Adversarial Autoencoders (AAE)

Adapted from Makhzani et al. 2015

Giovanni Papini

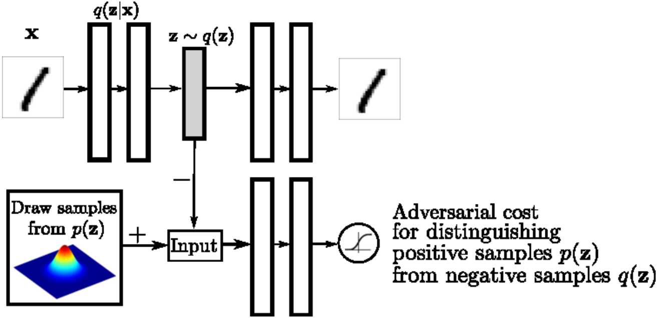
Universitá degli Studi di Firenze

February 12, 2019

Context

- Scalable generative models to capture rich distributions
- Graphical models such as RBMs , DBNs, DBMs are based on MCMC algorithms for doing inference
- VAEs, GANs, GMMNs are trained via direct back-propagation
- The AAE is trained with dual objectives:
  - **minimizing the reconstruction error** — $||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2$
  - **adversarial training criterion** — matching the aggregated posterior distribution of the latent representation to an arbitrary prior

The AAE architecture



$$p(\boldsymbol{z}) = \int_{\Omega_x} q(\boldsymbol{z}|\boldsymbol{x}) p_d(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$$

The encoder

- **Deterministic** — (used in the paper)
- **Gaussian posterior** — $z_i \sim \mathcal{N}(\mu_i(\boldsymbol{x}), \sigma_i(\boldsymbol{x}))$
- **Universal approximator posterior** — $\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\eta} \sim \delta(\boldsymbol{z} - f(\boldsymbol{x}, \boldsymbol{\eta}))$
  where $\boldsymbol{\eta}$ is random noise with a fixed distribution

$$q(\boldsymbol{z}|\boldsymbol{x}) = \int_{\Omega_\eta} q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\eta}) p_\eta(\boldsymbol{\eta}) \mathrm{d}\boldsymbol{\eta}$$

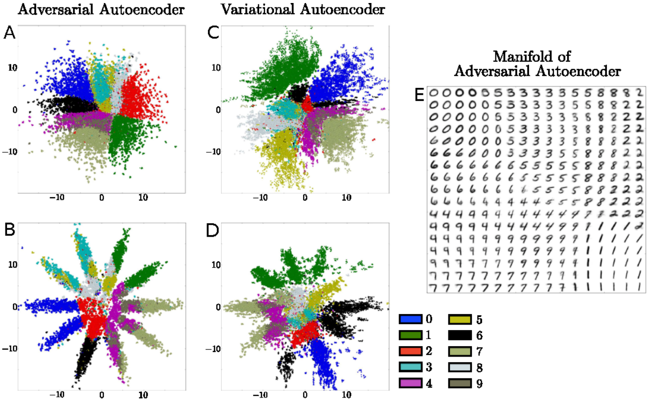
Relationship with VAEs

VAE aims to minimize the negative ELBO:

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{x}}[-\log p(\boldsymbol{x})] < \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}[-\log p(\boldsymbol{x}|\boldsymbol{z})] + \mathbb{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}))\right] \\
= \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{z}}[-\log p(\boldsymbol{x}|\boldsymbol{z})]\right] + \mathbb{E}_{\boldsymbol{x}}[-H(q(\boldsymbol{z}|\boldsymbol{x}))] \\
+ \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}[-\log p(\boldsymbol{z})] \\
= \text{Reconstr. Err.} - \text{Entropy} + \text{Crossentropy}
\end{aligned}
$$

VAE	AAE
◦ encourages $q(\boldsymbol{z})$ to match $p(\boldsymbol{z})$ modes because of crossentropy penalty	◦ encourages $q(\boldsymbol{z})$ to match the *whole* $p(\boldsymbol{z})$ because of the adversarial training
◦ needs the exact functional form of the prior distribution in order to back-propagate through KL divergence	◦ can impose even complicated distributions just through the capability of sampling from them

Relationship with VAE



Relationship with GAN and GMMN

GAN	AAE
◦ imposes the data distribution to the output of a neural network	◦ relies on the autoencoder to capture the data distribution ◦ shapes a much lower dimensional space into a much simpler distribution
GMMN	**AAE**
◦ (first) trains a dropout autoencoder (then) fits a distribution in the code-space of the pretrained network	◦ uses adversarial training as a regularizer that shapes the code distribution while training the autoencoder from scratch

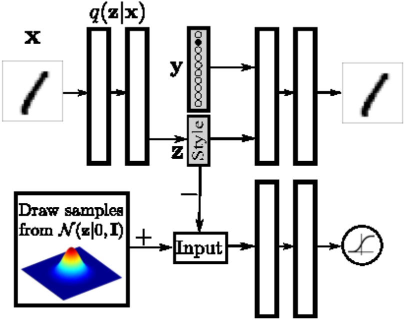
Likelihood analysis

- Benchmarks on: MNIST, Toronto Face Dataset (TFD)
- Not direct likelihood measure, but lower bound approximation (KDE using Gaussian Parzen window, σ selected by cross-validation)
- Samples of 10K and 10M units to estimate the test set log-likelihood

	MNIST(10K)	MNIST(10M)	TFD(10K)	TFD(10M)
DBN	138 ± 2	-	1909 ± 66	-
Stacked CAE	121 ± 1.6	-	2110 ± 50	-
Deep GSN	214 ± 1.1	-	2890 ± 29	-
GAN	225 ± 2	386	2057 ± 26	-
GMMN + AE	282 ± 2	-	2204 ± 20	-
AAE	340 ± 2	427	2252 ± 16	2522

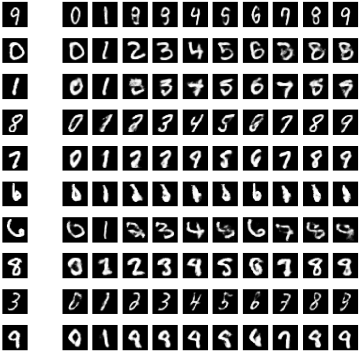
Semi-supervised approach
Architecture

- Incorporate one-hot vector in the latent representation



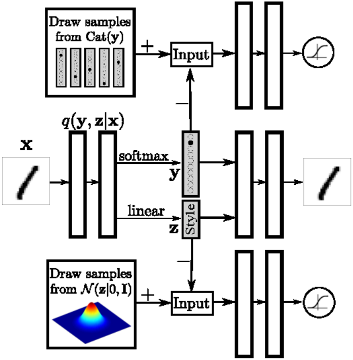
Semi-supervised approach

- The AAE disentangles *style* features from *content/semantic* features
- Experiments on MNIST and Street View House Number dataset (SVHN)



Semi-supervised classification

Architecture



- Improve classification performance using both labeled and unlabeled data
- Assume the latent space is a mixed Categorical and Gaussian distribution

$$p(\boldsymbol{y}) = \mathrm{Cat}(\boldsymbol{y}) \quad p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; 0, \mathbf{I})$$

- For each distribution a different adversarial network regularizes the latent representation

Semi-supervised classification

Training

3-phase training, all with SGD:

1. *reconstruction* — the AE updates the encoder $q(\boldsymbol{z}, \boldsymbol{y}|\boldsymbol{x})$ to minimize the reconstruction error
2. *regularization* — each adversarial network
   - first updates the discriminative network to distinguish true samples from the generated samples
   - then update the encoder to confuse the discriminative networks
3. *semi-supervised classification* — the AE updates $q(\boldsymbol{z}, \boldsymbol{y}|\boldsymbol{x})$ to minimize the cross-entropy cost on a labeled minibatch

Semi-supervised classification

Results

	MNIST(100)	MNIST(1000)	MNIST(All)	SVHN(1000)
NN Baseline	25.80	8.73	1.25	47.5
VAE (M1) + TSVM	11.82 ± 0.025	4.24 ± 0.07	-	55.33 ± 0.11
VAE (M2)	11.97 ± 1.71	3.60 ± 0.56	-	36.02 ± 0.10
VAE (M1 + M2)	3.33 ± 0.14	2.40 ± 0.02	0.96	24.63
CatGAN	1.91 ± 0.1	1.73 ± 0.18	0.91	-
Ladder Networks	1.06 ± 0.37	0.84 ± 0.08	0.57 ± 0.02	-
ADGM	0.96 ± 0.10	-	-	16.61 ± 0.24
AAE	1.90 ± 0.10	1.60 ± 0.08	0.85 ± 0.02	17.70 ± 0.30

Table: Error rate on semi-supervised classification.

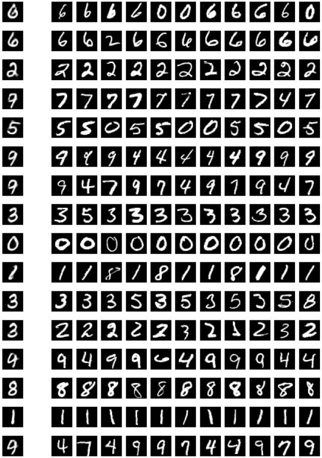
Unsupervised clustering

- Similar to the semi-supervised architecture, but without the semi-supervised training stage
- Not necessarily 10 classes, arbitrary number of clusters
- Benchmark criterion: find $\arg\max_{\boldsymbol{x}_n} q(y_i|\boldsymbol{x}_n)$ and assign label of x_n to all elements of i-th cluster, then compute error rate based on the assigned class labels

	MNIST (Unsupervised)
CatGAN (20 clusters)	9.70
AAE (16 clusters)	9.55 ± 2.05
AAE (30 clusters)	4.10 ± 1.13

Table: Error-rate on unsupervised clustering

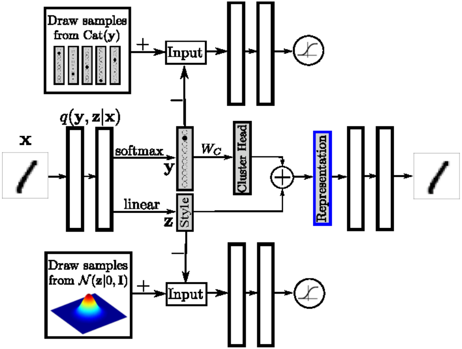
Unsupervised clustering



Dimensionality reduction

Architecture

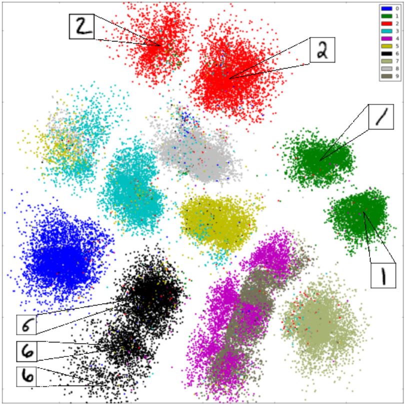
- Typically, non-regularized autoencoders fracture the manifolds ⇒ very different codes for similar images
- Little modification label-integrated AAE architecture

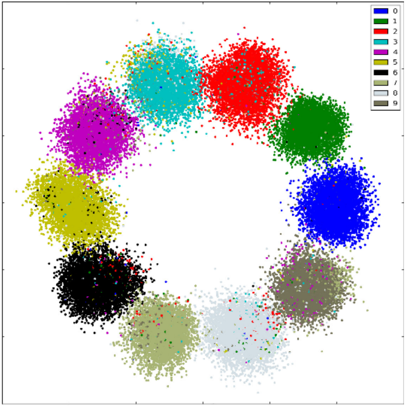


- Additional cost to penalize euclidean distance between cluster heads

Dimensionality reduction

- Reducing the dimensionality of the hidden space (from 10 to 2) can impact on the predictive power of the whole network
- Ad hoc tricks can balance this trade-off





Conclusions

Pros

- it is framework capable of modeling complex distributions only requiring to be able to sample from it
- the AAE is highly flexible, could be combined with variational objectives (see Rosca et al. 2017)

Cons

- like GANs, it requires much hyper-parameter tuning to perform at the top
- it could suffer from complex adversarial game dynamics
- it could be an overkill just to treat Gaussian/Gaussian-mixture distributions

* It would need some real-world testing.

References

- Bengio, Yoshua et al. (2013). "Better mixing via deep representations". In: *International Conference on Machine Learning*, pp. 552–560.
- Bengio, Yoshua et al. (2014). "Deep generative stochastic networks trainable by backprop". In: *International Conference on Machine Learning*, pp. 226–234.
- Goodfellow, Ian et al. (2014). "Generative adversarial nets". In: *Advances in neural information processing systems*, pp. 2672–2680.
- Kingma, Diederik P and Max Welling (2013). "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114*.
- Li, Yujia, Kevin Swersky, and Rich Zemel (2015). "Generative moment matching networks". In: *International Conference on Machine Learning*, pp. 1718–1727.
- Maaløe, Lars et al. (2016). "Auxiliary deep generative models". In: *arXiv preprint arXiv:1602.05473*.
- Makhzani, Alireza et al. (2015). "Adversarial autoencoders". In: *arXiv preprint arXiv:1511.05644*.
- Rosca, Mihaela et al. (2017). "Variational approaches for auto-encoding generative adversarial networks". In: *arXiv preprint arXiv:1706.04987*.