Adversarially Constrained Autoencoder Interpolation using Wasserstein Autoencoder

Machine Learning

Lorenzo Palloni

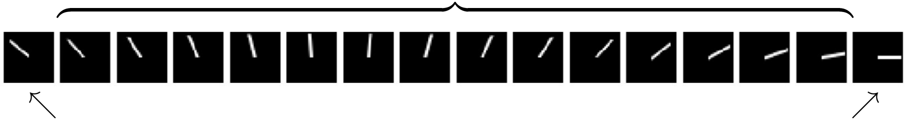
University of Florence

lorenzo.palloni@stud.unifi.it

April 18, 2020

Introduction

- **Unsupervised Learning** context
- We aim to obtain "high-quality" **interpolations**
- An interpolation example:



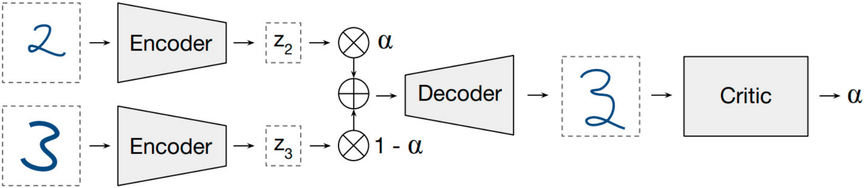
- An "high-quality" interpolation point have two characteristics:
  - is indistinguishable from real data
  - represent a semantically smooth morphing between the endpoints

Motivation and Techniques

- Uncover underlying structure of dataset
- Better representations $\rightarrow$ better results in other tasks

- Implemented techniques (using pytorch [1]):
    - Adversarially Constrained Autoencoder Interpolation (ACAI) [2]
    - Wasserstein Autoencoder (WAE) [3]
    - Wasserstein-Wassertein Autoencoder (WWAE) [4]

- Graphical representation of ACAI structure:



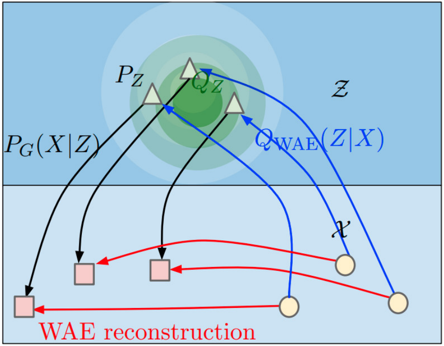
- Loss functions (discriminator and autoencoder respectively):

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

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

WAE

- Graphical representation of WAE structure:



- Loss function:

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

Different penalties, different WAEs (1/3)

- Setting
$$\mathcal{D}_Z(Q_Z, P_Z) = D_{JS}(Q_Z, P_Z)$$
$\rightarrow$ we obtain WAE-GAN
(WAE using a GAN[1]-based penalty)

- where $D_{JS}(\cdot, \cdot)$ is the Jensen-Shannon Divergence:
$$D_{JS}(Q_Z, P_Z) := \frac{1}{2} D_{KL}(Q_Z, \frac{Q_Z + P_Z}{2}) + \frac{1}{2} D_{KL}(P_Z, \frac{Q_Z + P_Z}{2}) \quad (3)$$
- and D_{KL} is the Kullback-Leibler Divergence:
$$D_{KL}(Q_Z, P_Z) := \int Q_Z \log \left(\frac{Q_Z}{P_Z} \right) \quad (4)$$

[1] Generative Adversarial Network [5]

Different penalties, different WAEs (2/3)

- Setting
$$\mathcal{D}_Z(Q_Z, P_Z) = MMD_k(Q_Z, P_Z)$$
$\rightarrow$ we obtain WAE-MMD
(WAE with Maximum Mean Discrepancy as penalty)

- where $MMD_k(\cdot, \cdot)$ is the Maximum Mean Discrepancy with *characteristic* kernel k:
$$MMD_k(Q_Z, P_Z) := \left\| \int_{\mathcal{Z}} k(z, \cdot) dQ_Z(z) - \int_{\mathcal{Z}} k(z, \cdot) dP_Z(z) \right\|_{\mathcal{H}_k}$$

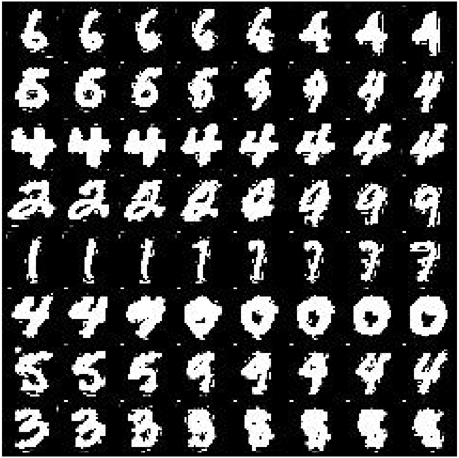
- Setting

$$\mathcal{D}_Z(Q_Z, P_Z) = \underbrace{\|\mu_{P_Z} - \mu_{Q_Z}\|^2 + \mathrm{Tr}\left(\Sigma_{P_Z} + \Sigma_{Q_Z} - 2(\Sigma_{P_Z}\Sigma_{Q_Z})^{\frac{1}{2}}\right)}_{\text{2-Wasserstein distance between two multivariate normal distributions [6]}}$$

$\rightarrow$ we obtain WWAE

(WAE with 2-Wasserstein penalty)

- Example interpolations on MNIST with ACAI + WWAE:



Conclusion

-

Appendix - Wasserstein distance

References

Paszke, Adam, et al. "Automatic differentiation in pytorch." (2017).

Berthelot, David, et al. "Understanding and improving interpolation in autoencoders via an adversarial regularizer." arXiv preprint arXiv:1807.07543 (2018).

Tolstikhin, Ilya, et al. "Wasserstein auto-encoders." arXiv preprint arXiv:1711.01558 (2017).

Zhang, Shunkang, et al. "Wasserstein-Wasserstein auto-encoders." arXiv preprint arXiv:1902.09323 (2019).

Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.

Dowson, D. C., and B. V. Landau. "The Fréchet distance between multivariate normal distributions." Journal of multivariate analysis 12.3 (1982): 450-455.