
Model Compression with Generative Adversarial Networks

Ruishan Liu
Stanford University
Stanford, CA, USA
ruishan@stanford.edu

Nicolo Fusi & Lester Mackey
Microsoft Research
Cambridge, MA, USA
{fusi,lmackey}@microsoft.com

Abstract

More accurate machine learning models often demand more computation and memory at test time, making them difficult to deploy on CPU- or memory-constrained devices. *Model compression* (also known as *distillation*) alleviates this burden by training a less expensive student model to mimic the expensive teacher model while maintaining most of the original accuracy. However, when fresh data is unavailable for the compression task, the teacher’s training data is typically reused, leading to suboptimal compression. In this work, we propose to augment the compression dataset with synthetic data from a generative adversarial network (GAN) designed to approximate the training data distribution. Our *GAN-assisted model compression* (GAN-MC) significantly improves student accuracy for expensive models such as deep neural networks and large random forests on both image and tabular datasets. Building on these results, we propose a comprehensive metric—the *Compression Score*—to evaluate the quality of synthetic datasets based on their induced model compression performance. The Compression Score captures both data diversity and discriminability, and we illustrate its benefits over the popular Inception Score in the context of image classification.

1 Introduction

Modern machine learning models have achieved remarkable levels of accuracy, but their complexity can make them slow to query, expensive to store, and difficult to deploy for real-world use. Ideally, we would like to replace such cumbersome models with simpler models that perform equally well. One way to address this problem is to perform *model compression* (also known as *distillation*), which consists of training a student model to mimic the outputs of a teacher model (Bucila et al., 2006; Hinton et al., 2015).

An important degree of freedom in the model compression problem is the *compression set*¹ used to train the student. Ideally, fresh (unlabeled) data from the training distribution would fuel this task, but often no fresh data remains after the teacher is trained (Bucila et al., 2006; Ba & Caruana, 2014), leading to suboptimal compression performance.

In this paper, we propose GAN-assisted model compression (GAN-MC), a simple approach to improving teacher-student compression by augmenting the compression set with GAN data. On CIFAR-10 image classification, we show GAN-MC consistently improves student test accuracy for deep neural network teacher-student pairings. For random forest teachers, we demonstrate 25 to 336-fold reductions in execution and storage costs with less than 1.2% loss in test performance across a suite of real-world tabular datasets.

¹To avoid ambiguity, we will refer to the dataset used for compression as “compression set” and reserve the name “training set” for the data used to train the teacher.

We further develop a *Compression Score*, which uses GAN-MC to evaluate the quality of synthetic datasets and their generators. We show it offers a robust, goal-driven metric for synthetic data quality and illustrate its advantages over the popular Inception Score on CIFAR-10.

2 Model Compression with GANs

2.1 Deep Neural Network Compression

In the standard teacher-student approach to compressing a neural network classifier, a relatively inexpensive prediction rule, like a shallow neural network, is trained to predict the unnormalized log probability values—the *logits* z —assigned to each class by a previously trained deep network classifier. The inexpensive model is termed the *student*, and the expensive deep network is termed the *teacher*. Given a compression set of n feature vectors paired with teacher logit vectors, $\{(x^{(1)}, z^{(1)}), \dots, (x^{(n)}, z^{(n)})\}$, Ba & Caruana (2014) proposed framing the compression task as a multitask regression problem with L^2 loss,

$$L(\theta) = \|g(x; \theta) - z\|_2^2. \quad (1)$$

Here, θ represents any student model parameters to be learned (e.g., the student network weights), and $g(x; \theta)$ is the vector of logits predicted by the student model for the input feature vector x .

2.2 Random Forest Compression

Random forests (Breiman, 2001) construct highly accurate prediction rules by averaging the predictions of a diverse and often large collection of learned decision trees. Effectively mimicking a large random forest with a single decision tree or a small forest has the potential to reduce prediction computation and storage costs by multiple orders of magnitude (Bucila et al., 2006; Joly et al., 2012; Begon et al., 2017; Painsky & Rosset, 2016, 2018). Focusing on the common setting of binary classification with labels in $\{0, 1\}$, we propose to train a student regression random forest to predict a teacher forest’s outputted probability p of a datapoint x having the label 1.

2.3 GAN-assisted Model Compression (GAN-MC)

In a typical compression setting, as much data as possible has been dedicated to training the highly accurate teacher model, leaving little fresh data for training the student model. To boost student performance and compression efficiency, we propose a simple solution applicable to tabular and image data alike: augment the compression set with synthetic feature vectors generated by a high-quality GAN. These synthetic feature vectors are then labeled with the true outputted teacher class probabilities or logits, as described in Secs. 2.1 and 2.2. We call this approach *GAN-assisted model compression* (GAN-MC).

To generate high-quality GAN feature vectors which capture the salient features of each class, we use the auxiliary classifier GAN (AC-GAN) of Odena et al. (2017). The AC-GAN generator G produces a synthetic feature vector $X_{fake} = G(W, C)$ given a random noise vector W and an independent target class label C drawn from the real data class distribution. For any given feature vector x , the AC-GAN discriminator D predicts both the probability of each class label $P(C | x)$ and the probability of the data source being real or fake, $P(S | x)$ for $S \in \{real, fake\}$. Given a training dataset \mathcal{D}_{real} of labeled feature vectors, two components contribute to the AC-GAN training objective:

$$L_{source} = \frac{1}{|\mathcal{D}_{real}|} \sum_{(x,c) \in \mathcal{D}_{real}} \log P(S = real | x) + \mathbb{E}_{W,C \sim p_c} [\log P(S = fake | G(W, C))] \text{ and} \\ L_{class} = \frac{1}{|\mathcal{D}_{real}|} \sum_{(x,c) \in \mathcal{D}_{real}} \log P(C = c | x) + \mathbb{E}_{W,C \sim p_c} [\log P(C | G(W, C))], \quad (2a)$$

representing the expected conditional log-likelihood of the correct source and the correct class of a feature vector, respectively. In the adversarial game, the generator G is trained to maximize $L_{class} - L_{source}$, and the discriminator D is trained to maximize $L_{class} + L_{source}$.

3 Deep Neural Network GAN-MC

We now investigate how GAN-MC performs when used to compress convolutional deep neural network (CNN) classifiers trained on the CIFAR-10 dataset of Krizhevsky & Hinton (2009). CIFAR-10 consists of 32×32 RGB images from 10 classes, divided into 50,000 training and 10,000 test

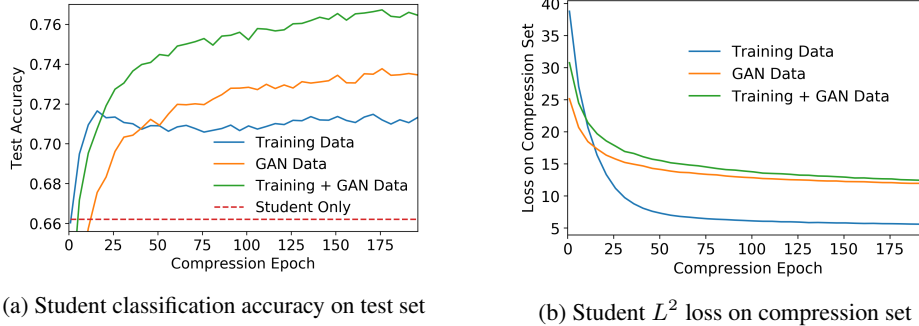


Figure 1: Student performance for L^2 logit-matching compression as training progresses on CIFAR-10 (see Sec. 3). The teacher and student are NIN and LeNet with test accuracies of 78.1% and 66.2% (red dashed curve in (a)) when trained without compression. Compression is performed using only real training data (blue curve), only synthetic GAN data (green curve), or a mixture of training data and GAN data (orange curve, $p_{fake} = 0.8$). Results are averaged over 3 independent runs.

Table 1: Test accuracy (Higgs) and test AUC (Evergreen and MAGIC) of the learned student in random forest compression. Here a forest with 500 trees is compressed into a single decision tree.

Dataset	Training Data Size	Teacher Only	Student Only	Student after Compression with		
				Training Data	GAN Data	Training & GAN
Higgs	1k	66.4%	56.2%	56.5%	59.0%	57.7%
	100k	72.6%	62.1%	62.7%	69.6%	64.7%
MAGIC	10k	0.935	0.785	0.895	0.918	0.912
Evergreen	5k	0.889	0.731	0.856	0.882	0.849

images. The teacher and the student are NIN (Lin et al., 2014) and LeNet (LeCun et al., 1998) models. The uncompressed networks are pre-trained by Caffe (Chan, 2016; Jia et al., 2014). For compression training, we use the Adam optimizer in Tensorflow (Abadi et al., 2015) with learning rate 10^{-4} and the L^2 loss in Eq. 1. The performance is evaluated after 200 training epochs.

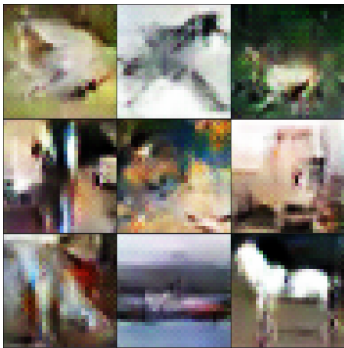
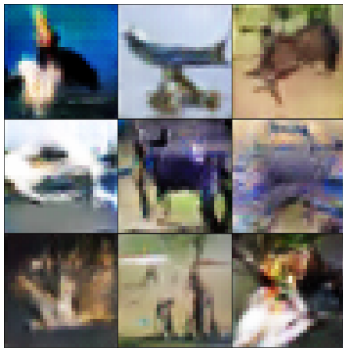
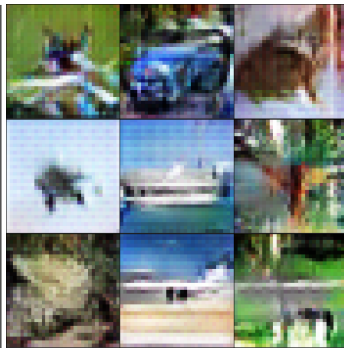
We compare the standard approach of compression using only the teacher’s training dataset to two versions of GAN-MC: compression using only GAN data and compression using a mixture of training and GAN data. The GAN data is produced in real time during the stochastic optimization training. The mixture of training and GAN data is realized by generating GAN data with probability p_{fake} and by sampling from the training set with probability $1 - p_{fake}$. For each teacher-student pair, we select the value of p_{fake} in $\{0.0, 0.1, 0.2, \dots, 1.0\}$ that yields the highest validation set accuracy and report performance on the held-out test set. This results in the choice $p_{fake} = 0.8$.

Fig. 1a displays student test accuracy following each epoch of compression training with the L^2 logit-matching objective. In the end, both versions of GAN-MC significantly outperform compression on training data alone and training without compression (‘Student Only’). The results are particularly striking for the mixture of GAN and training data which doubles the impact of training data compression. In this case, student accuracy increases by 10.5 percentage points (from 66.2% to 76.7%) with GAN-MC as opposed to 5.3 percentage points (from 66.2% to 71.5%) with training data alone.

4 Random Forest GAN-MC

We next use three tabular datasets — Evergreen (StumbleUpon Evergreen) dataset from Kaggle and Higgs and MAGIC (MAGIC Gamma Telescope) datasets from the UCI Machine Learning Repository — to explore how GAN-MC performs when used to compress large random forests for binary classification. We study three scenarios: compression using training data only, GAN data only or a mixture of training and GAN data. We generate $n_{fake} = 9n_{real}$ GAN datapoints for the compression set, where n_{real} is the number of real training datapoints. The mixture compression set is generated by pooling the n_{real} training datapoints and the n_{fake} GAN datapoints together.

Table 2: Inception and Compression Scores for CIFAR-10 images; larger scores should signify higher quality images. Inferior data generated by training well-trained GAN for 10 additional epochs using only the classification objective L_{class} (see Sec. 5.2). Inception Score increases for inferior images despite evident unrealistic artifacts. Compression Score decreases for inferior images.

Real Data	Well-trained GAN	Inferior GAN
		
Inception: 11.2 ± 0.1 Compression: 0.994 ± 0.003	Inception: 5.80 ± 0.06 Compression: 0.778 ± 0.002	Inception: 5.93 ± 0.06 Compression: 0.702 ± 0.002

The results of compressing a random forest with 500 trees into a single decision tree are given in Table 1. We use test accuracy as our performance metric for the balanced Higgs dataset and test AUC for the unbalanced MAGIC and Evergreen datasets. For all datasets and a variety of training dataset sizes, compression with GAN data outperforms compression with training data and substantially outperforms the student model trained without compression.

5 Evaluating GANs with a Compression Score

5.1 The Compression Score

To evaluate the quality of a generated dataset \mathcal{D} relative to a real dataset \mathcal{D}_{real} , we define a *Compression Score* based on the test accuracy $acc(\mathcal{D})$ of a student trained with compression set \mathcal{D} to mimic a teacher pre-trained on \mathcal{D}_{real} :

$$\text{CompressionScore}(\mathcal{D}; \mathcal{D}_{real}) = \frac{acc(\mathcal{D}) - acc_{mode}}{acc(\mathcal{D}_{real}) - acc_{mode}}.$$

The term acc_{mode} represents the accuracy obtained by always predicting the most common class in \mathcal{D}_{real} . A higher Compression Score is designed to indicate a higher quality dataset \mathcal{D} .

5.2 Evaluating GANs: An Illustration with CIFAR-10

To illustrate the potential benefit of the Compression Score over the commonly-used Inception Score, we reinstate the CIFAR-10 experimental setup of Fig. 1. The standard error is obtained from 3 independent runs. Each student is trained only for one epoch. We evaluate the compression score on real data, well-trained GAN data (i.e., data from the AC-GAN described in Sec. 3), and inferior data which have high confidence classifications under the teacher network but do not resemble real data. The inferior data are generated by training the well-trained AC-GAN for 10 additional epochs using only the classification objective L_{class} given in Eq. 2a. That is, both the generator G and discriminator D are trained to maximize L_{class} , while ignoring the traditional GAN objective component L_{source} .

In Table 2, the quality of the GAN data degrades noticeably after the additional training with only L_{class} . Unrealistic artifacts are evident in the inferior images, but the Inception Scores of those images are higher than those of the well-trained images. In contrast, the Compression Score decreases in accordance with our expectations as the GAN images become evidently worse. Using the Inception Score code of (Salimans et al., 2016) and an NVIDIA Tesla V100 GPU, the Inception Score required 1436.6s and the Compression Score 350.1s.

References

- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/>. Software available from [tensorflow.org](https://www.tensorflow.org/).
- Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pp. 2654–2662, 2014.
- Jean-Michel Begon, Arnaud Joly, and Pierre Geurts. Globally induced forest: A prepruning compression scheme. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 420–428, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.
- Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pp. 535–541, 2006. doi: 10.1145/1150402.1150464. URL <http://doi.acm.org/10.1145/1150402.1150464>.
- James Chan. https://github.com/chengshengchan/model_compression, 2016.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
- Arnaud Joly, François Schnitzler, Pierre Geurts, and Louis Wehenkel. L1-based compression of random forest models. In *20th European Symposium on Artificial Neural Networks*, 2012.
- Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *ICLR*, 2014.
- Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2642–2651, 2017.
- Amichai Painsky and Saharon Rosset. Compressing random forests. In *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, pp. 1131–1136, 2016. doi: 10.1109/ICDM.2016.0148. URL <https://doi.org/10.1109/ICDM.2016.0148>.
- Amichai Painsky and Saharon Rosset. Lossless (and lossy) compression of random forests. *arXiv preprint arXiv:1810.11197*, 2018.
- Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.