LEARNED DATA AUGMENTATION.

Chandramouli Rajagopalan
chanduiyer.raja@gmail.com

June 7, 2021

ABSTRACT

Dataset augmentation, the process of applying simple and complex transformations to help overcome the increasingly large requirements of Deep Learning Models, is a standard procedure in almost every supervised learning task. We try to approximate the set of transformations to which a particular label is invariant and improve the generalization of the VAE using data augmentation. To attain quicker results, we constrain our dataset to a small section of the ImageNet dataset (TinyImageNet), and CIFAR-10 and compare a set of data augmentation techniques. The potential benefit of such data augmentation is that it can include viewpoint changes, changes in scene lighting, and many nonlinear transformations. We propose a method to learn the data augmentations on the observations to obtain improved generalization. Finally, we validate the generalization of the trained VAE by classifying the data.

1 Introduction

Autoencoders are an unsupervised learning technique in which we utilize neural networks for the task of representation learning. Specifically, we'll design a neural network architecture such that we apply a bottleneck in the network which forces a compressed knowledge representation of the original input. A Variational autoencoder (VAE)[Kingma and Welling, 2014][Goodfellow et al., 2016] uses a probabilistic approach of representing an observation in the latent space. Instead of representing each latent observation into a single discrete value, VAE tries to illustrate each latent observation as a probability distribution. However, we try to understand if VAEs trained on augmented observations can help improve the generalization of the VAEs. VAEs trained on data augmented observations can include shifts in viewpoint, changes in scenic lighting, and many such nonlinear transformations in it.

Rather than working on large datasets, we try to limit ourselves to small and carefully examined datasets and augment them to improve the trained models' performance. The datasets we examine are the tiny-imagenet-200 data and CIFAR-10[Ho-Phuoc, 2019]. Tiny-imagenet-200[tin] consists of 100ktraining, 10k validation, and 10k test images of dimensions 64x64x3. There are a total of 500 images per class with 200 distinct classes. The CIFAR-10 dataset contains 60,000 32x32 color images in 10 different classes. There are 6,000 images of each class. To experiment with different algorithms, we limit our evaluation to these two datasets. Subsequently, we try to evaluate the performance by building a neural network classifier on top of the VAE to identify the class.

We train a VAE on the augmented data and learn the transformations on the data by decoding it back to the original observed space. Once we have a trained VAE on augmented data, we compare its performance with the VAE trained on the original observed space without any augmentation. To compare the performance, we add fully connected layers on top of the encoder for the classification task on the dataset. For the experiment above, we will measure the performance on the test set of the dataset. We then test the VAE learned on augmented data to approximate the set of transforms to which the label is invariant on the test set. Finally, we show the different transformations the VAE learned on augmented data can include.

2 Related Work

Dataset augmentation has been a standard procedure while training supervised models to reduce the overfitting of the training data. Dataset augmentation is an increasingly popular and highly demanded part as new data can be easily generated by applying image transformations such as cropping, scaling, shifting, rotating, and other affine transformations. Several augmentations can be applied without effectively changing the labels.

Denoising autoencoders[Im et al., 2016] take in partially corrupted data and are trained to recover the original undistorted input. When calculating the loss function, it compares the output values with the original input, not with the corrupted input. That way, the risk of learning the identity function instead of extracting features is eliminated.

Variational autoencoders are generative models that use a probabilistic approach of representing the observation in the latent space. Variational autoencoders consist of an encoder and a decoder. Instead of encoding the observation into a single value, VAE encodes as a distribution over the latent space.

Certain techniques[DeVries and Taylor, 2017] have used data augmentation in feature space that has shown promising results to improve generalization when there is limited data. These techniques add noise to the encoded data in the feature space and then decode it to generate new data.

We attempt to simulate a similar feature extracting method using a VAE, wherein the augmented data is fed as an input and then the decoded output values are compared with the original input to improve the generalization of the data.

3 Method

We propose two methods to learn the data augmentation and validate the generalization[Perez and Wang, 2017] of the VAE. In the first approach, we use augmented data as an input fed to the encoder of the network, and try to reconstruct the original observed data at the decoder's end. The motivation is to identify the augmented part of the image and attempt to remap the latent space to the original observed space. The second approach utilizes the already learned encoder part of the VAE and an additionally appended fully connected network. We validate the performance of the two different VAEs by classifying the CIFAR-10 test data and visualize the difference in generalization on the data.

3.1 Data Augmentation.

We mainly test our VAEs on three different kinds of augmentations. For each observation, we generate augmented images that are cropped, zoomed-in, rotated, and include a change in scenic lighting as well as a change in the color space of the image.

3.2 Learning the augmentation using VAE.

Since both the observations are almost similar and are taken from similar latent distribution, we train the VAE to learn the data augmentation on the observation and learn the higher-level representations. At training time, augmented data of 64x64x3 with random augmentations act as an input to the encoder, while the decoder tries to reconstruct the undistorted image.

3.3 Classification.

The learned VAE on augmented data is tested on the test data by appending a fully connected network on top of the encoder to identify the classes of the input labels. This helps us to understand whether the VAE trained on augmented data has better generalization than the VAE trained on undistorted images.

We use a small 2 layer fully connected network and a 5 layer CNN as an encoder of the VAE. We attempt to explore the potential benefits of learning the data augmentation and visualize the different transformations that VAE contains. Details about the architecture and the loss function will be presented in the experiments section.

4 Dataset and Features

There are primarily two different datasets that we work on. Initially, we use the CIFAR-10 dataset to learn the data augmentation and to validate the generalization of the VAE. Then, we use tiny-imagenet-200 dataset to identify the transformations to which the label is invariant. We don't classify our VAEs on tiny-imagenet-200 data due to a large number of classes and increasing experimentation time.

Our initial dataset is taken from CIFAR-10. It consists of 60k 32x32x3 color images of 10 different classes. The training set has 50k images and the rest 10k test set images are used for testing. The images are resized to 64x64x3 during preprocessing.

Lastly, our final data set is taken from tiny-imagenet-200. We take 100k images from all the classes. We then test it on the validation set of the tiny-imagenet-200 which contains 10k images to identify the set of transformations the label is invariant. The images are 64x64x3.

5 Experiments

Validation of the trained VAE, we run 5 experiments on different augmentations on the tiny-imagenet-data and a single augmentation on the CIFAR-10 to confirm the generalization of the VAE. We use Adam optimizer for both the classification task and learning the data augmentation task. We use a learning rate of 0.0001.

Encoder Architecture

- Conv with 32 channels and 3x3 filters. Batch normalizations. Relu activations.
- Conv with 64 channels and 3x3 filters. Batch normalizations. Relu activations.
- Conv with 128 channels and 3x3 filters. Batch normalizations. Relu activations.
- Conv with 256 channels and 3x3 filters. Batch normalizations. Relu activations.
- Conv with 512 channels and 3x3 filters. Batch normalizations. Relu activations.

Two parallel fully connected networks with an output latent dimension of 128 represent the mean and variance of the distribution. A similar architecture is used for the Decoder of the VAE as well.

Classification Network Architecture

- A fully connected network with output dimension 32. Batch normalizations. Tanh activations.
- A fully connected network with output dimension 10. Batch normalization. Relu activations.

We carry out the experiment in two different steps. In the first step, we train the VAE to learn the augmented images and then try to reconstruct the undistorted input observations. We test the trained VAE on undistorted images as well as augmented images to validate the transformations included in it. The experiments are performed using several augmentations such as random crop, random rotation, change in brightness and contrast, and change in hue and saturation values. In the second step, the parameters of the encoder are frozen and a fully connected network tries to classify based on the latent distribution of the encoded image. We then test the VAEs on the test data to compare the generalization. The variational autoencoder loss function is the negative log-likelihood with a regularizer. The loss function is the summation of two main parts, the reconstruction loss is the expected likelihood over the encoder's distribution over the observations, and the divergence loss measures the information loss while reconstructing. Classification loss is a multi-class cross-entropy loss on the scores produced by the classification network.

VAE loss formula

$$L = \sum_{i=1}^{N} l_i(\theta, \phi) = \sum_{i=1}^{N} \mathbb{E}_{Q_i(z_i)}[log p_\theta(x_i|z_i)] - D_{KL}(Q_i(z_i|x_i)||p(z_i)) \tag{1}$$

Cross entropy loss formula for classification

$$L = \sum_{i=1}^{N} y_{o,n} log(p_{o,n}) \tag{2}$$

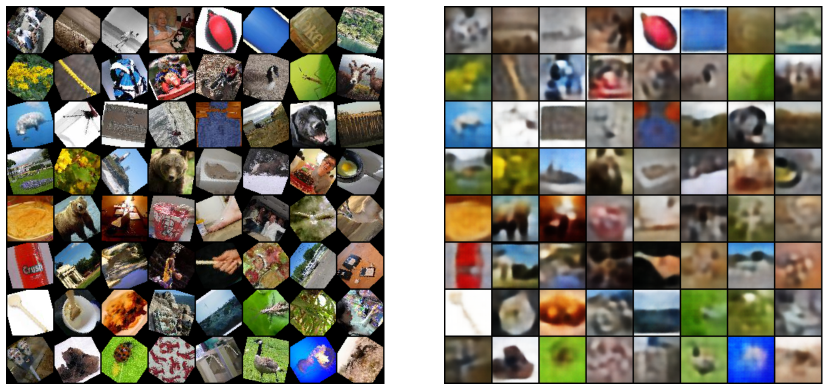
Pytorch on GPU (around 60 seconds per epoch) was used to train the models although they train reasonably fast on CPU as well. The classification experiment was run for 40 epochs whereas VAE was trained for around 100 epochs.

6 Results

VAE on learned data augmentation.When a set of images are fed into the learned VAE, it includes inverse of the transformations on which the VAE was trained.

6.1 Experiments on Tiny-Imagenet Dataset

6.1.1 Experiment with rotation angle augmentation

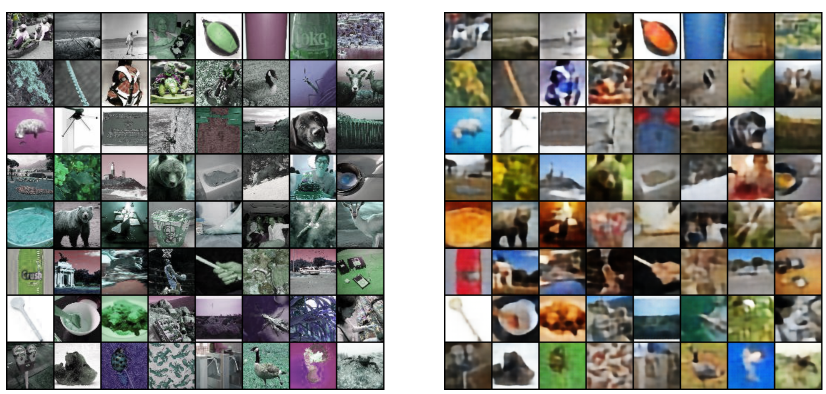


(a) Shows the observations rotated before inputting it to the VAE.

(b) Shows the observations rotated back to it's original undistorted state by the VAE.

Figure 1: VAE learned on augmented data rotates the input images.

6.1.2 Experiment with hue and saturation augmentation



(a) Shows the observations color changed before inputting it to the VAE.

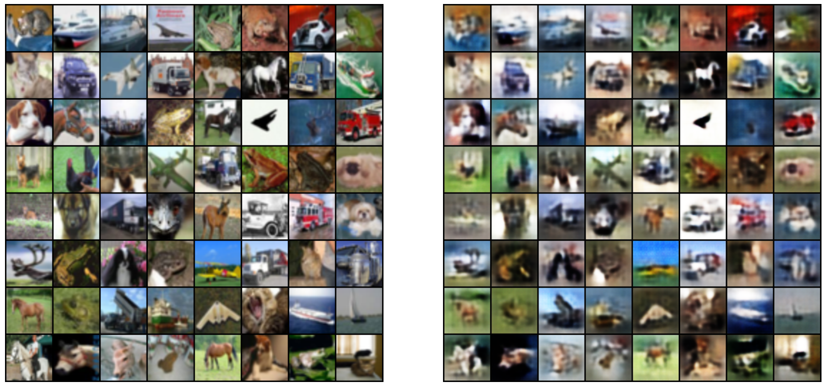
(b) Shows the observations transformed back to it's original undistorted state by the VAE.

Figure 2: VAE learned on augmented data changes the color of the input images.

The label is invariant to almost every transforms except for a change in hue and saturation, where the color change, changes the label. Color change intrinsically affects the label properties which changes the label. Geometric transforms that don't affect intrinsic properties of the label help in improvising the generalization of the VAE.

6.2 Experiments on CIFAR-10 Dataset

6.2.1 Experiment with center crop augmentation

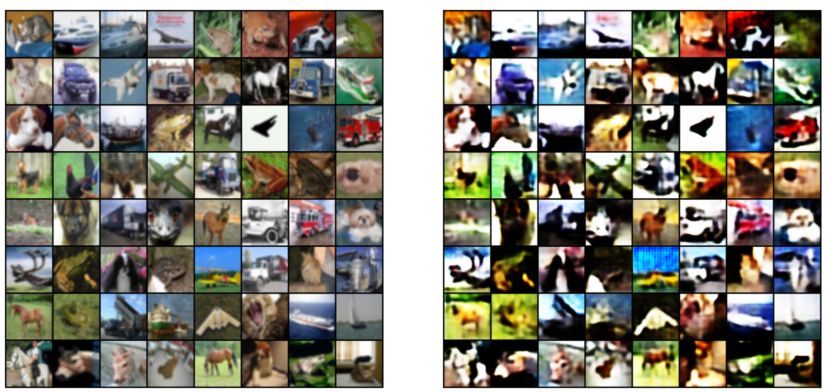


(a) Shows the undistorted observations before inputting it to the VAE.

(b) Shows the observations transformed with change in viewpoint by the VAE.

Figure 3: VAE learned on augmented data changes the viewpoint of the images.

6.2.2 Experiment with brightness and contrast augmentation



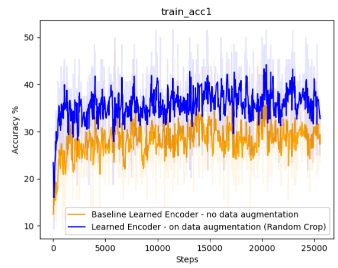
(a) Shows the undistorted observations before inputting it to the VAE.

(b) Shows the observations transformed with change in lighting by the VAE.

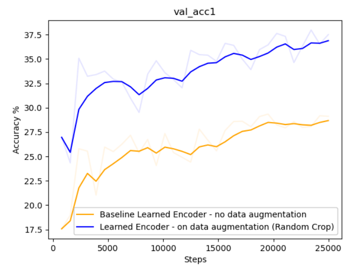
Figure 4: VAE learned on augmented data changes the lighting of the images.

6.2.3 Classification experiment on CIFAR-10

For the classification task, we test the learned VAE on the CIFAR-10 dataset due to the lesser number of classes when compared to the Tiny-Imagenet dataset. We test the VAE trained with augmentations included and VAE trained on undistorted observations. We then make a comparison of both the VAEs and visualize improvised generalization of the VAE trained on augmented observations.

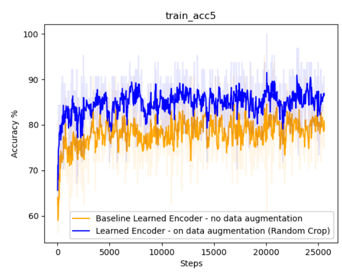


(a) Top-1 training accuracy comparison of VAE trained on undistorted observations and VAE trained on augmented observations.

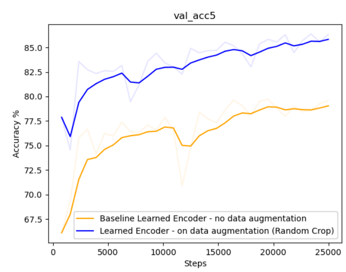


(b) Top-1 validation accuracy comparison of VAE trained on undistorted observations and VAE trained on augmented observations.

Figure 5: Top-1 accuracy of the VAE trained on augmented observations.

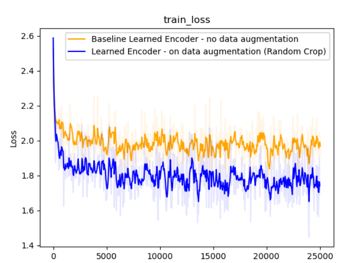


(a) Top-5 training accuracy comparison of VAE trained on undistorted observations and VAE trained on augmented observations.

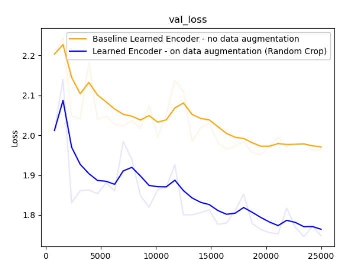


(b) Top-5 validation accuracy comparison of VAE trained on undistorted observations and VAE trained on augmented observations.

Figure 6: Top-5 accuracy of the VAE trained on augmented observations.



(a) Training loss comparison of VAE trained on undistorted observations and VAE trained on augmented observations.



(b) Validation loss comparison of VAE trained on undistorted observations and VAE trained on augmented observations.

Figure 7: Magnitude of the loss function of the VAE trained on augmented observations.

7 Conclusion

We introduced a method of training the variational autoencoder on augmented data to learn the data augmentations. We then trained the VAE on multiple augmentations one at a time to identify the set of transforms for which the label is invariant. We predict that the augmentations that do not hinder the intrinsic properties of the label improve the generalization of the VAE (color augmentations in most cases make the label corrupt). We tried to validate the performance of the VAE learned on augmented data which resulted in proving that VAE learned on augmented data does generalize better. The potential benefits of training the VAE on augmented data as its potential benefits are changes in scenic lighting, change in viewpoint, and other nonlinear transformations which were included in the results section. Furthermore, we could try to utilize the nonlinear transformations of the VAE to improve several classification tasks. If the resultant nonlinear transformations are successful in improvisation, we could minimize the burdensome task of collecting data and labeling them. It would also be interesting to see if reinforcement learning techniques could benefit from VAE learned on augmented data.

References

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2014.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Tien Ho-Phuoc. Cifar10 to compare visual recognition performance between deep neural networks and humans. *arXiv preprint arXiv:1811.07270*, 2019.

Tiny-imagenet dataset. `http://cs231n.stanford.edu/tiny-imagenet-200.zip`.

Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning, 2017.

Daniel Jiwoong Im, Sungjin Ahn, Roland Memisevic, and Yoshua Bengio. Denoising criterion for variational auto-encoding framework. *arXiv preprint arXiv:1511.06406*, 2016.

Terrance DeVries and Graham W. Taylor. Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538*, 2017.

Alexander J. Ratner, Henry R. Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. *arXiv preprint arXiv:1709.01643*, 2017.