“What you saw is not what you get”

Domain adaptation for deep learning

Kate Saenko
Successes of Deep Learning in AI

Google's DeepMind Masters Atari Games

Deep Learning for self-driving cars

Google Translate

Time flies like an arrow

Face Recognition
So is AI solved?

pedestrian detection FAIL

https://www.youtube.com/watch?v=w2pxv8rFkU
Major limitation of deep learning

**Not data efficient:** Learning requires **millions** of labeled examples,
models do not generalize well to new domains; not like humans!
“What you saw is not what you get”

What your net is trained on

What it’s asked to label

“Dataset Bias”
“Domain Shift”
“Domain Adaptation”
“Domain Transfer”
Example: scene segmentation

Train on Cityscapes, Test on Cityscapes

Domain shift: Cityscapes to SF

Train on Cityscapes, Test on San Francisco Dashcam

No tunnels in CityScapes?

Applications to different types of domain shift

- From dataset to dataset
- From RGB to depth
- From simulated to real control
- From CAD models to real images
Today

- Show that deep models can be adapted without labels
- Propose two deep adaptation methods:
  - adversarial alignment
  - correlation alignment
- Show applications
Background: Domain Adaptation from source to target distribution

Source Domain $\sim P_S(X, Y)$
- lots of **labeled** data

$D_S = \{(x_i, y_i), \forall i \in \{1, \ldots, N\}\}$

Target Domain $\sim P_T(Z, H)$
- unlabeled or limited labels

$D_T = \{(z_j, ?), \forall j \in \{1, \ldots, M\}\}$
Background: unsupervised domain adaptation

- NO labels in target domain
- Roughly, three categories of methods
  - Sample re-weighting
  - Subspace matching
  - Deep methods
How to adapt a deep network?
How to adapt a deep network?

- Applying source classifier to target domain can yield inferior performance…
How to adapt a deep network?

- Fine tune?
  .....Zero or few labels in target domain

- Siamese network?
  .....No paired / aligned instance examples!

IDEA: align feature distributions
Deep distribution alignment

- by minimizing distance between distributions, e.g.
  - CORrelation ALignment Sun and Saenko, AAAI 2016

- ...or by adversarial domain alignment, e.g.
  - Domain Confusion E. Tzeng et al. ICCV 2015
  - Reverse Gradient Y. Ganin and V. Lempitsky ICML 2015
Adversarial Domain Adaptation

Eric Tzeng
UC Berkeley

Judy Hoffman
UC Berkeley

Trevor Darrell
UC Berkeley
Adversarial networks
Adversarial networks
Adversarial domain adaptation
Adversarial domain adaptation

Source Data + Labels
- backpack
- chair
- bike

Unlabeled Target Data

Encoder

can be shared

Encoder

Classifier
- classification
- loss

Adversarial domain adaptation can be shared.
Adversarial domain adaptation

- **Source Data + Labels**
  - backpack
  - chair
  - bike

- **Unlabeled Target Data**

- **Encoder**
  - can be shared

- **Classifier**
  - classification loss

- **Discriminator**
  - Adversarial loss

Adversarial domain adaptation can be shared.
Adversarial domain adaptation

Source Data + Labels
- backpack
- chair
- bike

Unlabeled Target Data

Encoder

Classifier

Discriminator

Classification loss

Adversarial loss

can be shared
Design choices in adversarial adaptation

Generative or discriminative? ✔

Shared or not? ✔

“confusion” Which loss?

<table>
<thead>
<tr>
<th>Method</th>
<th>Base model</th>
<th>Weight sharing</th>
<th>Adversarial loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient reversal [16]</td>
<td>discriminative</td>
<td>shared</td>
<td>minimax</td>
</tr>
<tr>
<td>Domain confusion [12]</td>
<td>discriminative</td>
<td>shared</td>
<td>confusion</td>
</tr>
<tr>
<td>CoGAN [13]</td>
<td>generative</td>
<td>unshared</td>
<td>GAN</td>
</tr>
</tbody>
</table>

Deep domain confusion

Adversarial Training of domain label predictor and domain confusion loss:

\[
\min_{\theta_D} \mathcal{L}_D(x_S, x_T, \theta_{\text{repr}}; \theta_D) = -\sum_d \mathbb{1}[y_D = d] \log q_d
\]

\[
\min_{\theta_{\text{repr}}} \mathcal{L}_{\text{conf}}(x_S, x_T, \theta_D; \theta_{\text{repr}}) = -\sum_d \frac{1}{D} \log q_d.
\]

Domain Label Cross-entropy with uniform distribution

[Tzeng ICCV15]
Deep domain confusion

Train a network to minimize classification loss AND confuse two domains

\[ \mathcal{L}_D(x_S, x_T, \theta_{\text{repr}}; \theta_D) = -\sum_d 1[y_D = d] \log q_d \]

\[ q = \text{softmax}(\theta_D^T f(x; \theta_{\text{repr}})) = p(y_D = 1|x) \]

\[ \mathcal{L}_{\text{conf}}(x_S, x_T, \theta_D; \theta_{\text{repr}}) = -\sum_d \frac{1}{D} \log q_d \]

(cross-entropy with uniform distribution)

[Deep domain confusion] (Tzeng ICCV15)
Applications to different types of domain shift

From dataset to dataset

From simulated to real control

From RGB to depth

From CAD models to real images
ImageNet adapted to Caltech [Tzeng ICCV15]

![Graph showing multiclass accuracy vs. number of labeled target examples. The graph compares different methods: Source+Target CNN, Ours: softlabels only, Ours: dom confusion+softlabels.]
Results on Cityscapes to SF adaptation

Before domain confusion

After domain confusion

Adversarial Loss Functions

**Confusion loss** [Tzeng 2015]
\[
\max_D \mathbb{E}_{x \sim p_S(x)} \left[ \log D(M_S(x)) \right] + \mathbb{E}_{x \sim p_T(x)} \left[ \log(1 - D(M_T(x))) \right]
\]
\[
\max_{M_S, M_T} \sum_{d \in \{S, T\}} \mathbb{E}_{x \sim p_d(x)} \left[ \frac{1}{2} \log D(M_{d}(x)) + \frac{1}{2} \log(1 - D(M_{d}(x))) \right]
\]

**Minimax loss** [Ganin 2015]
\[
\min_{M_S, M_T} \max_D V(D, M_S, M_T) = \mathbb{E}_{x \sim p_S(x)} \left[ \log D(M_S(x)) \right] + \mathbb{E}_{x \sim p_T(x)} \left[ \log(1 - D(M_T(x))) \right]
\]

**GAN loss** [Goodfellow 2014]
\[
\max_D \mathbb{E}_{x \sim p_S(x)} \left[ \log D(M_S(x)) \right] + \mathbb{E}_{x \sim p_T(x)} \left[ \log(1 - D(M_T(x))) \right]
\]
\[
\max_{M_T} \mathbb{E}_{x \sim p_T(x)} \left[ \log D(M_T(x)) \right].
\]

“stronger gradients”
Adversarial Discriminative Domain Adaptation (ADDA) (in submission)

Which loss? Shared or not? Generative or discriminative?

<table>
<thead>
<tr>
<th>Method</th>
<th>Base model</th>
<th>Weight sharing</th>
<th>Adversarial loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient reversal [16]</td>
<td>discriminative</td>
<td>shared</td>
<td>minimax</td>
</tr>
<tr>
<td>Domain confusion [12]</td>
<td>discriminative</td>
<td>shared</td>
<td>confusion</td>
</tr>
<tr>
<td>CoGAN [13]</td>
<td>generative</td>
<td>unshared</td>
<td>GAN</td>
</tr>
<tr>
<td>ADDA</td>
<td>discriminative</td>
<td>unshared</td>
<td>GAN</td>
</tr>
</tbody>
</table>
ADDA: Adaptation on digits

<table>
<thead>
<tr>
<th>Method</th>
<th>MNIST $\rightarrow$ USPS</th>
<th>USPS $\rightarrow$ MNIST</th>
<th>SVHN $\rightarrow$ MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source only</td>
<td>0.752 ± 0.016</td>
<td>0.571 ± 0.017</td>
<td>0.601 ± 0.011</td>
</tr>
<tr>
<td>Gradient reversal</td>
<td>0.771 ± 0.018</td>
<td>0.730 ± 0.020</td>
<td>0.739 [16]</td>
</tr>
<tr>
<td>Domain confusion</td>
<td>0.791 ± 0.005</td>
<td>0.665 ± 0.033</td>
<td>0.681 ± 0.003</td>
</tr>
<tr>
<td>CoGAN</td>
<td>0.912 ± 0.008</td>
<td>0.891 ± 0.008</td>
<td>did not converge</td>
</tr>
<tr>
<td>ADDA (Ours)</td>
<td>0.894 ± 0.002</td>
<td>0.901 ± 0.008</td>
<td>0.760 ± 0.018</td>
</tr>
</tbody>
</table>
Applications to different types of domain shift

From dataset to dataset

From simulated to real control

From RGB to depth

From CAD models to real images
ADDA: Adaptation on RGB-D (in submission)

Train on RGB

Test on depth

<table>
<thead>
<tr>
<th></th>
<th>bathtub</th>
<th>bed</th>
<th>bookshelf</th>
<th>box</th>
<th>chair</th>
<th>counter</th>
<th>desk</th>
<th>door</th>
<th>dresser</th>
<th>garbage bin</th>
<th>lamp</th>
<th>monitor</th>
<th>night stand</th>
<th>pillow</th>
<th>sink</th>
<th>sofa</th>
<th>table</th>
<th>television</th>
<th>toilet</th>
<th>overall</th>
</tr>
</thead>
<tbody>
<tr>
<td># of instances</td>
<td>19</td>
<td>96</td>
<td>87</td>
<td>210</td>
<td>611</td>
<td>103</td>
<td>122</td>
<td>129</td>
<td>25</td>
<td>55</td>
<td>144</td>
<td>37</td>
<td>51</td>
<td>276</td>
<td>47</td>
<td>129</td>
<td>210</td>
<td>33</td>
<td>17</td>
<td>2401</td>
</tr>
<tr>
<td>Source only</td>
<td>0.000</td>
<td>0.010</td>
<td>0.011</td>
<td>0.124</td>
<td>0.188</td>
<td>0.029</td>
<td>0.041</td>
<td>0.047</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.039</td>
<td>0.587</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.010</td>
<td>0.000</td>
<td>0.139</td>
</tr>
<tr>
<td>ADDA (Ours)</td>
<td>0.000</td>
<td>0.146</td>
<td>0.046</td>
<td>0.229</td>
<td>0.344</td>
<td>0.447</td>
<td>0.025</td>
<td>0.023</td>
<td>0.000</td>
<td>0.018</td>
<td>0.292</td>
<td>0.081</td>
<td>0.020</td>
<td>0.297</td>
<td>0.021</td>
<td>0.116</td>
<td>0.143</td>
<td>0.091</td>
<td>0.000</td>
<td>0.211</td>
</tr>
<tr>
<td>Train on target</td>
<td>0.105</td>
<td>0.531</td>
<td>0.494</td>
<td>0.295</td>
<td>0.619</td>
<td>0.573</td>
<td>0.057</td>
<td>0.120</td>
<td>0.291</td>
<td>0.576</td>
<td>0.189</td>
<td>0.235</td>
<td>0.630</td>
<td>0.362</td>
<td>0.248</td>
<td>0.357</td>
<td>0.303</td>
<td>0.647</td>
<td>0.468</td>
<td></td>
</tr>
</tbody>
</table>
ADDA: Adaptation on RGB-D

Train on target

True label

stand
pillow
sink
sofa
table
television
toilet
Applications to different types of domain shift

From dataset to dataset

From RGB to depth

From simulated to real control

From CAD models to real images
Adapting Deep Visuomotor Representations with Weak Pairwise Constraints

Eric Tzeng\textsuperscript{1}, Coline Devin\textsuperscript{1}, Judy Hoffman\textsuperscript{1}, Chelsea Finn\textsuperscript{1}, Pieter Abbeel\textsuperscript{1}, Sergey Levine\textsuperscript{1}, Kate Saenko\textsuperscript{2}, Trevor Darrell\textsuperscript{1}

\textsuperscript{1}University of California, Berkeley
\textsuperscript{2}Boston University
From simulation to real world control [Tzeng, Devin, et al 16]
Weak pairwise constraints

[Tzeng, Devin, et al 16]
Robotic task: place rope on scale

<table>
<thead>
<tr>
<th>Method</th>
<th># Sim</th>
<th># Real (unlabeled)</th>
<th>Success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic only</td>
<td>4000</td>
<td>0</td>
<td>38.1% ± 8%</td>
</tr>
<tr>
<td>Autoencoder (100)</td>
<td>0</td>
<td>100</td>
<td>28.6% ± 25%</td>
</tr>
<tr>
<td>Autoencoder (500)</td>
<td>0</td>
<td>500</td>
<td>33.2% ± 15%</td>
</tr>
<tr>
<td>Domain alignment with randomly assigned pairs</td>
<td>4000</td>
<td>100</td>
<td>33.3% ± 16%</td>
</tr>
<tr>
<td>Domain alignment with weakly supervised pairwise constraints</td>
<td>4000</td>
<td>100</td>
<td><strong>76.2% ± 16%</strong></td>
</tr>
<tr>
<td>Oracle</td>
<td>0</td>
<td>500 (labeled)</td>
<td>71.4% ± 14%</td>
</tr>
</tbody>
</table>

[Tzeng, Devin, et al 16]
Applications to different types of domain shift

From dataset to dataset

From RGB to depth

From simulated to real control

From CAD models to real images
Domain Adaptation via Correlation Alignment

Baochen Sun
Microsoft

Xingchao Peng
Boston University
Deep CORAL: Correlation Alignment for Deep Domain Adaptation

\[ \mathcal{L} = \mathcal{L}_{\text{CLASS}} \]

Classification Loss
Deep CORAL: Correlation Alignment for Deep Domain Adaptation

\[ \mathcal{L} = \mathcal{L}_{\text{CLASS}} + \sum_{i=1}^{i} \lambda_i \mathcal{L}_{\text{CORAL}} \]

\[ \| C_S - C_T \|^2_F \]

[Sun 2016]
Generative CORAL Network

Realistic Image $I_t$

CAD-synthetic Image $I_s$

$\mathcal{L}_{\text{conv}1-1}^{\text{coral}}$, $\mathcal{L}_{\text{conv}2-1}^{\text{coral}}$, $\mathcal{L}_{\text{conv}3-1}^{\text{coral}}$, $\mathcal{L}_{\text{conv}4-1}^{\text{coral}}$, $\mathcal{L}_{\text{conv}5-1}^{\text{coral}}$

$\mathcal{L}_{\text{conv}3-2}^{\text{feat}}$

Deep Generative Correlation Alignment Net (VGG-16)
Synthetic to real adaptation for object recognition

Train on synthetic

Test on real
Synthetic to Real Adaptation with Deep Generative Correlation Alignment Networks (in submission)
Synthetic to Real Adaptation with Deep Generative Correlation Alignment Networks (in submission)
Synthetic to Real Adaptation with Deep Generative Correlation Alignment Networks (in submission)
Summary

- Deep models can be adapted to new domains without labels
- Proposed two deep feature alignment methods:
  - adversarial alignment
  - correlation alignment
- Many potential applications
Thank you

References

• Eric Tzeng, Judy Hoffman, Trevor Darrell, Kate Saenko, Simultaneous Deep Transfer Across Domains and Tasks, ICCV 2015

• Eric Tzeng, Coline Devin, Judy Hoffman, Chelsea Finn, Pieter Abbeel, Sergey Levine, Kate Saenko, Trevor Darrell. Adapting Deep Visuomotor Representations with Weak Pairwise Constraints, WAFR 2016

• Baochen Sun, Jiashi Feng, Kate Saenko, Return of Frustratingly Easy Domain Adaptation, AAAI 2016

• Baochen Sun, Kate Saenko, Deep CORAL: Correlation Alignment for Deep Domain Adaptation, TASK-CV Workshop at ICCV 2016

• Adversarial Discriminative Domain Adaptation, in submission

• Synthetic to Real Adaptation with Deep Generative Correlation Alignment Networks, in submission