Using WGAN for Improving Imbalanced Classification Performance

Snehal Bhatia & Rozenn Dahyot

School of Computer Science and Statistics
Trinity College Dublin, Ireland
{bhatias,dahyotr}@tcd.ie

Abstract. This paper investigates data synthesis with a Generative Adversarial Network (GAN) for augmenting the amount of data used for training classifiers (in supervised learning) to compensate for class imbalance (when the classes are not represented equally by the same number of training samples). Our data synthesis approach with GAN is compared with data augmentation in the context of image classification. Our experimental results show encouraging results in comparison to standard data augmentation schemes based on image transforms.

1 Introduction

Image classification is a standard process in image processing and many machine (deep) learning techniques are routinely evaluated [14] using labelled images datasets (e.g. FMNIST [16], CIFAR-10 [6]). The occurrence of imbalance in datasets collected from real-life domains is sometimes unavoidable due to the cost of collecting and labelling data, or due to privacy issue, or simply due to rare event scenarios. This imbalance of number of training examples per class often has a detrimental effect on the performance of classifiers [9]. In this research we explore the potential of data synthesis with GANs to address this issue, more specifically we compare data synthesis with the Wasserstein Generative Adversarial Networks (WGANs) against the traditional data augmentation approach traditionally used for training deep learning architectures. GANs and WGANs are first introduced in Section 2, and our approach is introduced next (Sec. 3). We show experimentally that WGAN data augmentation outperforms traditional data augmentation with image transforms on both FMNIST and CIFAR-10 datasets (Sec. 4).

2 Related Work

The Generative Adversarial Network (GANs) framework was first introduced for artificially generating realistic images from scratch [3]. Since then, GANs have been employed for a variety of image processing and computer vision tasks, such as generating high resolution images from low resolution input images [7], texture synthesis in images [8] and human face synthesis [5].

This generative capacity of GANs makes them suitable for the purpose of data augmentation, and many recent studies have shown how to tackle the problem of imbalanced datasets in classification by using variations of the GAN architecture. Mariani et al [9] introduced a new architecture called "BAGAN" or Balancing GAN, and achieved a significantly better classification performance in comparison to ACGAN [11] and simple GAN [3], when tested on various artificially imbalanced distributions of the image datasets of MNIST, CIFAR-10, Flowers and GTSRB datasets. Mariani et al [9]'s original work with GAN (introduced Paragraph 2.1) is extended in our paper by evaluating the performance of Wasserstein GAN (WGAN and WGAN-GP [2, 4], explained in Paragraph 2.2) with the same methodology (Sec. 3 and 4).

2.1 Generative Adversarial Networks

The first and original GAN architecture introduced by Ian Goodfellow et al [3] consists of two sub-networks, which are competing Artificial Neural Networks (ANNs), namely the 'Generator' (G) and the 'Discriminator' (D). The Generator learns to transfer the distribution of input data (e.g. distribution of noise images) into a target distribution (e.g. distribution of images of horses). At the same time, the Discriminator, which is essentially a binary classifier, is trained to distinguish between the real data points (e.g. real images of horses) and artificially generated data points by the generator (e.g. synthesised images of horses created by the generator transfer function).

A GAN [3] is often defined as a two-player minimax game in which Generator G wants to minimize the cost function whereas the discriminator D aims to maximize it:

$$\min_G \max_D \mathbb{E}_{x\sim p_r}[\log D(x)] + \mathbb{E}_{\tilde{x}\sim p_g}[\log(1 - D(\tilde{x})] \tag{1}$$

where $\tilde{x} = G(z)$ with $z \sim p(z)$ (input of the generator sampled from a simple noise distribution such as Uniform or Normal), and p_r is the (real) data distribution [4]. The GAN value function (Equation (1)) is essentially the Jensen-Shanon Divergence between the real data and the artificially generated data. The training process of a GAN can be viewed as a double feedback-loop where the discriminator is in a feedback loop with the real images and the generator uses the feedback from the discriminator to learn how to produce images that are realistic enough to fool the discriminator. The feedback process is repeated throughout the training phase, until Nash Equilibrium is achieved[1]. This process is called Adversarial Training.

Although Vanilla GANs have achieved state of the art results in many domains, they suffer from certain drawbacks:

[1] In Game Theory, when multiple interacting, non-cooperating participants are involved, Nash Equilibrium is the state of stability achieved when no participant can benefit solely from changing its own strategy or actions if the other players' strategies remain constant.

– ***Nash Equilibrium is Hard to Achieve:*** The training process of GAN is based on gradient descent. The two models, generator and discriminator, are simultaneously trained to find a Nash Equilibrium. However, since both models update their loss functions concurrently and independently, there is no guarantee of convergence

– ***Vanishing Gradient Problem:*** Training a GAN loss function poses a dilemma. If the discriminator is trained perfectly (especially early on in the training process), then $\boldsymbol{D(x_{real})=1}$ and $\mathbf{D(x_{real})=0}$. From Equation (1) it can be observed that in this case, the value of the loss function would become 0, and there would be no gradient left to update during the training iterations, hence leading to the vanishing gradient problem. However, if the discriminator function is not trained to perfection then the generator would not receive relevant feedback, meaning that the learned loss function would not be good enough to generate realistic images

– ***Mode Collapse:*** This is a common failure observed in GANs where the generator achieves a state where it always produces the same images as outputs. This may in some cases be enough to fool the discriminator, but the low variety of images generated is not representative of the complexities observed in real-world data distribution [1]

Therefore, various modifications of the original GAN have been proposed, one of which is introduced below.

2.2 Wasserstein GAN (WGAN) and WGAN-GP

The WGAN architecture proposed by Arjovsky et al [2] replaces the Jensen-Shannon divergence from the original GAN architecture [3] with the Wasserstein Distance function. The Wasserstein Distance is also called the "Earth Mover Distance", as it can be informally interpreted as the minimal cost of moving and transforming some quantity of mass (say, a pile of dirt) from the shape of one probability distribution P to that of another probability distribution Q. The cost of moving in this scenario is calculated as the product of the amount of mass moved and the distance by which it has been moved. $W(P, Q)$ which is a measure of distance between the points in probability distributions P and Q. The WGAN value function corresponds to:

$$\min_{G} \max_{D \in \mathcal{D}} \mathbb{E}_{x \sim p_r}[D(x)] - \mathbb{E}_{\tilde{x} \sim p_g}[(D(\tilde{x})] \tag{2}$$

In Equation (2), $\mathcal{D}$ denotes the set of 1-Lipschitz functions[2], meaning that the discriminator loss should follow the Lipschitz constraint. Gulrajani et al [4] extended WGAN to WGAN-GP (gradient penalty) to improve the training of WGANs. The Weight-Clipping method used to enforce the Lipschitz Constraint introduces numerous problems such as vanishing gradient (when the clipping

[2] A Lipschitz function is a function $\boldsymbol{f}$ such that $\boldsymbol{|f(x)-f(y)| \leq K|x-y|}$ for all $\boldsymbol{x}$ and $\boldsymbol{y}$, where $\boldsymbol{K}$ is a constant independent of $\boldsymbol{x}$ and $\boldsymbol{y}$.

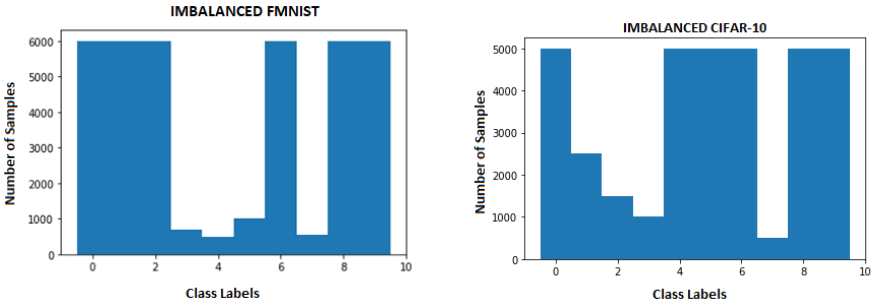
window is too large), slow convergence (when the clipping window is too small) and it is not very suitable for very complex data [2, 4]. One solution is to impose a Gradient Penalty instead of weight clipping as a means to enforce Lipschitz constraint [4]. The value function for WGAN-GP can be observed in Equation (3). Here, $\boldsymbol{L}$ represents the loss function, x' represents a sample from fake or generated data, and $\hat{x}$ represents randomly sampled data. Note that a "soft penalty" is imposed (i.e. only on the randomly sampled data) to prevent tractability issues. The last term in the equation is the penalty term, with λ being the penalty coefficient.

$$\min_{G} \max_{D \in \mathcal{D}} \mathbb{E}_{\tilde{x} \sim p_g}[D(\tilde{x})] - \mathbb{E}_{x \sim p_r}[D(x)] + \lambda \mathbb{E}_{\hat{x} \sim p_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{3}$$

$p_{\hat{x}}$ is defined for sampling uniformally on straight edges (see [4] for details). Generally, for a 1-Lipschitz function, the maximum gradient norm should be 1. Therefore, instead of applying weight-clipping, WGAN-GP loss function imposes a penalty if the gradient norm moves away from its maximum target norm value of 1.

3 Method

In this study, we compared the traditional image data augmentation technique of using geometric and photometric transformations [13], with our proposed technique of using Generative Adversarial Networks (specifically, WGAN-GP [4]) for the same purpose. We first artificially introduce imbalance in two benchmark balanced datasets of FMNIST [16] and CIFAR-10 [6], and the class distribution can be observed in Figure 1.



FMNIST CIFAR-10

Fig. 1: Frequency of Classes after introducing imbalance

To effectively study the effects of dataset imbalance on classification performance, we use a Hybrid CNN-SVM architecture. The overall workflow of the

study can be observed in Figure 2. The classifier is separately trained on each of these datasets and these trained models are used for evaluation. However, no change is made to the test set of these datasets.

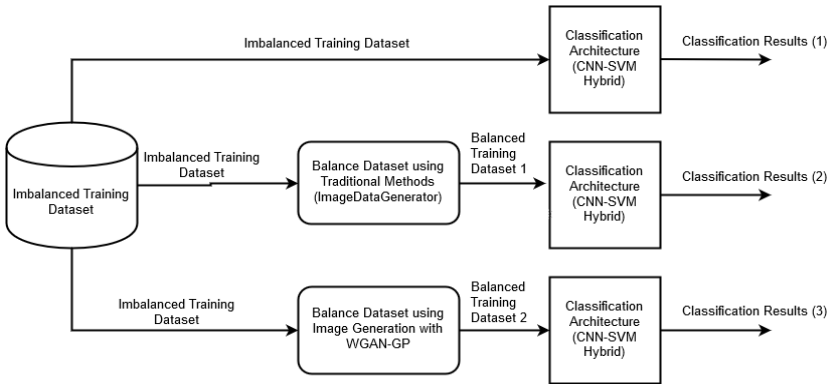


Fig. 2: Workflow of our study.

3.1 Oversampling using Image Transformations

Some of the most common geometric and photo-metric transformations applied on images to generate additional data while preserving the context of the image are rotation, translation, shearing, scaling, flipping, zooming, blurring, whitening etc [13]. This has been achieved using ImageDataGenerator class of the image pre-processing module of Keras, and a sample of images generated using this methodology can be observed in Figure 3.

3.2 Oversampling using WGAN-GP

WGAN-GP [2] is a superior GAN architecture which is known to achieve convergence without facing the issues of vanishing or exploding gradients. The training process is very stable for this architecture, and it has also proven to generate highly diverse data samples with low noise. It can achieve high quality results with almost no hyperparameter tuning, making it a suitable choice for a wide variety of applications and datasets. Therefore, we have adopted the WGAN-GP architecture for our study.

The general methods for GAN architecture construction and the choices made in our model are described below:

- ***Use of Leaky ReLU:*** The Rectified Linear Unit (ReLU) activation function is a widely adopted, efficient activation function that returns the input directly as the output, or returns 0 when the input is 0.0 or less. However, the best practice for GANs is to use a variation called LeakyReLU, which

allows some values lesser than 0, and learns the optimum cut-off for each node. In our architecture, we have used LeakyRelu in both the generator as well as discriminator, with slope values being of the order of the default value, 0.2.

- ***Use of Batch Normalization:*** Batch normalization is a technique used to improve the speed, performance and stability of neural networks by normalizing the input layer by adjusting and scaling the activations. We employ batch normalization after the convolution layers. In the case of GANs, it helps avoid vanishing and exploding gradients, as well as mode collapse.
- ***Using Gaussian Weight Initialization:*** Before starting the training process, the weights (parameters) of the neural network must be initialized with small random variables to prevent the activation layer from producing vanishing or exploding outputs, which would cause very small or very large gradient updates, giving rise to convergence problems. It is considered to be a good practice to initialize all weights using a zero-centred Gaussian distribution, with mean value as 0 and variance value as 1/N, where N specifies the number of input neurons. Therefore, we have used Xavier initialization in our architecture, which is based on the same principle.
- ***Not using Max Pooling:*** A max-pooling layer is often used in CNNs to after each convolution layer to downsample the input and feature maps. However, we would not be using this approach as in the case of Generative Adversarial Networks, it has been shown that having all convolutional layers allows the network to learn its own spatial down-sampling, which leads to an increase in performance.

Qualitative Evaluation Metrics for WGAN-GP Since our goal is to use the WGAN-GP model to generate additional minority class images in order to augment an imbalanced dataset and balance it, we aim to fulfil the following criterion for our generated images through visual inspection [9]:

- Generated images should be similar to the other images of the class in question. If this target is not met, it would mean that the generator is not trained enough to produce quality, realistic images.
- Generated images must not all be the same, or repetitive. This will ensure that the generator does not suffer from mode collapse problem.
- Generated images should be different from the images which are already present in the training set. If not, it would mean that we have simply trained our generative model to repeat the training data.

In this study, the visual inspection has been done by the researchers themselves, by randomly mixing real and generated samples and testing if the fake samples can be spotted in that mix. The generated images sampled at various epochs of WGAN-GP training phase for the class *horse* can be observed in Figure 5.

3.3 Classification Architecture

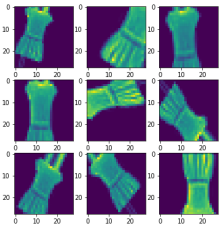
We adopted the hybrid CNN-SVM architecture as the classifier as proposed by Niu and Suen [10]. In this hybrid architecture, the original CNN (with the output layer) is trained on the input dataset until convergence is achieved. Then, the output layer is replaced with the Radial Bias Function (RBF) of SVM. The output from the CNN hidden layer is taken as a feature vector for training the SVM. Once trained, this SVM is able to perform the classification task on unseen data.

4 Experimental Results

New data samples generated by WGAN-GP (Figure 4) result in a superior augmented (balanced) dataset as compared to the dataset obtained by augmentation with images generated using geometric transforms (Figure 3). This can be attributed to the following qualities of WGAN-generated images:

- Realistic looking, and impossible to tell apart from the original dataset images of the same class by a human observer
- Preservation of context or semantic information of the class in question
- Samples are not repetitive, signifying that there is very less possibility of over-fitting
- Samples generated are variable or diverse in nature, therefore resulting in an efficiently augmented dataset which contains samples representing many possibilities

Original Image

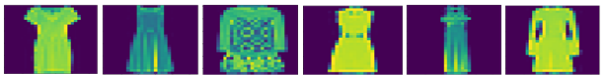


New images generated using Image Transformations

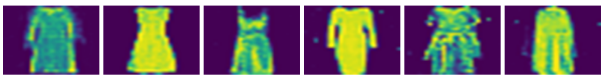
Fig. 3: An example of Generating New Data using Image Transforms on *Dress* sample of FMNIST dataset.

4.1 Results for FMNIST

The results of classification on the CNN-SVM architecture are reported in Table 1 when using FMNIST dataset. We note that imbalance causes the classifier's



(a) Original Images



(b) Synthesised Images Generated using WGAN-GP

Fig. 4: Comparison of Real Images and Synthesised Images generated using WGAN-GP for *Dress* class of FMNIST

performance to decrease (e.g. Accuracy drops by 5.25% , and similarly for the F1-Score). While using traditional image transforms results in an improvement of approximately 4% in testing Accuracy, and of around 4.3% in the F1-Score, it can be observed that WGAN-GP does a better job and increases the Accuracy and F1-Score by 5% and 4.5% in comparison to the imbalanced dataset.

	Accuracy	Precision	Recall	F1 Score
Original (Balanced)	0.932	0.931	0.932	0.932
Imbalanced	0.883	0.897	0.884	0.883
Data Augmentation (Image Transforms)	0.919	0.922	0.92	0.921
Data Augmentation (WGAN)	0.928	0.925	0.921	0.923

Table 1: Testing Metrics for Classification of Variations of FMNIST.

4.2 Results for CIFAR10

The results of classification on the CNN-SVM architecture are reported in Table 2 for the dataset CIFAR-10. While the class imbalance creates a 8% drop in accuracy, data augmentation approaches both manage to restore the performance with an increase of 5% for Image transforms and 6% for WGAN.

	Accuracy	Precision	Recall	F1 Score
Original (Balanced)	0.8342	0.837	0.833	0.834
Imbalanced	0.7569	0.787	0.758	0.75
Data Augmentation (Image Transforms)	0.8084	0.812	0.808	0.806
Data Augmentation (WGAN)	0.8189	0.824	0.815	0.812

Table 2: Testing Metrics for Classification of Variations of CIFAR-10

5 Conclusion

We have shown the potential benefit of using WGAN for improving the performance of a classifier when the training dataset suffers from class imbalance. For further testing, it would not only be vital to observe the performance of the classifier with different distributions of imbalanced classes, but to also take into account the impact of noise, dataset shift problem (where the train and test sets follow different distributions) and the cases where there is possibility of class overlapping. It would also be useful to introduce quantitative metrics to assess the images generated by GANs, such as SSIM (Structural Similarity Index) [15], Inception Score [12], GAN Quality Index [17], which would reduce or eliminate the need for human visual inspection.

Acknowledgements. This work has emanated from research conducted as part of the MSc in Computer Science programme (Future Networked Systems) at Trinity College Dublin, Ireland. This research is partly supported by the ADAPT Centre for Digital Content Technology funded under the SFI Research Centres Programme (Grant 13/RC/2106) and co-funded under the European Regional Development Fund.

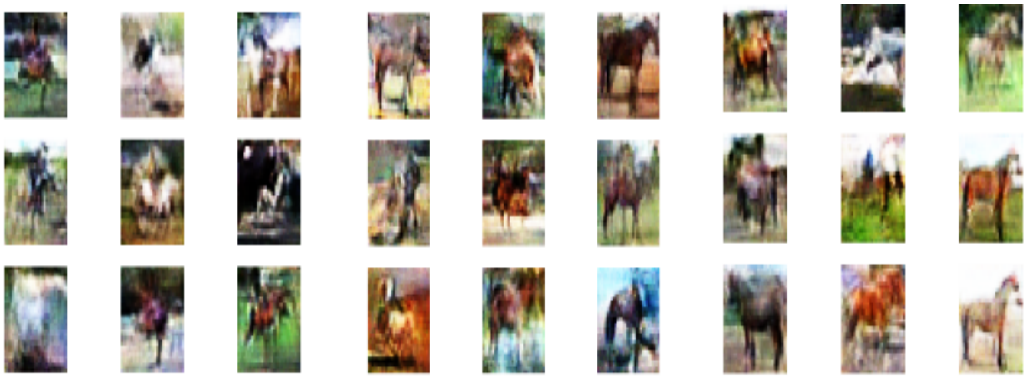
References

1. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. In: International Conference on Learning Representation (2017)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 214–223 (06–11 Aug 2017)
3. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems 27. pp. 2672–2680 (2014)
4. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in neural information processing systems. pp. 5767–5777 (2017)
5. Huang, R., Zhang, S., Li, T., He, R.: Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. 2017 IEEE International Conference on Computer Vision (ICCV) pp. 2458–2467 (2017)
6. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep., CIFAR-10 (Canadian Institute for Advanced Research) (2009), `http://www.cs.toronto.edu/~kriz/cifar.html`
7. Ledig, C., Theis, L., Huszár, F., Caballero, J., Aitken, A.P., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 105–114 (2016)
8. Li, C., Wand, M.: Precomputed real-time texture synthesis with markovian generative adversarial networks. CoRR abs/1604.04382 (2016), `http://arxiv.org/abs/1604.04382`
9. Mariani, G., Scheidegger, F., Istrate, R., Bekasa, C., Malossi, A.C.I.: Bagan: Data augmentation with balancing gan. In: ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models (2018)

10. Niu, X.X., Suen, C.Y.: A novel hybrid cnn–svm classifier for recognizing handwritten digits. Pattern Recognition 45(4), 1318–1325 (2012)
11. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier GANs. In: Proceedings of the 34th International Conference on Machine Learning. vol. 70, pp. 2642–2651. PMLR (06–11 Aug 2017)
12. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems 29. pp. 2234–2242 (2016)
13. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of Big Data 6(1), 60 (Jul 2019), https://doi.org/10.1186/s40537-019-0197-0
14. Ulicny, M., Krylov, V., Dahyot, R.: Harmonic networks for image classification. In: British Machine Vision Conference (BMVC). Cardiff UK (9-12 September 2019)
15. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., et al.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13(4), 600–612 (2004)
16. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. CoRR abs/1708.07747 (2017), http://arxiv.org/abs/1708.07747
17. Ye, Y., Wang, L., Wu, Y., Chen, Y., Tian, Y., Liu, Z., Zhang, Z.: Gan quality index (gqi) by gan-induced classifier (2018), https://openreview.net/forum?id=S1CIev1vM

(a) Epoch 5 (b) Epoch 21 (c) Epoch 27

(d) Epoch 54 (e) Epoch 71 (f) Epoch 102



(g) Epoch 127 (h) Epoch 260 (i) Epoch 406

Fig. 5: Data Generated for the Minority class 'Horse' in the Imbalanced CIFAR-10 using WGAN-GP through the training epochs. Note that at Epoch 406, we first observe realistic-looking generated images