Exercise 5

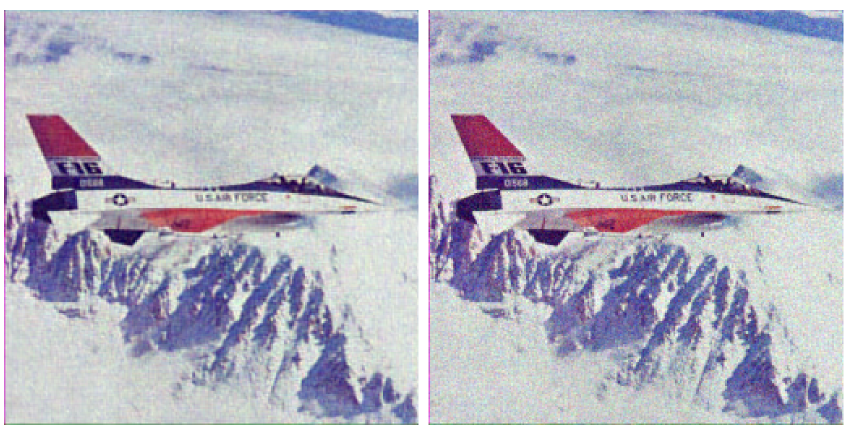
Shahaf & Omri

February 8, 2019

1 1D

1.1

1. As we can see by the results we achieved ,The inductive bias assumption does hold , and indeed we can see that a somewhat cleaned image emerges before the noise is being reconstructed.

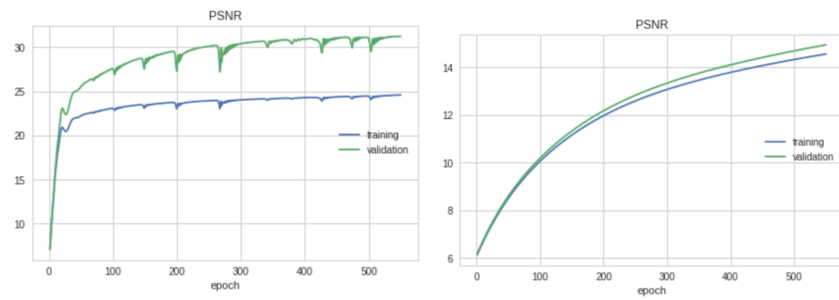


(a) 1000 epochs (b) 5000 epochs

Figure 1: A cleaner image emerges before noise is starting to be reconstructed

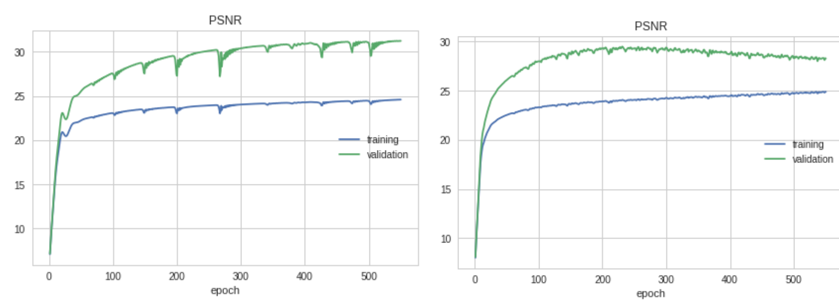
1.2

We found using ADAM for optimization resulted in more pleasing results visually and faster convergence. In addition, using batch normalization was important for good results and a smoother image. We found learning rate of 1e-3 was a good fit, while 1e-5 was too slow, and bigger learning rates, such as 1e1, lead to bad results. Increasing the number of filters also helped to achieve better results. Our architecture is 5 convolution layers and 1 fully connected layer with 4 up-samples. We choose to use LeakyRelu as our activation function, and we used interpolation and then convolution in attempt to get less artifacts.



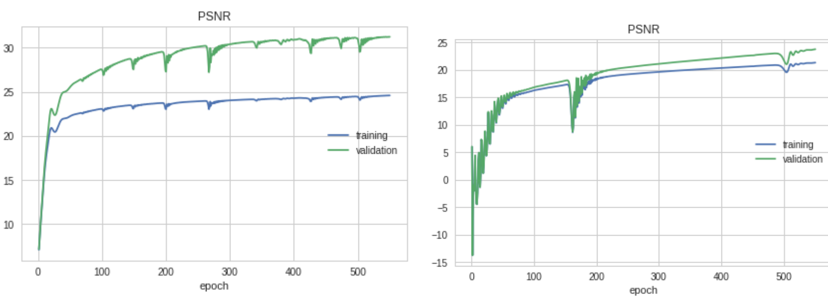
(a) using ADAM (b) using SGD

Figure 2: A major difference in results with using ADAM instead of SGD can be seen



(a) Using L2 (b) Using L1

Figure 3: As we see, using L1 gets lower maximum, for the best results we used L2.



(a) using 1e-3 learning rate (b) using 1e-1 learning rate

Figure 4: We see that a high learning rate will make it harder to converge , while a too small learning rate will cause convergence to be too slow.

1.3 PSNR

First image 26.3 second 29.96 third 30.729 forth 30 the fifth 29.45 the six 28.14 the seven 30.36, the eight image 30.98 and the last has 30.3 , average of 29.58.

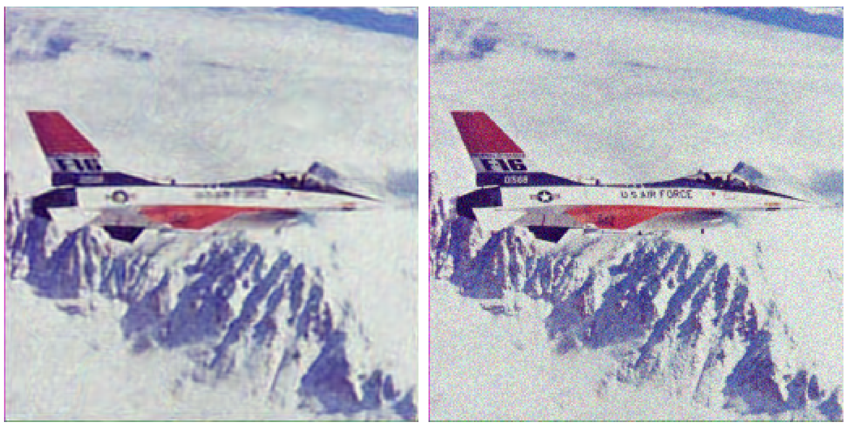
1.4

we have $6,468,835$ trainable parameters are in the model, $1,376,780,288$ addition operations and $1,376,772,096$ multiplication operations in our model.

2 2D

2.1

1. As we can see in 5, the inductive bias assumption does hold , and indeed we can see that a somewhat cleaned image emerges before the noise is being reconstructed.

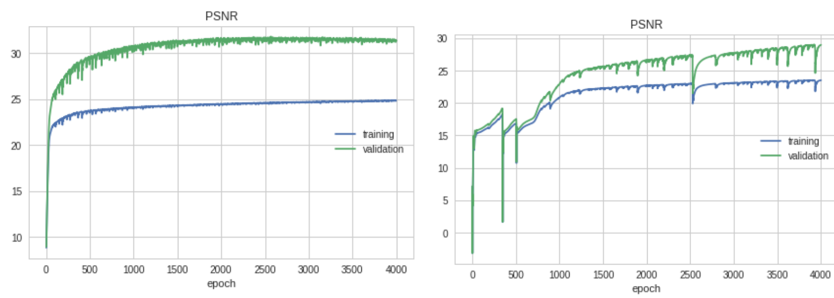


(a) 4000 epochs (b) Image with the noise

Figure 5: As we can see a lot of the noise was indeed removed

2.2

We found that using batch normalization was important for good results and a smoother image. Increasing the number of filters and parameters also helped achieving better results. We noticed that L2 is still better for us then using L1. Our net is made of 9 convolution layers with 4 upsamples and 4 downsamples. We used at the beginning layers with convolution with strides (to downsample) and activation function and then we used layers with interpolation and then convolution, batch-norm and activation in attempt to get less artifacts.



(a) Using 1e-3 learning rate (b) Using 1e-1 learning rate

Figure 6: Learning rate of 1e-3 performs better.)

2.3 PSNR

First image 27.81 second 29.49 third 29.56 forth 27.95 the fifth 28.36 the six 27.72 the seven 28.56, the eight image 31.41 and the last has 28.61, in total we have average of 28.83.

2.4

we have $254,371$ trainable parameters are in the model, $892,862,464$ addition operations and $892,862,464$ multiplication operations in our model.

3 2D with skip-connections

3.1

1. The architecture is the same as before but with "skip connections" , in which the information is propagated through residual links. The network performance is similar to the 2D case. The network does output a cleaner image in comparison to the noisy image but not completely smooth, as can be seen in Figure 7. Indeed, it's easier for the network to output a smoother, neutral image. The network does manage to over-fit the noise but very slowly. After 10,000 epochs the PSNR with respect to the noisy image, equalizes the PSNR with respect to clean image, As shown in Figure 8. If we let the network train longer we will probably see the rend continuing albeit slowly.

(a) Noisy (b) Result (c) Clean

Figure 7: 2D with skip-connections after 1500 epochs

3.2

We Tried using half the amount of filters across all layers but we pretty much the same results except that with more filters PSNR grow faster, as can be seen in Figure 9. We decided to keep using the same amount of filters.

We also tried switching to a concatenation architecture. The PSNR for the residual links network was slightly better as shown in figure 10.

3.3 PSNR

Results for residual links network: PSNR by image: 26, 29.92, 30.61, 29.59, 30.05, 28.16, 30.1, 31.39, 30.1. Total Average: 29.54.

3.4

we have 1866211 trainable parameters are in the model, 3093692416 addition operations and 3098542080 multiplication operations in our model.

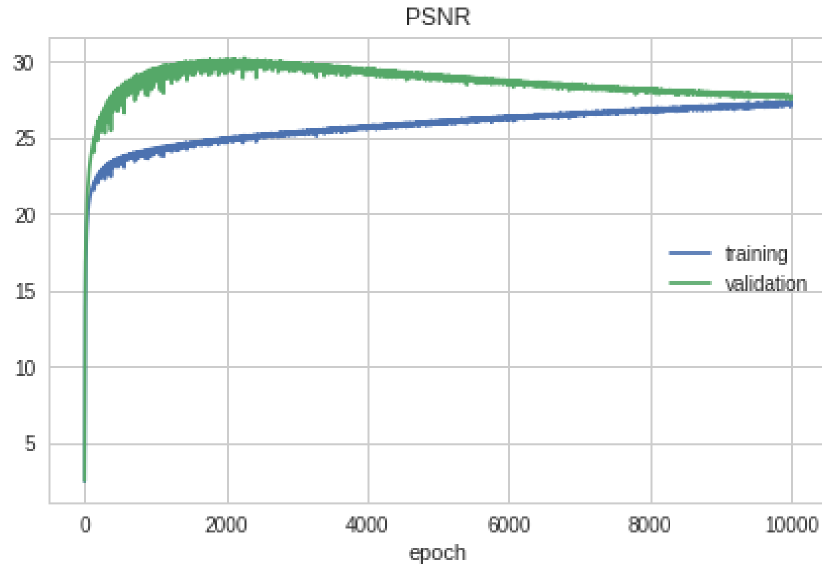
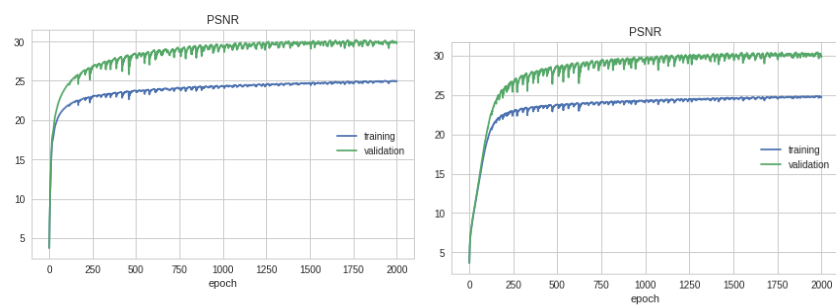
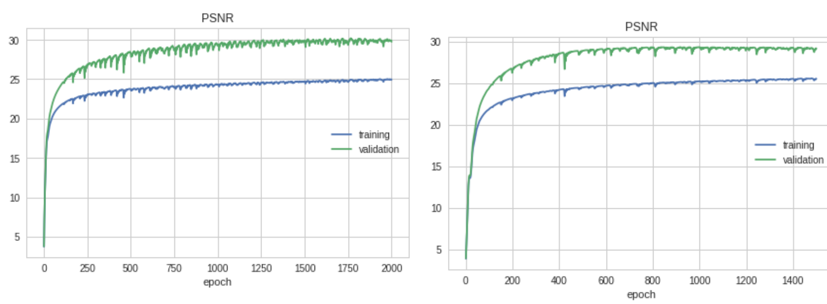


Figure 8: PSNR for 10,000 epochs on image no. 5



(a) Vanilla architecture (b) Half the filters

Figure 9: PSNR using different amount of filters across all layers.



(a) Residual Links (b) With concatenation

Figure 10: PSNR - Residual links vs. Concatenation