Adversarial oversampling for multi-class imbalanced data classification with convolutional neural networks

Adam Wojciechowski AWOJCIECHOWSKI@MAN.POZNAN.PL
Poznan Supercomputing and Networking Center, Poznań, Poland
Faculty of Computing and Telecommunications, Poznan University of Technology, Poznań, Poland

Mateusz Lango MLANGO@CS.PUT.POZNAN.PL
Institute of Computer Science, Poznan University of Technology, Poznań, Poland

Editors: Nuno Moniz, Paula Branco, Luís Torgo, Nathalie Japkowicz, Michał Woźniak and Shuo Wang.

Abstract

Although many methods have been proposed for dealing with class imbalance, the problem of multi-class imbalanced classification still received significantly smaller attention. This problem is particularly important in image imbalanced classification since it has many critical applications, e.g., in the medical domain. One group of effective methods for imbalanced data are oversampling algorithms; however, they are usually not designed to work with image data. The current methods also work in separation from the learning algorithm, not considering the difficulties encountered during the training. In this work, we propose a new oversampling algorithm for neural networks that changes oversampled instances during training to further expand the decision region of minority classes, providing better recognition of minority classes. Experiments performed on various datasets with several configurations of class-imbalanced distributions demonstrate that the proposed method provides significant F-measure and G-mean improvements on imbalanced classification tasks.

Keywords: multi-class imbalanced data, image classification, adversarial examples, resampling methods

1. Introduction

Learning from imbalanced datasets still poses a significant challenge for modern machine learning methods (Branco et al., 2016). Such problems include one or several underrepresented classes, called minority classes, that usually are crucial from the application point of view (detecting rare disease vs. heathy patients). Unfortunately, such classes are difficult to learn for standard classifiers and, as a consequence, are poorly represented by the induced model. Since imbalanced problems occur in many important application domains, for instance in medicine (Wang et al., 2020) or in business analytics (Lango, 2019), the problem received significant research attention and many algorithms dealing with it has been proposed (He and Garcia, 2009; Fernández et al., 2018). Nevertheless, some problems remains open and require further investigation.

One of such problems is dealing with multi-class imbalanced datasets (Krawczyk, 2016), particularly in the high-dimensional settings that naturally occur while handling image datasets. In general, the number of proposed methods for learning from multi-class imbalanced datasets is relatively small, especially in the comparison to the large amount of methods for binary problems. The proposed algorithms can be roughly divided into decomposition methods that transform problems into a series of binary ones, and ad-hoc approaches that mostly include various resampling methods (Fernández et al., 2018). The resampling ad-hoc methods are the most popular ones both in research and in practice, since they are universal i.e. they can be applied together with virtually any learning algorithm, and at the same time they do not ignore the global view of the multi-class problem (e.g. they can take into account class interrelations).

Despite these advantages, the performance of standard resampling methods while training a deep neural network classifier on image data is rather limited (Buda et al., 2018). This can be partially attributed to the fact that standard methods based on SMOTE (Chawla et al., 2002) construct new data points as linear interpolations of existing ones, which in case of image data usually results in producing examples out of the distribution. Therefore, constructing such examples do not impact significantly the classification performance on minority classes. Recent resampling methods for image data (Zhang et al., 2020; Ali-Gombe and Elyan, 2019) try to alleviate this problem by producing new instances with deep generative models like Generative Adversarial Networks (GAN) (Goodfellow et al., 2016). However, such methods come with a significant computational overhead, since training deep generative models is computationally-intensive. Moreover, successful training of GAN itself is known to be problematic due to the existing challenges that include mode collapse, instability, etc. (Saxena and Cao, 2021). These challenges are especially prominent while training from small datasets that often arise in imbalanced domains like medicine.

Additionally, a recent systematic study of the difficulty of multi-class problems (Lango and Stefanowski, 2022) highlighted the importance of dealing with so-called data difficulty factors like class-overlapping or different class size configuration. Unfortunately, taking into account data difficulty factors in high-dimensional data is problematic since most methods detects them leveraging k-nearest neighbours or local kernel estimation (Stefanowski, 2013) that suffer from the curse of high-dimensionality. Another arising issue is that influential data difficulty factors like class-overlapping depend on the example's position in the feature space, which, in the case of training neural networks, is automatically constructed by the model and constantly changes during training. For instance, an example can be in the class-overlapping region in the original space, but be far away from the decision boundary in the constructed hidden space.

In this paper, we present a method for training deep neural networks on multi-class imbalanced data, called Adversarial OverSampling (AOS), that successfully produces new instances of minority classes that improve the recognition of images from underrepresented classes by a neural network. To generate new examples, the method leverages Fast Gradient Sign Method (FGSM) proposed by Goodfellow et al. (2015) as a method of attacking machine learning systems with spurious images that are consistently misclassified. In this work, FGSM is utilized in a new context of multi-class imbalanced data and generating minority instances in the unsafe regions of the feature space. The AOS method actively oversamples minority instances, adapting its working according to difficulty factors in the examples' hidden space that changes during network training. Contrary to the other imbalanced methods for deep neural networks based on GANs, AOS do not require a cost-intensive training of additional deep neural model and can be easily integrated in the training loop of any classifier's architecture. Despite the fact that AOS uses a rather basic method of generating adversarial examples, the performed experimental evaluation demonstrated that it obtains better results in terms of G-mean and F-score measures than other popular resampling methods like random oversampling, random undersampling and SMOTE.

2. Related works

A dataset is imbalanced when the numbers of examples representing each class are not equal. However, when talking about learning from imbalanced data, we have in mind datasets in which disproportions between class sizes are significant (He and Garcia, 2009). The level of class imbalance can be easily measured with *imbalance ratio* defined as a ratio between the size of the largest majority class and the number of examples in the smallest minority class:

$$\rho = \frac{\max_i\{|C_i|\}}{\min_i\{|C_i|\}}$$

where C is a set of all classes and $|C_i|$ denotes i-th class cardinality. In the multi-class setting, the imbalanced datasets are additionally characterized by class sizes configuration (e.g. whether they contain intermediate classes, many majority classes or many minority classes etc.). One very simplistic indicator of class configuration is the fraction of classes being minority ones (Buda et al., 2018):

$$\mu = \frac{\sum_{i\in\{1,2,\ldots,|C|\}} \mathbb{1}_{[C_i \text{ is a minority class}]}}{|C|}$$

where C is a set of all classes and $\mathbb{1}$ is the indicator function.

As mentioned in the introduction, the number of proposed methods for imbalanced classification is very large and making even a short review of them is out of scope of the current paper. We refer the interested reader to one of the excellent reviews of the field (He and Garcia, 2009; Branco et al., 2016; Fernández et al., 2018) and limit this chapter to a brief description of the methods later used in the experiments or being directly related to the current research.

Resampling methods for imbalanced data can be roughly divided into random resampling and informed resampling methods. One of the most popular random methods is random undersampling (RUS) that randomly removes majority instances from the dataset until the cardinalities of all classes are equal. Another technique is random oversampling (ROS) that duplicates randomly selected minority instances until obtaining a fully balanced distribution. In the context of image classification and convolutional neural networks, both these methods were investigated by Buda et al. (2018). The study revealed that random resampling methods offer improvement over the baseline neural network. It was also demonstrated that while working with image data and deep classifiers, ROS usually obtains better results than RUS. Only in the most extremely imbalanced cases, RUS and ROS achieved similar classification performance.

The most prominent example of informed resampling methods is SMOTE (Chawla et al., 2002) which is an oversampling method that constructs new instances by taking a linear interpolation of two randomly selected minority class instances. Even though it was originally proposed for binary problems, its simple extension that iteratively runs SMOTE for each minority class is also used for multi-class data (Fernández-Navarro et al., 2011). The usefulness of SMOTE method was also confirmed in the context of convolutional neural networks and image classification by several earlier works (Gao et al., 2019; Özdemir et al., 2021), however, its performance can be limited since a linear interpolation of two images, unlike in the case of tabular data, usually result in an example that is out of the data distribution.

More recent works on imbalanced image classification try to solve this problem by training an additional deep generative model of the data and using it to generate new instances. An example of such method is BAGAN (Mariani et al., 2018) that first trains an autoencoder and use it to initialize weights in a GAN model. Later, the GAN model is trained and used to generate minority class instances. BAGAN was also further extended to a BAGAN-GP (Huang and Jafari, 2021) which adopts additional gradient penalty and different autoencoder initialization. Another method is MFC-GAN (Ali-Gombe and Elyan, 2019) that trains a GAN model with additional fake classes to oversample the dataset. Despite being effective, these methods come with a significant additional training cost: they require training one or two deep learning models that serve to generate additional data.

3. Adversarial OverSampling

In this work, we explore another possibility to oversample image data through generating adversarial examples using machine learning attack methods that do not require training an additional learning model.

The idea behind AOS algorithm is best explained when considering images as points in a data space. Consider other oversampling methods like Random Oversampling that copies and pastes examples from minority classes into the existing dataset. In the data space, it results in adding data points exactly in the same places as existing ones. We end up with the same empirical class' data distribution with only the prior class probability changed. Random Oversampling has no means to alter the topology of data points. Contrary to ROS, algorithms from SMOTE family do change the topology of original data points by producing new examples as linear interpolations. However, new examples can never be produced outside the convex hull of minority class data points. SMOTE, therefore, only intensifies representation of minority class in its convex hull but is not able to increase its area.

Our proposed approach influences the model's decision boundary more directly. It extorts the change of the decision boundary by generating exogenous minority examples placed closer to the decision boundary than existing examples, or even exceeding it, i.e. pushing additional minority examples towards majority ones in the class overlapping regions. Such behaviour results in forcing the classifier to move the decision boundary away from the minority class examples, possibly assigning more feature space to the minority class. Subsequently, it leads to classification performance gains since in the case of imbalanced data, the decision boundary of the model is expected to be falsely misplaced towards minority class concepts (Wallace et al., 2011). In comparison to the aforementioned methods, AOS is capable of enlarging the area designated by convex hulls of the minority class data points.

In order to achieve it, AOS alters existing minority examples in a specific way. First, it is seeking the direction towards the currently induced decision boundary for every minority example. Later, a selected portion of minority instances is moved according to the computed vectors. More concretely, AOS employs Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015) to alter examples with a gradient of loss function optimized during classifier training with respect to a given example. Such gradient indicates the direction of the fastest local increase of loss function, which in practice means that the examples moved according to it will have higher probability of misclassification, i.e., they will be relatively closer to the decision boundary. Since FGSM is a method of attacking machine learning models, increasing the number of misclassified examples, many of such produced minority examples will effectively be in the region dominated by majority class examples. AOS method has two configurable parameters. The length of the translation vector that is applied to minority examples is controlled by ϵ parameter and roughly corresponds to the degree of the aggressiveness of influencing decision boundary of the model. Another parameter is τ which controls the percentage of minority instances that will be altered according to the translation vector.

The pseudocode of AOS is presented in Algorithm 1. The procedure starts with an imbalanced dataset X, which is oversampled with standard Random Oversampling (line 1). This step is performed because at the beginning of training the network weights are randomly initialized and generation of new samples basing on completely noisy hidden space will not be beneficial. Additionally, this step provides a fully balanced dataset which will be modified in an online fashion during training. Datasets with new instances effectively will be copies of randomly oversampled data with a portion of examples modified with a non-random noise.

In every epoch of neural network training, standard actions like forward-pass, back-propagation and optimization of the model's parameters are performed (lines 15-17). However, every second epoch[1], AOS performs a sampling procedure that alters the oversampled datasets (lines 3-14). The algorithm iterates over examples in all minority classes and with probability τ (method's parameter) decides whether the given image will be altered in the current training loop iteration (line 7). For each selected image, an adversarial mask is computed as the gradient of loss function with respect to the original image (line 9). The computed mask is multiplied by ϵ to control the mask intensity and added to the original image (line 10). More formally, a new synthetic example is computed according to the following FGSM formula:

$$x' = x + \epsilon \cdot sign(\nabla_x J(NN_w, x, y))$$

where x is the original image, sampled from a minority class y; ϵ is a parameter controlling masking intensity NN_w are the neural network's parameters; J is the optimized loss function (e.g. cross-entropy).

4. Experiments

In this chapter, we present experiments performed with convolutional neural networks in order to investigate the effectiveness of AOS in handling multi-class imbalanced image datasets.

[1] The procedure is not performed every epoch to ensure more stable training by give the optimizer more time to exploit newly constructed examples.

Algorithm 1: Adversarial OverSampling (AOS)

Input: X - dataset, C_{min} - set of minority classes, $NoEpochs$ - number of traning epochs, ϵ, τ - parameters of the method

Output: NN, the trained neural network

```
$X \leftarrow RandomOversampling(X)$
for $epoch \leftarrow 0$ to $NoEpochs - 1$ do
    if $epoch \% 2 = 0$ then
        $aos\_X \leftarrow X$
        for $x \in aos\_X$ do
            if $C(x) \in C_{min}$ then
                decision $\leftarrow$ draw($\mathcal{X} \sim \mathcal{B}(1, \tau)$)
                if $decision{=}1$ then
                    $mask \leftarrow sign(\nabla_x J(NN_w, X, Y))$
                    $x \leftarrow x + \epsilon \cdot mask$
                end
            end
        end
    end
    NN.forward(aos_X)
    NN.backpropagate(aos_X)
    NN.optimization_step()
end
```

4.1. Experimental setup

We have compared the performance of AOS with three other resampling algorithms that were used in the literature in the context of image classification: Random Oversampling (ROS), Random Undersampling (RUS) and SMOTE[2]. Additionally, we also report results on the imbalanced dataset without any oversampling or undersampling method applied (BASELINE). We have excluded from the experiments methods that require training additional deep generative models as they are much more computationally heavy than AOS or other standard methods.

We conduct experiments on four different image datasets, which are diverse with respect to the number of classes, image resolution or color format. The datasets are the following:

- CIFAR-10: CIFAR-10 is a more difficult dataset that consists of images of every day objects like boats or planes. It has 10 classes. The images are of size 32x32x3 and are of RGB format. The dataset has 50k images for training, which gives approximately 5k images per class.

- Intel Image Classification: dataset provided by Intel, which contains images of natural sceneries. It has 6 classes. The images are of size 150x150x3 and are of RGB format. The dataset has 14k images for training, which gives approximately 2.3k images per class.

- MIT Indoor Scene Recognition: dataset provided by Massachusetts Institute of Technology, which contains photos of many different indoor scenes. It has 67 classes. The images are of size 240x240x3 and are of RGB format. The dataset has 15620 images for training with at least 100 images per class.

- MNIST: MNIST is a classic dataset for image classification, it consists of images of digits with 10 classes corresponding to 10 digits. The images are of size 28x28 and are gray-scale. The dataset has 60k images for training, which gives approximately 6k images per class.

Since most of the presented datasets are balanced, in the experiments we have used modified versions of them that are class imbalanced. Additionally, to better evaluate the area of competence of the algorithm, we have prepared datasets with different numbers of minority classes. Therefore, the problems in the resulting collection of datasets are ranging from multi-minority to multi-majority problems, where the percentage of minority classes is denoted by μ.

The procedure of artificially inducing class imbalance with the imbalance ratio ρ is the following:

1. Find the class with the lowest cardinality among all the classes in the dataset.

2. Randomly delete images from all the other classes, until all class cardinalities are equal to the cardinality of class selected in step 1.

3. Assuming $|C|$ is the number of classes, choose $\lfloor \mu |C| \rfloor$ classes randomly and mark them as minority classes. Mark the rest of the classes as majority ones.

[2] Implementation of SMOTE from imbalanced-learn package was used with default parameters.

4. Randomly delete examples in every minority class, until the ratio between the minority class cardinality and majority class cardinality is equal to ρ.

For each original dataset, we created four artificially imbalanced datasets with $\mu = \{0.2, 0.4, 0, 6, 0.8\}$, where $\mu = 0.2$ variation might be considered as the most multi-majority one (20% of all classes are the minority classes) and $\mu = 0.8$ might be considered as the most multi-minority setup (80% of all classes are the minority classes). The datasets constructed from the same original dataset have the same imbalanced ratio ρ, however, ρ for each original dataset were selected to different values. Our goal was to construct datasets with as high class imbalance as possible to better observe performance differences between methods handling imbalance. However, to ensure stable neural network training, we picked high ρ values that leave each class with a sufficient number of observations (50-100). Therefore, we picked larger imbalanced ratios for datasets with originally large classes and smaller imbalanced ratio for these with smaller classes. The specific parameters of the datasets, including selected values of ρ, can be found in Table 2.

As a base classifier, we use standard convolutional neural networks build from several blocks of convolutional layers, Batch Normalization layers, MaxPooling layers and ended with a softmax layer. As the activation function, a very popular ReLU was used. In order to provide a fair comparison for every algorithm, each model for a given dataset was trained with the same number of epochs. Later, the models' weights from the epoch with the highest f1-score calculated on the validation set was saved and used for final evaluation on a test set. We did not employ either learning rate decay or any form of additional regularization like weight decay or Dropout. The models' parameter counts were adjusted proportionally to the dataset size[3] and the exemplary network architecture for CIFAR-10 dataset can be found in Table 1. The network was trained by Adam optimizer with standard cross-entropy as the loss function.

All reported results are computer on out-of-sample, test set data. For Intel, MNIST and CIFAR-10 train-test set splits provided by the authors of the datasets were utilized. For the MIT dataset we have performed a stratified random split, ensuring the same class distributions in train, validation and test sets. Every test set was used in its original form and was not artificially made imbalanced. All results were averaged over 3 independent runs.

4.2. Results

In this section, we present the results of our experiments measured with two commonly used metrics for imbalanced data: macro-averaged F-score and G-mean. The results of F-score can be found in the Table 3 and the results of G-mean measure can be found in the Table 5. For every dataset, we added a row named *Avg.*, which is a mean value over all μ variants for a given dataset and method. The bottom-most row named *Global Avg.* is a mean value over all datasets for a given method.

We also present the number of wins/loses for each pair of methods in Table 4 for *f1-score* and in Table 6 for *G-mean*.

[3] Details on network architectures for all datasets can be found at the following link: https://www.cs.put.poznan.pl/mlango/publications/aos.pdf

Table 1: Neural network architecture for CIFAR-10 dataset

Layer	**Dimensions**	**Kernel size**	**Stride**
Input	(32x32x3)	-	-
Convolution	(32x32x32)	(3x3)	(1x1)
Batch Normalization	(32x32x32)	-	-
Convolution	(30x30x32)	(3x3)	(1x1)
Batch Normalization	(30x30x32)	-	-
Max pooling	(15x15x32)	(2x2)	(2x2)
Convolution	(15x15x64)	(3x3)	(1x1)
Batch Normalization	(15x15x64)	-	-
Convolution	(13x13x64)	(3x3)	(1x1)
Batch Normalization	(13x13x64)	-	-
Max pooling	(6x6x64)	(2x2)	(2x2)
Fully Connected	(1x1x10)	-	-

Table 2: List of tested dataset variants (left) along with the parameters used for network training (τ - AOS percentage of altered instances, LR - learning rate, epochs - number of epoches)

Dataset	ρ	μ	τ	LR	epochs
CIFAR-10	50	0.2	0.4	10^{-4}	20
	50	0.4	0.4	10^{-4}	20
	50	0.6	0.4	10^{-4}	20
	50	0.8	0.4	10^{-4}	20
INTEL	20	0.2	0.6	10^{-3}	35
	20	0.4	0.8	10^{-3}	35
	20	0.6	0.8	10^{-3}	35
	20	0.8	0.8	10^{-3}	35
MIT	2	0.2	0.6	10^{-3}	60
	2	0.4	0.8	10^{-3}	60
	2	0.6	0.6	10^{-3}	60
	2	0.8	0.6	10^{-3}	60
MNIST	60	0.2	0.4	10^{-4}	20
	60	0.4	0.4	10^{-4}	20
	60	0.6	0.8	10^{-4}	20
	60	0.8	0.6	10^{-4}	20

Table 3: The results of F1-score measure for different datasets and preprocessing methods.

Dataset	ρ	μ	AOS	SMOTE	ROS	RUS	BASELINE
CIFAR-10	50	0.2	0.6023	0.5948	0.5929	0.0802	0.6007
		0.4	0.5116	0.4623	0.4695	0.0795	0.4536
		0.6	0.4294	0.3813	0.3741	0.0849	0.3345
		0.8	0.3748	0.3415	0.3666	0.0969	0.2986
		Avg.	**0.4795**	**0.4450**	**0.4508**	**0.0854**	**0.4219**
INTEL	20	0.2	0.7844	0.7746	0.7868	0.1634	0.7601
		0.4	0.7369	0.7261	0.7166	0.1666	0.7268
		0.6	0.6633	0.6414	0.6248	0.1415	0.6132
		0.8	0.6516	0.6564	0.6456	0.1569	0.6112
		Avg.	**0.7090**	**0.6996**	**0.6934**	**0.1571**	**0.6778**
MIT	2	0.2	0.3154	0.3191	0.3038	0.2761	0.3148
		0.4	0.2983	0.2863	0.2768	0.2503	0.2726
		0.6	0.2860	0.2993	0.2738	0.2618	0.2729
		0.8	0.2810	0.2605	0.2635	0.2662	0.2709
		Avg.	**0.2950**	**0.2913**	**0.2795**	**0.2636**	**0.2828**
MNIST	60	0.2	0.9824	0.9806	0.9808	0.9133	0.9769
		0.4	0.9748	0.9726	0.9732	0.9160	0.9669
		0.6	0.9641	0.9574	0.9583	0.9195	0.9481
		0.8	0.9550	0.9484	0.9548	0.9203	0.9456
		Avg.	**0.9691**	**0.9648**	**0.9668**	**0.9173**	**0.9594**
Global Avg.			**0.6132**	**0.6002**	**0.5976**	**0.3558**	**0.5855**

Table 4: The number of wins/loses while comparing algorithms pairwise on F1-score

	AOS	SMOTE	ROS	RUS	BASELINE
AOS	-	13/3	15/1	16/0	16/0
SMOTE	3/13	-	8/8	16/0	13/3
ROS	15/1	8/8	-	15/1	12/4
RUS	0/16	0/16	1/15	-	0/16
BASELINE	0/16	3/13	4/12	16/0	-

Table 5: The results of G-mean measure for different datasets and preprocessing methods.

Dataset	ρ	μ	AOS	SMOTE	ROS	RUS	BASELINE
CIFAR-10	50	0.2	0.5169	0.5115	0.5015	0.0059	0.4901
		0.4	0.4467	0.3806	0.4030	0.0000	0.3252
		0.6	0.3390	0.2880	0.2761	0.0000	0.1907
		0.8	0.3005	0.2533	0.2775	0.0000	0.1651
		Avg.	**0.4008**	**0.3583**	**0.3645**	**0.0015**	**0.2928**
INTEL	20	0.2	0.7756	0.7596	0.7789	0.0028	0.7475
		0.4	0.7226	0.7116	0.6926	0.0270	0.7112
		0.6	0.6446	0.6075	0.5895	0.0064	0.5829
		0.8	0.6257	0.6413	0.6237	0.0024	0.5831
		Avg.	**0.6921**	**0.6800**	**0.6712**	**0.0096**	**0.6562**
MIT	2	0.2	0.2910	0.2833	0.2802	0.2528	0.2792
		0.4	0.2718	0.2643	0.2499	0.2334	0.2425
		0.6	0.2617	0.2721	0.2475	0.2409	0.2488
		0.8	0.2604	0.2348	0.2316	0.2497	0.2419
		Avg.	**0.2712**	**0.2636**	**0.2523**	**0.2442**	**0.2531**
MNIST	60	0.2	0.9823	0.9804	0.9806	0.9127	0.9765
		0.4	0.9745	0.9722	0.9728	0.9150	0.9661
		0.6	0.9635	0.9565	0.9575	0.9189	0.9464
		0.8	0.9545	0.9477	0.9543	0.9195	0.9447
		Avg.	**0.9687**	**0.9642**	**0.9663**	**0.9165**	**0.9584**
Global Avg.			**0.5832**	**0.5665**	**0.5636**	**0.2930**	**0.5401**

Table 6: The number of wins/loses while comparing algorithms pairwise on G-mean

	AOS	SMOTE	ROS	RUS	BASELINE
AOS	-	14/2	15/1	16/0	16/0
SMOTE	2/14	-	9/7	15/1	15/1
ROS	15/1	9/7	-	15/1	13/3
RUS	0/16	1/15	1/15	-	1/15
BASELINE	1/15	3/13	3/13	15/1	-

Starting the analysis from the *Global Avg.* values, we can see that on average the AOS algorithm yields the best results for both metrics, followed by SMOTE and ROS. RUS is the only imbalanced learning method studied in the experiment that, on average, does not achieve higher results than the baseline. Considering averages over results on a given dataset, AOS is always better than the next best method which oscillates between SMOTE and ROS, depending on the dataset. AOS is also substantially better than the baseline results for both metrics. On some dataset configurations, AOS offer very significant improvements, up to 5% on both metrics in comparison to the second-best method. It also seems that the advantage of AOS over SMOTE/ROS grows as μ is getting larger, i.e. the problem is having more minority classes. Note that multi-minority datasets are considered to be more difficult than multi-majority ones (Wang and Yao, 2012; Lango and Stefanowski, 2022).

By analyzing the numbers of win/loses for both metrics, we can see that AOS is the only method that yielded better results than the baseline approach 100% of the time. This could indicate, that our proposed method is the most flexible one respective to the wide range of μ parameters we have tested and can be treated as the most universal approach. We can also notice, that AOS has the biggest win count (summarized number of wins against every other competitor method) among all of the tested methods. The biggest competitor is SMOTE, but it provides better results than AOS only for 3 among 16 datasets for F-score and only for 2 for G-mean. The comparison of informed SMOTE with random ROS is also interesting, since both methods obtain (almost) equal number of wins for both metrics. This provides an evidence that using SMOTE on image data is not as effective for standard tabular data, possibly for the reasons discussed earlier in the paper.

5. Summary

This work explores the possibility of using machine learning attack methods to alleviate the issue of multi-class imbalance. We have presented Adversarial OverSampling, a new technique for oversampling imbalanced image datasets while training a deep neural network classifier. The method can be applied with any backpropagable network architecture and generates minority data close to or even behind the decision boundary, enlarging the feature space for minority classes. Conducted experiments demonstrated that the method achieves better results than other resampling methods used in image classification that do not train additional deep model for sampling.

The presented work can be further extended. First, AOS use the most basic method of generating adversarial examples called FGSM which can be replaced with more advanced methods like, for instance, DeepFool (Moosavi-Dezfooli et al., 2016) or C&W attack (Carlini and Wagner, 2017) which could lead to better performance of trained network. Moreover, currently the AOS method generates new instances basing on randomly selected minority instances which can be suboptimal. Future research can explore the possibilities of generating more instances basing on most unsafe minority examples e.g. these lying in the class overlapping area.

Acknowledgments

Adam Wojciechowski was supported by AI Tech project sponsored from POPC programme POPC.03.02.00-00-0001/20. Mateusz Lango was supported by the Polish National Science Center under grant No. 2016/22/E/ ST6/00299.

References

Adamu Ali-Gombe and Eyad Elyan. Mfc-gan: Class-imbalanced dataset classification using multiple fake class generative adversarial network. *Neurocomputing*, 361:212–221, 2019. ISSN 0925-2312.

Paula Branco, Luís Torgo, and Rita P. Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Comput. Surv.*, 49(2), August 2016. ISSN 0360-0300. doi: 10.1145/2907070. URL https://doi.org/10.1145/2907070.

Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. doi: 10.1109/SP.2017.49.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, pages 321–357, 2002.

Alberto Fernández, Salvador García, Mikel Galar, Ronaldo Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from Imbalanced Data Sets*. Springer, 2018.

Francisco Fernández-Navarro, César Hervás-Martínez, and Pedro Antonio Gutiérrez. A dynamic over-sampling procedure based on sensitivity for multi-class problems. *Pattern Recognition*, 44(8):1821 – 1833, 2011. ISSN 0031-3203.

Ruihan Gao, Jiawei Peng, Long Nguyen, Yunfeng Liang, Steven Thng, and Zhiping Lin. Classification of non-tumorous facial pigmentation disorders using deep learning and smote. In *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2019.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

Gaofeng Huang and Amir Jafari. Enhanced balancing gan: minority-class image generation. *Neural Computing and Applications*, 06 2021. doi: 10.1007/s00521-021-06163-8.

Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

Mateusz Lango. Tackling the problem of class imbalance in multi-class sentiment classification: An experimental study. *Foundations of Computing and Decision Sciences*, 44(2): 151–178, 2019.

Mateusz Lango and Jerzy Stefanowski. What makes multi-class imbalanced problems difficult? an experimental study. *Expert Syst. Appl.*, 199(C), aug 2022. ISSN 0957-4174. doi: 10.1016/j.eswa.2022.116962. URL https://doi.org/10.1016/j.eswa.2022.116962.

Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi. Bagan: Data augmentation with balancing gan, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016. doi: 10.1109/CVPR.2016.282.

Divya Saxena and Jiannong Cao. Generative adversarial networks (gans): Challenges, solutions, and future directions. *ACM Comput. Surv.*, 54(3), may 2021. ISSN 0360-0300.

Jerzy Stefanowski. Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data. In *Emerging paradigms in machine learning*, pages 277–306. Springer, 2013.

Byron C. Wallace, Kevin Small, Carla E. Brodley, and Thomas A. Trikalinos. Class imbalance, redux. In *11th International Conference on Data Mining (ICDM)*, pages 754–763. IEEE, 2011.

Linda Wang, Zhong Qiu Lin, and Alexander Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, 10(1):1–12, 2020.

Shuo Wang and Xin Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4): 1119–1130, 2012.

Wei Zhang, Xiang Li, Xiao-Dong Jia, Hui Ma, Zhong Luo, and Xu Li. Machinery fault diagnosis with imbalanced data using deep generative adversarial networks. *Measurement*, 152:107377, 2020. ISSN 0263-2241.

Akın Özdemir, Kemal Polat, and Adi Alhudhaif. Classification of imbalanced hyperspectral images using smote-based deep learning methods. *Expert Systems with Applications*, 178: 114986, 2021. ISSN 0957-4174.