Quantized Generative Models for Solving Inverse Problems

Nareddy Kartheek Kumar Reddy, Vinayak Killedar, Chandra Sekhar Seelamantula
Department of Electrical Engineering, Indian Institute of Science, Bengaluru, 560012.
nareddyreddy@iisc.ac.in, vinayak.killedar@gmail.com, css@iisc.ac.in

Abstract

Generative priors have been shown to be highly successful in solving inverse problems. In this paper, we consider quantized generative models i.e., the generator network weights come from a learnt finite alphabet. Quantized neural networks are efficient in terms of memory and computation. They are ideally suited for deployment in a practical setting involving low-precision hardware. In this paper, we solve non-linear inverse problems using quantized generative models. We introduce a new meta-learning framework that makes use of proximal operators and jointly optimizes the quantized weights of the generative model, parameters of the sensing network, and the latent-space representation. Experimental validation is carried out using standard datasets – MNIST, CIFAR10, SVHN, and STL10. The results show that the performance of 32-bit networks can be achieved using 4-bit networks. The performance of 1-bit networks is about 0.7 to 2 dB inferior, while saving significantly (32×) on the model size.

1. Introduction

Consider $\boldsymbol{x} \in \mathbb{R}^n$ compressed to measurement $\boldsymbol{y} \in \mathbb{R}^m$, $m \ll n$, through a linear or nonlinear measurement operator $\mathcal{A}$. The problem is ill-posed since $m \ll n$, and infinite solutions exist. Stated formally,

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}), \tag{1}$$

where $\mathcal{A}$ is the sensing operator. In compressed sensing, the sensing operator is linear and involves a matrix multiplication: $\mathcal{A}(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x}$, whereas in compressed phase retrieval: $\mathcal{A}(\boldsymbol{x}) = |\mathbf{A}\boldsymbol{x}|^2$. In neural network based sensing, $\mathcal{A} = \mathcal{A}_{\boldsymbol{\phi}}$ is a network with parameters $\boldsymbol{\phi}$.

The reconstruction of $\boldsymbol{x}$ from measurement $\boldsymbol{y}$ involves two steps — enforcing measurement consistency and structural constraints. Let $\hat{\boldsymbol{x}}$ be the reconstructed signal, then first step is to ensure the measurement consistency, i.e. $\mathcal{A}_{\boldsymbol{\phi}}(\hat{\boldsymbol{x}})$ has to be close to $\boldsymbol{y}$ in minimum ℓ_2-norm sense. The second step imposes the constraint that $\hat{\boldsymbol{x}}$ has the desired structure, which in our case, $\hat{\boldsymbol{x}}$ must be an image. We rely on quantized generative models as an image prior to satisfy the conditions from the second step.

Generative models for solving inverse problems: Generative models have been used to solve inverse problems [2,4,6,12–16,33] and are shown to outperform the traditional optimization methods using a small number of measurements. Bora *et al.* enforced the constraint that the signal $\boldsymbol{x}$ lies in the range-space of the pre-trained generator network $\mathbf{G}_{\boldsymbol{\theta}}$ [4]. Yan *et al.* introduced meta-learning in the context of compressed sensing to jointly optimize the generator weights and latent space [33]. Killedar *et al.* enforced sparsity in the latent space to obtain superior performance over the state-of-the-art methods [14–16]. In a survey paper, Zhao *et al.* discusses the use of generative models for solving inverse problems [36] ranging from X-ray computed tomography to synthetic aperture radar. Song et al. used pre-trained diffusion models to solve the nonlinear inverse problems [28].

Quantization in neural networks: The downside with neural networks is the large model size, and significant computational requirement. Quantization of weights is a promising strategy to reduce the overall model footprint and to simplify arithmetic operations. Neural networks with binary parameters and/or activations (BNNs) have been shown to be promising for solving classification problems [10,11,26,35,37], continual learning [21], language modeling [9,24], semantic segmentation [32], video processing [23], compressed image recovery [27], etc. In the context of generative modeling, Dong and Yang [7] use binary neurons at the output layer of the generator network for modeling discrete distributions. Using linear combinations of $2k$ binary values, Wan *et al.* were able to compress the generative models effectively [29]. Wang *et al.* developed a variant of the expectation-maximization (EM) algorithm for training the quantized GANs [31]. By utilizing multiple compression techniques such as model distillation, channel pruning, and quantization, GAN slimming method achieved 47× reduction in model size [30]. Andreev [1] experimented with various quantization methods specific to StyleGAN, Self-Attention GAN, and CycleGAN for efficient inference.

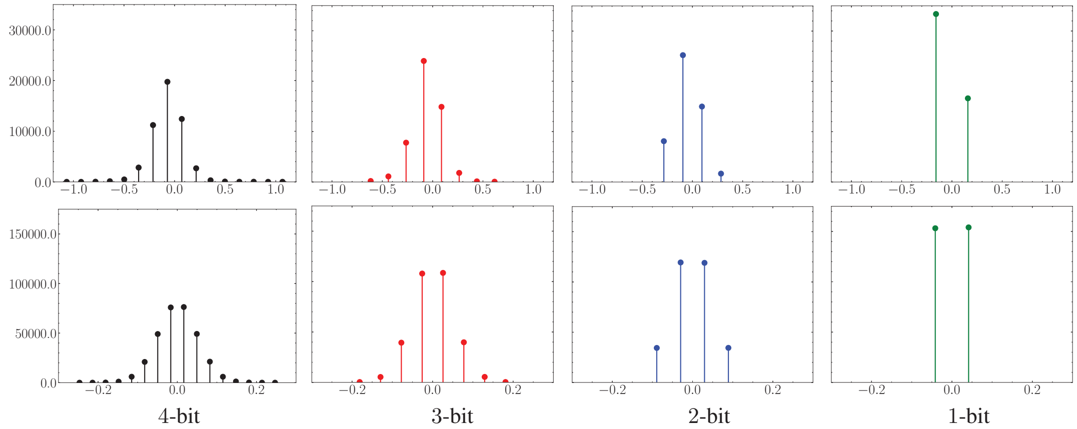

Figure 1. The first row shows the distribution of quantized weights taken from a fully connected network used as the generator for MNIST dataset. The second row shows the distribution of quantized weights from DCGAN generator employed in CIFAR10, SVHN, and STL10 datasets. The presence of a few large weights in 4-bit networks resulted in their superior performance over the 1-bit counterparts.

2. Quantized Generative models as an Image Prior for Solving Inverse Problems

In this paper, we use quantized generative models ($\mathbf{G}_{\boldsymbol{\theta}_Q}$) as an image prior. We investigate the performance-quantization trade-off when the generator weights are quantized to a low precision: 1, 2, 3, and 4 bits. We refer to the quantized generative model as Q-GEN. The sensing operator $\mathcal{A}_{\boldsymbol{\phi}}$ is a neural network with parameters $\boldsymbol{\phi}$. We pursue the following optimization objective:

$$\min_{\boldsymbol{z},\boldsymbol{\theta}_Q,\boldsymbol{\phi}} \|\boldsymbol{z}\|_0 \text{ s.t. } \boldsymbol{y} = \mathcal{A}_{\boldsymbol{\phi}}(\boldsymbol{x}), \;\; \boldsymbol{x} = \mathbf{G}_{\boldsymbol{\theta}_Q}(\boldsymbol{z}). \quad (2)$$

Solving for the optimal network parameters $\boldsymbol{\phi}, \boldsymbol{\theta}_Q$ requires minimizing an appropriate loss function. We define a joint loss function that captures measurement inconsistency, isometric properties in sensing network, and image gradient loss. Consider the following optimization problem

$$\min_{\boldsymbol{z},\boldsymbol{\theta}_Q,\boldsymbol{\phi}} \mathcal{L} = \mathcal{L}_{\mathbf{G}} + \mathcal{L}_{\mathcal{A}_{\boldsymbol{\phi}}} + \mathcal{L}_{\mathbf{GDL}}, \quad \text{s.t.} \quad \|\boldsymbol{z}\|_0 \leq s, \quad (3)$$

where

$$\begin{aligned}
\mathcal{L}_{\mathbf{G}} &= \mathop{\mathbb{E}}_{\boldsymbol{z}}\{\|\boldsymbol{y} - \mathcal{A}_{\boldsymbol{\phi}}(\mathbf{G}_{\boldsymbol{\theta}_Q}(\boldsymbol{z}))\|_2^2\}, \\
\mathcal{L}_{\mathcal{A}_{\boldsymbol{\phi}}} &= \mathop{\mathbb{E}}_{\boldsymbol{x}_1,\boldsymbol{x}_2}\{\|\mathcal{A}_{\boldsymbol{\phi}}(\boldsymbol{x}_1) - \mathcal{A}_{\boldsymbol{\phi}}(\boldsymbol{x}_2)\|_2 + \delta - \gamma\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2\}, \\
\mathcal{L}_{\mathbf{GDL}} &= \sum_{i,j} \|\nabla_i \boldsymbol{x} - \nabla_i \mathbf{G}_{\boldsymbol{\theta}_Q}(\boldsymbol{z})\|_2^2 + \|\nabla_j \boldsymbol{x} - \nabla_j \mathbf{G}_{\boldsymbol{\theta}_Q}(\boldsymbol{z})\|_2^2,
\end{aligned}$$

with δ and γ being the parameters in set-restricted eigenvalue condition (S-REC) loss $\mathcal{L}_{\mathcal{A}_{\boldsymbol{\phi}}}$ [33]. The objective function is a sum of three terms: $\mathcal{L}_{\mathbf{G}}$ ensures measurement consistency; $\mathcal{L}_{\mathcal{A}_{\boldsymbol{\phi}}}$, the S-REC loss, restricts the isometry in the sensing operator $\mathcal{A}_{\boldsymbol{\phi}}$; and the gradient difference loss $\mathcal{L}_{\mathbf{GDL}}$ promotes reconstructions that are sharp [20]. We explain the importance of S-REC loss next. The signals $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are sampled from the true data distribution and generator network $\mathbf{G}_{\theta_Q}$, respectively. When $\boldsymbol{x}_1$ is different from $\boldsymbol{x}_2$, we prefer that the corresponding measurements are also different. $\mathcal{L}_{\mathcal{A}_{\boldsymbol{\phi}}}$ ensures that this property holds by penalizing the difference between $\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2$ and $\|\mathcal{A}_{\phi}(\boldsymbol{x}_1) - \mathcal{A}_{\phi}(\boldsymbol{x}_2)\|_2$. This loss is inspired by the restricted isometry condition from compressed sensing.

Algorithm 1: Training a quantized generative model (Q-GEN) and nonlinear sensing network.

```
Input: Training data $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$, sensing operator $\mathcal{A}_{\boldsymbol{\phi}}$,
learning rate $\alpha$, number of iterations $T$, and sparsity $s$
Initialization: Full-precision generator network
parameters $\boldsymbol{\theta}$
repeat
    $\boldsymbol{\theta}_Q \leftarrow$ Q($\boldsymbol{\theta}$) as per LinearQuantize method [11]
    for $i = 1$ to $N$ do
        Initialize $\boldsymbol{z}$
        for $t = 1$ to $T$ do
            Compute $\boldsymbol{z} \leftarrow \mathcal{P}_s(\boldsymbol{z} - \beta\nabla_{\boldsymbol{z}} f(\boldsymbol{y}, \boldsymbol{z}))$
        end for
    end for
    Compute $\mathcal{L}$ per Eq. (3)
    Update $\boldsymbol{\theta}$ per Eq. (6)
    Update $\boldsymbol{\phi} \leftarrow$ ADAM($\boldsymbol{\phi}$, $\nabla_{\boldsymbol{\phi}}\mathcal{L}_{\mathcal{A}_{\boldsymbol{\phi}}}$)
until iteration limit
```

The sparsity constraint in the latent representation $\boldsymbol{z}$ results in the range-space of the generator network $\mathbf{G}_{\boldsymbol{\theta}_Q}$ to be composed of a union-of-manifolds [15, 16]. The optimization of $\boldsymbol{z}$ involves minimizing $\mathcal{L}_G$ while satisfying the feasibility condition $\|\boldsymbol{z}\|_0 \leq s$. This approach is referred to as sparsity-driven latent-space sampling (SDLSS). An analysis of the distribution of the learned quantized weights from different quantized models is provided in Fig. 1. These weights are obtained after training the quantized generated models on MNIST and CIFAR-10 datasets.

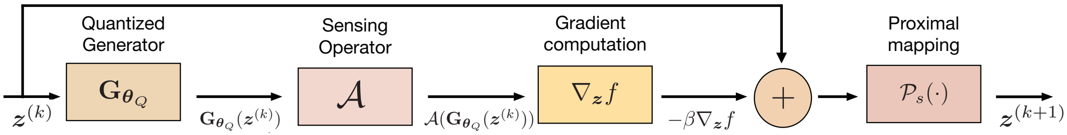
Quantized Generator
$\mathbf{G}_{\boldsymbol{\theta}_Q}$
$\boldsymbol{z}^{(k)}$
$\mathbf{G}_{\boldsymbol{\theta}_Q}(\boldsymbol{z}^{(k)})$
Sensing Operator
$\mathcal{A}$
$\mathcal{A}(\mathbf{G}_{\boldsymbol{\theta}_Q}(\boldsymbol{z}^{(k)}))$
Gradient computation
$\nabla_{\boldsymbol{z}} f$
$-\beta\nabla_{\boldsymbol{z}} f$
+
Proximal mapping
$\mathcal{P}_s(\cdot)$
$\boldsymbol{z}^{(k+1)}$

Figure 2. The flow diagram for updating the latent space variable $\boldsymbol{z}$ through proximal GD method.

3. Meta-Learning for Quantized Generative Image Prior

Inspired by [33], we use the meta-learning framework to jointly optimize the parameters $\boldsymbol{\phi}$ of the full-precision sensing network, $\boldsymbol{\theta}_Q$ of the coarsely quantized generative model, and the latent space variable $\boldsymbol{z}$. Meta-learning, or *learning to learn* is a training strategy used to improve the performance of models that involve multiple tasks [8].

The sparse code $\boldsymbol{z}$ is learned via proximal gradient-descent, whereas the sensing network parameters $\boldsymbol{\phi}$ are updated using the ADAM optimizer [17]. The quantized weights of the generator $\boldsymbol{\theta}_Q$ are updated using the straight-through-estimation method (STE) [3].

The optimization problem for obtaining the sparse code is given by

$$\min_{\boldsymbol{z}} f(\boldsymbol{y},\boldsymbol{z}) = \|\boldsymbol{y} - \mathcal{A}_{\boldsymbol{\phi}}(\mathbf{G}_{\boldsymbol{\theta}_Q}(\boldsymbol{z}))\|_2^2 \quad \text{s.t. } \|\boldsymbol{z}\|_0 \leq s. \tag{4}$$

The solution to the above problem can be obtained using the proximal gradient method, and the update equation is given by

$$\boldsymbol{z} \leftarrow \mathcal{P}_s\left(\boldsymbol{z} - \beta\nabla_{\boldsymbol{z}} f(\boldsymbol{y},\boldsymbol{z})\right), \tag{5}$$

where $\mathcal{P}_s(\cdot)$ is the hard-thresholding operator that sets all entries in the vector to zero apart from the s-largest entries in magnitude, and $\beta > 0$ is the step-size parameter.

The optimization of the quantized parameters of the generator network follows a standard STE procedure [3]. The full-precision weights are also stored and used for computing the quantized weights at the beginning of every forward pass. We use the LinearQuantize method [11] to obtain the low-precision weights from the full-precision weights. In the forward pass, a generator with coarsely quantized weights is used for reconstructing the image. The optimization cost contains the measurement loss $\mathcal{L}_{\mathbf{G}}$, $\mathcal{S}$-REC loss $\mathcal{L}_{\mathcal{A}_{\boldsymbol{\phi}}}$ and image gradient difference loss $\mathcal{L}_{\mathbf{GDL}}$. The gradients $\nabla_{\boldsymbol{\theta}}\mathcal{L}|_{\boldsymbol{\theta}=\boldsymbol{\theta}_Q}$ are computed at the quantized weights and are used for updating the full-precision weights $\boldsymbol{\theta}$. The weight updates are performed using ADAM [17] optimizer:

$$\boldsymbol{\theta} \leftarrow \text{ADAM}(\boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}}\mathcal{L}|_{\boldsymbol{\theta}=\boldsymbol{\theta}_Q}, \alpha, \beta_1, \beta_2), \tag{6}$$

where α is the learning rate and β_1, β_2 are the ADAM optimization variables (default values of 0.9, 0.999, respectively). The procedure for updating the sparse latent representation, training the quantized generative model and sensing network is given in Algorithm 1.

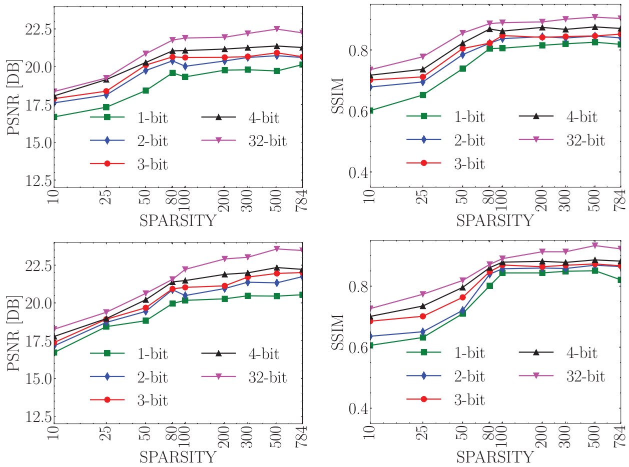

Figure 3. Performance trends of quantized versus full-precision models on MNIST dataset. The first row corresponds to $m = 25$, and the second row corresponds to $m = 100$

4. Experimental Results

We perform experimental validation on standard datasets MNIST [19], CIFAR10 [18], SVHN [22], and STL10 [5]. The generator network and sensing network in the case of MNIST are constructed using a two-layered feed-forward neural network with 500 neurons in each hidden layer and leaky ReLU as the activation. In contrast, a standard DCGAN [25] generator is used for CIFAR10, SVHN, and STL10 datasets, while a convolutional network with eight layers is used for sensing. Two objective metrics namely, peak signal-to-noise ratio (PSNR $= -10\log(\|\boldsymbol{x} - \mathbf{G}_{\boldsymbol{\theta}}(\boldsymbol{z})\|_2^2)$ dB), with peak value considered to be unity, and the structural similarity index metric (SSIM) [34] are used for comparing the performance of different models.

Figure 3 shows the variability in the performance of different models with sparsity in latent space for two different compression scenarios. The sparsity level in the latent space is denoted by s, i.e. $\|\boldsymbol{z}\|_0 \leq s$. The quantized models exhibit a close performance with 32-bit models, while the 4-bit quantized models having a superior performance over the 1-bit quantized models in both PSNR and SSIM metrics. One-bit quantization reduces the model size by 32×, while trading-off $\sim$ 2 dB PSNR and $\sim$ 0.07 SSIM. The two-bit network has a superior performance over the one-bit counterpart. The performance of the 3-bit and 4-bit models is comparable to that of the 32-bit model for $s < 100$.

Table 1. Performance comparison of quantized versus full-precision models. The 4-bit models perform close to 32-bit SDLSS models across the datasets. The 32-bit DCS model outperforms the 32-bit SDLSS and its quantized counterparts in STL10 and SVHN datasets. The 32-bit SDLSS model has a superior performance over 32-bit DCS model on CIFAR10 dataset. The 4-bit SDLSS model performs on-par with the 32-bit DCS model on the CIFAR10 dataset.

	Proposed methods				Literature	
	1-bit SDLSS	2-bit SDLSS	3-bit SDLSS	4-bit SDLSS	32-bit SDLSS	32-bit DCS
Dataset	(PSNR, SSIM)	(PSNR, SSIM)	(PSNR, SSIM)	(PSNR, SSIM)	(PSNR, SSIM)	(PSNR, SSIM)
CIFAR10	22.07, 0.6787	22.29, 0.6961	22.51, 0.7036	22.57, 0.7060	22.84, 0.7259	22.60, 0.7094
SVHN	31.71, 0.9213	32.06, 0.9201	33.26, 0.9370	33.65, 0.9414	34.01, 0.9549	34.63, 0.9501
STL10	20.65, 0.5543	21.24, 0.5996	21.45, 0.6051	21.63, 0.6279	22.04, 0.6723	23.16, 0.7136

Illustrative reconstructions from MNIST and SVHN are shown in Fig 4 and Fig 5, respectively. The performance of the full precision models (32-bit DCS, 32-bit SDLSS) is superior to that of the quantized models. However, visually, the results generated by the coarsely quantized models are comparable to the images generated from full-precision models. In some cases the quantized models perform slightly superior to the full-precision models, for instance, the generated digit '0' highlighted using the white boundary has a better loop closure than that output by the full-precision model.

We tested the Q-GEN model on images coming from CIFAR10, SVHN, and STL10 datasets. We used DCGAN generator and quantized its convolutional filters to 1, 2, 3, and 4 bits. The reconstruction performance is given in Table 1. The difference between full-precision and the quantized models varies from as low as 0.27 dB in the case of 4-bit models to about 2.3 dB in the 1-bit case. The performance of the quantized models in reconstructing color images inspires greater confidence for deploying them in energy-constrained settings. The quantized networks offer lower model sizes, which is attractive for resource-constrained devices, with a marginal performance drop.

5. Conclusion

We addressed the problem of compressed image recovery with a neural network used as the nonlinear sensing operator. We used a generative model with quantized weights as an image prior and incorporated sparsity into the latent space. Experimental results show that 4-bit quantized generative models perform close to full-precision (32-bit) models while reducing the model footprint by 8×. The 1-bit model has a 32× smaller footprint with a mild degradation in performance (PSNR of 0.74 dB in CIFAR10, 1.39 dB in STL10, and 2.3 dB in SVHN datasets). The analysis of the quantized weights shows that the presence of a few large weights makes the 4-bit model superior to the 1-bit counterpart. With a rapidly growing concern for saving energy in the context of AI, low-precision models offer a viable energy saving alternative. Future work would include using quantized generative image priors for other imaging applications such as ultrasound imaging and lensless imaging.

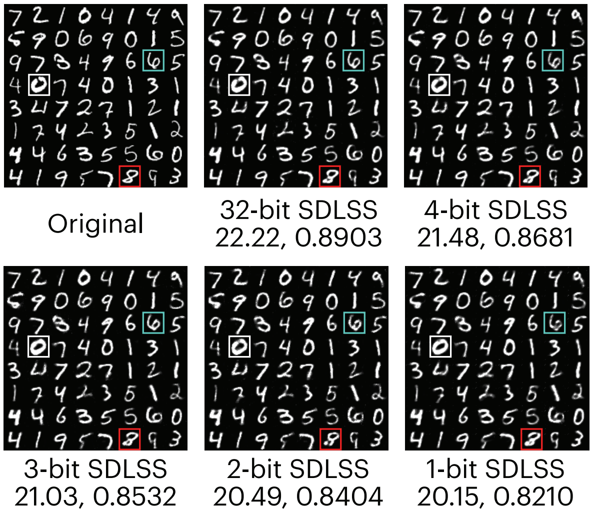

Figure 4. Images reconstructed using Q-GEN for various bit precision in comparison with the original from the MNIST dataset. The numbers at the bottom of the image are PSNR, SSIM.

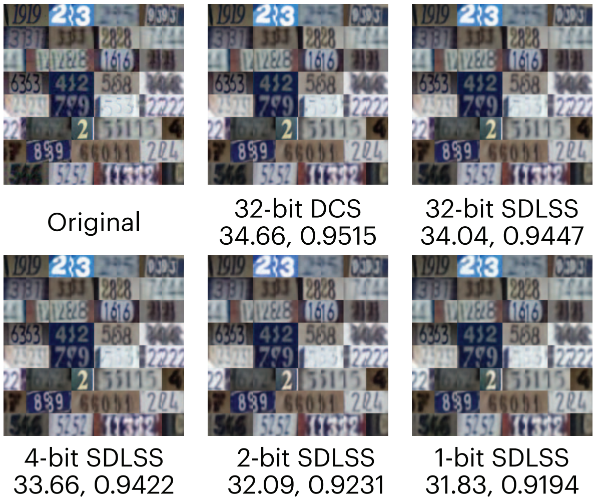

Figure 5. SVHN images reconstructed using Q-GEN models and 32-bit DCS model are compared here. We observe comparable visual performance from low-precision Q-GEN models compared to full-precision models.

References

[1] P. Andreev and A. Fritzler. Quantization of generative adversarial networks for efficient inference: A methodological study. pages 2179–2185, Los Alamitos, CA, USA, aug 2022. IEEE Computer Society. 1

[2] R. Anirudh, S. Lohit, and P. Turaga. Generative patch priors for practical compressive image recovery. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2535–2545, January 2021. 1

[3] Y. Bengio, N. Léonard, and A. C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013. 3

[4] A. Bora, A. Jalal, E. Price, and A. G. Dimakis. Compressed sensing using generative models. *Int. Conf. Mach. Lear.*, pages 537–546, 2017. 1

[5] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. volume 15 of *Proceedings of Machine Learning Research*, pages 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. 3

[6] M. Dhar, A. Grover, and S. Ermon. Modeling sparse deviations for compressed sensing using generative models. *Proceedings of the 35th International Conference on Machine Learning*, 80:1214–1223, 10–15 Jul 2018. 1

[7] H. W. Dong and Y. Yang. Training generative adversarial networks with binary neurons by end-to-end backpropagation. *CoRR*, abs/1810.04714, 2018. 1

[8] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *ICML*, 2017. 3

[9] L. Hou, Q. Yao, and J. T. Kwok. Loss-aware binarization of deep networks. *CoRR*, abs/1611.01600, 2016. 1

[10] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. Binarized neural networks. volume 29, pages 4107–4115. Curran Associates, Inc., 2016. 1

[11] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18(187):1–30, 2018. 1, 2, 3

[12] A. Jalal, L. Liu, A. G. Dimakis, and C. Caramanis. Robust compressed sensing using generative models. *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020. 1

[13] A. Kamath, E. Price, and S. Karmalkar. On the power of compressed sensing with generative models. *Proceedings of the 37th International Conference on Machine Learning*, 119:5101–5109, 13–18 Jul 2020. 1

[14] V. Killedar, P. K. Pokala, and C. S. Seelamantula. Learning generative prior with latent space sparsity constraints. *CoRR*, abs/2105.11956, 2021. 1

[15] V. Killedar, P. K. Pokala, and C. S. Seelamantula. Sparsity driven latent space sampling for generative prior based compressive sensing. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2895–2899, 2021. 1, 2

[16] V. Killedar and C. S. Seelamantula. Compressive phase retrieval based on sparse latent generative priors. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1596–1600, 2022. 1, 2

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *in Proceedings of the International Conference on Learning Representations*, 2015. 3

[18] A. Krizhevsky, V. Nair, and G. Hinton. Cifar-10 (canadian institute for advanced research). 3

[19] Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010. 3

[20] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. Jan. 2016. 4th International Conference on Learning Representations, ICLR 2016. 2

[21] X. Meng, R. Bachmann, and M. E. Khan. Training binary neural networks using the Bayesian learning rule. volume 119, pages 6852–6861. PMLR, 2020. 1

[22] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011. 3

[23] P. O'Connor and M. Welling. Sigma delta quantized networks. *CoRR*, abs/1611.02024, 2016. 1

[24] J. Ott, Z. Lin, Y. Zhang, S. Liu, and Y. Bengio. Recurrent neural networks with limited numerical precision. *CoRR*, abs/1608.06902, 2016. 1

[25] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*, 2016. 3

[26] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks, 2016. cite arxiv:1603.05279. 1

[27] Nareddy Kartheek Kumar Reddy, Mani Madhoolika Bulusu, Praveen Kumar Pokala, and Chandra Sekhar Seelamantula. Quantized proximal averaging networks for compressed image recovery. pages 4632–4642, June 2023. 1

[28] J. Song, A. Vahdat, M. Mardani, and J. Kautz. Pseudoinverse-guided diffusion models for inverse problems. 2023. 1

[29] D. Wan, F. Shen, L. Liu, F. Zhu, L. Huang, M. Yu, H. T. Shen, and L. Shao. Deep quantization generative networks. *Pattern Recognition*, 105:107338, 2020. 1

[30] H. Wang, S. Gui, H. Yang, J. Liu, and Z. Wang. Gan slimming: All-in-one gan compression by a unified optimization framework. *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV*, page 54–73, 2020. 1

[31] P. Wang, D. Wang, Y. Ji, X. Xie, H. Song, X. Liu, Y. Lyu, and Y. Xie. QGAN: quantized generative adversariatural imageal networks. *CoRR*, abs/1901.08263, 2019. 1

[32] H. Wen, S. Zhou, Z. Liang, Y. Zhang, D. Feng, X. Zhou, and C. Yao. Training bit fully convolutional network for fast semantic segmentation. *CoRR*, abs/1612.00212, 2016. 1

[33] Y. Wu, M. Rosca, and T. Lillicrap. Deep compressed sensing. *ICML*, 26:1349–1353, 2019. 1, 2, 3

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 3

[35] D. Zhang, J. Yang, D. Ye, and G. Hua. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. *CoRR*, abs/1807.10029, 2018. 1

[36] Zhizhen Zhao, Jong Chul Ye, and Yoram Bresler. Generative models for inverse imaging problems: From mathematical foundations to physics-driven applications. *IEEE Signal Processing Magazine*, 40(1):148–163, 2023. 1

[37] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016. 1