Published in final edited form as:
Nat Mach Intell. 2019 August ; 1(8): 373–380. doi:10.1038/s42256-019-0077-5.

Learning with Known Operators reduces Maximum Training Error Bounds

Andreas K. Maier, **Christopher Syben**, **Bernhard Stimpel**, **Tobias Würfl**, **Mathis Hoffmann**, **Frank Schebesch**, **Weilin Fu**, **Leonid Mill**
[*]Department of Computer Science, Friedrich-Alexander University Erlangen-Nürnberg, Germany

Lasse Kling,
[†]Helmholtz Zentrum Berlin für Materialien und Energie, Germany

Silke Christiansen
[‡]Physics Department, Free University Berlin and the Helmholtz Zentrum Berlin für Materialien und Energie, Germany

Abstract

We describe an approach for incorporating prior knowledge into machine learning algorithms. We aim at applications in physics and signal processing in which we know that certain operations must be embedded into the algorithm. Any operation that allows computation of a gradient or sub-gradient towards its inputs is suited for our framework. We derive a maximal error bound for deep nets that demonstrates that inclusion of prior knowledge results in its reduction. Furthermore, we also show experimentally that known operators reduce the number of free parameters. We apply this approach to various tasks ranging from CT image reconstruction over vessel segmentation to the derivation of previously unknown imaging algorithms. As such the concept is widely applicable for many researchers in physics, imaging, and signal processing. We assume that our analysis will support further investigation of known operators in other fields of physics, imaging, and signal processing.

1 Introduction

Pattern analysis and machine intelligence have been focussed predominantly on tasks that mimic perceptual problems. These are typically modelled as classification or regression tasks in which the actual reference stems from a human observer that defines the *ground-*

andreas.maier@fau.de.

Author Contributions
Andreas Maier is the main author of the paper and is responsible for the writing of the manuscript, theoretical analysis, and experimental design. Christopher Syben and Bernhard Stimpel contributed to the writing of Section 3.3 and the supporting experiments. Tobias Würfl and Mathis Hoffmann supported writing Section 3.1 and performed the experiments reported in this section. Frank Schebesch contributed to the mathematical analysis and the writing thereof. Weilin Fu conducted the experiments supporting Section 3.2 and contributed to their description. Leonid Mill, Lasse Kling, and Silke Christiansen contributed to the experimental design and the writing of the manuscript.

Competing Interests
The authors declare that they do not have any competing interests.

truth. As we have only limited understanding on how these man-made classes emerge from the human mind, there is only limited knowledge available. As such, pattern recognition has relied on expert knowledge to design features that are suited towards a particular recognition task [1]. In order to alleviate the task of feature-design, researchers started also learning feature descriptors as a part of the training procedure [2]. Implementation of such on efficient hardware gave rise to first models that could outperform classical feature extraction methods significantly [3] and was one of the milestone works in the emerging field of *deep learning*.

With the rise of deep learning [4], researchers became aware that these methods of general function learning are applicable to a much wider range than mere perceptual tasks. Today, machine learning is applied in a much wider range of applications. Examples range from image super resolution [5], image denoising and inpainting [6], or even computed tomography [7]. In these fields, the methods from deep learning are often directly applied and often show performances that are either en par or even significantly better than results found with state-of-the-art methods. Yet, there are also reports that present surprising results in which parts of the image are hallucinated [8, 9]. In particular [9] demonstrates that mismatches in training and test data leads to dramatic changes in the produced result. Hence, *blind* deep learning methods have to be performed with care in order to be successful.

In this article, we explore the use of known operations within machine learning algorithms. First, we analyze the problem from a theoretical perspective and study the effect of prior knowledge in terms of maximal error bounds. This is followed by three applications in which we use prior operators to study to their effect on the respective regression or classification problem. Lastly, we discuss our observations in relation to other works in literature and give an outlook on future work. Note that some of the work presented here is based on prior conference publications [10, 11, 12, 13].

2 Known Operator Learning

The general idea of known operator learning is to embed entire operations into a learning problem. Figure 1 presents the idea graphically. We generally refer to the N_D-dimensional input of our trained algorithm as $\mathbf{x}' \in \mathbb{R}^{N_D}$. In order to increase readability, we use an extended version $\mathbf{x} \in \mathbb{R}^{N_D+1}$ such that inner products with some weight vector $\mathbf{w}'$ plus bias w_0 can be conveniently written, i. e. $\mathbf{w}'^{\top} \mathbf{x}' + w_0 = \mathbf{w}^{\top} \mathbf{x}$. Before looking into the properties of this approach and in particular maximal error bounds, we shortly summarize the Universal Approximation Theorem as it is closely related to our analysis. Note that the supplementary material to this article contains all proofs for the presented the-orems in this section.

2.1 Universal Approximation

Theorem 1 (Universal Approximation Theorem). *Let* $\varphi(t) : \mathbb{R} \to \mathbb{R}$ *be a non-constant, bounded, and continuous function and* $u(\mathbf{x})$ *be a continuous function on a compact set* $\mathcal{D} \subset \mathbb{R}^{N_D+1}$. *Then, there exist an integer N, weights* $\mathbf{w} \in \mathbb{R}^{N_D+1}$, *and* $u_i \in \mathbb{R}$ *that form an approximation* $\hat{u}(\mathbf{x})$

$$\hat{u}(\mathbf{x}) = \sum_{i=0}^{N-1} u_i \varphi(\mathbf{w}_i^\top \mathbf{x}) \tag{1}$$

such that the inequality

$$|\hat{u}(\mathbf{x}) - u(\mathbf{x})| \leq \epsilon_u \tag{2}$$

holds for all $\mathbf{x} \in \mathcal{D}$ *and* $\epsilon_u > 0$.

Theorem 1 states that for any continuous function $u(\mathbf{x})$ an approximation $\hat{u}(\mathbf{x})$ can be found such that the difference between true function and approximation is bounded by ϵ_u. With increasing number of nodes N, ϵ_u will decrease. In literature, this result is often referred to as Universal Approximation Theorem [14, 15] and forms the fundamental result that neural networks with just a single hidden layer are general function approximators. Yet, this type of approximation may result in a very high requirement for the choice of N which is the reason why stacked layers of different type are known to be more successful [16].

We can extend Theorem 1 to vector-valued functions $\mathbf{u}(\mathbf{x}) : \mathcal{D} \longrightarrow \mathbb{R}^{N_O}$ on $\mathcal{D}$ by postulating Theorem 1 for each of their components k

$$|U_k(\mathbf{x}) - u_k(\mathbf{x})| \leq \epsilon_{u_k}. \tag{3}$$

Hence, universal approximation generally also applies to N_O-dimensional functions.

2.2 Known Operator Error Bounds

Knowing the limits of general function approximation, we are now interested in finding limits for mixing known and approximated operators. As previously mentioned, deep networks are never constructed out of a single layer, but rather take the form of the configuration shown in Figure 1. Hence, we need to consider layered networks to analyze the maximal error bounds. Instead of investigating entire networks, we choose to simplify our theoretical analysis to the special case

$$f(\mathbf{x}) = g(\mathbf{u}(\mathbf{x})) : \mathcal{D} \to \mathbb{R}$$

with $g(\mathbf{x}) : S \longrightarrow \mathbb{R}$, $\mathbf{u}(\mathbf{x}) : \mathcal{D} \longrightarrow \mathcal{S}$, and compact sets $\mathcal{S} \subset \mathbb{R}^{N_I+1}$ and $\mathcal{D} \subset \mathbb{R}^{N_D+1}$. Note that this simplification does not limit the generality of our analysis, as we can map any knowledge on the structure of the network architecture either onto the output function $g(\mathbf{x})$, the intermediate function $\mathbf{u}(\mathbf{x})$, or directly as a transform of the inputs $\mathbf{x}$. Generalisation to N_O-dimensional functions is also possible following the idea shown in Eq. 3.

Previous definition of $f(\mathbf{x})$ allows us to investigate different forms of approximation. In particular, we are able to introduce approximations $\hat{\mathbf{u}}(\mathbf{x})$ and $\hat{g}(\mathbf{x})$ following Theorem 1:

$$\hat{f}_g(\mathbf{x}) = \hat{g}(\mathbf{u}(\mathbf{x})) = f(\mathbf{x}) - e_g \tag{4}$$

$$\hat{f}_u(\mathbf{x}) = g(\hat{\mathbf{u}}(\mathbf{x})) = f(\mathbf{x}) - e_u \tag{5}$$

$$\hat{f}(\mathbf{x}) = \hat{g}(\hat{\mathbf{u}}(\mathbf{x})) = f(\mathbf{x}) - e_f \tag{6}$$

Here $|e_u| \leq \epsilon_u$, $|e_g| \leq \epsilon_g$, and $|e_f| \leq \epsilon_f$ denote the errors that are introduced by respective approximation of $\mathbf{u}$, g, and f.

Next, we are interested in finding bounds on $|e_f|$ using above approximations. For the case of known $\mathbf{u}(\mathbf{x})$, we can substitute $\mathbf{x}^* \coloneqq \mathbf{u}(\mathbf{x})$, as $\mathbf{u}(\mathbf{x})$ is a fixed function. In this case Theorem 1 directly applies and a bound on $|e_f|$ is found as $|e_f| \leq \epsilon_g$ with $|e_g| \leq \epsilon_g$. If we would know $g(\mathbf{x})$ in addition, e_g would be 0 and the bound would shrink to the case of equality.

The case described in Eq. 5 is slightly more complicated, but we are also able to find general bounds as shown in Theorem 2.

Theorem 2 (Known Output Operator Theorem). *Let* $\varphi(\mathbf{x}) : \mathbb{R} \rightarrow \mathbb{R}$ *be a non-constant, bounded, and continuous function and* $f(\mathrm{x}) = g(\mathbf{u}(\mathrm{x})) : \mathscr{D} \rightarrow \mathbb{R}$ *be a continuous function on* $\mathscr{D} \subset \mathbb{R}^{N_D+1}$. *Further let* $g(\mathbf{x}) : \mathscr{S} \rightarrow \mathbb{R}$ *be Lipschitz-continuous function with Lipschitz constant* $l_g = \sup\{\|\nabla g(\mathbf{x})\|_p\}$ *with* $p \in \{1, 2\}$ *on* $\mathscr{S} \subset \mathbb{R}^{N_I+1}$ *and*

$$\hat{u}_k(\mathbf{x}) = \sum_{i=0}^{N_{\hat{u}_k} - 1} u_i \varphi(\mathbf{w}_{i,k}^\top \mathbf{x}) \tag{7}$$

be a general function approximator of $\mathbf{u}(\mathbf{x})$ *with integer* $N_{\hat{u}_k}$, *weight* $\mathbf{w}_{i,k} \in \mathbb{R}^{N_D+1}$ *, and* $u_i \in \mathbb{R}$. *Then,* $e_f = f(\mathbf{x}) - \hat{f}(\mathbf{x})$ *with* $\hat{f}(\mathbf{x}) = \hat{g}(\hat{\mathbf{u}}(\mathbf{x})) = g(\hat{\mathbf{u}}(\mathbf{x}))$ *— as g is known — is generally bounded for all* $\mathbf{x} \in \mathscr{D}$ *by*

$$|e_f| \leq \ell_g \cdot ||\mathbf{e}_u||_p \tag{8}$$

with $e_u = e_f$ *and component-wise approximation errors* $\mathbf{e}_u = [e_{u_0}, \ldots e_{u_{N_I}}]^\top$.

The bound for $|e_f|$ is found using a Lipschitz constant l_g on $g(\mathbf{x})$ which implies that the theorem will only hold, if Lipschitz-bounded functions are used for $g(\mathbf{x})$. Analysis of Eq. 8 reveals that knowing $\mathbf{u}(\mathbf{x})$ in this case, would imply $\mathbf{e}_u = \mathbf{0}$ which also yields equality on both sides.

We further explore this idea in Theorem 3. It describes a bound for the case that both $g(\mathbf{x})$ and $\mathbf{u}(\mathbf{x})$ are approximated.

Theorem 3 (Unknown Operator Theorem). *Let* $\varphi(\mathbf{x}) : \mathbb{R} \rightarrow \mathbb{R}$ *be a non-constant, bounded, and continuous function with Lipschitz-bound* l_φ *and* $f(\mathbf{x}) = g(\mathbf{u}(\mathbf{x})) : \mathscr{D} \rightarrow \mathbb{R}$ *be a continuous function on* $\mathscr{D} \subset \mathbb{R}^{N_D+1}$ *Further let*

$$\hat{g}(\mathbf{x}) = \sum_{j=0}^{N_{\hat{g}}-1} g_j \varphi(\mathbf{w}_j^\top \mathbf{x}) \quad \text{and} \tag{9}$$

$$\hat{u}_k(\mathbf{x}) = \sum_{i=0}^{N_{\hat{u}_k}-1} u_i \varphi(\mathbf{w}_{i,k}^\top \mathbf{x}) \tag{10}$$

be general function approximators of $g(\mathbf{x}) : \mathscr{S} \to \mathbb{R}$ *and* $\mathbf{u}(\mathbf{x}) : \mathscr{D} \to \mathscr{S}$ *with integers* $N_{\hat{g}}$ *and* $N_{\hat{u}_k}$, *weights* $\mathbf{w}_j \in \mathbb{R}^{N_I+1}$, $\mathbf{w}_{i,k} \in \mathbb{R}^{N_D+1}$, g_j, $u_i \in \mathbb{R}$, *and compact sets* $\mathscr{S} \subset \mathbb{R}^{N_I+1}$ *and* $\mathscr{D} \subset \mathbb{R}^{N_D+1}$. *Then,* $e_f = f(\mathbf{x}) - \hat{f}(\mathbf{x})$ *with* $\hat{f}(\mathbf{x}) = \hat{g}(\hat{\mathbf{u}}(\mathbf{x}))$ *is generally bounded for all* $\mathbf{x} \in \mathscr{D}$ *by*

$$|e_f| \leq \sum_{j=0}^{N_{\hat{g}}-1} |g_j| \cdot l_\varphi \cdot |e_{\mathbf{u}_j}| + \epsilon_g \ . \tag{11}$$

where $\epsilon_g \geq |e_g|$, $e_{\mathbf{u}_j} = \mathrm{w}_j^\top \mathbf{e}_u$, *and* $\mathbf{e}_u = [e_{u_0}, \ldots e_{u_{N_I}}]^\top$ *is the vector of errors introduced by the components of* $\hat{\mathrm{u}}(\mathbf{x})$.

The bound is comprised of two terms in an additive relation:

$$\underbrace{\sum_{j=0}^{N_{\hat{g}}-1} |g_j| \cdot l_\varphi \cdot |e_{\mathrm{u}_j}|}_{\text{only dependent on } \hat{\mathbf{u}}(\mathbf{x})} + \underbrace{\epsilon_g}_{\text{only dependent on } \hat{g}(\mathbf{x})} \tag{12}$$

where the first term vanishes, if $\mathbf{u}(\mathbf{x})$ is known as $|e_{\mathbf{u}_j}| = 0 \ \forall j$ and the second term vanishes for known $g(\mathbf{x})$ as $e_g = 0$. Hence for all of the considered cases, knowing $g(\mathbf{x})$ or $\mathbf{u}(\mathbf{x})$ is beneficial and allows to shrink the maximal training error bounds.

Given the previous observations, we can now also explore deeper networks that try to mimic the structure of the original function. This gives rise to Theorem 4.

Theorem 4 (Unknown Operators in Deep Networks). *Let* $\mathbf{u}_\ell(\mathbf{x}_\ell) : \mathscr{D}_\ell \mapsto \mathscr{D}_{\ell-1}$ *be a continuous function with Lipschitz-bound* l_{u_ℓ} *on compact set* $\mathscr{D}_\ell \subset \mathbb{R}^{N_\ell}$ *with integer* $\ell > 0$. *Further let* $\mathbf{f}_\ell(\mathbf{x}_\ell) : \mathscr{D}_\ell \mapsto \mathscr{D}$ *be a function composed of* ℓ *layers / function blocks defined as recursion* $\mathbf{f}_\ell(\mathbf{x}_\ell) = \mathbf{f}_{\ell-1}(\mathbf{u}_\ell(\mathbf{x}_\ell))$ *with* $\mathbf{f}_{\ell=0}(\mathbf{x}) = \mathbf{x}$ *on compact set* $\mathscr{D} \subset \mathbb{R}^{N_D+1}$ *bound by Lipschitz constant* ℓ_{f_ℓ} *with* $\ell_{\mathrm{f}_{\ell=0}} = 1$. *Recursive function* $\hat{\mathrm{f}}_\ell(\mathbf{x}_\ell) = \hat{\mathrm{f}}_{\ell-1}(\hat{\mathrm{u}}_\ell(\mathbf{x}_\ell))$ *with* $\hat{\mathrm{f}}_{\ell=0}(\mathbf{x}) = \mathbf{x}$ *is then an approximation of* $\mathbf{f}_\ell(\mathbf{x}_\ell)$. *Then,* $\mathrm{e}_{f,\ell} = \mathrm{f}_\ell(\mathbf{x}_\ell) - \hat{\mathrm{f}}_\ell(\mathbf{x}_\ell)$ *is generally bounded for all* $\mathbf{x}_\ell \in \mathscr{D}_\ell$ *and for all* $\ell > 0$ *in each component k by*

$$|e_{f,\ell,k}| \le \sum_{\ell_i = 1}^{\ell} ||\mathbf{e}_{u,\ell_i}||_p \cdot l_{\mathbf{f}_{\ell_i - 1}} \tag{13}$$

where $\mathbf{e}_{u,\ell} = [e_{u,\ell,0}, \dots e_{u,\ell,N_\ell}]^\top$ *is the vector of errors introduced by* $\hat{\mathbf{u}}_\ell(\mathbf{x}_\ell)$.

If we investigate Theorem 4 closely, we identify similar properties to Theorem 3. The errors of each layer / function block $\mathbf{u}_\ell(\mathbf{x}_\ell)$ are additive. If a layer is known, the respective error vector $\mathbf{e}_{u,\ell} = \mathbf{0}$ vanishes and the respective part of the bound cancels out. Furthermore, later layers have a multiplier effect on the error as their Lipschitz constants amplify $\|\mathbf{e}_{u,\ell}\|_p$. Note that the relation $l_{\mathbf{f}_\ell} \le \prod_{\ell_i=1}^{\ell} l_{\mathbf{u}_{\ell_i}}$ is shown in the supplementary material. A large advantage of Theorem 4 over Theorem 3 is that the Lipschitz constants $l_{\mathbf{f}_\ell}$ that appear in the error term $\|\mathrm{e}_{u,\ell_i}\|_p \cdot l_{\mathrm{f}_{\ell_i - 1}}$ are the ones of the true function $\mathbf{f}_\ell(\mathbf{x}_\ell)$. Therefore, the amplification effects are only dependent of the structure of the true function and independent of the actual choice of the universal function approximator. The approximator only influences the actual error $\mathbf{e}_{u,\ell}$.

Above observations pave the way to incorporating prior operators into different architectures. In the following, we will highlight several applications in which we explore blending deep learning with prior operators.

3 Application Examples

We believe that known operators have a wide range of applications in physics and signal processing. Here, we highlight three approaches to use such operators. All three applications are from the domain of medical imaging, yet the method is applicable to many more disciplines to be discovered in the future. The results presented here are based on conference contributions [10, 11, 13]. Note that the supplementary material contains descriptions of experiments, data, and additional figures that were omitted here for brevity.

3.1 Deep Learning Computed Tomography

In computed tomography, we are interested in computing a reconstruction **y** from a set of projection images **x**. Both are related by the X-ray transfrom **A**:

$$\mathbf{Ay} = \mathbf{x}$$

Solving for **y** requires inversion of above formula. The Moore-Penrose inverse of **A** yields the following solution:

$$\mathbf{y} = \mathbf{A}^\top \left(\mathbf{A}\mathbf{A}^\top\right)^{-1} \mathbf{x}$$

This type of inversion gives rise to the class of filtered back-projection methods, as it can be shown that $(\mathbf{AA}^\top)^{-1}$ takes the form of a circulant matrix **K**, i. e. $\mathbf{K} = (\mathbf{AA}^\top)^{-1} = \mathbf{F}^H\mathbf{CF}$, where **F** denotes the Fourier transform, $\mathbf{F}^H$ its inverse, and **C** a diagonal matrix that corresponds to the Fourier transform of **K**. As **K** typically is associated with a large receptive field, it is typically implemented in Fourier space. In order to be applicable for other geometries, such as fan-beam reconstruction additional Parker and cosine weights have to be incorporated that can elegantly be summarised in an additional diagonal matrix **W** to yield

$$\mathbf{y} = \mathrm{ReLu}\left(\mathbf{A}^{\top}\mathbf{KWx}\right) \tag{14}$$

where ReLu(·) suppresses negative values as the final reconstruction algorithm.

Following the paradigm of known operator learning, Eq. 14 can also interpreted as a neural network structure as it only contains diagonal, circulant, and fully connected matrix operators displayed in Figure 2. A practical limitation of **A** is that it typically is a very large and sparse matrix. In practice, it is therefore never instantiated, but only evaluated on the fly using fast ray-tracing methods. For 3-D problems, the full matrix size is way beyond the memory restrictions of today’s compute systems. Furthermore, none of the parameters need to be trained as all of them are known for complete data acquisitions.

Incomplete data cannot be reconstructed with this type of algorithm and would lead to strong artifacts. We can still tackle limited data problems if we apply additional training of our network. As $\mathbf{A}^{\top}$ is large, we treat it as fixed during the training and only allow modification of **W** and **K**. Results and experimental details are demonstrated in the supplementary material. Training of both matrices clearly improves the image reconstruction result. In particular, the trained algorithm learns to compensate for the loss of mass in areas of the reconstruction in which rays are missing.

As the trained algorithm is mathematically equivalent to the original filtered back-projection method, we are able to map the trained weights back onto their original interpretation which allows comparison to state-of-the-art weights. In Figure 3, we can see that the trained weights show similarity with the approach published by Schäfer et al. [18]. In contrast to Schäfer et al. who arrived at their weights following intuition, our approach is optimal with respect to our training data. In our present model, we have to re-train the algorithm for every new geometry. This could be avoided by modelling the weights using a continuous function which is sampled by the reconstruction network.

3.2 Learning from Heuristic Algorithms

Incorporating known operators generally allows blending of deep learning methods with traditional image processing approaches. In particular, we are able to choose heuristic methods that are well understood and augment them with neural network layers.

One example for such a heuristic method is Frangi’s vesselness [19]. The vesselness values for dark tubes are calculated using the following formula:

$$V_0(\sigma) = \begin{cases} 0, & \text{if } \lambda_2 < 0, \\ \exp(-\frac{R_B^2}{2\beta^2})(1 - \exp(-\frac{S^2}{2c^2})), & \text{otherwise,} \end{cases} \tag{15}$$

where $|\lambda_1| < |\lambda_2|$ are the eigenvalues, $S = \sqrt{\lambda_1^2 + \lambda_2^2}$ is the second order structureness, $R_B = \frac{\| \lambda_1 \|}{\| \lambda_2 \|}$ is the blobness measure, β, c are image-dependent parameters for blobness and structureness terms, and V_0 stands for the vesselness value.

The entire multi-scale framework of Frangi filter can be mapped onto a neural network architecture [11]. In Frangi-Net, each step of the Frangi filter is replaced with a network counterpart and data normalization layers are added to enable end-to-end training. Multi-scale analysis is formed as a series of trainable filters, followed by eigenvalue computation in specialized fixed function network blocks. This is followed by another fixed function – the actual vesselness measure as described in Eq. 15. Figure 4

We compare the segmentation result of the proposed Frangi-Net with the original Frangi filter, and show that the Frangi-Net outperforms Frangi filter regarding all evaluation metrics. In comparison to the state-of-the-art image segmentation model U-Net, Frangi-Net contains less than 6% the number of trainable parameters, while achieving an AUC score around 0.960, which is only 1% inferior to that of the U-Net. Adding a trainable guided-filter before Frangi-Net as preprocessing step yields an AUC 0.972 with only 8.5% of the trainable parameters of U-Net which is statistically not distinguishable from U-Net's AUC of 0.974.

Hence using our approach of known operators, we are able to augment heuristic methods by blending them with methods of deep learning saving many trainable parameters.

3.3 Deriving Networks

A third application of known operator learning that we would like to highlight in this paper, is the derivation of new network architectures from mathematical relations of the signal processing problem at hand. In the following, we are interested in hybrid imaging of magnetic resonance imaging (MRI) and X-ray imaging simultaneously. One major problem is that MRI k-space acquisitions typically allow parallel projection geometries, i. e. a line through the center k-space, while X-rays are associated with a divergent geometry such as fan- or cone-beam geometries. Both modalities allow different contrast mechanisms and simultaneous acquisition and overlay in the same image would be highly desirable for improved interventional guidance.

In the following, we assume to have sampled MRI projections $\mathbf{x}$ in k-space. By inverse Fourier Transform $\mathbf{F}^H$, they can be transformed into parallel projections $\mathbf{p}_{\mathrm{PB}} = \mathbf{F}^H\mathbf{x}$. Both parallel and cone-beam projections $\mathbf{p}_{\mathrm{CB}}$ are related to the volume under consideration $\mathbf{v}$ by associated projection operations $\mathbf{A}_{\mathrm{PB}}$ and $\mathbf{A}_{\mathrm{CB}}$:

$$\mathbf{p}_{\mathrm{PB}}=\mathbf{A}_{\mathrm{PB}}\mathbf{v} \qquad (16)$$

$$\mathbf{p}_{\mathrm{CB}}=\mathbf{A}_{\mathrm{CB}}\mathbf{v} \qquad (17)$$

As **v** appears in both relations, we can solve Eq. 16 for **v** using the Moore-Penrose Pseudo Inverse:

$$\mathbf{v}=\mathbf{A}_{\mathrm{PB}}^{\top}\left(\mathbf{A}_{\mathrm{PB}}\mathbf{A}_{\mathrm{PB}}^{\top}\right)^{-1}\mathbf{p}_{\mathrm{PB}}=\mathbf{A}_{\mathrm{PB}}^{\top}\left(\mathbf{A}_{\mathrm{PB}}\mathbf{A}_{\mathrm{PB}}^{\top}\right)^{-1}\mathbf{F}^{H}\mathbf{x}$$

Next, we can use **v** in Eq. 17 to yield

$$\mathbf{p}_{\mathrm{CB}}=\mathbf{A}_{\mathrm{CB}}\mathbf{A}_{\mathrm{PB}}^{\top}\left(\mathbf{A}_{\mathrm{PB}}\mathbf{A}_{\mathrm{PB}}^{\top}\right)^{-1}\mathbf{F}^{H}\mathbf{x}.$$

Note that all operations on the path from k-space to **p** $_{\mathrm{CB}}$ are known. Yet, $\left(\mathbf{A}_{\mathrm{PB}}\mathbf{A}_{\mathrm{PB}}^{\top}\right)^{-1}$ is expensive to determine and may need significant amounts of memory. As we know from reconstruction theory, this matrix often takes the form of a circulant matrix, i. e. a convolution. As such, we can approximate it with the chain of operations $\mathbf{F}^{H}\mathbf{CF}$ where **C** is a diagonal matrix. In order to add a few more degrees of freedom, we further add another diagonal operator in spatial domain **W** to yield

$$\mathbf{p}_{\mathrm{CB}}=\mathbf{A}_{\mathrm{CB}}\mathbf{A}_{\mathrm{PB}}^{\top}\mathbf{W}\mathbf{F}^{H}\mathbf{C}\mathbf{F}\mathbf{F}^{H}\mathbf{x}=\mathbf{A}_{\mathrm{CB}}\mathbf{A}_{\mathrm{PB}}^{\top}\mathbf{W}\mathbf{F}^{H}\mathbf{C}\mathbf{x} \qquad (18)$$

as parallel to cone rebinning formula. In this formulation, only **C** and **W** are unknown and need to be trained. By design both matrices are diagonal and therewith only have few unknown parameters.

Even though the training was conducted merely on numerical phantoms we can apply the learned algorithm on data acquired with a real MRI system without any loss of generality. Using only 15 parallel-beam MR projections we were able to compute a stacked fan-beam projection with both approaches. In Figure 5 the results of the analytical and learned algorithms are shown. The result of the learned algorithm has much sharper visual impression compared to the analytical approach which intrinsically suffers from ray-by-ray interpolation and thus from a blurring effect. Note that additional smoothing could be incorporated into the network by regularization of the filter or additional hard-coded filter steps at request.

4 Discussion

For many applications, we do not know which operation is required in the ideal processing pipeline. Most machine learning tasks focus either on perceptual problems or man-made classes. Therefore, we only have limited knowledge on the ideal processing chain. In many cases, the human brain seems to have identified suitable solutions. Yet, our knowledge of the human brain is incomplete and search for well-suited deep architectures is a process of trial and error. Still, deep learning has shown to be able to solve tasks that were deemed as hard or close to impossible [20].

Now that deep learning also starts addressing fields of physics and classical signal processing, we are entering areas in which we have much better understanding of the underlying processes and therefore know that kind of mathematical operations need to be present in order to solve specific problems. Yet, during the derivation of our mathematical models, we often introduce simplifications that allow more compact descriptions and a more elegant solution. Still these simplifications introduce slight errors along the way and are often compensated using heuristic correction methods [21].

In this paper, we have shown that inclusion of known operators is beneficial in terms of maximal error bounds. We demonstrated that in all cases in which we are able to use partial knowledge on the function at hand, the maximal errors that may remain after training of the network are reduced even for networks of arbitrary depth. Note that in the future tighter error bounds than the ones described in this work might be identified that are independent of the use of known operators. Yet, our error analysis is still useful, as for the case of increasing number of known operations in the network, the magnitude of the bound shrinks up to the point of identity, if all operations are known. To the knowledge of the authors, this is the first paper to attempt such a theoretical analysis of the use of known operators in neural network training.

In our experiments with CT reconstruction, we could demonstrate that we are able to tackle limited angle reconstructions using a standard filtered back-projection-type of algorithm. In fact, we only adopted weights while run-time, behaviour, and computational complexity remained unchanged. As we can map the trained algorithm back onto its original interpretation, we could also investigate shape and function of the learned weights. They demonstrated similarity to a heuristic method that could previously only be explained by intuition rather than by showing optimality. For the case, of our trained weights, we can demonstrate that they are optimal with respect to the training data.

Based on Frangi’s vesselness, we could develop a trainable network for vessel detection. In our experiments, we could demonstrate that training of this net already yields improved filters for vessel detection that are close in terms of performance with a much more complex U-Net. Further inclusion of a trainable denoising step yielded an accuracy that is statistically not distinguishable from U-Net.

As last application of our approach, we investigated rebinning of MR data to a divergent beam geometry. For this kind of rebinning procedure, a fast convolution-based algorithm was previously unknown. Prior approaches relied on ray-by-ray interpolation that is typically introducing blurring. With our hypothesis that the inverse matrix operator takes the form of a circulant matrix in spatial domain in combination with an additional multiplicative weight, we could train a new algorithm successfully. The new approach is not just computationally efficient, it also features images of a degree of sharpness that was previously not reported in literature.

Although only applications from the medical domain are shown in this paper, this does not limit the generality of our theoretical analysis. Similar problems are found in many fields, e. g. computer vision [22], image super resolution [23], or audio signal processing [24].

Obviously, known operators have been embedded into neural networks already for a long time. Already, LeCun et al. [2] suggested convolution and pooling operators. Janderberg et al. introduced differentiable spatial transformations and their resampling into deep learning [25]. Lin et al. use this for image generation [26]. Kulkarni et al. developed an entire deep convolutional graphics pipeline [27]. Zhu et al. include differentiable projectors to disentangle 3D shape information from appearance [28]. Tewari et al. integrate a differentiable model-based image generator to synthesize facial images [29]. Adler et al. shows an approach to partially learn the solution for ill-posed inverse problems[30]. Ye et al. [31] introduced the Wavelet transform as multi-scale operator, Hammernik et al. [32] mapped entire energy minimization problems onto networks, and Wu et al. even included the guided filter as layer into a network [33]. As this list could be continued with many more references, we see this as an emerging trend in deep learning. In fact, any operation that allows the computation of a sub-gradient [34] is suited to be used in combination with the back-propagation algorithm. In order to integrate a new operator, only the partial derivatives / sub-gradients with respect to its inputs and its parameters have to be implemented. This allows inclusion of a wide range of operations. To the best of our knowledge, this is the first paper giving a general argument for the effectiveness of such approaches.

Next, the introduction of a known operator is also associated with a reduction of trainable parameters. We demonstrate this in this paper in all of our experiments. This allows us to work with much fewer training data and helps us to create models that can be transferred from synthetic training data to real measured data. Zarei et al. [35] drive this approach so far that they are able to train user-dependent image denoising filters using only few clicks from alternate forced-choice experiments. Thus, we believe that known operators may be a suitable approach to problems for which only limited data is available.

At present we are unaware how to predict the benefit of using known operators before the actual experiment. Our analysis only focuses on maximum error bounds. Therefore, investigation of expected errors following for example the approach of Barron seems interesting for future work [36]. Also analysis of the bias variance trade-off seems interesting. In [37, Chapter 9] Duda and Hart already hinted at the positive effect of prior knowledge on this trade-off.

Lastly, we believe that known operators may be key in order to gain better understanding of deep networks. Similar to our experiments with Frangi-Net, we can start replacing layers with known operations and observe the effect on the performance of the network. From our theoretical analysis, we expect that inclusion of a known operation will not or only insignificantly reduce the system’s performance. This may allow us to find configurations for networks that only have few unknown operations while showing large parts that are explainable and understood. Figure 6 shows a variant of this process that is inspired by [38]. Here, we offer a set of known operations in parallel and determine their optimal superposition by training of the network. In a second step, connections with low weights can be removed to iteratively determine the optimal sequence of operations. Furthermore, any known operator sequence can also be regarded as a hypothesis for a suitable algorithm for the problem at hand. By training, we are able to validate of falsify our hypothesis similar to our example of the derivation of a new network architecture.

5 Conclusion

We believe that the use of known operators in deep networks is a promising method. In this paper, we demonstrate that the use of such reduces maximal error bounds and experimentally show an reduction in the number of trainable parameters. Furthermore, we applied this to the case of learning CT reconstruction yielding networks that are interpretable and that can be analysed with classical signal processing tools. Also mixing of deep and known operator learning is beneficial, as it allows us to build smaller networks with only 6% of the parameters of a competing U-Net while being close with respect to their performance. Lastly, the known operators can also be found using mathematical derivation of networks. While keeping large parts of the mathematical operations, we only replace inefficient or unknown operations with deep learning techniques to find entirely new imaging formulas. While all of the applications shown in this paper stem only from the medical domain, we believe that this approach is applicable to all fields of physics and signal processing which is the focus of our future work.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgment

The research leading to these results has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (ERC Grant No. 810316).

Data Availability Statement

All data in this publication are publicly available. The experiments in Section 3.1 use data of the low-dose CT challenge [39]. Section 3.2 uses the DRIVE database [40]. The data for Section 3.3 is available in a Code Ocean Capsule available at https://doi.org/10.24433/CO.8086142.v2 [41].

Code Availability Statement

The code and data for this article, along with an accompanying computational environment, are available and executable online as a Code Ocean Capsule. Experiments in Section 3.1 can be found at https://doi.org/10.24433/CO.2164960.v1 [42]. The code on learning vesselness in Section 3.2 are published at https://doi.org/10.24433/CO.5016803.v2 [41]. Code for Section 3.3 is available at https://doi.org/10.24433/CO.8086142.v2 [43]. The code capsules for the experiments in Section 3.1 and Section 3.3 were implemented using the open source framework PYRO-NN [44].

References

[1]. Niemann, H. Pattern Analysis and Understanding. Vol. 4. Springer Science & Business Media; 2013.

[2]. LeCun Y, Bengio Y. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks. 1995; 3361:1995.

[3]. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems. 2012:1097–1105.

[4]. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015; 521:436. [PubMed: 26017442]

[5]. Dong, C; Loy, CC; He, K; Tang, X. Learning a deep convolutional network for image super-resolution. European Conference on Computer Vision; Springer; 2014. 184–199.

[6]. Xie J, Xu L, Chen E. Image denoising and inpainting with deep neural networks. Advances in Neural Information Processing Systems. 2012:341–349.

[7]. Wang G, Ye JC, Mueller K, Fessler JA. Image reconstruction is a new frontier of machine learning. IEEE Transactions on Medical Imaging. 2018; 37:1289–1296. [PubMed: 29870359]

[8]. Cohen, JP, Luck, M, Honari, S. Distribution matching losses can hallucinate features in medical image translationMedical Image Computing and Computer Assisted Intervention – MICCAI 2018. Frangi, AF, Schnabel, JA, Davatzikos, C, Alberola-López, C, Fichtinger, G, editors. Springer International Publishing; Cham: 2018. 529–536.

[9]. Huang, Y, , et al. Some investigations on robustness of deep learning in limited angle tomographyMedical Image Computing and Computer Assisted Intervention – MICCAI 2018. Frangi, AF, Schnabel, JA, Davatzikos, C, Alberola-López, C, Fichtinger, G, editors. Springer International Publishing; Cham: 2018. 145–153.

[10]. Würfl, T; Ghesu, FC; Christlein, V; Maier, A. Deep learning computed tomography. International Conference on Medical Image Computing and Computer-Assisted Intervention; Springer; 2016. 432–440.

[11]. Fu, W, , et al. Frangi-Net: A Neural Network Approach to Vessel SegmentationBildverarbeitung für die Medizin 2018. Maier, A, , et al., editors. 2018. 341–346.

[12]. Maier, A; , et al. Precision Learning: Towards Use of Known Operators in Neural Networks. In: Tan, JKT, editor. 24rd International Conference on Pattern Recognition (ICPR); 2018. 183–188. URL https://www5.informatik.uni-erlangen.de/Forschung/Publikationen/2018/Maier18-PLT.pdf

[13]. Syben C, et al. Deriving neural network architectures using precision learning: Parallel-to-fan beam conversion. German Conference on Pattern Recognition (GCPR). 2018

[14]. Cybenko G. Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems. 1989; 2:303–314.

[15]. Hornik K. Approximation capabilities of multilayer feedforward networks. Neural networks. 1991; 4:251–257.

[16]. Maier A, Syben C, Lasser T, Riess C. A gentle introduction to deep learning in medical image processing. Zeitschrift für Medizinische Physik. 2019; 29:86–101. [PubMed: 30686613]

[17]. Parker DL. Optimal short scan convolution reconstruction for fan beam ct. Medical Physics. 1982; 9:254–257. [PubMed: 7087912]

[18]. Schäfer, D; van de Haar, P; Grass, M. Modified parker weights for super short scan cone beam ct. Proc 14th Int Meeting Fully Three-Dimensional Image Reconstruction Radiol Nucl Med; 2017. 49–52.

[19]. Frangi, AF; Niessen, WJ; Vincken, KL; Viergever, MA. Multiscale vessel enhancement filtering. International Conference on Medical Image Computing and Computer-Assisted Intervention; Springer; 1998. 130–137.

[20]. Silver D, et al. Mastering the game of go with deep neural networks and tree search. Nature. 2016; 529:484. [PubMed: 26819042]

[21]. Zhu B, Liu JZ, Cauley SF, Rosen BR, Rosen MS. Image reconstruction by domain-transform manifold learning. Nature. 2018; 555:487. [PubMed: 29565357]

[22]. Fürsattel P, Plank C, Maier A, Riess C. Accurate Laser Scanner to Camera Calibration with Application to Range Sensor Evaluation. IPSJ Transactions on Computer Vision and Applications. 2017; 9

[23]. Köhler T, et al. Robust Multiframe Super-Resolution Employing Iteratively Re-Weighted Minimization. IEEE Transactions on Computational Imaging. 2016; 2:42–58.

[24]. Aubreville M, et al. Deep Denoising for Hearing Aid Applications. IEEE (ed.) 16th International Workshop on Acoustic Signal Enhancement (IWAENC). 2018:361–365.

[25]. Jaderberg M, Simonyan K, Zisserman A, Kavukcuoglu K. Spatial transformer networks. Advances in Neural Information Processing Systems. 2015:2017–2025.

[26]. Lin, C-H; Yumer, E; Wang, O; Shechtman, E; Lucey, S. St-gan: Spatial transformer generative adversarial networks for image compositing. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2018.

[27]. Kulkarni TD, Whitney WF, Kohli P, Tenenbaum J. Deep convolutional inverse graphics network. Advances in Neural Information Processing Systems. 2015:2539–2547.

[28]. Zhu J-Y, et al. Visual object networks: Image generation with disentangled 3d representations. Advances in Neural Information Processing Systems. 2018:118–129.

[29]. Tewari, A; , et al. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. The IEEE International Conference on Computer Vision (ICCV) Workshops; 2017.

[30]. Adler J, Öktem O. Solving ill-posed inverse problems using iterative deep neural networks. Inverse Problems. 2017; 33

[31]. Ye JC, Han Y, Cha E. Deep convolutional framelets: A general deep learning framework for inverse problems. SIAM Journal on Imaging Sciences. 2018; 11:991–1048.

[32]. Hammernik K, et al. Learning a variational network for reconstruction of accelerated mri data. Magnetic Resonance in Medicine. 2018; 79:3055–3071. [PubMed: 29115689]

[33]. Wu, H; Zheng, S; Zhang, J; Huang, K. Fast end-to-end trainable guided filter. 2018. CoRR abs/1803.05619 URL http://arxiv.org/abs/1803.05619.1803.05619

[34]. Rockafellar, R. Convex Analysis. Princeton University Press; 1970. Princeton landmarks in mathematics and physics. URL https://books.google.de/books?id=1TiOka9bx3sC

[35]. Zarei, S, Stimpel, B, Syben, C, Maier, A. Bildverarbeitung für die Medizin 2019. Informatik aktuell; 2019. User Loss A Forced-Choice-Inspired Approach to Train Neural Networks Directly by User Interaction; 92–97. URL https://www5.informatik.uni-erlangen.de/Forschung/Publikationen/2019/Zarei19-ULA.pdf

[36]. Barron AR. Approximation and estimation bounds for artificial neural networks. Machine learning. 1994; 14:115–133.

[37]. Duda, RO, Hart, PE, Stork, DG. Pattern classification. John Wiley & Sons; 2012.

[38]. Szegedy, C; , et al. Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015. 1–9.

[39]. McCollough C. Tu-fg-207a-04: Overview of the low dose ct grand challenge. Medical Physics. 2016; 43:3759–3760.

[40]. Staal J, Abràmoff MD, Niemeijer M, Viergever MA, Van Gin-neken B. Ridge-based vessel segmentation in color images of the retina. IEEE Transactions on Medical Imaging. 2004; 23:501–509. [PubMed: 15084075]

[41]. Fu W. Frangi-net on high-resolution fundus (HRF) image database. Code Ocean. 2019; doi: 10.24433/CO.5016803.v2

[42]. Syben C, Hoffmann M. Learning CT reconstruction. Code Ocean. 2019; doi: 10.24433/CO.2164960.v1

[43]. Syben C. Deriving neural networks. Code Ocean. 2019; doi: 10.24433/CO.8086142.v2

[44]. Syben C, et al. PYRO-NN: Python reconstruction operators in neural networks. 2019

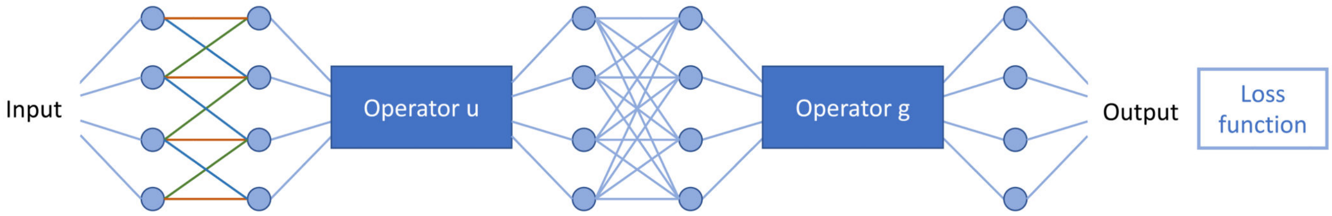


Figure 1. Schematic of the idea of known operator learning

One or more known operators (here Operator u and Operator g) are embedded into a network. Doing so, allows dramatic reduction of the number of parameters that have to be estimated during the learning process. The minimal requirement for the operator is that it must allow the computation of a sub-gradient for use in combination with the back-propagation algorithm. This requirement is met by a large class of operations.

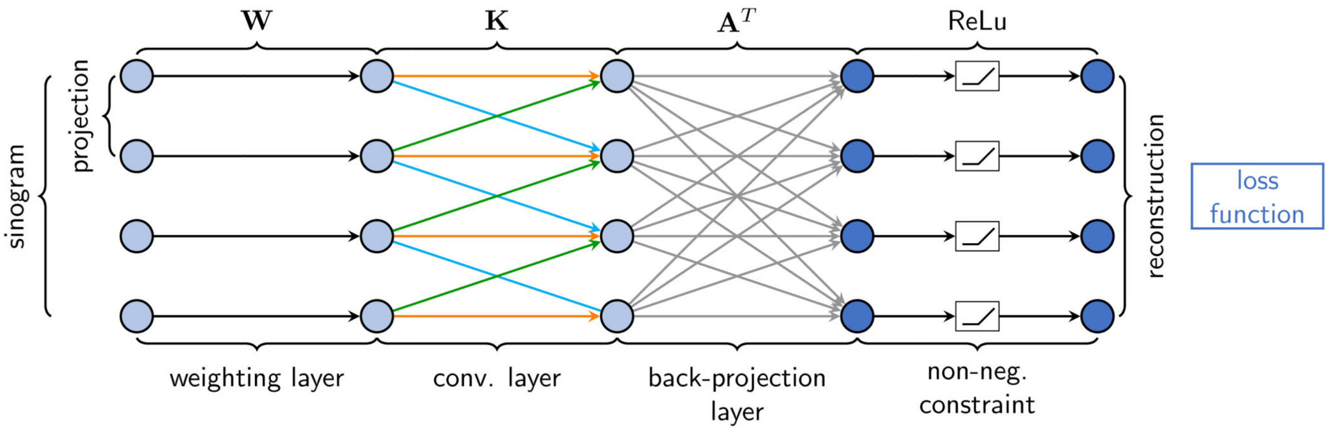
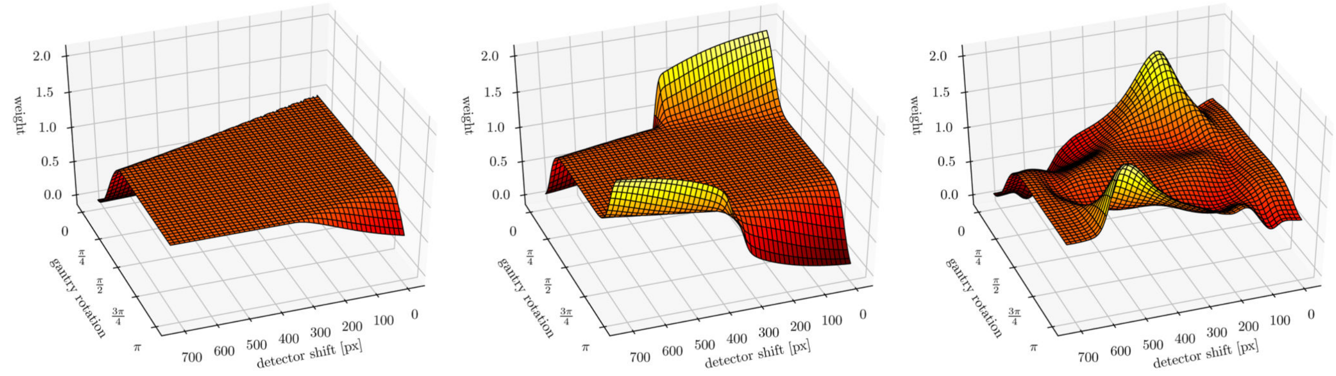


Figure 2. Deep Learning Computed Tomography

Reconstruction network for **y**= ReLu(**A** ⊤ **KWx**) from projections **x** to image **y**. As **W** is a diagonal matrix, it is merely a point-wise multiplication followed by convolution **K** and back-projection **A** ⊤.



(a) Parker Weights (b) Schäfer Weights (c) Trained Weights

Figure 3. Improved interpretability in deep networks

The trained reconstruction algorithm can be mapped back into its original interpretation. Hence, we can compare them to reconstruction weights after (a) Parker [17] and (b) Schäfer [18]. (c) expresses significant similarity to (b) which is also able to compensate for the loss of mass. While (b) was only arrived at heuristically (c) can be shown to be data optimal here.

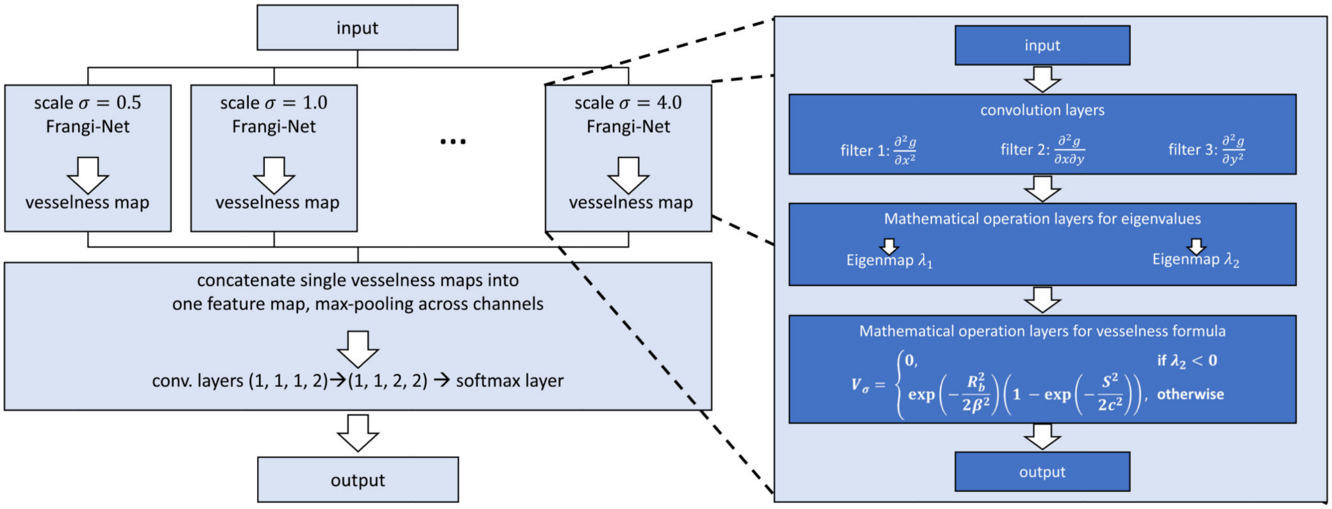
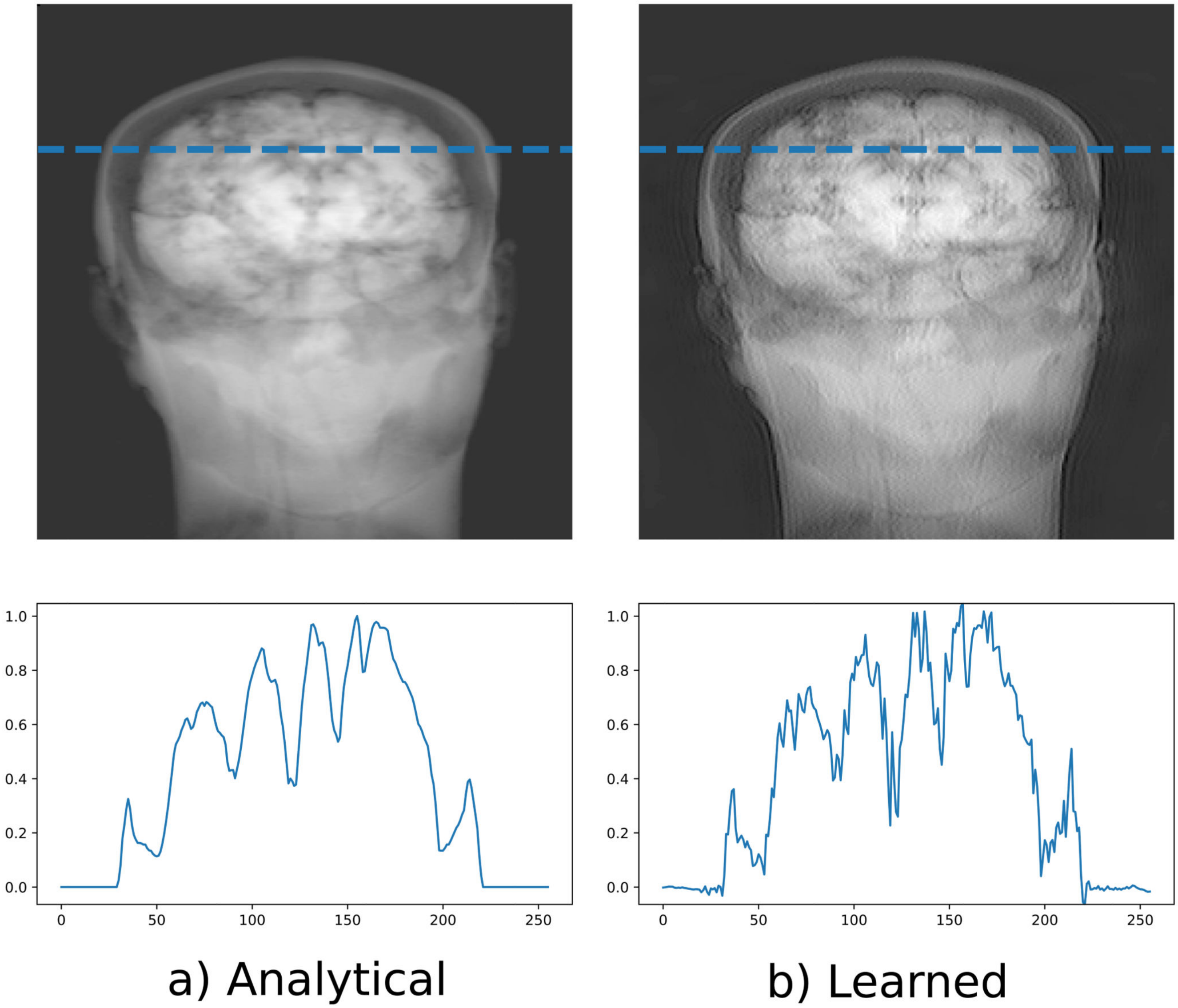

input
scale $\sigma = 0.5$
Frangi-Net
vesselness map
scale $\sigma = 1.0$
Frangi-Net
vesselness map
...
scale $\sigma = 4.0$
Frangi-Net
vesselness map
concatenate single vesselness maps into
one feature map, max-pooling across channels
conv. layers (1, 1, 1, 2)→(1, 1, 2, 2) → softmax layer
output
input
convolution layers
filter 1: $\frac{\partial^2 g}{\partial x^2}$
filter 2: $\frac{\partial^2 g}{\partial x \partial y}$
filter 3: $\frac{\partial^2 g}{\partial y^2}$
Mathematical operation layers for eigenvalues
Eigenmap λ_1
Eigenmap λ_2
Mathematical operation layers for vesselness formula
$V_\sigma = \begin{cases} 0, & \text{if } \lambda_2 < 0 \\ \exp\left(-\frac{R_b^2}{2\beta^2}\right)\left(1 - \exp\left(-\frac{S^2}{2c^2}\right)\right), & \text{otherwise} \end{cases}$
output

Figure 4. Architecture of Frangi-Net over 8 scales σ

For each single-scale a Frangi-Net computes spatial derivatives $\frac{\partial^2 g}{\partial x^2}$, $\frac{\partial^2 g}{\partial x \partial y}$, and $\frac{\partial^2 g}{\partial y^2}$. These are used to form a Hessian matrix of which eigenvalues λ_1 and λ_2 are extracted. Both are used to compute structureness S and blobness R_b which are required to compute the final vesselness at each pixel V_σ.



a) Analytical

b) Learned

Figure 5. Classical analytical rebinning vs. derived neural networks

The trained rebinning algorithm can directly applied to real MR projection data. Parallel-beam MR projection data is rebinned to a stacked fan-beam geometry with the analytical (a) and the learned algorithm (b). Note that the result of the learned method is much sharper as it avoids ray-by-ray interpolation.

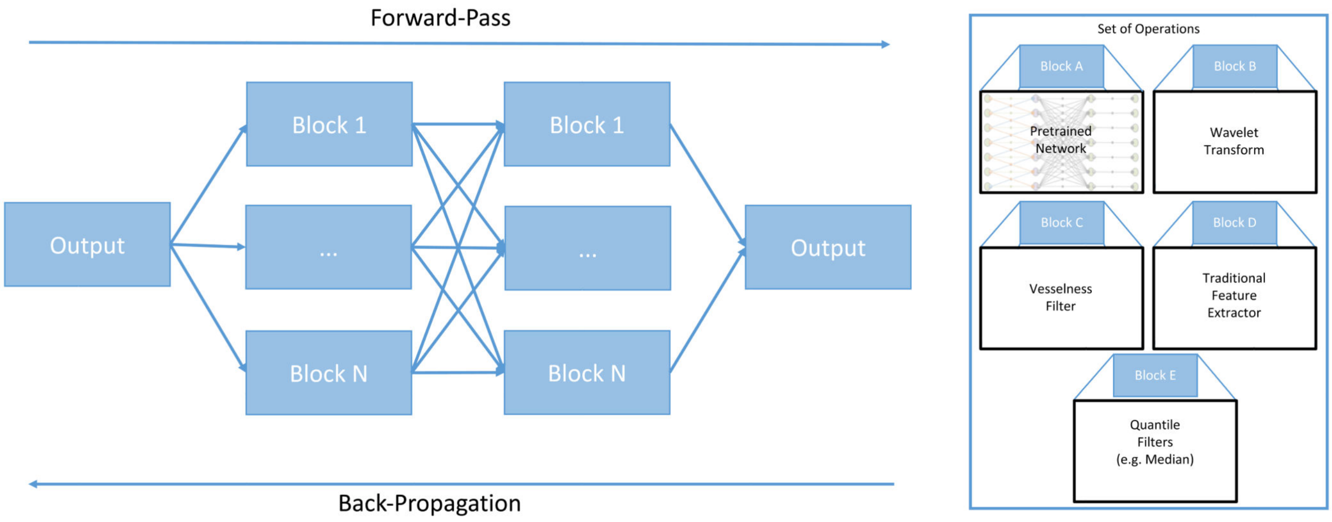


Figure 6. Towards operator discovery and sequence analysis

We hypothesise that Known Operator Learning may also be used to disentangle information efficiently. Offering several operators in parallel allows the network to find the best sequence of operations during the training process. In a subsequent step, blocks can be removed step-by-step to determine the minimal block networks.