Variational Ladder Autoencoder with SSIM Loss

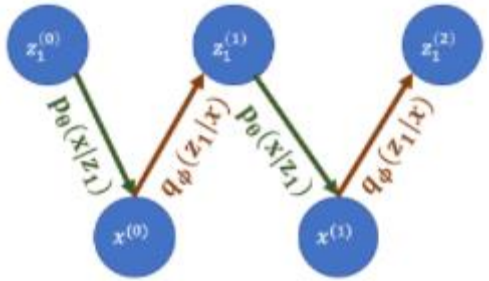
Alokendu Mazumder
PhD (1st Year), SPECTRUM Lab, Dept. of EE
IISc Bangalore

Email: alokendum@iisc.ac.in

SR No: 04-03-00-10-12-21-1-20134

Background

- ✓ Idea of hierarchical learning is applied to generative models ,but unfortunately didn't work that well.
- ✓ One way is, stacking of generative models on top of one another to attain hierarchy.
- ✓ The bottom layer alone contains enough information to reconstruct the data distribution, and the layers above the first one can be ignored.
- ✓ Assume that we already have a perfect generative model ***p*** and its corresponding inference model ***q*** that performs perfect posterior inference at every layer, such that at the first layer $\boldsymbol{p}(\mathbf{z}_1|\mathbf{x})=\boldsymbol{q}(\mathbf{z}_1|\mathbf{x})$. Then it's just a Gibbs sampling chain $\boldsymbol{p}(\mathbf{x}, \mathbf{z}_1)$.



Proposed Approach

- ✓ **Approach** is based on intuition that, if $\mathbf{z_i}$ is more abstract than $\mathbf{z_j}$, then the inference mapping $\boldsymbol{q}(\mathbf{z}_i|\mathbf{x})$ and generative mapping $\boldsymbol{p}(\mathbf{x}|\mathbf{z}_i)$ requires a more expressive network to capture.
- ✓ Hence, rather stacking up, we let the network first learn abstract representations, then map each latent code to one deep layer and sample from there. In this way, each distribution is being conditioned over an abstract representation of layer. As one moves deep in network, we can generate distributions over complex features of input.
- ✓ In order to make visual quality of images intact, I added a SSIM loss term.
- ✓ In VAE's KL divergence acts as regularizer, here I have used MMD loss (described in paper) with RBF kernel.

Technical Details

- ✓ Layers from input onwards are $\mathbf{h_1, h_2, h_3}$ are sequence of Conv2D layers, BN and ReLu activation.
- ✓ The encoder outputs $\mathbf{h_1, h_2, h_3}$ that are given to input of three stochastic layers, which generates $\mathbf{z_1, z_2, z_3}$ respectively. Here $\mathbf{z_j}$ is sampled from gaussian with mean $\boldsymbol{\mu_j(h_j)}$ and standard deviation $\boldsymbol{\sigma_j(h_j)}$.
- ✓ Then $\mathbf{z'_1}$ is obtained from $\mathbf{z_3}$ with three Dense/BN/ReLu layer. $\mathbf{z'_2}$ is output of a neural network ***f***(.) who's input is concatenated $\mathbf{z'_3}$ and $\mathbf{z_2}$. $\mathbf{z'_3}$ is output of same neural network with concatenated inputs $\mathbf{z'_2}$ and $\mathbf{z_1}$,.
- ✓ From $\mathbf{z'_1}$, decoder starts, we keep on increasing the dimension until we reach original.
- ✓ Here ***f***(.) is a single dense layer with ReLu activation.

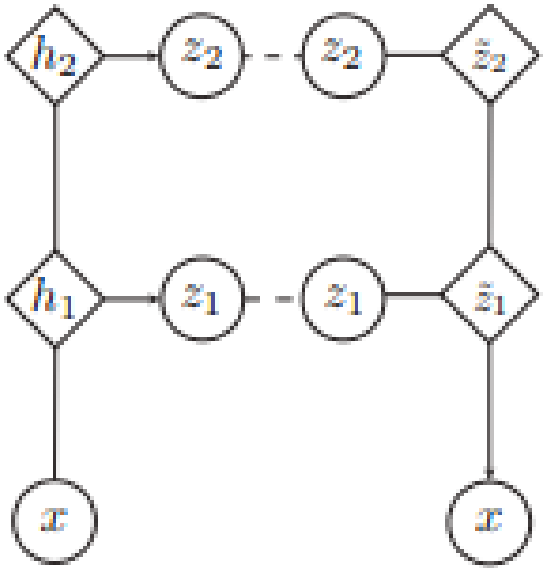


Fig 1: Graphical Model of VLAE

Contributions (Novelty)

- ✓ Unlike simply optimising the ELBO criteria, I introduced an additional loss term, SSIM Loss to make the reconstructed image perceptually better to viewers.
- ✓ Also, instead of KL divergence regularizer, MMD based divergence with RBF kernel is used.
- ✓ These are two major addon’s over baseline paper.

Results & Conclusion

- ✓ The model is tested over MNIST handwritten digits data set with 60,000 training samples and 10,000 validation samples.
- ✓ I trained it once with SSIM Loss and without SSIM Loss in main loss function.
- ✓ The generated images which had SSIM Loss in the model, is more perceptually clear as compared when not trained with SSIM Loss. They tend to have clear curvatures while results obtained without SSIM Loss have more blurs around curvatures.

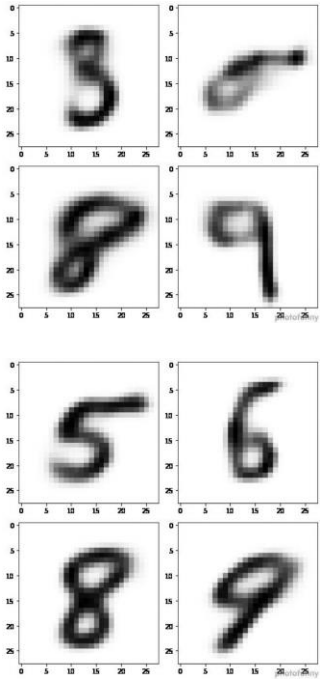


Fig 2: Results without SSIM Loss (Top) & with SSIM Loss (Bottom)