Deep Transfer Learning

Mingsheng Long

School of Software
National Engineering Lab for Big Data Software
Tsinghua University

https://github.com/thuml

Outline

1. **Deep Transfer Learning**
2. **Problem 1:** $P(\mathbf{X}) \neq Q(\mathbf{X})$
3. **Problem 2:** $P(\mathbf{X}, Y) \neq Q(\mathbf{X}, Y)$
   - Joint Adaptation Network (JAN)
   - Conditional Domain Adversarial Network (CDAN)
4. **Evaluation**

Deep Learning

Learner: $f : \boldsymbol{x} \rightarrow y$ **Distribution:** $(\boldsymbol{x}, y) \sim P(\boldsymbol{x}, y)$

fish

bird

mammal

tree

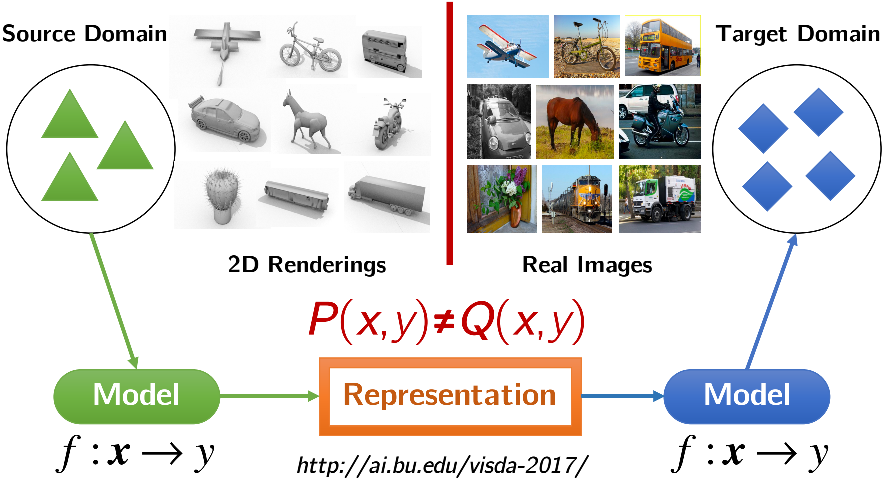
flower

......

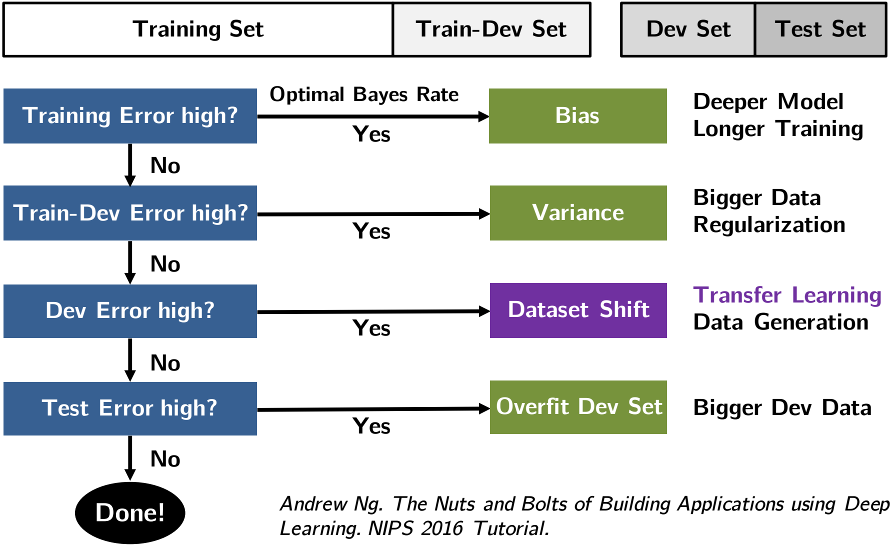
Error Bound: $\epsilon_{\text{test}} \leq \hat{\epsilon}_{\text{train}} + \sqrt{\frac{\text{complexity}}{n}}$

Deep Transfer Learning

- Deep learning across domains of different distributions $P \neq Q$



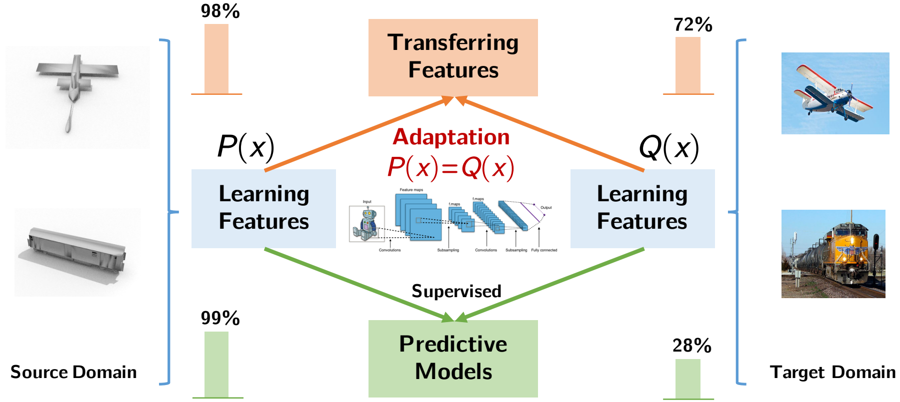
Deep Transfer Learning: Why?



Andrew Ng. The Nuts and Bolts of Building Applications using Deep Learning. NIPS 2016 Tutorial.

Deep Transfer Learning: How?

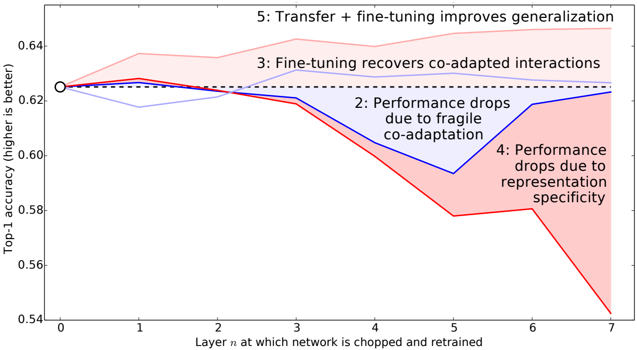
- Learning predictive models on transferable features s.t. $P(\mathbf{x}) = Q(\mathbf{x})$
- Distribution matching: **MMD** (ICML'15), **GAN** (ICML'15, JMLR'16)



How Transferable Are Deep Features?

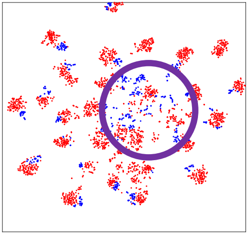
Transferability is restricted by (Yosinski et al. 2014; Glorot et al. 2011)

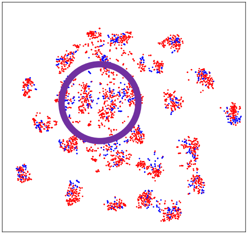
- Specialization of higher layer neurons to original task (new task ↓)
- Optimization difficulty in splitting nets between co-adapted neurons
- Disentangling of variations in higher layers enlarges task discrepancy
- Transferability of features decreases while task discrepancy increases

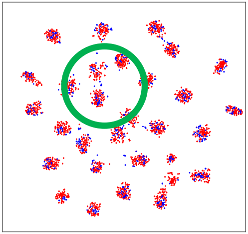


Distribution Mismatch

- Marginal distribution mismatch: $P(\mathbf{X}) \neq Q(\mathbf{X})$
- Conditional distribution mismatch: $P(Y|\mathbf{X}) \neq Q(Y|\mathbf{X})$







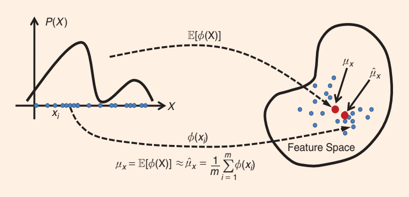
$P(\mathbf{x}) \neq Q(\mathbf{x})$
$P(\mathbf{y}|\mathbf{x}) \neq Q(\mathbf{y}|\mathbf{x})$

$P(\mathbf{x}) \approx Q(\mathbf{x})$
$P(\mathbf{y}|\mathbf{x}) \neq Q(\mathbf{y}|\mathbf{x})$

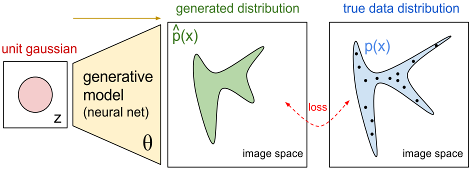
$P(\mathbf{x}) \approx Q(\mathbf{x})$
$P(\mathbf{y}|\mathbf{x}) \approx Q(\mathbf{y}|\mathbf{x})$

Distribution Matching

- Marginal distribution mismatch: $P(\mathbf{X}) \neq Q(\mathbf{X})$
- Conditional distribution mismatch: $P(Y|\mathbf{X}) \neq Q(Y|\mathbf{X})$



Kernel Embedding



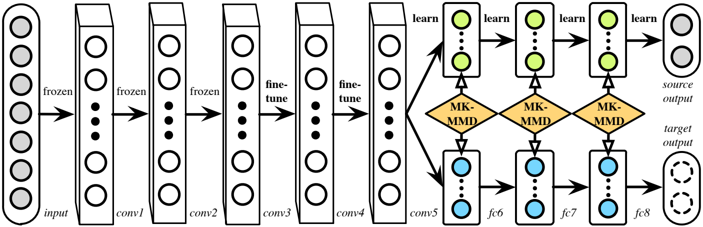
Adversarial Learning

Song et al. Kernel Embeddings of Conditional Distributions. ***IEEE****, 2013.*
Goodfellow et al. Generative Adversarial Networks. ***NIPS*** *2014.*

Outline

1. Deep Transfer Learning
2. **Problem 1: $P(\mathbf{X}) \neq Q(\mathbf{X})$**
3. Problem 2: $P(\mathbf{X}, Y) \neq Q(\mathbf{X}, Y)$
   - Joint Adaptation Network (JAN)
   - Conditional Domain Adversarial Network (CDAN)
4. Evaluation

Deep Adaptation Network (DAN)[1]



Deep adaptation: match distributions in multiple domain-specific layers

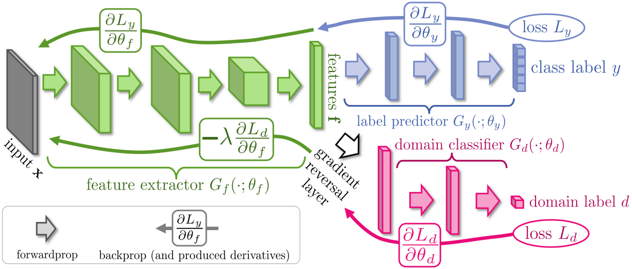
Optimal matching: maximize two-sample test power by multiple kernels

$$d_k^2(P, Q) \triangleq \left\| \mathbf{E}_P[\phi(\mathbf{x}^s)] - \mathbf{E}_Q\left[\phi\left(\mathbf{x}^t\right)\right] \right\|_{\mathcal{H}_k}^2 \quad (1)$$

$$\min_{\theta \in \Theta} \max_{k \in \mathcal{K}} \frac{1}{n_a} \sum_{i=1}^{n_a} J\left(\theta\left(\mathbf{x}_i^a\right), y_i^a\right) + \lambda \sum_{\ell=l_1}^{l_2} d_k^2\left(\mathcal{D}_s^\ell, \mathcal{D}_t^\ell\right) \quad (2)$$

[1] *Long et al. Learning Transferable Features with Deep Adaptation Networks. ICML '15.*

Domain Adversarial Neural Network (DANN)[2]



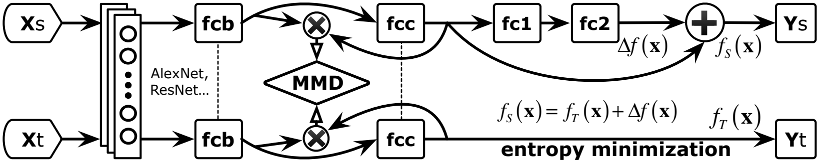
Adversarial adaptation: learning features indistinguishable across domains

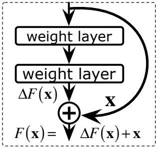
$$E(\theta_f, \theta_y, \theta_d) = \sum_{\mathbf{x}_i \in \mathcal{D}_s} L_y(G_y(G_f(\mathbf{x}_i)), y_i) - \lambda \sum_{\mathbf{x}_i \in \mathcal{D}_s \cup \mathcal{D}_t} L_d(G_d(G_f(\mathbf{x}_i)), d_i) \quad (3)$$

$$(\hat{\theta}_f, \hat{\theta}_y) = \arg\min_{\theta_f, \theta_y} E(\theta_f, \theta_y, \theta_d) \quad (\hat{\theta}_d) = \arg\max_{\theta_d} E(\theta_f, \theta_y, \theta_d) \quad (4)$$

[2] *Ganin et al. Domain Adversarial Training of Neural Networks. JMLR '16.*

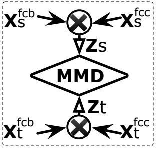
Residual Transfer Network (RTN)[3]





Classifier Adaptation

$$\min_{f_S=f_T+\Delta f} \frac{1}{n_s}\sum_{i=1}^{n_s} L\left(f_s\left(\mathbf{x}_i^s\right), y_i^s\right) + \frac{\gamma}{n_t}\sum_{i=1}^{n_t} H\left(f_t\left(\mathbf{x}_i^t\right)\right) + \lambda\, D_{\mathcal{L}}\left(\mathcal{D}_s, \mathcal{D}_t\right),$$



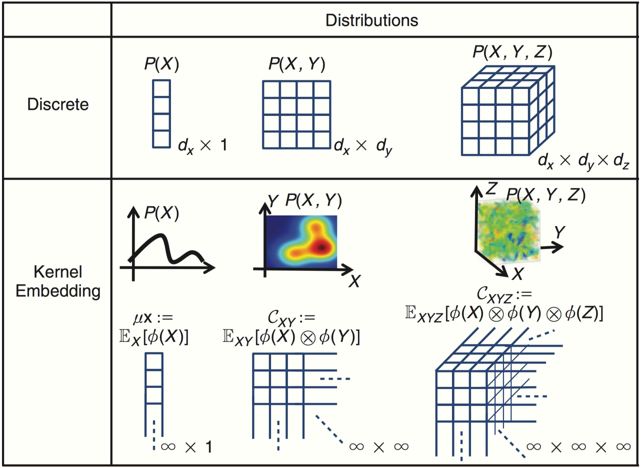
Feature Adaptation

[3] *Long et al. Unsupervised Domain Adaptation with Residual Transfer Networks. NIPS '16.*

Outline

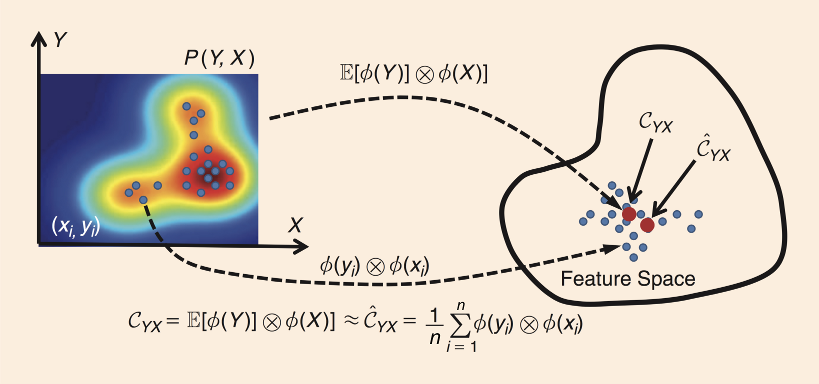
1. Deep Transfer Learning
2. Problem 1: $P(\mathbf{X}) \neq Q(\mathbf{X})$
3. **Problem 2: $P(\mathbf{X}, Y) \neq Q(\mathbf{X}, Y)$**
   - Joint Adaptation Network (JAN)
   - Conditional Domain Adversarial Network (CDAN)
4. Evaluation

Kernel Embedding of Distributions



Le Song et al. Kernel Embeddings of Conditional Distributions. IEEE, 2013.

Kernel Embedding of Joint Distributions



$$\mathcal{C}_{\mathbf{X}^{1:m}}(P) \triangleq \mathbb{E}_{\mathbf{X}^{1:m}}\left[\otimes_{\ell=1}^{m}\phi^{\ell}(\mathbf{X}^{\ell})\right] \approx \widehat{\mathcal{C}}_{\mathbf{X}^{1:m}} = \frac{1}{n}\sum_{i=1}^{n}\otimes_{\ell=1}^{m}\phi^{\ell}(\mathbf{x}_i^{\ell}) \qquad (5)$$

Le Song et al. Kernel Embeddings of Conditional Distributions. IEEE, 2013.

Joint Maximum Mean Discrepancy (JMMD)

Distance between *embeddings* of $P(\mathbf{Z}^{s1}, \ldots, \mathbf{Z}^{s|\mathcal{L}|})$ and $Q(\mathbf{Z}^{t1}, \ldots, \mathbf{Z}^{t|\mathcal{L}|})$

$$D_{\mathcal{L}}(P, Q) \triangleq \left\| \mathcal{C}_{\mathbf{Z}^{s,1:|\mathcal{L}|}}(P) - \mathcal{C}_{\mathbf{Z}^{t,1:|\mathcal{L}|}}(Q) \right\|^2_{\otimes_{\ell=1}^{|\mathcal{L}|} \mathcal{H}^{\ell}}. \tag{6}$$

$$\begin{aligned} \widehat{D}_{\mathcal{L}}(P, Q) = & \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} \prod_{\ell \in \mathcal{L}} k^{\ell}\left(\mathbf{z}_i^{s\ell}, \mathbf{z}_j^{s\ell}\right) \\ & + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} \prod_{\ell \in \mathcal{L}} k^{\ell}\left(\mathbf{z}_i^{t\ell}, \mathbf{z}_j^{t\ell}\right) \\ & - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \prod_{\ell \in \mathcal{L}} k^{\ell}\left(\mathbf{z}_i^{s\ell}, \mathbf{z}_j^{t\ell}\right). \end{aligned} \tag{7}$$

Theorem (Two-Sample Test (Gretton et al. 2012))

- $P = Q$ *if and only if* $\widehat{D}_{\mathcal{L}}(P, Q) = 0$ *(In practice,* $\widehat{D}_{\mathcal{L}}(P, Q) < \varepsilon$*)*

How to Understand JMMD?

- Set last-layer features $\mathbf{Z} = \mathbf{Z}^{L-1}$, classifier predictions $\mathbf{Y} = \mathbf{Z}^{L} \in \mathbb{R}^{C}$
- We can understand JMMD$(\mathbf{Z}, \mathbf{Y})$ by simplifying it to linear kernel
- This interpretation assumes classifier predictions $\mathbf{Y}$ be one-hot vector

$$\begin{aligned} \widehat{D}_{\mathcal{L}}(P, Q) &\triangleq \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{z}_i^s \otimes \mathbf{y}_i^s - \frac{1}{n_t} \sum_{j=1}^{n_t} \mathbf{z}_j^t \otimes \mathbf{y}_j^t \right\|^2 \\ &= \sum_{c=1}^{C} \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} y_{i,c}^s \mathbf{z}_i^s - \frac{1}{n_t} \sum_{j=1}^{n_t} y_{j,c}^t \mathbf{z}_j^t \right\|^2 \\ &\approx \sum_{c=1}^{C} \widehat{D}\left(P_{Z|y=c}, Q_{Z|y=c}\right) \end{aligned} \tag{8}$$

- Equivalent to matching P and Q conditioned on each class
- JMMD process with continuous softmax activations (probability)

Connection to Wasserstein-GAN (WGAN)

Different function spaces, and different powers in comparing distributions

- Wasserstein distance

$$D_W(P, Q) \triangleq \sup_{\|\phi\|_L \leq 1} \left(\mathbb{E}_{\mathbf{Z}^s}\left[\phi\left(\mathbf{Z}^s\right)\right] - \mathbb{E}_{\mathbf{Z}^t}\left[\phi\left(\mathbf{X}^t\right)\right]\right) \quad (9)$$

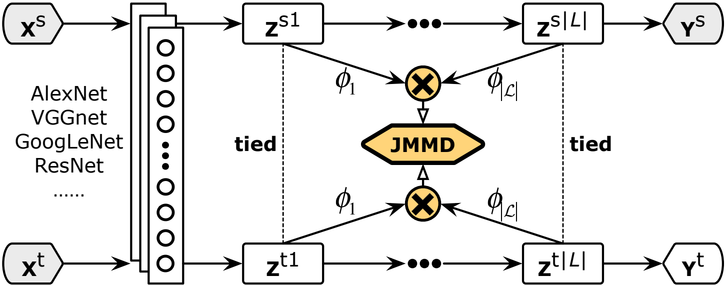
- MMD

$$D_{\mathcal{H}}(P, Q) \triangleq \sup_{\|\phi\|_{\mathcal{H}} \leq 1} \left(\mathbb{E}_{\mathbf{Z}^s}\left[\phi\left(\mathbf{Z}^s\right)\right] - \mathbb{E}_{\mathbf{Z}^t}\left[\phi\left(\mathbf{Z}^t\right)\right]\right) \quad (10)$$

- Joint MMD (JMMD)

$$D_{\mathcal{L}}(P, Q) \triangleq \sup_{\left\|\phi^\ell\right\|_{\mathcal{H}} \leqslant 1} \left(\mathbb{E}_{\mathbf{Z}^s}\left[\otimes_{\ell=1}^{|\mathcal{L}|} \phi^\ell(\mathbf{Z}^{s\ell})\right] - \mathbb{E}_{\mathbf{Z}^t}\left[\otimes_{\ell=1}^{|\mathcal{L}|}(\mathbf{Z}^{t\ell})\right]\right) \quad (11)$$

Joint Adaptation Network (JAN)[4]



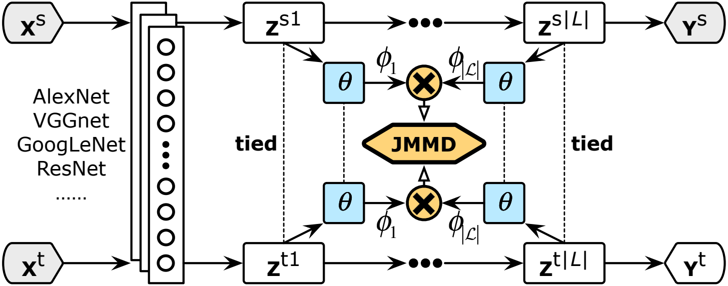
Joint adaptation: match joint distributions of multiple task-specific layers

$$\min_f \frac{1}{n_s} \sum_{i=1}^{n_s} J\left(f\left(\mathbf{x}_i^s\right), \mathbf{y}_i^s\right) + \lambda \widehat{D}_{\mathcal{L}}\left(P, Q\right) \tag{12}$$

$$D_{\mathcal{L}}\left(P, Q\right) \triangleq \left\| \mathcal{C}_{\mathbf{Z}^{s,1:|\mathcal{L}|}}\left(P\right) - \mathcal{C}_{\mathbf{Z}^{t,1:|\mathcal{L}|}}\left(Q\right) \right\|^2_{\otimes_{\ell=1}^{|\mathcal{L}|}\mathcal{H}^\ell} \tag{13}$$

[4] *Long et al. Deep Transfer Learning with Joint Adaptation Networks. ICML '17.*

Adversarial Joint Adaptation Network (JAN-A)



Optimal matching: maximize JMMD as semi-parametric domain adversary

$$\min_f \max_\theta \frac{1}{n_s} \sum_{i=1}^{n_s} J\left(f\left(\mathbf{x}_i^s\right), \mathbf{y}_i^s\right) + \lambda \widehat{D}_{\mathcal{L}}\left(P, Q; \theta\right) \tag{14}$$

$$\widehat{D}_{\mathcal{L}}\left(P, Q; \theta\right) = \frac{2}{n} \sum_{i=1}^{n/2} d\left(\left\{\theta^\ell\left(\mathbf{z}_{2i-1}^{s\ell}, \mathbf{z}_{2i}^{s\ell}, \mathbf{z}_{2i-1}^{t\ell}, \mathbf{z}_{2i}^{t\ell}\right)\right\}_{\ell \in \mathcal{L}}\right) \tag{15}$$

Learning Algorithm

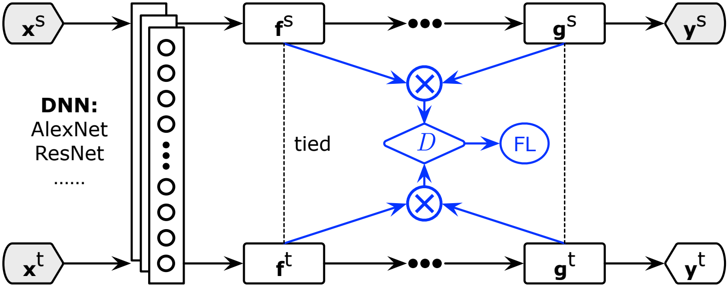
Linear-Time $O(n)$ Algorithm of JMMD (Streaming Algorithm)

$$\begin{aligned} \widehat{D}_{\mathcal{L}}(P, Q) &= \frac{2}{n} \sum_{i=1}^{n/2} \left(\prod_{\ell \in \mathcal{L}} k^{\ell}(\mathbf{z}_{2i-1}^{s\ell}, \mathbf{z}_{2i}^{s\ell}) + \prod_{\ell \in \mathcal{L}} k^{\ell}(\mathbf{z}_{2i-1}^{t\ell}, \mathbf{z}_{2i}^{t\ell}) \right) \\ &- \frac{2}{n} \sum_{i=1}^{n/2} \left(\prod_{\ell \in \mathcal{L}} k^{\ell}(\mathbf{z}_{2i-1}^{s\ell}, \mathbf{z}_{2i}^{t\ell}) + \prod_{\ell \in \mathcal{L}} k^{\ell}(\mathbf{z}_{2i-1}^{t\ell}, \mathbf{z}_{2i}^{s\ell}) \right) \\ &= \frac{2}{n} \sum_{i=1}^{n/2} d\left(\{\mathbf{z}_{2i-1}^{s\ell}, \mathbf{z}_{2i}^{s\ell}, \mathbf{z}_{2i-1}^{t\ell}, \mathbf{z}_{2i}^{t\ell}\}_{\ell \in \mathcal{L}} \right) \end{aligned} \tag{16}$$

SGD: for each layer ℓ and for each quad-tuple $\left(\mathbf{z}_{2i-1}^{s\ell}, \mathbf{z}_{2i}^{s\ell}, \mathbf{z}_{2i-1}^{t\ell}, \mathbf{z}_{2i}^{t\ell}\right)$

$$\nabla_{W^{\ell}} = \frac{\partial J\left(\mathbf{z}_{2i-1}^{s}, \mathbf{z}_{2i}^{s}, y_{2i-1}^{s}, y_{2i}^{s}\right)}{\partial W^{\ell}} + \lambda \frac{\partial d\left(\{\mathbf{z}_{2i-1}^{s\ell}, \mathbf{z}_{2i}^{s\ell}, \mathbf{z}_{2i-1}^{t\ell}, \mathbf{z}_{2i}^{t\ell}\}_{\ell \in \mathcal{L}}\right)}{\partial W^{\ell}} \tag{17}$$

Multilinear Conditioning[5]

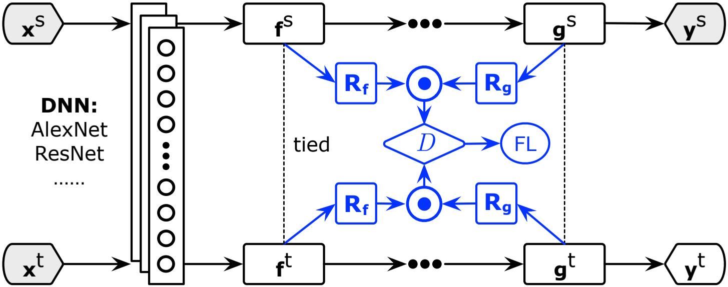


$$\min_G E(G) - \lambda E(D, G)$$
$$\min_D E(D, G) \tag{18}$$

$$E(D, G) = -\frac{1}{n_s}\sum_{i=1}^{n_s} \log\left(D\left(\mathbf{f}_i^s \otimes \mathbf{g}_i^s\right)\right) - \frac{1}{n_t}\sum_{j=1}^{n_t} \log\left(1 - D\left(\mathbf{f}_j^t \otimes \mathbf{g}_j^t\right)\right) \tag{19}$$

[5] *Long et al. Conditional Adversarial Domain Adaptation. arXiv '17.*

Randomized Multilinear Conditioning

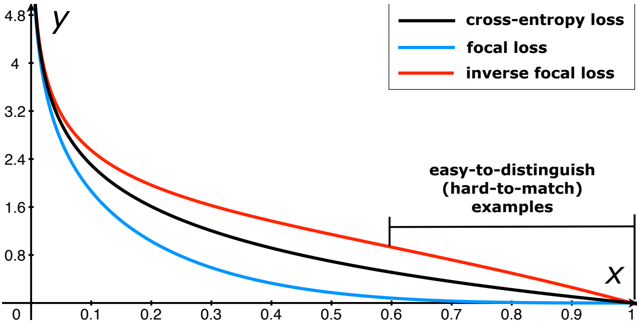


$$T_{\otimes}(\mathbf{f}, \mathbf{g}) = \mathbf{f} \otimes \mathbf{g} \tag{20}$$

$$T_{\odot}(\mathbf{f}, \mathbf{g}) = \frac{1}{\sqrt{d}} (\mathbf{R_f f}) \odot (\mathbf{R_g g}) \tag{21}$$

$$\phi(\mathbf{h}) = \begin{cases} T_{\otimes}(\mathbf{f}, \mathbf{g}) & \text{if } d_f \times d_g \leqslant 4096 \\ T_{\odot}(\mathbf{f}, \mathbf{g}) & \text{otherwise} \end{cases} \tag{22}$$

Inverse Focal Discriminator



$$E(D, G) = -\frac{1}{n_s}\sum_{i=1}^{n_s} \exp\left(D\left(\phi\left(\mathbf{h}_i^s\right)\right)\right) \log\left(D\left(\phi\left(\mathbf{h}_i^s\right)\right)\right) - \frac{1}{n_t}\sum_{j=1}^{n_t} \exp\left(1 - D\left(\phi\left(\mathbf{h}_j^t\right)\right)\right) \log\left(1 - D\left(\phi\left(\mathbf{h}_j^t\right)\right)\right) \tag{23}$$

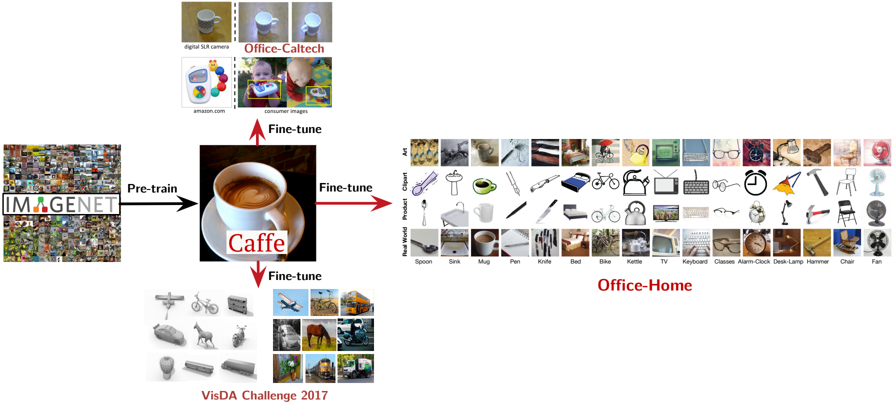
Optimization Problem

$$
\begin{aligned}
\min_G \quad & \frac{1}{n_s} \sum_{i=1}^{n_s} L\left(G\left(\mathbf{x}_i^s\right), \mathbf{y}_i^s\right) \\
& + \frac{\lambda}{n_s} \sum_{i=1}^{n_s} \exp\left(D\left(\phi\left(\mathbf{h}_i^s\right)\right)\right) \log\left(D\left(\phi\left(\mathbf{h}_i^s\right)\right)\right) \\
& + \frac{\lambda}{n_t} \sum_{j=1}^{n_t} \exp\left(1 - D\left(\phi\left(\mathbf{h}_j^t\right)\right)\right) \log\left(1 - D\left(\phi\left(\mathbf{h}_j^t\right)\right)\right) \\
\max_D \quad & \frac{1}{n_s} \sum_{i=1}^{n_s} \exp\left(D\left(\phi\left(\mathbf{h}_i^s\right)\right)\right) \log\left(D\left(\phi\left(\mathbf{h}_i^s\right)\right)\right) \\
& + \frac{1}{n_t} \sum_{j=1}^{n_t} \exp\left(1 - D\left(\phi\left(\mathbf{h}_j^t\right)\right)\right) \log\left(1 - D\left(\phi\left(\mathbf{h}_j^t\right)\right)\right)
\end{aligned}
\tag{24}
$$

Outline

1. Deep Transfer Learning
2. Problem 1: $P(\mathbf{X}) \neq Q(\mathbf{X})$
3. Problem 2: $P(\mathbf{X}, Y) \neq Q(\mathbf{X}, Y)$
   - Joint Adaptation Network (JAN)
   - Conditional Domain Adversarial Network (CDAN)
4. **Evaluation**

Datasets

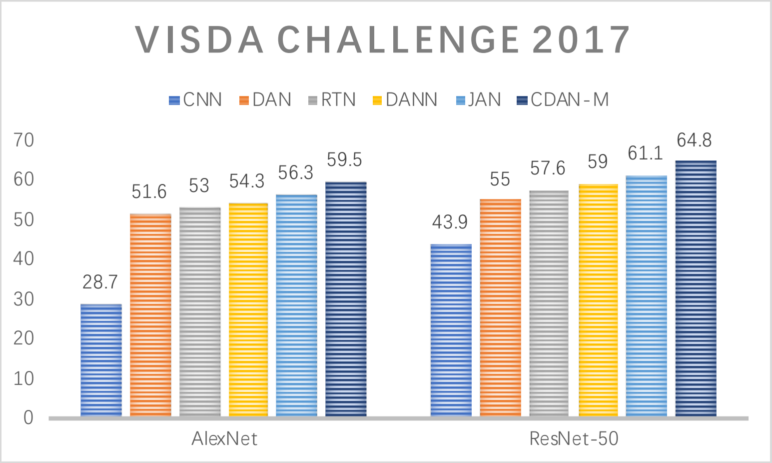

digital SLR camera
Office-Caltech
amazon.com
consumer images
Fine-tune
IMAGENET
Pre-train
Caffe
Fine-tune
Art
Clipart
Product
Real World
Spoon
Sink
Mug
Pen
Knife
Bed
Bike
Kettle
TV
Keyboard
Classes
Alarm-Clock
Desk-Lamp
Hammer
Chair
Fan
Office-Home
Fine-tune
VisDA Challenge 2017

Results

Table: Accuracy (%) on *Office-31* for unsupervised domain adaptation

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
AlexNet	61.6±0.5	95.4±0.3	99.0±0.2	63.8±0.5	51.1±0.6	49.8±0.4	70.1
TCA	61.0±0.0	93.2±0.0	95.2±0.0	60.8±0.0	51.6±0.0	50.9±0.0	68.8
GFK	60.4±0.0	95.6±0.0	95.0±0.0	60.6±0.0	52.4±0.0	48.1±0.0	68.7
DAN	68.5±0.5	96.0±0.3	99.0±0.3	67.0±0.4	54.0±0.5	53.1±0.5	72.9
RTN	73.3±0.3	96.8±0.2	99.6±0.1	71.0±0.2	50.5±0.3	51.0±0.1	73.7
DANN	73.0±0.5	96.4±0.3	99.2±0.3	72.3±0.3	53.4±0.4	51.2±0.5	74.3
ADDA	73.5±0.6	96.2±0.4	98.8±0.4	71.6±0.4	54.6±0.5	53.5±0.6	74.7
JAN	74.9±0.3	96.6±0.2	99.5±0.2	71.8±0.2	**58.3**±0.3	55.0±0.4	76.0
CDAN-RM	**77.9**±0.3	96.9±0.2	**100.0**±.0	**74.6**±0.2	55.1±0.3	**57.5**±0.4	**77.0**
CDAN-M	77.6±0.2	**97.2**±0.1	**100.0**±.0	73.0±0.1	57.3±0.2	56.1±0.3	76.9
ResNet-50	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
TCA	72.7±0.0	96.7±0.0	99.6±0.0	74.1±0.0	61.7±0.0	60.9±0.0	77.6
GFK	72.8±0.0	95.0±0.0	98.2±0.0	74.5±0.0	63.4±0.0	61.0±0.0	77.5
DAN	80.5±0.4	97.1±0.2	99.6±0.1	78.6±0.2	63.6±0.3	62.8±0.2	80.4
RTN	84.5±0.2	96.8±0.1	99.4±0.1	77.5±0.3	66.2±0.2	64.8±0.3	81.6
DANN	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
ADDA	86.2±0.5	96.2±0.3	98.4±0.3	77.8±0.3	69.5±0.4	68.9±0.5	82.9
JAN	85.4±0.3	97.4±0.2	99.8±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
CDAN-RM	93.0±0.2	98.4±0.2	**100.0**±.0	89.2±0.3	70.2±0.4	69.4±0.4	86.7
CDAN-M	**93.1**±0.1	**98.6**±0.1	**100.0**±.0	**93.4**±0.2	**71.0**±0.3	**70.3**±0.3	**87.7**

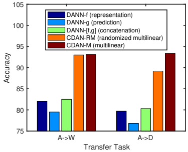
Results


VISDA CHALLENGE 2017
CNN
DAN
RTN
DANN
JAN
CDAN-M
70
60
50
40
30
20
10
0
28.7
51.6
53
54.3
56.3
59.5
43.9
55
57.6
59
61.1
64.8
AlexNet
ResNet-50

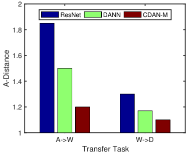
Analysis

Table: Accuracy (%) of CDAN variants for unsupervised domain adaptation

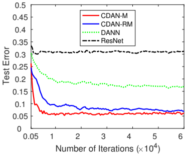
Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
CDAN-RM (bernoulli)	87.8±0.3	97.2±0.3	99.4±0.1	85.1±0.4	**70.9**±0.5	**71.7**±0.5	85.3
CDAN-RM (gaussian)	88.0±0.1	97.4±0.1	99.7±0.1	86.4±0.2	70.6±0.3	71.4±0.3	85.6
CDAN-RM (uniform)	**93.0**±0.2	**98.4**±0.2	**100.0**±.0	**89.2**±0.3	70.2±0.4	69.4±0.4	**86.7**
CDAN-M (no focal loss)	91.7±0.2	98.3±0.1	**100.0**±.0	92.5±0.2	70.0±0.2	67.8±0.2	86.8
CDAN-M (focal loss)	**93.1**±0.1	**98.6**±0.1	**100.0**±.0	**93.4**±0.2	**71.0**±0.3	**70.3**±0.3	**87.7**



(a) Conditioning



(b) Discrepancy



(c) Convergence

Figure: Analysis of CDAN: (a) Conditioning, (b) Discrepancy, (c) Convergence.

Open Problems

- Heterogeneous Transfer Learning

$$\mathbf{X}_s \neq \mathbf{X}_t \vee \mathbf{Y}_s \neq \mathbf{Y}_t$$

- Pixel-Level Transfer Learning

$$P(\mathbf{X}) \neq Q(\mathbf{X}) \vee P(\mathbf{Z}) \neq Q(\mathbf{Z})$$

- Learning Transferable Architectures

- Code available at: `https://github.com/thuml/Xlearn`