Partial Transfer Learning with Selective Adversarial Networks

Zhangjie Cao\textsuperscript{1}, Mingsheng Long\textsuperscript{1}, Jianmin Wang\textsuperscript{1}, and Michael I. Jordan\textsuperscript{2}

\textsuperscript{1}KLiss, MOE; School of Software, Tsinghua University, China
\textsuperscript{1}National Engineering Laboratory for Big Data Software
\textsuperscript{2}University of California, Berkeley, Berkeley, CA, USA

IEEE Conference on Computer Vision and Pattern Recognition
CVPR 2018 (Spotlight)
Deep Transfer Learning

- Deep learning across domains of different distributions $P \neq Q$

**Source Domain**

- 2D Renderings
- Real Images

**Target Domain**

- 2D Renderings
- Real Images

Model

$P(x,y) \neq Q(x,y)$

Representation


Model
## Deep Transfer Learning: Why?

<table>
<thead>
<tr>
<th>Training Set</th>
<th>Train-Dev Set</th>
<th>Dev Set</th>
<th>Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Error high?</td>
<td>Yes</td>
<td>Bias</td>
<td>Deeper Model</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Variance</td>
<td>Longer Training</td>
</tr>
<tr>
<td>Train-Dev Error high?</td>
<td>Yes</td>
<td>Dataset Shift</td>
<td>Bigger Data</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Overfit Dev Set</td>
<td>Regularization</td>
</tr>
<tr>
<td>Dev Error high?</td>
<td>Yes</td>
<td></td>
<td>Transfer Learning</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td></td>
<td>Data Generation</td>
</tr>
<tr>
<td>Test Error high?</td>
<td>Yes</td>
<td></td>
<td>Bigger Dev Data</td>
</tr>
</tbody>
</table>

Andrew Ng. The Nuts and Bolts of Building Applications using Deep Learning. NIPS 2016 Tutorial.
Partial Transfer Learning

- Deep learning across domains with different label spaces $C_s \supset C_t$
- Positive transfer across domains in shared label space $P_{C_t} \neq Q_{C_t}$
- Negative transfer across domains in outlier label space $P_{C_s \setminus C_t} \neq Q_{C_t}$
Partial Transfer Learning: How?

Matching distributions across the source and target domains s.t. $P \approx Q$
- Reduce marginal distribution mismatch: $P(X) \neq Q(X)$
- Reduce conditional distribution mismatch: $P(Y|X) \neq Q(Y|X)$

**Kernel Embedding**

**Adversarial Learning**

**Selective Adversarial Networks**

- $f = G_f(x)$: feature extractor
- $\hat{y}$: predicted data label
- $\hat{d}$: predicted domain label
- $G_y, L_y$: label predictor and loss
- $G_d^k, L_d^k$: domain discriminator
- GRL: gradient reversal layer

**Diagram:**

- $x$ is fed into a CNN to produce $f$.
- $f$ is then passed through $G_y$ to predict $\hat{y}$.
- $f$ is also passed through $G_d^k$ to predict $\hat{d}^k$.
- $\hat{y}$ and $\hat{d}^k$ are used in the loss functions $L_y$ and $L_d^k$.
- Back-propagation is used to update the parameters of $f$, $G_y$, and $G_d^k$.

---

Z. Cao et al. (Tsinghua University)
Selective Adversarial Networks

Instance Weighting (IW): probability-weighted loss for $G_d^k, k = 1, \ldots, |C_s|$

$$L'_d = \frac{1}{n_s + n_t} \sum_{k=1}^{|C_s|} \sum_{x_i \in D_s \cup D_t} \hat{y}_i^k L_d^k \left( G_d^k \left( G_f \left( x_i \right) \right), d_i \right)$$  (1)
Selective Adversarial Networks

Class Weighting (CW): down-weigh $G^k_d$, $k = 1, \ldots, |C_s|$ for outlier classes

$$L_d = \frac{1}{n_s + n_t} \sum_{k=1}^{\left|C_s\right|} \left\{ \left( \frac{1}{n_t} \sum_{x_i \in D_t} \hat{y}_k^i \right) \times \left( \sum_{x_i \in (D_s \cup D_t)} \hat{y}_k^i L_d^k \left( G^k_d \left( G_f (x_i) \right), d_i \right) \right) \right\}$$  \hspace{1cm} (2)
Selective Adversarial Networks

Entropy (uncertainty) minimization: $H(G_y(G_f(x_i))) = -\sum_{k=1}^{\left|C_s\right|} \hat{y}_i^k \log \hat{y}_i^k$

$E = \frac{1}{n_t} \sum_{x_i \in D_t} H(G_y(G_f(x_i)))$ (3)
Selective Adversarial Networks

\[ C \left( \theta_f, \theta_y, \theta_d^k \mid |C_s| \right) = \frac{1}{n_s} \sum_{x_i \in D_s} L_y (G_y (G_f (x_i)), y_i) + \frac{1}{n_t} \sum_{x_i \in D_t} H (G_y (G_f (x_i))) \]

\[ - \frac{1}{n_s + n_t} \sum_{k=1}^{\mid C_s \mid} \left\{ \left( \frac{1}{n_t} \sum_{x_i \in D_t} \hat{y}_i^k \right) \times \left( \sum_{x_i \in (D_s \cup D_t)} \hat{y}_i^k L_d^k \left( G_d^k (G_f (x_i)), d_i \right) \right) \right\} \]

(4)

(\hat{\theta}_f, \hat{\theta}_y) = \arg \min_{\theta_f, \theta_y} C \left( \theta_f, \theta_y, \theta_d^k \mid |C_s| \right)

(5)

(\hat{\theta}_d^1, ... , \hat{\theta}_d^{|C_s|}) = \arg \max_{\theta_d^1, ... , \theta_d^{|C_s|}} C \left( \theta_f, \theta_y, \theta_d^k \mid |C_s| \right)
**Transfer Tasks:** Office-31 (31 → 10), Caltech-Office (256 → 10) and ImageNet-Caltech (1000 → C84 and C256 → I84)
## Results

### Office-31

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A 31 → W 10</td>
<td>D 31 → W 10</td>
<td>W 31 → D 10</td>
<td>A 31 → D 10</td>
<td>D 31 → A 10</td>
<td>W 31 → A 10</td>
<td>Avg</td>
<td></td>
</tr>
<tr>
<td>AlexNet [2]</td>
<td>58.51</td>
<td>95.05</td>
<td>98.08</td>
<td>71.23</td>
<td>70.6</td>
<td>67.74</td>
<td>76.87</td>
<td></td>
</tr>
<tr>
<td>DAN [3]</td>
<td>56.52</td>
<td>71.86</td>
<td>86.78</td>
<td>51.86</td>
<td>50.42</td>
<td>52.29</td>
<td>61.62</td>
<td></td>
</tr>
<tr>
<td>RevGrad [1]</td>
<td>49.49</td>
<td>93.55</td>
<td>90.44</td>
<td>49.68</td>
<td>46.72</td>
<td>48.81</td>
<td>63.11</td>
<td></td>
</tr>
<tr>
<td>RTN [4]</td>
<td>66.78</td>
<td>86.77</td>
<td>99.36</td>
<td>70.06</td>
<td>73.52</td>
<td>76.41</td>
<td>78.82</td>
<td></td>
</tr>
<tr>
<td>ADDA [5]</td>
<td>70.68</td>
<td>96.44</td>
<td>98.65</td>
<td>72.90</td>
<td>74.26</td>
<td>75.56</td>
<td>81.42</td>
<td></td>
</tr>
<tr>
<td>SAN-selective</td>
<td>71.51</td>
<td>98.31</td>
<td>100.00</td>
<td>78.34</td>
<td>77.87</td>
<td>76.32</td>
<td>83.73</td>
<td></td>
</tr>
<tr>
<td>SAN-entropy</td>
<td>74.61</td>
<td>98.31</td>
<td>100.00</td>
<td>80.29</td>
<td>78.39</td>
<td>82.25</td>
<td>85.64</td>
<td></td>
</tr>
<tr>
<td>SAN</td>
<td>80.02</td>
<td>98.64</td>
<td>100.00</td>
<td>81.28</td>
<td>80.58</td>
<td>83.09</td>
<td>87.27</td>
<td></td>
</tr>
</tbody>
</table>

### Caltech-Office

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C 256 → W 10</td>
<td>C 256 → A 10</td>
<td>C 256 → D 10</td>
<td>Avg</td>
<td>C 256 → C 84</td>
<td>C 256 → I 84</td>
<td>Avg</td>
<td></td>
</tr>
<tr>
<td>AlexNet [2]</td>
<td>58.44</td>
<td>76.64</td>
<td>65.86</td>
<td>66.98</td>
<td>52.37</td>
<td>47.35</td>
<td>49.86</td>
<td></td>
</tr>
<tr>
<td>DAN [3]</td>
<td>42.37</td>
<td>70.75</td>
<td>47.04</td>
<td>53.39</td>
<td>54.21</td>
<td>52.03</td>
<td>53.12</td>
<td></td>
</tr>
<tr>
<td>RevGrad [1]</td>
<td>54.57</td>
<td>72.86</td>
<td>57.96</td>
<td>61.80</td>
<td>51.34</td>
<td>47.02</td>
<td>49.18</td>
<td></td>
</tr>
<tr>
<td>RTN [4]</td>
<td>71.02</td>
<td>81.32</td>
<td>62.35</td>
<td>71.56</td>
<td>63.69</td>
<td>50.45</td>
<td>57.07</td>
<td></td>
</tr>
<tr>
<td>ADDA [5]</td>
<td>73.66</td>
<td>78.35</td>
<td>74.80</td>
<td>75.60</td>
<td>64.20</td>
<td>51.55</td>
<td>57.88</td>
<td></td>
</tr>
<tr>
<td>SAN-selective</td>
<td>76.44</td>
<td>81.63</td>
<td>80.25</td>
<td>79.44</td>
<td>66.78</td>
<td>51.25</td>
<td>59.02</td>
<td></td>
</tr>
<tr>
<td>SAN-entropy</td>
<td>72.54</td>
<td>78.95</td>
<td>76.43</td>
<td>75.97</td>
<td>55.27</td>
<td>52.31</td>
<td>53.79</td>
<td></td>
</tr>
<tr>
<td>SAN</td>
<td>88.33</td>
<td>83.82</td>
<td>85.35</td>
<td>85.83</td>
<td>68.45</td>
<td>55.61</td>
<td>62.03</td>
<td></td>
</tr>
</tbody>
</table>
(a) Accuracy w.r.t #Target Classes

(b) Test Error

- SAN outperforms RevGrad even more for larger class-space difference
- SAN converges more stably and fast to lower test error than RevGrad
Visualization

Figure: t-SNE with class information (top) and domain information (bottom).
References

Domain-adversarial training of neural networks.

Imagenet classification with deep convolutional neural networks.
In *NIPS*, 2012.

M. Long, Y. Cao, J. Wang, and M. I. Jordan.
Learning transferable features with deep adaptation networks.
In *ICML*, 2015.

Unsupervised domain adaptation with residual transfer networks.

Adversarial discriminative domain adaptation.
Summary

- A novel selective adversarial network for partial transfer learning
  - Circumvent negative transfer by selecting out outlier source classes
  - Promote positive transfer by matching shared-class-space distributions

- Code will be available soon at: https://github.com/thuml/

- A work at CVPR 2018 follows our arXiv version: how fast they are!