Learning Transferable Features with Deep Adaptation Networks

Mingsheng Long[1,2], Yue Cao[1], Jianmin Wang[1], and Michael I. Jordan[2]

[1]School of Software, Tsinghua University, China [2]AMPLab, Department of EECS, UC Berkeley

ICML Lille

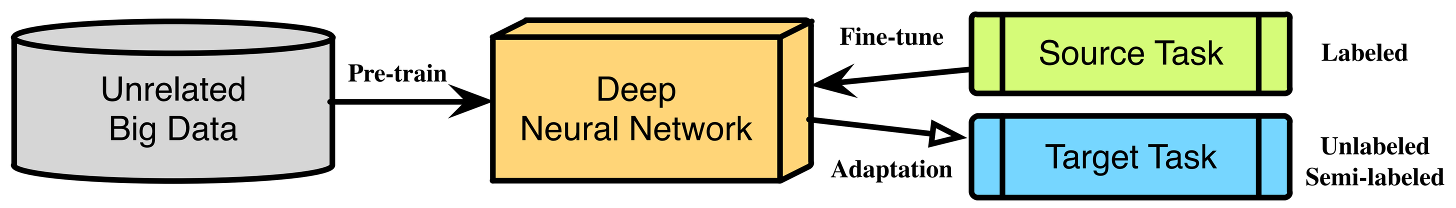
International Conference on Machine Learning

Summary

- A deep adaptation network for learning transferable features
- Two important improvements:
  - Deep adaptation of multiple task-specific layers (including output)
  - Optimal adaptation using multiple kernel two-sample matching
- A brief analysis of learning bound for the proposed deep network
- Outlook
  - Principled way of deciding the boundary of generality and specificity
  - Deeper adaptation of convolutional layers to enhance transferability
  - Fine-grained adaptation using structural embedding of distributions

Deep Learning for Domain Adaptation

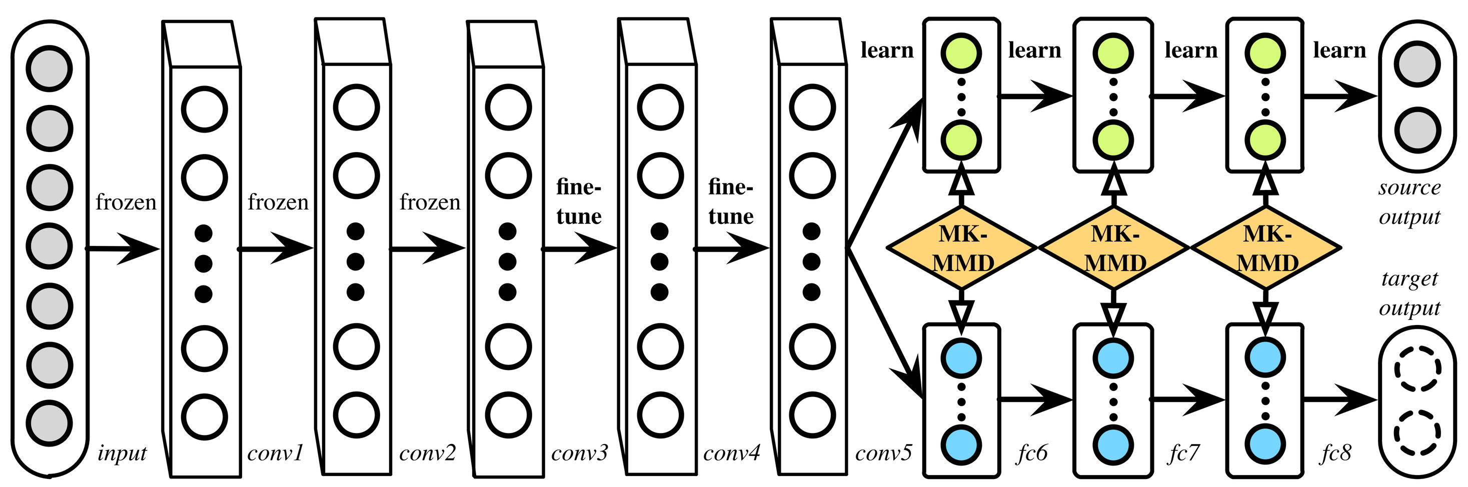
- None or very limited supervision in the target task (new domain)
  - Target classifier cannot be reliably trained due to over-fitting
  - Fine-tuning is impossible as it requires substantial supervision
- Leverage supervision (same categories) from related source task
  - Deep networks can learn more transferable features for adaptation
  - Transferability of features decreases as the task discrepancy increases
- Hard to find big source task for learning deep features from scratch
  - Fine-tune from deep networks pre-trained on unrelated big dataset
  - Transferring features from distant tasks is better than random features



Deep Adaptation Network (DAN)

Key Assumptions (AlexNet)

- Conv-layers learn general features: safely transferable
  - Safely freeze *conv1-conv3* & fine-tune *conv4-conv5*
- FC-layers fit domain-specific variations: NOT transferable
  - Deeply adapt *fc6-fc8* using optimized two-sample matching



Deep adaptation: match distributions in multi-layers, including output

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

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

Multiple Kernel Maximum Mean Discrepancy (MK-MMD)

$\triangleq$ RKHS distance between *kernel embeddings* of distributions p and q

$$d_k^2(p,q) \triangleq \left\|\mathbf{E}_p\left[\phi\left(\mathbf{x}^s\right)\right] - \mathbf{E}_q\left[\phi\left(\mathbf{x}^t\right)\right]\right\|_{\mathcal{H}_k}^2, \tag{2}$$

$k\left(\mathbf{x}^s,\mathbf{x}^t\right) = \left\langle \phi\left(\mathbf{x}^s\right), \phi\left(\mathbf{x}^t\right)\right\rangle$ is a convex combination of m PSD kernels

$$\mathcal{K} \triangleq \left\{k = \sum_{u=1}^m \beta_u k_u : \sum_{u=1}^m \beta_u = 1, \beta_u \geqslant 0, \forall u\right\}. \tag{3}$$

Two-Sample Test (Gretton et al. 2012)

- $p = q$ if and only if $d_k^2(p,q) = 0$ (In practice, $d_k^2(p,q) < \varepsilon$)
- $\max\limits_{k\in\mathcal{K}} d_k^2(p,q)\,\sigma_k^{-2} \Leftrightarrow$ min Type II Error ($d_k^2(p,q) < \varepsilon$ when $p \neq q$)

Learning CNN

Linear-Time Algorithm of MK-MMD (Streaming Algorithm)

$O(n^2)$: $d_k^2(p,q) = \mathbf{E}_{\mathbf{x}^s\mathbf{x}'^s} k(\mathbf{x}^s,\mathbf{x}'^s) + \mathbf{E}_{\mathbf{x}^t\mathbf{x}'^t} k(\mathbf{x}^t,\mathbf{x}'^t) - 2\mathbf{E}_{\mathbf{x}^s\mathbf{x}^t} k(\mathbf{x}^s,\mathbf{x}^t)$

$O(n)$: $d_k^2(p,q) = \frac{2}{n_s}\sum_{i=1}^{n_s/2} g_k(\mathbf{z}_i) \to$ linear-time unbiased estimate

- Quad-tuple $\mathbf{z}_i \triangleq \left(\mathbf{x}_{2i-1}^s, \mathbf{x}_{2i}^s, \mathbf{x}_{2i-1}^t, \mathbf{x}_{2i}^t\right)$
- $g_k(\mathbf{z}_i) \triangleq k(\mathbf{x}_{2i-1}^s,\mathbf{x}_{2i}^s) + k(\mathbf{x}_{2i-1}^t,\mathbf{x}_{2i}^t) - k(\mathbf{x}_{2i-1}^s,\mathbf{x}_{2i}^t) - k(\mathbf{x}_{2i}^s,\mathbf{x}_{2i-1}^t)$

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

$$\nabla_{\Theta^\ell} = \frac{\partial J(\mathbf{z}_i)}{\partial \Theta^\ell} + \lambda \frac{\partial g_k\left(\mathbf{z}_i^\ell\right)}{\partial \Theta^\ell} \tag{4}$$

Learning Kernel

Learning optimal kernel $k = \sum_{u=1}^m \beta_u k_u$ by minimizing Type II error

$$\max_{k\in\mathcal{K}} d_k^2\left(\mathcal{D}_s^\ell, \mathcal{D}_t^\ell\right)\sigma_k^{-2}, \tag{5}$$

where $\sigma_k^2 = \mathbf{E}_\mathbf{z} g_k^2(\mathbf{z}) - \left[\mathbf{E}_\mathbf{z} g_k(\mathbf{z})\right]^2$ is the estimation variance.

Quadratic Program (QP), scaling linearly to sample size: $O(m^2 n + m^3)$

$$\min_{\mathbf{d}^{\mathrm{T}}\beta=1,\beta\geqslant\mathbf{0}} \beta^{\mathrm{T}}\left(\mathbf{Q} + \varepsilon\mathbf{I}\right)\beta, \tag{6}$$

where $\mathbf{d} = (d_1, d_2, \ldots, d_m)^{\mathrm{T}}$, and each d_u is MMD using base kernel k_u.

Theorem (Adaptation Bound)

Let $\theta \in \mathcal{H}$ be a hypothesis, $\epsilon_s(\theta)$ and $\epsilon_t(\theta)$ be the expected risks of source and target respectively, then

$$\epsilon_t(\theta) \leqslant \epsilon_s(\theta) + 2d_k(p,q) + C, \tag{7}$$

where C is a constant for the complexity of hypothesis space and the risk of an ideal hypothesis for both domains.

VC-Dimension of neural nets with linear-threshold gates $O(W \log W)$.

Experiment Setup

- **Datasets:** pre-trained on ImageNet, fined-tuned on Office&Caltech
- **Tasks:** 12 adaptation tasks → An unbiased look at dataset bias
- **Baselines:** TCA, GFK, CNN, LapCNN, DDC (Tzeng et al. 2014)
- **Protocols:** unsupervised vs semi-supervised, full-set vs sub-sample
- **Parameter selection:** cross-validation by jointly assessing
  - validation errors of source classifier and two-sample classifier (MK-MMD)

(Fei-Fei et al. 2012) (Jia et al. 2014) (Saenko et al. 2010)

Results and Discussion

Learn transferable features by deep adaptation and optimal matching

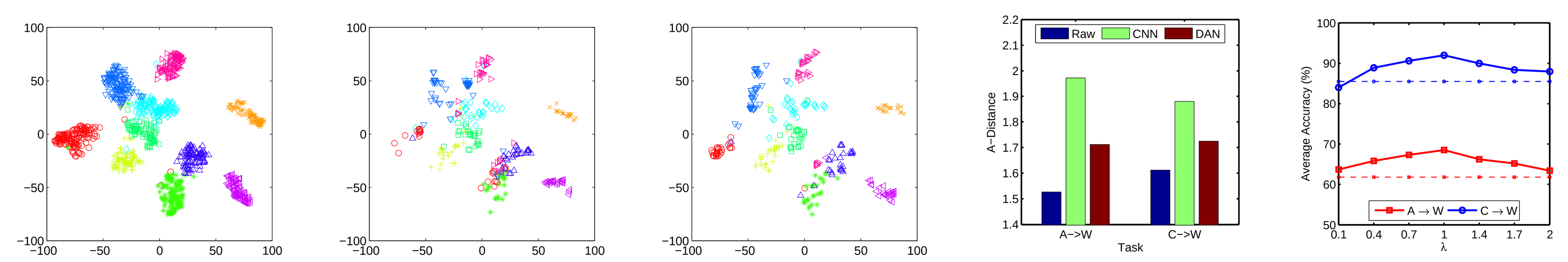
- Deep adaptation of multiple domain-specific layers (DAN) vs. shallow adaptation of one hard-to-tweak layer (DDC)
- Two samples can be matched better by MK-MMD vs. SK-MMD
- Semi-supervised adaptation: source vs. target supervision?
  - Limited target supervision is prone to over-fitting the target task
  - Source supervision can provide strong but inaccurate inductive bias
  - Via source inductive bias, target supervision is much more powerful
  - Two-sample matching is more effective for bridging dissimilar tasks

Paradigm	Method	A → W	D → W	W → D	Average
Un-supervised	DDC	59.4±0.8	92.5±0.3	91.7±0.8	81.2
	DAN	**66.0**± 0.4	**93.5**±0.2	**95.3**±0.3	**84.9**
Semi-Supervised	DDC	84.1±0.6	95.4±0.4	96.3±0.3	91.9
	DAN	**85.7**±0.3	**97.2**±0.2	**96.4**±0.2	**93.1**

Empirical Analysis

How generalization performance relates to two-sample discrepancy?

- $\hat{d}_\mathcal{A}$ on CNN & DAN features is larger than $\hat{d}_\mathcal{A}$ on Raw features
  - Deep features are salient for both category & domain discrimination
- $\hat{d}_\mathcal{A}$ on DAN feature is much smaller than $\hat{d}_\mathcal{A}$ on CNN feature
  - Domain adaptation can be boosted by minimizing domain discrepancy



(a) CNN on Source (b) DDC on Target (c) DAN on Target (d) $\mathcal{A}$-distance (e) Accuracy vs. λ

School of Software - Tsinghua University - China AMPLab - Department of EECS - UC Berkeley Mail: mingsheng@tsinghua.edu.cn WWW: ise.thss.tsinghua.edu.cn/∼mlong Code Available Soon!