A Generative Framework for Zero Shot Learning with Adversarial Domain Adaptation

Homanga and Varun

Computer Science, IIT Kanpur

2 May, 2019

Motivation

- In conventional image classification tasks, examples from all classes are available during training of the model
- This assumption rarely holds in real world problems.
- There exist a plethora of real world concepts that do not have the corresponding ubiquity of representative images.
- It is common knowledge that humans do not require prior visual evidence of a category to recognize an example from that category.

Motivation

(a) A Zebra

(b) A Horse

Figure 1: Given the image of a horse and a description about zebra's appearance, a child usually has no trouble identifying a zebra when he/she sees one.

Highlights

- Generative Zero-Shot Learning (ZSL) Framework
- Adversarial Domain Adaptation (ADA) to minimize domain gap between the ”actual” and the ”generated” distributions
- End-to-end training in the Generative Framework
- Enforcement of cyclic consistency in a latent space for ADA

Proposed Approach

- Generative Framework
  - We model the data distribution as a mixture of individual class conditional distributions:

$$\mathbf{x} \sim p(\mathbf{x}|\Theta_c)\ \forall c \in \mathcal{C}$$

  - Classify an unknown test sample $\mathbf{x}_+$ by

$$\hat{y}_+ = \operatorname*{argmax}_c p(c|\mathbf{x}_+, \Theta)$$

  - Once Θ is learnt new samples from a particular class can be generated by simply sampling from the above distribution
- Domain Adaptation
  - We use adversarial domain adaptation to adjust the model parameters Θ_c ($\forall c \in \mathcal{C}^{unseen}$) to mimic the test data distribution
  - This improves the classification accuracy by many folds

Generative Framework

- Need to infer parameters $\{\Theta_c\}$ from the class descriptions $\mathbf{a}_c$ which explain the given data well.
- Define a mapping $f : \mathcal{A} \rightarrow \Theta$ then $\Theta_c = f_\eta(\mathbf{a}_c)$ and objective becomes,

$$\underset{\Theta}{\text{argmax}}\ \mathbb{P}[\mathbf{X}^{S \cup U} | \Theta] = \underset{\eta}{\text{argmax}}\ \mathbb{P}[\mathbf{X}^{S \cup U} | f_\eta(\mathcal{A})]$$

- Upon simplification and taking *log*() we get

$$\underset{\eta}{\text{argmax}} \underset{\mathbf{x} \sim S, c \in S}{\mathbb{E}} log\ p(\mathbf{x} | f_\eta(\boldsymbol{a}_c))$$

Comparison with Other Frameworks (GZSL)

	SUN			CUB			AWA2		
Method	**U $\rightarrow$ S+U**	**S $\rightarrow$ S+U**	**H**	**U $\rightarrow$ S+U**	**S $\rightarrow$ S+U**	**H**	**U $\rightarrow$ S+U**	**S $\rightarrow$ S+U**	**H**
CONSE	6.8	39.9	11.6	1.6	**72.2**	3.1	0.5	**90.6**	1.0
CMT*	8.1	21.8	11.8	7.2	49.8	12.6	0.5	90.0	1.0
SSE	2.1	36.4	4.0	8.5	46.9	14.4	8.1	82.5	14.8
SJE	14.7	30.5	19.8	23.5	59.2	**33.6**	8.0	73.9	14.4
ESZSL	11.0	27.9	15.8	12.6	**63.8**	21.0	5.9	77.8	11.0
SYNC	7.9	**43.3**	13.4	11.5	70.9	19.8	10.0	90.5	18.0
SAE	8.8	18.0	11.8	7.8	54.0	13.6	1.1	82.2	2.2
LATEM	14.7	28.8	19.5	15.2	57.3	24.0	11.5	77.3	20.0
DEVISE	16.9	27.4	20.9	**23.8**	53.0	32.8	17.1	74.7	27.8
W/O ADA (Ours)	**19.4**	38.6	**25.8**	19.5	58.5	29.3	**30.3**	81.5	**44.2**

Table 1: Accuracy for GZSL, on proposed split(PS). U and S represents top-1 accuracy on unseen and seen class. H: Harmonic mean.

Comparison with Other Frameworks (ZSL)

METHOD	SUN	CUB	AWA2
IAP	19.4	24.0	35.9
CONSE	38.8	34.3	44.5
CMT	39.9	34.6	37.9
ESZSL	54.5	53.9	58.6
SAE	40.3	33.3	54.1
DEM	61.9	51.7	67.1
GFZSL	63.1	49.2	67.0
CVAE-ZSL	61.7	52.1	65.8
W/O ADA (Ours)	63.3	53.5	70.3
With ADA (Ours)	**65.4**	**56.7**	**77.2**

Table 2: Zero Shot Learning Accuracy on the SUN, CUB, and AWA2 dataset. The train-test split is is the proposed split as defined in a recent paper

Adversarial Domain Adaptation

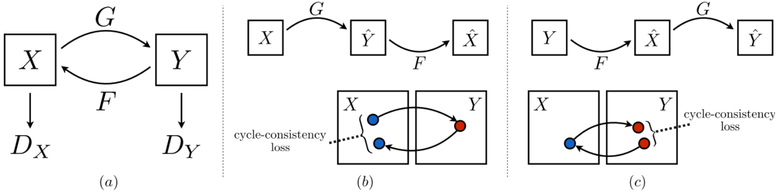

Figure 2: The high-level procedure of CycleGAN

- In the subsequent slides, for describing our method, we denote G as G_η^T and F as G_η^S. X and Y respectively correspond to the source S and target T distributions.

Adversarial Domain Adaptation

- Pre-training: The generative model is learned end-to-end during pre-training.
- Through ADA, we aim to bring the target distribution of $\{y_n\}_c$ closer to the source distribution $\{x_n\}_c$.
- Learn $G^T_\eta(\{y_n\}_c)$ that maps class conditionals from the generated distribution $\{y_n\}_c$ to a common latent space as that from the real test distribution$\{x_n\}_c$ for all classes $\{c\}_{c=1}^{S+U}$
- $\{\mathbf{x}_n\}_c$ denotes the entire test data comprising of samples from all the classes. Similarly, $\{y_n\}_c$ denotes the samples generated from the generative model for all the classes.
- Two functions learnt to enforce cyclic consistency: $G^T_\eta : S \to T$, $G^S_\eta : T \to S$

Adversarial Domain Adaptation

- Generator: $G_\eta^T(\{y_n\}_c)$ such that $\{y_n\}_c \sim N(f_\mu(a_c), di(f_\Sigma(a_c)))$
- Generator's Loss: $L_D^T = \mathbb{E}_{c \sim p_c}[D_w^T(G_\eta(\{y_n\}_c))] - \mathbb{E}_{\hat{c} \sim p_{\hat{c}}}[D_w^T(\{x_n\}_{\hat{c}})]$
- The Discriminator D takes as input features generated by the Generator G of the target domain inputs (fake) and features from the examples in the source domain (real).
- Discriminator's Loss: $L_D^T = \mathbb{E}_{c \sim p_c}[D_w^T(G_\eta(\{y_n\}_c))] - \mathbb{E}_{\hat{c} \sim p_{\hat{c}}}[D_w^T(\{x_n\}_{\hat{c}})]$

Adversarial Domain Adaptation

- Hence, we enforce the mappings G_η^T and G_η^S to be cycle consistent by enforcing forward ($S \to T$) and backward ($T \to S$) cycle-consistency between the source and target domains.
- This behavior is encouraged through a cycle consistency loss:
- $\mathcal{L}_{\mathrm{cy}}(G^T, G^S) = \mathbb{E}_{c \sim p_{\mathrm{c}}}[G^S(G^T(\{y_n\}_c)) - \{x_n\}_{c_p}] + \mathbb{E}_{c \sim p_{\mathrm{c}}}[G^T(G^S(\{x_n\}_c)) - \{y_n\}_{c_p}]$.

Adversarial Domain Adaptation

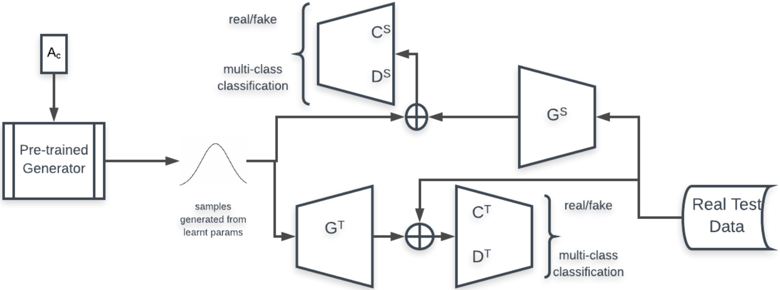

Figure 3: Overall Process of Adversarial Domain Adaptation

Adversarial Domain Adaptation

- The overall optimization objective for adversarial loss:
- $\mathcal{L}_{ov}(G^T, G^S, D^T, D^S) =$
  $\mathcal{L}_{\mathrm{GAN}}(G^T, D^T, S, T, H^T + \mathcal{L}_{\mathrm{GAN}}(G^S, D^S, T, S) +$
  $\xi \mathcal{L}_{\mathrm{C}}(G^S, C^S, T, S) + \xi \mathcal{L}_{\mathrm{C}}(G^T, C^T, S, T) + \chi \mathcal{L}_{\mathrm{cy}}(G^T, G^S),$

Empirical Gains from ADA

Method	SUN	CUB	AWA2
ALE	57.1	54.6	71.3
GFZSL	64.2	50.5	**78.6**
With ADA (Ours)	**65.4**	**56.7**	77.2

Table 3: Transductive Zero-Shot Learning results on the SUN, CUB, and AWA2 dataset. Transductive setting for our model corresponds to ADA.

Visualization of ADA Gains

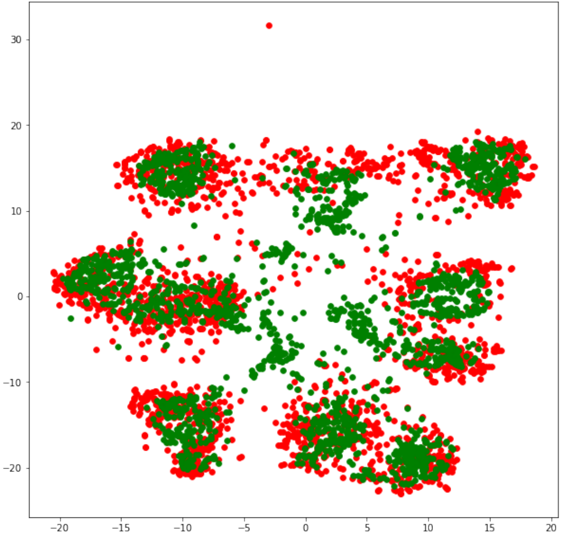

Figure 4: Red-Real Examples from Unseen classes, Green-Generated Examples from Unseen classes

Possible Extensions

- Currently we used pre-trained resnet-101 as a feature extractor.
- We explored the impact of feature learning in this model. Amongst various methods employed, siamese networks with contrastive loss gave accuracies equal to the model without feature learning. Rest of models performed worse.
- t-SNE plot shows that feature learning brought the resnet clusters closer to each other thereby hampering the classification
- We believe this can be improved by better hyper-param optimization but due to paucity of time we weren't able to experiment extensively on this approach.

Resnet Features

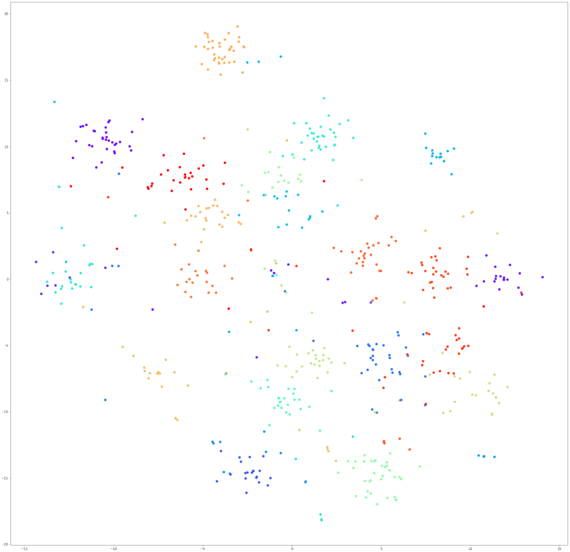

Figure 5: class wise clusters of AWA 2 using t-SNE