Post-Training Quantization of Generative Adversarial Networks

Lecture
Practical Applications of Deep Learning
Summer Semester 2020

Johannes Hötter, Samik Real-Enríquez, Henrik Wenck

Tutor:
Gonçalo Mordido

August 31, 2020

Contents

1 Introduction

In recent years, Machine Learning has enabled some significant breakthroughs in areas such as Computer Vision, Natural Language Processing or Robotics. In particular, Artificial Neural Networks (ANN) have been used to recognize complex patterns in large datasets. Due to their often large number of layers, this area is referred to as Deep Learning.

What is often neglected, however, is the actual complexity of the structure that artificial neural networks can exhibit. If ANNs consist of several layers, each containing hundreds of units, they become large in size. In many real-world applications, the size of a model becomes important - may it be due to economic reasons such as the needed computational power, or due to practical reasons such as the runtime of a model on a mobile device.

For this reason, the Post-Training Quantization of deep ANNs is an important research topic. Following this approach, complex models are reduced in their size while trying to maintain the performance they can achieve in their output. In this report, we are demonstrating the effects of quantization on so-called Generative Adversarial Networks (GAN).

2 Generative Adversarial Networks

GANs are a subclass of ANNs. Introduced in 2014 by Goodfellow et al [GPAM$^+$14] they are known to produce realistic-looking fake data from randomized input.

They are trained in an adversarial manner called two-player minimax-game. The actual generating model G is one of two models that are being trained. Given some randomized input, such as a 100-dimensional vector from a Gaussian distribution, this model creates an output. The second model acts as a discriminator D. Its goal is to differentiate fake data from real data. It receives as input samples from the original data distribution $p_{data}(x)$ or from the generator which in turn receives a sample out of the noise distribution $p_z(z)$, and must predict the set to which it believes the content is coming from.

In order to understand the minimax-game in more depth, one must understand the mathematical derivation. All starts with the binary cross-entropy loss:

$$L(\hat{y}, y) = -(y \cdot \log(\hat{y}) + (1-y) \cdot \log(1-\hat{y})) \tag{1}$$

If we now assume that y is the original sample and $\hat{y}$ is the generated one we can consider 2 cases. If the image comes from the training data, it carries the label y=1 and $\hat{y}$ is therefore D(x)

$$L(D(x), 1) = -\log(D(x)) \tag{2}$$

Otherwise, if the image comes from the generator, the label is y=0 and $\hat{y}$ =D(G(x)) :

$$L(D(G(z)),0) = -\log(1-D(G(z))) \tag{3}$$

Since both cases occur one can combine them:

$$L(D,G) = -(\log(D(x)) + \log(1-D(G(z)))) \tag{4}$$

The game is on. The Discriminator D now tries to minimize the loss, i.e. to predict the right label of the incoming image each time.

$$L^{(D)} = \max[\log(D(x)) + \log(1-D(G(z)))]^1 \tag{5}$$

At the same time the generator G tries to fool the discriminator as often as possible to maximize the loss.

$$L^{(G)} = \min[\log(D(x)) + \log(1-D(G(z)))]^1 \tag{6}$$

However, since they are playing simultaneously against each other and not alone, both formulas must be combined again. If the discriminator and the generator would only play the game once, the formula would be

$$\min_G \max_D V(D,G) = \min_G \max_D [\log(D(x)) + \log(1-D(G(z)))] \tag{7}$$

Since they do it more often, it results in:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log(D(x))] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{8}$$

which brings us to the initial equation in the original paper by Goodfellow et al [GPAM+14].

As with other ANNs, at first both models are creating arbitrary outputs, as their parameters are initialized. From time, however, as one of the two models improves, the other model needs to improve too. If the discriminator is able to differentiate real from fake data, the generator must produce better content to fool the discriminator. If the generator is good enough to fool the discriminator, it must improve in focusing on details in the content that differentiate between fake and real data.

For content-creation applications, the goal is to achieve an equilibrium. In this state, the discriminator will be correct in half of the cases, i.e. it has to guess whether data is real or generated.

In 2018, Karras et al developed StyleGAN [KLA18]. StyleGAN is a novel architecture following the GAN approach. Figure 1 shows exemplary outputs of the generative model.

[1] minus makes a min problem to a max problem and vice versa.

Figure 1: Exemplary images generated by StyleGAN

StyleGAN however consists of millions of parameters. In its original form, it can't be used for inference on e.g. mobile devices. Hence, such models need to be analyzed using quantization techniques to widen the space for real-world applications.

3 Analysis of quantization

The following two subsections demonstrate the used approach for this seminar as well as the scores used for an automated metric-based analysis of quantized GANs.

3.1 Quantization approach

At its core, quantization is a mapping from a continuous space M to a discrete space N, and then from N back to M again while transforming values during those transformation [WWJ+19]. This mapping Q usually consists of three steps: scaling, rounding, and re-scaling:

$$Q(x) = f^{-1}(round(f(x)))$$

where x is a full-precision value from M. f is the function for scaling. We chose the Q-min-max function for this:

$$f(x) = \frac{x - min(x)}{max(x) - min(x)} \times (2^k - 1)$$

where k is the desired bit size. For instance, an array of full-precision values of each 32 bits can be transformed into an array of each 4 bits. Of course, some Precision is lost during this transformation - still, the values are as close as possible to their original values.

This approach can be used to transform the parameters of an ANN. A trained ANN has its parameters as arrays of full-precision values of typically either 32 or 64 bits.

The goal of quantization therefore is to reduce the size of a model without losing too much parameter information. In this report, we analyze how the quantization of GANs affect their outputs. In order to be able to do this scientifically, we need quantifiable metrics.

3.2 Automated analysis of image quality and diversity

One obstacle in the analysis of quantized generators is to mimic human analysis. When we as humans take a look at images, we can say if an image is of high quality, or if several images are of diverse classes. For images created by a generative network, this is substantially more difficult since GANs lack an objective function to compare performances of different models [SGZ+16]. A generated image has no assigned class by default. During quantization, the complexity of patterns a generative model can create decreases as the bit size of its parameters shrink. Still, one wants to be able to tell how the images change with respect to their quality and diversity.

There are some approaches which correlate with the human analysis of images, which will be considered in this report. First, there is the so-called Inception Score [SGZ+16]. In its core, the Inception score uses a pre-trained Inception-V3 network to calculate the distance between activation distributions and thus assign class probabilities for a given set of images. If for each single image the class probabilities are tending towards one definitive class, then one can say that the quality of the images generated is high. If the class distributions are uniformly distributed over the whole set of images, one can say that the generated images are diverse. The Inception Score combines both of these aspects in a single score using the Kullback-Leibler Divergence.

Another approach, which is considered to be more accurate than the Inception Score, is the K Nearest Neighbor-based Precision and Recall Score [KKL+19]. In this approach, an estimated manifold is being created using the distributions of real and generated images. To calculate the Precision, for each generated image one needs to indicate whether one of the K Nearest Neighbors is an arbitrary image of the set of real images. The Precision then is the number of images with that condition being true divided by the number of generated images. For Recall, it's vice versa (i.e. for each real image, check whether a K Nearest Neighbor is of the set of generated images).

Last, there is the Realism score, which is derived from the K Nearest Neighbor-based Precision and Recall. The Realism score can be described as a metric that increases as a an image gets closer to a manifold and decreases as it gets further away from it. Thus, it can be understood as a continuous version of the binary indicator for the Precision and Recall Scores.

4 Methods

For our report, ANNs have been trained and images were generated. Then the weights of the models have been quantized with bit sizes ranging from 8 to 3 bits and their generated images as well as the generated images from the non-quantized models have been submitted to quality and diversity metric tests. Lastly, the quality and diversity metrics from the images of the non-quantized models were compared with the images from the quantized models.

For the experiments three datasets have been chosen: MNIST with 60,000 training and 10,000 test 28 X 28 grey scale images, Fashion-MNIST also with 60,000 training and 10,000 test 28 X 28 grey scale images and CIFAR-10 with 50,000 training and 10,000 test 32 X 32 RGB images [LCB99, XRV17, KH+09]. All three of these datasets have 10 different classes.

For the MNIST dataset two model architectures have been chosen. The first one is a fully connected architecture (FC) with 4 layers. The second architecture is a convolutional neural network (CNN) with 5 layers. For the Fashion-MNIST and CIFAR-10 datasets only a CNN model was used. Both of these datasests used the same model with 3 convolution layers. All architectures were trained for 200 epochs.

A total of 20,000 samples were used for each metric. For the Inception Score only generated images were used and for Precision and Recall half of the samples were from the generated images and the other half were real images from the respective dataset.

5 Results and Discussion

The results of the experiments specified in section 4 can be seen in Figure 2. Each row represents a different dataset-model combination and in each column, generated images from models with specific quantized bit size weights can be seen. An original image sample from each dataset is also included in the last column.

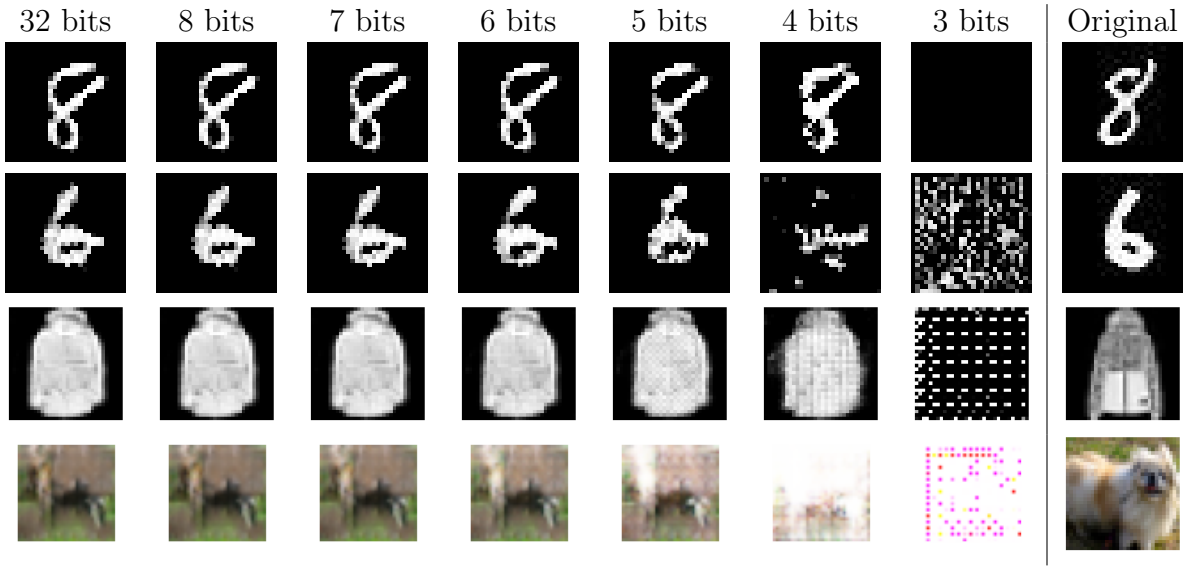

Figure 2: Generated images with different quantized model-weight bit sizes. Each row corresponds to a different dataset-model combination (from top to bottom): MNIST FC, MNIST CNN, Fashion-MNIST and CIFAR-10. Generated images from models with specific quantized bit size weights can be seen on each column. The last column corresponds to the original image from each dataset.

5.1 Inception Score

As observable in Figure 3, the Inception Score is rather stable throughout the various quantized models. For the MNIST dataset, for both FC and CNN architectures the model produces quite good results until a bit size of 5. Taking a look at the examples from above, this makes sense. A human could still easily identify the number displayed. However, the score is hard to interpret as its not clearly indicating either the quality or diversity. For a quantization of less than 5 bits, the score plummets. Here it is interesting to see, that for 4 bits, the FC model performs much better than the CNN model. This might be due to the fact that compressing CNNs could be harder in general.

For Fashion-MNIST, the quantization seems to be stable too. Interestingly, there even is a small increase in the Inception Score from bit size 6 to bit size 5. This might be due to the cause that with less bit size, images might be misunderstood to belong to another class, increasing the Recall, while still maintaing a rather well image quality. This is can be the effect of the Inception Score, as it essentially combines two metrics (image quality and diversity).

For the CIFAR-10 dataset, the model contains a good relative Inception Score even after quantization to 4 or less bits. This might be due to the cause that the model keeps a high confidence even if the image is in low quality, as the original images are already

hard to classify on human eyesight. The relative loss in the image quality and diversity therefore is more difficult to analyze. Hence, the Score is at a relatively high value even after strong quantization.

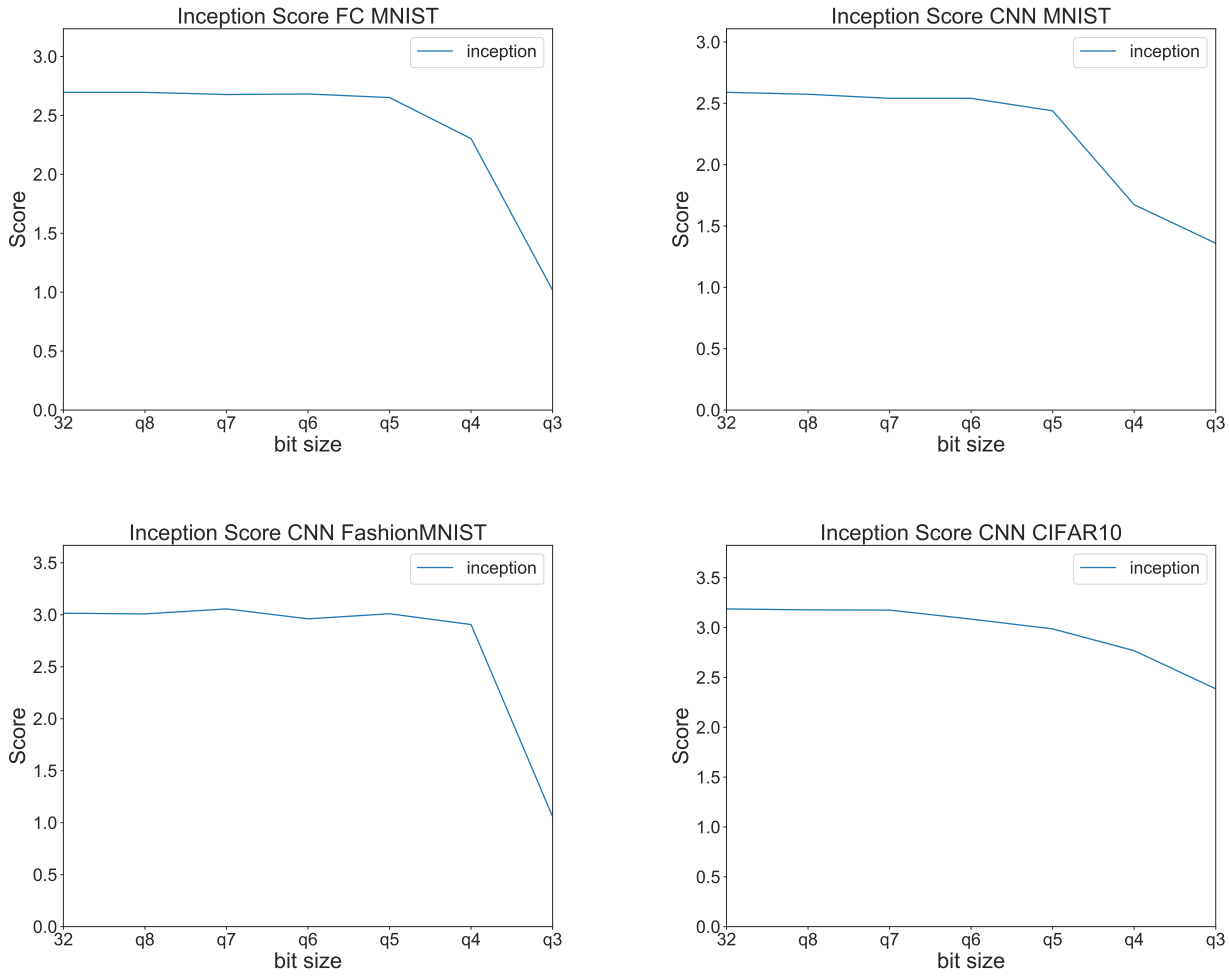

Figure 3: Inception score for datasets MNIST FC, MNIST CNN, Fashion-MNIST and CIFAR10 (from left to right, top to bottom). The x-axis on each plot represents the number of bits used for model-weight quantization.

5.2 K Nearest Neighbor-based Precision and Recall

Figure 4 showcases the results for the Precision and Recall metric for the experiments.

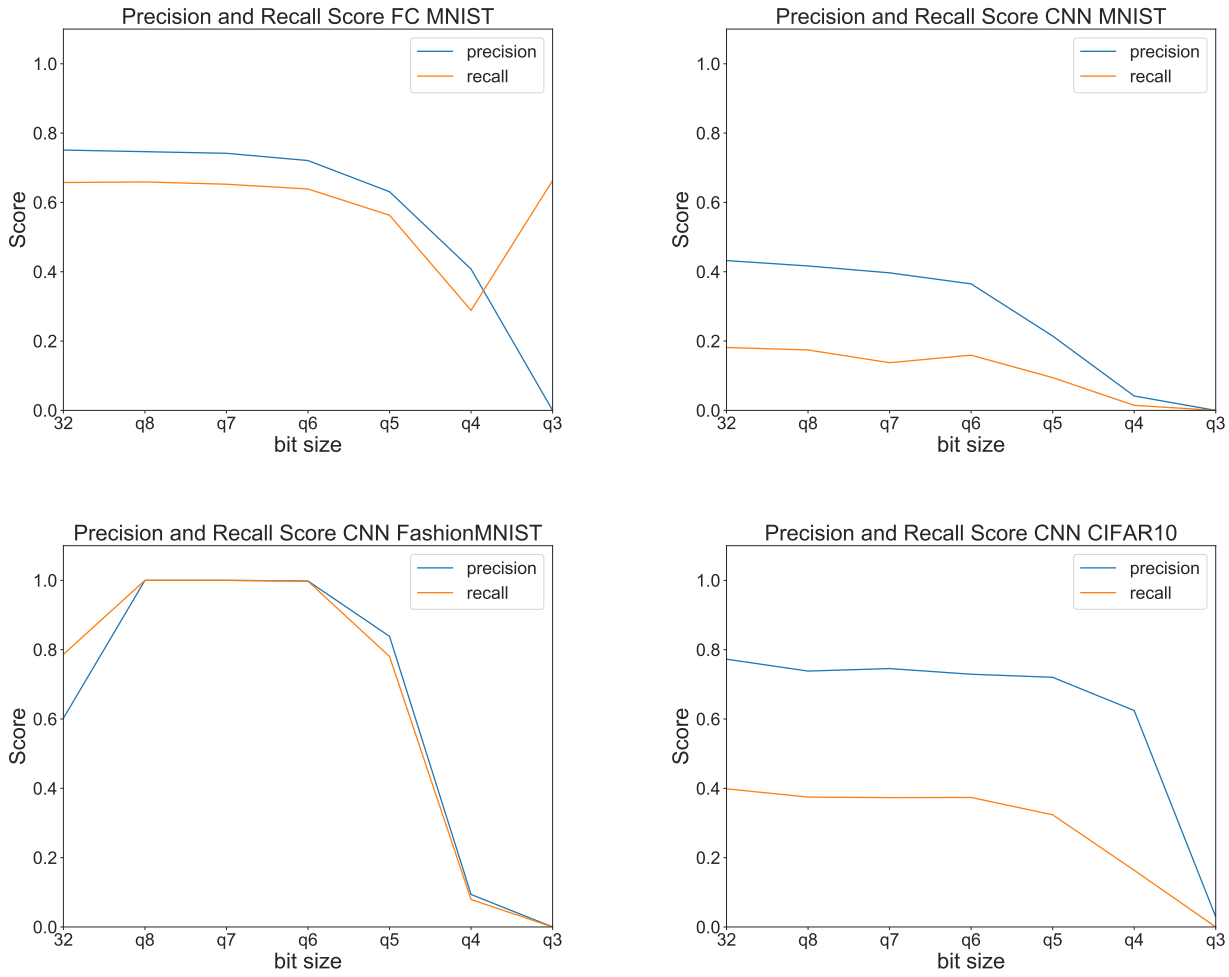

Figure 4: Precision and Recall Scores for datasets MNIST FC, MNIST CNN, Fashion-MNIST and CIFAR10 (from left to right, top to bottom). The x-axis on each plot represents the number of bits used for model-weight quantization.

The Precision metric for FC MNIST, CNN MNIST and CIFAR-10 correlates with the images, showing a gradual decrease in image quality the lower the bit size. Sharp decrease start at 6 and 5 bits, which is a little more conservative than the Inception Score and correlates a little better with the human quality assessment of the images. The increase in Precision on the Fashion-MNIST between the 32 and 8 bit images can be explained due to the low number of image samples used for the metric, thus creating random noise inside of the manifold used to determine the Precision metric. This behaviour is

rare since noise is unlikely to fall inside of the manifold. This problem is assumed to disappear when using a higher number of samples for the metric.

The Recall metric, which assesses image diversity, decreases gradually the less bits used. But this metric also behaves unexpectedly, randomly increasing its metric value on lower bit quantizations. This would means that the generated images become more diverse in certain bit quantizations. But this is identified again as an error due to the fact that this is an isolated point in each dataset, while the tendency shows a decrease. This can also be explained by the lack of samples when calculating the metric and by noise points landing inside of the manifolds by chance.

5.3 Realism Precision and Recall

Realism Score graphs are presented in Figure 5. The Realism Score is the continuous representation of the Precision and Recall Scores from Figure 4. The closer an image is to the manifold, the larger the Realism Score will be. Since this score represents a distance, the value range will vary from different datasets. In general the Realism Scores for Precision and Recall metrics have a very similar form in comparison to the discrete Precision and Recall Scores from Figure 4. The only exception to this is the Fashion-MNIST dataset which has a completeley different shape. The Realism Precision and Recall Scores of this dataset are also exactly the same. No possible explanations for the vast difference in form for this dataset could be found.

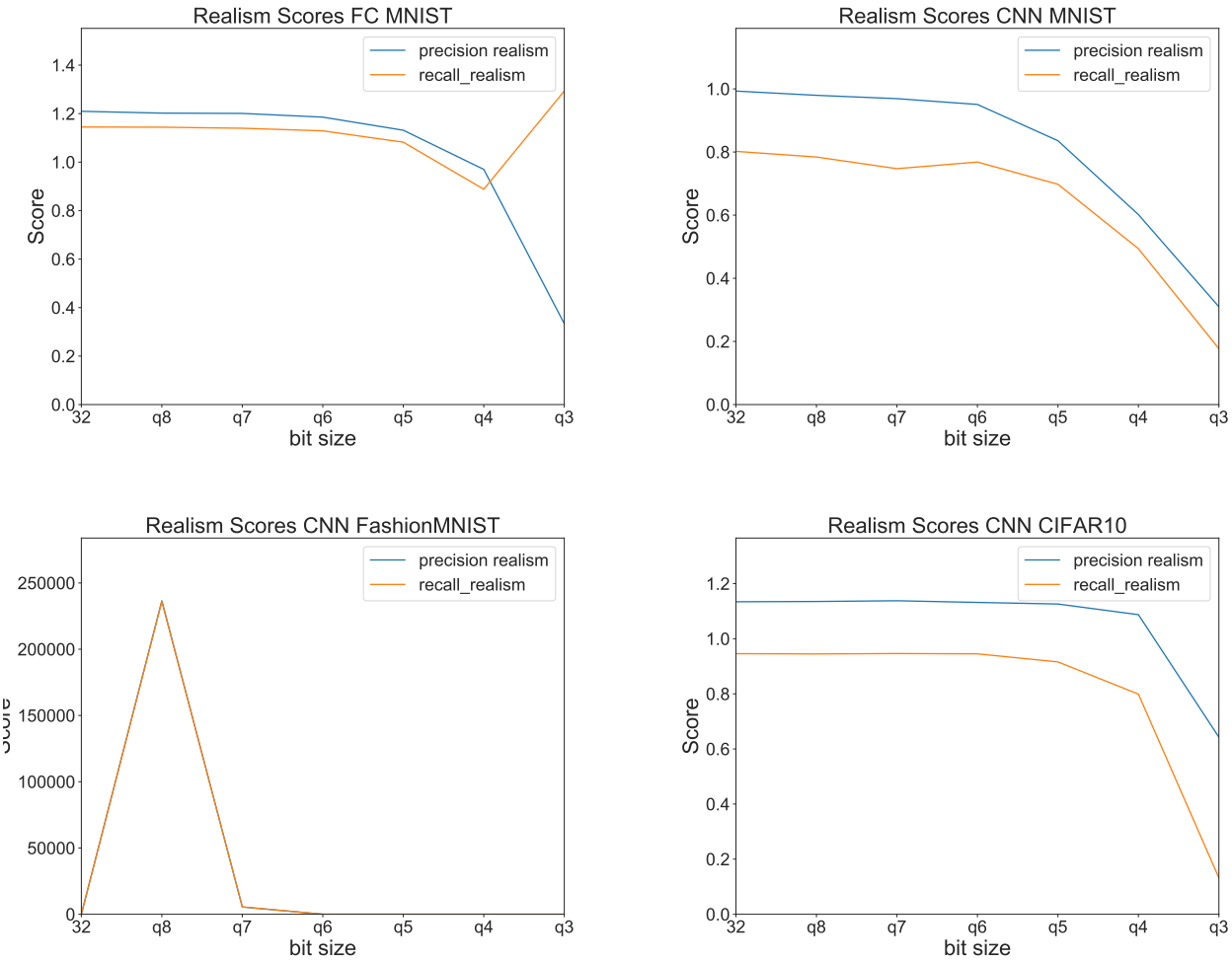

Figure 5: Realism Scores for Precision and Recall for datasets MNIST FC, MNIST CNN, Fashion-MNIST and CIFAR10 (from left to right, top to bottom). The x-axis on each plot represents the number of bits used for model-weight quantization.

6 Conclusion

During this seminar, we identified patterns in the quantization of GANs. First, we realized that the automated analysis of images is a non-trivial task. The Inception Score regularly correlates with human behaviour, but didn't work too well during quantization. As the image quality and diversity went significantly down by eyesight, the Inception Score was still quite good. The K Nearest Neighbor-based Precision and Recall Scores turned out to be more useful.

Based on both the analysis of Precision and Recall Scores, as well as pure eyesight, we were able to identify that shrinking the bit sizes of the model parameters resulted in worse model outputs. However, the amount by which the parameters can be shrinked without significantly worsening results has been astonishing. We were able to reduce the bit size from 32 to values around 6 to 5 while keeping both good image quality and diversity.

In our experiments, we also examined that during quantization, both Recall and Precision went down rather simultaneously. It did not happen that either Precision dropped while Recall stayed the same nor vice versa.

7 Future Work

An experiment that might be beneficial for image quality and diversity assessment is pre-training the VGG-16 network with the dataset that will be subject to the metric evaluations. This way it will be easier for the metric to determine diversity and quality of the generated images.

Using other quantization approaches will also allow for a better understanding of the influence of quantization on deep networks. Other quantization approaches include log/tahn quantization or Outlier Channel Splitting (OCS) [ZHD+19], Analytical Clipping for Integer Quantization (ACIQ) [BNHS18], QGAN [WWJ+19]. This last one shows promising results because of the quality it maintains. Additionally quantization of all other bit sizes that were not covered in this paper can be tested, using the Multi-Precision algorithm [WWJ+19] to identify the optimal bit size for generated image quality and diversity retainment. It would also be a good idea to revise the implementation of the Realism Score in Fashion-MNIST to identify the reason for the behaviour of this curve in Figure 5.

References

[BNHS18] Banner, Ron ; Nahshan, Yury ; Hoffer, Elad ; Soudry, Daniel: ACIQ: analytical clipping for integer quantization of neural networks. (2018)

[GPAM+14] Goodfellow, Ian ; Pouget-Abadie, Jean ; Mirza, Mehdi ; Xu, Bing ; Warde-Farley, David ; Ozair, Sherjil ; Courville, Aaron ; Bengio, Yoshua: Generative Adversarial Nets. Version: 2014. http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf. In: Ghahramani, Z. (Hrsg.) ; Welling, M. (Hrsg.) ; Cortes, C. (Hrsg.) ; Lawrence, N. D. (Hrsg.) ; Weinberger, K. Q. (Hrsg.): *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, 2672–2680

[KH+09] Krizhevsky, Alex ; Hinton, Geoffrey u. a.: Learning multiple layers of features from tiny images. (2009)

[KKL+19] Kynkäänniemi, Tuomas ; Karras, Tero ; Laine, Samuli ; Lehtinen, Jaakko ; Aila, Timo: Improved precision and recall metric for assessing generative models. In: *Advances in Neural Information Processing Systems*, 2019, S. 3927–3936

[KLA18] Karras, Tero ; Laine, Samuli ; Aila, Timo: *A Style-Based Generator Architecture for Generative Adversarial Networks*. 2018

[LCB99] Lecun, Y ; Cortes, C ; Burges, CJC: *The MNIST Dataset of Handwritten Digits(Images)*. 1999

[SGZ+16] Salimans, Tim ; Goodfellow, Ian ; Zaremba, Wojciech ; Cheung, Vicki ; Radford, Alec ; Chen, Xi: Improved techniques for training gans. In: *Advances in neural information processing systems*, 2016, S. 2234–2242

[WWJ+19] Wang, Peiqi ; Wang, Dongsheng ; Ji, Yu ; Xie, Xinfeng ; Song, Haoxuan ; Liu, XuXin ; Lyu, Yongqiang ; Xie, Yuan: QGAN: Quantized Generative Adversarial Networks. In: *CoRR* abs/1901.08263 (2019). http://arxiv.org/abs/1901.08263

[XRV17] Xiao, Han ; Rasul, Kashif ; Vollgraf, Roland: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. In: *arXiv preprint arXiv:1708.07747* (2017)

[ZHD+19] Zhao, Ritchie ; Hu, Yuwei ; Dotzel, Jordan ; De Sa, Christopher ; Zhang, Zhiru: Improving neural network quantization without retraining using outlier channel splitting. In: *arXiv preprint arXiv:1901.09504* (2019)