Geometry-aware Deep Transform

Jiaji Huang Qiang Qiu Robert Calderbank Guillermo Sapiro
Duke University
Durham, NC, 27708
{jiaji.huang, qiang.qiu, guillermo.sapiro, robert.calderbank}@duke.edu

Abstract

Many recent efforts have been devoted to designing sophisticated deep learning structures, obtaining revolutionary results on benchmark datasets. The success of these deep learning methods mostly relies on an enormous volume of labeled training samples to learn a huge number of parameters in a network; therefore, understanding the generalization ability of a learned deep network cannot be overlooked, especially when restricted to a small training set, which is the case for many applications. In this paper, we propose a novel deep learning objective formulation that unifies both the classification and metric learning criteria. We then introduce a geometry-aware deep transform to enable a non-linear discriminative and robust feature transform, which shows competitive performance on small training sets for both synthetic and real-world data. We further support the proposed framework with a formal (K, ϵ)*-robustness analysis.*

1. Introduction

Many recent efforts have been devoted to learning a mapping from low-level image features, e.g., image patches [16, 17], LBP descriptors [3, 14], to high-level discriminative representations. The learned feature mapping often increases the inter-class separation while reducing the intra-class variation. This idea dates back at least to the linear discriminant analysis (LDA) for linear cases; however, if we allow the feature mapping to be non-linear, e.g., deep convolutional neural network [8, 16, 15], the discriminability of the learned representation is often significantly enhanced compared to its linear counterpart.

Deep learning techniques achieve unprecedentedly high precision in object and scene classification, where an enormous volume of labeled training samples are often required to learn a rich set of parameters [6, 10, 15]. Despite such revolutionary advances, many real-world classification problems remain challenging, due to the large number of non-linearly separable classes and the scarcity of training samples. One such example is face verification [9], where recently reported successes mostly rely on huge proprietary training sets, e.g., 4.4 million labeled faces from 4,030 people in [17]; however, publicly available training datasets often consist of only a small set of subjects with several samples per subject. It is a notoriously difficult task to learn from limited training samples a deep structure that can generalize well on testing data [11].

While great current attentions are paid to smart manipulation of different deep architectures for more discriminative representations [7, 16, 17], in this paper, we focus on the generalization problem, i.e., how to encourage a mapping learned from limited training samples to generalize well over testing data. This issue is of significant importance when the training samples are scarce, in which case the network optimized on the training set is likely at the risk of overfitting. We provide both analytic and experimental illustrations on the generalization errors of a learned deep structure, under several popular objective functions.

We further propose a geometry-aware feature transformation framework, which balances between discriminability and generalization. The proposed framework encourages inter-class separation while at the same time penalizes the distortion of intra-class structure. This also extends the "shallow" setup in [12] to a deep architecture, also providing theoretical insights regarding robustness. In particular, we show that constraining feature mapping functions to be near-isometry in local sub-regions yields robust algorithms. We first motivate our framework with a synthetic example, and then support it through theoretical analysis. We further validate our framework using face verification experiments and report state-of-the-art results on the challenging LFW face dataset .

Our main contributions are:

- proposing a novel deep learning objective that unifies the classification and metric learning criteria.
- providing a theoretical argument showing that awareness of geometry leads to robustness;

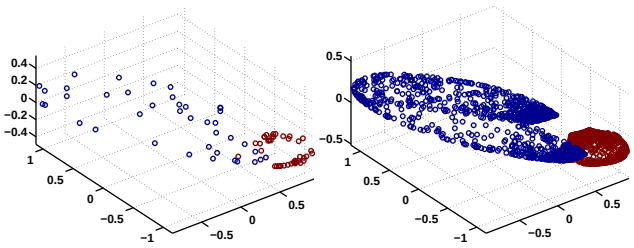
- motivating a general algorithmic framework which considers data geometry in the formulation;
- designing a learned deep transform, as a particular example of the proposed geometric framework, that achieves state-of-art results.

2. Geometry-aware deep transform

Deep networks are often optimized for a classification objective, where class-labeled samples are input as training [6, 10, 16, 17]; or a metric learning objective, where training data are input as positive and negative pairs [8, 15]. [1] In this section, we first propose a novel deep learning objective that unifies the classification and metric learning criteria. We then introduce a geometry-aware deep transform, and optimize it through standard back-propagation. We further support the proposed framework with a formal (K, ϵ)-robustness analysis [18].

2.1. Pedagogic formulation

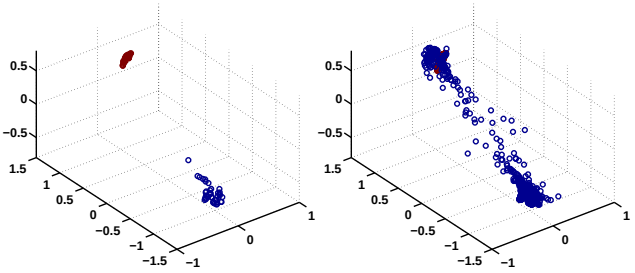
We use the following two-class problem as an illustration example: The first class is generated as $\mathbf{x} = \mathbf{U}\mathbf{v}/\|\mathbf{U}\mathbf{v}\|$, where $\mathbf{v}$ is with probability (w.p.) $1/2$ from a constrained plane $-y + z = 1, x \in [-1, 1], z \in [-3, 0]$, and w.p. $1/2$ from plane $y + z = 1, x \in [-1, 1], z \in [0, 3]$. $\mathbf{U}$ is a $d \times 3$ ($d \gg 3$, $d = 100$ in this case) matrix that embeds $\mathbf{x}$ into a d-dimensional space. Similarly, the second class is generated as $\mathbf{x} = \mathbf{U}\mathbf{u}/\|\mathbf{U}\mathbf{u}\|$, where $\mathbf{u}$ is w.p. $1/2$ from $-y + z = -1, x \in [-1, 1], z \in [-3, 0]$, and w.p. $1/2$ from $y + z = -1, x \in [-1, 1], z \in [0, 3]$. For each class, 40 training and 1000 testing samples are generated. Fig. 1 visualizes the training and testing data by randomly projecting it to a 3 dimensional coordinate system, with different colors representing different classes. Observe that the two classes are not linearly separable, which necessitates a non-linear feature transform.



(a) Training samples: 40 per class. (b) Testing samples: 1000 per class.

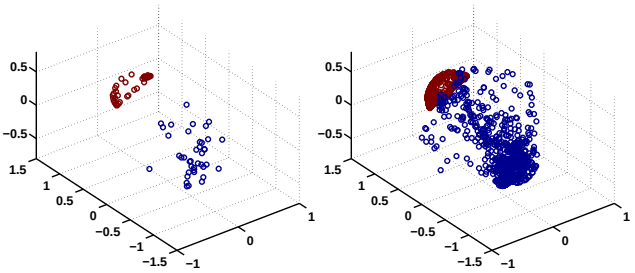
Figure 1: Training and testing samples.

[1] A positive pair contains two samples from the same class, and a negative pair contain two samples from different classes.



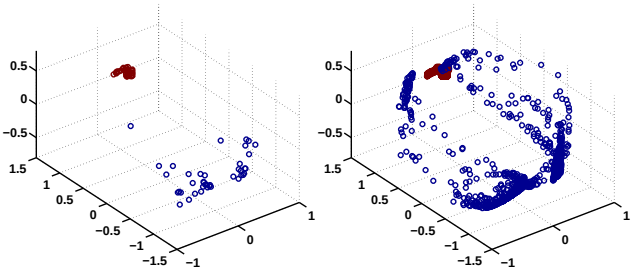
(a) Transformed training samples. (b) Transformed testing samples.

Figure 2: Transformed features using a metric learning formulation.



(a) Transformed training samples. (b) Transformed testing samples.

Figure 3: Transformed features using a classification formulation.



(a) transformed training samples (b) transformed testing samples

Figure 4: Transformed features using GDT with $\lambda = 0.4$.

We want to learn a mapping $f(\mathbf{x})$ that transforms the low-level feature $\mathbf{x}$ to a more discriminative one. In this paper, we are particularly interested in non-linear transforms $f(\cdot)$ implemented as a deep neural network. However, the method and theory we develop are general in the sense that any other family of $f(\cdot)$ can be adopted as well. In this example, $f(\cdot)$ is implemented as a 2-layer fully connected neural network with $\tanh$ as the squash function, $f(\cdot)$ taking the form

$$f(\mathbf{x}) = \tanh(\mathbf{A}_2 \tanh(\mathbf{A}_1 \mathbf{x})), \tag{1}$$

where $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{d\times d}$ are the linear coefficients in those two layers.

Metric learning formulation. The general goal of metric learning is to ensure that, after the transform, the distance between intra-class points is small, while the inter-class distance is large. The Euclidean distance is a common choice of metric; however, empirical results [3, 14] have shown that the cosine distance outperforms Euclidean distance on certain tasks such as face verification. Moreover, cosine distance is bounded and easier for us to design the loss function. We therefore adopt the cosine distance in this paper, and propose the metric learning formulation

$$\min_{\mathbf{A}_1,\mathbf{A}_2} \sum_{i\neq j} \left(\frac{f(\mathbf{x}_i)^\top f(\mathbf{x}_j)}{\|f(\mathbf{x}_i)\| \cdot \|f(\mathbf{x}_j)\|} - t_{i,j} \right)^2, \qquad (2)$$

where the indicator

$$t_{i,j} = \begin{cases} 1 & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \text{ same class,} \\ -1 & \text{otherwise.} \end{cases}$$

Notice that $\frac{f(\mathbf{x}_i)^\top f(\mathbf{x}_j)}{\|f(\mathbf{x}_i)\|\cdot\|f(\mathbf{x}_j)\|} \in [-1, 1]$ is the cosine of the angle between the transformed features $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$. The objective of (2) is to encourage the intra-class angle to be close to 0, and the inter-class ones to be as separated as π. We use back-propagation to optimize the parameters, $\mathbf{A}_1$ and $\mathbf{A}_2$, as explained later.

Fig. 2a visualizes the transformed training samples by the learned $f(\cdot)$. The learned transform significantly pulls apart the two classes and reduces the variations within each individual class. We then apply the learned $f(\cdot)$ to the testing samples (fig. 2b). However, we observe that the two classes are not well separated, raising our concerns about the robustness of the pure metric learning formulation in (2).

Classification formulation. Now let us consider a different objective function, where we encourage the intra-class angles to be preserved after the transformation. This new objective has a unified formulation as (2), but now the indicator becomes

$$t_{i,j} = \begin{cases} \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|} & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \text{ same class,} \\ -1 & \text{otherwise.} \end{cases}$$

We denote this objective function as a classification formulation, as it shares similar attributes to the classification objective commonly optimized for a deep network [16, 17]. Explicit constraints are imposed to separate different classes, e.g., $t_{i,j} = -1$ for negative pairs here, but only weak constraints are used to assign similar representation to the same class. This classification formulation is less ambitious than the metric learning formulation, as it does not require the variance in the same class being reduced. $f(\cdot)$ is implemented as the 2-layer neural network as described before, and optimized through back-propagation.

The transformed training and testing samples are visualized in Fig. 3. Comparing Fig. 2 and Fig. 3, we observe that although our metric learning formulation works well on the training data, it does not well discriminate the two classes on testing data, *i.e.*, it has a big generalization error. In contrast, following the classification formulation, the intra-class variance is not reduced, yet the deterioration from training to testing is not so significant. In other words, while the metric learning formulation is too optimistic about the discrimination we can achieve, the classification formulation is more robust but conservative.

2.2. Proposed formulation and algorithm

We introduce now a geometry-aware deep transform. We use $f_{\boldsymbol{\alpha}}(\cdot)$ to denote the feature transform, to emphasize that $\boldsymbol{\alpha}$ are parameters to be learned, e.g., filters in a neural network ($\mathbf{A}_1, \mathbf{A}_2$ in the previous section). $f_{\boldsymbol{\alpha}}$ can be a linear function or a non-linear function implemented by a neural network.

We formulate the transformation learning problem as:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{i\neq j} \left(\frac{f_{\boldsymbol{\alpha}}(\mathbf{x}_i)^\top f_{\boldsymbol{\alpha}}(\mathbf{x}_j)}{\|f_{\boldsymbol{\alpha}}(\mathbf{x}_i)\| \cdot \|f_{\boldsymbol{\alpha}}(\mathbf{x}_j)\|} - t_{i,j} \right)^2, \qquad (3)$$

where the indicator

$$t_{i,j} = \begin{cases} \lambda + (1-\lambda)\frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|} & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \text{ same class,} \\ -1 & \text{otherwise.} \end{cases}$$

and $\lambda \in [0, 1]$. We denote formulation (3) as Geometry-aware Deep Transform (GDT). The GDT objective is a weighted combination of the two pedagogic formulations discussed above. We can understand it as regularizing the metric learning formulation using the classification one.

We use gradient descent (back-propagation if $f_{\boldsymbol{\alpha}}(\cdot)$ is a deep neural network) to solve for the $\boldsymbol{\alpha}$ in (3). In particular, let us denote the objective in (3) as J and define

$$\begin{aligned} f_{\boldsymbol{\alpha}}(\mathbf{x}_i) &\triangleq \quad \mathbf{y}_i, \\ \frac{f_{\boldsymbol{\alpha}}^\top(\mathbf{x}_i) f_{\boldsymbol{\alpha}}(\mathbf{x}_i)}{\|f_{\boldsymbol{\alpha}}(\mathbf{x}_i)\| \cdot \|f_{\boldsymbol{\alpha}}(\mathbf{x}_j)\|} &\triangleq \quad C_{i,j}. \end{aligned} \qquad (4)$$

Then we have

$$\frac{\partial J}{\partial \mathbf{y}_i} = \frac{1}{\|\mathbf{y}_i\|} \sum_{j\neq i} (C_{i,j} - t_{i,j}) \left[\frac{\mathbf{y}_j}{\|\mathbf{y}_j\|} - C_{i,j} \cdot \frac{\mathbf{y}_i}{\|\mathbf{y}_i\|} \right]. \qquad (5)$$

$\frac{\partial J}{\partial \mathbf{y}_j}$ can be calculated in the same manner. Then we back-propagate this gradient through the network to update all the parameters. More specifically, we denote $\boldsymbol{\alpha}^{(k)}$ as the filter weights and bias in the k-th ($1 \leq k \leq K$) layer. And $\mathbf{x}_i^{(k)}$

as the output of the k-th layer excited by the input $\mathbf{x}_i^{(k-1)}$ (therefore $\mathbf{y}_i = \mathbf{x}_i^{(K)}$ and $\mathbf{x}_i = \mathbf{x}_i^{(0)}$). Then,

$$\frac{\partial J}{\partial \boldsymbol{\alpha}^{(K)}} = \sum_i \frac{\partial J}{\partial \mathbf{y}_i} \cdot \frac{\partial \mathbf{y}_i}{\partial \boldsymbol{\alpha}^{(K)}},$$
$$\frac{\partial J}{\partial \boldsymbol{\alpha}^{(k)}} = \sum_i \frac{\partial J}{\partial \mathbf{x}_i^{(k+1)}} \cdot \frac{\partial \mathbf{x}_i^{(k+1)}}{\partial \mathbf{x}_i^{(k)}} \cdot \frac{\partial \mathbf{x}_i^{(k)}}{\partial \boldsymbol{\alpha}^{(k)}}, 1 \leq k \leq K-1. \tag{6}$$

An overview of the GDT algorithm is summarized in Algorithm 1.

Algorithm 1 Gradient descent solver for GDT

Input: $\lambda \in [0,1]$, training pairs $\{(\mathbf{x}_i, \mathbf{x}_j, \ell_{i,j})\}$, a defined K-layer network ($f_{\boldsymbol{\alpha}}(\cdot)$ family), stepsize γ
Output: $\boldsymbol{\alpha}$
  while stable objective not achieved **do**
    Compute $\mathbf{y}_i = f_{\boldsymbol{\alpha}}(\mathbf{x}_i)$ by a forward pass
    Compute objective J
    Compute $\frac{\partial J}{\partial \mathbf{y}_i}$ as Eq. (5)
    for $k = K$ down to 1 **do**
      Compute $\frac{\partial J}{\partial \boldsymbol{\alpha}^{(k)}}$ as Eq. (6)
      $\boldsymbol{\alpha}^{(k)} \leftarrow \boldsymbol{\alpha}^{(k)} - \gamma \frac{\partial J}{\partial \boldsymbol{\alpha}^{(k)}}$
    end for
  end while

For an illustration of Algorithm 1, we apply it with $\lambda = 0.4$ to the illustrative example above. The transformed training and testing samples are shown as Fig. 4. Compared with the two pedagogic formulations (equivalent to $\lambda = 1$ and 0 in GDT respectively), this $\lambda = 0.4$ case is balancing between discriminability and robustness. Before more detailed experimental analysis are shown in Section 3, we now provide theoretical insights to support our robustness claim.

2.3. (K, ϵ)-robustness

GDT regularizes discriminative transform learning with intra-class structure preservation. In this section, we formally show that a local isometry regularization induces robustness. In the following, we assume a general objective that works with distance metrics of pairs of transformed features.

Let the low-level feature space be $\mathcal{X}$, and the class label set be $\mathcal{Y} = \{1, \ldots, L\}$, where L is the number of classes. $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ is the set of low-level features and their corresponding labels. The training set is

$$\mathcal{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \triangleq \{\mathbf{z}_1, \ldots, \mathbf{z}_n\} \in \mathcal{Z}^n,$$

which consists of n i.i.d. samples drawn from an unknown distribution $\mathcal{D}$ defined on $\mathcal{Z}$. The feature mapping is $f_{\boldsymbol{\alpha}}(\mathbf{x}) : \mathcal{X} \mapsto \mathcal{F}$, where $\mathcal{F}$ is the transformed feature space.

Denote ρ as a metric endowed with $\mathcal{X}$ and $\mathcal{F}$. Define pair label $\ell_{i,j} = 1$ if $y_i = y_j$, and -1 otherwise. We may adopt a loss function $g(\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_i), f_{\boldsymbol{\alpha}}(\mathbf{x}_j)), \ell_{i,j})$ that encourages $\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_i), f_{\boldsymbol{\alpha}}(\mathbf{x}_j))$ to be small (big) if $\ell_{i,j} = 1$ (-1). We require the Lipschtiz constant of $g(\cdot, 1)$ and $g(\cdot, -1)$ to be upper bounded by A $(0 < A < \infty)$. Examples of such g include the hinge loss

$$\max(-\ell_{i,j}(1 - \rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_i), f_{\boldsymbol{\alpha}}(\mathbf{x}_j))), 0), \tag{7}$$

as well as its smoothed version

$$\log(1 + e^{-\ell_{i,j}(1-\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_i), f_{\boldsymbol{\alpha}}(\mathbf{x}_j)))}), \tag{8}$$

both of which are commonly adopted in the metric learning literature [8]. In GDT formulation (3), the quadratic loss has bounded Lipschtiz w.r.t. the cosine distance $C_{i,j}$ as well. In the following, we denote

$$g(\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_i), f_{\boldsymbol{\alpha}}(\mathbf{x}_j)), \ell_{i,j}) \triangleq h_{\boldsymbol{\alpha}}(\mathbf{z}_i, \mathbf{z}_j)$$

for short.

The empirical loss on the training set (associated with parameter $\boldsymbol{\alpha}$) is

$$R_{emp}(\boldsymbol{\alpha}) \triangleq \frac{2}{n(n-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} h_{\boldsymbol{\alpha}}(\mathbf{z}_i, \mathbf{z}_j). \tag{9}$$

And the expected loss is

$$R(\boldsymbol{\alpha}) \triangleq \mathbb{E}_{\mathbf{z}_1', \mathbf{z}_2' \sim \mathcal{D}} \left[h_{\boldsymbol{\alpha}}(\mathbf{z}_1', \mathbf{z}_2') \right]. \tag{10}$$

The algorithm is a program that seeks

$$\boldsymbol{\alpha}_{\mathcal{T}} \triangleq \arg\min_{\boldsymbol{\alpha}} R_{emp}(\boldsymbol{\alpha}), \tag{11}$$

which minimizes the empirical loss on the training set $\mathcal{T}$. A metric learning type formulation, including our GDT, falls in the category of algorithm (11). The quantity $R_{emp}(\boldsymbol{\alpha}_{\mathcal{T}}) - R(\boldsymbol{\alpha}_{\mathcal{T}})$ is called the algorithm's generalization error. Smaller generalization error implies robustness.

The work [18] proposes a notion called (K, ϵ)-robustness, and [2] extends the definition of robustness to algorithms like (11) that work on pairs of samples. It also shows that (K, ϵ)-robust algorithms have generalization error bounded as

$$R_{emp}(\boldsymbol{\alpha}_{\mathcal{T}}) - R(\boldsymbol{\alpha}_{\mathcal{T}}) \leq \epsilon + O\left(\sqrt{\frac{K}{n}}\right). \tag{12}$$

We now rephrase the definition of (K, ϵ)-robustness in [2]:

Definition 1. *The algorithm* (11) *is* (K, ϵ)*-robust if* $\mathcal{Z}$ *can be partitioned into* K *disjoint set,* $\{C_k\}_{k=1}^K$*, such that for all* $\mathcal{T} \in \mathcal{Z}^n$*, the learned* $\boldsymbol{\alpha}_{\mathcal{T}}$ *satisfies:*
$\forall \mathbf{z}_i, \mathbf{z}_j \in \mathcal{T}$ *where* $i \neq j$*,*
$\forall \mathbf{z}_1', \mathbf{z}_2' \in \mathcal{Z}$*,*
If $\mathbf{z}_i, \mathbf{z}_1' \in C_p$*, and* $\mathbf{z}_j, \mathbf{z}_2' \in C_q$ *for any* $p, q \in \{1, \ldots, K\}$*, then*

$$|h_{\boldsymbol{\alpha}_{\mathcal{T}}}(\mathbf{z}_i, \mathbf{z}_j) - h_{\boldsymbol{\alpha}_{\mathcal{T}}}(\mathbf{z}_1', \mathbf{z}_2')| \leq \epsilon.$$

According to Definition 1, the (K, ϵ)-robustness essentially requires that with the learned $\boldsymbol{\alpha}_{\mathcal{T}}$, a testing pair $(\mathbf{z}'_1, \mathbf{z}'_2)$ incurs a similar loss with any training pair $(\mathbf{z}_i, \mathbf{z}_j)$ that is in the same subset (in a pair-wise sense). And according to the generalization error bound (12), the smaller ϵ is, the smaller the generalization error tends to be; therefore the more robust the algorithm is.

Before presenting our theory, we need to introduce the covering number, defined as follows:

Definition 2. *For a metric space $(\mathcal{S}, \rho)$, we say that $\hat{\mathcal{S}} \subset \mathcal{S}$ is a γ-cover of $\mathcal{S}$, if $\forall \mathbf{s} \in \mathcal{S}$, $\exists \hat{\mathbf{s}} \in \hat{\mathcal{S}}$ such that $\rho(\mathbf{s}, \hat{\mathbf{s}}) \leq \gamma$. The γ-covering number of $\mathcal{S}$ is*

$$\mathcal{N}_\gamma(\mathcal{S}, \rho) = \min\{|\hat{\mathcal{S}}| : \hat{\mathcal{S}} \text{ is a } \gamma\text{-cover of } \mathcal{S}\}$$

Remark 1. *The covering number describes how many balls (in ρ metric sense) we need to "cover" a space. Feature space of certain property, e.g., Gaussian distributed, sparsely representable [13], has certain covering number. The more complex the feature space is, the more balls we need to cover it. In a word, covering number reflects the geometry of the set $\mathcal{S}$. In particular, we notice that the set $\mathcal{S}$ with covering number $\mathcal{N}_{\gamma/2}(\mathcal{S}, \rho)$ can be partitioned into $\mathcal{N}_{\gamma/2}(\mathcal{S}, \rho)$ disjoint subsets, such that any two points within the same subset are separated by no more than γ.*

Lemma 1. *$\mathcal{Z}$ can be partitioned into $L\mathcal{N}_{\gamma/2}(\mathcal{X}, \rho)$ subsets, denoted as $\mathcal{Z}_1, \ldots, \mathcal{Z}_{L\mathcal{N}_{\gamma/2}(\mathcal{X},\rho)}$, such that for all $\mathbf{z}_1 \triangleq (\mathbf{x}_1, y_1), \mathbf{z}_2 \triangleq (\mathbf{x}_2, y_2)$ belonging to any one of these subsets, $y_1 = y_2$ and $\rho(\mathbf{x}_1, \mathbf{x}_2) \leq \gamma$.*

Proof. As noticed immediately after Definition 1, we can partition $\mathcal{X}$ into $\mathcal{N}_{\gamma/2}(\mathcal{X}, \rho)$ disjoint subsets, each with diameter no bigger than γ. Then we can partition $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ into $L\mathcal{N}_{\gamma/2}(\mathcal{X}, \rho)$ disjoint subsets, such that any two samples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2)$ in any one of these subsets have $y_1 = y_2$ and $\rho(\mathbf{x}_1, \mathbf{x}_2) \leq \gamma$. □

Lemma 1 also implies a partition of $\mathcal{X}$, denoted as $\mathcal{X}_1, \ldots, \mathcal{X}_{L\mathcal{N}_{\gamma/2}(\mathcal{X},\rho)}$ such that any $\mathbf{x}_i, \mathbf{x}_j$ from the same subset have $\rho(\mathbf{x}_i, \mathbf{x}_j) \leq \gamma$ and share the same label.

Theorem 1. *If $f_{\boldsymbol{\alpha}}(\mathbf{x})$ is a δ-isometry (i.e., distance distorted by at most δ after the transform) within each of $\mathcal{X}_1, \ldots, \mathcal{X}_{L\mathcal{N}_{\gamma/2}(\mathcal{X},\rho)}$ as described above, then an algorithm in the category of* (11) *is $(L\mathcal{N}_{\gamma/2}(\mathcal{X}, \rho), 2A(\gamma+\delta))$-robust.*

Proof. The proof follows the definition of (K, ϵ)-robustness. We pick any training samples $\mathbf{z}_i, \mathbf{z}_j$ and testing samples $\mathbf{z}'_1, \mathbf{z}'_2$ such that $\mathbf{z}_i, \mathbf{z}'_1 \in \mathcal{Z}_p$ and $\mathbf{z}_j, \mathbf{z}'_2 \in \mathcal{Z}_q$ for some $p, q \in \{1, \ldots, L\mathcal{N}_{\gamma/2}(\mathcal{X}, \rho)\}$. Then

$$\rho(\mathbf{x}_i, \mathbf{x}'_1) \leq \gamma \text{ and } \rho(\mathbf{x}_j, \mathbf{x}'_2) \leq \gamma.$$

Notice that $\mathbf{x}_i, \mathbf{x}'_1 \in \mathcal{X}_p$ and $\mathbf{x}_j, \mathbf{x}'_2 \in \mathcal{X}_q$. Therefore by the δ-isometry definition,

$$|\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_i), f_{\boldsymbol{\alpha}}(\mathbf{x}'_1)) - \rho(\mathbf{x}_i, \mathbf{x}'_1)| \leq \delta,$$

and

$$|\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_j), f_{\boldsymbol{\alpha}}(\mathbf{x}'_2)) - \rho(\mathbf{x}_j, \mathbf{x}'_2)| \leq \delta.$$

Rearranging the above gives

$$\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_i), f_{\boldsymbol{\alpha}}(\mathbf{x}'_1)) \leq \rho(\mathbf{x}_i, \mathbf{x}'_1) + \delta \leq \gamma + \delta,$$

and

$$\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_j), f_{\boldsymbol{\alpha}}(\mathbf{x}'_2)) \leq \rho(\mathbf{x}_j, \mathbf{x}'_2) + \delta \leq \gamma + \delta.$$

We need to bound the difference between $\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_i), f_{\boldsymbol{\alpha}}(\mathbf{x}_j))$ and $\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}'_1), f_{\boldsymbol{\alpha}}(\mathbf{x}'_2))$ so that we can further invoke the finite Lipschtiz assumption to bound the quantity $|h_{\boldsymbol{\alpha}}(\mathbf{z}_i, \mathbf{z}_j) - h_{\boldsymbol{\alpha}}(\mathbf{z}'_1, \mathbf{z}'_2)|$. Specifically,

$$\begin{aligned}
&|\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_i), f_{\boldsymbol{\alpha}}(\mathbf{x}_j)) - \rho(f_{\boldsymbol{\alpha}}(\mathbf{x}'_1), f_{\boldsymbol{\alpha}}(\mathbf{x}'_2))| \\
&\leq |\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_i), f_{\boldsymbol{\alpha}}(\mathbf{x}_j)) - \rho(f_{\boldsymbol{\alpha}}(\mathbf{x}'_1), f_{\boldsymbol{\alpha}}(\mathbf{x}_j))| \\
&\quad + |\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}'_1), f_{\boldsymbol{\alpha}}(\mathbf{x}_j)) - \rho(f_{\boldsymbol{\alpha}}(\mathbf{x}'_1), f_{\boldsymbol{\alpha}}(\mathbf{x}'_2))| \\
&\leq \rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_i), f_{\boldsymbol{\alpha}}(\mathbf{x}'_1)) + \rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_j), f_{\boldsymbol{\alpha}}(\mathbf{x}'_2)) \\
&\leq 2(\gamma + \delta),
\end{aligned}$$

where the second line follow from the triangle inequality, while the third line follows the definition of metric. Notice that $y_i = y'_1$ and $y_j = y'_2$. Therefore $|h_{\boldsymbol{\alpha}}(\mathbf{z}_i, \mathbf{z}_j) - h_{\boldsymbol{\alpha}}(\mathbf{z}'_1, \mathbf{z}'_2)|$ is either

$$|g(\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_i), f_{\boldsymbol{\alpha}}(\mathbf{x}_j)), 1) - g(\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}'_1), f_{\boldsymbol{\alpha}}(\mathbf{x}'_2)), 1)|,$$

or

$$|g(\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_i), f_{\boldsymbol{\alpha}}(\mathbf{x}_j)), -1) - g(\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}'_1), f_{\boldsymbol{\alpha}}(\mathbf{x}'_2)), -1)|.$$

Since the Lipschtiz constants of $g(\cdot, 1)$ and $g(\cdot, -1)$ are no bigger than A, we have

$$\begin{aligned}
&|h_{\boldsymbol{\alpha}}(\mathbf{z}_i, \mathbf{z}_j) - h_{\boldsymbol{\alpha}}(\mathbf{z}'_1, \mathbf{z}'_2)| \\
&\leq A|\rho(f_{\boldsymbol{\alpha}}(\mathbf{x}_i), f_{\boldsymbol{\alpha}}(\mathbf{x}_j)) - \rho(f_{\boldsymbol{\alpha}}(\mathbf{x}'_1), f_{\boldsymbol{\alpha}}(\mathbf{x}'_2))| \\
&\leq 2A(\gamma + \delta),
\end{aligned}$$

which concludes the proof. □

Remark 2. *Theorem 1 tells us that the algorithm will be robust if we constrain the function $f_{\boldsymbol{\alpha}}(\cdot)$ to be near isometric in local regions. And the robustness depends on how much of an isometry $f_{\boldsymbol{\alpha}}(\cdot)$ is in the local regions. The local regions are jointly defined by the class labels and the covering number, which, as we described in remark 1, depicts the geometry of the low-level feature space. Given that the algorithm is $(K, 2(\gamma + \delta))$ robust, by Eq.* (12)*, we can bound the generalization error of algorithms that belongs to the category of* (11) *by*

$$R_{emp}(\boldsymbol{\alpha}_{\mathcal{T}}) - R(\boldsymbol{\alpha}_{\mathcal{T}}) \leq 2(\gamma + \delta) + O\left(\sqrt{\frac{K}{n}}\right).$$

Remark 3. *In practice, we may resort to a formulation like GDT to encourage the mapping $f_{\boldsymbol{\alpha}}(\cdot)$ to be near isometry in the local regions. We can understand GDT as with small δ, resulting in small generalization error. This explains why GDT is more robust (Fig. 2 to 4) than the metric learning formulation.*

Remark 4. *In fact, GDT only partitions the $\mathcal{X}$ space into L subsets, implicitly assuming a trivial covering number of 1. One could further partition within each classes, corresponding to a nontrivial covering number. However, this is at the cost of learning local neighborhoods within each class, which is beyond the scope of this paper.*

3. Experiments

We provided a formal analysis in Section 2.3 to support the proposed geometry-aware deep transform as a robust framework for optimizing a deep network. In this section, we further present an experimental evaluation of GDT demonstrating its power in producing discriminative and robust features for classification. We compare GDT with two state-of-the-art deep learning objectives: DeepFace (DF) [17] and Deep Metric Learning (DML) [8]. As discussed before, DeepFace shares attributes with our pedagogic classification formulation, and DML is close to our pedagogic metric learning formulation.

3.1. Illustrative example revisited

We provide here more experimental evaluation using the illustrative example in Section 2. First we look at how λ influences the performance. The number of training samples per class ranges from 40 to 100. And λ is varied in the $[0, 1]$ interval.

Denote the training set as $\mathcal{T}$ and the testing set as $\mathcal{V}$. In our case, the empirical loss on the training set is

$$R_{emp} = \frac{1}{Z_{\mathcal{T}}} \sum_{\substack{i \neq j \\ \mathbf{x}_i, \mathbf{x}_j \in \mathcal{T}}} \left(\frac{f_{\boldsymbol{\alpha}_{\mathcal{T}}}(\mathbf{x}_i)^\top f_{\boldsymbol{\alpha}_{\mathcal{T}}}(\mathbf{x}_j)}{\|f_{\boldsymbol{\alpha}_{\mathcal{T}}}(\mathbf{x}_i)\| \cdot \|f_{\boldsymbol{\alpha}_{\mathcal{T}}}(\mathbf{x}_j)\|} - \ell_{i,j} \right)^2, \tag{13}$$

where $Z_{\mathcal{T}}$ is the number of pairs constructed from the training set. Note that the loss is not the objective in GDT formulation (3) averaged over $Z_{\mathcal{T}}$; the objective of GDT incorporates an intra-class structure-preserving regularization, which should be excluded in evaluating the empirical loss. The expected loss is empirically evaluated over the testing set,

$$\hat{R} = \frac{1}{Z_{\mathcal{V}}} \sum_{\substack{i \neq j \\ \mathbf{x}_i, \mathbf{x}_j \in \mathcal{V}}} \left(\frac{f_{\boldsymbol{\alpha}_{\mathcal{T}}}(\mathbf{x}_i)^\top f_{\boldsymbol{\alpha}_{\mathcal{T}}}(\mathbf{x}_j)}{\|f_{\boldsymbol{\alpha}_{\mathcal{T}}}(\mathbf{x}_i)\| \cdot \|f_{\boldsymbol{\alpha}_{\mathcal{T}}}(\mathbf{x}_j)\|} - \ell_{i,j} \right)^2, \tag{14}$$

where $Z_{\mathcal{V}}$ is the number of pairs constructed from the testing set. Here we use the notation $\hat{R}$ to indicate that it is an empirical estimate.

Fig. 5a shows R_{emp} and $\hat{R}$ for a variety of λ and $|\mathcal{T}|$. Note that the smaller λ is, the more the structure-preserving regularization is emphasized. We observe that R_{emp} is constantly lower than $\hat{R}$, indicating that R_{emp} always tends to be optimistic. As $|\mathcal{T}|$ increases, $\hat{R}$ decreases and R_{emp} approaches $\hat{R}$. Note that when $|\mathcal{T}|$ is small and λ is big, R_{emp} significantly underestimates $\hat{R}$. Fig. 5b shows an empirical estimate of the generalization error, $R_{emp} - \hat{R}$. Fixing a particular $|\mathcal{T}|$, the generalization error decreases as λ approaches zero, implying more robustness.

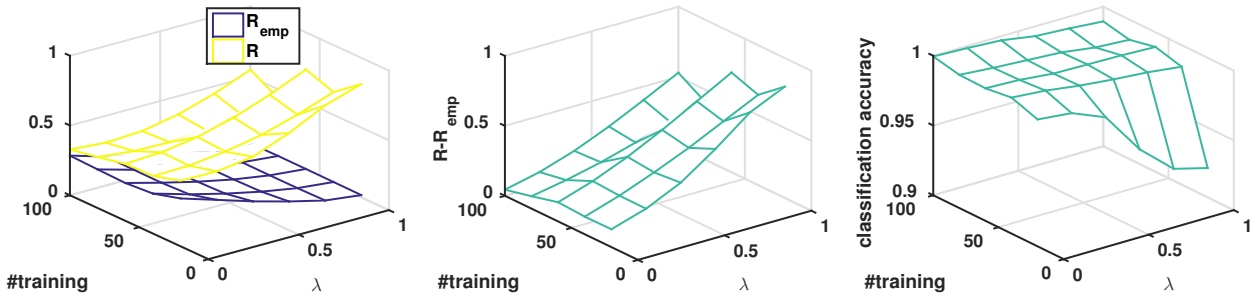
To see how the robustness influences classification, we apply a nearest neighbor (1-NN) classifier to the transformed testing data. The obtained classification accuracy is shown in Fig. 5c. When the number of training samples per class is small, there is a steady increase in classification accuracy as λ decreases, i.e., when more structure preservation is enforced; and such increase becomes less obvious when the training set size increases. The above observation clearly shows that, when only a small training set is given, the robustness gained from the structure preservation dominates the classification performance.

As discussed before, when $\lambda = 0$, the objective function is optimized for classification by imposing explicit constraints, $t_{i,j} = -1$ for negative pairs, to separate different classes; however, due to the structure preservation, weak constraints are used to enforce similar representation for the same class. This drawback cannot be overlooked for applications where it is critical to expect similar representations for the same class samples, such as face verification, and image retrieval. In the next section, we use face verification to demonstrate a scenario where the balance between robustness and discrimination is preferred.

3.2. Mnist

The last section shows an extreme case where the best classification performance is achieved when $\lambda = 0$. However, in general, R takes minimum at a nontrivial $\lambda \in (0, 1)$, as illustrated in this section. We apply GDT to Mnist dataset. The $f_{\boldsymbol{\alpha}}(\cdot)$ we adopted is a neural network made up of 3 convolutional layers. Between every two consecutive convolutional layer is a pooling layer. The original 28×28 images are mapped to 256 dimensional feature vectors.

We vary $\lambda \in [0, 1]$ and evaluate R_{emp}, R and generalization error for several training set sizes, as shown in Fig. 6. We observe that as λ varies from 0 to 1, the empirical loss keeps decreasing (Fig. 6a), implying increasing discrimination on training set. However, the generalization error keeps increasing (Fig. 6b), implying decreasing robustness. Therefore, to achieve smallest R (corresponding to best performance in testing set), we need to balance between dis-



(a) R_{emp} and $\hat{R}$ (b) Generalization error (c) 1-NN classification accuracy

Figure 5: Motivating example revisited.

crimination and robustness. And in general, the R takes minimum at some $\lambda \in (0, 1)$ (Fig. 6c).

As a comparison, we also run LeNet on the same training set. The LeNet's network structure is the same as the one adopted by GDT except that a fully connected layer and a softmax loss layer is added on the top. Fig. 6d compares the classification accuracy of 1-nn on GDT features and that of LeNet. GDT's accuracy constantly outperforms LeNet and peaks around $\lambda = 0.5$ where R is the smallest.

3.3. LFW

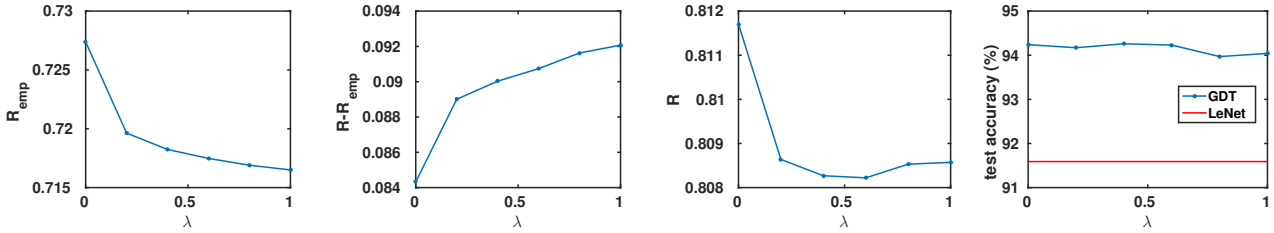
We further validate the effectiveness of the geometry-aware deep transform by performing face verification on the challenging LFW benchmark dataset [9]. Deep learning methods for face verification mostly use proprietary training data [15, 16, 17] and are therefore not reproducible. We adopt the experimental setting from [4], and train a deep network on the WDRef dataset [4]. The WDRef dataset contains 2995 subjects and about 20 samples per subject, which is significantly smaller than a typical (proprietary) training set for deep learning, e.g., 4.4 million labeled faces from 4,030 people in [17], or 202,599 face images from 10, 177 subjects in [15]. The goal of this paper is not to reproduce the success of deep learning in face verification [8, 17], but to compare the proposed GDT with several popular objectives optimized in a deep network. In our experiment, each face is described using a high dimensional LBP feature [5] available at [1], which is reduced to dimension 5,000 using PCA.

We compare the proposed GDT with two state-of-the-art deep learning objectives: DeepFace (DF) [17], and Deep Metric Learning (DML) [8]. To enable a fair comparison, we adopt the same network structure and input features for all compared methods, but keep their respective objective functions. DF feeds the output of the last layer to a K-way soft-max to predict the probability distribution over K classes, and minimizes a softmax loss. DML uses the Euclidean distance metric, and minimizes the loss defined in (8). The function $f_{\boldsymbol{\alpha}}(\cdot)$ in (3) is implemented as a two-layer fully connected network with $\tanh$ as the squash function, and the same network structure is used for DF and DML. Weight decay (conventional Frobenius norm regularization) is adopted in both DF and DML. And a range of weight decaying factor is tried and the best testing performance is reported. The network is trained on WDref and then applied to the LFW. To reflect the discriminability of the transformed features, we only use a simple verification method, by comparing the cosine distance between a given face pair to a threshold.

Table 1: Verification accuracy and AUC on LFW

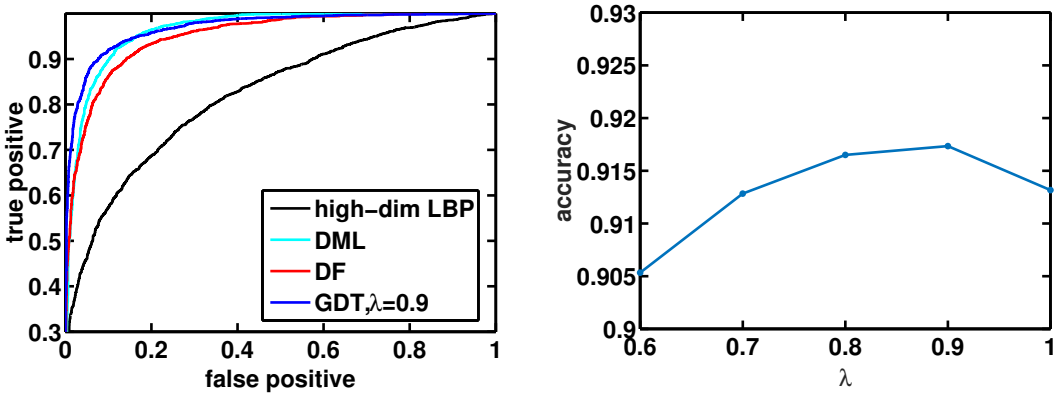
Method	accuracy (%)	AUC
High-dim LBP	74.73	0.8222±0.01
DF	88.72	0.9550±0.0029
DML	90.20	0.9640±0.0027
GDT	**91.72**	**0.9724**±0.0029

The ROCs for all methods are reported in Fig. 7a. Verification accuracies and area under the ROC curves (AUC) are listed in Table 1. High-dim LBP denotes the original features before transform. DF optimizes for a classification objective, the softmax loss, and separates well samples from different classes; however, it enforces no explicit constraints to assign similar representations to the same class. DML enforces discriminative pairwise distance; but, as illustrated before, becomes less robust when restricted to a small training set. As analyzed in Section 2.3, the proposed GDT is less conservative than DF for better discriminability; and, at the same time, expects smaller generalization errors than DML by preserving the local geometry (3). We observe that GDT outperforms both DF and DML by achieving a balance between discrimination and robustness. Face verification accuracies are shown in Fig. 7b by varying λ from 0.6 to 1; and peak accuracy is observed at $\lambda = 0.9$, illustrating the effectiveness of geometry preservation. Considering



(a) Empirical loss R_{emp} (b) Generalization error (c) Test loss R (d) Classification accuracy

Figure 6: GDT on Mnist: colors specify different training set sizes.



(a) Comparison with state-of-art. (b) Varying λ in GDT.

Figure 7: Verification accuracy on LFW.

the facts that both DF and DML are state-of-the-art deep learning methods that report revolutionary face verification results, the improvements reported here clearly demonstrate the strength of the geometry-aware deep transform.

We demonstrated here how the discriminability of original features, e.g., high-dim LBP here, can be improved with a learned feature transform. As emphasized, the goal is not to reproduce the success of deep learning in face verification (which can't be done due to the lack of availability of the data used in the corresponding papers); thus, we perform verification by simply comparing the cosine distance between each pair with a threshold. Note that more advanced verification techniques such as *JointBayes* [4] can always be adopted for improved accuracies; for example, [5] reports 95.17% accuracy by applying the JointBayes method on the high-dim LBP features. As observed in [15], we also expect steady improvements in verification accuracy by increasing the number of subjects used in training a deep network.

4. Conclusion

We proposed a geometry-aware deep transform that unifies both the classification and metric learning objectives commonly optimized in learning a deep network. We provided both experimental and theoretic illustrations to show that our method achieves a balance between discrimination and robustness, especially when restricted to a small training set. We demonstrated the effectiveness of the proposed deep learning objective using real-world data for applications such as face verification.

Acknowledgement

Work partially supported by NSF and DoD.

References

[1] `http://home.ustc.edu.cn/chendong/`.

[2] A. Bellet and A. Habrard. Robustness and generalization for metric learning. *arXiv:1209.1086v3*, 2014.

[3] Q. Cao, Y. Ying, and P. Li. Similarity metric learning for face recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2408–2415, 2013.

[4] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *European Conference on Computer Vision (ECCV)*, 2012.

[5] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[6] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M.Ranzato, and N. A. Y. Large scale distributed deep networks. in advances in neural information processing systems (pp.). In *Advances in Neural Information Processing Systems*, pages 1223–1231, 2012.

[7] H. Fan, Z. Cao, Y. Jiang, Q. Yin, and C. Doudou. Learning deep face representation. In *arXiv:1403.2802*, 2014.

[8] J. Hu, J. Lu, and Y. Tan. Discriminative deep metric learning for face verification in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1875–1882, 2014.

[9] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] R. Livni, S. Shalev-Shwartz, and O. Shamir. On the computational efficiency of training neural networks. In *In Advances in Neural Information Processing Systems*, number 855-863, 2014.

[12] Z. Lu, P. Jain, and I. S. Dhillon. Geometry-aware metric learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 673–680, 2009.

[13] S. Mendelson, A. Pajor, and N. Tomczak-Jaegermann. Uniform uncertainty principle for bernoulli and subgaussian ensembles. *Constructive Approximation*, 28:227–289, 2008.

[14] H. V. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *ACCV*, pages 709–720, 2010.

[15] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pages 1988–1996, 2014.

[16] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1891–1898, 2014.

[17] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708, 2014.

[18] H. Xu and S. Mannor. Robustness and generalization. *Machine Learning*, 86(3):391–423, 2012.