FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

Deep Learning:
Embedding of operators

Deep Learning: Embedding of operators

- Introduction
- Known Operators in Neural Networks:
  - Known Operators
  - Universal Approximation Theorem
  - Bounds for Sequences of Operators
- Examples:
  - X – Ray Material decomposition
  - Learning Projection-Domain Weights

Introduction

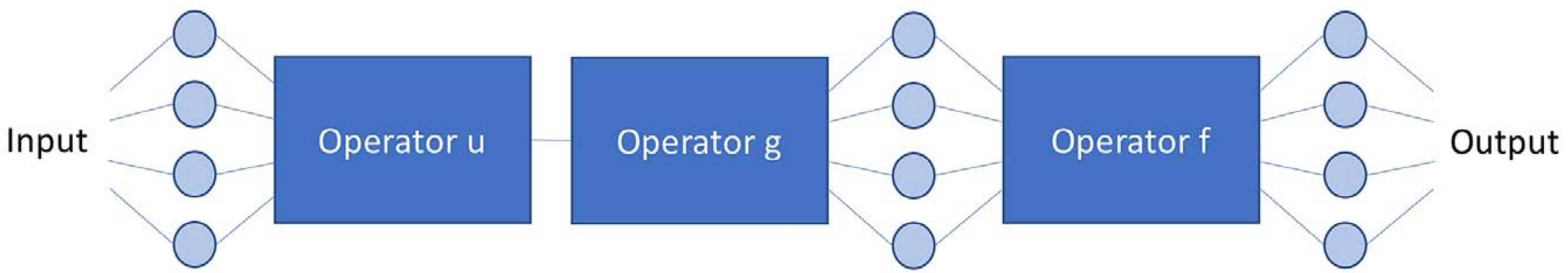
Known Operators in Neural Networks?

- Integrate knowledge and Properties from Physics / Signal Processing
- Reduce number of unknown parameters
- Less Training Samples
- Fewer Training Iterations

→ “Don’t reinvent the wheel…”

Known Operators embedded into a Network

- Schematic of the idea:



- Fix parts of the network by using prior information, to reduce number of parameters.

[1] Maier, Andreas, “Deep Learning Lecture SS 2018”, https://www.video.uni-erlangen.de/course/id/662
[2] Maier, Andreas, et al. "Precision learning: Towards use of known operators in neural networks." arXiv preprint arXiv:1712.00374 (2017).

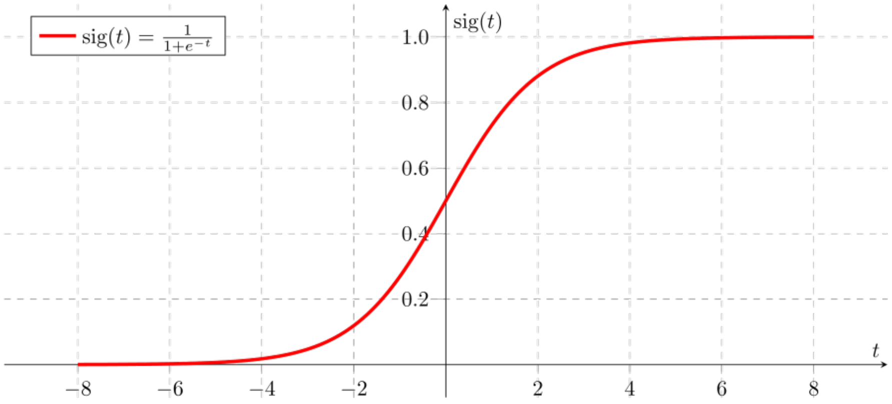
Recap: Universal Approximation Theorem

- Any continuous function can be approximated by Neural Net

$$u(\mathbf{x}) \approx U(\mathbf{x}) = \sum_i u_i s(\mathbf{w}_i^\top \mathbf{x} + w_{j,0})$$

- The error is bound by

$$|U(\mathbf{x}) - u(\mathbf{x})| \leq \epsilon_u$$



Approximation Sequences

- Specifically consider the use of two operators in sequence

$$f(\mathbf{x}) \quad = \quad g(\mathbf{u}(\mathbf{x}))$$

- Can be approximated in the following ways:

$$\begin{aligned} F_u(\mathbf{x}) = \quad & g(\mathbf{U}(\mathbf{x})) & = f(\mathbf{x}) - e_u \\ F_g(\mathbf{x}) = \quad & G(\mathbf{u}(\mathbf{x})) & = f(\mathbf{x}) - e_g \\ F(\mathbf{x}) = \quad & G(\mathbf{U}(\mathbf{x})) & = f(\mathbf{x}) - e_f \end{aligned}$$

Error of Approximation Sequences

- Approximation introduces error

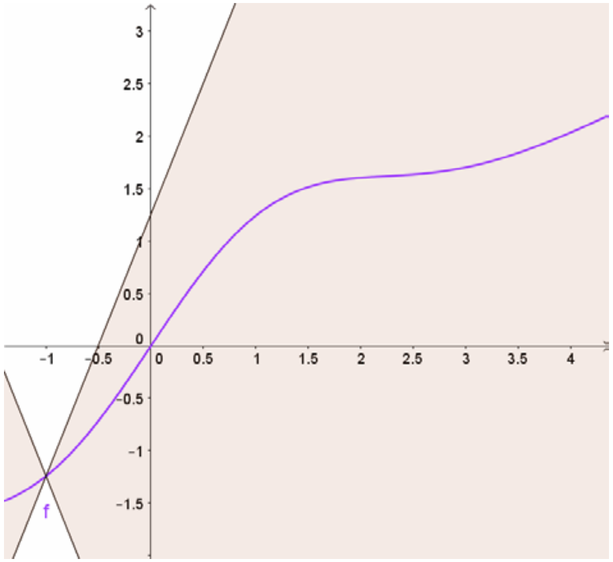
$$
\begin{aligned}
f(\mathbf{x}) &= g(\mathbf{u}(\mathbf{x})) = G(\mathbf{u}(\mathbf{x})) + e_g \\
&= \sum_j g_j s\left(u_j(\mathbf{x})\right) + g_0 + e_g \\
&= \sum_j g_j s\left(U_j(\mathbf{x}) + e_{u_j}\right) + g_0 + e_g
\end{aligned}
$$

- Now we want to find bounds to this errors, but how?

Error of Approximation Sequences

- We use the Lipschitz continuity of the sigmoid function

$$s(x+e) \leq s(x) + l_s \cdot |e|$$

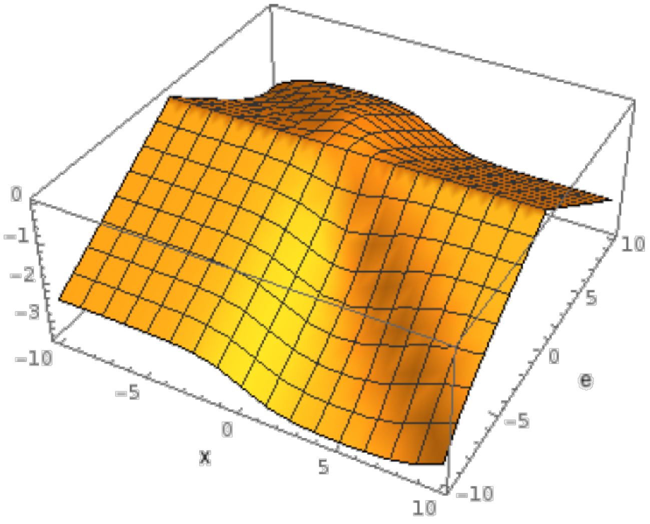


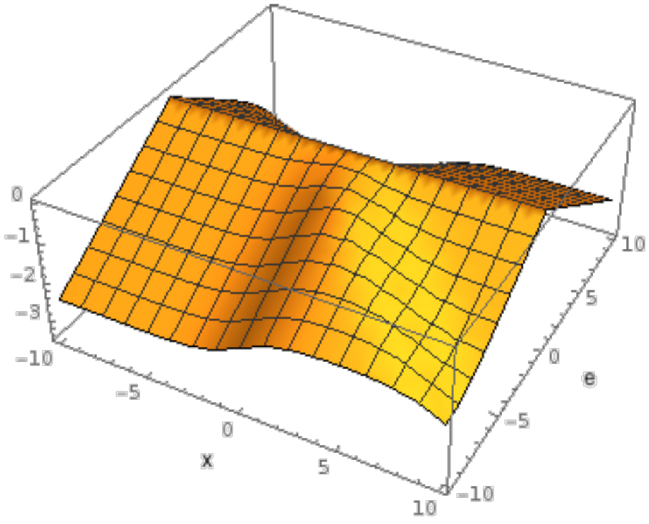
- For a Lipschitz continuous function: The graph is always entirely outside of the cone.

Error of Approximation Sequences

- But we have a linear combination of sigmoids!
- So combining with a multiplicative constant, we get an alternative formulation: *(without proof, but cool graph)*

$$g_j s(x+e) \leq g_j s(x) + |g_j| \cdot l_s \cdot |e|$$





Error of Approximation Sequences

- Now combining all equations yields:

$$
\begin{aligned}
f(\mathbf{x}) &= \sum_j g_j s\left(U_j(\mathbf{x}) + e_{u_j}\right) + g_0 + e_g \\
&\leq \underbrace{\sum_j g_j s(U_j(\mathbf{x})) + g_0}_{} + \textstyle\sum_j |g_j| \cdot l_s \cdot |e_{u_j}| + e_g \\
&\leq F(\mathbf{x}) + \textstyle\sum_j |g_j| \cdot l_s \cdot |e_{u_j}| + e_g \\
\underbrace{f(\mathbf{x}) - F(\mathbf{x})}_{} &\leq \sum_j |g_j| \cdot l_s \cdot |e_{u_j}| + e_g \\
e_f &\leq \sum_j |g_j| \cdot l_s \cdot |e_{u_j}| + e_g
\end{aligned}
$$

Error of Approximation Sequences

- Use the same idea for the lower bound

$$e_f \geq -\sum_j |g_j| \cdot l_s \cdot |e_{u_j}| - \epsilon_g$$

- And so we find a general bound:

$$|e_f| \leq \sum_j |g_j| \cdot l_s \cdot |e_{u_j}| + \epsilon_g$$

Error of Approximation Sequences

- We come to the following observations:

$$|e_f| \leq \underbrace{\sum_j |g_j| \cdot l_s \cdot |e_{u_j}|}_{\text{Error U(x)}} + \underbrace{\epsilon_g}_{\text{Error G(x)}}$$

- Error of U(x) and G(x) additive
- Error in U(x) amplified by g(x)
- Requires Lipschitz continuity
- This is an maximum error approximation

Deep Learning: Embedding of operators

- Introduction
- Known Operators in Neural Networks:
  - Known Operators
  - Universal Approximation Theorem
  - Bounds for Sequences of Operators
- Examples:
  - X – Ray Material decomposition
  - Learning Projection-Domain Weights

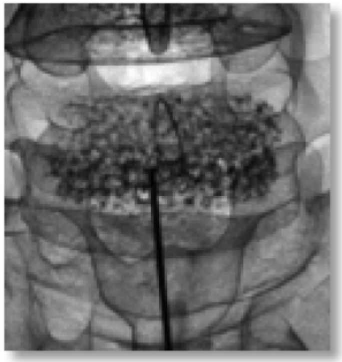
X – Ray Material decomposition

- Using a energy resolving detector we get multiple images at different energy levels
- This can be interpreted to be similar to using colors
- Many properties of the transform are known

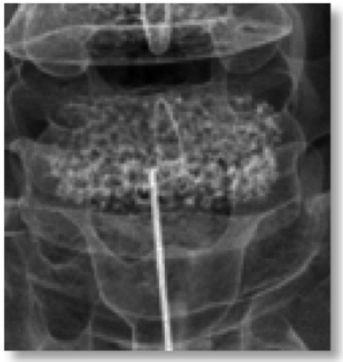
X – Ray Material decomposition

- Example application:

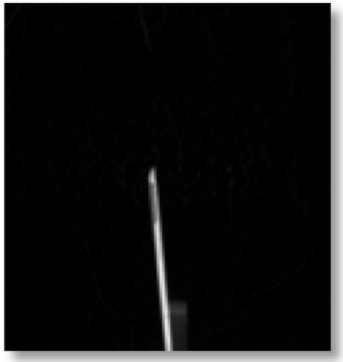
  We want to subtract a needle from an phantom



X-ray image I(x; y)
of the phantom
with the needle



data after example transform
u(I(x; y)), i.e. line integral
domain.



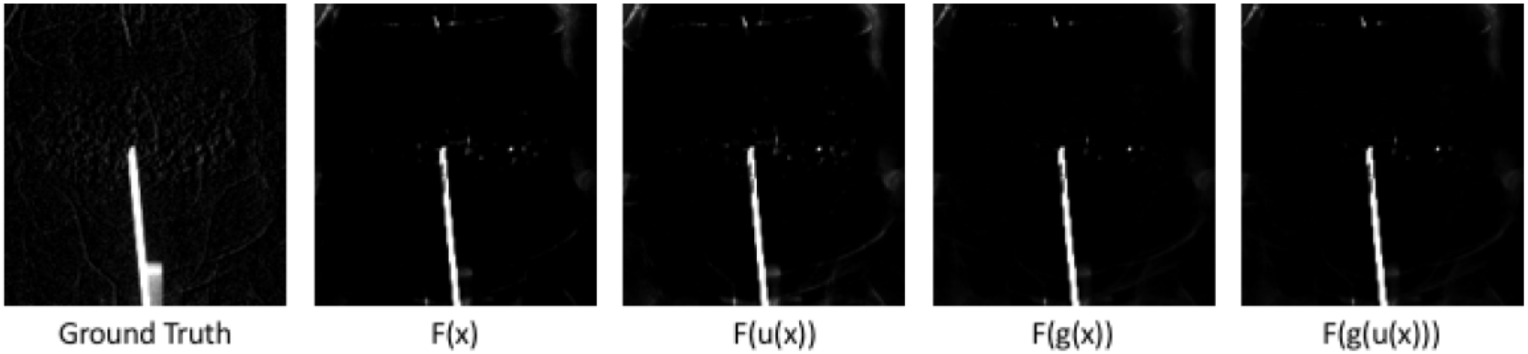
ground truth

- How do it?

  Use known transforms in the network that make use of the physics of the energy resolving detector that is employed

X – Ray Material decomposition: Results

- The more known transform the better the results got.
- This also is inline with the derivation at the beginning.



OVERVIEW ON THE RESULTS OF THE PREDICTION. PEARSON'S r IS GENERALLY HIGH, WHILE THE SSIM IS DRASTICALLY INCREASED WITH INCREASING PRIOR KNOWLEDGE.

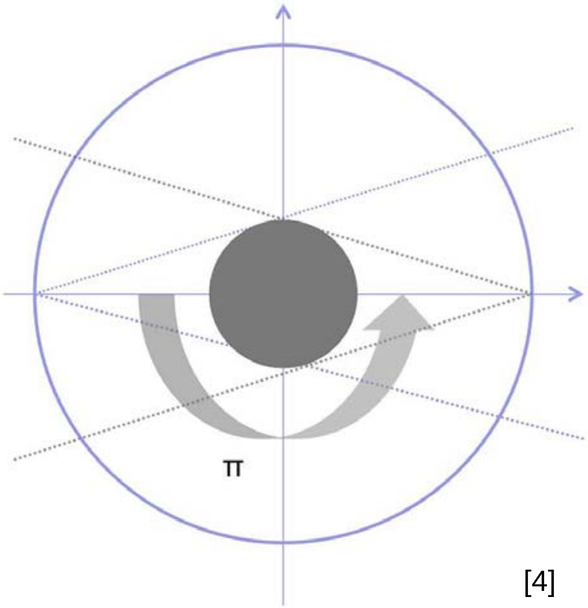
	$F(\mathbf{I})$	$F(u(\mathbf{I}))$	$F(g(\mathbf{I}))$	$F(g(u(\mathbf{I})))$
Pearson's r [%]	95.0	95.2	95.1	**95.5**
SSIM [%]	54.1	63.1	73.8	**88.4**

Deep Learning: Embedding of operators

- Introduction
- Known Operators in Neural Networks:
  - Known Operators
  - Universal Approximation Theorem
  - Bounds for Sequences of Operators
- Examples:
  - X – Ray Material decomposition
  - Learning Projection-Domain Weights

Learning Projection-Domain Weights from Image Domain in Limited Angle Problems [3]

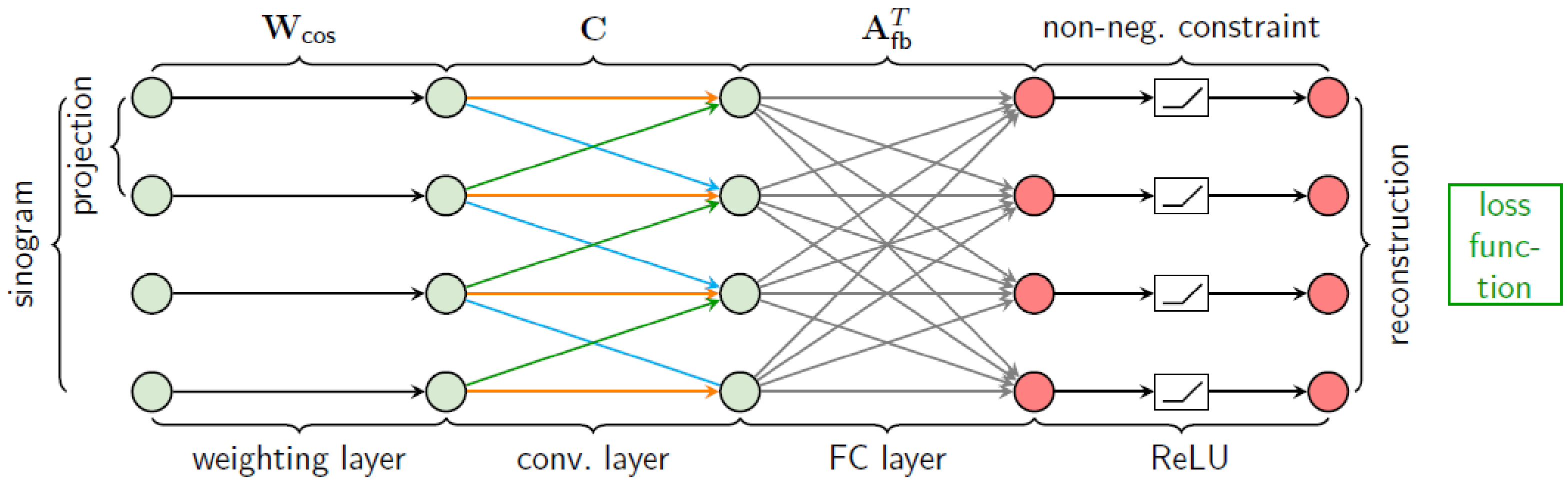
- Goal: to learn redundancy weights for our FBP-Algorithm
  - What are redundancy weights?
  - In Short-Scan some rays are measured twice, so we need to weigh them accordingly



[4]

Learning Projection-Domain Weights

- The proposed Network for Fan-Beam:



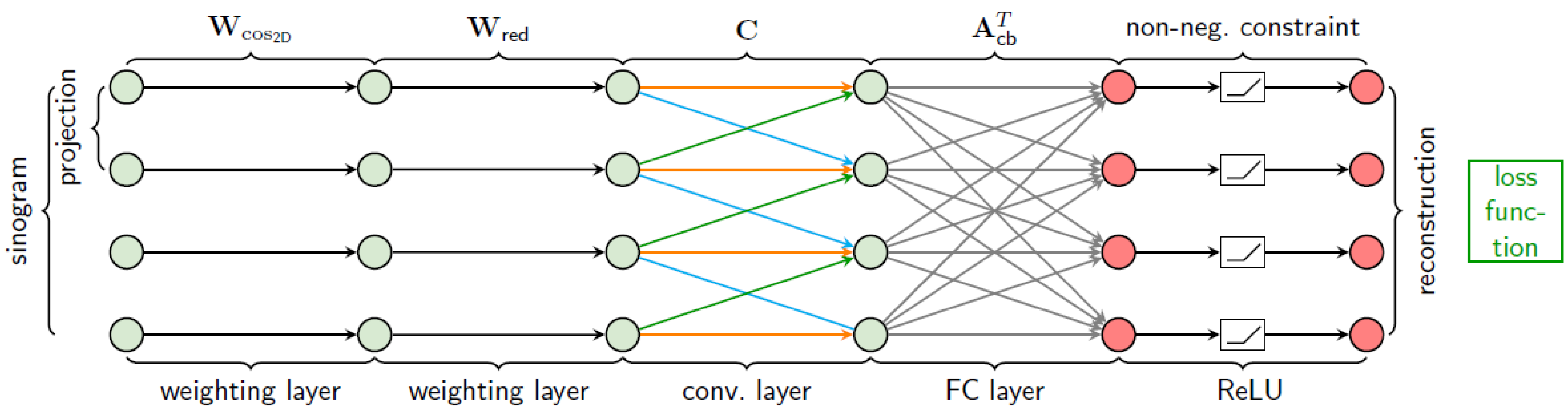
To be learnt

Filtering

Known Operator

Learning Projection-Domain Weights

- Fan-Beam is only 2D
- Transition clinically relevant cone-beam geometry (3D)
- Additional cosine weighting and use of the FDK - Alg



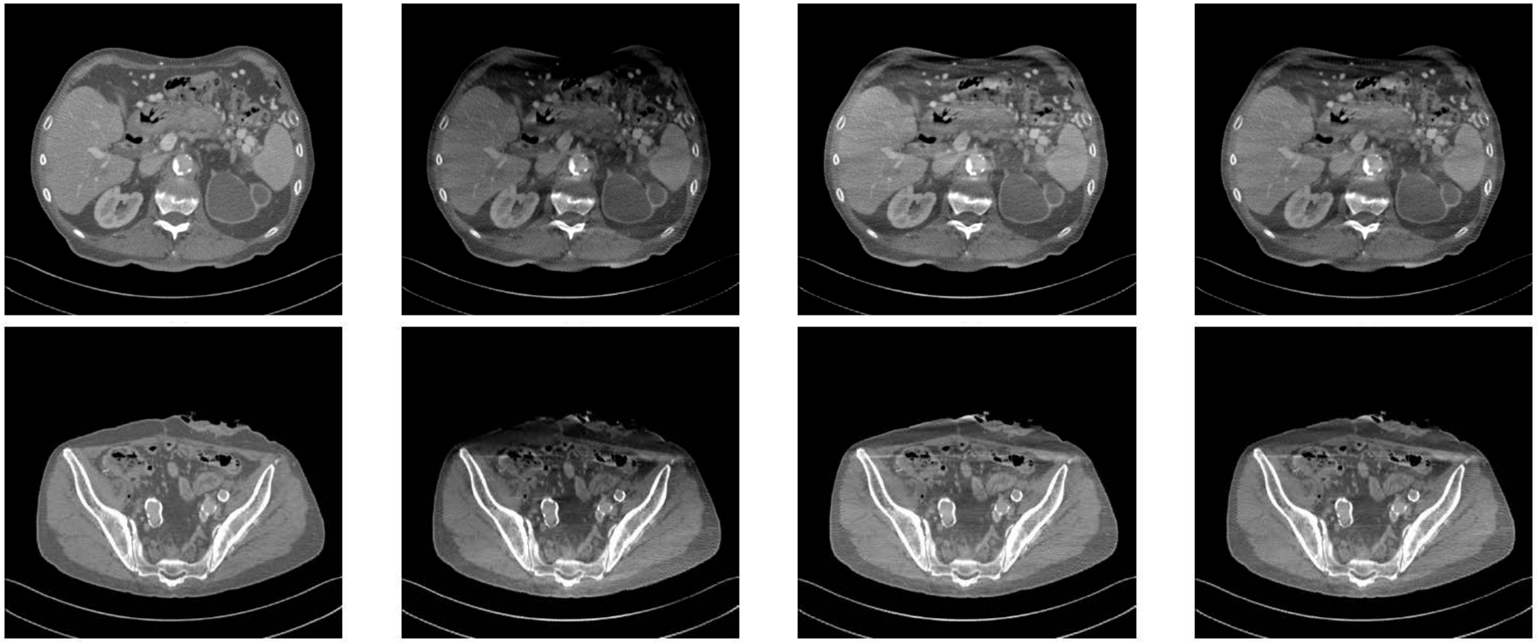
To be learnt

Filtering

Known Operator

Learning Projection-Domain Weights

- Results:



Ground truth | Half projections | Weights of Schäfer et. al | Learned Weights

Learning Projection-Domain Weights

- Results with noise:

Parker Weights

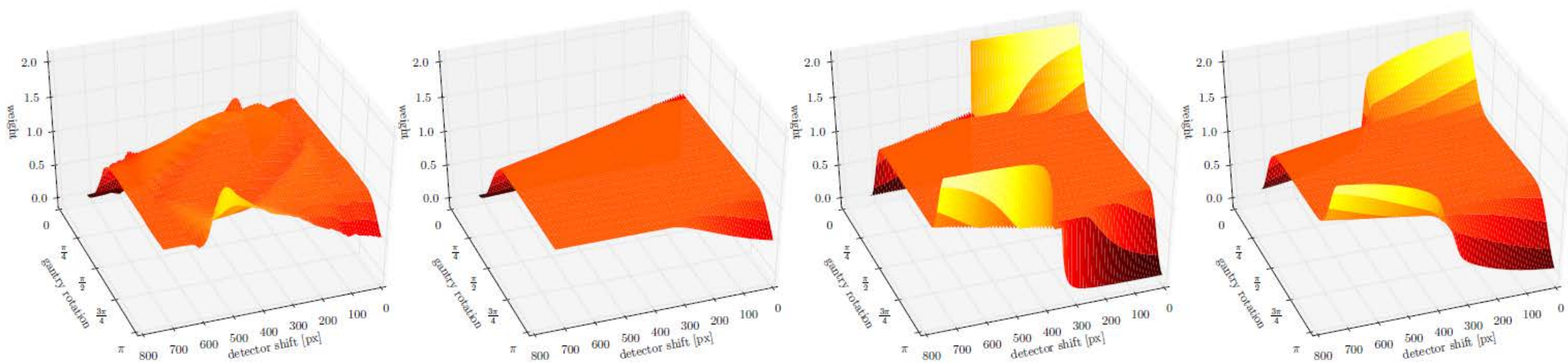
Learned Weights

Parker Weights

Learned Weights

Learning Projection-Domain Weights

- Results: Interpretation of learned weights is possible!

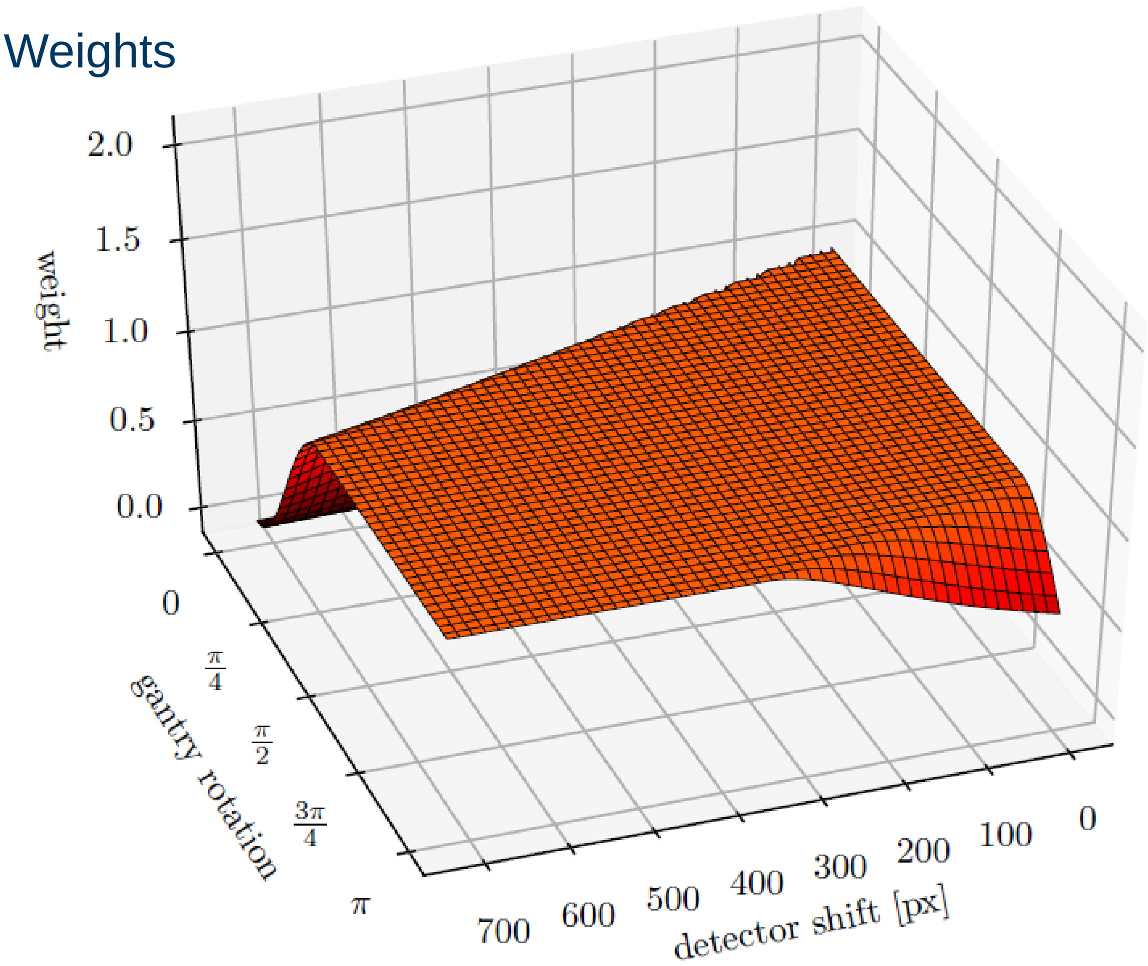


Learned Weights | Parker Weights | Weights by Riess et al. *(without gaussian smoothing)* | compensation weights proposed by Schäfer et al.

- The loss of mass typically caused by missing data can be corrected by learned compensation weights
- no additional computational effort

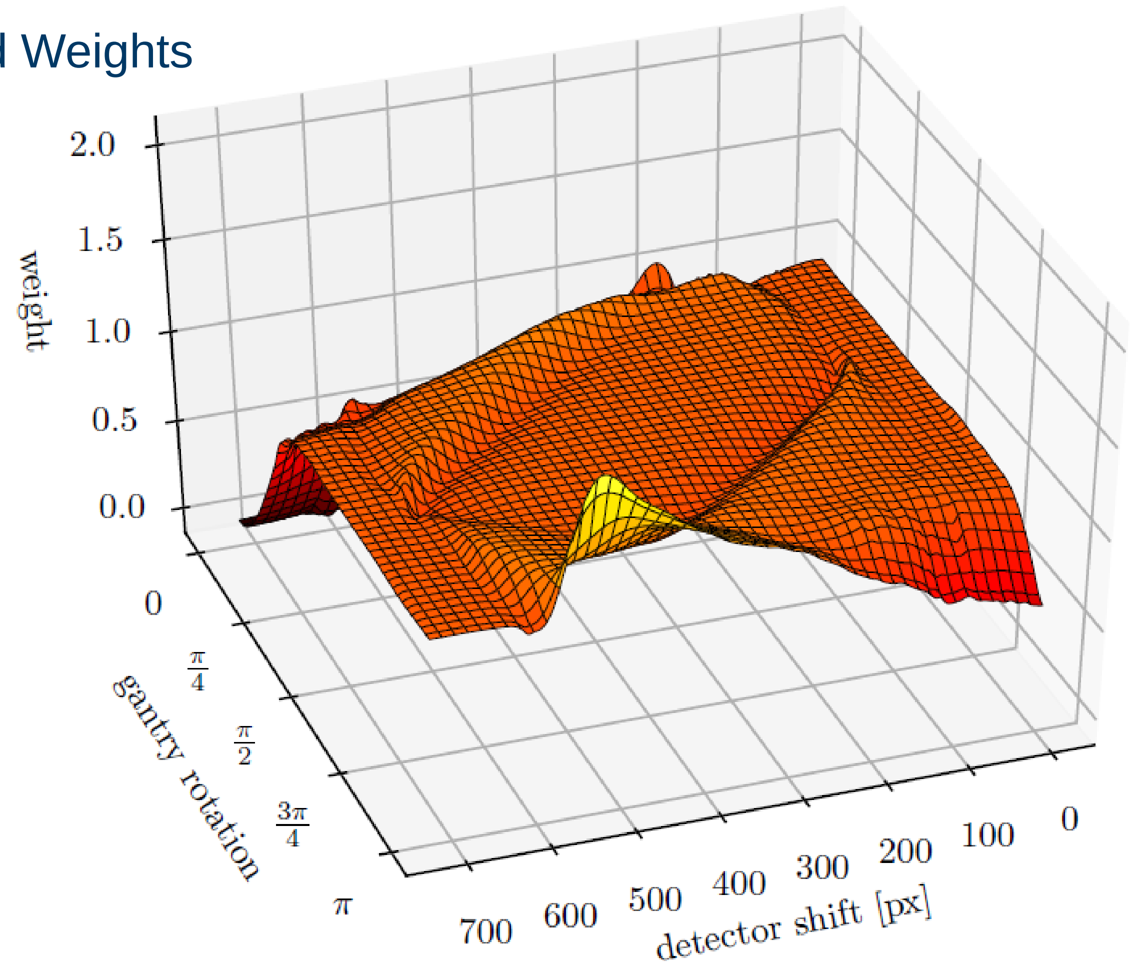
Learning Projection-Domain Weights

- Parker Weights



Learning Projection-Domain Weights

- Learned Weights



Literature

- [1] Maier, Andreas, “Deep Learning Lecture SS 2018”, https://www.video.uni-erlangen.de/course/id/662
- [2] Maier, Andreas, et al. "Precision learning: Towards use of known operators in neural networks." arXiv preprint arXiv:1712.00374 (2017).
- [3] Würfl, Tobias, et al. "Deep learning computed tomography: Learning projection-domain weights from image domain in limited angle problems." IEEE transactions on medical imaging 37.6 (2018): 1454-1463.
- [4] Maier, Andreas, “MIPDA Lecture SS 2018”