DEEP LEARNING FOR COMPUTER VISION

Summer Seminar UPC TelecomBCN, 4 - 8 July 2016

Instructors

Xavier Giró-i-Nieto
Elisa Sayrol
Amaia Salvador
Jordi Torres
Eva Mohedano
Kevin McGuinness

Organizers

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

telecom BCN

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

DCU Dublin City University
Ollscoil Chathair Bhaile Átha Cliath

Insight Centre for Data Analytics

NVIDIA GPU CENTER OF EXCELLENCE

Co-funded by the Erasmus+ Programme of the European Union

+ info: **TelecomBCN.DeepLearning.Barcelona**

Day 2 Lecture 5

Transfer learning and domain adaptation

Semi-supervised and transfer learning

Myth: you can't do deep learning unless you have a million labelled examples for your problem.

Reality

- You can learn useful representations from **unlabelled data**
- You can **transfer** learned representations from a related task
- You can train on a nearby **surrogate objective** for which it is easy to generate labels

Transfer learning: idea

Instead of training a deep network from scratch for your task:

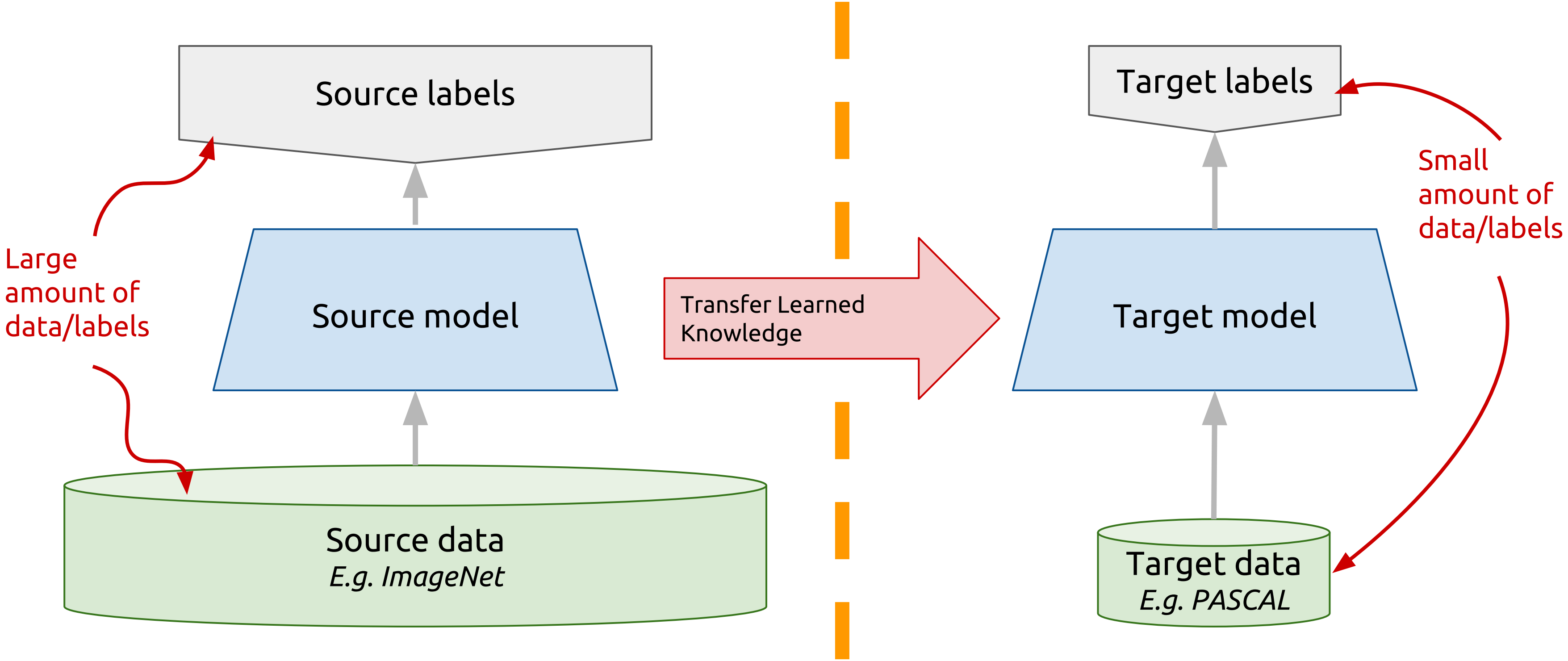
- Take a network trained on a different domain for a different **source task**
- Adapt it for your domain and your **target task**

This lecture will talk about how to do this.

Variations:

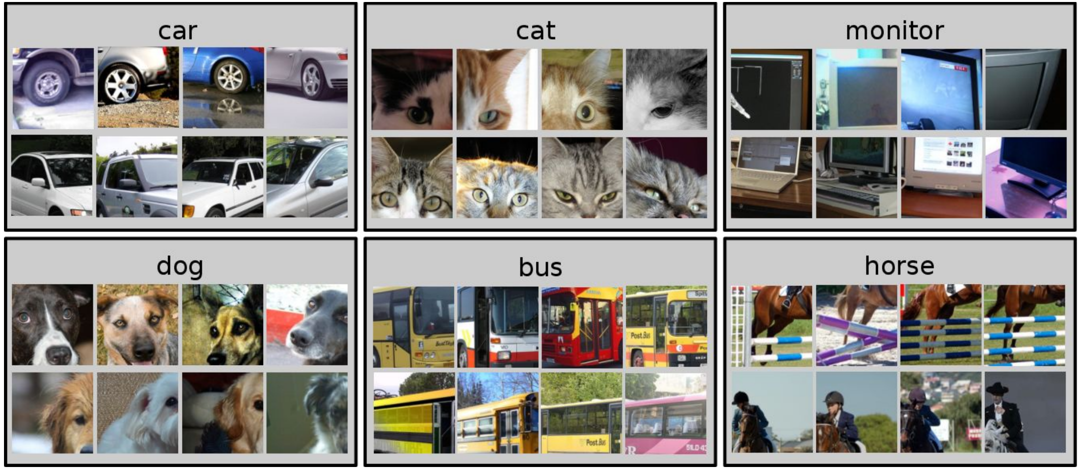
- Same domain, different task
- Different domain, same task

Transfer learning: idea



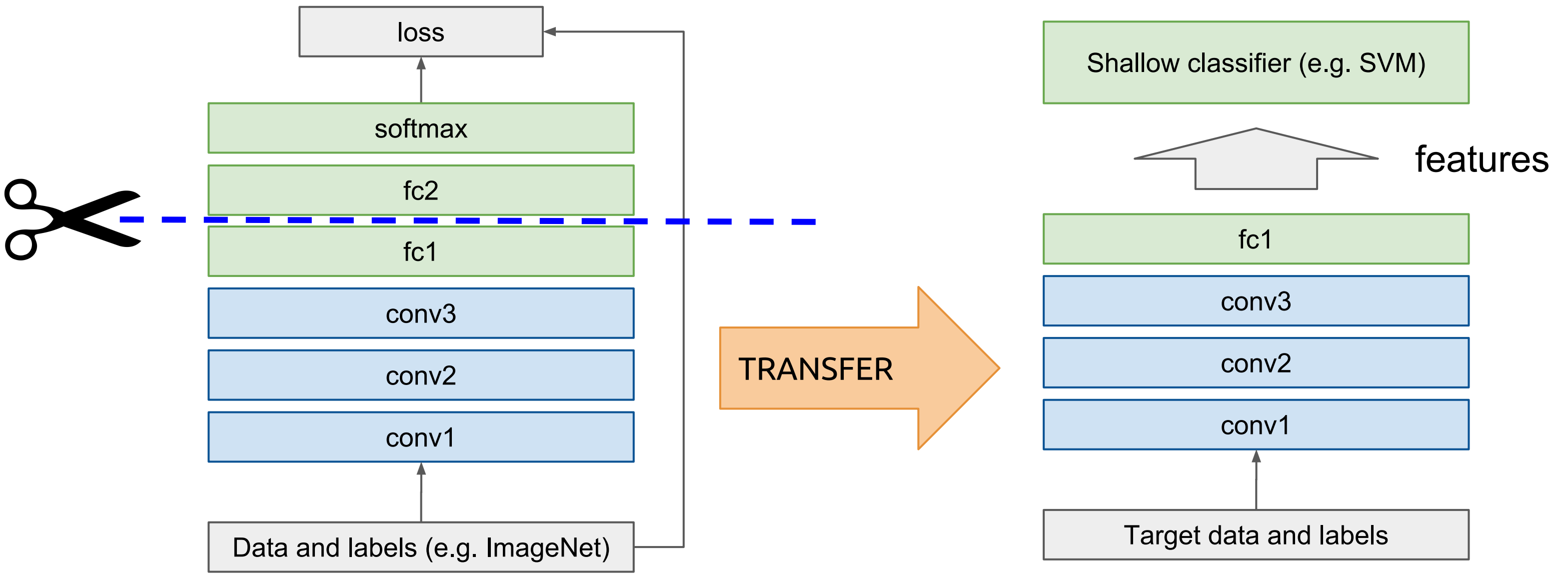
Example: PASCAL VOC 2007

- Standard classification benchmark, 20 classes, ~10K images, 50% train, 50% test
- Deep networks can have many parameters (e.g. 60M in Alexnet)
- Direct training (from scratch) using only 5K training images can be problematic. Model overfits.
- How can we use deep networks in this setting?



“Off-the-shelf”

Idea: use outputs of one or more layers of a network trained on a different task as generic feature detectors. Train a new shallow model on these features.



Off-the-shelf features

Works surprisingly well in practice!

Surpassed or on par with state-of-the-art in several tasks in 2014

Image classification:

- PASCAL VOC 2007
- Oxford flowers
- CUB Bird dataset
- MIT indoors

Image retrieval:

- Paris 6k
- Holidays
- UKBench

Method	mean Accuracy
HSV [27]	43.0
SIFT internal [27]	55.1
SIFT boundary [27]	32.0
HOG [27]	49.6
HSV+SIFTi+SIFTb+HOG(MKL) [27]	72.8
BOW(4000) [14]	65.5
SPM(4000) [14]	67.4
FLH(100) [14]	72.7
BiCos seg [7]	79.4
Dense HOG+Coding+Pooling[2] w/o seg	76.7
Seg+Dense HOG+Coding+Pooling[2]	80.7
CNN-SVM w/o seg	74.7
CNNaug-SVM w/o seg	**86.8**

Oxford 102 flowers dataset

Razavian et al, **CNN Features off-the-shelf: an Astounding Baseline for Recognition,** CVPRW 2014 http://arxiv.org/abs/1403.6382

Can we do better than off the shelf features?

Domain adaptation

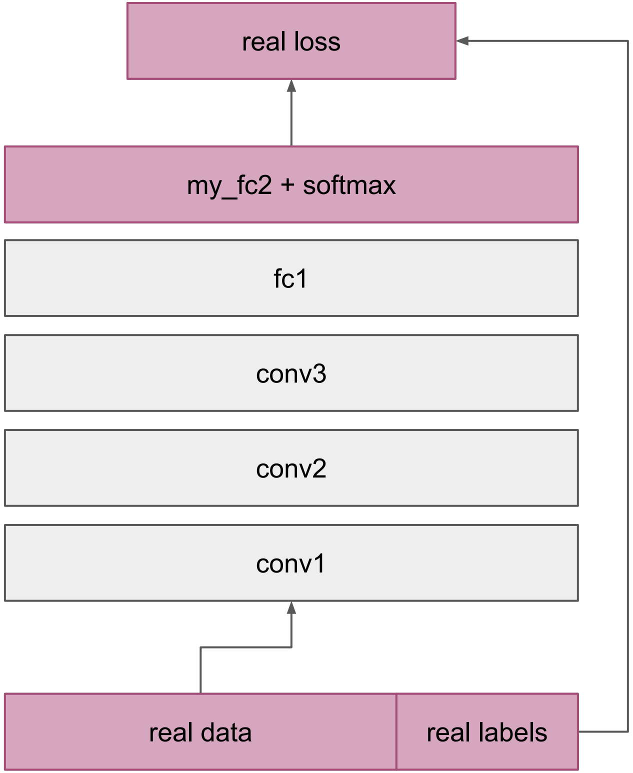
Fine-tuning: supervised domain adaptation

Train deep net on “nearby” task for which it is easy to get labels using standard backprop

- E.g. ImageNet classification
- Pseudo classes from augmented data
- Slow feature learning, ego-motion

Cut off top layer(s) of network and replace with supervised objective for target domain

Fine-tune network using backprop with labels for target domain until validation loss starts to increase



Freeze or fine-tune?

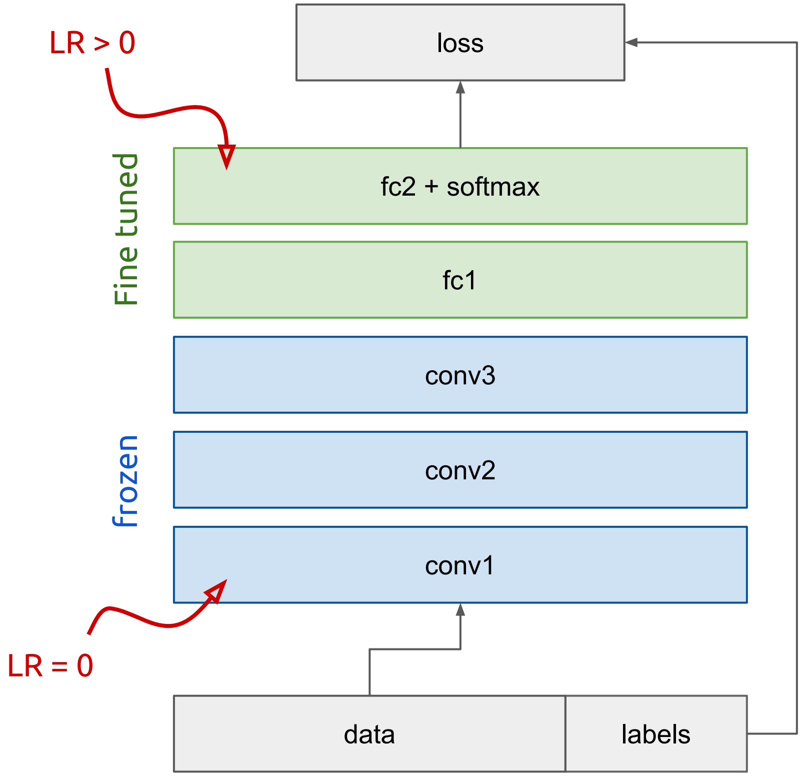
Bottom *n* layers can be frozen or fine tuned.

- **Frozen**: not updated during backprop
- **Fine-tuned**: updated during backprop

Which to do depends on target task:

- **Freeze**: target task labels are scarce, and we want to avoid overfitting
- **Fine-tune**: target task labels are more plentiful

In general, we can set learning rates to be different for each layer to find a tradeoff between freezing and fine tuning



How transferable are features?

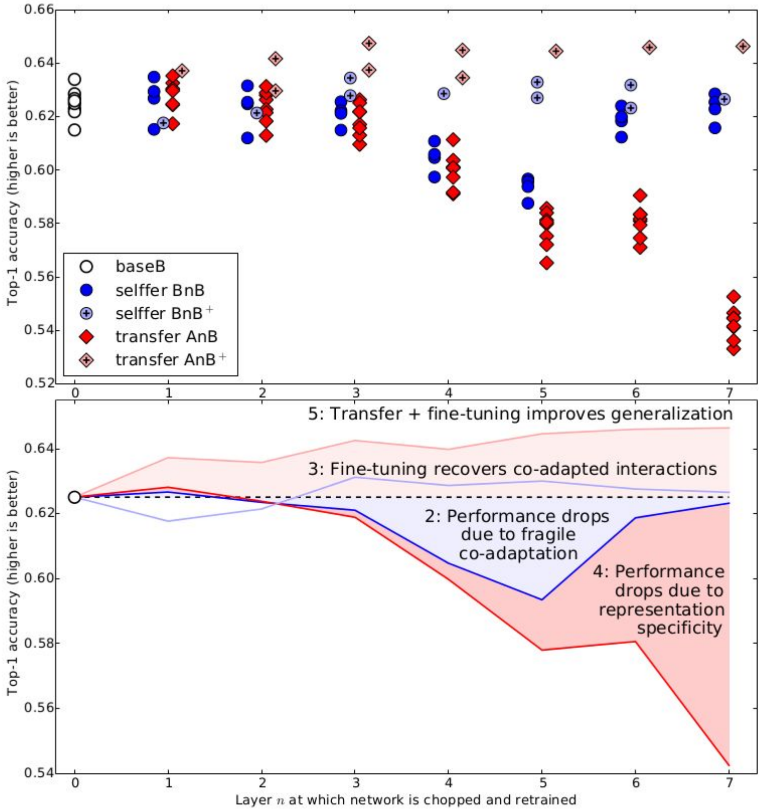
Lower layers: more general features. Transfer very well to other tasks.

Higher layers: more task specific.

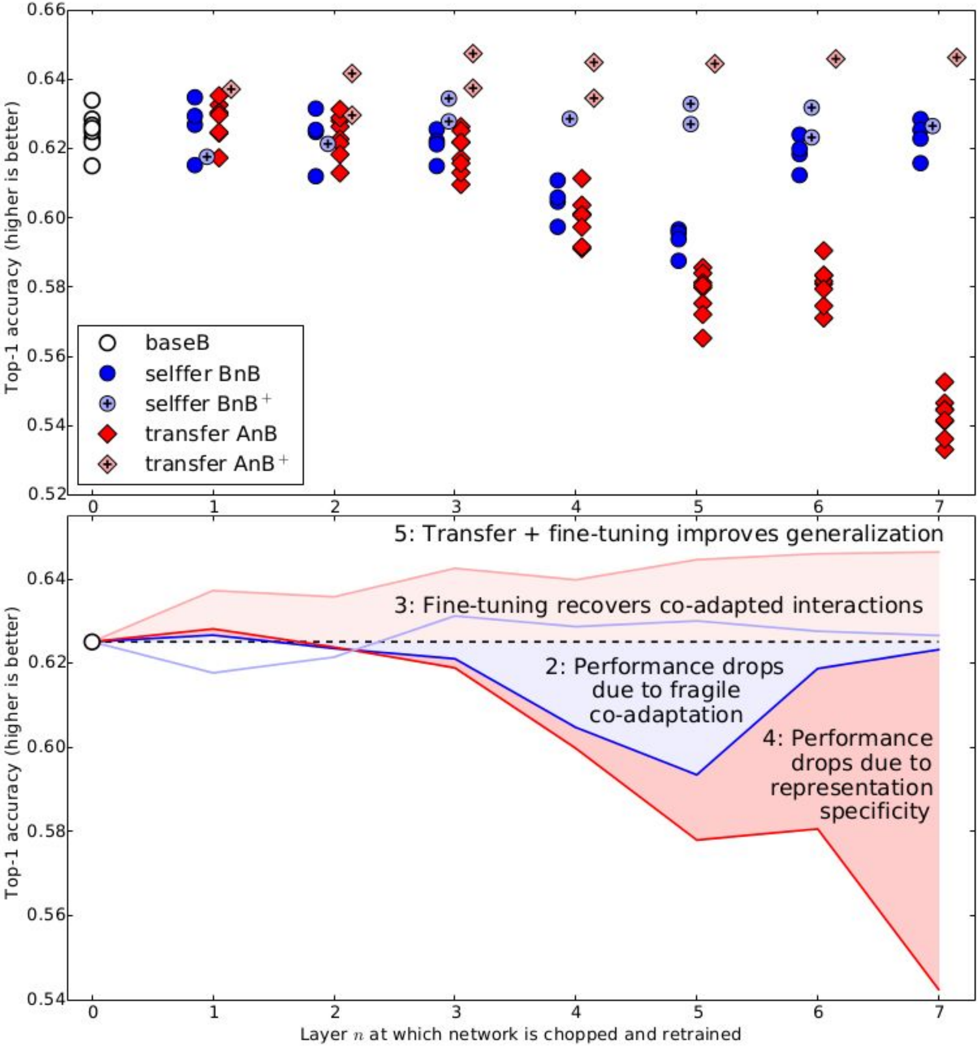
Fine-tuning improves generalization when sufficient examples are available.

Transfer learning and fine tuning often lead to better performance than training from scratch on the target dataset.

Even features transferred from distant tasks are often better than random initial weights!



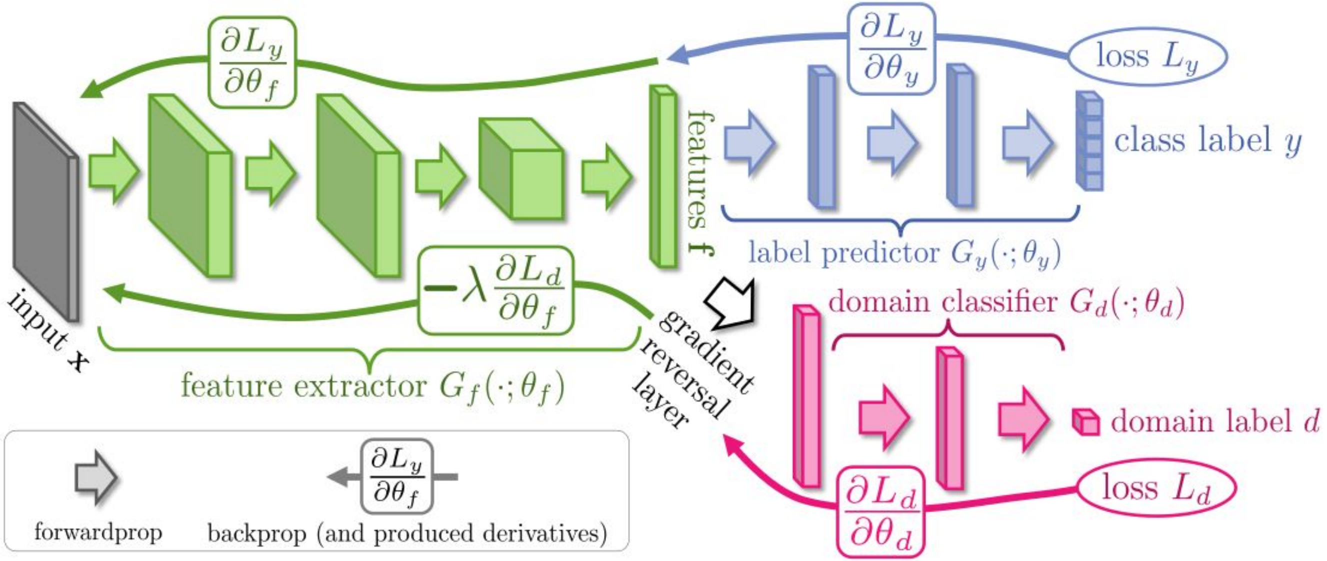
Yosinki et al. **How transferable are features in deep neural networks.** NIPS 2014. https://arxiv.org/abs/1411.1792



baseB
selffer BnB
selffer BnB$^+$
transfer AnB
transfer AnB$^+$
Top-1 accuracy (higher is better)
5: Transfer + fine-tuning improves generalization
3: Fine-tuning recovers co-adapted interactions
2: Performance drops due to fragile co-adaptation
4: Performance drops due to representation specificity
Layer n at which network is chopped and retrained

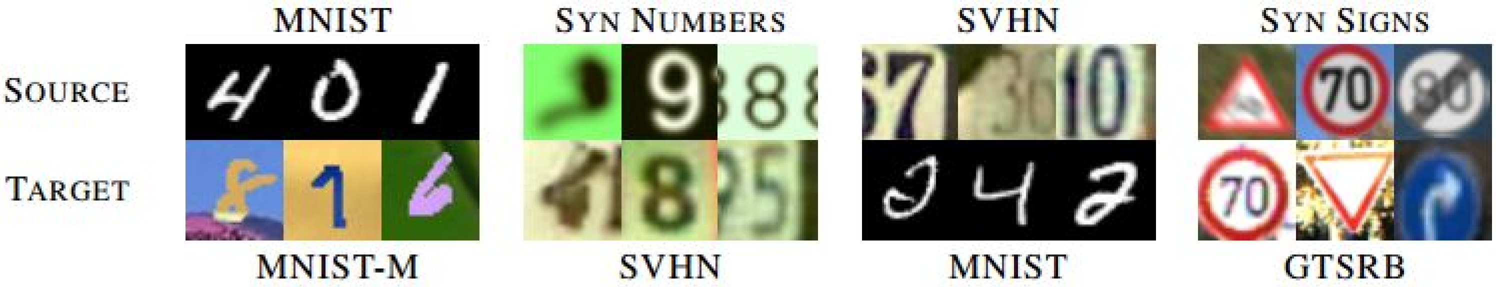
Unsupervised domain adaptation

Also possible to do domain adaptation without labels in target set.



Y Ganin and V Lempitsky, **Unsupervised Domain Adaptation by Backpropagation,** ICML 2015 https://arxiv.org/abs/1409.7495

Unsupervised domain adaptation



METHOD — SOURCE / TARGET	MNIST / MNIST-M	SYN NUMBERS / SVHN	SVHN / MNIST	SYN SIGNS / GTSRB
SOURCE ONLY	.5749	.8665	.5919	.7400
SA (FERNANDO ET AL., 2013)	.6078 (7.9%)	.8672 (1.3%)	.6157 (5.9%)	.7635 (9.1%)
PROPOSED APPROACH	**.8149** (57.9%)	**.9048** (66.1%)	**.7107** (29.3%)	**.8866** (56.7%)
TRAIN ON TARGET	.9891	.9244	.9951	.9987

Y Ganin and V Lempitsky, **Unsupervised Domain Adaptation by Backpropagation,** ICML 2015 https://arxiv.org/abs/1409.7495

Summary

Possible to train very large models on small data by using transfer learning and domain adaptation

Off the shelf features work very well in various domains and tasks

Lower layers of network contain very generic features, higher layers more task specific features

Supervised domain adaptation via fine tuning almost always improves performance

Possible to do unsupervised domain adaptation by matching feature distributions