SMate: Synthetic Minority Adversarial Technique

Pablo Rodriguez Bertorello
Computer Science
Stanford University
prodrig1@stanford.edu

Liang Ping Koh
Statistics
Stanford University
lpkoh@stanford.edu

Abstract

In important prediction scenarios, data-sets are naturally imbalanced, for instance in cancer detection: a small minority of people may exhibit the disease. This poses a significant classification challenge to machine learning algorithms. Data imbalance can cause lower performance for the class of interest, e.g. classifying with high precision that the person has cancer. When training data is abundant, a possible approach is to down-sample the majority class, thus restoring balance. Another prevalent approach is weighting, accelerating learning for minority class training examples[7]. Synthesis is a major alternative, producing examples of the minority class, adding them to the training set to overcome the class imbalance. The Synthetic Minority Over-sampling Technique, SMOTE[3] is widely applied, but it was not developed for image data. Rather, this research applies Generative Adversarial Networks[4], which generate image examples drawn from the minority class distribution. The novel SMate approach leverages GAN minority-class image generators, which benefit from Transfer Learning from majority-class image generators. Consequently, SMate outperforms SMOTE for imbalanced image data-sets.

Index Terms – Data-set Class Imbalance, Image Classification, Down-sampling, Example Weighting, Synthetic Minority Over-sampling, SMOTE, Adaptive Synthesis, ADASYN, Generative Adversarial Network, GAN, Transfer Learning, Deep Neural Network, DNN, Convolutional Neural Network, CNN

1. Introduction

The goal of this paper is to solve minority-class classification for imbalanced data-sets. The main contribution is an algorithm that outperforms SMOTE and ADASYN for image synthesis. We propose the use of Generative Adversarial Networks and Transfer Learning.

Section 3 introduces the CIFAR data-set. In Section 4, we provide a brief introduction to training neural networks. Experimental results are showcased in 5. Final conclusions are in Section 6

2. Related Work

Our work is inspired by Generative Adaptive Networks, Transfer Learning, and Classifier Boosting.

Data Synthesis: the Synthetic Minority Over-sampling Technique[3], SMOTE, synthesizes new examples of the minority class. For each member of the minority class, SMOTE finds its k nearest neighbors, and then randomly chooses a point on the line that connecting them. The point is used as a new synthetic training example. In many domains, this approach creates realistic examples. However, such linear combinations are unlikely to produce realistic images.

Adaptive Synthetic Sampling[5], ADASYN, creates minority examples in the more challenging regions of the dataset. It finds the minority training examples crowded by other classes, Euclidean distance wise. For them, ADASYN determines that more examples of the minority class should be created. However, Euclidean distance is not a good method to measure image proximity.

Classifier Boosting: SMOTEBoost[2]: was proposed for moderately imbalanced datasets, where the F-score on the minority class is the primary performance criterium. However, F-score for minority classes is not always the performance goal. In some cases, false negatives are of greater concern, as in cancer detection. In some others, false positives are paramount, as in advertising they result in greater expense.

3. Data and Software Libraries

3.1. CIFAR10

The entire CIFAR10 data-set is utilized. It spans 10 classes each with 5,000 training examples and 1,000 test examples. Example are shown in Figure 1.

Each image in the data-set is normalized, so that networks do not learn features specific to intrinsic qualities of an image, for example dead pixels or ambient lighting.

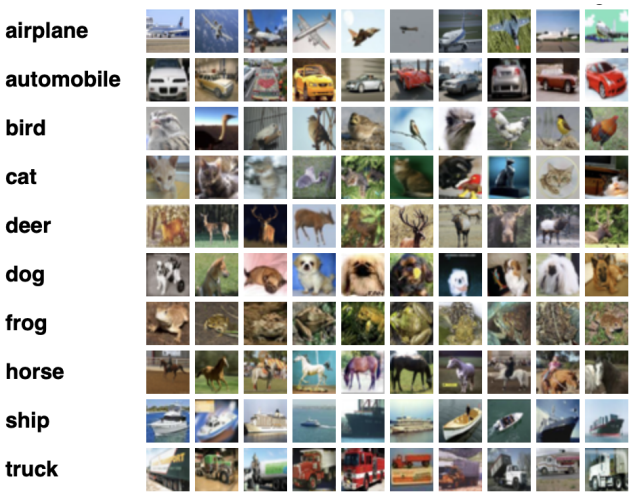

Figure 1. Sample CIFAR10 Class Images

3.2. Third Party Software

All code is implemented in Python leveraging open source packages: Keras, Tensorflow, SciPy, NumPy.

Amazon Web Service's SageMaker served as Integrated Development Environment, working across instances via their Elastic File System. The instance type used is conda_tensorflow_p36. For compute throughput, compute instances in the p2 GPU family are applied.

Visualizations of neural networks performance are implemented using wandb.com, matplotlib, and pydot. The first required the installation of graphql-core.

Data augmentation is performed with the imgaug library.

4. Technical Methods and Approach

4.1. Gradient Descent

Gradient Descent is applied to numerically minimize a loss function, the objective to best fit a neural network's parameters to a given data-set. See equation 1, where $W_{current}$ is the current network weight matrix, and the gradient F is followed each iteration with lr learning rate.

$$W_{new} = W_{current} - lr\ \nabla F(W_{current}) \tag{1}$$

4.2. Image Classification

The outputs of a neural network can be trained to non-linearly compute regression values.

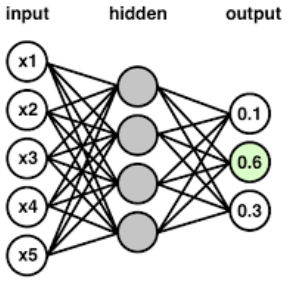

Figure 2. Neural Network Classifier Diagram[10]

For classification, the output layer of the neural network is fed into classification stage. For example, Softmax outputs a one-hot encoded vector indicating the predicted class.

$$softmax(x)_i = \frac{exp(x_i)}{\sum_j exp(x_j))} \tag{2}$$

4.3. Classifier Ensembles

Ensemble methods include Bagging and Boosting. Adaptive Boosting[8], AdaBoost, iteratively builds an set of classification models by adjusting the weights of mis-classified data during each iteration. Initially every example has equal weight. For each subsequent model, weights are recalculated such that a higher emphasis is assigned to samples mis-classified thus far. This weight determines the emphasis in the loss function that is placed on the example in the training of the next weak learner.

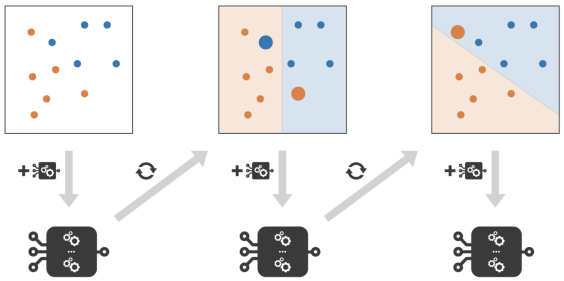

Figure 3. Ensemble Boosting Method[?]

AdaBoost was developed for binary classification problems with one-split decision trees, weak learner form of decision tree. Likewise, SAMME[11] generalizes boosting for multi-class classification.

Since we were unable to find an open source implementation for neural networks, we implemented it for multi-class image classification CNNs.

4.4. Generative Adversarial Network

Generative Adaptive Networks[4], GANs for short, estimate generative models following an adversarial process. A network D is trained to classify if an example is truly from

a chosen distribution, or fake. And a network G is trained to synthesize examples to fool D. However, in training G for a minority class, the network can over-fit to few of the known true examples. It can also produce examples that do not appear natural to a human.

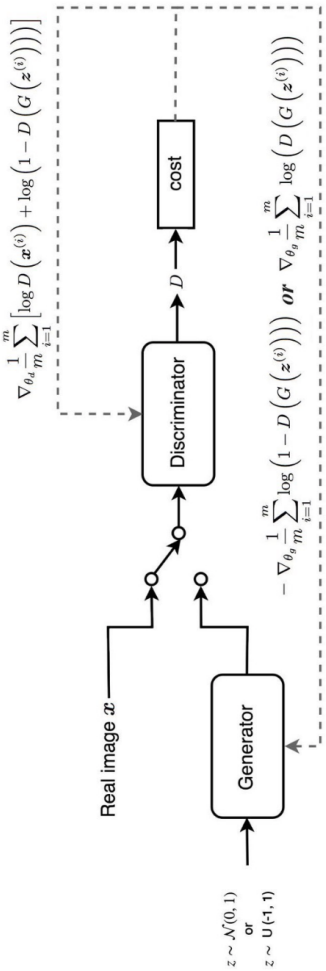

Figure 4. Generative Adversarial Network Diagram [6]

4.5. Transfer Learning

Transfer Learning [9] is the process of applying knowledge from previously-learned tasks to learn new related models. Specifically, given a network architecture, and its associated learned parameters, part or all of it can be used as input features into another model.

5. Results and Analysis

The idea is to rely on Generative Adversarial Networks to generate examples of a minority class, to balance an otherwise imbalanced data-set. Performance is validated by the accuracy of a Classifier that relies on those generated examples.

5.1. Problem: Imbalanced Data-set Performance

When a classifier architected to perform well on CIFAR10 images is trained with the entire data-set, its test set performance matches its training set performance.

To evaluate the performance of our overall method, we induce a data-set imbalance, down-sampling one of the classes to 10%. As can be seen in Figure 5 this severely impacts the classifier's test set performance.

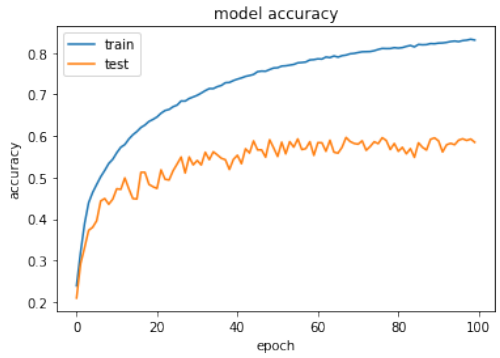

Figure 5. Accuracy per Training Epoch

5.2. Baseline: Prior Art Re-balancing Performance

Giving more weight to minority class examples proved to be the best prior art method, as can be seen for the chosen Truck class in Figure 6.

Model	Test Accuracy	Truck Class Recall
Full Data	77.5%	85%
Undersampled Data	76.1%	56%
Oversampled Data	66.6%	62%
SMOTE Data	75.7%	51%
ADASYN Data	76.6%	56%

Figure 6. Accuracy per Training Epoch

When a classifier is trained and tested with SMOTE, the minority class is re-balanced with SMOTE-generated examples. In the case of ADASYN, the algorithm identifies the hardest examples to classify, and generates new ones with SMOTE.

5.3. Novel SMate Approach

5.3.1 Data Pre-processing

The CIFAR10 training set has 50,000 examples, each image 32x32 pixels in 3 RGB channels. That is 5,000 examples for each of its ten classes. And after inducing class imbalance, sampling the Truck class down to 10% of its examples, this minority class had only 500 examples.

With so few minority examples to train a GAN on, we noticed the generated images over-fitted. Error analysis revealed, for instance, that a big red-colored blob could pass for a fire department trucks.

Thus we pursued Transfer Learning. We first trained the Generator on all the majority classes. Then froze 6 layers of the G sequence, leaving two convolutional layers unfrozen, proceeding to train for the minority class.

Various image augmentation approaches were tried for the minority class, Truck, resulting in the images in Figure 7. Random augmentation includes: Flip, Crop, GaussianBlur, ContrastNormalization, AdditiveGaussianNoise, Multiply, Afine. Heavier augmentations were abandoned as they frequently broke images.

Transfer learning worked well to a point. When we augmented the 500 Truck examples to 5,000 the GAN was able to converge. However, when the minority class was augmented to 50,000 examples the Generator was not powerful enough to converge within the allotted time.

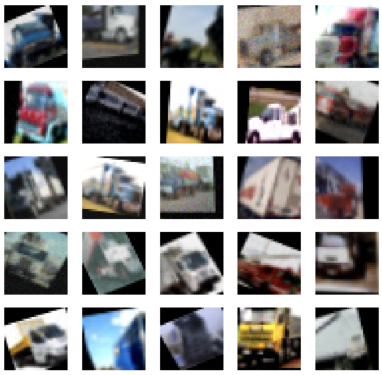

Figure 7. Image Augmentation of Truck Class with imgaug

5.3.2 GAN Training

We selected the Brownlee GAN architecture[1], we found it best suited to generate CIFAR10 images. Figure 8 shows the the G-sequence of the Generator's Convolutional Neural Network.

The code published with this research makes architecture replacement simple plug-and-play. For loss function, different functions are readily available: GAN, WGAN, LSGAN, DRAGAN, HINGE. We chose GAN understanding it sufficient to rebalance CIFAR10 data-sets.

In addition to architecture, we searched for hyper-parameters including optimizer, experimenting with the Adam and RMSProp optimizers, because of their momentum-driven fast convergence properties.

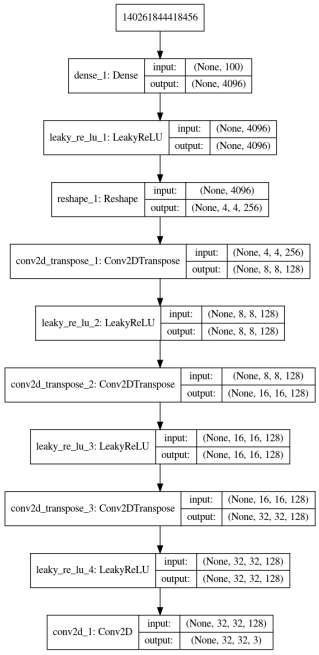

Figure 8. Brownlee CIFAR10 Generator

5.3.3 GAN Evaluation

Quantitative: We experimented with objective early GAN training termination criteria, but found none that minimized compute time while producing natural images.

Thus from a quantitative point of view, we primarily sought quick convergence to achieving the goal of generating images that fool the Discriminator. This can be confirmed by observing the Discriminator's fake catch accuracy fall to 50%. We found that images kept appearing more natural to a human well past that goal was met.

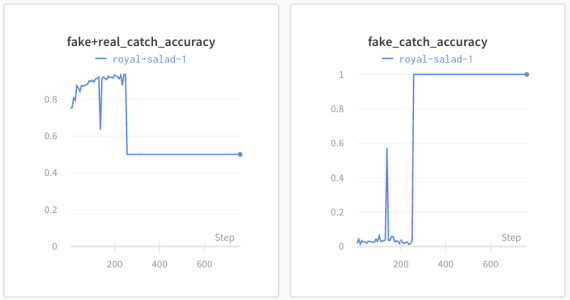

Figure 9. Generator and Discriminator Training Convergence

For Gradient Descent loss minimization we selected the Adaptive Momentum optimizer, aka Adam. The largest learning rate that did not suffer loss divergence was 0.0002.

With respect to the minority class, relying on the GAN's Generator to create images, the CIFAR data-set is re-balanced. Accuracy for our SMate approach is shown in Figure 10.

Please contrast to the classification accuracy on the imbalanced data-set of Figure 5. In fact, SMate performs better than every other approach: under-samplign, over-sampling, SMOTE, and ADASYN. The resulting confusion matrix is in 11.

Error analysis on the confusion matrix reveals that primarily the generated Truck images are sometimes confused for Car images. This points us to productive future research directions on loss functions for SMate, further discussed in Section 6.

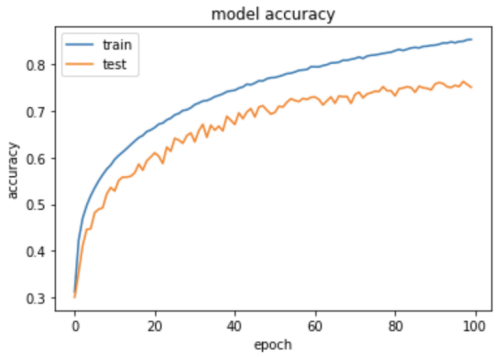

Figure 10. Truck Classification Accuracy for Truck Class after Rebalancing Data-set with SMate

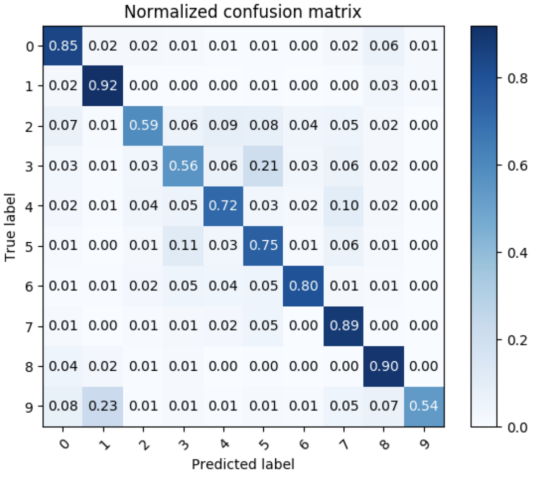

Figure 11. Truck Classification Confusion Matrix after Rebalancing Data-set with SMate

Qualitative: To assess convergence, a researcher subjectively evaluated image quality.

Given the adversarial training approach, the generator learns after every batch. Thus while image quality may generally increase, we saw it occasionally suffering setbacks. Figure 12 shows example generated images, which require some squinting and imagination to see them as Truck images.

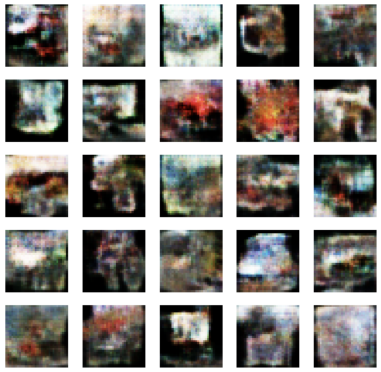

Figure 12. Generated Truck Images

5.3.4 SAMME Classifier Performance

We attempted to address minority class imbalance by also innovating in classifiers. However, our SAMME implementation of multi-class Ada-boost did not succeed. Normalization did not perform as expected, perhaps causing aggregations to be significantly biased from maximum likelihood.

6. Conclusion and Future Work

We propose the SMate method, which can be used to rebalance imbalanced data-sets. It relies on Transfer Learning a GAN generator trained for majority classes, leveraging it to learn to generate minority examples. It outperforms prior art methods including: under-sampling, over-sampling, SMOTE, and ADASYN.

As future direction, we believe that the GAN's loss function should be enhanced. Not only should it target generating examples of the minority class. It should also penalize generation of examples that get classified as majority objects.

In principle, this work is extensible to every data type. For example, investigation into SMate for time series data appears to hold high promise.

The project's repository is at https://github.com/pablo-tech/SMate–SyntheticMinirityAdversarialTechnique

7. Contributions

Both authors contributed to the overall direction, development, analysis, and conclusions.

Pablo: principally contributed to all aspects related to Generative Adversarial Networks.

Liang: principally contributed to all aspects related to Boosting.

8. Acknowledgements

The authors are grateful to Huizi Mao for his research guidance.

Special thanks to Professor Andrew Ng for educating us on all aspects of Deep Learning.

Authors

Pablo Rodriguez Bertorello leads Machine Learning at Sephora, where he launched their Deep Learning Practice. Previously he was CTO of Airfox, which completed a successful Initial Coin Offering. He is the co-inventor of a cloud platform company acquired by Oracle. And the original designer of the data bus for Intel's Itanium processor. Pablo has been issued over a dozen patents.

Liang Ping Koh is a Master's Degree student in the Statistics Department at Stanford University.

References

[1] J. Brownlee. How to develop a gan to generate cifar10 small color photographs. *Machine Learning Mastery*, https://machinelearningmastery.com/category/generative-adversarial-networks/, 2019. 4

[2] N. Chawla, A. Lazarevic, L. Hall, and K. Bowyer. Smoteboost: Improving prediction of the minority class in boosting. volume 2838, pages 107–119, 01 2003. 1

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June 2002. 1

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial networks. *ArXiv*, abs/1406.2661, 2014. 1, 2

[5] H. He, Y. Bai, E. Garcia, and S. Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. pages 1322 – 1328, 07 2008. 1

[6] J. Hui. What is generative adversary networks gan? *Medium*, 6 2018. 3

[7] G. Solon, S. Haider, and J. Wooldridge. What are we weighting for? *Journal of Human Resources*, 50(2):301–316, 2015. 1

[8] A. Tharwat. Adaboost classifier: an overview, 02 2018. 2

[9] K. Weiss, T. Khoshgoftaar, and D. Wang. A survey of transfer learning. *Journal of Big Data*, 3, 12 2016. 3

[10] A. Wong. Coding up a neural network classifier from scratch. *Towards Data Science*, 10 2017. 2

[11] J. Zhu, S. Rosset, H. Zou, and T. Hastie. Multi-class adaboost. *Statistics and its interface*, 2, 02 2006. 2