Mini-project 1: Noise2Noise

Mihaela Berezantev, Maximilian Gangloff, Gabriel Maquignaz
EE-559 Deep Learning, EPFL, Switzerland

***Abstract*—The U-Net for Biomedical image segmentation was first introduced in 2015 [1]. Using the U-Net as a base model, the Noise2Noise model was created in 2018 which enabled to train a model that restores images while only training with pairs of noisy images [2].**
In this paper, using only PyTorch, we will implement and compare the two models while adapting different modules and hyperparameters to make the models best fit our data while respecting the given time constraints of less than 10 minutes of training per epoch on an entry-level GPU. This resulted in a final model with Psnr of 25.54dB.

I. Introduction

The denoising of images is one of the most fundamental problems in image processing and computer vision. It represents the recovery of a high quality image from a noisy one. Different types of noises require different methods for the noise to be best removed. In this paper, we will focus on unbiased and uniformly distributed noise that is added to the images. Traditionally, the deep networks trained for denoising images are trained using a set of noisy images and a set of clean images which corresponds to the ground truth. Then during training, the quadratic loss is minimized . For additive Gaussian noise, this would correspond to

$$W = argmin_w \frac{1}{N} \sum_{n=1}^{N} ||x_n - \phi(x_n + \epsilon_n; w)||^2 \quad (1)$$

where x_n are the images and ϵ_n the random Gaussian noise.
However, sometimes it is not possible to obtain clean images especially in the medical field e.g. for MRI images. Nonetheless, even without having clean images, it is still possible to predict the ground truth. For this, pairs of images with independent additive and unbiased noise are needed for training. In this case we have that

$$\begin{aligned} &\mathbb{E}\left[||\phi(X+\epsilon;\theta) - (X+\theta)||^2\right] \\ &= \mathbb{E}\left[||\phi(X+\epsilon;\theta) - X||^2\right] + \mathbb{E}\left[||\theta||^2\right] \end{aligned} \quad (2)$$

Hence minimizing the quadratic loss between the pairs of noisy images is equivalent to minimizing equation 1.

$$\begin{aligned} &argmin_\theta\, \mathbb{E}\left[||\phi(X+\epsilon;\theta) - (X+\theta)||^2\right] \\ &= argmin_\theta\, \mathbb{E}\left[||\phi(X+\epsilon;\theta) - X||^2\right] \end{aligned} \quad (3)$$

II. Models and Methods

In this section, we will present the models that were implemented and the data that was used to train them.

A. Data Handling

The data handling is a major part in deep learning. In our case, we had 50'000 noisy pairs of images for the training. Each image corresponds to a downsampled, pixaleted image of size 32x32. An additional validation set with pairs of images containing a noisy image and their corresponding ground truth was provided to test our model.
To get a more stable model, we decided to augment our data with a horizontal and vertical flip. The total size of the training data was then 150'000x3x32x32. Adding additional noise or to blur the images to augment the data made no sense and did not improve the model since the images already contained noise.
The data was normalized to be in the range of -0.5 and 0.5 instead of 0 and 255 which helps the model to converge faster.

B. Model choices

Two different kind of models were implemented for this paper. The U-Net and the Noise2Noise model which is an adaption of the U-Net model.

1) U-Net
Introduced in 2015 for Biomedical image segmentation[1], its core feature is merging a convulational network with a residual network. It consists of an downsampling part and an upsampling part where each part is connected with a skip

connection. The downsampling part, also known as the contracting part helps to identify what is important in an image. The upsampling part, also know as the expansive part helps to locate where the important part is. These two parts are also known as the encoder and decoder. Since these two parts have multiple convolutional and deconvolutional layers, they suffer from the degradation problem.[3] This is where the skip connections comes into play and passes the feature maps forward which enables to use the fine-grained details learned in the encoder part in the decoder part.[4]

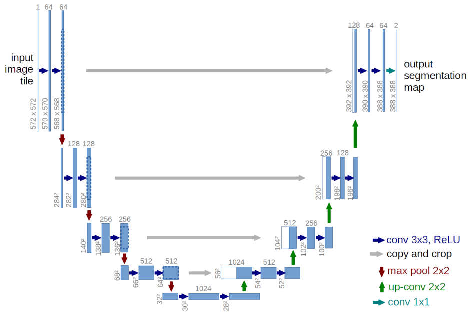

Figure 1. Original architecture of the U-Net.[1]

For our implementation of the U-Net, the sizes were adapted accordingly to fit our training data. Each downsampling block consists of the following modules:

- Conv2d with a 3x3 kernel and same padding
- Conv2d with a 3x3 kernel and same padding
- LeakyReLU

To then downsample to the next block, average pooling and LeakyReLU instead of max pooling and ReLU was used which increased the Psnr slightly. The upsampling modules are the same as the downsampling modules. To up sample to the next block, a ConvTranspose2d module with kernel size and stride of 2 was used. For the U-Net and Noise2Noise model, Adam was used as otimizer.

2) *Noise2Noise*

The Noise2Noise model takes the U-Net model as a base but mainly reduces its block and feature sizes. There, only the first block consists of a double convolution and all other blocks of the downsampling consists of a single convolution. The modules of the upsampling part are then the same with the exception of the sizes that are different. However, in the Noise2Noise model, the skip connections were added at slightly different positions. The main change that was made to the original Noise2Noise model is that average pooling was used instead of max pooling. This increased slightly the resulting Psnr of its model.

C. *Hyperparameter tuning*

Besides the structural optimization, there are also the hyperparameters that play a big role in the optimization of a model. Both models were initialized with the same parameters for the weights and the biases. The weights of the ConvTranspose2d and Conv2d modules where initialized after He et al. 2015[5]

Besides the weight initialization, the two main hyperparameters that were tuned for both models were the mini batch sizes and the learning rate. Both are of extreme importance for our model due to the required time constraints. The two hyperparameters were optimized using grid search.

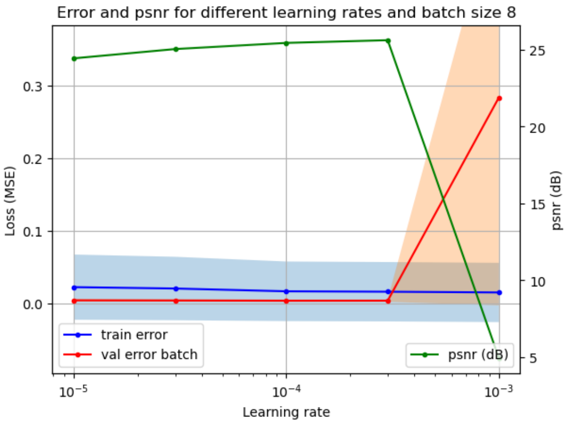

Figure 2. Tuned hyperparameters for the U-Net

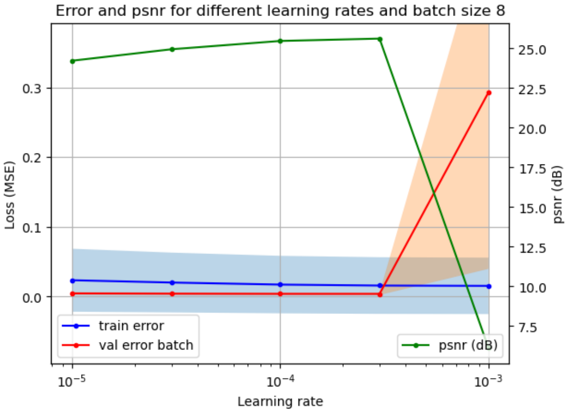

Figure 3. Tuned hyperparameters for the Noise2Noise model

We observe from the above figures, that the optimal hyperparameters for both models were xx

III. Results

Using the hyperparameters described before, our best results for both models after a small number of epoches of training were:

Model	epochs	learning rate	batch size	psnr (dB)	time/epoch (min)
U-Net	3	3e-4	8	25.63	3.9
Noise2Noise	3	3e-4	8	25.66	3.57

Table I
Psnr in dB for optimized U-Net and Noise2Noise

The training was done on a GTX 1050 Ti with 4GB of GDDR5 memory and a Tesla T4.
Different changes to try to improve the accuracy of the Noise2Noise model were made but did not result in any improvement. The changes consisted of adding batch normalization, changing the pooling layer from max pooling to average pooling and using recursive evaluation.

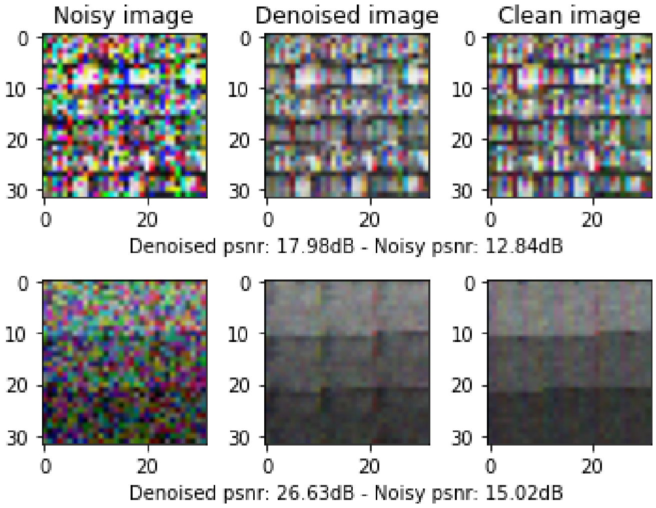

Figure 4. Resulting images using the Noise2Noise model

We can clearly see, that in most cases the model did correctly learn and has a good prediction of the ground truth. However, there are also images that are also images that are poorly predicted as the first one.

IV. Discussion

Both ReLU and LeakyReLU were tested but LeakyReLU slightly increased the Psnr.
In our case, the batch normalization did not improve the resulting Psnr. This can be explained by the fact that due to the optimal mini batch size being pretty low, when adding the batch normalization, the mini batch size needs to be increased for the batch normalization to have a robust estimate of the mean and std. However, when increasing the batch size too high, the resulting images started to become blurry.
The same goes for the recursive evaluation. After two to three evaluations, the images seemed to be even clearer than the provided ground truth. However, when evaluating even more, the images became blurry over time. With every recursive evaluation, the Psnr decreased further.
When looking at the predicted images with the best and worst Psnr, we can see that the first ground truth image contains much more colors compared to the second one. The model seems to be strong in predicting the shape but fails to predict the colors.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[2] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2noise: Learning image restoration without clean data," 2018.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[4] J. Dong, X. Mao, C. Shen, and Y. Yang, "Unsupervised feature learning with symmetrically connected convolutional denoising auto-encoders," *CoRR*, vol. abs/1611.09119, 2016. [Online]. Available: http://arxiv.org/abs/1611.09119

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *CoRR*, vol. abs/1502.01852, 2015. [Online]. Available: http://arxiv.org/abs/1502.01852