Brain tumor MRI image denoising and reconstruction using Autoencoders

Martin Palazzo

July 2019

Abstract

This report is presented as the final project of the doctoral course "Digital Processing of Images" at UTN Buenos Aires under the supervision of Professor Claudio Delrieux.

1 Background

This work has been made with the objective to show the denoising and reconstruction capacity of autoencoders using images as input data. The dataset is composed of MRI brain images of cancer patients.

1.1 Autoencoders

Autoencoders are feedforward networks that learn two functions simultaneously: an encoder and decoder. The encoder maps the original input domain X to a new domain named latent space Z of dimension L. The decoder then maps from Z to the original input space X. The output of the decoder is also a reconstruction feedforward network. Since we aim to have a lower dimensionality at the latent space Z, the autoencoder is forced to build an encoder function that captures all the salient features from the training data as much as possible [2]. The encoder and decoder functions are defined as $z = f(x)$ and $\tilde{x} = g(z)$ respectively where z are the samples at the learned latent space and $\tilde{x}$ are the reconstructed samples on $\mathcal{X}$. With the previous definition, the autoencoder loss function to minimize is formalized as

$$E\left(x, \tilde{x}\right) = E\left(x, g\left(f\left(x\right)\right)\right)$$

where E is the error function that penalizes $g\left(f\left(x\right)\right)$ to be different to x. In this work the measure of this error function is the cross entropy loss and the Mean Squared Error. Then the encoder F and decoder G functions can be defined as [3]

$$z = F\left(x, \mathbf{W}_F\right) = \sigma\left(\mathbf{W}_F x + \mathbf{b}_F\right)$$

$$\tilde{x} = G\left(z, \mathbf{W}_G\right) = \sigma\left(\mathbf{W}_G \mathbf{z} + \mathbf{b}_G\right)$$

where $F\left(\cdot, \mathbf{W}_F\right)$ and $G\left(\cdot, \mathbf{W}_G\right)$ correspond to the encoding and decoding functions respectively and $\sigma\left(\cdot\right)$ is an activation function. The original input sample is $\mathbf{x} \in \mathcal{X}$, $\tilde{x} \in \mathcal{X}$ is the reconstructed samples and z the corresponding latent ones which dimension is lower than x. The tensors $\mathbf{W}$ and $\mathbf{b}$ corresponds to the trained weights and biases of the encoder and decoder networks. These parameters are learned by backpropagation in order to minimize the loss function by the optimizer. This work uses Adaptive Moment Estimation (Adam) [4] optimizer to learn the weights of the network that minimizes the loss function. Adam is a novel first-order stochastic optimization technique. It computes an adaptive learning rate depending on the gradient mean.

1.2 Types of autoencoders for image denoising

This work uses different models within the Autoencoder family. In this work these are Convolutional Autoencoders (CAE), Denoising Autoencoders and Variational Autoencoder. In this work images X are modified with low pass filters, random noise and a combination of both to get X_n. The key idea is to train the autoencoder with X_n and force it to reconstruct the images similar to the original X. The loss function is computed between $L(x, \tilde{x})$. By this way the autoencoder learns how to filter the noise on the images.
When data are images, consider each sample as a vector is not the best option since the sense of surface is missed. Then the convolutional AE (CAE) has convolution layers at the beginning of the encoding function and at the end of the decoding one. The convolutional layers are composed by kernels of

3×3. Images contains only one channel since are under a gray scale.
Finally, a novel type of autoencoder known as Variational Autoencoder [5] is used to compare its performance with the standard ones. Variational Autoencoders are probabilistic models that learns a latent space composed by a mean and standard deviation vectors. This is possible thanks to the Kullback–Leibler divergence (KL divergence) added to the loss function.

2 Experiments

Different types of input data are used to test the denoising and reconstructing capacity of each type of autoencoder. From the original input data, 3 types of image corruption are generated.

2.1 Dataset

The dataset is composed by 200 squared images of 128 pixels per side. For training 180 images were used while the rest 20 are keep as an independent test set. Images are similar to the ones obtained in the [1].

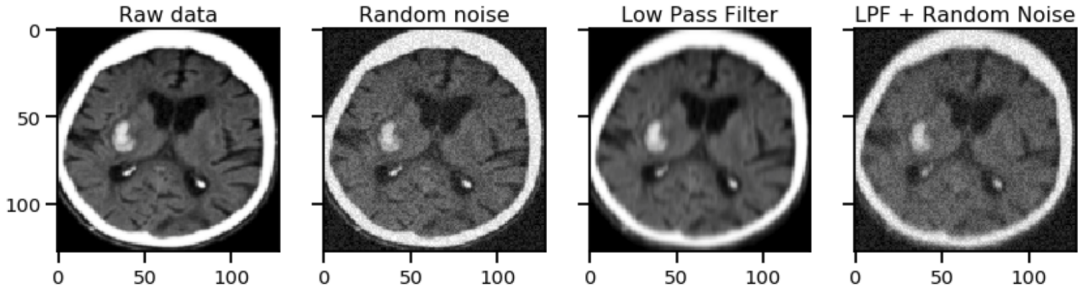

Figure 1: Types of input data

The input images consist in four subsets which are the original raw brain data X, the images plus a random noise $X + \alpha N$, the original images pre processed by a low pass filter (LPF) and finally images pre processed with LPF plus random noise. The parameter α represents the total amount of noise added to the images.

2.2 Results

First a standard convolutional autoencoder has been trained with the original input data to ensure its reconstruction capacity without adding any type of noise.

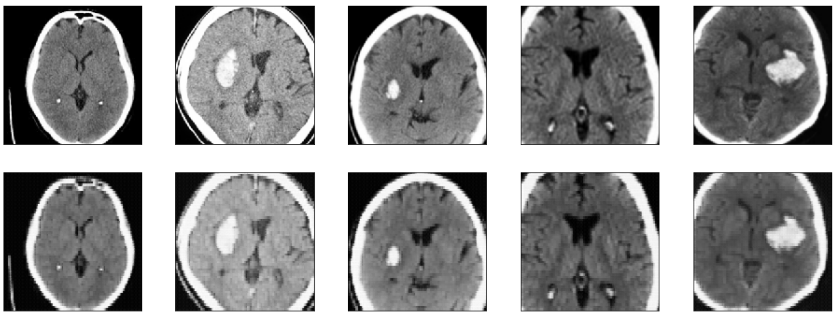

Figure 2: Reconstruction of raw images using a standard convolutional autoencoder. First row corresponds to the original raw data. Second row corresponds to raw data corrupted by LPF and random noise. Third row represents the reconstruction after convolutional autoencoder.

Figure 2 shows an acceptable reconstruction capacity of the model.
Then using the same standard convolutional autoencoder different types of noise were added to the original images. Figure 3 the corrupted images with low pass filter and noise (middle row) and the reconstructed images (3rd row) obtained from the model while using as target the original input data. Despite the autoencoder reveals some difficulties to fully reconstruct the output image, it is clear that the reconstruction has less noise and has more details in the low and high frequencies in comparison with the corrupted ones.

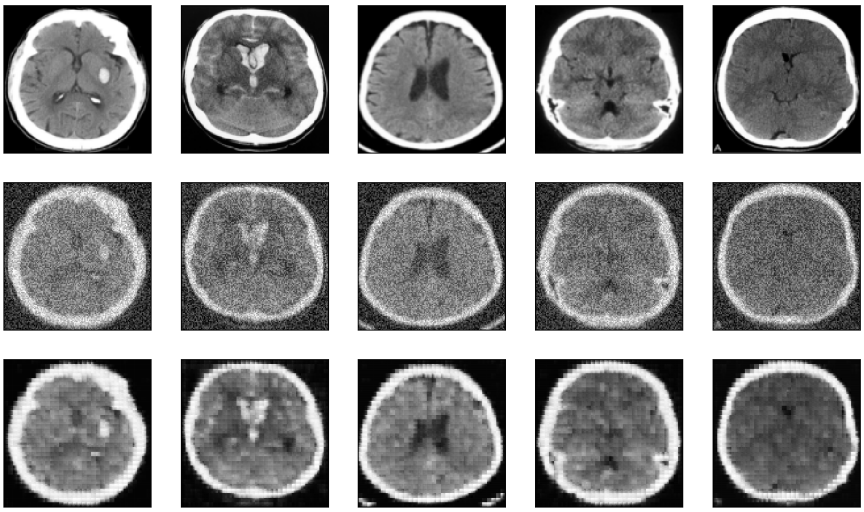

Figure 3: Reconstruction of images corrupted by LPF and noise using a standard convolutional autoencoder. First row are original images. Second row are the corrupted images. Third row corresponds to the output images which are reconstructed and denoised after autoencoder.

Figure 4 shows the loss function during the training epochs. It is shown that after 30 epochs the loss converges. This evidences that it would be necessary to add early stopping policies to the training strategy.

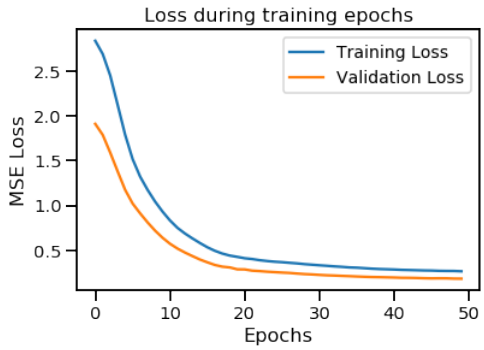

Figure 4: Loss of convolutional autoencoder during training epochs.

Finally, the experiment executed before on convolutional autoencoders is repeated with variational autoencoders to compare the denoising and reconstruction capacity of both models.

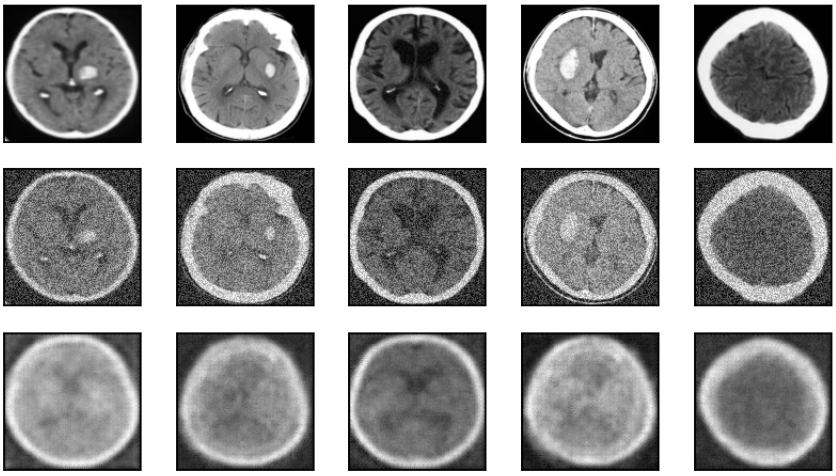

Figure 5: Reconstruction of corrupted images using a variational convolutional autoencoder. First row corresponds to the original raw data. Second row corresponds to raw data corrupted by LPF and random noise. Third row represents the reconstruction after variational convolutional autoencoder.

The reconstruction results using as input data corrupted by LPF and random noise reveals that the strategy implemented does not performs as expected. The images are poorly reconstructed although the morfology of the tumors remains almost the same. Variational autoencoders is a more complex model which requires a deeper design of the network.

2.3 Computational resources

All experiments where executed in Python 3.6 using Pytorch 1.1 version using a Intel i7-5600U 2.60GHz CPU.

3 Conclusions

In this work 200 MRI images of brain tumor patients were used to train different types of autoencoders to analyze the reconstruction and denoising capacity. Using a standard convolutional autoencoder the reconstruction is the best, nevertheless in this case training images are not corrupted so the problem is relatively easy to solve. Then the AE used before is trained with corrupted images with low pass filters and random noise in order to evaluate

the capacity of reconstruction and denoising. Despite the results are not perfect, there is clear evidence that the corrupted images are reconstructed with a significative lower level of noise. Finally the same experiment is executed with variational autoencoders. The last result is not enough to reconstruct properly the images. Next work would include a deep study of variational autoencoder training strategy to outperform the actual results. As a final conclusion, in this work is explained how autoencoders can be used to reconstruct and denoise corrupted images.

References

[1] Kenneth Clark, Bruce Vendt, Kirk Smith, John Freymann, Justin Kirby, Paul Koppel, Stephen Moore, Stanley Phillips, David Maffitt, Michael Pringle, et al. The cancer imaging archive (tcia): maintaining and operating a public information repository. *Journal of digital imaging*, 26(6):1045–1057, 2013.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[3] Michael Kampffmeyer, Sigurd Løkse, Filippo M Bianchi, Robert Jenssen, and Lorenzo Livi. Deep kernelized autoencoders. In *Scandinavian Conference on Image Analysis*, pages 419–430. Springer, 2017.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.