Exploring the Efficiency of Capsule Networks in GANs

Nitin Kishore Sai Samala
University of Massachusetts Amherst
nsamala@cs.umass.edu

Sruthi Chilamakuri
University of Massachusetts Amherst
schilamakuri@cs.umass.edu

Abstract

A GAN (Generative Adversarial Network) is a very popular machine learning technique created just two years ago that has found uses in almost every field. It has 2 components which are trained at the same time as adversaries in a minimax game: a generator that attempts to transform samples drawn from a prior distribution to samples from a complex data distribution with much higher dimensionality and a discriminator that decides whether the given sample is real or generated. Traditionally, GANS use CNNs but the internal data representation of a CNN doesn't capture important spatial hierarchies between simple and complex objects while Capsules are groups of neurons that are locally invariant and learn to recognize entities and output activation vectors that represent both the presence and their properties relevant to the visual task. Capsule networks were recently shown to outperform CNNs, we attempt designing a GAN using Capsule networks and employ several techniques devised to stabilize and make training more robust.

1. Introduction

Colloquially, GANs are known to be popular for generating very realistic fake images but a more fascinating use for gans is to learn features from data without any labels (unsupervised). They were initially proposed in [5] and have gone through several modifications and evolutions that it is now hard to keep track of them. The concept is fairly simple. The two components of a GAN, a generator and a discriminator are trained simultaneously so they both improve by competing against one another. GANs have a lot of promising research potential based on the wide range of applications they are employed in, such as face or font generation,painting [23],image completion, content-aware filling, synthetic data generation [20], drug discovery through generation of sample drug candidates, image generation from text description([3], [24], [16]), domain transfer(e.g. style-transfer, pix2pix, sketch2image)([14], [11], [9]), etc. There have been a lot of publications in this field making it a very active research problem.

Recently a new type of neural network called Capsule network [6] was introduced which makes use of dynamic routing by agreement between capsules [17] for learning. The cornerstone of deep learning's surge in popularity undoubtedly must be convolutional neural networks [12] but they do not exist without flaws. The convolutional layer is responsible for learning important high-level features as combinations of simpler features learnt from the lower layers. The setup of a CNN involves using a succession of convolution, non-linear activation and pooling layers repeatedly to reduce the spatial dimensions and the widen the scope of the higher-level neurons that detect higher order features. This gave results outperforming the state-of-the-art at the time albeit losing valuable information from images. The fact remains that orientation and relative spatial relation between components in an image are of less consequence to a CNN. The data representation internal to CNNs doesnt account for important spatial hierarchies between simple and complex objects.

Contrary to producing an image from an internal representation of objects, the human brain deconstructs the visual stimuli into a hierarchical representation matching it with existing learned patterns and relationships. This approach is modeled by a Capsule network to preserve hierarchical pose (translation plus rotation) relationships between object parts. The intuition behind Capsule networks is that a model has to understand and discriminate between different views of the same object i.e., the representation should not depend on viewing angle. Capsule networks can achieve an understanding of 3D space and do it using less data than a CNN requires, which makes them a very desirable alternative.

The fundamental unit of this network is called a capsule. Instead of outputting a single scalar value like a neuron, a capsule outputs a vector that encapsulates not only presence but also information about the state of the detected feature under consideration [7]. The probability of detection of this feature is encoded as the length of the vector while its orientation gives us the state of the feature. This makes feature detection pose invariant and allows for more powerful representational capabilities. Currently Capsule Networks are slower than most models and have been shown to do well on simpler datasets.

In this project, we attempt to take the evolution of GANs a bit further chronologically and aim to use Capsule networks to learn features from images. However, GANs are notorious for being difficult to train due to various factors such as minimax optimization, vanishing gradients causing unstable training, mode collapse, etc. Hence our goal is to implement several training hacks outlined in [18], [21], [1], [19], [22] to improve the fundamental stability of a ”CapsuleGAN” and make training more robust. We also hope to generate better results on more complex image datasets than MNIST such as CIFAR-10 and celebrities face dataset for this project.

2. Related Work and Background

Over the past few years a lot of relevant progress has been made and different types of GANs were proposed. The first significant enhancement on vanilla GAN that went on to become a solid baseline to implement and compare future models with, is the Deep Convolutional GAN. Deep Convolutional GAN (DCGAN) puts more emphasis on using ReLu activations and Batchnorm layers while avoiding fully-connected hidden layers and pooling layers in the convolutions. This architecture was further improved in [18] allowing generation of better high-resolution images by addressing the issues of training stability and convergence. Some of these enhancements are what we hope to apply to our model as well. DCGANs do not suffer from mode collapse [15] when generating complex images similar to that of the CIFAR-10 dataset which propelled more widespread adoption of GANs in novel applications and implementations.

Zi Yi Dou, (2017) proposed a new framework to train discriminators dynamically based on the distance between generated samples and real samples, called Metric Learning based Generative Adversarial Network (MLGAN) [4]. MLGANs update the generator under the newly learned metric and avoid the instability issues caused by the sigmoid cross entropy loss function used in traditional GAN discriminators. This approach has been found to empirically increase the stability of training in GANs. Another modification to the loss function involves including the Wasserstein distance [2] which correlates with image quality. This distance between points from one distribution to another, replaces Jensen-Shannon divergence as an objective function that greatly improves training stability. This approach also improves convergence by offering a numerical value that interprets how well the parameters are tuned and thus eliminates the need to constantly evaluate the generated samples for signs of convergence. Our approach is to use Capsule Networks instead of CNN for discriminators in our GAN.

3. Methodology

A discriminative model involves evaluating the conditional probability of a target random variable Y given an observed random variable X, while a generative model entails evaluating the joint probability P(X,Y). We can interpret images as samples from a probability distribution. In a generative adversarial network, both the discriminator and the generator are competitors trained simultaneously in a minimax game. Initially, the generator samples a vector noise Z from a simple distribution, say Gaussian, and then upsamples this vector up to an image which is predominantly noise at first. As the generator keeps learning, the discriminator classifies the generated images and the real images. This result is propagated backwards in a feedback loop enabling the generator's distribution to get as close as possible to the distribution of real images while the discriminator is trained to identify the fakes. This process continues until the probability of classification of an image becomes 0.5, indicating that the generated image can pass for a real one.

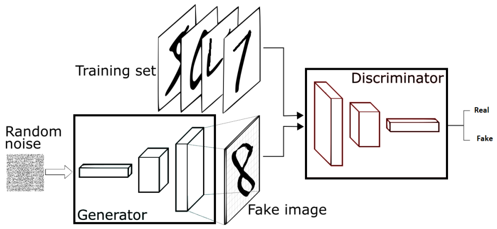


Figure 1. GAN architecture

3.1. Model

3.2. Capsule Network

Several components of the Capsule Network are described in further detail in the subsequent sections.

3.2.1 Capsule Architecture

A capsule is the most basic unit of Capsule networks that takes in a weighted sum of output vectors from the capsules in the layer below and produces an output vector. For a capsule s_j the input is given by:

$$s_j = \sum_i c_{ij}\hat{u}_{j|i} \;\; where \;\; \hat{u}_{j|i} = W_{ij}u_i$$

$\hat{u}_{j|i}$ are the output vectors from capsules in the layer below obtained by computing dot product with the weight matrix.

c_{ij} is the coupling coefficient between capsule s_j and each of the i capsules in the previous layer. The coupling coefficients sum to one and are the softmax probabilities of logits b_{ij}. Initial values for b_{ij} can all be zero indicating that all capsules have equal affinity to all parent capsules or can be initialized to values from a prior distribution that captures affinity between capsule i and parent j.

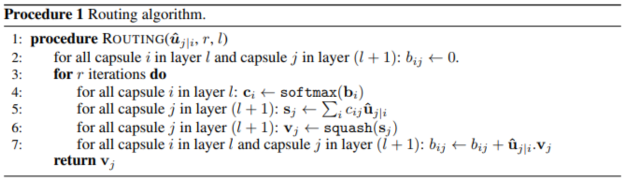
The output vector also known as the prediction vector v_j is computed using the following non linearity (a method termed squashing):

$$v_j = \frac{||s_j||^2}{1 + ||s_j||^2} \frac{s_j}{||s_j||}$$

Once the prediction vector is computed, the logits b_{ij} are updated with the agreement: the product $v_j \hat{u}_{j|i}$. If the product is high, then the low level features detected by i capsules are important to parent capsule j and naturally the coupling coefficients b_{ij} must be increased proportionally to propagate relevant information to the parent through routing by agreement.

3.2.2 Routing by Agreement

The dynamic routing by agreement algorithm in [17] is presented below:



Procedure 1 Routing algorithm.

```
procedure ROUTING($\hat{\mathbf{u}}_{j|i}$, $r$, $l$)
    for all capsule $i$ in layer $l$ and capsule $j$ in layer $(l+1)$: $b_{ij} \leftarrow 0$.
    for $r$ iterations do
        for all capsule $i$ in layer $l$: $\mathbf{c}_i \leftarrow \texttt{softmax}(\mathbf{b}_i)$
        for all capsule $j$ in layer $(l+1)$: $\mathbf{s}_j \leftarrow \sum_i c_{ij} \hat{\mathbf{u}}_{j|i}$
        for all capsule $j$ in layer $(l+1)$: $\mathbf{v}_j \leftarrow \texttt{squash}(\mathbf{s}_j)$
        for all capsule $i$ in layer $l$ and capsule $j$ in layer $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i}.\mathbf{v}_j$
    return $\mathbf{v}_j$
```

Figure 2. Routing by Agreement. Sabour, Frosst et al.”Dynamic Routing Between Capsules.”(2017)

3.2.3 Loss Function

A separate margin loss is used for each of the k classes in the final layer of the Capsule network.

$$L_k = T_k \max(0, m^+ - ||v_k||)^2 + \lambda(1 - T_k) \max(0, ||v_k|| - m^-)^2$$

T_k is an indicator variable which is equal to 1 if object of class k exists in the image and 0 otherwise. m^+ and m^- are hyperparameters and are set to 0.9 and 0.1 respectively. As mentioned in [17] the second term of the loss function concerning absent object classes helps prevent shrinking of the lengths of prediction vectors for these classes during initial stages of learning. Thus the total loss is the sum of k margin losses, one for each class.

3.3. GAN Architecture

4. Dataset

We have three datasets of interest in this project.The first is MNIST, a large collection of black and white handwritten digit images, widely popular and used for generating baselines in many academic research papers. Baseline and results in [17] were computed on the MNIST dataset. MNIST consists of 60,000 training images and 10,000 testing images.

Another dataset we will test our model on is the CIFAR-10 dataset which consists of 32x32 color images of 10 classes (airplane, automobile, bird, cat, deer, dog, horse, frog, ship and truck) with 6000 images per class. There are 50000 training images and 10000 test images in this dataset. CIFAR-10 is well recognized and has been used for computing baselines in several publications involving convolutional neural networks and computer vision.

The third dataset is the large-scale CelebFaces Attributes (CelebA) dataset [13]. It contains 202,599 diverse celebrity face images. Researchers at Nvidia recently used this dataset to progressively train a GAN by using low-resolution images first and gradually increasing image resolution and network size until they were able to generate very realistic faces [10]. This offers a challenge and it would be interesting to see how Capsule networks do on such a complex dataset.

5. Training

It is recommended to use 3 routing iterations for training. [1]

6. Evaluation

We want to identify improvements in GANs both in terms of training stability and quality of the samples generated so we consider using inception scores as one evaluation metric [18]. Human evaluation is prone to high variance and is expensive. In the beginning, Goodfellow et al. (2014) evaluated GANs using a comparison of the generated data sample and its nearest neighbors in the data. Evaluating the quality of images is very subjective and therefore it is hard to reach a general consensus on what is considered good. However, a GAN is ultimately meant to learn the hidden generative distribution so we can focus on measuring how close the model distribution is to the real distribution. For datasets like CIFAR-10 and MNIST, semi-supervised classification can be done by providing raw pixel values to any general classification algorithm using Label Spreading algorithm from [25] with the outputs of our model as the unlabeled examples and some real labeled samples. Alternatively generative adversarial metric (GAM) [8] can be considered. The idea behind this pairwise metric is to swap discriminators of two GANs and train the generators against their new counterparts in the same minimax fashion. We

[1] Code for this project will be maintained on https://github.com/snknitin/DeepfakeCapsuleGAN

h!

Dataset	Inception Score(mean+std) for 100 samples
MNIST	
CIFAR-10	
Celeb Faces	

Table 1. Results

can compare our model with an improved DCGAN using the GAM values to see if our model outperforms it.

7. Results

Please refer Table 1 for results. To be updated

8. Conclusion and Future Work

One possible extension we have identified as a potential future project would be to generate a large number of good quality face images and use them to employ a deep fake to embed the new character in an existing video. Future work includes using Capsule networks for the generator as well,

References

[1] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

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

[3] A. Dash, J. C. B. Gamboa, S. Ahmed, M. Z. Afzal, and M. Liwicki. Tac-gan-text conditioned auxiliary classifier generative adversarial network. *arXiv preprint arXiv:1703.06412*, 2017.

[4] Z.-Y. Dou. Metric learning-based generative adversarial network. *arXiv preprint arXiv:1711.02792*, 2017.

[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] G. Hinton, N. Frosst, and S. Sabour. Matrix capsules with em routing. 2018.

[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. Generative adversarial metric. 2016.

[9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[10] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[11] T. Kim, M. Cha, H. Kim, J. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[13] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[14] F. Luan, S. Paris, E. Shechtman, and K. Bala. Deep photo style transfer. *CoRR, abs/1703.07511*, 2017.

[15] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[16] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 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, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[19] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016.

[20] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, page 6, 2017.

[21] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

[22] T. White. Sampling generative networks: Notes on a few effective techniques. *arXiv preprint arXiv:1609.04468*, 2016.

[23] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5485–5493, 2017.

[24] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE Int. Conf. Comput. Vision (ICCV)*, pages 5907–5915, 2017.

[25] 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*, pages 321–328, 2004.