Latent Space Arithmetic with Images and Texts

Zhengyan Lyu [1]

Abstract

Despite the recent progress in high fidelity image synthesis with Generative Adversarial Networks (GANs), the attributes of synthesis are mostly binary so the synthesis is not transferable for unknown attributes. In this work, I propose a novel framework for semantic face editing with words by interpreting the latent semantics learned by GANs. In this framework, I conduct a study on how words can be encoded in the latent space of GANs. I find that the word embedding latent space can be projected to the latent space of GANs with semantics. The projection is continuous, since the word vectors that are close in the word embedding space can have similar projection. In addition, my model is able to learns a disentangled representation of the word in the GAN latent space by linear transformations.

1. Introduction

Generative Adversarial Network (GAN) (Goodfellow et al., 2014) is very popular as a technique for image synthesis in recent years. GANs can learn the mapping from a latent vector to the real data through adversarial training. After learning the non-linear mapping, the GAN generator can generate photo-realistic images from randomly sampled latent vectors. Taking face synthesis as an example, In recent years, many studies are conducted on semantic face editing. Existing work typically focuses on improving the synthesis quality of GANs of some certain attributes such as gender or hair color.

There are some previous works that utilizes GANs with word embeddings to generate images (Zhang et al., 2018), but generally they did not provide a detailed understanding about the connection between image latent space and word latent space.

In this work, I propose a novel framework for semantic face editing with words by interpreting the latent semantics learned by GANs. In this framework, I conduct a study on how words can be encoded in the latent space of GANs. I find that the word embedding latent space can be projected to the latent space of GANs with semantics. We observe that the projection is slightly continuous. More research may be conducted on the continuity of this projection

My contributions are summarized as follows:

- I propose a neural network model to explore how word embeddings are encoded in the latent space of GANs, such as DCGAN (Radford et al., 2016). The neural network spontaneously learn various latent subspaces corresponding to word.
- I propose a new training process and loss function that can help the model to achieve a balance between disentanglement and quality of the projection of the words by linear transformations.

2. Related Work

2.1. Generative Adversarial Networks

GAN (Goodfellow et al., 2014) has brought wide attention in recent years due to its great potential in producing photo-realistic images . GANs can learn the mapping from a latent vector to the real data through adversarial training. After learning the non-linear mapping, the GAN generator can generate photo-realistic images from randomly sampled latent vectors. To make GANs applicable for real image processing, existing methods proposed to reverse the mapping from the latent space to the image space or learn an additional encoder associated with the GAN training.

2.2. Latent Space of GANs

Latent space of GANs is generally treated as Riemannian manifold (Arvanitidis et al., 2018). Many prior work focused on exploring how to make the output image vary smoothly from one synthesis to another through interpolation in the latent space, regardless of whether the image is semantically controllable. Many works observed that when linearly interpolating two latent codes z1 and z2, the appearance of the corresponding synthesis changes continuously

[1] Brown University. Correspondence to: Zhengyan Lyu <zhengyan_lyu@brown.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

(Brock et al., 2019). Some work has observed the vector arithmetic property I use this observation as the precondition of the whole model. Despite the tremendous success of GANs, little work has been done on the connection between the GAN image latent space and the word latent space.

2.3. Semantic Face Editing with GANs

Compared to GANs which can generate image without conditions, semantic editing requires the model to generate an image that only modify the target attributes but keep other information of the input unchanged. To achieve this goal, current methods required carefully designed loss functions, or special architectures of the models. Semantic editing using a more complicated dataset with masks about. Shen et al. (Shen et al., 2020) show that linearly edition the GAN latent space can also be disentangled. However, most of their works were limited to binary attributes.

3. Methodology

3.1. Preprocessing

GloVe (Pennington et al., 2014) is used as the pre-trained word embedding. Since the attribute word inputs for the model is very limited, the embedding layer is not trainable.

Since the encoder cannot effectively learn the environment in the image, I use MTCNN (Zhang et al., 2016) to extract the faces from the images and thus create a new dataset for training.

3.2. Pretraining

3.2.1. Generative Adversarial Network (GAN)

DCGAN (Radford et al., 2016) is used as the GAN generator model. A vector generated by random sampling is fed into the decoder and a face image is generated based on this vector. The generated images is then mixed with the original images and the discriminator would try to distinguish between real images and generated images.

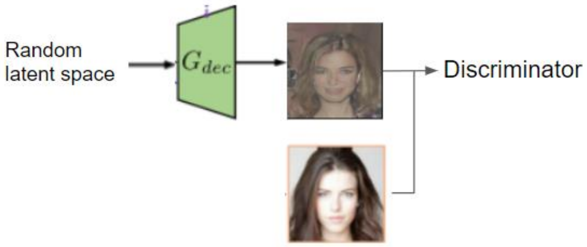

Figure 1. the GAN model

3.2.2. Encoder

The encoder can compress the original image into a vecto in the latent space of the trained GAN model. The structure of the encoder is similar to the generator except that convolutional layers instead of convolutional transpose layers are used for the encoder. MSE loss is used to train the encoder.

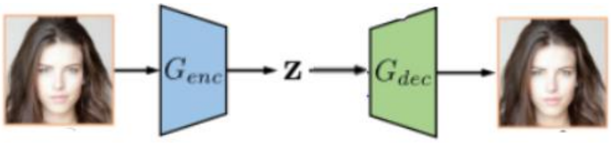

Figure 2. the Encoder model

3.3. Architecture

Two images are randomly selected from the dataset. The word forms of different attributes between two images are treated as the inputs of the embedding layers. The word embeddings are then fed into some fully connectly layers to obtain a vector with semantics in the GAN latent space. The assumption is that by editing the first image based on the words given, the model can generate an image that is similar to the second image.

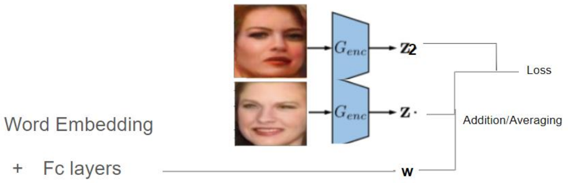

Figure 3. The Achitecture of model

3.3.1. Vector Averaging

In the training process, I assume that for any vocabulary, there exists a hyperplane in latent space such that all samples from the same side have this word as its attribute. Therefore, I would be able find a typical vector in the latent space that is far away from the hyperplane in every dimension for any vocabulary. The output is the average of the vector of the original image in the latent space and the average vector of all typical vectors of different words in the latent space.

$$z' = (z + E(w))/2 \tag{1}$$

This result is then compared with the target vector using MSE loss.

3.3.2. Vector Addition

Vector averaging has several shortcomings. First, it is hard to only modify the target attributes but keep other information of the input unchanged since the other characteristics of the typical image of the word will be brought through averaging. Second, it is hard to decide the proportion each word in the training process. In the experiment I used averaging, but it is not ideal because the distance between the projection of two points of the images in the latent space on the eigenvector of the hyperplane that seperates the word is different for every image pair.

$$z' = z + \Sigma(w) \tag{2}$$

Therefore, I propose a new loss function for training. By finding an ideal parameter θ that is multiplied to each word vector in the latent space before added to the latent vector of the original image, the euclidean distance between the latent vector of the generated image and that of the target image can be minimized. Assume that every word inputted into the model are independent to each other. This assumption is not correct in reality, but in this way, the model would try to learn vectors of the words in the latent space that are independent to all other vectors that could be inputted together. Here, independent is a synonym of perpendicular. Thus, when finding the ideal parameter for one word vector in the latent space, the other word vectors can be ignored. The ideal parameter for each vector can be found by derivation.

$$\frac{d|(z + \theta w) - z2|}{d\theta} = 0 \tag{3}$$

$$\frac{d\sum_{i=1}^{latent\_dim}((z_i - z2_i)^2 + 2\theta w_i(z_i - z2_i) + \theta^2 w_i^2)}{d\theta} = 0 \tag{4}$$

$$\sum_{i=1}^{latent\_dim}(2w_i(z_i - z2_i) + 2\theta w_i^2) = 0 \tag{5}$$

$$\theta = -\frac{\sum_{i=1}^{latent\_dim} w_i(z_i - z2_i)}{\sum_{i=1}^{latent\_dim} w_i^2} \tag{6}$$

The whole process is linear. Let $D = z - z2$, the number of input words be n, and latent dimension be l_d. Then the loss function becomes:

$$Loss = \frac{1}{l\_d}\sum_{k=1}^{l\_d}((\sum_{j=1}^{n}(-\frac{\sum_{i=1}^{l\_d} w_{ji}D_i}{\sum_{i=1}^{l\_d} w_{ji}^2}w_j))_k - D_k)^2 \tag{7}$$

where w is output with shape (n, l_d) and D is label with shape (l_d).

4. Experiment

4.1. Data

CeleA dataset (Liu et al., 2015) is used for training. Because of the limitaion of the hardware, I compressed the images into 80*80 pixels. For convinence, I converted word phrases in the dataset to single words by their meaning. In this way, the model do not need to learn a joint word embedding space.

4.2. Pretraining

Both the encoder and the decoder are trained for 20 epochs. For the original dataset, the generator can generate images with high quality, while the encoder cannot effectively learn the details and the environment in the image for the original dataset.

4.3. Results

Vector averaging fails to generate an image that only modify the target attributes but keep other information of the input unchanged. θ for vector addition can be inputted manually, indicating the degree of manipulation on the new image. θ can also be a negative value. Compared to vector averaging, vector addition is able to keep most other information in the image unchanged, indicating that the word latent vectors that may appear at the same time are generally perpendicular to each other. However, a completely independent vector in the vector space is not learnt by the model. For example, by modifying lipstick, the entire makeup becomes heavy because of the biases in the dataset.

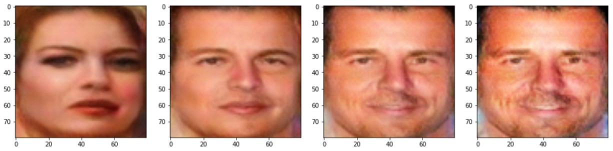

Figure 4. The results of semantic face editing with vector averaging. Word input: male

In addition, I feed the output latent space vectors of different words into generator for curious. It shows that the generated images are actually related to the semantics of the words even though it should not be outputted directly for both training and testing. Meanwhile, I find that the space is continuous. Words with similar meaning will be projected into similar vectors in the latent space, and the images generated based on the similar vectors are also similar.

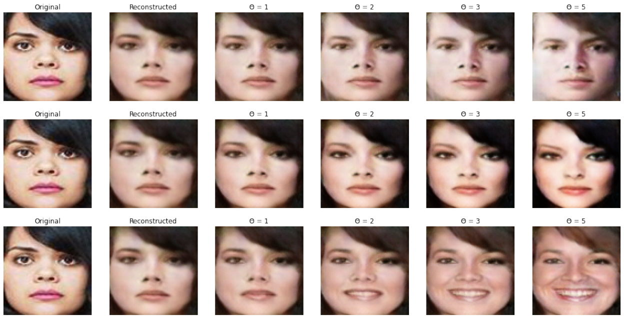

Figure 5. The results of semantic face editing with vector addition. X-axis: Original, Reconstructed, θ =1, θ =2, θ =3, θ =5 Y-axis: male, lipstick, smile

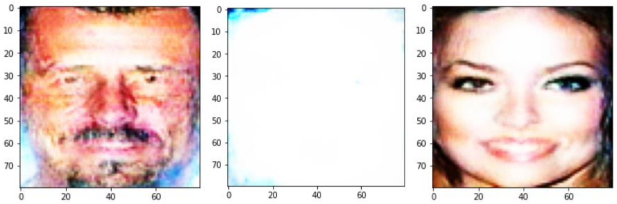

Figure 6. The images of the word projections

5. Future works

There are many related fields that can be further explored about this project. For the failure in the encoder for the original dataset, a more advanced GAN model with better interpretation in the latent space of GANs such as PGGAN (Karras et al., 2018) and StyleGAN (Karras et al., 2019) or inputs with higher resolution may fix this problem. Such model can also be trained on a larger dataset with more attribute words but more noise. Besides, some text to image (Zhang et al., 2018) models shows that joint text embedding can be used to generate continuous latent space. Joint embedding can be used to learn more complicated patterns in the text. Furthermore, I am curious of the performance of the model on a more detailed word attributes such as ”wearing necklace” if I am able to construct a better encoder.

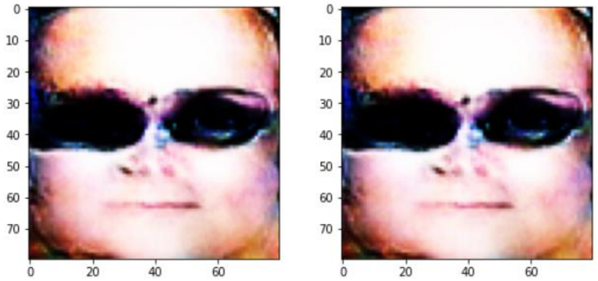

Figure 7. The comparison of images of the projections of similar words. Left: eyeglasses(trained), Right: eyeglass(untrained)

References

Arvanitidis, G., Hansen, L. K., and Hauberg, S. Latent space oddity: on the curvature of deep generative models, 2018.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis, 2019.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks, 2014.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation, 2018.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks, 2019.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation., 2014.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

Shen, Y., Gu, J., Tang, X., and Zhou, B. Interpreting the latent space of gans for semantic face editing, 2020.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. Stackgan++: Realistic image synthesis with stacked generative adversarial networks, 2018.

Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016. ISSN 1558-2361. doi: 10.1109/lsp.2016.2603342. URL http://dx.doi.org/10.1109/LSP.2016.2603342.