AdvAE and FlowAE : Sampling Arbitrary Latent Variable Distributions

Srivatsan Sridhar Varun Srivastava

Deep Generative Models (CS 236) - Stanford University

Motivation

- In a VAE, latent variable $\boldsymbol{z}$ is from a simple (usually Gaussian) prior $p_\theta(\boldsymbol{z})$
- VAE training minimizes KL divergence of posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ with prior $p_\theta(\boldsymbol{z})$
- Simple prior may be less expressive, and may not match the data distribution

Goals of the Project

- To train an autoencoder to learn an **arbitrarily distributed** latent variable $\boldsymbol{z} = f_\theta(\boldsymbol{x})$
- To **sample** from the arbitrary latent variable distribution to generate samples $\boldsymbol{x} = f_\phi(\boldsymbol{z})$
- To **estimate the density** $p(\boldsymbol{z})$ of the arbitrary latent variable distribution
- To observe characteristics of the arbitrarily distributed latent space

Training and Experiments

- The autoencoder is first trained with a reconstruction loss, and then fixed
- FlowAE/AdvAE generator is trained next
- Compare MNIST samples from a standard VAE, AAE, FlowAE and AdvAE
- **Frechet Classifier Distance** using last layer activations of an MNIST classifier
- Visualize the latent space using t-SNE

Previous Works

Adversarial Autoencoder (AAE)

- Adversarial training to match the posterior of the latent variable with a known prior
- Trains the autoencoder on reconstruction loss plus adversarial loss
- Latent variable sampled from known prior

Our Methods - AdvAE and FlowAE

1. Autoencoder (AE)

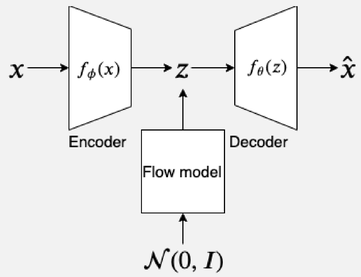
- Encoder $\boldsymbol{z} = f_\theta(\boldsymbol{x})$ and Decoder $\boldsymbol{x} = f_\phi(\boldsymbol{z})$
- Different from VAE which learns parameterized distributions $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ and $q_\phi(\boldsymbol{x}|\boldsymbol{z})$

$$\mathscr{L}_{AE}(x, \hat{x}) = x \log(\hat{x}) - (1-x)\log(1-\hat{x})$$

2. Flow Network + AE (FlowAE)

- Flow network $F(\mathbf{z_0})$ generates latent variable $\boldsymbol{z}$
- Maximum likelihood on $\boldsymbol{z} = f_\theta(\boldsymbol{x})$ from real data

$$\mathscr{L}_F(\mathbf{z}) = -\log \mathcal{N}(F^{-1}(\mathbf{z}); 0, \mathrm{I}) + \log \left| \frac{\partial f^{-1}(z)}{\partial z'} \right|$$

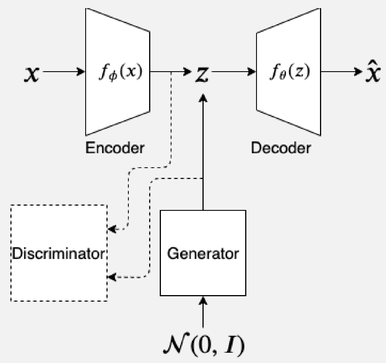


3. Adversarially trained AE (AdvAE)

- Generator $G(\mathbf{z_0})$ generates latent variable $\boldsymbol{z}$
- Discriminator $D(\boldsymbol{z})$ classifies $\boldsymbol{z} = G(\mathbf{z_0})$ from generator and $\boldsymbol{z} = f_\theta(\boldsymbol{x})$ from real data
- Alternate training of discriminator and generator using WGAN-GP loss

$$\mathscr{L}_D = -D(f_\theta(\boldsymbol{x})) + D(G(\mathbf{z_0})) + \lambda(\|\nabla D\|_2 - 1)^2$$

$$\mathscr{L}_G = -D(G(\mathbf{z_0}))$$



Results

Figure: VAE

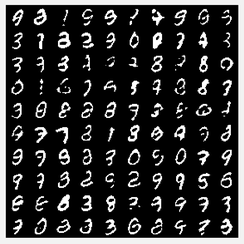


Figure: AAE

Figure: AdvAE

Figure: FlowAE

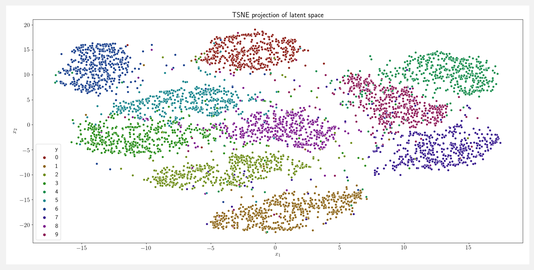


Figure: t-SNE on latent space of VAE (*Perplexity: 70*)

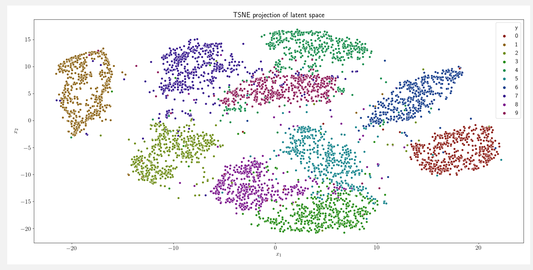


Figure: t-SNE on latent space of FlowAE, AdvAE (*Perplexity: 70*)

Results

Frechet Classifier Distance Scores

Model	FCD
Real Data	0.2
FlowAE	**4.65**
AdvAE	6.03
AAE	6.09
VAE	8.12

Conclusion

Decoupling the tasks of sampling the latent space and learning a flexible class of distributions over the latent space leads to both increase in sample quality, and simplified training procedures (in FlowAE). The t-SNE plots suggest that the latent space can be clustered by its labels.

Further Work

- Convolutional Autoencoders on CIFAR-10
- Convolutional Generators and flow networks for the latent space

References

[1] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey.
Adversarial autoencoders.
2015.

[2] Danilo Jimenez Rezende and Shakir Mohamed.
Variational inference with normalizing flows.
2015.

[3] Laurens van der Maaten and Geoffrey Hinton.
Visualizing data using t-sne.
2008.

[4] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner.
Understanding disentangling in β-vae, 2018.