use a cyclical cosine annealing learning rate schedule [35] (multiple cycles with ending epoch numbers $\{\eta_t\}_{t=1}^{T_\kappa}$ and with starting/target learning rate decayed by a multiplicative factor 0.85). It will suffer a short period of accuracy drop due to the surging learning rate at the beginning of every cycle (as in Figure 4) but can traverse more regions with different local minima on the loss landscape and eventually achieve better performance with faster long-term convergence. On each dataset, we tried a handful of schedules on mini-batch SGD, chose the one with the best validation accuracy, and used it for all the methods (but each method may select different numbers of samples per epoch). We apply standard data augmentation on all datasets and Mix-up [58] with $\alpha = 0.4$. More details about the datasets, options of $\{\eta_t\}_{t=1}^{T_\kappa}$ and other training hyperparameters shared across methods can be found in the appendix. In DoCL-NR and DoCL, we set $\gamma_k = \gamma_\lambda = 0.9, k_{\min} = 0.2n$. We tried a few common choices for them, e.g., $\gamma_k, \gamma_\lambda \in \{0.85, 0.9, 0.95\}$ and chose the best one. On some datasets, we further test the performance of Eq. (15) in reducing S_t and employ the "facility location" submodular function [9] $G(S) = \sum_{j\in S_t} \max_{i\in S} \omega_{i,j}$ as a diversity criterion, where $\omega_{i,j}$ represents the similarity between sample x_i and x_j. We utilize a Gaussian kernel for similarity measurement using neural net features (i.e., the inputs to the last fully connected layer in our experiments) $z(x)$ for each x, i.e., $\omega_{i,j} = \exp\left(-\|z(x_i)-z(x_j)\|^2/2\sigma^2\right)$, where σ is the mean value of all the $k(k-1)/2$ pairwise distances.

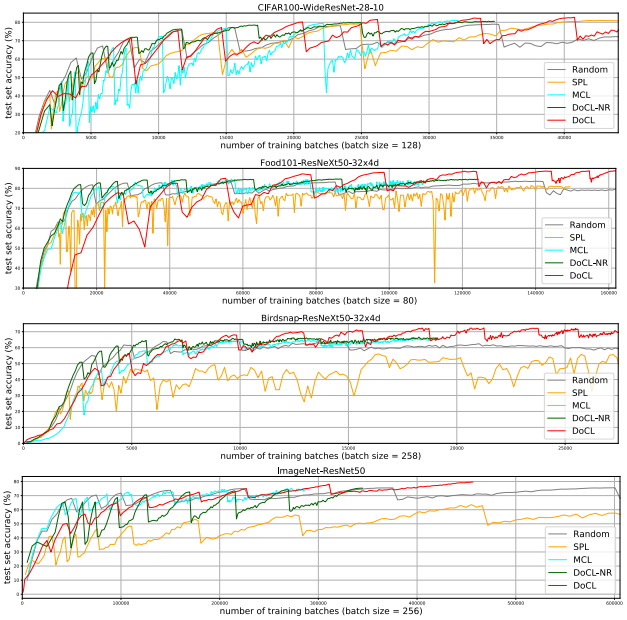

Figure 4: Training DNNs with DoCL, DoCL-NR, SPL [31], MCL [59], and random mini-batch SGD on 3 datasets: CIFAR100, Food101, Birdsnap and ImageNet. We report test accuracy changes v.s. the number of training batches.

Main Results. In Table 1, we summarize the final test accuracy achieved by every method on all the 9 datasets. DoCL achieves the highest test accuracy among all the evaluated methods and outperforms them by a large margin. On the four fine-grained classification datasets (i.e., Food101, Birdsnap, Aircraft and Cars) that traditional solutions usually rely on fine tuning a pre-trained model, DoCL significantly improves the training from scratch so the resulting accuracy can be comparable with the fine-tuned models. Furthermore, DoCL improves the state-of-the-art top-1 accuracy of ResNet50 on ImageNet from 79.29% [21] (after applying a bag of tricks that might be specific for ImageNet) to 79.54% without heavy tuning of tricks and hyperparameters.

Efficiency. In Figure 4, we report how the test accuracy improves with the increasing number of training batches on three datasets (more results in the appendix). During the first several cycles, DoCL has lower accuracies than some other baselines because it starts from the whole training set

Table 1: The test accuracy (%) achieved by random mini-batch SGD (Random), SPL, MCL, DoCL-NR and DoCL in training DNNs on 9 datasets (without pre-training). In MCL, DoCL-NR and DoCL, we apply lazier-than-lazy-greedy [38] for Eq. (15) on CIFAR10, CIFAR100, SVHN and FMNIST. DoCL achieves the highest test accuracy over all 9 datasets.

Curriculum	CIFAR10	CIFAR100	Food-101	ImageNet	SVHN	FMNIST	Birdsnap	Aircraft	Cars
Random	96.18	79.64	83.56	75.04	96.48	95.22	64.23	74.71	78.73
SPL [31]	93.55	80.25	81.36	73.23	96.15	92.09	63.26	68.95	77.61
MCL [59]	96.60	80.99	84.18	75.09	96.93	95.07	65.76	75.28	76.98
DoCL-NR	96.40	81.42	84.75	75.62	96.80	95.50	66.59	79.72	81.48
DoCL (Ours)	**97.43**	**83.23**	**87.45**	**79.54**	**97.36**	**95.89**	**71.37**	**82.40**	**86.26**

and gradually reduces the samples per epoch (i.e., line 20 of Algorithm 1) while other CL methods increase the samples per epoch from a small number. DoCL selects samples in decreasing numbers since it needs sufficient exploration of more samples in line with the fast-changing linear dynamics during earlier stages. Hence, given the same number of epochs, DoCL has longer cycles than others. As the cycle length decreases for later stages/cycles, the optimization of dynamics in DoCL prevails and improves the test accuracy much faster than other methods.

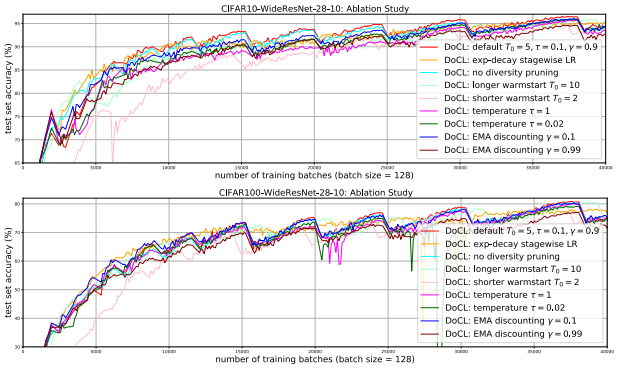

Figure 5: Ablation study and sensitivity analysis of hyperparameters for DoCL when applied to train WideResNet-28-10 on CIFAR10 (top) and CIFAR100 (bottom).

Ablation Study. In Figure 5, we conduct a thorough ablation study of DoCL (i.e., removing diversity pruning in line 10 of Algorithm 1, or changing the cyclical learning rate to exponential decaying learning rate over episodes) and compare the default hyperparameters (i.e., $T_0 = 5, \tau = 0.1, \gamma = 0.9$) used in above experiments with other options. It shows that DoCL is not very sensitive to most hyperparameters. Specifically, DoCL equipped with diversity pruning, cyclical learning rates, longer warm-starting epochs, lower temperature τ, and moderate γ works slightly better than their counterparts. DoCL with a very short warm-starting period (2 epochs) suffers from insufficient exploration over all samples in earlier stages and thus performs much poorer than other variants but it finally achieve similar test accuracy as others in later stages.

Regression. Although the above experiments mainly focus on classification tasks, we also evaluate the performance of DoCL and compare it with other baselines on a regression task called "knowledge distillation" [8, 42, 22] that aims to transfer the knowledge of a pre-trained large neural net to a smaller one (e.g., ResNet-18). In particular, we study an L2 regression minimizing the L2 loss between the output logits (the last-layer outputs before softmax) of a ResNet-18 model and the logits produced by a pre-trained ResNeXt-29 8x64d on the same data. We apply different methods to sequentially select the data subset in each epoch on which we minimize the L2 distillation loss. In Figure 6, we report their performance on CIFAR10 and CIFAR100, which shows that DoCL achieves the best test accuracy among all the methods. On CIFAR100, DoCL keeps outperforming the others starting at very early stages. On CIFAR10, DoCL improves slower during earlier stages due to the decreasing schedule of subset size adopted by DoCL (as illustrated before) but it surpasses others after 30,000 training batches.

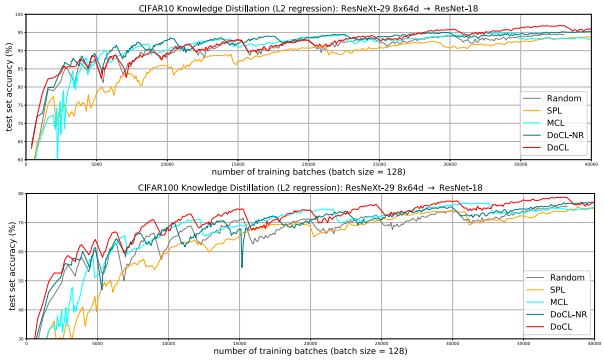

Figure 6: Regression (knowledge distillation with L2 loss on pre-softmax logits) by DoCL, DoCL-NR, SPL [31], MCL [59], and random mini-batch SGD for transferring the knowledge of a pre-trained ResNeXt-29 8x64d (34.4M parameters) to ResNet-18 (11.2M parameters) on CIFAR10 (top) and CIFAR100 (bottom).

6 Conclusion

We derive a general curriculum learning strategy (DoCL) from the optimization of training dynamics. DoCL selects training sample subsets that most quickly help the predictions for samples drawn from the data distribution get close to their targets (Eq. (1)). It uses an objective that combines training dynamics and gradient flow in both the regression and classification settings. We relate DoCL to recent studies on the neural tangent kernel. The DoCL scores depend only on training time byproducts and thus incur minimal extra computation. DoCL is built upon a time-moving average of this score and integrates it into a framework with several state-of-the-art DNN-training techniques. In experiments over 9 datasets, DoCL substantially improves the performance and efficiency over existing CL methods.

Acknowledgments

This research is based upon work supported by the National Science Foundation under Grant No. IIS-1162606, the National Institutes of Health under award R01GM103544, and by a Google, a Microsoft, and an Intel research award. It is also supported by the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. Some GPUs used to produce the experimental results are donated by NVIDIA. We would like to thank AISTATS area chairs and anonymous reviewers for their efforts in reviewing this paper and their constructive comments! We also thank Chandrashekhar Lavania, Lilly Kumari, and all the MELODI lab members for their helpful discussions and feedback.

References

[1] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems 32*, pages 8141–8150. 2019.

[2] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2003.

[3] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. In *Proceedings of the 23rd International Conference on Machine Learning*, page 65–72, 2006.

[4] Sumit Basu and Janara Christensen. Teaching classification boundaries to humans. In *AAAI*, pages 109–115, 2013.

[5] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, pages 41–48, 2009.

[6] Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L. Alexander, David W. Jacobs, and Peter N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2014.

[7] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014.

[8] Cristian Buciluundefined, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 535–541, 2006.

[9] G. Cornuéjols, M. Fisher, and G.L. Nemhauser. On the uncapacitated location problem. *Annals of Discrete Mathematics*, 1:163–177, 1977.

[10] Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *AAAI*, pages 746–751, 2005.

[11] Ido Dagan and Sean P. Engelson. Committee-based sampling for training probabilistic classifiers. In *ICML*, pages 150–157, 1995.

[12] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *ICML*, pages 208–215, 2008.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[14] Pedro Domingos. Every model learned by gradient descent is approximately a kernel machine, 2020.

[15] Yang Fan, Fei Tian, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. Learning to teach. In *International Conference on Learning Representations*, 2018.

[16] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[17] Satoru Fujishige. *Submodular functions and optimization*. Annals of discrete mathematics. Elsevier, 2005.

[18] 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*, volume 70, pages 1311–1320, 2017.

[19] 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*, volume 97, pages 2535–2544, 2019.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[21] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li. Bag of tricks for image classification with convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 558–567, 2019.

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[23] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems 31*, pages 8571–8580. 2018.

[24] Angela H. Jiang, Daniel L. K. Wong, Giulio Zhou, David G. Andersen, Jeffrey Dean, Gregory R. Ganger, Gauri Joshi, Michael Kaminksy, Michael Kozuch,

Zachary C. Lipton, and Padmanabhan Pillai. Accelerating deep learning by focusing on the biggest losers, 2019.

[25] Lu Jiang, Deyu Meng, Shoou-I Yu, Zhenzhong Lan, Shiguang Shan, and Alexander G. Hauptmann. Self-paced learning with diversity. In *NeurIPS*, pages 2078–2086, 2014.

[26] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G. Hauptmann. Self-paced curriculum learning. In *AAAI*, pages 2694–2700, 2015.

[27] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 2304–2313, 2018.

[28] Faisal Khan, Xiaojin (Jerry) Zhu, and Bilge Mutlu. How do humans teach: On curriculum learning and teaching dimension. In *NeurIPS*, pages 1449–1457, 2011.

[29] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[30] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[31] M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *NeurIPS*, pages 1189–1197, 2010.

[32] Ji Liu and Xiaojin Zhu. The teaching dimension of linear learners. *Journal of Machine Learning Research*, 17(162):1–25, 2016.

[33] Weiyang Liu, Bo Dai, Ahmad Humayun, Charlene Tay, Chen Yu, Linda B. Smith, James M. Rehg, and Le Song. Iterative machine teaching. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2149–2158, 2017.

[34] Weiyang Liu, Bo Dai, Xingguo Li, Zhen Liu, James Rehg, and Le Song. Towards black-box iterative machine teaching. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 3141–3149, 2018.

[35] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[36] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.

[37] Michel Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, volume 7 of *Lecture Notes in Control and Information Sciences*, chapter 27, pages 234–243. Springer Berlin Heidelberg, 1978.

[38] Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. Lazier than lazy greedy. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 1812–1818, 2015.

[39] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

[40] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[41] Kaustubh R Patil, Xiaojin Zhu, Ł ukasz Kopeć, and Bradley C Love. Optimal teaching for limited-capacity human learners. In *NeurIPS*, pages 2465–2473, 2014.

[42] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015.

[43] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[44] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction. In *CAIDA*, pages 309–318, 2001.

[45] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin, Madison, 2010.

[46] L. N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472, 2017.

[47] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. Baby Steps: How "Less is More" in unsupervised dependency parsing. In *NeurIPS 2009 Workshop on Grammar Induction, Representation of Language and Language Learning*, 2009.

[48] Weijie Su, Stephen Boyd, and Emmanuel J. Candès. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. *Journal of Machine Learning Research*, 17(153):1–43, 2016.

[49] James Steven Supancic III and Deva Ramanan. Self-paced learning for long-term tracking. In *CVPR*, pages 2379–2386, 2013.

[50] Kevin Tang, Vignesh Ramanathan, Li Fei-fei, and Daphne Koller. Shifting weights: Adapting object detectors from image to video. In *NeurIPS*, pages 638–646, 2012.

[51] Ye Tang, Yu-Bin Yang, and Yang Gao. Self-paced dictionary learning for image classification. In *MM*, pages 833–836, 2012.

[52] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[53] Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *ICML*, 2015.

[54] Andre Wibisono, Ashia C. Wilson, and Michael I. Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.

[55] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[56] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[57] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

[58] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[59] Tianyi Zhou and Jeff Bilmes. Minimax curriculum learning: Machine teaching with desirable difficulties and scheduled diversity. In *ICLR*, 2018.

[60] Tianyi Zhou, Shengjie Wang, and Jeff A. Bilmes. Curriculum learning by dynamic instance hardness. In *NeurIPS*, 2020.

[61] Tianyi Zhou, Shengjie Wang, and Jeff A. Bilmes. Time-consistent self-supervision for semi-supervised learning. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 11523–11533, 2020.

[62] Tianyi Zhou, Shengjie Wang, and Jeff A. Bilmes. Robust curriculum learning: From clean label detection to noisy label self-correction. In *ICLR*, 2021.

[63] Xiaojin Zhu. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *AAAI*, pages 4083–4087, 2015.