Data Parameters: A New Family of Parameters for Learning a Differentiable Curriculum

Shreyas Saxena, Oncel Tuzel, Dennis DeCoste
{shreyas_saxena, otuzel, ddecoste} @ apple.com

CODE: https://github.com/apple/ml-data-parameters

Motivation

Humans and machines learn differently. Can we bridge the gap by learning a curriculum?

Humans	Machine Learning
Different levels for learning a new language	Master all aspects of NLP/ speech together
Music: Learn chords, notes, song, compose	Music: Learn composition directly
Learn few set of visual concepts first	Learn all 1000 ImageNet classes
• • •	• • •

Limitations of Existing Work

Hand-crafted curriculums. Main issues which limit application of existing approaches:

(1) Not trivial to define what are easy examples or task,

(2) Assumes curriculum for humans and machine is correlated,

(3) Predefined curriculum might not be optimal for a dynamically learnt model.

Goal: Formulate a differentiable task agnostic curriculum learning setup.

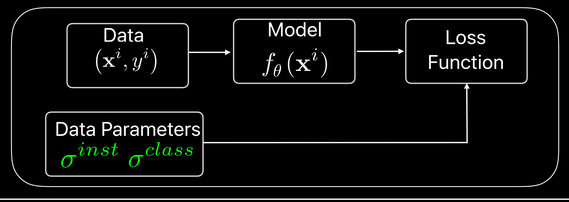
Highlight

- Introduced data parameters for every class and instance in the dataset. Data parameters are learnt by gradient descent, and doing so, amounts to learning a differentiable curriculum.
- **First curriculum learning method** to show gains on large-scale image classification dataset (**ImageNet, WebVision**) and detection task (**KITTI**)!
- In noisy setting on **CIFAR10** and **CIFAR100**, improves over the state-of-the-art by an **absolute 14%.**
- **Free gain:** Consistent gains with same training time and model complexity at inference.

Learning with Data Parameters

Standard setup: Model parameters determine function mapping, and are updated with gradient descent.

Proposed setup: Each class and instance has a corresponding data parameter, which controls their importance in the optimization process. Data parameters are also updated via gradient descent.



Learning With Class-Level Parameters

$\{(\mathbf{x}^i, y^i)\}_{i=1}^N$ denotes the data.

$\mathbf{x}^i \in \mathbb{R}^d$ denotes a single data point.
$y^i \in \{1, ..., k\}$ denotes target label.

$\sigma^{class} \in \mathbb{R}^k$ denotes class-level parameters for classes.

Key idea: Use label of training data point, and modify the entropy of predictive distribution

$$p^i_{y^i} = \frac{\exp(z^i_{y^i}/\sigma^{class}_{y^i})}{\sum_j \exp(z^i_j/\sigma^{class}_{y^i})} \qquad L^i = -\log(p^i_{y^i})$$

$$\frac{\partial L^i}{\partial z^i_j} = \frac{p^i_j - 1(j = y^i)}{\sigma^{class}_{y^i}} \qquad \frac{\partial L^i}{\partial \sigma^{class}_{y^i}} = \frac{(1 - p^i_{y^i})}{(\sigma^{class}_{y^i})^2}\Big(z^i_{y^i} - \sum_{j \neq y^i} q^i_j z^i_j\Big)$$

Can accelerate or decay learning rate of class conditioned on current state of model.

Where q^i_j is probability distribution over non-target classes with $j \neq y^i$.

Learning With Instance-Level Parameters

In practice instances within a class have varying levels of difficulty (noisy labels, occluded objects, etc.

$\sigma^{inst} \in \mathbb{R}^N$ denotes instance-level parameters for instances.

Key idea: Use index of training data point, and modify the entropy of predictive distribution.

$$p^i_{y^i} = \frac{\exp(z^i_{y^i}/\sigma^{inst}_i)}{\sum_j \exp(z^i_j/\sigma^{inst}_i)}$$

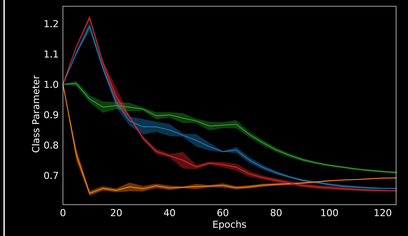
Results for Image Classification and Detection

Dataset	Model	Baseline	DCL	Class	Instance
CIFAR100	WRN-28-10	80.1 ± 0.2	$\mathbf{80.8 \pm 0.1}$	✓	✓
ImageNet	ResNet18	70.3 ± 0.1	$\mathbf{71.0 \pm 0.1}$	✓	✓
			70.8 ± 0.1	✓	✗
WebVision	ResNet18	66.3 ± 0.1	$\mathbf{67.5 \pm 0.1}$	✓	✓
			67.1 ± 0.1	✗	✓

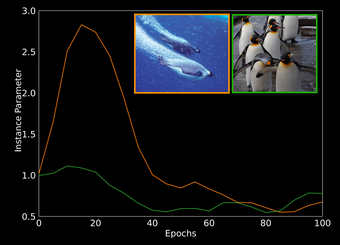
Consistent gains across different datasets and CNN architectures.

Is ImageNet a clean dataset?

Pastry Greenhouse



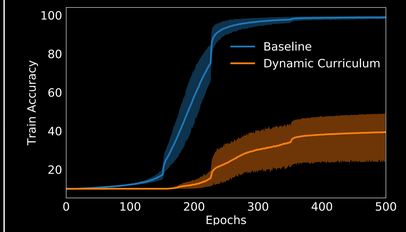
Class Level Dynamic Curriculum



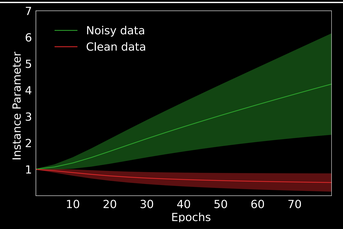
Instance Level Dynamic Curriculum

Dynamic curriculum over the course of training (left to right) for object detection.

Results on Noisy Datasets



CIFAR100 all random labels. Dynamic curriculum formulation resists memorization.



	CIFAR-10	CIFAR-100
Baseline [26]	67.97 ± 0.62	50.66 ± 0.24
Reed Hard [26]	69.66 ± 1.21	51.34 ± 0.17
S Model [26]	70.64 ± 3.09	49.10 ± 0.58
MentorNet PD [17]	76.6	56.9
DCL (ours)	$\mathbf{91.10 \pm 0.70}$	$\mathbf{70.93 \pm 0.15}$
Baseline on clean data (oracle)	94.24 ± 0.15	74.18 ± 0.19

Performance under 40% uniform label noise.

TM and © 2019 Apple Inc. All rights reserved. Redistribution or public display not permitted without written permission from Apple.