AUTO-ENCODER INTERPOLATION

Using Adversarial Constrained Autoencoder Interpolation As A Case Study

Sulaimon and Alex

University of Houston

May 4, 2020

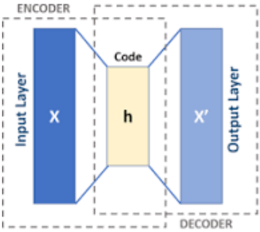
Outline

- What is the task?
- Datasets
- Adversarial Networks
- What model was used?
- How do you control the interpolatd datasets?
- Any results?
- Conclusion

What is the task?

Goal: Generating meaningful images of interpolated data-points in hidden layers.

Claim: a good interpolation should both reveal the hidden structure of the dataset, as well as be smooth and follow the true data distribution, i.e. produce realistic elements.



Datasets

What Datasets was it used on?

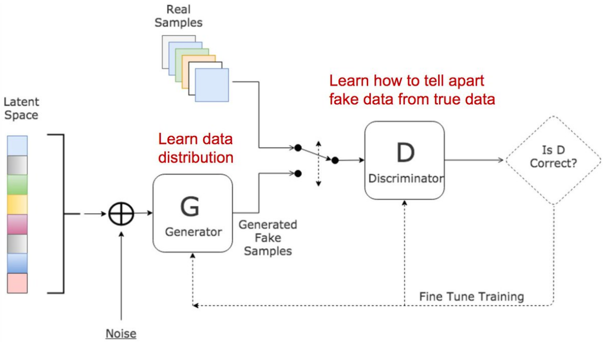
- MNIST
- Cifar10

What Datasets do we wish to try it on in the future?

- Fashion MNIST
- ImageNet
- lsun
- SVHN

Adversarial Networks

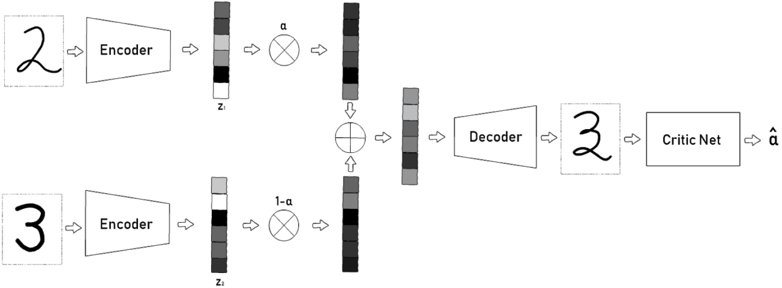
This model uses what is called an Adversary(Discriminator) Network to act against or upon another network. Generative Adversarial Networks, or GANs, are generative model comprising of a generator and a discriminator.



GANs are based on a game scenario in which the generator network must compete against an adversary. The generator network directly produces samples. Its adversary, attempts to distinguish between samples drawn from the training data and samples drawn from the generator.

What model was used?

A special kind of auto-encoder called **Adversarially Constrained Autoencoder Interpolation (ACAI)** was used. Below is the architecture of this model:



Is ACAI an example of a Generative Adversarial Networks?

How do control the interpolated datasets?

Let $X \subset \mathbb{R}^N$ be a dataset. Consider the case when we are given a manifold model for X, which consists of a decoder D from the latent $Z = \mathbb{R}^D$ to the input data space: $G : Z \to \mathbb{R}^N$.

We define the realism index on Z as a function $ri : Z \to [0, 1]$ such that high values of $ri(z)$ indicate that $G(z)$ is indistinguishable from the elements of X. In general, the optimal choice of ri can be nontrivial, and may depend on the generative model in question.

Let $\mathbb{X}$ be a random vector in Z, we define the realism index ri based on the function f by the formula

$$ri(z; f) := p(f(\mathbb{X}) \leq f(z)) = \int_{\{x : f(x) \leq f(z)\}} f(s) ds$$

where p denotes the probability

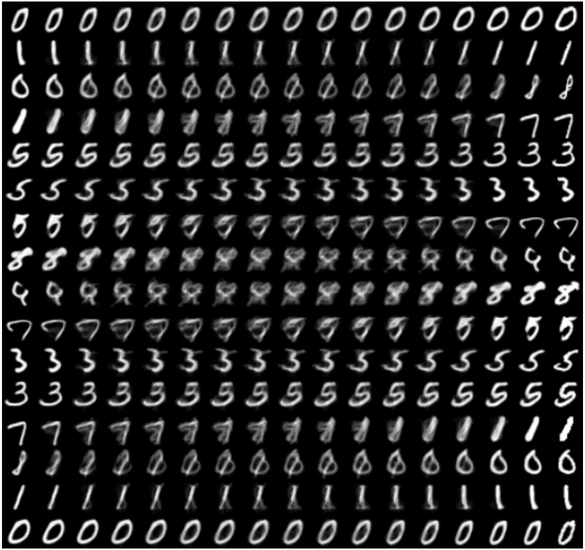
Any results?

Below is the table of single layer classifier accuracy achieved by different autoencoders:

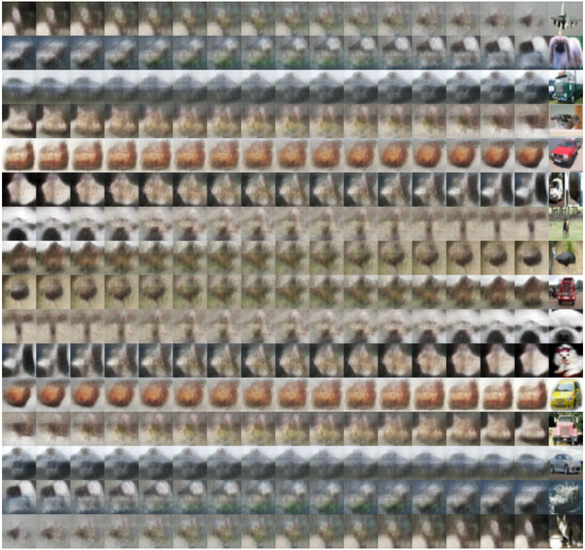
Dataset	d_z	Baseline	Dropout	Denoising	VAE	AAE	VQ-VAE	ACAI
MNIST	32	94.90±0.14	96.45±0.42	**96.00±0.27**	96.56±0.31	70.74±3.27	97.50±0.18	**98.25±0.11**
	256	93.94±0.13	94.50±0.29	**98.51±0.04**	98.74±0.14	90.03±0.54	97.25±1.42	**99.00±0.08**
SVHN	32	26.21±0.42	26.09±1.48	**25.15±0.78**	29.58±3.22	23.43±0.79	24.53±1.33	**34.47±1.14**
	256	22.74±0.05	25.12±1.05	**77.89±0.35**	66.30±1.06	22.81±0.24	44.94±20.42	**85.14±0.20**
CIFAR-10	256	47.92±0.20	40.99±0.41	**53.78±0.36**	47.49±0.22	40.65±1.45	42.80±0.44	52.77±0.45
	1024	51.62±0.25	49.38±0.77	**60.65±0.14**	51.39±0.46	42.86±0.88	16.22±12.44	**63.99±0.47**

What does our model produce when used on the datasets we tried it on?

On MNIST



On Cifar10



Conclusion

ACAI is leading in improved performance for feature learning and unsupervised clustering.

In future work, we are interested in investigating whether our regularizer improves the performance of autoencoders other than the standard autoencoder we applied it to.

Also, we focused was on image datasets due to the ease of visualizing interpolations, but we are also interested in applying these ideas to non-image datasets.

THANK YOU