Characterizing Structural Regularities of Labeled Data in Overparameterized Models

-Summary-

Introduction

Problem & Notations

- Mastering a domain involves knowing when to generalize and when not to generalize.

  (ex : kick -> kicked ⬄ seek ->sought ⬄ need ->needed) => Irregularity exists

➢ Knowing what to 'memorize' is crucial for improving performance of model

- Problem : How to characterize those irregular /sub-regular / regular examples?

Notations :

- $f(\cdot\,; D)$: model trained on D, where D = i.i.d. sample of size n following data distribution P
- Consistency profile : $C_{P,n}(x, y) = \mathbb{E}_{D\sim P}[P(f(x; D - \{(x, y)\}) = y)]$

Note : For 'dense mode' (regular examples), the model prediction is accurate even for small n. However, for 'sparse mode' (irregular examples), the prediction will be inaccurate for even large n

Empirical consistency profile

Empirical consistency profile

- Empirical consistency profile (we don't know P in general) :

$$\hat{C}_{\widehat{D},n}(x,y) = \widehat{\mathbb{E}}^r_{D\sim\widehat{D}}[P(f(x;D-\{(x,y)\}=y)]$$

where $\widehat{D}$ is empirical data distribution / $\widehat{\mathbb{E}}^r$ denotes empirical averaging with r i.i.d. samples of D.

(Note : To get reasonably accurate estimate, r should be large => Computationally intractable)

- With clever grouping and reuse, the number of models we need to train can be greatly reduced

Empirical consistency profile – estimation algorithm

Algorithm 1 Estimation of $\hat{C}_{\hat{\mathcal{D}},n}$

Require: Data set $\hat{\mathcal{D}} = (X, Y)$ with N examples
Require: n: number of instances used for training
Require: k: number of subset samples
Ensure: $\hat{C} \in \mathbb{R}^N$: $(\hat{C}_{\hat{\mathcal{D}},n}(x, y))_{(x,y)\in\hat{\mathcal{D}}}$

Initialize binary mask matrix $M \leftarrow 0^{k\times N}$
Initialize 0-1 loss matrix $L \leftarrow 0^{k\times N}$
for $i \in (1, 2, \ldots, k)$ **do**
  Sample n random indices I from $\{1, \ldots, N\}$
  $M[i, I] \leftarrow 1$
  Train $\hat{f}$ from scratch with the subset $X[I], Y[I]$
  $L[i, :] \leftarrow \mathbf{1}[\hat{f}(X) \neq Y]$
end for
Initialize score estimation vector $\hat{C} \leftarrow 0^N$
for $j \in (1, 2, \ldots, N)$ **do**
  $Q \leftarrow \neg M[:, j]$
  $\hat{C}[j] \leftarrow \mathrm{sum}(\neg L[:, Q])/\mathrm{sum}(Q)$
end for

Note :

- We need to get $\hat{C}_{\widehat{D},n}(x, y)$ for all $(x, y) \in \widehat{D}$
- the empirical expectation is taken over k i.i.d. samples of $\widehat{D}$ (notation change $r \to k$)

This can be replaced to $\mathbb{P}[\hat{f}(X) = y]$ with removal of ¬ on last line

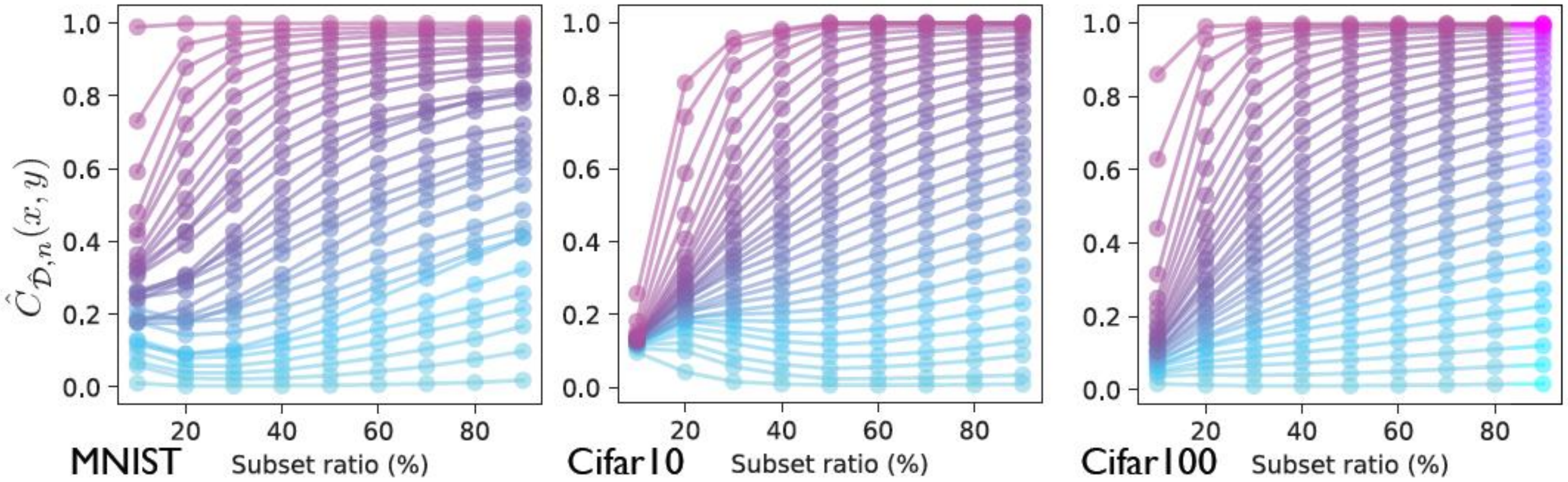
Use trained model effectively, avoid the case $(x, y) \in \widehat{D} - D$

Consistency score (C-score)

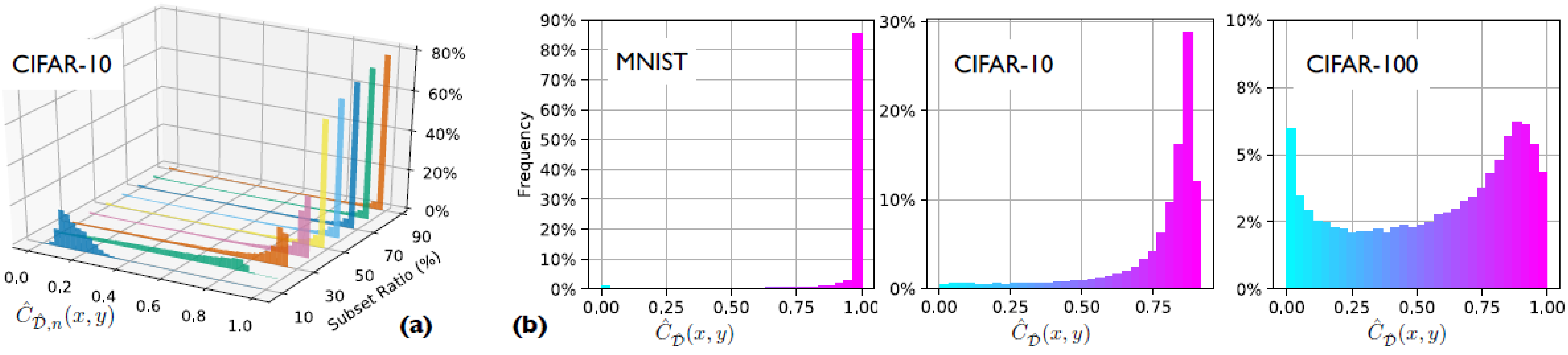
Consistency score (C-score)

- In particular, we differ n (size of training set on $\hat{C}_{\hat{D},n}$) dynamically according to the subset ratio $s \in \{10\%, \ldots, 90\%\}$ of the full available training set $\hat{D}$.
- To get a total ordering of the examples in a data set, we need to take the expectation of consistency profile over n
- Consistency score (C-score) : $\hat{C}_{\hat{D}}(x, y) = \mathbb{E}_n[\hat{C}_{\hat{D},n}(x, y)]$

Consistency profiles of training examples



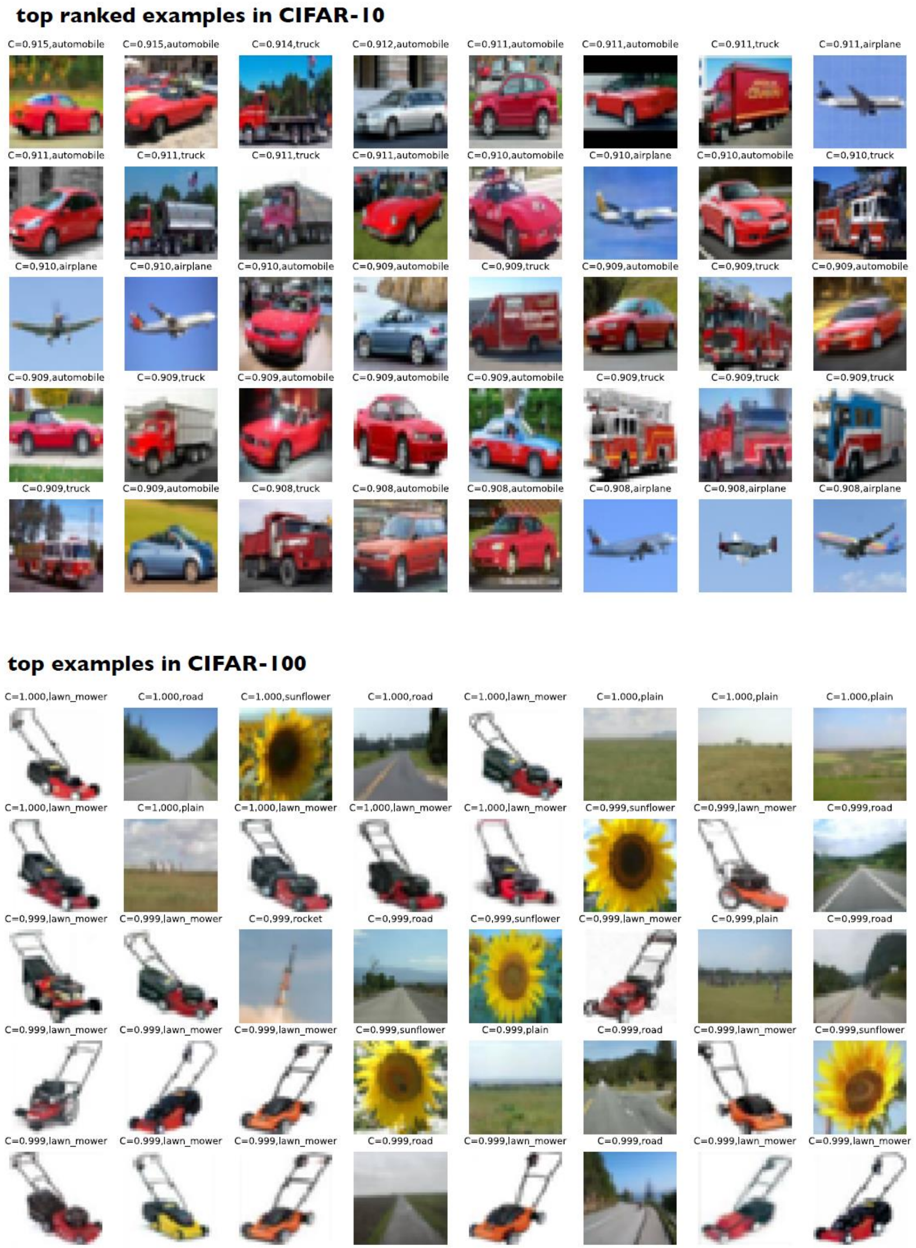
The structural regularities of common image data set

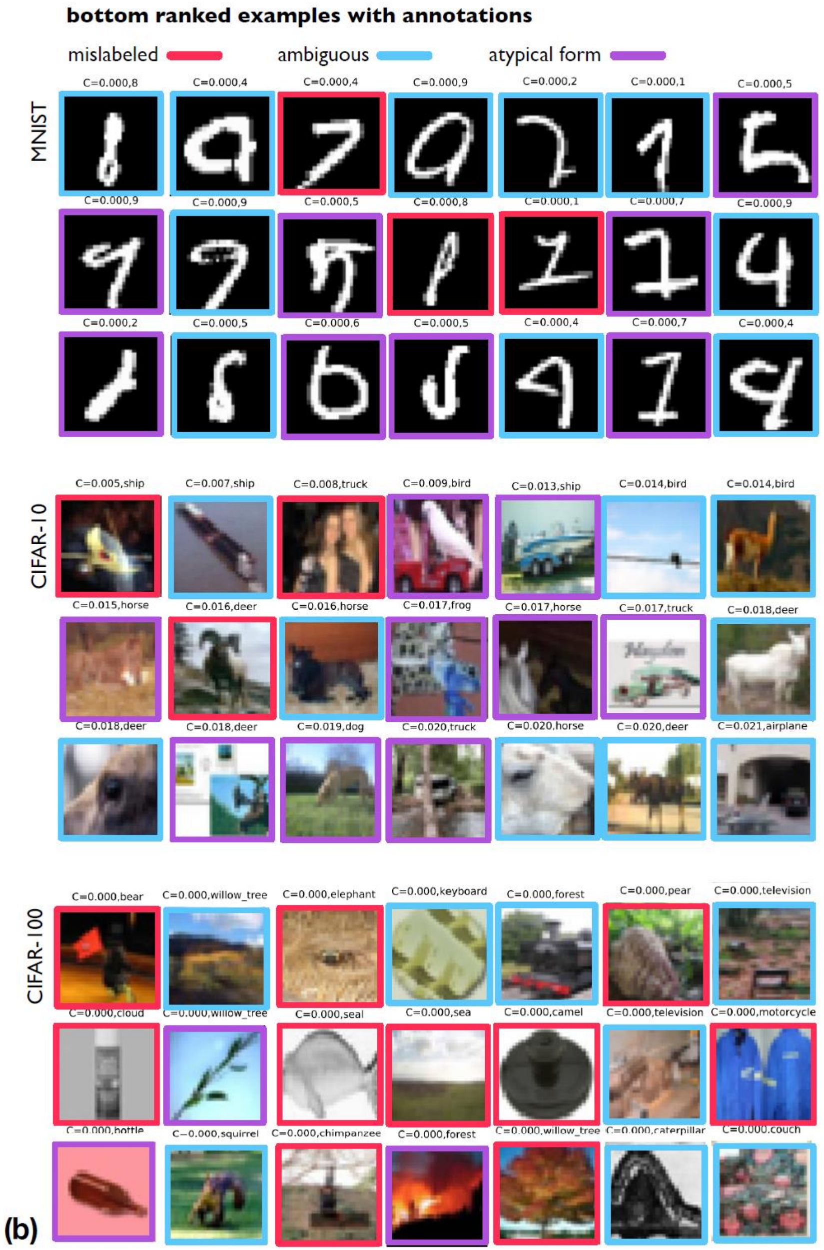


Note

- (a) : Histogram of $\hat{C}_{\hat{D},n}$ for each subset ratio s on CIFAR-10
- (b) : Histogram of the C-score averaged over all subset ratios $s \in \{10\%, \dots 90\%\}$ on 3 different data sets
- Similar to intuition, It turns out that the difficulties of data set : CIFAR-100 > CIFAR-10 > MNIST

The structural regularities of common image data set





(left) : Top ranked examples in CIFAR-10 / CIFAR-100

(right) : Bottom ranked examples with annotations

Point estimation of C-score

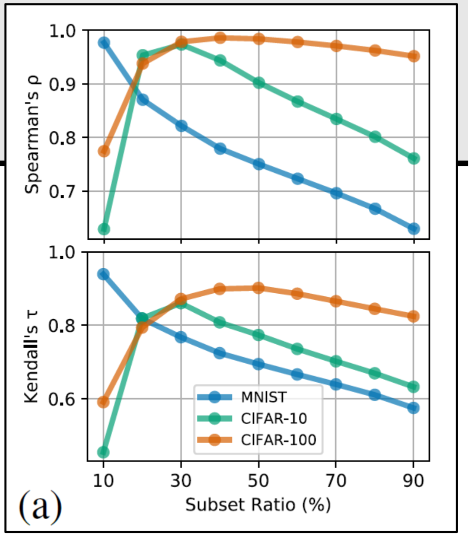
Point estimation of C-score

- C-score analysis on ImageNet data requires 10~100 times computational cost compared to CIFAR data -> Require some approximation (or estimation) of C-score

- **Suggested method : select s that best represent the integral C-score**
- For MNIST (less challenging), the correlation(between integral C-score) peak is lower ($s = 10$)
- As the dataset gets challenging, the correlation peak is higher (CIFAR-10 : $s = 40$, CIFAR-100 : $s = 50$)
- It is reasonable to choose $s = 70$ to estimate integral C-score for ImageNet

  (i.e : estimate $\hat{C}_{\hat{D}}(x, y) \cong \hat{C}_{\hat{D},0.7|\hat{D}|}(x, y)$

Rank correlation between integral C-score

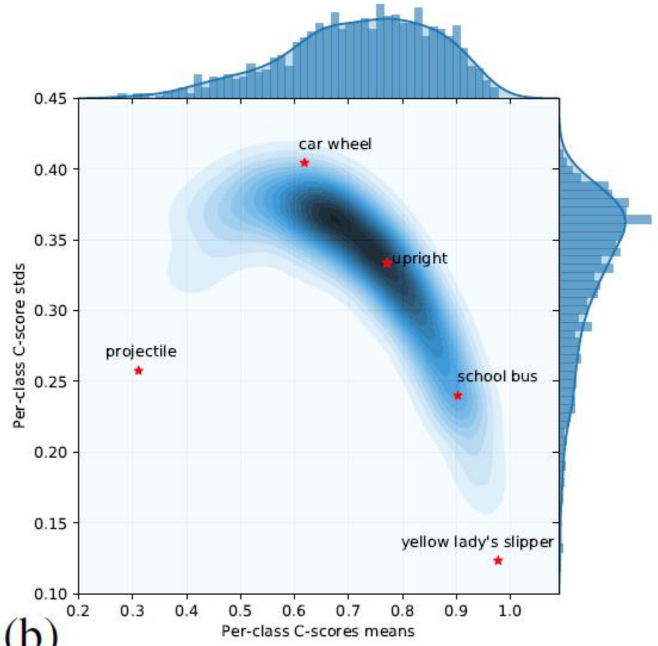


C-score across classes

C-score across classes

- How about C-score between classes?
- Calculate mean and standard deviation of the C-scores of all the examples in a particular class.
- Mean indicates the relative difficulty of classes
- Standard deviation indicates diversity of examples within each class

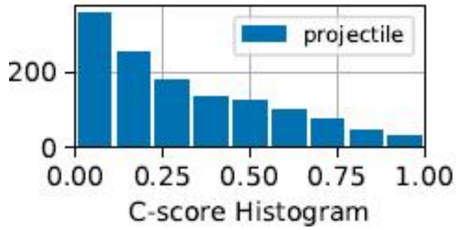
Joint distribution of C-score per-class means and standard deviations on ImageNet (1000 classes)



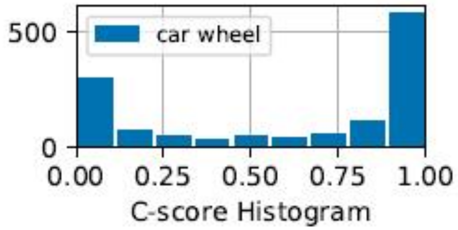
(b)

C-score across classes

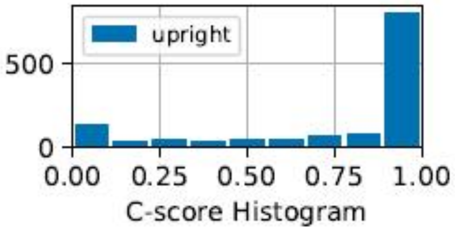
Projectile



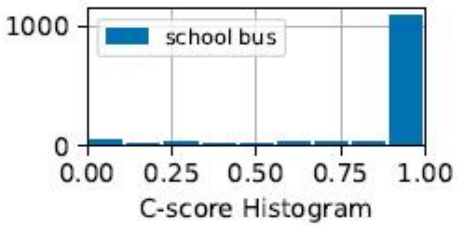
Car wheel



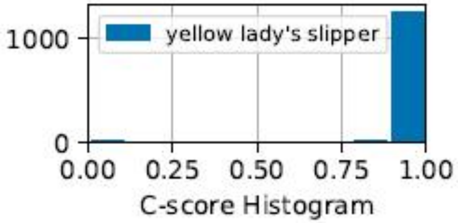
Upright



School bus



Yellow lady's slipper



Examples images from ImageNet (99%, 35%, 1% percentile ranked by C-score within each class, left to right)

C-score proxies

C-score proxies

- It is able to reduce the cost of estimating C-score from infeasible to feasible, but still very expensive due to multiple training of models.
- Is it possible to have ‘proxy’ that do not require training multiple models
- ‘proxy’ : any quantity that is well correlated with the C-score, but does not have a direct mathematical relation to it. (⇔ approximation)

- **Suggested proxy :**
  1. Pairwise distance based proxy
  2. Learning speed based proxy

Pairwise distance based proxy

Pairwise distance based proxy

- Intuitively, an example is consistent with the data distribution if it lies near other examples having the same label.
- However, if the example lies far from instances in the same class or lies near instances of different classes, we do not expect it to generalize. [Intuition of relative local-density score]
- Relative local-density score : $\hat{C}^{\pm L}(x, y) = \frac{1}{N}\sum_{i=1}^{N} 2\left(\mathbb{I}[y = y_i] - \frac{1}{2}\right) K(x_i, x)$

  where $K(x, x') = \exp(-\frac{\|x-x'\|^2}{h^2})$ is an RBF kernel with bandwidth h

- **Variants :**

  1. $\hat{C}^{L}(x, y) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[y = y_i] K(x_i, x)$
  2. $\hat{C}(x) = \frac{1}{N}\sum_{i=1}^{N} K(x_i, x)$
  3. $\hat{C}^{LOF}(x) = -LOF_{MinPts}(x)$ [Breunig, 2000] (negative value of local outlier factor)

Pairwise distance based proxy

		$\hat{C}$	$\hat{C}^L$	$\hat{C}^{\pm L}$	$\hat{C}^{\mathrm{LOF}}$
ρ	CIFAR-10	-0.064	-0.009	0.083	0.103
	CIFAR-100	-0.098	0.117	0.105	0.151
τ	CIFAR-10	-0.042	-0.006	0.055	0.070
	CIFAR-100	-0.066	0.078	0.070	0.101

Rank correlation between C-score and pairwise distance based proxies on inputs
(ρ : Spearman, τ : Kendall)

Note

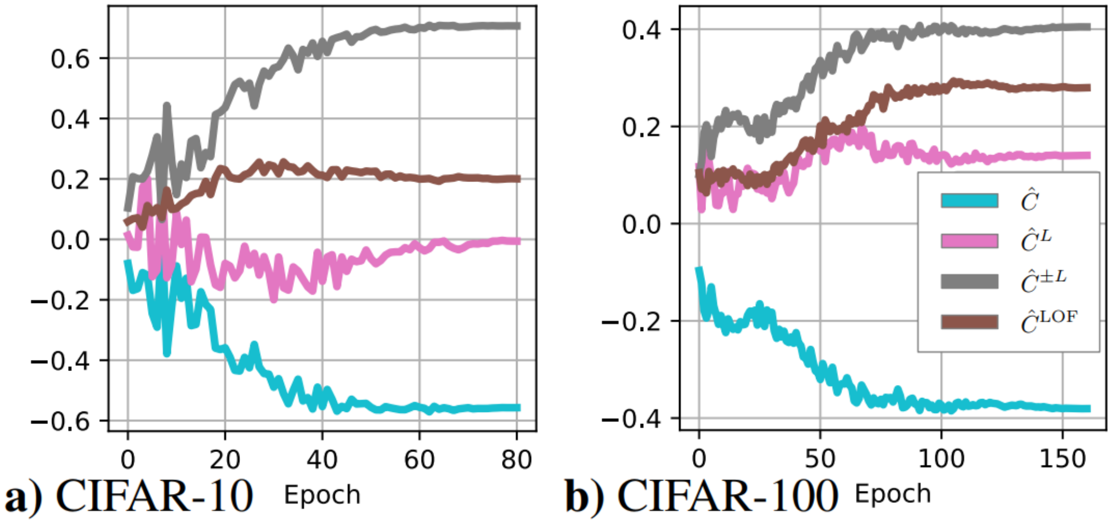
- $\hat{C}^{LOF}$ performs slightly better than the other proxies, but none of the proxies has high enough correlation to be useful

➢ Proposed reasoning : It is very hard to obtain semantically meaningful distance estimations from the raw pixels.

➢ How about evaluating the proxies using the penultimate layer as a representation of an image?

(Denote these proxies as $\hat{C}_h^{\pm L}, \hat{C}_h^L, \hat{C}_h, \hat{C}_h^{LOF}$ by appending h to mean 'hidden layer')

Pairwise distance based proxy



Spearman rank correlation between C-score and distance based proxies using learned hidden representations

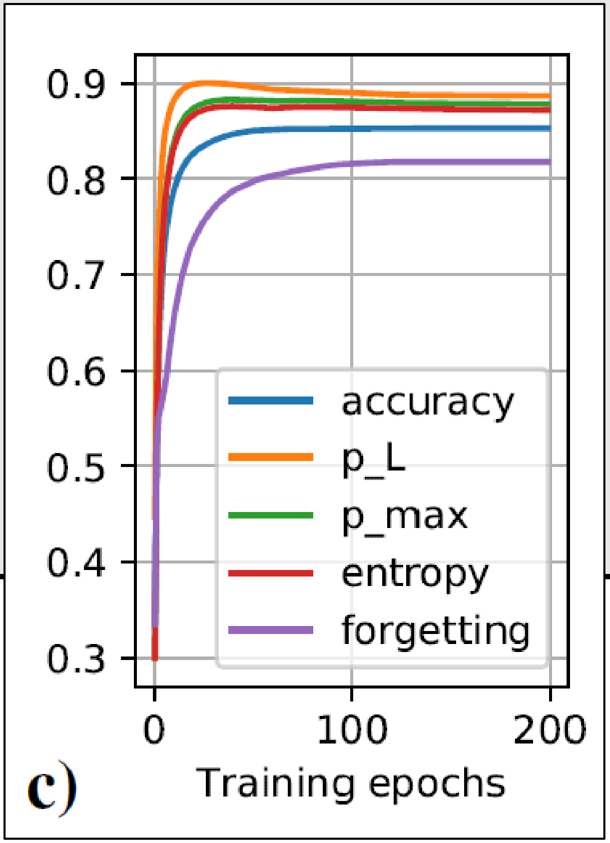
Note

- As training progress, the representation will optimize toward the classification loss and may discard inter-class relationships [Scott, 2018]
- However, the results suggest that $\hat{C}_h^{\pm L}$ does not decrease as a predictor of C-score even after training converges => Can be a good proxy to C-score

Learning speed based proxy

Learning speed based proxy

- Intuitively, a training example that is consistent with many others should be learned quickly due to aligned gradient descent step for all consistent examples.
- Note that C-score is defined after training, but learning speed based proxy is defined during training.
- Candidate proxies :
    1. Accuracy : 0-1 correctness of x
    2. p_L : softmax confidence on the correct class of x
    3. p_{max} : max softmax confidence across all classes
    4. Entropy : negative entropy of softmax confidence
    5. Forgetting event statistic
- Result : p_L reaches ρ (rank correlation) $\cong$ 0.9



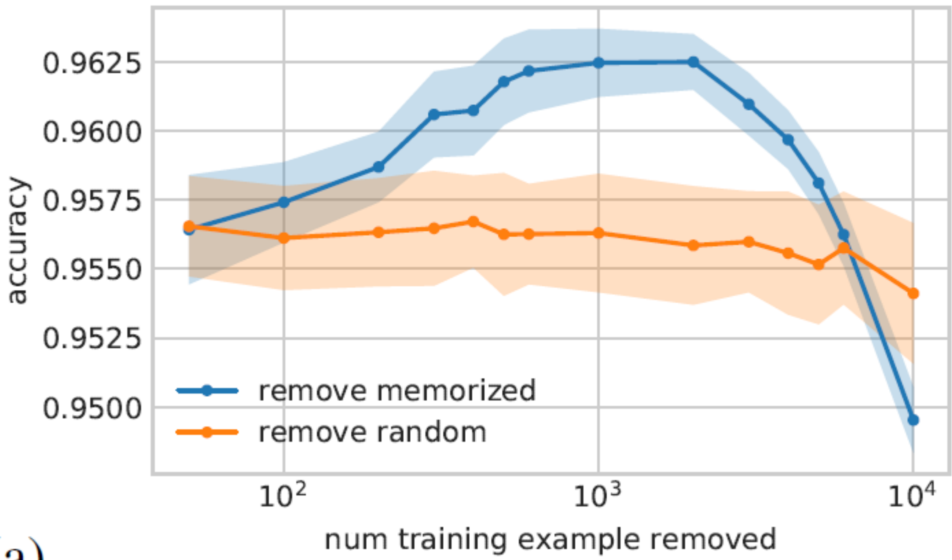
Spearman rank correlation between C-score and learning speed based proxies on CIFAR-10

Applications

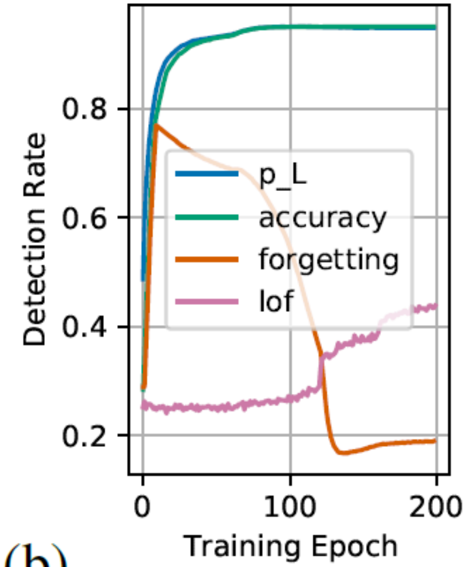
Applications
1. Removal of irregular training examples (C-score typically ranks mislabeled instances toward the bottom, followed by correctly labeled but rare instances, but removing rare instances can cause drop in performance => happen in CIFAR-10) **2. Outlier identification** (Corrupt random fraction $\gamma = 25\%$ of the CIFAR-10 training set with random label assignments, and identify the fraction γ with the lowest ranking by several proxies) **3. Study for behavior of different optimizers** (SGD with stagewise learning rate effectively enforces a sort of curriculum where the model focuses on learning the strongest regularities first, However Adam learns all examples at similar pace)

Applications

Model performance on SVHN (Removal of examples)



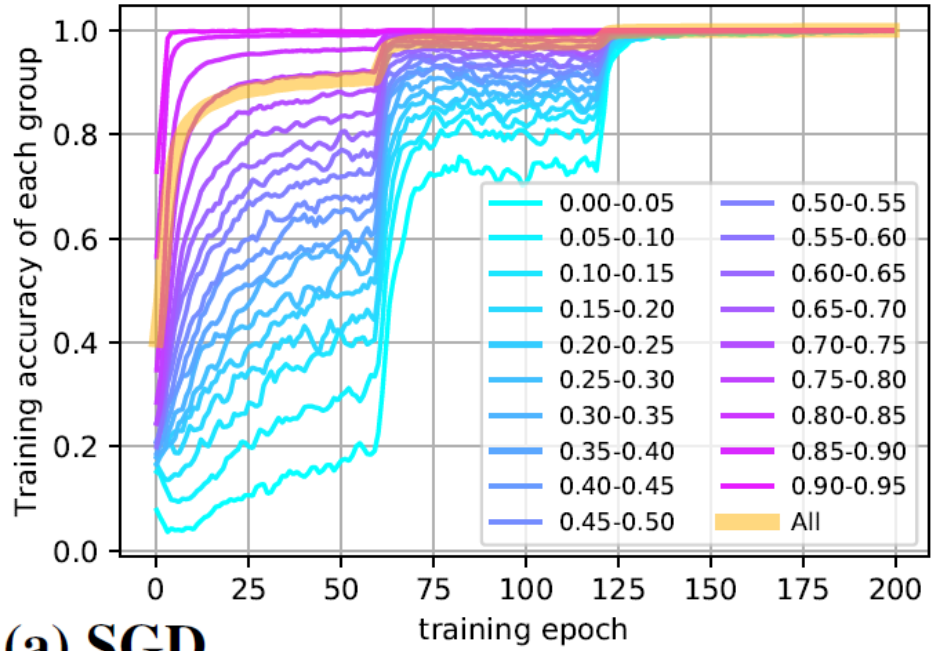
(a)



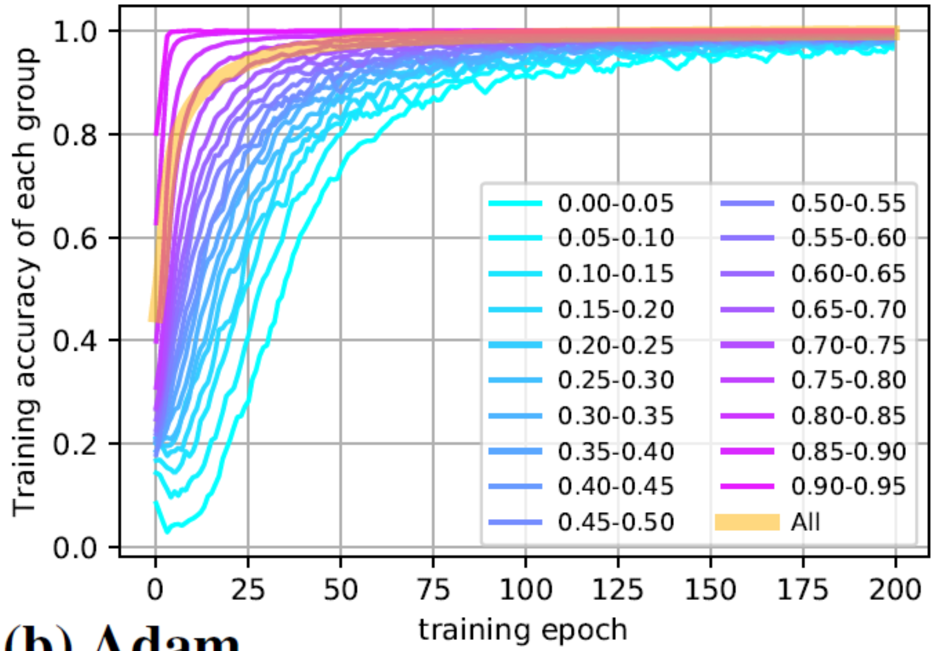
(b)

Detection rate of label-flipped outliers on CIFAR-10

Learning speed of CIFAR-10 examples grouped by C-score



(a) SGD



(b) Adam