Hierarchically Robust Representation Learning

Qi Qian[1] Juhua Hu[2] Hao Li[1]
[1]Alibaba Group
[2]School of Engineering and Technology
University of Washington, Tacoma, USA
{qi.qian, lihao.lh}@alibaba-inc.com juhuah@uw.edu

Abstract

With the tremendous success of deep learning in visual tasks, the representations extracted from intermediate layers of learned models, that is, deep features, attract much attention of researchers. Previous empirical analysis shows that those features can contain appropriate semantic information. Therefore, with a model trained on a large-scale benchmark data set (e.g., ImageNet), the extracted features can work well on other tasks. In this work, we investigate this phenomenon and demonstrate that deep features can be suboptimal due to the fact that they are learned by minimizing the empirical risk. When the data distribution of the target task is different from that of the benchmark data set, the performance of deep features can degrade. Hence, we propose a hierarchically robust optimization method to learn more generic features. Considering the example-level and concept-level robustness simultaneously, we formulate the problem as a distributionally robust optimization problem with Wasserstein ambiguity set constraints, and an efficient algorithm with the conventional training pipeline is proposed. Experiments on benchmark data sets demonstrate the effectiveness of the robust deep representations.

1. Introduction

Extracting appropriate representations is essential for visual recognition. In the past decades, various hand-crafted features have been developed to capture semantics of images, e.g., SIFT [16], HOG [7], etc. The conventional pipeline works in two phases. In the first phase, representations are extracted from each image with a given schema. Thereafter, a specific model (e.g., SVM [6]) is learned with these features for a target task. Since the hand-crafted features are task-independent, the performance of this pipeline can be suboptimal.

Deep learning proposes to incorporate these phases by training end-to-end convolutional neural networks. Without an explicit feature design like SIFT [16], a task-dependent representation will be learned through multiple layers and a fully connected layer is attached at the end as a linear classifier for recognition. Benefited from this coherent structure, deep learning promotes the performance on visual tasks dramatically, e.g., categorization [15], detection [21], etc. Despite the success of deep learning on large-scale data sets, deep neural networks (DNNs) are easy to overfit small data sets due to the large number of parameters. Besides, DNNs require GPU for efficient training, which is expensive.

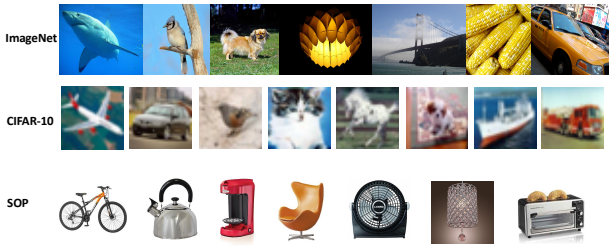


Figure 1. Examples from ImageNet, CIFAR-10 and SOP. Example-level distribution difference within a class can be observed from the 7th image of ImageNet and 2nd image of CIFAR-10 for the car class in various aspects, e.g., resolution and pose. Concept-level distribution difference is significant between ImageNet and SOP. ImageNet includes many classes from the concept "animal" while SOP only contains classes from "artifact".

Researchers attempt to leverage pre-trained DNNs to improve the feature design mechanism. Surprisingly, it is observed that the features extracted from the last few layers perform well on the generic tasks when the model is pre-trained on a large-scale benchmark data set, e.g., ImageNet [22]. Deep features, which are outputs from intermediate layers of a deep model, become popular as the substitute of training deep models for light computation. Systematic comparison shows that these deep features outperform the existing hand-crafted features with a large margin [8, 17, 20].

The objective of learning deep models for specific tasks and deep features for generic tasks can be different, but little efforts have been devoted to further investigating deep fea-

tures. When learning deep models, it focuses on optimizing the performance on the current training data set. In contrast, deep features should be learned for generic tasks rather than a single data set. In the applications of deep features, it is also noticed that the deep features can fail when the data distribution in a generic task is different from the benchmark data set [28]. By studying the objective of learning models for a given task, we find that it is a standard empirical risk minimization (ERM) problem that is optimized on the uniform distribution over examples. It is well known that the models obtained by ERM can generalize well on the data from the same distribution as training [3].

However, the data distribution from real applications can be significantly different from a benchmark data set, which can result in the performance degeneration when adopting the representations learned from ERM. The differences can come from at least two aspects. First, the distribution of examples in each class can be different between the generic task and the benchmark data set, which is referred as example-level distribution difference in this paper. Taking the 7th image of ImageNet and 2nd of CIFAR-10 in Fig. 1 as an example, they are of different resolutions and poses while they are both from the car class. This problem attracts much attention recently and some approaches to optimize the worst-case performance are developed to handle this issue [5, 18, 24]. Second, the distribution of concepts in an application is also different from that in the benchmark data set. It should be noted that each concept here can contain multiple classes, e.g., bulldog, beagle and so on under the concept "dog". This concept-level distribution difference has been less investigated but more crucial for deploying deep features due to the fact that the concepts in real applications may be only a subset of or partially overlapped by those in the benchmark data set. For instance, the concepts in SOP is quite different from those covered in ImageNet as shown in Fig. 1.

In this work, we propose to consider the difference in examples and that in concepts simultaneously and learn hierarchically robust representations from DNNs. Compared with ERM, our algorithm is more consistent with the objective of learning generic deep features. For the example-level robustness, we adopt Wasserstein ambiguity set [24] to encode the uncertainty from examples for the efficient optimization. Our theoretical analysis also illustrates that an appropriate augmentation can be better than the regularization in training DNNs, since the former one provides a tighter approximation for the optimization problem. For the concept-level robustness, we formulate it as a game between the deep model and the distribution over different concepts to optimize the worst-case performance over concepts. By learning deep features with the adversarial distribution, the worst-case performance over concepts can be improved. Finally, to keep the simplicity of the training pipeline, we develop an algorithm that leverages the standard random sampling strategy at each iteration and reweights the obtained gradient for an unbiased estimation. This step may increase the variance of the gradient and we reduce the variance by setting the learning rate elaborately. We show that the adversarial distribution can converge at the rate of $\mathcal{O}(\log(T)/T)$, where T denotes the total number of iterations. We employ ImageNet as a benchmark data set for learning deep features and the empirical study on real-world data sets confirms the effectiveness of our method.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the proposed method. Section 4 conducts the experiments on the benchmark data sets and Section 5 concludes this work with future directions.

2. Related Work

Deep Features: Deep learning becomes popular since ImageNet ILSVRC12 and various architectures of DNNs have been proposed, e.g., AlexNet [15], VGG [23], GoogLeNet [27], and ResNet [12]. Besides the success on image categorization, features extracted from the last few layers are applied for generic tasks. [8] adopts the deep features from the last two layers in AlexNet and shows the impressive performance on visual recognition with different applications. After that, [20] applies deep features for distance metric learning and achieves the overwhelming performance to the hand-crafted features on fine-grained visual categorization. [17] compares deep features from different neural networks and ResNet shows the best results. Besides the model pre-trained on ImageNet, [28] proposes to learn deep features with a large-scale scene data set to improve the performance on the scene recognition task. All of these work directly extract features from the model learned with ERM as the objective. In contrast, we develop an algorithm that is tailored to learn robust deep representations. Note that deep features can be extracted from multiple layers of deep models and we focus on the layer before the final fully-connected layer in this work.

Robust Optimization: Recently, distributionally robust optimization that aims to optimize the worst-case performance has attracted much attention [5, 18, 24]. [18] proposes to optimize the performance with worst-case distribution over examples that is derived from the empirical distribution. [5] extends the problem to a non-convex loss function, but they require a near-optimal oracle for the non-convex problem to learn the robust model. [24] introduces the adversarial perturbation on each example for robustness. Most of these algorithms only consider the example-level robustness. In contrast, we propose the hierarchically robust optimization that considers the example-level and concept-level robustness simultaneously, to learn the generic deep representations for real applications.

3. Hierarchical Robustness

3.1. Problem Formulation

Let $\mathbf{x}_i$ denote an image and $y_i \in \{1, \ldots, C\}$ be its corresponding label for a C-class classification problem. Given a benchmark data set $\{\mathbf{x}_i, y_i\}$ where $i = 1, \ldots, N$, the parameter θ in a deep neural network can be learned by solving the optimization problem as

$$\min_{\theta} \frac{1}{N} \sum_i \ell(\mathbf{x}_i, y_i; \theta) \tag{1}$$

where $\ell(\cdot)$ is a non-negative loss function (e.g., cross entropy loss). By decomposing the parameter θ as $\theta = \{\delta, \omega\}$, where ω denotes the parameter of the final fully-connected layer and δ denotes the parameter from other layers and can be considered as for a feature extraction function $f(\cdot)$, we can rewrite the original problem as

$$\min_{\theta} \frac{1}{N} \sum_i \ell(f(\mathbf{x}_i), y_i; \omega)$$

Considering that ω is for a linear classifier, which is consistent to the classifiers applied in real-world applications (e.g., SVM), the decomposition shows that the problem of learning generic deep features $f(\mathbf{x})$ can be addressed by learning a robust deep model on the benchmark data set.

The original problem in Eqn. 1 is an empirical risk minimization (ERM) problem that can be inappropriate for learning generic representations. In the following, we explore the hierarchical robustness to obtain robust deep representations for generic tasks.

First, we consider the example-level robustness. Unlike ERM, a robust model is to minimize the loss with the worst-case distribution derived from the empirical distribution. The optimization problem can be cast as a game between the prediction model and the adversarial distribution

$$\min_{\theta} \max_{i} \{\ell(\mathbf{x}_i, y_i; \theta)\}$$

which is equivalent to

$$\min_{\theta} \max_{\mathbf{p} \in \mathbb{R}^N; \mathbf{p} \in \Delta} \sum_i p_i \ell(\mathbf{x}_i, y_i; \theta)$$

where $\mathbf{p}$ is the adversarial distribution over training examples and Δ is the simplex as $\Delta = \{\mathbf{p} | \sum_i p_i = 1, \forall i, p_i \geq 0\}$. When $\mathbf{p}$ is a uniform distribution, the distributioanlly robust optimization becomes ERM.

Without any constraints, the adversarial distribution is sensitive to the outlier and can be arbitrarily far way from the empirical distribution, which has large variance from the selected examples. Therefore, we introduce a regularizer to constrain the space of the adversarial distribution, which provides a trade-off between the bias (i.e., to the empirical distribution) and variance for the adversarial distribution. The problem can be written as

$$\min_{\theta} \max_{\mathbf{p} \in \mathbb{R}^N; \mathbf{p} \in \Delta} \sum_i p_i \ell(\mathbf{x}_i, y_i; \theta) - \lambda_e \mathcal{D}(\mathbf{p} || \mathbf{p}_0) \tag{2}$$

where $\mathbf{p}_0$ is the empirical distribution. $\mathcal{D}(\cdot)$ measures the distance between the learned adversarial distribution and the empirical distribution. We apply squared L_2 distance in this work as $\mathcal{D}(\mathbf{p} || \mathbf{p}_0) = \|\mathbf{p} - \mathbf{p}_0\|_2^2$. The regularizer is to guarantee that the generated adversarial distribution is not too far way from the empirical distribution. It implies that the adversarial distribution is from an ambiguity set as

$$\mathbf{p} \in \{\mathbf{p} : \mathcal{D}(\mathbf{p} || \mathbf{p}_0) \leq \epsilon\}$$

where ϵ is determined by λ_e.

Besides the example-level robustness, concept-level robustness is more important for learning the generic features. A desired model should perform consistently well over different concepts. Assuming that there are K concepts in the training set and each concept consists of N_k examples, the concept-robust optimization problem is

$$\min_{\theta} \max_{k} \{\frac{1}{N_k} \sum_i^{N_k} \ell(\mathbf{x}_i^k, y_i^k; \theta)\}$$

With the similar analysis as the example-level robustness and adopting the appropriate regularizer, the problem becomes

$$\min_{\theta} \max_{\mathbf{q} \in \mathbb{R}^K; \mathbf{q} \in \Delta} \sum_k \frac{q_k}{N_k} \sum_i^{N_k} \ell(\mathbf{x}_i^k, y_i^k; \theta) - \lambda_c \mathcal{D}(\mathbf{q} || \mathbf{q}_0) \tag{3}$$

where $\mathbf{q}_0$ can be set as $q_0^k = N_k / N$.

Combined with the example-level robustness, the hierarchically robust optimization problem becomes

$$\min_{\theta} \max_{\substack{\mathbf{p} \in \mathbb{R}^N; \mathbf{p} \in \Delta \\ \mathbf{q} \in \mathbb{R}^K; \mathbf{q} \in \Delta}} \sum_k \frac{q_k}{N_k} \sum_i^{N_k} p_i \ell(\mathbf{x}_i^k, y_i^k; \theta) - \lambda_e \mathcal{D}(\mathbf{p} || \mathbf{p}_0) - \lambda_c \mathcal{D}(\mathbf{q} || \mathbf{q}_0)$$

In this formulation, each example is associated with a parameter p_i and q_k. Therefore, a high dimensionality with this coupling structure makes an efficient optimization challenging. Due to the fact that $K \ll N$, we decouple the hierarchical robustness with an alternative formulation for the example-level robustness as follows.

3.2. Wasserstein Ambiguity Set

In Eqn. 2, the ambiguity set is defined with the distance to the uniform distribution over the training set. It introduces the adversarial distribution by re-weighting each

example, which couples the parameter with that of the concept-level problem. To simplify the optimization, we generate the ambiguity set for the adversarial distribution with Wasserstein distance [24]. The property of Wasserstein distance can help to decouple the example-level robustness from concept-level robustness.

Assume that P is a data-generating distribution over the data space and P_0 is the empirical distribution from where the training set is generated as $\mathbf{x} \sim P_0$. The ambiguity set for the distribution P can be defined as

$$\{P : W(P, P_0) \leq \epsilon\}$$

$W(P, P_0) = \inf_{M \in \Pi(P, P_0)} E_M[d(\hat{\mathbf{x}}, \mathbf{x})]$ is the Wasserstein distance between distributions [24] and we denote the example generated from P as $\hat{\mathbf{x}}$. $d(\cdot, \cdot)$ is the transportation cost between examples.

The problem of example-level robustness can be written as

$$\min_{\theta} \max_{P} E_P[\ell(\hat{\mathbf{x}}, y; \theta)] - \frac{\lambda_w}{2} W(P, P_0)$$

According to the definition of Wasserstein distance [24] and let the cost function be the squared Euclidean distance, the problem is equivalent to

$$\min_{\theta} \max_{\hat{\mathbf{x}} \in \mathbf{X}} \sum_i \ell(\hat{\mathbf{x}}_i, y_i; \theta) - \frac{\lambda_w}{2} \sum_i \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_F^2$$

where $\mathbf{X}$ is the data space. In [24], they obtain the optimal $\hat{\mathbf{x}}_i$ by solving the subproblem for each example at each iteration. To accelerate the optimization, we propose to minimize the upper bound of the subproblem, which also provides an insight for the comparison between regularization and augmentation.

The main theoretical results are stated in the following theorems and their proofs can be found in the supplementary. First, we give the definition of smoothness as

Definition 1. *A function f is called L_z-smoothness in z w.r.t. a norm $\|\cdot\|$ if there is a constant L_z such that for any values of z as z' and z'', it holds that*

$$f(z'') \leq f(z') + \langle \nabla f(z'), z'' - z' \rangle + \frac{L_z}{2} \|z'' - z'\|^2$$

Theorem 1. *Assuming $\ell(\cdot)$ is $L_{\mathbf{x}}$-smoothness in $\mathbf{x}$ and $\nabla_{\mathbf{x}} \ell$ is L_θ-Lipschitz continuous for θ, we have*

$$\max_{\hat{\mathbf{x}}_i \in \mathbf{X}} \ell(\hat{\mathbf{x}}_i, y_i; \theta) - \frac{\lambda_w}{2} \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_F^2 \leq \ell(\mathbf{x}_i, y_i; \theta) + \frac{\gamma}{2} \|\theta\|_F^2$$

where λ_w is sufficiently large such that $\lambda_w > L_{\mathbf{x}}$ and $\gamma = \frac{L_\theta^2}{\lambda_w - L_{\mathbf{x}}}$.

Theorem 2. *With the same assumption in Theorem 1 and considering an additive augmentation with $\mathbf{z}$ for the original image*

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i + \tau \mathbf{z}_i$$

we have

$$\max_{\hat{\mathbf{x}}_i \in \mathbf{X}} \ell(\hat{\mathbf{x}}_i, y_i; \theta) - \frac{\lambda_w}{2} \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_F^2 \leq \ell(\tilde{\mathbf{x}}_i, y_i; \theta) + \frac{\gamma}{2} \|\theta\|_F^2 - \alpha$$

where

$$\tau = \frac{\langle \nabla_{\mathbf{x}_i} \ell, \mathbf{z}_i \rangle}{3 L_{\mathbf{x}} \|\mathbf{z}_i\|_F^2}$$

and α is a non-negative constant as

$$\alpha = \frac{\lambda_w}{\lambda_w - L_{\mathbf{x}}} \frac{\langle \nabla_{\mathbf{x}_i} \ell, \mathbf{z}_i \rangle^2}{6 L_{\mathbf{x}} \|\mathbf{z}_i\|_F^2}$$

Theorem 1 shows that learning the model using the original examples with a regularization on the complexity of the model, e.g., weight decay with γ, can make the learned model robust for examples from the ambiguity set. A similar result has been observed in the conventional robust optimization [1]. However, the regularization is not sufficient to train good enough DNNs and many optimization algorithms have to rely on augmented examples to obtain models with better generalization performance.

Theorem 2 interprets the phenomenon by analyzing a specific augmentation that adds a patch $\mathbf{z}$ to the original image and shows that augmented examples can provide a tighter bound for the loss of the examples in the ambiguity set. Besides, the augmented patch $\mathbf{z}_i$ is corresponding to the gradient of the original example $\mathbf{x}_i$. To make the approximation tight, it should be identical to the direction of the gradient. So we set $\mathbf{z}_i = \frac{\nabla_{\mathbf{x}_i} \ell}{\|\nabla_{\mathbf{x}_i} \ell\|_F}$, which is similar to that in adversarial training [11].

Combining with the concept-level robustness in Eqn. 3, we have the final objective for learning the hierarchically robust representations as

$$\min_{\theta} \max_{\mathbf{q} \in \mathbb{R}^K; \mathbf{q} \in \Delta} \mathcal{L}(\mathbf{q}, \theta) = \sum_k \frac{q_k}{N_k} \sum_i \ell(\tilde{\mathbf{x}}_i^k, y_i^k; \theta) + \frac{\gamma}{2} \|\theta\|_F^2 - \frac{\lambda}{2} \|\mathbf{q} - \mathbf{q}_0\|_2^2 \quad (4)$$

3.3. Efficient Optimization

The problem in Eqn. 4 can be solved efficiently by stochastic gradient descent (SGD). In the standard training pipeline for ERM in Eqn. 1, a mini-batch of examples are randomly sampled at each iteration and the model is updated with gradient descent as

$$\theta_{t+1} = \theta_t - \eta_\theta \frac{1}{m} \sum_i^m \nabla_\theta \ell(\mathbf{x}_i, y_i; \theta_t)$$

where m is the size of a mini-batch.

For the problem in Eqn. 4, each example has a weight as q_k / N_k and the gradient has to be weighted for an unbiased estimation as

$$\theta_{t+1} = \theta_t - \eta_\theta (\frac{1}{m} \sum_i^m \frac{N}{N_k} q_k \nabla_\theta \ell(\tilde{\mathbf{x}}_i^k, y_i^k; \theta_t) + \gamma \theta_t) \quad (5)$$

For the adversarial distribution $\mathbf{q}$, each concept has a weight q_k and the straightforward way is to sample a mini-batch of examples from each concept to estimate the gradient of the distribution. However, the number of concepts varies and it can be larger than the size of a mini-batch. Besides, it results in the different sampling strategies for computing the gradient of deep models and the adversarial distribution, which increases the complexity of the training system. To address the issue, we take the same random sampling pipeline and update the distribution with weighted gradient ascent as

$$\hat{q}_k^{t+1} = q_k^t + \eta_q^t\big(\frac{1}{m}\sum_j^{m_k}\frac{N}{N_k}\ell(\tilde{\mathbf{x}}_j^k, y_j^k; \theta_t) - \lambda(q_k^t - q_0^k)\big)$$
$$\mathbf{q}_{t+1} = \mathcal{P}_\Delta(\hat{\mathbf{q}}_{t+1}) \tag{6}$$

where m_k is the number of examples from the k-th concept in the mini-batch and $\sum_k m_k = m$. $\mathcal{P}_\Delta(\cdot)$ projects the vector onto the simplex as in [9].

The re-weighting strategy makes the gradient unbiased but introduces the additional variance. Since batch-normalization [13] is inapplicable for the parameters of the adversarial distribution that is from the simplex, we develop a learning strategy to reduce the variance from gradients.

First, to illustrate the issue, let δ_1 and δ_2 be two binary random variables as

$$\Pr\{\delta_1 = 1\} = \frac{1}{N_k}; \quad \Pr\{\delta_2 = 1\} = \frac{1}{N}$$

Obviously, we have $E[\delta_1] = \frac{1}{N_k}$; $E[\frac{N\delta_2}{N_k}] = \frac{1}{N_k}$. It demonstrates that the gradient after re-weighting is unbiased. However, the variance can be different as

$$\mathrm{Var}[\delta_1] = \frac{1}{N_k} - \frac{1}{N_k^2}; \quad \mathrm{Var}[\frac{N\delta_2}{N_k}] = \frac{N}{N_k^2} - \frac{1}{N_k^2}$$

where the variance is roughly increased by a factor of N/N_k.

By investigating the updating criterion in Eqn. 6, we find that the gradient is rescaled by the learning rate η_q^t. If we let $\eta_q^t = \mathcal{O}(\frac{1}{t})$, the norm of the gradient will be limited after a sufficient number of iterations. Besides, for any distribution $\mathbf{q}'$, the norm of $\|\mathbf{q} - \mathbf{q}'\|_2^2$ is bounded by a small value of 2 since the distribution is from the simplex. It inspires us to deal with the first several iterations by adopting a small learning rate. The algorithm is summarized in Alg. 1. In short, we use the learning rate as $\eta_t = \frac{1}{c\lambda t}$ where $c > 1$ for the first s iterations and then the conventional learning rate $\eta_t = \frac{1}{\lambda t}$ is applied.

The convergence about the adversarial distribution is stated as follows.

Theorem 3. *Assume the gradient of the adversarial distribution* $\mathbf{q}$ *is bounded as* $\forall t, \|g_q^t\|_2 \leq \mu$ *and set the learning rate as*

$$\eta_q^t = \begin{cases} \frac{1}{c\lambda t} & t \leq s \\ \frac{1}{\lambda t} & o.w. \end{cases}$$

We have

$$\max_{\mathbf{q}^* \in \Delta} \frac{1}{T}\sum_t^T E[\mathcal{L}(\mathbf{q}^*, \theta_t) - \mathcal{L}(\mathbf{q}_t, \theta_t)]$$
$$\leq \frac{1}{T}(\frac{\mu^2}{2\lambda}(\log(T) + 1) - \beta)$$

where β *is a non-negative constant as* $\beta = (\mu\sqrt{\frac{\log(s)}{2\lambda}} - \sqrt{s\lambda})^2$ *and* $c = \frac{\mu}{\lambda}\sqrt{\frac{\log(s)}{2s}}$ *should be larger than* 1.

Algorithm 1 Hierarchically Robust Representation Learning (HRRL)

Input: Dataset $\{\mathbf{x}_i, y_i\}$, iterations T, mini-batch size m, λ, γ, τ, s, c
for $t = 1, \cdots, T$ **do**
  if $t <= s$ **then**
    $\eta_q^t = \frac{1}{c\lambda t}$
  else
    $\eta_q^t = \frac{1}{\lambda t}$
  end if
  Sample a mini-batch of examples $\{\mathbf{x}_i, y_i\}_{i=1,\ldots,m}$
  Generate the augmented data as $\tilde{\mathbf{x}}_i = \mathbf{x}_i + \tau\mathbf{z}_i$
  Update model with gradient descent as in Eqn. 5
  Update distribution with gradient ascent as in Eqn. 6
end for
return A feature extraction function $f(\cdot)$ from θ_T

Theorem 3 shows a $\mathcal{O}(\log(T)/T)$ convergence rate for the adversarial distribution. The gain of the adaptive learning rate is indicated in β, that is, a larger β provides a better convergence. When applying the conventional learning rate i.e. $c = 1$, it is easy to show $\beta = 0$. To further investigate the properties of β, we let $h(s) = \mu\sqrt{\log(s)/(2\lambda)} - \sqrt{s\lambda}$, i.e., $\beta = h(s)^2$, and study its behavior.

Proposition 1. $h(s)$ *is non-negative.*

Proof. Since $c = \frac{\mu}{\lambda}\sqrt{\frac{\log(s)}{2s}} > 1$, we have $\mu > \lambda\sqrt{\frac{2s}{\log(s)}}$. Therefore

$$h(s) = \mu\sqrt{\frac{\log(s)}{2\lambda}} - \sqrt{s\lambda} > \sqrt{s\lambda} - \sqrt{s\lambda} = 0$$

□

It implies that we can benefit from the variance reduction as long as the variance μ is sufficiently large. Then, we fix $\lambda = 1$ and plot the curve of $h(s)$ when varying μ in Fig. 2. We can find that $h(s)$ achieves its maximum after

thousands of iterations, which suggests that s should not be too large. It is consistent with our claim that the gradient will be shrunk by the learning rate and the additional variance has little influence when t is large.

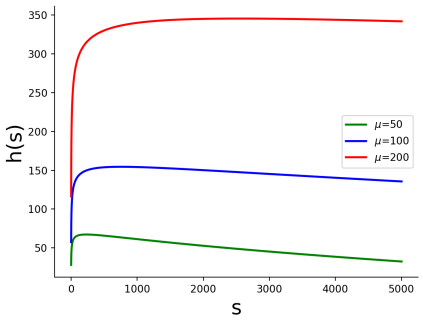


Figure 2. Curves of $h(s)$ with different μ's.

4. Experiments

We adopt ImageNet ILSVRC12 [22] as the benchmark data set to learn models for generic feature extraction in the experiments. ImageNet includes $1,000$ classes, where each class has about $1,200$ images in training and 50 images in test. We summarize these $1,000$ classes into 11 concepts according to the structure of WordNet [10] that is the default class structure of ImageNet. The statistics of the concepts and classes are summarized in Table 1. Apparently, ImageNet is biased to specific animals. For example, it contains 59 classes of birds and more than 100 classes of dogs. This bias can result in the performance degeneration when applying the model learned by ERM to generate representations for different tasks.

Concept	An	Ar	B	C	De	Do
#Classes	121	107	59	56	129	118
Concepts	I	M	S	V	O	
#Classes	106	100	57	67	80	

Table 1. Concepts in ImageNet. The initials "An", "Ar", "B", "C", "De", "Do", "I", "M", "S", "V", "O" denote "Animal", "Artifact", "Bird", "Container", "Device", "Dog", "Instrumentality", "Mammal", "Structure", "Vehicle", "Others", respectively.

We apply ResNet-18 [12], which is a popular network as the feature extractor [17], to learn the representations. We train the model with stochastic gradient descent (SGD) on 2 GPUs. Following the common practice [12], we learn the model with 90 epochs and set the size of mini-batch as 256. The initial learning rate is set to 0.1, and then it is decayed by a factor of 10 at $\{30, 60\}$. The weight decay is 10^{-4} and the momentum in SGD is 0.9. All model training includes random crop and horizontal flipping as the data augmentation. We set $s = 1000$ as suggested by Fig. 2 for the proposed algorithm. For the setting of c, we calculate the variance for μ from several mini-batches and set $c = 10$ according to Theorem 3. After obtaining deep models, we extract deep features from the layer before the last fully-connected layer, which generates a 512-dimensional feature for a single image. Given the features, we learn a linear SVM [4] to categorize examples. τ, λ and the parameter of SVM are searched in $\{10^i\}(i = -3, \ldots, 1)$. Four different deep features with SVM as follows are compared in the experiments, where SVM_{ERM} is the conventional way to extract features with models trained by ERM and the others are our proposals.

- SVM_{ERM}: deep features learned with ERM.
- SVM_{EL}: deep features learned with example-level robustness only.
- SVM_{CL}: deep features learned with concept-level robustness only.
- SVM_{HRRL}: deep features learned with both example-level and concept-level robustness.

Experiments are repeated 3 times and the average results with standard deviation are reported.

4.1. CIFAR-10

First, we study the scenario when example-level distribution difference exits between the target task and the benchmark data set. We conduct experiments on CIFAR-10 [14], which contains 10 classes and $60,000$ images. $50,000$ of them are for training and the rest are for test. CIFAR-10 has the similar concepts as those in ImageNet, e.g., "bird", "dog", and the difference in concepts is negligible. On the contrary, each image in CIFAR-10 has a size of 32×32, which is significantly smaller than that of images in ImageNet. As shown in Fig. 1, the example-level distribution changes dramatically and the example-level robustness is important for this task.

Table 2 summarizes the comparison. First, we observe that the accuracy of SVM_{ERM} can achieve 85.77%, which surpasses the performance of SIFT features [2], i.e., 65.6%, by more than 20%. It confirms that representations extracted from a DNN model trained on the benchmark data set can be applicable for generic tasks. Compared with representations from the model learned with ERM, SVM_{EL} outperforms it by a margin about 1%. It shows that optimizing with Wasserstein ambiguity set can learn the example-level robust features and handle the difference in examples better than ERM. SVM_{CL} has the similar performance as SVM_{ERM}. It is consistent with the fact that the difference of concepts between CIFAR-10 and ImageNet is small. Finally, the performance of SVM_{HRRL} is comparable to that of SVM_{EL} due to negligible concept-level distribution difference but it is significantly better than SVM_{ERM}, which demonstrates the effectiveness of the proposed algorithm.

Methods	Acc(mean±std)
SVM_{ERM}	85.77±0.12
SVM_{EL}	86.62±0.18
SVM_{CL}	85.64±0.26
SVM_{HRRL}	86.49±0.19

Table 2. Comparison of accuracy (%) on CIFAR-10.

4.2. Stanford Online Products (SOP)

In this subsection, we demonstrate the importance of concept-level robustness. We have Stanford Online Products (SOP) [25] as the target task to evaluate the learned representations. SOP collects product images from eBay.com and consists of $59,551$ images for training and $60,502$ images for test. We adopt the super class label for each image, which leads to a 12-class classification problem. As shown in Fig. 1, we can find that the example-level distribution difference is not significant (e.g., resolution), while the distribution of concepts (i.e., concept-level distribution) is relatively different. ImageNet includes many natural objects, e.g., animals, while SOP only contains artificial ones. Handling the difference in concepts is challenging for this task.

Table 3 shows the performance comparison. Apparently, SVM_{EL} has the similar performance as SVM_{ERM} due to the minor changes in the example-level distribution. However, SVM_{CL} demonstrates a better accuracy, which is about 1% better than SVM_{ERM}. It demonstrates that the deep features learned with the proposed algorithm is more robust than those from ERM when the distribution of concepts varies. Besides, the performances of SVM_{HRRL} and SVM_{CL} are comparable, which confirms that deep features obtained with hierarchical robustness work well consistently in different scenarios.

Methods	Acc(mean±std)
SVM_{ERM}	73.47±0.09
SVM_{EL}	73.48±0.08
SVM_{CL}	74.34±0.05
SVM_{HRRL}	74.23±0.08

Table 3. Comparison of accuracy (%) on SOP.

4.3. Street View House Numbers (SVHN)

Finally, we deal with a task when both example-level and concept-level distribution differences exist. We evaluate the robustness of deep features on Street View House Numbers (SVHN) [19] data set. It consists of $73,257$ images for training and $26,032$ for test. The target is to identify one of 10 digits from each 32×32 image. The image has the same size as CIFAR-10, which is very different from ImageNet. Moreover, SVHN has the concepts of digits, which is also different from ImageNet.

We compare the different deep features in Table 4. First, as observed in CIFAR-10, SVM_{EL} outperforms SVM_{ERM} by a large margin. It is because features learned with example-level robustness is more applicable than those from ERM when examples are from a different distribution. Second, SVM_{CL} improves the performance by more than 2%. It is consistent with the observation in SOP, where features learned with concept-level robustness perform better when concepts vary. Besides, we can observe that the performance of SVM_{CL} surpasses that of SVM_{EL}. It implies that controlling concept-level robustness, which has not been investigated sufficiently, may be more important than example-level robustness for representation learning. Finally, by combining example-level and concept-level robustness, SVM_{HRRL} shows an improvement of more than 4%. It demonstrates that example-level and concept-level robustness are complementary. Incorporating both of them can further improve the performance of deep features, when the example-level and concept-level distributions are different from these of the benchmark data set.

Methods	Acc(mean±std)
SVM_{ERM}	63.23±0.35
SVM_{EL}	65.01±0.37
SVM_{CL}	65.47±0.27
SVM_{HRRL}	67.33±0.39

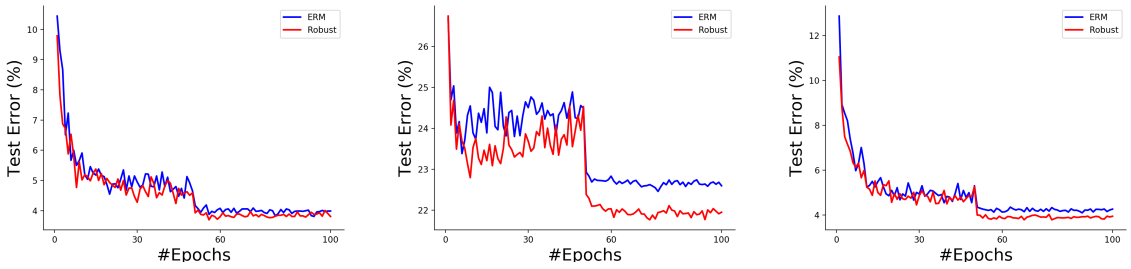
Table 4. Comparison of accuracy (%) on SVHN.

4.4. Fine-tuning

Besides extracting features, a pre-trained model is often applied as an initialization for training DNNs on the target task when GPUs are available. Since initialization is crucial for the final performance of DNNs [26], we conduct the experiments that initialize the model with parameters trained on ImageNet and then fine-tune the model on CIFAR-10, SOP and SVHN. After initialization, the model is fine-tuned with 100 epochs, where ERM is adopted as the objective for each task. The learning rate is set as 0.01 and decayed once by a factor of 10 after 50 epochs. Fig. 3 illustrates the curve of test error. We let "ERM" denote the model initialized with that pre-trained with ERM and "Robust" denote the one initialized with the model pre-trained with the proposed algorithm. Surprisingly, we observe that the models initialized with the proposed algorithm still surpass those with ERM. It implies that the learned robust models can be used for initialization besides feature extraction.

4.5. Effect of Robustness

Finally, we investigate the effect of the proposed method on ImageNet task itself to further illustrate the impact of robustness. First, we demonstrate the results of example-level robustness. We generate the augmented examples for vali-



(a) CIFAR-10 (b) SOP (c) SVHN

Figure 3. Comparison of fine-tuning with different initializations.

dation set as in Theorem 2 and report the accuracy of different models in Fig. 4. The horizontal axis shows the step size for generating the augmented examples. When step size is 0, the original validation set is used for evaluation. Otherwise, each image in the validation set is modified with the corresponding step size, and only modified images are used for evaluation. Intuitively, larger step size implies larger example-level distribution change compared to the original ImageNet data set.

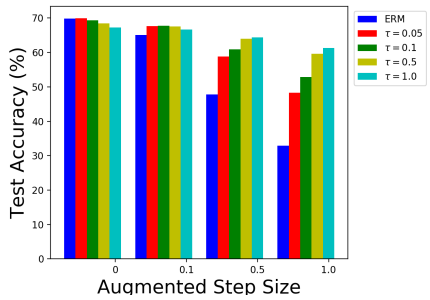


Figure 4. Comparison of accuracy on augmented examples.

Besides ERM, four different models are included in the comparison. Each model is trained with the example-level robustness and the corresponding parameter τ is denoted in the legend, where larger τ should theoretically provide a more robust model.

We can observe that ERM performs well when there is no augmentation but its performance degrades significantly when the augmentation step size increases. It confirms that ERM cannot generalize well when the example-level distribution changes. Fortunately, we can observe that more robust models (i.e., τ increases) can provide better generalization performance as expected. It is because that the proposed algorithm focuses on optimizing the worst-case performance among different distributions derived from the original distribution.

Second, we show the influence of concept-level robustness. We train models with different λ for regularization and summarize the accuracy of concepts in Fig. 5. We sort the accuracy in ascending order to make the comparison clear. As illustrated, ERM aims to optimize the uniform distribution of examples and ignores the distribution of concepts. Consequently, certain concept, e.g., "bird", has much higher accuracy than others. When decreasing λ in our proposed method, the freedom of adversarial distribution increases. With more freedom, the proposed method will focus on the concepts with bad performance. By optimizing the adversarial distribution, the model will balance the performance between different concepts as illustrated in Fig. 5.

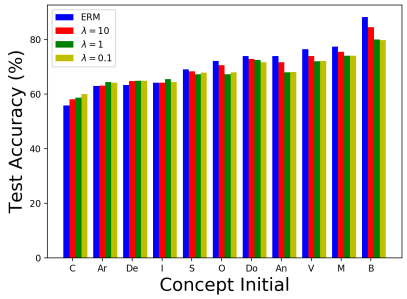


Figure 5. Comparison of accuracy on concepts in ImageNet.

In summary, Figs. 4 and 5 demonstrate the different influences of example-level and concept-level robustness. Evidently, our method can deal with the perturbation from different aspects. It further confirms that improving the hierarchical robustness is important for applying deep features or initializing models in real-world applications.

5. Conclusion

In this work, we study the problem of learning deep features using a benchmark data set for generic tasks. We propose a hierarchically robust optimization algorithm to learn robust representations from a large-scale benchmark data set. The theoretical analysis also demonstrates the importance of augmentation when training DNNs. The experiments on real-world data sets demonstrate the effectiveness of the learned features from the proposed method. The framework can be further improved when side information is available. For example, given the concepts of the target domain, we can obtain the specific reference distribution $\mathbf{q}_0$ accordingly, and then learn the features for the desired task. This direction can be our future work.

References

[1] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton University Press, 2009.

[2] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Kernel descriptors for visual recognition. In *NeurIPS*, pages 244–252, 2010.

[3] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[4] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[5] Robert S. Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. In *NeurIPS*, pages 4708–4717, 2017.

[6] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[8] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014.

[9] John C. Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l_1-ball for learning in high dimensions. In *ICML*, pages 272–279, 2008.

[10] Christine Fellbaum. 1998, wordnet: An electronic lexical database, 1998.

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[14] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1106–1114, 2012.

[16] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[17] Romain Mormont, Pierre Geurts, and Raphaël Marée. Comparison of deep transfer learning strategies for digital pathology. In *CVPR Workshops*, pages 2262–2271, 2018.

[18] Hongseok Namkoong and John C. Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *NeurIPS*, pages 2208–2216, 2016.

[19] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[20] Qi Qian, Rong Jin, Shenghuo Zhu, and Yuanqing Lin. Fine-grained visual categorization via multi-stage metric learning. In *CVPR*, pages 3716–3724, 2015.

[21] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[24] Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. In *ICLR*, 2018.

[25] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016.

[26] Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, pages 1139–1147, 2013.

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[28] Bolei Zhou, Àgata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NeurIPS*, pages 487–495, 2014.