Variational Autoencoders

A Deep Generative Model

Ayush Raina

Indian Institute of Science

October 6, 2024

What are Autoencoders ?

1. Autoencoders are neural networks that aim to learn the compact representation of the input data whose dimension is much smaller than the input data.
2. It consists of two parts: an *encoder*(ϕ) and a *decoder*(θ), where encoder and decoder are neural networks.

What are Autoencoders ?

1. Autoencoders are neural networks that aim to learn the compact representation of the input data whose dimension is much smaller than the input data.
2. It consists of two parts: an *encoder*(ϕ) and a *decoder*(θ), where encoder and decoder are neural networks.

Goal

The goal of encoder network is to learn a hidden representation of the input data, and the goal of decoder network is to reconstruct the input data from the hidden representation.

Training Autoencoders

- The training of autoencoders is done by minimizing the reconstruction error between the input data and the reconstructed data.
- The loss function used for training autoencoders is Mean Squared Error (MSE).

Training Autoencoders

- The training of autoencoders is done by minimizing the reconstruction error between the input data and the reconstructed data.
- The loss function used for training autoencoders is Mean Squared Error (MSE).
- The loss function is given by:

$$L(\phi, \theta) = \frac{1}{N} \sum_{i=1}^{N} ||x_i - Decoder_\theta(Encoder_\phi(x_i))||^2 \quad (1)$$

where x_i is the input data, $Encoder_\phi(x_i)$ is the hidden representation of x_i and $Decoder_\theta(Encoder_\phi(x_i))$ is the reconstructed data.

Reconstructions of Autoencoders

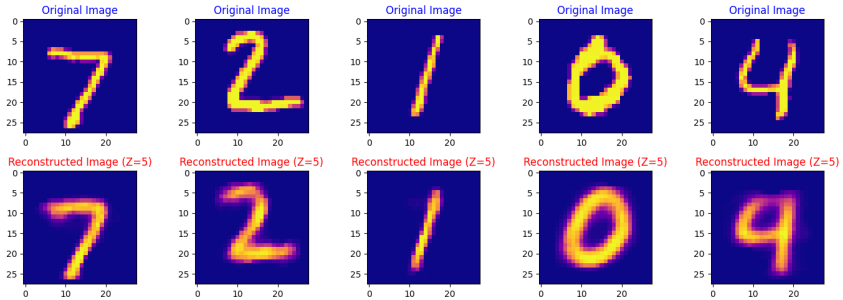

Figure 1: Reconstruction of Autoencoders without CNN

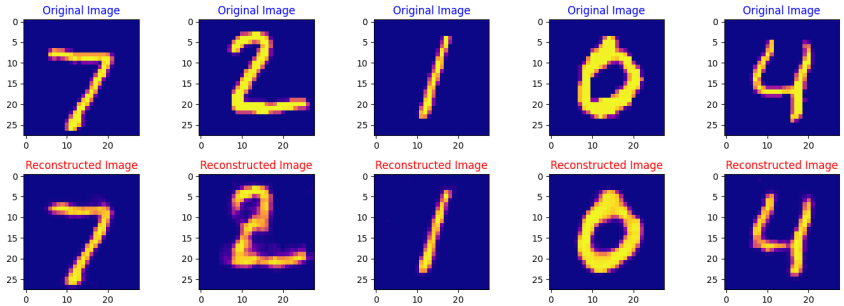

Figure 2: Reconstruction of Autoencoders with CNN

In both the cases we trained for over 20 epochs.

Reconstructions of Autoencoders

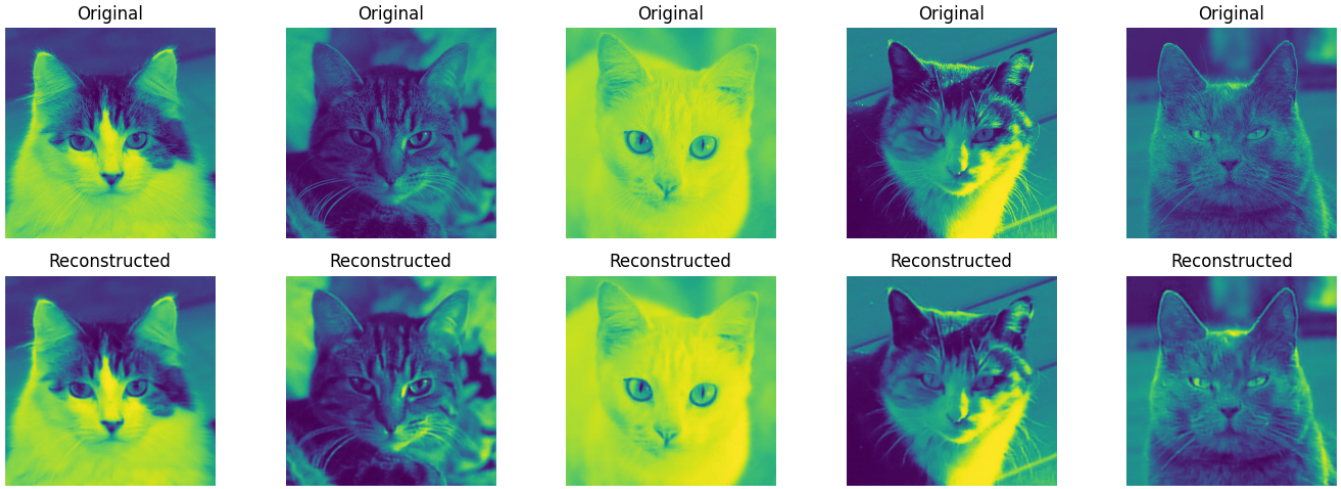

Figure 3: Reconstruction of Autoencoders on Animal Face Dataset

This reconstruction was done on grayscale version, we also obtained similar output on colored version.

Generation

Can we generate new data with Autoencoders ?

After the training of autoencoders, we can generate new data by feeding random hidden representation to the decoder network. But we may simply get noise in the output.

This is because the hidden representation lies in very small subspace of the input space.

Generation

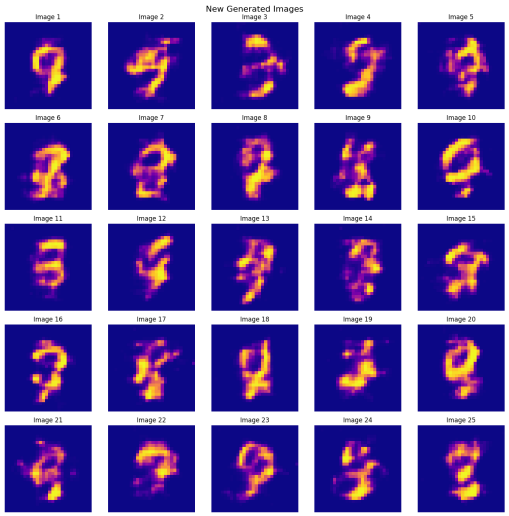

Figure 4: Generation of new handwritten digits using Autoencoders

Generation

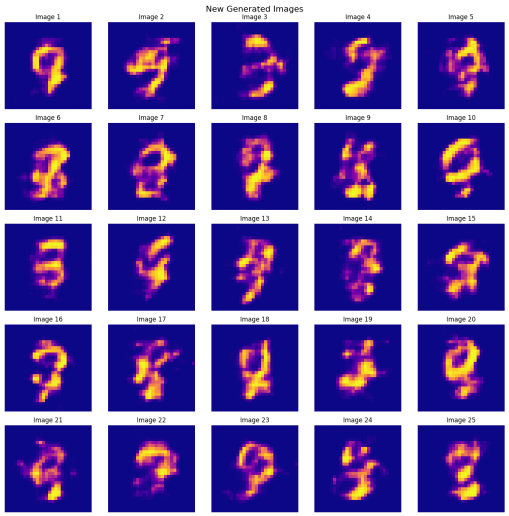

Figure 4: Generation of new handwritten digits using Autoencoders

This is the generation when we fed mean of the hidden representations generated by the encoder network. Otherwise output was simply a noise.

Generation

What should be done ?

If we somehow feed the highly likely hidden representation z, then we can expect meaningful output.

Generation

What is highly likely hidden representation ?

In fact we want to sample from $P(z^{(i)}|x^{(i)})$ where $z^{(i)}$ is the hidden representation of $x^{(i)}$.

Here both encoder and decoder networks are deterministic, which means for every input data $x^{(i)}$, the hidden representation $z^{(i)}$ is fixed and vice versa.

Introduction to VAE

Variational Autoencoders (VAE) have same structure as autoencoders, but here we learn the distribution of the hidden representation, rather than learning the fixed hidden representation.

Introduction to VAE

Variational Autoencoders (VAE) have same structure as autoencoders, but here we learn the distribution of the hidden representation, rather than learning the fixed hidden representation.

2 things to care about

1. We are interested in learning the distribution $P(z^{(i)}|x^{(i)})$ so that we can sample highly likely $z^{(i)}$ for a given $x^{(i)}$.
2. We are also interested in learning the distribution $P(x^{(i)}|z^{(i)})$ so that we can generate new data by sampling $z^{(i)}$ from $P(z^{(i)}|x^{(i)})$.

With above choice, neither encoder nor decoder is deterministic.

Modelling Assumptions

Assumption 1

$z^{(i)} \sim N(0, I_{k \times k})$

Assumption 2

Posterior distribution $P(z^{(i)}|x^{(i)}) \sim N(\mu(x^{(i)}), \Sigma(x^{(i)}))$, where $\mu(x^{(i)})$ and $\Sigma(x^{(i)})$ are the functions of $x^{(i)}$.

Assumption 3

$P(x^{(i)}|z^{(i)}) \sim N(\mu(z^{(i)}), \Sigma(z^{(i)}))$, where $\mu(z^{(i)})$ and $\Sigma(z^{(i)})$ are the functions of $z^{(i)}$.

Achieving the goals

1. Since we assumed that $P(z^{(i)}|x^{(i)}) \sim N(\mu(x^{(i)}), \Sigma(x^{(i)})$, the goal of encoder network is to learn mean and variance of the distribution.

2. We also assumed that $P(x^{(i)}|z^{(i)}) \sim N(\mu(z^{(i)}), \Sigma(z^{(i)})$, the goal of decoder network is to learn mean and variance of the distribution.

Visualizing the VAE

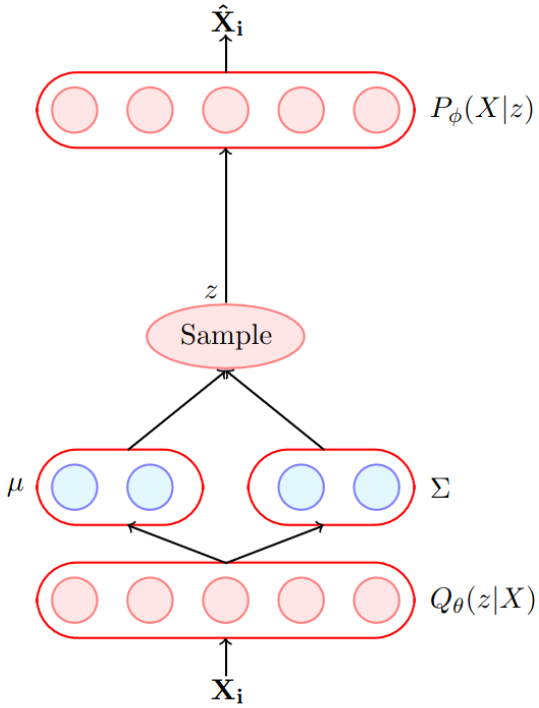

Figure 5: Variational Autoencoder

Loss Function in VAE

We optimize the following MSE Loss + KL Divergence Loss where KL Divergence Loss is given by:

$$KL(P(z^{(i)}|x^{(i)})||P(z^{(i)})) = \frac{1}{2}\sum_{j=1}^{k}(\mu_j^2 + \sigma_j^2 - \log(\sigma_j) - 1) \quad (2)$$

where μ_j and σ_j are the mean and variance of the latent space distribution $P(z^{(i)}|x^{(i)})$.In practice we assume that covariance matrix is diagonal.

Loss Function in VAE

We optimize the following MSE Loss + KL Divergence Loss where KL Divergence Loss is given by:

$$KL(P(z^{(i)}|x^{(i)})||P(z^{(i)})) = \frac{1}{2}\sum_{j=1}^{k}(\mu_j^2 + \sigma_j^2 - \log(\sigma_j) - 1) \quad (2)$$

where μ_j and σ_j are the mean and variance of the latent space distribution $P(z^{(i)}|x^{(i)})$.In practice we assume that covariance matrix is diagonal. This can be derived from the expression of KL Divergence between two Gaussian distributions.

VAE's on MNIST dataset

Figure 6: Generation of new handwritten digits using VAE

We can clearly see that VAE has generated better handwritten digits than autoencoders.

VAE's on MNIST dataset

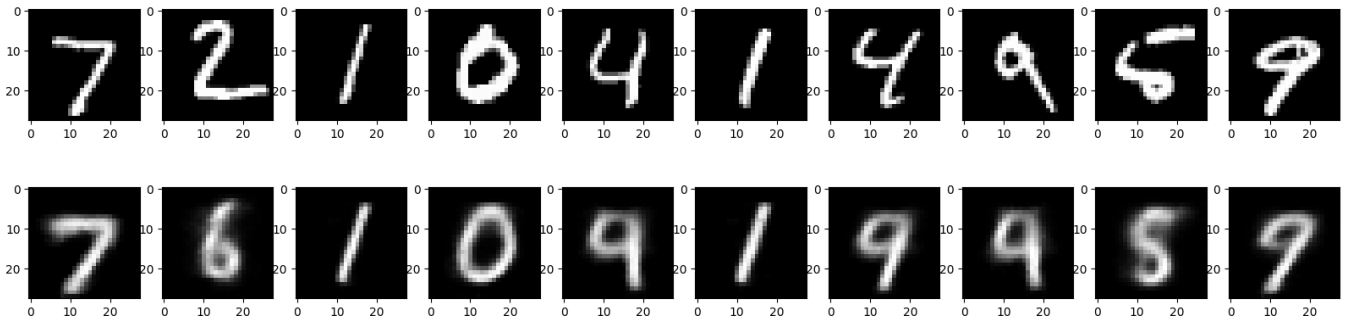

Figure 7: Reconstructions of VAE on MNIST dataset

Reconstructions are also similar to the original data.

Key observations

1. Autoencoders are deterministic, while VAE's are probabilistic.
2. VAE's are better in generating new data than autoencoders.
3. Autoencoders are better in reconstruction than VAE's.

About Dataset

Dataset	Animal Faces
Size	512x512
Training Images	14630
Validation Images	1500
Number of Classes	3

For all out experiments we have resized our images to 128x128.

Training for cats only

We trained VAE for 5153 cat images and used 500 images for validation. We trained over 40 epochs. Here are the results:

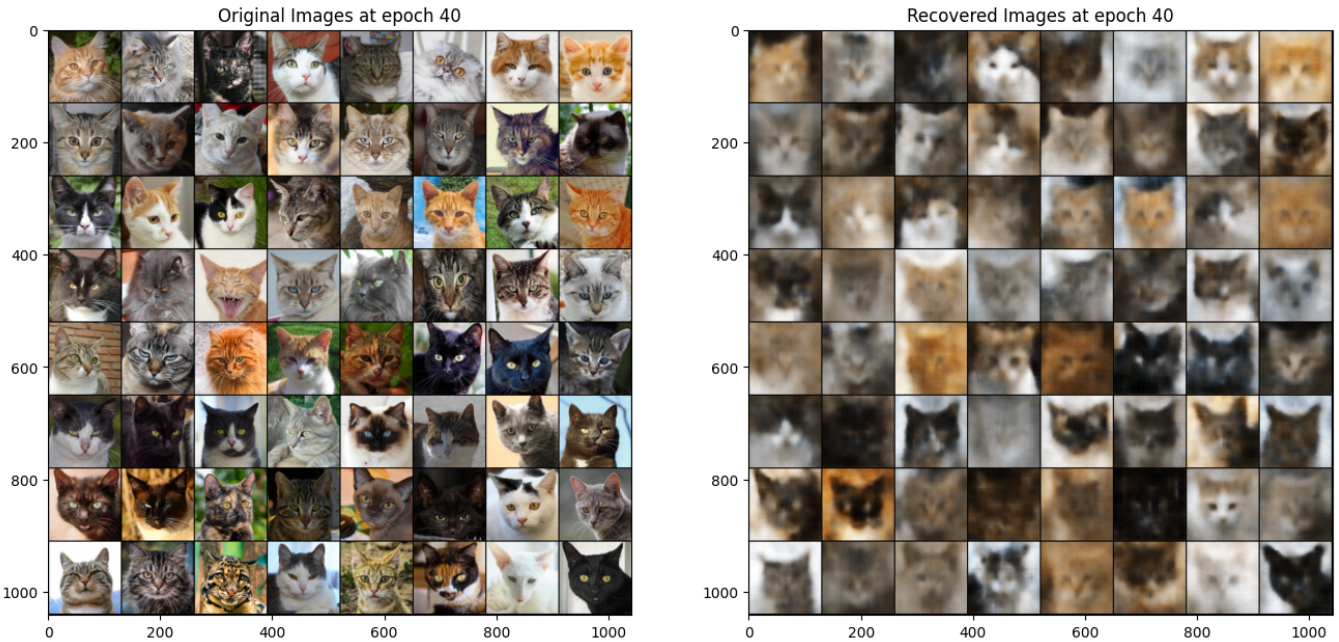

Figure 8: Reconstructions of VAE on Cat dataset

Generation

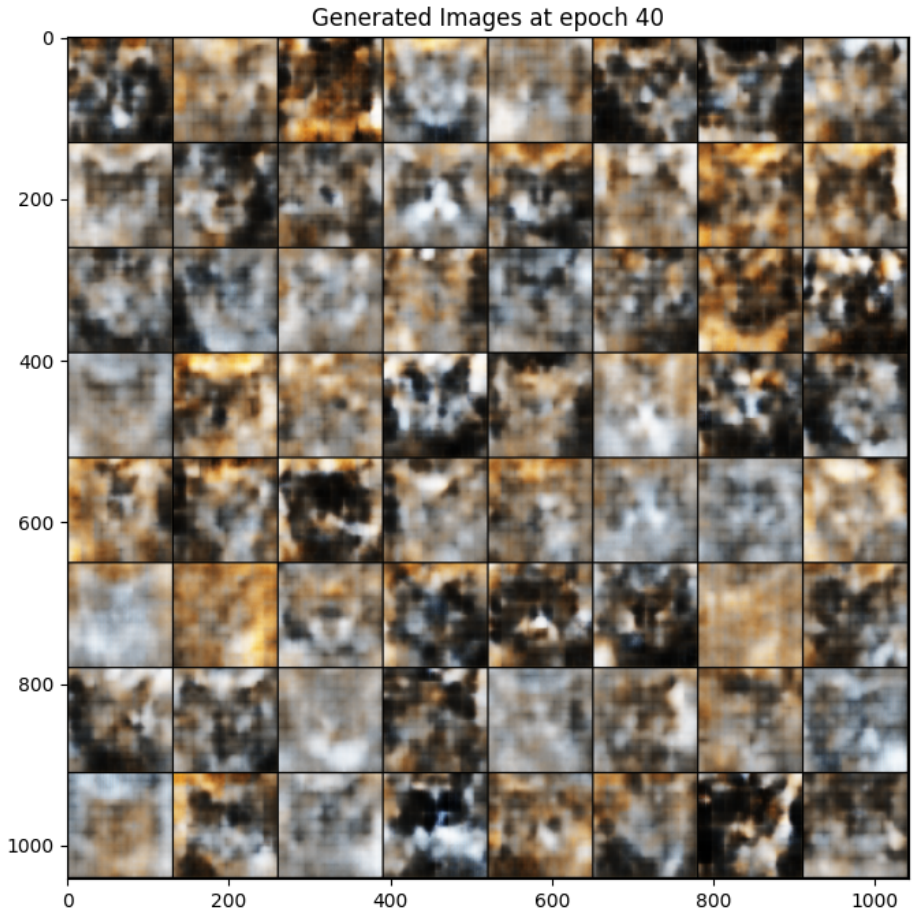

Figure 9: Generation of new cat images using VAE

Loss Curves

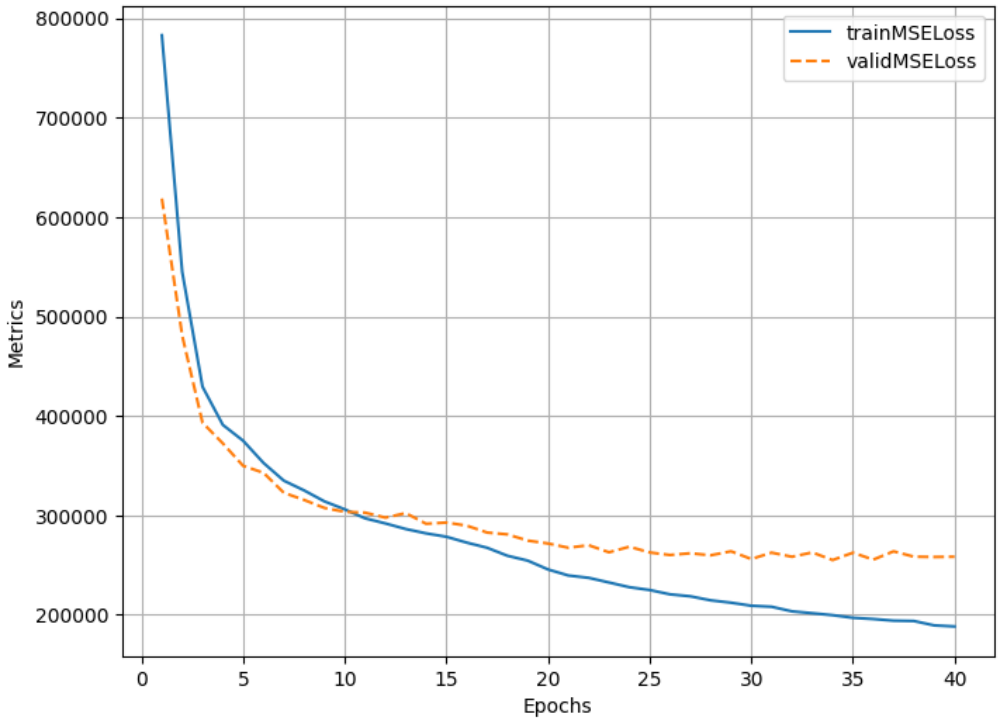

Figure 10: Loss Curves for VAE on Cat dataset

Training for wild animals only

We trained VAE for 4738 images of wild animals and used 500 images for validation. We trained over 70 epochs.

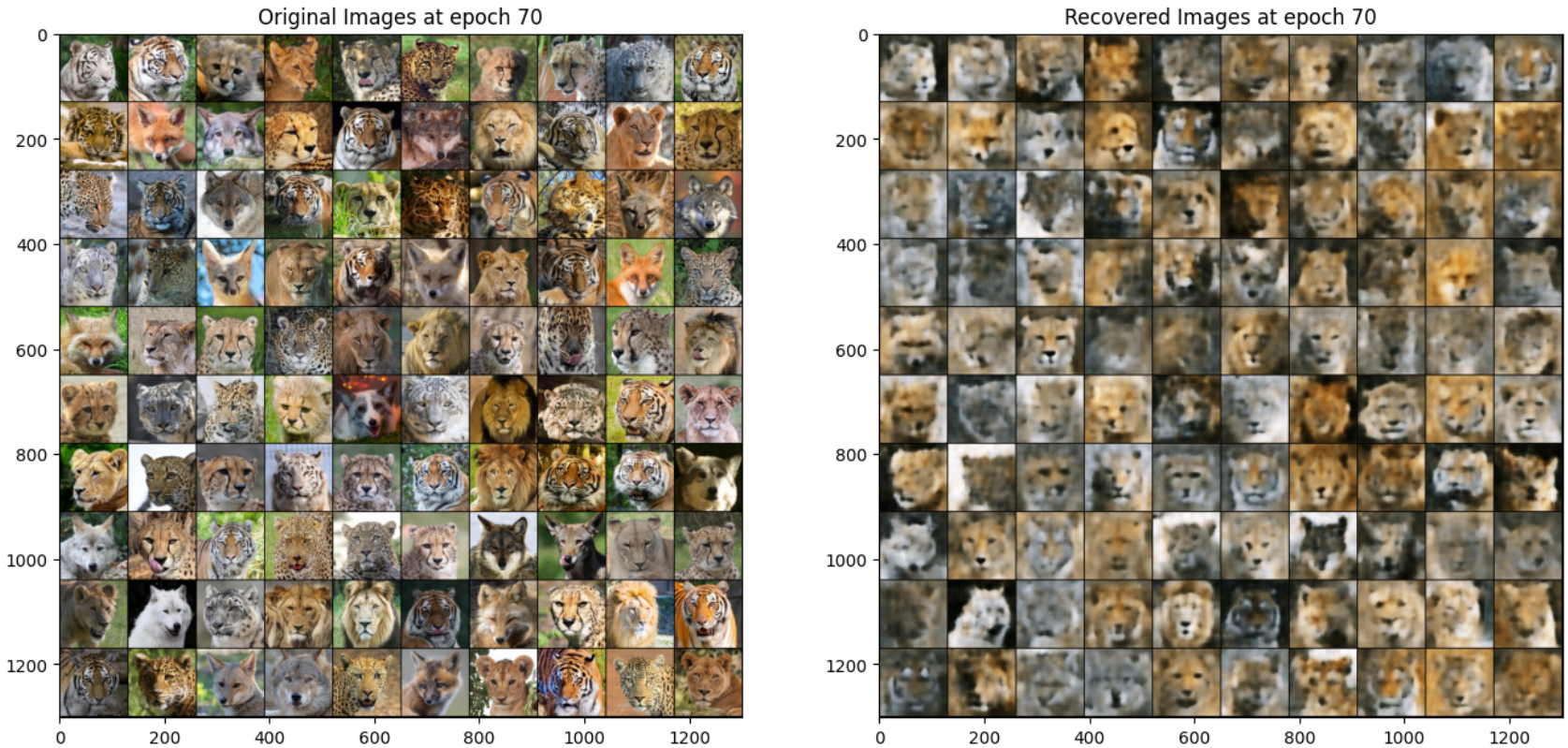

Figure 11: Reconstructions of VAE on Wild Animals

Generation

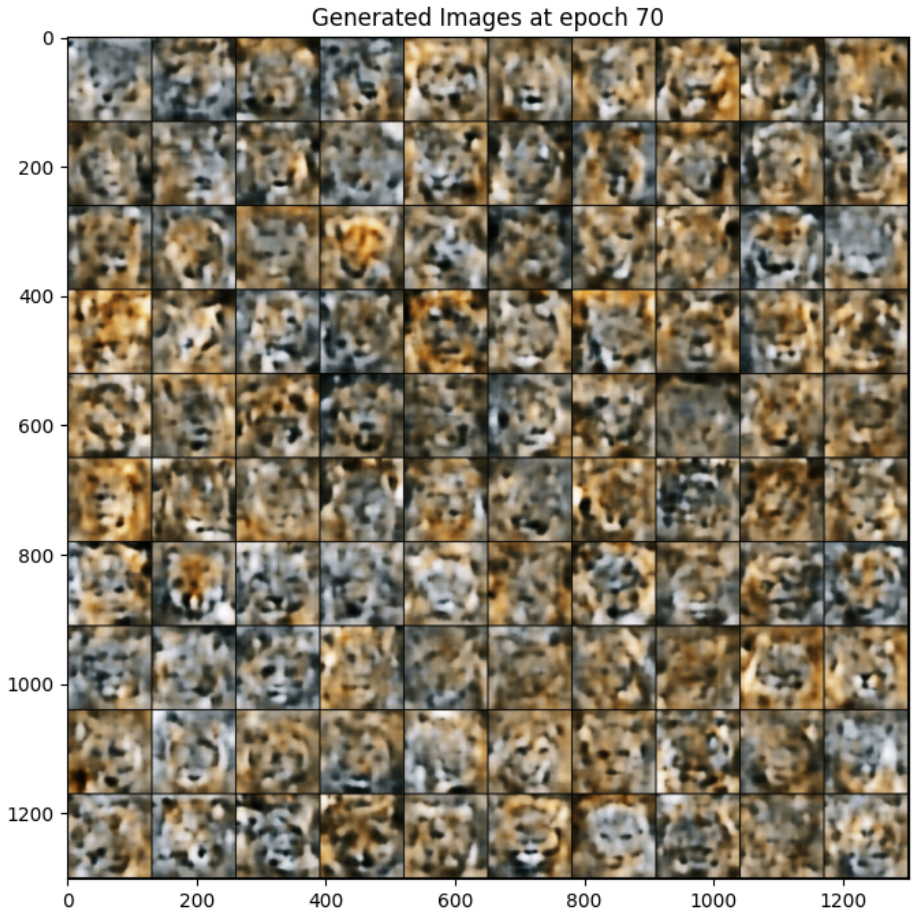

Figure 12: Generation of new wild animal images using VAE

Loss Curves

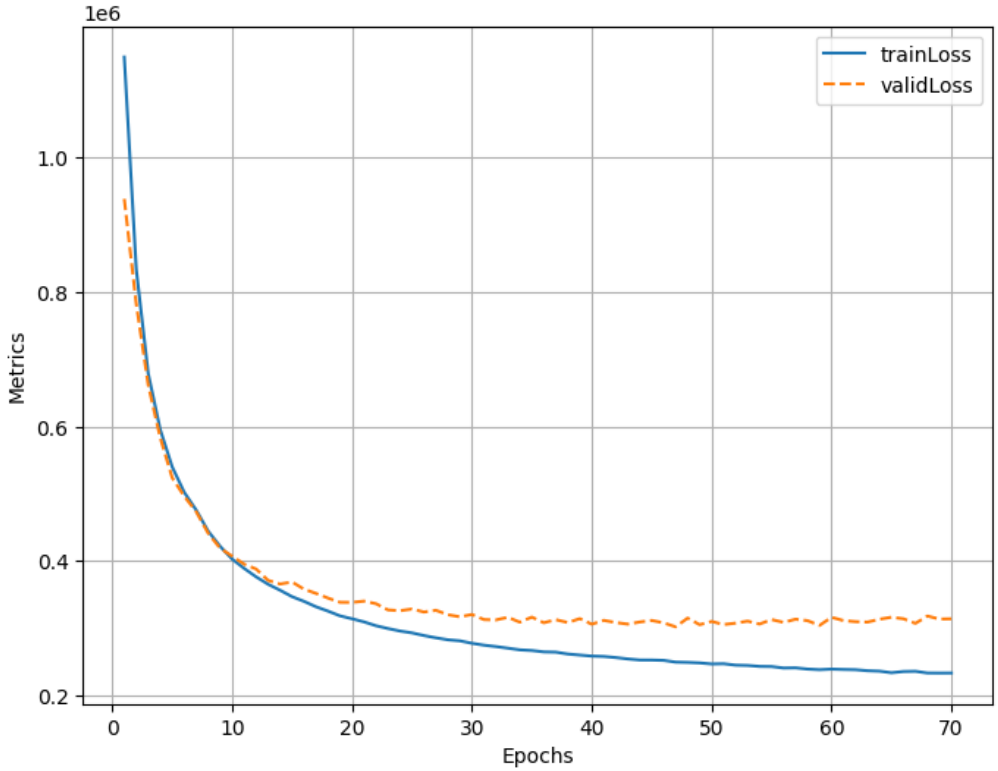

Figure 13: Loss Curves for VAE on Wild Animals

Training on full dataset

We trained VAE for 14630 images of animals and used 1500 images for validation. We trained over 100 epochs.

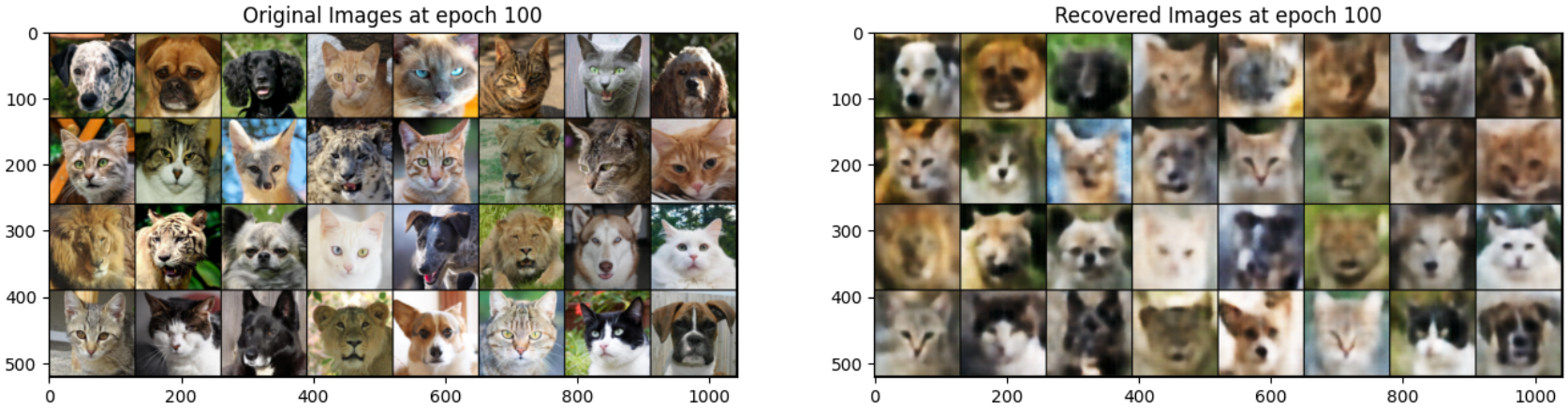

Figure 14: Reconstruction of VAE on Animal Face Dataset

Generation

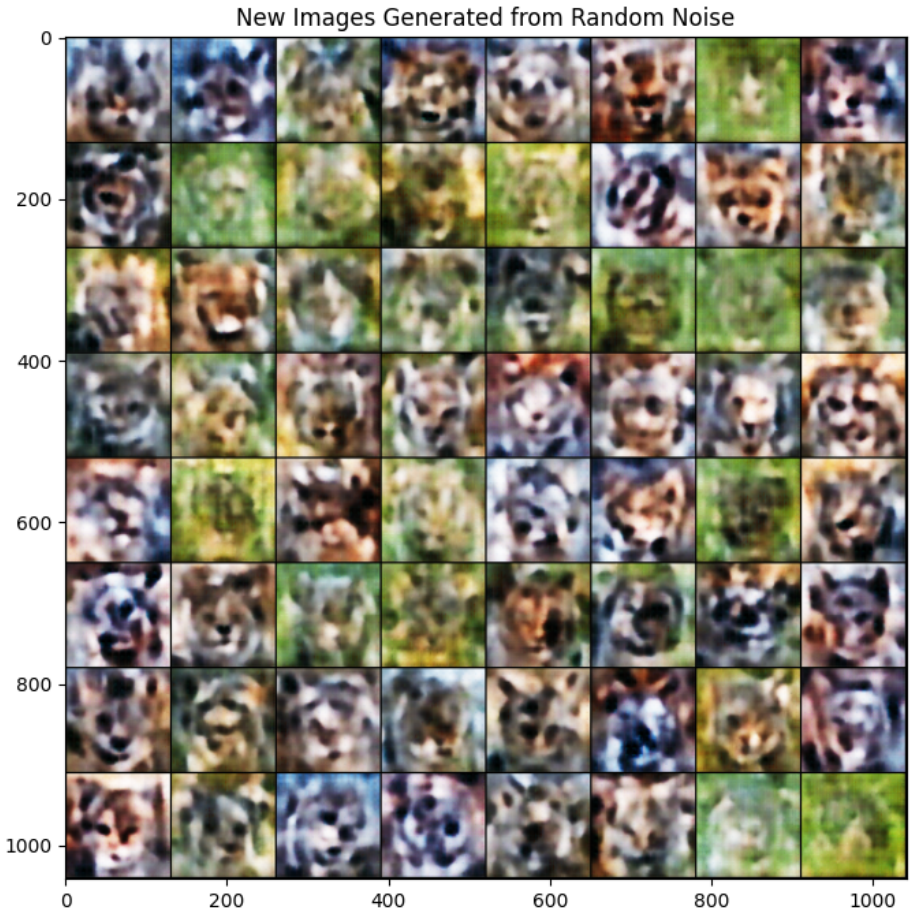

Figure 15: Generation of new animal face images using VAE

Loss Curve 1

Here are plot for Train Loss and Validation Loss for VAE on Animal Face Dataset.

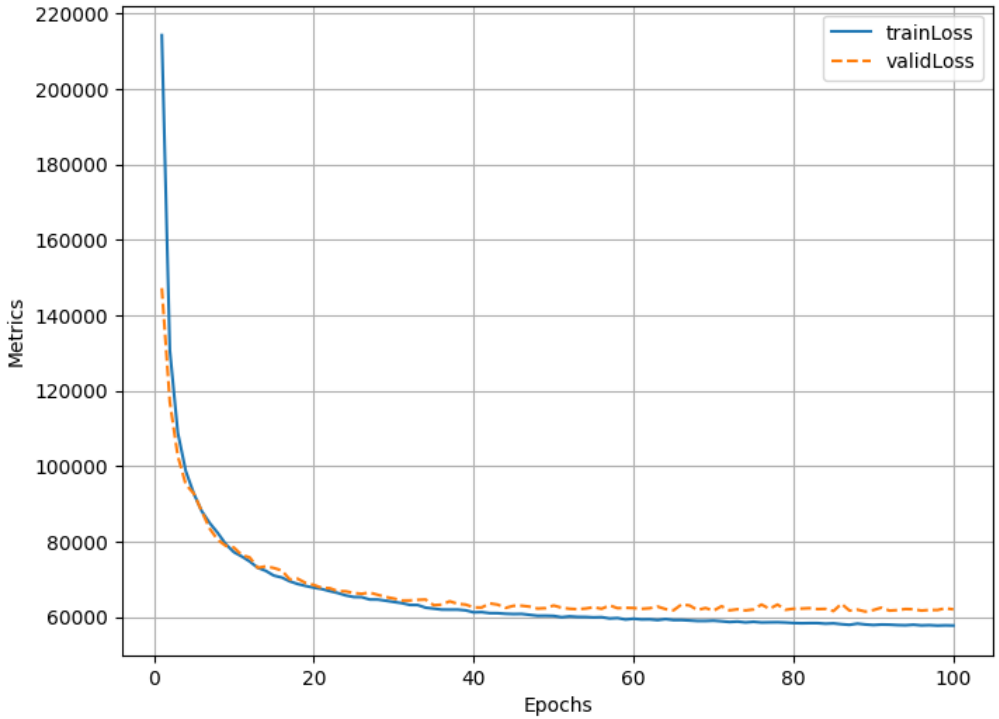

Figure 16: Loss Curves for VAE on Animal Face Dataset

Loss Curve 2

Here are plot for KL Divergence Loss during training and validation for VAE on Animal Face Dataset.

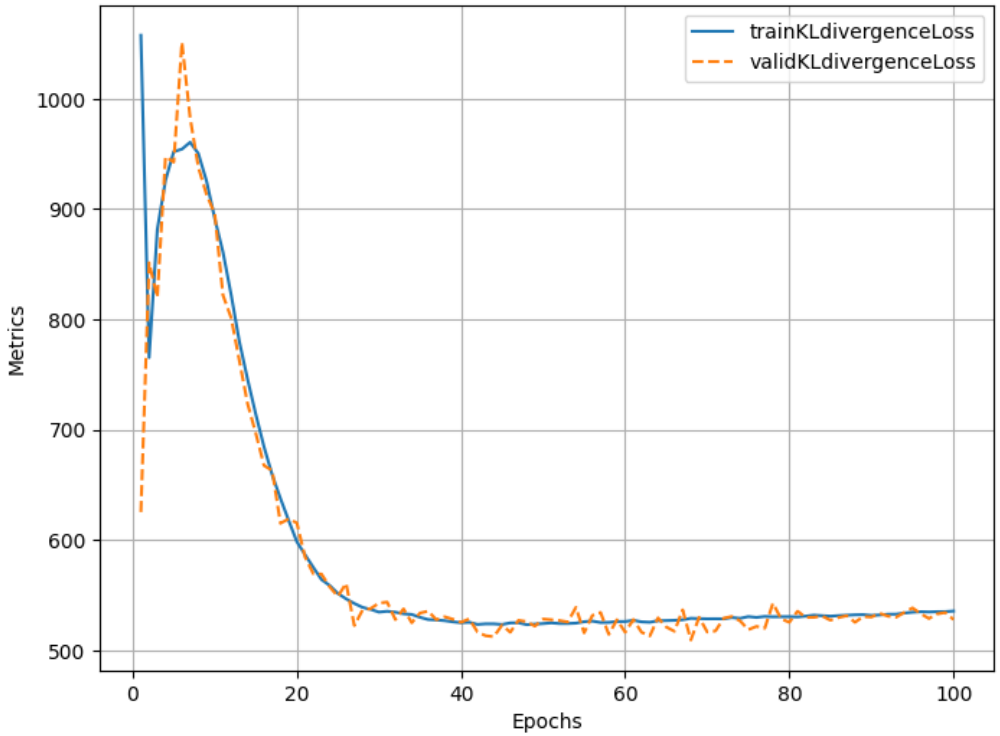

Figure 17: KL Divergence Loss for VAE on Animal Face Dataset

Frechet Inception Distance

We took randomly choosen 2000 images resized to 128x128 from training set and generated 2000 images using VAE. Our calculated FID score is 269.98.

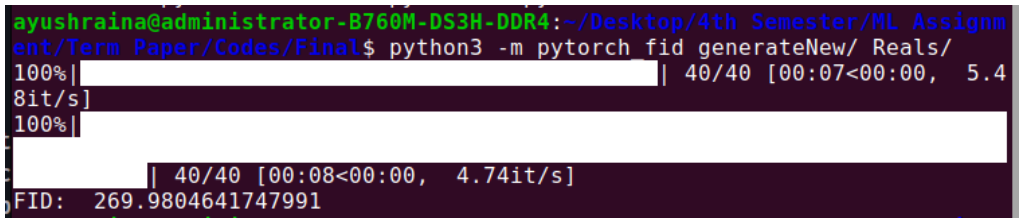

Figure 18: Frechet Inception Distance for VAE on Animal Face Dataset

Key Takeaways

- Importance of Learning Rate.

Key Takeaways

- Importance of Learning Rate.
- Importance of Batchsize when training with GPU's.

Key Takeaways

- Importance of Learning Rate.
- Importance of Batchsize when training with GPU's.
- More is KL weight, more better the generation, but reconstruction may suffer.

Key Takeaways

- Importance of Learning Rate.
- Importance of Batchsize when training with GPU's.
- More is KL weight, more better the generation, but reconstruction may suffer.
- Number of epochs is also important to prevent overfitting sometimes.

Key Takeaways

- Importance of Learning Rate.
- Importance of Batchsize when training with GPU's.
- More is KL weight, more better the generation, but reconstruction may suffer.
- Number of epochs is also important to prevent overfitting sometimes.
- Playing with architecture can also give better results.

Some more results

These are the results that we obtained during experiments but we lost the parameters onwhich these were generated.

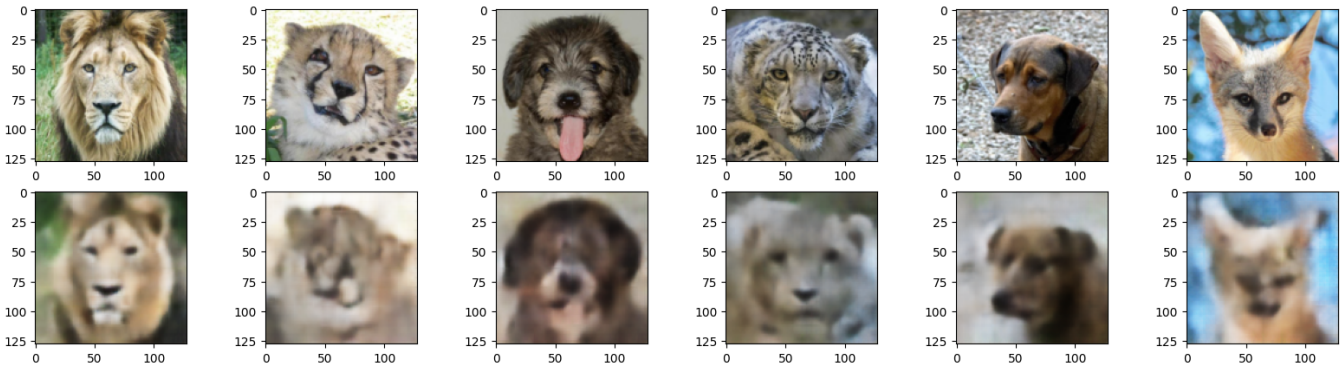

Figure 19: Reconstruction

Some more results

Figure 20: Generation

Variational Inference

In VAE, our first goal is to learn the distribution $P(z^{(i)}|x^{(i)})$ we cannot calculate this distribution directly because terms involved are neural networks. So, we use Variational Inference to approximate this distribution.

Variational Inference

Log likelihood

$$\log P(x^{(i)}) = ELBO(x; Q) + D_{KL}(Q||P_{z|x}) \tag{3}$$

We want Q to be as close as possible to $P_{z|x}$, which means $D_{KL}(Q||P_{z|x}) \longrightarrow 0$.If we maximise ELBO wrt Q, then we are minimising $D_{KL}(Q||P_{z|x})$.
So our problem reduces to $\max_{q \sim Q} ELBO(x; q)$. We will assume all q belong to same family Q which is Gaussian.

Thank You!

Thank You!

Here are some references:

1. ELBO Surgery: Yet Another Way to Carve Up the Variational Evidence Lower Bound, https://approximateinference.org/2016/accepted/HoffmanJohnson2016.pdf
2. CS229 Lecture Notes for VAE, https://cs229.stanford.edu/summer2019/cs229-notes8.pdf
3. Tutorial on Variational Autoencoders, https://arxiv.org/pdf/1606.05908.pdf