Image denoising with multi-layer perceptrons

HAX907X - Apprentissage statistique

Esteve Nathan, Fattouhy Mohamed, Nguyen Louis, Vernay Amélie

October 12, 2021

1 Introduction

The article we worked on [1] aims to learn the mapping from a noisy image 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.

2 Multi-layer perceptron

A multi-layer perceptron is a particular architecture of neural network. In this architecture we have many hidden layers, and 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. So with this kind of neural network, the weight matrix might be dense.

The computationally most intensive operations in an MLP are the matrix-vector multiplications. 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 use an MLP for image denoising, they estimate the parameters by training on pairs of noisy and clean image patches using stochastic gradient descent.

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.

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

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 the imagenet dataset and they performed no pre-processing, but the transform to grey-scale on the training images, which belong to six different datasets.

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 performances of methods.

5 Results and comparison with existing algorithms

5.1 Existing algorithms

There are many algorithms to denoise an image. In order to evaluate the efficiency of their method, they compared the results obtained against four algorithms, which we will briefly describe.

The algorithms are the following :
They choose those algorithms because they achieve good results.

5.2 Comparaison on AWG noise

In this section, we present results achieved with an MLP on AWG noise with $\sigma = 25$.

The MLP had the architecture $(39 \times 2, 3072, 3072, 2559, 2047, 17 \times 2)$, and they chose that because it delivered the best results. Furthermore, the MLP was trained for approximately 3.5×10^8 backprops.

On the 11 standard test images, their approach achieves the best result on 7 of the 11 test images and is the runner-up on one image. However, their method is clearly inferior to BM3D and NLSC on images which contain a lot of regular structure but outperform KSVD on those images, even though KSVD is also an algorithm that is well-suited for these types of images. In addition, they also outperform both KSVD and EPLL on every image.

Now we compare their method to EPLL, BM3D and NLSC on the five larger test sets : *Berkeley BSDS*500, *Pascal VOC* 2007, *Pascal VOC* 2011, *McGill*, *ImageNet*.
Each dataset contains 500 images, giving us a total of 2500 test images.

5.3 Comparison on different noise variances

Now we present the results obtained by their approach on other noise levels. They tested four noise levels : $\sigma = 10$ (low noise), $\sigma = 50$ (high noise), $\sigma = 75$ (very high noise) and $\sigma = 170$ (extremely high noise).

References

[1] Stefan Harmeling Harold Christopher Burger Christian J. Schuler. “Image denoising with multi-layer perceptrons, part 1: comparison with existing algorithms and with bounds”. In: *Journal of Machine Learning Research* (2012). DOI: https://arxiv.org/abs/1211.1544.