Adversarial Autoencoders

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, Brendan Frey

Presented by: Paul Vicol
Outline

- **Adversarial Autoencoders**
  - AAE with *continuous* prior distributions
  - AAE with *discrete* prior distributions
  - AAE vs VAE

- **Wasserstein Autoencoders**
  - Generalization of Adversarial Autoencoders
  - Theoretical Justification for AAEs
Classical *unregularized* autoencoders minimize a reconstruction loss \( ||x - \hat{x}||^2 \).

This yields an *unstructured latent space*.

- Examples from the data distribution are mapped to codes scattered in the space.
- No constraint that similar inputs are mapped to nearby points in the latent space.
- We cannot sample codes to generate novel examples.

VAEs are one approach to regularizing the latent distribution.
Adversarial Autoencoders - Motivation

- **Goal:** An approach to *impose structure* on the latent space of an autoencoder

- **Idea:** Train an autoencoder with an *adversarial loss* to match the distribution of the latent space to an arbitrary prior
  - Can use *any prior that we can sample from* either continuous (*Gaussian*) or discrete (*Categorical*)
Adversarial autoencoders are generative autoencoders that use adversarial training to impose an arbitrary prior on the latent code.
1. The **reconstruction phase**: Update the encoder and decoder to minimize reconstruction error

\[
\begin{align*}
q(z|x) & \quad z \sim q(z|x) \quad p(x|z)
\end{align*}
\]
2. **Regularization phase**: Update discriminator to distinguish true prior samples from generated samples; update generator to fool the discriminator.

\[ q(z|x) \quad z \sim q(z|x) \]

\[ p(z) \quad z \sim p(z) \]

Diagram:
- Encoder / GAN Generator
- Discriminator
- \([0, 1]\)
AAE vs VAE

- VAEs use a KL divergence term to impose a prior on the latent space.
- AAEs use adversarial training to match the latent distribution with the prior.

\[ \mathcal{L} = \mathbb{E}_x \left[ \mathbb{E}_{q(z|x)} \left[ - \log p(x|z) \right] \right] + \mathbb{E}_x \left[ \text{KL}(q(z|x)||p(z)) \right] \]

Reconstruction Error \hspace{2cm} KL Regularizer

- Why would we use an AAE instead of a VAE?
  - To backprop through the KL divergence we must have access to the functional form of the prior distribution \( p(z) \)
  - In an AAE, we just need to be able to sample from the prior to induce the latent distribution to match the prior.
AAE vs VAE: Latent Space

- Imposing a *Spherical 2D Gaussian prior* on the latent space

Gaps in the latent space; not well-packed
AAE vs VAE: Latent Space

- Imposing a mixture of 10 2D Gaussians prior on the latent space

VAE emphasizes the modes of the distribution; has systematic differences from the prior
Core idea: Use a discriminator to check that a latent variable is discrete
GAN for Discrete Latent Structure

- $D_{cat}$ induces the softmax output $y$ to be highly peaked at one value
- Similar to continuous relaxation with temperature annealing, but does not require setting a temperature or annealing schedule
Semi-Supervised Adversarial Autoencoders

- Model for semi-supervised learning that exploits the generative description of the unlabeled data to improve classification performance
- Assume the data is generated as follows:

\[
p(y) = \text{Cat}(y)
\]

\[
p(z) = \mathcal{N}(z|0, I)
\]

- Now the encoder predicts both the discrete class \( y \) (content) and the continuous code \( z \) (style)
- The decoder conditions on both the class label and style vector
Semi-Supervised Adversarial Autoencoders

Imposes a *discrete* (categorical) distribution on the latent class variable

Imposes a *continuous* (Gaussian) distribution on the latent style variable
### Semi-Supervised Classification Results

- AAEs outperform VAEs

<table>
<thead>
<tr>
<th></th>
<th>MNIST (100)</th>
<th>MNIST (1000)</th>
<th>MNIST (All)</th>
<th>SVHN (1000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN Baseline</td>
<td>25.80</td>
<td>8.73</td>
<td>1.25</td>
<td>47.50</td>
</tr>
<tr>
<td>VAE (M1) + T SVM</td>
<td>11.82 (±0.25)</td>
<td>4.24 (±0.07)</td>
<td>-</td>
<td>55.33 (±0.11)</td>
</tr>
<tr>
<td>VAE (M2)</td>
<td>11.97 (±1.71)</td>
<td>3.60 (±0.56)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>VAE (M1 + M2)</td>
<td>3.33 (±0.14)</td>
<td>2.40 (±0.02)</td>
<td>0.96</td>
<td>36.02 (±0.10)</td>
</tr>
<tr>
<td>VAT</td>
<td>2.33</td>
<td>1.36</td>
<td>0.64 (±0.04)</td>
<td>24.63</td>
</tr>
<tr>
<td>CatGAN</td>
<td>1.91 (±0.1)</td>
<td>1.73 (±0.18)</td>
<td>0.91</td>
<td>-</td>
</tr>
<tr>
<td>Ladder Networks</td>
<td>1.06 (±0.37)</td>
<td>0.84 (±0.08)</td>
<td>0.57 (±0.02)</td>
<td>-</td>
</tr>
<tr>
<td>ADGM</td>
<td>0.96 (±0.02)</td>
<td>-</td>
<td>-</td>
<td>16.61 (±0.24)</td>
</tr>
<tr>
<td><strong>Adversarial Autoencoders</strong></td>
<td><strong>1.90 (±0.10)</strong></td>
<td><strong>1.60 (±0.08)</strong></td>
<td><strong>0.85 (±0.02)</strong></td>
<td><strong>17.70 (±0.30)</strong></td>
</tr>
</tbody>
</table>

Table 2: Semi-supervised classification performance (error-rate) on MNIST and SVHN.
Unsupervised Clustering with AAEs

● An AAE can disentangle *discrete class variables* from continuous latent style variables without supervision
● The inference network \( q(y|x) \) predicts one-hot vector with \( K = \text{num clusters} \)

Figure 9: Unsupervised clustering of MNIST using the AAE with 16 clusters. Each row corresponds to one cluster with the first image being the cluster head. (see text)
Pros
- Flexible approach to impose arbitrary distributions over the latent space
- Works with any distribution you can sample from, continuous and discrete
- Does not require temperature/annealing hyperparameters

Cons
- May be challenging to train due to the GAN objective
- Not scalable to many latent variables → need a discriminator for each
Generative models (VAEs & GANs) try to minimize discrepancy measures between the data distribution $P_X$ and the model distribution $P_G$.

WAE minimizes a penalized form of the Wasserstein distance between the model distribution and the target distribution:

$$D_{WAE}(P_X, P_G) := \inf_{Q(Z|X) \in Q} \mathbb{E}_{P_X} \mathbb{E}_{Q(Z|X)}[c(X, G(Z))] + \lambda \cdot \mathcal{D}_Z(Q_Z, P_Z)$$

- **Reconstruction cost**
- **Regularizer** encourages the encoded distribution to match the prior.
Theoretical justification for AAEs:

- When $c(x, y) = \|x - y\|^2$ WAE = AAE
- AAEs minimize the 2-Wasserstein distance between $P_X$ and $P_G$

WAE generalizes AAE in two ways:

1. Can use any cost function $c(x, y)$ in the input space $\mathcal{X}$
2. Can use any discrepancy measure $D_Z$ in the latent space $\mathcal{Z}$
   - Not just an adversarial one
Thank you!