

Article

# Regularization, Bayesian Inference, and Machine Learning Methods for Inverse Problems <sup>†</sup>

Ali Mohammad-Djafari <sup>1,2,3</sup> 

<sup>1</sup> Laboratoire des Signaux et Systèmes, CNRS, CentraleSupélec-University Paris Saclay, 91192 Gif-sur-Yvette, France; djafari@lss.supelec.fr

<sup>2</sup> International Science Consulting and Training (ISCT), 91440 Bures-sur-Yvette, France

<sup>3</sup> Scientific Leader of Shanfeng Company, Shaoxing 312352, China

<sup>†</sup> Presented as a Tutorial at the 40th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, GTU, Austria, 4 July 2021.

**Abstract:** Classical methods for inverse problems are mainly based on regularization theory, in particular those, that are based on optimization of a criterion with two parts: a data-model matching and a regularization term. Different choices for these two terms and a great number of optimization algorithms have been proposed. When these two terms are distance or divergence measures, they can have a Bayesian Maximum A Posteriori (MAP) interpretation where these two terms correspond to the likelihood and prior-probability models, respectively. The Bayesian approach gives more flexibility in choosing these terms and, in particular, the prior term via hierarchical models and hidden variables. However, the Bayesian computations can become very heavy computationally. The machine learning (ML) methods such as classification, clustering, segmentation, and regression, based on neural networks (NN) and particularly convolutional NN, deep NN, physics-informed neural networks, etc. can become helpful to obtain approximate practical solutions to inverse problems. In this tutorial article, particular examples of image denoising, image restoration, and computed-tomography (CT) image reconstruction will illustrate this cooperation between ML and inversion.

**Keywords:** inverse problems; regularization; Bayesian inference; machine learning; artificial intelligence; Gauss–Markov–Potts; Variational Bayesian Approach (VBA); physics-informed ML



**Citation:** Mohammad-Djafari, A. Regularization, Bayesian Inference, and Machine Learning Methods for Inverse Problems. *Entropy* **2021**, *23*, 1673. <https://doi.org/10.3390/e23121673>

Academic Editors: Wolfgang von der Linden and Sascha Ranftl

Received: 1 November 2021

Accepted: 7 December 2021

Published: 13 December 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Inverse problems arise in almost any scientific and engineering application. In fact, they arise whenever we want to infer a quantity that is not directly measured. Noting the unknown quantity  $f$  and the measurement data  $g$ , we may have a mathematical relation between them:  $g = \mathcal{H}(f)$  where  $f$  can be a 1D function (signal), a 2D function (image), a 3D function, or more (e.g., video, hyperspectral images, etc.).  $\mathcal{H}$  is a mathematical model, called a forward operator, and  $g$  can also be a 1D, 2D, 3D, or more function. In practice, we may only have discrete values of it available, and, for this reason, the inverse problem that is inferring  $f$  from this limited data is an ill-posed problem. When discretized, we may write the relations between them as  $g = \mathbf{H}(f) + \epsilon$  where  $g$  contains all the data,  $f$  contains all the discretized representations of the unknown quantity, and  $\mathbf{H}$  is a multidimensional operator connecting them. Finally,  $\epsilon$  represents all the errors of discretization and measurement uncertainties.

Handling inverse problems, even in the discretized version linear model  $g = \mathbf{H}f + \epsilon$  is not easy, at least for two reasons: one is the ill-conditioning of the matrix  $\mathbf{H}$  and its great dimensions; the second is accounting for the errors.

Classical methods for inverse problems are mainly based on regularization theory, particularly those that are based on optimization of a criterion with two parts: a data-model-matching part  $\Delta_1(g, \mathbf{H}f)$  and a regularization term  $\Delta_2(f, f_0)$  with a balancing term between them:  $J(f) = \Delta_1(g, \mathbf{H}f) + \lambda \Delta_2(f, f_0)$  where  $\Delta_1$  and  $\Delta_2$  are two distances (L2, L1, etc.) or

divergence measure such as Kullback–Leibler (KL) or any other divergence, when, for example,  $f$  and  $f_0$  are densities.  $f_0$  can be equal to zero or any other prior default solution. Different choices for these two terms and a great number of optimization algorithms have been proposed with success in very diversified domains and applications [1–5].

Bayesian-inference-based methods also had great success for handling inverse problems, in particular, when the data are noisy, uncertain, some are missing and with some outliers, and where there is a need to account and to quantify uncertainties. In fact, the two terms of the regularization methods can have a Bayesian Maximum A Posteriori (MAP) interpretation where these two terms correspond to the likelihood and prior models, respectively. Indeed, the Bayesian approach gives more flexibility in choosing these terms and particularly the prior term via hierarchical models and hidden variables [6–9]. However, the Bayesian computations can become very heavy computationally. The machine-learning (ML) methods, such as classification, clustering, segmentation, and regression, based on neural networks (NN), such as convolutional NN and deep NN, physics-informed neural networks, etc., can become helpful to obtain approximate but good-quality and practical solutions to inverse problems [10–13].

However, even if in many domains of machine learning such as classification and clustering these methods have shown success, their use in real scientific problems is limited. The main reasons are twofold. First, the users of these tools cannot explain the reasons why they are successful and why they are not. The second is that, in general, these tools cannot quantify the remaining uncertainties.

Model-based and Bayesian-inference approaches have been very successful in linear inverse problems. However, adjusting the hyperparameters is complex, and the cost of the computation is high. The convolutional-neural-networks (CNN) and deep-learning (DL) tools can be useful for pushing these limits further. On the other side, the model-based methods can be helpful for the selection of the structure of CNN and DL, which are crucial in ML success. In this tutorial article, first an overview and a survey of the aforementioned methods are presented, and the possible interactions between them are explored [14,15].

The rest of the article is organized as follows: First, a survey of inverse-problem examples, analytical inversion methods, generalized inversion and regularization methods, and finally the Bayesian inference methods is presented. Then, a discussion on the process and final objectives of imaging systems, for example, in health survey systems, going from the data acquisition to image reconstruction, its segmentation, its feature extraction, and finally its interpretation and usage is presented to prepare the more advanced part of this tutorial—for example, the Bayesian joint reconstruction and segmentation using Gauss–Markov–Potts prior modeling [16–19]. In the third part, first an introduction to machine-learning (ML) tools and processes and basic notions and notations on neural networks (NN) are given. The last part is related to the relations between all these methods via forward modeling, identification, learning, and inversion. These relations are shown via a few simple examples, and then we discuss the fully learned and physics-informed partially learned ML methods for inverse problems.

After mentioning some successful case studies in which the ML tools have been successful [20–35], we arrive at the main conclusions of this article and the future of the possible interactions between model-based and machine-learning tools. We conclude by mentioning the open problems and challenges in both the classical, model-based and the ML tool.

## 2. Inverse Problems Example

Inverse problems arise almost everywhere in science and engineering— everywhere we want to infer an unknown quantity  $f$  that is not accessible (observable) directly. We only have access to another observable quantity  $g$  that is related to it via a linear or a non-linear relation  $H$  [36–38].

As you can see, I am going to use a color code: red for unknown quantities and blue for observed or assumed, known quantities. The forward operator linking the two quantities

is noted  $H$ . In general, the forward operator is well-posed, but the inverse problem is ill-posed. This means that either the classical inverse operator does not exist (existence), or we can define many generalized inverse operators, so many solutions to the problem can be defined (uniqueness), or even if we can define an inverse operator, it may be unstable (stability) [39].

Let us mention a few examples of common inverse problems here.

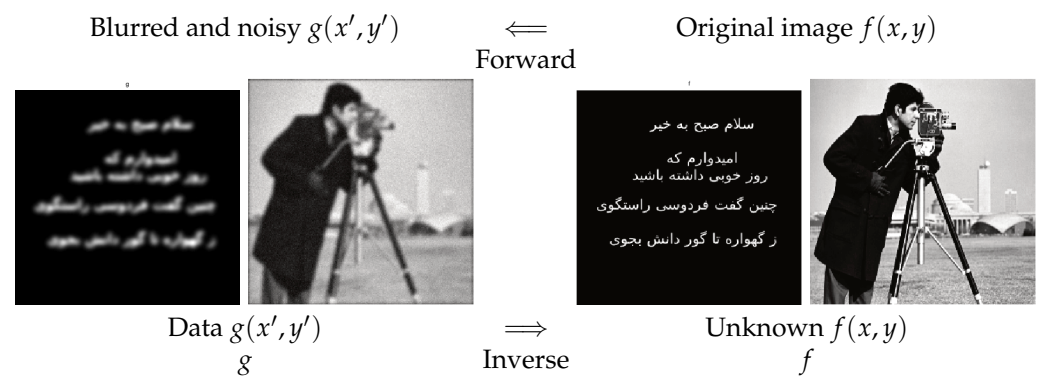
### 2.1. Image Restoration

Any photographic system (camera, microscope, or telescope) has a limited field of view and a limited resolution. If we note by  $f(x, y)$  the original image and by the  $g(x, y)$  the observed image and if we assume a linear and space-invariant operator between them, then the forward relation can be written as a convolution operator:

$$g(x', y') = \int f(x, y)h(x' - x, y' - y) dx dy, \tag{1}$$

where  $h(x, y)$  represents the point spread function (psf) of the imaging system.

Many examples can be given [40,41]. In Figure 1, two synthetic examples are shown.



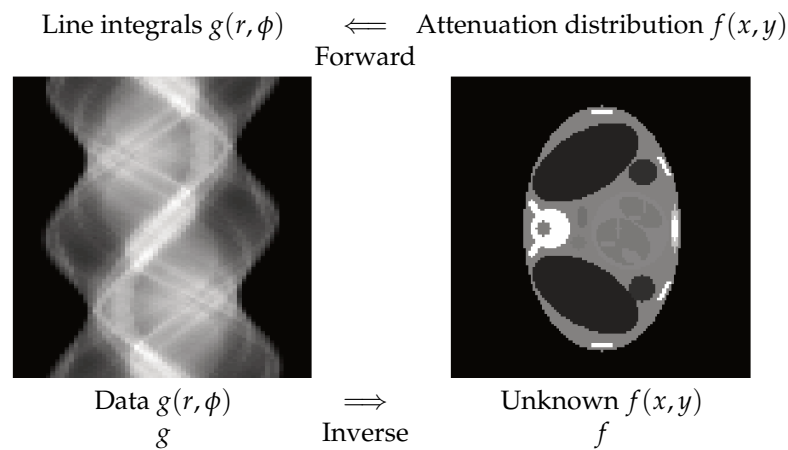
**Figure 1.** Forward and inverse problems in image restoration. Forward operation is a *convolution*, and the inverse operation is called *deconvolution*.

### 2.2. X-ray Computed Tomography

In X-ray computed tomography (CT), the relation between the data and the object can be modeled via the radon transform:

$$g(r, \phi) = \iint f(x, y)\delta(r - x \cos \phi - y \sin \phi) dx dy, \tag{2}$$

where  $\delta$  is the Dirac function; thus,  $g(r, \phi)$  represents the line integrals over the lines of angles  $\phi$  of the object function  $f(x, y)$ . Forward operation is called *projection*, and the inversion process is called *image reconstruction*. In Figure 2, one synthetic example is shown.



**Figure 2.** Forward and inverse problems in computed tomography. The horizontal axis on the left is  $r$ , the vertical is  $\phi$ , and the values of  $g(r, \phi)$  are presented as the gray levels. On the right, the object section  $f(x, y)$  is presented. Forward operation is called *projection*, and the inversion process is called *image reconstruction*.

### 2.3. Acoustical Imaging

Acoustic source localization in acoustical imaging can also be considered as an inverse problem, where the positions and intensities of acoustical sources have to be estimated from the signal received by the microphone arrays. If we represent the distribution of the sources by

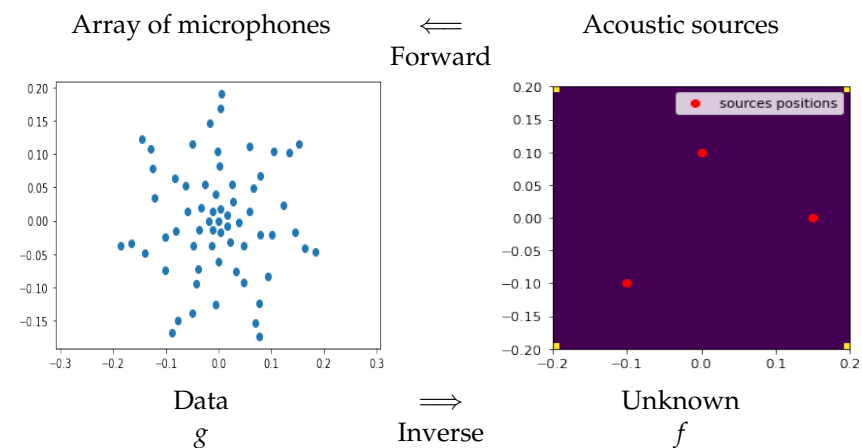
$$f(x, y) = \sum_n s_n \delta(x - x_n, y - y_n),$$

each microphone receives the sum of the delayed sources' sounds [42–45]:

$$g_m(t) = \sum_n s_n(t - \tau_{mn}),$$

where  $\tau_{mn}$  is the delay of transmission from the source position  $n$  to the microphone position  $m$ . This delay is a function of the speed of the sound and the distance between the source position  $(x_n, y_n)$  and the microphone position  $(x_m, y_m)$ .

In Figure 3, one synthetic example is shown to explain the main idea.



**Figure 3.** Forward and inverse problems in acoustical imaging. Each microphone receives the sum of the delayed sources' sounds. The inverse problem is to estimate the sources' distribution from the received signals by the microphones array.

#### 2.4. Microwave Imaging for Breast-Cancer Detection

In microwave imaging, the body is illuminated by microwaves. As the electrical properties (conductivity and permeability) of the healthy and tumor tissues are different, their corresponding induced sources are different. These differences can be measured via the electrodes outside of the breast. The inverse problem, in this case, consists in estimating these induced sources or even directly the distribution of the conductivity and permeability inside the breast. Looking at such images, the tumor area can be visualized [46,47].

The forward problem here is described by two linked equations:

$$\begin{cases} g(\mathbf{r}_i) = \iint_D G_m(\mathbf{r}_i, \mathbf{r}') \phi(\mathbf{r}') f(\mathbf{r}') d\mathbf{r}', \mathbf{r}_i \in S; \\ \phi(\mathbf{r}) = \phi_0(\mathbf{r}) + \iint_D G_o(\mathbf{r}, \mathbf{r}') \phi(\mathbf{r}') f(\mathbf{r}') d\mathbf{r}', ; \mathbf{r} \in D \end{cases}$$

where  $f(\mathbf{r})$  represents the variation of the conductivity,  $\phi(\mathbf{r})$  the induced field due to the incident field  $\phi_0(\mathbf{r})$ , and  $G_m(\mathbf{r}_i, \mathbf{r}')$  and  $G_o(\mathbf{r}, \mathbf{r}')$  are the Green functions.

The first one relates the measured diffracted field on the sensors  $g(\mathbf{r})$  as a function of the induced currents  $J(\mathbf{r}) = \phi(\mathbf{r})f(\mathbf{r})$  inside the brain due to the external field via the Green functions, and the second relates the total field as the sum of the incident and the induced field. So, the forward problem is nonlinear, as  $\phi(\mathbf{r})$  appears in both sides of the equation. However, it can be approximated by a linear relation if we assume that the induced field inside is very small compared to the incident field: ( $\phi(\mathbf{r}') \simeq \phi_0(\mathbf{r}')$ ). This is called Born approximation:

$$g(\mathbf{r}_i) = \iint_D G_m(\mathbf{r}_i, \mathbf{r}') \phi_0(\mathbf{r}') f(\mathbf{r}') d\mathbf{r}', \mathbf{r}_i \in S.$$

Both the bi-linear relations and the linear Born approximations are used in microwave imaging. The first one is more common in industrial non-destructive testing (NDT) and the second for biological and medical applications.

#### 2.5. Brain Imaging

In brain imaging, the electrical activity of the neurons inside the brain are propagated and can be measured at the surface of the skull via the electrodes fixed on it. These signals are called electroencephalography (EEG). It is also possible to measure the magnetic field created by this activity. This time, the signals are called (MEG). In both cases, the inversion process consists in estimating the distribution of the brain activity from the measured signals. If the brain electrical activity can be modeled as the electrical mono- or dipoles distributed over the surface of the brain, then the simplified forward model can be almost similar to acoustical sources localization of the previous example. Here, the distribution of the sources are in the 3D space of the brain, and the EEG electrodes are positioned on the skull. The signal received by each EEG electrode can be compared to the signals received by the microphones in the previous example. There are a great number of forward models, analysis, and inversion methods that have been proposed for this application. A very good toolbox is the EEGLAB, which can be searched on the internet and easily obtained.

#### 2.6. Other Applications

Many other imaging systems to see inside the human body or inside any industrial object in non-destructive testing (NDT) applications exist. Here, a few of them are illustrated. We can just mention a few more: magnetic-resonance imaging (MRI), ultrasound imaging such as echography, positron emission tomography (PET), single-emission computed tomography (SPECT), electrical impedance tomography, and eddy current tomography [48].

### 3. Classification of Inverse Problems' Methods

Inverse problems' methods can be classified in the following categories:

- Analytical inversion methods.
- Generalized inversion approach.
- Regularization methods.
- Bayesian inference methods.

In the first category, the main idea is to recognize the forward operators as one of the well-known mathematical invertible operator and thus to use the appropriate inversion operator. Typical examples are Fourier transform (FT) and radon transform (RT). In the second category, the notion of generalized inversion is used. The corresponding methods are either based on singular value decomposition (SVD) or the iterative projection based algorithms. The regularization methods are mainly based on the optimization of a criterion, often made in two parts: data–model adequacy and the regularization with a regularization parameter. Finally, the Bayesian inference approach, which I consider to be the most general and complete, has all the necessary tools to go beyond the regularization methods.

### 4. Analytical Methods

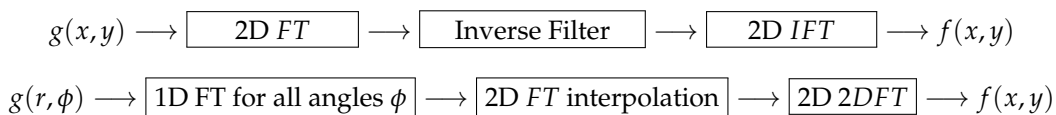
Figure 4 shows the main idea behind the analytical methods via two classical cases of image deconvolution and X-ray image reconstruction. In the first case, as the forward model is a 2D convolution:  $g(x, y) = h(x, y) * f(x, y)$  or equivalently a 2D Fourier-transform (FT) filtering:  $G(u, v) = H(u, v)F(u, v)$ ; the operation consists in going to the Fourier domain, doing inverse filtering, and coming back. However, the inverse filtering  $\frac{1}{H(u,v)}$  must be regularized either by limiting the band width or by applying an appropriate window mask before doing the inverse Fourier transform (IFT).

In the second case, the forward model is the radon transform (RT). Using the relation between FT and RT (Fourier slice theorem), the analytical inversion process becomes:

1. For each angle  $\phi$ , compute the 1D FT of  $g_\phi(r) = g(r, \phi)$ ,
2. Relate it to the 2D FT of  $f(x, y)$  via the Fourier slice theorem and interpolate to obtain the full 2D FT of  $f(x, y)$ ; and
3. Compute 2D IFT to obtain  $f(x, y)$

For more details, refer to [49,50].

The main difficulty with the analytical method is that the forward relations are given for continuous functions or densities. They can give satisfactory results if the inversion process is regularized and if the data are dense, complete, and without any measurement or discretization errors. In practical situations, rarely are all these conditions satisfied.



**Figure 4.** Transform-based analytical methods. Two examples are given: image deconvolution by inverse filtering and image reconstruction in CT by using the relation between RT and FT.

### 5. Generalized Inversion Approach

In this approach, the main idea is based on the fact that the forward operator is in general a singular one. This means there are many possible solutions to the inverse problem. In this approach, there are mainly two categories of methods. The first are based on singular-values decomposition (SVD). The second is based on optimization of a criterion such as the least squares (LS). In both, the main idea is to define a set of possible solutions, called generalized inverse solutions:

$$\{f^\dagger : Hf^\dagger = g\} \tag{3}$$

or pseudo solutions:

$$\{f^\dagger : \|Hf^\dagger - g\|^2 < \epsilon\}. \tag{4}$$

Then, between those possible solutions, one tries to define a criterion, such as the minimum norm, to choose a solution. For the linear inverse problems, the corresponding solutions are given by

$$f^\dagger = [H^t H]^{-1} H^t g = H^\dagger g \quad \text{or} \quad f^\dagger = H^t [H H^t]^{-1} g = H^\dagger g. \tag{5}$$

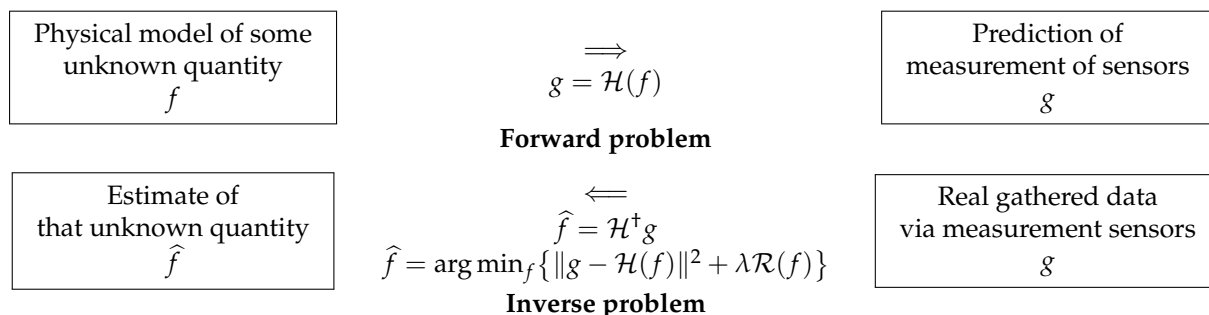
In great dimensional problems, even if we have these analytical expressions, in practice, the solutions are computed by using iterative optimization algorithms, for example, to optimize the LS criterion  $J(f) = \|Hf - g\|^2$  by a gradient-based algorithm:

$$f^{(k+1)} = f^{(k)} + \alpha H^t (g - Hf^{(k)}), \tag{6}$$

with a stopping criteria or just after some fixed number of iterations. We will see in the next sections how this can lead to a deep-learning NN structure.

### 6. Model-Based and Regularization Approach

The model-based methods are related to the notions of the forward-model and the inverse-problems approach. Figure 5 shows the main idea:



**Figure 5.** Model-based methods: forward and inverse problems. The solution of the inverse problem is defined either by the generalized inversion or by a regularization method.

Given the forward model  $\mathcal{H}$  and the source  $f$ , the prediction of the data  $g$  can be done, either in a deterministic way:  $g = \mathcal{H}(f)$  or via a probabilistic model:  $p(g|f, \mathcal{H})$  as we will see in the next section. In the same way, given the forward model  $\mathcal{H}$  and the data  $g$ , the estimation of the unknown source  $f$  can be done either via a deterministic method or a probabilistic one. One of the deterministic methods is the generalized inversion:  $f = \mathcal{H}^\dagger(g)$ . A more general method is the regularization:

$$\hat{f} = \arg \min_f \{ J(f) \} \quad \text{with} \quad J(f) = \|g - \mathcal{H}(f)\|^2 + \lambda \mathcal{R}(f). \tag{7}$$

As we will see later, the only probabilistic method that can be efficiently used for the inverse problems is the Bayesian approach.

#### Regularization Methods

Let consider the discretized linear inverse problem:  $g = Hf + \epsilon$ , and the regularization criterion

$$J(f) = \frac{1}{2} \|g - Hf\|_2^2 + \lambda R(f). \tag{8}$$

The first main issue in such a regularization method is the choice of the regularizer. The most-common examples are:

$$R(f) = \left\{ \|f\|_2^2, \|f\|_{\beta'}^\beta, \|Df\|_2^2, \|Df\|_{\beta'}^\beta, \sum_j \phi([Df]_j) \right\}, 1 \leq \beta \leq 2, \quad (9)$$

where  $D$  is a linear operator, generally a first-order derivation, a gradient, or a second-order derivation. The function  $\phi$  has to be convex to ensure the uniqueness of the solution. Many such functions have been proposed, but some non-convex ones have also been proposed, which then need global optimization techniques.

The second main issue in regularization is the choice of an appropriate optimization algorithm. This mainly depends on the type of the criterion, and we have:

- $R(f)$  quadratic: gradient-based and conjugate gradient algorithms are appropriate.
- $R(f)$  non-quadratic but convex and differentiable: here too the gradient-based and the conjugate gradient (CG) methods can be used, but there are also a great number of convex criterion optimization algorithms.
- $R(f)$  convex but non-differentiable: here, the notion of a sub-gradient is used.

Specific cases are:

- L2 or quadratic:  $J(f) = \frac{1}{2}\|g - Hf\|_2^2 + \lambda\|Df\|_2^2$  ;  
In this case we have an analytic solution:  $\hat{f} = (H^t H + \lambda D^t D)^{-1} H^t g$ . However, in practice, this analytic solution is not usable in high-dimensional problems. In general, as the gradient  $\nabla J(f) = -H^t(g - Hf) + 2\lambda D^t Df$  can be evaluated analytically, gradient-based algorithms are used.
- L1 (TV): convex but not differentiable at zero:  $J(f) = \frac{1}{2}\|g - Hf\|_2^2 + \lambda\|Df\|_1$  ;  
The algorithms in this case use the notions of the Fenchel conjugate, the dual problem, the sub gradient, and the proximal operator [11,51–53]
- Variable splitting and augmented Lagrangian

$$(f, \hat{z}) = \arg \min_{f, z} \left\{ \frac{1}{2}\|g - Hf\|_2^2 + \lambda\|z\|_1 + q\|z\|_2^2 \right\} \text{ s.t. } z = Df. \quad (10)$$

A great number of optimization algorithms have been proposed: ADMM, ISTA, FISTA, etc. [1,5,54].

The main limitations of deterministic regularization methods are:

- A limited choice of the regularization term. Mainly, we have: (a) smoothness (Tikhonov) and (b) sparsity, piecewise continuous (total variation).
- Determination of the regularization parameter. Even if there are some classical methods such as the L-curve and cross validation, there are still controversial discussions about this.
- Quantification of the uncertainties: this is the main limitation of the deterministic methods, particularly in medical and biological applications where this point is important.

The best possible solution to push further all these limits is the Bayesian approach, which has: (a) many possibilities to choose prior models, (b) the possibility of the estimation of the hyperparameters, and, most important, (c) an accounting for the uncertainties.

## 7. Bayesian-Inference Methods

### 7.1. Basic Idea

The simple case of the Bayes rule is:

$$p(f|g, \mathcal{H}) = \frac{p(g|f, \mathcal{H}) p(f|\mathcal{H})}{p(g|\mathcal{H})} \text{ where } p(g|\mathcal{H}) = \int p(g|f, \mathcal{H}) p(f|\mathcal{H}) df, \quad (11)$$



where  $\mathcal{H}$  is a model,  $p(g|f, \mathcal{H})$  is the likelihood of  $f$  in the data through the model,  $p(f|\mathcal{H})$  is the prior knowledge about the unknown quantity  $f$ , and  $p(f|g, \mathcal{H})$  called the posterior is the result of the combination of the likelihood and the prior. The denominator  $p(g|\mathcal{H})$ , called the evidence, is the overall likelihood of the model in the data  $g$ .

When there are some hyperparameters, for example, the parameters of the likelihood and those of the prior law, which have also to be estimated, we have:

$$p(f, \theta|g, \mathcal{H}) = \frac{p(g|f, \theta, \mathcal{H}) p(f|\theta, \mathcal{H}) p(\theta|\mathcal{H})}{p(g|\mathcal{H})} \text{ where } p(g|\mathcal{H}) = \int p(g|f, \theta, \mathcal{H}) p(f|\theta, \mathcal{H}) d\theta df \quad (12)$$

This is called the joint posterior law of all the unknowns. From that joint posterior distribution, we may also obtain the marginals:

$$p(f|g, \mathcal{H}) = \int p(f, \theta|g, \mathcal{H}) d\theta \text{ and } p(\theta|g, \mathcal{H}) = \int p(f, \theta|g, \mathcal{H}) df. \quad (13)$$

7.2. Gaussian Priors Case

To be more specific, let us consider the case of linear inverse problems  $g = \mathbf{H}f + \epsilon$ . Then, assuming Gaussian noise, we have:

$$p(g|f) = \mathcal{N}(g|\mathbf{H}f, v_\epsilon \mathbf{I}) \propto \exp\left\{ \frac{-1}{2v_\epsilon} \|g - \mathbf{H}f\|_2^2 \right\}. \quad (14)$$

Assuming also a Gaussian prior:

$$p(f) \propto \exp\left\{ \frac{-1}{2v_f} \|f\|_2^2 \right\} \text{ or } \exp\left\{ \frac{-1}{2v_f} \|\mathbf{D}f\|_2^2 \right\}, \quad (15)$$

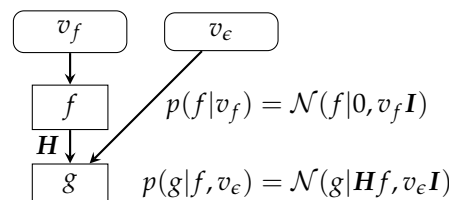
it is easy to see that the posterior is also Gaussian, and the MAP and posterior mean (PM) estimates become the same and can be computed as the minimizer of :  $J(f) = \|g - \mathbf{H}f\|_2^2 + \lambda R(f)$ :

$$p(f|g) \propto \exp\left\{ \frac{-1}{2v_\epsilon} J(f) \right\} \rightarrow \hat{f}_{MAP} = \arg \max_f \{p(f|g)\} = \arg \min_f \{J(f)\}. \quad (16)$$

In summary, we have:

$$\begin{cases} p(g|f) = \mathcal{N}(g|\mathbf{H}f, v_\epsilon \mathbf{I}) \\ p(f) = \mathcal{N}(f|0, v_f \mathbf{I}) \end{cases} \rightarrow \begin{cases} p(f|g) = \mathcal{N}(f|\hat{f}, \hat{\Sigma}) \\ \hat{f} = [\mathbf{H}^t \mathbf{H} + \lambda \mathbf{I}]^{-1} \mathbf{H}^t g \\ \hat{\Sigma} = v_\epsilon [\mathbf{H}^t \mathbf{H} + \lambda \mathbf{I}]^{-1}, \quad \lambda = \frac{v_\epsilon}{v_f} \end{cases}. \quad (17)$$

This case is also summarized in (Figure 6).



$$p(f|g, v_f, v_\epsilon) = \mathcal{N}(f|\hat{f}, \hat{\Sigma})$$

Figure 6. Bayesian-inference scheme in linear systems and Gaussian priors. The posterior is also Gaussian, and all the computations can be done analytically.

We may note that, in this case, we have an analytical expression for the posterior law, which is also a Gaussian law and entirely specified by its mean  $\hat{f}$  and covariance  $\hat{\Sigma}$ .

However, for great dimensional problems where  $f$  is a great size vector, the computation of  $\hat{\Sigma}$  can become very costly. The computation of the posterior mean  $\hat{f}$  can be done by optimization as it is the same as the MAP solution.

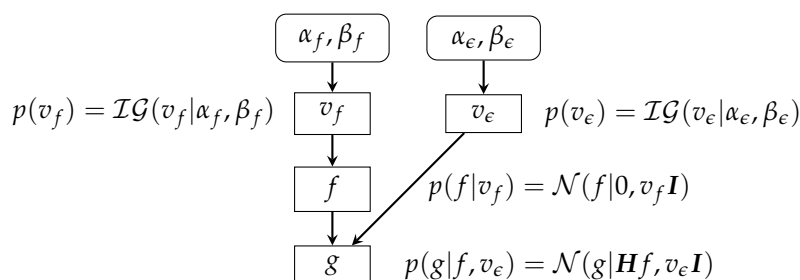
### 7.3. Gaussian Priors with Unknown Parameters

For the case where the hyperparameters  $v_\epsilon$  and  $v_f$  are unknown (unsupervised case), we can derive the following:

$$\left\{ \begin{array}{l} p(g|f, v_\epsilon) = \mathcal{N}(g|\mathbf{H}f, v_\epsilon \mathbf{I}) \\ p(f|v_f) = \mathcal{N}(f|0, v_f \mathbf{I}) \\ p(v_\epsilon) = \mathcal{IG}(v_\epsilon|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \\ p(v_f) = \mathcal{IG}(v_f|\alpha_{f_0}, \beta_{f_0}) \end{array} \right. \rightarrow \left\{ \begin{array}{l} p(f|g, v_\epsilon, v_f) = \mathcal{N}(f|\hat{f}, \hat{\Sigma}) \\ \hat{f} = [\mathbf{H}^t \mathbf{H} + \hat{\lambda} \mathbf{I}]^{-1} \mathbf{H}^t g \\ \hat{\Sigma} = v_\epsilon h [\mathbf{H}^t \mathbf{H} + \hat{\lambda} \mathbf{I}]^{-1}, \hat{\lambda} = \frac{v_\epsilon h}{v_f h} \\ p(v_\epsilon|g, f) = \mathcal{IG}(v_\epsilon|\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon) \\ p(v_f|g, f) = \mathcal{IG}(v_f|\tilde{\alpha}_f, \tilde{\beta}_f) \\ \tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon, \tilde{\alpha}_f, \tilde{\beta}_f \end{array} \right., \quad (18)$$

where all the details and, in particular, the expressions for  $\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon, \tilde{\alpha}_f, \tilde{\beta}_f$  can be found in [19]. As we can see, the expressions of  $\hat{f}$  and  $\hat{\Sigma}$  are the same as in previous case, except that the values of  $v_\epsilon h, v_f h$ , and  $\hat{\lambda}$  have to be updated. They are obtained from the conditionals  $p(v_\epsilon|g, f)$  and  $p(v_f|g, f)$  at each iteration.

This case is also summarized in Figure 7.



$$p(f, v_f, v_\epsilon | g) \propto \mathcal{N}(g|\mathbf{H}f, v_\epsilon \mathbf{I}) \mathcal{N}(f|0, v_f \mathbf{I}) \mathcal{IG}(v_\epsilon|\alpha_\epsilon, \beta_\epsilon) \mathcal{IG}(v_f|\alpha_f, \beta_f)$$

**Figure 7.** Bayesian-inference scheme in linear systems and Gaussian priors. The posterior is also Gaussian, and all the computations can be done analytically.

The joint posterior can be written as:

$$p(f, v_\epsilon, v_f | g) \propto \exp\{-J(f, v_\epsilon, v_f)\}. \quad (19)$$

From this expression, we have different possible expansions:

- JMAP: alternate optimization with respect to  $f, v_\epsilon, v_f$ :

$$J(f, v_\epsilon, v_f) = \frac{1}{2v_\epsilon} \|g - \mathbf{H}f\|_2^2 + \frac{1}{2v_f} \|f\|_2^2 + (\alpha_{\epsilon_0} + 1) \ln v_\epsilon + \frac{\beta_{\epsilon_0}}{v_\epsilon} + (\alpha_{f_0} + 1) \ln v_f + \frac{\beta_{f_0}}{v_f}. \quad (20)$$

Each iteration can be done in two steps:

1. Fix  $v_\epsilon$  and  $v_f$  to previous values and optimize with respect to  $f$ ;
2. Fix  $f$  and optimize with respect to  $v_\epsilon$  and  $v_f$ .

The first step results in a quadratic criterion with respect to  $f$ , which results in an analytical expression for the solution, which can be used for small-dimension problems, or it can be optimized easily by any gradient-based algorithm. The second step, in this case, results in two separate explicit solutions: one for  $v_\epsilon$  and one for  $v_f$ .

- Gibbs sampling and Markov Chain Monte Carlo (MCMC):

$$f \sim p(f, v_\epsilon, v_f | g) \rightarrow v_\epsilon \sim p(v_\epsilon | g, f) \rightarrow v_f \sim p(v_f | g, f). \tag{21}$$

These steps can be done using the expressions of the conditional given in Equation (18). These methods are used generally when we not only want to have a point estimator such as MAP or the posterior mean but also to quantify the uncertainties by estimations of the variances and covariances.

- Variational Bayesian Approximation (VBA): approximate  $p(f, v_\epsilon, v_f | g)$  by a separable one  $q(f, v_\epsilon, v_f) = q_1(f | v_\epsilon, v_f) q_2(v_\epsilon) q_3(v_f)$  minimizing  $KL(q | p)$  [19,55–58]. We can see that the alternate optimization of  $KL(q_1, q_2, q_3 | p)$  with respect to  $q_1, q_2$ , and  $q_3$  result in the same expressions as in Equation (18), only the expressions for updating the parameters  $\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon, \tilde{\alpha}_f$ , and  $\tilde{\beta}_f$  are different.

The Approximate Bayesian Computation (ABC) method and, in particular, the VBA and mean-field-approximation methods are used when Gibbs sampling and MCMC methods are too expensive and we still want to quantify uncertainties, for example, estimating the variances.

### 8. Imaging inside the Body: From Data Acquisition to Decision

To introduce the link between the different model-based methods and the machine-learning tools, let us consider the case of medical imaging, from the acquisition to the decision steps:

- Data acquisition :

$$\text{Object } f \rightarrow \boxed{\text{CT scan, MRI, TEP, US, Microwave imaging}} \rightarrow \text{Data } g$$

- Image reconstruction by analytical methods:

$$\text{Data } g \rightarrow \boxed{\text{Reconstruction}} \rightarrow \text{Image } \hat{f}$$

- Post-processing (segmentation, contour detection, and selection of region of interest):

$$\text{Image } \hat{f} \rightarrow \boxed{\text{Segmentation}} \rightarrow \hat{z}$$

- Understanding and decision:

$$\begin{array}{c} \text{Image } \hat{f} \\ \text{Segmentation } \hat{z} \end{array} \rightarrow \boxed{\begin{array}{c} \text{Interpretation} \\ \text{Decision} \end{array}} \rightarrow \begin{array}{c} \text{Tumor or} \\ \text{Not Tumor} \end{array}$$

#### 8.1. Bayesian Joint Reconstruction and Segmentation

The questions now are: can we join any of these steps? Can we go directly from the image to the decision? For the first one, the Bayesian approach can provide a solution:

$$\text{Data } g \rightarrow \boxed{\begin{array}{c} \text{Reconstruction} \\ \text{Segmentation} \end{array}} \rightarrow \begin{array}{c} \text{Reconstruction } \hat{f} \\ \text{Segmentation } \hat{z} \end{array}$$

The main tool here is to introduce a hidden variable that can represent the segmentation. A solution is to introduce a classification hidden variable  $z$  with  $z_j = \{1, 2, \dots, K\}$ , which can be used to show the segmented image. See Figure 8.

Figures 8 and 9 summarize this scheme.

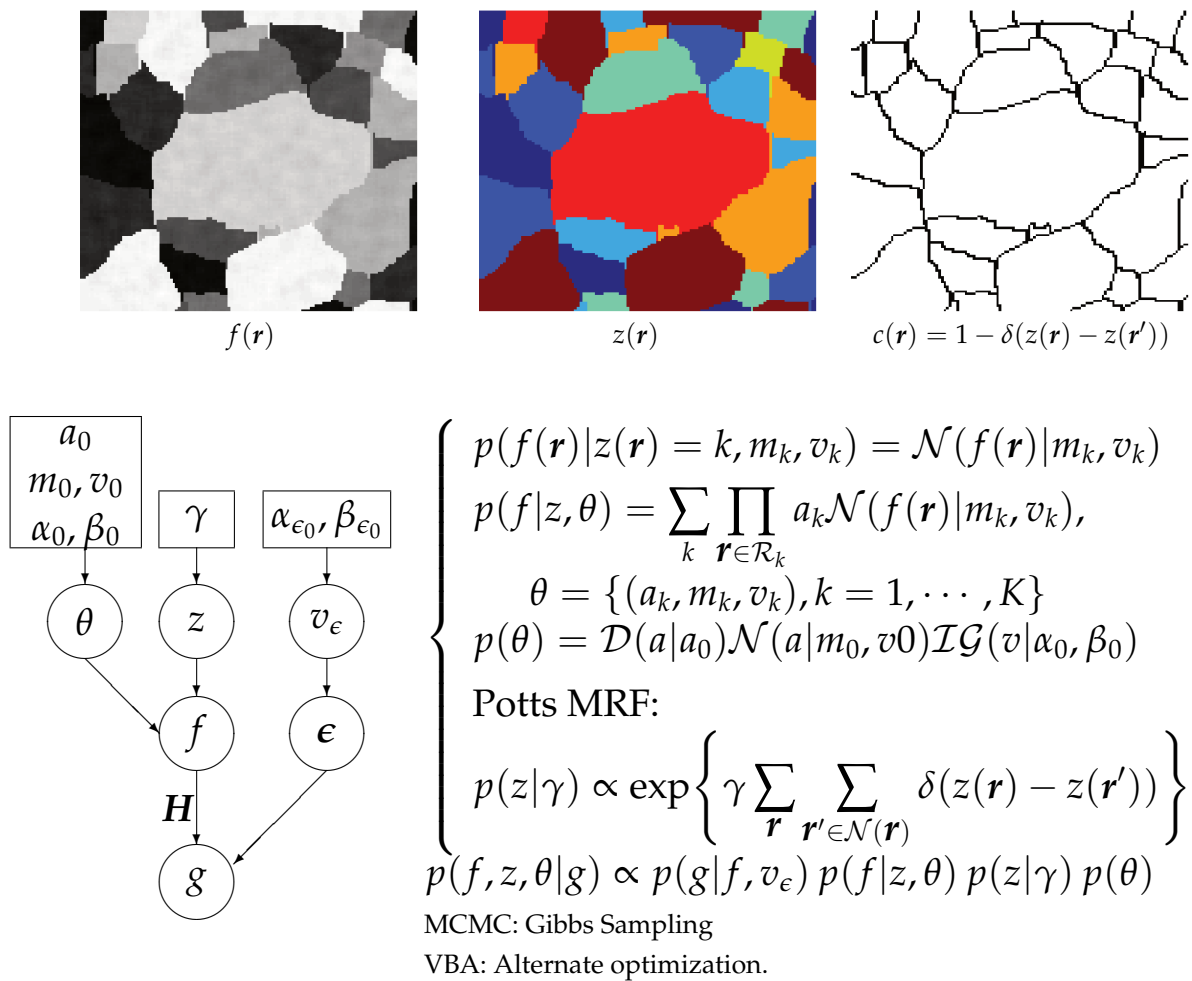


Figure 8. Gauss–Markov–Potts prior model for Bayesian image reconstruction and segmentation.

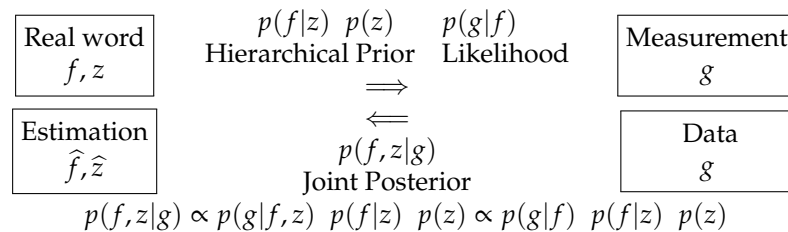


Figure 9. Bayesian approach with hierarchical prior model for joint reconstruction and segmentation.

A few comments for these relations:

- $p(g|f, z)$  does not depend on  $z$ , so it can be written as  $p(g|f)$ .
- We used a Markovian Potts model for  $p(z)$  to obtain more compact homogeneous regions [18,19].
- If we choose for  $p(f|z)$  a Gaussian law, then  $p(f, z|g)$  becomes a Gauss–Markov–Potts model [19].
- We can use the joint posterior  $p(f, z|g)$  to infer on  $(f, z)$ : we may just do JMAP:  $(\hat{f}, \hat{z}) = \arg \max \{p(f, z|g)\}$  or trying to access to the expected posterior values by using the Variational Bayesian Approximation (VBA) techniques [17,19,58–62].
- When the iterations finished, we obtain an estimate of the reconstructed image  $f$  and its segmentation  $z$  when using JMAP and also the covariance of  $f$  as well as the parameters of the posterior laws of  $z$

This scheme can be extended to consider the estimation of the hyperparameters too. Figure 10 shows this.

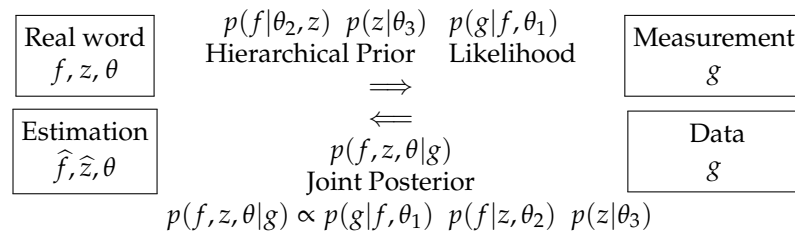


Figure 10. Advanced Bayesian approach for joint reconstruction, segmentation and parameter estimation.

Again, here, we can use the joint posterior  $p(f, z, \theta|g)$  to infer on all the unknowns. When the iterations are finished, we get an estimate of the reconstructed image  $f$ , its segmentation  $z$ , as well as all the unknown parameters such as the means and variances of the reconstructed image at each of its segments. Giving more details is out of the focus of this overview article. They can be found more specifically in [17].

### 8.2. Advantages of the Bayesian Framework

Between the main advantages of the Bayesian framework for inverse problems, we can mention the following:

- Large flexibility of prior models.
  - Smoothness (Gaussian and Gauss–Markov).
  - Direct sparsity (double exp., heavy-tailed distributions).
  - Sparsity in the transform domain (double exp., heavy-tailed distributions on the WT coefficients).
  - Piecewise continuous (DE or Student-t on the gradient).
  - Objects composed of only a few materials (Gauss–Markov–Potts), ...
- Possibility of estimating hyperparameters via JMAP or VBA.
- Natural ways to take account for uncertainties and to quantify the remaining uncertainties.

### 8.3. Imaging inside the Body: From Data to Decision: Classical or Machine Learning

From previous sections, we see that we have many solutions to go from data to an image by inversion (image reconstruction), then extraction of interesting features (segmentation), and finally the interpretation and decision. The question that we may ask now is: *can we do all together in a more easy way?* Machine-learning and artificial-intelligence tools may propose such a solution. See Figure 11. To be able to use ML to go from data to decision, there is a crucial need of a great and rich database obtained by experts to let the machine learn from that great database. In the next section, we add a little more in detail to see the advantages, limitations, and drawbacks.

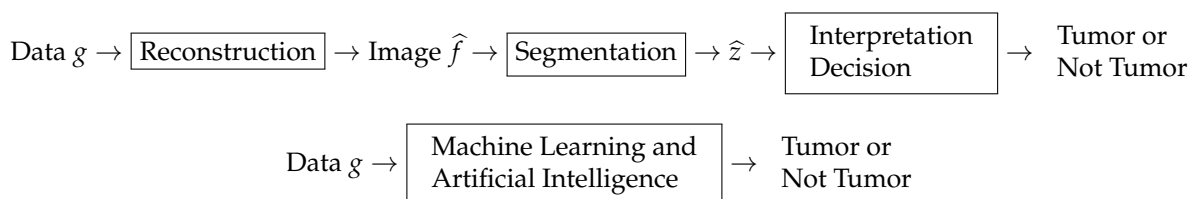
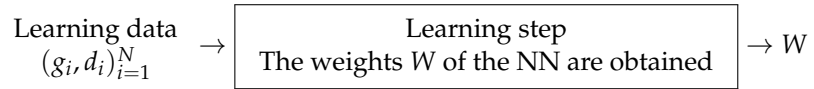


Figure 11. Two approaches going from the data to decision. Top: from data, first, reconstruct an image via inversion, then post-process to obtain segmentation. Perform pattern recognition to extract the contours of the region of interest and finally make a decision. Bottom: try to use machine-learning methods to go directly from data to decision.

### 9. Machine Learning's Basic Idea

The main idea in machine learning is first to learn from a great number of data-decisions:  $(g_i, d_i), i = 1, \dots, N$ :



and then, when a new case (Test  $g_j$ ) appears, it uses the learned weights  $W$  to give a decision  $d_j$

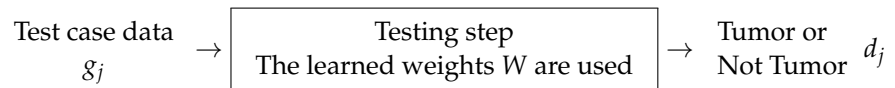


Figure 12 shows the main process of ML.

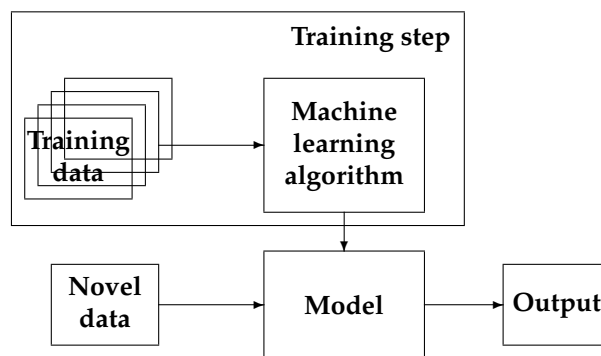


Figure 12. Basic machine-learning process: first, learn a model, then use it. Learning step needs a rich-enough database, which is expensive. When the model is learned and tested, its use is easy and fast, and its cost is low.

Nowadays, ML methods and tools have made great progress in many different areas of applications. There is no need here to go more in detail, instead mentioning a few main components of all of them. Between the basic tasks we can mention:

- Classification (supervised and semi-supervised);
- Clustering (unsupervised classification when the data do not yet have labels);
- Regression (continuous parameter estimation).

Figure 13 shows these three main tasks.



Figure 13. Basic machine-learning tasks: classification, clustering, and regression.

Between the existing ML tools, we may mention: support vector machines (SVM), decision-tree learning (DT), artificial neural networks (ANN), Bayesian networks (BN), HMM and random forest (RF), mixture models (GMM, SMM, etc.), KNN, Kmeans, etc.

Additionally, the combination of Imaging technology and systems, image processing, computer vision, machine learning, and artificial intelligence has been the seed for much great progress in all areas of health and our environment. The frontiers between science and technology has become less precise as is shown in Figure 14.

Image technology	2D, 3D, hyperspectral acquisition, compression, transmission;
Image processing	Representation, compression, segmentation; Enhancement, restoration;
Computer vision	Segmentation, contour detection; Segments, edges, patterns, RoIs, features extraction;
Machine learning	Pattern matching and localization; Objects detection and identification;
Artificial intelligence	2D and 3D pattern recognition, interpretation; Classification, clustering, recognition, decision making, etc.

**Figure 14.** Frontiers between image technology, image processing (IP), computer vision (CV), machine Learning (ML) and artificial intelligence (AI).

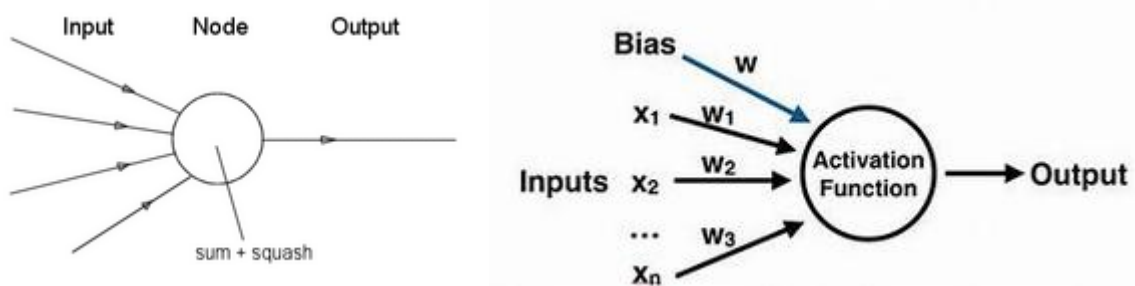
Between the machine-learning tools using NN, the convolutional NN (CNN), recurrent NN (RNN), deep learning (DL), generative artificial networks (GAN) had greater success in different area such as speech recognition, computer vision, and specifically in segmentation, classification and clustering, and in multi-modality and cross-domain information fusion.

However, there are still many limitations: a lack of interpretability, reliability, and uncertainty and no reasoning and explaining capabilities. To overcome this, there os still much to do with the fundamentals.

### 10. Neural Networks, Machine Learning, and Inverse Problems

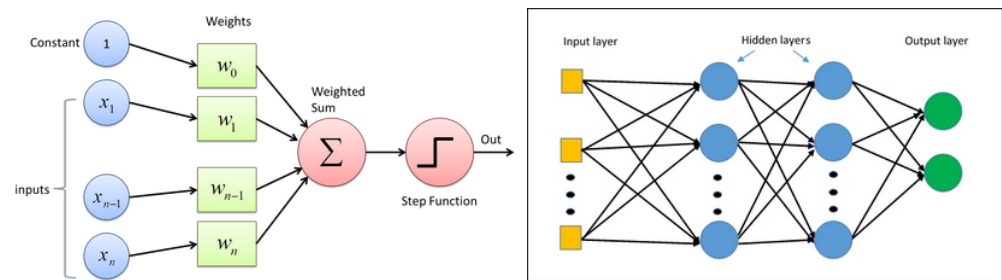
#### 10.1. Neural Networks

Let us start this section with a few words on neurons and neural networks. The following figures show the basic idea. The following figure shows the main idea about a neuron in a mathematical framework. Figure 15 shows this graphically.



**Figure 15.** A neuron and its mathematical representation.

Figure 16 shows the components of a neuron and an example of a two-layer NN.



**Figure 16.** A neuron with its inputs and outputs and a neural network with two-hidden-layer neurons.

10.2. NN and Learning

A neural network can be used for modeling a universal relation between its inputs  $X$  and outputs  $Y$ . This model can be written as  $Y = F_W(X)$  where  $W$  represents the parameters of the model represented by the weights of the network nodes relation. They are commonly used for:

- Classification (supervised learning)  
A set of data  $\{(x_i, y_i)\}$  with labels (classes)  $\{c_i\}$  are given. The objective during the training is to use them for training the network, which is then used for classifying a new income  $(x_j, y_j)$ .
- Clustering (unsupervised learning)  
A set of data  $\{(x_i, y_i)\}$  is given. The objective is to cluster them in different classes  $\{c_i\}$ .
- Regression with all data (supervised learning).  
A set of data  $\{(x_i, y_i)\}$  are given. The objective is to find a function  $F$  describing the relation between them:  $F(x, y)$  or explicitly  $y = F(x)$  for any  $x$  (extrapolation or interpolation).

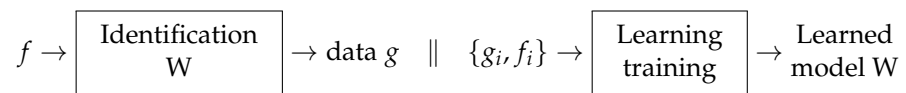
10.3. Modeling, Identification, and Inversion

Here, we make a connection between the classical and ML tools and show the links between forward modeling and inversion or inference, model identification and learning or training, and inversion and using the NN:

- Forward modeling and inversion



- Identification of a system and the training step of NN



- Inversion (inference) or using the NN-trained model



11. ML for Inverse Problems

To show the possibilities of the interaction between inverse problems' methods, machine learning, and NN methods, the best way is to give a few examples.



11.1. First Example: A Linear One- or Two-Layer Feed-Forward NN

The first one is the case of linear inverse problems and quadratic regularization of the Bayesian with Gaussian priors. The solution has an analytic expression, and we have the following relations:

$$g = \mathbf{H}f + \epsilon \rightarrow \hat{f} = (\mathbf{H}^t\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^t g = \mathbf{A} g = \mathbf{B}\mathbf{H}^t g \text{ or still } \hat{f} = \mathbf{H}^t(\frac{1}{\lambda}\mathbf{H}\mathbf{H}^t + \mathbf{I})^{-1}g = \mathbf{H}^t\mathbf{C} g$$

where  $\mathbf{A} = (\mathbf{H}^t\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^t$ ,  $\mathbf{B} = (\mathbf{H}^t\mathbf{H} + \lambda\mathbf{I})^{-1}$  and  $\mathbf{C} = (\frac{1}{\lambda}\mathbf{H}\mathbf{H}^t + \mathbf{I})^{-1}$ .

These relations can be presented schematically as

$$g \rightarrow \boxed{\mathbf{A}} \rightarrow \hat{f}, \quad g \rightarrow \boxed{\mathbf{H}^t} \rightarrow \boxed{\mathbf{B}} \rightarrow \hat{f}, \quad g \rightarrow \boxed{\mathbf{C}} \rightarrow \boxed{\mathbf{H}^t} \rightarrow \hat{f}$$

As we can see, these relations directly induce a linear feed-forward NN structure. In particular, if  $\mathbf{H}$  represents a convolution operator, then  $\mathbf{H}^t$ ,  $\mathbf{H}^t\mathbf{H}$ , and  $\mathbf{H}\mathbf{H}^t$  are too, as well as the operators  $\mathbf{B}$  and  $\mathbf{C}$ . Thus, the whole inversion can be modeled by CNN [63].

For the case of computed tomography (CT), the first operation is equivalent to an analytic inversion; the second corresponds to back-projection first followed by 2D filtering in the image domain; and the third corresponds to the famous filtered back-projection (FBP), which is implemented on classical CT scans. These three cases are illustrated on Figure 17.

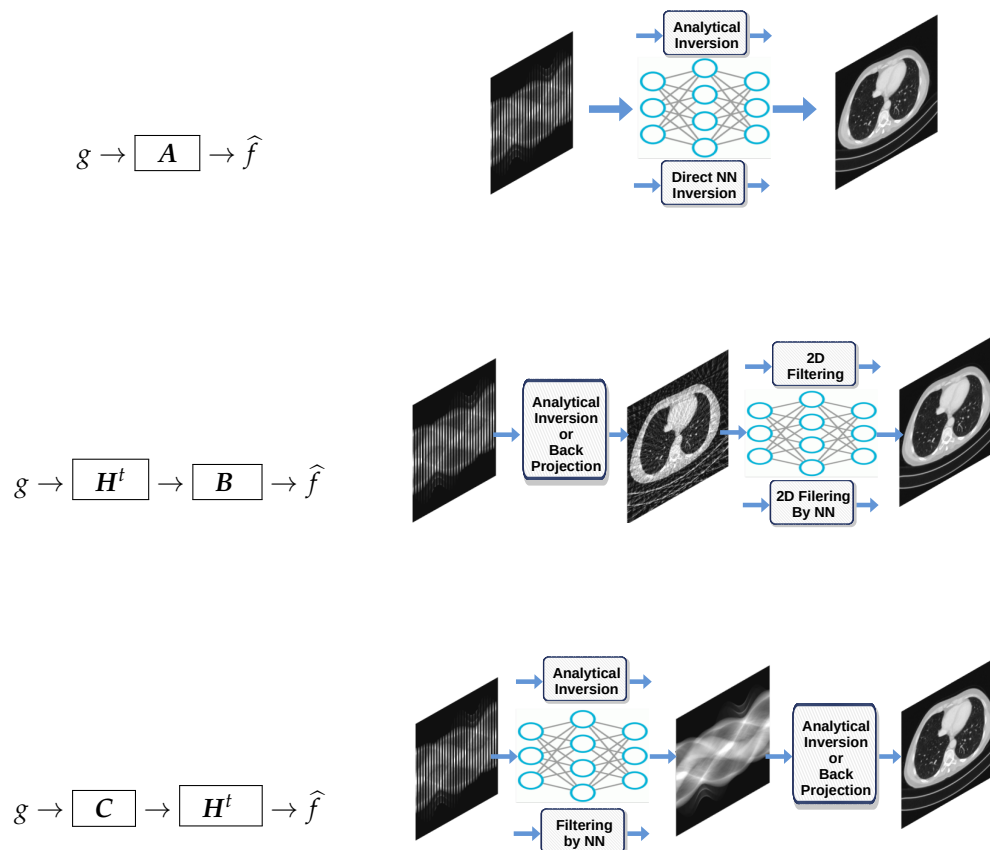


Figure 17. Three linear NN structures that are derived directly from quadratic regularization inversion method.

11.2. Second Example: Image Denoising with a Two-Layer CNN

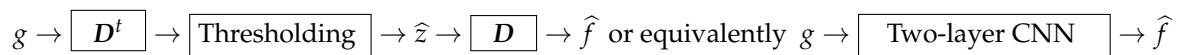
The second example is the denoising  $g = f + \epsilon$  with an L1 regularizer:

$$\hat{f} = D\hat{z} \text{ and } \hat{z} = \arg \min_z \{J(z)\} \text{ with } J(z) = \|g - Dz\| + \lambda \|z\|_1, \tag{22}$$

where  $D$  is a filter, i.e., a convolution operator. This can also be considered as the MAP estimator with a double exponential prior. It is easy to show that the solution can be obtained by a convolution followed by a thresholding [64–66].

$$\hat{f} = D\hat{z} \text{ and } \hat{z} = S_{\frac{\lambda}{\alpha}}(D^t g)$$

where  $S_{\lambda}$  is a thresholding operator.



11.3. Third Example: A Deep-Learning Equivalence

One of the classical iterative methods in linear inverse-problems algorithms is based on a gradient-based method to optimize  $J(f) = \|g - Hf\|^2$ :

$$f^{(k+1)} = f^{(k)} + \alpha H^t(g - Hf^{(k)}) = \alpha H^t g + (I - \alpha H^t H)f^{(k)}, \tag{23}$$

where the solution of the problem is obtained recursively. Everybody knows that when the forward model operator  $H$  is singular or ill-conditioned, this iterative algorithm starts by converging, but it may diverge easily. One of the experimental methods to obtain an acceptable approximate solution is just to stop the iterations after  $K$  iterations. This idea can be translated to a deep-learning NN by using  $K$  layers. Each layer represents one iteration of the algorithm. See Figures 18 and 19.

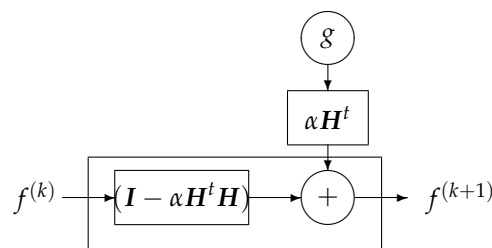


Figure 18. One bloc of iteration that can be considered as one layer of a NN.

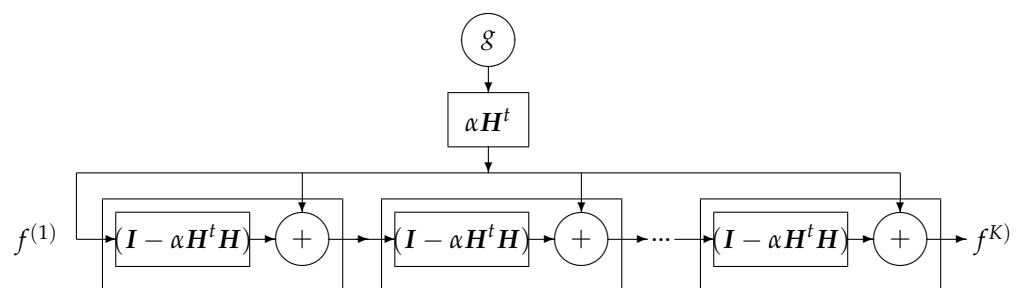


Figure 19. A  $K$  layers DL NN equivalent to  $K$  iterations of the basic optimization algorithm.

This DL structure can easily be extended to a regularized criterion:  $J(f) = \frac{1}{2} \|g - Hf\|^2 + \lambda \|Df\|^2$ , where

$$f^{(k+1)} = f^{(k)} + \alpha [H^t(g - Hf^{(k)}) - \lambda D^t D] = \alpha H^t g + (I - \alpha H^t H - \alpha \lambda D^t D)f^{(k)}. \tag{24}$$

We just need to replace  $(I - \alpha H^t H)$  by  $(I - \alpha H^t H - \alpha \lambda D^t D)$ .

This structure can also be extended to all the sparsity-enforcing regularization terms such as  $\ell_1$  and total variation (TV) using appropriate algorithms such as ISTA, FISTA, ADMM, etc. by replacing the update expression and by adding a NL operation much like the ordinary NNs. A simple example is given in the following subsection.

11.4. Fourth Example:  $\ell_1$  Regularization and NN

Let us consider the linear inverse problem  $g = Hf + \epsilon$  with  $\ell_1$  regularization criterion:

$$J(f) = \|g - Hf\|_2^2 + \lambda \|f\|_1,$$

and an iterative optimization algorithm, such as ISTA:

$$f^{(k+1)} = \text{Prox}_{\ell_1}(f^{(k)}, \lambda) \triangleq \mathcal{S}_{\lambda\alpha}(\alpha H^t g + (I - \alpha H^t H)f^{(k)}),$$

where  $\mathcal{S}_\theta$  is a soft thresholding operator, and  $\alpha \leq |\text{eig}(H^t H)|$  is the Lipschitz constant of the normal operator. When  $H$  is a convolution operator, then:

- $(I - \alpha H^t H)f^{(k)}$  can also be approximated by a convolution and thus considered as a filtering operator;
- $\frac{1}{\alpha} H^t g$  can be considered as a bias term and is also a convolution operator; and
- $\mathcal{S}_{\theta=\lambda\alpha}$  is a nonlinear point-wise operator. In particular, when  $f$  is a positive quantity, this soft thresholding operator can be compared to the ReLU activation function of NN. See Figure 20.

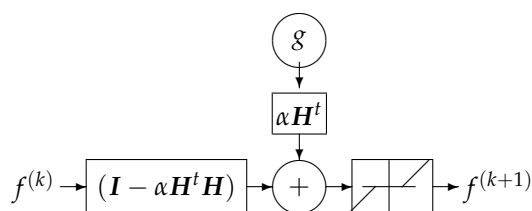


Figure 20. One block of a NN corresponds to one iteration of  $\ell_1$  regularization.

In all these three examples, we directly could obtain the structure of the NN from the forward model and known parameters. However, in these approaches, there are some difficulties, which consist in the determination of the structure of the NN. For example, in the first example, obtaining the structure of  $B$  depends on the regularization parameter  $\lambda$ . The same difficulty arises for determining the shape and the threshold level of the thresholding bloc of the network in the second example. The same need of the regularization parameter as well as many other hyperparameters is necessary to create the NN structure and weights. In practice, we can decide, for example, on the number and structure of a DL network, but as their corresponding weights depend on many unknown or difficult to fix parameters, ML may become of help. In the following, we first consider the training part of a general ML method. Then, we see how to include the physics-based knowledge of the forward model in the structure of learning.

12. ML General Approach

The ML approach can become helpful if we could have a great amount of data: inputs-outputs  $\{(f, g)_k, k = 1, 2, \dots, K\}$  examples. Thus, during the training step, we can learn the coefficients of the NN and then use it for obtaining a new solution  $\hat{f}$  for a new data  $g$ .

The main issue is the number of data input-output examples  $\{(f, g)_k, k = 1, 2, \dots, K\}$  we can have for the training step of the network.

*Fully Learned Method*

Let us consider a one-layer NN where the relation between its input  $g_k$  and output  $f_k$  is given by  $f_k = \phi(Wg_k)$  where  $W$  is the weighting parameters of the NN, and  $\phi$  is the point-wise non-linearity function of the output NN output layer. The estimation of  $W$  from the training data in the learning step is done by an optimization algorithm, which optimizes a loss function  $\mathcal{L}$  defined as

$$\mathcal{L} = \sum_{k=1}^K \ell_k(f_k, \phi(Wg_k)) \tag{25}$$

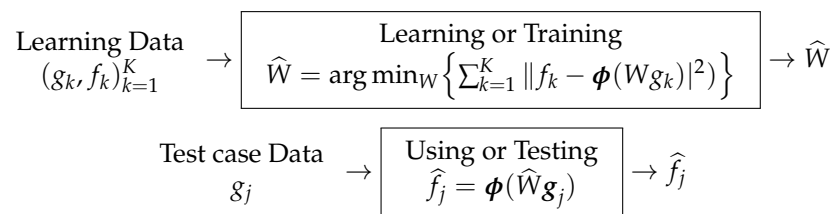
with

$$\ell_k(f_k, \phi(Wg_k)) = \|f_k - \phi(Wg_k)\|^2, \tag{26}$$

a quadratic distance or any other appropriate distance or divergence or a probabilistic one

$$\ell_k(f_k, \phi(Wg_k)) = E\{\|f_k - \phi(Wg_k)\|^2\}. \tag{27}$$

When the NN is trained and we obtain the weights  $\hat{W}$ , then we can use it easily when a new case (Test  $g_j$ ) appears, just by applying:  $f_j = \phi(\hat{W}g_j)$ . These two steps of training and using (also called testing) are illustrated in Figure 21.



**Figure 21.** Training (top) and testing (bottom) steps in a ML approach.

The scheme that we presented is general and can be extended to any multi-layer NN and DL. In fact, if we had a great number of data-ground-truth examples  $\{(f, g)_k, k = 1, 2, \dots, K\}$  with  $K$  much more than the number of elements  $W_{m,n}$  of the weighting parameters  $W$ , then, we would not even have any need for forward model  $H$ . This can be possible for very low dimensional problems [67–70]. However, in general, in practice, we do not have enough data. So, some prior or regularizer is needed to obtain a usable solution. This can be done just by adding a regularizer  $R(W)$  to the loss function (25) and (26), but we can also use the physics of the forward operator  $H$ .

**13. Physics-Based ML**

As mentioned above, in general, in practice, a rich-enough and complete data set is not often available, particularly for inverse problems. So, some prior or regularizer is needed to obtain a usable solution. Using a regularizer  $R(W)$  to the loss function (25) is good but is not enough. We have to use the physics of the forward operator  $H$ . This can be done in different ways.

*13.1. Decomposition of the NN Structure to Fixed and Trainable Parts*

The first, easiest, and understandable method consists in decomposing the structure of the network  $W$  in two parts: a fixed part and a learnable part. As the simplest example, we can consider the case of the analytical expression of the quadratic regularization:  $\hat{f} = (HH^t + \lambda DD^t)^{-1}H^t g = BH^t g$  which suggests to have a two-layer network with a fixed part structure  $H^t$  and a trainable one  $B = (HH^t + \lambda DD^t)^{-1}$ . See Figure 22.

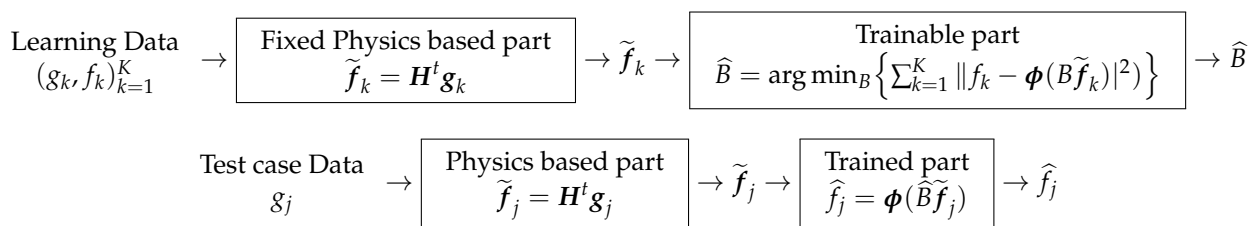


Figure 22. Training (top) and testing (bottom) steps in the first use of a physics-based ML approach.

It is interesting to note that, in X-ray computed tomography (CT), the forward operator  $H$  is called *projection*, the adjoint operator  $H^t$  is called *back-projection (BP)*, and the  $B$  operator is assimilated to a 2D filtering (convolution).

13.2. Using Singular-Value Decomposition of Forward and Backward Operators

Using the eigenvalues and eigenvectors of the pseudo or generalized inverse operators

$$H^\dagger = [H^t H]^{-1} H^t \quad \text{or} \quad H^\dagger = H^t [H H^t]^{-1} \tag{28}$$

and singular-value decomposition (SVD) of the operators  $[H^t H]$  and  $[H H^t]$  give another possible decomposition of the NN structure. Let us note

$$H H^t = U \Delta V' \quad \text{or equivalently} \quad H^t H = V \Delta U', \tag{29}$$

where  $\Delta$  is a diagonal matrix containing the singular values,  $U$  and  $V$ , containing the corresponding eigenvectors. This can be used to decompose the  $W$  to four operators:

$$W = V' \Delta U H^t \quad \text{or} \quad W = H^t V \Delta U', \tag{30}$$

where three of them can be fixed, and only one  $\Delta$  can be trainable. It is interesting to know that when the forward operator  $H$  has a shift-invariant (convolution) property, then the operators  $U$  and  $V'$  will correspond to the FT and IFT operators, respectively, and the diagonal elements of  $\Delta$  correspond to the FT of the impulse response of the convolution forward operator. So, we will have a fixed layer corresponding to  $H^t$ , which can be interpreted as a matched filtering, and a fixed FT layer, which is a feed-forward linear network, a trainable filtering part corresponding to the diagonal elements of  $\Delta$ , and a fourth fixed layer corresponding to IFT. See Figure 23.

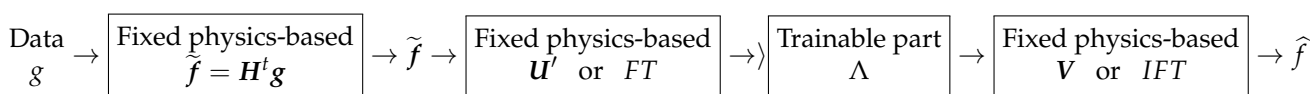
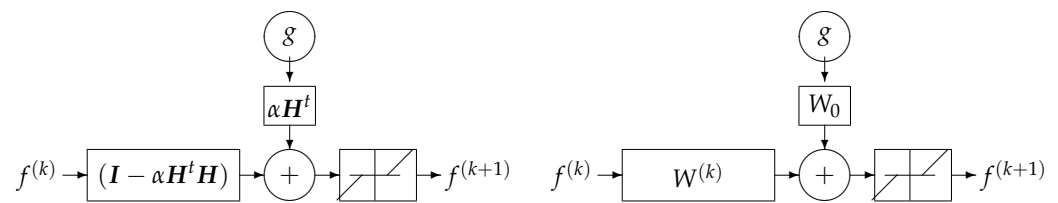


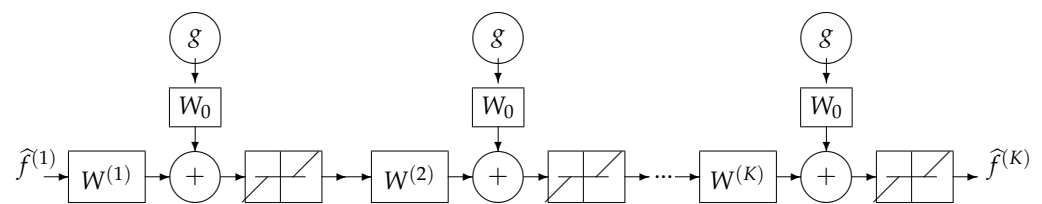
Figure 23. A four-layer NN with three physics-based fixed operators corresponding to  $H^t$ ,  $U'$  or  $FT$ , and  $V$  or  $IFT$  layers and one trainable layer corresponding to  $\Lambda$ .

DL Structure Based on Iterative Inversion Algorithm

Using the iterative gradient-based algorithm with a fixed number of iterations for computing a GI or a regularized one as explained in previous section can be used to propose a DL structure with  $K$  layers,  $K$  being the number of iterations before stopping. Figure 24 shows this structure for a quadratic regularization, which results to a linear NN and Figure 25 for the case of  $\ell_1$  regularization.



**Figure 24.** A  $K$  layers DL NN equivalent to  $K$  iterations of a basic gradient-based optimization algorithm. A quadratic regularization results in a linear NN, while a  $\ell_1$  regularization results in a classical NN with a nonlinear activation function. Left: supervised case. Right: unsupervised case. In both cases, all the  $K$  layers have the same structure.



**Figure 25.** All the  $K$  layers of DL NN equivalent to  $K$  iterations of an iterative gradient-based optimization algorithm. The simplest solution is to choose  $W_0 = \alpha H$  and  $W^{(k)} = W = (I - \alpha H^t H)$ ,  $k = 1, \dots, K$ . A more robust but more costly method is to learn all the layers  $W^{(k)} = (I - \alpha^{(k)} H^t H)$ ,  $k = 1, \dots, K$ .

**14. Conclusions and Challenges**

Signal and image processing (SIP), imaging systems (IS), computer vision (CV), machine learning (ML), and artificial intelligence (AI) have made great progress in the last forty years. The first category of the methods in signal and image processing was based on linear transformation followed by a thresholding or windowing and coming back. The second generation was model based: the forward-modeling and the inverse-problems approach. The main successful approach was based on regularization methods using a combined criterion. The third generation was model based but probabilistic and used the Bayes rule, which is the Bayesian approach.

Classical methods for inverse problems are mainly based on regularization methods, particularly those that are based on the optimization of a criterion with two parts: a data-model matching part and a regularization term. A great number of methods have been proposed for choosing these two parts and proposing appropriate optimization algorithms. A Bayesian Maximum A Posteriori (MAP) interpretation for these regularization methods can be given where these two terms correspond to the likelihood and prior probability models, respectively.

The Bayesian approach gives more flexibility in different aspects: (i) in choosing these terms and, in particular, the prior term via hierarchical models and hidden variables; (ii) a more-extended class of prior models can be obtained, particularly via the hierarchical prior models; (iii) determination of the regularization parameter, and more generally all the hyperparameters, can also be estimated; (iv) all the uncertainties are accounted for, and all the remaining uncertainties can be evaluated.

However, the Bayesian computations can become very heavy computationally, particularly when we want compute the uncertainties (variances and covariances) and when we want also to estimate the hyperparameters. Recently, the machine-learning (ML) methods have become a good help for some aspects of these difficulties.

Nowadays, ML, neural networks (NN), convolutional NN (CNN), and deep-learning (DL) methods have obtained great success in classification, clustering, object detection, speech and face recognition, etc. However, they need a great number of training data, still lack explanation, and they may fail very easily.

For inverse problems, they still need progress. In fact, using only data-based NN without any specific structure coming from the forward model (physics) is just an illusion. However, the progress arrives via their interaction with the model-based methods. In fact, the success of CNN and DL methods greatly depends on the appropriate choice of the network structure. This choice can be guided by the model-based methods.

In this work, a few examples of such interactions are described. As we could see, the main contribution of ML and NN tools can be in reducing the costs of the inversion method when an appropriate model is trained. However, to obtain a good model, there is a need for sufficiently rich data and a good network structure obtained from the physics knowledge of the problem in hand.

For inverse problems, when the forward models are non-linear and complex, NN and DL may be of great help. However, we may still need to choose the structure of the NN via an approximate forward model and approximate Bayesian inversion.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Bioucas-Dias, J.M.; Figueiredo, M.A. An iterative algorithm for linear inverse problems with compound regularizers. In Proceedings of the 2008 15th IEEE International Conference on Image Processing, San Diego, CA, USA, 12–15 October 2008; pp. 685–688.
2. Ghadimi, E.; Teixeira, A.; Shames, I.; Johansson, M. Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems. *IEEE Trans. Autom. Control* **2015**, *60*, 644–658. [[CrossRef](#)]
3. Chambolle, A.; Dossal, C. On the convergence of the iterates of “FISTA”. *J. Optim. Theory Appl.* **2015**, *166*, 25.
4. Ayasso, H.; Duchêne, B.; Mohammad-Djafari, A. MCMC and variational approaches for Bayesian inversion in diffraction imaging. *Regul. Bayesian Methods Inverse Probl. Signal Image Process.* **2015**, *201*, 224.
5. Florea, M.I.; Vorobyov, S.A. A robust FISTA-like algorithm. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 4521–4525.
6. Mohammad-Djafari, A. Inverse problems in imaging science: From classical regularization methods to state of the art Bayesian methods. In Proceedings of the International Image Processing, Applications and Systems Conference, New York, NY, USA, 12–14 December 2014; pp. 1–2. [[CrossRef](#)]
7. Mohammad-Djafari, A. Bayesian inference with hierarchical prior models for inverse problems in imaging systems. In Proceedings of the 2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA), Algiers, Algeria, 12–15 May 2013; pp. 7–18. [[CrossRef](#)]
8. Mohammad-Djafari, A. Bayesian approach with prior models which enforce sparsity in signal and image processing. *EURASIP J. Adv. Signal Process.* **2012**, *2012*, 52. [[CrossRef](#)]
9. Ayasso, H.; Duchêne, B.; Mohammad-Djafari, A. Bayesian inversion for optical diffraction tomography. *J. Mod. Opt.* **2010**, *57*, 765–776. [[CrossRef](#)]
10. Ren, D.; Zhang, K.; Wang, Q.; Hu, Q.; Zuo, W. Neural Blind Deconvolution Using Deep Priors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020), Online, 14–19 June 2020; pp. 3338–3347.
11. Bertocchi, C.; Chouzenoux, E.; Corbineau, M.C.; Pesquet, J.C.; Prato, M. Deep Unfolding of a Proximal Interior Point Method for Image Restoration. *Inverse Probl.* **2019**, *36*, 034005. [[CrossRef](#)]
12. Jospin, L.V.; Buntine, W.L.; Boussaid, F.; Laga, H.; Bennamoun, M. Hands-on Bayesian Neural Networks—A Tutorial for Deep Learning Users. *arXiv* **2020**, arXiv:2007.06823.
13. Monga, V.; Li, Y.; Eldar, Y.C. Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing. *IEEE Signal Process. Mag.* **2021**, *38*, 18–44. [[CrossRef](#)]
14. Gilton, D.; Ongie, G.; Willett, R. Model Adaptation for Inverse Problems in Imaging. *IEEE Trans. Comput. Imaging* **2021**, *7*, 661–674. [[CrossRef](#)]
15. Repetti, A.; Pereyra, M.; Wiaux, Y. Scalable Bayesian Uncertainty Quantification in Imaging Inverse Problems via Convex Optimization. *SIAM J. Imaging Sci.* **2019**, *12*, 87–118. [[CrossRef](#)]
16. Mohammad-Djafari, A. Gauss-Markov-Potts Priors for Images in Computer Tomography resulting to Joint Optimal Reconstruction and Segmentation. *Int. J. Tomogr. Stat.* **2008**, *11*, 76–92.

17. Ayasso, H.; Mohammad-Djafari, A. Joint NDT image restoration and segmentation using Gauss–Markov–Potts prior models and variational bayesian computation. *IEEE Trans. Image Process.* **2010**, *19*, 2265–2277. [[CrossRef](#)] [[PubMed](#)]
18. Féron, O.; Duchêne, B.; Mohammad-Djafari, A. Microwave imaging of inhomogeneous objects made of a finite number of dielectric and conductive materials from experimental data. *Inverse Probl.* **2005**, *21*, S95. [[CrossRef](#)]
19. Chapdelaine, C.; Mohammad-Djafari, A.; Gac, N.; Parra, E. A 3D Bayesian Computed Tomography Reconstruction Algorithm with Gauss-Markov-Potts Prior Model and its Application to Real Data. *Fundam. Inform.* **2017**, *155*, 373–405. [[CrossRef](#)]
20. Chun, I.Y.; Huang, Z.; Lim, H.; Fessler, J. Momentum-Net: Fast and convergent iterative neural network for inverse problems. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**. [[CrossRef](#)] [[PubMed](#)]
21. Sandhu, A.I.; Shaikat, S.A.; Desmal, A.; Bagci, H. ANN-assisted CoSaMP Algorithm for Linear Electromagnetic Imaging of Spatially Sparse Domains. *IEEE Trans. Antennas Propag.* **2021**. [[CrossRef](#)]
22. Fang, Z. A High-Efficient Hybrid Physics-Informed Neural Networks Based on Convolutional Neural Network. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–13. [[CrossRef](#)] [[PubMed](#)]
23. Husain, Z.; Madjid, N.A.; Liatsis, P. Tactile sensing using machine learning-driven Electrical Impedance Tomography. *IEEE Sens. J.* **2021**. [[CrossRef](#)]
24. Ongie, G.; Jalal, A.; Metzler, C.A.; Baraniuk, R.G.; Dimakis, A.G.; Willett, R. Deep Learning Techniques for Inverse Problems in Imaging. *IEEE J. Sel. Areas Inf. Theory* **2020**, *1*, 39–56. [[CrossRef](#)]
25. Gilton, D.; Ongie, G.; Willett, R. Neumann Networks for Linear Inverse Problems in Imaging. *IEEE Trans. Comput. Imaging* **2020**, *6*, 328–343. [[CrossRef](#)]
26. Gong, D.; Zhang, Z.; Shi, Q.; van den Hengel, A.; Shen, C.; Zhang, Y. Learning Deep Gradient Descent Optimization for Image Deconvolution. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 5468–5482. [[CrossRef](#)] [[PubMed](#)]
27. Sun, J.; Zhang, J.; Zhang, X.; Zhou, W. A Deep Learning-Based Method for Heat Source Layout Inverse Design. *IEEE Access* **2020**, *8*, 140038–140053. [[CrossRef](#)]
28. Peng, P.; Jalali, S.; Yuan, X. Solving Inverse Problems via Auto-Encoders. *IEEE J. Sel. Areas Inf. Theory* **2020**, *1*, 312–323. [[CrossRef](#)]
29. Zhang, X.; Li, B.; Jiang, J. Hessian Free Convolutional Dictionary Learning for Hyperspectral Imagery With Application to Compressive Chromo-Tomography. *IEEE Access* **2020**, *8*, 104216–104231. [[CrossRef](#)]
30. de Haan, K.; Rivenson, Y.; Wu, Y.; Ozcan, A. Deep-Learning-Based Image Reconstruction and Enhancement in Optical Microscopy. *Proc. IEEE* **2020**, *108*, 30–50. [[CrossRef](#)]
31. Lu, F.; Wu, J.; Huang, J.; Qiu, X. Restricted-Boltzmann-Based Extreme Learning Machine for Gas Path Fault Diagnosis of Turbofan Engine. *IEEE Trans. Ind. Inform.* **2020**, *16*, 959–968. [[CrossRef](#)]
32. Zheng, J.; Peng, L. A Deep Learning Compensated Back Projection for Image Reconstruction of Electrical Capacitance Tomography. *IEEE Sens. J.* **2020**, *20*, 4879–4890. [[CrossRef](#)]
33. Ren, S.; Sun, K.; Tan, C.; Dong, F. A Two-Stage Deep Learning Method for Robust Shape Reconstruction With Electrical Impedance Tomography. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 4887–4897. [[CrossRef](#)]
34. Yang, G.; Yu, S.; Dong, H.; Slabaugh, G.; Dragotti, P.L.; Ye, X.; Liu, F.; Arridge, S.; Keegan, J.; Guo, Y.; et al. DAGAN: Deep De-Aliasing Generative Adversarial Networks for Fast Compressed Sensing MRI Reconstruction. *IEEE Trans. Med. Imaging* **2018**, *37*, 1310–1321. [[CrossRef](#)] [[PubMed](#)]
35. Stahlhut, C.; Morup, M.; Winther, O.; Hansen, L.K. Hierarchical Bayesian model for simultaneous EEG source and forward model reconstruction (SOFOMORE). In Proceedings of the 2009 IEEE International Workshop on Machine Learning for Signal Processing, Grenoble, France, 1–4 September 2009; pp. 1–6. [[CrossRef](#)]
36. Idier, J. *Bayesian Approach to Inverse Problems*; John Wiley & Sons: Hoboken, NJ, USA, 2008.
37. Mohammad-Djafari, A. *Problèmes Inverses en Imagerie et en Vision en Deux Volumes Inséparables*; Traité Signal et Image, IC2; ISTE-WILEY: Hoboken, NJ, USA, 2009.
38. Mohammad-Djafari, A. *Inverse Problems in Vision and 3D Tomography*; ISTE-Wiley: Hoboken, NJ, USA, 2010.
39. Penrose, R. A Generalized Inverse for Matrices. *Math. Proc. Camb. Philos. Soc.* **1955**, *51*, 406–413. [[CrossRef](#)]
40. Carasso, A.S. Direct Blind Deconvolution. *SIAM J. Appl. Math.* **2001**, *61*, 1980–2007. [[CrossRef](#)]
41. Chan, T.; Wong, C.K. Total variation blind deconvolution. *IEEE Trans. Image Process.* **1998**, *7*, 370–375. [[CrossRef](#)] [[PubMed](#)]
42. Chu, N.; Mohammad-Djafari, A.; Gac, N.; Picheral, J. A variational Bayesian approximation approach via a sparsity enforcing prior in acoustic imaging. In Proceedings of the 2014 13th Workshop on Information Optics (WIO), Neuchatel, Switzerland, 7–11 July 2014; pp. 1–4. [[CrossRef](#)]
43. Chu, N.; Mohammad-Djafari, A.; Picheral, J. Robust Bayesian super-resolution approach via sparsity enforcing a priori for near-field aeroacoustic source imaging. *J. Sound Vib.* **2013**, *332*, 4369–4389. [[CrossRef](#)]
44. Chu, N.; Ning, Y.; Yu, L.; Huang, Q.; Wu, D. A Fast and Robust Localization Method for Low-Frequency Acoustic Source: Variational Bayesian Inference based on Non-Synchronous Array Measurements. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 1–18. [[CrossRef](#)]
45. Chu, N.; Zhao, H.; Yu, L.; Huang, Q.; Ning, Y. Fast and High-Resolution Acoustic Beamforming: A Convolution Accelerated Deconvolution Implementation. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 1–15. [[CrossRef](#)]
46. Gharsalli, L.; Duchêne, B.; Mohammad-Djafari, A.; Ayasso, H. Microwave tomography for breast cancer detection within a variational Bayesian approach. In Proceedings of the 21st European Signal Processing Conference (EUSIPCO 2013), Marrakech, Morocco, 9–13 September 2013; pp. 1–5.



47. Gharsalli, L.; Duchêne, B.; Mohammad-Djafari, A.; Ayasso, H. A Gauss-Markov mixture prior model for a variational Bayesian approach to microwave breast imaging. In Proceedings of the 2014 IEEE Conference on Antenna Measurements Applications (CAMA), Vasteras, Sweden, 3–6 September 2014; pp. 1–4. [\[CrossRef\]](#)
48. Premel, D.; Mohammad-Djafari, A. Eddy current tomography in cylindrical geometry. *IEEE Trans. Magn.* **1995**, *31*, 2000–2003. [\[CrossRef\]](#)
49. Kak, A.C.; Slaney, M. *Principles of Computerized Tomographic Imaging*; SIAM: Philadelphia, PA, USA, 2001.
50. Jackson, J.I.; Meyer, C.H.; Nishimura, D.G.; Macovski, A. Selection of a convolution function for Fourier inversion using gridding [computerized tomography application]. *IEEE Trans. Med. Imaging* **1991**, *10*, 473–478. [\[CrossRef\]](#)
51. Osher, S.; Burger, M.; Goldfarb, D.; Xu, J.; Yin, W. An iterative regularization method for total variation-based image restoration. *Multiscale Model. Simul.* **2005**, *4*, 460–489. [\[CrossRef\]](#)
52. Wang, Y.; Yang, J.; Yin, W.; Zhang, Y. A new alternating minimization algorithm for total variation image reconstruction. *SIAM J. Imaging Sci.* **2008**, *1*, 248–272. [\[CrossRef\]](#)
53. Goldstein, T.; Osher, S. The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci.* **2009**, *2*, 323–343. [\[CrossRef\]](#)
54. Sun, Y.; Wu, Z.; Xu, X.; Wohlberg, B.; Kamilov, U.S. Scalable Plug-and-Play ADMM With Convergence Guarantees. *IEEE Trans. Comput. Imaging* **2021**, *7*, 849–863. [\[CrossRef\]](#)
55. Mohammad-Djafari, A. Joint estimation of parameters and hyperparameters in a Bayesian approach of solving inverse problems. In Proceedings of the 3rd IEEE International Conference on Image Processing, Lausanne, Switzerland, 16–19 September 1996; Volume 1, pp. 473–476. [\[CrossRef\]](#)
56. Mohammad-Djafari, A.; Ayasso, H. Variational Bayes and mean field approximations for Markov field unsupervised estimation. In Proceedings of the 2009 IEEE International Workshop on Machine Learning for Signal Processing, Grenoble, France, 1–4 September 2009; pp. 1–6.
57. Blei, D.; Kucukelbir, A.; McAuliffe, J. Variational Inference: A Review for Statisticians. *J. Am. Stat. Assoc.* **2016**, *112*, 859–877. [\[CrossRef\]](#)
58. Wang, L.; Mohammad-Djafari, A.; Gac, N. X-ray Computed Tomography using a sparsity enforcing prior model based on Haar transformation in a Bayesian framework. *Fundam. Inform.* **2017**, *155*, 449–480. [\[CrossRef\]](#)
59. Wang, L.; Gac, N.; Mohammad-Djafari, A. Bayesian 3D X-ray computed tomography image reconstruction with a scaled Gaussian mixture prior model. *AIP Conf. Proc.* **2015**, *1641*, 556–563.
60. Chapdelaine, C.; Mohammad-Djafari, A.; Gac, N.; Parra, E. A Joint Segmentation and Reconstruction Algorithm for 3D Bayesian Computed Tomography Using Gauss-Markov-Potts Prior Model. In Proceedings of the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017.
61. Chapdelaine, C.; Gac, N.; Mohammad-Djafari, A.; Parra, E. New GPU implementation of Separable Footprint Projector and Backprojector: First results. In Proceedings of the 5th International Conference on Image Formation in X-Ray Computed Tomography, Baltimore, MD, USA, 4 November 2018.
62. Chapdelaine, C. Variational Bayesian Approach and Gauss-Markov-Potts prior model. *arXiv* **2018**, arXiv:1808.09552.
63. Ciresan, D.C.; Meier, U.; Masci, J.; Gambardella, L.M.; Schmidhuber, J. Flexible, High Performance Convolutional Neural Networks for Image Classification. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011; pp. 1237–1242.
64. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [\[CrossRef\]](#)
65. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. *arXiv* **2013**, arXiv:1311.2901.
66. Nan, Y.; Quan, Y.; Ji, H. Variational-EM-Based Deep Learning for Noise-Blind Image Deblurring. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020), Online, 13–19 June 2020.
67. Saul, L.K.; Roweis, S.T. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *J. Mach. Learn. Res.* **2003**, *4*, 119–155.
68. Arel, I.; Rose, D.C.; Karnowski, T.P. Deep machine learning—A new frontier in artificial intelligence research. *Comput. Intell. Mag.* **2010**, *5*, 13–18. [\[CrossRef\]](#)
69. Cho, K.; Raiko, T.; Ilin, A. Enhanced gradient for training restricted Boltzmann machines. *Neural Comput.* **2013**, *25*, 805–831. [\[CrossRef\]](#)
70. Cho, K. Foundations and Advances in Deep Learning. Ph.D. Thesis, Aalto University School of Science, Espoo, Finland, 2014.