VARIATIONAL LADDER AUTO ENCODER WITH SSIM LOSS

Alokendu Mazumder

SPECTRUM Lab, Dept. of EE, IISc, Bangalore (PhD, 1^{st} year)

ABSTRACT

Idea of hierarchical learning is applied to generative models , but unfortunately didn't worked that well. In this project we propose an efficient way to stack up latent variables so that the model can be benefited from hierarchical structure along with a modified loss fucntion.

Index Terms— Generative models, hierarchical learning, loss fucntion

1. INTRODUCTION

Stacking generative models on top of each other recursively is one way of introducing hierarchy [1]. Once fully trained, the bottom layer itself contains enough information that can reconstruct the data distribution, making the higher latent codes redundant. They also lack in learning disentangled features. In this work, a ladder based VAE [2] model with SSIM loss term & MMD [3] regularizer is proposed that can learn rich feature hierarchies over images.

2. TECHNICAL DETAILS

Approach: If $\mathbf{z}_i$ is more abstract than $\mathbf{z}_j$, then $q(\mathbf{z}_i|\mathbf{x})$ mapping and generative mapping requires a more expensive network.

2.1. Ladder structure: Encoder

A three layer encoder is used, each having two Conv2D blocks. Latent codes are decomposed like $\mathbf{z}$ = [$\mathbf{z}_1$, $\mathbf{z}_1$, $\mathbf{z}_1$,...]. We choose $q(\mathbf{z}\|\mathbf{x})$ as follows:

$$\mathbf{h}_j = \mathbf{g}_j(\mathbf{h}_{j-1}) \tag{1}$$

$$\mathbf{z}_j \sim \mathcal{N}(\mu_j(\mathbf{h}_j), \sigma_j(\mathbf{h}_j)) \tag{2}$$

where $j = 1, 2, ..., L$, $\mathbf{g}_j$, μ_j, σ_j are neural networks and $\mathbf{h}_0$ is $\mathbf{x}$.

2.2. Ladder structure: Decoder

We choose latent pior, $p(\mathbf{z})$ = $p(\mathbf{z}_1, \mathbf{z}_2,..., \mathbf{z}_L) \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. We define conditional distribution $p(\mathbf{x}\|\mathbf{z}_1, \mathbf{z}_2,..., \mathbf{z}_L)$ as:

$$\begin{aligned} \tilde{\mathbf{z}_L} &= \mathbf{f}_L(\mathbf{z}_L) \\ \tilde{\mathbf{z}_j} &= \mathbf{f}_j([\tilde{\mathbf{z}}_{j+1} : \mathbf{z}_j]) \; \forall j < L \end{aligned} \tag{3}$$

Email: alokendum@iisc.ac.in, SR No: 04-03-00-10-12-21-1-20134

Here, $\mathbf{x}$ is a fixed variance gaussian with mean μ_r=$\mathbf{f}_0(\tilde{\mathbf{z}_1})$ and [:] represents concatenation of two vectors.

2.3. Loss function

For learning, we use the loss as:

$$\boldsymbol{L}(\mathbf{x}) = E_{q(\mathbf{z}|\mathbf{x})}[logp(\mathbf{x}|\mathbf{z})] + MMD[q(\mathbf{z}|\mathbf{x})|\mathbf{p}(\mathbf{z})] + S(\mathbf{x}, \hat{\mathbf{x}}) \tag{4}$$

Here, $\mathbf{p}(\mathbf{z}) \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ is prior of $\mathbf{z}$. $S(,)$ is the SSIM loss defined in [4] between real data and generated one. SSIM loss is used to generate visually pleasant images. MMD is termed as maximum mean discrepancy, it implies that two distributions are same if and only if their all moments are same. It is a similarity measure. We define a MMD based divergence as:

$$\begin{aligned} MMD(p(x)|q(x)) = E_{p(x),p(x')}[k(x, x')] + \\ E_{q(x),q(x')}[k(x, x')] - 2E_{p(x),p(x')}[k(x, x')] \end{aligned} \tag{5}$$

Here, $k(,)$ is the standard RBF kernel.

3. RESULTS

The model is trained & validated over MNIST dataset [5] with 60,000 and 10,000 samples respectively. It is run for 200 epochs with appropriate callbacks. Below are few generated images:

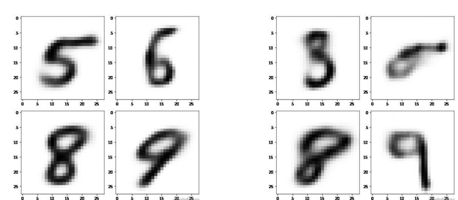

Fig. 1. Results with SSIM Loss (L) & without SSIM Loss (R)

Clearly, model with SSIM loss gives visually better generated images. It has less blur around curvatures.

4. CONTRIBUTION

The codes are re-implemented by myself. Major contribution is the introduction of SSIM loss in loss function to keep visual quality intact and better for generated images. Also, instead of KL divergence, I have used MMD based divergence.

5. RESOURCES

The paper can be found **here**.

6. REFERENCES

[1] Donahue, Jeff, Krahenb uhl, Philipp, and Darrell, Trevor "Adversarial feature learning," arXiv preprint arXiv:1605.09782, 2016.

[2] S. Zhao, J. Song, and S. Ermon "Learning Hierarchical Features from Generative Models," Proceedings of the 34^{th} International Conference on Machine

[3] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola "A kernel two-sample test," JMLR, 2012. Learning, Sydney, Australia, PMLR 70, 2017

[4] H. Zhao, O. Gallo, I. Forosio, and J. Kautz "Loss Functions for Neural Networks for Image Processing," IEEE Transactions on Computational Imaging, 2017

[5] Y. LeCun, C. Cortez, C.J.C Burges "The MNIST database of handwritten digits", 1999.