Deep Decoder: Evaluation, Improvements And Benchmarks

Tyler Sellmayer, Salman Naqvi

Introduction

We implement a simple untrained neural network based image model as recently described by [1] that can represent and generate natural images well. Image models allow for representing images with fewer parameters, which enable them to solve a variety of inverse problems in imaging like denoising, inpainting and super-resolution.

Deep Decoder, [1], is a deterministic function G that maps a set of parameters C to an image, x:

x = G(C),

where the number of weight parameters is much smaller than the number of free parameters in image x. This highly under-parameterized network is effectively a method for image dimensionality reduction, which compresses the image into a concise set of network weights. This under-parameterization provides a barrier to overfitting to noise in images, giving it state-of-the-art denoising performance.

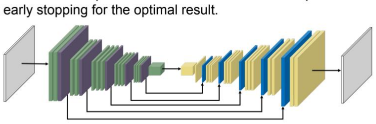
In this project, we conducted a number of experiments with Deep Decoder to establish the best hyperparameters. We further evaluated its performance on a recently published real-world noise image denoising benchmark dataset. Finally, we investigated the use of different cost functions which are recently shown to be optimal for image restoration tasks.

Related Work

Deep neural networks have been outperforming traditional methods for image restoration. However, they typically:

a. Have deep Convolutional architectures
b. Are over-parameterized and
c. Need pre-training on Large Datasets

Recent "Deep Image Prior" [4] made breakthrough by showing state-of-the-art performance without the need of pre-training. **However** it also had deep convolutional over-parameterized architecture with skip connections,and relied on early stopping for the optimal result.



Evaluation Metric

We investigated several commonly used image quality evaluation metric to establish the accuracy of our network. These metric determine how much does the network output (restored image) differ from the target clean image. These included:

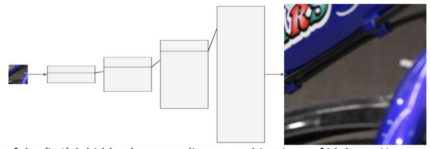
1. PSNR (Peak Signal-To-Noise Ratio), derived from MSE (mean-squared error) or L2 loss. Maximized by minimizing the MSE loss.
2. L1 loss.
3. SSIM (**S**tructural **SIM**iliarity) between the goal and denoised images. Computed as the average over sliding windows on each image.. (x and y are windows on the goal and denoised images resp., u and σ are mean and covariance, & c_1 and c_2 are constants to account for the image type)
4. MS-SSIM (Multi-scale SSIM) similar to SSIM.

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \qquad \mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Model Architecture

Our deep decoder, in contrast with other neural network models:

a. Is under-parameterized (doesn't overfit to noise),
b. has simple non-convolutional architecture (easy to study),
c. does not require pre-training of weight parameters



Outputs of the (i+1)th hidden layer are linear combinations of ith layer, Upsampled and passed through a non-linear activation.

$$\mathbf{B}_{i+1} = \mathrm{cn}(\mathrm{relu}(\mathbf{U}_i\mathbf{B}_i\mathbf{C}_i))$$

Input: to the network, also found to be optimal after hyperparameters search, is a small uniformly sampled random noise, with regularization noise added b/w each iteration.

Update: The network uses Adam Optimization and updates its parameters at each iteration using an L2 loss between its output and the target noisy image.

Hyperparameter search

Our randomized hyperparameter search determined the optimal denoising network.

1. It searched along 5 axes, trying 100 potential combinations for 1000 iterations each.
2. We then fixed the kernel size at 1x1 and ran 1000 training iterations per hyperparameter configuration.
3. We then kept the top 3 in terms of PSNR score.

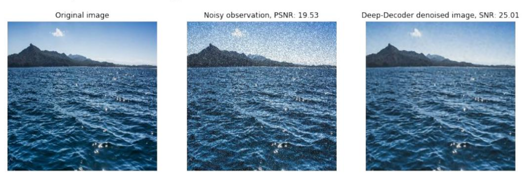
❖ Our "winning" denoising model uses 4 layers with 1x1x196 convolutional channels, with a 2x linear upsampling between each layer.

Hyperparameter	Value in best-scoring denoiser
Number of channels in convolutions	196
Number of convolutional layers	4
Input regularization noise	1.7502e-7
Activation function of conv layers	LeakyReLU(negative_slope=0.1)
Random fixed input or downsampled image?	random

Fine Structure Preservation

In experiments with artificial gaussian white noise, deep decoder preserves fine structure even in images which are already noise-like, like the ocean in the image..

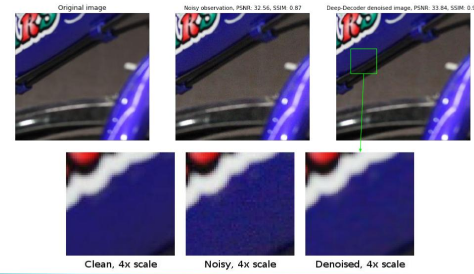
This experiment used sigmoid as the activation function and MSE as the loss.



Denoising Benchmark Dataset

- The original paper only showed the results on synthetic gaussian noise
- We thus investigated network's performance on real-world noisy images.
- The benchmark database contains low-ISO and high-ISO (grainy) images.
- After 10k iterations of updating parameters, the network generates the output
  - With measurable noise reduction and
  - Subjective visual proximity to the benchmark "clean" images

  using the best hyperparameter configuration found in the prior search.



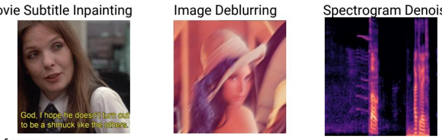
Loss Functions

We tried l1, l2, SSIM, and MS-SSIM loss functions as well as "Mix" (linear comb. of MS-SSIM+L1) loss, shown to be optimal for image restoration tasks [3].

❖ Mix (right most) provided greater contrast in higher frequency regions but MSE alone (middle) gave the best performance on our evaluation metrics.

Future Work

Movie Subtitle Inpainting | Image Deblurring | Spectrogram Denoising



References

1. R. Heckel, P. Hand. "Deep Decoder: Concise Image Representations from Untrained Non-convolutional Networks". In: arXiv preprint arXiv:1810.03982v1, 2018
2. Jun Xu, Hui Li, Zhetong Liang, David Zhang, and Lei Zhang. "Real-world Noisy Image Denoising: A New Benchmark". https://arxiv.org/abs/1804.02603, 2018.
3. Hang Zhao, Orazio Gallo, Iuri Frosio and Jan Kautz. "Loss Functions for Neural Networks for Image Processing". arXiv:1511.08861. In: IEEE Transactions on Computational Imaging, 2017
4. D. Ulyanov, A. Vedaldi, and V. Lempitsky. "Deep image prior". In: Conference on Computer Vision and Pattern Recognition, 2018

Video https://youtu.be/0c6ettmdlm0