# GAN-SMOTE: A Generative Adversarial Network approach to Synthetic Minority Oversampling for One-Hot Encoded Data

Mitchell Scott and Jo Plested

Research School of Computer Science, Australian National University

**Abstract:** Small unbalanced datasets remain a serious impediment to the implementation of cutting-edge machine learning in an industry setting. This paper proposes GAN-SMOTE, a novel approach to synthetic minority class oversampling using a generative adversarial network that can be applied to boost the performance of classifiers learning from small and imbalanced one-hot encoded datasets. This paper also introduces techniques that are key for ensuring the stability and variance of the generative adversarial network in this setting. The proposed method demonstrates meaningful improvement on a well-studied petrographical dataset with significant class imbalance.

**Keywords:** GAN-SMOTE, synthetic data, minority class oversampling, generative adversarial network, unbalanced dataset, small dataset, minibatch discrimination, decreasing random bit-flips

## 1 Introduction

As a general rule, the more data that a machine learning algorithm has access to, the more accurate its predictions will be [1]. Small datasets present serious challenges to the effective use of machine learning techniques in real world applications because of their propensity to overfit on the available data [1]. The problems associated with small datasets are often compounded by class imbalance within the dataset, which can cause machine learning algorithms like Convolutional Neural Networks (CNNs) to converge slower during training and generalise poorly on unseen data [2].
Class imbalance occurs when one or more of the categories in a dataset has significantly more samples than others and can have significantly detrimental effects when training CNNs [2].

There are several traditional techniques for handling class imbalance. The first is undersampling, which involves removing instances of the majority class so there is an even number of samples in each class [2]. Although undersampling balances the number of samples in each class the classifier misses out on valuable data about the majority class [2]. In small datasets each datapoint is even more valuable, so removing samples in these cases will further increase the risk of sampling error and overfitting [3].

Another method is oversampling, where instances of the minority class are duplicated so that the classes become evenly distributed [2]. This technique again balances the number of samples in each class but can cause overfitting instead on the minority class as the idiosyncrasies of each minority sample are repeatedly shown to the classifier [4]. In neural networks, practitioners can manipulate the loss function to penalise misclassification of the minority class, called cost-sensitive learning, although in effect this has the same issues as oversampling [2].

Another canonical technique is Synthetic Minority Oversampling Technique (SMOTE) [5]. SMOTE is an over-sampling technique that creates synthetic samples by interpolating between members of the minority class in "feature space" (as opposed to "data space") [5]. Concretely, the SMOTE algorithm chooses *k* nearest neighbors for each data point based upon the similarities in feature vectors and creates a synthetic sample at a random point on the line between each data point and chosen neighbor [5]. See Figure 1 for a visualisation of SMOTE.
This technique boosts class membership whilst introducing enough variety to encourage classifier generalisation and grow minority class decision regions [5]. Since its release in 2002, SMOTE and its derivatives have had a profound impact on the way practitioners preprocess data [6] and is now the considered the "de facto" technique for learning from imbalanced datasets [6].

SMOTE has demonstrated meaningful results in many lower-dimensional problems, but class imbalance literature has increasingly found that it is a suboptimal choice for synthesis of samples in high-dimensional space [7, 8]. High-dimensional data types, including natural language and image data, are unable to be modeled by simple linear interpolation and require a different approach to create meaningful synthetic samples [7]. In the case of image synthesis, success has been found in "data-level" techniques such as image transformation where operations such as image rotations, cropping or scaling are used to boost existing datasets [9, 10]. However, there is a need for effective domain-agnostic "feature-level" techniques for synthetic oversampling in most high dimensional domains.

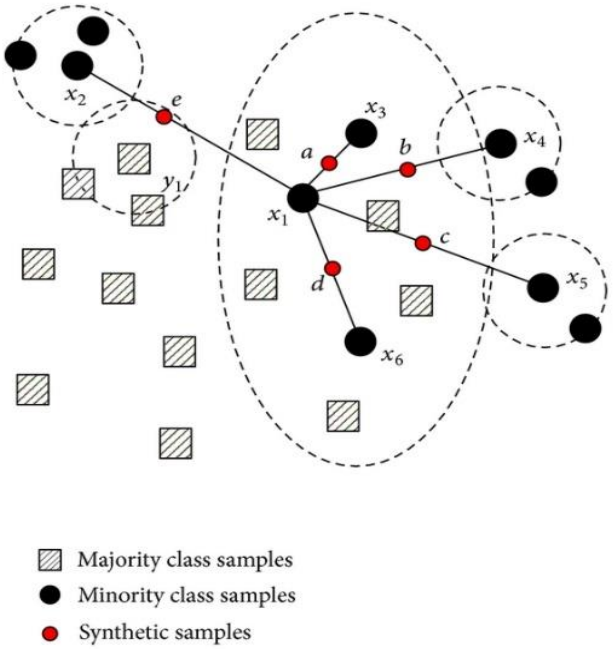


**Fig. 1.** SMOTE creates synthetic samples by interpolating between neighbouring minority class samples in "feature space" [11].

Generative Adversarial Networks (GANs) are a potential source of "feature-level" data augmentation for high dimensional data [12, 13, 14, 15]. The GAN architecture uses two neural networks in an adversarial set-up [12]**.** The first neural network, the generator, produces synthetic data samples from a random input vector [12]. The second neural network, the discriminator, is given both real and synthetic samples and must classify them as either real or synthetic [12]. The loss of the discriminator reflects the accuracy of its predictions, and the loss of the generator becomes the inverse of the loss of the discriminator [12]**.** In this way, the two networks compete in a zero-sum "minimax" game where one network's failure becomes the other network's gain [12]. The end result is that the generator is trained to produce realistic samples in order to "fool" the discriminator.

There is a growing number of publications demonstrating that a GAN-based approach to synthetic oversampling can improve performance in high-dimensional class-imbalance problems. Antinou et al. propose a Data Augmentation GAN (DAGAN) which produces augmented face images to increase classifier accuracy [13]. The authors demonstrate that adding additional augmented images with the original images during classification can significantly increase classification accuracy on a vanilla DenseNet classifier [13]. Variations of augmentation GANs have also been used to boost classification accuracy in the medical imaging domain [15] and create additional training images for radiology trainees [14]. While there exists a number of implementations of GAN-based synthetic oversampling for images, the literature lacks implementations that target other high-dimensional data types.

To add to this growing body of work, this paper implements a novel GAN-based approach to synthetic minority oversampling for sparse one-hot encoded data (GAN-SMOTE). It also introduces a number of techniques to ensure GAN performance when using a small dataset of binary values.

This technique is tested, as a proof of concept, on a dataset of petrographical descriptions of core samples from the North West Shelf in offshore Australia [16]. The class for prediction in this dataset is the sample's porosity, rated as either Very Poor, Poor, Fair or Good. The factors provided to determine the sample's porosity are detailed lithographical descriptions summarised into six characters: grain size, sorting, matrix, roundness, bioturbation and laminae. The data was provided in a sparse 'one-hot' encoded format, with a 1 in the relevant column when a sample has the associated attribute for a character and a 0 otherwise. There is not always one attribute per characteristic, with some samples having none or several.

The dataset is small, consisting of 140 samples in total. The distribution of the attributes is also quite uneven which can make classification difficult for outlier samples.

Our experiment demonstrates that the use of GAN-SMOTE leads to meaningful classifier performance increases on a vanilla neural network using this data compared to traditional techniques such as SMOTE or random oversampling.

# 2 Method

## 2.1 GAN-SMOTE Basic Architecture

The GAN-SMOTE architecture is based upon the original GAN architecture proposed in Goodfellow's 2014 paper 'Generative Adversarial Networks' [12].

GAN-SMOTE is made up of two neural networks, a generator and a discriminator. The generator is a 2 hidden layer neural network utilising leaky ReLU in each hidden layer. It takes 16 neurons of random input and produces one output neuron for each attribute of the data (in this case 58) plus an extra neuron to enable mini-batch discrimination. Our implementation of mini-batch discrimination is described in '2.3 Improving GAN-SMOTE Variance'.

The discriminator is a 2 hidden layer neural network utilising dropout and leaky ReLU in each of its hidden layers. It has the same number of input neurons as the final layer of the generator. A sigmoid output function creates an output between 0 and 1.

In every epoch, the generator produces synthetic datapoints. The discriminator is shown a set of real datapoints from a specific class of the data and predicts if the data points were made by the generator. The correct output in the first round should be a score of 1.0 for each data point. The results are used to calculate Mean Squared Error loss, which is backpropagated through the discriminator. The discriminator is then shown a set of synthetic datapoints created by the generator. In this round the target output is 0.0. The results are again used to calculate loss, which is backpropagated through the discriminator.

To calculate loss from the generator, the discriminator is given fake data produced by the generator and the loss for the generator is calculated based on a target output of 1.0. This decreases the loss for the generator when the discriminator does badly and increases it when the discriminator is more accurate. In effect, the two networks are engaged in a 'minimax game', where each is rewarded for the failure of the other [12]**.** Over many epochs of training, the discriminator improves its ability to recognise the real data samples, and the generator improves its ability to produce realistic synthetic data.

The GAN is trained on one class of the dataset at a time, and at the end of training the generator can be used to produce an arbitrary number of synthetic data samples for this class. In the case of the one-hot encoded petrographical dataset, this data was rounded up to 1 or down to 0 for consistency.

## 2.2 Improving GAN-SMOTE Stability

One of the most difficult tasks when training a GAN is ensuring that the network remains stable [17]. When creating the GAN-SMOTE architecture, a number of changes had to be made to ensure that the networks were competitive with one another.

The discriminator often out-performed the generator which caused the generator to stop improving. This is caused by an unchanging loss function which reflects the discriminator's 100% accuracy.

The first change that was made was the introduction of random noise to the input data. As a result of the one-hot encoding the input data is encoded as either 1.0 or 0.0, so any uncertainty in the generator made it obvious that the sample was synthetic. Because a rounding activation function has a gradient of 0, the output of the generator couldn't be coerced to produce only rounded numbers. As an alternative, the real data was altered with a random amount of noise between 0% and 2%.

The GAN-SMOTE architecture also relies on a novel GAN stabilisation technique for one-hot encoded input data that we've termed "decreasing random bit-flips". As each sample was an array of binary numbers, random noise alone won't prevent the discriminator from overfitting on the real data (the discriminator quickly learns that 0.97 is the same as 1.0). A predetermined proportion of the real data had the binary data reversed so that 1s became 0s and vice versa. This helps the generator remain competitive with the discriminator at the cost of the synthetic data's accuracy. The amount of random bit-flips were then slowly reduced to zero over the course of many epochs, allowing the generator to produce more accurate samples by the end of training. We found that this technique allowed the two networks to remain competitive for longer and improved the synthetic samples that were produced.

The learning rate was adjusted in line with "SGDR: Stochastic Gradient Descent with Warm Restarts" to prevent the networks from settling into local minima [18]. The learning rate was increased by a factor of 10 and degraded back to the original learning rate using cosine annealing over the course of 1000 epochs. This is repeated for every 1000 epoch interval. This technique was particularly useful for ensuring the generator moved past suboptimal local minima.

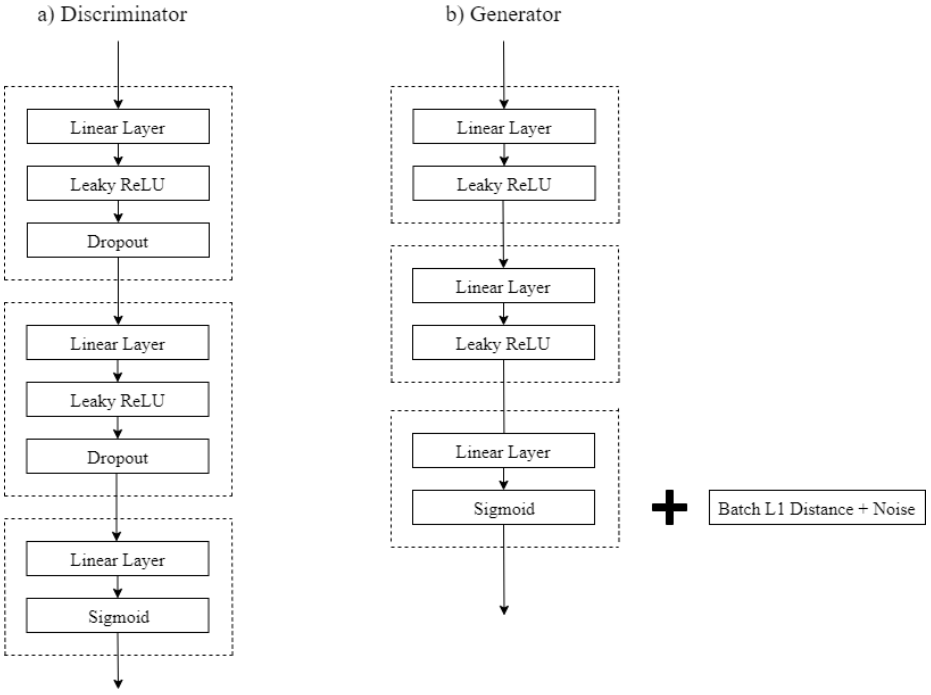


**Fig. 2.** GAN-SMOTE architecture diagram showing a) discriminator network and b) generator network. The networks are feedforward neural networks utilising Leaky ReLU activation functions in the hidden layers and sigmoid output functions. The discriminator network utilises dropout in its hidden layers. The generator concatenates the sum of L1 distances within the batch with a predetermined amount of noise to the output to implement minibatch discrimination as described in '2.3 Improving GAN-SMOTE Variance'.

### 2.3 Improving GAN-SMOTE Variance

The GAN-SMOTE architecture had issues with "mode collapse", where the generator ignores input and produces the same output each time. Minibatch discrimination was implemented as described in "Improved Techniques for Training GANs" [19]. The Manhattan distance was computed between each sample in a mini-batch and every other sample, and then the sum of these distances is appended to the output of the generator. This additional information about the distribution of the samples in the batch allows the discriminator to identify and penalise mode collapse and encourages the generator to achieve a similar distribution of data to the real samples. Because of the small amount of data, it was necessary to add a certain amount of random noise (30%) to each instance where the sum of distances was appended so that the generator remained competitive with the discriminator.

Optimising the GAN-SMOTE architecture required consideration of many different metrics to ensure the generator did not succumb to mode collapse. Deciding on optimal hyperparameters for GAN-SMOTE was possible by visualising these metrics over the course of training. Mode collapse could be quickly visualised by plotting the synthetic and real data using PCA. In order to adjust the hyperparameters for minibatch discrimination and decreasing random bit-flips we plotted the sum of Manhattan distances within synthetic and real data batches, as well as the sum of Manhattan distances between the two datasets. These metrics gave us insight into the synthetic data's variance and its similarity to the real data, which allowed us to determine the optimal levels for noise to be added to mini-batch discrimination and the schedule required for decreasing random bit-flips.

### 2.4 Benchmarking GAN-SMOTE performance

A simple neural network baseline was implemented. The network had one hidden layer of 100 neurons, a cross entropy loss function and a stochastic gradient descent optimiser. A constant weight initialisation seed was used to fix the starting point of the network. Early stopping was used to determine the ideal number of epochs.
10-fold cross-validation was used for the experiment. For each random fold, the training data was used to create GAN-SMOTE samples for each of the four classes.

To assess the performance of GAN-SMOTE, the synthetic data was used to "top up" class imbalances in training data.

Benchmark tests were performed using SMOTE and random oversampling to "top up" class imbalances. Finally, the original unbalanced data was used as a baseline.

The dataset was somewhat imbalanced. The four classes had 32, 37, 30 and 41 samples respectively. In the each "top up" experiment, the class was topped up to the maximum class size seen in that fold. The mean of the maximum accuracy scores was taken for each technique, and this set of values was used to compute the standard deviation.

## 3 Results and Discussion

The results of the benchmarking tests are shown in Table 1.

**Table 1.** Average test set accuracy with standard deviation, and F1 score on the original petrographical dataset.

| | GAN-SMOTE "top up" | SMOTE "top up" | Original training data | Basic oversampling "top up" |
|---|---|---|---|---|
| Average accuracy | **69.29 ± 12.16** | 68.57 ± 11.76 | 68.57 ± 15.13 | 67.86 ± 12.26 |
| F1 Score | **0.693** | 0.686 | 0.686 | 0.679 |

These results show a meaningful increase in training accuracy and F1 score when using GAN-SMOTE. The small decrease in accuracy seen with random oversampling compared to the unaltered data indicates that GAN-SMOTE's improved performance can be attributed to better modelling of the original data's "feature space".

Both GAN-SMOTE and SMOTE lowered the classifier's variance when compared to the unaltered data, however only GAN-SMOTE was able to improve both the accuracy and F1 score on this dataset.

## 4 Conclusion and Future Work

This paper proposed a novel method for dealing with class imbalance in one-hot encoded datasets, utilising a GAN to synthesise data for the purpose of oversampling minority classes. The benchmark tests on the synthesised one-hot encoded data suggests that GAN-SMOTE can increase classifier performance, in terms of both raw accuracy and class balanced performance metrics.

This paper also introduced techniques that are key for ensuring the stability and variance of GAN training with sparse one-hot encoded data. Decreasing random bit-flips is a novel approach to encourage GAN stability in a binary context. When combined with noising of the real data and stochastic gradient descent with restarts [18], this technique had a remarkable effect on GAN stability. Minibatch discrimination [19] was used to combat mode-collapse and allowed the generator to produce a realistic array of samples for minority oversampling.

There still remains a lot of work to discover the best practices when utilising the techniques applied in this paper. GANs are difficult to train, and techniques such as decreasing random bit-flips may apply in different degrees to other types of data. Many changes needed to be made to the GAN-SMOTE architecture to account for the one-hot encoded dataset, so a robust understanding of the raw data being fed into the GAN will allow data scientists to use this technique successfully and repeatably.

Although GAN-SMOTE did outperform SMOTE in this experiment, future work should investigate GAN-SMOTE's performance using much higher dimensional data in order to prove that GAN-SMOTE can significantly outperform traditional techniques in these classification problems.

Another area for investigation is the potential use of this technique to create synthetic data in scenarios with small datasets but without class imbalance. The improved training stability and reduced propensity to overfit derived from greater sample sizes can be a viable alternative to simply adding noise to training data because the variance produced by GAN-SMOTE is contextual. The technique appears to produce 'acceptable' variance while retaining key features, which is highly beneficial for datasets of high dimensionality such as this one.

The main benefit of developing new GAN-based oversampling techniques is to address SMOTE's weakness working with high-dimensional data [7, 8]. GANs have been successfully used to augment high-dimensional structured data such as images, so they have the credentials to do so in other similar domains. As GAN-based data augmentation techniques

are applied to a greater range of problems, they will benefit from the large volume of ongoing research occurring in the field of GAN optimisation.

## 5 References

[1] Perez L., Wang J.: The Effectiveness of Data Augmentation in Image Classification using Deep Learning. arXiv preprint arXiv: 1712.04621 (2017).

[2] Buda M., Maki A., Mazurowski MA.,: A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks 106, 249-259 (2018).

[3] Wasikowski M.: Combating the Class Imbalance Problem in Small Sample Data Sets. Thesis submitted to the Department of Electrical Engineering & Computer Science and the Graduate Faculty of the University of Kansas School of Engineering (2009).

[4] Yap BW., Rani KA., Rahman HAA., Fong S., Khairudin Z., Abdullah NN.: An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets. In: Proceedings of the First International Conference on Advanced Data and Information Engineering. 13-22 (2013).

[5] Chawla N., Bowyer K., Hall L., Kegelmeyer P.: SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16, 321-357 (2002).

[6] Fernández A., Garcia S., Herrera F., Chawla NV.: SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. Journal of Artificial Intelligence Research 61, 863-905 (2018).

[7] Bellinger C., Drummond C., Japkowicz.: Beyond the Boundaries of SMOTE: A Framework for Manifold-Based Synthetically Oversampling. In: ECML PKDD 2016 European Conference on Machine Learning and Knowledge Discovery in Databases. 248-263 (2016).

[8] Blagus R., Lusa A.: SMOTE for high-dimensional class-imbalanced data. BMC Bioinformatics 14, 106 (2013).

[9] Buslaev A., Parinov A., Khvedchenya E., Iglovikov VI., Kalinin AA. Albumentations: fast and flexible image augmentations. arXiv preprint arXiv: 1809.06839 (2018).

[10] Ratner AJ., Ehrenberg HR., Hussain Z., Dunnmon., Ré.: Learning to Compose Domain-Specific Transformation for Data Augmentation. arXiv preprint arXiv: 1709.01643 (2017).

[11] Hu F., Li H.: A Novel Boundary Oversampling Algorithm Based on Neighborhood Rough Set Model: NRSBoundary-SMOTE. Mathematical Problems in Engineering (2013).

[12] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.: Generative Adversarial Nets. Advances in neural information processing systems, 2672-2680 (2014).

[13] Antoniou A., Storkey A., Edwards H.: Data Augmentation Genrative Adversarial Networks. arXiv preprint arXiv: 1711.04340 (2018).

[14] Finlayson SG., Lee H., Kohan IS., Oakden-Rayner.: Toward generative adversarial networks as a new paradigm for radiology education. arXiv preprint arXiv: 1812.01547 (2018).

[15] Lima JLP., Macêdo D., Zanchettin C.: Heartbeat Anomaly Detection using Adversarial Oversampling. arXiv preprint arXiv: 1901.09972 (2019).

[16] Gedeon T, Tamhane D, Lin T, Wong P. Use of linguistic petrographical descriptions to characterise core porosity: contrasting approaches. Journal of Petroleum Science & Engineering 31, 193-199 (2001).

[17] Mescheder L., Geiger A., Nowozin S..: Which Training Methods for GANs do actually Converge? arXiv preprint arXiv: 1801.04406 (2018).

[18] Loshchilov I., Hutter F.: SGDR: Stochastic Gradient Descent with Warm Restarts. ILCR 2017 Conference Submission (2017).

[19] Salimans T., Goodfellow I., Zaremba W., Cheung V., Radford A., Chen X.: Improved Techniques for Training GANs. Proceedings of the 29th International Conference on Neural Information Processing Systems, 2234-2242 (2016).