Distributionally Robust Post-hoc Classifiers Under Prior Shifts

Jiaheng Wei💜💙, Harikrishna Narasimhan💙, Ehsan Amid💙, Wen-sheng Chu💙, Yang Liu💜, Abhishek Kumar💙

💙: Google Research
💜: University of California. Santa Cruz

ICLR, 2023

Google Research

THE UNIVERSITY OF CALIFORNIA 1868 LET THERE BE LIGHT

UNIVERSITY OF CALIFORNIA SANTA CRUZ

Brief Overview

Setting

Improve the distribution robustness when there exist prior shifts between training and test datasets, i.e.,

- Shifts in Class Prior $\mathbb{P}(Y = i)$, with label Y;
- Shifts in Group Prior $\mathbb{P}(G = i)$, with hidden attribute G;

Brief Overview

One sentence summary

A post-hoc approach that performs scaling adjustments to predictions from a pre-trained model, via minimizing a distributionally robust loss around a target distribution.

Google Research

Background

Model Robustness under
Distribution Shifts

Google Research

Distribution Shifts in Class Priors

AKA: Class-Imbalanced Learning

- **Basic Setting:**
  For an m-class classification task, denote $\{(x_j, y_j)\}_{j=1}^{n}$ the training samples drawn from $(X, Y) \sim \mathcal{D}$.
  Assume $\pi_i \coloneqq \mathbb{P}(Y = i)$ is the class prior.

- **Class-level distribution robustness:**
  Train on imbalanced π
  ↓
  Aim to perform well on a *target prior distribution* at the test time
  (Test on target π)

Google Research

Distribution Shifts in Group Priors

AKA: Group Distributional Robustness

- **Basic Setting:**
  For a m -class classification task, denote $\{(x_j, y_j, a_j)\}_{j=1}^{n}$ the training samples drawn from $(X, Y, A) \sim \mathcal{D}$.
  a_j is the attribute/group information, i.e., male/female.

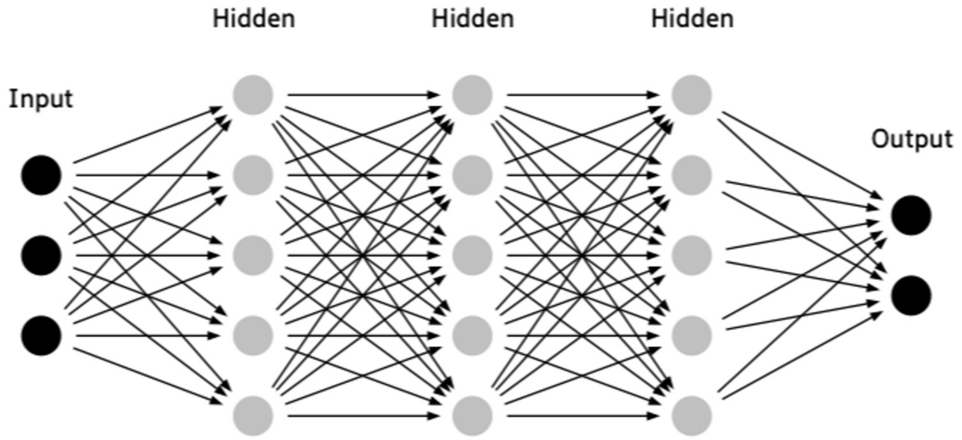
- **Group-level distribution robustness:**

Train on imbalanced group priors
(without group information in training)
↓
Aim to perform well on a *target group prior* during the test time

Motivations

Deep neural nets
capture core features well

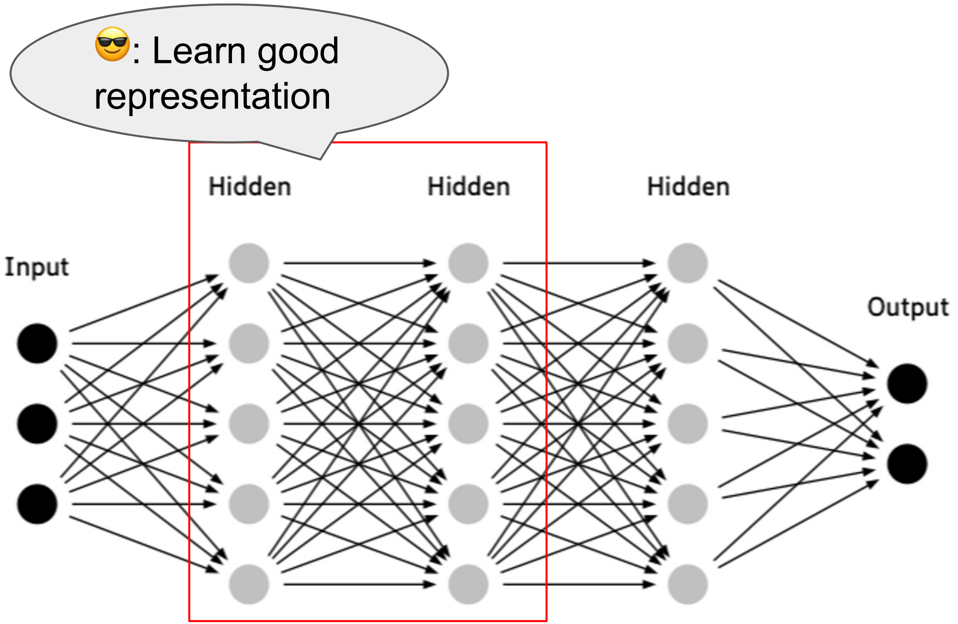
Motivation 1



Deep neural networks could learn the core features sufficiently well, even if they appear to perform poorly on minority classes/groups.

[1] Last layer re-training is sufficient for robustness to spurious correlations. [ICML 2022 workshop]

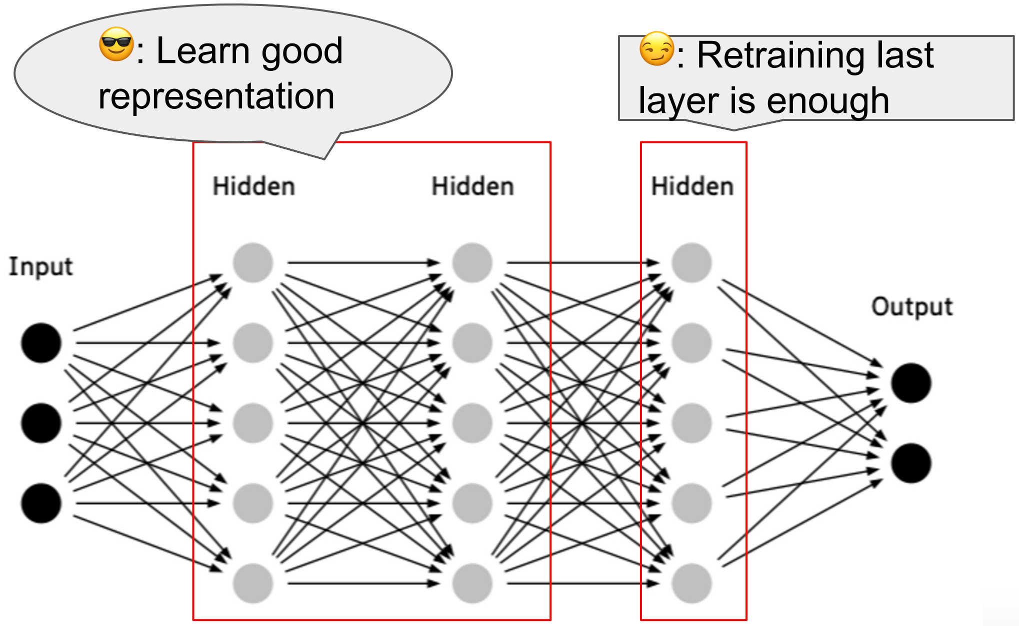
Motivation 1



Deep neural networks could learn the core features sufficiently well, even if they appear to perform poorly on minority classes/groups.

[1] Last layer re-training is sufficient for robustness to spurious correlations. [ICML 2022 workshop]

Motivation 1



Deep neural networks could learn the core features sufficiently well, even if they appear to perform poorly on minority classes/groups.

[1] Last layer re-training is sufficient for robustness to spurious correlations. [ICML 2022 workshop]

Motivation 2

Google Research

A spectrum of controlled distribution shifts:

Distribution **R**obust **E**valuation (DRE metric)

$$\text{DRE}(D, \delta): \quad \min_{g} \; g_i \, Acc_i \, , \qquad s.t. \sum_{i \in [m]} g_i = 1 \, , g_i \geq 0, D(g, u) \leq \delta.$$

Minimize the weighted sum of per-class/group accuracy Acc_i, where:

D: the divergence; u: a *target distribution*; δ: the perturbation;

Special cases

$\delta = 0$ evaluates w.r.t. any target distribution u;

$\delta = \infty$ returns the worst class/group accuracy.

Summary: DRE metric measures the worst expected accuracy in a δ-radius ball around the target distribution u.

: Improve the distributional robustness

Goal:

$$\min_{\theta} \max_{g} g_i \, \mathbb{P}_{(X,Y=i)\sim\mathcal{D}}(h_\theta(X) \neq Y),$$
$$s.t. \sum_{i\in[m]} g_i = 1, g_i \geq 0, D(g,u) \leq \delta.$$

h_θ denotes the deep neural nets.

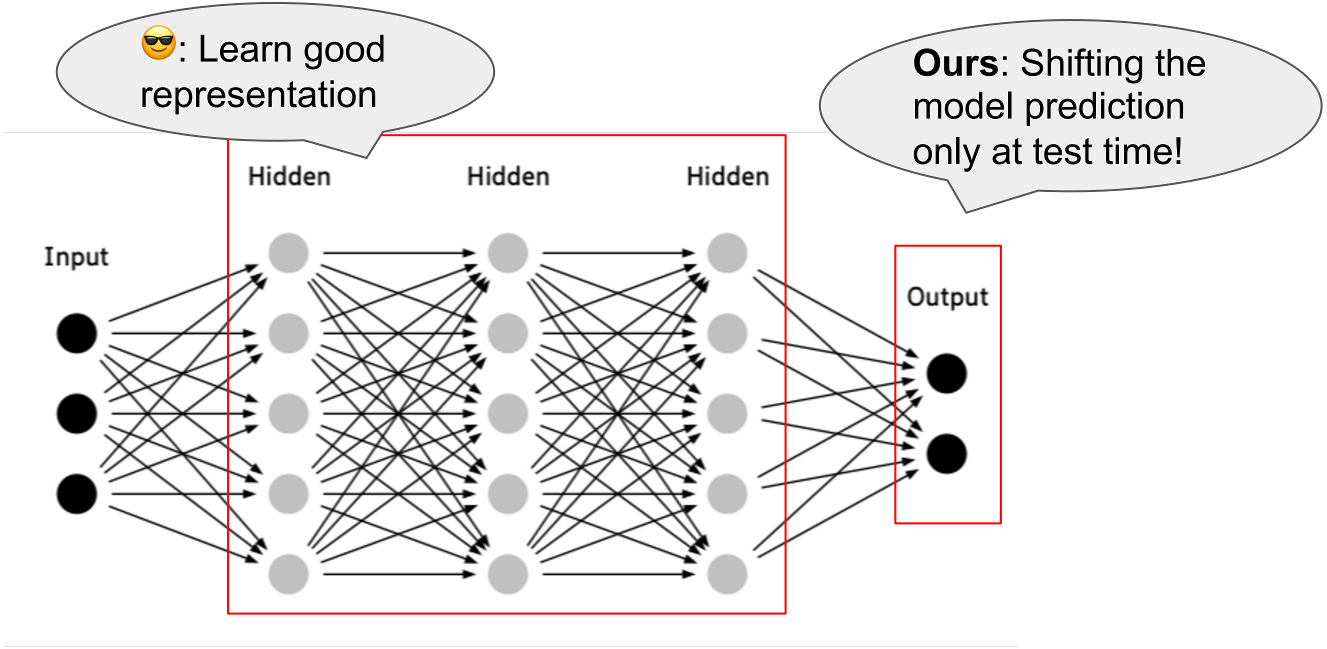
: Can we optimize the model performance under the controlled distribution shifts, by **only** shifting the model prediction at the **test time**?

Google Research

Distributional RObust PoSt-hoc Approach

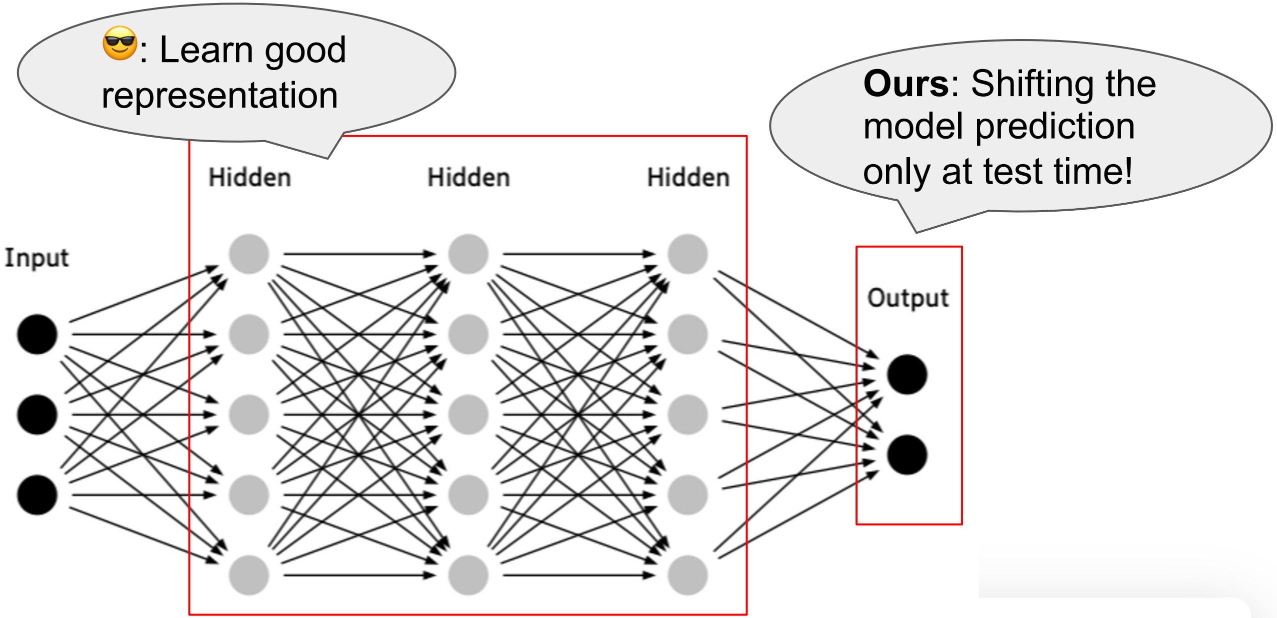
Scale the model prediction
at test time (DROPS)

DROPS



👍: Test time scaling helps with improving the robustness under any DRE metric, efficiently.

DROPS



The intuition

1. **Down-scale:**
   The model predicted probability of majority training classes at the test time.
1. **Up-scale:**
   The model predicted probability of minority training classes at the test time.

How to scale model predictions

Minimize the sum of worst-case re-weighted per-class loss

Class-imbalanced setting

$$\min_{f:\mathcal{X}\to\Delta_m} \boxed{\max_{g\in\mathbb{G}(\delta)} \sum_{i=1}^{m} g_i \ell_i(f)}. \qquad (1)$$

Generates optimal class-weights g_i^*

Clarifications

$\ell_i(f)$ is the expected loss for each class,
$\mathbb{G}(\delta) = \{g \in \Delta_m | D(g,u) \leq \delta\}$, where:
$D\colon \Delta_m \times \Delta_m \to \mathbb{R}_+$ is the divergence;
u is a *target distribution*, δ is the perturbation.

How to scale model predictions

Class-imbalanced setting

$$\textbf{DROPS:} \qquad h(x) \in \underset{i\in[m]}{\operatorname{argmax}} \frac{g_i^*}{\pi_i} \cdot \hat{\eta}_i(x)\,. \qquad (2)$$

Clarifications

$\hat{\eta}_i(x)$ is the predicted probability of sample x belonging to class i;

Insights

Model prediction of class i is upscaled by **DROPS** if:

(1) A large weight g_i^* is assigned; or (2) Class i has a small prior π_i.

An Empirical Sketch (DROPS)

Solve the Lagrangian form of Eqn. (1) via a validation set for a number of iterations.

At iteration t, do:

- **Step 1:** updating the Lagrangian multiplier $\lambda^{(t)}$;
- **Step 2:** updating the weights $g^{(t)}$;
- **Step 3:** scaling the predictions.

Experiments

On class-imbalanced learning
& Group distributional robustness

Google Research

Experiments on Class-Imbalanced Learning

Synthetic class-imbalanced CIFAR-10 & CIFAR-100

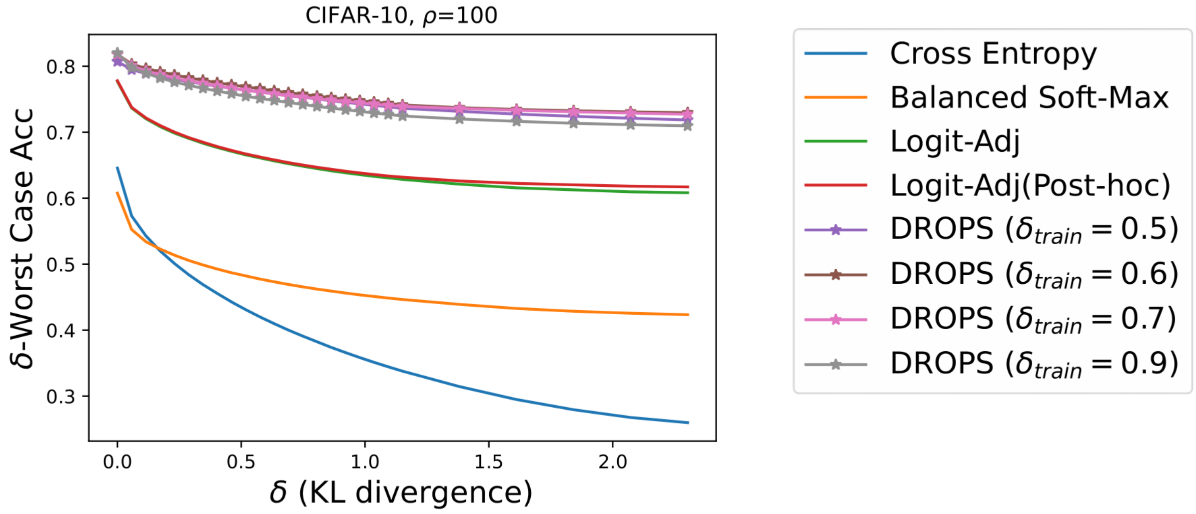
- Simulation: with the increasing of class index, # of selected samples per class has an exponential decay.
  Imbalance ratio: $\rho = \frac{\max_i \pi_i}{\min_i \pi_i}$.
- Train on imbalanced dataset; validate and test on balanced datasets.
- For DROPS, get g_i^* under different perturbation level δ_{train}, under D_{KL}.
- Performance evaluation: for both D_{KL} and $D_{\text{R-KL}}$, report the model performance in the metric $\text{DRE}(D, \delta_{\text{test}})$, with a list of δ_{test}.

Google Research

Experiments on Class-Imbalanced Learning

CIFAR-10 (imbalance ratio $\rho = 100$) – under DRE$(D_{\mathrm{KL}}, \delta_{\mathrm{test}})$

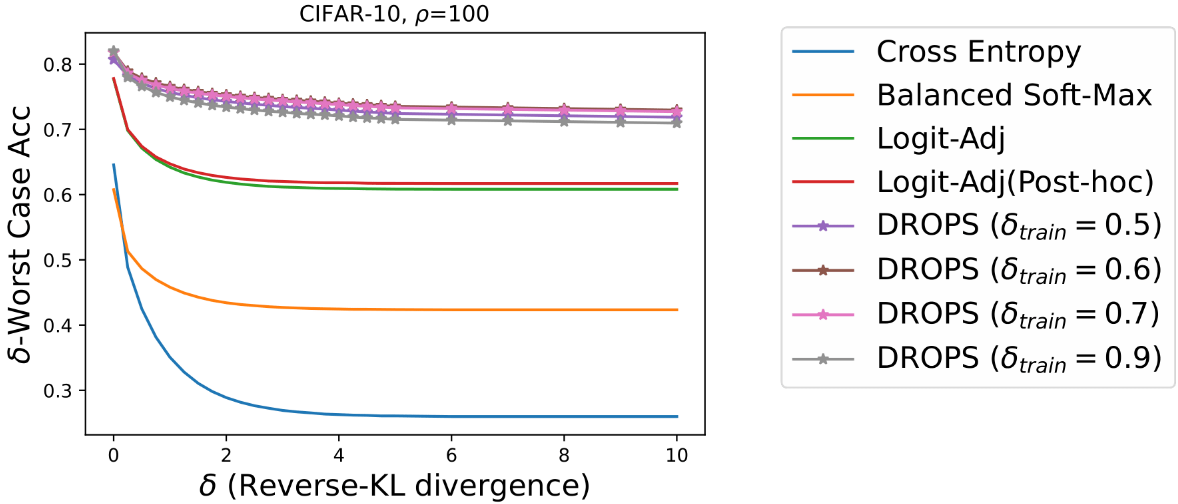
X axis: divergence from the target distribution δ_{test};
Y-axis: DRE$(D, \delta_{\mathrm{test}})$; Higher curve → More robustness



Google Research

Experiments on Class-Imbalanced Learning

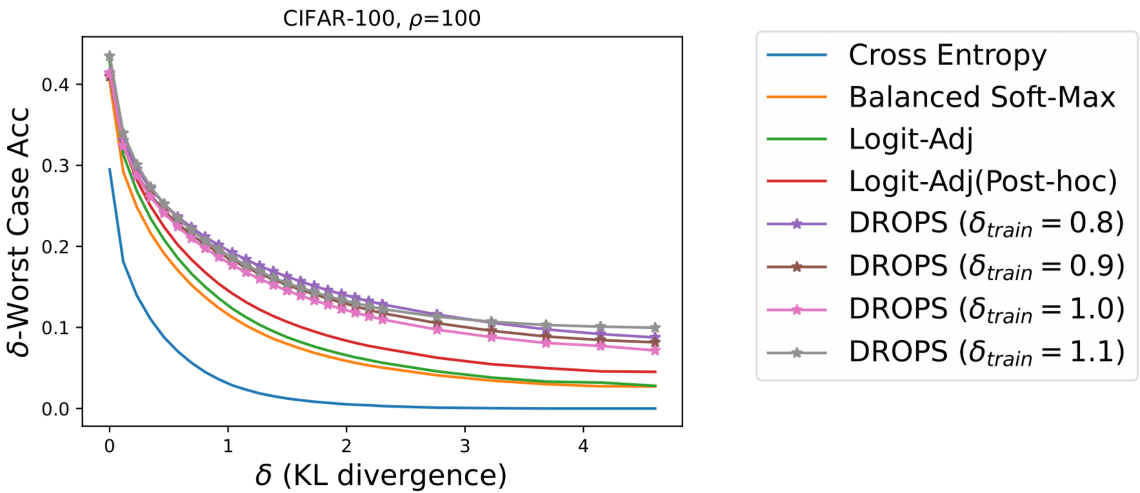
CIFAR-10 (imbalance ratio $\rho = 100$) – under DRE$(D_{\mathrm{R-KL}}, \delta_{\mathrm{test}})$



Google Research

Experiments on Class-Imbalanced Learning

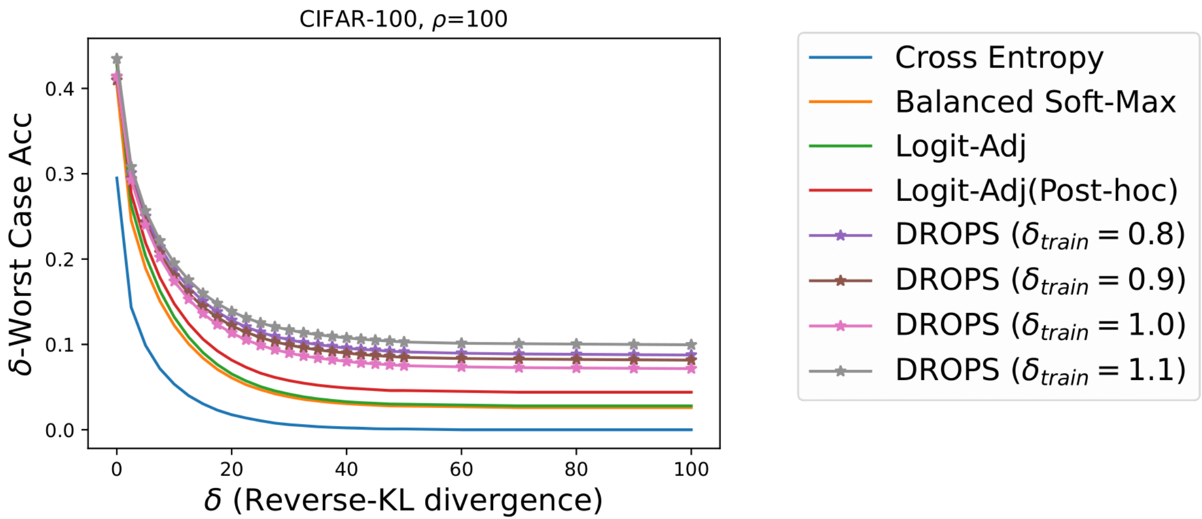
CIFAR-100 (imbalance ratio $\rho = 100$) – under DRE$(D_{\mathrm{KL}}, \delta_{\mathrm{test}})$



Google Research

Experiments on Class-Imbalanced Learning

CIFAR-100 (imbalance ratio $\rho = 100$) – under DRE$(D_{\mathrm{R-KL}}, \delta_{\mathrm{test}})$



Experiments on Group Distributional Robustness (DRO)

Table 2: Performance comparisons on Waterbirds and CelebA: averaged accuracy with training prior weight and uniform weight, δ-worst accuracy for several δ perturbations, and worst group-level test accuracy are reported. The *Group Info* column indicates whether the group labels are available to the methods on train/validation sets. The symbol ✓✓means the method also re-trains the last layer on the validation set (w/o needing the group information for training).

Method	Group Info Train	Group Info Val	(Train prior) Averaged	*Waterbirds* ($r \to$ uniform prior) Averaged	$\delta = 0.05$-**Worst**	$\delta = 0.1$-**Worst**	$\delta = 0.2$-**Worst**	$\delta = 0.5$-**Worst**	**Worst**
ERM	✗	✓	98.08±0.20	88.09±0.90	84.31±1.24	82.71±1.41	80.51±1.68	76.65±2.25	70.65±3.32
JTT	✗	✓	93.05±0.36	89.56±0.69	88.59±0.74	88.24±0.75	87.81±0.77	87.13±0.85	86.18±1.08
DROPS	✗	✓	97.95±0.16	88.55±1.15	85.50±1.51	84.33±1.64	82.81±1.79	80.41±2.00	77.14±2.15
G-DRO	✓	✓	93.03±0.34	91.67±0.22	91.23±0.33	91.06±0.34	90.83±0.40	90.44±0.52	89.85±0.73
SUBG	✓	✓	91.97±0.50	90.05±0.44	89.46±0.40	89.24±0.39	88.98±0.40	88.59±0.49	88.12±0.76
$\text{DFR}^{\text{Tr}}_{\text{Tr}}$	✓	✓	95.83±0.94	93.45±0.49	92.77±0.48	92.51±0.51	92.16±0.55	91.58±0.67	90.72±0.91
$\text{DFR}^{\text{Val}}_{\text{Tr}}$	✗	✓✓	93.17±1.30	93.29±0.80	92.98±0.84	92.86±0.85	92.70±0.87	92.42±0.88	92.01±0.88
DROPS*	✗	✓✓	93.01±1.32	93.42±0.61	93.08±0.81	92.98±0.90	92.76±1.02	92.45±1.23	91.99±1.56

Method	Group Info Train	Group Info Val	(Train prior) Averaged	*CelebA* ($r \to$ uniform prior) Averaged	$\delta = 0.05$-**Worst**	$\delta = 0.1$-**Worst**	$\delta = 0.2$-**Worst**	$\delta = 0.5$-**Worst**	**Worst**
ERM	✗	✓	95.33±0.12	81.18±1.60	73.78±2.36	70.53±2.69	65.97±3.15	57.82±3.98	44.89±5.30
JTT	✗	✓	87.78±0.73	85.04±1.05	83.34±1.35	82.89±1.49	82.35±1.71	81.51±2.12	79.71±2.85
DROPS	✗	✓	90.67±0.76	89.05±0.88	86.80±1.34	85.89±1.55	84.67±1.88	82.59±2.50	79.44±3.58
G-DRO	✓	✓	92.59±0.87	90.95±0.52	90.18±0.46	89.89±0.43	89.53±0.37	88.97±0.26	88.21±0.17
SUBG	✓	✓	91.09±0.62	88.75±0.34	87.69±0.31	87.27±0.41	86.72±0.73	85.80±0.29	84.45±0.28
$\text{DFR}^{\text{Tr}}_{\text{Tr}}$	✓	✓	90.36±0.64	89.25±0.73	87.30±0.97	86.53±1.09	85.50±1.26	83.77±1.63	81.22±2.31
$\text{DFR}^{\text{Val}}_{\text{Tr}}$	✗	✓✓	90.90±0.79	91.13±0.40	90.32±0.54	90.02±0.58	89.64±0.65	89.05±0.77	88.25±1.03
DROPS*	✗	✓✓	95.69±0.41	93.59±0.36	92.64±0.46	92.28±0.51	91.82±0.58	91.10±0.68	90.19±0.81

Guidelines for More Results

Theoretical Results

- Theorem 1 and Lemma 2: the optimal scaling of DROPS;
- Section 4.2: empirical implementation of DROPS;
- Theorem 3: convergence analysis of DROPS.

Empirical Results

- Table1: Experiments results of class-imbalanced CIFAR datasets;
- Table2: Experiments of group distributional robustness on Waterbirds, CelebA.

Paper and Code

Thanks for watching!

Paper

Code