# CS-433 FINAL PROJECT

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Reproducibility of results obtained in research papers is an important part of the research process, as it allows to support the conclusions of the paper, or otherwise find potential inconsistencies in the results. The paper we have chosen to reproduce is called *MAE : Mutual Posterior-Divergence Regularization for Variational Autoencoders*, and aims to improve the performances of Variational Autoencoders (VAE) by constraining their latent variables as to ensure they do contain useful information. VAEs are one of two very powerful generative models and can be useful in many different fields, which is why we decided to study this paper.

## 1 INTRODUCTION

Our research history.

The paper *MAE : Mutual Posterior-Divergence Regularization for Variational Autoencoders* is based on research on Variational Autoencoders (VAEs) and tries to mitigate performance issues araising when the latent representation becomes uninformative, collapsing the model to an unconditional generative model. We had to read multiple papers about previous research on VAEs, as well as different Neural Network models used in the studied paper. We will here introduce those concepts and talk about the need for a better model than simple VAE as described in the first paper introducing the model (Kingma & Welling (2006)).

### 1.1 VARIATIONAL AUTOENCODERS

We learned about VAE, how it works

what's an autoencoder

An autoencoder is a generative model and can be decomposed into two parts : an *encoder* and a *decoder*, which both can be many kinds of networks. The encoder generates a so-called *latent representation* of the original data received as input, which the decoder will then use to try and recover the original information. The goal is then to train the encoder to generate the most informative latent variables, and the decoder to reconstruct the original data as closely as possible. Such a model can be very useful for many reasons : an efficient latent representation can be useful for analysis and understanding of some natural process or for data representation tasks, and when used along with the decoder, it can be used to generate realistic data from small representations.

For the rest of our discussion, let $x$ be an observable random variable which is assumed to be generated from a hidden continuous random variable $z$, following a conditional distribution $p_\theta(x|z)$.

Kingma and Welling in their paper wish to design an autoencoder that works well even on latent variables with intractable posterior distributions $p_\theta(x|z)$, i.e. that cannot be evaluated or differenciated. To this end, they introduce what they call a *recognition model* $q_\phi(z|x)$, an approximation to the true posterior $p_\theta(z|x)$. They also call $q_\phi(z|x)$ the *encoder* and consequently the conditional distribution $p_\theta(x|z)$ the *decoder*.

In fact, they point out that the marginal likelihood $\log p_\theta(x)$ can be lower bound by the following :

$$\log p_\theta(x) \geq \mathcal{L}(\theta, \phi; x) = -D_{KL}\left(q_\phi(z|x)||p_\theta(z)\right) + \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] \tag{1}$$

which they explain has too high a variance to be practical when estimated using a standard Monte Carlo gradient estimator, and where $D_{KL}$ is the KL divergence of two distributions. To solve this problem, they introduce a new lower bound that takes advantage of a *parameterization trick*, which consists in rewriting the distribution of $z$ as a function of a noise variable $\varepsilon$ :

$$z \sim q_\phi(z|x) = g_\phi(\varepsilon|x) \tag{2}$$

where $\varepsilon$ follows an appropriate distribution $p(\varepsilon)$. This makes it possible, after replacing the lower bound in 1, to apply a Monte Carlo estimator on the second term, yielding :

$$\tilde{\mathcal{L}}(\theta, \phi; x) = -D_{KL}\left(q_\phi(z|x)||p_\theta(z)\right) + \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(x, g_\phi(\varepsilon_l, x)) \tag{3}$$

where $L$ is some sample size and $\varepsilon_l$ is a sample from the distribution of $\varepsilon$.

### 1.2 Autoencoding capabilities of VAEs

Read about VLAE, which explains well why VAE is not enough. Talk about KL-varnishing at some point

As a followup reading, cited in our paper of interest about MAEs, we looked at *Variational Lossy Autoencoders* (Chen et al. (2017)), where we found good insight as to why VAEs on their own are not always sufficient for autoencoding.

They note that when the decoder is made too expressive, it is easy for the model to set the approximate posterior $q_\phi(z|x)$ to equal the prior $p_\theta(z)$ and hence avoid any cost from the regularizing term $D_{KL}(p_\phi(z|x)||p_\theta(z))$ in 3. Although this is explained by "optimization challenges" of VAE in most research , they present another observation that, even on a perfectly optimized VAE, ignoring the latent variables should still result in optimum results in most instances of VAE with intractable true posterior and powerful enough decoders.

### 1.3 ConvNet

Finally, as most experiments in our paper of interest make use of Neural networks for the VAE encoder and decoder, we describe here in essence what they are. Note that we focus on Convolutional Neural Networks (ConvNet) rather than Residual Neural Networks (ResNet), even though the latter is also used in the paper. We unfortunately have had technical and timing constraints, and Resnet being a fairly big model, we decided to focus on having good results on ConvNet alone.

**Regular Neural Nets** A Neural Network is composed of a sequence of hidden layers, each consisting of neurons which connect to all neurons of the previous layer. The first layer takes a single vector as input, and the last layer outputs the result. For example, in a classification problem, it outputs a score for each class.

**Convolutional Neural Nets** An issue with Regular Neural Networks is the fact that each neuron must be connect to all neurons of the previous layer, which makes the model impractically big for large input data. Convolution Neural Networks (ConvNet) take advantage of the fact that, in inputs that represent images, localized information in the image can be enough for learning. Each layer organizes its neurons in a three dimensions (width, height, depth) and each neuron is not constrained to be fully connected to the previous layer. The first layer then usually has the same dimensions as the input images, and for the example of a classification problem with $c$ classes, the output data will have dimension $1 \times 1 \times c$.

## 2 Experiments

### 2.1 Our journey towards MAE implementation

As a first step, we wanted to reproduce the VAE model. In order to do this we used the pytorch framework. We generated an encoder decoder structure, with the encode and the decoder as one layer

### 2.2 Experiment results

### 2.3 Evaluating our experiments

## 3 Conclusion

Ouverture, future work.

- extend MAE to other forms of data, in particular text (on which VAEs suffer a more serious KL-varnishing problem).
- Hypespherical Variational Auto-Encoders link. Use another distribution than Gaussian to get different latent space (hypespherical structured latent space), better suited for some types of data.

## References

Xi Chen, Diederik Kingma, Time Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder, mar 2017. arXiv:1611.02731v2.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes, may 2006. arXiv:1312.6114v10.