TVAE: Deep Metric Learning Approach for Variational Autoencoder

Haque Ishfaq
Department of Statistics
Stanford University
hmishfaq@stanford.edu

Ruishan Liu
Electrical Engineering
Stanford University
ruishan@stanford.edu

Abstract

Deep metric learning has been demonstrated to be highly effective in learning semantic representation and encoding data information. People are able to do similarity measurement for data, based on the embedding learned from metric learning. At the same time, variational autoencoder (VAE) has widely been used to approximate inference and proved to have a good performance for directed probabilistic models. However, for traditional VAE, the data label or feature information are intractable. Similarly, traditional representation learning approaches fail to represent many salient aspects of the data. To this end, in this project, we propose a novel structure to learn latent embedding in VAE by incorporating deep metric learning. The features are learned by a triplet loss on the mean vectors of VAE in conjunction with reconstruction loss of VAE. This approach, which we call Triplet based Variational Autoencoder (TVAE), allows us to capture more fine-grained information in the embedding. Our model is first tested on MNIST data set. A high triplet accuracy of around 95.60% *is achieved while the VAE is found to perform well at the same time. We further implement our structure on Zappos50k shoe dataset [32] to show the efficacy of our method.*

1. Introduction

Learning semantic similarity between pairs of images is a core part of visual competence and learning. Functions such as Euclidean distances, Mahalanobis distance, cosine similarity are commonly used for measuring similarity distances. When applied on raw complex input datasets directly, these functions usually provide poor measure of similarity. But if applied on proper embedding of the input data, these functions result in superior metric for similarity measurement and reduces many learning problems to simpler problems. For example, given a proper image embedding and similarity measurement function, an image classification task would simply reduce to a generic nearest neighbor problem. Traditionally, such image embeddings were learned as a part of larger classification task. But this approach has various practical limitations for several scenarios. In extreme classification problems [8, 2] where the number of possible categories is very large or possibly unknown, conventional classification learning approaches are essentially useless since the availability of training examples for each class becomes scarce, if not totally unavailable. Hence, a new line of approach, namely metric learning [27, 23, 13] has gained much popularity for its ability to learn image embedding directly using the concept of relative distances rather than relying on specific category information. This way, it is able to learn a metric space where nearest neighbor based methods would naturally give superior performance due to the higher quality representation of input images in the learned embedding space.

On the other hand, Variational autoencoder has attracted much attention recently because of its ability to do efficient inference. A probabilistic model is learned with latent variables [17, 25]. VAE is considered as a powerful method in unsupervised learning, which is highly expressive with its stochastic variables. Recent advance in deep neural work has enabled VAE to achieve desirable performance. Despite its ability in model expression, the latent embedding space learned in VAE lacks many salient aspects of the original data.

In this project, we designed a new architecture motivated from metric learning and VAE, which is capable of two tasks at the same time - learning image representations with fine-grained information and doing stochastic inference. As a proof of concept, we first implement our idea on the MNIST data [3]. Along with VAE and metric loss of desireable order, we achieve about 95.60% metric accuracy. We further implement our structure on Zappos50k shoe dataset [32] to show the efficacy of our method.

2. Related Work

2.1. Variational Autoencoder (VAE)

A VAE consists of two networks. The first one, an encoder network, allows us to encode an image x to a latent

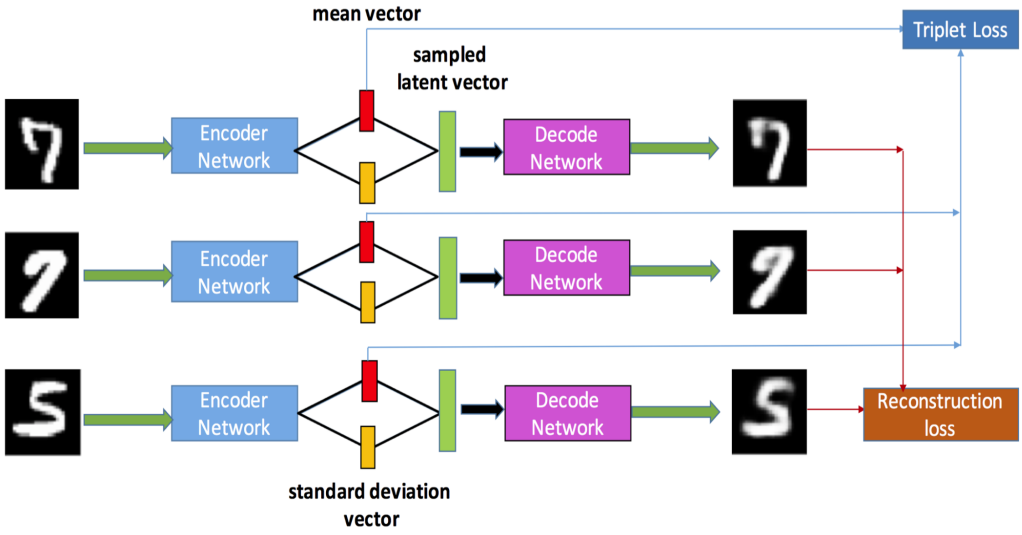


Figure 1: Model overview. As input a triplet of digit images (7,7,5) is given to three identical encoder networks. The mean latent vectors of three input images are used to calculate the triplet loss and the reconstructed images by the identical decoders are used to calculate the reconstruction error.

vector $z = Encoder(x) \sim q(z|x)$. The second one, a decoder network is used to decode the latent vector z back to an image $\bar{x} = Decoder(z) \sim p(x|z)$. To regularize the encoder, the VAE imposes a prior over the latent distribution $p(z)$. Usually the prior is set to independent unit Gaussian distribution. The VAE loss consists of two parts: the reconstruction loss and the KL Divergence loss. The reconstruction loss $\mathcal{L}_{rec}$ is the negative expected log-likelihood of the observations in x. And the KL-Divergence loss $\mathcal{L}_{KL}$ characterizes the distance between the distribution $q(z|x)$ and the unit Gaussian distribution. VAE models are trained by optimizing the sum of the reconstruction loss and the KL divergence loss using gradient descent.

$$\mathcal{L}_{vae} = \mathcal{L}_{rec} + \mathcal{L}_{KL}, \tag{1}$$

where

$$\mathcal{L}_{KL} = \mathrm{KL}[q(z|x)||p(z)] \tag{2}$$

$$\mathcal{L}_{rec} = -\mathbb{E}_{q(z|x)}[\log p(x|z)] \tag{3}$$

VAE has been widely used in recent unsupervised learning researches as a highly expressive model. Different approaches are used to generate the variational distributions, such as Gaussian processes [29], importance weighted approach [5] and a combination with auxiliary generative models [21]. Much progress has been achieved in recent research to increase the expression ability in VAE and enhance the learning process, including approaches like extension to semi-supervised learning [16], introducing multi-scale structural-similarity score [26], adding novel regularization [19], implementing deep feature consistency [11], and combining VAE with generative adversarial network [24].

2.2. Deep Metric Learning

Siamese networks for signature verification [4] first showed the possible usage of neural network for compact embedding learning. Recent work on metric learning using neural network are heavily based on various CNN architectures that are trained using triplet [23, 10], pairwise [7, 9] or quadruplet constraints [6]. In this approach, CNN is trained to learn an embedding for images that would capture the semantic similarity among images. Apart from triplet and pairwise loss, there are also approaches that use quadratic loss [13] and lifted structured embeddings [23]. Deep metric learning approaches have recently been used in various vision related problems such as face recognition and verification [27], style matching, image retrieval [31] and product design [1].

3. Methods

In this section we first provide background on triplet loss and then we introduce our method for combining VAE and triplet-based metric learning along with feature perceptual loss. Our proposed hybrid model is motivated as a way to improve VAE, so that it can learn latent representation enriched with more fine-grained information.

3.1. Triplet-based Variational Autoencoder.

The triplet based variational autoencoder framework is illustrated in Fig. 1. In each iteration of training, the input

triplet (x, x_p, x_n) is sampled from the training set in such a way that the anchor x is more similar to the positive x_p than the negative x_n. Then the triplet of three images are fed into encoder network simultaneously to get their mean latent embedding $f(x)$, $f(x_p)$ and $f(x_n)$. We then define a loss function $\mathcal{L}_{triplet}(\cdot)$ over triplets to model the similarity structure over the images. We use triplet loss same as the one described in Wang et al. [31]. The triplet loss can be expressed as

$$\mathcal{L}_{triplet}(x_a, x_p, x_n) = \max\{0, D(x_a, x_p) - D(x_a, x_n) + m\}, \quad (4)$$

where $D(x_i, x_j) = ||f(x_i) - f(x_j)||_2$ is the Euclidean distance between the mean latent vector of images x_i and x_j. Here m is a hyper-parameter that controls the distance margin in the latent embedding. This triplet loss function will produce a non-zero penalty of $D(x_a, x_p) - D(x_a, x_n) + m$ if the Euclidean distance between x_a and x_n is not more than the Euclidean distance between x_a and x_p plus margin m in the latent space.

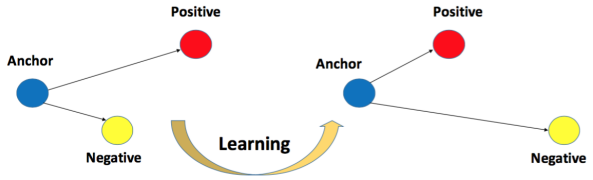


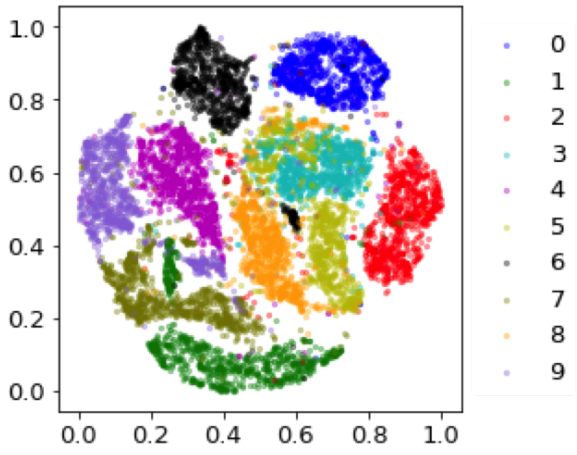
Figure 2: The **Triplet Loss** encourages to minimize the distance between the *anchor* and the *positive* maximize the distance between the *anchor* and the *negative*.

While various sampling strategies are proposed, such as random sampling, hard mining and semi-hard mining [27] for training triplet loss based deep metric learning models, in our case such approach would alter the real distribution of the data and would negatively affect the training of VAE. Thus, in our project, we used random sampling for constructing training triplets. We performed this by first randomly sampling an anchor and a positive image from a class and then randomly sampling a negative image from a different class.

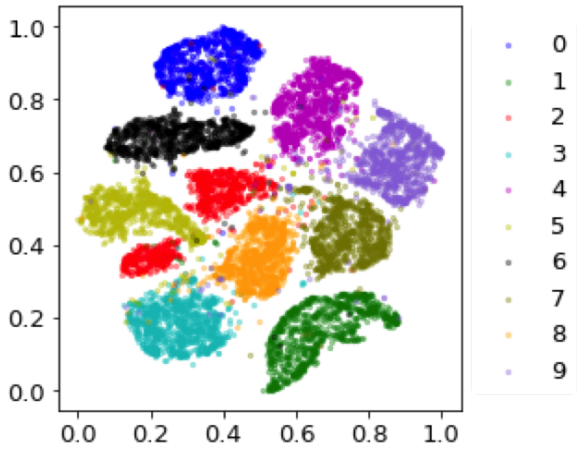
To ensure that we have a good representation of every class in the set of triplets, we construct an equal number of triplets per class, with an image from the class being the anchor point.

3.2. Variational Autoencoder Network Architecture

We construct both our encoder and decoder based on deep CNN like VGGNet [28] and AlexNet [18]. In the encoder network, we use 4 convolutional layers with 4 x 4 kernels a fixed stride of 2. We use a batch normalization layer and a LeakyReLU activation layer after each convolutional



(a) Plain VAE



(b) Triplet based VAE

Figure 3: t-SNE projection for the latent mean vector for the MNIST dataset.

layer. Finally, two fully-connected output layers are added to the encoder: one for mean and the other for variance of the latent embedding. As explained in [17], the mean and the variance is then used to compute the KL divergence loss and sample latent embedding z. In the case of decoder, we use 4 convolutional layers with 3 x 3 kernels and fixed stride of 1. For upsampling, we use nearest neighbor method by a scale of 2. Like in encoder, we use batch normalization layer and LeakyReLU activation layer after each convolution layer.

3.3. Perceptual Loss with Triplet Loss

Based on ideas in [14], [12], we replace pixel-to-pixel VAE reconstruction loss with a feature perceptual loss. In this case, we input both the original image x and the reconstructed image $\bar{x}$ through pre-trained 16-layer VGG net-

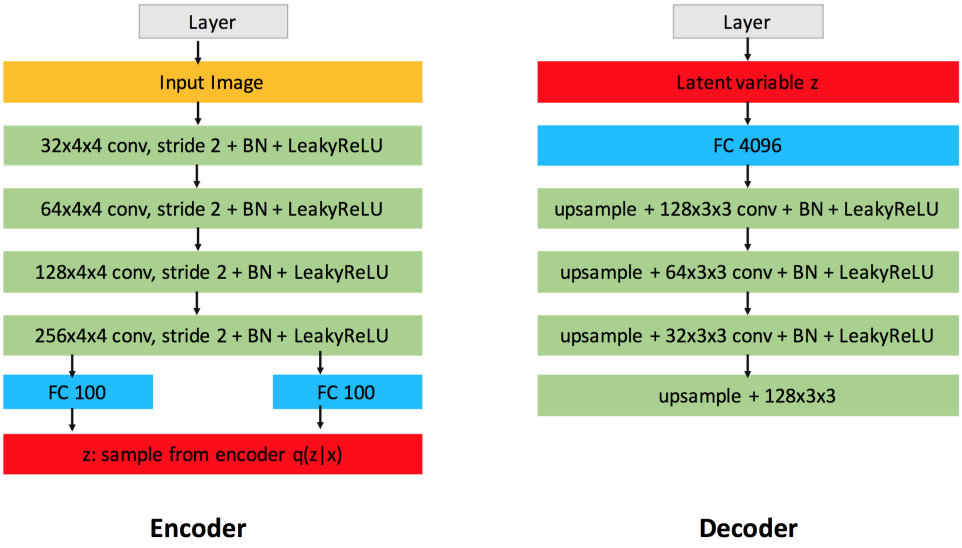


Figure 4: Autoencoder network architecture.

work [28]. We then use the squared Euclidean distance between the output of the l^{th} layer of the VGG net as the VAE loss. This loss function is less sensitive to actual pixel value compared to pixel-to-pixel $L2$ loss.

Thus our final loss function for an input triplet is given by:

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{KL} + \beta \sum_{i}^{l} \mathcal{L}_{total}^{l} + \gamma \mathcal{L}_{triplet} \qquad (5)$$

4. Experiments

We focus our experiments on preservation of the semantic structure in the learned latent embedding and image generation ability compared to traditional VAE.

4.1. Datasets

For our project, we perform experiments on two different datasets. First for proof of concept, we use MNIST dataset which includes 10 classes of hand-written digits (60000 training images and 10000 test data) [20]. The individual images are of size 28 by 28 pixels.

We then test our proposed structure on a more complicated dataset - the Zappos50k shoe dataset [32]. The dataset contains 50000 images of individual shoes each one richly annotated. We resize the images from 136 by 102 pixels to 128 by 128 pixels. For the purpose of this project, we focus into shoe characteristics based on the height of the heels (numerical measurement from 0 to 5 inches). We also performed experiments based on other characteristics: the types of the shoes, the suggested gender of the shoes. But due to time constraint of the course and limited computing resource, we were not able to finish training models based on those characteristics and thus omit those results in this report.

4.2. Baseline model and Proposed Model

Plain VAE with Perceptual Loss: For our baseline, we trained a plain VAE without using any triplet loss. For reconstruction loss, we used perceptual loss as explained in section 3.3 in the case of Zappos dataset.

Triplet-based Variational Autoencoder: Our proposed architecture is illustrated in Fig. 1. The input images are first fed through an encoder network. Mean vector and standard deviation vector are learned. Then the two vectors are combined as a sampled latent vector and pass through a decoder network. For MNIST, we adopted a simple network structure with two fully connected layers as encoder and decoder and used pixel-to-pixel $L2$ distance loss function as reconstruction loss. The dimension of the mean vector and the are both 20. For the Zappos50k shoe dataset, we adopted a network as illustrated in Fig. 5.

4.3. Training Details

For the MNIST dataset, the model was trained for 10 epochs.

For the Zappos dataset, we train the model with a mini-batch size of 64 and optimize using ADAM [15] with learning rate 5E-5, $\beta_1 = 0.1$, $\beta_2 = 0.001$. For experiment using the Zappos dataset, we use latent embedding dimension of 100. In equation 5, we set $\alpha = 1$, $\beta = 0.5$ and $\gamma = 10$. In each mini-batch, we sample triplets uniformly. Each model

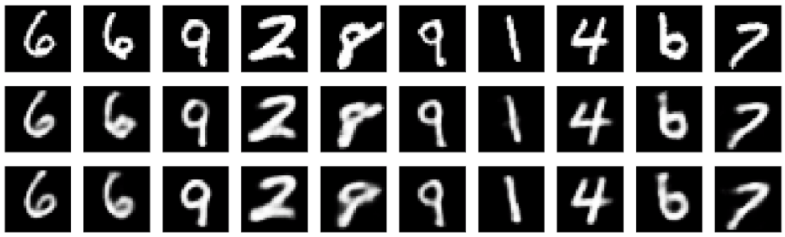


Figure 5: Comparison of reconstructed images from the MNIST dataset. The first row is the input images from the MNIST test set. The second row is the reconstructed images generates by the plain VAE. The third row is the reconstructed images generated by the TVAE.

is trained for 8 epochs, each epoch consisting 50,000 unique triplets. Even though for similar networks in relevant literature, it's suggested that the model should be trained for at least 100 epoch with unique triplet as much as possible (in the order of 100K), we had to limit ourselves to only 8 epochs with each having only 50K unique triplets, due to time constraint and limited computational resource. We then run the model snapshot with the best validation performance on the test set.

4.4. Visual Exploration of the Learned Latent Space

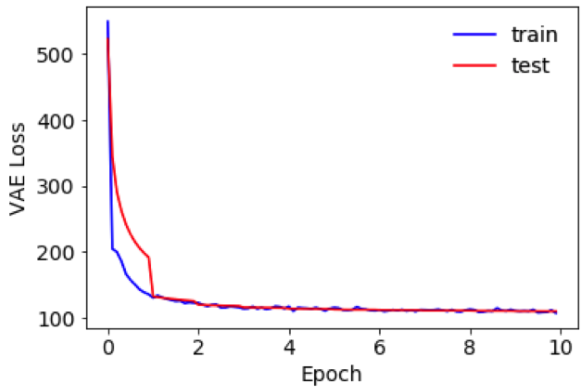
We visually explore the learned embedding distribution for the mean vector. Fig.3 shows the two dimensional projection of 20-dimensional learned latent embedding of the MNIST dataset. We used t-SNE [22] for creating the two dimensional projection image.Here we use different colors to stand for different digit classes. For instance, images with digit 1 are plotted in green color. With an additional triplet loss term, the clusters are more compactly clustered in the mean vectors, as shown in Fig. 3b. On the other hand, without the added triplet loss, the image clusters are less compact and seem to spreading out in the spatial space as seen in Fig. 3a. In this case, we also observe that images from one class are more likely to be divided into more than one cluster and images from different classes encounter more mixing issues.

4.5. Results on Triplet Prediction

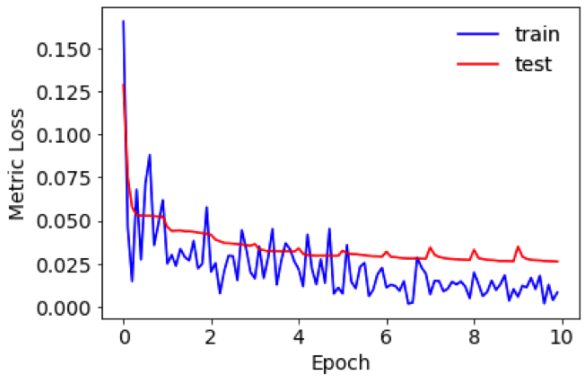
In order to evaluate the structure quality of the learned latent embedding, we analyze learned latent embedding of unseen triplets. We calculate triplet accuracy which is defined by the percentage of triplets that incur a loss of zero in Eq.4. First we train our model using training triplets. Then once the training is done, for each triplet x_a, x_p, x_n in the test set, we evaluate whether the distance between x_a and x_p is smaller than the distance between x_a and x_n by the distance margin. Clearly a random guessing would yield an error rate of 50% due to the task's binary nature.

From Fig. 6, we see that after 10 epochs of training on the MNIST data the network learning converges and the plain VAE learns latent space that obtains triplet accuracy of only 75.08%. On the other hand, triplet loss based VAE achieves 95.6% triplet accuracy. From Table. 1, we see that VAE loss for plain VAE and TVAE is in the same order. But on top of achieving lower VAE loss, the TVAE were able to learn latent vector space that would also preserve the relative distance metric.

In the case of, Zappos shoe dataset, we observe that the



(a) Triplet loss



(b) VAE loss

Figure 6: VAE loss and Triplet loss learning curve for TVAE with MNIST dataset.

Table 1: Triplet Accuracy and VAE Perceptual Loss

	Model	VAE Loss	Triplet Accuracy
MNIST	Plain VAE	104.66	75.08%
(10 epochs)	Triplet VAE	**110.34**	**95.6%**
Zappos	Plain VAE	0.5966	53.66%
(8 epochs)	Triplet VAE	**0.6204**	**73.8%**

TVAE starts with test triplet accuracy of around 50%, almost a random guessing, but with each epoch it steadily increases. By the 8^{th} epoch it achieves test triplet accuracy of around 75%. On the other hand, the test triplet accuracy in plain VAE always stays around 50% even after several epochs of learning. This validates that the plain VAE, while might be able to get latent embedding good enough for future reconstruction of the input image, it fails to capture salient features like relative distance metric in it's learned latent embedding. Whereas, in the case of TVAE, we are able to get almost same VAE reconstruction error compared to plain VAE while incorporating more salient features in the learned latent embedding. Here, we emphasize that the learning curve for Zappos dataset in Table 1, is based on less than 10 epochs of learning since we were constrained by time and computing resource within the time-line of our course CS231N. But based on other relevant literature, the training for triplet based models should be done for at least 50 epochs after which it starts to give much stronger result.

5. Conclusion and Future Work

Triplet based Variational Autoencoders (TVAEs) provide a new set of tools for learning latent embedding that leverage both traditional VAE and deep metric learning techniques. By incorporating triplet constraint in the learning process, TVAEs can learn interpretable latent representation that preserves semantic structure of the original dataset. Our method provides an initial framework for learning latent embedding that would be able to encode various notions of similarity. We demonstrate that TVAEs achieve perceptual reconstruction loss almost as same as the traditional VAE while encoding more semantic structural information in the latent embedding.

For future work, we would first like to run more experiments on larger datasets like Zappos and train CNN bases TVAEs for larger number of epochs which we were not able to do here due to time and computational resource constraints. We would also like to see how our approach performs for different notions of similarity in the dataset and explore further if we can incorporate ideas as in Conditional Similarity Network [30] in our framework.

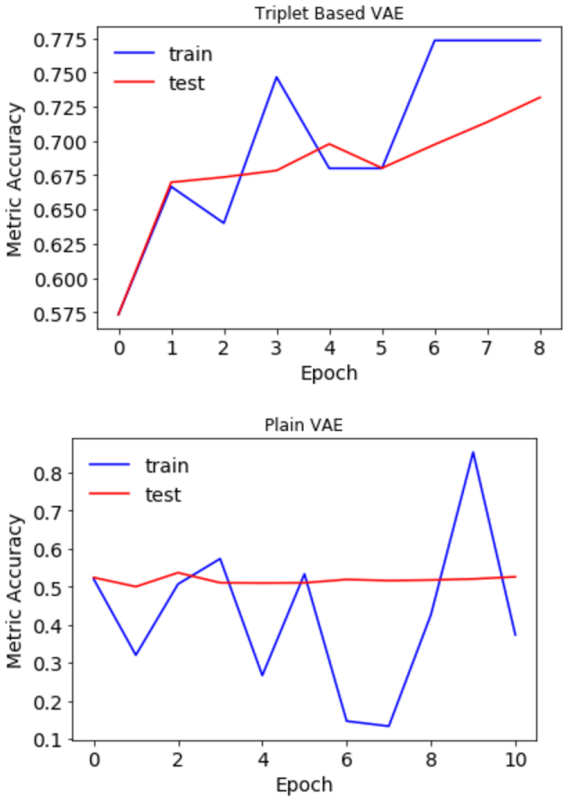


Figure 7: Triplet accuracy learning curve for TVAE and plain VAE with Zappos shoe dataset. We see that even within the first 8 epochs, the triplet accuracy for validation triplet set in TVAE steadily increases while in the case of plain VAE it always stays around 50% which is same as random guessing. We emphasize that this image is based on only 8 epochs.

6. Acknowledgements

We would like to thank Assaf Hoogi and Timon Dominik Ruban for insightful discussions, Google for providing Google Cloud credits to use GPUs for the experiments. We would also like to thank Andreas Veit for answering our questions regarding his paper Conditional Similarity Networks [30] and open-sourcing the code for the paper which helped us tremendously in setting up the initial environment for our project experiments.

References

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

[2] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In *Advances in Neural Information Processing Systems*, pages 163–171, 2010.

[3] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. D. Jackel, Y. LeCun, U. A. Muller, E. Sackinger, P. Simard, et al. Comparison of classifier methods: a case study in handwritten digit recognition. In *Pattern Recognition, 1994. Vol. 2-Conference B: Computer Vision & Image Processing., Proceedings of the 12th IAPR International. Conference on*, volume 2, pages 77–82. IEEE, 1994.

[4] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah. Signature verification using a "siamese" time delay neural network. *IJPRAI*, 7(4):669–688, 1993.

[5] Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

[6] W. Chen, X. Chen, J. Zhang, and K. Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. *arXiv preprint arXiv:1704.01719*, 2017.

[7] 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.

[8] A. Choromanska, A. Agarwal, and J. Langford. Extreme multi class classification. In *NIPS Workshop: eXtreme Classification, submitted*, 2013.

[9] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 1735–1742. IEEE, 2006.

[10] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.

[11] X. Hou, L. Shen, K. Sun, and G. Qiu. Deep feature consistent variational autoencoder. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 1133–1141. IEEE, 2017.

[12] X. Hou, L. Shen, K. Sun, and G. Qiu. Deep feature consistent variational autoencoder. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 1133–1141. IEEE, 2017.

[13] X. Huang and Y. Peng. Cross-modal deep metric learning with multi-task regularization. *arXiv preprint arXiv:1703.07026*, 2017.

[14] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[17] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] 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.

[19] A. Lamb, V. Dumoulin, and A. Courville. Discriminative regularization for generative models. *arXiv preprint arXiv:1602.03220*, 2016.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[21] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

[22] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[23] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4004–4012, 2016.

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

[25] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[26] K. Ridgeway, J. Snell, B. D. Roads, R. S. Zemel, and M. C. Mozer. Learning to generate images with perceptual similarity metrics. *arXiv preprint arXiv:1511.06409*, 2015.

[27] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

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

[29] D. Tran, R. Ranganath, and D. M. Blei. The variational gaussian process. *arXiv preprint arXiv:1511.06499*, 2015.

[30] A. Veit, S. Belongie, and T. Karaletsos. Conditional similarity networks. 2017.

[31] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014.

[32] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, 2014.