Unsupervised Domain Adaptation with Residual Transfer Networks

Mingsheng Long[1], Han Zhu[1], Jianmin Wang[1], and Michael I. Jordan[2]

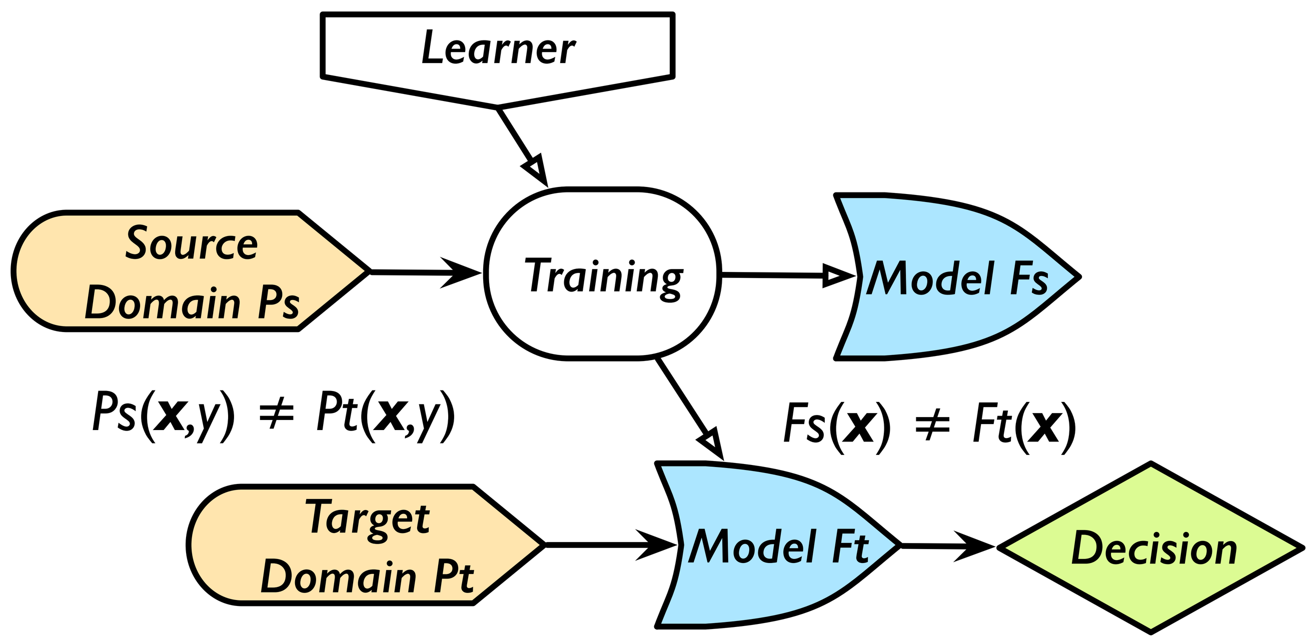
[1]School of Software, Tsinghua University, China [2]Department of EECS, UC Berkeley, USA

Summary

- A residual transfer network for unsupervised domain adaptation
- An end-to-end deep architecture for jointly learning
  - Transferable features with joint distribution adaptation
  - Adaptive classifiers with deep residual learning and entropy minimization
- Open problem
  - More efficient joint distribution adaptation with fast kernel approximation
  - Extending residual transfer network to semi-supervised domain adaptation

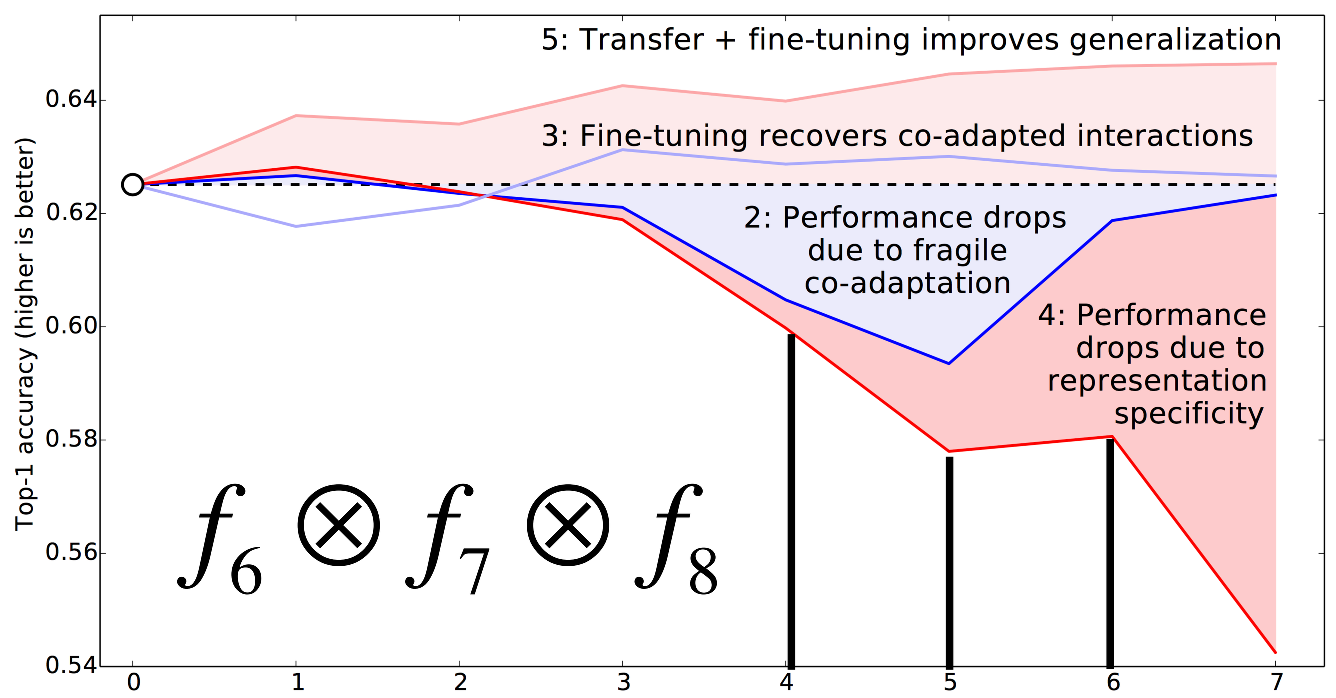
Unsupervised Domain Adaptation

- Source domain: $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ of n_s labeled examples
- Target domain: $\mathcal{D}_t = \{\mathbf{x}_j^t\}_{j=1}^{n_t}$ of n_t unlabeled examples
- Setting: different feature distributions $P_s(\mathbf{x}, y) \neq P_t(\mathbf{x}, y)$
- Challenge: different classification models $f_s(\mathbf{x}) \neq f_t(\mathbf{x})$
- Problem: bound the target risk $R_t(f_t) = \mathbb{E}_{(\mathbf{x},y)\sim P_t}[f_t(\mathbf{x}) \neq y]$
- Key: jointly learn transferable features and adaptive classifiers such that $P_s(\mathbf{x}, y) \approx P_t(\mathbf{x}, y)$ and $f_S(\mathbf{x}) = f_T(\mathbf{x}) + \Delta f(\mathbf{x})$



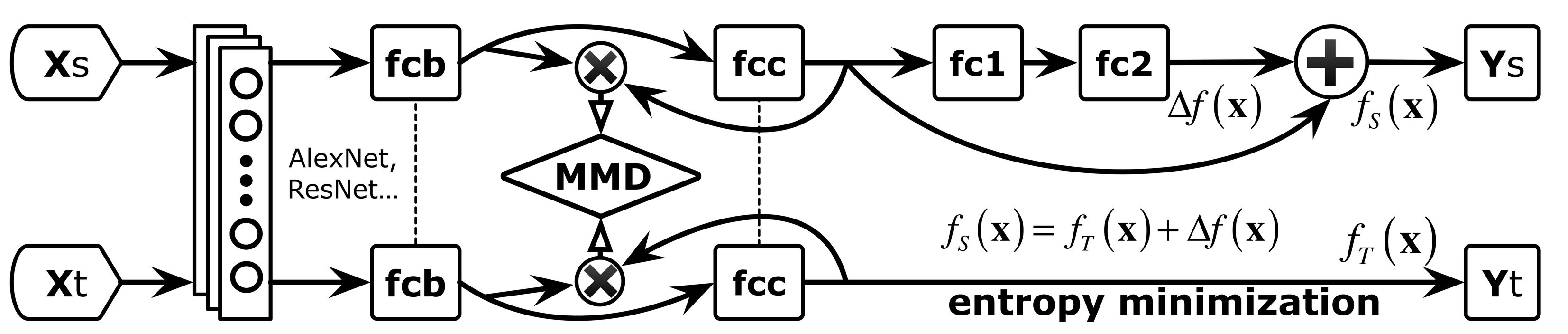
Deep Learning for Domain Adaptation

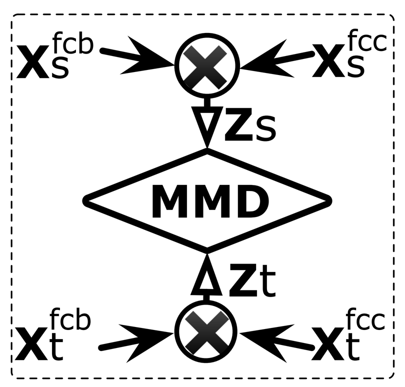
- Feature transferability decreases in multiple task-specific layers
- Feature transferability decreases as domain difference increases



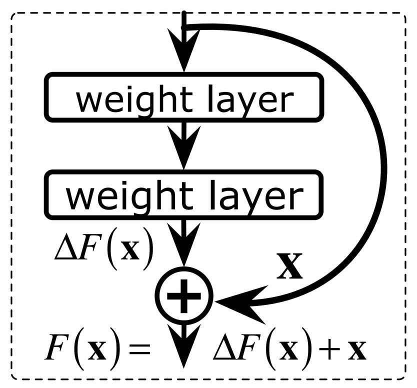
Residual Transfer Network (RTN)

- Goal: end-to-end learning of transferable features and classifiers





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



Feature Adaptation

- Modeling the joint distribution of features in multiple layers
- Feature fusion via tensor product: $\mathbf{z}_i^s \triangleq \otimes_{\ell\in\mathcal{L}} \mathbf{x}_i^{s\ell}, \mathbf{z}_j^t \triangleq \otimes_{\ell\in\mathcal{L}} \mathbf{x}_j^{t\ell}$
- Maximum Mean Discrepancy (MMD) to compare distributions P_s and P_t in RKHS: $D(P_s, P_t) \triangleq \left\| \mathbf{E}_{P_s}[\phi(\mathbf{x}^s)] - \mathbf{E}_{P_t}[\phi(\mathbf{x}^t)] \right\|_{\mathcal{H}}^2$
- Feature adaptation via MMD over fused features (tensor MMD)

$$\min_{f_s, f_t} D_{\mathcal{L}}(\mathcal{D}_s, \mathcal{D}_t) = \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} \frac{k(\mathbf{z}_i^s, \mathbf{z}_j^s)}{n_s^2} + \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} \frac{k(\mathbf{z}_i^t, \mathbf{z}_j^t)}{n_t^2} - 2 \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \frac{k(\mathbf{z}_i^s, \mathbf{z}_j^t)}{n_s n_t} \tag{2}$$

Classifier Adaptation

- Modeling cross-domain classifier shift by deep residual learning
- Build the connection across classifiers: $f_S(\mathbf{x}) = f_T(\mathbf{x}) + \Delta f(\mathbf{x})$
- Set residual block: $\mathbf{x} \triangleq f_T(\mathbf{x})$, $F(\mathbf{x}) \triangleq f_S(\mathbf{x})$, $\Delta F(\mathbf{x}) \triangleq \Delta f(\mathbf{x})$
- Probabilistic predictions: $f_s(\mathbf{x}) \triangleq \sigma(f_S(\mathbf{x}))$, $f_t(\mathbf{x}) \triangleq \sigma(f_T(\mathbf{x}))$
- Low-density separation of target data by entropy minimization

$$\min_{f_t} \frac{1}{n_t} \sum_{i=1}^{n_t} H\left(f_t(\mathbf{x}_i^t)\right) \tag{3}$$

- The class-conditional distribution $f_j^t(\mathbf{x}_i^t) = p(y_i^t = j | \mathbf{x}_i^t; f_t)$
- The entropy function $H(f_t(\mathbf{x}_i^t)) = -\sum_{j=1}^{c} f_j^t(\mathbf{x}_i^t) \log f_j^t(\mathbf{x}_i^t)$

Experimental Setup

- **Datasets:** Office-31 (31 classes), Office-Caltech (10 classes)
- **Tasks:** 6+12 transfer tasks → unbiased look at dataset bias
- **Methods:** TCA, GFK, AlexNet, DDC, DAN, RevGrad
- **Model Selection:** (1) Cross-Validation (CV) on source data, (2) CV on **A** → **W** (1 labeled point per class as validation set)

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

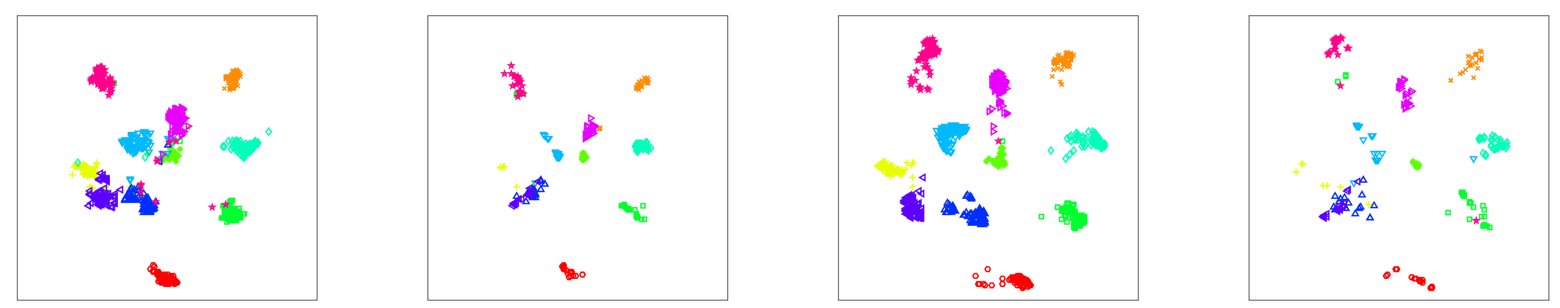
Results

- RTN can learn adaptive classifiers and transferable features
- Entropy minimization can make residual transfer more effective

Table: Accuracy on *Office-Caltech* dataset for unsupervised domain adaptation

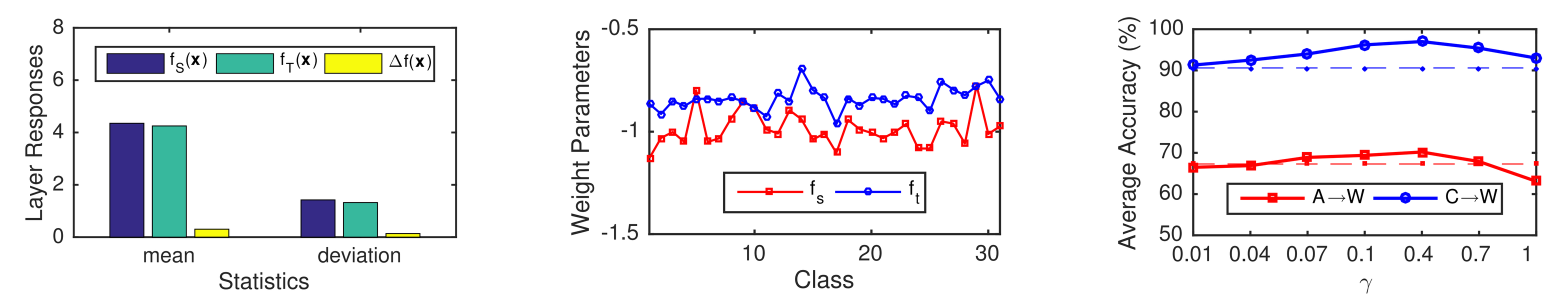
Method	A→W	D→W	W→D	A→D	D→A	W→A	A→C	W→C	D→C	C→A	C→W	C→D	Avg
TCA	84.4	96.9	99.4	82.8	90.4	85.6	81.2	75.5	79.6	92.1	88.1	87.9	87.0
GFK	89.5	97.0	98.1	86.0	89.8	88.5	76.2	77.1	77.9	90.7	78.0	77.1	85.5
AlexNet	79.5	97.7	**100.0**	87.4	87.1	83.8	83.0	73.0	79.0	91.9	83.7	87.1	86.1
DDC	83.1	98.1	**100.0**	88.4	89.0	84.9	83.5	73.4	79.2	91.9	85.4	88.8	87.1
DAN	91.8	98.5	**100.0**	91.7	90.0	92.1	84.1	81.2	80.3	92.0	90.6	89.3	90.1
RevGrad	91.4	99.0	**100.0**	92.1	**94.9**	**93.7**	85.6	86.5	84.3	93.2	93.6	89.8	91.9
RTN (mmd)	93.2	98.5	**100.0**	91.7	88.0	90.7	84.0	81.3	80.4	91.0	89.8	90.4	90.0
RTN (mmd+ent)	93.8	98.6	**100.0**	92.9	93.6	92.7	87.8	84.8	83.4	93.2	96.6	93.9	92.6
RTN (mmd+ent+res)	**95.2**	**99.2**	**100.0**	**95.5**	93.8	92.5	**88.1**	**86.6**	**84.6**	**93.7**	**96.9**	**94.2**	**93.4**

Discussion



(a) DAN: Ds=A (b) DAN: Dt=W (c) RTN: Ds=A (d) RTN: Dt=W

Figure: Prediction visualization: (a)-(b) t-SNE of DAN; (c)-(d) t-SNE of RTN.



(a) Layer Responses (b) Classifier Shift (c) Accuracy w.r.t. γ

Figure: Analysis: (a) layer responses; (b) classifier shift; (c) sensitivity of γ.

- Classifiers f_s & f_t trained on ground-truth data are very different

School of Software - Tsinghua University - China Department of EECS - UC Berkeley - USA Mail: `mingsheng@tsinghua.edu.cn` WWW: `http://ise.thss.tsinghua.edu.cn/~mlong`