CS230

Data Synthesis with Stable Diffusion for Dataset Imbalance - Computer Vision

Torstein Ø. Eliassen
Department of Electrical Engineering
Stanford University
torsteoe@stanford.edu

Yuntao Ma
Department of Electrical Engineering
Stanford University
yma42@stanford.edu

Abstract

As Neural Networks are entering new and diverse industries and domains, situations with imbalanced datasets often occur. Despite improving performance of Neural Nets, classical data augmentation techniques still suffer from dataset biasXu et al. [2020]. Generated data from GANs Goodfellow et al. [2014]can alleviate this bias, but often suffers from unstable training. This paper proposes using Stable DiffusionRombach et al. [2021] together with Active Learning to rebalance datasets and achieves considerable increase in classification accuracy compared to the mentioned techniques.

1 Introduction

When training neural networks, the data fed into the network is often just as important (if not more) than the architecture of the network. However, labeled data can be expensive and sometimes, regardless of budget, difficult to obtain. As an example, to avoid hitting pedestrians on the highway, it would be preferable if the object-detection networks of self-driving cars had trained on photos with pedestrians on a highway before, but generating these photos in the real world is clearly very dangerous. This project is trying to solve the problem of incomplete datasets with synthetic data-generation and will use the open source Stable diffusion Rombach et al. [2021] model to generate realistic images. The input to the algorithms will be text inputs including the label and the output will be realistic images containing the desired labels. We propose a hybrid technique using both generative models and active learning to learn from the best synthetic images. A GAN is also built and trained from scratch as a baseline comparatative model to the diffusion model.

The diffusion model implemented in this project will also be used in the CS299 final project and the shared repository is found at `https://github.com/Tma2333/StableDiffusionProject`.

2 Related work

When dealing with dataset imbalance, some of the traditional techniques are 1. Dataset re-sampling to create a balanced set Japkowicz and Stephen [2002], 2. Augmenting existing data to inflate the number of samples of the underrepresented class Simard et al. [2003] , 3. Using weighted cost to encourage learning from the underrepresented class Thai-Nghe et al. [2010].

In 2014, the proposed GANs Goodfellow et al. [2014] were able to make highly realistic images synthetically and have also been used for data augmentation Sandfort et al. [2019]. However, they still suffer from unstable training and the recent diffusion models Sohl-Dickstein et al. [2015] are shown to be able to create better resulting images Dhariwal and Nichol [2021]. A lot of the models

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

are unfortunately behind locked doors. We therefore use Stable Diffusion's open source code to do data synthesis in this project and implement a GAN as a baseline model for comparatative purposes.

3 Dataset and Features

In order to objectively evaluate our synthetic method, we start with an existing dataset and create an artificial imbalanced dataset from it. We choose CIFAR-10 Krizhevsky [2009] due to its relative small size allowing us to iterate through different methods and synthetically generate new data for the classification task efficiently.

CIFAR-10 consist of 60000 32x32 labeled color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. Our dataset will have 3 general categories: a full set, an imbalanced set, and a synthetic set. The original CIFAR-10 training set will be the full set. To construct an imbalanced set, we select one class, "cats", and randomly remove 99% of the image of that class from the training set. Then, we use classical data augmentation techniques, a diffusion model and a GAN to generate images representing the imbalanced class. The test set will be left untouched.

For generating images with a GAN, we train a GAN on a merged dataset of cat faces (Ferlito) from 5 different sources. The dataset consists of approximately 30000 images.

4 Methods

4.1 Non-synthetic Method

Though inexpensive compared to collecting or labeling more data, generating millions of synthetic images is still a higher commitment versus simple non-synthetic methods. In order to justify our synthetic dataset, we would also like to see how it stacks up against the non-synthetic methods described below:

1. Naively over-sample the imbalanced class to have same amount of learning samples in the train set.
2. Perform random data augmentations on existing imbalanced class to create more learning samples for underrepresented class.

The extra augmentation used for oversampled images are as follow:

- **Random Color Jitter** with 0.5 brightness factor and 0.3 hue factor.
- **Random Perspective Shift** with 0.2 distortion scale, and constant 0 padding.
- **Random Rotation** between 0 to 180 degrees and constant 0 padding.
- **Random Solarize** wit threshold at 0.75 and 0.5 apply probability

4.2 Synthetic Method: GAN

For our baseline GAN model, we modify the DCGAN architecture(Radford et al. [2015]) [1]. This architecture was used on human facesLiu et al. [2015] which may be easier than cat faces that vary a lot. After some unsuccesful trainings on the cat dataset, we therefore increased the complexity of the architecture.

The generator consists of 4 Up-Convolution blocks with output feature sizes: 1024, 512, 256, 128. Each block contains one transpose convolution layer with kernel size of 4 and stride of 2 and padding of 1, batch normalization layer and a ReLU layer. Then the output of last Up-Convolution block is passed through a transpose convolution layer with output channel of 3 and a tanh layer to generate the final image.

The discriminator consists of 4 Down-Convolution blocks with output feature sizes: 128, 256, 512, 1024. Each block contains one convolution layer with kernel size of 4 and stride of 2 and padding of

[1] `https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html`

1, batch normalization layer and a leaky ReLU layer with factor of 0.2. Then the output of the last Down-Convolution block is passed through a convolutional layer with output size of 1 and a sigmoid layer to predict image source.

We chose learning rates from 0.0002 to 0.00001 and found 0.00005 to give the most stable training. Initial training was for 100 epochs, but after seeing continuous improvement, we trained for 300 epochs to improve the results. In implementation, we removed the final sigmoid layer from the discriminator and use for loss, a Binary Cross Entropy Loss With Logits. This is equivalent to the original Sigmoid + BCE, but is said to be more numerically stable, this was also confirmed empirically. Random vertical flip and random sharpness adjustments were implemented for data augmentation.

4.3 Synthetic Method: Stable Diffusion

Stable Diffusion is an open-source implementation of a *latent diffusion model*. The architecture consists of a variational autoencoder, a latent-space scheduler and conditional U-NetRonneberger et al. [2015], and a CLIP text embeddingRadford et al. [2021].

During inference, a random vector in the latent space is denoised with consecutive passes through the U-Net architecture where at each step, the noise is predicted and subtracted from the vector. Parallel to this, every text input is embedded. The embedding is fed into the U-Net as pre-conditioning, making our denoising step conditional on the prompt. The image is obtained by passing the final latent vector through VAE's decoder.

We implemented the Stable Diffusion model with pretrained weights. For demonstration purposes, one can often engineer a particular prompt to generate a few impressive images. However, we need a large amount of data (4950 unique images for CIFAR-10). We need an efficient strategy to sample diverse, high-quality data. We create a prompt template

$$< \text{BREED} > \text{cat} < \text{PREPOSITIONS} > \text{the} < \text{FURNITURE} > \tag{1}$$

We collected 109 cat breeds, 85 types furniture, and 11 prepositions. In total we generated 101,915 images with all combinations and randomized seeds.

4.4 Active Learning

Active Learning is a semi-supervised machine learning process, which builds a dataset iteratively. Training starts out with our small known dataset and at each iteration the best samples are chosen and added to the training set. As the dataset grows, the accuracy of the predictor improves and we increase the chance of finding better samples. Samples are added until we reach a desired number of samples. We choose Deep Bayesian Active Learning (DBAL) Gal et al. [2017] as our active learning framework. By adding dropout layers we create a Monte Carlo dropout network. By keeping dropout during inference, we form an approximation of the posterior of the data. The posterior is given by

$$P(y|x_i) = \sum_{j=1}^{T} \frac{1}{T} P(y|x_i, \theta_j) \tag{2}$$

where T is number of forward passes performed through the Bayesian network. θ_j is the effective parameters (non-dropout parameters) for each forward pass. We can maximize this approximate posterior to determine generated samples that will bring maximum entropy (i.e. information gain). We start training with the remaining 50 images from the imbalanced class, and test on all Stable Diffusion synthetic images. Every iteration, we add the top 50 images to the training set. By using active learning, we should hopefully build up a better synthetic dataset than simply random sampling.

4.5 Classifier Network

We use ResNet18 with minor modification to adjust for CIFAR's small image size. We replace the first convolutional layer's (7, 7) kernel to (3, 3) kernel. We also remove the first max pool layer to reduce over-all downsample factor from 32 to 16.

5 Experiments, Results and Discussion

We generate 5000 cat face image using GAN in resolution of 64x64. We also generated 101,915 cat images using Stable Diffusion with prompt strategy mentioned in Section 4.3 in resolution of 512x512. All images are re-scaled to 32x32. In total we created following 7 dataset:

1. **Imbalance**: Original CIFAR-10 dataset with 99% of the *Cat* class removed
2. **Oversample**: Oversample remaining *Cat* class in **Imbalance** to match number of images in each class.
3. **Oversample + Augmentation**: Apply random augmentation mentioned in Section 4.1 to all oversampled images.
4. **GAN**: **Imbalance** set with 4,950 randomly sampled GAN synthetic images added to *Cat* class.
5. **Stable Diffusion Random**: **Imbalance** set with 4,950 randomly sampled Stable Diffusion synthetic images added to *Cat* class.
6. **Stable Diffusion + Active Learning**: **Imbalance** set with 4,950 Stable Diffusion synthetic image added to *Cat* class sampled using active learning.
7. **Full**: original CIFAR-10 dataset.

All data regardless of dataset is standardized with full set mean and standard deviation and padded random crop and random flip are applied.

Examples from each set are shown in Figure 1.

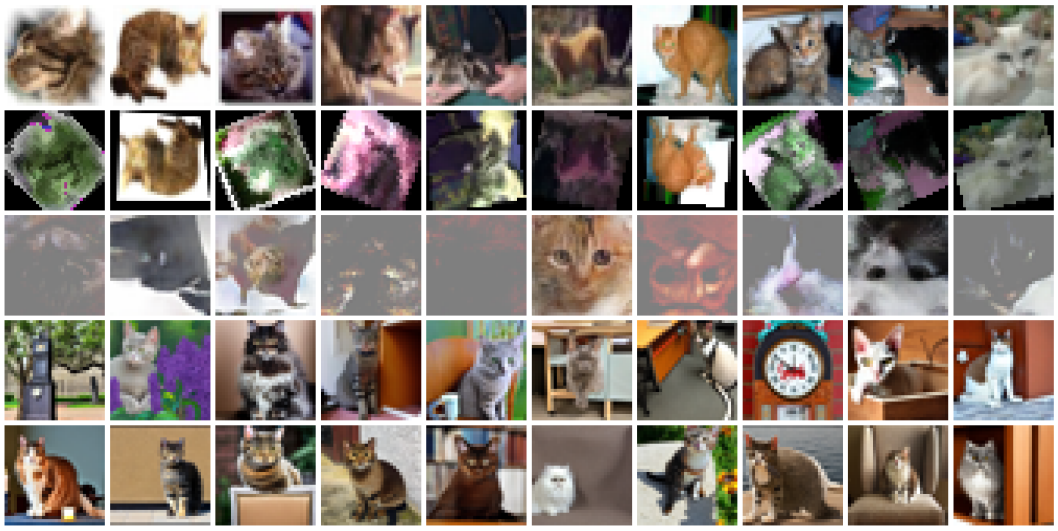

Figure 1: **1st row:** 10 random samples from original training set (CIFAR-10). **2nd row:** Augmentation on 1st row - images. **3rd row:** Synthetic images generated with GAN after 300 epochs **4th row:** Stable Diffusion synthetic images sample at random **5th row:** Stable Diffusion images sampled using active learning

We trained ResNet18 on all above mentioned datasets. We trained with a batch size of 256 and an SGD optimizer with learning rate of 0.1, weight decay of 0.0005, and momentum of 0.9. We use one cycle of cosine annealing decay with 5 warm up epochs. We trained all datasets for 100 epochs and fixed seed to minimize run to run variance. We test the best model from each run using the unaltered test set. See results in Table 1.

The imbalanced set performs the worst overall, and did not learn cat class at all due to the large discrepancy in number of data in each class. The altered cat class only has 50 training samples while the other class has 5000 training samples. However oversampling and augmentation did little to help bridge the gap, as we cannot learn rich and different sets of features for cat class.

Dataset	Test Acc	Cat	Bird	Horse	Deer	Airplane	Automobile	Truck	Dog	Frog	Ship
Imbalanced	85.689	0	91.66	96.82	94.15	95.84	97.15	94.87	95.03	95.15	96.22
Oversample	85.404	3.08	91.3	96.08	93.42	94.72	97.15	92.58	95.85	94.23	95.63
Oversample + Augmentation	85.414	5.61	90.86	96.55	94.07	95.87	97.06	90.11	95.09	93.15	95.77
GAN	85.66	9.95	90.93	96.25	93.74	91.41	96.75	91.51	95.97	93.5	96.59
Stable Diffusion Random	89.022	34.41	90.22	96.5	93.58	95.54	96.94	95.47	96.13	95.8	95.63
Stable Diffusion + Active Learning	90.805	51.37	90.84	96.5	93.94	95.49	97.78	94.84	95.33	94.99	96.97
Full set	93.77	88.51	90.52	95.46	93.44	94.95	97.88	95.99	91.5	92.46	97.01

Table 1: result of the experiments, targeted imbalance class is highlighted in gray

We see more significant improvement when training with the synthetic dataset, as it provides a rich learning signal for the network to learn different features of the cats. One of the significant challenges we saw is that, the synthetic data might not perfectly represent the distribution of the original classes. Both GAN and Random Sample Stable Diffusion images suffer from this problem. In Figure 1, we can see a image has a cat clock rather than a cat next to the clock. By utilizing active learning, we found the resulting images to be more representative of the original dataset.

To investigate this further, we perform PCA on the output features vector of the model trained on original CIFAR-10 dataset.

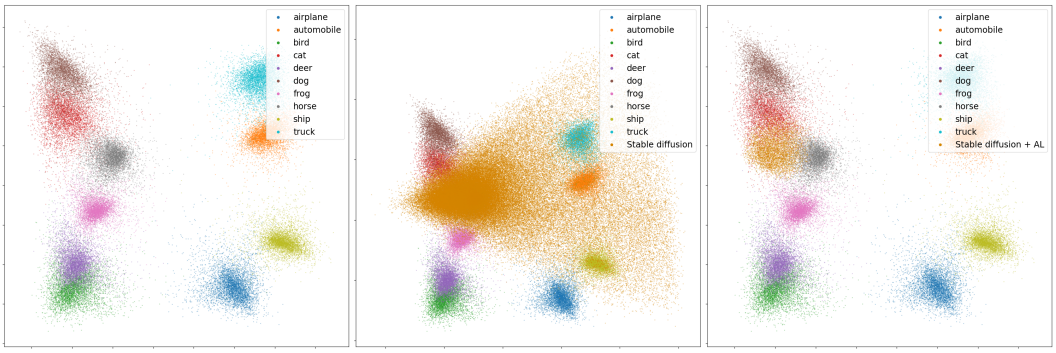

Figure 2: **Left:** First two PC of the features produced from original training set **Middle:** First two PC of the features produced from original training set and all Stable diffusion synthetic images, **Right:** First two PC of the features produced from original training set and synthetic image sampled using Active learning

Figure 2 indicates that Stable Diffusion generates many image out of the distribution of the targeted class. Randomly sampling from these spaces is not ideal. Using active learning, we can select images that are relevant and this should help improve the performance. However, more sophisticated prompting or generation method should be adapted to minimize out of distribution generation.

6 Conclusion and Future work

In this project, we investigate using generative models, such as GAN and Stable Diffusion to create synthetic datasets to address the class imbalance problem. We use CIFAR-10 as our reference dataset, and create am imbalanced set by removing 99% of the cat class. We compare synthetic methods with traditional oversampling and augmentation methods. We found that synthetic methods outperform the oversample and augmentation method with a large margin. However, synthetic methods cannot consistent create data that represent the original distribution. We found that over-generation plus active learning could help with this issue.

In the future, we wish to explore synthetic strategies that can consistently generate image that can represent the original data distribution using limited data. For example, a more sophisticated prompt generation schema should be explored. We can also potentially use remaining data to condition the de-noise network in diffusion model.

7 Contributions

Yuntao Ma implemented the Stable Diffusion model and also set up the pipeline for testing the different methods on imbalanced datasets. Torstein implemented, tuned and trained the GAN. Joint work was done on prompt generation for Stable Diffusion, result analysis and report writing.

References

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. URL https://arxiv.org/abs/2105.05233.

Federico Ferlito. Cat-faces-dataset. https://github.com/fferlito/Cat-faces-dataset.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. *CoRR*, abs/1703.02910, 2017. URL http://arxiv.org/abs/1703.02910.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, pages 429–449, 2002.

Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015. URL https://arxiv.org/abs/1511.06434.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Veit Sandfort, Ke Yan, Perry J Pickhardt, and Ronald M Summers. Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks. *Scientific reports*, 9(1):1–9, 2019.

P.Y. Simard, D. Steinkraus, and J.C. Platt. Best practices for convolutional neural networks applied to visual document analysis. pages 958–963, 2003. doi: 10.1109/ICDAR.2003.1227801.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. URL https://arxiv.org/abs/1503.03585.

Nguyen Thai-Nghe, Zeno Gantner, and Lars Schmidt-Thieme. Cost-sensitive learning methods for imbalanced data. pages 1–8, 2010. doi: 10.1109/IJCNN.2010.5596486.

Yi Xu, Asaf Noy, Ming Lin, Qi Qian, Hao Li, and Rong Jin. Wemix: How to better utilize data augmentation, 2020. URL https://arxiv.org/abs/2010.01267.