Mat-VAE

CS726: Advanced Machine Learning

Group: ClosedAI

Advait Padhye, 200100014
Pulkit Paliwal, 20D100021
Sanchit Jindal, 200020120
Lakshya Gupta, 200050066

Project Report

Indian Institute of Technology, Bombay

Contents

1 Introduction

Representation Learning or Feature Learning is an important subfield in Machine Learning which is concerned with learning representations of the data that make it easier to extract useful information when building classifiers or other predictors. In the case of probabilistic models, a good representation is the one that captures the posterior distribution of the underlying explanatory factors for the observed input. These learned representations play a crucial role in modern ML systems, serving various downstream tasks such as classification or retrieval. Upon the advent of Deep Learning, several new techniques have been developed to learn representations from all sorts of data. For eg. Convolutional Neural Networks have been used to extract features from input images which are then used for various applications such as object recognition, edge detection. One such deep learning model that is used to learn representations is Variational Autoencoder which is the focus of this project.
An ideal representation should be expressive, meaning that a reasonably-sized learned representation should capture a huge number of possible input configuration. In addition, the process of learning representations is expected to not be computationally expensive.

1.1 Balancing Learning and Inference in VAEs

Variational autoencoders (VAEs) are a deep learning technique for learning latent representations. VAEs consist of an encoder network which models the posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$ and a decoder network which models the likelihood distribution $p_\theta(\mathbf{x}|\mathbf{z})$. They have been used for several applications such as dimensionality reduction and image generation.

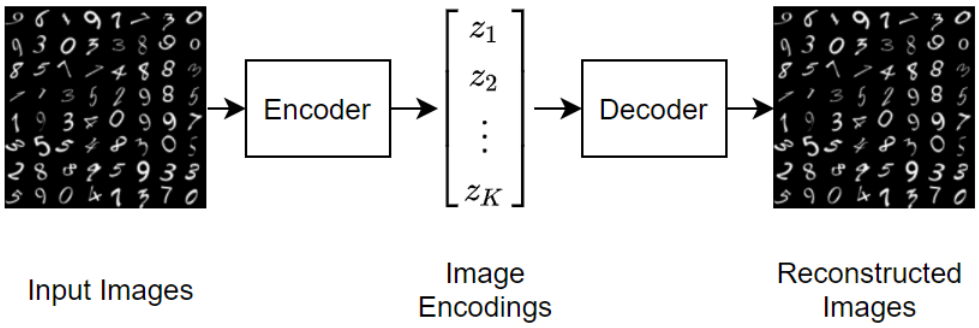

Figure 1: Variational Autoencoder Architecture

The classical Variational Autoencoders are trained using the Evidence Lower Bound (ELBO) objective. After training the VAE on the ELBO objective, the learned $q_\phi(z|x)$ can be used to generate the representation. The representation for a given x' can be obtained by either sampling from the posterior, or finding the most likely representation. However, in practice, it has been found that the approximate inference distribution learnt by VAEs is often significantly different from the true posterior. Zhao et al. [2] claims that this problem is rooted in the ELBO objective.

Another problem pointed out by the authors is that when the conditional distribution is sufficiently expressive, the latent variables are often ignored. The ELBO objective

encourages message $q_\phi(z \mid x)$ to be a random sample from $p(z)$ for each input x making the message uninformative about the input. Thus, if the decoder is very flexible, then a trivial strategy of only producing from $p(z)$ can globally maximize the ELBO objective leading to uninformative latent representations.

1.2 Obstacles in Representation Learning

Often times, when training these representations, it's common to encounter unknown computational and statistical constraints for each downstream task. For example, in a client-server setting, the server might have the necessary computational resources to handle data of 128 dimensions however, the client may not possess the same and can handle only 32-dimensional data. In such cases, we often restrict the capabilities of our models to meet the requirements of the user with the least resources. This results in under-utilization of resources for all other users in the system and a significant drop in the overall performance of the model. This issue can be resolved if the information present in the input data could be encoded at different granularities in a single embedding. However, the existing deep learning models tend to diffuse information across the entire representation vector due to their gradient-based training. In order to overcome this, Kusupati et al. [1] presented Matryoshka Representation Learning (MRL) which encodes information at different granularities and allows a single embedding to adapt to the computational constraints of downstream tasks.

The modification allows for usage of smaller embedding sizes with a lower loss of performance as compared to training an individual model with a smaller embedding size. The authors demonstrated the performance of Matryoshka Representation for classification tasks by modifying the ResNet-50 model and comparing its performance on ImageNet-1k dataset.

2 Problem Formulation

This project seeks to integrate Matryoshka Representation Learning and InfoVAEs to create a VAE model capable of delivering quality flexible representations. With the help of MRL, we intend to overcome the need of training multiple low-dimensional models and instead have a singular model which produces a single embedding that can be deployed adaptively at no additional cost during inference. We showcase the effectiveness of MRL-MMD-VAEs by evaluating their performance against individually trained MMD-VAEs and standard VAEs in a basic classification task using the MNIST, Fashion-MNIST and CIFAR10 datasets. For implementing these models, we primarily referred to the following Github repositories:

1. InfoVAE
2. MRL

3 Approach

3.1 InfoVAEs

Our first goal was to improve the latent space representations for better performance in downstream classification task. In order to overcome these limitations of VAE, we implemented InfoVAE, a new family of Variational Autoencoders proposed by Zhao et al. [2]. Instead of the classical ELBO objective, InfoVAE uses Maximum Mean Discrepancy (MMD) objective which was shown to address the issues of uninformative latent code and variance over-estimation in feature space.
In particular, the authors used Maximum Mean Discrepancy (MMD) instead of KL Divergence to quantify the distance between two distributions. MMD can be efficiently implemented using the kernel trick. Let $k(.\,,.)$ be any positive definite kernel then,

$$\begin{aligned}\mathcal{L}_{\text{MMD}}(q\,||\,p) &= \mathbb{E}_{p(z),p(z')}[k(z,\,z')] \\ &+ \mathbb{E}_{q(z),q(z')}[k(z,\,z')] \\ &- 2\mathbb{E}_{q(z),p(z')}[k(z,\,z')]\end{aligned}$$

In our project, we utilized Gaussian Kernel with $\sigma = 1$ to implement Maximum Mean Discrepancy.

$$k(z,\,z') = e^{-\dfrac{||\,z - z'\,||^2}{2\sigma^2}}$$

For computing the MMD distance, we first simply generated n samples from the prior distribution $p(z)$ and compared these generated samples with the encoder output.

3.2 Matryoshka Representation Learning

Our objective in the latter half of the project would be to make the representations learnt from VAEs flexible. In order to achieve this goal, we implement Matryoshka Representation Learning proposed by Kusupati et al. [1] in the VAE and InfoVAE architecture.

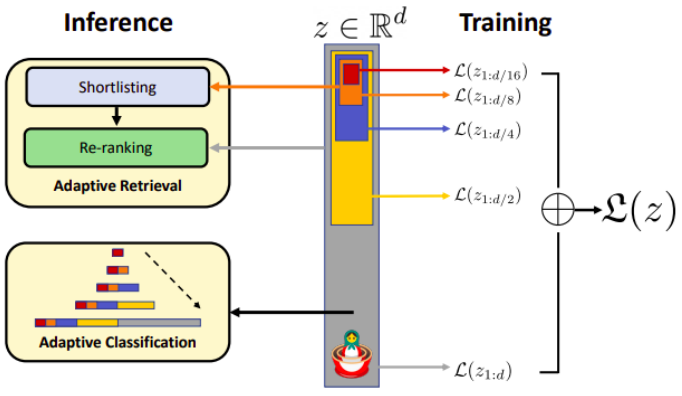

Figure 2: Matryoshka Representation Learning

3.2.1 Preliminaries

For $d \in \mathbb{N}$, consider a set $\mathcal{M} \subset [d]$ of representation sizes. For a datapoint x in the input domain $\mathcal{X}$, our goal is to learn a d-dimensional flexible representation vector $z \in \mathbb{R}^d$ using a deep neural network $F(\ \cdot\ ; \theta_F) : X \to \mathbb{R}^d$ parameterized by weights θ_F. For every $m \in \mathcal{M}$, Matryoshka Representation Learning (MRL) enables each of the first m dimensions of the embedding vector, $z_{1:m} \in \mathbb{R}^m$ to be independently capable of being a transferable and general purpose representation of the datapoint x. Suppose we are given a labelled dataset $\mathcal{D} = \{(x_1, y_1), ..., (x_N, y_N)\}$ where $x_i \in \mathcal{X}$ is an input point and $y_i \in [L]$ is the label of x_i. MRL optimizes the multi-class classification loss for each of the nested dimension $m \in \mathcal{M}$ using a separate linear classifier, parameterized by $W^{(m)} \in \mathbb{R}^{L \times m}$

$$\min_{\theta_F,\ \{W^{(m)}\}_{m \in \mathcal{M}}} \frac{1}{N} \sum_{i \in [N]} \sum_{m \in \mathcal{M}} \mathcal{L}(W^{(m)} \cdot F(x_i\ ; \theta_F)_{1:m}\ ;\ y_i)$$

3.2.2 MRL in VAE architecture

We adopt the objective of Matryokshka Representation Learning to the Variational Autoencoder architecture in the following way :-

1. The Encoder network $q(\ \cdot\ ; \phi)$ is taken as $F(\ \cdot\ ; \theta_F)$.
2. For every $m \in \mathcal{M}$, we use a separate Decoder network $p(\ \cdot\ ; \theta_m)$
3. This new VAE architecture is trained upon the following objective :-

$$\min_{\phi,\ \{\theta_m\}_{m \in \mathcal{M}}} \frac{1}{N} \sum_{i \in [N]} \sum_{m \in \mathcal{M}} \mathcal{L}_{\text{vae}}(x_i,\ q(x_i\ ; \phi)_{1:m},\ p(q(x_i\ ; \phi)_{1:m}\ ; \theta_m))$$

After training the Encoder Network, we train a classifier separately for every $m \in \mathcal{M}$ using the first m dimensions of the representation $z = q(x\ ;\ \phi)$

4 Experimental Details

4.1 InfoVAE Architecture

For the downstream classification task, we used a feedforward network consisting of 2 linear layers. The encoder and decoder networks of VAE consist of 2 convolutional layers and 2 linear layers. The classifier and VAE have been implemented in PyTorch and we have used Adam Optimizer for training the models. We varied the size of latent representations as [2, 8, 16, 32, 64, 128]. VAE and classifier have been trained for 20 epochs with a batch size of 32. We implemented an early stopping module to prevent overfitting. For evaluation, we employed 5-fold cross validation.

4.2 Matyrokshka Representation Learning

For implementing MRL, we used the same architecture for classifier and VAEs. We varied the size of latent representations as [2, 8, 16, 32, 64, 128]. VAE and classifier have been

trained for 20 epochs with a batch size of 32. We implemented an early stopping module to prevent overfitting. We compared the results of MRL with individually trained models. We have provided results on 3 datasets :-

1. MNIST
2. FashionMNIST
3. CIFAR10

5 Results

5.1 MNIST

Embedding Size	Mean Accuracy	Best Accuracy
2	10.442	10.59
8	10.464	10.89
16	10.276	10.51
32	10.526	10.7
64	10.17	10.38
128	10.388	10.56

Table 1: VAE Sampling

Embedding Size	Mean Accuracy	Best Accuracy
2	12.088	15.46
8	15.888	23.05
16	19.73	21.34
32	17.122	20.12
64	22.156	26.87
128	30.332	39.95

Table 2: VAE MAP

Embedding Size	Mean Accuracy	Best Accuracy
2	10.262	10.44
8	94.938	95.19
16	97.586	97.77
32	98.138	98.22
64	98.288	98.45
128	98.458	98.61

Table 3: MMD Sampling

Embedding Size	Mean Accuracy	Best Accuracy
2	11.662	17.19
8	95.422	96.19
16	97.918	98.04
32	98.352	98.43
64	98.464	98.55
128	98.498	98.6

Table 4: MMD MAP

Embedding Size	Mean Accuracy	Best Accuracy
2	10.238	10.47
4	10.244	10.43
8	10.562	10.94
16	10.44	11.03
32	10.314	10.514
64	9.937	10.25
128	10.196	10.64

Table 5: VAE-MRL Sampling

Embedding Size	Mean Accuracy	Best Accuracy
2	10.18	10.28
4	9.536	9.8
8	9.404	9.8
16	9.622	10.09
32	9.81	10.09
64	9.844	10.1
128	9.956	10.1

Table 6: VAE-MRL MAP

Embedding Size	Mean Accuracy	Best Accuracy
2	15.012	16.01
4	50.074	53.98
8	90.596	92.49
16	96.367	96.69
32	97.982	98.23
64	98.412	98.56
128	98.428	98.58

Table 7: MMD-MRL Sampling

Embedding Size	Mean Accuracy	Best Accuracy
2	13.791	25.48
4	43.344	54.61
8	89.411	92.07
16	96.295	96.52
32	97.978	98.2
64	98.4	98.56
128	98.436	98.52

Table 8: MMD-MRL MAP

5.2 Fashion-MNIST

Embedding Size	Mean Accuracy	Best Accuracy
2	10.14	10.46
8	10.032	10.35
16	10.104	10.62
32	10.258	10.46
64	10.144	10.39
128	10.036	10.52

Table 9: VAE Sampling

Embedding Size	Mean Accuracy	Best Accuracy
2	10.2	10.83
8	14.012	17.54
16	19.908	22.54
32	21.31	28.11
64	22.386	29.61
128	26.742	40.46

Table 10: VAE MAP

Embedding Size	Mean Accuracy	Best Accuracy
2	9.816	9.97
8	82.248	83.16
16	86.484	86.78
32	88.338	88.56
64	88.99	89.12
128	89.206	89.44

Table 11: MMD Sampling

Embedding Size	Mean Accuracy	Best Accuracy
2	68.204	69.38
8	83.514	83.74
16	86.562	86.83
32	88.256	88.42
64	89.166	89.24
128	89.354	89.68

Table 12: MMD MAP

Embedding Size	Mean Accuracy	Best Accuracy
2	9.912	10.23
4	9.898	10.17
8	10.027	10.61
16	9.833	10.2
32	10.214	10.43
64	9.874	10.34
128	10.074	10.77

Table 13: VAE-MRL Sampling

Embedding Size	Mean Accuracy	Best Accuracy
2	40.007	55.81
4	69.076	70.86
8	78.876	79.63
16	83.124	83.77
32	85.528	86.27
64	86.522	86.94
128	86.794	87.24

Table 14: MMD-MRL Sampling

Embedding Size	Mean Accuracy	Best Accuracy
2	36.594	55.6
4	68.679	71.85
8	79.11	79.74
16	83.262	83.9
32	86.24	86.24
64	86.5	86.88
128	86.833	87.21

Table 15: MMD-MRL MAP

5.3 CIFAR-10

Embedding Size	Mean Accuracy	Best Accuracy
2	10.124	10.38
8	9.876	10.15
16	10.24	10.64
32	10.01	10.47
64	10.172	10.87
128	10.236	10.54

Table 16: VAE Sampling

Embedding Size	Mean Accuracy	Best Accuracy
2	10.022	10.81
8	10.242	10.77
16	10.578	12.45
32	11.75	13.11
64	11.726	13.43
128	13.884	17.36

Table 17: VAE MAP

Embedding Size	Mean Accuracy	Best Accuracy
2	9.984	10.58
8	9.878	10.15
16	30.154	36.2
32	41.466	42.44
64	51.034	51.42
128	53.276	53.83

Table 18: MMD Sampling

Embedding Size	Mean Accuracy	Best Accuracy
2	12.178	15.68
8	16.224	21.94
16	37.658	38.19
32	44.106	45.45
64	51.348	51.64
128	53.648	53.89

Table 19: MMD MAP

Embedding Size	Mean Accuracy	Best Accuracy
2	10.09	10.52
4	10.064	10.28
8	9.898	10.12
16	10.122	10.38
32	9.98	10.17
64	9.93	10.28
128	9.64	10.26

Table 20: VAE-MRL Sampling

Embedding Size	Mean Accuracy	Best Accuracy
2	12.128	12.71
4	19.308	20.41
8	26.94	27.92
16	33.788	36.0
32	38.814	40.72
64	41.023	43.62
128	41.968	44.36

Table 21: MMD-MRL Sampling

Embedding Size	Mean Accuracy	Best Accuracy
2	10.924	13.18
4	15.95	18.89
8	26.532	29.24
16	36.278	36.93
32	40.394	41.21
64	42.857	44.65
128	43.62	45.82

Table 22: MMD-MRL MAP

6 Conclusions

We implemented VAE and InfoVAE with sampling and MAP queries, along with their MRL variations. We ran our experiments on the aforementioned three datasets and observed that InfoVAEs outperformed VAEs, as expected. We also obtained comparable results for MRL-implementation of InfoVAEs and VAEs with their standard counterparts. All-in-all, we observed that MRL works well for InfoVAEs and provide us with a flexible and better representation as compared to classical Variational Autoencoders.

References

[1] Kusupati, A., Bhatt, G., Rege, A., Wallingford, M., Sinha, A., Ramanujan, V., Howard-Snyder, W., Chen, K., Kakade, S., Jain, P. and Farhadi, A., 2022. Matryoshka representation learning. Advances in Neural Information Processing Systems, 35, pp.30233-30249.

[2] Zhao, S., Song, J. and Ermon, S., 2017. Infovae: Information maximizing variational autoencoders. arXiv preprint arXiv:1706.02262.