

Deep Learning Computed Tomography

Tobias Würfl^(✉), Florin C. Ghesu, Vincent Christlein, and Andreas Maier

Pattern Recognition Laboratory, Friedrich-Alexander-University
Erlangen-Nuremberg, Erlangen, Germany
`wuerflts@outlook.com`

Abstract. In this paper, we demonstrate that image reconstruction can be expressed in terms of neural networks. We show that filtered back-projection can be mapped identically onto a deep neural network architecture. As for the case of iterative reconstruction, the straight forward realization as matrix multiplication is not feasible. Thus, we propose to compute the back-projection layer efficiently as fixed function and its gradient as projection operation. This allows a data-driven approach for joint optimization of correction steps in projection domain and image domain. As a proof of concept, we demonstrate that we are able to learn weightings and additional filter layers that consistently reduce the reconstruction error of a limited angle reconstruction by a factor of two while keeping the same computational complexity as filtered back-projection. We believe that this kind of learning approach can be extended to any common CT artifact compensation heuristic and will outperform hand-crafted artifact correction methods in the future.

1 Introduction

X-ray computed tomography scanning has become a standard procedure in diagnosis of certain diseases or trauma. In virtually every scanning system heuristic steps exist to compensate for artifacts from scatter, beam-hardening, or other sources of artifacts. A simple heuristic approach for compensation of limited angle artifacts is shown in the work of Riess et al. [9]. The method introduces heuristic compensation weights for the filtered back-projection (FBP) algorithm that reduce the loss in mass from the reduced number of views. While demonstrating significant improvements in image quality, the approach lacks any theoretical guarantees for the weighting procedure. The approach is one amongst many “hand-crafted” artifact reduction methods.

Neural networks have been employed for reconstruction. Argyrou et al. [1] already show an approach of learning 2-D reconstruction with a zero hidden layer neural network. The main disadvantage of their approach is the large number of synapses required. Thus, this method requires an impractical amount of memory and training examples and an extension to 3-D seems impossible.

Recent developments in deep learning also suggest that deeper architectures should be explored and regularization is necessary for training such networks. A method presented by Cierniak [2] uses a back-projection then filtering approach and employs a Hopfield neural network to solve the image deblurring problem.

This approach bypasses the problem of too many parameters by fixing them in the back-projection step and degenerates the reconstruction problem to an image-based filtering approach.

A different approach by de Medeiros et al. [6] exploits the sparsity of the back-projection matrix to fit it into memory. However, this approach is not able to involve any training since the sparse structure would be undone in the training step.

In MRI-reconstruction Hammernik et al. [5] have shown an approach of learning sparsifying transforms and potential functions.

In this paper, we propose a fully differentiable back-projection layer for parallel-beam and fan-beam projection as well as a weighting layer. This enables various neural network architectures for reconstruction to be trained end-to-end. The distinctive advantage of our approach is that we are also able to learn heuristics that are applied in the projection domain *before* back-projection. This way we can replace heuristic correction methods with learned optimal solutions. Also we can jointly optimize different correction methods to improve their results when applied simultaneously. We present an example where we jointly optimize compensation weights, the reconstruction filter and a non-linear image filtering algorithm. In addition we propose a method for regularizing the optimization by pre-training. We evaluate our method using realistic data.

2 Methodology

We first describe the mapping of a filtered back-projection algorithm to a basic neural network architecture for reconstruction in Sect. 2.1. For this architecture we derive the forward and backward computation of our novel back-projection layer, cf. Sect. 2.2). In Sect. 2.3 we extend the architecture to fan-beam reconstruction by deriving a novel weighting layer that implements the special topology of this operation, and a fan-beam layer (Sect. 2.3). Eventually, we evaluate these architectures in Sect. 2.5).

2.1 Mapping FBP to Neural Networks

The input-vector \mathbf{x} to the neural network corresponds to the whole sinogram, while the output-vector \mathbf{y} corresponds to the whole reconstruction. As loss function every regression loss function is applicable, e.g., the l_2 norm: $\|\mathbf{x} - \mathbf{y}\|_2$. The filtering is a convolution with a high pass filter and can directly be mapped to a convolutional layer. Typically, convolutional layers in neural networks use comparably small filter kernels which are calculated in spatial domain. In comparison, a convolution in the Fourier domain is advantageous for reconstruction because (i) the high pass filters have infinite support, thus, they are as large as the number of detector pixels, and (ii) reduce computational complexity. Rectified linear units (ReLU) as non-linear activation functions can enforce the constraint of non-negativity of the reconstruction. As last step back-projection has to be

mapped to a layer in a neural network. We begin with the discrete formulation of FBP:

$$f(u, v) \approx \frac{\pi}{N} \sum_{n=1}^N q(u \cos(\theta_n) + v \sin(\theta_n), \theta_n), \quad (1)$$

where $f(u, v)$ denotes the function to be reconstructed, s is a position on the detector, N denotes the number of projections and $q(s, \theta_n)$ denotes the filtered projections. Since we sample the function $q(s, \theta_n)$ only at discrete positions of s (denoted as $q_{m,n}$), a one-dimensional interpolation has to be performed, i. e.,:

$$f(u, v) \approx \frac{\pi}{N} \sum_{n=1}^N \sum_{m=1}^M w_m(u, v, \theta_n) \cdot q_{\lceil u \cos(\theta_n) + v \sin(\theta_n) - \frac{M+2}{2} + m \rceil, n}, \quad (2)$$

where $w_m(u, v, \theta_n)$ are the interpolation weights and M is an even integer denoting the number of interpolation coefficients.

A well known activation model of a neuron is [3]:

$$f(y_i) = f\left(\sum_{j=1}^N w_{ij}x_j + w_{j0}\right). \quad (3)$$

When we set our activation function to be the identity $f(x) = x$ and all bias weights w_{j0} to zero it follows $f(y_i) = \sum_{j=1}^N w_{ij}x_j$. Let us change the indexation of $f(y_i)$ to $f(x_i, y_j)$ which denotes a pixel of a reconstruction of Size $I \times J$. Similarly we change the indexation of \mathbf{x} to that of the filtered sinogram $q_{m,n}$:

$$f(x_i, y_j) = \sum_{n=1}^N \sum_{m=1}^M w_{i+(j-1) \cdot I, m+(n-1) \cdot M} \cdot q_{m,n}. \quad (4)$$

We can compute Eq. (2) only at some discrete u, v positions and choose, without loss of generality, the interpolation size big enough to cover the length of the detector by zero-padding the signal as needed.

$$f(u_i, v_j) \approx \frac{\pi}{N} \sum_{n=1}^N \sum_{m=1}^M w_m(u_i, v_j, \theta_n) \cdot q_{m,n} \quad (5)$$

Equation (5) is equivalent to Eq. (4) if we choose:

$$w_{i+j \cdot I, m+(n-1) \cdot M} = \frac{\pi}{N} w_m(u_i, v_j, \theta_n). \quad (6)$$

This general result holds for arbitrary interpolation techniques. The most important case being the linear interpolation which will yield only up to two non-zero coefficients for every M interpolation coefficients resulting in an extremely sparse matrix.

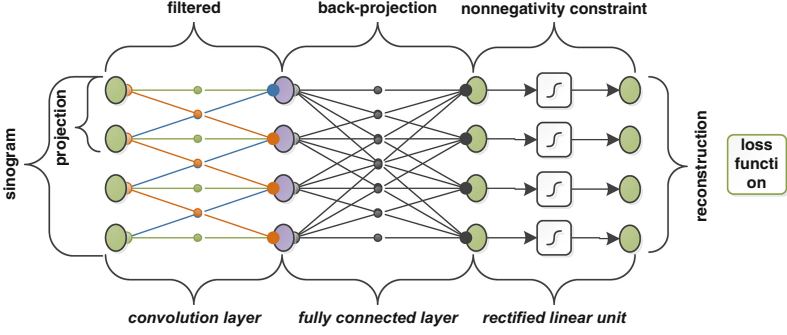


Fig. 1. Parallel-beam neural network architecture

Plugging in our convolution layer, the output for our neural network architecture is finally calculated as:

$$f(x_i, y_j) = \max \left[0, \sum_{n=1}^N \sum_{m=1}^M \frac{\pi}{N} w_m(u_i, v_j, \theta_n) \cdot \left(\sum_{k=-M/2}^{M/2} w_k \cdot p_{m-k,n} \right) \right]. \quad (7)$$

This shows that our neural network architecture implements a filtered back-projection algorithm. Note that the network's weights for initialization are known from the original derivation of the filtered back-projection. Figure 1 shows this basic architecture for parallel-beam reconstruction. We mapped each step of the FBP algorithm into a corresponding layer.

2.2 Parallel-Beam Back-Projection Layer

To solve the problem of fitting the parameters of the FCL, which represents the back-projection operator, into memory we propose a novel back-projection layer. This layer has no adjustable parameters. During the forward-pass of the network the coefficients $w_{i,j}$ of the matrix \mathbf{W}_l are computed, where l denotes the index of this layer. This is calculated incrementally using the update rule

$$\mathbf{y}_l = \mathbf{W}_l \mathbf{y}_{l-1} \quad (8)$$

similar to the traditional back-projector in FBP. Since this layer has no adjustable weights in its backward pass only the gradient with respect to the lower layers has to be calculated. For FCLs this corresponds to calculating:

$$\mathbf{E}_{l-1} = \mathbf{W}_l^T \mathbf{E}_l, \quad (9)$$

where \mathbf{E}_{l-1} represents the error of the next lower layer of the network. Since the back-projection operator is the transpose of the projection operator [11], one algorithm for the backward-pass is readily available. An alternative way is to recalculate the weights as in the forward-pass and apply them to the error.

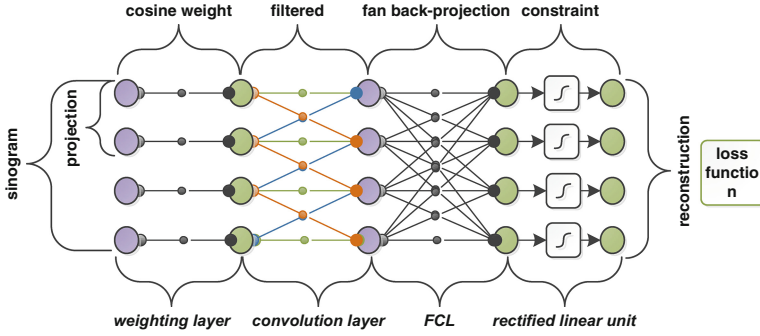


Fig. 2. Fan-beam neural network architecture

2.3 Extension to Fan-Beam Reconstruction

The extension of the well known FBP algorithm to the fan-beam projection model consists of a cosine weighting of the projections and weighted back-projection. Figure 2 shows this extended architecture for fan-beam reconstruction.

Weighting Layer. A layer that performs only weighing has the very sparse structure of a diagonal matrix. This can be exploited to construct a special weighting layer that enforces sparsity. This layer has exactly N training parameters, where N is the dimensionality of the input and output vectors. The forward-pass can be calculated using Eq. (8) which corresponds to an element-wise multiplication of the input with the weights. For the backward-pass we employ Eq. (9). Since $\mathbf{W}^T = \mathbf{W}$ holds for diagonal matrices, the backward-pass of this layer is an element-wise multiplication of the weights with the error of the next higher layer. The gradient with respect to the weights of a FCL is calculated by $\mathbf{G}_l = \mathbf{E}_l \mathbf{y}_{l-1}$.

Fan-Beam Back-Projection Layer. The derivation of the forward and backward pass of a fan-beam layer is identical to the parallel-beam layer. The weights are calculated differently and a distance weighting is applied to every element of the sum [11]. But the backward pass can also be implemented as fan-beam projection.

2.4 Convergence and Overfitting

The advances in deep learning have shown that regularization is crucial to the performance of neural networks. Because of the high dimensionality of our reconstruction networks, regularization is important to achieve convergence and to prevent overfitting. Popular methods are weight-decay, dropout [10] and pre-training. Weight-decay is not effective for this learning problem because the

scaling of the data is crucial when using the l_2 norm as a loss function. Dropout is normally very effective for fully connected layers because it can prevent co-adaptation of features. However for this large scale regression problem, co-adaptation is required and all weights depend upon each other. Pre-training can be applied directly using knowledge of existing FBP algorithms. Discretized solutions are known for all layers: the convolutional layer uses the ramp filter, and the weighting layer accounts for cosine-weighting or redundancy weighting. These initializations can be used for a very effective pre-training.

2.5 Experiments

We present two applications of our proposed neural network architecture. For both we use slices of reconstructions of real patient data of size 512×512 . These slices are downsized by a factor of 0.7 and embedded into a zero image of the original size to prevent truncation artifacts. We perform a ten-fold cross validation on 2378 slices of ten different patients.

The presented layers have been implemented with GPU-acceleration using the Caffe Framework. We also implemented the weight initializations used as pre-training as so called weight-fillers in the framework. The implementation will be released upon publication of the paper.

Limited Angle Parallel-Beam Reconstruction. In our first experiment, we explore improvements of parallel-beam architectures for limited angle reconstruction with five degree of missing data. We use 175 projections with an angular increment of one degree. The training target is the original image, which was used to simulate the projections. Our neural network architecture is the presented basic architecture for parallel beam reconstruction, with an additional weighting layer between the filtering and the back-projection operation. Since the data is incomplete, this problem can only be solved approximately without any regularizing assumptions. Thus, we place a *maxout* [4] filtering cascade on top of the network to introduce a non-linear filtering. We change the reconstruction filter to a 2D filter with a kernel size of 5. The values of the reconstruction kernel outside the third column are initialized to zero. The weighting layer is initialized with every value set to one, while the reconstruction filter is set to the well known Ramachandran-Lakshminarayan filter [8] discretization. The maxout network is initialized with Gaussian distributed random values.

Limited Angle Fan-Beam Reconstruction. In the second experiment, we learn compensation weights for a limited-angle problem with 180 projections up to 180 degree. We use our basic architecture for fan-beam reconstruction. The weights of the weighting layer are initialized with the well known Parker weights [7] multiplied with cosine weights. The reconstruction filter is set to the Ramachandran-Lakshminarayan filter discretization. As training target we employ a reconstruction with 360 projections of full scan data. During the training, only the weights receive a non-zero learning rate.

3 Evaluation

We found experimentally that the different layers require individual learning rates. The reconstruction filter was found to be extremely sensitive. The weighting-layer before the back-projection layer generally tolerates large learning rates.

For the parallel-beam problem we found Caffe’s “RMSPROP” solver with the decay of 0.02 effective. The global learning rate was set to 10^{-6} while the reconstruction filter’s learning rate had to be set to 10^{-15} to prevent divergence. For this problem online training turned out to be more effective than mini-batch or batch learning. The relative root-mean-square error, averaged over the test sets, drops from $6.78\text{e-}03\%$ for the classical FBP to only $3.54\text{e-}3\%$. Hereby, our method outperforms the conventional FBP with enforcing the non-negativity constraint for every case by a factor of nearly two.

For the fan-beam experiment, we used Caffe’s “ADAGRAD” solver and online training. We could set our learning rate for the weighting layer to $2 \cdot 10^2$ without divergent behaviour. The relative root-mean-square error averaged over the test sets dropped from $5.31\text{e-}03\%$ for FBP to $3.92\text{e-}03\%$ for our method.

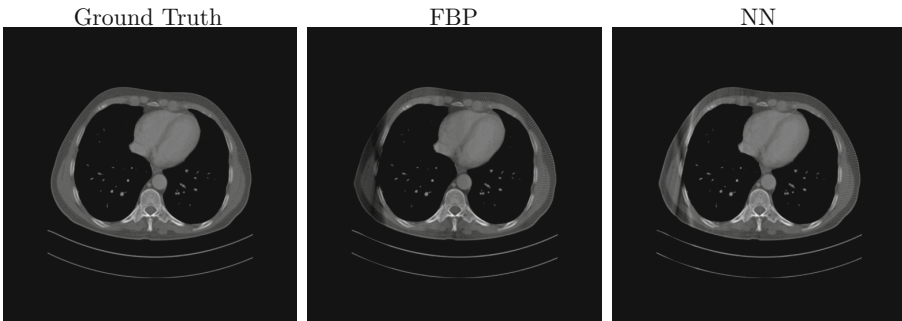


Fig. 3. Reconstruction results using 360° , 180° FBP, and 180° NN.

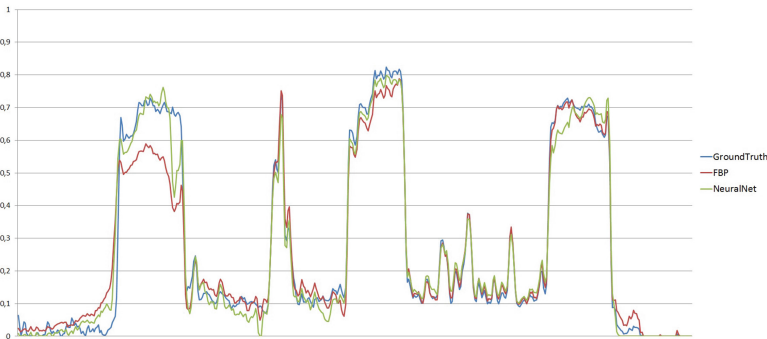


Fig. 4. Cross sections through the images of Fig. 3

Figures 3 and 4 show that the loss of mass could be corrected well, despite the architecture being equivalent to FBP.

4 Conclusion

We propose to use deep learning techniques to replace heuristic compensation steps in CT reconstruction. This enables various new architectures which can account for many artifact types. Evaluations of the reconstruction results show improved image quality compared to FBP while still retaining the same computational demands. We could successfully learn compensation layers for limited-angle tomography. Presumably, most artifact compensation methods in CT can be mapped to convolutional neural networks which will also enable to learn scatter compensation and beam-hardening.

Acknowledgments. The authors would like to thank Dr. Cynthia McCollough, the Mayo Clinic, the American Association of Physicists in Medicine funded by grants EB017095 and EB017185 from the National Institute of Biomedical Imaging and Bioengineering for providing the used data.

References

1. Argyrou, M., Maintas, D., Tsoumpas, C., Stiliaris, E.S.: Tomographic image reconstruction based on artificial neural network (ANN) techniques. In: Proceedings of 2012 IEEE Nuclear Science Symposium and Medical Imaging Conference Record (NSS/MIC), pp. 3324–3327. IEEE (2012)
2. Cierniak, R.: A new approach to image reconstruction from projections using a recurrent neural network. *Int. J. Appl. Math. Comput. Sci.* **18**, 147–157 (2008). AMCS
3. Duda, R.O., Hart, P.E., Stork, D.G. (eds.): *Pattern Classification*. Wiley, New York (2000)
4. Goodfellow, I.J., Warde-farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. In: Proceedings of The 30th International Conference on Machine Learning, vol. 28, pp. 1319–1327 (2013)
5. Hammernik, K., Knoll, F., Sodickson, D., Pock, T.: Learning a variational model for compressed sensing mri reconstruction. In: Proceedings of the International Society of Magnetic Resonance in Medicine (ISMRM) (2016)
6. de Medeiros, L.F., da Silva, H.P., Ribeiro, E.P.: Tomographic image reconstruction of fan-beam projections with equidistant detectors using partially connected neural networks. In: *Learning and Nonlinear Models - Revista da Sociedade Brasileira de Redes Neurais*, vol. 1, pp. 122–130. Sociedade Brasileira de Redes Neurais (2003)
7. Parker, D.L.: Optimal short scan convolution reconstruction for fan beam CT. *Med. Phys.* **9**(2), 254–257 (1982)
8. Ramachandran, G.N., Lakshminarayanan, A.V.: Three-dimensional reconstruction from radiographs and electron micrographs: application of convolutions instead of fourier transforms. *Proc. Nat. Acad. Sci.* **68**(9), 2236–2240 (1971)

9. Riess, C., Berger, M., Wu, H., Manhart, M., Fahrig, R., Maier, A.: TV or not TV? that is the question. In: Leahy, R.M., Qi, J. (eds.) *Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, pp. 341–344, June 2013
10. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014)
11. Zeng, G.L. (ed.): *Medical Image Reconstruction*. Springer, New York (2009)