Machine Learning Project: Multimodal VAE
— Milestone 1 —

Pablo López 453865
Mohammad Orabe 402831
Mert Akil 473470

December 15, 2023

Contents

1 MVAE: Conceptual Summary

Variational Autoencoders (VAE's) is a type of explicit generative model with an encoder and decoder structure. The encoder (eq. (3)) takes an input sample x and maps it to a latent space representation denoted by z trough a probabilistic mapping. The probabilistic mapping consist of the parameters, μ and σ of a multivariate Gaussian distribution in latent space.

$$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x)) \tag{1}$$
$$p_\theta(x|z) = \mathcal{N}(\mu_\theta(z), \sigma_\theta(x)) \tag{2}$$

The decoder (eq. (4)) takes a sampled latent code z and reconstructs the input data point x again and is trained to map the latent space back to the input space. Hence, the VAE has the form $p_\theta(x, z) = p(z)p_\theta(x|z)$, where $p(z)$ is the prior, which is usually a spherical Gaussian. In that sense the goal of the VAE is to maximize the marginal likelihood of the data, which is intractable. Therefore, the evidence lower bound (ELBO) is optimized instead, where the decoder serves as a tractable importance distribution.

In machine learning, Variational Autoencoders (VAEs) are a noteworthy category of explicit generative models. Their structured architecture involves an encoder and decoder, offering a compelling approach for capturing latent representations of input data. The encoder, as defined in Eq. 3, undertakes the task of mapping an input sample x to a latent space representation denoted as z. This mapping is probabilistic, characterized by the parameter μ and σ, which define a multivariate Gaussian distribution in the latent space.

$$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x)) \tag{3}$$
$$p_\theta(x|z) = \mathcal{N}(\mu_\theta(z), \sigma_\theta(x)) \tag{4}$$

On the other side of the VAE architecture is the decoder, articulated in Eq. 4. The decoder takes a sampled latent code z and endeavors to reconstruct the original input data point x. Its training is centered around mapping the latent space back to the input space. Conceptually, the VAE can be expressed as $p_\theta(x, z) = p(z)p_\theta(x|z)$, with $p(z)$ representing the prior, often modeled as a spherical Gaussian distribution. In essence, the vAE aims to maximize the marginal likelihood of the data, a task rendered intractable. Consequently, the optimization process focuses on the Evidence Lower Bound (ELBO), wherein the decoder functions as a tractable importance distribution.

$$\text{ELBO} = E_{q_\phi(z|x)}[\log p_\theta(x|z)] - \text{KL}[q_\phi(z|x), p(z)] \tag{5}$$

The crux of the VAE is encapsulated in the ELBO equation (Eq. 5). The first ter, serves as the reconstruction loss, quantifying the fidelity of the model in reconstructing the input data. This crucial metric gauges the models ability to faithfully recreate images from their latent space representatoins. Meanwhile, the second term represents the Kullback-Leibler divergence between the approximate posterior and the prior. This KL-divergence term plays a pivotal role in regulating the learned latent space, encouraging its proximity to a standard distribution.

Upon the completion of the training process, the model is not only able to reconstruct an image but also possesses the remarkable capability to generate entirely new data points from the learned latent Gaussian distribution. However, it is important to note that this capability is currently limited to a singular modality.

Herein lies the innovation of the Multimodal VAE (MVAE); it extends this idea to multiple modalities using a unified model. The modalities, denoted as $x_1, \ldots, x_N$ are assumed to be conditionally independent given the common latent variable z. Consequently, the generative model adopts the form $p_\theta(x_1, x_2, \ldots, x_N) = p(z)p_\theta(x_1|z)p_\theta(x_2|z)\ldots p_\theta(x_N|z)$. Accounting for this conditional independence among modalities, the ELBO takes on a modified structure as can be seen in Eq. 6. The terms λ_i and β are weights balancing the terms in the ELBO, where $\lambda = 1$ and β is slowly annealed to 1.

$$\text{ELBO} = E_{q_\phi(z|X)}[\sum_{x_i \in X} \lambda_i \log p_\theta(x|z)] - \beta \text{KL}[q_\phi(z|X), p(z)], \tag{6}$$

In order to train the the inference networks $q(z|X)$ the authors present a relationship among joint- and single-modality posteriors (Eq. 7).

$$p(z|x_1, \ldots, x_N) \propto \frac{\prod_{i=1}^{N} p(z|x_i)}{\prod_{i=1}^{N-1} p(z)} \approx p(z) \prod_{i=1}^{N} \tilde{q}(z|x_i) \tag{7}$$

This expression illuminates a profound insight: product of experts can be employed, complemented by a prior expert, as the approximating distribution for the joint posterior. However, the product distribution is typically non-trivial to solve in closed form. Fortunately, when the prior $p(z)$, and the approximating posterior $\tilde{q}(z|x_i)$ are Gaussian it can be solved as *the product of Gaussians is itself a Gaussian.* The precise calculation for the mean and covariance in the product of experts can be seen in Eq. 9 and a visual representation is shown in Figure 1.

$$\mu = \frac{\sum_i \mu_i T_i}{\sum_i T_i} \tag{8}$$

$$V = \frac{1}{\sum_i T_i} \tag{9}$$

This section provides an insight into the methodology employed to develop the MVAE model. Upon thorough examination of its architecture, the authors determined that MVAE achieves state-of-the-art results on four bi-modal datasets during the time of publication—specifically, MNIST, FashionMNIST, MultiMNIST, and CelebA. Given the aim of this section to emphasize the methodology and core concepts of MVAE, we encourage readers to delve into the original paper for a more detailed exploration of the experiments and evaluation (Wu and Goodman (2018)).

2 MNIST Dataset

Mike Wu and *Noah Goodman* used five different data-sets for the implementation of the Multimodal Variational Autoencoder. These are: MNIST, BinaryMNIST,

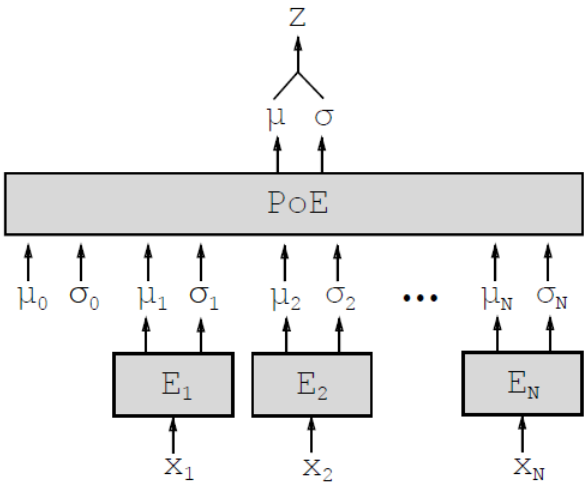


Figure 1: Architecture of MVAE with N modalities. Each E_i corresponds to the i-th inference network. The parameters μ_0 and σ_0 denote the prior parameters, whereas μ_i and σ_i represents the i-th variational parameter.

FashionMNIST, MultiMNIST and CelebA. For the project's initial milestone, we will only study the implementation for the MNIST data-set – a widely recognized collection of handwritten digit images which comprises 60,000 training images and 10,000 testing images. The enduring popularity of MNIST arises from its simplicity and clarity: each image has a resolution of 28x28 pixels and portrays a distinct single digit (ranging from 0 to 9) originated from diverse contributors. Consequently, all the images are labeled with its corresponding digit. The intensity values of each pixel within the images range from 0 to 255, representing the darkness of the pixel. The following figure illustrates a sample image for each digit number.

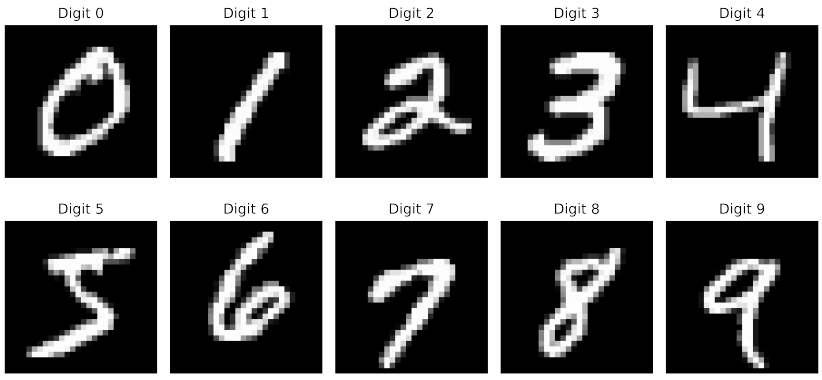


Figure 2: Sample images from MNIST Data-set

The resolution of the images facilitates a straightforward visualization and interpretation of the results. One can easily examine the generated images and latent space representations to get valuable insights into the model's learning performance. All in all, the MNIST data-set has become a standard benchmark in machine learning. Hence, we first test and validate our model on it before moving to more complex data-sets such as FashionMNIST or CelebA.

To extend the MNIST data-set for use in a Multimodal Variational Autoencoder (MVAE), we can introduce a second modality to enrich the information available for each digit. In this instance, the second modality is comprised of the labels assigned

to the images which are contained in the data-set itself. Therefore, no additional modifications are required for the data-set to be implemented in an MVAE.

The existing labels serve as a readily available and inherent source of categorical information. This inclusion enables the MVAE to jointly model the visual features of the handwritten digits along with the information provided by their associated labels.

3 Our Approach

As explained in the preceding sections, our strategy is to implement a MVAE on the MNIST data-set within a bi-modal setting. Accordingly, we consider two different modalities ($N = 2$): x_1 representing the image modality and x_2 as the label modality. In this initial milestone, our primary focus is to gain a comprehensive understanding of this bi-modal framework and to define the architecture of the model for the chosen data-set.

3.1 Defining the model architecture.

To implement a bi-modal VAE on the MNIST, we need to design the architecture of two distinct Variational Autoencoders. Specifically, we must establish a encoder and a decoder dedicated to processing images (corresponding to the x_1 modality) and an other VAE for interpreting labels (associated to the x_2 modality).

Similar to the architecture presented in the paper, we propose the following framework.

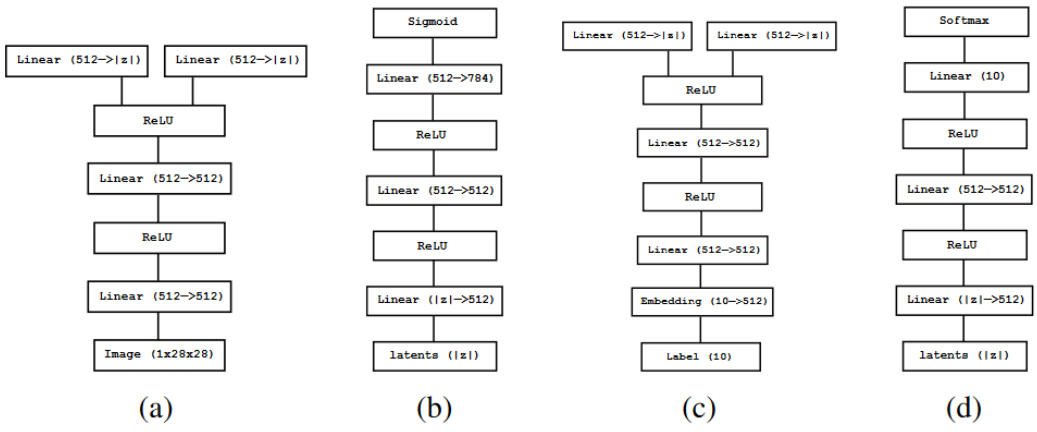


Figure 3: Used MVAE architecture on MNIST

As represented in Figure 3, we define two networks (Encoder-Decoder structure) with 2 hidden layers of 512 units each and set $p(x_1|z)$ as Bernoulli and $p(x_2|z)$ as a multinomial distribution (categorical distribution). The models (a) and (b) constitute the architecture for the image VAE given by $q(z|x_1)$ and $p(x_1|z)$ where x_1 denotes a fixed image and z its representation in the latent space. Moreover, (c) and (d) delineate the specifications for the VAE in the second modality, modeled by $q(z|x_2)$ and $p(x_2|z)$ with x_2 as the label. Since the dimensionality of the latent space (portrayed as $|z|$ in Figure 3) was not specified in the paper, we decided to set it as a hyper-parameter for the model and set it to 20 by default.

3.2 Evaluation metrics.

While the training phase has not been executed yet, we also set our focus in this milestone towards comprehending the potential methods/metrics for evaluating the eventual results, i.e. the generated images. The authors did explain in detail this evaluating process for the bi-modal setting. We distinguish this process in two types: quantitative and qualitative. In this milestone, we will only study the evaluation quantitatively and leave the qualitative analysis to be studied in the next milestone where we will implement the training phase.

3.2.1 Quantitative Evaluation.

For evaluating the results in a quantitative manner, we estimate the test log-likelihoods which capture the ability of the model to learn the data distribution and its conditionals. Hence, it can be roughly seen as the negative reconstruction error since a higher probability means a better capability of the model for generating proper samples and converting between the diverse modalities. In Suzuki et al. (2016) the following estimates for the log-likelihoods were presented:

$$\log p(x_1) \approx \log E_{q(z|x_1)}\left[\frac{p(x_1|z)p(z)}{q(z|x_1)}\right] \tag{10}$$

$$\log p(x_1|x_2) \approx \log E_{q(z|x_2)}\left[\frac{p(x_1|z)p(x_2|z)p(z)}{q(z|x_2)}\right] - \log E_{p(z)}[p(x_2|z)] \tag{11}$$

where $\log E_{p(z)}[p(x_2|z)]$ in eq. (11) can be estimated via $\log \frac{1}{n}\sum_i^n p(x_2|z_i)$ with $z_i \sim p(z)$ and n denoting the number of test images (convergence can be easily proven with the law of large numbers).

Since we assumed the modalities x_1 and x_2 to be conditionally independent, we can also estimate the test joint log-likelihood $\log p(x_1, x_2)$ using eq. (10).

4 Discussion

In our endeavor to implement the Multimodal Variational Autoencoder (MVAE) method, following the approach outlined in the original paper with some modifications, we are currently immersed in comprehending the presentation of multimodal datasets, with a specific focus on the MNIST dataset, and constructing a model prototype. At this milestone, our primary objective was to implement the foundational model and explore its complexities. The development process is facilitated using PyTorch.

Presently, our model has not undergone training; nevertheless, substantial progress has been made in our implementation efforts. We've loaded the data, conducted visualizations for inspection, and implemented preprocessing steps, including loading the MNIST dataset, normalizing pixel values, and other necessary shape transformations. The existing prototype on the main branch on GitHub includes essential components like the encoder, decoder, reparameterization trick, and the product-of-experts (PoE) structure, resulting in a comprehensive forward pass. A notable divergence from the original paper in our implementation lies in our choice of activation functions in the model architecture. For simplicity, we opted for the ReLU activation function instead of the Leaky ReLU function in the forward pass for both the encoder and decoder across the two modalities. We want to emphasize that, up to this point, the training process has not been initiated in the submitted

implementation. Our focus on a fully supervised method has yet to address missing data or modalities.

One initial challenge we faced during the model implementation was effectively managing multiple modalities, particularly in the process of combining images and labels in the initial MNIST dataset. As we progress, we anticipate encountering additional complexities in the next phases of our project. This will be especially relevant when working with more intricate datasets such as CelebA (a large-scale dataset for face recognition by Liu et al. (2015)), and exploring other case studies like machine translation with a weakly supervised presentation of datasets.

Our early findings indicate that the MVAE model shows promise in learning joint representations effectively, especially in extracting meaningful information from diverse modalities. In practical terms, we expect our implementation to demonstrate robustness in handling incomplete supervision, making it suitable for scenarios where only a small subset of examples contains all required observations. Our ongoing analysis focuses on adapting the implementation to address challenges in weakly-supervised learning. Moving forward, our exploration includes experimenting with larger and more diverse datasets such as CelebA to understand MVAE's scalability and generalizability. The immediate next steps involve continued implementation, training with varying hyperparameters, model evaluation, and comprehensive testing.

5 Code and Data Availability

Our project's codebase and relevant datasets are accessible on GitHub. You can find the latest code, implementations, and project updates in the main branch of our GitHub Repository.

References

Z. Liu, P. Luo, X. Wang, and X. Tang. Celeba: A large-scale dataset for face recognition, 2015.

M. Suzuki, K. Nakayama, and Y. Matsuo. Joint multimodal learning with deep generative models, 2016.

M. Wu and N. Goodman. Multimodal generative models for scalable weakly-supervised learning, 2018.

Machine Learning Project: Multimodal VAE
— Milestone 2 —

Pablo López 453865
Mohammad Orabe 402831
Mert Akil 473470

January 8, 2024

Contents

1 Introduction

In our initial milestone, we introduced the Multimodal Variational Auto-encoder, a progressive extension of the well-established VAE. We outlined our strategy approach for its implementation within a bi-modal context using the MNIST data-set. Our efforts involved defining the architecture of the model and providing valuable insights into the training and evaluation processes.

Now, in this pivotal second milestone, our primary goal is to implement the training phase for the MVAE on the MNIST data-set. The details of this process will be explained in the upcoming chapter titled *Methodology*. The successful realization of this phase, coupled with the implementation of a sampling method, will mark the achievement of a fully functional MVAE for this specific data-set.

2 Methodology

We let the prior distribution $p(z)$ and the approximate posterior $q_\theta(z|x_i)$ be defined as in the first milestone. There, we also explained the calculation process with the Product of Experts to understand how the joint parameters are computed. In this section, we will define the whole generative model for the MNIST setting and specify how the ELBO loss is practically implemented to train the model.

2.1 Neural Networks as probabilistic decoders

In Multimodal Variational Auto-encoders, neural networks are used as probabilistic decoders. The used architecture consists of two networks with 2 hidden layers of 512 units each. Moreover, we model the image decoder $p(x_1|z)$ with a (multivariate) Bernoulli likelihood and the label decoder $p(x_2|z)$ with a multinomial (categorical) likelihood.

2.1.1 Image Decoder

For the image decoder $p(x_1|z)$, we use a multi-layered perceptron (MLP) with a Bernoulli likelihood. Practically, given an image $x_1 \in \mathbb{R}^{784}$ and the latent variable z from its distribution $p(z)$, this corresponds to the decoder

$$p_\theta(x_1|z) = \prod_{i=1}^{784} \mu_i^{x_1}(1-\mu_i)^{(1-x_1)}$$

where μ is the mean of the multivariate Bernoulli distributed random variable x_1 and, practically, is computed as the output of the Auto-encoder network given x_1 (Note: in our code it is named as *gen-image*).

Since it models a probability distribution, the outputs must lie between 0 and 1. To ensure that, we make use of a sigmoid layer at the output.

Moreover, the logarithm of $p_\theta(x_1|z)$ is used for the computation of the ELBO loss. It turns out to be the binary cross-entropy loss, which reads

$$\log p_\theta(x_1|z) = \sum_{i=1}^{784} x_1 \log \mu_i + (1-x_1)\log(1-\mu_i).$$

2.1.2 Label Decoder

For the label decoder $p(x_2|z)$, we use a MLP with a multinomial likelihood, in particular a categorical likelihood since we consider 10 different classes and 1 single try. Hence, we model it as the categorical cross-entropy loss with 10 classes.

It consists of a Log Softmax and a cross-entropy loss. That reads

$$\log p_\theta(x_2|z) = -\sum_{i=1}^{10} t_i f(z)_i$$

where

$$f(z)_i = \log\left(\frac{e^{z_i}}{\sum_{l=1}^{10} e^{z_l}}\right)$$

defines the Log Softmax activation function (which corresponds to *F.log_softmax* in PyTorch).

In our specific case of Multi-Class classification, the labels are one-hot encoded. That means that there is only one unique element of the Target vector t which is not zero. So discarding the elements of the summation which are zero due to target labels, we can simplify to

$$\log p_\theta(x_2|z) = -\log\left(\frac{e^{z_p}}{\sum_{i=1}^{10} e^{z_i}}\right)$$

where p denotes the only positive class.

2.2 Computation of the KL divergence

Recall the presented ELBO loss for the multimodal setting

$$ELBO(X) := \mathbb{E}_{z\sim q_\theta(z|X)}[\sum_{x_i\in X} \lambda_i \log p_\theta(x_i|z)] - \beta KL[q_\theta(z|X)||p(z)].$$

In the MNIST case, this can be rewritten to

$$ELBO(X) := \mathbb{E}_{z\sim q_\theta(z|X)}[\lambda_1 \log p_\theta(x_1|z) + \lambda_2 \log p_\theta(x_2|z)] - \beta KL[q_\theta(z|X)||p(z)].$$

After establishing the computation of the likelihoods $p_\theta(x_1|z)$ and $p_\theta(x_2|z)$ within the model, the remaining key aspect to comprehend for an understanding of the ELBO loss calculation is how the Kullback-Leibler (KL) divergence is computed.

The paper Kingma and Welling (2022) presents an easy solution for its calculation. Since

$$\begin{aligned} -KL(q_\theta(z|X)||p(z)) &= \int q_\theta(z|X)(\log p(z) - \log q_\theta(z|X))\, dz \\ &= \int q_\theta(z|X) \log p(z)\, dz - \int q_\theta(z|X) \log q_\theta(z|X)\, dz \end{aligned}$$

one can easily proof using the Gaussian assumptions that

$$-KL(q_\theta(z|X)||p(z)) = \frac{1}{2}\sum_{i=1}^{D}(1 + \log \sigma_i^2 - \mu_i^2 - \sigma_i^2)$$

where μ and σ denote the mean and variance of the input data X and are outputs of the model (*forward function*), and D is the latent dimensionality which is defined in the upcoming chapter.

2.3 Model selection

In this section, we outline the hyperparameter choices crucial for training our machine learning model, as they significantly impact its performance. The selected hyperparameters aim to strike a balance between model expressiveness and generalization capabilities.

2.3.1 Lambda Weights and Beta Term

Our model incorporates two lambda weights, denoted as λ_1 and λ_2, alongside the beta term (β) within the Evidence Lower Bound (ELBO) term. λ_1 and λ_2 are employed to weigh the reconstruction error, with a higher weight assigned to the low-dimensional modality. This practice aligns with established methodologies, promoting effective learning of the joint distribution (Wu and Goodman (2018)). The beta term is gradually annealed to 1 over the first 200 epochs to ensure a valid lower bound on the evidence.

2.3.2 Standard Hyperparameters

In addition to the specialized hyperparameters mentioned above, we employ standard settings for fundamental aspects of our model:

- **Train-Val-Test Split Sets:** Following the approach detailed in the last milestone, our training phase involves 50,000 sample images, with separate sets of 10,000 samples each allocated for validation and testing. This distribution is consistent with the standard setup for the MNIST dataset, providing diversity in samples to enhance the model's generalization capacity.
- **Model Optimizer:** We choose the Adam Optimizer with a learning rate of 10^{-3}, a widely recognized selection for tasks of this nature due to its empirically proven effectiveness.
- **Batch Size:** The input batch size for training is set to 100, as per the main paper's recommendation.
- **Latent Dimensions:** We define the latent embedding with 64 dimensions, a common choice for the MNIST dataset. This dimensionality effectively captures underlying representations while mitigating overfitting.
- **Number of Training Epochs:** We set the number of training epochs to 500.

These hyperparameter choices are consistent with the original implementation, aligning with the authors' settings (Wu and Goodman (2018)). This decision was deliberate, aiming to initially replicate the model as closely as possible to validate expected results. Subsequent iterations of the project may involve exploring alternative hyperparameter configurations for further refinement.

3 Sampling Method

This section provides a detailed exploration of the Sampling Method employed by the (MVAE), specifically applied to the MNIST dataset.

Given the generative nature of Variational Autoencoders (VAEs) and Multimodal Variational Autoencoders (MVAE), the trained models possess the capability to generate data from the learned distribution. Given a context, where the MVAE has learned a joint latent space for images and labels of the MNIST dataset, the model can generate corresponding images and labels. Four distinct approaches are employed for the sampling process.

The first approach utilizes the unconditioned sampling method, denoted as $z \sim p(z)$. In this method, random samples are drawn from a normal distribution using the normal mean (0) and standard deviation (1). These samples are then input into the MVAE decoder, generating new images and labels.

The second approach involves sampling conditioned on images, denoted as $z \sim q(z|x_1)$. Given an input image representing a specific number, the corresponding mean and standard deviation are obtained by passing the image through the encoder and calculating the Product of Experts. Subsequently, random samples are generated from a normal distribution using this mean and standard deviation. These samples are then fed into the MVAE decoder, producing labels that represent the numbers in the images.

The third approach is analogous to the second, with the distinction that sampling is now conditioned on labels, denoted as $z \sim q(z|x_2)$. This process results in generated images that represent the label content.

The fourth and final approach involves sampling conditioned on both image and label, denoted as $z \sim q(z|x_1, x_2)$. Similar to approaches 2 and 3, an image and a label are used to derive the mean and standard deviation for sampling, providing a comprehensive method for generating multimodal data.

4 Evaluation metrics

In the previous milestone, we explored towards comprehending the potential methods and metrics for evaluating the generated images, particularly within this bimodal setting. We differentiated this evaluation process into two categories: quantitative and qualitative.

4.1 Quantitative Evaluation

We briefly recall from the preceding milestone our approach for evaluating the results quantitatively, and analyze our actual outcomes in this manner.

To capture the ability of the model to learn the data distribution and its conditionals, we estimate the marginal probabilities $\log p(x_1)$, $\log p(x_1|x_2)$ and $\log p(x_1, x_2)$. An adequate estimation for their computation reads

$$\log p(x_1) \approx \log \mathbb{E}_{z\sim q(z|x_1)}\left[\frac{p(z|x_1)p(z)}{q(z|x_1)}\right] \tag{1}$$

$$\log p(x_1|x_2) \approx \mathbb{E}_{z\sim q(z|x_2)}\left[\frac{p(x_1|z)p(x_2|z)p(z)}{q(z|x_2)}\right] - \log \mathbb{E}_{z\sim p(z)}[p(x_2|z)] \tag{2}$$

$$\log p(x_1, x_2) = \log p(x_1)p(x_2) \tag{3}$$

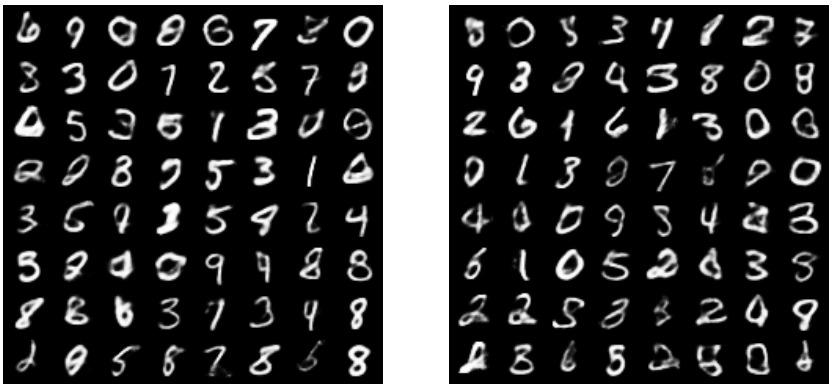
As of this report, the quantitative evaluation, including the computation of marginal probabilities, has not been implemented. The computational estimation outlined above remains pending, and this aspect of the evaluation will be addressed in the subsequent phases of our project.

4.2 Qualitative Evaluation

Assessing the outcomes in a qualitative manner involves a subjective evaluation based on visual inspection, circumventing the need for mathematical computations. In the context of our model, the Qualitative Evaluation leverages the sampling method for both the unconditional case and the case conditioned on the label. The emphasis here lies on visualizing sampled image results.

For the unconditional generation, Figure 1.a illustrates the sampled images. Notably, the generated images aptly reflect numbers in the range 0-10. However, upon closer inspection, certain flaws emerge, especially in samples generating the digit 0. Distinguishing between 0 and 8 becomes challenging due to a perceptible level of noise. While real numbers are indeed generated, a nuanced presence of noise persists.

Moving to conditioned sampling for $x_2 = 5$ in Figure 1.b, it becomes apparent that the model fails to accurately reflect the number 5 in the images. The generated results even exhibit images that are difficult to interpret as any discernible numbers. The intricacies behind these outcomes will be elucidated and explored further in Section 5.



(a) (b)

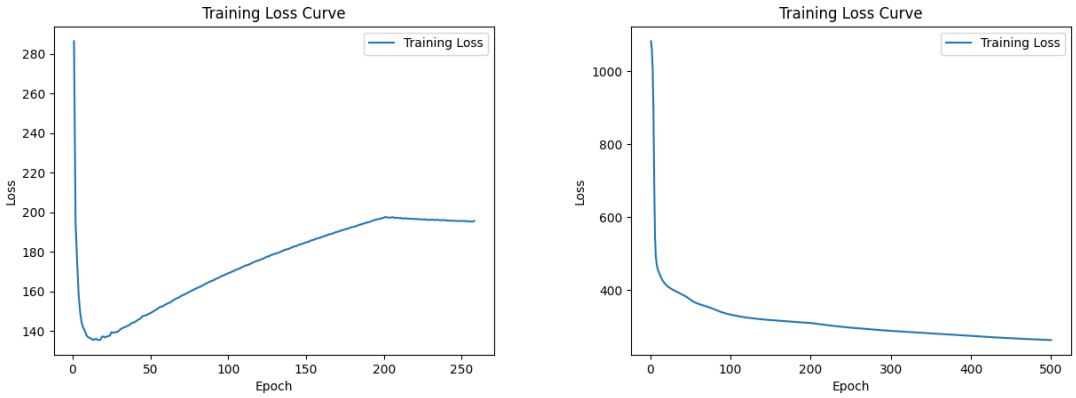
Figure 1: 64 images sampled using MVAE by (a) sampling $z \sim p(z)$ and generating them via $p(x_1|z)$, and by (b) sampling conditional image by $z \sim q(z|x_2 = 5)$

5 Further Discussion

In this section, we delve into the challenges and issues encountered while training our Multimodal Variational Autoencoder (MVAE).

5.1 Image Generation and Conditional Sampling

Observing the results of image generation through sampling from the trained model, we note that the unconditional images exhibit satisfactory quality, accurately representing digits ranging from 0 to 10. However, challenges arise when examining



(a) (b)

Figure 2: Curve of the training loss for (a) learning_rate= 10^{-3}, and (b) learning_rate= 10^{-6}

samples conditioned on text, specifically for the case when $x_2 = 2$. In these instances, the generated images do not consistently reflect the intended digit, revealing a limitation in the conditional generation process.

5.2 Divergence in Training

A notable challenge emerged during training when utilizing a learning rate of 10^{-3}. After the 250th epoch, both the training and validation losses exhibited NaN (Not a Number) values, indicative of a divergence issue. This is evident in the training loss curve illustrated in Figure 2. The model consistently diverges after a few epochs, indicating potential instability.

Interestingly, training the model with a lower learning rate of 10^{-6} led to a different outcome. While the model did not exhibit divergence, the convergence was notably slow. This prompts a reevaluation of the learning rate hyperparameter and its impact on training dynamics. Despite the authors' recommendation of 10^{-3}, our observations suggest a need for further exploration of alternative learning rates in future iterations.

5.3 Next Steps

In the upcoming phases, we plan to address the aforementioned challenges and take the following steps:

- **Hyperparameter Exploration:** Given the divergence and slow convergence issues, we aim to experiment with various learning rates to identify a balance that ensures stability and efficiency in training.
- **Model Debugging:** In addition to the Hyperparameter Exploration, we further aim to inspect the model during training to identify potential sources of divergence and inefficiency. This involves scrutinizing intermediate representations, examining gradients, and monitoring layer-wise activations.
- **Implementation of Evaluation Metrics:** To gauge the model's performance accurately, we intend to implement the evaluation metrics mentioned

in Section 4.1 (insert relevant reference). These metrics will provide a quantitative assessment of the model's effectiveness in capturing multimodal dependencies.

6 Code and Data Availability

Our project's codebase and relevant datasets are accessible on GitHub. You can find the latest code, implementations, and project updates in the main branch of our GitHub Repository.

Note: Unfortunately we overlooked creating a pull request for the recent commits in the ongoing (second) milestone, instead we merged the branches that contained the implementations. As a result, the option to generate a pull request for those specific branches was lost. The new commits associated with the current milestone starting from commit hash (9595ce5) mark the new continuation of our work.

References

D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2022.

M. Wu and N. Goodman. Multimodal generative models for scalable weakly-supervised learning, 2018.

Machine Learning Project: Multimodal VAE
— Milestone 3 —

Pablo López 453865
Mohammad Orabe 402831
Mert Akil 473470

February 2, 2024

Contents

1 Introduction & Outline

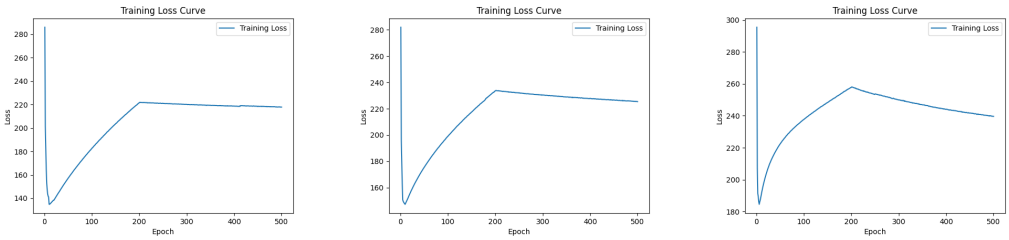
In the previous milestone, we detailed the execution of the training phase of a MVAE within a bi-modal context for the MNIST data-set. Furthermore, we developed a sampling function to qualitatively assess the model using four distinct approaches. Consequently, we were able to quantify the model's capacity to generate new samples in both unconditional and conditional scenarios.

This final milestone centers on addressing challenges encountered in the initial two phases. We will elaborate on how these issues were effectively resolved, particularly in comparison to the original paper Wu and Goodman (2018). We will hence propose a final version of our MVAE for the MNIST data-set. Additionally, we delve into a detailed exploration of the quantitative evaluation. Acquiring a deeper understanding will ultimately allow us to numerically evaluate the generated samples.

2 Addressing the Divergence Problem

To briefly revisit our previous milestone, we encountered a divergence issue when training the model with a learning rate of 1e-3. Subsequently, employing a learning rate of 1e-6 resulted in non-convergence, albeit at an exceptionally sluggish pace.

Upon careful analysis of the outcomes, a pivotal observation emerged: the model appears to exhibit effective learning at a learning rate lower than 1e-3, as evidenced by the absence of divergence at 1e-6. Consequently, we are optimistic that the implementation of a learning rate scheduler could serve as a potential solution to mitigate the divergence problem. Additionally, we plan to incorporate gradient clipping, setting the threshold to 1.0 when necessary to further enhance the model's stability and convergence.



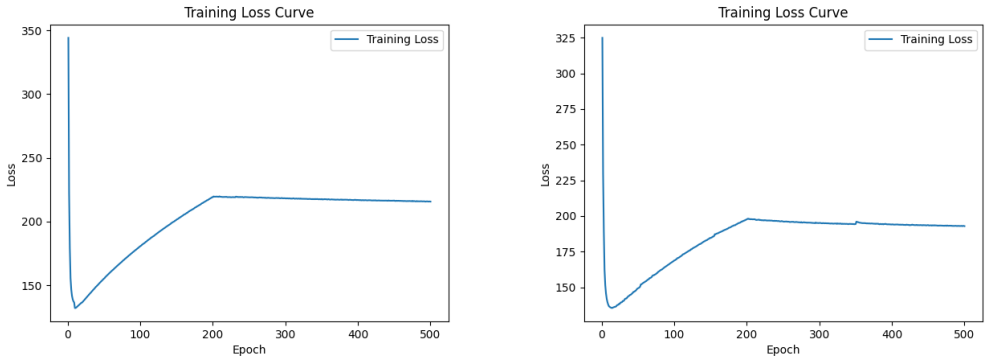
(a) (b) (c)

Figure 1: Training loss trajectories with a learning rate scheduler, gradually reducing the initial 1e-3 learning rate by a factor of 10 every (a) tenth, (b) fifth, and (c) third epoch until reaching 1e-6.

Initiating the learning rate scheduler with an initial setting to decrease the learning rate every 10 epochs, dividing by 10 until reaching $1e-6$, was our initial approach. The learning rate trends are visualized in Fig. 1a. Surprisingly, the model still exhibits divergence around the same epoch as the baseline model utilizing a fixed learning rate of $1e-3$. In response, we experimented with alternative configurations, training two additional models where the learning rate scheduler is invoked every fifth (Fig. 1b) and every

(a) (b) (c)

Figure 2: Generated samples from the model with a learning rate scheduler adjusting the initial learning rate every (a) tenth, (b) fifth, and (c) third epoch until reaching 1e-6.



(a) (b)

Figure 3: Evolution of training loss under the influence of gradient clipping and a learning rate scheduler, systematically diminishing the initial 1e-3 learning rate by a factor of 10 every (a) tenth, and (b) third epoch until reaching 1e-6.

third epoch (Fig. 1c). Regrettably, these adjustments did not break the pattern of divergence in the learning curve. Furthermore, an examination of the generated samples from the three models reveals a deviation from the expected images, despite conditioning on the label 5 (Fig. 2).

In a subsequent effort, we reintroduced the same learning rate scheduler while incorporating gradient clipping, with a constraint set at 1. This dual strategy was employed with a learning rate scheduler modifying the rate every third (3b) and tenth (3a) epoch until the learning rate reached $1e-6$. Despite these adjustments, the recurrent pattern of divergence persisted. Intriguingly, the learning rate scheduler's specific configuration seemed immaterial; after the initial epochs, the model consistently veered into divergence. Analyzing the generated samples from these models further corroborates this, as the desired images, akin to those in Fig. 2, remained elusive (Fig. 4).

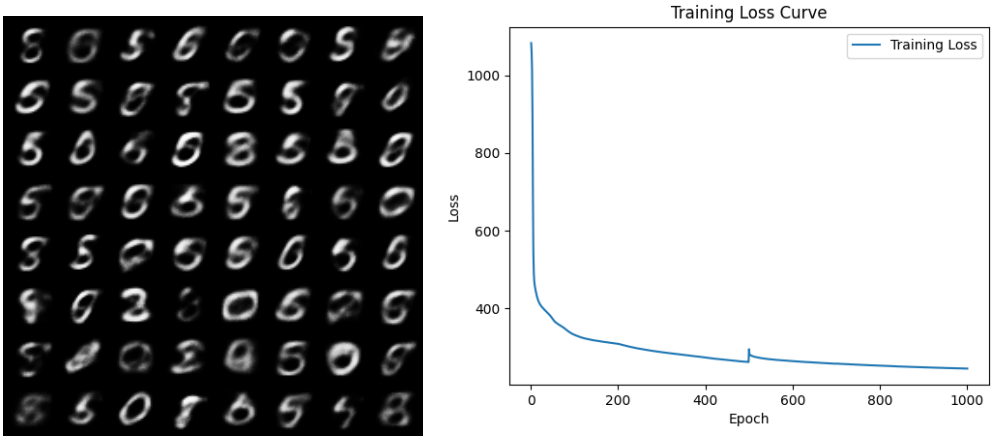
Recalling the model's gradual convergence with a learning rate of $1e-6$, an unconventional yet effective approach was employed. The model, initially trained with a learning rate of $1e-6$, was extended for an additional 500 epochs using a comparatively higher learning rate of $1e-3$. While this strategy

(a) (b)

Figure 4: Generated samples from the model incorporating gradient clipping and a learning rate scheduler, fine-tuning the initial learning rate every (a) tenth and (b) third epoch until reaching 1e-6.

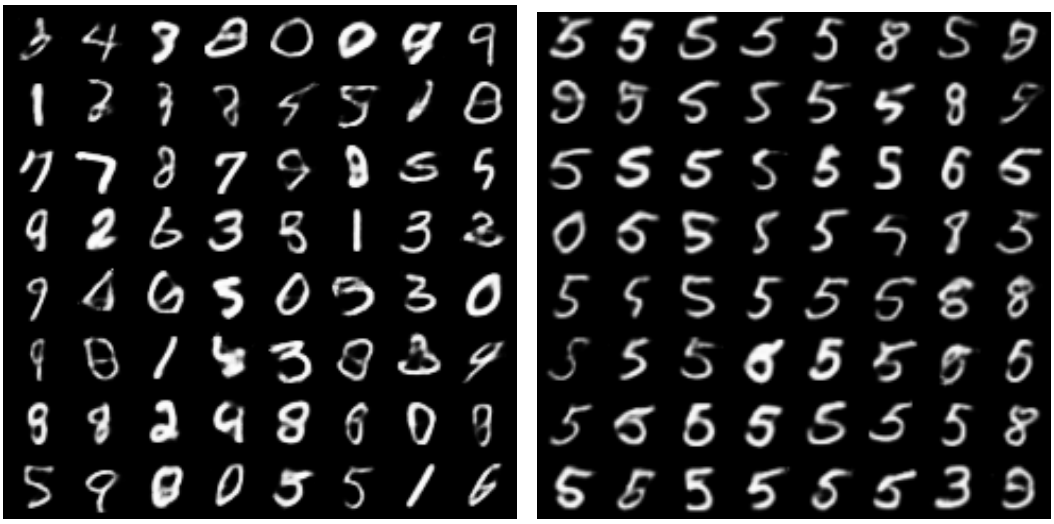
might seem counterintuitive and deviates from conventional technical wisdom, it yielded promising results. As depicted in Fig. 5b, the model exhibited convergence, and the quality of the generated samples (Fig. 5a) surpassed those generated under alternative conditions. However, it's important to note that, despite these improvements, the generated samples still fall short of the benchmark set by the results published in the referenced paper (Wu and Goodman (2018)).

Following a series of unsuccessful attempts, we turned to the repository provided by the authors (Wu (2018)) of the paper for a comprehensive sanity check. Intriguingly, even in their original implementation, the model encountered divergence during training. Notably, the generated samples from their repository (Fig. 6a) exhibited similarities to those produced by our models, diverging from the anticipated results showcased in the paper (Fig. 6b). This unexpected alignment in challenges highlights the complexity of the task and suggests a potential intrinsic difficulty in achieving the desired outcomes, even with the authors' own implementation.



(a) (b)

Figure 5: (a) Generated samples and (b) Training curve for the model trained over 500 epochs with a learning rate of 1e-6, followed by an additional 500 epochs with a learning rate of 1e-3.



(a) (b)

Figure 6: (a) Generated samples from the model trained using the official repository, and (b) Samples presented in the paper for comparison.

3 Evaluation

In the preceding report, we left the implementation of quantitative evaluation and the computation of the marginal probabilities for future implementation.

During this milestone, our emphasis is hence directed towards gaining a thorough understanding of the quantitative evaluation of the inferred probability measure and the samples generated from it. This will allow us not only to ultimately evaluate the samples when generated, but to use the defined probabilities as the loss functions for the testing step of the model. In this manner, we will obtain better final results. This will be explained in detail in the next section 4.

3.1 Quantitative Evaluation

To quantitatively evaluate the model's ability to grasp the data distribution and its conditionals, we proceed in the same way as in the paper and estimate the logarithmic probabilities for the marginal events $\log p(x_1)$ and $\log p(x_1, x_2)$. While various quantitative evaluation methods such as Inception Score or Kernel Density Estimation are available for generative models, they demand additional effort and comprehension. Consequently, we will exclusively assess our model by calculating the log-likelihoods, as done in Wu and Goodman (2018).

The likelihoods $p(x_1)$ and $p(x_1, x_2)$ serve as indicators of how effectively the generative model comprehends and emulates the inherent distribution of the data. These probabilities represent the model's assessment of the probability associated with the individual image data point x_1 and the joint probability of the sequence x_1, x_2, respectively. Evaluating these likelihoods provides insight into the model's capacity to capture the intricate patterns within the data, gauging its ability to mimic the underlying data distribution. Lately, we perform the logarithmic function to simplify calculations and avoid numerical instability for very small probabilities.

While the authors in Wu and Goodman (2018) do not provide details on how the approximations are derived, we express our interest in understanding their derivation. Specifically, there is no equation presented for the computation of the joint probability $p(x_1, x_2)$. Consequently, we undertook additional research and effort to derive a meaningful and effective approximation. In the following, we provide a detailed description of an approximate computation of the different marginal probabilities, through variational inference.

We propose the approximates

$$\begin{aligned}
\log p(x_1) &= \mathbb{E}_{z\sim q(z|x_1)}[\log p(x_1)] = \log \mathbb{E}_{z\sim q(z|x_1)}[p(x_1)] \\
&= \log \mathbb{E}_{z\sim q(z|x_1)}\left[\frac{p(x_1,z)}{p(z|x_1)}\right] \\
&\approx \log \mathbb{E}_{z\sim q(z|x_1)}\left[\frac{p(x_1,z)}{q(z|x_1)}\right] \\
&= \log \mathbb{E}_{z\sim q(z|x_1)}\left[\frac{p(x_1|z)p(z)}{q(z|x_1)}\right] \\
&= \mathbb{E}_{z\sim q(z|x_1)}[\log p(x_1|z) + \log p(z) - \log q(z|x_1)]
\end{aligned}$$

and

$$\begin{aligned}
\log p(x_1,x_2) &= \log \mathbb{E}_{z\sim q(z|x_1)}[p(x_1,x_2)] \\
&= \log \mathbb{E}_{z\sim q(z|x_1)}\left[\frac{p(x_1,x_2,z)}{p(z|x_1,x_2)}\right] \\
&\approx \log \mathbb{E}_{z\sim q(z|x_1)}\left[\frac{p(x_1,x_2,z)}{q(z|x_1,x_2)}\right] \\
&= \log \mathbb{E}_{z\sim q(z|x_1)}\left[\frac{p(x_1,x_2|z)p(z)}{q(z|x_1,x_2)}\right] \\
&\approx \log \mathbb{E}_{z\sim q(z|x_1)}\left[\frac{p(x_1|z)p(x_2|z)p(z)}{q(z|x_1)}\right] \\
&= \mathbb{E}_{z\sim q(z|x_1)}[\log p(x_1|z) + \log p(x_2|z) + \log p(z) - \log q(z|x_1)].
\end{aligned}$$

3.2 Hypothetical Use Case

Consider the following hypothetical scenario that sheds light on the practicality of our model:

Let us assume that a human needs x seconds on average with perfect performance to perform the generation and that generating a sample using our method takes 0 seconds.

For this simplistic dataset, we find ourselves in a unique position where, given an input label, we can effortlessly generate an image with the corresponding label and vice versa. Given the inherently generative nature of our model and the project's scope, quantifying the accuracy of the generated samples becomes a nuanced challenge. In this context, we propose a use case where a human expert reviews each generated result, accepting or rejecting it and correcting errors as needed. Examining Fig. 5a, we generously recognize 25 samples as the number 5, yielding an accuracy of $25/64 \approx 39\%$. Similarly, we assume the accuracy of label generation to be around $\sim 39\%$, leading to a consistent analysis. In a scenario where the accepted error rate is set at 0%, the human expert would invest x seconds in correcting errors 61 out of 100 times. As the accepted error rate increases, the human's time investment decreases linearly. This implies that even at a 39% accuracy rate, our model could potentially save a substantial amount of time, roughly $(39 + \text{err\_rate})\%$ of the total time. For instance, at an accepted error rate of 61%, which is less than ideal, human interaction would still be minimal or non-existent.

Considering that our model currently specializes in generating images featuring a single numeric content and accompanying textual numbers, the immediate appeal may seem limited. Realistically, the process may take just a few seconds at worst.

However, the true potential of this architectural framework transcends its current application. Imagine its adaptation for diverse domains, such as generating facial images from textual descriptions of facial features or conceptualizing novel shoe designs based on specific textual attributes. The versatility of this approach extends beyond numbers, presenting exciting possibilities. In practical terms, envision the time-saving impact in such domains. For instance, generating intricate facial images or innovative shoe designs could potentially save several hours of manual design work. This represents a significant leap forward and underscores the substantial gains achievable through the utilization of this model in various creative and practical contexts.

4 Outlook

We evaluated the mean marginal log- likelihood, $\log p(x_1)$, and joint log-likelihood, $\log p(x_1, x_2)$, across a range of samples (1 - 5000) selected randomly from the test set. In examining joint likelihood, we considered conditions on images, labels, and both combined, observing values between 200,000 and 900,000 for both joint and marginal log-likelihoods without a clear relationship to varying evaluation parameters. Although we anticipated these measures to reflect the model's capacity to approximate the data distribution and its conditionals—implying that higher scores indicate better model performance in generating accurate samples and transitioning between modalities—our findings starkly contrasted with those reported in Wu and Goodman's work. Their results, in the range of -90, suggest a discrepancy in magnitude by thousands and are notably negative, prompting us to rely on the ELBO loss for model evaluation during training (as detailed in our 'training.py' script's validation function).

Upon comparing samples generated by our model to those from the author's official implementation and those presented in their paper, we observed significant differences. Despite these discrepancies, the samples produced by our model were of high quality, with conditional samples accurately reflecting the intended labels, opening questions about the reported test results on marginal and joint log-likelihoods from the original study.

Furthermore, as the best performing model required training over 1000 epochs, this duration might be considered insufficient, particularly when compared to the 500 epochs used by the original authors. This discrepancy suggests that extending the training period could potentially yield weights that better capture the underlying data patterns, even if initial training phases exhibit divergence without subsequent recovery of training loss.

Finally, we found a noticeable lack of resources on multimodal generative models, which significantly complicated our efforts, particularly when outcomes diverged from our expectations. This scarcity underscores the inherent difficulties in implementing both the project and the concepts discussed in the paper, highlighting the challenges faced in navigating this project.

5 Code Availability

Our project's codebase and the trained model are accessible on GitHub. You can find the latest code, implementations, and project updates in the main branch of our GitHub Repository.

Note that some of the updates for the third milestone were directly made to the main branch, without bypassing a typical pull request procedure. To review the work specific to the third milestone, please revisit repository's history following the commit identified by SHA: '7306ec8', which marks the the last commit of the second milestone's contributions.

References

M. Wu. multimodal-vae-public. https://https://github.com/mhw32/multimodal-vae-public, 2018.

M. Wu and N. Goodman. Multimodal generative models for scalable weakly-supervised learning, 2018.