Machine Learning at Amazon Scale

Ragav Venkatesan

Outline

1. Personal Biography
2. Domain Adaptation
3. Out-of-the-box Pruned Networks

EDUCATION & EXPERIENCE

Present 2017

Amazon Alexa AI
Applied Scientist
Amazon Web Services AI Labs
Applied Scientist
Amazon Web Services SageMaker
Research Scientist

2014 2013

Intel
Computer Vision
Researcher Intern

Ph.D. 2017

Doctor of Philosophy
Advisor: Dr. Baoxin Li
Computer Science
Arizona State University

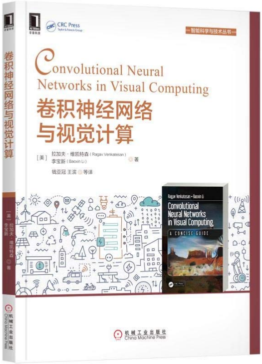
M.S. 2012

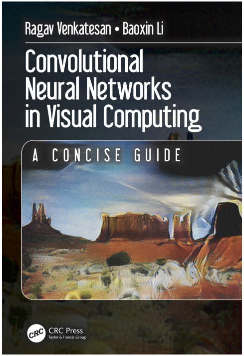
Master of Science
Advisor: Dr. David Frakes
Electrical Engineering
Arizona State University

B.E. 2010

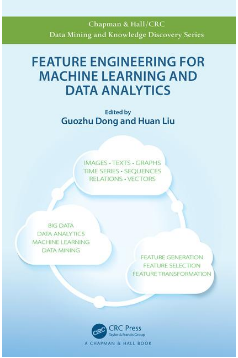
Bachelor of Engineering
Electronics and Communication Engineering
Anna University

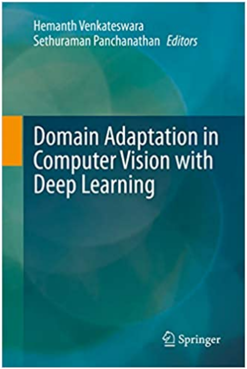
Books


CRC Press
Convolutional Neural Networks in Visual Computing
卷积神经网络与视觉计算
钱亚冠 王滨 等译
机械工业出版社
China Machine Press

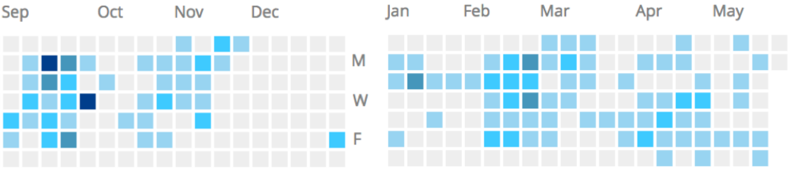

Ragav Venkatesan • Baoxin Li
Convolutional Neural Networks in Visual Computing
A CONCISE GUIDE
CRC Press
Taylor & Francis Group

BOOK CHAPTERS





SOFTWARE



YANN
yet another neural network toolbox

DMLC

GLUON

Amazon SageMaker

Career Highlights

- Recent Research Highlights:
  - EMNLP Negative Insights Oral, 2020 – ENAS not for BERT.
  - CVPR Oral, 2019 – Domain Adaptation using SNE.
- Production Highlights:
  - AWS Sagemaker Tensorflow and Mxnet.
  - AWS Sagemaker Image Classification.
  - AWS Sagemaker Object Detection.
  - AWS Sagemaker Semantic Segmentation.
  - AWS Sagemaker RL.
  - **AWS Sagemaker GroundTruth.**
  - **AWS Sagemaker Neo Model Compression using RL.**
  - AutoAlexa: a NAS and HPO platform.
  - AlexaFrugal: A cost efficent model training platform.

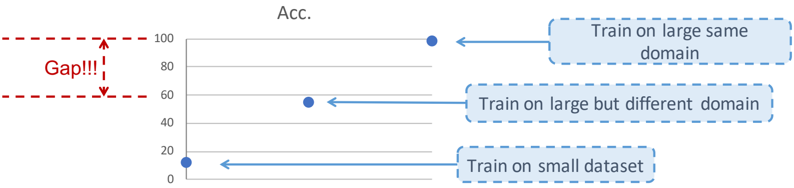
Outline

Motivation

Deep nets today are data inefficient.

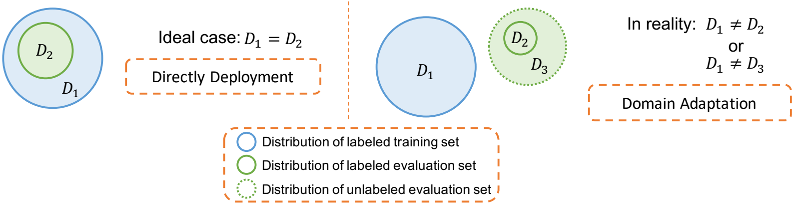
- Collect more labeled data: AWS GroundTruth, incremental learning.
- Use large public datasets such as ImageNet to get efficient Pretrained models.
- Self-supervised learning.
- Generating more synthetic data: Phantom Sampling.

There will always be a performance gap between the training and testing set.



MOTIVATION

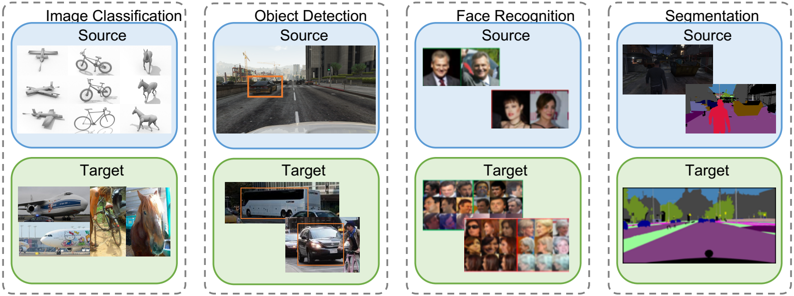
- Training domain $\mathcal{D}_1$.
- Generalization domain $\mathcal{D}_2$ or $\mathcal{D}_3$.



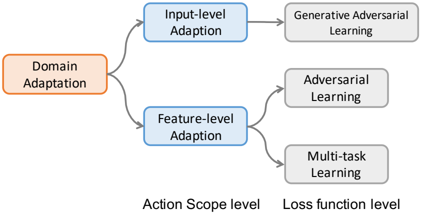
DEFINITIONS

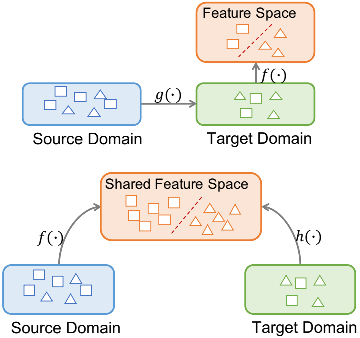
- The class of such problems where the knowledge from another domain is recycled to work to a new target domain is called **domain adaptation**.
- If the solution can perform equally-well in both domains, it is called as **domain generalization**.

Applications



CATEGORIES OF DOMAIN ADAPTATION





Categories of Domain Adaptation

- **Domain Transformation:** To build a transformation from target data to source domain and reuse the source feature extractor and classifier ($x^t \rightarrow x^s$).
- **Latent-Space Transformation:** To build a transformation of features extracted from source and features extracted from target into each other or into a common latent space.

Inter-domain distances

Consider the distance between a sample from the source domain and one from a target domain in the latent-space,

$$d(x_i^s, x_j^t) = \|\Phi_{\mathcal{D}^s}(x_i^s) - \Phi_{\mathcal{D}^t}(x_j^t)\|^2, \tag{1}$$

where,

- $\Phi_{\mathcal{D}^s}(\cdot) \rightarrow \mathbb{R}^d$
- $\Phi_{\mathcal{D}^t}(\cdot) \rightarrow \mathbb{R}^d$

are deep neural networks.

PROBABILITY OF INTER-DOMAIN SIMILARITY

In this latent-space,

$$p_{ij} = \frac{\exp(-d(x_i^s, x_j^t))}{\sum_{x \in \mathcal{D}^s} \exp(-d(x, x_j^t))}, \tag{2}$$

is the probability that the target sample $x_j^t \in \mathcal{D}^t$ has the same label as the source sample $x_i^s \in \mathcal{D}^s$.

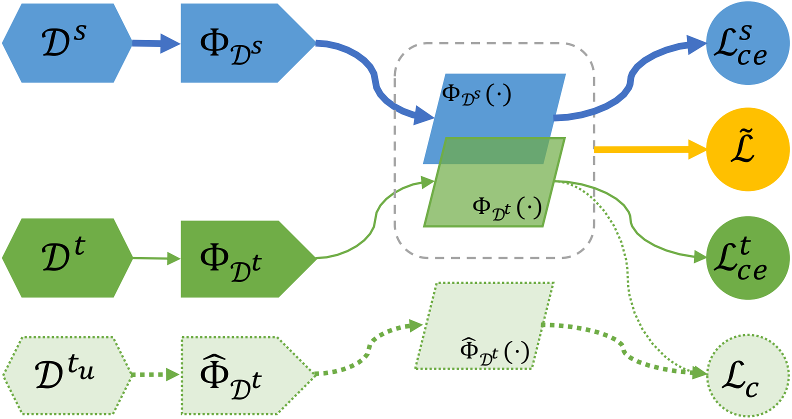
- We actually have the label for both x_i^s and x_j^t, $\rightarrow y_i^s$ and y_j^t.
- If $y_i^s = y_j^t$, we want p_{ij} to be maximized.
- If otherwise, we want p_{ij} to be minimized.

Hausdorffian Likelihood

$$\tilde{\mathcal{L}} = \sum_{x_j \in \mathcal{D}^t} \left[\frac{\sup_{x \in \mathcal{D}^s_k} \{a | a \in d(x, x_j)\}}{\inf_{x \in \mathcal{D}^s_{\not k}} \{b | b \in d(x, x_j)\}}, \text{ for } k = y_j \right]. \tag{3}$$

- Only minimize the largest distance between the samples of the same class.
- Only maximize the smallest distance between the samples of different classes.

d-SNE ARCHITECTURE



- $\tilde{\mathcal{L}}$ is the d-SNE loss.
- $\mathcal{L}_c$ is a clustering loss that we add as an unsupervised extension.

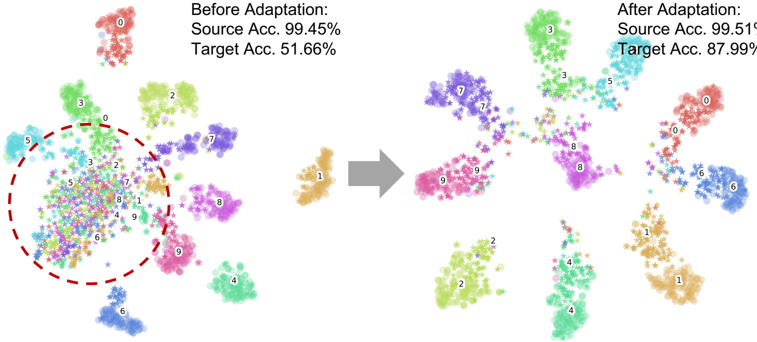
Digits Dataset

Method	Setting	k	MNIST→MNIST-M	MNIST→USPS	USPS→MNIST	MNIST→SVHN	SVHN→MNIST
DIRT-T[1]	U		98.90	-	-	54.50	99.40
SE[2]	U		-	98.23	99.54	71.40	92.00
SBADA-GAN[3]	U		99.40	95.04	97.60	61.08	76.14
G2A[4]	U		-	95.30	90.80	-	92.40
FADA[5]	S	7	-	94.40	91.50	47.00	87.20
CCSA[6]	S	10	78.29	97.22	95.71	37.63	94.57
d-SNE	S	7	84.62	97.53	97.52	53.19	95.68
d-SNE	S	10	87.80	99.00	98.49	61.73	96.45
d-SNE	SS	10	94.12	-	-	77.63	97.60

[1] R. Shu, H. H. Bui, H. Narui, and S. Ermon. A DIRT-T Approach to unsupervised Domain Adaption. In *Proc. ICLR*, 2018
[2] G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for Visual Domain Adaptation. In *Proc. ICLR*, 2018
[3] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo. From source to target and back: symmetric bi-directional adaptive GAN. In *Proc. CVPR*, 2018.
[4] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. Generate To Adapt: Aligning Domains using Generative Adversarial Networks. In *Proc. CVPR*, 2018
[5] S. Motiian, Q. Jones, S. M. Iranmanesh, and G. Doretto. Few-Shot Adversarial Domain Adaptation. In *Proc. NIPS 2018*
[6] S.Motiian, M.Piccirilli, D.A. Adjeroh, and G.Doretto. Unified Deep Supervised Domain Adaptation and Generalization. In *Proc. IEEE ICCV*, 2017

- d-SNE achieved the best results in the supervised setting.
- Extended model in the semi-supervised setting further improved the performance.

Digits Dataset - t-SNE



MNIST -> MNIST-M

Other Datasets

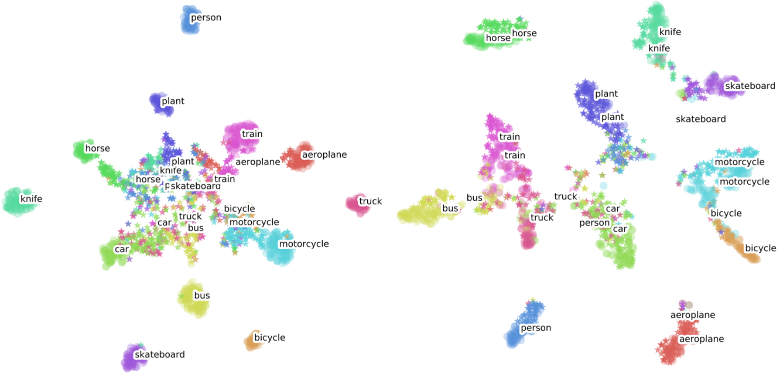
Method	Setting	k	A → D	A → W	D → A	D → W	W → A	W → D
DANN[1]	U		-	73.00	-	96.40	-	-
DRCN[2]			68.70	67.10	56.00	96.40	54.09	99.00
G2A[3]			87.70	89.50	72.80	97.90	71.40	99.80
CCSA[4]	S	3	89.00	88.20	**71.80**	96.40	**72.10**	**97.60**
FADA[5]		3	88.20	88.10	68.10	96.40	71.10	97.50
d-SNE (VGG16)		0	62.40	61.49	48.92	82.24	47.52	90.42
		3	**91.44**	**90.13**	71.06	**97.10**	71.74	97.46
d-SNE (ResNet101)		0	80.41	75.26	67.39	96.39	65.55	98.31
		3	**94.65**	**96.58**	**75.51**	**99.10**	**74.20**	**100.00**

[1] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain Adversarial Training of Neural Networks. *JMLR*, 17:1–35, 2016
[2] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for un- supervised domain adaptation. In Proc. ECCV, 2016
[3] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo. From source to target and back: symmetric bi-directional adaptive GAN. In *Proc. CVPR*, 2018.
[3] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. Generate To Adapt: Aligning Domains using Generative Adversarial Networks. In *Proc. CVPR*, 2018
[4] S. Motiian, M. Piccirilli, D.A. Adjeroh, and G.Doretto. Unified Deep Supervised Domain Adaptation and Generalization. In *Proc. IEEE ICCV*, 2017
[5] S. Motiian, Q. Jones, S. M. Iranmanesh, and G. Doretto. Few-Shot Adversarial Domain Adaptation. In *Proc. NIPS 2018*

Method	Setting	Source	Target
G2A[1]	U	44.50	77.10
SE[2]	SS	52.80	85.40
CCSA[3]	S	52.80	76.89
d-SNE	S	52.80	80.66
	SS	52.80	**86.15**

[1] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. Generate To Adapt: Aligning Domains using Generative Adversarial Networks. In *Proc. CVPR*, 2018
[2] G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for Visual Domain Adaptation. In *Proc. ICLR*, 2018
[3] S. Motiian, M. Piccirilli, D.A. Adjeroh, and G.Doretto. Unified Deep Supervised Domain Adaptation and Generalization. In *Proc. IEEE ICCV*, 2017

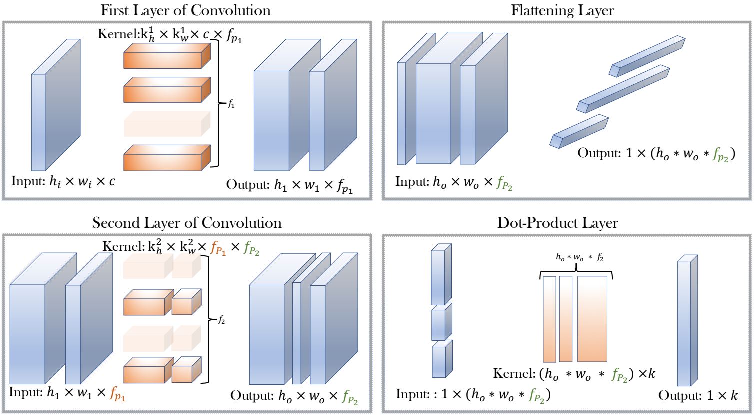
VISDA-C t-SNE


person
plant
horse
train
aeroplane
aeroplane
plant
knife
horse
skateboard
train
knife
truck
bicycle
truck
car
bus
motorcycle
car
motorcycle
bus
bicycle
skateboard
horse
horse
knife
knife
skateboard
skateboard
plant
plant
train
train
motorcycle
motorcycle
bus
bus
truck
car
truck
person
car
bicycle
bicycle
person
aeroplane
aeroplane

Outline

1. Personal Biography
2. Domain Adaptation
3. Out-of-the-box Pruned Networks
   - Formalization
   - Taxonomy
   - Initialization?
   - Which to prune?
   - How much?
   - Random Search
   - Transferabilty
   - RL search

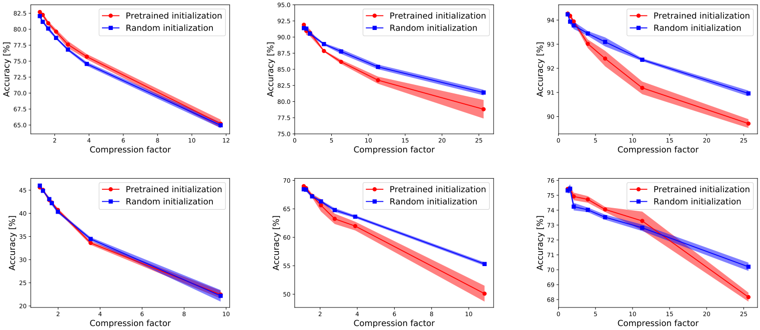
Formalization



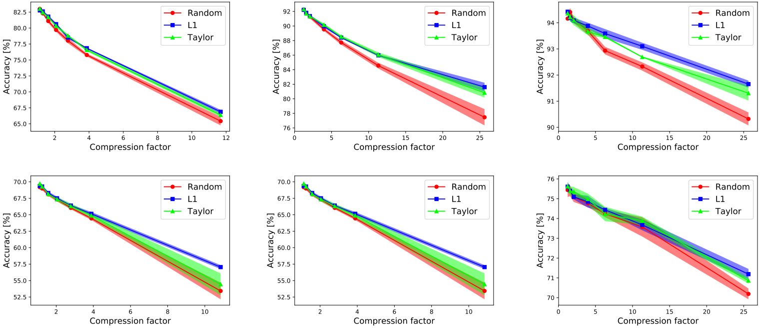
TAXONOMY OF CHANNEL PRUNING

Policy	**How much to prune?**		**Which to prune?**	**Inits for fine-tuning**
	Level	*Policy*		
Online Offline Deterministic	Network Layer	Heuristics Learned Online	Random Max-metric Learned Online	Random Trained Weights Rewound

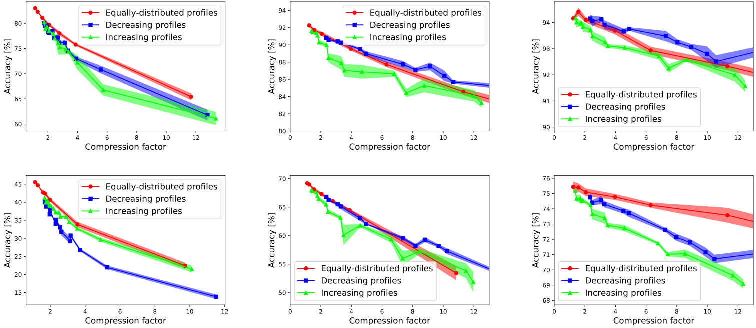
Inits ain't a thing



WHICH CHANNELS YOU PRUNE DOENS'T MATTER.



Heuristic baselines for how much to prune.



Ragav Venkatesan

Biography

Domain Adaptation

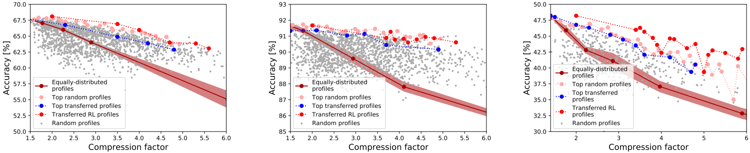
Motivation
Literature
d-SNE
Results

OOTB

Formalization
Taxonomy
Initialization?
Which to prune?
How much?
Random Search
Transferabilty
RL search

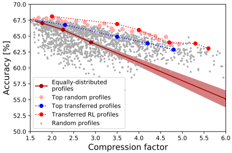
Fin.

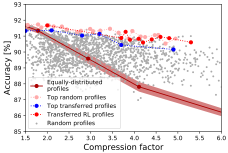
Brute-force your way into finding good profiles.

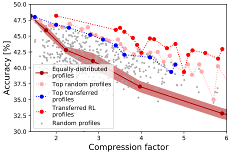


Ragav Venkatesan

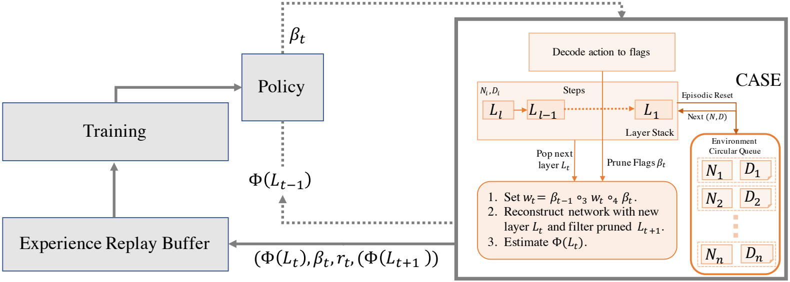
Pruning profiles are transferable



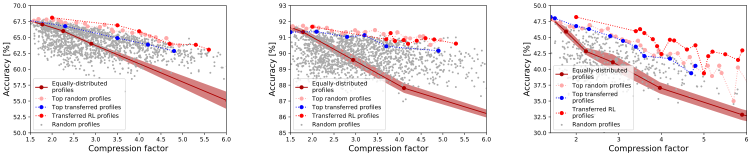




Find better transferable profiles?



YEP ...


Accuracy [%]
Compression factor
Equally-distributed profiles
Top random profiles
Top transferred profiles
Transferred RL profiles
Random profiles

Questions ?