FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

Deep Learning for Image Reconstruction

Wenyu Zhang

Overview

- AUTOMAP
- CT Image Reconstruction
- Variational Network

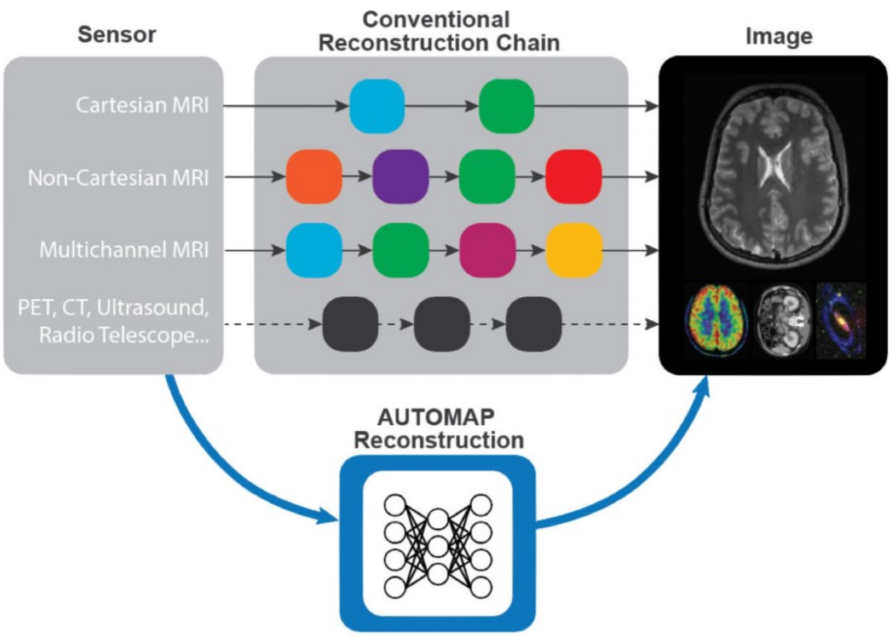
Overview

- **AUTOMAP**
- CT Image Reconstruction
- Variational Network

AUTOMAP

What is Automated Transform by Manifold Approximation?

AUTOMAP recasts image reconstruction as a data-driven, supervised learning task implemented with a deep neural network that allows a mapping between sensor and image domain.



AUTOMAP

Layers:

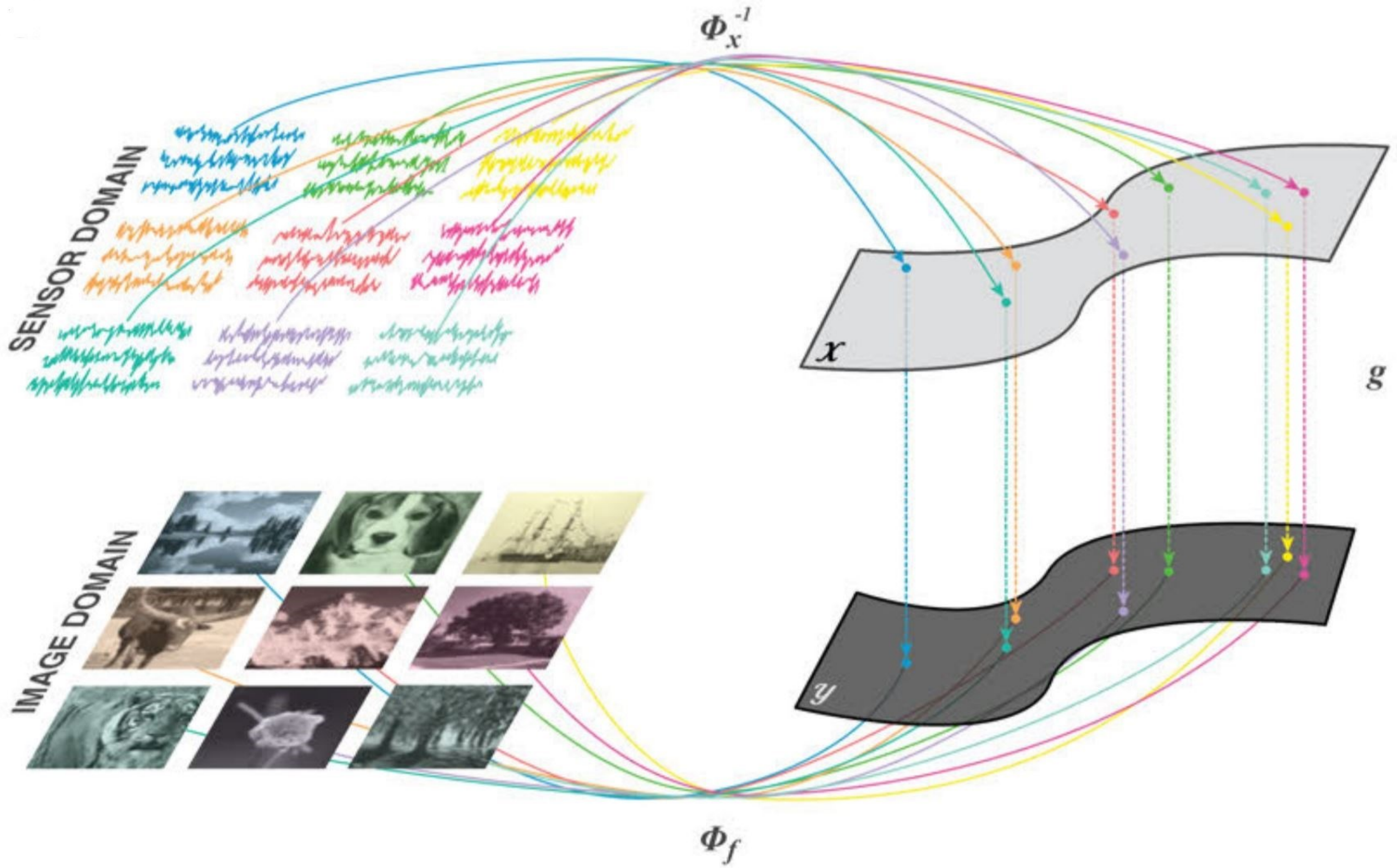
- The fully-connected layers approximate the between-manifold projection from the **sensor domain** to the **image domain**.
- The convolutional layers extract high-level features from the data and force the image to be represented **sparsely**

A composite transformation:

$$f(x) = \phi_y \circ g \circ \phi_x^{-1}(x)$$

- ϕ_x^{-1} is an inversed transform original encoding signals from sensor domain to the decoded domain
- g is projection from manifold of decoded inputs x to manifold of output images y
- ϕ_y decompress the images from manifold g back to the representation in euclidean space.

AUTOMAP



AUTOMAP

$\tilde{x}$ is noisy observation, x is sensor domain inputs. First task is to learn the stochastic projection operator onto mainifold $\mathcal{X}$:

$$P(\tilde{x}) = P(x \mid \tilde{x})$$

After obtaining x, second task is to obtain the reconstruct function f(x) :

$$\min \quad L(\tilde{f}(x), \quad f(x))$$

Denote the data as $(y_i, x_i)_{i=1}^{n}$, x_i is i-th obervation indicates nxn paramters, y_i indicates nxn underlying images. Assum that:

- Smooth and homeomorphic function f: $y = f(x)$ exists
- Manifolds $\mathcal{X}$ and $\mathcal{Y}$are embedded in the ambient space $\Re^{n^2}$

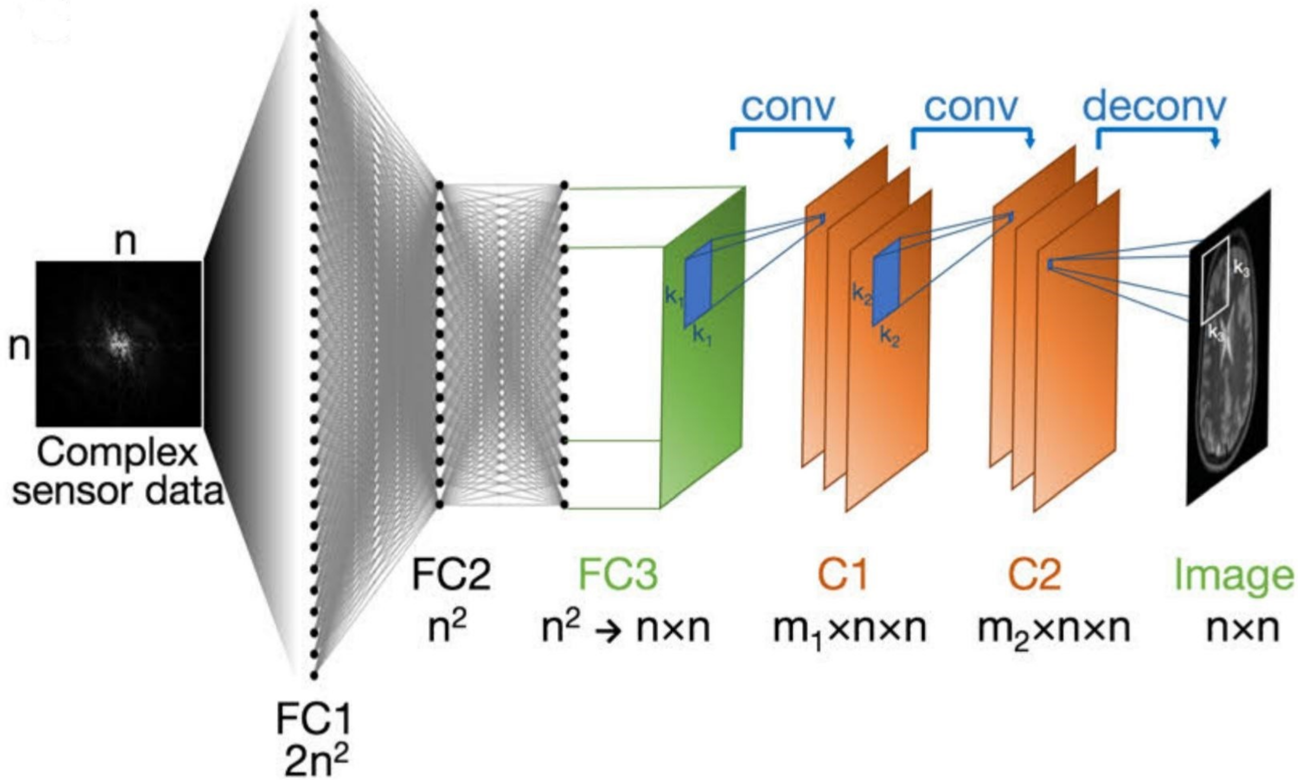
Joint manifold $M_{x,y} = \mathcal{X} \times \mathcal{Y}$, that the $(y_i, x_i)_{i=1}^{n}$ lies in.
Exist a mapping $\phi = (\phi_x, \phi_y)$, between (x,f(x)) and (z,g(z))
$(x = \phi_x(z), f(x) = \phi_y \circ g(z))$

$$f(x) = \phi_y \circ g \circ \phi_x^{-1}(x)$$

Model Architecture

- The input to the neural network consists of a vector of sensor domain sampled data produced by the preprocessing
- Complex data must be separated into real and imaginary components concatenated in the input vector(nxn-2n^2x1)
- The input layer FC1 is fully connected to an n^2 x 1 dimensionality FC2 and activated by the tanh
- Fully connected to another n^2 × 1 dimensionality hidden layer FC3
- C1,C2 convolve 64 filters of 5 × 5 with stride 1 followed by a rectifier nonlinearity
- The the final output layer deconvolves the C2 layer with 64 filters of 7 × 7 with stride 1, reconstruct the magnitude image

Model Architecture



Training Details

Data:

- ImageNet
- Human Connectome Project (HCP)
- Random value noise

Sensor-domain encoding:

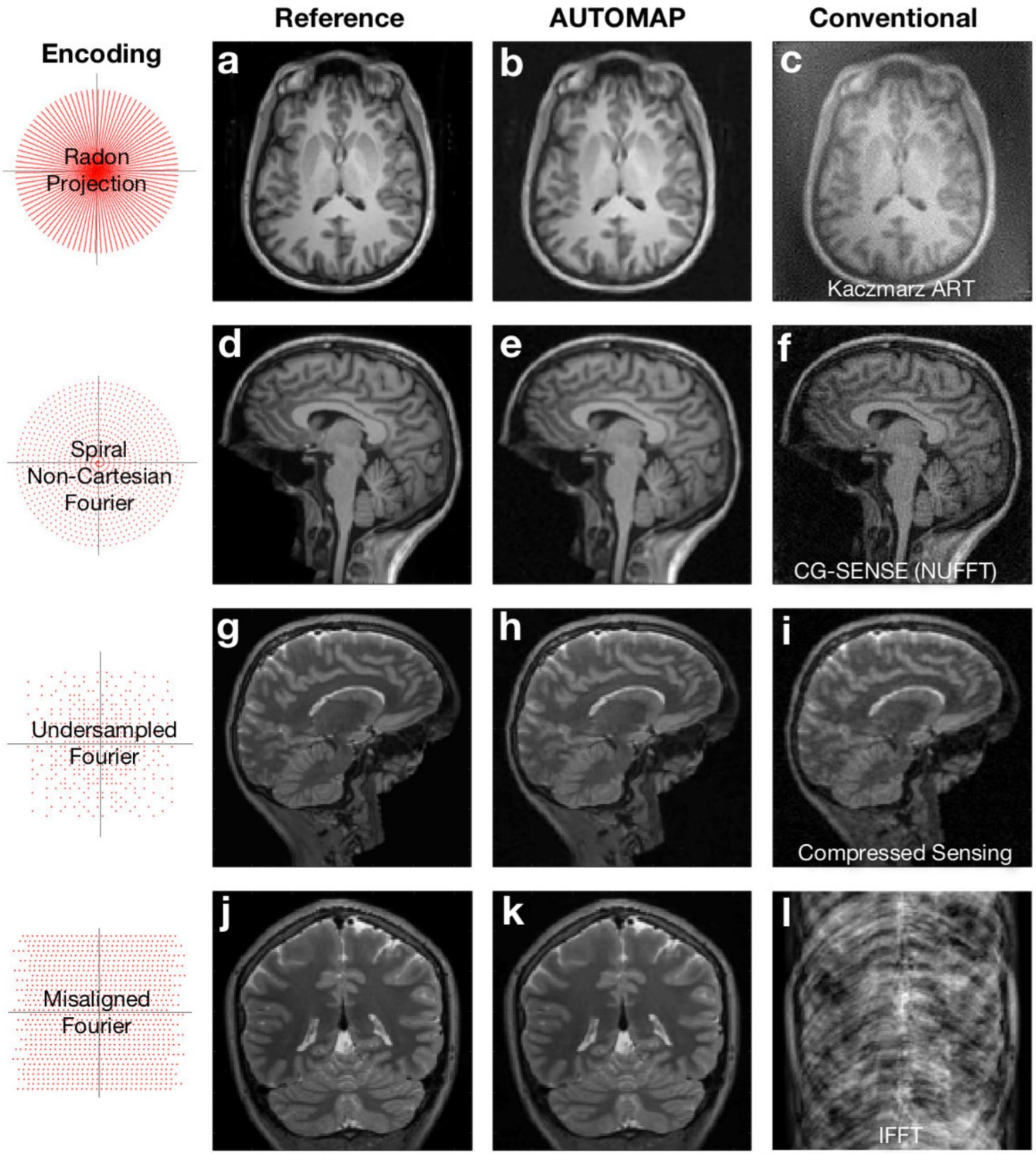
- Discrete Radon Transform with 180 projection angles and 185 parallel rays
- Nonuniform Fast Fourier Transform (NUFFT) was used the Spiral k-space experiment
- Undersampled Cartesian k-space experiment used a Poisson-disc sampling pattern
- The misaligned Cartesian k-space experiment used Fourier Transformed

Training Details

Training the network:

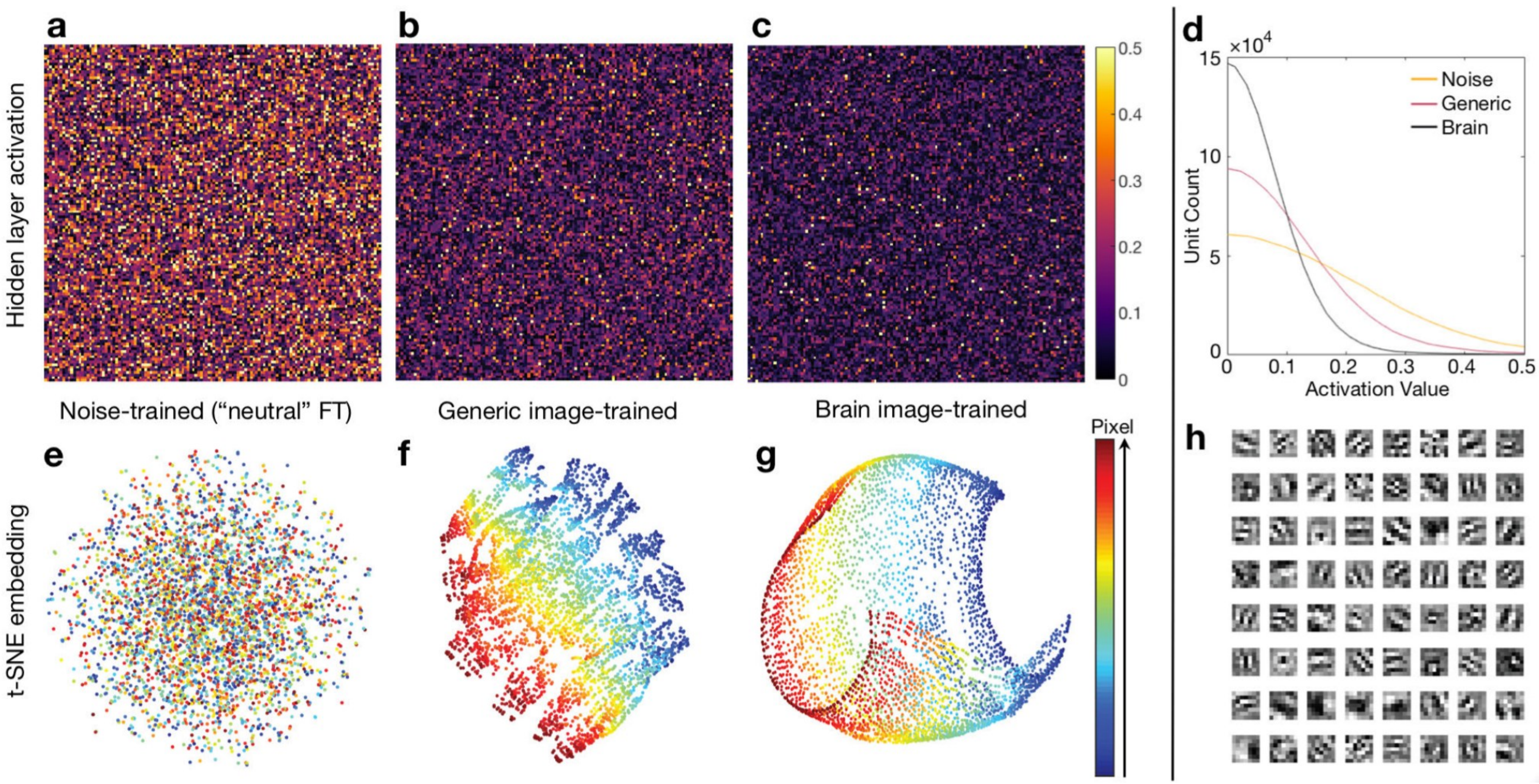
- Multiplicative noise to the inputs to force the network to learn robust presentation
- Squared loss
- L1 norm penalty applied to the feature map activations in the final hidden layer C2
- Gaussian noise that was only applied during evaluation

Resul



Reference
AUTOMAP
Conventional
Encoding
Radon Projection
a
b
c
Kaczmarz ART
Spiral Non-Cartesian Fourier
d
e
f
CG-SENSE (NUFFT)
Undersampled Fourier
g
h
i
Compressed Sensing
Misaligned Fourier
j
k
l
IFFT

Hidden-layer Activity



t-SNE: https://lvdmaaten.github.io/tsne/

Learn Reconstruction of Image Phase

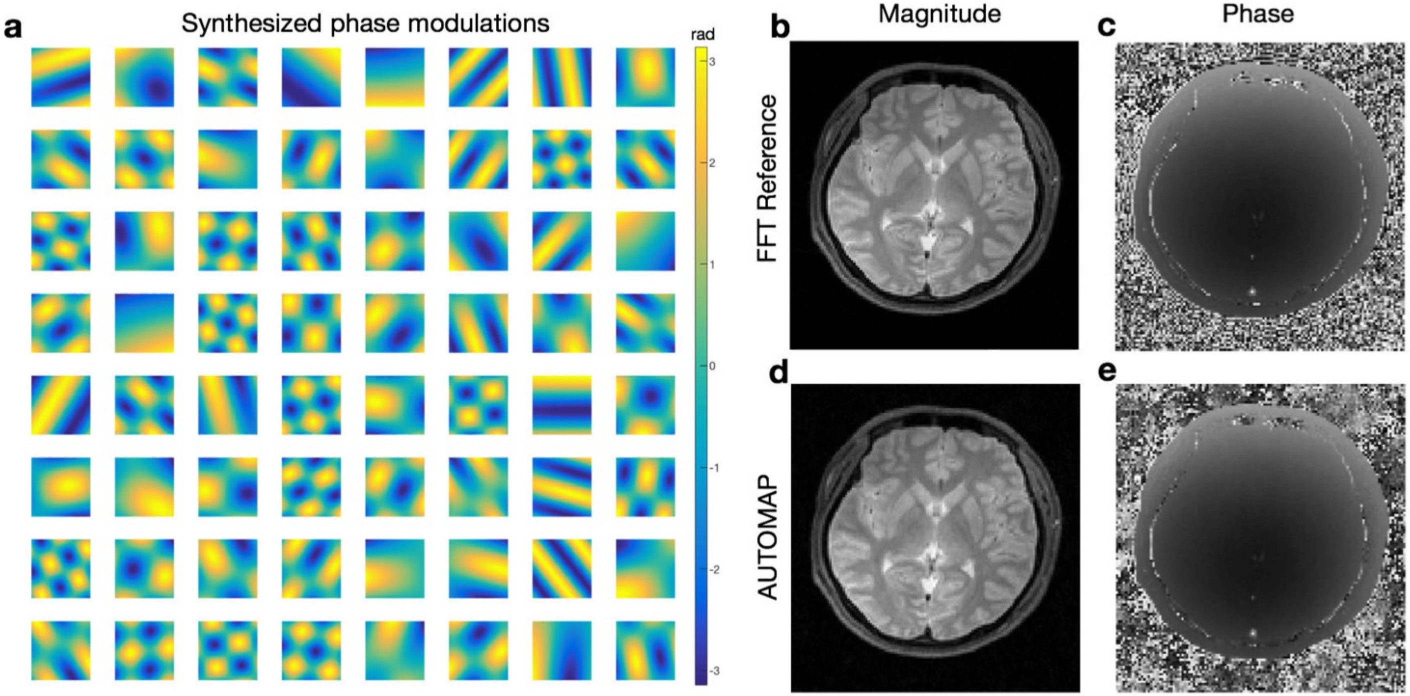


Figure 4. Learning reconstruction of phase for *in vivo* data. The inclusion of synthetic phase to the training dataset enables AUTOMAP to properly reconstruct both the magnitude and phase. **a,** The magnitude-only Human Connectome Project (HCP) *k*-space data was phase-modulated by two-dimensional sinusoids of varying spatial frequencies to generate the training dataset. After training, the magnitude (**b**) and phase (**c**) of a test T2-weighted *k*-space dataset are properly reconstructed by AUTOMAP (**d, e**).

Overview

- AUTOMAP
- **CT Image Reconstruction**
- Variational Network

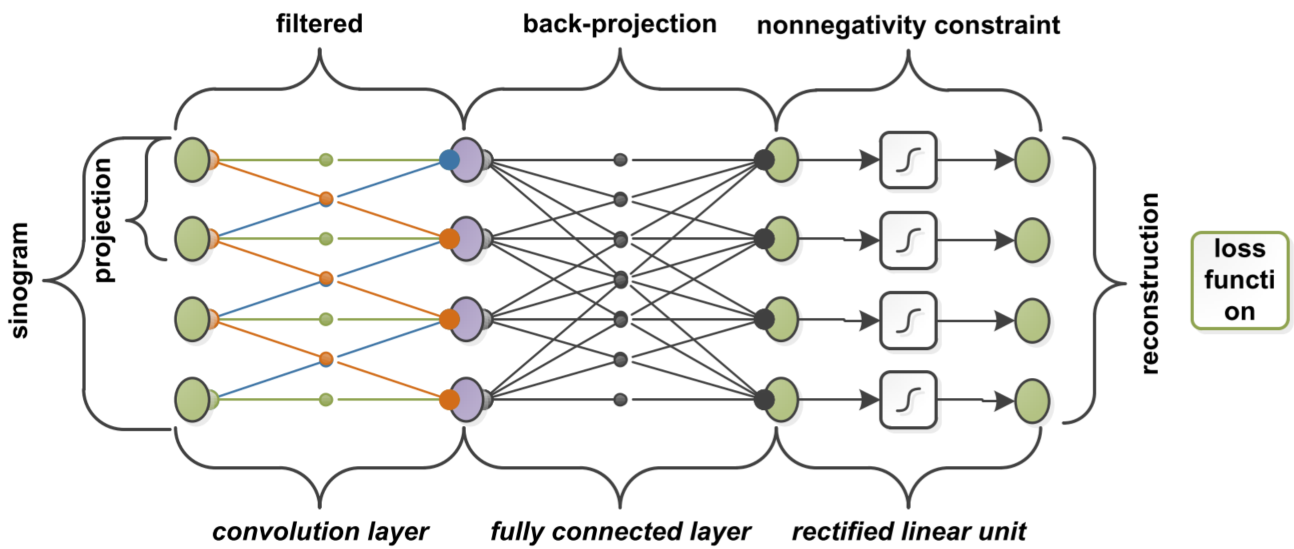
Filtered Back-projection

$$f(u,v) = \int_0^{\pi} q(s,\theta) * h(s) d\theta \Big|_{s = u\cos\theta + v\sin\theta}$$

Mapping there steps to neural network:

- High pass filter —— convolution layer
- Backprojection along θ —— fully connected layer
- Suppress negative values —— non-negative constrain (ReLU)
- Loss function: e.g. l2 norm: $\|x - y\|_2$

Parallel-Beam Neural Network Architecture



Mapping FBP to Neural Networks

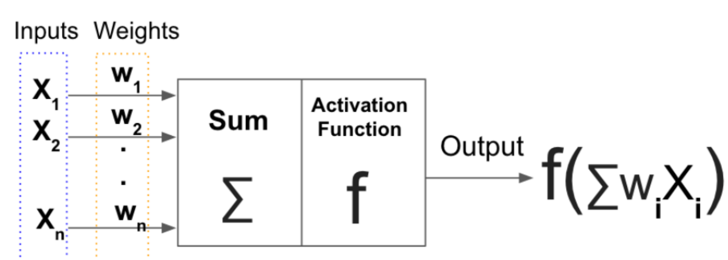
$$f(u,v) \approx \frac{\pi}{N} \sum_{n=1}^{N} q(u\cos(\theta_n) + v\sin(\theta_n), \theta_n)$$

and one-dimensional interpolation:

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

A well known activation model of a neuron is:

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



A Practical Introduction to Deep Learning with Caffe and Python, Adil Moujahid

Mapping FBP to Neural Networks

$f(x_i, y_j)$ denotes a pixel of a reconstruction of Size $I \times J$,

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

then we could compare the function before:

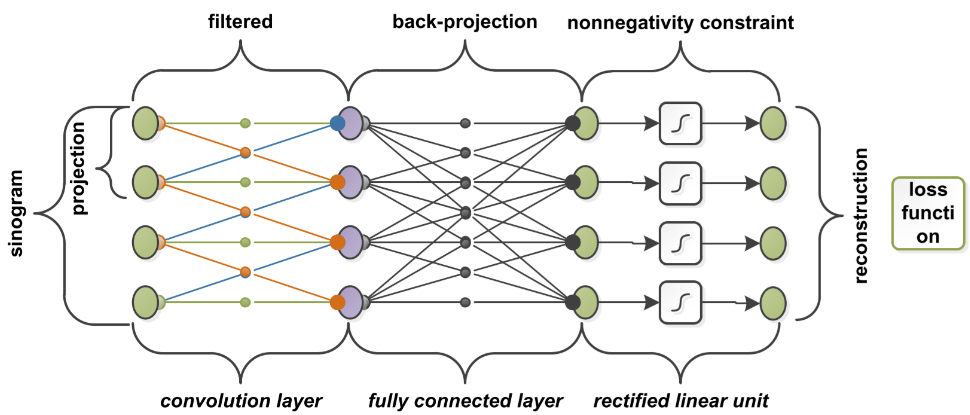
$$f(u_i, v_j) \approx \frac{\pi}{N} \sum_{n=1}^{N} \sum_{m=1}^{M} w_m(u_i, v_j, \theta_n) \cdot q_{m,n}$$

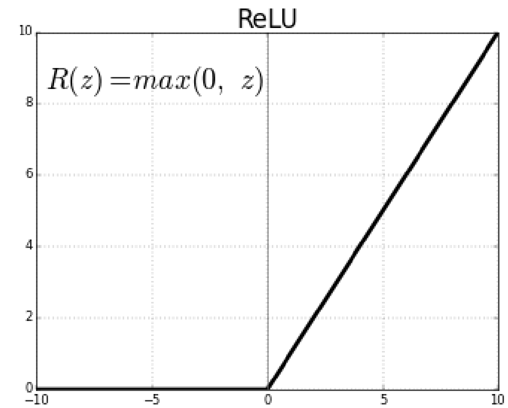
they are equivalent if:

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

Mapping FBP to Neural Networks

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





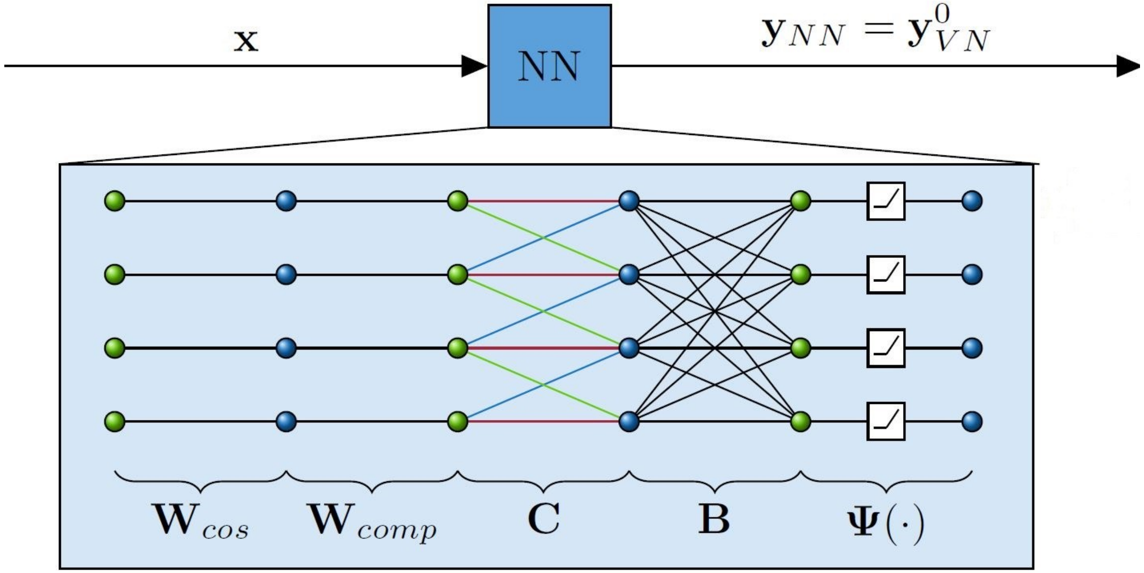
https://medium.com/@kanchansarkar/relu-not-a-differentiable-function-why-used-in-gradient-based-optimization-7fef3a4cecec

Parallel-Beam Back-Projection Layer

To solve the parameters in the fully connected layer:

- During the forward-pass, the coefficients are computed,and update: $y_l = W_l y_{l-1}$
- Backward-pass: $E_{l-1} = W_l^T E_l$

Fan-beam Neural Network Architecture



Fan-beam Neural Network Architecture

$$y_{NN} = \Psi(BCW_{comp} W_{\cos} x)$$

- B denotes the backprojection operator
- C implements filtering with one-dimensioal convolution kernel
- $W_{\cos}$ is weighting operators
- W_{comp} is compensation weights
- The non-negativity constraint is realized via operator Ψ

Fan-beam Neural Network Architecture

Main parts of the architecture:

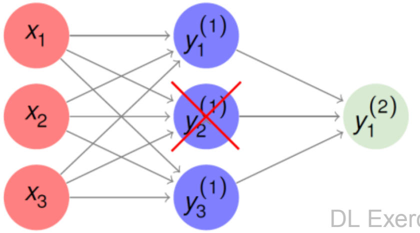
- Weighting Layer:
  W is sparse structure of a diagonal matrix;
  Forward: element-wise multiplication of the input with the weights;
  Backward: element-wise multiplication of the weights with the error.

- Fan-Beam Back-Projection Layer :
  identical to the parallel-beam FCL

Convergence and Overfitting

Regularization is important to achieve convergence and to prevent overfitting:

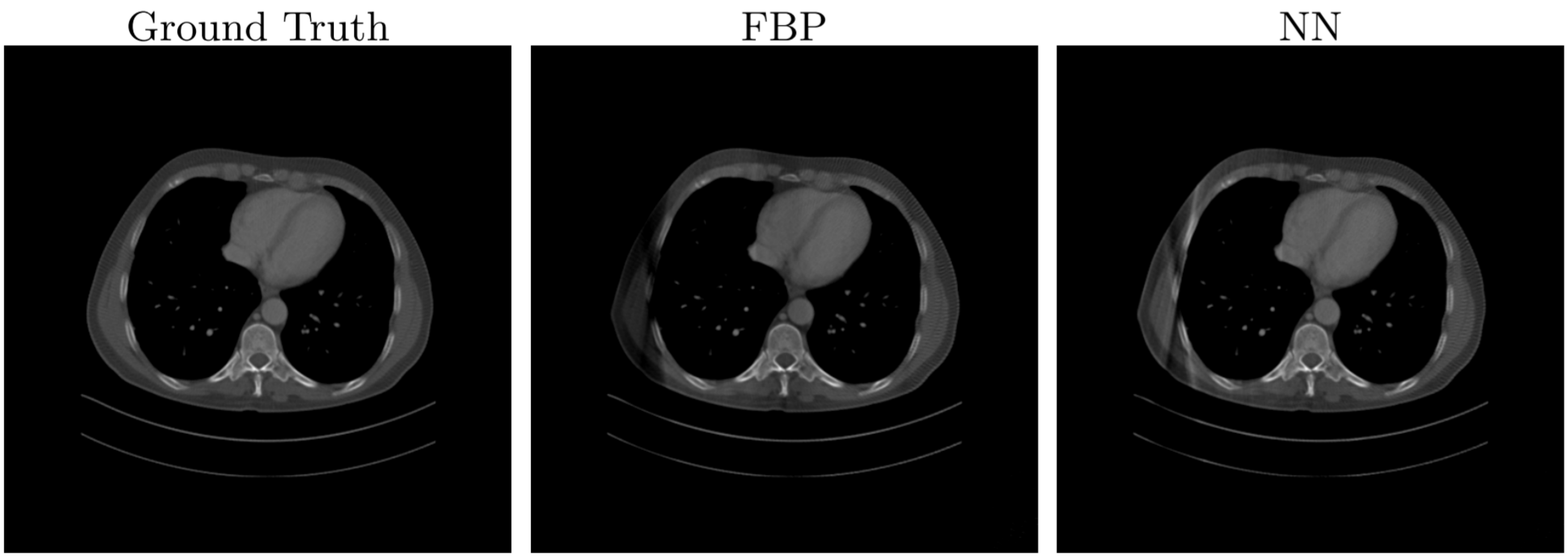
- Dropout: individual nodes are either "dropped out" of the net with probability 1 − p or kept with probability p



DL Exercise 2: Regularization

- **Pre-training** can be applied directly using knowledge of existing FBP algorithms.
  e.g. The convolutional layer uses the ramp filter.

Reconstruction Results

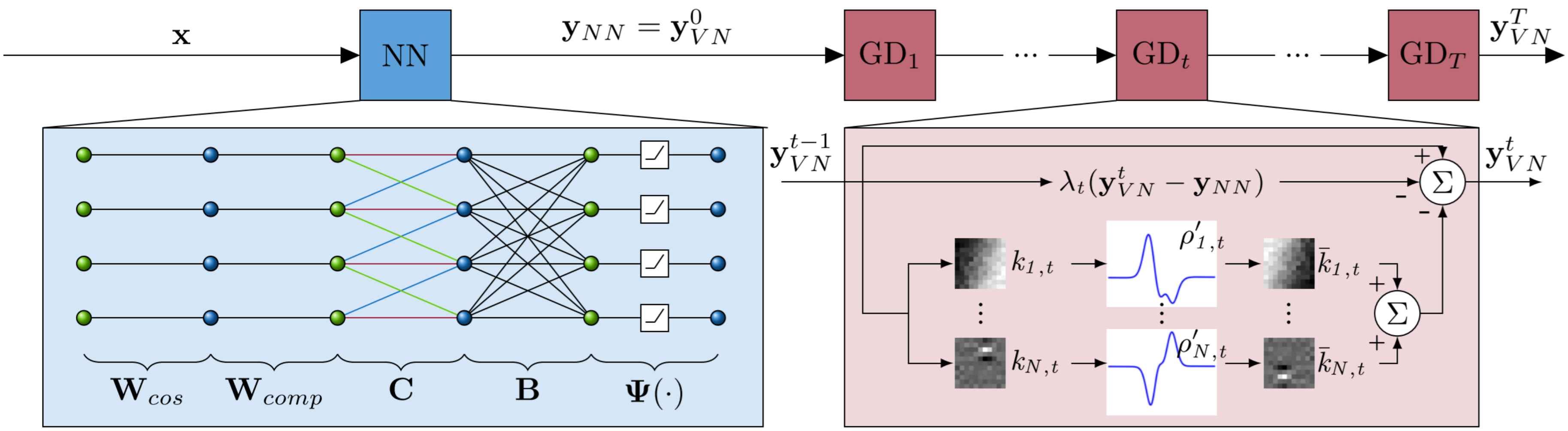


Reconstruction results using 360◦, 180◦ FBP, and 180◦ NN

Overview

- AUTOMAP
- CT Image Reconstruction
- **Variational Network**

Variational Network



Variational Network

To remove the streaking artifacts, the VN formulates non-linear filtering. In each step t, these parameters are learned:

- filters $k_{i,t}$
- derivative of potential functions $\mathrm{p}_{i,t}$
- the regularization parameter λ_t

Variational Network

Formulating a network for non-linear filtering as a fixed number of T unrolled gradient descent steps:

$$y_{VN}^{t} = y_{VN}^{t-1} - g^{t}\left(y_{VN}^{t-1}\right)$$

$$g^{t}\left(y_{VN}^{t-1}\right) = \nabla_{y} E(y)\Big|_{y = y_{VN}^{t-1}}$$

The variational image restoration problem is given as,

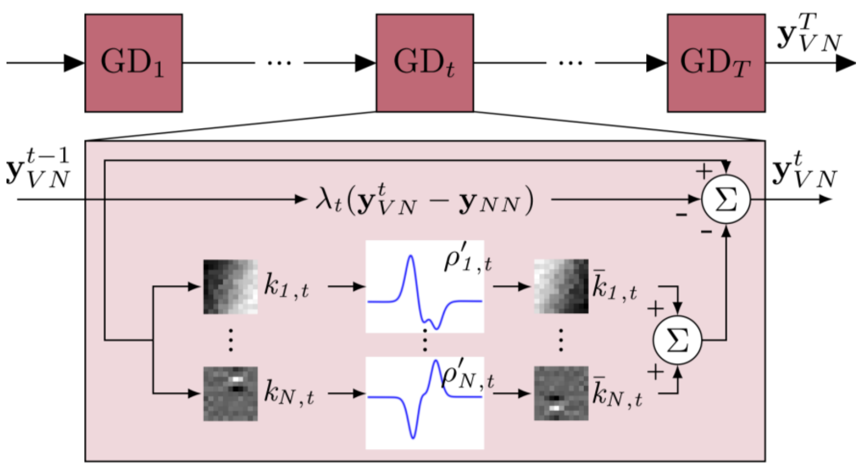
$$\mathrm{E(y)} = \frac{\lambda}{2}\left\| y_{VN} - y_{NN} \right\|_2^2 + \sum_{i=1}^{N_k} p_i (K_i y_{VN})$$

Variational Network

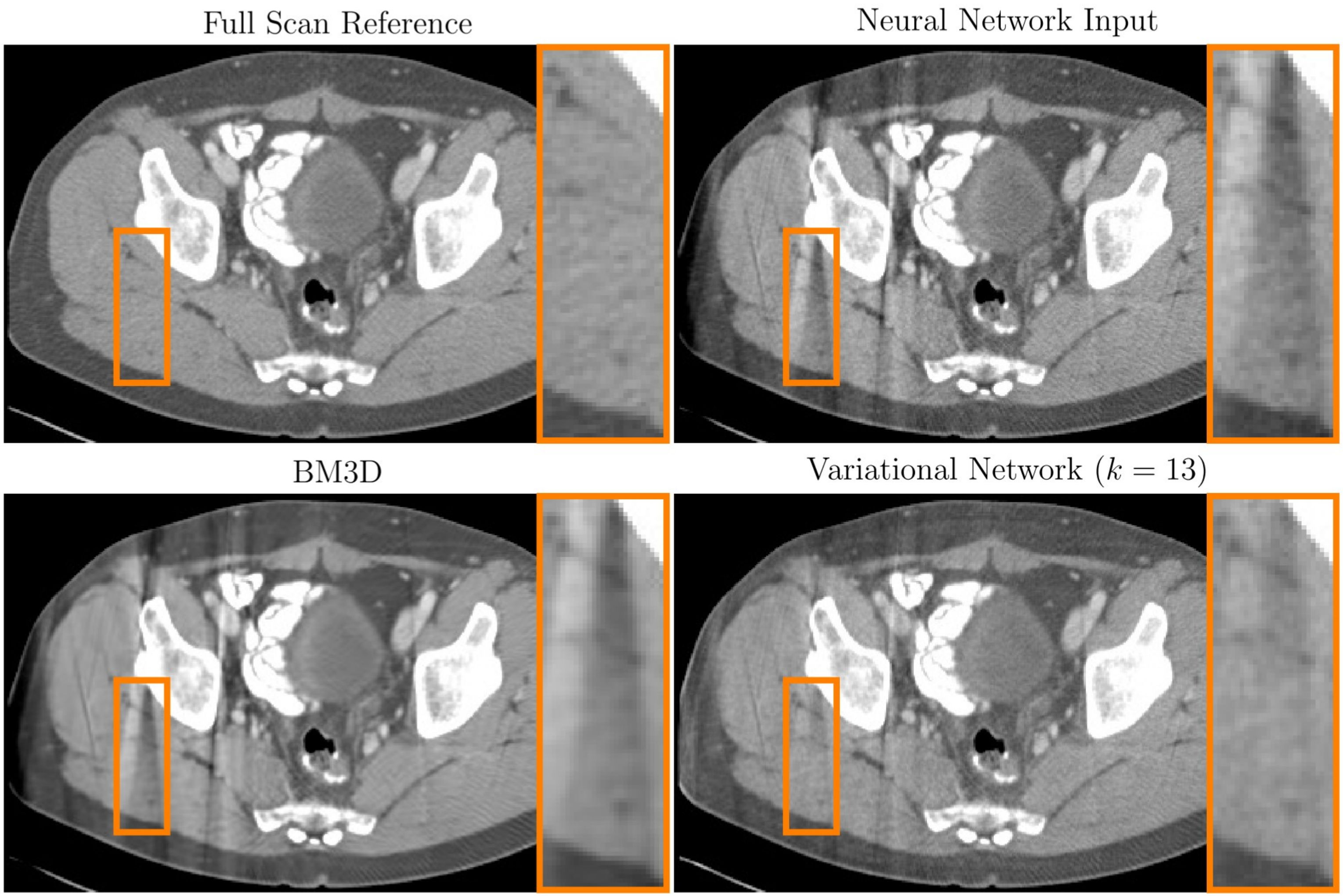
$$y_{VN}^{t} = y_{VN}^{t-1} - \sum_{i=1}^{N_k} K_{i,t}^{T} p'_{i,t}(K_{i,t} y_{VN}^{t-1}) - \lambda_t (y_{VN}^{t-1} - y_{NN})$$

Then the loss function is the minimization of the mean-squared error (MSE):

$$L = \frac{1}{2S} \sum_{s=1}^{S} \left\| y_{VN}^{T} - z_s \right\|_2^2$$



Results Comparison



References

[1] Zhu, B., Liu, J.Z., Cauley, S.F., Rosen, B.R. and Rosen, M.S., 2018. Image reconstruction by domain-transform manifold learning.Nature,555(7697), p.487.
[2] Würfl T, Ghesu FC, Christlein V, Maier A. Deep Learning Computed Tomography. In: Medical Image Computing and Computer-Assisted Intervention; 2016. p. 432–440.
[3] Hammernik, K., Würfl, T., Pock, T. and Maier, A., 2017. A deep learning architecture for limited-angle computed tomography reconstruction. In Bildverarbeitung für die Medizin 2017 (pp. 92-97). Springer Vieweg, Berlin, Heidelberg.
[4] Hammernik, K., Klatzer, T., Kobler, E., Recht, M.P., Sodickson, D.K., Pock, T. and Knoll, F., 2018. Learning a variational network for reconstruction of accelerated MRI data. Magnetic resonance in medicine, 79(6), pp.3055-3071.

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

Thank you :)