On the Latent Space of Deep Generative Models

Honors Thesis Presentation

Naren Manoj

August 2018

Outline

Outline

Generative Models

Problem

Given training data, generate samples from the same distribution as the training data (true distribution).

Deep Generative Models

Definition (Deep Generative Model)

A *deep generative model* is a (possibly randomized) function, denoted g and typically parameterized as a neural network, that accepts some vector $z \in \mathbb{R}^d$ for some d as input and outputs some image $X \in \mathcal{I}$, where $\mathcal{I}$ represents the space of images.

Deep Generative Models

Definition (Deep Generative Model)

A *deep generative model* is a (possibly randomized) function, denoted g and typically parameterized as a neural network, that accepts some vector $z \in \mathbb{R}^d$ for some d as input and outputs some image $X \in \mathcal{I}$, where $\mathcal{I}$ represents the space of images.

1. Sample $z \sim \mathcal{N}(0, I_d)$ from *latent space*

Deep Generative Models

Definition (Deep Generative Model)

A *deep generative model* is a (possibly randomized) function, denoted g and typically parameterized as a neural network, that accepts some vector $z \in \mathbb{R}^d$ for some d as input and outputs some image $X \in \mathcal{I}$, where $\mathcal{I}$ represents the space of images.

1. Sample $z \sim \mathcal{N}(0, I_d)$ from *latent space*
2. Obtain $X = g(z)$

Deep Generative Models

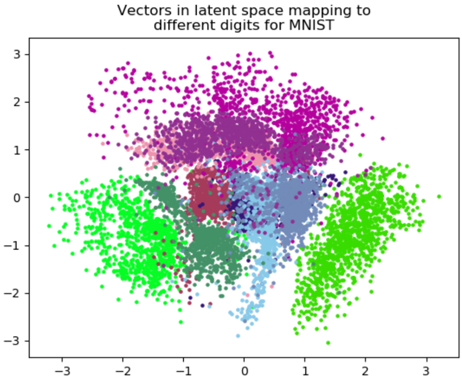
Definition (Deep Generative Model)

A *deep generative model* is a (possibly randomized) function, denoted g and typically parameterized as a neural network, that accepts some vector $z \in \mathbb{R}^d$ for some d as input and outputs some image $X \in \mathcal{I}$, where $\mathcal{I}$ represents the space of images.

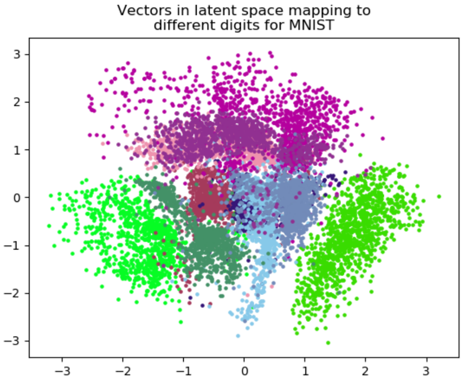
1. Sample $z \sim \mathcal{N}(0, I_d)$ from *latent space*
2. Obtain $X = g(z)$
3. Hope that distribution of outputs is close to true distribution

Outline

Examples

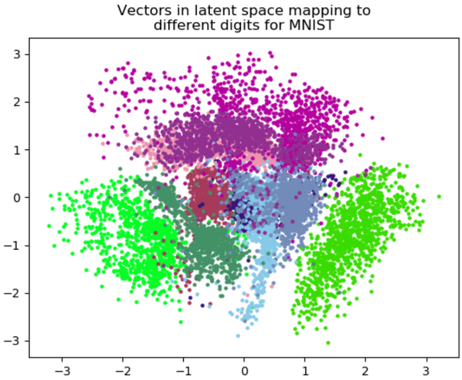

Vectors in latent space mapping to
different digits for MNIST
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

Examples



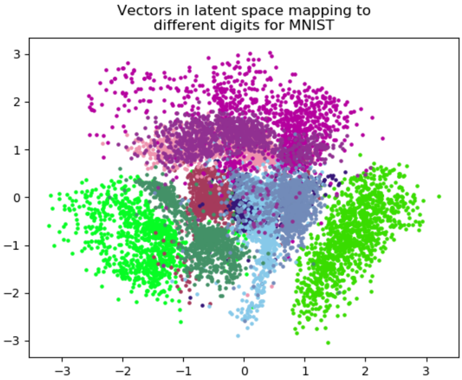
- Latent space of deep generative model mapping latent codes to handwritten digits

Examples



- Latent space of deep generative model mapping latent codes to handwritten digits
- Each colored region maps to a specific digit (class)

Examples



- Latent space of deep generative model mapping latent codes to handwritten digits
- Each colored region maps to a specific digit (class)
- Everything else: "holes"

Outline

Lack of Density in Latent Space is Bad

Two necessary and related conditions to achieve good coverage of the latent space:

Lack of Density in Latent Space is Bad

Two necessary and related conditions to achieve good coverage of the latent space:

- Most of the latent space (as measured by probability mass) maps to some image in the true distribution via a generator

Lack of Density in Latent Space is Bad

Two necessary and related conditions to achieve good coverage of the latent space:

- Most of the latent space (as measured by probability mass) maps to some image in the true distribution via a generator
    - Informal: Holes are bad (Makhzani et al. (2015))

Lack of Density in Latent Space is Bad

Two necessary and related conditions to achieve good coverage of the latent space:

- Most of the latent space (as measured by probability mass) maps to some image in the true distribution via a generator
  - Informal: Holes are bad (Makhzani et al. (2015))
- Most of the classes are close to each other

Lack of Density in Latent Space is Bad

Two necessary and related conditions to achieve good coverage of the latent space:

- Most of the latent space (as measured by probability mass) maps to some image in the true distribution via a generator
  - Informal: Holes are bad (Makhzani et al. (2015))
- Most of the classes are close to each other
  - Distant classes is bad for sample diversity

Lack of Density in Latent Space is Bad

Two necessary and related conditions to achieve good coverage of the latent space:

- Most of the latent space (as measured by probability mass) maps to some image in the true distribution via a generator
  - Informal: Holes are bad (Makhzani et al. (2015))
- Most of the classes are close to each other
  - Distant classes is bad for sample diversity (our theoretical contribution)

Lack of Density in Latent Space is Bad

Two necessary and related conditions to achieve good coverage of the latent space:

- Most of the latent space (as measured by probability mass) maps to some image in the true distribution via a generator
  - Informal: Holes are bad (Makhzani et al. (2015))
- Most of the classes are close to each other
  - Distant classes is bad for sample diversity (our theoretical contribution)
  - Attempted resolution: bring classes closer to each other and cover more probability

Lack of Density in Latent Space is Bad

Two necessary and related conditions to achieve good coverage of the latent space:

- Most of the latent space (as measured by probability mass) maps to some image in the true distribution via a generator
  - Informal: Holes are bad (Makhzani et al. (2015))
- Most of the classes are close to each other
  - Distant classes is bad for sample diversity (our theoretical contribution)
  - Attempted resolution: bring classes closer to each other and cover more probability (our empirical contribution)

Outline

Notation

- d is dimension of latent space

Notation

- d is dimension of latent space
- K is number of classes

Notation

- d is dimension of latent space
- K is number of classes
- K_i is set of latent vectors in class i

Notation

- d is dimension of latent space
- K is number of classes
- K_i is set of latent vectors in class i
- $p_i := \frac{\mathbb{P}_{z\sim\mathcal{N}(0,I_d)}[z\in K_i]}{\sum_{j=1}^{K}\mathbb{P}_{z\sim\mathcal{N}(0,I_d)}\left[z\in K_j\right]}$

Notation

- d is dimension of latent space
- K is number of classes
- K_i is set of latent vectors in class i
- $p_i := \frac{\mathbb{P}_{z\sim\mathcal{N}(0,I_d)}[z\in K_i]}{\sum_{j=1}^{K}\mathbb{P}_{z\sim\mathcal{N}(0,I_d)}[z\in K_j]}$
  - What is the chance that a randomly selected latent vector belongs to class i given that the latent vector belongs to some class?

Notation

- d is dimension of latent space
- K is number of classes
- K_i is set of latent vectors in class i
- $p_i := \frac{\mathbb{P}_{z\sim\mathcal{N}(0,I_d)}[z\in K_i]}{\sum_{j=1}^{K}\mathbb{P}_{z\sim\mathcal{N}(0,I_d)}[z\in K_j]}$
  - What is the chance that a randomly selected latent vector belongs to class i given that the latent vector belongs to some class?
  - p is the vector of the probabilities

Notation

- d is dimension of latent space
- K is number of classes
- K_i is set of latent vectors in class i
- $p_i := \frac{\mathbb{P}_{z\sim\mathcal{N}(0,I_d)}[z\in K_i]}{\sum_{j=1}^{K}\mathbb{P}_{z\sim\mathcal{N}(0,I_d)}[z\in K_j]}$
    - What is the chance that a randomly selected latent vector belongs to class i given that the latent vector belongs to some class?
    - p is the vector of the probabilities
- If $z \in K_i$, then $r(z) := \inf_{z' \in \bigcup_{j \neq i} K_j} \|z - z'\|_2$

Notation

- d is dimension of latent space
- K is number of classes
- K_i is set of latent vectors in class i
- $p_i := \frac{\mathbb{P}_{z\sim\mathcal{N}(0,I_d)}[z\in K_i]}{\sum_{j=1}^{K}\mathbb{P}_{z\sim\mathcal{N}(0,I_d)}[z\in K_j]}$
  - What is the chance that a randomly selected latent vector belongs to class i given that the latent vector belongs to some class?
  - p is the vector of the probabilities
- If $z \in K_i$, then $r(z) := \inf_{z'\in\bigcup_{j\neq i}K_j} \|z - z'\|_2$
  - How far away am I from the closest point in a different class?

Assumptions

Probability Decay

$$p_i \leq 1/5$$
$$p_i \leq o(1)$$

Assumptions

Probability Decay

$$p_i \leq 1/5$$
$$p_i \leq o(1)$$

No class is asymptotically dominant.

Assumptions

Probability Decay

$$p_i \leq 1/5$$
$$p_i \leq o(1)$$

No class is asymptotically dominant.

Partition of Latent Space

$$\bigcup_{i=1}^{K} K_i = \mathbb{R}^d$$

Assumptions

Probability Decay

$$p_i \leq 1/5$$
$$p_i \leq o(1)$$

No class is asymptotically dominant.

Partition of Latent Space

$$\bigcup_{i=1}^{K} K_i = \mathbb{R}^d$$

Every latent vector maps to some image in the true distribution.

Outline

Class Radius to Diversity

Theorem

Under the aforementioned assumptions, we have the following regarding the expected class radius of each image in the latent space of our generative model:

$$\mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,I_d)} r(z) \leq \frac{\log\left(4\pi\log\left(1/\left\|p\right\|_2^2\right)\right)}{\sqrt{2\log\left(1/\left\|p\right\|_2^2\right)}}$$

Class Radius to Diversity

Theorem

Under the aforementioned assumptions, we have the following regarding the expected class radius of each image in the latent space of our generative model:

$$\mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,I_d)} r(z) \leq \frac{\log\left(4\pi\log\left(1/\left\|p\right\|_2^2\right)\right)}{\sqrt{2\log\left(1/\left\|p\right\|_2^2\right)}}$$

- Upper bound depends *only* on $\|p\|_2^2$

Class Radius to Diversity

Theorem

Under the aforementioned assumptions, we have the following regarding the expected class radius of each image in the latent space of our generative model:

$$\mathop{\mathbb{E}}_{z \sim \mathcal{N}(0, I_d)} r(z) \leq \frac{\log\left(4\pi \log\left(1/\left\|p\right\|_2^2\right)\right)}{\sqrt{2\log\left(1/\left\|p\right\|_2^2\right)}}$$

- Upper bound depends *only* on $\|p\|_2^2$
- Inverting yields an *upper bound* on $\log\left(1/\left\|p\right\|_2^2\right)$

Class Radius to Diversity

Theorem

Under the aforementioned assumptions, we have the following regarding the expected class radius of each image in the latent space of our generative model:

$$\mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,I_d)} r(z) \leq \frac{\log\left(4\pi\log\left(1/\left\|p\right\|_2^2\right)\right)}{\sqrt{2\log\left(1/\left\|p\right\|_2^2\right)}}$$

- Upper bound depends *only* on $\|p\|_2^2$
- Inverting yields an *upper bound* on $\log\left(1/\left\|p\right\|_2^2\right)$
- How do we interpret $\log\left(1/\left\|p\right\|_2^2\right)$?

Outline

$||p||_2^2$ as a Measure of Diversity

Problem

I draw two samples from some distribution p over K types. What is the probability that both my samples are of the same type?

$\|p\|_2^2$ as a Measure of Diversity

Problem

I draw two samples from some distribution p over K types. What is the probability that both my samples are of the same type?

Solution

- $\mathbb{P}\left[\textit{both samples are of type } i\right] = p_i^2$

$\|p\|_2^2$ as a Measure of Diversity

Problem

I draw two samples from some distribution p over K types. What is the probability that both my samples are of the same type?

Solution

- $\mathbb{P}\left[\textit{both samples are of type } i\right] = p_i^2$
- $\sum p_i^2 = \|p\|_2^2$

$\|p\|_2^2$ as a Measure of Diversity

Definition (α-order Rényi Entropy)

$$H_\alpha(p) = \frac{1}{1-\alpha} \log \left(\sum_{i=1}^{K} p_i^\alpha \right)$$

$\|p\|_2^2$ as a Measure of Diversity

Definition (α-order Rényi Entropy)

$$H_\alpha(p) = \frac{1}{1-\alpha} \log \left(\sum_{i=1}^{K} p_i^\alpha \right)$$

Special Case: Shannon Entropy

$$\lim_{\alpha \to 1} H_\alpha(p) = -\sum_{i=1}^{K} p_i \log(p_i)$$

$\|p\|_2^2$ as a Measure of Diversity

Definition (α-order Rényi Entropy)

$$H_\alpha(p) = \frac{1}{1-\alpha} \log \left(\sum_{i=1}^{K} p_i^\alpha \right)$$

Special Case: Shannon Entropy

$$\lim_{\alpha \to 1} H_\alpha(p) = -\sum_{i=1}^{K} p_i \log(p_i)$$

Special Case: Collision Entropy

$$H_2(p) = -\log \left(\sum_{i=1}^{K} p_i^2 \right) = \log \left(\frac{1}{\|p\|_2^2} \right)$$

$\|p\|_2^2$ as a Measure of Diversity

Fact (Maximum Entropy Distribution)

For all α, the distribution maximizing the α-order Rényi Entropy is the uniform distribution, yielding:

$$H_\alpha(p) \leq \log(K)$$

$\|p\|_2^2$ as a Measure of Diversity

Fact (Maximum Entropy Distribution)

For all α, the distribution maximizing the α-order Rényi Entropy is the uniform distribution, yielding:

$$H_\alpha(p) \leq \log(K)$$

Thus, our bound is *smallest* when p is the uniform (most diverse) distribution.

$\|p\|_2^2$ as a Measure of Diversity

Fact (Maximum Entropy Distribution)

For all α, the distribution maximizing the α-order Rényi Entropy is the uniform distribution, yielding:

$$H_\alpha(p) \leq \log(K)$$

Thus, our bound is *smallest* when p is the uniform (most diverse) distribution.

$$\mathop{\mathbb{E}}_{z \sim \mathcal{N}(0, I_d)} r(z) \leq \frac{\log(4\pi \log(K))}{\sqrt{2\log(K)}}$$

Outline

Proof Outline of Main Theorem

1. Get upper bound on $\mathbb{P}\left[r(z) \geq \eta\right]$

Proof Outline of Main Theorem

1. Get upper bound on $\mathbb{P}\left[r(z) \geq \eta\right]$
   - Obtained by Fawzi et al. (2018)

Proof Outline of Main Theorem

1. Get upper bound on $\mathbb{P}\left[r(z) \geq \eta\right]$
   - Obtained by Fawzi et al. (2018)
2. Use $\mathbb{E}\left[X\right] = \int \mathbb{P}\left[X \geq \eta\right] \mathrm{d}\eta$

Proof Outline of Main Theorem

1. Get upper bound on $\mathbb{P}\left[r(z) \geq \eta\right]$
   - Obtained by Fawzi et al. (2018)
2. Use $\mathbb{E}\left[X\right] = \int \mathbb{P}\left[X \geq \eta\right] \mathrm{d}\eta$
   - Obtained by Fawzi et al. (2018)

Proof Outline of Main Theorem

1. Get upper bound on $\mathbb{P}\left[r(z) \geq \eta\right]$
   - Obtained by Fawzi et al. (2018)
2. Use $\mathbb{E}\left[X\right] = \int \mathbb{P}\left[X \geq \eta\right] \mathrm{d}\eta$
   - Obtained by Fawzi et al. (2018)
3. Massage a little

Proof Outline of Main Theorem

1. Get upper bound on $\mathbb{P}\left[r(z) \geq \eta\right]$
   - Obtained by Fawzi et al. (2018)
2. Use $\mathbb{E}\left[X\right] = \int \mathbb{P}\left[X \geq \eta\right] \mathrm{d}\eta$
   - Obtained by Fawzi et al. (2018)
3. Massage a little
   - Our contribution

Some Details

(Slightly rewritten) Result from Fawzi et al. (2018)

$$\mathbb{E}_{z\sim\mathcal{N}(0,I_d)} r(z) \leq \sum_{i=1}^{K} \Phi^{-1}(p_i) \cdot p_i + \frac{e^{-\Phi^{-1}(p_i)^2/2}}{\sqrt{2\pi}}$$

Some Details

(Slightly rewritten) Result from Fawzi et al. (2018)

$$\mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,I_d)} r(z) \leq \sum_{i=1}^{K} \Phi^{-1}(p_i)\cdot p_i + \frac{e^{-\Phi^{-1}(p_i)^2/2}}{\sqrt{2\pi}}$$

Rewrite some:

$$\sum_{i=1}^{K} \Phi^{-1}(p_i)\cdot p_i + \frac{e^{-\Phi^{-1}(p_i)^2/2}}{\sqrt{2\pi}} \leq \sum_{i=1}^{K} p_i \left(\frac{\log\left(4\pi\log\left(1/p_i\right)\right)}{\sqrt{2\log\left(1/p_i\right)}} \right)$$

Some Details

Rewrite some:

$$\sum_{i=1}^{K} \Phi^{-1}(p_i) \cdot p_i + \frac{e^{-\Phi^{-1}(p_i)^2/2}}{\sqrt{2\pi}} \leq \sum_{i=1}^{K} p_i \left(\frac{\log\left(4\pi \log\left(1/p_i\right)\right)}{\sqrt{2\log\left(1/p_i\right)}} \right)$$

Some Details

Rewrite some:

$$\sum_{i=1}^{K} \Phi^{-1}(p_i) \cdot p_i + \frac{e^{-\Phi^{-1}(p_i)^2/2}}{\sqrt{2\pi}} \leq \sum_{i=1}^{K} p_i \left(\frac{\log\left(4\pi \log\left(1/p_i\right)\right)}{\sqrt{2\log\left(1/p_i\right)}} \right)$$

Use concavity of:

$$\gamma(x) := \frac{\log\left(4\pi \log\left(1/x\right)\right)}{\sqrt{2\log\left(1/x\right)}}$$

Some Details

Rewrite some:

$$\sum_{i=1}^{K} \Phi^{-1}(p_i) \cdot p_i + \frac{e^{-\Phi^{-1}(p_i)^2/2}}{\sqrt{2\pi}} \leq \sum_{i=1}^{K} p_i \left(\frac{\log\left(4\pi \log\left(1/p_i\right)\right)}{\sqrt{2\log\left(1/p_i\right)}} \right)$$

Use concavity of:

$$\gamma(x) := \frac{\log\left(4\pi \log\left(1/x\right)\right)}{\sqrt{2\log\left(1/x\right)}}$$

Then Jensen:

$$\mathop{\mathbb{E}}_{z \sim \mathcal{N}(0, I_d)} r(z) \leq \gamma\left(\sum_{i=1}^{K} p_i \cdot p_i\right) = \gamma\left(\|p\|_2^2\right) = \boxed{\frac{\log\left(4\pi \log\left(1/\|p\|_2^2\right)\right)}{\sqrt{2\log\left(1/\|p\|_2^2\right)}}}$$

Outline

Test for Diversity

Problem

Suppose you have two generators g_1 and g_2, classifier f, and access to $\mathbb{E}[r(z)]$. Compare the diversity of the generators.

Test for Diversity

Problem

Suppose you have two generators g_1 and g_2, classifier f, and access to $\mathbb{E}\left[r(z)\right]$. Compare the diversity of the generators.

Solution

Compare $\mathbb{E}\left[r(z)\right]$ for g_1 and g_2; claim that the generator with higher class radius is less diverse.

Test for Diversity

Problem

Suppose you have two generators g_1 and g_2, classifier f, and access to $\mathbb{E}\left[r(z)\right]$. Compare the diversity of the generators.

Solution

Compare $\mathbb{E}\left[r(z)\right]$ for g_1 and g_2; claim that the generator with higher class radius is less diverse.

Unfortunately, it is unclear that computing $\mathbb{E}\left[r(z)\right]$ is easier than computing $\|p\|_2^2$.

Adversarial Examples - A Tangential Connection

- Recall:

$$r(z) = \inf_{z' \in \bigcup_{j \neq c(g(z))} K_j} \left\| z - z' \right\|$$

Adversarial Examples - A Tangential Connection

- Recall:

$$r(z) = \inf_{z' \in \bigcup_{j \neq c(g(z))} K_j} \left\| z - z' \right\|$$

- "Find an adversarial example in z-space"

Adversarial Examples - A Tangential Connection

- Recall:

$$r(z) = \inf_{z' \in \bigcup_{j \neq c(g(z))} K_j} \left\| z - z' \right\|$$

- "Find an adversarial example in z-space"
- If g is L-Lipschitz, then on average, you don't have to perturb too much in image space:

$$L \cdot \frac{\log \left(4\pi \log \left(1/ \left\| p \right\|_2^2 \right) \right)}{\sqrt{2 \log \left(1/ \left\| p \right\|_2^2 \right)}}$$

Adversarial Examples - A Tangential Connection

- Recall:

$$r(z) = \inf_{z' \in \bigcup_{j \neq c(g(z))} K_j} \left\| z - z' \right\|$$

- "Find an adversarial example in z-space"
- If g is L-Lipschitz, then on average, you don't have to perturb too much in image space:

$$L \cdot \frac{\log\left(4\pi \log\left(1/\left\|p\right\|_2^2\right)\right)}{\sqrt{2\log\left(1/\left\|p\right\|_2^2\right)}}$$

- Under the probability decay assumption, above approaches 0 as the number of classes K grows arbitrarily large

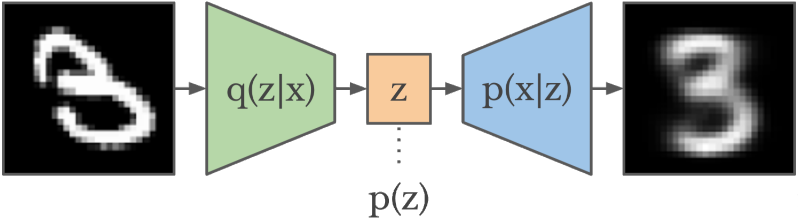
Outline

1 Introduction
- Deep Generative Models Overview
- Latent Space Overview
- Main Claim

2 Theory
- Notation and Assumptions
- Main Result
- Relationship to Diversity
- Proofs
- Potential Implications

3 Experiments
- Introduction to VAEs
- VAE Modification
- Experimental Setup
- Results

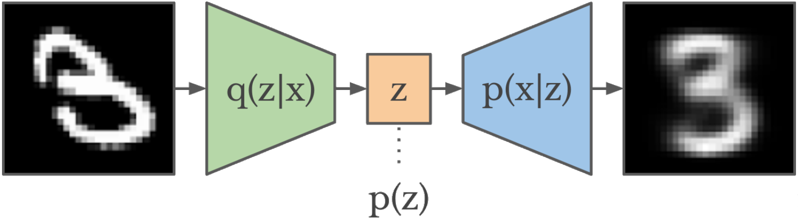
VAE Architecture



[1]

[1]Diagram courtesy of Danijar Hafner.

VAE Architecture

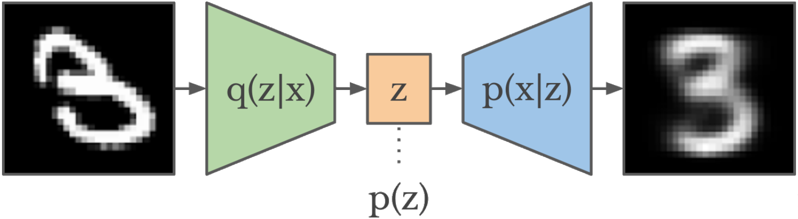


[1]

- $X \sim \mathcal{X}$ is image sampled from data distribution $\mathcal{X}$

[1] Diagram courtesy of Danijar Hafner.

VAE Architecture

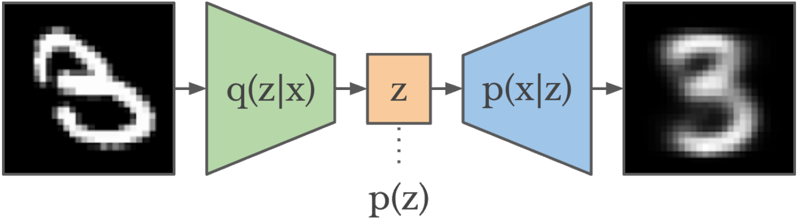


[1]

- $X \sim \mathcal{X}$ is image sampled from data distribution $\mathcal{X}$
- $Q(Z|X = x)$ is a distribution of encoder outputs for input image x

[1] Diagram courtesy of Danijar Hafner.

VAE Architecture

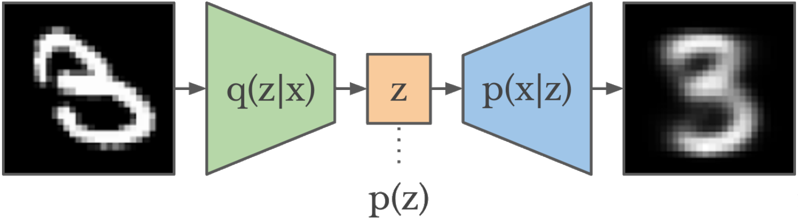


[1]

- $X \sim \mathcal{X}$ is image sampled from data distribution $\mathcal{X}$
- $Q(Z|X = x)$ is a distribution of encoder outputs for input image x
- Objective: Match $Q(Z|X)$ to a target distribution over latent vectors ($\mathcal{N}(0, I_d)$)

[1] Diagram courtesy of Danijar Hafner.

VAE Architecture

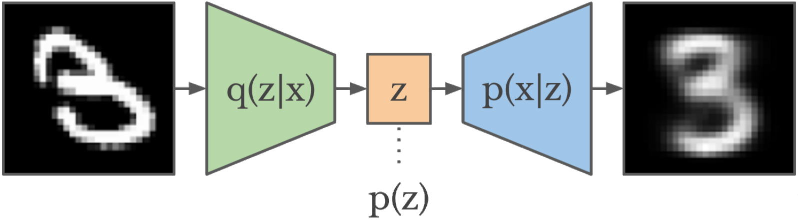


[1]

- $X \sim \mathcal{X}$ is image sampled from data distribution $\mathcal{X}$
- $Q(Z|X = x)$ is a distribution of encoder outputs for input image x
- Objective: Match $Q(Z|X)$ to a target distribution over latent vectors $(\mathcal{N}(0, I_d))$
- $P(X|Z = z)$ is a distribution of decoder outputs for input latent vector z

[1] Diagram courtesy of Danijar Hafner.

VAE Architecture



[1]

- $X \sim \mathcal{X}$ is image sampled from data distribution $\mathcal{X}$
- $Q(Z|X=x)$ is a distribution of encoder outputs for input image x
- Objective: Match $Q(Z|X)$ to a target distribution over latent vectors ($\mathcal{N}(0, I_d)$)
- $P(X|Z=z)$ is a distribution of decoder outputs for input latent vector z
- Objective: Minimize expected reconstruction loss

[1]Diagram courtesy of Danijar Hafner.

Outline

Motivation

Problem

How do we bring different classes closer together while training a VAE?

Motivation

Problem

How do we bring different classes closer together while training a VAE?

Solution

Penalize distant encodings.

Motivation

Problem

How do we bring different classes closer together while training a VAE?

Solution

Penalize distant encodings.

$$\omega\left(I_1, I_2\right) := \frac{\left\|f\left(I_1\right) - f\left(I_2\right)\right\|_2^2}{\left\|I_1 - I_2\right\|_2^2}$$

Motivation

Problem

How do we bring different classes closer together while training a VAE?

Solution

Penalize distant encodings.

$$\omega\left(I_1, I_2\right) := \frac{\left\|f\left(I_1\right) - f\left(I_2\right)\right\|_2^2}{\left\|I_1 - I_2\right\|_2^2}$$

Regularize with $\omega\left(\cdot, \cdot\right)$ *summed over many pairs of images.*

Motivation

Problem

How do we bring different classes closer together while training a VAE?

Solution

Penalize distant encodings.

$$\omega\left(I_1, I_2\right) := \frac{\left\|f\left(I_1\right) - f\left(I_2\right)\right\|_2^2}{\left\|I_1 - I_2\right\|_2^2}$$

Regularize with $\omega\left(\cdot,\cdot\right)$ summed over many pairs of images. To force ω to be low, enforce Lipschitz constant via network weight clipping.

Outline

Experimental Setup

- Used a modified VAE implementation inspired from PyTorch examples repository

Experimental Setup

- Used a modified VAE implementation inspired from PyTorch examples repository
    - DCGAN-like encoder and decoder

Experimental Setup

- Used a modified VAE implementation inspired from PyTorch examples repository
  - DCGAN-like encoder and decoder
- Vanilla VAE objective

Experimental Setup

- Used a modified VAE implementation inspired from PyTorch examples repository
  - DCGAN-like encoder and decoder
- Vanilla VAE objective
- Regularizer (batch size b; set $\lambda = 30$):

$$\lambda \cdot \sum_{i=1}^{b/2} \omega \left(I_i, I_{i+b/2} \right)$$

Experimental Setup

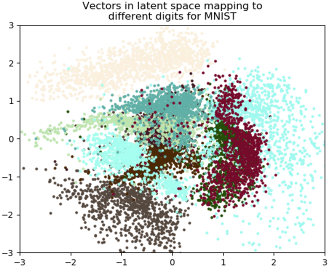
- Used a modified VAE implementation inspired from PyTorch examples repository
  - DCGAN-like encoder and decoder
- Vanilla VAE objective
- Regularizer (batch size b; set $\lambda = 30$):

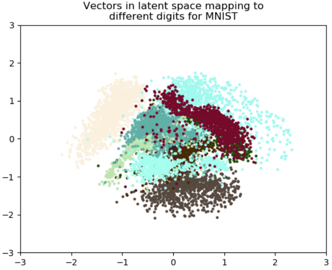
$$\lambda \cdot \sum_{i=1}^{b/2} \omega\left(I_i, I_{i+b/2}\right)$$

- Clip weights to $[-0.3, 0.3]$

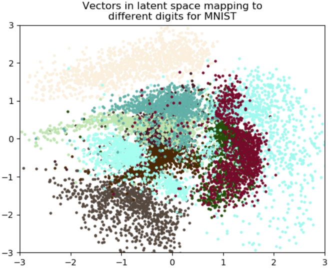
Outline

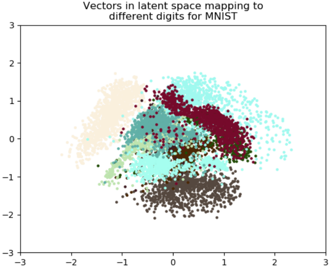
Geometric Implications





Geometric Implications





Conclusion: Modified VAE makes classes come closer to one another but decreases probability mass of space covered

Sample Quality

Sample Quality

Conclusion: Modified VAE does not change quality of generated samples noticeably

Interpretation of Results

Hyperparameter selection is tricky

Interpretation of Results

Hyperparameter selection is tricky

- Better weight clip constant might lead to better coverage of the latent space

Interpretation of Results

Hyperparameter selection is tricky

- Better weight clip constant might lead to better coverage of the latent space

Regularizer itself is not particularly useful

Interpretation of Results

Hyperparameter selection is tricky

- Better weight clip constant might lead to better coverage of the latent space

Regularizer itself is not particularly useful

- Network could be learning a (almost) scaled-down version of the latent space

Interpretation of Results

Hyperparameter selection is tricky

- Better weight clip constant might lead to better coverage of the latent space

Regularizer itself is not particularly useful

- Network could be learning a (almost) scaled-down version of the latent space

Problem

For future: Devise an optimization objective that favors both closer classes and better coverage of the latent space.

Conclusion

Summary

We care about coverage of the latent space.

Conclusion

Summary

We care about coverage of the latent space.

- As previously identified, this is indicative of the model having better learned the data manifold.

Conclusion

Summary

We care about coverage of the latent space.

- As previously identified, this is indicative of the model having better learned the data manifold.

We care about density of the latent space.

Conclusion

Summary

We care about coverage of the latent space.

- As previously identified, this is indicative of the model having better learned the data manifold.

We care about density of the latent space.

- As we prove, this is indicative of the model learning a more diverse distribution.

Conclusion

Summary

We care about coverage of the latent space.

- As previously identified, this is indicative of the model having better learned the data manifold.

We care about density of the latent space.

- As we prove, this is indicative of the model learning a more diverse distribution.

Future Work

We want to leverage these findings empirically.

Conclusion

Summary

We care about coverage of the latent space.

- As previously identified, this is indicative of the model having better learned the data manifold.

We care about density of the latent space.

- As we prove, this is indicative of the model learning a more diverse distribution.

Future Work

We want to leverage these findings empirically.

- Develop optimization procedures to achieve these properties

Conclusion

Summary

We care about coverage of the latent space.

- As previously identified, this is indicative of the model having better learned the data manifold.

We care about density of the latent space.

- As we prove, this is indicative of the model learning a more diverse distribution.

Future Work

We want to leverage these findings empirically.

- Develop optimization procedures to achieve these properties
- Test generative models for diversity

Conclusion

Summary

We care about coverage of the latent space.

- As previously identified, this is indicative of the model having better learned the data manifold.

We care about density of the latent space.

- As we prove, this is indicative of the model learning a more diverse distribution.

Future Work

We want to leverage these findings empirically.

- Develop optimization procedures to achieve these properties
- Test generative models for diversity
- Meta-Problem: Theoretically understand generative models better

References I

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Sanjeev Arora, Andrej Risteski, and Yi Zhang. Do GANs learn the distribution? some theory and empirics. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=BJehNfW0-`.

Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards AI. In *Large Scale Kernel Machines*. MIT Press, 2007.

Ali Borji. Pros and cons of gan evaluation measures. *arXiv preprint arXiv:1802.03446*, 2018.

GM Bosyk, M Portesi, and A Plastino. Collision entropy and optimal uncertainty. *Physical Review A*, 85(1):012108, 2012.

Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

References II

Lutz Duembgen. Bounding standard gaussian tail probabilities. *arXiv preprint arXiv:1012.2063*, 2010.

Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. Adversarial vulnerability for any classifier. *arXiv preprint arXiv:1802.08686*, 2018.

Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18: 1527–1554, 2006.

References III

He Huang, Phillip Yu, and Changhu Wang. An introduction to image synthesis with generative adversarial nets. *arXiv preprint arXiv:1803.04469*, 2018.

Matt Jordan, Naren Manoj, Surbhi Goel, and Alex Dimakis. Combined adversarial attacks. *NIPS 2018 (submitted)*, 2018.

Valentin Khrulkov and Ivan Oseledets. Geometry score: A method for comparing generative adversarial networks. *arXiv preprint arXiv:1802.02664*, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 2010. URL `http://yann.lecun.com/exdb/mnist/`.

References IV

Fei-Fei Li, Justin Johnson, and Serena Yeung. Lecture notes for convolutional neural networks for visual recognition, May 2018.

Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. *arXiv preprint arXiv:1712.04086*, 2017.

Erik Linder-Norn. Pytorch-gan. `https://github.com/eriklindernoren/PyTorch-GAN`, 2018.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

References V

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

Shibani Santurkar, Ludwig Schmidt, and Aleksander Madry. A classification-based perspective on gan distributions. *arXiv preprint arXiv:1711.00970*, 2017.

Shibani Santurkar, Ludwig Schmidt, and Aleksander Madry. A classification-based study of covariate shift in GAN distributions. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4487–4496, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL `http://proceedings.mlr.press/v80/santurkar18a.html`.

References VI

Akash Srivastava, Lazar Valkoz, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pp. 3308–3318, 2017.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=HkL7n1-0b`.