Visual Recommender System with Adversarial Generator-Encoder Networks

Bowen Yao
Stanford University
450 Serra Mall, Stanford, CA 94305
boweny@stanford.edu

Yilin Chen
Stanford University
450 Serra Mall
yilinc2@stanford.edu

Mingxiang Chen
Stanford Unversity
450 Serra Mall
ming1993@stanford.edu

Abstract

A deep-learning-based visual recommender system was built in an unsupervised fashion. The system uses Adversarial Generative-Encoder Network[14] to learn embeddings for images and then K-nearest neighboring images of the query image in the embedding space is output as recommendation results. Such system will be most useful for E-commerce companies where visual recommendation can be used to alleviate cold start issue of common non-deep-learning-based recommender system.

1. Introduction

Convolution neural network has so far done a great job in image classification and recognition[9][11][12]. This technology has been introduced to many industrial areas, where the demand in computer vision is growing higher and higher, especially in E-commerce. Scientists and engineers used new technologies to assist customers with product recommendations. For instance, we can use *Mobile Taobao* or *Amazon Rekognition* to find the ideal product by uploading photos or pictures from Internet.

However, currently most such recommendation systems are built in a supervised way. In this project, We seek to build a deep-learning-based visual recommender system in an unsupervised fashion. Since in a real situation, adding labels to each single picture would be expensive and time-consuming. We will use Adversarial Generative-Encoder Network to achieve this goal, which can embed images into a simpler space by learning a pair of mappings between the data distribution and a given simple distribution. Given a query image, the system can then recommend to the user the top k images in the database that are closest to the query image in the simple embedding space.

2. Problem Statement

Using crawler technology to steal image data of retailers or from E-commerce websites is illegal, while we do not have access to these database, data from ImageNet and other standard deep learning dataset were used instead. For example, since there are more than 14 million images in the ImageNet dataset [1], we can simply assume that once we can do a satisfactory recommendation on ImageNet, we can also recommend pictures of high similarity on retailers' or other image datasets.

Our approach is also evaluated on ImageNet dataset and other standard deep learning dataset such as CelebA, SVHN, and CIFAR10. Since each image in these datasets is labeled to a specific class, we can then define a reasonable recommendation to be those recommended images that are in the same class as the input query image. Then we define the *recommendation precision* to be the proportion of reasonable recommendations within all recommended images output by the model. We will use *Recommendation precision* as our main metric for measuring the performance of our system. We will also compare our results with some baseline models such as simple K-Nearest-Neighbor in the data space.

3. Related Work

Many efforts have been made to resort to the great power of neural networks do visual recommendation in a supervised fashion. In [3], the author proposes a Siamese Network that uses two channels of identical convolutional neural networks to learn a similarity metric for images. The method was tested on 3 face recognition datasets and other 2 small datasets. The results are promising but the datasets are rather simple. [2] further implements the idea of [3] on a larger dataset. The dataset contains mainly room decoration and home design image and is carefully collected by the author by building a whole platform for human workers to mark two images to be similar. The data collected takes great effort of quality control and the article takes a large volume to describe how it is actually done. The results are promising but the whole process of collecting labeled data reveals the most obvious cons of supervised recommender system that it requires massive amount of hand labeled data.

Generative Adversarial Network(GAN)[7] is a powerful framework for estimating generative models via an adversarial process, in which a generative model G that captures the data distribution and a discriminative model D that estimates the probability that a sample came from the training data rather than G are trained adversarially. Works have been done to modify GAN so that an inverse mapping of G can be learned to map data into a simpler space. [5] and [4] are two papers published at about the same time. They both independently proposed a Bi-directional GAN structure that train an encoder E together with the G and D of common GAN. This model gives good generative and reconstruction quality of the output image, whereas the downside is that Bi-GAN turns out to tricky to train.

4. Technical Approach

We plan to use Adversarial Generator-Encoder Network (AGE Network)[14] to learn image feature representation in an unsupervised fashion. Such feature representation is useful for many downstream tasks such as finding similar images semantically and build a visual recommender system.

Adversarial Generator-Encoder Network (AGE Network) is appropriate for extracting features from images. Compared with other similar generative adversarial models, such as Adversarially-Learned Inference (ALI)[5] and Bi-directional Generative Adversarial Nets (BiGAN)[4], a promising feature for AGE Networks is that it doesn't have external discriminator. Instead of using discriminator to output a binary result representing whether a image is fake or not, the AGE network can take a batch of samples and thus it can compare the distribution of fake image and real image. The advantage of this feature is that it can address the mode collapse issue of regular GAN model (usually happens when generator learns to map several different input z values to the same output point[6]). And it has been shown that the feature embeddings learned by AGE Network is useful in downstream task. Therefore, we plan to use the representation of image in latent space learned by AGE Network to build a recommendation system in a semantic way.

In short, the overall AGE Network model includes a generator and an encoder, which define the mapping between a given distribution in latent space and the data distribution. The generator will try to generate images as indistinguishable from the real data as possible, while the encoder will try to distinguish them from real data. In other words, the generator try to make the distribution of real image and fake image in latent space as close as possible, while the encoder will try to make the two distribution as different as possible. During the "battle" between the generator and the encoder, the joint model gradually "learn" the optimal mappings between the given distribution and the data distribution. Figure 1 shows the overall architecture of the AGE network.

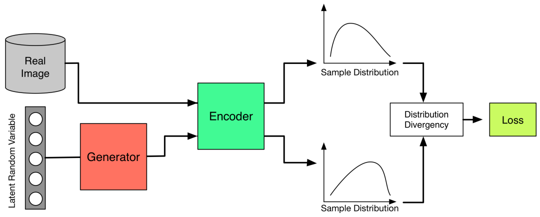


Figure 1: AGE Model Architecture

4.1. Distribution Divergence Loss

Formally speaking, the optimization problem for the encoder and generator in AGE Network is defined as the following:

$$\max_{e} \min_{g} V(g, e) \tag{1}$$

where e represents the mapping from image distribution to latent space defined by encoder, g represents the mapping from latent space to image distribution defined by generator, and V is the distribution divergence. It should be emphasized that the divergence between the outcome distribution of encoder and generator is measured in latent space. In practice, we usually train the network adversarially by train each of them several steps in turn. That is, we can break down the overall optimization objective into the objective of encoder and generator respectively:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}(V(g_{\theta}, e_{\bar{\psi}})) \tag{2}$$

$$\hat{\psi} = \underset{\psi}{\operatorname{argmax}}(V(g_{\bar{\theta}}, e_{\psi})) \tag{3}$$

where θ and ψ denote the value of the encoder and generator parameters. Equation (1) corresponds to generator's objective to make two distribution indistinguishable, while equation (2) corresponds to encoder's objective to encode them as different as possible. As mentioned before, we only compare them in latent space, since the encoder e maps distributions X and g(Z) in the latent space to the distributions e(X) and e(g(Z)) in the latent space. Thus, we can rewrite the above objective into following formula:

$$V(g, e) = \Delta(e(g(Z))||e(X)) \tag{4}$$

Where Z is the given distribution in latent space, which usually serves as noise in latent space, X is the real data distribution, which depends on the dataset we training the AGE network, Δ is the distribution difference measure in latent space, and it has to be nonnegative and zero if and only if the distributions are identical.

However, it should be noticed that (4) requires to compare the two general form distributions, which is complicated if we are only given in the form of samples in practice. Therefore, we further claim that the divergence measure $V(g, e)$ is equivalent to their relative divergence difference with respect to a fixed distribution, That is,

$$V(g, e) = \Delta(e(g(Z))||Y) - \Delta(e(X)||Y) \tag{5}$$

where Y is a fixed distribution in the latent space. In our model, we keep Y fixed as standard Gaussian distribution. Remember that Δ has to be nonnegative and zero if and only if the distributions are identical, and thus we choose KL-divergence to measure the distribution difference in latent space. That is,

$$V(g, e) = KL(g||N(0; I)) - KL(e||N(0; I)) \tag{6}$$

In order to analytically compute the above divergence for a mini-batch of examples, we introduce a parametric estimator giving the distribution divergence as below:

$$KL(g||N(0; I)) \approx -\frac{M}{2} + \sum_{j=1}^{M} \frac{\mu_{g_j}^2 + \sigma_{g_j}^2}{2} - log(\mu_{g_j}) \tag{7}$$

where M is the dimension of encoding vector, and μ_{g_j}, σ_{g_j} are the first and second moment of the sample.

4.2. Reconstruction Loss

Although the above analysis can ensure that if we minimize the distribution divergence loss, the generator and encoder can represent the optimal mapping between data space and latent space. However, it doesn't necessarily entails reciprocity of the e and g mappings at the level of individual samples. That is, given a real image, the generated image according to its embedding from encoder might be very different from its original image. Therefore, we also add a term of reconstruction loss L in the loss function. Specifically,

$$L_X(g_\theta, e_\psi) = E_X||x - g_\theta(e_\psi(x))||^2 \tag{8}$$

$$L_Z(g_\theta, e_\psi) = E_Z||z - e_\psi(e_\theta(z))||^2 \tag{9}$$

As shown above, the reconstruction loss can be either measured in latent space or data space. The reconstruction loss in data space is the traditional loss used within auto-encoders. In experiments, however, we found that measuring the reconstruction loss in latent space for generator can help avoid possible blurring issues for reconstruction. Therefore, the overall optimization problem is finally defined as:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}(V_2(g_\theta, e_{\bar{\psi}}) + \lambda L_Z(g_\theta, e_{\bar{\psi}})) \tag{10}$$

$$\hat{\psi} = \underset{\psi}{\operatorname{argmax}}(V_2(g_{\bar{\theta}}, e_\psi) - \mu L_X(g_{\bar{\theta}}, e_\psi)) \tag{11}$$

where λ and μ are hyperparameters to balance the reconstruction loss and divergence loss. The prove of equation (3) - (4) are given in the paper [14].

4.3. Architecture

The architecture of our encoder and generator follow the same structure as DCGAN[10], except that the output of the encoder is vector of length 128 instead of a single number. Besides, we found that the architecture of AGE network is very sensitive to image size, and thus we build two AGE networks for image size of 32 × 32 and 64 × 64 respectively, the detailed architectures are shown in Figure 2. We train the AGE Network based on the loss function defined above.

4.4. Making Recommendation

After training the AGE network, a KNN recommender is built using the features extracted from the AGE network. More specifically, we embed both all images in our dataset as well the query image using the encoder of the AGE network, then the K-nearest images in the dataset to the query image are output as the model's recommendation, where the distance measure is the $L2$ distance in the extracted feature space.

Since the images in the dataset are pre-labeled, we can use the labels to evaluate the performance of this KNN recommender. we can define a reasonable recommendation to be those recommended images that are in the same class as the input query image. Then we define the *recommendation precision* to be the proportion of reasonable recommendations within all recommended images output by the model. We will use *Recommendation precision* as our main metric for measuring the performance of our system.

5. Experiment

5.1. Data

The AGE model was trained on 4 different datasets (SVHN, CIFAR10, CelebA, and Tiny ImageNet). The Street View House Numbers (SVHN) Dataset has 70,000 elements in the training set, 10,000 in the validation set and 16,000 in the test set. CIFAR-10 is an established computer-vision dataset used for object recognition containing 10 classes. It has 40,000 images in the training set, 10,000 in the validation set, and 10,000 in the test set. CelebFaces Attributes Dataset (CelebA) is a large-scale face attributes dataset containing 9,000 faces in the training set, and 500 each in the validation set and the test set. ImageNet is an image database organized according to the WordNet hierarchy, while the tiny Imagenet is a smaller one provided in the

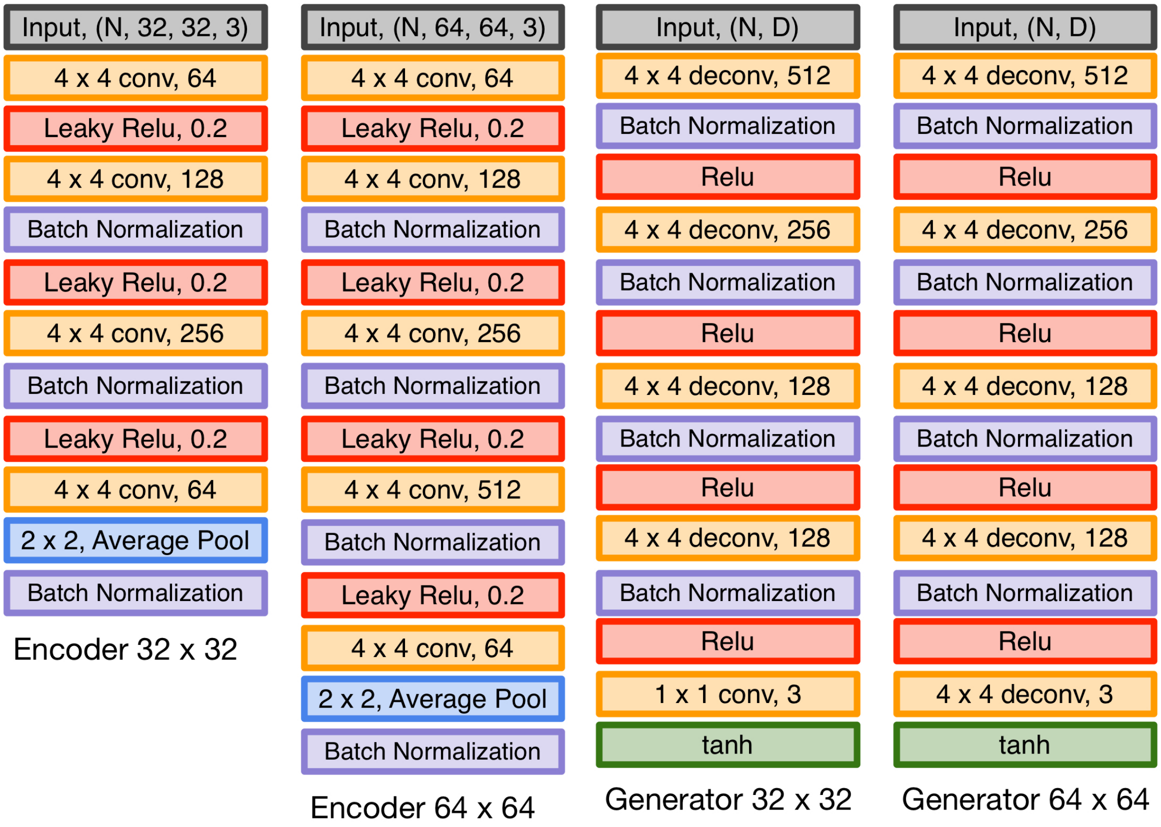


Figure 2: Encoder and Generator Architecture

default course project for Stanford CS231N. It has 100,000 pictures in the training set, 10,000 each in the validation set and the test set.

5.2. Training Detail

We train the AGE network on all the 4 datasets above. For each dataset, we do hyper parameter tuning on λ from the set of (500, 1000, 2000) and on μ from the set of (10, 50). We choose the pair that gives the best generating quality. We monitor the training process by checking the element-wise mean and variance of the $e(X)$, which, if functioning properly, should give 0 and $dim(Z)$ respectively by the property of uniform distribution on the unit sphere[14].

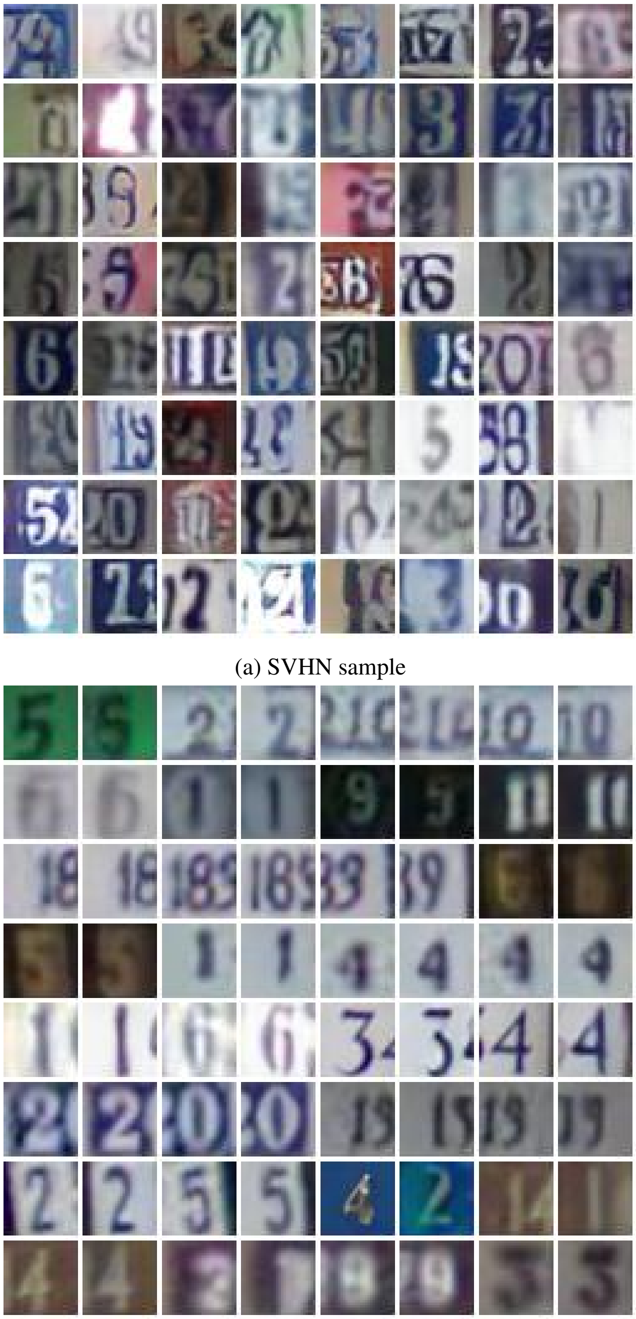
5.3. Qualitative Result

We now show the generated image (sample) $g(Z)$ and reconstruction $g(e(X))$ for all 4 datasets. Figure 3456 show results for SVHN, CelebA, Cifar10 and Tiny ImageNet, respectively. As we can see, the AGE network can give high quality samples and reconstructions, indicating the model is trained well.

5.4. Quantitative Result

5.4.1 KNN Recommendation Results

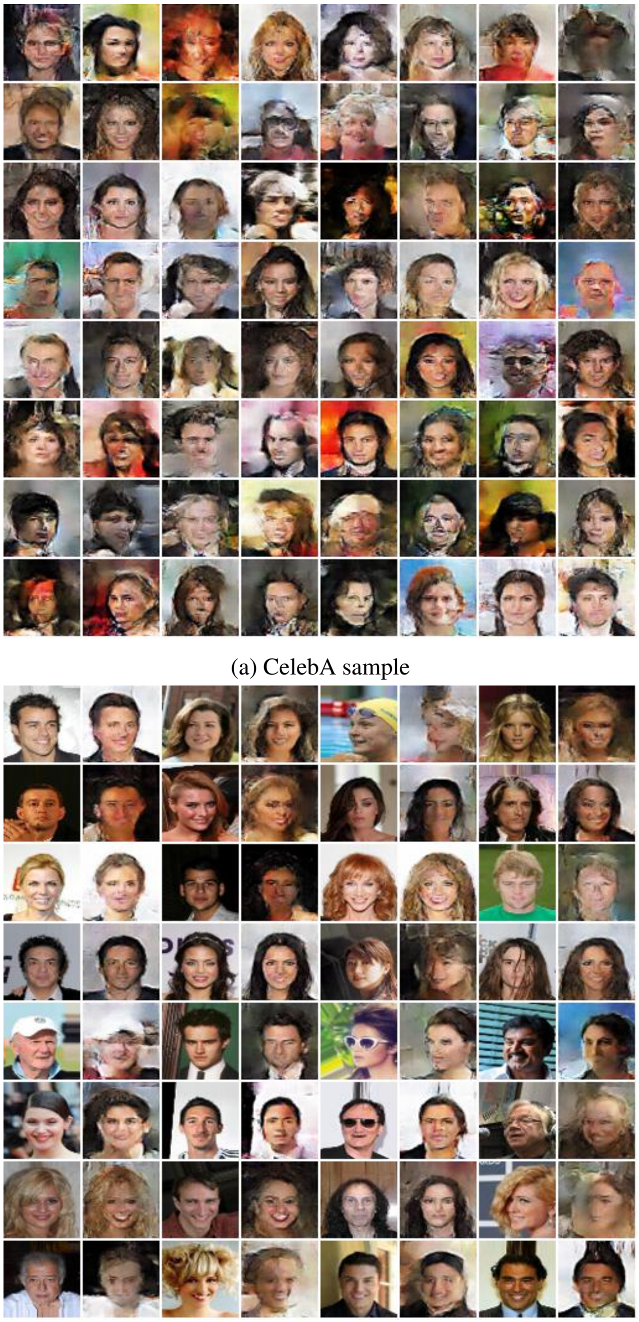
As mentioned in the previous section, a KNN classifier is used for the task of recommendation on the SVHN dataset. However the result is not satisfying. In this model, using the embeddings calculated from the last three hidden layers as well as the output layer of the encoder, with K equals to 3, the *Recommendation precision* is 43.2%, but the validation *Recommendation precision* is only 16.2%. This result is surprising to us at first, so we further train an SVM classifier to evaluate if the feature itself is useful.



(a) SVHN sample

(b) SVHN reconstruction

Figure 3: AGE model's generation and reconstruction for SVHN dataset



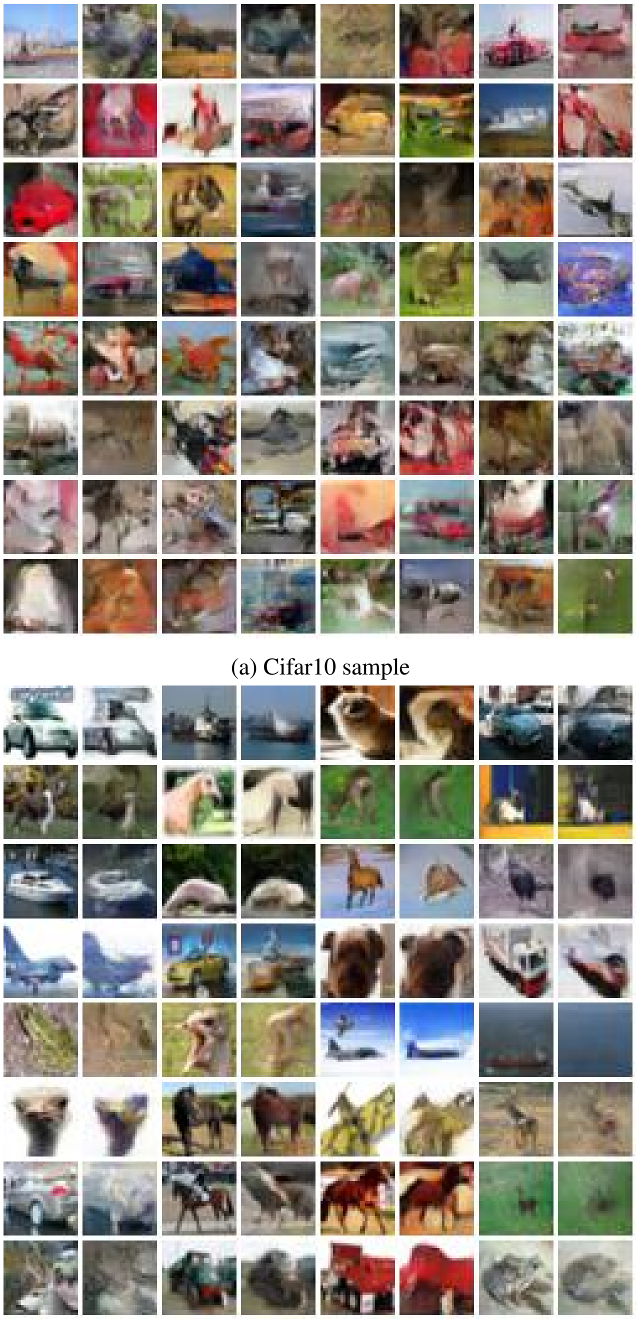
(a) CelebA sample

(b) CelebA reconstruction

Figure 4: AGE model's generation and reconstruction for CelebA dataset

5.4.2 SVM results

The input of the SVM is a matrix with the shape of $N * 4416$, where 4416 is the result of stacking the embeddings of the last three hidden layers as well as the output layer of the encoder. The training accuracy is 23.54%, and the validation accuracy is 19.46%. In comparison, a same SVM classifier is trained on the raw pixel features, which got a training accuracy of 46.49%, and a validation accuracy of 12.79%. We hence conclude that the embeddings learned using the AGE network is indeed informative in that it out performs raw pixel. However, the accuracy is still too low

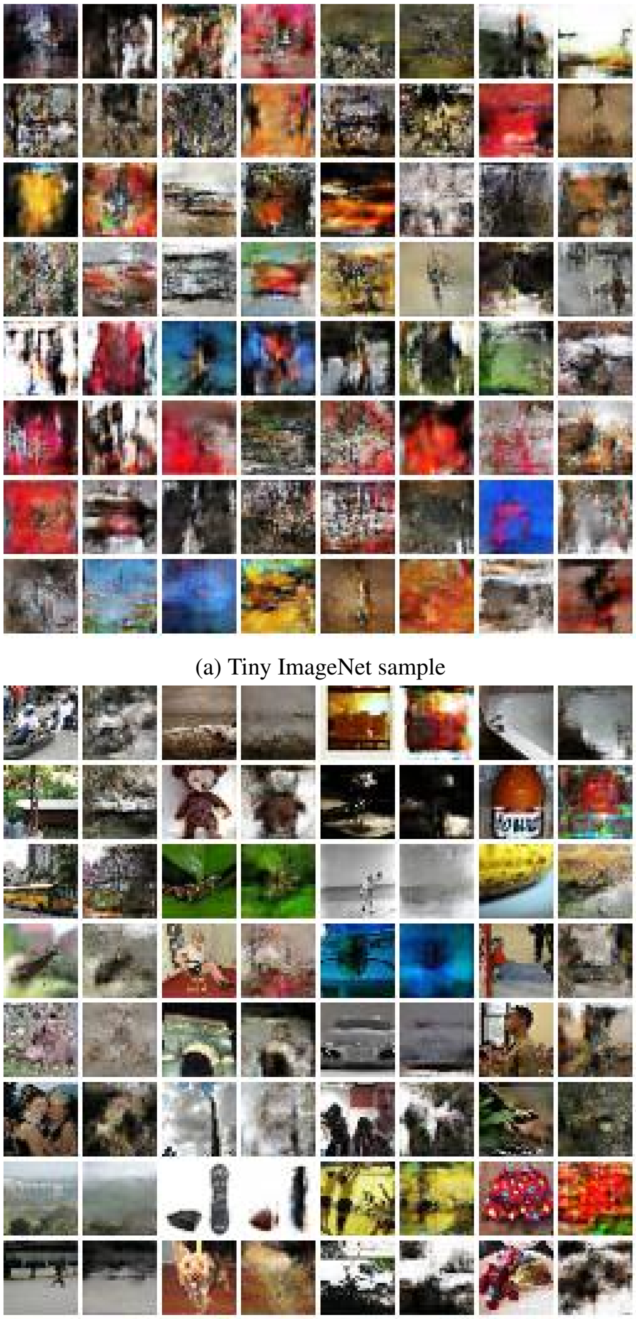


(a) Cifar10 sample

(b) Cifar10 reconstruction

Figure 5: AGE model's generation and reconstruction for Cifar10 dataset

to build a good recommender system on top of it. This is probably because of the fact unsupervised learning lack the semantic information incorporated in the labels of the data and hence is unable to give high quality embeddings.



(a) Tiny ImageNet sample

(b) Tiny ImageNet reconstruction

Figure 6: AGE model's generation and reconstruction for Tiny ImageNet dataset

6. Conclusion and Future Work

We train a AGE network in order to build a visual recommender system in an unsupervised fashion. Our AGE network is able to generate and reconstruct good quality images across various datasets. The embeddings drawn from the AGE network are better than raw pixels on the downstream classification task, however they are not good enough to build a recommender system on top of it. This is probably because that the AGE network is trained completely unsupervised and lack semantic information encoded in the labels to give high quality embeddings.

As for future work, we will consider modifying the AGE network to incorporate some label information to boost the model's performance.

Acknowledgement

We thank Dmitry Ulyanov who describes the base architecture of AGE network[14] and publishes the source code[13], and Justin Johnson for providing the code to load tiny image-net data[8]. We would also like to show our gratitude to Ben Poole for mentoring our project.

References

[1] About ImageNet summary and statistics (updated on april 30, 2010). http://image-net.org/about-stats. Accessed: 2017-05-16.

[2] S. Bell and K. Bala. Learning visual similarity for product design with convolutional neural networks. *ACM Transactions on Graphics (TOG)*, 34(4):98, 2015.

[3] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

[4] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[5] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[6] I. Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[8] J. Johnson. load tiny-imagenet data code. https://github.com/jcjohnson/tiny-imagenet, 2016. Accessed: 2017-04-28.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[13] D. Ulyanov. Age network source code. https://github.com/DmitryUlyanov/AGE, 2016. Accessed: 2017-04-20.

[14] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Adversarial generator-encoder networks. *arXiv preprint arXiv:1704.02304*, 2017.