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, which means that the image pixels 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.

2 Multi-layer perceptron for image denoising

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 being connected to every neuron of the previous and the next ones. Signals between neurons are weighted by randomly initialized weights, updated by the backpropagation algorithm minimizing a loss function. With MLP the weight matrix might be dense.

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 PSNR . Denoising quality is measured with the Peak Signal-To-Noise Ratio (PSNR) : 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 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.

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.

---

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

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 is correctly trained, the next step is to compare the results obtained with other denoising method.
Once the MLP is trained, the next step is to compare the results obtained with other denoising methods.

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** (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, using a maximisation of likelihood of several patch.
- **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.

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

The MLP $(39 \times 2, 3072, 3072, 2559, 2047, 17 \times 2)$ was trained for approximately 3.5×10^8 backprops and 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 BSDS500*, *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.

Comparison on different noise variances -   Now we 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).

On 2500 test images, the resulst show that MLP is not very efficient for low level noise ($\sigma = 10$) compared to BM3D. Indeed, MLP outperform BM3D on 1876 of the 2500 images (75.04%). For ($\sigma = 50$), MLP outperform BM3D on 2394 (95.76%) of the 2500 images, and for ($\sigma = 75$) on 2440 (97.60%) of the 2500 images.

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 $\sigma = 75$).

So the conclusion is the following : their approach is particularly well suited for medium to high noise levels ($\sigma \geq 50$). MLPs outperform the previous state-of-the-art on all noise levels, but for $\sigma = 10$. However, MLPs has to be trained on each noise level in order to achieve good results.

Comparison on different type of noise -   Their learning method can be use on other type of noise just changing the training data set, by corrupting the image dataset with the type of noise given. For stripe noise, Salt and Pepper noise, MLP outperform BM3D and the other classical methods.
For JPEG quantization artifacts and mixed Poisson-Gaussian noise MLP seems to be competitive with the state-of-the-art.

4 Theoritical bound

Clustering-based bounds - Actual 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 outperform 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 on how well any denoising algorithm can perform, which depends on the patch size.
is about patch-based method estimate denoising bounds in a Bayesian framework, for a given patch size. Their approach results exceed these bounds, that's beacause they use larger patches than assumed by Levin and Nadler (2011).
Similar bounds estimated for infinite patch sizes are estimated by Levin et al (2012). MLPs make significant progress in achieving these bounds: their approach achieves almost half the theoretically possible gain over BM3D. Levin et al (2012) agree with Chatterjee and Milanfar (2010) that there is little room for improvement on patches with complex textures.

5 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 MLPs can achieve better results, especially images with repeating structures, on which BM3D and NLSC outperform MLPs.
They 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 by the block-matching MLP on the 11 *standard test images* is $0,07$dB higher than the plain MLP. It outperfoms 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 : plain MLPs perform better on images with smooth surfaces, and block-matching MLPs provide 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.

6 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.