Understanding and Improving Interpolation in Autoencoders via an Adversarial Regularizer

Discussion Lead: Mahsa Rahimi
Discussion Facilitator: Andrew Martin, Armando Benitez

Overview

Basic concepts
Autoencoders
Interpolation

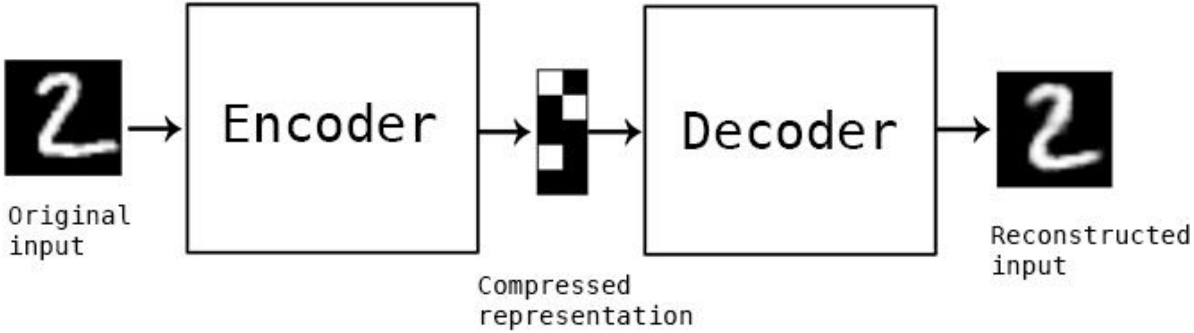
Model
Adversarially Constrained Autoencoder Interpolation (ACAI)
Benchmark task
Interpolation metrics

Experiments Results
Benchmark synthetic data
Real data

Takeaways

Discussion points

Autoencoders



$$x \in \mathbb{R}^{d_x} \Longrightarrow f_\theta \Longrightarrow \begin{array}{c} z = f_\theta(x) \\ z \in \mathbb{R}^{d_z} \end{array} \Longrightarrow g_\phi \Longrightarrow \begin{array}{c} \hat{x} = g_\phi(z) \\ \hat{x} \in \mathbb{R}^{d_x} \end{array}$$

Objective: $||x - \hat{x}||^2$

Interpolation

https://hackernoon.com/latent-space-visualization-deep-learning-bits-2-bd09a46920df

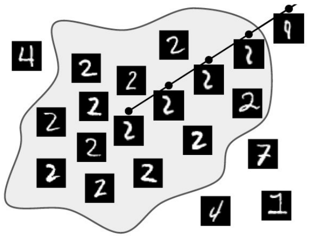
Typically convex combination of the two latent codes

$$\hat{x}_\alpha = g_\phi(\alpha z_1 + (1-\alpha) z_2)$$

$$\alpha \in [0, 1]$$

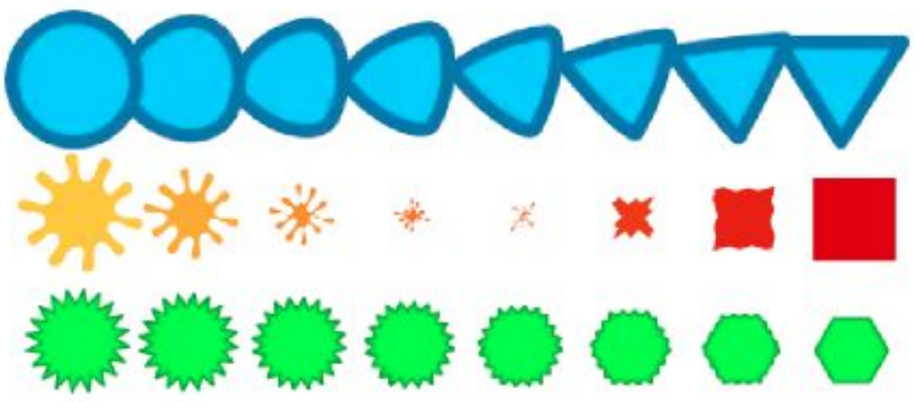
Why do we want to look at interpolations?

- if we're able to smoothly interpolate through the latent space then the autoencoder has uncovered structure in the data.



- If the interpolation is smooth and well resolved it can have useful **creative** applications

Ideal interpolation



https://inkscapetutorials.wordpress.com/2008/01/30/official-inkscape-tutorial-using-the-interpolation-extension/

$$\hat{x}_\alpha = g_\phi(\alpha z_1 + (1-\alpha) z_2)$$

$$\alpha \in [0, 1]$$

How do we measure the quality of an interpolation?

- Interpolation is a somewhat ill-defined concept: It relies on the notion of a “semantically meaningful combination” which is problem-dependent and vague.

- None of the objectives or structures used for autoencoders explicitly enforce good interpolation.

The purpose of this work

To formalize and improve interpolation in autoencoders and to test whether simply encouraging better interpolation behavior produces a better representation for downstream tasks with the following contributions:

- Proposing an adversarial regularization strategy which explicitly encourages high-quality interpolations in autoencoders
- Developing a simple benchmark where interpolation is well-defined and quantifiable
- Quantitatively evaluating the ability of common autoencoder models to achieve effective interpolation and show that the proposed regularizer exhibits superior interpolation behavior
- Showing that regularizer improves the performance of autoencoders

Adversarially Constrained Autoencoder Interpolation (ACAI)

A high-quality interpolation should have two characteristics:

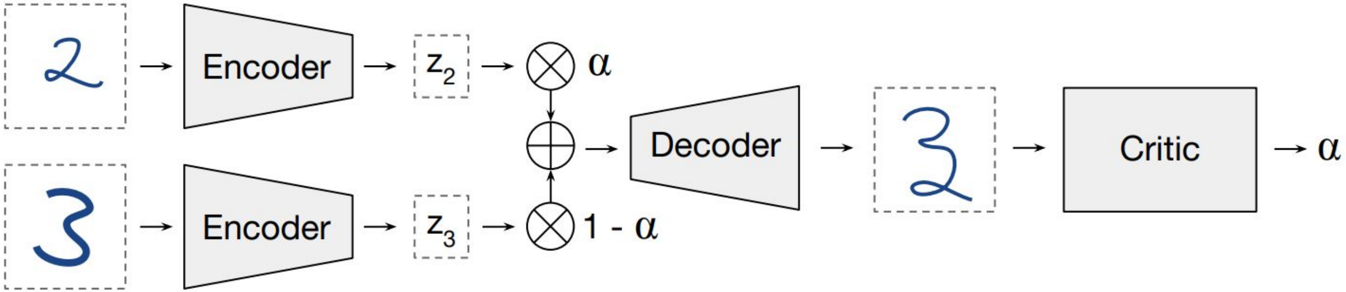
- Intermediate points along the interpolation are indistinguishable from real data.

- Intermediate points provide a semantically smooth morphing between the endpoints.

Semantically smooth morphing is hard to enforce and explicitly codify:

They propose a regularizer constraint results in realistic and smooth interpolations in practice.

Autoencoder and critic model with ACAI



$$\hat{x}_\alpha = g_\phi(\alpha z_1 + (1-\alpha) z_2)$$

Model Objectives

Discriminator $d_\omega(x)$

$$\mathcal{L}_d = \|d_\omega(\hat{x}_\alpha) - \alpha\|^2 + \|d_\omega(\gamma x + (1-\gamma) g_\phi(f_\theta(x))\|^2$$

γ Is a hyperparameter

$$\hat{x} = g_\phi(f_\theta(x))$$

First term: recovering α from xˆα.

Second term serves as a regularizer:

- Enforces that the critic consistently outputs 0 for non-interpolated inputs.
- Ensures the critic is exposed to realistic data even when the AE generator reconstructions are poor.

The 2nd part was helpful for stabilizing the learning process but it was not crucial for their approach

Generator

$$\mathcal{L}_{f,g} = \|x - g_\phi(f_\theta(x))\|^2 + \lambda \|d_\omega(\hat{x}_\alpha)\|^2$$

2nd term: The autoencoder is trained to fool the critic to think that α is always zero.

Results:

- Encouraging this behavior also produces semantically smooth interpolations
- Providing improved learning performance

How can measure whether ACAI improves autoencoders interpolations?

A good Benchmark Task should:

- Be well defined, unambiguous and give the “corrects” interpolation between two data points
- Allow us to quantitatively evaluate successful interpolation

Benchmark: Autoencoding lines

Benchmark Task: Autoencoding 32 × 32 greyscale images of lines.

Data: 16-pixel-long lines

Data manifold: It can be defined entirely by a single variable: Λ which is the angle between the line and an a reference point (example x-axis)

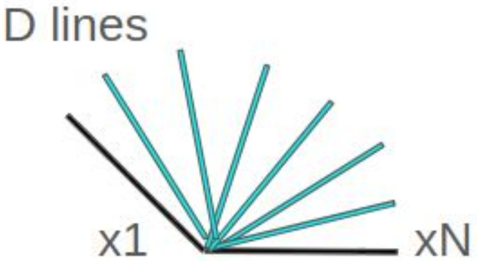
Valid interpolation

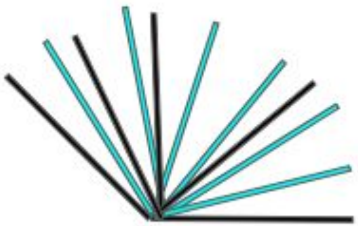
- Interpolate from x1 to x2 smoothly and linearly adjust Λ from the angle of the line in x1 to the angle in x2.
- The interpolation traverses the shortest path possible along the data manifold.

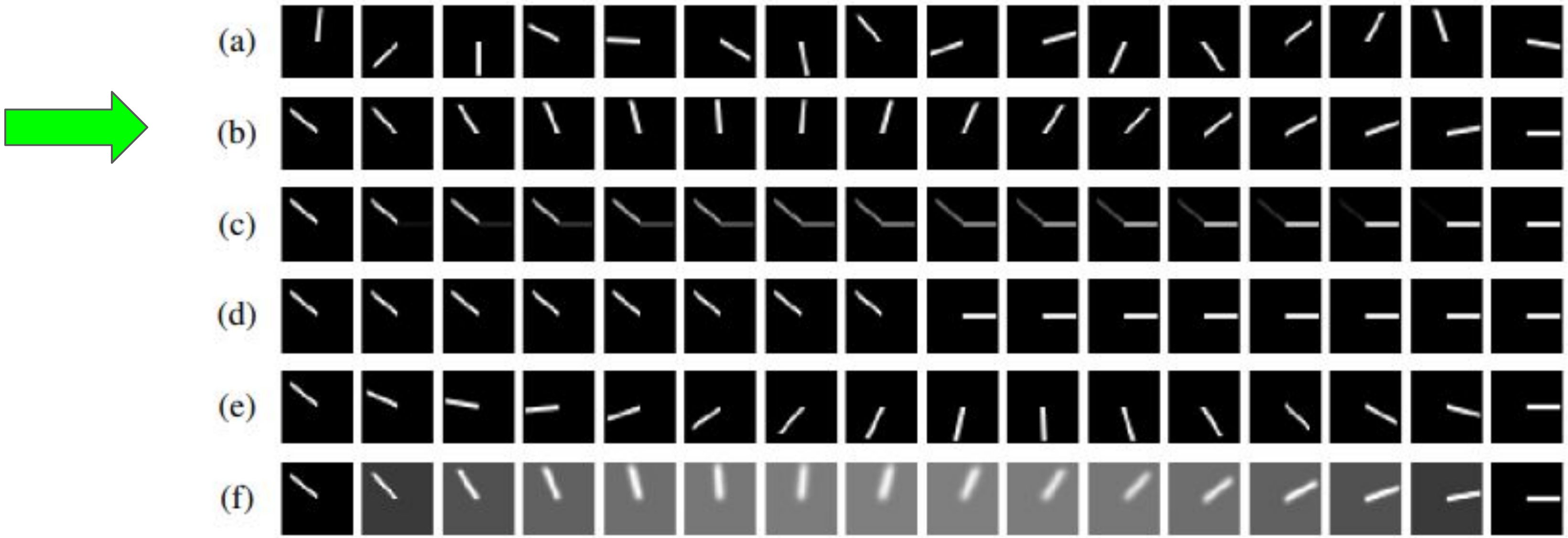
Visualizing interpolation metrics

1) **Mean distance**

2) **Smoothness**





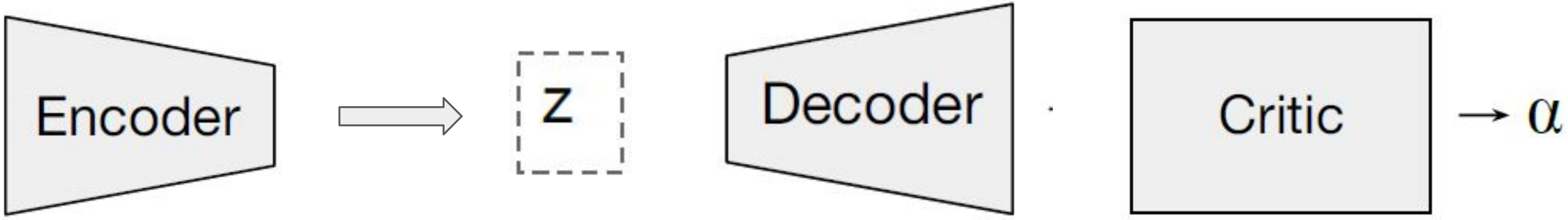


(a)
(b)
(c)
(d)
(e)
(f)

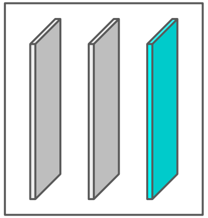
Pixel vs latent space interpolation

~2 minutes break~

Base model architecture and training procedure



Basic block: 2 conv (3x3) + pooling (2x2)



Latent vector dimension = 64

Lambda = 0.5

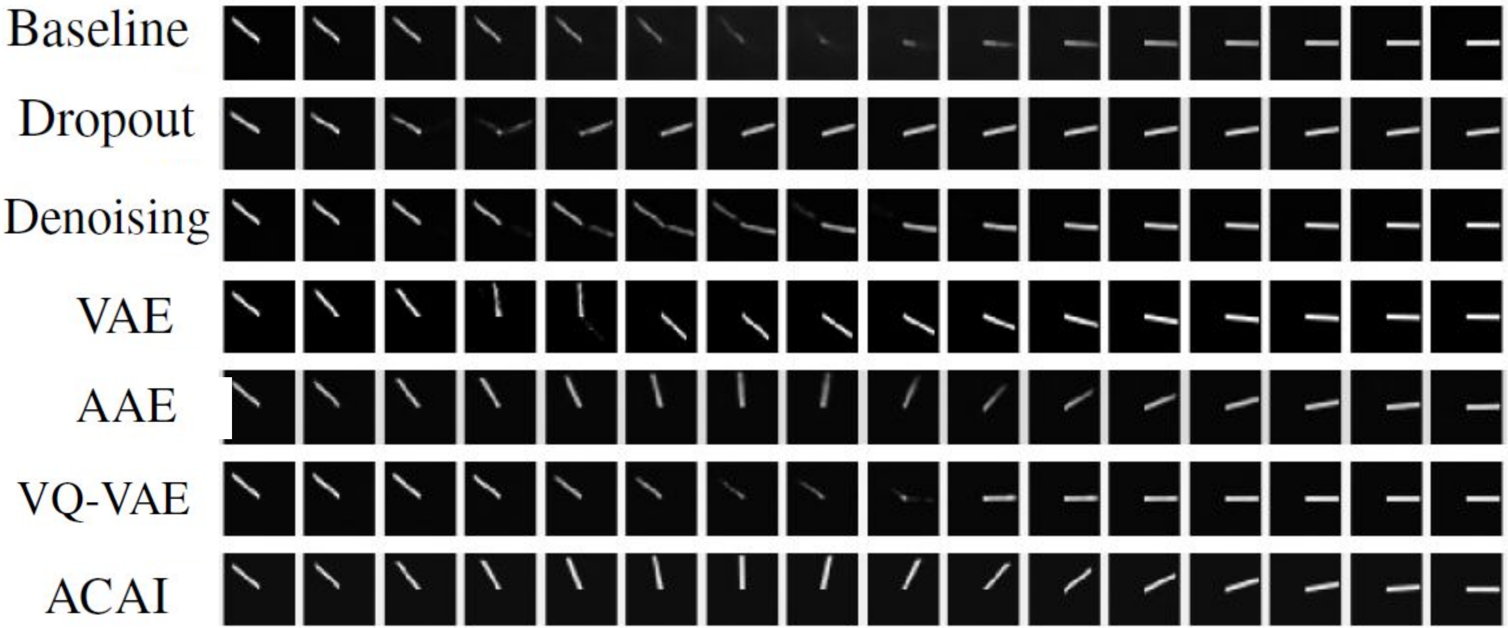
Optimizer= Adam with the learning rate of 0.0001

Batch size =64

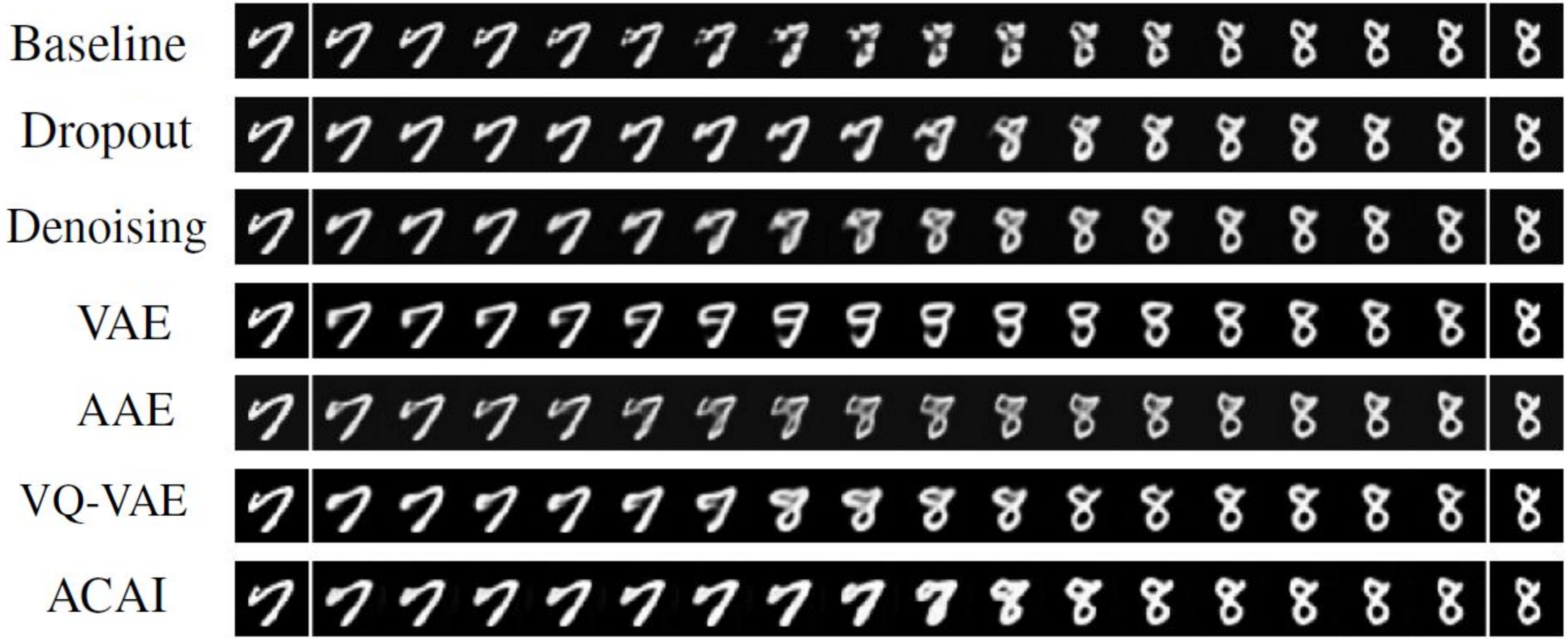
number of samples = 2^24

Experiments results

Metric	Baseline	Dropout	Denoising	VAE	AAE	VQ-VAE	ACAI
Mean Distance $(\times 10^{-3})$	6.88±0.21	2.85±0.54	4.21±0.32	1.21±0.17	3.26±0.19	5.41±0.49	**0.24±0.01**
Smoothness	0.44±0.04	0.74±0.02	0.66±0.02	0.49±0.13	0.14±0.02	0.77±0.02	**0.10±0.01**



Interpolation experiments on real data



SVHN

CelebA

Performance experiment on real data

Table 2: 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**

Takeaways

- The paper provided in-depth perspective on interpolation in autoencoders.
- It proposed Adversarially Constrained Autoencoder Interpolation (ACAI).
- Proposed a synthetic benchmark and showed that ACAI substantially outperformed common autoencoder models.
- It studied the effect of improved interpolation on downstream tasks.
- It showed that ACAI led to improved performance for feature learning.

Discussion Points

- Does the technique help in getting better random samples? If the proposed regularizer is applied to a VAE, does it help in getting better random samples by decoding z?
- How does ACAI benefit models with non visual results?
- How would ACAI extend to more complicated data sets rather than toy models?

Appendix A

Formalizing interpolation on synthetic line data

characteristics of a successful interpolation as specific evaluation metrics

Mean distance: Average distance between interpolated points and “real” datapoints

$$\text{Mean Distance}(\{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N\}) = \frac{1}{N}\sum_{n=1}^{N} C_{n,q_n^\star}$$

n ∈ {1, . . . , N} and x^n is nth interpolated image between x1 and x2

q ∈ {1, . . . , D} collection of line images D with corresponding angles Λq spaced evenly between 0 and 2π

Cn,q is the cosine distance between xˆn and the qth entry of D.

Smoothness: Measures whether the angles of the interpolated lines follow a linear trajectory between the angle of the start and endpoint

$$\text{Smoothness}(\{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N\}) = \frac{1}{|\tilde{\Lambda}_{q_1^\star} - \tilde{\Lambda}_{q_N^\star}|} \max_{n \in \{1,\ldots,N-1\}} \left(\tilde{\Lambda}_{q_{n+1}^\star} - \tilde{\Lambda}_{q_n^\star}\right) - \frac{1}{N-1}$$