Image denoising with multi-layer perceptrons

HAX907X - Apprentissage statistique

Esteve Nathan, Fattouhy Mohamed, Nguyen Louis, Vernay Amélie

October 17, 2021

1 Introduction

The article we worked on [**denoise**] aims to learn the mapping from a noisy image (the image pixels undergo random fluctuations), to a noise-free image directly with plain multi-layer perceptrons (MLP).

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 smaller areas, called patches, denoised separately and then placed at the location of their noisy counterparts. 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, the authors mainly focused on additive white and Gaussian-distributed noise with known variance (AWG noise), but they will show that the method can also be adapted to mixed Poisson-Gaussian noise, JPEG artifacts, salt and pepper noise and noise that resembles stripes.

2 Multi-layer perceptron for image denoising

A multi-layer perceptron is a particular architecture of neural network. In this architecture we have an input layer, an output layer and many hidden layers, each neuron of a hidden layer being connected to every neuron of the previous and the next ones. Signals between neurons are weighted by randomly initialized weights, updated by backpropagation to compute the gradient of a loss function.

To find a denoising function with MLP, the authors use pairs of noisy image patch (input) and clean image patch (output). To make it efficient, the data is normalized and the weights w sampled from an uniform distribution :

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

where n_j and n_{j+1} are the number of neurons in the input and output sides of the layer, respectively.

To update those weights, they use the stochastic gardient descent, applied to our loss function, defined as the mean squared error between $f(x)$ (denoised patch) and y (clean patch), minimizing pixel-wise. With this choise of loss function, we maximise the Peak Signal-To-Noise Ratio : $\text{PSNR} = 20 \times \log_{10}(m/\sqrt{\text{MSE}})$ (dB), where m is the maximum possible pixel value of a given image. It is the ratio between the maximum possible power of a signal (interpretable signal) and the power of corrupting noise (stray signal)[1].

Furthermore, to keep a steady learning rate while modifying the number N of units per hidden 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.

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.

For each type of noise and each noise level, an MLP is trained. This allows us to obtain a denoising function for each configuration. Once the MLP's training is done, the authors compare the results obtained with other denoising methods.

[1] https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio

3 Results and comparison with existing algorithms

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** does not explicitly use an image prior, but rather the fact that images often contain self-similarities. It's considered the state-of-the-art in image denoising ;
- **NLSC** is a dictionary-based algorithm that exploits images self-similarities like BM3D, also achieving excellent results ;
- **EPLL** is a learning-based approach, based on patch-based priors. It sometimes outperforms BM3D ;
- **KSVD** is a dictionary-based method, that approximates a noisy patch by a sparse linear combination of dictionary elements. It achieves better results than previous state-of-the-art algorithms

The authors chose these algorithms for their comparison because they manage excellent results, with different approaches.

Comparaison on AWG noise - Let us present the results obtained with MLP-based algorithm on AWG noise with $\sigma = 25$.

The MLP $(39 \times 2, 3072, 3072, 2559, 2047, 17 \times 2)$ was trained for approximately 3.5×10^8 backpropagations and delivered the best results.

Out of the 11 *standard test images*, the multi-layer perceptron approach achieves the best result on 7 images. The method with MLP 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, MLP-based denoising algorithm also outperforms both KSVD and EPLL on every image of the dataset.

They now compare the MLP-based method to EPLL, BM3D and NLSC on the five larger test sets : *Berkeley BSDS500*, *Pascal VOC 2007*, *Pascal VOC 2011*, *McGill*, and *ImageNet*, with a total of 2500 test images.

MLP-based 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.

Comparison on different noise variances - The results obtained by the approach with MLP on low noise ($\sigma = 10$), high noise ($\sigma = 50$), very high noise ($\sigma = 75$) and extremely high noise ($\sigma = 170$) are presented below :

MLP-based algorithm outperforms BM3D on 75.04% of the 2500 test images for $\sigma = 10$, on 95.76% of the test images fo $\sigma = 50$, and it outperforms BM3D on 97.60% of the image set. For very high noise, MLP-based method outperforms all other methods. The most-performing methods for $\sigma = 170$ are prior-based methods.

To know how MLP behaves on the noise levels they have been trained on, they varied σ between 5 and 100 in steps of 5. Results shows that MLPs achieve better results than BM3D on the noise levels they have been trained on.

However, for noise levels they have not been trained on, the MLPs performance degrades quickly.
Exceptions are the MLPs trained on $\sigma = 50$ and $\sigma = 75$, which also outperform BM3D on $\sigma = 45$ and $\sigma = 55$ (MLP was trained on $\sigma = 50$) and $\sigma = 70$ and $\sigma = 80$ (MLP trained on $\sigma = 75$).

To conclude, the MLP-based algorithm is particularly well-suited for medium to very high noise levels ($\sigma \geq 50$), and still outperforms BM3D with $\sigma = 10$. However, MLPs have to be trained on each noise level in order to achieve good results.

4 Theoritical bound

Clustering-based bounds - Recent denoising algorithms, especially BM3D, are very close to an inherent limit on denoising quality (estimated in [**deaddenoise**]) for images with rich geometric structure. Yet, MLP outperforms BM3D by approximatively 0.4 dB on this type of images, which is a significant improvement.

Bayesian bounds - Levin and Nadler [**Levin2011NaturalID**] estimated theoretical bounds, in a Bayesian framework, on how well any denoising algorithm can perform, which depends on the patch size. The results with MLP-based approach exceed these bounds, by using larger patches than assumed by Levin and Nadler.
Similar bounds for infinite patch sizes were estimated by Levin. MLP-based algorithm makes significant progress in reaching these bounds: their approach achieves almost half the theoretically possible gain over BM3D.

5 Comparison on other types of noise

Most denoising algorithms assume the noise to be AWG. It is actually not always the case, and for such algorithms, a transformation needs to be applied to obtain AWG noise. However, in most cases, such transformation is impossible to find.
The method with MLP enables to learn a denoising algorithm for a given noise type, if it can be simulated. The authors use the architecture that achieved good results for AWG noise.
For stripe noise, MLP-based approach outperforms BM3D on 82 million training examples.
For salt and pepper noise, the method with MLP outperforms both BM3D and median filtering, the latter being a common algorithm for denoising salt and pepper noise.
With regard to JPEG quantization artifacts

6 Block-matching MLPs

Recent denoising algorithms such as BM3D and NLSC rely on block-matching procedures. The idea is to find the patches most similar to a reference patch.
Knowing that block-matching procedures are effective, let's see if a combination with MLP-based algorithm can achieve better results, especially images with repeating structures, on which BM3D and NLSC outperform the approach with plain MLP.
The authors train MLPs that take as input a reference patch and its nearest neighbors (similar patches).
The block-matching procedure slows down the training by approximatively 2. The MLP they used has four hidden layers (4095,2047,2047,2047). Compared to BM3D, this method always select the same number of neighbors, directly in the noisy image, and has fewer hyper-parameters.

Results - The mean result achieved with block-matching MLP on the 11 *standard test images* is $0,07$dB higher than the approach with plain MLP. It outperforms the plain MLP on 7 images, especially on images with repeating structure. However, BM3D and NLSC still provide better results on this kind of images.
On larger test sets, both methods achieve quite the same results : method with plain MLPs performs better on images with smooth surfaces, while method with block-matching MLPs provides better results on images with repeating structure, whithout outperforming BM3D and NLSC.
However, it's noticeable that block-matching MLPs use less information as input than plain MLPs.

7 Conclusion

Through this paper we can improve the MLP. Whether in the improvement to reach the theoretical limits, or the debriefing. Indeed, when the MLP is trained on a specific type of noise, it provides very good results on the same noise. However on some image paternity, or with low variances of other method (BM3D) seems to compete with the MLP.