Unequal-training for Deep Face Recognition

Yaoyao Zhong, Weihong Deng, Mei Wang, Jiani Hu
[Beijing University of Posts and Telecommunications]

Jianteng Peng, Xunqiang Tao, Yaohai Huang
[Cannon Information Technology (Beijing) Co., Ltd]

CVPR 2019

Sungman, Cho.

Introduction

Introduction

- Real-world face datasets are usually large-scale and exhibit a **long-tailed distribution**, which present **three challenges** for model training.

1. extremely unbalanced data
2. million level identities
3. inherent noise

long-tailed distribution

Introduction

- Real-world face datasets are usually large-scale and exhibit a **long-tailed distribution**, which present **three challenges** for model training.

1. **<u>extremely unbalanced data</u>**

   A model trained on the whole long-tailed dataset will perform worse than that trained on a specific proportion of the whole dataset (cutting 50% tail) [CVPR, 2017]

2. million level identities

3. inherent noise

Introduction

- Real-world face datasets are usually large-scale and exhibit a **long-tailed distribution**, which present **three challenges** for model training.

1. extremely unbalanced data
2. **<u>million level identities</u>**

   Batch-size will be lowered, which will make the training loss difficult to converge. [ICCV Workshops, 2017]
3. inherent noise

Introduction

- Real-world face datasets are usually large-scale and exhibit a **long-tailed distribution**, which present **three challenges** for model training.

1. extremely unbalanced data
2. million level identities
3. **<u>inherent noise</u>**

   Million level datasets typically would even have a noise ratio higher than 30% [arXiv:1807.11649, 2018]

Introduction

- Real-world face datasets are usually large-scale and exhibit a **long-tailed distribution**, which present **three challenges** for model training.

- Current methods **cannot stably make full use of discriminative information in the long-tailed** noisy dataset.
  - re-sampling [Journal of Artificial Intelligence Research, 2002]
  - cost-sensitive weighting [ICML, 2000]
  - compensating the tail data. [ACM Multimedia, 2017] [CVPR, 2017]

Introduction

- The **head identities** with abundant samples are suitable to characterize the **intra-class variability**
- the large number of **tail identities** could offer abundant **inter-class** information.

→ Two training data streams

Introduction

Two training data streams

The first stream, based on the **head data**, is used for training relative discriminative face representation supervised by a **noise-resistant loss**

The second stream, based on the **tail data**, is used to learn the stable inter-class discriminative information by **hard identities mining.**

The Approach

Overview

The head data
images per person
The tail data
Identities
Split
The head data
The tail data
gradually
Forward
Backward
Fc Layer
Share weights
Noise Resistance Loss
Noise
W3
Hypothetical W
for a Noisy sample
ID3
W1
ID1
W2
ID2
Center-dispersed Loss
Hard Identities Mining
Center
Training sample

Three Steps

1. Splitting the training dataset
2. Constructing a noise resistance model
3. Joint training with the tail data

Constructing a noise resistance model

Analyze the types of noise in the training set, and the differences between noisy training data and correctly labeled data, so that **we could distinguish noisy data from clean ones.**

Constructing a noise resistance model

Noise Resistance Loss

$$L_{NASB} = -\frac{1}{N}\sum_{i=1}^{N}\left(\alpha(P_{y_{i_p}})\log(P_{y_i}) + \beta(P_{y_i})\log\left(P_{y_{i_p}}\right)\right), \quad (1)$$

$$\alpha(P) = \begin{cases} \rho, & P > t \\ 0, & P \le t \end{cases}, \quad \beta(P) = \begin{cases} 1-\rho, & P > t \\ 0, & P \le t \end{cases}. \quad (2)$$

P_{y_i} : predict probability of "true" class

$P_{y_{i_p}}$: current predict class

<u>At the beginning of training,</u>

ρ is set to 1 and t is set to 0.

<u>when the model is trained relatively sufficient,</u>

ρ is reduced slightly
t is set to a small value.

Noise Resistance Softmax loss

$$P_{y_i} = \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}, \quad P_{y_{i_p}} = \frac{e^{W_{y_{i_p}}^T x_i + b_{y_{i_p}}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}, \quad (3)$$

Constructing a noise resistance model

Noise Resistance Loss

$$L_{NASB} = -\frac{1}{N}\sum_{i=1}^{N}\left(\alpha(P_{y_{i_p}})\log(P_{y_i}) + \beta(P_{y_i})\log\left(P_{y_{i_p}}\right)\right), \quad (1)$$

At the beginning of Training

$$L_{NASB} = -\frac{1}{N}\sum_{i=1}^{N}\left(\log(P_{y_i})\right), \quad (1)$$

P_{y_i} : predict probability of "true" class

$P_{y_{i_p}}$: current predict class

<u>At the beginning of training,</u>

ρ is set to 1 and t is set to 0.

<u>when the model is trained relatively sufficient,</u>

ρ is reduced slightly
t is set to a small value.

$$\alpha(P) = \begin{cases} \rho, & P > t \\ 0, & P \le t \end{cases}, \quad \beta(P) = \begin{cases} 1-\rho, & P > t \\ 0, & P \le t \end{cases}. \quad (2)$$

Constructing a noise resistance model

Noise Resistance Loss

$$L_{NASB} = -\frac{1}{N}\sum_{i=1}^{N}\left(\alpha(P_{y_{i_p}})\log(P_{y_i}) + \beta(P_{y_i})\log\left(P_{y_{i_p}}\right)\right), \quad (1)$$

When the model is trained sufficiently

$$L_{NASB} = -\frac{1}{N}\sum_{i=1}^{N}\left(\alpha(P_{y_{i_p}})\log(P_{y_i}) + \beta(P_{y_i})\log\left(P_{y_{i_p}}\right)\right), \quad (1)$$

$$\alpha(P) = \begin{cases} \rho, & P > t \\ 0, & P \le t \end{cases}, \quad \beta(P) = \begin{cases} 1-\rho, & P > t \\ 0, & P \le t \end{cases}. \quad (2)$$

P_{y_i} : predict probability of "true" class

$P_{y_{i_p}}$: current predict class

At the beginning of training,

ρ is set to 1 and t is set to 0.

when the model is trained relatively sufficient,

ρ is reduced slightly
t is set to a small value.

Constructing a noise resistance model

Noise Resistance CosFace

$$
\begin{aligned}
P_{y_i} &= \frac{e^{s\left(\cos\theta_{y_i} - m_C\right)}}{e^{s\left(\cos\theta_{y_i} - m_C\right)} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos\theta_j}}, \\
P_{y_{i_p}} &= \frac{e^{s\cos\theta_{y_{i_p}}}}{e^{s\left(\cos\theta_{y_i} - m_C\right)} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos\theta_j}},
\end{aligned}
\tag{5}
$$

Noise Resistance ArcFace

$$
\begin{aligned}
P_{y_i} &= \frac{e^{s\left(\cos\left(\theta_{y_i} + m_A\right)\right)}}{e^{s\left(\cos\left(\theta_{y_i} + m_A\right)\right)} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos\theta_j}}, \\
P_{y_{i_p}} &= \frac{e^{s\cos\theta_{y_{i_p}}}}{e^{s\left(\cos\left(\theta_{y_i} + m_A\right)\right)} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos\theta_j}},
\end{aligned}
\tag{6}
$$

Joint training with the tail data

Three challenges in tail data.

1. There exists a large number of identities in the tail data.
2. Each tail identity only contains rarely samples.
3. Considerable portion of them is noisy.

→ Have to extract the most credible information of the tail.

: Center-dispersed Loss

Center-dispersed Loss

Extract features of tail identities using the base model; then add the tail data gradually.

$$L_{CD} = \min \frac{1}{m(m-1)} \sum_{k=1}^{m(m-1)} {S_{i,j}}^2 .$$

$S_{i,j}$: simlilarity between identity i and j in mini-batch.

$$S_{i,j} = \left(\frac{C_i}{\|C_i\|}\right)^T \left(\frac{C_j}{\|C_j\|}\right).$$

C_i , C_j : represent identity i and j . (formulated as the center of normalized features)

$$C_i = \frac{1}{n} \sum_{t=1}^{n} \frac{x_t}{\|x_t\|}, n \leqslant n_i,$$

Algorithm 1 The joint training process in the second stage

Input:

The head data D_h, the tail data D_t, base model $(\theta_{ResNet}, W_{fc})$.

Output:

The model $(\theta_{ResNet}, W_{fc})$.

Global parameters:

Hyperparameter in Noisy Resistance (NR) loss ρ, t. Mini-batch size (NR) $s1$.

Weight of Center-dispersed (CD) Loss η. Number of identities in a mini-batch (CD) m, in a candidate bag (CD) M. Samples of an identity n. So mini-batch size (CD) $s2 = m \times n$, Candidate bag size (CD) $s = M \times n$.

Initialization at the beggining of an epoch:

// Constructing queues $Q1$ for NR and $Q2$ for CD.

$Q1$=$Q2$={}.

for x_i **in** D_h **do**

  if $P_{y_{i_p}} > t$ **then**

    $Q1$.append(x_i).

  end if

end for

for ID_i **in** D_t **do**

  Randomly select n samples $\{x_1, \cdots, x_n\}$ of ID_i.

  $Q2$.append($\{x_1, \cdots, x_n\}$).

end for

shuffle $Q1$ and $Q2$.

Optimization in an epoch:

while $Q1$ is not empty **do**

  $B_1 \leftarrow$Take out a mini-batch with $s1$ samples in $Q1$.

  L_{NASB} (1), $\nabla L_{NASB} \leftarrow B_1$.

  $(\theta_{ResNet}, W_{fc}) \leftarrow \nabla L_{NASB}$

end while

while $Q2$ is not empty **do**

  $C1 \leftarrow$Take out a candidate bag with M identities in $Seq2$, extract their features using θ_{ResNet}.

  $list_{ID} = \{ID_1, \cdots, ID_M\} \leftarrow$Calculate and sort $S^k_{i,j}$ (8) in $C1$.

  $B_2 \leftarrow$Constructing a mini-batch with first m identities in $list_{ID}$. // Hard Identities mining

  L_{CD} (7),$\nabla L_{CD} \leftarrow B_2$.

  $\theta_{ResNet} \leftarrow \eta \nabla L_{CD}$.

end while

Experiments

Distribution of imbalanced datasets

Training Data→	WebFace			WebFace+tail(one shot)			WebFace+tail(low shot)		
Method↓	LFW	CPLFW	YTF	LFW	CPLFW	YTF	LFW	CPLFW	YTF
Softmax Loss	99.25	83.72	94.74	99.37	83.32	95.10	99.15	83.10	94.94
Ours(NRS+CD)	**99.28**	**83.75**	**95.00**	**99.40**	**84.75**	**95.78**	**99.40**	**84.97**	**95.58**
CosFace(LMCL) [36]	99.55	87.67	95.52	99.47	87.85	96.12	99.50	87.25	95.60
Ours(NRC+CD)	99.43	**87.92**	**95.64**	**99.48**	**88.18**	**96.12**	99.47	**88.00**	**95.98**
ArcFace [7]	99.52	87.20	95.60	99.55	86.32	93.82	99.40	88.15	94.96
Ours(NRA+CD)	**99.55**	**87.53**	95.50	99.53	**88.03**	**96.04**	**99.55**	**88.22**	**95.76**

Table 1. Results on the controlled experiments by varying training datasets and training methods.

Distribution of imbalanced datasets

Training Data→	WebFace			WebFace+tail(one shot)			WebFace+tail(low shot)		
Method↓	LFW	CPLFW	YTF	LFW	CPLFW	YTF	LFW	CPLFW	YTF
Softmax Loss	99.25	83.72	94.74	99.37	83.32	95.10	99.15	83.10	94.94
Ours(NRS+CD)	**99.28**	**83.75**	**95.00**	**99.40**	**84.75**	**95.78**	**99.40**	**84.97**	**95.58**
CosFace(LMCL) [36]	99.55	87.67	95.52	99.47	87.85	96.12	99.50	87.25	95.60
Ours(NRC+CD)	99.43	**87.92**	**95.64**	**99.48**	**88.18**	**96.12**	99.47	**88.00**	**95.98**
ArcFace [7]	99.52	87.20	95.60	99.55	86.32	93.82	99.40	88.15	94.96
Ours(NRA+CD)	**99.55**	**87.53**	95.50	99.53	**88.03**	**96.04**	**99.55**	**88.22**	**95.76**

Table 1. Results on the controlled experiments by varying training datasets and training methods.

MegaFace Challenge

Comparison with 64-layer ResNet results.

Method	Protocol	Rank1 Acc.	Ver.
SphereFace(A-Softmax) [21]	Large	71.17	84.22
CosFace(LMCL) [36]	Large	74.11	86.77
Rangeloss [44]	Large	69.54	82.67
LMLE [13]	Large	74.76	87.78
CLMLE [14]	Large	76.26	89.41
Ours(ResNet64)	Large	**78.12**	88.03

Comparison with top results on MF2 Leaderboard.

Method	Protocol	Rank1 Acc.	Ver.
3DiVi	Large	57.05	66.46
NEC	Large	62.12	66.85
GRCCV	Large	75.77	74.84
Yang Sun	Large	75.79	84.03
CosFace(LMCL) [36]	Large	74.11	86.77
Ours(ResNet50)	Large	**80.02**	**89.93**

Thank You.