Deep Compressed Sensing with relationship to Generative Models

Yan Wu [1] **Mihaela Rosca** [1] **Timothy Lillicrap** [1]
DeepMind

Liu Wen

https://arxiv.org/pdf/1905.06723.pdf
https://arxiv.org/pdf/1703.03208.pdf
https://www.zhihu.com/question/21589619/answer/23212905

Compressed Sensing

- Sensing

- Compressed

Compressed Sensing (CS)

（1）什么是感知（**sensing**）？

Sensing描述的是，为了表达和恢复某一个信号或者对象，对其采取某种感知手段（这里的感知包括了对图像的imaging, 对连续信号的sampling,等等），从而得到这种感知模态下的measurement，以便之后的信号重建以及分析等应用，这样一个过程。

我们拿图像重建举一个例子：

如果我们要感知一个10x10像素的图像，最简单最直接的方式，就是直接在空间域里面感知，得到并储存图像的100个像素点,这样的话，你可以通过把这100个像素点排列为矩阵的方式，重建这副目标图像。这里获取这100个点的过程，就称之为sensing。

https://www.zhihu.com/search?type=content&q=%E5%8E%8B%E7%BC%A9%E6%84%9F%E7%9F%A5

Compressed Sensing (CS)

（2）什么是压缩（**compressed**）？

在(1)的例子中，我们感知了一个10x10图像全部的100个像素点，我们一般称这种sensing叫做**全采样（full sampling）**，因为目标信号是一个100维的信号，而你也获得了对应的100维measurement，你可以很直接地完美恢复图像。这样的感知方式没有任何的压缩，i.e., **感知到的维度=信号本身的维度**。当然你也可以采集更高维度的measurement，我们称之为over-sampling / over-sensing，但多余的采样并不会进一步提高你的图像恢复（信号已经达到完美恢复了）。

在不借助信号先验的前提下，我们把需要的最少的感知样本维度（这里是100），记做是**critical sampling rate**。那么你也可以选择感知少于critical rate的measurement，比如你只采集98个点，这种情况我们称之为是**under-sampling**，这样就起到了对full sensing的压缩，也就是compressed了。

https://www.zhihu.com/search?type=content&q=%E5%8E%8B%E7%BC%A9%E6%84%9F%E7%9F%A5

Compressed Sensing (CS)

- aims to **recover signal $\boldsymbol{x}$** from a linear measurement $\boldsymbol{m}$
- $m = Fx + \eta$
- F is a sampling matrix, and it is a "wide" matrix with $C \times D$, $C \ll D$, known.

ill-posed problem!

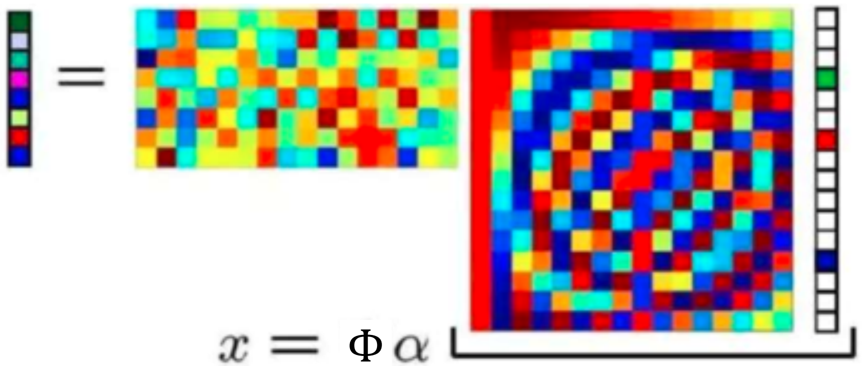
- η is the measurement noise which is usually assumed to be Gaussian distributed.
- $$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{m} - \mathbf{F}\boldsymbol{x}\|_2^2$$

Compressed Sensing (CS)

$$(1-\delta)\,\|\boldsymbol{x}_1-\boldsymbol{x}_2\|_2^2 \le \|\mathbf{F}(\boldsymbol{x}_1-\boldsymbol{x}_2)\|_2^2 \le (1+\delta)\,\|\boldsymbol{x}_1-\boldsymbol{x}_2\|_2^2$$

- Traditional CS [1] :
  - Restricted Isometry Property (RIP), such as Sparsity prior, ;
  - One can nearly perfectly recover x with high probability given a **random matrix** F and **sparse** x;
  - Sometime x is not sparse, e.g. the natural images,
    - x be sparse can be replaced by sparsity in a set of basis Φ, such as the Fourier basis or wavelet, so that $\Phi\alpha$ can be non-sparse signals such as natural images

$m =$ F Φ $\alpha = Fx$



[1] Donoho, D. L. Compressed sensing. IEEE Transactions on information theory, 52(4):1289–1306, 2006.

Compressed Sensing (CS)

- General CS :
  - Beyond sparsity prior! Statistic model, even **deep prior**!
  - Compressed Sensing using Generative Models (CSGM) [1];
  - Deep Compressed Sensing (DCS) [2].

[1] Bora, A., Jalal, A., Price, E., and Dimakis, A. G. Compressed sensing using generative models. arXiv preprint arXiv:1703.03208, 2017.

[2] Yan Wu, Mihaela Rosca, Timothy Lillicrap *Deep Compressed Sensing*. ICML 2019.

Compressed Sensing using Generative Models (**CSGM**)

- $m = Fx + \eta$
  - $F \in R^{C \times D}$
  - $x^* = \arg\min ||m - Fx||_2^2 + ||x||_1$

Sparsity Prior

- $m = FG(z) + \eta$
  - a distribution P_z over $z \in R^k$;
  - $\boldsymbol{x} = \boldsymbol{G}(\boldsymbol{z})$ **is a generative model, like GAN, VAE**, $G: R^k \rightarrow R^D, k \ll D$;
  - $z^* = \arg\min ||m - FG(z)||_2^2$

$\boldsymbol{G}(\boldsymbol{z})$: **learnable prior !**

  - Recover $x^* = G(z^*)$

CSGM: experiments-Super Resolution

- $m = FG(z) + \eta$
  - m is the low-resolution image;
  - $G(\mathrm{z})$ is a pretrained generative model (DCGAN) in ImageNet dataset;
  - We need to recover the high-resolution image x^*;
  - $z^* = \arg\min ||m - FG(z)||_2^2$, gradient decent over z, and we get the optimal z^*, then arrive the high-resolution image $x^* = G(z^*)$.

CSGM: experiments-Super Resolution

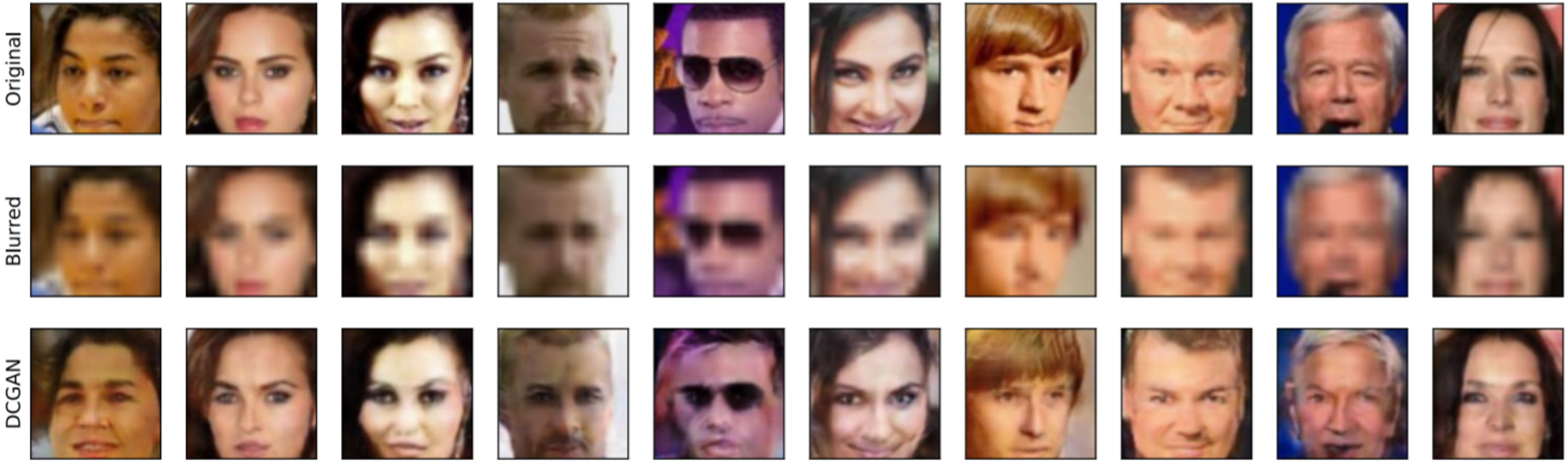


Figure 4: Super-resolution results on celebA. Top row has the original images. Second row shows the low resolution ($4x$ smaller) version of the original image. Last row shows the images produced by our algorithm.

CSGM: experiments-Reconstruction

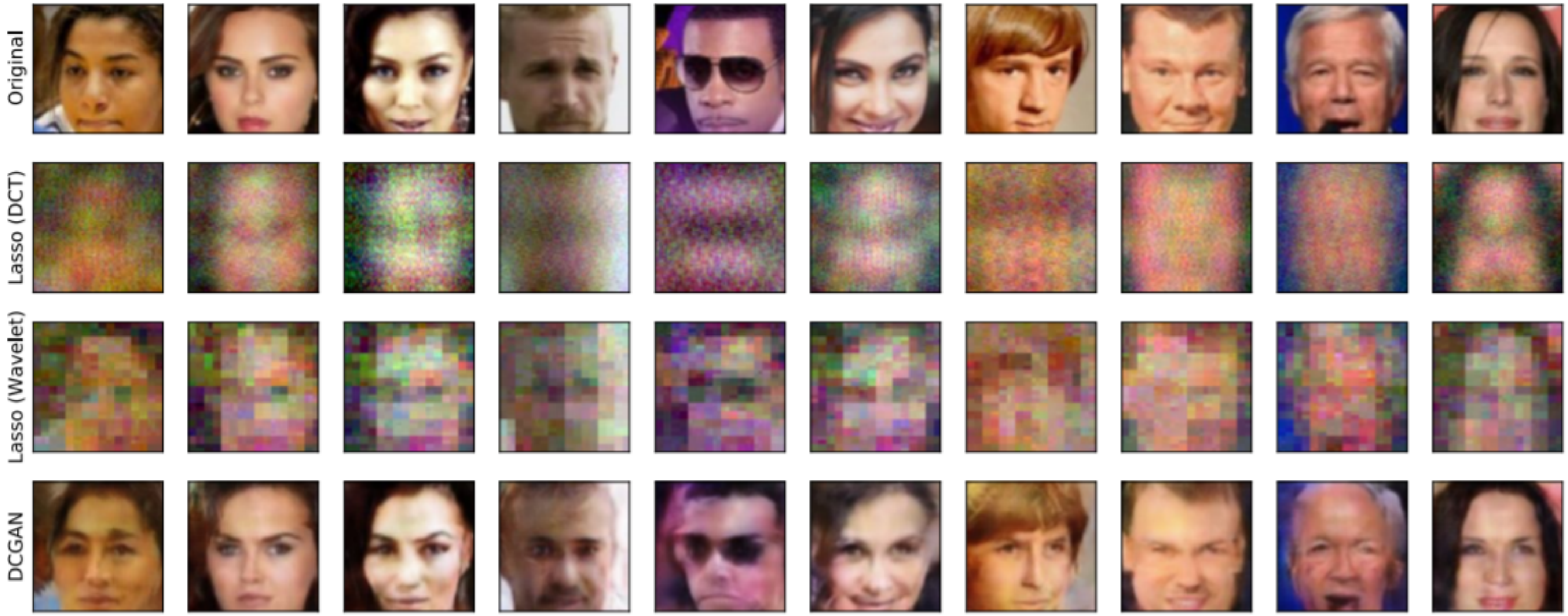


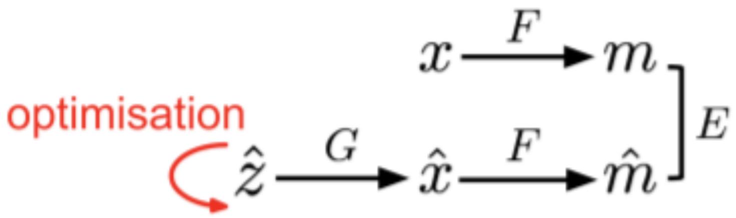
Figure 3: Reconstruction results on celebA with $m = 500$ measurements (of $n = 12288$ dimensional vector). We show original images (top row), and reconstructions by Lasso with DCT basis (second row), Lasso with wavelet basis (third row), and our algorithm (last row).

Deep Compressed Sensing (DCS)

- $z^* = \arg\min ||m - FG_\theta(z)||_2^2$, and recover $x^* = G(z^*)$;
- Can we update G_θ instead of using a pretrained GAN and fixing it?
  - a general case of Dictionary Learning!
- Can we learn measurement function, instead of only a linear combination of F_ϕ?
  - Similar to the discriminator in GAN;
  - Learned image similarity, VGG Loss;
  - Learnable reward function in Reinforcement Learning!

DCS: training G_θ

- $m = Fx$;
- $z^* = \arg\min ||m - FG_\theta(z)||_2^2$;
- $E_\theta(m, z) = ||m - FG_\theta(z)||_2^2$;



$$\mathcal{L}_F = \mathbb{E}_{\boldsymbol{x}_1, \boldsymbol{x}_2}\left[\left(\|\mathbf{F}\left(\boldsymbol{x}_1 - \boldsymbol{x}_2\right)\|_2 - \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2\right)^2\right]$$

Generalized RIP

Algorithm 1 Compressed Sensing with Meta Learning

Input: minibatchs of data $\{\boldsymbol{x}_i\}_{i=1}^N$, random matrix $\mathbf{F}$, generator G_θ, learning rate α, number of latent optimisation steps T

repeat

  Initialize generator parameters θ

  for $i = 1$ **to** N **do**

    Measure the signal $\boldsymbol{m}_i \leftarrow \mathbf{F}\,\boldsymbol{x}_i$

    Sample $\hat{\boldsymbol{z}}_i \sim p_{\boldsymbol{z}}(\boldsymbol{z})$

    for $t = 1$ **to** T **do**

      Optimise $\hat{\boldsymbol{z}}_i \leftarrow \hat{\boldsymbol{z}}_i - \frac{\partial}{\partial \boldsymbol{z}} E_\theta(\boldsymbol{m}_i, \hat{\boldsymbol{z}}_i)$

    end for

  end for

  $\mathcal{L}_G = \frac{1}{N}\sum_{i=1}^N E_\theta(\boldsymbol{m}_i, \hat{\boldsymbol{z}}_i)$

  Compute $\mathcal{L}_F$ using eq. 12

  Update $\theta \leftarrow \theta - \frac{\partial}{\partial \theta}(\mathcal{L}_G + \mathcal{L}_F)$

until reaches the maximum training steps

DCS: learn measurement function

- $m = F_{\phi}(x)$;
- $z^* = \arg\min ||m - F_{\phi}(G_{\theta}(z))||_2^2$;
- $E_{\phi,\theta}(m, z) = ||m - F_{\phi}(G_{\theta}(z))||_2^2$;

$$\mathcal{L}_F = \mathbb{E}_{\boldsymbol{x}_1, \boldsymbol{x}_2}\left[\left(\|F_\phi(\boldsymbol{x}_1 - \boldsymbol{x}_2)\|_2 - \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2\right)^2\right]$$

Algorithm 2 Deep Compressed Sensing

Input: minibatchs of data $\{\boldsymbol{x}_i\}_{i=1}^N$, measurement function F_ϕ, generator G_θ, learning rate α, number of latent optimisation steps T

repeat

  Initialize generator parameters θ

  for $i = 1$ **to** N **do**

    Measure the signal $\boldsymbol{m}_i \leftarrow F_\phi(\boldsymbol{x}_i)$

    Sample $\hat{\boldsymbol{z}}_i \sim p_{\boldsymbol{z}}(\boldsymbol{z})$

    for $t = 1$ **to** T **do**

      Optimise $\hat{\boldsymbol{z}}_i \leftarrow \hat{\boldsymbol{z}}_i - \frac{\partial}{\partial \boldsymbol{z}} E_\theta(\boldsymbol{m}_i, \hat{\boldsymbol{z}}_i)$

    end for

  end for

  $\mathcal{L}_G = \frac{1}{N}\sum_{i=1}^N E_\theta(\boldsymbol{m}_i, \hat{\boldsymbol{z}}_i)$

  Compute $\mathcal{L}_F$ using eq. 12

  Option 1 : joint update $\theta \leftarrow \theta - \frac{\partial}{\partial \theta}(\mathcal{L}_G + \mathcal{L}_F)$

  Option 2 : alternating update

$$\boxed{\theta \leftarrow \theta - \frac{\partial}{\partial \theta}\mathcal{L}_G \qquad \phi \leftarrow \phi - \frac{\partial}{\partial \phi}\mathcal{L}_F}$$

until reaches the maximum training steps

Relationship to GAN

- $m = F_{\phi}(x)$, m is one-dimension measurement, real is 1, fake is 0.
- $z^* = \arg\min ||m - F_{\phi} G_{\theta}(z)||_2^2$, $E_{\phi,\theta}(m, z) = ||m - F_{\phi} G_{\theta}(z)||_2^2$;

$$\mathcal{L}_F = \begin{cases} \|F_\phi(\boldsymbol{x}) - 1\|_2^2 & \boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x}) \\ \|F_\phi(\hat{\boldsymbol{x}})\|_2^2 & \hat{\boldsymbol{x}} \sim G_\theta(\hat{\boldsymbol{z}}), \forall \hat{\boldsymbol{z}} \end{cases} \quad (15)$$

LS-GAN!

- If E is binary cross-entropy criterion,

$$\mathcal{L}_F = t(\boldsymbol{x}) \ln [D_\phi(\boldsymbol{x})] + (1 - t(\boldsymbol{x})) \ln [1 - D_\phi(\boldsymbol{x})] \quad (16)$$

where the binary scalar t is an indicator function identifies whether $\boldsymbol{x}$ is a real data point.

$$t(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x}) \\ 0 & \boldsymbol{x} \sim G_\theta(\boldsymbol{z}), \forall \boldsymbol{z} \end{cases} \quad (17)$$

Vanilla GAN!

Experiments

Figure 2. Reconstructions using 10 measurements from random linear projection (top), trained linear projection (middle), and trained neural network (bottom).