Deep Feature Consistent Variational Autoencoder

Gony Idan and Hila Malka

March 2022

Abstract

In this report we summarize the findings of the article "Deep Feature Consistent Variational Autoencoder" [1] and present the results obtained using the method described in that study.

1 Introduction

The main novel result of the paper is an improved general Variational Autoencoder (VAE). The enhancement that yields better results is attained by changing the loss function that the model works with, while using Convolutional Neural Networks (CNN). Before we address the paper and it's findings, we review some backgrounds concepts and terminologies.

Convolutional Neural Networks

CNN is a type of neural network model which allows us to extract better representations for image content. Unlike the classical image recognition where we define the image features, CNN takes the raw pixel data of an image and extracts the features automatically for better classification. It can successfully infer the spatial and temporal dependencies in an image through the application of relevant filters.

When a CNN processes an image, each convolution layer breaks the image down to kernels. Each kernel is studied in order to detect the features in it. As the images goes through the convolution layers, more complex features are detected, such as objects, eyes, nose, etc. Also, to each convolution layers we can add padding layers, which adds pixels to the frame of the image when it's being processed by the kernel and by that extends the area of which the CNN processes the image and reaches a more accurate analysis of the image. In this study Replication Padding was used which copes the input being padded, reverses it and then applies it in the image. Overall, CNN learns an image by studying it's features, meaning it takes into account the spatial correlation and the perceptual loss.

Variational Autoencoder

VAE is used for encoding and decoding an image. Its first element is the Encoder that takes an image represented by x and compresses into a vector z of a smaller dimension, i.e., $z = Encoder(x) \sim q(z|x)$. The second element is the Decoder. It takes the vector z and reconstructs an image $\hat{x}$. The goal of the VAE is to obtain an image $\hat{x}$ that is as close as possible to x.

For a given set of possible encoders and decoders, we are looking for the pair that keeps the maximum ~~of~~ information when encoding and also has the minimum of reconstruction error when decoding. The reconstructed loss is defined as the minimum of the negative log likelihood, $\mathcal{L}_{rec} = -\mathbb{E}_{q(z|x)}[log(p(x|z))]$. This implies that we want to maximize the likelihood of reconstructing the vector x given the latent vector z over all the possible latent vectors Z.

Another property of VAE is the ability to control the distribution of the latent vector z and make it independent in other parameters, using a normal distribution: $z \sim \mathcal{N}(0, 1)$. The distance between $q(z|x)$ and the standard normal distribution is quantified by KL-divergence. KL-divergence is used to measure the difference between two probability distributions over the same variable x. Minimizing KL-divergence indicates that we are getting closer to the original distribution of the latent vector.

VAE models can be trained by optimizing the sum of the reconstruction loss ($\mathcal{L}_{rec}$) and KL-divergence loss ($\mathcal{L}_{kl}$) by gradient descent. There is a trade-off in minimizing both arguments in the loss function. Minimizing only the KL-divergence will result images that look like a reconstructed image from a uniform distribution, causing all of them to look the same with lack of details. Minimizing only the reconstructed loss will yield the closest reconstruction to the original image x.

2 Challenges

VAE is an autoencoder whose encoding distribution is regularised during the training in order to ensure that its latent space has good properties allowing us to generate some new data. When VAE is used for encoding and decoding images, the common approach these days is to use a pixel-by-pixel loss function, like the $\mathcal{L}_2$ norm, while two given images are compared pixel-by-pixel. The problem with this approach is that it does not consider the spatial correlation and the perceptual loss of the two images being compared, which might cause that the images may seem similar to the human eye but receive a high loss. In this study, a new loss function is suggested in order to deal with said problem. In the next section we will dive in into the solution given in the paper.

3 Main Topic Summary

As stated earlier, the problem with the current loss function used for VAE is that it does not take into account the spatial correlation and the perceptual loss which cause the reconstructed images to be blurry. Also, as we have seen, CNN does consider these elements when passing images through its hidden layers. Hence, the main goal of this paper is to combine these strong elements of the CNN into the loss function of the VAE in order to have better comparison between the input image and its reconstructed image.

In order to improve the loss function and replace the pixel-by-pixel loss, the paper suggests using a new loss function: the feature perceptual loss. This method tries to find the consistency between two images by examining their outputs in the hidden layers of a pre-trained CNN. These hidden layers have the ability to capture important perceptual quality

features such as spatial correlation. Consequently, when comparing the representation of a hidden layer between two images, when there is a smaller difference it indicated that there is a better consistency of the spatial correlation. Minimizing the distance between representations of each hidden layer will cause the model to be more aware of the spatial correlation, unlike the pixel-by-pixel loss.

The perceptual loss will be calculated as follows: given a CNN model ϕ, for both input image and reconstructed image, we will calculate the output for the hidden layer l and compare between the representation received, denoted as $\phi(x)^l$ when x is a given image. We will be using simple squared euclidean distance to compare between the representation of the input image x and the representation of the reconstructed image $\bar{x}$ for the l hidden layer denoted as $\mathcal{L}^l_{rec}$. The complete perceptual loss will be $\mathcal{L}_{rec} = \sum \mathcal{L}^l_{rec}$.

In total, we combine the perceptual loss with the KL-divergence to yield the new loss for the VAE: $\mathcal{L}_{total} = \alpha\mathcal{L}_{kl} + \beta\mathcal{L}_{rec}$, where α and β are weights for the perceptual loss and the KL-divergence loss, respectively.

4 Experiments

The paper presents two kinds of experiments. One to generate images out of latent vector z, and the other to reconstruct images, along with exploring the semantic relationship between different latent representations. The model is trained on CelebFaces Attributes (CelebA) data set and using a pre-trained VGG19 model as the loss network ϕ to construct the feature perceptual loss for image reconstruction. Two different models were created with two combination of layers from VGG19 network used to calculate the perceptual loss: VAE-123 and VAE-345, while using Relu layers 1, 2, 3 and Relu layers 3, 4, 5, respectively.

4.1 Image Reconstruction Experiment

In order to compare the performance of the model, an additional generative models was trained: a classic VAE model, PVAE. In this experiment, the comparison was based on PVAE, VAE-123 and VAE-345 and their ability to reconstruct images. The results of the study showed that the VAE-123 and VAE-345 reconstructed clearer images than the PVAE VAE-123 was great in keeping the original colors of the input image, and create faces with clear face parts. The VAE-345 was better in keeping hair texture but changed the color sometimes.

4.2 Image Generation Experiment

In this experiment in order to compare the performance of the model, two additional generative models were trained: the same PVAE as in the previous experiment and the other is a Deep Convolutional Generative Adversarial Networks (DCGAN). The last has shown the ability to generate high quality images from noise vectors. In the experiment, fake face images generated from 100-dimension latent vectors $z \sim \mathcal{N}(0, 1)$. The models mentioned above showed the following results: As in the previous experiment, PVAE model create very blurry images, especially in face parts like nose and eyes, as expected. DCGAN model

generated clean and sharp face images containing clearer facial features, however it has the facial distortion problem and sometimes generates weird faces. Both VAE-123 and VAE-345 generated better results. VAE-123 managed to generate faces of different genders, ages with clear noses, eyes and teeth, which are better than VAE-345. However, VAE-345 is better at generating hair with different textures.

4.3 Linear Interpolation of Latent Vector and Facial Attribute Manipulation

The paper investigated the linear combination of two latent vectors and the images generated from the combination. The latent vectors were denoted as z_{left} and z_{right}. The interpolation is defined by linear transformation $z = (1-\alpha) \cdot z_{left} + \alpha \cdot z_{right}$, for different values of α. Then z is fed to the decoder network to generate new face images. The attribute manipulation is based on creating latent vector that is related to the attribute, and add it to another z vector. This way the model creates a new image containing the original image and adding the attribute. Both experiments showed smooth and natural changes examples. These results generate hope that we can use such a method in order to produce images based on features soon.

4.4 Facial Attribute Prediction

In this experiment the challenge was to predict each binary feature, from the 40 labeled features in the data set. This mission is very challenging, due to the quantity of features, and the fact that there is a large scale for every attribute (for example, small changes between different smiling faces). The method based on cropping face area, and fed the two models VAE-123 and VAE-345 to extract latent vectors. The latent vectors were used to train a standard Linear SVM classifiers, one classifier for each attribute. The results were significantly better for VAE-345 than VAE-123. There were some easy features, that all models had good performances to predict them like Eyeglasses, and some really difficult features to predict like Oval face.

5 Summary of the Paper

The paper presents various uses for VAE with Feature Perceptual Loss function. This method successfully manages to overcome the problems that were mentioned in the beginning, how in traditional methods of reconstruction images using VAE there is a sensitivity to small changes causing a blurry reconstructed image. The method is also useful to other tasks like identify images, changing images and predict features. We believe that future implements and developments using the perceptual loss are intriguing.

6 Practical - Implementation

We decided to implement the new VAE model that was presented in the paper. First let's review the architecture of the model.

Both the encoder and decoder of the VAE are implemented based on CNN. The encoder contains 4 convolution layers each with a 4x4 kernel, with a stride of 2. Between each convolution layer the data is passed through a batch normalization layer and a LeakyRelu activation function, to help stabilize the data. After the encoder, two fully connected output layers were added, that represent the mean and variance and will be used to sample a latent variable z. The decoder contains 4 convolution layers each with a 3x3 kernel and a stride of 2 and replication padding. Between each layer we will have a batch normalization layer and a LeakyRelu function, like the encoder. This architecture matches the one from the paper, with a slight difference where in the decoder where we used a stride of 2 instead of 1, because of technical issues.

The perceptual loss will be calculated using the hidden layers of a pre-trained VGG19 network that was trained on Image-Net data base.

We build the same architecture using Pytorch and for the pre-trained model we used the same pre-trained VGG19 model in Pytorch that was also trained on Image-Net. As stated in the Experiments section we created two models, one that when calculating the perceptual loss uses the Relu layers 1, 2, 3 from the pre-trained model and a second that uses the Relu layers 3, 4, 5. We also created a model that uses all the layers in the pre-trained model for calculating the perceptual loss, as stated it the paper. We also trained a regular VAE model for comparison. When calculating the loss for the VAE we used $\alpha = 1$ and $\beta = 0.5$, as used in the study.

We ran our models on Comic faces (paired, synthetic)[1] data set which contains images of faces and cartoon faces. We used the human faces which includes 10,000 images. We ran the models with 15 epochs and used Adam optimizer with learning rate of 0.0005. In the study the models were trained on CelebA data set which contains 200,000 images of faces. Each model trained for 5 epochs and used Adam optimizer with learning rate of 0.0005.

6.1 Results

Since we worked on a significantly smaller data set, as we had limited computational power and space we needed more epochs for the models to train efficiently. Also this can be the main explanation to why we did not receive such clear results as the study presented. Also after a few test runs, we received that all the reconstructed images looked the same. This result happened because the strong weight (α) the KL divergence has on the loss function, and as we stated in the beginning it causes all the reconstructed images to looks like they were taken from uniform distribution. In order to deal with this problem we decreased the weight of the KL divergence by factor of 0.001.

6.1.1 Reconstructing images

One of the main uses of VAE models is to reconstruct an image that passes through the encoder and the decoder and receive a similar result as possible to the original image. We tested our models ability to reconstruct images given and original input Fig. 1.

The study presents that the VAE-123 model works better on reconstruct facial parts and the VAE-345 is better on reconstruct textural features like hair. In our work we did

[1] https://www.kaggle.com/defileroff/comic-faces-paired-synthetic

Figure 1: The input images

not receive a significant difference between the two models, though they both worked well reconstructing the face features, like eyes, nose, mouth etc. Though we can see that the VAE-123 model is more blurry than the VAE-345, as is depicted in Figs. 2a and 2b.

We also saw that the VAE model that uses all the hidden layers in the pre-trained VGG19 model to calculate the perceptual loss reconstruct better images than all models we have trained, as is depicted in Fig. 2c. This result matches our expectations since it worked with all the layers in the VGG19 model. As stated at the beginning, each hidden layer is better at detecting more complex features. Therefore, using all these layers will result in comparing more complex features between the input and reconstructed images and receiving a more precise loss function and as explained in the study the model will be better at reconstructing images.

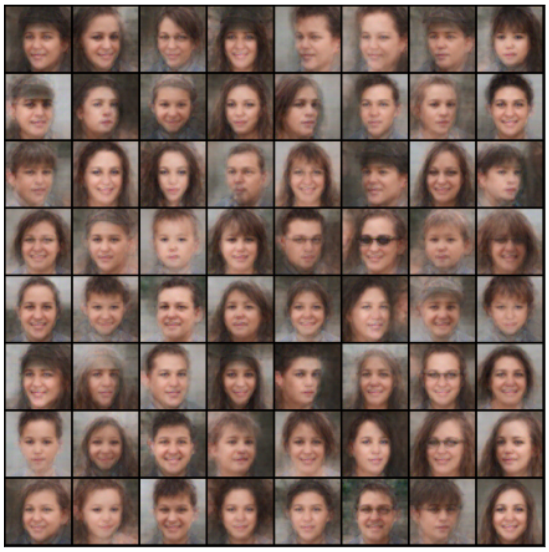
When comparing our models, VAE-123 VAE-345 VAE-all, to the original VAE model with the same architect, VAE-plain, which uses the original loss function we can see that the results are blurrier, as is depicted in Fig. 2d. This result matches the results of the study.

6.1.2 Generating Images

We also examined our models behavior in generating images. As mentioned in the previous part, one of the experiments presented in the paper was generating images from normally distributed vectors. The ability to generate images is a sought-after ability in many fields. In our implementation 8 vectors from the Normal distributions, $\mathcal{N}(0, 1)$ were drawn and fed to the decoders of the different models to generate new images. We used the same models as trained for reconstructing the images before, VAE-123, VAE-345, VAE-All and VAE-plain. It is clear to see that VAE-All generated images that had clearer facial features than the original VAE-Plain model, Fig. 3. The paper describe some difference for the generative models VAE-123 and VAE-345. VAE-123 was better in generating clear face parts, and VAE-345 was better in generating hair texture. Similar differences exist in our results, but not as clearly as in the paper, which again is caused mainly by the different size of training data.

(a) Reconstructed images using VAE-123 model

(b) Reconstructed images using VAE-345 model



(c) Reconstructed images using perceptual loss and all layers of the model

(d) Reconstructed images of PVAE model

Figure 2: Results of trained models in reconstructing the same images

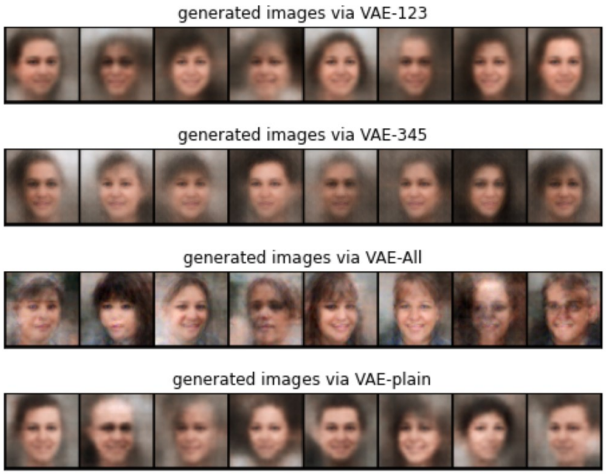


Figure 3: The generated images

References

[1] Xianxu Hou, Linlin Shen, Ke Sun, and Guoping Qiu. Deep feature consistent variational autoencoder. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, mar 2017.