CT reconstruction using neural network

Hoyeon Lee

KAIST

July 3, 2017

Contents

- After neural netowork showed better performance in many fields, many researchers tried to adopt it to reconstruction
- In medical application, most of their effort concentrated to post- or preprocessing

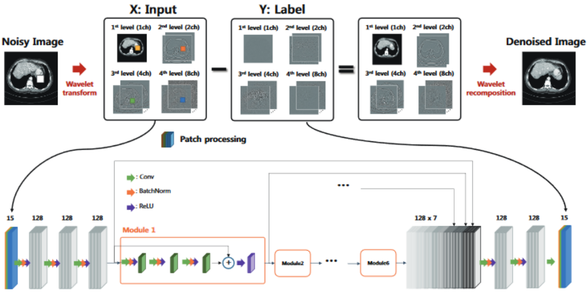

Fig. 1. The proposed WavResNet architecture for low-dose X-ray CT reconstruction.

Figure 1. CNN for reducing noise using wavlet transform

Introduction

- After neural netowork showed better performance in many fields, many researchers tried to adopt it to reconstruction
- In medical application, most of their effort concentrated to post- or preprocessing

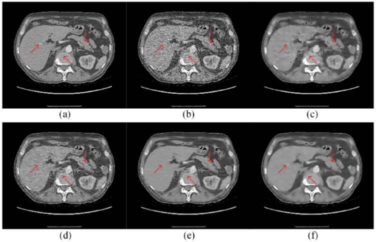

Fig. 3. Results of an abdomen image. (a) Original normal-dose image; (b) the low-dose image; (c) the ASD-POCS image; (d) the KSVD image; (e) the BM3D image; (f) the CNN processed low-dose image.

Figure 2. Filter optimization for low-dose CT

Introduction

- After neural netowork showed better performance in many fields, many researchers tried to adopt it to reconstruction
- In medical application, most of their effort concentrated to post- or preprocessing

Algorithm 1 ANN boost of FBP.

1) Create training data based on M clean CT images:
   For every training image:
   a) Simulate low count CT sinogram with combined Poisson and Gaussian noise.
   b) Reconstruct the image from the simulated sinogram using FBP with K low-pass filters.
   c) From the set of all overlapped patches randomly extract a subset of patches for training, with probability proportional to the average gradient norm in the patch (in order to better treat edges).
   d) Central regions of the corresponding clean patches are extracted to form output training data.
   e) Normalize the data according to equation V.2.
2) Train ANN with e.g. stochastic gradient descent using the above training data.
3) Image fusion with ANN:
   a) Perform K reconstructions with the same low-pass filters as in the training.
   b) ANN application: for each pixel do
      - Normalize the data according to equation V.2.
      - Apply trained ANN.
      - De-normalize using stored α_1 and α_2 (V.2).
      - Position each patch in its place: each estimated pixel is computed by averaging over all the contributions.

Figure 3. Spatially optimized filtering using ANN

- After neural netowork showed better performance in many fields, many researchers tried to adopt it to reconstruction
- In medical application, most of their effort concentrated to post- or preprocessing

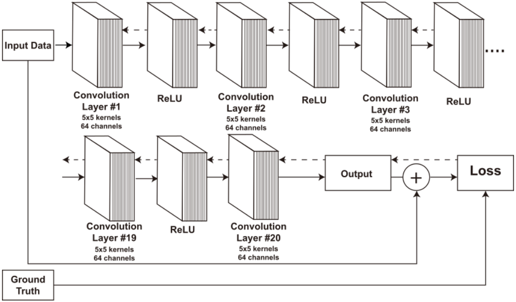

Figure 4. Network for view-interpolation using CNN

"Can we reconstruct projection data using neural network?"

- Actually, more than a decade ago,
- Some researchers applied it for reconstruction

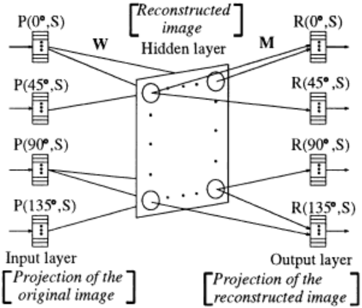

Fig. 2 Structure of the network.

Figure 5. Autoencoder based reconstruction

- Actually, more than a decade ago,
- Some researchers applied it for reconstruction

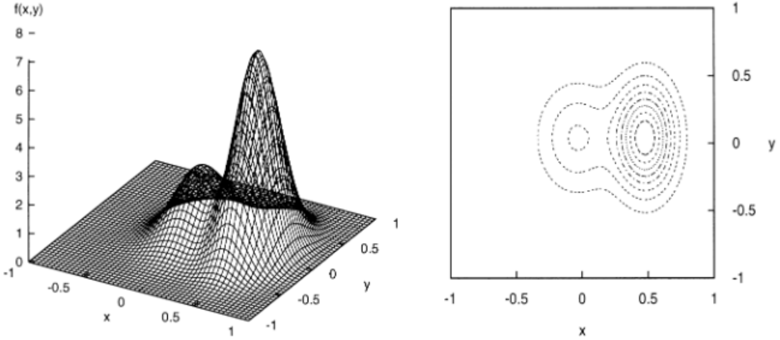

Figure 6. Reconstructed signal using neural network

- Actually, more than a decade ago,
- Some researchers applied it for reconstruction, or optimization of cost function

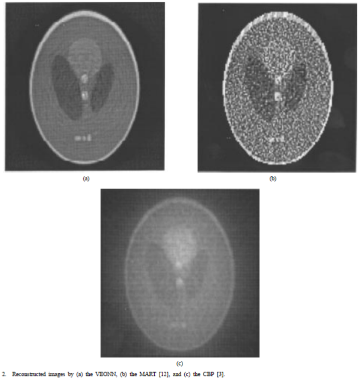

Figure 7. Optimizing using neural network

Purpose

- Purpose of this study train neural network to perform backprojection for given projection data

Methods

- Projection operator is linear process represented as following equation

$$P = Af$$

- Where f is reconstructed image, A is system matrix, and P is projection data
- If detector has m different pixels, and obtained projection for k different angles, the P would have $M = m \times k$ elements
- And if the f has N elements total, then the system matrix A would have $M \times N$ shape

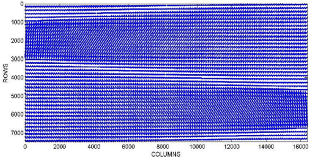

Fig 2. Sparsity pattern of the system matrix. The figure shows a part of the matrix data structure that corresponds to 7000 rows and 16000 columns. doi:10.1016/j.procs.2013.05.308.

Figure 8. A part of system matrix

Methods

- By inverting the system matrix to the other hand, we can reconstruct images from projection data

$$f = A^{-1}P$$

- It is still linear problem and can be solved using simple neural network

- Since, neural network is founded based on the following equation

$$y = \sigma(Ax + b)$$

,where y is output vector of a network, x is input vector of a network σ is activation function, A is weight, and b is bias term

- I started with simple parallel projection
- I forward-projected Shepp-Logan phantom using `radon` function in *Matlab* into 1440 views

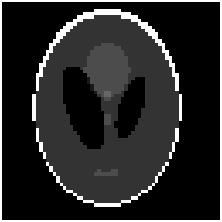

Figure 9. Shepp-Logan phantom

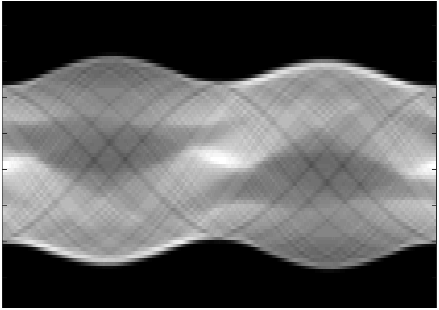

Figure 10. Radon transformed Shepp-Logan phantom

- Constructed neural network to train the backprojection operation

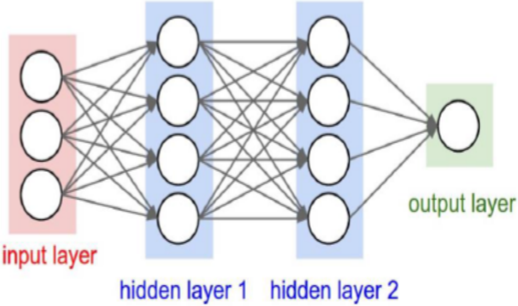

Figure 11. Basic structure of neural network

- And used activation function for non-linearity of the network

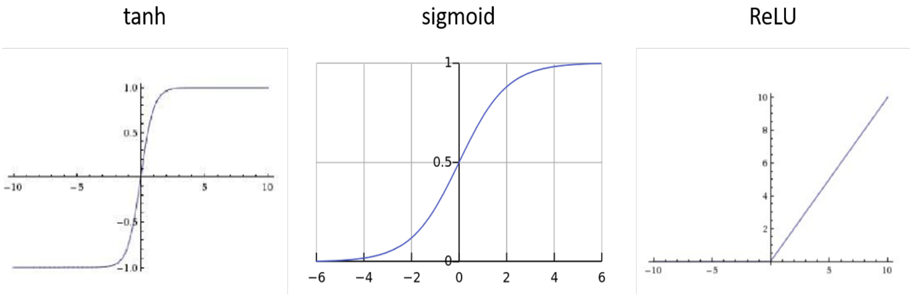

Figure 12. Type of possible activation functions

Methods

- I stacked fully-connected layers in `caffe` for nueral network and used ReLU for activation function
- And trained the network to minimize euclidean distance between output of the network and backprojected images

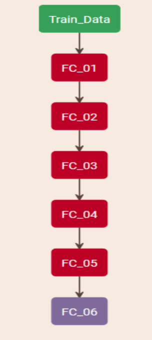

Figure 13. Network without ReLU, and Dropout

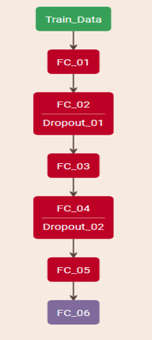

Figure 14. Network without ReLU, and wiht Dropout

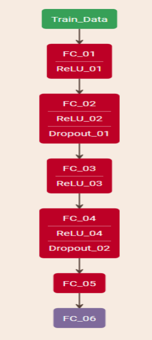

Figure 15. Network with ReLU, and Dropout

Results

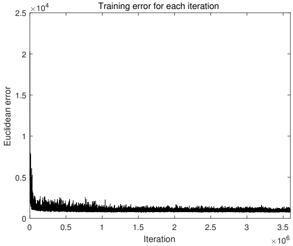

Figure 16. Training loss of network #1

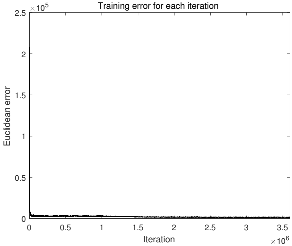

Figure 17. Training loss of of network #2

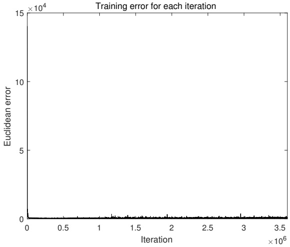

Figure 18. Training loss of of network #3

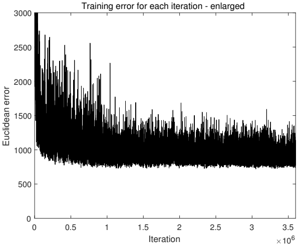

Figure 19. Training loss of network #1

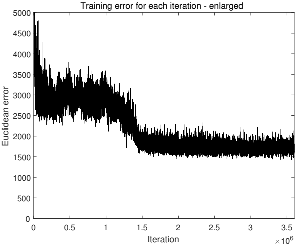

Figure 20. Training loss of of network #2

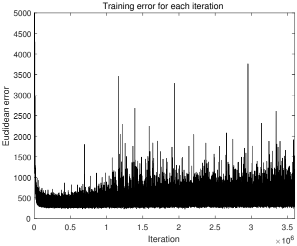

Figure 21. Training loss of of network #3

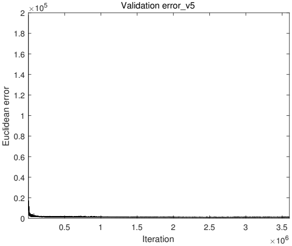

Figure 22. Validation loss of network #1

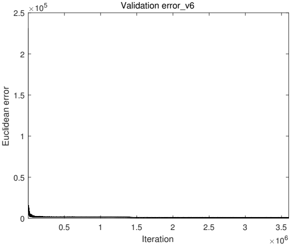

Figure 23. Validation loss of of network #2

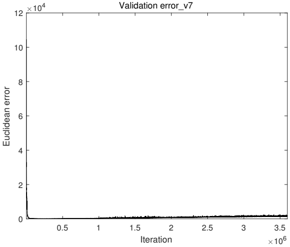

Figure 24. Validation loss of of network #3

- **It is strange validation error of the third network getting bigger!**

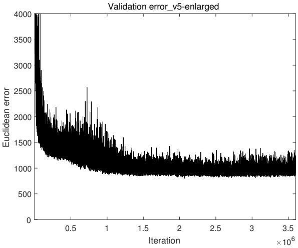

Figure 25. Validation loss of network #1

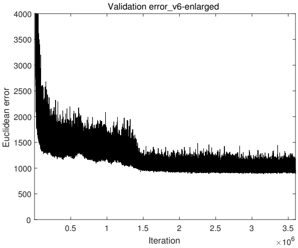

Figure 26. Validation loss of of network #2

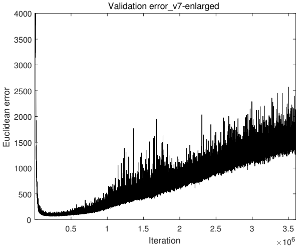

Figure 27. Validation loss of of network #3

Results

- Comparison between backprojected image using `iradon` function and using neural network #1

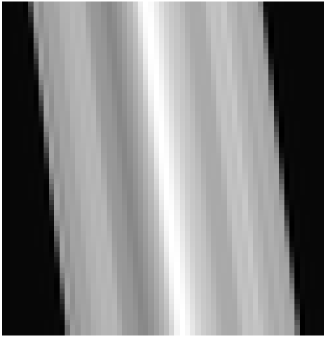

Figure 28. Backprojected image using `iradon`

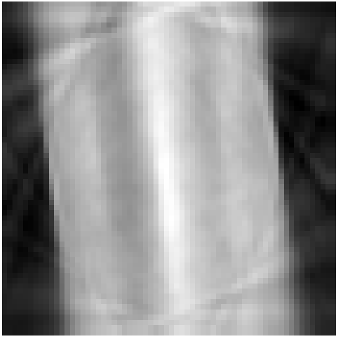

Figure 29. Backprojected image using nueral network

- And reconstructed images

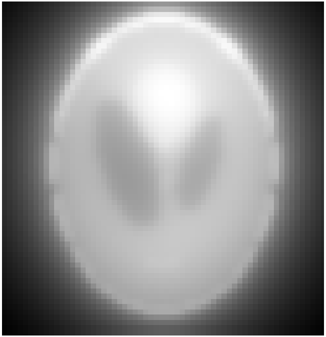

Figure 30. Reconstructed image `iradon`

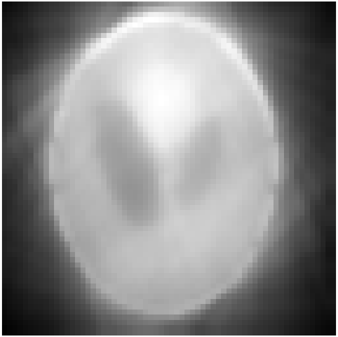

Figure 31. Reconstructed image using nueral network

- Comparison between image using `iradon` function and using neural network #2

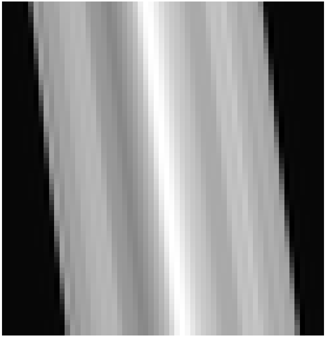

Figure 32. Backprojected image using `iradon`

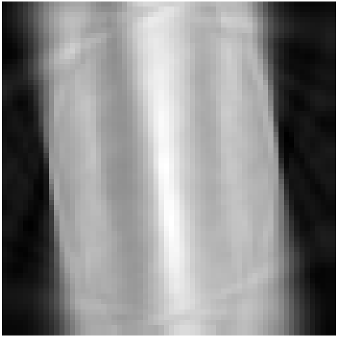

Figure 33. Backprojected image using nueral network

- And reconstructed images

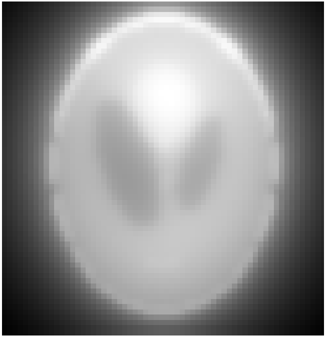

Figure 34. Reconstructed image `iradon`

Figure 35. Reconstructed image using nueral network

- Comparison between backprojected image using `iradon` function and using neural network #3

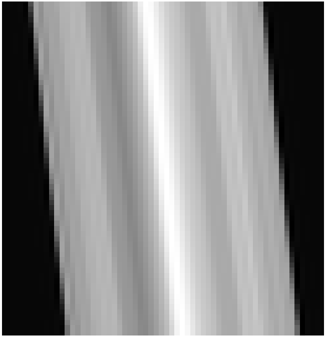

Figure 36. Backprojected image using `iradon`

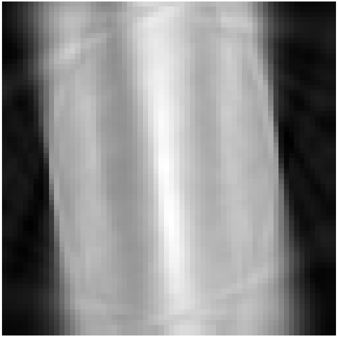

Figure 37. Backprojected image using nueral network

- And reconstructed images

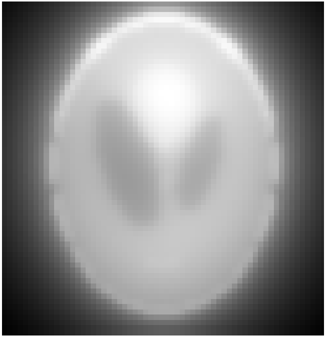

Figure 38. Reconstructed image `iradon`

Figure 39. Reconstructed image using nueral network

Results

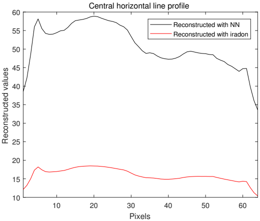

Figure 40. Central horizontal line profile of network #1

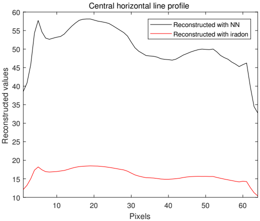

Figure 41. Central horizontal line profile of network #2

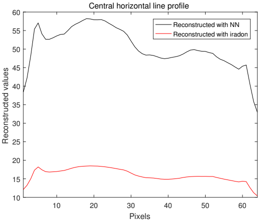

Figure 42. Central horizontal line profile of network #3

Table 1. RMSE between `iraon` results and neural network result

	Backprojected images	Reconstructed image
Network #1	0.69	27.79
Network #2	0.68	27.68
Network #3	0.88	27.86

Conclusion

- In this study, I trained neural network to perform backprojection of sinogram
- The networks trained enough to mimick backprojection
- However, the reconstructed value of pixels are not exactly same as reconstructed image using inverse radon transformation

Conclusion

- Possible causes of the error are;
  - Lack of variety of data
  - To shallow network to learn backprojection exactly
  - Inappropriate activation function
  - Inappropriate error function to learn backprojection

- Maybe, it is impossible to teach a neural network to learn the exact backprojection operation

What can we do with this?

- Reconstructing projection data obtained with task-optimized orbit
- Optimizing filtering to reduce artifacts
- Excluding acquisition angle from acquisition/reconstruction parameter
- Build fully-automatic reconstruction system

Next questions

- Are activation functions are really required?
  - The reconstruction process are all linear process, including filtering
  - However, the non-linearity made by activation function is essential feature of deep neural network
  - ReLU is linear above zero and reconstruction problem normally have value greater than zero, which means activation function may not act as we expeteced
- How to scale the network or normalize projection data to apply a trained network globally?
- Will this network helpful to constructing CNN, which takes project data as input, and desired output is reconstructed images?

Future works

- Decrease training loss and error between reconstructed images
- Add convolutional layer before backprojection as filtering process
- Increase size of reconstructed images for more realistic results
- Train the network with different beam geometry
- Make the network scalable or normalize projection data to deal with various geometries