Unsupervised Disentangled Representation Learning

Atharva Khandait
I.I.T. Bombay
160010020
Email: akhandait45@gmail.com

***Abstract*—Learning an interpretable disentangled representation of the independent data generative factors of the world without supervision is an important precursor for the development of artificial intelligence that is able to learn and reason in the same way that humans do. In this Research Project, I explored some GAN and VAE based unsupervised disentangled representation learning models. I speculated that the common problem of Mode Collapse in GANs adversely affects the disentanglement performance of models based on it. I performed some experiments to show that it is indeed true and used it it improve disentanglement performance of some common models. I further performed some experiments on VAE based models and tried to combine them with GAN based models.**

I. Introduction

A disentangled representation can be defined as one where single latent units are sensitive to changes in single generative factors, while being relatively invariant to changes in other factors. For example, a model trained on a dataset of 3D objects might learn independent latent units sensitive to single independent data generative factors, such as object identity, position, scale, lighting or colour, thus acting as an inverse graphics model. According to [1], disentangled representations could boost the performance of state-of-the-art AI approaches in situations where they still struggle but where humans excel. Such representations could be very useful for relevant but unknown tasks.

A significant fraction of unsupervised learning research is driven by generative modelling. It is motivated by the belief that the ability to synthesize, or create the observed data entails some form of understanding, and it is hoped that a good generative model will automatically learn a disentangled representation, even though it is easy to construct perfect generative models with arbitrarily bad representations. The most prominent generative models are the Generative Adversarial Network (GAN)[2] and the Variational Autoencoder (VAE)[3]. The most prominent disentangled representation learning models based on these are the InfoGAN[4] and the β-VAE[5].

II. Papers

A. InfoGAN

InfoGAN is a simple modification to the generative adversarial network objective that encourages it to learn interpretable and meaningful representations. It achieves this by maximizing the mutual information between a fixed small subset of the GANs noise variables and the observations. The paper shows that it was able to discover highly semantic and meaningful hidden representations on a number of image datasets like MNIST, CelebA, SVHN. The results suggest that generative modelling augmented with a mutual information cost could be a fruitful approach for learning disentangled representations.

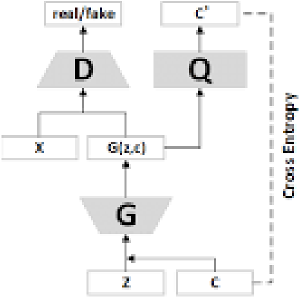

Fig. 1. InfoGAN model

The minimax game of a GAN is given by the following expression:

$$\min_G \max_D V(D,G) = E_{x\sim P_{data}}[\log D(x)]+E_{z\sim noise}[\log(1-D(G(z)))] \tag{1}$$

1) Maximizing Mutual Information: This formulation uses a simple factored continuous input noise vector z, while imposing no restrictions on the manner in which the generator may use this noise. As a result, it is possible that the noise will be used by the generator in a highly entangled way, causing the individual dimensions of z to not correspond to semantic features of the data.

In an InfoGAN, rather than using a single unstructured noise vector, we decompose the input noise vector into two parts: (i) z, which is treated as source of incompressible noise; (ii) c, which we will call the latent code and will target the salient structured semantic features of the data distribution.

We now provide the generator network with both the incompressible noise z and the latent code c, so the form of the generator becomes $G(z, c)$. The mutual information between latent codes c and generator distribution $G(z, c)$, $I(c; G(z, c))$ should be high.

So, the information-regularized minimax game of the InfoGAN is given by:

$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda I(c; G(z,c)) \tag{2}$$

$$\min_{G,Q} \max_D V_{InfoGAN}(D,G,Q) = V(D,G) - \lambda L_I(G,Q) \tag{3}$$

2) Implementation: In practice, we parametrize the auxiliary distribution Q as a neural network. In most experiments, Q and D share all convolutional layers and there is one final fully connected layer to output parameters for the conditional distribution $Q(c|x)$.

For categorical latent code c_{i} , we use the natural choice of softmax nonlinearity to represent $Q(c_{\text{i}}|x)$. For continuous latent code c_{j}, we consider $Q(c_{\text{j}}|x)$ to be a Gaussian. The individual units in this Gaussian are linearly independent.

3) The Mode Collapse problem: Mode Collapse(sometimes called Helvetica Scenario) is one of the biggest problems with the training of GANs. The basic idea is that the generator can accidently start to produce several copies of exactly the same image(extreme case), or a majority of the images will have some similar traits(partial case) which is more common. Examples of these traits are reddish tint on all the images, all lips bright red or all faces smiling on a faces dataset like CelebA.

Mathematically, G tries to minimize $E_{z\sim noise}[\log(1 - D(G(z)))]$. In other words, to generate the point $x^* = G(z)$ such that $x^* = argmaxD(x)$(We are assuming D is fixed). x^* is fixed regardless of the value of z. There's nothing in the objective function that forces the generator to generate different samples given the input.

The reason for this problem is, in the original minimax formulation(eq. (1)), what we really want is minimization over G in the outer loop and maximization over D in the inner loop. But, if we swap the order, we end up encouraging the generator to behave as explained above.

$$\min_G \max_D V(D,G) \neq \max_D \min_G V(D,G) \tag{4}$$

In practice, what we generally do is we train both the generator and discriminator simultaneously. Unfortunately, we usually get results that look more like $\max_D \min_G V(D,G)$ than $\min_G \max_D V(D,G)$.

I will explain how this general problem of GANs affects the disentangling performance of InfoGAN later in section III. First, let's look at a solution to the problem of Mode Collapse.

B. Unrolled GANs

This paper[6] defines the generator objective with respect to an unrolled optimization of the discriminator. This allows training to be adjusted between using the optimal discriminator in the generators objective, which is ideal but infeasible in practice, and using the current value of the discriminator, which is often unstable and leads to poor solutions. The authors have shown how this approach stabilizes training of GANs and solves the problem of mode collapse, thus increasing the diversity and coverage of the data distribution by the generator.

1) Differentiating through optimization: Many optimization schemes, including SGD, RMSProp, and Adam, consist of a sequence of differentiable updates to parameters. Gradients can be backpropagated through unrolled optimization updates in a similar fashion to backpropagation through a recurrent neural network.

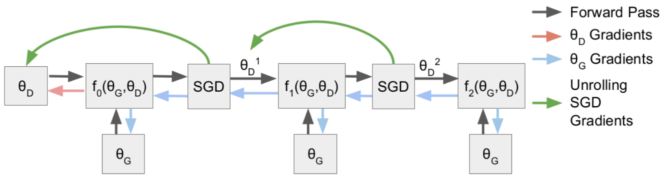

Fig. 2. The computation graph for an unrolled GAN with 3 unrolling steps.

C. Unrolling

The original learning problem of the GAN is to find optimal parameters of the generator θ_G^*:

$$\theta_D^*\left(\theta_G\right) = \theta_D argmax f\left(\theta_G, \theta_D\right) \tag{5}$$

$$\theta_G^* = \theta_G argmin \max_{\theta_D} f\left(\theta_G, \theta_D\right) \tag{6}$$

$$\theta_G^* = \theta_G argmin f\left(\theta_G, \theta_D^*\left(\theta_G\right)\right) \tag{7}$$

Here, f is $V(D,G)$ from eq. (1)

Explicitly solving for the optimal discriminator parameters $\theta_D^*\left(\theta_G\right)$ for every update step of the generator G is computationally infeasible for discriminators based on neural networks. Therefore, this minimax optimization problem is typically solved by alternating gradient descent on θ_G and ascent on θ_D. That is what usually leads to unstable training and mode collapse.

In order to address this, the paper introduces a surrogate objective function $f_K\left(\theta_G, \theta_D\right)$ for the generator which is closer to the true generator objective $f\left(\theta_G, \theta_D^*\left(\theta_G\right)\right)$.

A local optimum of the discriminator parameters θ_D^* can be expressed as the fixed point of an iterative optimization procedure:

$$\theta_D^{k+1} = \theta_D^k + \eta \frac{\mathrm{d}f\left(\theta_G, \theta_D^k\right)}{\mathrm{d}\theta_D^k} \tag{8}$$

$$\theta_D^*\left(\theta_G\right) = \lim_{k\to\infty} \theta_D^k \tag{9}$$

where η is the learning rate. By unrolling for K steps, we create a surrogate objective for the update of the generator:

$$f_K\left(\theta_G, \theta_D\right) = f\left(\theta_G, \theta_D^K\left(\theta_G, \theta_D\right)\right) \tag{10}$$

When $K = 0$ this objective corresponds exactly to the standard GAN objective, while as $K \to \infty$, it corresponds to the true generator objective function $f\left(\theta_G, \theta_D^*(G)\right)$.

The gradient of the surrogate loss w.r.t. the generator parameters:

$$\frac{\mathrm{d}f_K(\theta_G,\theta_D)}{\mathrm{d}\theta_G} = \frac{\partial f\left(\theta_G,\theta_D^K(\theta_G,\theta_D)\right)}{\partial\theta_G} + \frac{\partial f\left(\theta_G,\theta_D^K(\theta_G,\theta_D)\right)}{\partial\theta_D^K(\theta_G,\theta_D)}\frac{\mathrm{d}\theta_D^K(\theta_G,\theta_D)}{\mathrm{d}\theta_G} \tag{11}$$

Fig. 3 shows how unrolling leads to the generator capturing all the modes of the dataset distribution. Without unrolling, the generator jumps from one mode to another, whatever fools the discriminator at a particular point in training, but never really converges.

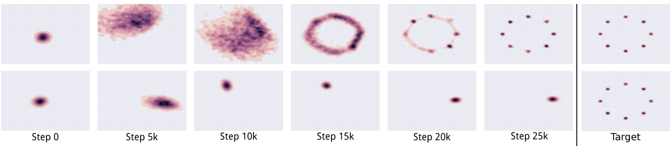

Fig. 3. Unrolling the discriminator stabilizes GAN training on a toy 2D mixture of Gaussians dataset. The top row shows training with 10 unrolling steps, no unrolling in the bottom row.

III. Unrolled InfoGAN - original contribution

Now, let's come back to InfoGANs. As we saw in Fig. 3, a standard GAN framework will frequently(for a lot of random seeds and hyperparameter choices) collapse to one or more modes of the data and not capture all the modes. What a InfoGAN is trying to do is capture these modes of the data distribution in a way that individual latent units are sensitive to changes in a single mode. This is basically what disentangling is. But, it can't do this if the generator can't even capture all the modes of the data. So, I theorize that solving the problem of Mode Collapse will in turn also improve the disentangling performance of InfoGAN, and for that matter, any GAN based disentangled representation learning model.

A. Modified objective

Using eq. (3)(formulation of the InfoGAN) and eq. (10)(Surrogate objective for the generator with an unrolled discriminator), let's define the objective for the generator of this model as:

$$g_K\left(\theta_G, \theta_D, \theta_Q\right) = f_K\left(\theta_G, \theta_D\right) - \lambda L_I(G, Q) \tag{12}$$

which gives us:

$$g_K\left(\theta_G, \theta_D, \theta_Q\right) = f\left(\theta_G, \theta_D^K\left(\theta_G, \theta_D\right)\right) - \lambda L_I(G, Q) \tag{13}$$

Updating the parameters of the generator w.r.t. $L_I(G,Q)$(loss from the Q network) will be independent of the unrolled discriminator. Since, the Q network shared most of its parameters with the discriminator, this part of the objective will be optimized using the parameters of an non-unrolled discriminator.

B. Experiments

The goal of the experiments is to verify that capturing more modes of the distribution leads to better disentanglement performace. To do this, I trained the Unrolled InfoGAN on the MNIST dataset with $K = 0$(no unrolling), $K = 5$, $K = 7$, $K = 10$. For the Q network, I used 1 discrete latent variable(1-10) and 2 continuous variables. I inspected and compared their disentangling performance both visually and numerically(by comparing $L_I(G,Q)$).

In all figures of latent code manipulation, we will use the convention that in each one continuous latent code varies from left to right, the discrete latent code varies vertically, while the other latent codes and noise are fixed.

It is to be noted that these results have been obtained using barely any hyperparameter tuning. The InfoGAN paper has better results. The problems I am trying to solve includes the problem that training of GANs is stable and converges to the distribution for only a few choices of hyperparameters. Unrolling not only solves the problem of mode collapse but also stabilizes and assures convergence for a much larger range of hyperparameters.

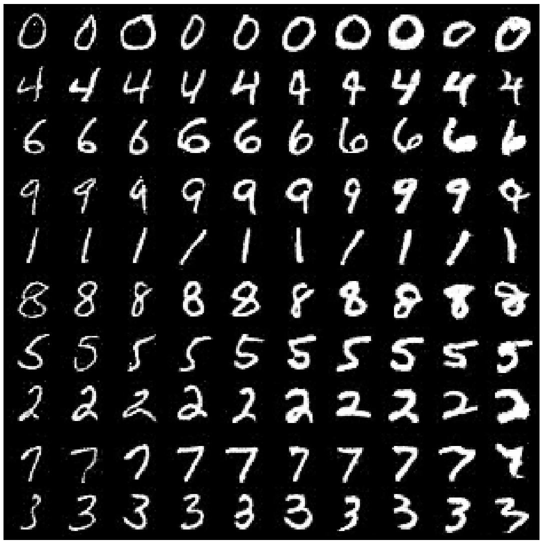

Fig. 4. Continuous latent variable 1 for $K = 0$

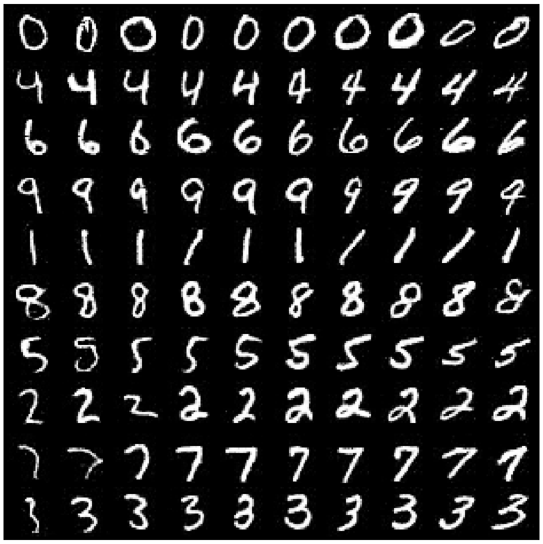

Fig. 5. Continuous latent variable 2 for $K = 0$

We can see in Fig. 4 that this latent variable has mostly tried to disentangle stroke and digit thickness. However, it frequently rotates some of the digits which shows it has not been completely disentangled. Also, the discrete latent code has almost completely disentangled all the digit types into its 10 classes. Although, for the case of digit 9, it collapses to 4 at the end. In Fig. 5, the 2nd latent variable has mostly learnt to disentangle digit rotation.

We see improvement in both the continuous latent variables

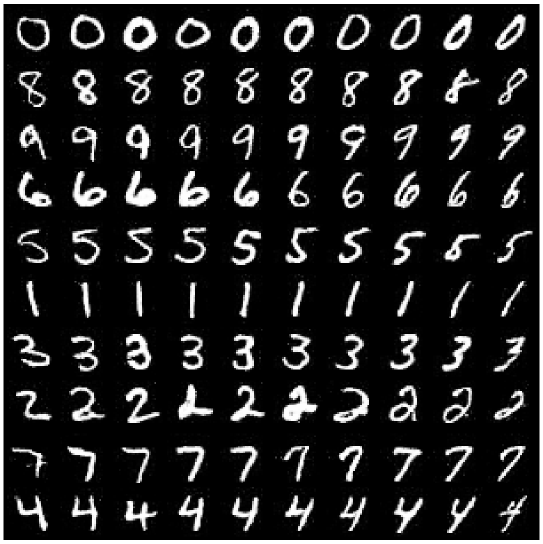

Fig. 6. Continuous latent variable 1 for $K = 5$

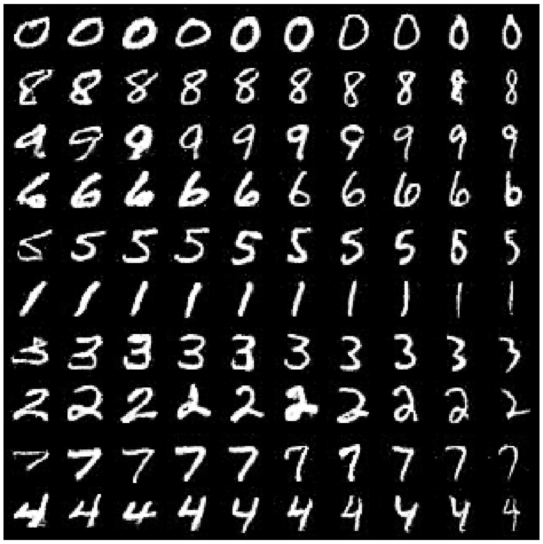

Fig. 7. Continuous latent variable 2 for $K = 5$

when we unroll with $K = 5$. In Fig. 6, it learns rotation of digits with minor thickening in some cases. In Fig. 7, it learns digit thickness with some entangled rotation from left to right. We can see that the overall quality has clearly improved.

Unrolling with $K = 7$ further improves the disentanglement performance of both the latent variables(Fig. 8 and 9). Also, the digit 9 no longer collapses to the digit 4 int he discrete latent variable.

Now, with $K = 10$, both continuous latent variables and the discrete latent variable have been almost perfectly disentangled from each other. In Fig. 10, the variable completely disentangled digit width and stroke thickness with no signs of rotation and in Fig. 11, the latent variable completely disentangled digit rotation.

Let's verify these observation numerically by plotting $L_I(G, Q)$ for the continuous latent variables. The discrete latent variable is relatively easy to learn and its loss drops to near 0 quickly irrespective of the value of K.

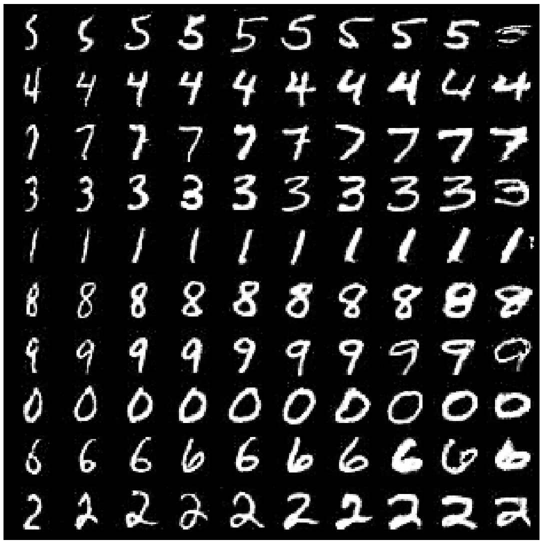

Fig. 8. Continuous latent variable 1 for $K = 7$

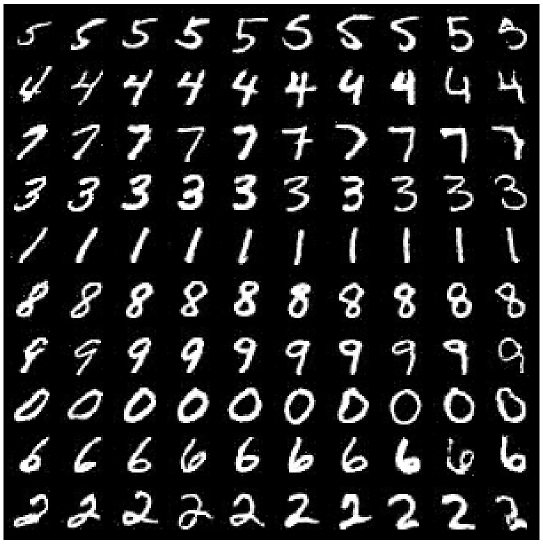

Fig. 9. Continuous latent variable 2 for $K = 7$

Q Loss	K=0	K=5	K=7	K=10
Continuous	0.14	0.076	0.072	0.012
Discrete	0.02	0.006	0.0015	0.0006

From Fig. 12 and the above table, we can see a clear correlation between the unrolling steps K and the disentangling performance of the model.

I can say with fair confidence that this technique will work not only for InfoGANs but any GAN based disentangling frameworks.

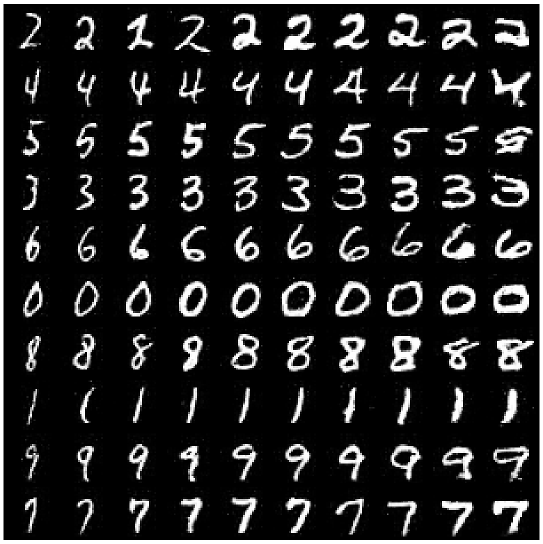

Fig. 10. Continuous latent variable 1 for $K = 10$

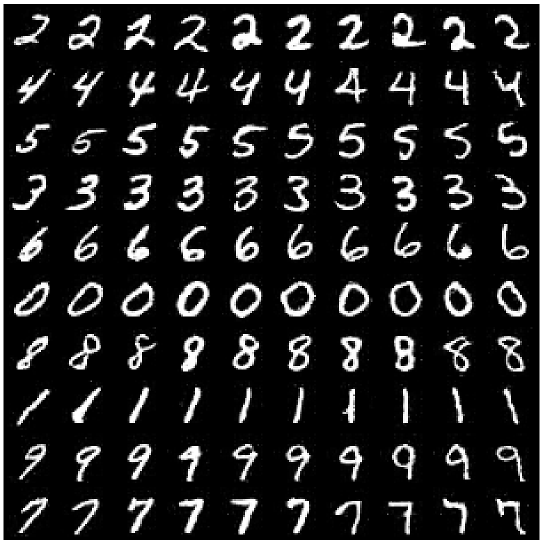

Fig. 11. Continuous latent variable 2 for $K = 10$

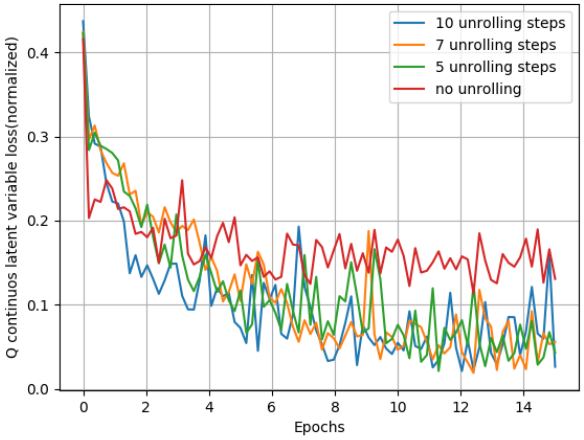

Fig. 12. Variation of Q continuous variable loss with training epochs for different K values.

IV. Variational Autoencoder Based Models

One of the properties of GANs which can limit its application for certain tasks is that it does not learn an explicit distribution for the dataset. In other words, it cannot map(encode) an image from the dataset to the latent space(dimensionality reduction $(X \rightarrow Z)$). The variational autoencoder(VAE) is capable of that. With such models, we can map any image from the data distribution to the latent space, modify the latent variables, map them back to the data distribution(generate images). Thus, we can see see the effect of the latent variables on images of our choice.

The β-VAE is a prominent disentangled representation learning model based on the VAE. They augment the original VAE objective by a hyperparameter β that is multiplied with the KL Divergence term. This constrains the limit of the capacity of the latent information channel and control the emphasis on learning statistically independent latent factors.

However, VAEs have one problem of their own. They are optimized by used a pixel-wise cross-entropy or mean-squared loss. Element-wise metrics are simple but not very suitable for image data, as they do not model the properties of human visual perception. E.g. a small image translation might result in a large pixel-wise error whereas a human would barely notice the change. This leads to their generated images being blurry. This limits its use to generating images for practical purposes.

The paper [7] has tried to solve this problem by combining GANs with VAEs. The paper achieves this by combining a VAE with a GAN as shown in Fig. 13. They collapse the VAE decoder and the GAN generator into one by letting them share parameters and training them jointly. For the VAE training objective, they replace the typical element-wise reconstruction metric with a feature-wise metric expressed in the discriminator.

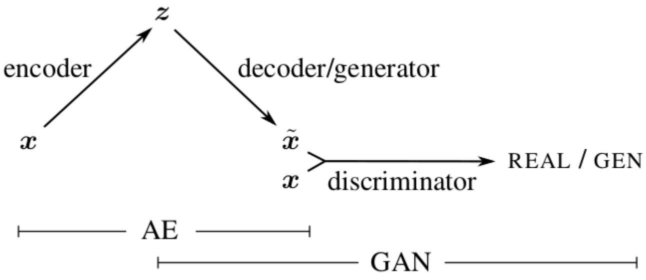

Fig. 13. Overview of the VAE-GAN network.

We can use the β-VAE hyperparameter even when it is combined with a GAN and get results with high visual fidelity. However, InfoGAN ranks higher than β-VAE on a lot of recent disentanglement metrics. I theorize that it is due to a strong assumption InfoGAN makes about the data distribution. It assumes that most of the input variables to the generator are incompressible noise and it maximizes information for only a few of its input variables. I think this assumption works well because a lot of complex datasets have some factors that are heavily correlated and cannot be disentangled. A lot of them can be minor factors like lighting and skin color. On the other hand, β-VAE does not make any such assumption and imposes its constraint on all of the latent variables. It leaves it to the model to figure out which factors can be disentangled and which cannot, a task which may prove too difficult for bigger and complex datasets.

I try to combine the best of both worlds in the following experiments.

A. Experiments

The aim of this experiment was to construct a model which can encode an image to a disentangled latent space, while making an assumption that only a few latent variables can learn disentangled representations. I tried to do this by combining a VAE with an InfoGAN. We maximize the mutual information between some of the latent variables and the output of the Q network. The GAN network has been unrolled as we have seen it improves disentanglement performance.

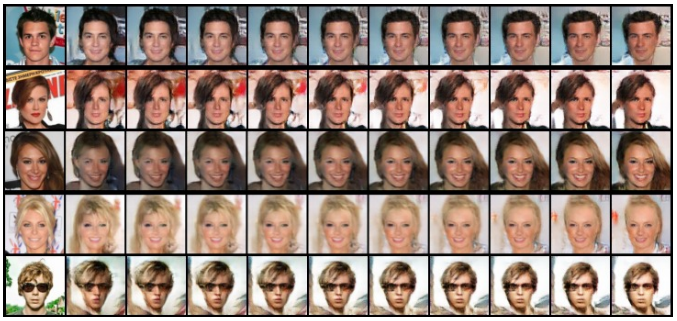

Fig. 14. We vary latent variables which have learnt disentangled representations of some factors. These latent variables are encoded from images from the dataset(1st column) which is not possible with InfoGAN.
From top to bottom, the variables have learnt the following factors of variation: 1) Age 2) Gender 3) Hair color 4) Hair length 5) Sunglasses add/remove

We can see in Fig 15. that the KL losses for variables that were assumed to be incompressible is lower than the ones for which we maximized information w.r.t. the Q network. We can say from this observation, that variables that learn disentangled representations are further away from the standard normal distribution than elements that don’t.

V. Conclusion

In this project, I visited various frameworks for disentangled representation, tried to understand how and why they work, and tried to come up with modifications/ideas from other papers that will improve their performance. Future work could include comparing their performance using disentangling metrics, diving deeper into the theory and implementing these frameworks on real-world problems. More work can also go in the last experiment which combines VAE-InfoGAN to improve its performance and understand its latent space.

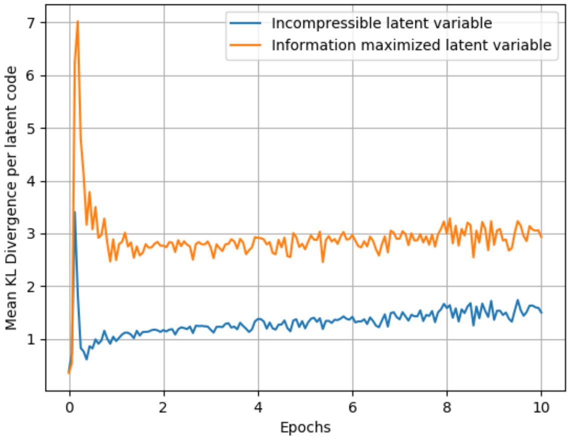

Fig. 15. The average per latent variable KL loss for variables that were assumed to be incompressible and variables for which Information was maximized.

References

[1] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman, *Building machines that learn and think like people.*

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

[3] D. P. Kingma and M. Welling, *Auto-encoding variational bayes.*

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel, *Interpretable representation learning by information maximizing generative adversarial nets.*

[5] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A, *Beta- VAE: Learning basic visual concepts with a constrained variational framework.*

[6] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, *Unrolled Generative Adversarial Networks.*

[7] Anders Boesen Lindbo Larsen, Sren Kaae Snderby, Hugo Larochelle and Ole Winther, *Autoencoding beyond pixels using a learned similarity metric.*