Image denoising with multi-layer perceptrons

HAX907X - Apprentissage statistique

Esteve Nathan, Fattouhy Mohamed, Nguyen Louis, Vernay Amélie

October 14, 2021

1 Introduction

The article we worked on [**denoise**] aims to learn the mapping from a noisy image, that is the image pixels is undergo random fluctuations, to a noise-free image directly with plain multi-layer perceptrons (MLP), applied to smaller areas, called patches. The denoised image is obtained by placing the denoised patches at the location of their noisy counterparts.

Images are invariably corrupted by some degree of noise, which strength and type depends on the imaging process. Image denoising seeks to find a clean image given only its noisy version.

Its complexity requires to split the image into possibly overlapping patches, denoised separately.

However, the size of the patches affect the quality of the denoising function : large patches potentially lead to better result, but the function might be difficult to model.

Among the numerous existing types of noise, we will mainly focus on additive white and Gaussian-distributed noise with known variance (AWG noise), but the method can also adapted to mixed Poisson-Gaussian noise, JPEG artifacts, salt and pepper noise and noise that resembles stripes.

The signal-to-noise ratio (SNR) measures the quality of a signal, more precisely it measures the share of information (interpretable signal) in relation to noise (stray signal). The SNR is obtained by the ratio of the value the pixels should have if there was no noise, and the standard deviation of the noise.

The SNR can be expressed in decibel (dB) by the following transformation : $SNR\ (dB) = 20 \times log(SNR)$.

2 Multi-layer perceptron

A multi-layer perceptron is a particular architecture of neural network. In this architecture we have input layer, output layer and many hidden layers, each neuron of a hidden layer is connected to every neuron of the previous and the next ones. In addition, weights are used to weight the signals between neurons, those weights are randomly initialized, and updated by the backpropagation algorithm minimizing a loss function. With this kind of neural network, the weight matrix might be dense.

The MLP are very expensive in calculation time during their learning phase, indeed the calibration of such a network requires a lot of algebraic calculation, more precisely matrix-vector products, and that's the computationally most intensive operations. So for their experiments they used Graphics Processing Units (GPUs) rather than Central Processing Units (CPUs), because of their ability to efficiently parallelize operations.

3 MLP for image denoising

To find an desoising function they use a MLP, they use pairs of noisy image patch(input) and clean image patche (output). To make it efficient, they normalize the data and initialize the weights, which are sampled from an uniform distribution. Those two steps ensure that all parts of the sigmoid function are reached.

$$w \sim [-\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}]$$

Where n_j are the number of neurons in the input side and output side of the layer. To optimised those weights, they use the stochastic gardient descent, apply to our loss function.

The loss function is defined as the mean squared error between $f(x)$ (denoised patch) and y (clean patch). With this choise of loss function, we maximise the PSNR values.

Furthermore, to keep a steady learning rate while modifying the number N of hidden units per layer, they divide it by N in each layer. The basic learning rate was set to 0.1.

The number of hidden layers, as well as N, determine the capacity of the model. In pratice, it's often better to use a large number of hidden layers with fewer hidden units each.

After 3.5×10^8 backpropagations we obtain the denoising fonction that will be use to compare with the other method.

4 Experimental design

All experiments are performed on grey-image images, but the MLPs could also be trained on color images. They used images from six different datasets, and performed no pre-processing but the transform to grey-scale on the training images.

To evaluate their approche, they mainly focused on a standard test dataset, *standard test images*, and AWG noise with $\sigma = 25$. However, they show results for others noise levels, other types of noise and other image sets, to compare the performance of different methods.

5 Results and comparison with existing algorithms

5.1 Existing algorithms

Image denoising is a well-known problem, thus denoising methods are numerous and diverse. In order to evaluate the efficiency of the method, they compared the results against the following algorithms :

- **BM3D** (2007), this method does not explicitly use an image prior, it use the fact that images contain self-similarities.

- **NLSC** (2010), a dictionary-based algorithm witch exploit self-similarities in image like **BM3D**

Both these methods are considered the state-of-the-art in image denoising.

- **EPLL** (2011) is a learning-based approach, shown to be sometimes superior to BM3D

- **KSVD** (2006) is a dictionary method based on sparse linear combination of dictionary elements.
They chose these algorithms for their comparison because they achieve excellent results, with different approaches.

5.2 Comparaison on AWG noise

Let's present the results achieved with an MLP on AWG noise with $\sigma = 25$.

The MLP $(39 \times 2, 3072, 3072, 2559, 2047, 17 \times 2)$ delivered the best results.

Out of the 11 *standard test images*, the MLP approach achieves the best result on 7 images and is the runner-up on one image. Their method is clearly inferior to BM3D and NLSC on both of the images which contain a lot of regular structure. However, it outperforms KSVD on these images, even though KSVD is also an algorithm well-suited for images with regular structure. Furthermore, they also outperform both KSVD and EPLL on every image of the dataset.

They now compare the MLP method to EPLL, BM3D and NLSC on the five larger test sets : *Berkeley BSDS*500, *Pascal VOC* 2007, *Pascal VOC* 2011, *McGill*, and *ImageNet*, with a total of 2500 test images.

Their method outperforms EPLL on 99.5% of the 2500 images, and BM3D on 92% of it. It also outperforms NLSC on 80% of the test sets ; the initial dictionary of NLSC was trained on a subset of *Pascal VOC* 2007, which explains its good results.

5.3 Comparison on different noise variances

They now present the results obtained by their approach on four other noise levels : $\sigma = 10$ (low noise), $\sigma = 50$ (high noise), $\sigma = 75$ (very high noise) and $\sigma = 170$ (extremely high noise).
Le MLP semble etre plus efficace quand σ est elevé. En effet on observe sur la comparaison de plus de 2500 images, avec $\sigma = 75$ il surpasse les autres methodes sur plus de 97.60% des images du data set. Pour $\sigma = 50$ c'est 95.76% et quand $\sigma = 10$ nous obtenons de meilleur résultat sur 75% des images.