Solving Inverse Problems with Deep Learning

–

Lecture 2: Neural Networks in practice and post-processing

Andreas Hauptmann

Jesus College, Cambridge

21 September 2023

Outline

Outline

Training the network: Optimisation

We have discussed how well we can approximate certain functions with a given class and in particular that deep networks are more efficient in representing general classes with less weights: We still need to find this approximation

$\Rightarrow$ which leads to the optimisation problem, referred to as *training task*:

- Consider now fixed architectures, that is we have a class $\mathbb{L}$ with a fixed parameter set Θ. That is, we set here again $\mathbb{L} := \{\Lambda_\theta\}_{\theta \in \Theta}$.
- The goal is to find $\theta \in \Theta$ through optimisation
- In contrast to allowing for varying parameterisations of variable width, as considered in the approximation section.

Recall: Empirical risk minimisation

Recall: the total risk does not only depend on the approximation error, but also on how well we can train the network weights $\rightarrow$ We have only access to the empirical risk.

- Given a training set of pairs $\{v^{(i)}, u^{(i)}\}_{i=1}^{N}$.
- Find the optimal parameter θ^* that minimises the empirical risk Φ for fixed architectures $\Lambda_\theta \colon V \rightarrow U$ in the class $\mathbb{L}$:

$$\theta^* = \arg\min_{\theta \in \Theta} \Phi(\Lambda_\theta) = \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} L_U\big(\Lambda_\theta(v^{(i)}), u^{(i)}\big). \tag{1.1}$$

Remember: Ideally we would like to find a minimiser over the full risk $\widehat{\Phi}$, i.e., over infinite data pairs to provide the best available approximation error. However, due to finite data we only have access to the empirical risk Φ which additionally limits approximation properties of the found estimator Λ_{θ^*}. We refer to this discrepancy as the *training error*.

Gradient descent

- For neural networks this optimisation is widely performed by first-order methods and in particular variations of gradient descent, where the gradient is computed with respect to the parameters.
- A gradient descent scheme for eq. (1.1) is given as

$$\theta_{i+1} := \theta_i - \eta \nabla_\theta \Phi(\Lambda_\theta(\cdot)),$$

- $\eta > 0$ is a step-length, often referred to as the *learning rate*.
- We need to compute the gradient $\nabla_\theta \Phi(\Lambda_\theta(\cdot))$ with respect to all samples in the training data $\{v^{(i)}, u^{(i)}\}_{i=1}^N$.

Stochastic gradient descent

- Computing the full gradient is computationally intractable for large amounts of data. (large sets of high resolution CT scans)
- Empirical evidence suggests that basic gradient descent tend to get stuck in sub-optimal local minima.
- Consequently, we consider stochastic variants, we compute updates as

$$\theta_{i+1} := \theta_i - \eta g_i,$$

  where g_i represents a descent direction with respect to only a subset of the data.
- The samples (training pairs) over which the gradient is computed are chosen randomly.
- Under suitable random sampling one can then show that in expectation we indeed obtain the gradient at the i-th iteration

$$\mathbb{E}[g_i|\theta_i] = \nabla_\theta \Phi(\Lambda_{\theta_i}).$$

- In practice, to compute the directions g_i so-called *batches* of several training pairs are chosen, instead of single samples.

A usual convergence result

The usual convergence results we obtain for stochastic gradient descent provide convergence bounds in expectation. In the simplest case assuming differentiability (in the nonlinearity) and convex Φ, we obtain a convergence rate to the optimal parameter θ^* as

$$\mathbb{E}\big[\Phi(\Lambda_{\overline{\theta}})\big] - \Phi(\Lambda_{\theta^*}) \leq \frac{1}{\sqrt{T}} \text{ where } \overline{\theta} = \frac{1}{T}\sum_{i=1}^{T} \theta_i \tag{1.2}$$

- The convergence rate follows $1/\sqrt{T}$ with number of iterations. This is not particularly encouraging, but computation of the direction g_i is assumed to be fast.
- Keep in mind that we do not expect to compute the best approximator possible (only access to the empirical risk), and there is no need to iterate until convergence in practice.

Some remarks on generalisation

Generalisation is difficult to study, there are a few attempts but mostly limited to classification problems:

- Margin maximisation and implicit bias - assumes separable data (no regression problems).
- Rademacher complexity measures generalisation bounds depending on network size (with very pessimistic bounds)

Exception: The work (de Hoop, Lassas, Wong, 2022) does provide a result on approximating nonlinear operators in inverse problems with neural networks and provides (pessimistic) complexity dependent bounds.
$\Rightarrow$ Most generalisation results are empirical, but show good results on regression tasks.

Outline

Deep networks: recap

We have introduced neural networks in their most basic form as a classic feed forward fully connected network. To discuss what that means and how we can include other design principles let us first recall the basic deep neural network $\Lambda : V \to U$:

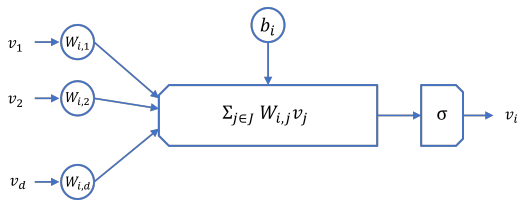
$$\Lambda_\theta(\mathbf{v}) = \mathbf{a}^{\mathrm{T}} \sigma_L(\mathrm{W}^{(L)}\, \sigma^{(L-1)}(\cdots \sigma_1(\mathrm{W}^{(1)}\, \mathbf{v} + \mathrm{b}^{(1)}) \cdots) + \mathrm{b}^{(L)}). \tag{2.1}$$

- Interpret the above network as a series of affine linear transformations followed by a nonlinear function.
- We can design already a variety of different mappings following this simple strategy.
- Do not require the linear transformations to map between the same spaces in each layer.
- Allow for connections between non-neighbouring layers $\Rightarrow$ design more efficient architectures or change the *learning problem*

Dense/fully connected layers

(fully connected layer) In a classic neural networks each weight matrix $\mathrm{W}\colon \mathbb{R}^d \to \mathbb{R}^m$ is chosen as a dense matrix, i.e., all matrix entries are learnable parameters: all the inputs $\mathbf{v} \in \mathbb{R}^d$ are related to all the outputs $\mathbf{u} \in \mathbb{R}^m$.

(neuron) The mapping from the vector v to the output element u_i is the i-th neuron:



- In inverse problems, the input, f or g, must be vectorised for the input layer.
- Each neuron aggregates data from all inputs to a single element in the output vector. This may lead to highly redundant representations and encodes a strong locality (possibly undesired).

Why convolutional layers

- The values in image pixels are related to neighbouring pixels $\Rightarrow$ Take spatial relations, and especially local relations, into account.
- Convolutions, especially with small filters - say 3×3 - are a popular and very successful choice: translation equivariant and hence encode the same local features under translation of the image and are agnostic of image size.
- Localised filters lead to linear mappings (the matrix W) with very sparse structure that can be efficiently implemented without an explicit matrix representation.
- Instead of learning the whole matrix W, one needs to learn only the filter coefficients. Usually multiple such filters are used, each one referred to as a *channel* here.

$\Rightarrow$ Convolutional Neural Networks (CNN)

Comparing the structure

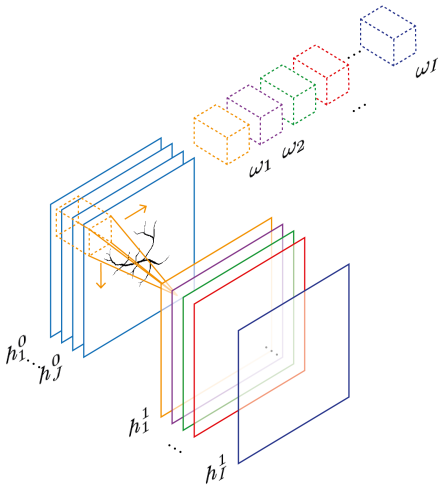
In imaging, the input is either a single or multichannel image $h^0 = \{h_j^0 \in \mathbb{R}^{m\times m}\}_{j=1}^{J} \in \mathbb{R}^{m\times m\times J}$, where $j \in \mathcal{J} = \{1, \ldots, J\}$ denotes the input channels.

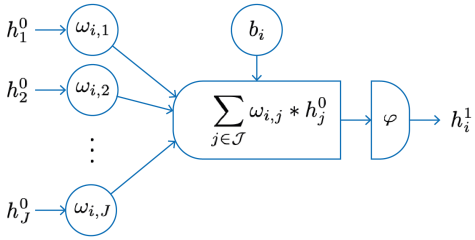
The output $h^1 = \{h_i^1 \in \mathbb{R}^{m\times m}\}_{i=1}^{I} \in \mathbb{R}^{m\times m\times I}$ where $i \in \mathcal{I} = \{1, \ldots, I\}$ denotes the output channels.

The affine linear mapping is then defined by a set of I filters $\omega_i \in \mathbb{R}^{m_\omega \times m_\omega \times J}$ where $\omega_i = \{\omega_{i,j} \in \mathbb{R}^{m_\omega \times m_\omega}\}_{j=1}^{J}$, and biases $b \in \mathbb{R}^I$, where each output channel has one bias. The convolutional layer that maps between the two multichannel images h^0 and h^1 is then defined for each channel as

$$h_i^1 = \varphi\left(\sum_{j\in\mathcal{J}} \omega_{i,j} * h_j^0 + b_i\right) \text{ for each } i \in \mathcal{I}, \tag{2.2}$$

Graphical Illustration





Typical network elements

Pooling: The process of reducing dimension of the input layer by a local (low-pass) filtering: Average, maximum, mean pooling

Residual connections: A connection between (multiple) layers as residual addition:

$$h^L = h^{L-i} + \Lambda_\theta(h^{L-i}).$$

Skip connections: Similar to a residual connection, except that the two layers h^L, h^{L-i} are concatenated by channels into a larger tensor.

$\Rightarrow$ Both, residual and skip connections, help in the optimisation to allow for the gradients to propagate "around" certain layers.

Outline

Learning a reconstruction operator

- We aim to solve the inverse problem $\mathcal{A} f = g$, given $g \in Y$.
- For (many) ill-posed problems the inverse is discontinuous $\Rightarrow$ Rather learn a regularized map
- Here we follow the paradigm to learn a *Reconstruction Operator* $\mathcal{R} : Y \to X$ instead

$\Rightarrow$ The underlying map we are trying to approximate is well-posed, such as a regularised reconstruction

$\Rightarrow$ This learned reconstruction operator may consist of learned and model-based parts

Classifying learned reconstructions

We classify learning a reconstruction operator into two paradigms:

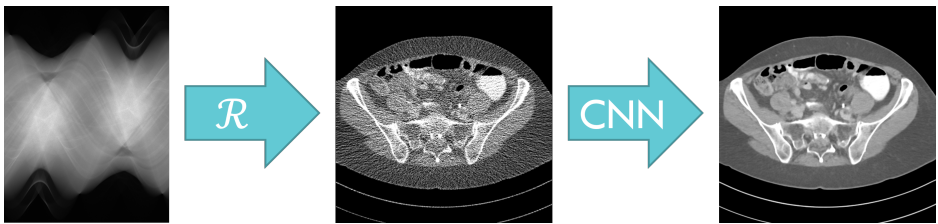
Uncoupled This section considers a setting where the training process (of the network) is decoupled from the model for how data is generated in the inverse problem, i.e., training does not involve evaluating the forward operator or the adjoint.

Learned iterative schemes Here network components and forward/adjoint are intertwined and training the reconstructions operator necessarily requires evaluation of the model.

Outline

Post-processing reconstructions

- Improve an initial reconstruction obtained from noisy, possibly under-sampled, measurement data.
- The neural network $\Lambda_\theta : X \to X$ maps image to image and it is trained to improve the reconstruction.
- We concentrate on the supervised setting: pairs of high- and low-quality reconstructions are available, such as low-dose and high-dose CT scans.
- The high-dose reconstruction provides (ground-truth) f and the low-dose measurements provides the data g to obtain an initial reconstruction by applying a handcrafted reconstruction operator $\mathcal{R}\colon Y \to X$.



Formalise the training task

(Supervised data) Assume we are given training pairs $\{g^{(i)}, f^{(i)}\}_{i=1}^{n} \subset X \times Y$, where $f^{(i)}$ refers to the ground-truth and matching noisy measurement pairs $g^{(i)}$.

(Uncoupled post-processing) Compute an initial reconstruction $\mathcal{R}(g^{(i)})$, which is noise corrupted and/or shows undersampling artefacts.

(Restoration network) $\Lambda_\theta \colon X \to X$ to remove noise and reconstruction artefacts generated by $\mathcal{R} \colon Y \to X$.

- The learning problem for the network itself is then formulated as the task to find the optimal parameter by empirical risk minimisation:

$$\theta^* \in \operatorname*{arg\,min}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} L_X\big((\Lambda_\theta \circ \mathcal{R})(g^{(i)}), f^{(i)}\big) \tag{3.1}$$

for some loss $L_X \colon X \times X \to \mathbb{R}$.

Note: Pre-compute $\mathcal{R}(g^{(i)})$, i.e., the reconstruction operator $\mathcal{R}$ does not enter into the training procedure.

Statistical view

- View as an approximation of the regression function that is given by the Bayes estimator, so $\Lambda_{\theta^*} \approx \widehat{\Lambda}$ where

$$\widehat{\Lambda} = \operatorname*{arg\,min}_{\Lambda\colon X \to X} \mathbb{E}_{(\mathsf{f},\mathsf{g})}\Big[L_X\big((\Lambda \circ \mathcal{R})(\mathsf{g}), \mathsf{f}\big)\Big]. \tag{3.2}$$

- Naturally, we cannot expect to compute Λ^* (risk vs. empirical risk), so we instead settle with the empirical risk eq. (3.1).
- The formulation eq. (3.2) specifies what one seeks to compute: In particular, if the loss $L_X\colon X \times X \to \mathbb{R}$ is the squared 2-norm, then $\Lambda^*(g)$ corresponds to the conditional mean given the reconstruction $\mathcal{R}(g)$, i.e.,

$$(\Lambda^* \circ \mathcal{R})(g) = \mathbb{E}\big[\mathsf{f} \mid \widetilde{\mathsf{f}} = \mathcal{R}(g)\big] \quad \text{for } g \in Y.$$

Interpretation as learned reconstruction operator

- Interpret the reconstruction process as a whole. Then the learned reconstruction operator $\mathcal{R}_\theta : Y \to X$ is defined by the composition

$$\mathcal{R}_\theta(g) = (\Lambda_\theta \circ \mathcal{R})(g) = \Lambda_\theta(\widetilde{f}). \tag{3.3}$$

- We could write the learning problem in for the reconstruction operator with the loss functions $L(\mathcal{R}(g), f)$.
- Joint training could be viable (but expensive): Optimising parameters in $\mathcal{A}^\dagger$.
- We will primarily use the formulation (3.1) of the training problem as approximating a regression function in the reconstruction space X.

Outline

U-Net: Backbone architecture for image processing

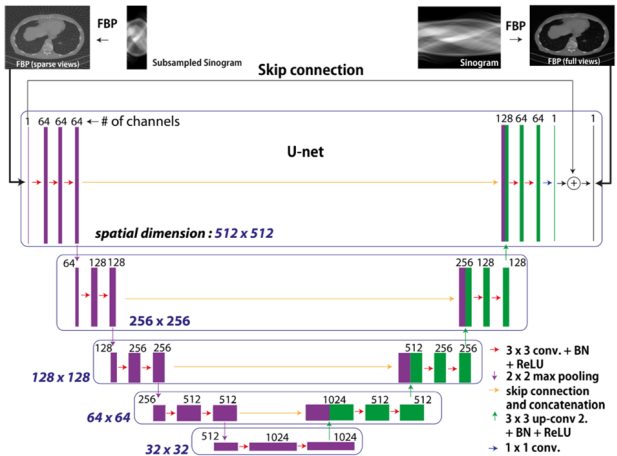
Originally proposed for image segmentation tasks, the U-Net and its variants have seen a wide application to various applications in imaging: from denoising and undersampling artefact removal to more advanced compensation of limited-view artefacts and inpainting applications, as well as in parts for generating networks. This can be attributed to its high expressiveness and particular (empirical) stability in the training procedure.

(Ronneberger, Fischer, and Brox, 2015)

FBPConvNet

[JIN ET AL., *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 2017]

- Reconstruction by FBP
- Learned denoising of reconstructed image
- Residual U-Net architecture



Formalising the post-processing approach

- The aim: define a network Λ_θ that processes an initial reconstruction $\widetilde{f} = \mathcal{A}^\dagger g$ such that the ground-truth image f is approximated, i.e., $f \approx \Lambda_\theta(\widetilde{f})$.
- Nevertheless, the problem is often reformulated as a residual problem

$$f \approx (\mathrm{Id} + \Lambda_\theta)\widetilde{f} = \widetilde{f} + \Lambda_\theta(\widetilde{f}), \tag{3.4}$$

  that means the network is trained to provide a residual $r = f - \widetilde{f}$ correction to the initial reconstruction.
- Network identifies and extracts noise and artifacts to be removed from the image. We refer to as residual networks or residual learning problem.

Quality of initial reconstruction: Primary limitation

- In the post-processing approach the reconstruction result depend only on the quality of the initial reconstruction (conditioned on the reconstruction) and the training data provided.
- Dependent on the capability of the network to correct corrupted features.
- Primary limitation: we cannot guarantee that the reconstructed images are consistent with the data, i.e., that

$$\| \mathcal{A} \Lambda_\theta(\widetilde{f}) - g \|_Y \text{ is small}$$

Example: Cardiovascular Magnetic Resonance Imaging

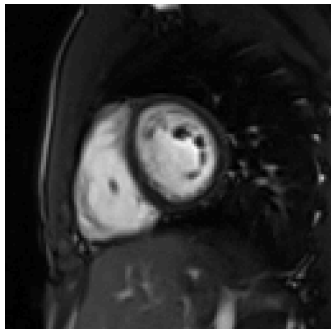
Forward model: Fourier transform $\mathcal{F}_k$

Data given in k-space: $y = \mathcal{F}_k x$

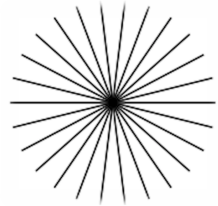
Reconstruction by inverse Fourier transform: $\mathcal{F}_k^{-1} y$

- Gold-standard taken under breath-hold, $\sim$10 seconds
- Need for real-time imaging for pediatric imaging

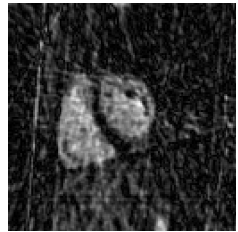
Gold-standard breath-hold



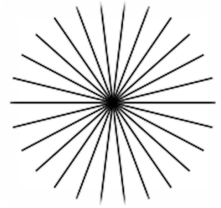
Real-time imaging in CMR and deep learning



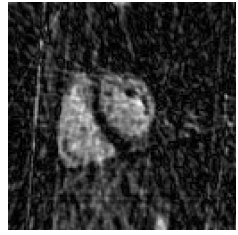
Direct reconstruction
F_k^{-1}



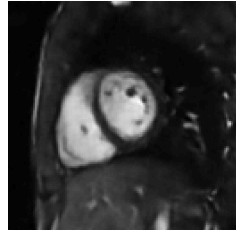
Real-time imaging in CMR and deep learning



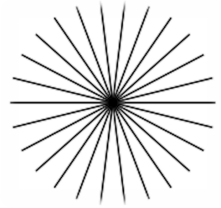
Direct reconstruction
F_k^{-1}



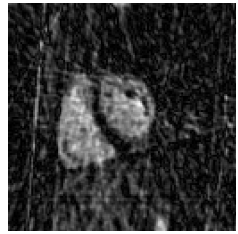
CNN



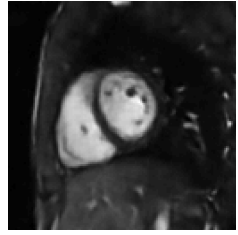
Real-time imaging in CMR and deep learning



Direct reconstruction
F_k^{-1}



CNN



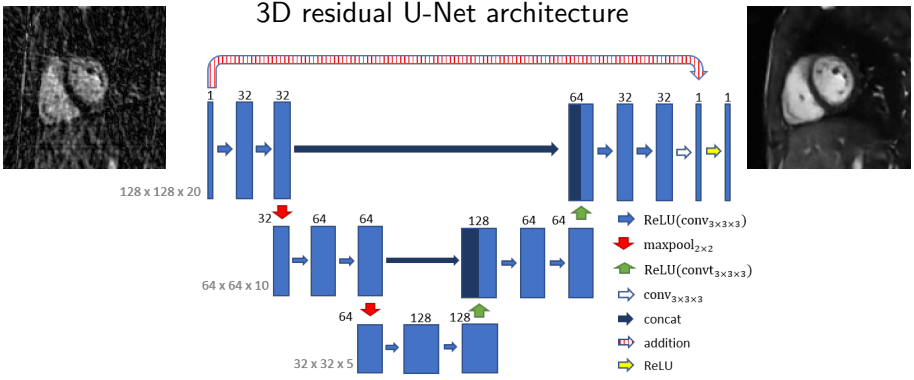
Pro:

- Fast reconstruction and post-processing
- Training with magnitude images possible

Contra:

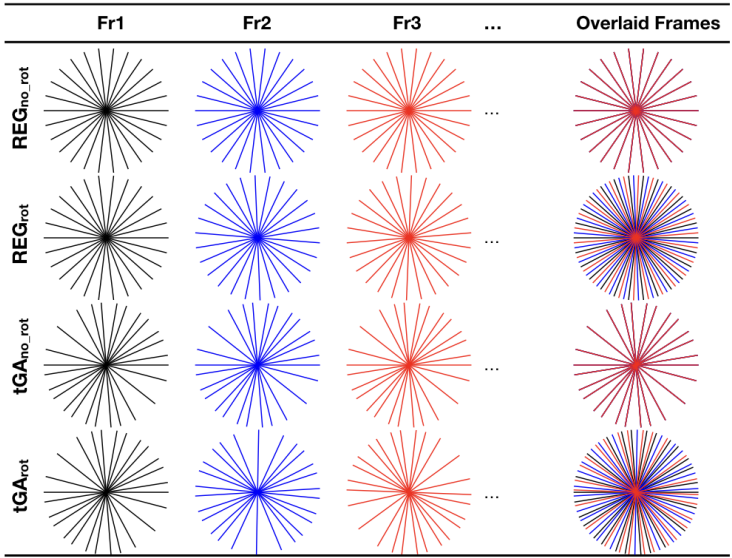
- Many samples needed for training
- Artifacts ideally to be noise-like

Network architecture

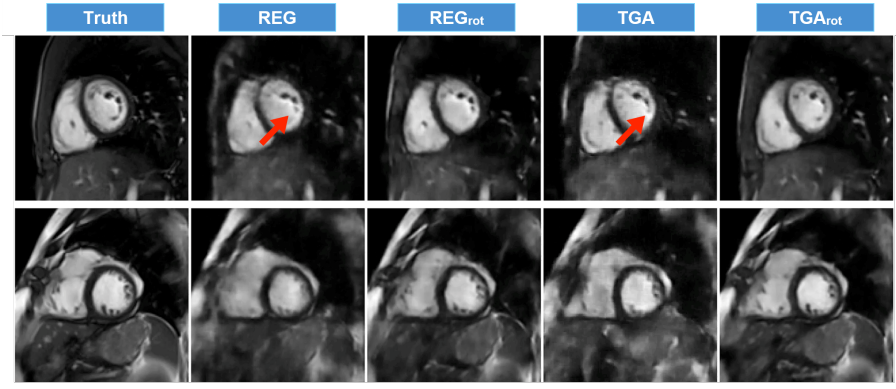


- Trained to minimise ℓ^2-loss of output to the desired ground truth:
- Trained with 2276 (2D+time) data sets from $\sim$250 patients
- Real-time data 13x accelerated, simulated from magnitude images
- Training time 12 hours

Comparison of radial sampling strategies



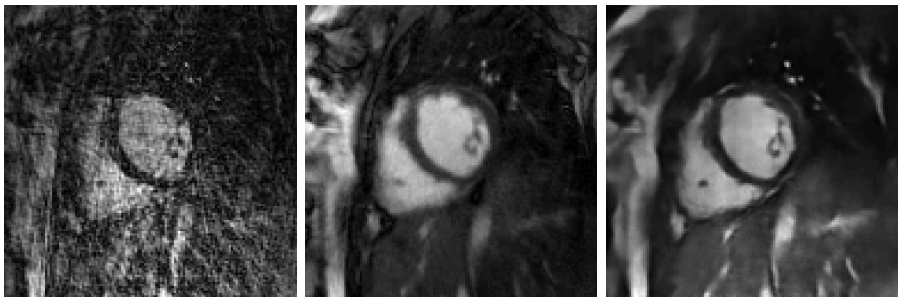
Comparison of sampling strategies - reconstructions



RMSE ($x10^{-2}$)	8.0 ± 1.5 *	4.9 ± 1.0 *	8.4 ± 1.6 *	4.2 ± 1.3
SSIM	0.64 ± 0.04 *	0.83 ± 0.03 *	0.63 ± 0.05 *	0.87 ± 0.03

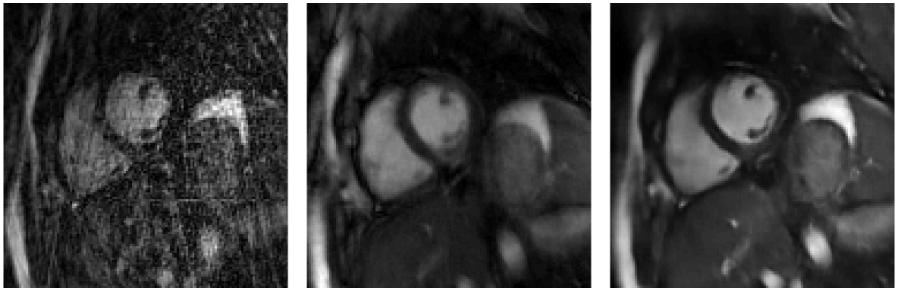
Application to prospective data

Prospective data GRASP recon. Post-processed image



Application to prospective data: Extra

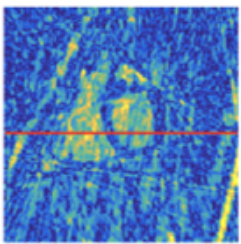
Prospective data | GRASP recon. | Post-processed image

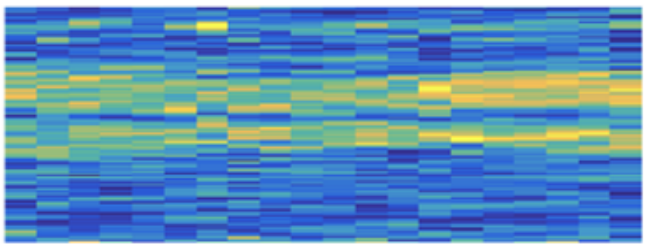


When to consider post-processing?

- If artefacts are incoherent/noise-like in time, CNN primarily needs to learn interpolation in time. $\Rightarrow$ Denoising task
- If this is not the case, CNN needs to rely on features learned from the training data. $\Rightarrow$ Inpainting task

$\Rightarrow$ Domain and model knowledge needed to design a robust learning task!





Deep Convolutional Framelets

The two papers (Ye, Han, Cha, 2018) and (Han, Ye, 2018) aim to provide some insights into the expressive power of U-Net architectures.

- Relation to multiresolution analysis provided a first intuition.
- A perfect reconstruction condition can be met in finite dimensions under some architectural improvements:
  The question is whether a given network architecture can represent a given digitised signal $\mathbf{f} \in \mathbb{R}^n$ error free. This is similar to the study of frames in signal processing which asks whether an existing set of vectors $\{e_k\}$ in the vector space $\mathbb{R}^n$ can be used to express any element $\mathbf{f} \in \mathbb{R}^n$ by a linear combination given the coefficients c_k as
  $$f = \sum_k c_k e_k. \tag{3.5}$$
- In the case of frames, in contrast to a basis, the set $\{e_k\}$ is overcomplete and there does not exist a unique decomposition of f.

Residual vs. direct learning

The takeaways from (Ye, Han, Cha, 2018) and (Han, Ye, 2018) are:

- Learned encoding and hand-crafted pooling operations can indeed guarantee to represent a signal in a learned basis and reconstruct it perfectly, in finite dimensions.

  Residual learning: If the underlying signal has a lower dimensional structure compared to the artefacts to be removed, than the residual approach can be interpreted as learning an annihilating filter that only leaves the higher dimensional artefacts, this can be compared to a low-pass filtering.

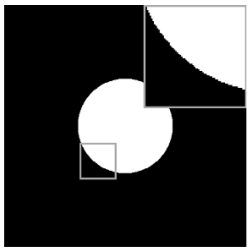
  Direct learning If the residual is of lower dimensional structure, then a direct learning approach is recommended. The second case is relevant for limited-view problems, where the network effectively needs to generate the missing regions.

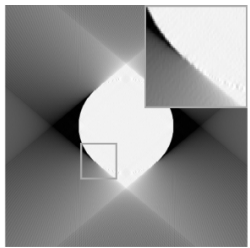
- Denoising vs. inpainting

Learning the visible vs. invisible

The study by (Bubba et al., 2019) follows a similar philosophy for limited-angel CT:
Separate the reconstruction in parts contained in g (visible) and does are not (invisible) $\Rightarrow$ *Only learn the invisible part*

- Microlocal analysis used to separate the wavefront set of the data into visible and invisible parts
- Shearlets representation of the reconstruction can be similarly separated into visible and invisible parts
- Train network to inpaint the missing shearlet coefficients
- Reconstruct from full shearlet coefficients: Maintain approximate data-consistency

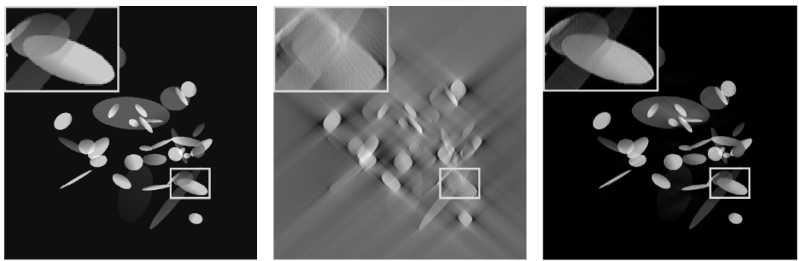




Learning the visible vs. invisible

The study by (Bubba et al., 2019) follows a similar philosophy for limited-angel CT:
Separate the reconstruction in parts contained in g (visible) and does are not (invisible) $\Rightarrow$ *Only learn the invisible part*

- Microlocal analysis used to separate the wavefront set of the data into visible and invisible parts
- Shearlets representation of the reconstruction can be similarly separated into visible and invisible parts
- Train network to inpaint the missing shearlet coefficients
- Reconstruct from full shearlet coefficients: Maintain approximate data-consistency



Post-processing as regularisation strategy: Nullspace networks

- The reconstruction operator is usually parametrised as $\mathcal{R}_\theta := \Lambda_\theta \circ \mathcal{B}$, where $\mathcal{B}\colon \mathsf{Y} \to \mathsf{X}$ denotes a classical reconstruction method (with no or few tune-able parameters, e.g., FBP or TV in X-ray CT).
- The reconstructions fail to satisfy the data-consistency criterion: a small value of $\left\| \mathcal{A} f^\dagger - g^\delta \right\|$ does not necessarily imply a small value for the data-fidelity term $\left\| \mathcal{A} \Lambda_\theta(f^\dagger) - g^\delta \right\|$ corresponding to the output of Λ_θ, where $\mathsf{x}^\dagger = \mathcal{B}(g^\delta)$.
- (Schwab, Antholzer, Haltmeier, 2019) suggested to parametrise the operator Λ_θ as $\Lambda_\theta = \mathrm{Id} + \left(\mathrm{Id} - \mathcal{A}^\dagger \mathcal{A}\right) \mathcal{Q}_\theta$, where $\mathcal{Q}_\theta$ is a Lipschitz-continuous DNN. Since $\left(\mathrm{Id} - \mathcal{A}^\dagger \mathcal{A}\right)$ is the projection operator onto the null-space of $\mathcal{A}$, the operator Λ_θ (referred to as null-space network) always satisfies $\mathcal{A} \Lambda_\theta(\mathsf{x}^\dagger) = \mathcal{A} \mathsf{x}^\dagger$, ensuring that the output of Λ_θ explains the observed data.
- Null-space networks are shown to provide convergent regularisation schemes, when inherited from $\mathcal{B}$.

Conclusion on U-Net post-processing

Data-driven methods linked to signal processing theory

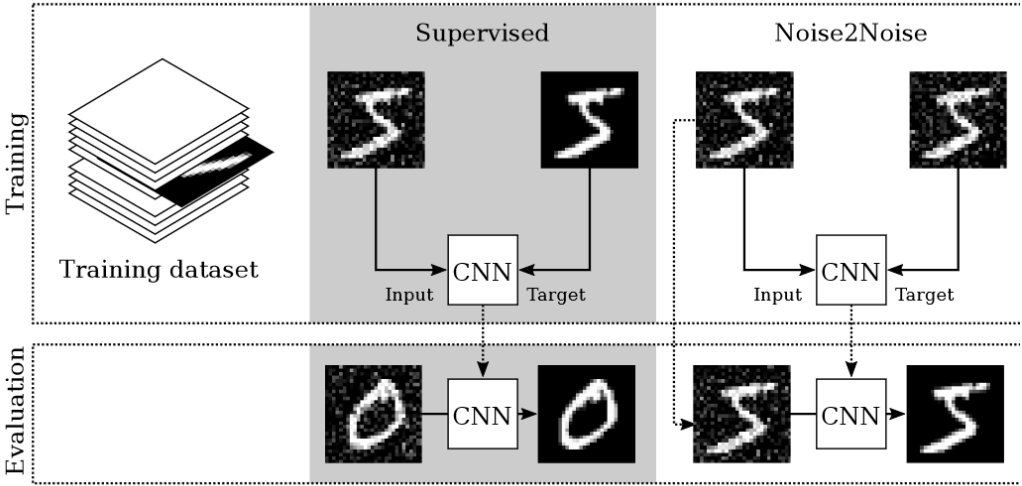
"The success of deep learning stems not from a magical black box, but rather from the power of a novel signal representation using a nonlocal basis combined with a data-driven local basis, which is indeed a natural extension of classical signal processing theory." Quoted from (Ye, Han, Cha, 2018).

Outline

Unsupervised data

- In medical imaging it would be ideal if we could use only noisy or undersampled measurements for the training process.
- If only $g \in Y$ are available for training, we need speak of *unsupervised data* (in Inverse Problems).
- We need to formulate a training task / loss function that can measure correctness.
- We observe that that convolutional neural networks will first learn to represent low frequency features before being able to represent the high frequency (noise) structures. $\Rightarrow$ We can exploit this when one has the possibility to sample noise $\eta \sim \pi_{\text{noise}}$ to create multiple instantiations of either noisy images f^η or data g^η.

Noise2Noise - Illustration



Proposed by (Lehtinen et al., 2018).
Image from (Hendriksen, Pelt, Batenburg, 2020).

Noise2Noise: Denoising images

- Given two noisy realisations of the same image $\mathbf{f}^{\eta_1}$ and $\mathbf{f}^{\eta_2}$ with independent draws of $\eta_i \sim \pi_{\text{noise}}$
- Reformulate the supervised training setting to

$$\theta^* = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} L\left(\Lambda_\theta(\mathbf{f}_i^{\eta_1}), \mathbf{f}_i^{\eta_2}\right). \tag{3.6}$$

- If the noise is zero-mean $\mathbb{E}[\eta] = 0$, then the expected prediction error

$$\Lambda^* = \arg\min_{\Lambda} \mathbb{E}_{\mathbf{f},\eta_1,\eta_2}\left[L\Big(\Lambda(\mathbf{f} + \eta_1), \mathbf{f} + \eta_2\Big)\right] \tag{3.7}$$

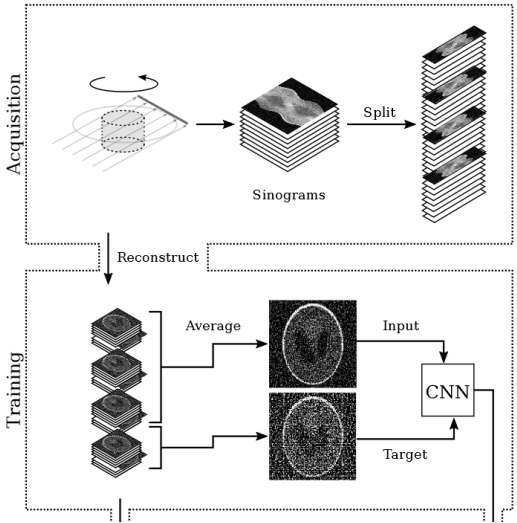
  is minimised by the same (regression) network as in the supervised case.
- Note that in practice, minimising (3.6) does not guarantee that we can obtain Λ^*.
- Empirical experience shows that the performance of denoisers trained by (3.6) supervised manner.
- Unfortunately, the need for two independent noisy measurements is not practical.

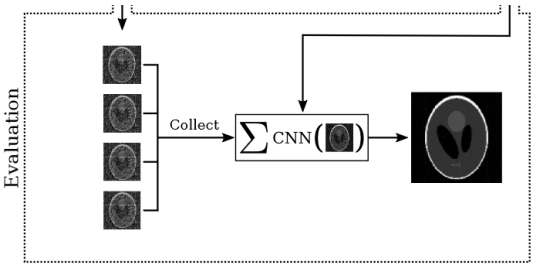
Noise2: Inverse?

In the context of inverse problems, the above concept of utilising two uncorrelated noise instantiations can be highly useful $\Rightarrow$ it eliminates the need to obtain ground-truth images.

- To obtain these two measurements we can now make use of the characteristics of the forward operator, and by that the reconstruction operator, to obtain two reconstructions with uncorrelated noise.
- When $\mathcal{A}$ is given by the Radon transform, or any other ray transform, then for each angle ω the measurement $g(\omega, s) = (\mathcal{A} f)(\omega, s) + \eta$ the noise $\eta \sim \pi_{\text{noise}}$ will be independent.
- Use to obtain two reconstructions with uncorrelated noise to train an unsupervised denoiser as in the Noise2Noise case.

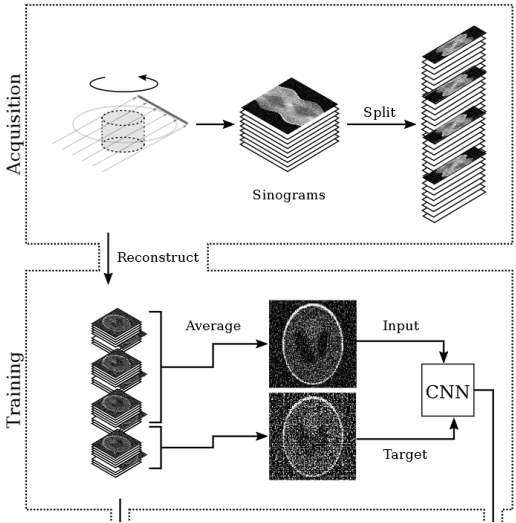
Noise2Inverse - Illustration

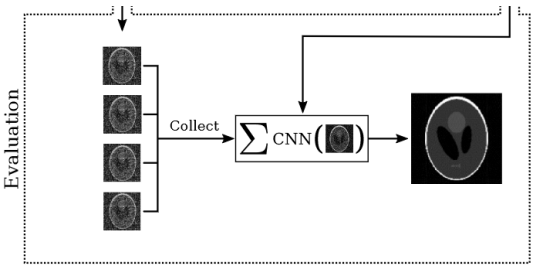




From (Hendriksen, Pelt, Batenburg, 2020).

Noise2Inverse - Illustration





From (Hendriksen, Pelt, Batenburg, 2020).

Formalising Noise2Inverse

- Given full measured sinogram $g(\omega, s)$ for $\omega \in \Phi$, where Φ denotes the full angular range according to the chosen measurement geometry.
- We split the measurement angles into randomly drawn bins $\tilde{\Phi} \subset \Phi$ and assign its (non intersecting) complement $\tilde{\Phi}^C$.
- The reconstructions are obtained by filtered backprojection (or any other linear reconstruction operator) as $f_{\tilde{\Phi}} = \mathcal{A}^\dagger g_{\tilde{\Phi}}$ and $f_{\tilde{\Phi}^C} = \mathcal{A}^\dagger g_{\tilde{\Phi}^C}$.
- By linearity of the reconstruction operator $\mathcal{A}^\dagger$ both, data and reconstruction, will have uncorrelated noise realisations and the training loss is then

$$L\left(\Lambda_\theta(f_{\tilde{\Phi}^C}), f_{\tilde{\Phi}}\right) = L\left(\Lambda_\theta\left(\mathcal{A}^\dagger g_{\tilde{\Phi}}\right), \mathcal{A}^\dagger g_{\tilde{\Phi}^C}\right). \tag{3.8}$$

Expected prediction error

The expected prediction error can now be decomposed into a noise dependent term and a reconstruction dependent one.

Theorem 3.1 (Decomposition of expected prediction error (hendriksen2020))

Consider a set of angles $\widetilde{\Phi} \subset \Phi$ and its complement $\tilde{\Phi}^C$. Given the noisy reconstructions $f_{\tilde{\Phi}} = \mathcal{A}^\dagger g_{\tilde{\Phi}}$, $f_{\tilde{\Phi}^C} = \mathcal{A}^\dagger g_{\tilde{\Phi}^C}$, and the clean reconstruction $f^*_{\Phi} = \mathcal{A}^\dagger g^*_{\Phi}$ where g^*_{Φ} denotes noise free data. Denote the joint measure of $f, \eta, \widetilde{\Phi}$ with μ, then for measurable Λ we have

$$\mathbb{E}_\mu \left[\|\Lambda(f_{\tilde{\Phi}^C}) - f_{\tilde{\Phi}}\|_X^2 \right] = \mathbb{E}_\mu \left[\|\Lambda(f_{\tilde{\Phi}^C}) - f^*_{\tilde{\Phi}}\|_X^2 \right] + \mathbb{E}_\mu \left[\|f^*_{\tilde{\Phi}} - f_{\tilde{\Phi}}\|_X^2 \right]. \tag{3.9}$$

- The expected prediction error can be decomposed into the supervised error given a clean reconstruction from noise free data and the error between clean reconstruction and noisy reconstruction, which ideally have mean-free noise.
- In other words, we can expect the denoiser to produce the same result as if it would be trained supervised on clean reconstructions given by the filtered backprojection (or any other linear reconstruction operator $\mathcal{A}^\dagger$).

Final remarks

- Learning the reconstruction operator vs. learning the inverse
- Uncoupled training of a denoiser (relatively) easy to do $\Rightarrow$ But, we need to be careful how to design the learning problem (Inpainting vs. denoising)
- Theoretical extensions: Explain expressivity, regularisation strategy, unsupervised training.