CapsuleGAN: Generative Adversarial Capsule Network

Ayush Jaiswal, Wael AbdAlmageed, Premkumar Natarajan
USC Information Sciences Institute
Marina del Rey, CA, USA
{ajaiswal, wamageed, pnataraj}@isi.edu

Abstract

We present Generative Adversarial Capsule Network (CapsuleGAN), a framework that uses capsule networks (CapsNets) instead of the standard convolutional neural networks (CNNs) as discriminators within the generative adversarial network (GAN) setting, while modeling image data. We provide guidelines for designing CapsNet discriminators and the updated GAN objective function, which incorporates the CapsNet margin loss, for training CapsuleGAN models. We show that CapsuleGAN outperforms convolutional-GAN at modeling image data distribution on the MNIST dataset of handwritten digits, evaluated on the generative adversarial metric and at semi-supervised image classification.

1 Introduction

Generative modeling of data is a challenging machine learning problem that has garnered tremendous interest recently, partly due to the invention of generative adversarial networks (GANs) [5] and its several sophisticated variants [1]. A GAN model is typically composed of two neural networks; (1) a generator that attempts to transform samples drawn from a prior distribution to samples from a complex data distribution with much higher dimensionality, and (2) a discriminator that decides whether the given sample is real or from the generator's distribution. The two components are trained by playing an adversarial game. GANs have shown great promise in modeling highly complex distributions underlying real world data, especially images. However, they are notorious for being difficult to train and have problems with stability, vanishing gradients, mode collapse and inadequate mode coverage [15, 18, 4]. Consequently, there has been a large amount of work towards improving GANs by using better objective functions [1, 6, 2], sophisticated training strategies [18], using structural hyperparameters [15, 14] and adopting empirically successful tricks [2].

Radford et al. [15] provide a set of architectural guidelines, formulating a class of convolutional neural networks (CNN) that have since been extensively used to create GANs (referred to as Deep Convolutional GANs or DCGANs) for modeling image data and other related applications [16, 10]. More recently, however, Sabour et al. [17] introduced capsule networks (CapsNets) as a powerful alternative to CNNs, which learn a more *equivariant* representation of images that is more robust to changes in pose and spatial relationships of parts of objects in images [7] (information that CNNs lose during training, by design). Inspired by the working mechanism of optic neurons in the human visual system, capsules were first introduced by Hinton et al. [7] as locally invariant groups of neurons that learn to recognize visual entities and output activation vectors that represent both the presence of those entities and their properties relevant to the visual task (such as object classification). CapsNets have been shown to outperform CNNs on MNIST digit classification and segmentation of overlapping digits [17]. This motivates the question whether GANs can be designed using CapsNets (insteated of CNNs) to improve their performance.

---

[1]https://github.com/hindupuravinash/the-gan-zoo
[2]https://github.com/soumith/ganhacks

We propose Generative Adversarial Capsule Network (CapsuleGAN) as a framework that incorporates capsules within the GAN framework. In particular, CapsNets are used as discriminators in our framework as opposed to the conventionally used CNNs. We show that CapsuleGANs perform better than CNN-based GANs at modeling the underlying distribution of MNIST [12] handwritten digits both qualitatively, and quantitatively using the generative adversarial metric (GAM) [9] and at semi-supervised classification using unlabeled GAN-generated images with a small number of labeled real images.

The rest of the paper is organized as follows. Section 2 discusses related work. In Section 3 we provide a brief introduction to GANs and CapsNets. Section 4 describes our CapsuleGAN framework along with implementation guidelines. Qualitative and quantitative analyses of our model are presented in Section 5. Section 6 concludes the paper and provides directions for future research.

2   Related Work

GANs were originally implemented as feedforward multi-layer perceptrons, which did not perform well on generating complex images like those in the CIFAR-10 dataset [11]. They suffered from mode collapse and were highly unstable to train [15, 18]. In an attempt to solve these problems, Radford et al. [15] presented a set of guidelines to design GANs as a class of CNNs, giving rise to DCGANs, which have since been a dominant approach to GAN network architecture design. Im et al. [8] later proposed the use of Recurrent Neural Networks instead of CNNs as generators for GANs, creating a new class of GANs referred to as Generative Recurrent Adversarial Networks or GRANs. On a related note, Odena et al. [14] proposed an architectural change to GANs in the form of a discriminator that also acts as a classifier for class-conditional image generation. This approach for designing discriminators has also been a popular choice for conditional GANs [13] recently. Our work is similar in line with [14] in the sense that we propose an architectural change to discriminators. We propose to transition from designing GAN discriminators as CNNs to formulating them as CapsNets, creating a new class of GANs called CapsuleGANs. This idea can be extended to encoder-based GANs like BiGAN [3] where the encoder can be modeled as a CapsNet also.

3   Preliminaries

3.1   Generative Adversarial Networks

Goodfellow et al. [5] introduced GANs as a framework for generative modeling of data through learning a transformation from points belonging to a simple prior distribution ($\mathbf{z} \sim p_z$) to those from the data distribution ($\mathbf{x} \sim p_{data}$). The framework is composed of two models that play an adversarial game: a generator and a discriminator. While the generator attempts to learn the aforementioned transformation $G(\mathbf{z})$, the discriminator acts as a critic $D(\cdot)$ determining whether the sample provided to it is from the generator's output distribution ($G(\mathbf{z}) \sim p_G$) or from the data distribution ($\mathbf{x} \sim p_{data}$), thus giving a scalar output ($y \in \{0, 1\}$). The goal of the generator is to fool the discriminator by generating samples that resemble those from the real data while that of the discriminator is to accurately distinguish between real and generated data. The two models, typically designed as neural networks, play an adversarial game with the objective as shown in Equation 1.

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} \left[\log D(\mathbf{x}) \right] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} \left[\log(1 - D(G(\mathbf{z}))) \right] \qquad (1)$$

3.2   Capsule Networks

The concept of capsules was first introduced by Hinton et al. [7] as a method for learning robust unsupervised representation of images. Capsules are locally invariant groups of neurons that learn to recognize the presence of visual entities and encode their properties into vector outputs, with the vector length (limited to being between zero and one) representing the presence of the entity. For example, each capsule can learn to identify certain objects or object-parts in images. Within the framework of neural networks, several capsules can be grouped together to form a capsule-layer where each unit produces a vector output instead of a (conventional) scalar activation.

Sabour et al. [17] introduced a routing-by-agreement mechanism for the interaction of capsules within deep neural networks with several capsule-layers, which works by pairwise determination of the passage of information between capsules in successive layers. For each capsule $h_i^{(l)}$ in layer l and each capsule $h_j^{(l+1)}$ in the layer above, a coupling coefficient c_{ij} is adjusted iteratively based on the agreement (cosine similarity) between h_i's prediction of the output of h_j and its actual output given the product of c_{ij} and h_i's activation. Thus, the coupling coefficients inherently decide how information flows between pairs of capsules. For a classification task involving K classes, the final layer of the CapsNet can be designed to have K capsules, each representing one class. Since the length of a capsule's vector output represents the presence of a visual entity, the length of each capsule in the final layer ($\|\mathbf{v_k}\|$) can then be viewed as the probability of the image belonging to a particular class (k). The authors introduce a margin loss L_M for training CapsNets for multi-class classification, as show in Equation 2:

$$L_M = \sum_{k=1}^{K} T_k \max(0, m^+ - \|\mathbf{v_k}\|)^2 + \lambda(1 - T_k) \max(0, \|\mathbf{v_k}\| - m^-)^2 \qquad (2)$$

where T_k represents target labels, $m^+ = 0.9$, $m^- = 0.1$ and $\lambda = 0.5$, a down-weighting factor for preventing initial learning from shrinking the lengths of the capsule outputs in the final layer. The authors also add regularization to the network in the form of a weighted image reconstruction loss, where the vector outputs $\mathbf{v_k}$ of the final layer are presented as inputs to the reconstruction network.

4 Generative Adversarial Capsule Networks

GANs have been mostly used for modeling the distribution of image data and associated attributes, as well as for other image-based applications like image-to-image translation [10] and image synthesis from textual descriptions [16]. The generator and the discriminator have conventionally been modeled as deep CNNs following the DCGAN guidelines [15]. Motivated by the stronger intuition behind and the superior performance of CapsNets with respect to CNNs [17], we design our CapsuleGAN framework to incorporate capsule-layers instead of convolutional layers in the GAN discriminator, which fundamentally performs a two-class classification task.

The CapsuleGAN discriminator is similar in architecture to the CapsNet model presented in [17]. CapsNets, in general, have a large number of parameters because, firstly, each capsule produces a vector output instead of a single scalar and, secondly, each capsule has *additional* parameters associated with all the capsules in the layer above it that are used for making predictions about their outputs. However, it is necessary to keep the number of parameters in the CapsuleGAN discriminator low due to two reasons: (1) CapsNets are very powerful models and can easily start harshly penalizing the generator early on in the training process, which will cause the generator to either fail completely or suffer from mode collapse, and (2) current implementations of the dynamic routing algorithm are slow to run. It is important to note that first reason for keeping the number of parameters of the CapsNet low falls in line with the popular design of convolutional discriminators as relatively shallow neural networks with low numbers of relatively large-sized filters in their convolutional layers.

The final layer of the CapsuleGAN discriminator contains a single capsule, the length of which represents the probability whether the discriminator's input is a real or a generated image. We use margin loss L_M instead of the conventional binary cross-entropy loss for training our CapsuleGAN model because L_M works better for training CapsNets. Therefore, the objective of CapsuleGAN can be formulated as shown in Equation 3.

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} [-L_M(D(\mathbf{x}), \mathbf{T} = \mathbf{1})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [-L_M(D(G(\mathbf{z})), \mathbf{T} = \mathbf{0})]$$
$$(3)$$

In practice, we train the generator to minimize $L_M(D(G(\mathbf{z})), \mathbf{T} = \mathbf{1})$ instead of minimizing $-L_M(D(G(\mathbf{z})), \mathbf{T} = \mathbf{0})$. This essentially eliminates the down-weighting factor λ in L_M when training the generator, which does not contain any capsules.

(a) GAN   (b) CapsuleGAN
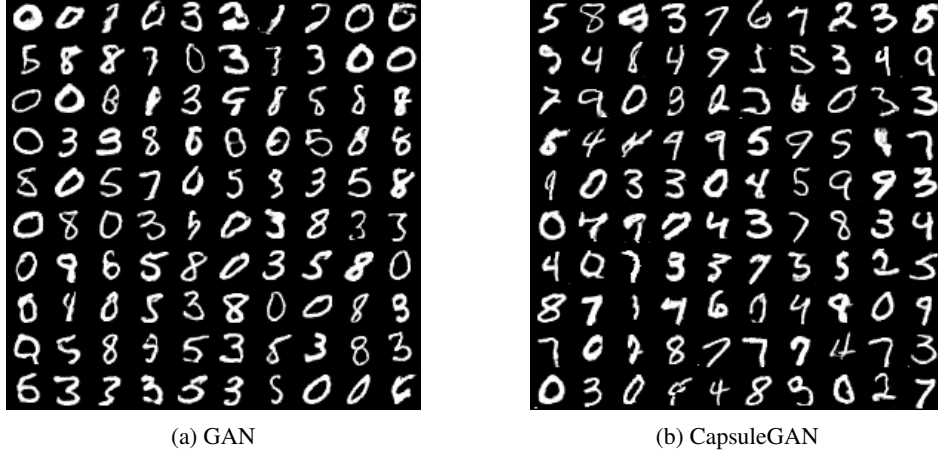
Figure 1: Randomly generated MNIST images

5   Experimental Evaluation

We evaluate the performance of CapsuleGANs at randomly generating images through a series
of experiments as described below, in which we compare the performance of CapsuleGANs with
convolutional GANs both qualitatively and quantitatively. We implement both the GAN models with
the same architecture for their generators. We provide results of our experiments on the MNIST [12]
dataset of 28×28 sized grayscale images of handwritten digits. Both convolutional GAN and the
proposed CapsuleGAN models are implemented using the publicly available `keras-adversarial` [3]
and `CapsNet-Keras` [4] packages.

5.1   Visual Quality of Randomly Generated Images

We qualitatively compare images generated randomly using both GAN and CapsuleGAN. Figure 1a
shows images generated using the standard GAN while Figure 1b shows those for CapsuleGAN.
Qualitatively, both CapsuleGAN and the standard CNN-GAN produce crisp images of similar quality,
that sometimes do not resemble any digit. However, the image-grid generated using GAN seems
to have less diversity in terms of generated classes of digits. We provide results of our quantitative
evaluation in the following subsections for deeper analyses of the image generation performance.

5.2   Generative Adversarial Metric

Im et al. [9] introduced the generative adversarial metric (GAM) as a pairwise comparison metric
between GAN models by pitting each generator against the opponent's discriminator, i.e., given two
GAN models $M_1 = (G_1, D_1)$ and $M_2 = (G_2, D_2)$, G_1 engages in a battle against D_2 while G_2
against D_1. The ratios of their classification errors on real test dataset and on generated samples are
then calculated as r_{test} and $r_{samples}$. Following their implementation [5], in practice, the ratios of
classification accuracies are calculated instead of errors to avoid numerical problems, as shown in
Equations 4 and 5

$$r_{samples} = \frac{A(D_{GAN}(G_{CapsuleGAN}(\mathbf{z})))}{A(D_{CapsuleGAN}(G_{GAN}(\mathbf{z})))} \tag{4}$$

$$r_{test} = \frac{A(D_{GAN}(\mathbf{x_{test}}))}{A(D_{CapsuleGAN}(\mathbf{x_{test}}))} \tag{5}$$

Table 1: Results of semi-supervised classification - MNIST

Model	Error Rate		
	n = 100	n = 1,000	n = 10,000
GAN	0.2900	0.1539	0.0702
CapsuleGAN	**0.2724**	**0.1142**	**0.0531**

Therefore, for CapsuleGAN to win against GAN, both $r_{samples} < 1$ and $r_{test} \simeq 1$ must be satisfied. In our experiments, we achieve $r_{samples} = 0.79$ and $r_{test} = 1$. Thus, CapsuleGAN performs better than convolutional GAN on this metric.

5.3 Semi-supervised Classification

We evaluate the performance of the convolutional GAN and the proposed CapsuleGAN on semi-supervised classification of MNIST digits. In this experiment, we randomly generate $50,000$ images using both GAN and CapsuleGAN. We use the Label Spreading algorithm [19] with the generated images as the unlabeled examples and n real labeled samples, with $n \in \{100, 1000, 10000\}$. We use the `scikit-learn` [6] package for these experiments. Table 1 shows the results of our experiments. The results show that the proposed CapsuleGAN consistently outperforms convolutional GAN for all the tested values of n with a margin of $1.7 - 3.97$ percentage points. Thus, CapsuleGAN generates images that are more similar to real MNIST images and more diverse than those generated using convolutional GAN, leading to better semi-supervised classification performance on the test dataset.

6 Discussion and Future Work

Generative adversarial networks are extremely powerful tools for generative modeling of complex data distributions. Research is being actively conducted towards further improving them as well as making their training easier and more stable. Motivated by the success of CapsNets over CNNs at image-based inference tasks, we presented the generative adversarial capsule network (CapsuleGAN), a GAN variant that incorporates CapsNets instead of CNNs as discriminators when modeling image data. We presented guidelines for designing CapsuleGANs as well as an updated objective function for training CapsuleGANs. We showed that CapsuleGANs outperform convolutional-GANs on the generative adversarial metric and at semi-supervised classification with a large number of unlabeled generated images and a small number of real labeled ones. This indicates that CapsNets should be considered as potential alternatives to CNNs for designing discriminators and other inference modules in future GAN models.

In this paper, we evaluated the proposed CapsuleGAN model on the MNIST dataset. However, we plan to evaluate it on more complex datasets like CIFAR-10 in the future. We also plan to conduct theoretical analysis of the use of margin loss within the GAN objective. We purposefully did not incorporate many GAN training tricks to fairly evaluate our contributions. The results presented in this paper motivate the use of CapsNets as opposed to CNNs for encoders in GAN variants like BiGAN. We see this as an important direction for future research.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[2] D. Berthelot, T. Schumm, and L. Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[3] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial Feature Learning. In *International Conference on Learning Representations*, 2017.

[4] I. Durugkar, I. Gemp, and S. Mahadevan. Generative Multi-Adversarial Networks. In *International Conference on Learning Representations*, 2017.

---

[6] http://scikit-learn.org/

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[6] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

[7] G. E. Hinton, A. Krizhevsky, and S. D. Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011.

[8] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic. Generating images with recurrent adversarial networks. *arXiv preprint arXiv:1602.05110*, 2016.

[9] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic. Generative adversarial metric. 2016.

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-To-Image Translation With Conditional Adversarial Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[11] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] M. Mirza and S. Osindero. Conditional Generative Adversarial Nets. *arXiv preprint arXiv:1411.1784*, 2014.

[14] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier GANs. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2642–2651, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[15] A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *International Conference on Learning Representations*, 2016.

[16] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative Adversarial Text-to-Image Synthesis. In *Proceedings of The 33rd International Conference on Machine Learning*, 2016.

[17] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3859–3869, 2017.

[18] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc., 2016.

[19] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328. MIT Press, 2004.