Deep Learning on Computational Accelerators Mini-Project Report

Sean Metlitski and Guy Or

September 26, 2024

1 Training an Auto-Decoder on Fashion MNIST

In this section of the mini-project, the goal was to train an Auto Decoder model on a subset of the Fashion MNIST dataset. The dataset contains images of various clothing items such as shirts, shoes, and bags, with the purpose being to reconstruct these images using latent vectors. Unlike Variational Auto Decoders (VADs), we focused solely on training an Auto Decoder (AD), where each image has its own latent vector representation instead of learning a distribution. The aim was to minimize the reconstruction loss by learning these latent vectors alongside the model's parameters. Below is an explanation of the architecture and training choices made during the project.

1.1 Model Architecture

For the Auto Decoder, a fully connected neural network (FCNN) was used to map random latent vectors back to the image space. The key details of the architecture are as follows:

- **Latent Space:** A 64-dimensional latent vector was initialized for each sample in the dataset. This latent space was chosen to balance between compression and feature retention.
- **Decoder Network:** The decoder was designed to map the 64-dimensional latent vectors back into 784-dimensional vectors, corresponding to the 28x28 pixel images from the Fashion MNIST dataset.
  - Layer 1: Fully connected layer transforming the latent vector (64 units) to 128 units, followed by a ReLU activation.
  - Layer 2: Fully connected layer from 128 to 256 units, followed by another ReLU activation to increase the network's capacity to capture features.
  - Layer 3: Fully connected layer from 256 units to 512 units, followed by a ReLU activation.

– Output Layer: Fully connected layer mapping from 512 units to 784 units (28x28 pixels). A Sigmoid activation was used to ensure the pixel values are between 0 and 1.

1.2 Training Parameters

The training was performed using the following choices:

- **Loss Function:** Mean Squared Error (MSE) was selected as the reconstruction loss function, which computes the pixel-wise difference between the input image and the decoded output.
- **Optimizer:** The Adam optimizer was used, with a learning rate of 0.001. Adam was chosen due to its adaptive learning rate and fast convergence.
- **Batch Size:** A batch size of 32 was used to ensure stable gradient updates and avoid memory issues.
- **Epochs:** The model was trained for 400 epochs, and latent vectors were optimized for 200 epochs for both training and test sets.

2 Model Evaluation

After training the Auto Decoder, the model was evaluated on both the training and test datasets. The key results are as follows:

2.1 Training and Test Results

- **Training Loss:** The final training loss was approximately **0.000265**, showing excellent convergence on the training set.
- **Test Loss:** The final test loss was approximately **0.00135**, indicating that the model generalized well to unseen data.

3 Latent Vector Sampling and Decoding

In this section, we evaluate the model by comparing the reconstruction of two sets of latent vectors:

- **Test Set Latent Vectors:** Latent vectors that were optimized for specific test set samples.
- **Random Latent Vectors:** Latent vectors sampled randomly from a uniform distribution $U(0, I)$.

3.1 Decoded Images

The figure below shows a comparison between images decoded from test set latent vectors and those decoded from randomly sampled latent vectors.

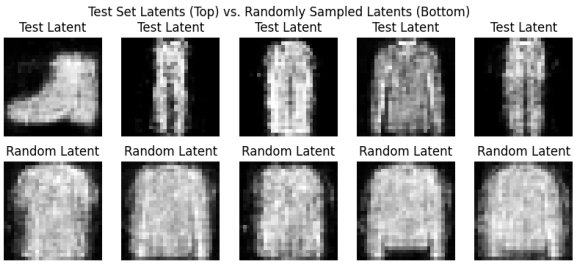

Figure 1: Comparison of decoded images from test set latent vectors (top) and randomly sampled latent vectors (bottom).

3.2 Why Test Set Latents Performed Better

The latent vectors from the test set produced significantly better reconstructions because they were specifically optimized during evaluation for each sample. This allowed the model to adjust the latent vectors to represent the important features of the clothing items, leading to more diverse outputs that resemble the original images.

In contrast, the randomly sampled latent vectors do not capture any meaningful structure from the dataset, resulting in blurry or distorted images. Additionally, the randomly sampled latents often look similar, such as resembling shirts in this case, because the model's latent space may be biased toward more common structures in the dataset (e.g., shirts), which dominate the random regions of the latent space.

4 Overview of Variational Auto Decoder Implementation

4.1 Model Architecture

The Variational Auto Decoder (VAD) uses a 64-dimensional latent space to balance compression and feature retention. The model has fully connected layers to output the mean and log variance for each input, enabling the use of the reparameterization trick for latent space sampling. The decoder has 3 layers: 128, 256, and 512 units, with ReLU activations, and an output layer with 784 units (28x28 pixels) and Sigmoid activation to ensure pixel values are between 0 and 1.

4.2 Training Parameters

The VAD is trained by optimizing two losses: Mean Squared Error (MSE) for reconstruction and KL Divergence to regularize the latent space. The model is optimized using the Adam optimizer with a learning rate of 0.001. Latent vectors are initialized as random Gaussian vectors and trained alongside model parameters. The model was trained with a batch size of 32 over 50 epochs.

4.3 Justification for Choices

The 64-dimensional latent space balances representational power with efficiency. The fully connected decoder layers ensure effective mapping from latent to image space. The Adam optimizer, with its adaptive learning rate, was chosen for stability, and KL Divergence regularizes the latent space to a Gaussian distribution, enabling effective sampling for generating new data.

5 Visualizing the Latent Space using t-SNE

In this section, we visualize the latent space of the model trained in section 1.3.1 using t-SNE (t-Distributed Stochastic Neighbor Embedding), which is a dimensionality reduction technique that helps visualize high-dimensional data in 2D space.

5.1 Latent Space Visualization

The latent vectors produced by the AutoDecoder for the Fashion MNIST dataset were projected onto a 2D plane using t-SNE. The visualization provides insight into how the model organized the latent space for the dataset. Each point in the plot represents a latent vector corresponding to an image, and the points are colored according to their label (i.e., their clothing item category).

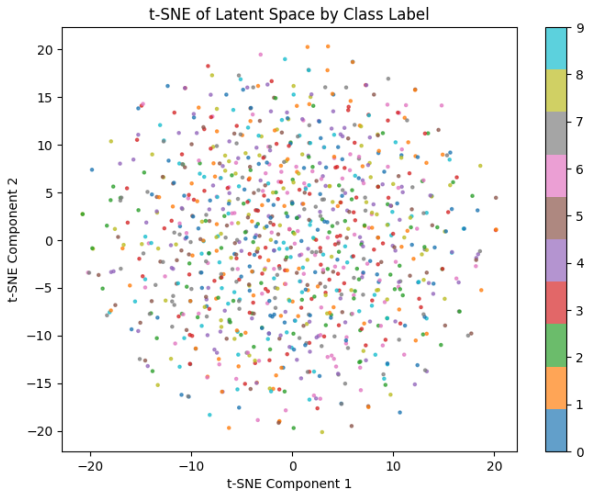

Figure 2: t-SNE visualization of the latent space with color labels representing different clothing items.

5.2 Analysis of the Latent Space

In the current model, we only minimized the reconstruction loss. This has caused the latent space to become disorganized, as shown in the t-SNE visualization. The technique projects the high-dimensional latent space into 2D, allowing us to see how similar samples are in this space. However, without a regularization term, the latent vectors are scattered and fail to form distinct clusters for different labels.
To resolve this, we can introduce a loss term that normalizes the latent variables to follow a known distribution. This would make it easier to generate new data consistently and organize similar samples closer together in the latent space. Additionally, with a well-structured latent space, interpolating between latent vectors would produce smoother transitions between the samples.

6 Training a Variational Auto-Decoder on Fashion MNIST

In this section, we applied the Variational Auto-Decoder (VAD) model to the Fashion MNIST dataset. The key difference between the VAD and AD models is the introduction of a latent distribution learned during training, which allows the VAD to generate better reconstructions from random latent vectors by sampling from the learned distribution.

6.1 Model Evaluation

After training the VAD model, we evaluated it on both the training and test datasets. The results are as follows:

- **Training Loss:** The final training loss for the VAD was **0.02179**, indicating successful convergence.
- **Test Loss:** The final test loss for the VAD was **0.03529**, showing that the model generalized well to unseen data.

6.2 Latent Vector Sampling and Decoding in the VAD

We decoded two sets of latent vectors:

- **Test Set Latent Vectors:** Latent vectors optimized for specific test set samples.
- **Random Latent Vectors:** Latent vectors sampled randomly from a Gaussian distribution $N(0, I)$, learned during training.

6.3 Decoded Images from VAD

The figure below compares the decoded images from test set latent vectors and randomly sampled latent vectors:

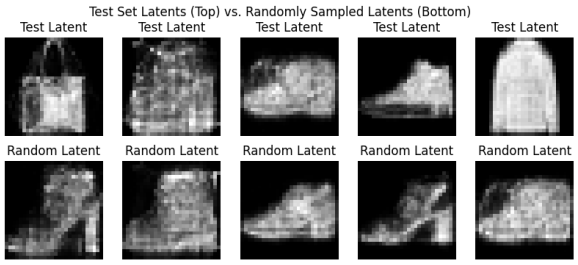

Figure 3: Comparison of decoded images from test set latent vectors (top) and randomly sampled latent vectors (bottom) using the VAD.

6.4 t-SNE Visualization of VAD Latent Space

We used t-SNE to visualize the latent space learned by the VAD. Each point in the plot represents a latent vector, with colors corresponding to the vector norms (magnitude).

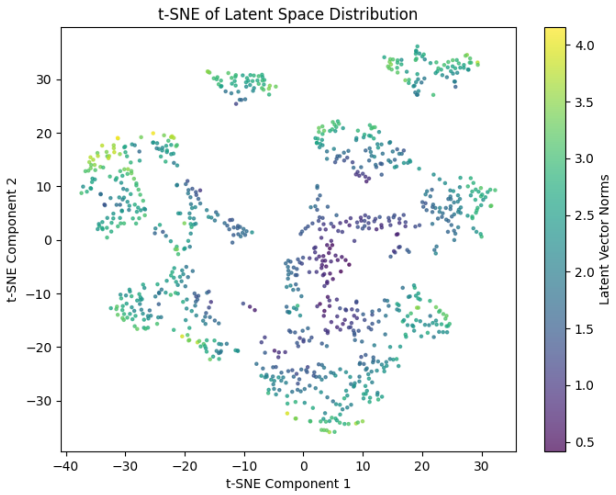

Figure 4: t-SNE of the Latent Space learned by the VAD. Colors represent latent vector norms.

The plot shows distinct clusters in the latent space. Latent vectors with smaller norms (in purple) are near the center, while larger ones (in yellow and green) are spread outward, reflecting how the VAD organizes the data for smooth interpolations and sampling.

6.5 Why Test Set Latents Performed Better in the VAD

In the VAD model, the latent vectors from the test set produced clearer and more accurate reconstructions because they were optimized for each specific sample. This allowed the model to effectively capture the underlying features of each image in the latent space, resulting in high-quality reconstructions.
In contrast, the randomly sampled latent vectors, while still capturing the overall structure of the dataset, produced blurrier images. This occurs because random latents are drawn from the latent distribution and are not fine-tuned to any specific image. However, the performance of the random latents in the VAD is significantly better than in the AD because the VAD's learned distribution ensures that the latent space is organized and can generalize to new data better than the unstructured AD latent space.

7 Distributions and Reparameterization Trick

In this project, we trained the Variational AutoDecoder (VAD) using two different distributions for the latent space: the **gaussian** distribution and the **uniform** distribution.

7.1 Gaussian Distribution

The normal distribution assumes that the latent variables follow a Gaussian distribution with a mean (μ) and standard deviation (σ). We used the reparameterization trick to allow gradients to pass through the sampling process:

$$z = \mu + \sigma \cdot \epsilon$$

where ϵ is sampled from a standard normal distribution $\epsilon \sim \mathcal{N}(0, 1)$.

7.2 Uniform Distribution

The uniform distribution assumes that the latent variables are sampled from a uniform range. We used the following reparameterization trick:

$$z = \mu + \epsilon \cdot \sigma$$

where ϵ is sampled from a uniform distribution $\epsilon \sim U(0, 1)$, and scaled by the standard deviation σ to match the spread of the distribution.

Both models were trained using the same dataset, and their performances were compared in terms of reconstruction quality and latent space exploration.

8 Gaussian and Uniform Variational AutoDecoder Results

In this section, we present the results for the two Variational AutoDecoder (VAD) models trained on Gaussian and Uniform distributions. We discuss the model parameters, evaluate the performance on training and test sets, show decoded sample images, and visualize the latent space using t-SNE.

8.1 Gaussian Distribution Variational AutoDecoder

8.1.1 Model Parameters

The Gaussian VAD model uses a latent space that follows a Gaussian distribution with a mean (μ) and standard deviation (σ). The reparameterization trick used for this model is:

$$z = \mu + \sigma \cdot \epsilon \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, 1)$$

The model was trained with a batch size of 32, learning rate of 0.001, and a latent dimension of 64 for 400 epochs.

8.1.2 Training and Test Losses

After training the Gaussian VAD model, the final training and test losses were as follows:

- **Training Loss:** *0.0228*
- **Test Loss:** *0.03641*

8.1.3 Sampled and Test Latent Vector Decoding

We sampled 5 latent vectors from the Gaussian distribution $z \sim \mathcal{N}(0, 1)$ and decoded them, along with 5 latent vectors from the test set. The comparison between the test set latents and the sampled latents is shown below.

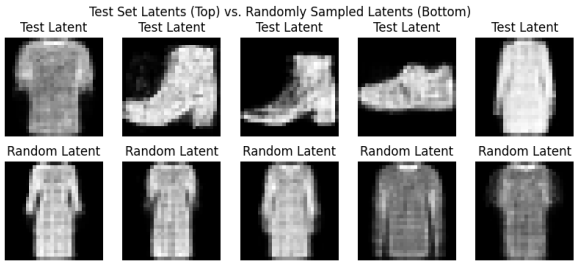

Figure 5: Gaussian VAD: Test Set Latents (Top) vs. Randomly Sampled Latents (Bottom).

The test set latents produced better reconstructions compared to the sampled latents, as the latent vectors were specifically optimized during training for each test image.

8.1.4 t-SNE Visualization of the Latent Space

The t-SNE plot below visualizes the latent space of the Gaussian VAD model. The latent vectors were projected into 2D space, and their magnitudes (norms) were used to color the points in the plot.

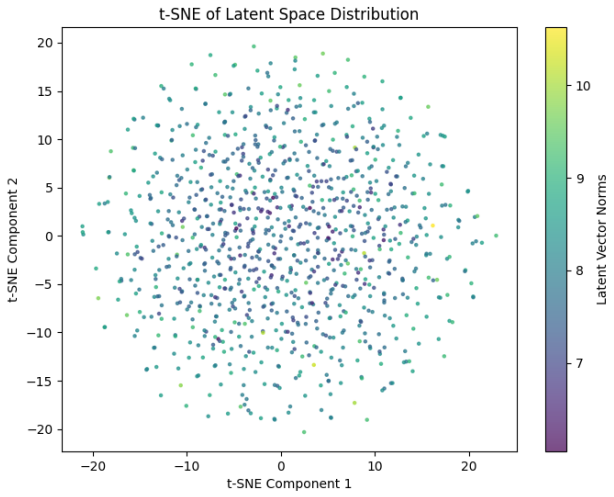

Figure 6: t-SNE of the Latent Space using Gaussian Distribution.

8.2 Uniform Distribution Variational AutoDecoder

8.2.1 Model Parameters

The Uniform VAD model assumes that latent variables follow a uniform distribution. The reparameterization trick used for this model is:

$$z = \mu + \epsilon \cdot \sigma \quad \text{where} \quad \epsilon \sim U(0, 1)$$

The model was also trained with a batch size of 32, learning rate of 0.001, and a latent dimension of 64 for 400 epochs.

8.2.2 Training and Test Losses

After training the Uniform VAD model, the final training and test losses were as follows:

- **Training Loss:** *0.0219*
- **Test Loss:** *0.03549*

8.2.3 Sampled and Test Latent Vector Decoding

Similar to the Gaussian VAD, we sampled 5 latent vectors from a uniform distribution and decoded them, along with 5 latent vectors from the test set. The decoded images are shown below.

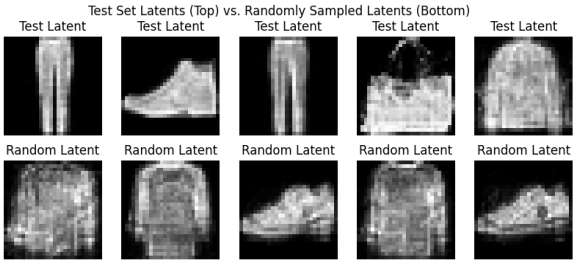

Figure 7: Uniform VAD: Test Set Latents (Top) vs. Randomly Sampled Latents (Bottom).

The test set latents produced better reconstructions, as expected, because they were optimized for the specific test set images.

8.2.4 t-SNE Visualization of the Latent Space

Below is the t-SNE plot of the latent space for the Uniform VAD model. As with the Gaussian model, the latent vectors were projected into 2D space using t-SNE, and the point colors represent their norms.

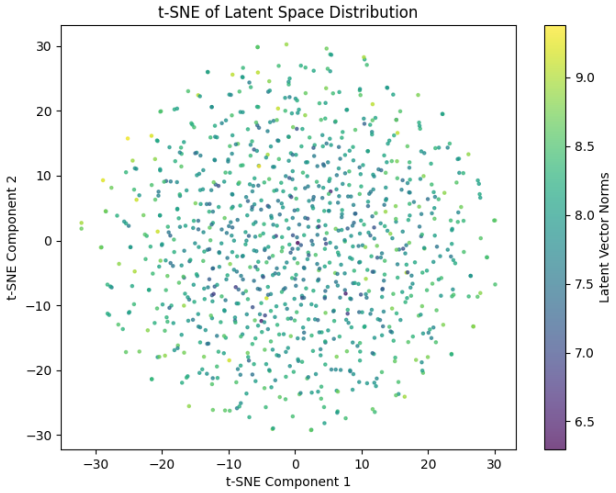

Figure 8: t-SNE of the Latent Space using Uniform Distribution.

8.3 Discussion

The Gaussian VAD model produced a more structured latent space in comparison to the Uniform VAD model. The regularization effect of the KL divergence

in the Gaussian model helps to organize the latent space more effectively. This is also reflected in the t-SNE visualizations, where the Gaussian VAD shows a clearer separation of latent vectors, while the Uniform VAD latent space appears less structured.

9 Latent Space Interpolation

We selected two samples from the test set, each belonging to a different class. For each model, we generated a latent vector for both samples. Then we used spherical linear interpolation to calculate 5 evenly spaced interpolations between the two latent vectors. Slerp ensures smoother transitions by accounting for the geometry of the latent space, resulting in more gradual transitions between the samples. Both the original latent vectors and their slerp-based interpolations were decoded back into images, and the results were plotted in a single line. Figures 9 and 10 show the smooth transition between the two different classes for the Gaussian and Uniform models, respectively.

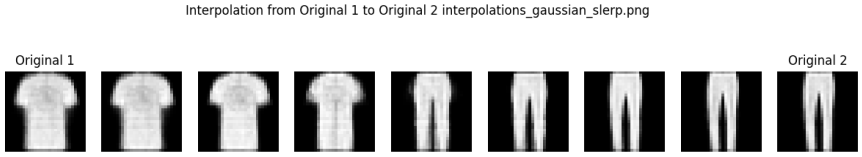

Figure 9: Latent space interpolation for the Gaussian model.

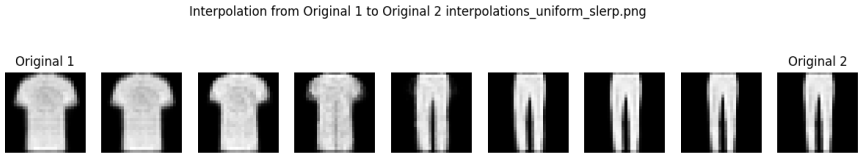

Figure 10: Latent space interpolation for the Uniform model.