Learning Image Perturbations from Examples (Generative Modeling)

Ying Sheng, Haoze Wu

Abstract

While significant progress has been made toward improving the robustness of neural networks against limited classes of transformations, a significant gap exists between the transformations that can be handled by current methods and the distribution shifts encountered in real-world applications. Broadly speaking, this gap between theory and practice exists because existing methods only extend to transformations that can be described by simple analytical sets, whereas the majority of real-world transformations cannot be defined in this way. In this work, we use data to learn perturbation sets in the latent space of conditional generative models which accurately capture the real-world shifts present in the data. Building on top of existing work, we improve the interpretability of the latent vector by reducing its dimension and introducing regularization term that represents our prior knowledge about the perturbation being learned. We are able to train models that generate meaningful perturbed images and preliminary results show benefits of the proposed regularization technique.

1 Problem Definition

In the past decade, Neural networks have achieved state-of-the-art performance in various application domains. Despite their general success, it is well-known that Neural Networks are susceptible to input perturbations that humans are naturally invariant to. This calls into questions the trustworthiness of neural networks and hinders their application in safety-critical domains. Indeed, training models that are robust to human invariants has become a core research area in the Machine Learning community.

Although much progress has been made in this area, most of the current work in robust machine learning have been focused on robustness against mathematically well-defined perturbation sets, for example, perturbation in l_p norm balls. However, in practice it is usually difficult, if not impossible, to define real-world perturbations analytically as a set of equations, making it challenging to learn or evaluate models that are robust against real-world perturbations in a systematic way. One way to address this challenge is to model the perturbation set learning from the perturbed data.

More concretely, given a dataset that consists of pairs of images (x, x'), where x is an image and x' is a perturbed version of x, our goal is to learn two Neural Networks: 1) an encoding network P, which takes in a pair (x, x') and outputs a latent vector that encodes the underlying perturbation represented by the (x, x'); 2) a decoding network Q, which takes in an unperturbed image x and the latent vector encoding of the perturbation z, and produces a perturbed version of x.

The perturbation set corresponding to an image x, would be defined as $\{Q(x, z) | \|z\| < \epsilon\}$. But we are not restricted on this definition of perturbation. The idea behind a small ball of ϵ reflects that a deep learning model should be robust according to local noises, or say, the output should be "smooth" according to the input. On the other hand, we may also want the model to ignore irrelevant factors. For example, the labeling of an image of hand writing number should not be affected by the color, rotation, etc. In this case, the perturbation may not lying in a small ball.

CS230: Deep Learning, Spring 2021, Stanford University, CA.

A possible application for our perturbation generator is to use it for checking or guiding the robustness of deep-learning models.

2 Challenges

The main challenge is to train the encoder such that it produces interpretable encoding of the perturbation. For instance, suppose the perturbation we are considering is rotation. The encoder's output should have low dimension (e.g., 1). In addition, suppose y is the intermediate state between two images, x and z (i.e., x is the original image, z is x rotated by θ, and y is x rotated by $\frac{\theta}{2}$). We should ideally have closure properties like $Q(x, P(x, y)) \approx y$, and linear properties like $Q(Q(x, P(x, y)), P(x, y)) \approx Q(x, P(x, z))$ ($P(x, y)$ is the latent vector that encodes $\theta/2$, $P(x, z)$ encodes θ and $\theta/2 + \theta/2 = \theta$).

More generally, for any three images x, y, z, we should ideally have $Q(Q(x, P(x, y)), P(y, z)) \approx Q(x, P(x, z))$ ($x \to y \to z = x \to z$).

3 Related work

The approaches proposed in [7, 5, 6, 1, 2] aim to learn perturbation sets from data in order to characterize real-world shifts for robust training and evaluation of neural networks. In [7], the authors use a conditional generative model to define perturbation sets over a constrained region of the latent space to perform robust training; we use a similar approach for learning perturbation sets in this paper.

4 Dataset and Features

We use the `mnist_corrupted` [1] [4] to evaluate our algorithm. `mnist_corrupted` is a dataset generated by adding 15 corruptions to the test images in the MNIST-10 dataset, which contains 10000 28×28 gray-scale images. For a given corruption, we use 70% as training examples, 15% as validation examples, and 15% as testing examples. We plan to organized each data point into a pair of images (x, x'), where x is the original test image and x' is the perturbed image. We plan to normalize the image from pixel value to $[0, 1]$.

5 Methods

One promising approach for our task is laid out in [7], which uses a Conditional Variational Autoencoder (CVAE) trained to generate $x' \in R^m$ from a latent vector $z \in R^k$ conditioned on x, where x is the unperturbed version of an image. In our case m is 28×28, an k is a hyper-parameter that we shall tune specifically for each type of corruption. An implementation[2] is provided in [7]. We aim to improve previous results by adding regularization terms to the cost function to enforce desired properties of the latent encoding (e.g. the closure property described in Section 2), similar to the techniques laid out in [3].

5.1 Dimentionality reduction for latent vector

Our implementation inherit from the original implementation in [7]. We focus on the rotation perturbation[3]. For each image x in the training set, we pair it with two perturbed images x_1, x_2, which are generated by sampling a $\theta \in \mathbb{R}$ uniformly from $[-20, 20]$ and then rotating with degree θ and 2θ.

For each original image x, we send (x, x_1) and (x, x_2) as a bundle to the network. The encoder is expected to output a distribution μ_1^x, σ_1^x corresponding to (x, x_1), and μ_2^x, σ_2^x corresponding to (x, x_2). The generator will takes in $\mu_1^x, \sigma_1^x, \mu_2^x, \sigma_2^x$ to output perturbed images $\hat{x_1}, \hat{x_2}$.

[1] `https://www.tensorflow.org/datasets/catalog/mnist_corrupted`

[2] `https://github.com/locuslab/perturbation_learning`

[3] See `perturbation_learning/configs/mnist_fc_pair_rotation.json` and `perturbation_learning/perturbation_learning/perturbations.py`

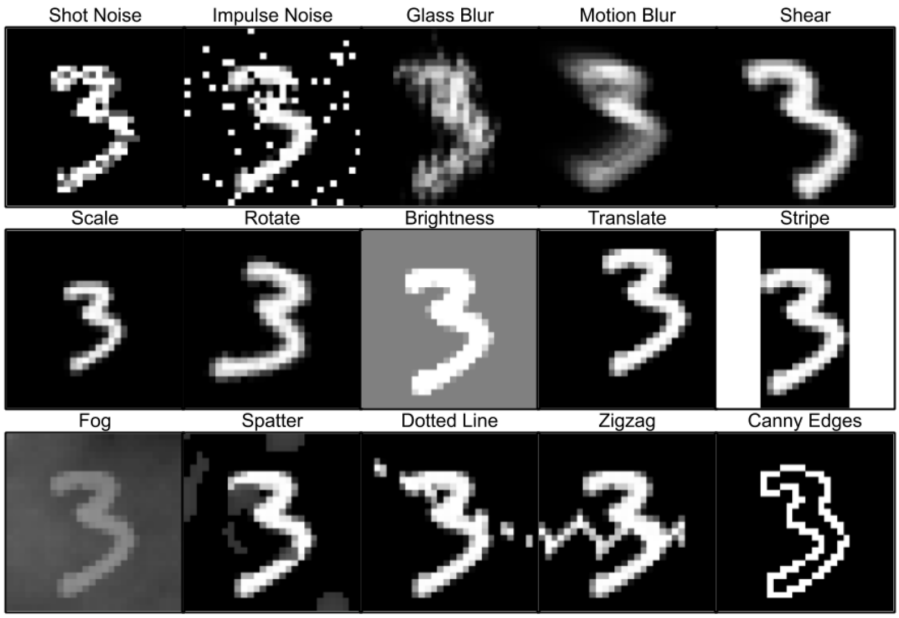

Figure 1: Examples of corruted images, taken from [4].

Together with the prior distribution $\mu_{prior}, \sigma_{prior}$ which corresponding to the whole training set, the baseline cost function[7] is the canonical CVAE loss:

$$\sum_x (x - \hat{x}) + \sum_x (D_{KL}(\mu^x, \sigma^x, \mu_{prior}, \sigma_{prior}))$$

Originally, the original code in [7] uses a latent vector dimension of 784. However, not only is this unnecessary for modelling transformations like rotation, it also makes the latent vector uninterpretable. Therefore, we set the latent vector of perturbation to 1-dimensional, with the intuition that rotation can be embedded with a one dimension latent variable. We refer to the latent vector as z.

Dimentionality reduction on latent vector already gives us a good interpretation on z, as its magnitude appears to be related to the rotation degree. See Section 6.

5.2 Linear-cost regularization

Additional to the model above, we add a regularization term that aim to help with the interpretation. Recall that we introduce a unit called sample bundle. Intuitively, the magnitude of μ_2^x deviated from μ_{prior} should be as two times larger as μ_1^x deviated from μ_{prior}. To enforce this property, we introduce the following regularization term:

$$\sum_x |2(\mu_1^x - \mu_{prior}) - (\mu_2^x - \mu_{prior})|$$

After adding our regularization, we observe that in the generated perturbed images (see Section 6), the latent vector is more concentrated on the rotation degree. As we shall see, the latent vector can still control the rotation degree, while the quality of the generated images is higher.

6 Evaluation

We evaluate our model on canonical CVAE metrics, such as KL divergence and reconstruction error. We also verify that the encoding of a perturbation has the desired closure properties described in Section 2, qualitatively by examining the generated images.

We train for 5 epochs for both the model without linear-cost regularization or the one with linear-cost regularization. The loss for these two models are similar, as the regularization term has a relatively small value. See Table 6.

model, dataset	reconstruction loss	kl-divergence (+ linear-cost regularization)	total loss
wt. reg. , train	133.2788	0.0057	133.2846
wt. reg. , test	92.2626	0.0079	92.2705
with reg. , train	132.784	0.9545	133.7385
with reg. , test	92.3044	0.4175	92.7219

We enumerate the value of the latent vector z from $\mu - \sigma$ to $\mu + \sigma$. As the z increases, we expect to see that the image is rotated by a higher degree. We pick an example to show in this report. For an image with hand-writing 6, we extract the μ, σ from the Encoder for this image, and call the generator to generate 11 perturbed images according to 11 different values of z. The values of z are chosen in the range $\mu - \sigma$ to $\mu + \sigma$ with equal distance:

```
for i in range(11):
    output = sess.run([output_name],
             {image_name: image.astype(numpy.float32),
              z_name : mean + (-5 + i)*std/5})[0].reshape(28,28)
```

Figure 2 shows the 11 perturbed images generated by CVAE trained without linear-cost regularization. And we can easily see that the pattern has been gradually rotated w.r.t. the increasing z. We also observe that the pattern becomes more blur and distorted when increase z. We believe the reason is that z also embedded with some meanings about other noises.

On the other hand, in Figure 3, we can also see the rotation resulting from the increasing z. And at the same time, the patterns are clearer and has higher quality compare to the model without regularization. We believe this is because the linear-cost regularization concentrates the model on the underlying perturbation over the random noises.

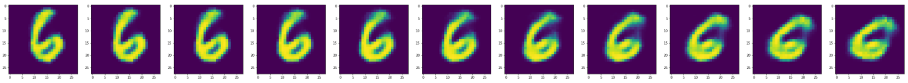

Figure 2: Images generated by CVAE trained without linear cost.

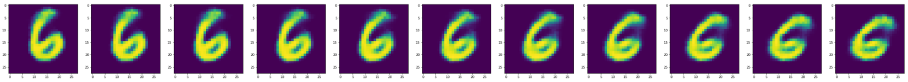

Figure 3: Images generated by CVAE trained with linear cost.

7 Conclusion

In this work, we aim to learn an interpretable and effective latent encoding of perturbations. We extended existing work [7] by reducing the dimension of the latent vector and adding regularization terms that correspond to our prior knowledge about the perturbation. We have obtained some interesting and promising initial results. Next steps would be trying to learn more complex transformations and devising other meaningful regularization terms.

8 Contributions

Most of the the programming and implementation are accomplished via pair programming. The writing is also done in a collaborative manner.

References

[1] S. Gowal, C. Qin, P.-S. Huang, T. Cemgil, K. Dvijotham, T. Mann, and P. Kohli. Achieving robustness in the wild via adversarial mixing with disentangled representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1211–1220, 2020.

[2] C. Laidlaw, S. Singla, and S. Feizi. Perceptual adversarial robustness: Defense against unseen threat models. *arXiv preprint arXiv:2006.12655*, 2020.

[3] M. Li, D. P. Losey, J. Bohg, and D. Sadigh. Learning user-preferred mappings for intuitive robot control. *arXiv preprint arXiv:2007.11627*, 2020.

[4] N. Mu and J. Gilmer. Mnist-c: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*, 2019.

[5] A. Robey, H. Hassani, and G. J. Pappas. Model-based robust deep learning. *arXiv preprint arXiv:2005.10247*, 2020.

[6] A. Robey, G. J. Pappas, and H. Hassani. Model-based domain generalization. *arXiv preprint arXiv:2102.11436*, 2021.

[7] E. Wong and J. Z. Kolter. Learning perturbation sets for robust machine learning. *arXiv preprint arXiv:2007.08450*, 2020.