Domain Separation Networks

Konstantinos Bousmalis[1] George Trigeorgis[2] Nathan Silberman[3]
Dilip Krishnan[3] Dumitru Erhan [1]

[1]Google Brain

[2]Imperial College London

[3]Google Research

NIPS, 2016
Presenter: Xueying Bai

Outline

1. Introduction
   - Motivation
   - Overview
   - Related Work
2. Method(DSN)
   - Overview
   - Learning
   - Similarity Losses
3. Evaluation
   - Settings
   - Accuracy
   - Analysis

Outline

1. Introduction
   - Motivation
   - Overview
   - Related Work
2. Method(DSN)
   - Overview
   - Learning
   - Similarity Losses
3. Evaluation
   - Settings
   - Accuracy
   - Analysis

Motivation

Two problems faced in the machine learning application:

- **Large data collection and annotation is expensive.**
  Can create large-scale synthetic dataset instead.
- **Enough synthetic data, but not perform well on realistic domains.**
  Learn representations that are domaininvariant in scenarios where the data distributions during training and testing are different.

The source data is labeled for a particular task. The task in this paper is to transfer knowledge from the source to the target domain which has no ground truth labels.

Outline

1. Introduction
   - Motivation
   - Overview
   - Related Work
2. Method(DSN)
   - Overview
   - Learning
   - Similarity Losses
3. Evaluation
   - Settings
   - Accuracy
   - Analysis

Overview

- **Different data distributions in the source and target domain:**
  low level: noise, resolution, illumination and color
  high level: number of classes, type of objects, geometric variations.
  In this paper, data from source and target domains have low level distribution difference, while have similar high-level distributions and the same label space.
- **Method:**
  A private subspace for each domain: capture specific domain properties.
  A shared subspace: capture representations shared by domains.

Outline

1. Introduction
    - Motivation
    - Overview
    - Related Work
2. Method(DSN)
    - Overview
    - Learning
    - Similarity Losses
3. Evaluation
    - Settings
    - Accuracy
    - Analysis

Related Work

- **Study of upper bounds on a domain-adapted classifier in the target domain:** Train a binary classifier to distinguish source and target domains. Errors on each domains are used to decide the bound.
- **Domain Adversarial Neural Networks (DANN):**An architecture with two classifiers trained simultaneously: the first is trained to correctly predict task-specific class labels on the source data while the second is trained to predict the domain of each input.
- **Maximum Mean Discrepancy(MMD) metric:** A metric to calculate the domain classification loss.

Outline

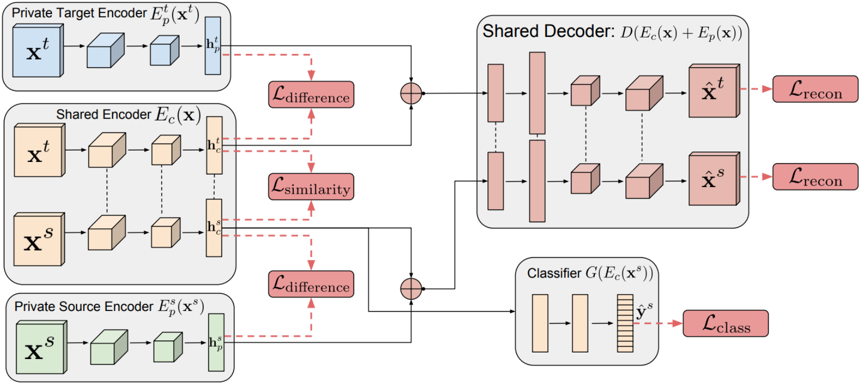
1 Introduction
- Motivation
- Overview
- Related Work

2 Method(DSN)
- Overview
- Learning
- Similarity Losses

3 Evaluation
- Settings
- Accuracy
- Analysis

Overview

Explicitly model both private and shared components of the domain representations.


Private Target Encoder $E_p^t(\mathbf{x}^t)$
$\mathbf{x}^t$
$\mathbf{h}_p^t$
Shared Encoder $E_c(\mathbf{x})$
$\mathbf{x}^t$
$\mathbf{h}_c^t$
$\mathbf{x}^s$
$\mathbf{h}_c^s$
Private Source Encoder $E_p^s(\mathbf{x}^s)$
$\mathbf{x}^s$
$\mathbf{h}_p^s$
$\mathcal{L}_{\text{difference}}$
$\mathcal{L}_{\text{similarity}}$
$\mathcal{L}_{\text{difference}}$
Shared Decoder: $D(E_c(\mathbf{x}) + E_p(\mathbf{x}))$
$\hat{\mathbf{x}}^t$
$\hat{\mathbf{x}}^s$
$\mathcal{L}_{\text{recon}}$
$\mathcal{L}_{\text{recon}}$
Classifier $G(E_c(\mathbf{x}^s))$
$\hat{\mathbf{y}}^s$
$\mathcal{L}_{\text{class}}$

$X^S = \{(x_i^s, y_i^s)\}_{i=0}^{N_s}$: labeled dataset from the source domain.

$X^t = \{x_i^t\}_{i=0}^{N_t}$: unlabeled dataset from the target domain

Outline

1 Introduction
- Motivation
- Overview
- Related Work

2 Method(DSN)
- Overview
- Learning
- Similarity Losses

3 Evaluation
- Settings
- Accuracy
- Analysis

Learning

$$L = L_{task} + \alpha L_{recon} + \beta L_{difference} + \gamma L_{similarity}$$

$$L_{task} = -\sum_{i=0}^{N_s} y_i^s \cdot \log \widehat{y_i^s}$$

$$\widehat{y} = G(E_c(x)), \widehat{x} = D(E_c(x) + E_p(x))$$

$$L_{recon} = \sum_{i=0}^{N_s} L_{si\_mse}(x_i^s, \widehat{x_i^s}) + \sum_{i=0}^{N_t} L_{si\_mse}(x_i^t, \widehat{x_i^t})$$

$$L_{si\_mse}(x, \widehat{x}) = \frac{1}{k} ||x - \widehat{x}||_2^2 - \frac{1}{k^2}([x - \widehat{x}] \cdot 1_k)^2$$

k is the number of pixels in the input x. L_{si_mse} is the scale-invariant MSE.

$$L_{difference} = \left\| {H_c^s}^T H_p^s \right\|_F^2 + \left\| {H_c^t}^T H_p^t \right\|_F^2$$

H_c^s: each row is the hidden shared representation $h_c^s = E_c(x_s)$

H_p^s: each row is the hidden private representation $h_p^s = E_p^s(x_s)$

Outline

1. Introduction
   - Motivation
   - Overview
   - Related Work
2. Method(DSN)
   - Overview
   - Learning
   - Similarity Losses
3. Evaluation
   - Settings
   - Accuracy
   - Analysis

DANN Similarity

The domain adversarial similarity loss is used to train a model to produce representations such that a classifier cannot reliably predict the domain of the encoded representation.

- Gradient Reversal Layer (GRL):

$$Q(f(u)) = f(u)$$

$$\frac{d}{du} Q(f(u)) = -\frac{d}{du} f(u)$$

- Domain Classifier:

$$Z(Q(h_c); \theta_z) \rightarrow \widehat{d}$$

- Loss Function:

$$L_{similarity} = \sum_{i=0}^{N_s+N_t} \left\{ d_i \log \widehat{d_i} + (1 - d_i) \log(1 - \widehat{d_i}) \right\}$$

MMD Similarity

The Maximum Mean Discrepancy (MMD) loss is a kernel-based distance function between pairs of samples.

- MMD loss:

$$L^{MMD}_{similarity} = \frac{1}{(N^s)^2}\sum_{i,j=0}^{N^s} k(h^s_{ci}, h^s_{cj}) - \frac{2}{N^s N^t}\sum_{i,j=0}^{N^s,N^t} k(h^s_{ci}, h^t_{cj})$$

$$+\frac{1}{(N^t)^2}\sum_{i,j=0}^{N_t} k(h^t_{ci}, h^t_{cj})$$

- Linear combination of multiple RBF kernels:

$$k(x_i, x_j) = \sum\nolimits_n \eta_n \exp\{-\frac{1}{2\sigma_n}\|x_i - x_j\|^2\}$$

Outline

1. Introduction
   - Motivation
   - Overview
   - Related Work
2. Method(DSN)
   - Overview
   - Learning
   - Similarity Losses
3. Evaluation
   - Settings
   - Accuracy
   - Analysis

Settings

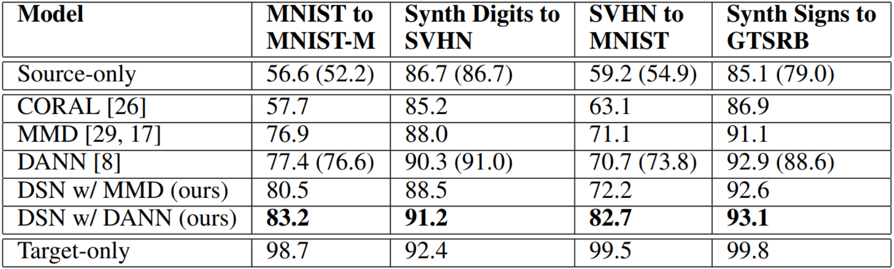
Evaluation is based on training on a clean dataset and testing on noisy dataset.

- **Data for training:**
  Source domain training: labeled training data from the source domain.
  Domain adaptation training: unlabeled data from the target domain.
  Hyperparameters: labeled data from the target domain(test set).
  Test: test set from the target domain.
- **5 transfer scenarios:**
  (a). MNIST to MNIST-M: MNIST-M was created by using each MNIST digit as a binary mask and inverting with it the colors of a background image.
  (b). Synthetic Digits to SVHM
  (c). SVHN to MNIST
  (d). Synthetic Signs to GTSRB
  (e). Synthetic Objects to LineMod: pose estimation.

Outline

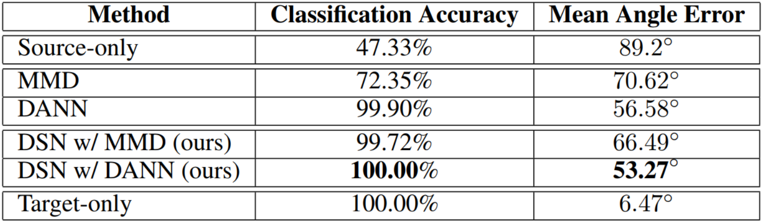
1. Introduction
   - Motivation
   - Overview
   - Related Work
2. Method(DSN)
   - Overview
   - Learning
   - Similarity Losses
3. Evaluation
   - Settings
   - Accuracy
   - Analysis

Accuracy



Model	MNIST to MNIST-M	Synth Digits to SVHN	SVHN to MNIST	Synth Signs to GTSRB
Source-only	56.6 (52.2)	86.7 (86.7)	59.2 (54.9)	85.1 (79.0)
CORAL [26]	57.7	85.2	63.1	86.9
MMD [29, 17]	76.9	88.0	71.1	91.1
DANN [8]	77.4 (76.6)	90.3 (91.0)	70.7 (73.8)	92.9 (88.6)
DSN w/ MMD (ours)	80.5	88.5	72.2	92.6
DSN w/ DANN (ours)	**83.2**	**91.2**	**82.7**	**93.1**
Target-only	98.7	92.4	99.5	99.8

Mean classification accuracy and pose error for the “Synth Objects to LINEMOD” scenario.

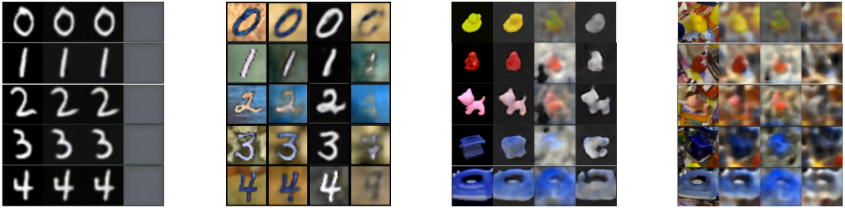


Method	Classification Accuracy	Mean Angle Error
Source-only	47.33%	89.2°
MMD	72.35%	70.62°
DANN	99.90%	56.58°
DSN w/ MMD (ours)	99.72%	66.49°
DSN w/ DANN (ours)	**100.00%**	**53.27°**
Target-only	100.00%	6.47°

Outline

1. Introduction
   - Motivation
   - Overview
   - Related Work
2. Method(DSN)
   - Overview
   - Learning
   - Similarity Losses
3. Evaluation
   - Settings
   - Accuracy
   - Analysis

Analysis of Reconstruction



(a) MNIST (source) (b) MNIST-M (target) (c) Synth Objects (source) (d) LINEMOD (target)

Figure 2: Reconstructions for the representations of the two domains for “MNIST to MNIST-M” and for “Synth Objects to LINEMOD”. In each block from left to right: the original image $\mathbf{x}_t$; reconstructed image $D(E_c(\mathbf{x}^t) + E_p(\mathbf{x}^t))$; shared only reconstruction $D(E_c(\mathbf{x}^t))$; private only reconstruction $D(E_p(\mathbf{x}^t))$.

Analysis of Loss Functions

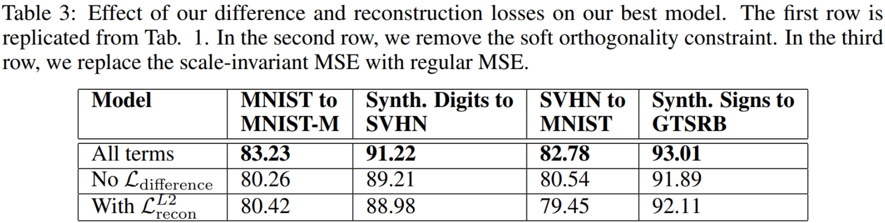


Table 3: Effect of our difference and reconstruction losses on our best model. The first row is replicated from Tab. 1. In the second row, we remove the soft orthogonality constraint. In the third row, we replace the scale-invariant MSE with regular MSE.

Model	**MNIST to MNIST-M**	**Synth. Digits to SVHN**	**SVHN to MNIST**	**Synth. Signs to GTSRB**
All terms	**83.23**	**91.22**	**82.78**	**93.01**
No $\mathcal{L}_{\text{difference}}$	80.26	89.21	80.54	91.89
With $\mathcal{L}_{\text{recon}}^{L2}$	80.42	88.98	79.45	92.11