Transferable Normalization: Towards Improving Transferability of Deep Neural Networks

Ximei Wang, Ying Jin, Mingsheng Long (✉), Jianmin Wang, and Michael I. Jordan♯

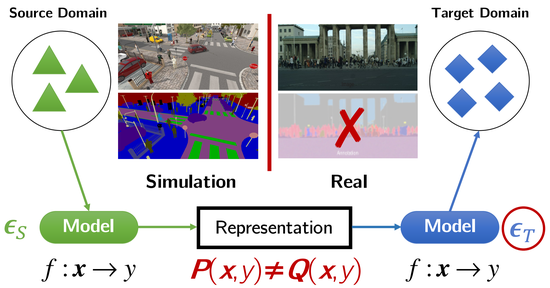
School of Software, BNRist, NEL-BDS, Research Center for Big Data, Tsinghua University, China, ♯University of California, Berkeley, USA

Summary

- TransNorm: a simple, general and end-to-end trainable normalization layer towards improving transferability of Deep Neural Networks (DNNs).
- TransNorm can be easily plugged into various DNNs and DA methods, without introducing any extra learnable parameters or hyper-parameters.
- Improves classification accuracies and accelerates convergence for mainstream DNN-based DA methods.
- Code available @ `github.com/thuml/TransNorm`

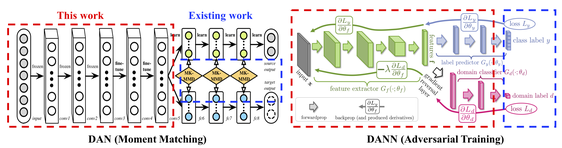
Domain Adaptation (DA)

- Deep learning across domains: $(P \neq Q)$
  - Independent and Differently Distributed (IDD) distributions



Mainstream Approaches to DA

- Existing works tackle domain adaptation by reducing the domain shift from the perspective of loss function designs.
- Rare attention has been paid to the other side of the coin, i.e. network design, an indispensable part of DA.



Batch Normalization (BN)

$$\mu^{(j)} = \frac{1}{m}\sum_{i=1}^{m} x_i^{(j)}, \quad \sigma^{2(j)} = \frac{1}{m}\sum_{i=1}^{m}(x_i^{(j)} - \mu^{(j)})^2 \tag{1}$$

$$\widehat{x}^{(j)} = \frac{x^{(j)} - \mu^{(j)}}{\sqrt{\sigma^{2(j)} + \epsilon}}, \quad y^{(j)} = \gamma^{(j)}\widehat{x}^{(j)} + \beta^{(j)} \tag{2}$$

Transferable Normalization (TransNorm)

- **Domain Specific Mean and Variance**

$$\widehat{x}_s = \frac{x_s - u_s}{\sqrt{\sigma_s^2 + \epsilon}}, \quad \widehat{x}_t = \frac{x_t - u_t}{\sqrt{\sigma_t^2 + \epsilon}} \tag{3}$$

- **Domain Sharing Gamma and Beta**

$$z_s = \gamma\widehat{x}_s + \beta, \quad z_t = \gamma\widehat{x}_t + \beta, \tag{4}$$

- **Domain Adaptive Alpha**
  - Calculating Distance

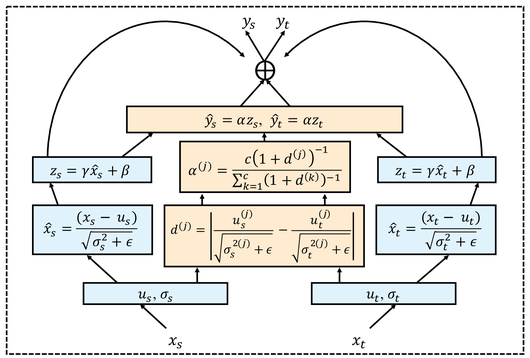
$$d^{(j)} = \left| \frac{\mu_s^{(j)}}{\sqrt{\sigma_s^{2(j)} + \epsilon}} - \frac{\mu_t^{(j)}}{\sqrt{\sigma_t^{2(j)} + \epsilon}} \right|, \quad j = 1, 2, ..., c, \tag{5}$$

  - Generating Probability

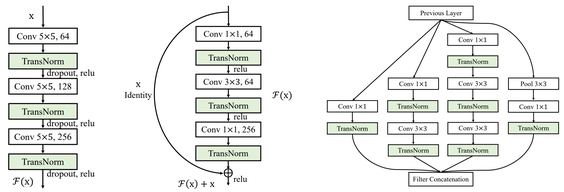
$$\alpha^{(j)} = \frac{c(1 + d^{(j)})^{-1}}{\sum_{k=1}^{c}(1 + d^{(k)})^{-1}}, \quad j = 1, 2, ..., c, \tag{6}$$

  - Residual Connection

$$y_s = (1 + \alpha)z_s, \quad y_t = (1 + \alpha)z_t. \tag{7}$$



TransNorm Layer



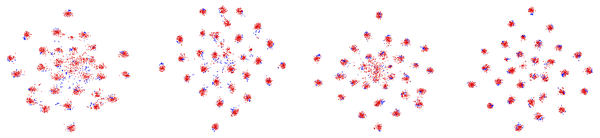
CNN | Residual Block | Inception Block

Experimental Results

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
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
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
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
MADA	90.0 ± 0.1	97.4 ± 0.1	99.6 ± 0.1	87.8 ± 0.2	70.3 ± 0.3	66.4 ± 0.3	85.2
SimNet	88.6 ± 0.5	98.2 ± 0.2	99.7 ± 0.2	85.3 ± 0.3	73.4 ± 0.8	71.6 ± 0.6	86.2
GTA	89.5 ± 0.5	97.9 ± 0.3	99.8 ± 0.4	87.7 ± 0.5	72.8 ± 0.3	71.4 ± 0.4	86.5
DANN+BN	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
DANN+TransNorm	**91.8** ± 0.4	**97.7** ± 0.2	**100.0** ± 0.0	**88.0** ± 0.2	**68.2** ± 0.2	**70.4** ± 0.3	**86.0**
CDAN+BN	94.1 ± 0.1	98.6 ± 0.1	**100.0** ± 0.0	92.9 ± 0.2	71.0 ± 0.3	69.3 ± 0.3	87.7
CDAN+TransNorm	**95.7** ± 0.3	**98.7** ± 0.2	**100.0** ± 0.0	**94.0** ± 0.2	**73.4** ± 0.4	**74.2** ± 0.3	**89.3**

Method (S:T)	Ar:Cl	Ar:Pr	Ar:Rw	Cl:Ar	Cl:Pr	Cl:Rw	Pr:Ar	Pr:Cl	Pr:Rw	Rw:Ar	Rw:Cl	Rw:Pr	Avg
ResNet-50	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
JAN	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
DANN+BN	**45.6**	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
DANN+TransNorm	43.5	**60.9**	**72.1**	**51.0**	**61.5**	**62.5**	**49.6**	**46.8**	**70.4**	**63.7**	**52.2**	**77.9**	**59.3**
CDAN+BN	**50.7**	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
CDAN+TransNorm	50.2	**71.4**	**77.4**	**59.3**	**72.7**	**73.1**	**61.0**	**53.1**	**79.5**	**71.9**	**59.0**	**82.9**	**67.6**

Adaptation Method		Network Backbone				Dataset	
Method	Avg	Method (Inception)	Avg	Method (DTN)	S→M	Method	VisDA
DANN	82.2	Inception-BN	75.5	CyCADA	90.4	GTA	69.5
CDAN	87.7	AdaBN	76.7	MCD	94.2	MCD	69.7
MDD+BN	88.8	**DANN+BN**	82.1	**DANN+BN**	73.9	**DANN+BN**	63.7
		DANN+TransNorm	**83.2**	**DANN+TransNorm**	**77.0**	**DANN+TransNorm**	**66.3**
MDD+TransNorm	**89.5**	**CDAN+BN**	84.9	**CDAN+BN**	89.2	**CDAN+BN**	70.0
		CDAN+TransNorm	**86.0**	**CDAN+TransNorm**	**94.7**	**CDAN+TransNorm**	**71.4**



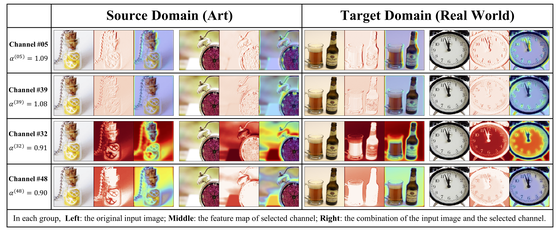
DANN+BN | DANN+TransNorm | CDAN+BN | CDAN+TransNorm

Comparison and Ablation Study

Method	DANN				CDAN			
Normalization	BN	AdaBN	AutoDIAL	**TransNorm**	BN	AdaBN	AutoDIAL	**TransNorm**
A→W	82.0	82.4	84.8	**91.8**	94.1	88.8	92.3	**95.7**
D→W	96.9	**97.7**	**97.7**	**97.7**	98.6	98.6	98.6	**98.7**
W→D	99.1	99.8	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
A→D	79.7	81.0	85.7	**88.0**	92.9	92.7	93.0	**94.0**
D→A	**68.2**	67.2	63.9	**68.2**	71.5	70.8	71.5	**73.4**
W→A	67.4	68.2	68.7	**70.4**	69.3	70.0	72.2	**74.2**
Avg	82.2	82.7	83.5	**86.0**	87.7	86.8	87.9	**89.3**

Type	Weight	Distance Type			Probability Type		
Ablation	$\alpha = 1$	μ	Wasserstein Distance	$\mu/\sqrt{\sigma^2 + \epsilon}$	Softmax	Gaussian	Student-t
A→W	94.6	95.0	94.5	**95.7**	94.9	94.8	**95.7**
D→W	98.6	**98.7**	**98.7**	**98.7**	97.9	98.6	**98.7**
W→D	**100.0**	**100.0**	**100.0**	**100.0**	99.8	**100.0**	**100.0**
A→D	93.4	93.0	91.0	**94.0**	91.4	93.1	**94.0**
D→A	71.5	72.8	72.6	**73.4**	69.7	72.4	**73.4**
W→A	72.9	73.7	73.6	**74.2**	73.2	73.0	**74.2**
Avg	88.5	88.9	88.4	**89.3**	87.9	88.7	**89.3**

Channel Visualization



In each group, **Left**: the original input image; **Middle**: the feature map of selected channel; **Right**: the combination of the input image and the selected channel.

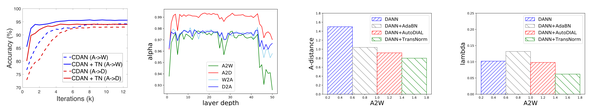
Theoretical Analysis

Ben-David *et al.* $\mathcal{E}_{\mathcal{T}}(h) \leq \mathcal{E}_{\mathcal{S}}(h) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda$

(a) The expected error of h on the source domain, $\mathcal{E}_{\mathcal{S}}(h)$;

(b) A-distance $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) = 2(1 - 2\epsilon)$, a measure of domain discrepancy, where ϵ is the error rate of a domain classifier trained to discriminate source domain and target domain;

(c) The error λ of the ideal joint hypothesis h^* on both source and target domains.



Convergence | α in AutoDIAL | A-distance | lambda