ClusterGAN : Latent Space Clustering in Generative Adversarial Networks

Lu Xiao

SMILE Lab, UESTC

nbshawnlu@hotmail.com

December 7, 2018

Overview

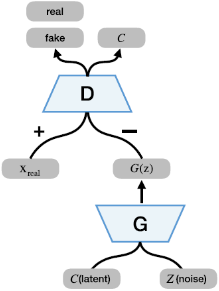
Representation learning

Learning representations of the data that make it easier to extract useful information when building classifiers or other predictors.
Deep learning methods are formed by the composition of multiple non-linear transformations, with the goal of yielding more abstract and ultimately more useful representations.

Generative Adversarial Network (GAN)

The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency.

InfoGAN



GAN loss:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim P_{data}}[logD(x)] + \mathbb{E}_{z \sim noise}[log(1 - D(G(z)))]$$

InfoGAN:

$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda I(c; G(z,c))$$

Motivation

- Representation learning enables machine learning models to decipher underlying semantics in data and disentangle hidden factors of variation.
- It would be even nicer if such clustering was obtained along with dimensionality reduction where the real data actually seems to come from a lower dimensional manifold.
- In recent times, much of unsupervised learning is driven by deep generative approaches, the two most prominent being Variational Autoencoder (VAE) and Generative Adversarial Network (GAN).

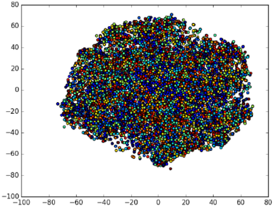
Motivation

Can we design a GAN training methodology that clusters in the latent space?

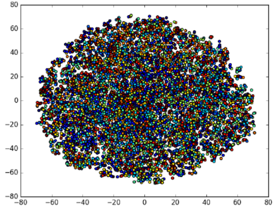
Challenge

- Back-propagate the data into the latent space does not cluster well. The key reason is that, if indeed, back-propagation succeeds, then the back-projected data distribution should look similar to the latent space distribution.
- GANs with a Gaussian mixture failed to cluster. Gaussian components tend to crowd and become redundant.

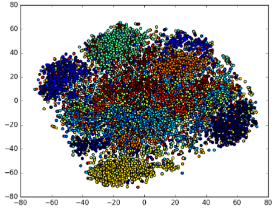
Challenge



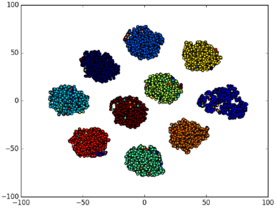
(a) $z \sim$ Uniform



(b) $z \sim$ Normal



(c) $z \sim$ Gaussian Mix



(d) $z \sim (z_n, z_c)$

TSNE visualization of latent space : MNIST

Architecture & Discrete-Continuous Mixtures

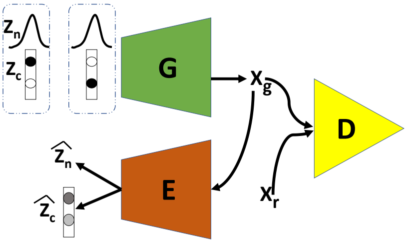


Figure: ClusterGAN Architecture

Sample from a prior that consists of normal random variables cascaded with one-hot encoded vectors:

$z = (z_n, z_c)$

$z_n \sim \mathcal{N}(0, \sigma^2 I_{d_n})$

$z_c = e_k, k \sim \mathcal{U}\{1, 2, \ldots, K\}$

e_k is the k^{th} elementary vector in $\mathbb{R}^K$

K is the number of clusters in the data.

Modified Backpropagation Based Decoding

Input: Real sampler x, Generator function $\mathcal{G}$, Number of Clusters K, Regularization parameter λ, Adam iterations τ
Output: Latent embedding z^*
for $k \in \{1, 2, \dots K\}$ **do**
  Sample $z_n^0 \sim \mathcal{N}(0, \sigma^2 I_{d_n})$
  Initialization $z_k^0 \leftarrow (z_n^0, e_k)$ (e_k is k^{th} elementary unit vector in K dimensions)
  for $t \in \{1, 2, \dots \tau\}$ **do**
    Obtain the gradient of loss function
    $g \leftarrow \nabla_{z_n} \left(\|\mathcal{G}(z_k^{t-1}) - x\|_1 + \lambda \|z_n^{t-1}\|_2 \right)$
    Update z_n^t using g with Adam iteration to minimize loss.
    Clipping of z_n^t, i.e., $z_n^t \leftarrow \mathcal{P}_{[-0.6, 0.6]}(z_n^t)$
    $z_k^t \leftarrow (z_n^t, e_k)$
  end
  Update z^* if z_k^τ has lowest loss obtained so far.
end
return z^*

Algorithm 1: DECODE_LATENT

Accuracy on MNIST

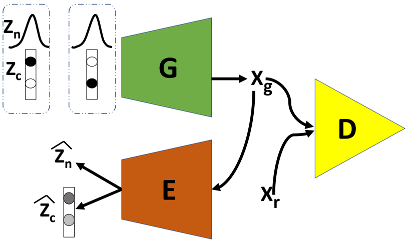


Figure: ClusterGAN Architecture

Sample from a prior that consists of normal random variables cascaded with one-hot encoded vectors:

Mode Accuracy (0.97):

$k \xrightarrow{G} x_g \xrightarrow{D} \hat{y}$

Reconstruction Accuracy (0.96):

$x_r \xrightarrow{E} z \xrightarrow{G} x_g \xrightarrow{D} \hat{y}$

Cluster Accuracy (0.95):

$X \xrightarrow{E} Z \xrightarrow{z_c \ part} \hat{Y}$

Clustering Specific Loss

Even though the above approach enables the GAN to cluster in the latent space, it may be able to perform even better if we had a clustering specific loss term in the minimax objective.
Introduce an encoder $\mathcal{E} : \mathcal{X} \mapsto \mathcal{Z}$, a neural network parameterized by Θ_E.
The GAN objective now takes the following form:

$$\min_{\Theta_G, \Theta_E} \max_{\Theta_D} \mathop{\mathbf{E}}_{x \sim \mathbb{P}_x^r} q(\mathcal{D}(x)) + \mathop{\mathbf{E}}_{z \sim \mathbb{P}_z} q(1 - \mathcal{D}(\mathcal{G}(z))) + \beta_n \mathop{\mathbf{E}}_{z \sim \mathbb{P}_z} \|z_n - \mathcal{E}(\mathcal{G}(z_n))\|_2^2 + \beta_c \mathop{\mathbf{E}}_{z \sim \mathbb{P}_z} \mathcal{H}(z_c, \mathcal{E}(\mathcal{G}(z_c)) \quad (1)$$

Algorithm

Input: Functions $\mathcal{G}$, $\mathcal{D}$ and $\mathcal{E}$, Regularization parameters β_n, β_c, learning rate η, parameters Θ_G^t, Θ_E^t

Output: $\Theta_G^{(t+1)}, \Theta_E^{(t+1)}$

Sample $z^{(i)}{}_{i=1}^{m}$ from $\mathbb{P}^z, z = (z_n, z_c)$

$g_{\Theta_G} \leftarrow$

$$\nabla_{\Theta_G}\left(-\sum_{i=1}^{m} q(\mathcal{D}(\mathcal{G}(z^{(i)})) + \beta_n \sum_{i=1}^{m} \|z_n^{(i)} - \mathcal{E}(\mathcal{G}(z_n^{(i)}))\|_2^2 + \beta_c \sum_{i=1}^{m} \mathcal{H}(z_c^{(i)}, \mathcal{E}(\mathcal{G}(z_c^{(i)})))\right)$$

$$g_{\Theta_E} \leftarrow \nabla_{\Theta_E}\left(\beta_n \sum_{i=1}^{m} \|z_n^{(i)} - \mathcal{E}(\mathcal{G}(z_n^{(i)}))\|_2^2 + \beta_c \sum_{i=1}^{m} \mathcal{H}(z_c^{(i)}, \mathcal{E}(\mathcal{G}(z_c^{(i)})))\right)$$

Update Θ_G using $(g_{\Theta_G}, \Theta_G^t)$ with Adam ; similarly for Θ_E.

return Θ_G, Θ_E

Algorithm 2: UPDATE_PARAM

Experiments

Dataset	Algorithm	ACC	NMI	ARI
Fashion-10	ClusterGAN	0.63	0.64	0.50
	Info-GAN	0.61	0.59	0.44
	GAN with bp	0.56	0.53	0.37
	GAN with Disc. ϕ	0.43	0.37	0.23
	AGGLO.	0.55	0.57	0.37
	NMF	0.50	0.51	0.34

Table: Comparison of clustering metrics across datasets

Experiments

Dataset	Algorithm	ACC	NMI	ARI
MNIST	ClusterGAN	0.95	0.89	0.89
	Info-GAN	0.87	0.84	0.81
	GAN with bp	0.95	0.90	0.89
	GAN with Disc. ϕ	0.70	0.62	0.52
	DCN	0.83	0.81	0.75
	AGGLO.	0.64	0.65	0.46
	NMF	0.56	0.45	0.36

Table: Comparison of clustering metrics across datasets

Experiments

Dataset	Algorithm			
	Cluster GAN	WGAN (Normal)	WGAN (One-Hot)	Info GAN
MNIST	0.81	0.88	0.94	1.88
Fashion	0.91	0.95	6.14	11.04
10x_73k	2.50	2.02	2.24	25.59
Pendigits	9.56	6.45	13.44	87.80

Table: Comparison of Frechet Inception Distance (FID) (Lower distance is better)

Figure: Digits generated from distinct modes : MNIST

Experiments

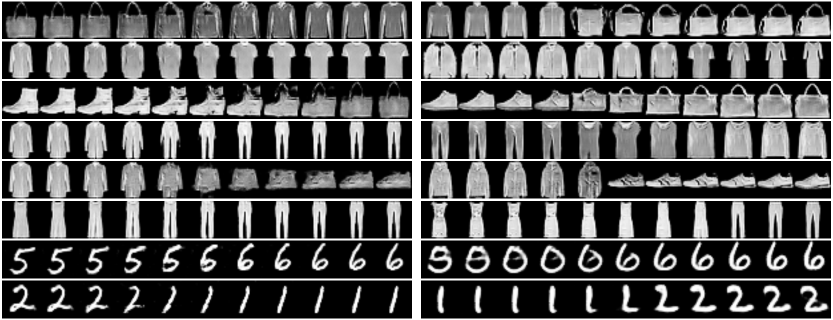


Figure: (a) ClusterGAN (left) (b) vanilla WGAN (right)

Dataset : MNIST, Algorithm : ClusterGAN **ACC**				
K = 7	**K = 9**	**K = 10**	**K = 11**	**K = 13**
0.60	0.84	0.95	0.80	0.84

Table: Robustness to Cluster Number K

Summary

Pros

- Utilize a **mixture of discrete and continuous** latent variables in order to create a non-smooth geometry in the latent space.
- Propose a **novel backpropogation algorithm** accommodating the discrete-continuous mixture.
- Retains good interpolation across the different classes.

Cons

- Need Cluster Number K.

Feature Works

- Better data-driven priors for the latent space.
- Improve results for problems that have a sparse generative structure such as compressed sensing.

References

Yoshua Bengio (2014)
Representation Learning: A Review and New Perspectives
IEEE Transactions on Pattern Analysis & Machine Intelligence 35.8(2012):1798-1828.

Xi Chen (2016)
InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
Advances in neural information processing systems 2016

Zachary C Lipton (2017)
Precise recovery of latent vectors from generative adversarial networks
arXiv 1702.04782

Mukherjee Sudipto (2018)
ClusterGAN: Latent Space Clustering in Generative Adversarial Networks
arXiv 1809.03627

Thank You