Owen MacKenzie & Leslie Dees, Group 1 Y3

Repairing Images with Fisher Autoencoder

1 Introduction

Autoencoders are composed of two stages. The first stage is the encoding stage, in which an input image is encoded into a compressed representation of the image called the latent vector. As this latent vector is restricted to a smaller dimensional space, only the most relevant parts of the input are preserved. The next stage is the decoding stage, where the compressed representation is converted back into a reconstructed image. Autoencoders are successful when the latent representation is unique to each encoding, as such differentiation allows for the decoder to preserve these differences when reconstructing the model. Even standard autoencoders are able to reduce noise present in an image. Since the noisy image is encoded down to a lower dimensional representation, this process can be tailored such that the features preserved are a product of distinct characteristics of the image alone and not the noise. Thus, after decoding these noise indifferent encodings, the resulting reconstructed image should be less noisy than the original input. However, there are many flaws with this approach. A standard autoencoder described as such would be highly specific to the data used to train it. All it knows is the weights that it learned through backpropagation in order to make the reconstructed image as similar to the original input as possible. Although this is enough to cluster similar images together in latent space **[1]**, this system is not very robust because the latent space is not completely covered by all possible images. This means that if one of the images is distorted, such as by blurring or through the addition of noise, the corresponding latent space encoding may drift beyond the reaches of one of these image class clusters.

The issue with this is that this area is unexplored and not correlated with any of the reconstructed classes, so the corresponding image is likely to be a scrambled mess. Furthermore, if the goal is to develop new images by sampling the space, this is more risky in a laten space with many empty holes. In such an autoencoder, the latent space is highly dependent on the encoder and input, and is not given the freedom to compose an understanding of the space or manipulate it in order to lead to meaningful sampling. In regular autoencoders the latent vector is not a distribution so it can't be sampled, and an attempt "sample" by choosing a nearby vector would be foolish because the latent space has many holes and areas that do not correlate to any realistic reconstruction that can be generated. Generation is especially important if there is noise or black box corruption that requires the model to fill in the missing pieces using knowledge of the latent space. This sets the stage for the need of a model that can better cover the latent space in order to capture more deviations from the class clusters and generate realistic outputs through sampling, thus resulting in a more robust model.

Through this project we hope to demonstrate the ability of several variants of variational autoencoders, specifically standard VAEs, Beta-VAEs, and Fisher Autoencoders, to achieve this generation through training on damaged KMNIST images. By training these autoencoders on damaged images and validating their ability to reconstruct images through the minimization of the loss between reconstructed images created from noisy inputs and the ground truth images our goal is to generate the most realistic replications as possible.

2 Related works

2.1 Variational Autoencoders

Variational encoders are able to do this by treating the latent vector as a distribution rather than a simple vector. Although the type of distribution can be altered, the most common and our choice for this project is a Gaussian distribution. A gaussian distribution is defined by two parameters: the mean where the distribution is centered, and a covariance matrix that defines the shape of the distribution by specifying how quickly the probability that a point exists in the distribution trails off in each dimensional direction away from the mean. For simplicity, this can be assumed to be an hyper-ellipsoid, meaning that a covariance parameter controls the width of the distribution in each dimension, but there is no tilting. This is a result of a covariance matrix with no cross covariance, so the matrix is zero apart from the diagonal which includes the covariance of each respective dimension. Thus the gaussian distribution requires two parameters: a vector mu containing the mean of the distribution in each dimension, and a variance vector that composes the diagonal of the covariance matrix. Instead of encoding a latent vector, variational autoencoders encode these two parameters. Then by a reparametrization trick, which is just creating a Gaussian distribution from the parameters, these parameter encodings can be turned into a latent space distribution.

To go from a latent distribution to a reconstructed image, a specific latent vector can be sampled from this distribution. In the project and homework, this was implemented by epsilon which was sampled from a 0 centered distribution of the specified covariance using the torch random function. Adding this to the mean resulted in the correct distribution sampling. Now with this sampled latent vector, the standard decoding procedure can be used to output a reconstructed image. This sampling also serves to increase the robustness of the model because not only does a latent vector located at the mean have to be able to be decoded into a similar reconstructed image, but also any surrounding point in the distribution that could be sampled must also be decoded into the same reconstruction. This nuance is what allows VAEs to be robust to slight deviations in the input vector, such as through added noise. More importantly a distribution is defined with a volume of realistic values, where new latent vectors can be sampled from. These novel latent vectors can be decoded, generating previously unseen examples of reconstructed images. Although simple mean squared error loss is sufficient to quantify the similarity between a reconstructed output and an input image, there needs to be a different loss function for assessing the distance between distributions.

With variational autoencoders' new goal of creating a normalized latent space that is conducive to generation of new reconstructed outputs, the focus shifts from the reconstructed layer to the distribution. Standard autoencoders are just trying to minimize the loss between the input and reconstructed image, but VAEs are focused on manipulating latent space, trusting that a regularized and optimal latent space will lead itself to producing realistic outputs. Since this is now the focus, it needs to be emphasized through a loss function that will create model parameter updates based on this goal. The goal itself is explicitly to align the models Gaussian distribution encoding of a specific image input x called $q_\theta(z|x)$ with the ideal distribution encoding $p(z|x)$. This distribution is unknown which is why the model chooses to approximate it as Gaussian. Now the difference between the model's encoding and the ideal target encoding

can be quantified through a type of divergence known as Kullback-Leibler divergence. The equation of this divergence is $D_{KL}(q(z|x) || p(z|x)) = \int q(z|x) \log(q(z|x) / p(z|x))$ which integrates over the entire z space, and results in a large value if q(z|x) and p(z|x) are not aligned because dividing q(z|x) by p(z|x) would be 1 if they are similar or small or big if they are different. Taking the log of this ratio would be 0 if it was 1, but be positive if q was bigger and negative if q was smaller. This is then scaled by the q distribution and summed up over all of the space. Although minimizing this divergence would accomplish the goal of aligning the model's encoding distribution with the desired target distribution, nuanced calculation is required to reformat this into a quantity that is computable.

The evidence lower bound (ELBO) is used to define the log likelihood of the given dataspace, and is used as the loss function for standard variational autoencoders through formula (1). It utilizes the difference between the expected estimated posterior that is derived from the latent posterior and the KL-Divergence between this estimated posterior and an assumed standard normal prior.

$$\log p(xi) \geq L = L(\theta, \varphi; x, z, \beta) = E\_q\varphi(z|x) [\log p\theta(x|z)] - DKL(q\varphi(z|x)||p(z)) \quad \textbf{(1)}$$

Through the ELBO, we are able to determine the ability of outputs generated from the VAE's extrapolated distribution to fit to the latent space of the ground truth distribution. The difference between the expectation of the log likelihood that an output z belonging to the generated distribution $q\varphi$ is generated from the ground truth distribution $p\theta$ of the input x and the KL-Divergence of this generated distribution and an assumed prior is treated as the log likelihood of any given x belonging to the true distribution p. This can be used as the loss function for a standard variational autoencoder. By maximizing the loss function, we are in essence maximizing the likelihood of reconstructed values matching the ground truth distribution.

2.2 Fisher Variational Autoencoder

The robustness of the standard VAE can be improved by using Fisher divergence instead of Kullback-Leibler Divergence. Fisher divergence is defined by equation (2).

$$D\nabla [p^*||p\theta] = Ep^*(x) \; 1/2 \; k\nabla x \log p^*(x) - \nabla x \log p\theta(x)k\hat{}2 \quad \textbf{(2)}$$

Intuitively it is the expected value of the difference between the gradients of the log likelihood of each distribution. The gradient of a distribution curve represents the certainty about that point; if it is steeply decreasing on either side, the point is a confident maxima because most of the distribution lies beneath it. For the fisher variational autoencoder, we are seeking to minimize the divergence between the models encoded latent distribution $q_{*,\varnothing}(x,y)$ and the posterior latent distribution $p_{\eta,\theta}(x,y)$. This can be simplified into the following equation

$$
\begin{aligned}
&\simeq \mathcal{L}^{(L)}_{\text{F-AE}}(\mathbf{x};\phi,\eta,\theta) \\
&= \frac{1}{2L}\sum_{l=1}^{L}\Big[\|\nabla_{\mathbf{z}}\log q_\phi(\mathbf{z}^{(l)}|\mathbf{x}) - \nabla_{\mathbf{z}}\log p_\eta(\mathbf{z}^{(l)}) \\
&\qquad - \nabla_{\mathbf{z}}\log p_\theta(\mathbf{x}|\mathbf{z}^{(l)})\|^2 \\
&\qquad + \left\|\mathbf{x} - f_\theta(\mathbf{z}^{(l)})\right\|^2 + \left\|\nabla_{\mathbf{x}}\log q_\phi(\mathbf{z}^{(l)}|\mathbf{x})\right\|^2\Big]
\end{aligned}
\qquad \textbf{(3)}
$$

The model seeks to minimize this fisher divergence loss to better align the model latent space with the posterior, leading to an optimized latent space with and an improved ability to generate new reconstructions from samples.

2.3 Beta Variational Autoencoder

β-variational autoencoders operate near-identically to vanilla VAEs with the exception of an included β hyperparameter included in the loss. This β is used to help modulate the learning constraints model, and emphasizes latent factors by adding a multiple to the KL Divergence included in the loss function **(4)**. Due to the simplistic nature of β-VAEs, it is trivial to perform hyperparameter tuning on the model to determine the optimal β value for each task.

$$F(\theta, \varphi, \beta; x, z) \geq L(\theta, \varphi; x, z, \beta) = E_{q\varphi(z|x)}[\log p\theta(x|z)] - \beta\, D_{KL}(q\varphi(z|x)||p(z)) \qquad \textbf{(4)}$$

Similarly to a vanilla VAE, the β-VAE attempts to describe the latent posterior configurations of the inputs using the probability distribution $q\varphi(z|x)$. The goal of the autoencoder is to capture the explicit independence between latent variables to ensure that reconstruction occurs along these factors. We constrain the latent posterior along an assumed prior $p(z)$ that follows a gaussian standard normal distribution N(0, I). This is where the β parameter begins to act as a regularization coefficient for the KL Divergence of the latent posterior conditioned on the assumed prior. Varying β changes the degree of disentanglement desired from the model, with greater values heavily pressuring the model to discern a more efficient data representation.

3 Details of the project

3.1 Implementation

Our implementation was heavily inspired and borrowed from the example VAE provided in HW4. To implement fisher this required changing the loss function from KL divergence to fisher divergence by calculating the 5 terms in equation 3. A was given in the paper as ∇z log qφ(z (l) |x) = − (l) σ(x), B was -z because we assumed the prior to be uniform, and D was just the norm of the difference between the input and the reconstructed image. Computing C required taking the gradient of the log likelihood, which came from the cross entropy loss between the decoded output and the original x, similar to the standard VAE. E also required calculating the gradient of the log likelihood but this time the distribution was Gaussian. We encoded x and reparameterize by using the mu and logvar outputs to specify a gaussian distribution. The log likelihood was

calculated using the distribution method log_var and then this value was used to compute the gradient with respect to z.

Artificial damage to the data was performed by randomly applying gaussian noise across all pixels following the distribution N(0, sigma), where sigma varied from [0.05, 0.25, 0.5]. The original dataset was preserved via a non-damaging transform in order to later on plot the differences in image reconstruction. The gaussian noise was added in the transform to training, validation, and test data. Black box random erasure was also implemented as a transform to test the ability of image reconstruction when given a missing section of the image, however since the images were created from such small quantities of pixels it was found that significant black box sizes resulted in poor ability of the VAEs to generate accurate images.

3.2 Contribution of each member of the team

We worked together on several sections of the project in order to accomplish debugging and interpretation of papers for implementation. Meeting together to debug code and communicate when questions arose led to hasty progress on more difficult sections of the project.
Leslie wrote the code to corrupt the images with noise and used it to generate and plot all of our results. He also implemented the Beta-VAE and wrote the explanation of the model. Owen implemented the fisher autoencoder and wrote the explanation of fisher as well as the intro and background on standard autoencoders and VAEs.

4 Experimental results

4.1 Data and Hyperparameter Construction

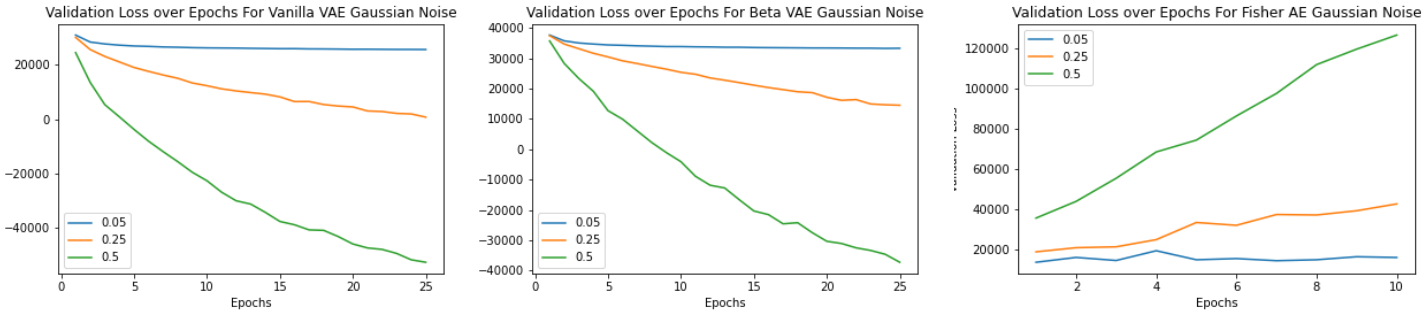
Experimental results for this project were derived from trials of different levels of damage that were added to KMNIST datasets. In order to examine the quality of reconstruction for each of the autoencoder variants, Gaussian noise within the distribution $X \sim N(\mu, \sigma^2)$ with $\mu=0$ and $\sigma^2 = [0.05, 0.25, 0.5]$ was generated on each of the images. “Black box” images were also attempted, in which a ratio of the entire image was deleted and replaced with a box of 0’s. This however was only used as an example of reconstruction, and was not used for final trials due to the inability of sampled VAEs to adequately reconstruct black box missing sections.

Since a β-VAE was used for trials in this project, hyperparameter tuning was performed in order to optimize model performance. β values of [2, 6] were tested at increments of 0.25, with minimum validation loss being recorded at $\beta=4$. This value was used for all further trials.

4.2 Validation Losses

Validation losses were generated using 20% of the total KMNIST datasets. For both the β-VAE and Vanilla VAE validation losses decreased as epochs increased. The quality of image reconstruction as defined in **4.3** is directly proportional to the maximum validation loss as depicted in the below plots for the Beta Vae and the standard VAE, while the minimum validation loss for the Fisher autoencoder equates to better image reconstruction performance. Maximizing the validation loss for both the Beta and Vanilla VAEs equated to better overall image reconstruction as opposed to the minimization of the validation loss according to the Fisher autoencoder. We see that for both the vanilla VAE and Beta VAE, an increase in training epochs results in a decrease in the overall validation loss. Lower levels of noise added through the gaussian transform tend to result in higher loss values for both of these models. For the

Fisher autoencoder however, we see that higher levels of noise result in higher validation losses. As we see the number of training epochs increase for this model its validation losses do tend to slightly increase. This behavior is somewhat unexpected, since training the model should hypothetically reduce validation loss. Despite this, the lowest value of gaussian noise still appears fairly constant at the level of loss. The monotonically increasing 0.25 and 0.5 sigma value gaussian noise samples do show that as trials increase, the loss between the outputs and expected images does tend to increase.

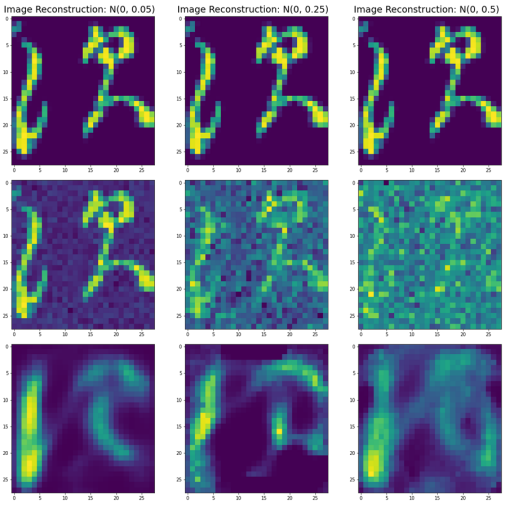


4.3 Image Reconstruction

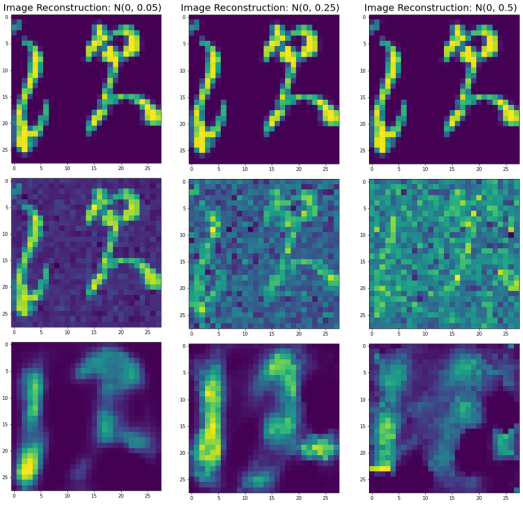
In order to visualize the ability of each of the models to reconstruct images from damaged inputs, test KMNIST samples were used as inputs and the outputs were plotted in a grid of visualizations. The first image in each column is the undamaged original KMNIST image. The second is the visualization of that same image with random gaussian noise applied across at varying levels of sigma from [0.05, 0.25, 0.5]. The final image in each column is the model's reconstruction of the noisy image. As observed, the images generally tend to worsen as the standard deviation of the applied gaussian increases. Since it is more difficult for the models to determine the latent factors of the original image's distribution when there is more noise applied, the reconstruction tends to favor less of these factors and appears more blurry to the viewer.

The best performing reconstructions belong to the Fisher Autoencoder. Since this autoencoder tends to more accurately map the latent factors of the ground truth distribution, it is more capable of reconstructing the original images from noisy damaged inputs. The β-VAE outputs appear to be reconstructed fairly well too. The extra regularization in this model assists in generalizability of the reconstruction to different image classes when compared to the baseline VAE. As we can observe, when the gaussian noise increases across the multitude of trials the reconstructions tend to look blurrier. This is due to the models having a more difficult time discerning latent variables from the original distribution. For the low-noise images however, very accurate and legible outputs are observed, especially for the Fisher autoencoder.

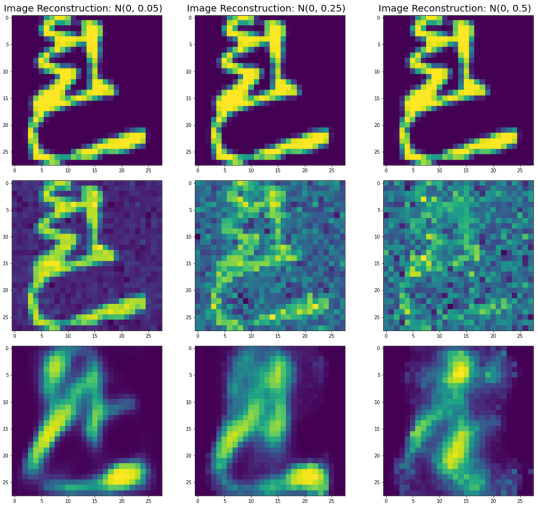
Beta VAE Image Reconstruction



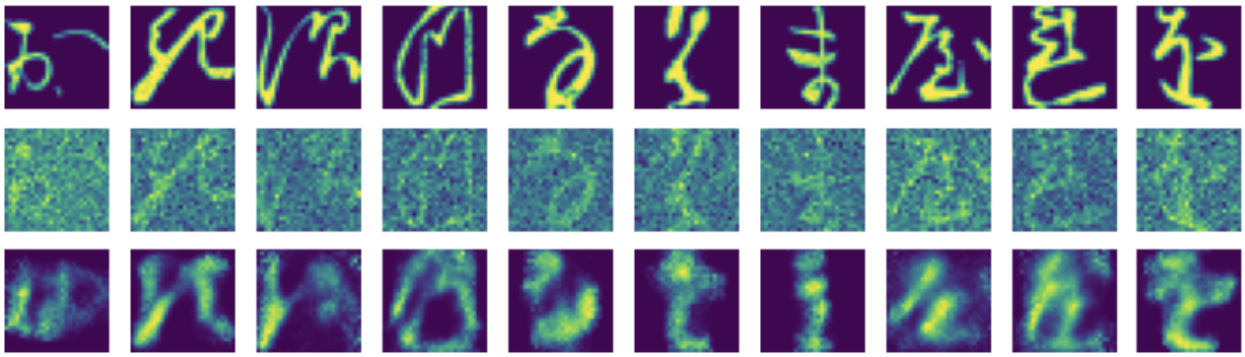
Vanilla VAE Image Reconstruction



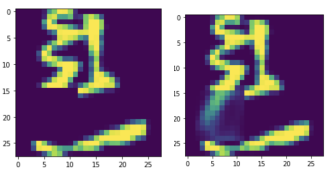
Fisher VAE Image Reconstruction



We can see across the multitude of examples generated from the input KMNIST images transformed by the strongest used gaussian of N(0, 0.5) that fisher is able to very well recreate the original images. This is independent of character class. The primary issue with the reconstructions is that for very small features as depicted by the first column, the Fisher autoencoder is sometimes less capable of discerning the difference between the noise and original feature. These reconstructions are mostly legible across character classes and especially compared between each other, characters are differentiable.



It also shows an ability to recover the missing information from a black box.



5 Concluding remarks

This project hoped to demonstrate the ability of variants of variational autoencoders to extrapolate reconstruction across a wide array of damaged input data. By utilizing the separate classes of KMNIST across noisy images, we expected to see adequate reconstruction of the original image despite gaussian noise being applied to individual images, since the variational autoencoders would capture the latent distribution of the pixels rather than the random gaussian noise applied to the observations. We were able to perform accurate reconstructions for the Fisher Autoencoder along with the original VAE and Beta VAE variants. Overall, this project served to demonstrate how an increase in the ability of a model to represent the latent distribution of the data results in more realistic generated outputs. As follow ups to the experiments performed here, utilizing different types of artificial damage on the KMNIST dataset could be performed to examine the ability of each of the models to reconstruct black box damage for example, or pixel inversion.

References

[1] Anwar, Aqeel. “Difference between AutoEncoder (AE) and Variational AutoEncoder (VAE).” *Medium*, Towards Data Science, 4 Nov. 2021, https://towardsdatascience.com/difference-between-autoencoder-ae-and-variational-autoencoder-vae-ed7be1c038f2.

[2] raptorAcrylycraptorAcrylyc 17511 gold badge11 silver badge88 bronze badges, and zoozoozoozoo 39822 silver badges66 bronze badges. “How Should I Intuitively Understand the KL Divergence Loss in Variational Autoencoders?” *Cross Validated*, 1 Apr. 1966, https://stats.stackexchange.com/questions/394296/how-should-i-intuitively-understand-the-kl-divergence-loss-in-variational-autoen.

[3] *CMU CS Academy*, https://academy.cs.cmu.edu/.

[4] Higgins, I., Matthey, L., Pal, A., Burgess, C.P., Glorot, X., Botvinick, M.M., Mohamed, S., & Lerchner, A. (2017). beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *International Conference on Learning Representations*.

[5] Elkhalil, K., Hasan, A., Ding, J., Farsiu, S., & Tarokh, V.. (2020). Fisher Auto-Encoders.

[6] Stephen G. Odaibo (2019). Tutorial: Deriving the Standard Variational Autoencoder (VAE) Loss Function. *CoRR, abs/1907.08956.*