Boulder

Computer Vision; Image Classification; Domain Adaptation

Maziar Raissi

Assistant Professor

Department of Applied Mathematics

University of Colorado Boulder

maziar.raissi@colorado.edu

Domain-Adversarial Training of Neural Networks

Domain Adaptation

Data at training and test time come from similar but different distributions!

$X \rightarrow$ input space

$Y = \{0, 1, \ldots, L-1\} \rightarrow$ set of L possible labels

$\mathcal{D}_S \rightarrow$ source domain

$\mathcal{D}_T \rightarrow$ target domain

$\mathcal{D}_S, \mathcal{D}_T \rightarrow$ distributions over $X \times Y$

$S \rightarrow$ labeled source sample drawn i.i.d from $\mathcal{D}_S$

$T \rightarrow$ unlabeled target sample drawn i.i.d from $\mathcal{D}_T^X$

$\mathcal{D}_T^X \rightarrow$ marginal distribution of $\mathcal{D}_T$ over X

$S = \{(x_i, y_i)\}_{i=1}^{n} \sim (\mathcal{D}_S)^n$

$T = \{x_i\}_{i=n+1}^{N} \sim (\mathcal{D}_T^X)^{n'}$

$N = n + n' \rightarrow$ total number of samples

Build a classifier $\eta : X \rightarrow Y$ with a low target risk:

$\mathcal{R}_{\mathcal{D}_T}(\eta) = \Pr_{(x,y)\sim\mathcal{D}_T}(\eta(x) \neq y)$

Theorem

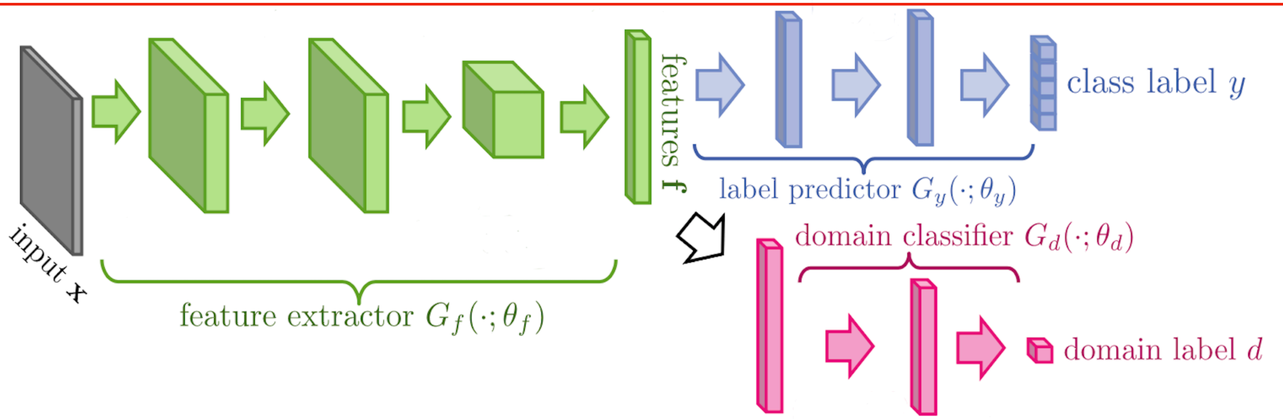
> For effective domain transfer to be achieved, predictions must be made based on features that cannot discriminate between the training (source) and test (target) domains.

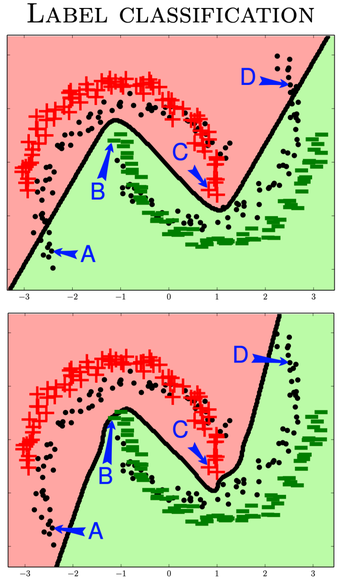
Domain-Adversarial Neural Network (DANN)

$G_f(\cdot;\theta_f) \rightarrow$ feature extractor

$G_y(\cdot;\theta_y) \rightarrow$ label predictor

$G_d(\cdot;\theta_d) \rightarrow$ domain classifier





$\mathcal{L}_y^i(\theta_f, \theta_y) = \mathcal{L}_y(G_y(G_f(x_i;\theta_f);\theta_y), y_i) \rightarrow$ prediction loss

$\mathcal{L}_d^i(\theta_f, \theta_d) = \mathcal{L}_d(G_d(G_f(x_i;\theta_f);\theta_d), d_i) \rightarrow$ domain loss

$\mathcal{L}_d(G_d(G_f(x_i)), d_i) = -d_i \log(G_d(G_f(x_i))) - (1-d_i)\log(1 - G_d(G_f(x_i)))$

$d_i = 1$ if $x_i \sim \mathcal{D}_T^X$

$d_i = 0$ if $x_i \sim \mathcal{D}_S^X$

$\mathcal{L}_y(G_y(G_f(x_i)), y_i) = -\log(G_y(G_f(x_i))_{y_i})$

$$(\hat{\theta}_f, \hat{\theta}_y) = \underset{\theta_f,\theta_y}{\operatorname{argmin}} E(\theta_f, \theta_y, \hat{\theta}_d)$$

$$\hat{\theta}_d = \underset{\theta_d}{\operatorname{argmax}} E(\hat{\theta}_f, \hat{\theta}_y, \theta_d)$$

$$E(\theta_f, \theta_y, \theta_d) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_y^i(\theta_f, \theta_y) - \lambda\left(\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_d^i(\theta_f, \theta_d) + \frac{1}{n'}\sum_{i=n+1}^{N} \mathcal{L}_d^i(\theta_f, \theta_d)\right)$$

Unsupervised Hyper-parameter Selection

Split S into training set S' and validation set S_V

Split T into training set T' and validation set T_V

Use S' and T' to learn a classifier η

$\{(x, \eta(x))\}_{x\in T'} \rightarrow$ self labeled set (new source)

Unlabeled $S' \rightarrow$ new target

$\eta_r \rightarrow$ reverse classifier

Evaluate η_r on S_V

$\mathcal{R}_{S_V}(\eta_r) \rightarrow$ reverse validation risk

$S_V \rightarrow$ used for early stopping the training of η

$T_V \rightarrow$ used for early stopping the training of η_r

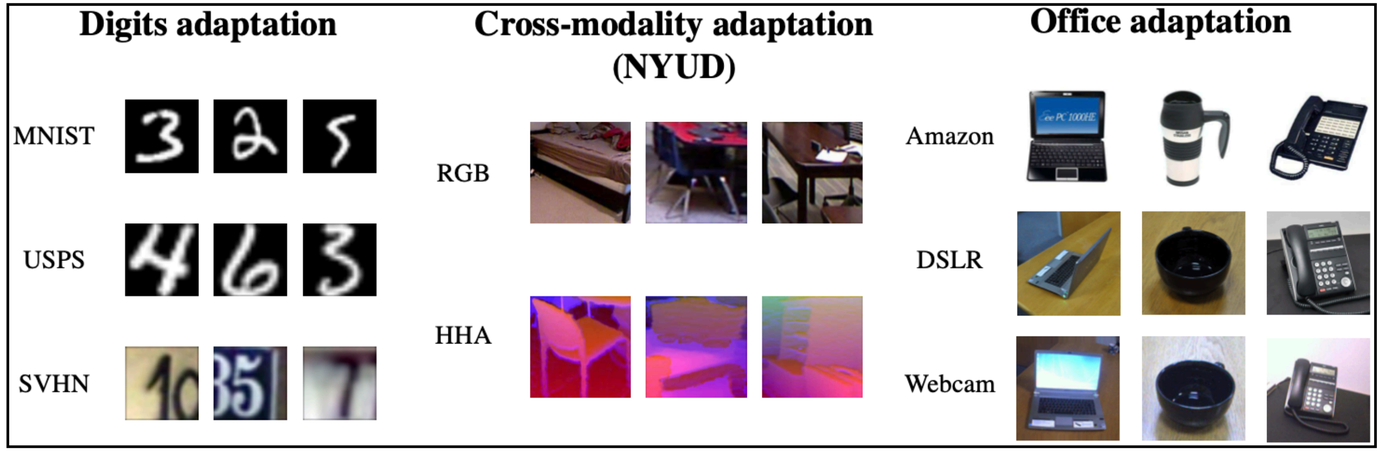
– document sentiment analysis

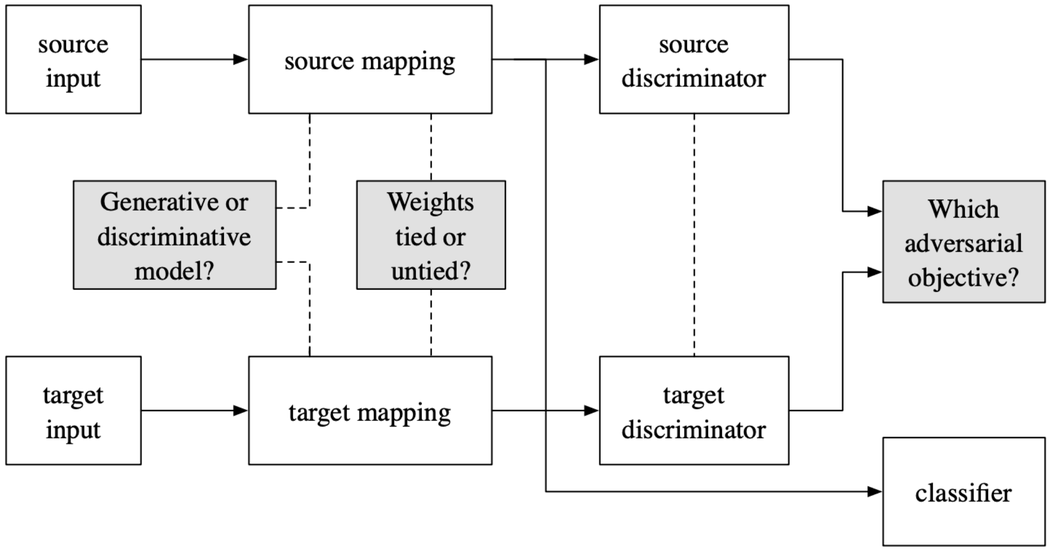
– image classification

–person re-identification

Ganin, Yaroslav, et al. "Domain-adversarial training of neural networks." *The journal of machine learning research* 17.1 (2016): 2096-2030.

Adversarial Discriminative Domain Adaptation





– dataset bias or domain shift

$(\mathbf{X}_s, Y_s) \rightarrow$ source images & labels $\quad \mathbf{X}_t \rightarrow$ target images (no labels)

$M_s, M_t \rightarrow$ source & target representation mappings

$C \rightarrow$ classifier $\quad K \rightarrow$ number of categories

$$\min_{M_s, C} \mathcal{L}_{\text{cls}}(\mathbf{X}_s, Y_s) =$$

$$- \mathbb{E}_{(\mathbf{x}_s, y_s) \sim (\mathbf{X}_s, Y_s)} \sum_{k=1}^{K} \mathbb{1}_{[k=y_s]} \log C(M_s(\mathbf{x}_s))$$

$$\min_{D} \mathcal{L}_{\text{adv}_D}(\mathbf{X}_s, \mathbf{X}_t, M_s, M_t) =$$

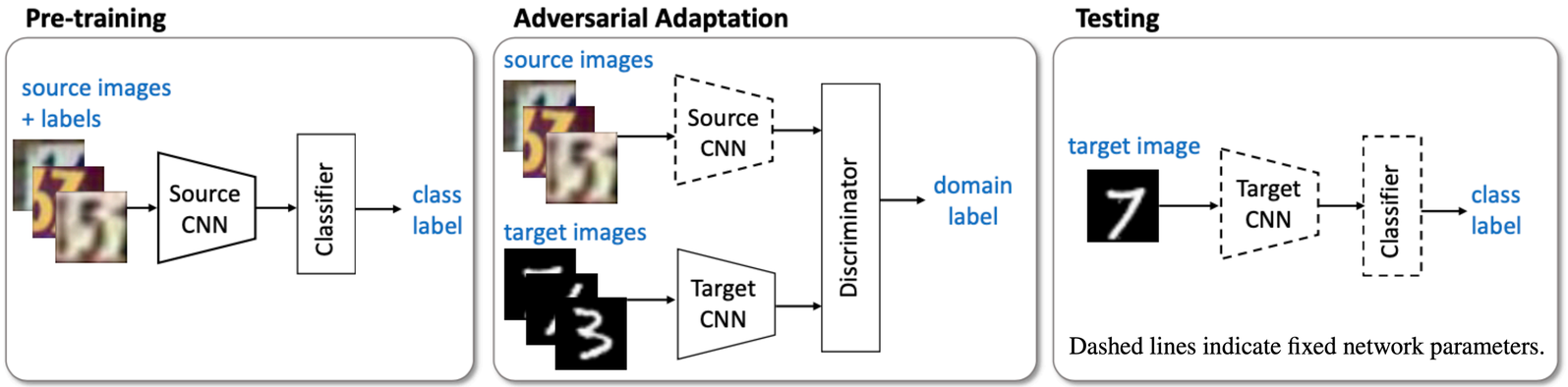
$$- \mathbb{E}_{\mathbf{x}_s \sim \mathbf{X}_s}[\log D(M_s(\mathbf{x}_s))] - \mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t}[\log(1 - D(M_t(\mathbf{x}_t)))]$$

$$\min_{M_t} \mathcal{L}_{\text{adv}_M}(\mathbf{X}_s, \mathbf{X}_t, D) =$$

$$- \mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t}[\log D(M_t(\mathbf{x}_t))].$$

Method	Base model	Weight sharing	Adversarial loss
Gradient reversal [19]	discriminative	shared	minimax
Domain confusion [12]	discriminative	shared	confusion
CoGAN [13]	generative	unshared	GAN
ADDA (Ours)	discriminative	unshared	GAN

Method	MNIST → USPS	USPS → MNIST	SVHN → MNIST
Source only	0.752 ± 0.016	0.571 ± 0.017	0.601 ± 0.011
Gradient reversal	0.771 ± 0.018	0.730 ± 0.020	0.739 [19]
Domain confusion	0.791 ± 0.005	0.665 ± 0.033	0.681 ± 0.003
CoGAN	0.912 ± 0.008	0.891 ± 0.008	did not converge
ADDA (Ours)	0.894 ± 0.002	0.901 ± 0.008	0.760 ± 0.018



Tzeng, Eric, et al. "Adversarial discriminative domain adaptation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

CyCADA: Cycle-Consistent Adversarial Domain Adaptation

Boulder

Adversarial Adaptation Models:
– discovering domain invariant representations (feature space methods)
– mapping between unpaired image domains (image space methods)

Unsupervised Adaptation

$X_S \rightarrow$ source data $\quad Y_S \rightarrow$ source labels
$X_T \rightarrow$ target data $\quad$ no target labels!

Learn a model f_T that correctly predicts the label for the target data X_T

Pre-train Source Task Model

$$\mathcal{L}_{\text{task}}(f_S, X_S, Y_S) = -\mathbb{E}_{(x_s,y_s)\sim(X_S,Y_S)} \sum_{k=1}^{K} \mathbb{1}_{[k=y_s]} \log\left(\sigma(f_S^{(k)}(x_s))\right)$$

σ: softmax function

domain shift leads to reduced performance when evaluating on target data

Pixel-level Adaptation

$G_{S\rightarrow T} \rightarrow$ mapping from source to target
↳trained to produce target samples that fool adversarial discriminator D_T

$$\mathcal{L}_{\text{GAN}}(G_{S\rightarrow T}, D_T, X_T, X_S) = \mathbb{E}_{x_t\sim X_T}[\log D_T(x_t)] + \mathbb{E}_{x_s\sim X_S}[\log(1 - D_T(G_{S\rightarrow T}(x_s)))]$$

$\mathcal{L}_{\text{task}}(f_T, G_{S\rightarrow T}(X_S), Y_S) \rightarrow$ learn a target model f_T

No way to guarantee that $G_{S\rightarrow T}(x_s)$ preserves the structure and content of x_s!

$$\mathcal{L}_{\text{GAN}}(G_{T\rightarrow S}, D_S, X_S, X_T)$$

$$\mathcal{L}_{\text{cyc}}(G_{S\rightarrow T}, G_{T\rightarrow S}, X_S, X_T) = \mathbb{E}_{x_s\sim X_S}[||G_{T\rightarrow S}(G_{S\rightarrow T}(x_s)) - x_s||_1] + \mathbb{E}_{x_t\sim X_T}[||G_{S\rightarrow T}(G_{T\rightarrow S}(x_t)) - x_t||_1]$$

Prevent Label Flipping

semantic consistency $\leftarrow \mathcal{L}_{\text{sem}}(G_{S\rightarrow T}, G_{T\rightarrow S}, X_S, X_T, f_S) =$

$$\mathcal{L}_{\text{task}}(f_S, G_{T\rightarrow S}(X_T), p(f_S, X_T)) + \mathcal{L}_{\text{task}}(f_S, G_{S\rightarrow T}(X_S), p(f_S, X_S))$$

$p(f, X) = \arg\max f(x)$
↳predicted label

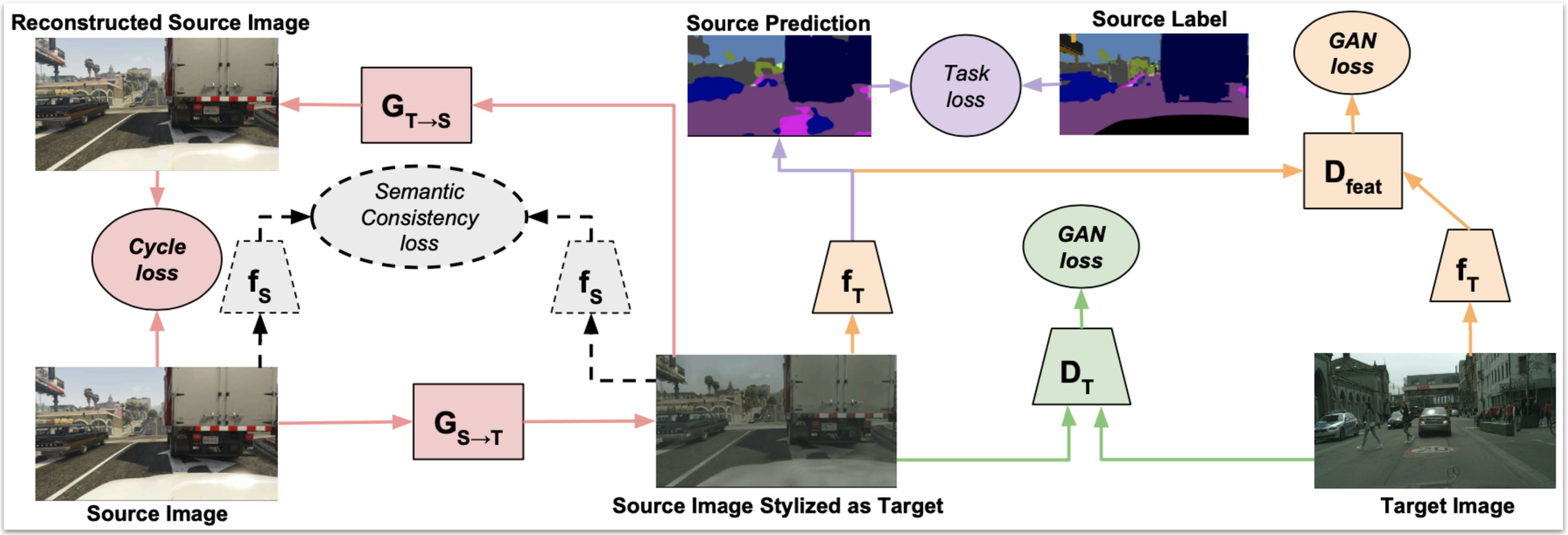
$f_S \rightarrow$ pretrained and fixed

Feature-level Adaptation

$$\mathcal{L}_{\text{GAN}}(f_T, D_{\text{feat}}, f_S(G_{S\rightarrow T}(X_S)), X_T)$$

Complete Objective

$$f_T^* = \arg\min_{f_T} \min_{\substack{G_{S\rightarrow T} \\ G_{T\rightarrow S}}} \max_{D_S, D_T} \mathcal{L}_{\text{CyCADA}}$$



SVHN→MNIST

Model	Accuracy (%)
Source only	67.1
CyCADA - no feat adapt, no semantic loss	70.3
CyCADA - no feat adapt	71.2
CyCADA - no cycle consistency	75.7
CyCADA - no pixel adapt	83.8
CyCADA (Full)	**90.4**
Target Fully Supervised	99.2

Hoffman, Judy, et al. "Cycada: Cycle-consistent adversarial domain adaptation." *International conference on machine learning*. PMLR, 2018.

Boulder

Questions?