Encouraging Categorical Meaning in the Latent Space of a VAE

Angelina Wang
26124698
angelina.wang@berkeley.edu

Nathan Blair
3031828892
nblair@berkeley.edu

Suneel Belkhale
3031804231
skbelkhale@berkeley.edu

Abstract

Variational Autoencoders (VAEs) learn latent representations of data that can be useful for downstream tasks. However, the VAE loss function encourages all latent embeddings to cluster around the origin, which while useful for certain tasks, disincentivizes semantically meaningful representations. Furthermore, VAE training is unsupervised, even when label information may be available. We propose enforced clustering in the VAE latent space for a more categorically meaningful learned representation. We achieve this with both supervised and unsupervised methods for latent clustering, each of which outperform our benchmarks on reconstruction and sampling tasks.

1. Introduction

The Variational Autoencoder (VAE) was introduced as a way to efficiently learn a compact latent space that can be used for inference and other learning tasks [11]. Since then, VAEs have been succesfully implemented for denoising, interpolating between data feature and learning compact latent spaces [9, 11, 1, 14]. We can also use VAEs for generative processes like creating random video game models, handwritten text samples, and photorealistic images [6, 8, 4].

In this project, we hope to improve on the latent space representation of VAEs. VAEs take input data and use an Encoder network to convert them into a shortened latent vector that represents a distribution over learned attributes of the data. Then, a Decoder network samples from this distribution and constructs a representation of the original high dimensional data [11].

The original VAE paper does not enforce any clustering of data based on class. We propose both supervised and unsupervised clustering of the data in the latent space. We show that this enables cluster-specific sampling of the latent space in the unsupervised case, and class-specific in the supervised case. Like in the setting of conditional-VAE [15], we are able to interact with high level features of the output image by moving in the latent space. We also show that enforcing clustering in the latent space also may slightly improve reconstruction error.

2. Related Works

The original VAE implementation is strictly unsupervised, and hence has no clustering based on classes [11]. However, later works encourage clustering using Gaussian mixtures as a prior in conjunction with a minimum information constraint [3, 10]. Variational Autoencoders have also been used as a tool for clustering [10].

In the setting of Generative Adversarial Networks (GANs), clustering in the latent space has been shown to be an effective tool for semantic clustering [13]. We take inspiration from the paper by Sudipto et al. which performs clustering in the GAN latent space [13].

3. Background

Before diving into the experiments we performed and different variations of VAEs we used, it is important to first understand VAEs and some of its existing variations. As mentioned in the introduction, VAEs are made up of two networks, the encoder and the decoder. If we denote the image data point as x and the latent noise vector as z, the encoder can be represented as $q_\theta(z|x)$, outputting the parameters of a Gaussian function. Then, when we sample from this distribution, we can feed it into the decoder. This is modeled as $p_\phi(x|z)$, and outputs the probability density of the reconstructed image.

Given the model mechanics of a VAE, we can look more closely at the loss function. It is represented for a single datapoint x_i as $l_i(\theta, \phi) = \mathbb{E}_{z\ q_\theta(z|x_i)}[\log p_\phi(x_i|z) + \mathbb{KL}(q_\theta(z|x_i)||p(z))$. We can decompose this loss term by term. The first term represents the reconstruction loss, based on how well the encoder-decoder network is able to reconstruct a given data point. The second term is the KL-divergence loss, which enforces the constraint that the distribution datapoints are encoded to be close to a normal Gaussian distribution centered at the origin. Intuitively, this term serves as a regulizer and encourages the least amount of information is lost between q and p.

3.1. Beta-VAE

Beta-VAEs [7] are a variation the VAE that, much like our work, attempts to imbue more meaning into the latent space. It hopes to disentangle the different latent dimensions, and thus provide a more meaningful latent space. This is similar to what the InfoGAN [2] does for the GAN [5]. In practice, this goal presents itself through a small modification to the loss function, in which the KL-divergence loss term now has a multiplicative weight of β. Likewise, regular VAE can be seen as Beta-VAE with a β value of 1.0. This bottleneck encourages the model to be much more efficient in its latent representation. This results in a more disentangled and meaningful latent space.

3.2. Conditional VAE

Conditional VAEs (CVAE) [15] are yet another variation of VAE that attempt to embed class information into the latent space. Whereas we hope to provide a disentangled latent space based on categorically meaningfully representations, this VAE maintains an entangled latent space like the original VAE, as can be seen in Figure 3.2. Conditional VAE is able to use class information by first concatenating each data point x_i, as well as its corresponding latent vector z_i, with a one hot encoding of its ground truth class label before being fed into the network. Thus, CVAE uses the concatenated label, along with a latent represenation centered around 0 for all classes, to perform class specific sampling.

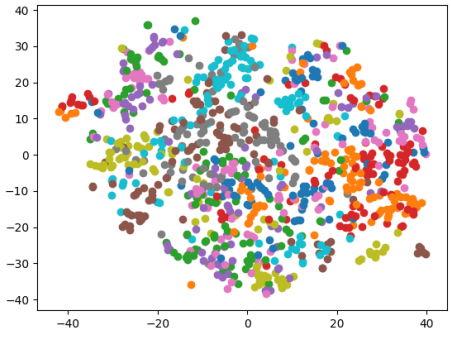

Figure 1. The latent space of a conditional VAE, inherently not structured categorically.

4. Methods

The two methods we used to enforce a more categorically meaningful latent space both intuitively attempt to pushing groups of the same class, away from groups of a different class. Our inspiration for this idea comes from the Siamese Network's contrastive loss function [12], which has the same intended effect of pushing the same classes closer together and different ones further apart.

Our first method, which we will term Lambda-Hot Clustering, comes from the idea that VAEs ignore class information that may be available. Because we know how we want the latent space to look in an ideal world (clusters of each class to be fairly separate from each other), we might as well impose this constraint onto the training. Whereas in the original VAE we had a loss term of the KL divergence from a normal Gaussian distribution, we are now looking at a different KL divergence. If we take a look at $\mathbb{KL}(q_\theta(z|x_i)||p(z))$, what we are doing is changing the prior that we have, $p(z)$, do be a different lambda-hot encoding (one hot encoding but λ instead of one) depending on what our known label y_i is. For example, if x_i has a label of 2, z dimension of 10, and λ value of 5, we will enforce the mean of the encoded latent vector to be $[0, 0, 5, 0, 0, 0, 0, 0, 0, 0]$. What this does, is takes advantage of the knowledge that we have to create class centers at various points in the latent space. We found λ values greater than 1 to be better at this, because the variance is already 1, so enforcing a mean of 1 does not push it out nearly enough.

Sometimes, however, we do not have class labels for our image. In this context, the unsupervised setting of the VAE is essential. To address this, while still pursuing our goal of clustering in the latent space, we propose k-means clustering in the latent space. The k-means algorithm finds clusters that represent different groups of the data without looking at classes. We use the clusters found by k-means to alter the KL divergence loss in a similar manner to that of Lambda-Hot clustering. That is, we compute the KL divergence between the latent vector and its nearest k-means cluster. Furthermore, we update the k-means clusters each epoch.

This allows us to enforce that the natural clusters found in the data are preserved. If two datapoints are initially encoded into separate clusters, those clusters will not be destroyed by the VAE training. Without our method, all latent points are pushed to be clustered around the origin, which is counterintuitive when there is variation in the data. Instead, the natural clusters in the data are enforced by using them to compute loss, and also because cluster centers are recomputed after each epoch. One downside of this method is that we are forced to create our initial clusters based on the data encoded by the untrained network. This means that the clusters may not necessarily be as meaningful. We show in section 5.2 that qualitatively, this method is still able to produce semantically meaningful groupings. Another downside of this method is that it requires the number of clusters to be set as a hyperparameter. In the unsupervised setting, the number of clusters within in the data in not always obvious, especially when some classes may be split into multiple clusters, or multiple classes sent to the same cluster. We argue that this weakness is minor compared to the benefits of doing k-means clustering in the latent space.

4.1. Comparison Metrics

In order to compare our methods with the baselines of a regular VAE, conditional VAE, and Beta-VAE, we decided upon two sets of metrics: qualitative and quantitative. Each was used when relevant on comparisons.

4.1.1 Qualitative Metrics

For our qualititative metrics, there were two that we employed. The first was sampling from different places in the latent space, and looking at what this resulted in the image space. For example, if we cluster the regular VAE in the latent space and find 10 clusters, when we sample will we find that each cluster corresponds to a different digit, or rather would we discover that each cluster corresponds to some aspect of writing such as line thickness. Sampling and feeding the noise vector through the decoder can answer these questions for us.

The second qualitative metric we employed was looking at the t-SNE [16] of the latent space. We colored each data point by the color of its label in order to help better grasp how well class-relevant groupings were learned in the latent space.

4.1.2 Quantitative Metrics

Although incredibly insightful, Qualitative metrics prove a little hard to compare amongst different methods, especially when they are very similar. Thus, we also employed a number of quantitative metrics that allow us to better rank algorithms against each other. The first quantitative metric we used is reconstruction loss. We want to make sure that a model does not solely learn to cluster and have a meaningful latent space, but also still retains the important encoder-decoder relationship that does not lose information when it is fed through the model. The next metric we used has two variations, but the general idea is performing a version of classification using the VAE. We can do this by assigning each of the 10 clusters we find in the latent space to a label, and classifying a test point with whatever label of the cluster that it falls into. We can also do this by adding a simple linear classifier that takes a noise vector from the latent space as input and outputs a class label. The reason we chose such a metric is because if the latent representation of a data point is categorically meaningful, then a classifier should be able to work very well when it only has access to this vector. However, if the learned embedding is not very categorically meaningful, then this end classifier will have a very hard time determining the class label from simply the embedded vector.

5. Experiments

We conducted our experiments on the MNIST [CITE] and FashionMNIST [17] datasets.

5.1. Lambda-Hot Clustering

We tried many different values of λ for these experiments, and found that there were some differences between datasets, but 5.0 struck a nice balance between pushing each class far from others in the latent space, without pushing them so far that the shared features would not be able to be taken advantage of. The latent space had a dimension of 20, where the latter 10 were able to be used to represented the shared features amongst the classes. This is akin to how the InfoGAN [2] has the first few dimensions of its latent space used to provide more specific meaning, and the latter dimensions are for shared characteristics. In our case we hope that the first ten dimensions will provide the class-specific meaning, and the latter will be able to focus on other aspects of the image.

Using the metrics specified in Section 4.1, we perform experiments using a $beta$ value of 5.0 in our baseline experiments. For the qualitative comparisons, we present visuals for the latent space using t-SNE. We can see that in comparison to the regular VAE and beta-VAE, whose purpose is to make the latent space more meaningful, the lambda-hot method very clearly clusters the points more categorically than the other methods, as shown in Figure 5.1 and 5.1 for the two datasets.

For another way to measure how well this clustering works, we apply our second qualitative metric for our method comparing against conditional VAE's. Comparing the latent space of our method to that of a conditional VAE's does not make sense, because a conditional VAE inherently has a latent space without clusters, because the label is provided as extra information that is appended to the latent space. Thus, by comparing against the conditional VAE, whose entire purpose is to be able to sample specifically for each class, we are comparing against the most difficult baseline. In Figures 5.1 and 5.1, it can be seen that by sampling from the latent space from a vector where the first ten dimensions are structured so that we are selecting for a particular class, we are able to successfully enforce what kind of an image it will look like in the latent space. Our method works very well and gives us complete control in selecting the image of the class label we want, simply by carefully sampling from the latent space. Surprisingly, our method qualitatively outperforms Conditional VAE in reliable class specific sampling. This is unexpected because the main point of CVAE is to be able to reliably perform class specific sampling. We hypothesize that by allowing for many dimensions of the latent space to be shared amongst all the classes, and providing the first 10 dimensions with the freedom to pick a particular class, the model is able to delegate

shared features to certain dimensions, and more unique ones to others. This more robust latent space may be the source of our method's success.

After all of these qualitative results that seem to support the idea that our Lambda-Hot Cluster VAE method works better at creating a categorically meaningful latent space, we look to quantitative metrics to make sure that our method has not sacrificed reconstruction loss. Table 5.1 shows that our method has the lowest reconstruction loss (binary cross-entropy loss) for both datasets. Thus, it is clear that our model has preserved reconstruction ability in creating a categorically meaningful latent space, and qualitatively outperforms CVAE in class specific sampling.

As a sanity test to see just how well our model naively performs, we try to use our encoder as a classifier in a very simple way. Given a datapoint x_i, we feed it into the encoder and receive z_i as the embedded latent vector. Then, we take the argmax amongst the first 10 dimensions, and classify the point. This comes from the idea that the distribution the point z_i comes from is centered at $[\lambda, 0, 0, 0, 0, 0, 0, 0, 0, 0]$ for $y_i = 0$. When we perform this naive classification, we find that it has a 91.9% accuracy for MNIST and 81.27% accuracy for FashionMNIST. This shows us that the distributions are indeed very good indicators of what class the datapoints are coming for. There is no reasonable other VAE algorithm we could compare to, because no other model to our knowledge is built to create this categorical structure.

Method/Dataset	MNIST	FashionMNIST
Regular VAE	80.97	228.87
Beta VAE	120.76	250.37
Conditional VAE	80.15	228.14
Lambda-Hot VAE	79.92	227.39
K-Means VAE	**77.59**	**225.68**

Table 1. Reconstruction loss for the different models

5.2. K-Means Clustering

K-Means clustering in the latent space is an unsupervised method, so it is most reasonable to compare it to other unsupervised methods such as Beta-VAE and regular VAE. First, we evaluate the latent space clustering of our K-Means method by looking at the tsne representation of the latent space, as depicted in Figures 5.1 and 5.1. Our method has clear clusters in the latent space, as we expected. However, these clusters do not always correspond to class labels. At least in the tsne representation, some clusters apparently include multiple classes. Thus, our latent space is clearly more separable than regular VAE. However, it is not clear whether K-Means Clustering learns a better latent space representation than Beta-VAE. We claim that our method provides a comparable latent space to Beta-VAE.

Next, we evaluate the K-Means Clustering VAE's ability to sample from its latent space. Specifically we sample from specific cluster centers in the latent space. Note, however, that these clusters do not inherently correspond to class labels. The clusters were simply found during training and represent the natural variation in the data. We see in Figure 5.2 that the K-Means clusters have learned information about classes. For example, in the Fashion-MNIST network, there are clusters that clearly represent short sleeve shirts, long sleeve shirts, pants, purses, boots, and shoes. Interestingly, some of the classes were split into multiple clusters. For example, the boot class was split into boots with low heels and boots with very high heels. Furthermore, some classes, like the sandals class, is underrepresented in our sampled data. We see similar results with MNIST. Zeros and sixes seem to have their own clusters. Ones, on the other hand, have been split into multiple clusters: diagonal ones and straight ones. Furthermore, some clusters are less clear cut. For example, eights and threes are grouped together, as are fours and nines. This makes sense intuitively, as eights and threes, as well as fours and nines are visually similar. For an unsupervised method, K-Means provides a strong ability to sample from semantic groups in the data. The cluster samples are mostly visually compelling. Although, the reconstructed samples are less clearly handwritten digits in the clusters that include multiple classes. For example, the samples from the first cluster of MNIST that seems to represent fours and nines produces visually distorted images.

Finally, we evaluate the reconstruction ability of the K-Means Clustering VAE. Qualitatively, looking at Figure 5.2, we see that the K-Means algorithm is able to successfully reconstruct images with high precision. We examine this quantitatively as well. In Table 5.1, we show that K-Means VAE performs better than all other methods for reconstruction. Essentially, K-Means VAE has all of the reconstruction benefits of regular VAE with all of the clustering and latent space benefits of Beta-VAE. In that sense, K-Means is the best method to use according to our metrics in the unsupervised setting. It has the lowest reconstruction on both MNIST and FashionMNIST of all methods, and also has a semantic latent space. The downsides of the K-Means method is that the clusters that it creates do not always correspond to actual classes. However, this is completely reasonable as we are not providing the network with class information.

6. Conclusion

We presented two new methods of modifying Variational Autoencoders to achieve semantically meaningful latent representations: Lambda-Hot Clustering and K-Means Clustering. Both of these methods modify the KL Divergence loss term in the VAE loss function to support cluster-

ing in the latent space.

Lambda-Hot Clustering is a supervised method similar to Conditional-VAE because it allows for class-specific sampling of the latent representation for generating new data. We show that this method outperforms Conditional-VAE in both image reconstruction and qualitatively in class-specific sampling. Furthermore, Lambda-Hot VAE provides a more intuitive visualization of class information in the latent space than Conditional-VAE. This is because Conditional-VAE intentionally avoids clustering in the latent space and instead relies on the addition of a separate non-random label to be concatenated to the latent vector for sampling. This results in a visually messy latent space.

K-Means Clustering is an unsupervised method like the original VAE and like Beta-VAE. We show that our K-Means Clustering method provides similar latent clustering to Beta-VAE. However, better clustering in Beta-VAE results in high reconstruction error. K-Means Clustering VAE has the semantic latent space of VAE with the reconstruction error of (better than) regular VAE. In fact, of all the methods we tried, K-Means clustering gave the best reconstruction error on both Fashion-MNIST and MNIST. Thus, by our metrics, K-Means Clustering VAE is easily the best unsupervised method to use.

6.1. Future Work

We present the following as possible avenues for future work:

- Expanding the size of our experiments. Due to computation constraints, we evaluated our method on only small datasets with small models. In the future, we should experiment with our method on larger, more complex datasets using deeper and more complex encoder-decoder models.

- Expanding the scope of our experiments. For the purpose of this research, we focused on the image domain. However, we could just as easily apply our method to text, audio, and more. While we expect that our clustering methods will hold up in varying domains, this should still be experimentally confirmed.

- Movable supervised clusters. In this research we experimented with fixed Lambda-Hot clusters in the supervised setting. However, it is also conceivable to allow flexible clusters with a fixed Lambda-Hot prior. This would combine our Lambda-Hot method with our K-Means method. In our Lamdba-Hot method, all classes are given linearly independent and equidistant clusters. In real life, though, some pairs of classes will be more similar than others. Implementing movable clusters would allow our latent representation to better represent the variation in classes.

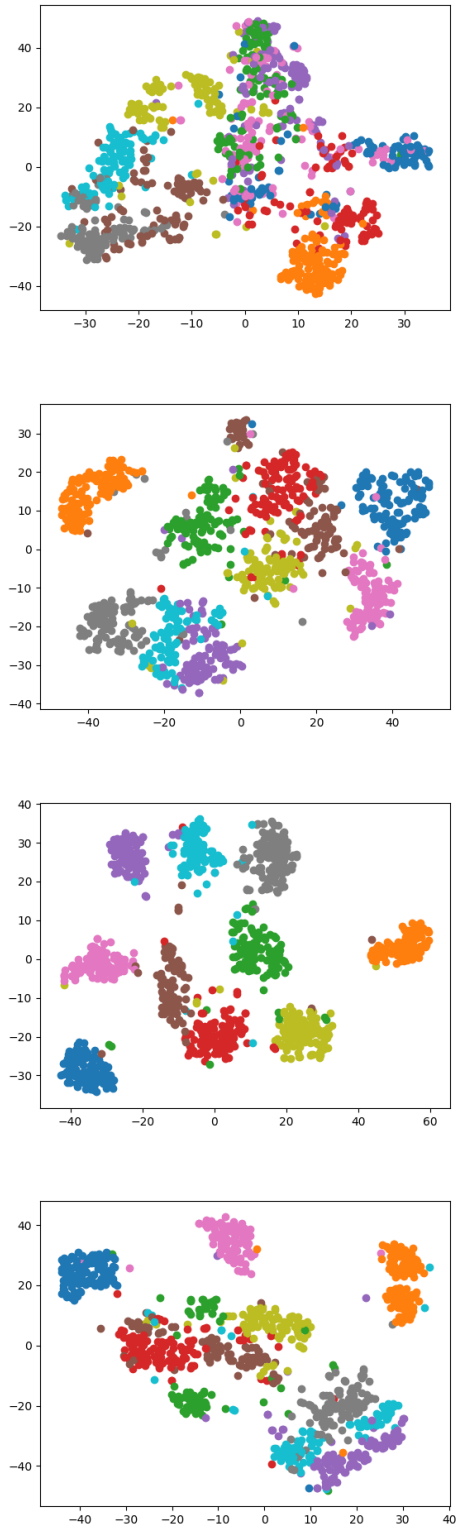

Figure 2. Latent space from encoding test points, with each color denoting a particular class label for MNIST. From top to bottom: Regular VAE, Beta-VAE, Lambda-Hot Cluster VAE, K-Means Cluster VAE

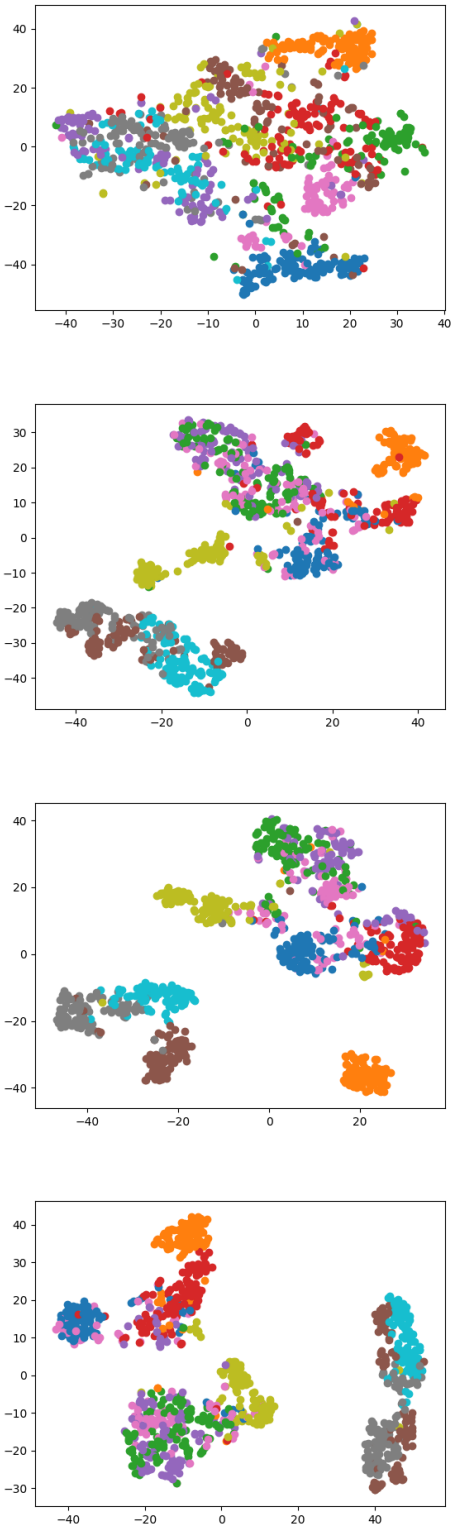

Figure 3. Latent space from encoding test points, with each color denoting a particular class label for FashionMNIST. From top to bottom: Regular VAE, Beta-VAE, Lambda-Hot Cluster VAE, K-Means Cluster VAE

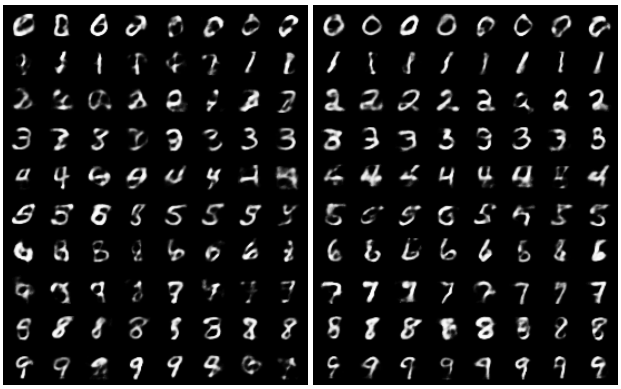

Figure 4. Comparison of generated images between Conditional VAE (left) and Lambda-Hot Cluster VAE (right) on MNIST. Each row shows samples that are specifically picked from the latent space to be of that label.

Figure 5. Comparison of generated images between Conditional VAE (left) and Lambda-Hot Cluster VAE (right) on FashionMNIST. Each row shows samples that are specifically picked from the latent space to be of that label.

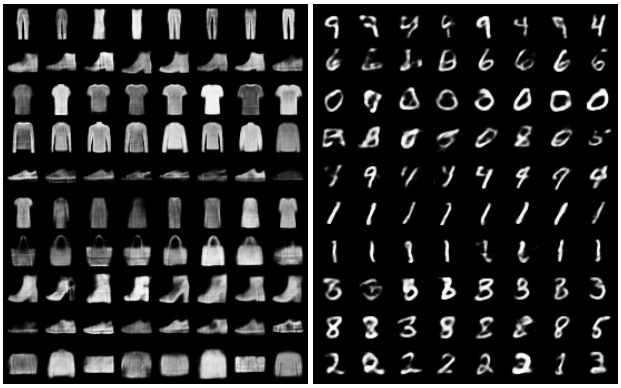

Figure 6. Random Samples around the K-Means cluster centers for Fashion-MNIST (left) and MNIST (Right). Each row corresponds to one of the K-Means clusters.

Figure 7. Reconstructions (bottom) of original images (top) for Fashion-MNIST (left) and MNIST (Right)

References

[1] D. Berthelot, C. Raffel, A. Roy, and I. J. Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *CoRR*, abs/1807.07543, 2018. 1

[2] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Interpretable representation learning by information maximizing generative adversarial nets. *NIPS*, 2016. 2, 3

[3] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *CoRR*, abs/1611.02648, 2016. 1

[4] C. Doersch. Tutorial on variational autoencoders, 2016. cite arxiv:1606.05908. 1

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *NIPS*, 2014. 2

[6] D. Ha and J. Schmidhuber. Recurrent world models facilitate policy evolution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 2450–2462. Curran Associates, Inc., 2018. 1

[7] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017. 2

[8] H. Huang, Z. Li, R. He, Z. Sun, and T. Tan. Introvae: Introspective variational autoencoders for photographic image synthesis. *CoRR*, abs/1807.06358, 2018. 1

[9] D. J. Im, S. Ahn, R. Memisevic, and Y. Bengio. Denoising criterion for variational auto-encoding framework. *CoRR*, abs/1511.06406, 2015. 1

[10] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou. Variational deep embedding: A generative approach to clustering. *CoRR*, abs/1611.05148, 2016. 1

[11] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 1

[12] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. *ICML*, 2015. 2

[13] S. Mukherjee, H. Asnani, E. Lin, and S. Kannan. Clustergan : Latent space clustering in generative adversarial networks. *CoRR*, abs/1809.03627, 2018. 1

[14] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin. Variational autoencoder for deep learning of images, labels and captions. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2352–2360. Curran Associates, Inc., 2016. 1

[15] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3483–3491. Curran Associates, Inc., 2015. 1, 2

[16] L. van der Maaten and G. E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning*, 2008. 3

[17] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arxiv:1708.07747*, 2017. 3