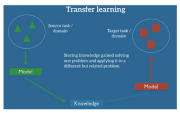
Transferable Adversarial Cycle Alignment for Domain Adaption

Yingcan Wei
The University of Hong Kong
Email/Skype:weiyingcan@gmail.com

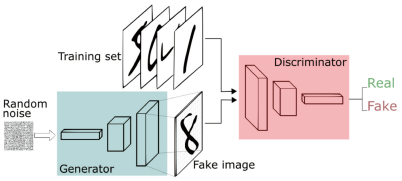
ICANN19
28th International Conference on Artificial Neural Networks

Background

Transfer learning:
• A research problem belonging to machine learning.
• Focus on storing knowledge while modeling one problem and applying it to a different but related problem.
• Obtain an ability of human to learn with help of analogy.
• Learn to walk could be used to learn to run.
• Learn to recognize cars could apply to recognize trucks.



Domain Adaption:
•Learning an invariant model in the presence of a shift between distributions.
• Building a mapping between the domains.
• The model learned for the one domain can be applied to target domain directly.



GAN consists of a generator G and a discriminator D.
• G: synthesize images resembling real images.
• D: distinguish the real from synthesized ones.
• G and D are realized as Deep Neural Networks.

Summary

Cycle-TBiGAN

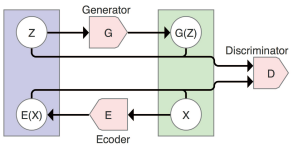
• A novel method (Cycle-TBiGAN) that integrating the Transferable Bi-GAN with Cycle Consistent Constraints by learning a mapping function for an image translation task.
• Aim to reduce even remove the distribution discrepancy between domains and relieved the training instability and model collapse issues.
• Combine with domain alignment objective to transfer the invariant feature representation to a target domain.
• Show the surprising results in the task of image translation without prior ground-truth knowledge.

Yesterday	Today	Tomorrow
Deep Learning	Reinforcement Learning	Transfer Learning
Lots of Data	Lots of Data	FewData
Only the Rich	Only the Rich	Everyone

Transferable Bidirectional Gernerative Adversarial Networks - (TBiGAN)

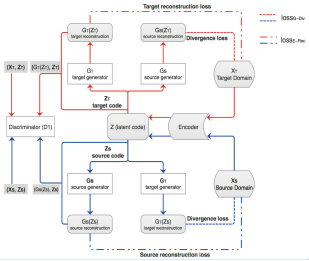
Bidirectional GAN (BiGAN) :
• Feature representation z acts as additional information combines with original and reconstruction images.
• The discriminator distinguishes input from both latent coder and original space.
The objective is defined as:

$$min_{G,E} max_D V(E,G,D)$$
$$= E_{x\sim p_x(x)}[\log D(x,z)] + E_{z\sim p_G(\cdot|Z)}\left[E_{x\sim p_G(\cdot|Z)}[\log(1-D(G(\hat{X},Z))]\right]$$
Where $\hat{X} = G_{z\sim p_z}(Z)$ and $Z = E_{x\sim p_X}(X)$



Transferable BiGANs (TBiGAN) :
• Model the data distribution as a transformation of a latent distribution P(z).
• Transformation is to train domain-specific generators.
• Force the output to P[Gs(x)] = P(Xs) and P[Gt(x)] = P(Xt) .

$$min_{G,E} max_{D1} V(E,G,D1) = E_{x\sim p_x(x)}[logD1(x,E(x))] + E_{z\sim p_z(z)}[1 - logD1(G(z),z)]$$
$$min_{D1} loss_{D1} = max_{D1} V(E,G,D1) = loss_{D1-E} + loss_{D1-G}$$



where $loss_{D1-E}$, $loss_{D1-G}$ defined as:

$$loss_{D1-E} = L_b\left(D1(x,E(x)),1\right)$$
$$loss_{D1-G} = L_b\left(D1(z,G(z)),0\right)$$
$$min_E loss_E = min_E V(E,G,D1) = (loss_{D1-E} + loss_{E-Rec})$$
$$min_G loss_G = min_G V(E,G,D1) = (loss_{D1-G} + loss_{G-Div})$$

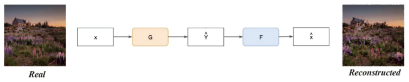
where $loss_{E-Rec}$ and $loss_{G-Div}$ defined as:

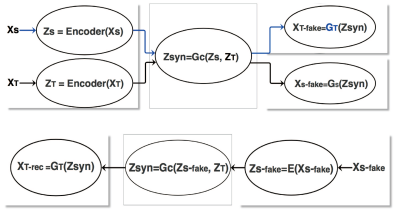
$$loss_{E-Rec} = \begin{cases} L_1\left(x_s, G_S(E(x_s))\right) \\ L_1\left(x_t, G_T(E(x_t))\right) \end{cases}$$
$$loss_{G-Div} = \begin{cases} L_d\left(x_s, G_T(E(x_s))\right) \\ L_d\left(x_t, G_S(E(x_t))\right) \end{cases}$$

Cycle Consistency Alignment Transformation - (Cycle-TBiGAN)

Cycle Consistent GAN

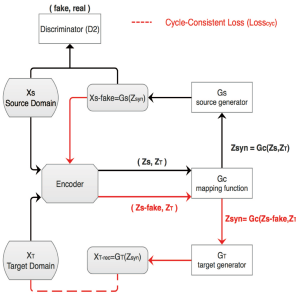
• Treat the original problem as image reconstruction.
• Using a deep network G to convert image x to y.
• Reverse the G with another deep network F to reconstruct the image.





Cycle Alignment for Transferable Bidirectional GAN

• Cycle-TBiGAN relies on pre-trained generators from source and target domain, an invariant feature space learned from TBiGAN.
• The objective is jointly to learn a transformation function Gc (·), which maps the latent coder of Zs and Zt to a synthesized code Zsyn.



• Synthesize a fake source image Xs-fake corresponding to target input image XT.

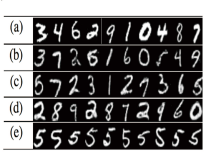
$$min_{D2} loss_{D2} = L_b\left(D2(G_S(Z_{syn})),0\right) + L_b(D2(X_S),1)$$
$$min_{G_C}\ loss_{G_C} = L_b\left(D2(G_S(Z_{syn})),1\right) + loss_{cyc}$$

Where $L_b(\cdot)$ is a binary cross-entropy loss, $loss_{cyc}$ is defined in

$$min\ loss_{cyc} = L_p(X_T,\ X_{T-rec})$$
$$X_{T-rec} = G_T\left(G_C(E(X_{S-fake}), E(X_T))\right)$$

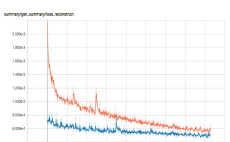
• Lp is "Perceptual Loss", a predefined image quality measurement, such as Laplacian pyramid, or L1 loss.

Experiment

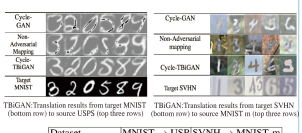


TBiGAN: Example results for MNIST are reconstructed by the different frameworks' Generator at the different training epochs. (a) TBiGAN (40 epochs). (b) GAN (300 epochs). (c) BiGAN (400 epochs). (d) Cycle-GAN (200 epochs). (e) Real MNIST.

TBiGAN :Example results for MNIST m that reconstructed by the different rameworks' Generator at same training epoch (50 epochs). (a) TBiGAN. (b)GAN. (c) BiGAN. (d) Cycle-GAN. (e) Real MNIST_m.



TBiGAN: Comparison reconstruction loss(MNIST_m) between training from scratch (orange) and training on pre-trained Mnist model(blue)



TBiGAN:Translation results from target MNIST (bottom row) to source USPS (top three rows)

TBiGAN:Translation results from target SVHN (bottom row) to source MNIST m (top three rows)

Dataset	MNIST → USP	SVNH → MNIST_m
Cycle-GAN	0.077± 0.05	0.105± 0.05
Non-Adversarial	0.089± 0.02	0.153± 0.02
Cycle-TBiGAN	**0.053**± 0.01	**0.087**± 0.02

TBiGAN: Image-to-image median pixel difference from linear comparison (lower is better)