Probabilistic inversions of electrical resistivity tomography data with a machine learning-based forward operator

Mattia Aleardi°, Alessandro Vinciguerra*,°, Eusebio Stucchi°, Azadeh Hojat+,§

°University of Pisa, Earth Sciences Department, via S. Maria 53, 56126, Pisa, Italy

*University of Florence, Earth Sciences Department, via. G. La Pira 4, 50121, Florence, Italy

+Shahid Bahonar University of Kerman, Department of Mining Engineering, Jomhouri Blvd.,
Kerman, 76188, Iran

§Politecnico di Milano, Department of Civil and Environmental Engineering, Piazza Leonardo da
Vinci 32, 20133, Milano Italy

Corresponding author: Mattia Aleardi, mattia.aleardi@unipi.it

ABSTRACT

Casting a geophysical inverse problem into a Bayesian setting is often discouraged by the computational workload needed to run many forward modeling evaluations. Here we present probabilistic inversions of electrical resistivity tomography data in which the forward operator is replaced by a trained residual neural network that learns the non-linear mapping between the resistivity model and the apparent resistivity values. The use of this specific architecture can provide some advantages over standard convolutional networks as it mitigates the vanishing gradient problem that might affect deep networks. The modeling error introduced by the network approximation is properly taken into account and propagated onto the estimated model uncertainties. One crucial aspect of any machine learning application is the definition of an appropriate training set. We draw the models forming the training and validation sets from previously defined prior distributions, while a finite element code provides the associated datasets. We apply the approach to two probabilistic inversion frameworks: a Markov Chain Monte Carlo algorithm is applied to synthetic data, while an ensemble-based algorithm is employed for the field measurements. For both the synthetic and field tests, the outcomes of the proposed method are benchmarked against the predictions obtained when the finite element code constitutes the forward operator. Our experiments illustrate that the network can effectively approximate the forward mapping even when a relatively small training set is created. The proposed strategy provides a forward operator three that is orders of magnitude faster than the accurate but computationally expensive finite element code. Our approach also yields most likely solutions and uncertainty quantifications comparable to those estimated when the finite element modeling is employed. The presented method allows solving the Bayesian electrical resistivity tomography with a reasonable computational cost and limited hardware resources.

Keywords: Electrical Resistivity Survey; Inversion;

INTRODUCTION

Geophysical inversion exploits the measured experimental data to infer the distribution of physical parameters in the subsurface. From the mathematical point of view, this process is an inverse problem that is often ill-posed (Tarantola, 2005; Aster et al. 2018) meaning that many models reproduce the observed data equally well. For this reason, one key aspect of geophysical inversions is the quantification of the uncertainties affecting the recovered solution. The Bayesian setting is usually adopted to accurately propagate the uncertainties from the available geophysical data onto the estimated parameters. In this context, the solution of an inverse problem is expressed by the posterior probability density (PPD) function in the model space (Tarantola, 2005). However, the PPD can be expressed in a closed-form only for linear problems with Gaussian assumptions about the data and model parameter distributions. For a non-linear relation linking the model to the data, or for non-parametric prior assumptions, a numerical assessment of the PPD is needed.

In this context, Markov Chain Monte Carlo (MCMC) algorithms (Sambridge and Mosegaard, 2002) constitute a possible approach to numerically solve a probabilistic inversion. The increasing computational power provided by modern parallel architectures has considerably promoted the applications of these methods to solve geophysical problems (Dosso et al. 2012; Sen and Stoffa 2013; Ray et al. 2017; Grana et al. 2021). However, their use remains a formidable computational task in problems with expensive forward operators and with many unknown parameters. Several strategies have been proposed to mitigate this issue for example by employing compression strategies (Grana et al. 2019) to reduce the dimensionality of the parameter space or by exploiting the information about the gradient of the error function to guide the probabilistic sampling (Sen and Biswas, 2017; Fichtner et al. 2019; Gebraad et al. 2020; Zhao and Sen, 2021; Aleardi and Salusti, 2020; Aleardi 2021).

Ensemble-based data assimilation methods such as ensemble smoother with multiple data assimilation (ES-MDA) (Emerick and Reynolds, 2013) constitute an efficient alternative to MCMC algorithms because they are computationally faster but might underestimate the model uncertainty in high-dimensional parameter and data spaces. This undesirable phenomenon is usually called ensemble collapse (Sætrom and Omre, 2013) and can be mitigated either by compressing model and data spaces (Luo et al., 2018) or by enlarging the number of models forming the ensemble, although this strategy increases the computational demanding.

Over the last years, machine learning algorithms (Monajemi et al., 2016; Goodfellow et al. 2016) have been increasingly applied to solve geophysical problems (Araya-Polo et al. 2018; Richardson, 2018; Waldeland et al. 2018; Wang et al. 2019; Wu and McMechan 2019; Puzyrev 2019; Park and Sacchi, 2020; Sun 2020; Aleardi 2020b; Moghadas 2020; Aleardi and Salusti, 2021). In particular, Convolutional Neural Networks (CNNs) are very popular architectures that exploit sparse connectivity and sharing weights among convolutional layers to reduce the computational cost of the training phase and improve the generalization ability (LeCun et al, 2015; Krizhevsky et al. 2017). The number of convolutional layers in a CNN plays a crucial role because a deeper network can potentially better approximate non-linear functions. It is usually believed that the deeper the network, the higher is the accuracy (match between actual and desired and network responses). However, it is often experienced that as the network gets deeper its performances degrade on both training and validation sets. This should not be confused with the well-known overfitting issue that usually manifests as a higher prediction error on the validation set. Instead, it may result from the optimization function, activation function, initialization of the network, or more importantly, from the vanishing gradient problem. This represents a crucial limitation when training deep networks (i.e., constituted by many convolutional layers) with derivative-based backpropagation algorithms (LeCun et al., 1998) and occurs when the network is unable to backpropagate the gradient information from deep to shallow layers. As a consequence the degradation problem arises: the accuracy gets saturated for a given number of layers and then starts degrading rapidly if additional layers are added. He et al. (2016) proposed a Residual Neural Network (ResNet) to solve this problem, in which layers are connected through skip connections that add the outcomes of a shallow layer to the output of a deeper one.

The electrical resistivity tomography (ERT) method is one of the most widely used geophysical techniques that provides the subsurface resistivity distribution for a variety of hydrogeological, environmental, and engineering problems (e.g., Dahlin 2001; Rucker et al. 2011; Chambers et al. 2014; Uhlemann et al. 2015; Moradipour et al. 2016; Whiteley et al. 2017; Bièvre et al. 2018; Hojat et al. 2019a; Dahlin 2020; Hermans and Paepen 2020; Aleardi et al. 2020; Loke et al. 2020; Norooz et al. 2021). Due to incomplete data coverage and noise contamination, the ERT is an ill-posed problem affected by non-uniqueness and instability of the solution (i.e., small variations of the data produce large perturbations in the predictions), and hence, an accurate estimation of the model uncertainty is of primary importance. However, the ERT is routinely solved through deterministic approaches in which optimization algorithms minimize a predefined objective function. Such methods are generally computationally efficient but provide an estimation of the model (i.e., the most likely solution) without accurately quantifying the associated uncertainty. On the other hand, the computing time needed for multiple forward evaluations (e.g., through a finite element code) hampers the application of probabilistic approaches to solving the ERT problem and in this context, either advanced MCMC recipes or model and data compression strategies are usually needed (Aleardi et al. 2020; Vinciguerra et al. 2021) to keep the computational cost affordable.

In this work, we reduce the computational burden of the probabilistic ERT inversion by replacing the code for the forward evaluation with a trained ResNet. The idea is to use a trained network as a computationally efficient approximation to the forward problem, while also accounting for the approximation error introduced by the network and propagating it onto the final PPD (using the same strategy presented in Hansen and Cordua, 2017). Multiple models and associated apparent resistivity data are used to make the ResNet learn the non-linear mapping between the model and the data space. The models forming the training and validation sets are generated according to prior model assumptions, while a 2.5D finite-elements (FE) Matlab modeling code constitutes the forward operator (Karaoulis et al., 2013) that computes the associated apparent resistivity data. We first demonstrate the method by inverting synthetic data, then the method is tested on field measurements. In the synthetic case, we employ the Differential Evolution Markov Chian (DEMC; Vrugt, 2016) to numerically assess the PPD, while the field inversion is solved through the ES-MDA algorithm. The outcomes of the proposed approach are also benchmarked with those yielded by DEMC and ES-MDA inversions in which the FE code constitutes the forward modeling engine. Other studies have already employed machine learning techniques to solve the ERT problem and especially to approximate the non-linear inverse mapping (Liu et al. 2020; Vu and Jardani, 2021; Aleardi et al. 2021a), but as far as the authors are aware, this is the first paper in which machine learning is used to speed up probabilistic ERT inversions. In this study, all the inversion codes have been written in Matlab, and the Matlab deep learning toolbox has been used to implement CNN and ResNet.

METHODOLOGY

The Bayesian framework and the probabilistic inversion

The Bayesian solution of an inverse problem is expressed by the posterior probability density (PPD) function in model space:

$$p(\mathbf{m}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{m})p(\mathbf{m})}{p(\mathbf{d})}, \quad (1)$$

where $p(\mathbf{m}|\mathbf{d})$ denotes the PPD, $p(\mathbf{d}|\mathbf{m})$ is the so-called data likelihood function, whereas $p(\mathbf{m})$ and $p(\mathbf{d})$ are the prior distributions of model parameters $\mathbf{m}$ and observed data $\mathbf{d}$, respectively. In most cases the data likelihood is derived from the L2 norm difference between predicted and observed data, under the assumption of Gaussian-distributed noise:

$$p(\mathbf{d}|\mathbf{m}) \propto -0.5 \times \left(\mathbf{d} - G(\mathbf{m})\right)^T \mathbf{C}_\mathbf{d}^{-1}\left(\mathbf{d} - G(\mathbf{m})\right), \quad (2)$$

in which $\mathbf{C_d}$ is the data covariance, and G is the forward operator. For non-linear problems, a numerical evaluation of the posterior can be derived using, for example, MCMC sampling algorithms or ensemble-based methods. The main computational demand when solving a Bayesian non-linear inverse problem lies in the computation of the likelihood because it requires running a forward evaluation for each sampled model. For this reason, our work aims to replace the computationally expensive forward operator G with the predictions of a trained network. The use of such approximation introduces a modeling error that if ignored can generate overfitting with the observed data and introduce artifacts in the final solution. Therefore, we also properly propagate the error introduced by the network approximation onto the final PPD. To this end the data covariance matrix $\mathbf{C_d}$ is computed as the sum of the noise contaminating the data $\mathbf{C_n}$ and the modeling error that takes into account the imperfect physics relating the model to the data $\mathbf{C_p}$ (Menke, 2018): $\mathbf{C_d} = \mathbf{C_n} + \mathbf{C_p}$. Both noise and modeling errors are considered to be Gaussian-distributed with a zero mean value. The modeling error matrix is derived by evaluating the covariance of the difference between desired and actual network outputs and is computed on the validation set (see Hansen and Cordua, 2017 for details). With desired output, we mean the data generated by the FE routine that are assumed to be perfect, error-free predictions of the apparent resistivity values.

The differential evolution Markov chain

Markov chain Monte Carlo methods sample the target PPD by adopting the Metropolis-Hasting rule that defines the probability to move from the current state of the chain (i.e. the current model $\mathbf{m}$) to the proposed (perturbed) state $\mathbf{m}'$ as follows:

$$\boldsymbol{\alpha} = \boldsymbol{p}(\mathbf{m}'|\mathbf{m}) = \ \mathbf{min}\left[\mathbf{1}, \frac{\boldsymbol{p}(\mathbf{m}')}{\boldsymbol{p}(\mathbf{m})} \times \frac{\boldsymbol{p}(\mathbf{d}|\mathbf{m}')}{\boldsymbol{p}(\mathbf{d}|\mathbf{m})} \times \frac{\boldsymbol{q}(\mathbf{m}|\mathbf{m}')}{\boldsymbol{q}(\mathbf{m}'|\mathbf{m})}\right]. \quad (\mathbf{3})$$

q() denotes the proposal distribution that defines the new state $\mathbf{m}'$ as a random deviate from a probability distribution $q(\mathbf{x}'|\mathbf{x})$ conditioned only on the current state $\mathbf{m}$. If $\mathbf{m}'$ is accepted $\mathbf{m} = \mathbf{m}'$. Otherwise, $\mathbf{m}$ is repeated in the chain, and another state is generated. The ensemble of sampled models after the burn-in period is used to numerically compute the statistical properties of the PPD (e.g. mean, mode, standard deviations, marginal densities). Many sampling recipes have been proposed to reduce the computational cost of MCMC inversions (Vrugt, 2016), and here, we employ the Differential Evolution Markov Chain (DEMC) that exploits differential evolution principles to guide the sampling procedure. In DEMC *N* Markov chains and multivariate proposals are generated from the collection of chains at each iteration. Let the d-vector $\mathbf{m}$ be the state of a single chain, then at each *t*-1 iteration the *N* chains define a population $\mathbf{M} = \{\mathbf{m}_{t-1}^1, \mathbf{m}_{t-1}^2, \dots, \mathbf{m}_{t-1}^N\}$. A proposal model $\mathbf{m}_p$ is then generated for each chain according to:

$$\mathbf{m}_p^i = \mathbf{m}_{t-1}^i + \gamma\left(\mathbf{m}_{t-1}^a - \mathbf{m}_{t-1}^b\right) + \epsilon, \quad a \neq b \neq i \quad (4)$$

where γ is the jump rate, i denotes the considered chain, whereas a and b are integer values drawn from {1,…,*i*-1,*i*+1,…,*N*} without replacement; ϵ is a normally distributed random deviate $\epsilon = \mathcal{N}(0, \sigma)$, where σ is properly set for the problem at hand. Each proposal is accepted according to the Metropolis-Hasting rule. The probability of $\gamma = 1$ is usually set to 10% and this allows mode-jumping which is a significant strength of DEMC over more conventional MCMC algorithms (i.e., random walk Metropolis). In this work, we solve the probabilistic ERT inversion using the DEMC algorithm presented in Vinciguerra et al. (2021) in which the Discrete Cosine Transform (DCT) reparameterization is used to compress the model space and to make the MCMC sampling computationally feasible. We refer the reader to that publication for further details.

The ensemble smoother with multiple data assimilation

The ES-MDA is an iterative procedure in which the updated models are used as the prior in the next iteration. The method starts with an ensemble of models generated according to the prior assumptions. Then, these models are updated by applying a Bayesian updating step to a stochastic observation of the data $\tilde{\mathbf{d}}_k$ under model and data Gaussian assumptions with empirical parameters estimated from the ensemble members. A single ES-MDA iteration can be written as:

$$\mathbf{m}_k^u = \mathbf{m}_k^p + \tilde{\mathbf{C}}_{\mathbf{md}}^p(\tilde{\mathbf{C}}_{\mathbf{dd}}^p + \mathbf{C}_{\mathbf{d}})^{-1}\left(\tilde{\mathbf{d}}_k - \mathbf{d}_k^p\right), \qquad (5)$$

where:

$$\tilde{\mathbf{C}}_{\mathbf{md}}^p = \frac{1}{Q-1}\sum_{k=1}^{Q}\left(\mathbf{m}_k^p - \bar{\mathbf{m}}^p\right)\left(\mathbf{d}_k^p - \bar{\mathbf{d}}^p\right)^T, \qquad (6)$$

$$\tilde{\mathbf{C}}_{\mathbf{dd}}^p = \frac{1}{Q-1}\sum_{k=1}^{Q}(\mathbf{d}_k^p - \bar{\mathbf{d}}^p)(\mathbf{d}_k^p - \bar{\mathbf{d}}^p)^T, \qquad (7)$$

with k=1,…,Q, where Q represents the number of models in the ensemble and $\tilde{\mathbf{d}}_k$ is a random perturbation of the observed data according to the Gaussian distribution $\mathcal{N}(\mathbf{d}, \mathbf{C}_{\mathbf{d}})$. The subscripts u and p denote the updated (current iteration) and prior (previous iteration) variables, respectively; $\tilde{\mathbf{C}}_{\mathbf{md}}^p$ and $\tilde{\mathbf{C}}_{\mathbf{dd}}^p$ represent the empirical covariance matrices estimated from the ensemble members, whereas $\bar{\mathbf{m}}^p$ and $\bar{\mathbf{d}}^p$ are the empirical ensemble mean of the model parameters and predicted data, respectively.

The following steps are implemented for the ES-MDA:

1. Define the number of models in the ensemble Q, the maximum number of iterations k, and the inflation coefficient α for each iteration with $\sum_{i=1}^{Q} \frac{1}{\alpha_i} = 1$;
2. Generate realizations according to the prior $p(\mathbf{m})$;
3. For each iteration:
    a. Apply the forward operator and compute the Q data vectors data generated by the ensemble members: $\{\mathbf{d}^p\}_{1,\ldots,Q}$;
    b. Perturb the observations of each ensemble member according to: $\tilde{\mathbf{d}}_k = \mathbf{d} + \sqrt{\alpha_i}\mathbf{C}_\mathbf{d}^{-1/2}\mathbf{n}$, with $\mathbf{n} = \mathcal{N}(0, \mathbf{I})$, where $\mathbf{I}$ is the identity matrix and $\mathbf{d}$ denotes the observed data;
    c. Update the ensemble using equations 5-7 with $\mathbf{C}_\mathbf{d}$ replaced by $\alpha_i\mathbf{C}_\mathbf{d}$.

All the ensemble members at the last iteration represent possible subsurface scenarios in agreement with the acquired geophysical data and with the prior assumptions. From this ensemble of models, the PPD can be numerically evaluated. Theoretically, the method converges when the ensemble size N tends to infinity. In practical applications, a sensitivity analysis is generally required to determine the optimal number of ensemble members that guarantees accurate posterior uncertainty assessments. In particular, the number of ensemble members should be large enough to get an accurate estimate of the $\mathbf{C}_{\mathbf{dd}}^p$ and $\mathbf{C}_{\mathbf{md}}^p$ matrices but small enough not to make the forward evaluations computationally impractical. Usually, the number of ensemble members needed to get accurate uncertainty assessments increases with the dimension of the model space (Aleardi et al. 2021b).

The selected ResNet architecture

In the present work, a neural network is viewed as functions F that get input $\mathbf{I}$ and through the internal parameters $\mathbf{P}$ comuptes the output response $\mathbf{O}$:

$$\mathbf{O} = F(\mathbf{P}, \mathbf{I}). \qquad (8)$$

Both convolutional and residual networks use convolutional filters (also known kernels) and fully connected layers to extract features (forming the so-called features maps) from mono- or multi-dimensional inputs. The feature mapping from one arbitrary layer to the next can be written as (Sun et al. 2020):

$$O_j^h = f\left(B_j + \sum_{i=1}^{L} O_i^{h-1} * W_j\right), \quad j = 1, 2, \dots, J \qquad (9)$$

where L denotes the number of feature maps in the (h-1)-th layer; J is the number of feature maps in the h-th layer; B_j is a matrix with the same size as O_j^h expressing the biases for the h-th layer; O_j^h represents the j-th feature map in the h-th layer, O_i^{h-1} is the i-th feature map in the (h-1)-th layer, and W_j represents the j-th filter of the h-th layer that is the weight matrix connecting O_j^h with O_i^{h-1}; $f()$ is the activation function that includes non-linearity in the mapping process. Finally, $*$ represents the convolution process. Therefore, in a traditional convolutional neural network, each layer feeds into the next one (Figure 1, left panel), and in each training iteration the weights are updated proportionally to the partial derivative of the loss function with respect to the current weights. With an increasing number of hidden layers, it might happen that the gradient will become vanishingly small, thus preventing the update of the weights. Therefore the learning capabilities of the network degrade rapidly, leading to higher prediction error. An effective solution to this issue is provided by ResNet that makes use of shortcuts and skip connections to add the result of a shallow layer directly to the corresponding output of a deeper layer so that the information is passed through the network as an identity function (Figure 1, right panel). The idea behind this approach is to assume that the residual mapping (R(**x**) in Figure 1, right panel) is easier to optimize than the mapping f(**x**) of traditional CNNs (Figure 1, left panel). This strategy helps to prevent the loss of information that occurs when backpropagating the gradient, thus ensuring that deeper networks do not perform any worse than shallower counterparts. Many different types of residual blocks exist but here we use the original configuration as depicted in Figure 1.

The resistivity model is the input of the network, whereas the flattened apparent resistivity section constitutes the output response. We employ the same ResNet architecture represented in Figure 2 in both synthetic and field applications. Note that we use skip connections to adjust features dimensions before addition layers, while the zero-padding preserves the dimensions after convolution. We use the Leaky ReLU activation function (Hahnloser et al. 2000) with a leakage value of 0.1. After the last convolutional layer, the feature maps are flattened and passed to the first fully connected layer. Dropout is used to prevent overfitting, in which a given percentage of randomly selected neurons is ignored during the training phase (i.e., in our case the 10%). Batch normalization is used as an additional regularization operator (Santurkar et al. 2018), while we set the batch size to 32. The He method (He et al., 2015) is used to initialize the internal network parameters **P**. The root-mean-square error (RMSE) between the desired and the computed output is considered as the loss function, while the updating of the learnable weights is driven by the Adam optimizer (Kingma and Ba, 2014) running for 20 epochs. The initial learning rate is set to 0.001 and this value is scaled by 0.95 every epoch. Some tests have been performed to define the optimal hyperparameter configurations (i.e., type of activation function, number of convolutional layers, size of the filters, strategy to initialize the weights, learning rate value). The final choice has been determined through a trial-and-error procedure and according to the accuracy evaluated on the validation set. In our several experiments (not all shown here for brevity) we found that many different ResNet architectures (i.e., with different numbers of layers, filter dimensions) worked similarly. The final one has been selected as a reasonable compromise between the computational cost of the training phase and the accuracy of the predictions. This means that the applicability of the approach does not critically depend on the specific network configuration employed. Some tests with different hyperparameter configurations are presented in Appendix A.

SYNTHETIC INVERSIONS

We consider a schematic subsurface resistivity model represented by a rectangular block with a resistivity of 50 Ωm hosted in a homogeneous half-space with resistivity equal to 150 Ωm (Figure 3). The study area is discretized with $11 \times 35 = 385$ rectangular cells with vertical and lateral dimensions of 0.5 m and 1 m, respectively. The resistivity values within the cells correspond to the model parameters to be estimated. We simulate a Wenner acquisition layout with 36 electrodes with a=1 m. The maximum a value is 11. This configuration results in 198 data points. In this example, we employ the Wenner layout because it has been also used for the field data acquisition, but the presented inversion framework can be applied to other electrode configurations as well. The FE code was used to compute the noise-free observed dataset that we contaminated with uncorrelated Gaussian noise with a standard deviation equal to 20% of the total standard deviation of the noise-free data.

Figure 4 represents the prior model assumptions used to generate the training and validation sets and also used in the following probabilistic inversion. We employ a stationary log-Gaussian prior with mean and variance values directly derived from the true model of Figure 3, while a Gaussian variogram is used as the spatial continuity pattern with horizontal and vertical variogram ranges equal to 4 and 1.5 m, respectively. Note that this prior constitutes a simplification of the actual distribution of the resistivity values in the synthetic model. Indeed, the true resistivity values are not Gaussian distributed as clearly shown by the true model of Figure 3.

To keep the number of FE forward modeling evaluations needed for the probabilistic inversion to a minimum, we are particularly interested in assessing the network accuracies and generalization capabilities as the size of the training set decreases. Figure 5 represents the RMSE error computed for the synthetic experiment on the validation set with the selected network architecture (Figure 2) when the size of the training is reduced. We observe that the accuracy rapidly increases passing from 200 to 2000 training examples, then it stabilizes if additional examples are provided. For this reason, in all the following examples the ResNet learns the forward mapping using 2000 examples for training, whereas 500 is the size of the validation set. Lower RMSE values can be obtained with a larger training set (e.g., with 3500 examples) but we deem that this improvement is not worth the extra computational cost required. According to our experience, we might say that a final RMSE lower than 30-35 is acceptable for the synthetic inversion. Indeed for an RMSE value higher than this threshold, the modeling error becomes comparable to the error related to noise contamination in the data (see for example Appendix A). This substantially deteriorates the accuracy and the precision of the results. Obviously, the acceptable RMSE value is case-dependent as it is related to the statistical properties of data and noise.

Figure 6 shows a direct comparison between the RMSE values computed on the validation set for the ResNet and CNN as the total number of convolutional layers varies. The first nine ResNet layers maintain the hyperparameter setting previously shown in Figure 2, while the convolutional layers 10-13 use the same configuration of layer 9 (20 filters with dimensions 3×3). The architectures of CNN and ResNet are the same, as they only differ in the fact that skip connections and addition layers are not used by CNN. It emerges that for seven convolutional layers the CNN gets saturated and then the accuracy decreases if other layers are added. Conversely, the ResNet accuracy steadily increases and eventually reaches a stable value for nine convolutional layers. This problem affected both the validation (Figure 6) and training set (not shown here for brevity), and hence it can not simply be associated with an overfitting issue. We also observed that this accuracy decrease also affected other CNN architectures with different hyperparameter settings (i.e., type of activation function, size of convolutional filters). Therefore, we deem that this result is not even related to the specific network architecture employed. We finally interpret this result as a possible consequence of the vanishing gradient problem as it disappears when the ResNet architecture is employed. However, we also have to say that the CNN provided a quite accurate forward approximation with a final RMSE lower than the optimal value of 30-35. This means that also the CNN forward can effectively replace the FE code and that the use of CNNs does not hamper the applicability of the proposed approach.

Figure 7 displays the evolution of the RMSE on the training and validation sets for the finally selected network configuration (Figure 2) and using 2000, and 500 training and validation examples, respectively. We observe that the RMSE attains a stable value after 10 epochs. Similar errors on the validation and training sets prove that overfitting has been prevented. Figure 8 compares some examples of apparent resistivity pseudosections predicted by the trained ResNet and the associated FE datasets taken from the validation set. The close similarity between the actual and desired output (computed through the FE code) confirms that the network can effectively approximate the non-linear relation linking the model to the data. All the tests discussed in this work have been run on a common notebook equipped with an Intel i7-10750H CPU@2.60GHz with 16Gb of RAM, and with NVIDIA GeForce RTX 2060. Considering a parallel code, the generation of the 2000 training examples takes 15 minutes, approximately, while the training running on the GPU is completed in less than 5 minutes.

Before including the trained network in the inversion framework we quantify the reduction in the computing time for the forward evaluation guaranteed by the ResNet approximation. To this end, we run 100 forwards using both the FE code and the trained network (Figure 9) and it turns out that the ResNet guarantees a time reduction of three orders of magnitudes. This will translate into a dramatic difference in the computational cost of the probabilistic inversions.

Figure 10 compares the diagonal entries of the $\mathbf{C_n}$ and $\mathbf{C_p}$ matrices in the synthetic data application. We observe that the data variations related to noise contamination are of one order of magnitude higher than the variations associated with the modeling error. This also means that the posterior uncertainties will be mainly related to noise contamination and different parameter illumination rather than the approximation error introduced by the ResNet.

Figure 11 represents the histogram of the prediction error computed on the validation set and the corresponding normal probability plot. The histogram proves that, as desired, the modeling error distribution has a null mean value, thus demonstrating that the ResNet forward operator does not add any systematic bias in the predicted apparent resistivity values. The normal probability plot illustrates that, despite some minor deviations, the modeling error distribution can be reasonably assumed to be Gaussian.

We now discuss the MCMC inversion results obtained in this synthetic test. As previously mentioned we applied the same inversion code described in Vinciguerra et al. (2021) in which the DEMC algorithm is used to sample the PPD in a DCT compressed parameter space. As in that paper, only 15 DCT coefficients are considered in the inversion, thus meaning that the 385D full model domain has been sparsely represented by 15 unknown parameters. The prior assumptions in the DCT space have been analytically derived from the prior information defined in the original, uncompressed space. In what follows we also compare the results we obtain when the ResNet and FE forward modeling codes are used. In both cases, the DEMC sampling starts from prior realizations and makes use of 30 interactive chains evolving for 3000 iterations with a burn-in period of 500. Figure 12 represents the most likely solutions and associated posterior uncertainties when the two forward operators are employed. In both inversions, we obtain congruent and extremely similar results, with no significant differences. The low rectangular resistivity body is successfully located and, as expected, the posterior uncertainties are lower in the central and well-illuminated part of the model and increase towards the bottom and lateral edges of the investigated area due to lower parameter illumination. For a more complete overview of the results, Figures 13 and 14 compare the marginal prior, posterior, and true model parameter values in the DCT domain. The true coefficients in the compressed space have been analytically derived by projecting the resistivity model of Figure 3 onto the DCT domain. All the considered fifteen coefficients are displayed. Again we observe that very similar posterior evaluations are estimated with the two forward modelings. This demonstrates the reliability and stability of the presented approach.

Figure 15 shows the evolution of the negative log-likelihood for the 30 chains and for the two DEMC inversions. In both examples, the inversion attains the same stable misfit values within the same number of iterations (i.e., 500). Figures 16a-c compare the observed data with the apparent resistivity pseudosection generated on the most likely solution of Figure 12a when the ResNet and FE codes are employed. On the one hand, the similarity between Figure 16a and 16b illustrates that the predictions provided by the DEMC inversion successfully reproduce the observed data. Moreover, the good agreement between Figures 16b and 16c is a further demonstration of the capability of the trained ResNet to predict the forward mapping. Figure 16d shows instead the predicted data associated with the inversion tests in which the FE code has been used. Also in this case the predicted model of Figure 12b successfully reproduces the observed data.

As a final and more quantitative assessment of the results, we list in Table 1 the 90% coverage ratio, the root-mean-square errors (RMSE) between true and predicted models, observed and predicted data, together with the computing times for the two probabilistic inversions. These computing times refer to the hardware resources previously described and to parallel codes in which the forward evaluations are distributed across different cores. We remind that the 90% coverage ratio quantifies the percentage of resistivity values in the true model that fall within the 90% confidence interval as estimated by the probabilistic inversion. The two inversions provide very similar models, data predictions, and also uncertainty estimations. However, there is a dramatic difference in the computational demand: the inversion takes only 10 minutes when the ResNet forward operator is used, while approximately 20 hours are needed with the FE code. These results demonstrate that the proposed approach not only drastically reduces the computational workload of the probabilistic sampling but, more importantly, also provides model estimations and uncertainty assessments comparable to those achieved with the FE forward modelling.

FIELD DATA APPLICATION

We now apply the presented approach to invert a field dataset acquired for levee monitoring along the Parma River in Colorno (Italy). We refer the interested reader to Hojat et al. (2019b) for more information about the study area. Due to the considerable computing time (several days) needed to tackle this inversion with the previously considered DEMC approach (see Vinciguerra et al. 2021), we resort to applying the ES-MDA algorithm. We invert a single dataset acquired with electrodes buried in a 0.5 m-deep trench and employing the Wenner configuration using 48 electrodes with a unit electrode spacing of a = 2 m. The dataset is corrected for the effect of the soil covering the electrodes (Hojat et al. 2021). The investigated site covers an area that is 94 m wide and 14 m deep and it is discretized with rectangular cells with vertical and lateral dimensions of 1 m and 2 m, respectively. This configuration results in 15×47= 705 resistivity values to be estimated from 360 data points.

We exploit all the available information about the investigated site to define the prior distribution of model parameters (see Hojat et al. 2019b). In particular, we still simplify the actual distribution of the subsurface resistivity with a log-Gaussian prior, and we employ a spatial variability pattern described by a Gaussian variogram with lateral and vertical ranges equal to 6 m and 2 m, respectively (Figure 17). In this area, we mainly expect a low-resistivity clay body that around 2-3 m depth hosts a more permeable layer with higher resistivity values associated with the presence of sand and gravel.

Similar to the synthetic case we draw from the prior 2000 models for training and 500 for validation. We maintain the same ResNet architecture that was previously described with the only difference related to the dimension of the input image (in this case a model with 15 rows and 47 columns) and in the output size (a vector of 360 apparent resistivity values). As demonstrated in Figure 18 the similarity of the ResNet predictions with the corresponding FE responses extracted from the validation set proves the capability of the trained machine learning model to predict the forward relation. Figure 19a shows the histogram of the prediction error derived from the validation set and just from a visual inspection we can affirm that the Gaussian assumption is reasonable also for this field data application. Figure 19b again shows that the data variability related to the noise contamination as expressed by the $\mathbf{C_n}$ matrix is larger than that expected from the modeling error. In this case, the $\mathbf{C_n}$ matrix has been derived from the variance of repeated measures of the apparent resistivity values within a short time frame, under the assumption of Gaussian distributed noise.

For the inversion, we employ the ES-MDA algorithm with an ensemble of 2000 models initially drawn from the prior assumptions and updated during five consecutive iterations. Similar to the synthetic example, we compare the results obtained when the ResNet and FE forward operators are used. No data or model space compressions are applied in this case. Figure 20 represents the final results in terms of the most likely solution and associated standard deviation. Similar and congruent outcomes are achieved in both cases: the inversion predicts a sand-gravel body at shallow depth hosted in shales. As expected the uncertainty tends to increase in correspondence of the high resistivity body and at the lateral and bottom edges of the model.

Figure 21 shows the observed data and the data generated by the ResNet and FE on the model of Figure 20a, together with the apparent resistivity pseudosection derived with the FE code from the model of Figure 20b. Both inversions provide most likely solutions that accurately reproduce the field measurements with very minor differences in the data generated by the ResNet and FE codes. This demonstrates that the trained network can properly approximate the forward relation even for realistic resistivity models not seen during the learning procedure. As a further demonstration of the generalization ability of the trained ResNet network Figure 22 compares, for the inversion test running with the ResNet forward, the observed data and the data predicted with both the ResNet and FE forwards on the initial and final (i.e., at the last iteration) ensemble of models. Again very minor differences can be seen in the apparent resistivity values provided by the two forward operators.

Figures 23 and 24 compare (for the inversion running with the ResNet forward) some models forming the initial ensemble and the corresponding predictions at the last iteration. We observe that the inversion satisfactorily converges towards congruent results. Indeed all the models at the very last iteration show similar characteristics such as the low resistivity anomaly in the shallowest and central part of the study area, and the high resistivity body buried around 3 m depth.

The computing times for the ES-MDA inversion with the ResNet and FE forward modelings are equal to 1 minute and 1.3 hours, respectively. So also in this case the trained ResNet model guarantees a significant speed-up of the inversion procedure, while still guaranteeing reliable predictions.

CONCLUSIONS

This work was aimed at reducing the computational cost of the probabilistic ERT inversion. To this end, we trained a ResNet network to learn the non-linear mapping between the model parameters and the associated data. This approach replaces the computationally complex FE forward modeling with the more computationally efficient network predictions. This reduced the average time needed for a forward evaluation of several orders of magnitude. In both synthetic and field experiments, the modeling error introduced by the network approximation was one order of magnitude lower than the expected data variability related to noise contamination. This modeling error has been quantified and properly propagated into the final estimates. The synthetic and field examples demonstrated that a neural network can be used to describe a (relatively) complex forward operator. Our examples also illustrated the applicability of the approach and that it provides most likely solutions and uncertainty assessments comparable to those achieved with the FE forward modeling.

The main challenge of this methodology might be generating a large enough training set. Indeed the forward model needs to be evaluated as many times as the size of the training set. This procedure may be time-consuming, but need to be done only once, and it is also easily parallelizable. Once trained the ResNet model is applicable only for the specific acquisition layout assumed for the training phase. If the recording geometry changes a new training set needs to be generated, and a new learning phase must be run. Moreover, note that in our examples the training data was generated according to a specific prior distribution. Thus, if the prior assumptions change, a new training set needs to be created and a new network needs to be trained. An alternative approach might be generating the training set according to a much broader prior model, namely, a prior model that generates realizations with larger spatial variability. Such a training set could be used to learn a forward mapping to be used for a wider range of possible prior assumptions without the need for multiple retraining. Another possibility is to apply transfer learning techniques to update the internal parameters of a previously trained network when the distribution of the new prior differs from that assumed during the training procedure. These may be subjects for future researches. In any case, it is worth noting that our results demonstrated that the network can be successfully trained also with a relatively small training set. Moreover, both the MCMC and ES-MDA algorithms will need many more forward evaluations than the size of the training set we considered. The main benefit of the presented approach is that it provides an alternative forward that is much faster than the FE code. This allows using sampling-based approaches to solve the probabilistic ERT inversion with a reasonable computational cost and using limited hardware resources. The approach can be also applied to reduce the computational complexity of other non-linear inverse problems that have to be solved through a stochastic sampling over the parameter space.

Conflict of interest

The authors declare no conflict of interest

Data availability

Data available on request from the corresponding author.

REFERENCES

Aleardi, M., Vinciguerra, A., and Hojat, A. (2020). A geostatistical Markov Chain Monte Carlo inversion algorithm for electrical resistivity tomography. Near Surface Geophysics, 19(1), 7-26.

Aleardi, M. (2020). Combining discrete cosine transform and convolutional neural networks to speed up the Hamiltonian Monte Carlo inversion of pre-stack seismic data. Geophysical Prospecting, 68(9), 2738-2761.

Aleardi, M., Vinciguerra, A., and Hojat, A. (2021). A convolutional neural network approach to electrical resistivity tomography. Journal of Applied Geophysics, 193, 104434.

Aleardi, M., and Salusti, A. (2020). Hamiltonian Monte Carlo algorithms for target-and interval-oriented amplitude versus angle inversions. Geophysics, 85(3), R177-R194.

Aleardi, M., Vinciguerra, A., and Hojat, A. (2021). Ensemble-based electrical resistivity tomography with data and model space compression. Pure and Applied Geophysics, 1-23.

Aleardi, M., and Salusti, A. (2021). Elastic prestack seismic inversion through discrete cosine transform reparameterization and convolutional neural networks. Geophysics, 86(1), R129-R146.

Aleardi, M. (2021). A gradient-based Markov chain Monte Carlo algorithm for elastic pre-stack inversion with data and model space reduction. Geophysical Prospecting, 69(5), 926-948.

Araya-Polo, M., Jennings, J., Adler, A., and Dahlke, T. (2018). Deep-learning tomography. The Leading Edge, 37(1), 58-66.

Aster, R. C., Borchers, B., and Thurber, C. H. (2018). Parameter estimation and inverse problems. Elsevier.

Bièvre, G., Oxarango, L., Günther, T., Goutaland, D. and Massardi M. (2018). Improvement of 2D ERT measurements conducted along a small earth-filled dyke using 3D topographic data and 3D computation of geometric factors. Journal of Applied Geophysics 153, 100-112.

Biswas, R., and Sen, M. (2017). 2D full-waveform inversion and uncertainty estimation using the reversible jump Hamiltonian Monte Carlo. In SEG Technical Program Expanded Abstracts 2017 (pp. 1280-1285). Society of Exploration Geophysicists.

Dahlin, T. (2020). Geoelectrical monitoring of embankment dams for detection of anomalous seepage and internal erosion - experiences and work in progress in Sweden. Fifth International Conference on Engineering Geophysics (ICEG), Al Ain, UAE, DOI: 10.1190/iceg2019-053.1.

Dosso, S. E., Holland, C. W., and Sambridge, M. (2012). Parallel tempering for strongly nonlinear geoacoustic inversion. The Journal of the Acoustical Society of America, 132(5), 3030-3040.

Emerick, A. A., and Reynolds, A. C. (2013). Ensemble smoother with multiple data assimilation. Computers & Geosciences, 55, 3-15.

Fichtner, A., Zunino, A., and Gebraad, L. (2019). Hamiltonian Monte Carlo solution of tomographic inverse problems. Geophysical Journal International, 216(2), 1344-1363.

Gebraad, L., Boehm, C., and Fichtner, A. (2020). Bayesian elastic full-waveform inversion using Hamiltonian Monte Carlo. Journal of Geophysical Research: Solid Earth, 125(3), e2019JB018428.

Goodfellow, I., Bengio, Y., and Courville, A. (2016), Deep learning: Cambridge, MIT Press.

Grana, D., de Figueiredo, L., and Mosegaard, K. (2021). Markov chain Monte Carlo for petrophysical inversion. Geophysics, 87(1), 1-60.

Grana, D., Passos de Figueiredo, L., and Azevedo, L. (2019). Uncertainty quantification in Bayesian inverse problems with model and data dimension reduction. Geophysics, 84(6), M15-M24.

Hansen, T. M., and Cordua, K. S. (2017). Efficient Monte Carlo sampling of inverse problems using a neural network-based forward—Applied to GPR crosshole traveltime inversion. Geophysical Journal International, 211(3), 1524-1533.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, 1026-1034.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 770-778.

Hermans, T., and Paepen, M. (2020). Combined inversion of land and marine Electrical resistivity tomography for submarine groundwater discharge and saltwater intrusion characterization. Geophysical Research Letters, 47(3), e2019GL085877.

Hojat, A., Arosio, D., Di Luch, I., Ferrario, M., Ivanov, V.I., Longoni, L., Madaschi, A., Papini, M., Tresoldi, G. and Zanzi, L. (2019a). Testing ERT and fiber optic techniques at the laboratory scale to monitor river levees. 25th European Meeting of Environmental and Engineering Geophysics, The Hague, Netherlands, DOI: 10.3997/2214-4609.201902440.

Hojat, A., Arosio, D., Longoni, L., Papini, M., Tresoldi, G., and Zanzi, L. (2019b). Installation and validation of a customized resistivity system for permanent monitoring of a river embankment. EAGE-GSM 2nd Asia Pacific Meeting on Near Surface Geoscience and Engineering, Kuala Lumpur, Malaysia. Doi: 10.3997/2214-4609.201900421.

Karaoulis, M., Revil, A., Tsourlos, P., Werkema, D. D., and Minsley, B. J. (2013). IP4DI: A software for time-lapse 2D/3D DC-resistivity and induced polarization tomography. Computers & Geosciences, 54, 164-170.

Hojat, A., Tresoldi, G. and Zanzi, L. (2021). Correcting the effect of the soil covering buried electrodes for permanent electrical resistivity tomography monitoring systems. Accepted: 4th Asia Pacific Meeting on Near Surface Geoscience & Engineering, Ho Chi Minh, Vietnam, Virtual.

Kingma, D. P., and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

Liu, B., Guo, Q., Li, S., Liu, B., Ren, Y., Pang, Y., Guo, X., Liu, L., and Jiang, P. (2020). Deep learning inversion of electrical resistivity data. IEEE Transactions on Geoscience and Remote Sensing, 58(8), 5715-5728.

Loke, M. H., Papadopoulos, N., Wilkinson, P. B., Oikonomou, D., Simyrdanis, K., and Rucker, D. F. (2020). The inversion of data from very large three-dimensional electrical resistivity tomography mobile surveys. Geophysical Prospecting, 68(8), 2579-2597.

Luo, X., Bhakta, T., Jakobsen, M., and Nævdal, G. (2018). Efficient big data assimilation through sparse representation: A 3D benchmark case study in petroleum engineering. PloS one, 13(7), e0198586.

Menke, W. (2018). *Geophysical data analysis: Discrete inverse theory*. Academic press.

Moghadas, D. (2020). One-dimensional deep learning inversion of electromagnetic induction data using convolutional neural network. Geophysical Journal International, 222(1), 247-259.

Monajemi, H., Donoho, D. L., and Stodden, V. (2016). Making massive computational experiments painless. In 2016 IEEE International Conference on Big Data, 2368-2373.

Moradipour, M., Ranjbar, H., Hojat, A., Karimi-Nasab, S. and Daneshpajouh, S. (2016). Laboratory and field measurements of electrical resistivity to study heap leaching pad no. 3 at Sarcheshmeh copper mine. 22nd European Meeting of Environmental and Engineering Geophysics, DOI: 10.3997/2214-4609.201602140.

Norooz, R., Olsson, P.I., Dahlin, T., Günther, T., Bernston, C. (2021). A geoelectrical pre-study of Älvkarleby test embankment dam: 3D forward modelling and effects of structural constraints on the 3D inversion model of zoned embankment dams. Journal of Applied Geophysics, 191, 104355.

Park, M. J., and Sacchi, M. D. (2020). Automatic velocity analysis using Convolutional Neural Network and Transfer learning. Geophysics, 85(1), V33-V43.

Puzyrev, V. (2019). Deep learning electromagnetic inversion with convolutional neural networks. Geophysical Journal International, 218(2), 817-832.

Ray, A., Kaplan, S., Washbourne, J., and Albertin, U. (2017). Low frequency full waveform seismic inversion within a tree based Bayesian framework. Geophysical Journal International, 212(1), 522-542.

Richardson, A. (2018). Seismic full-waveform inversion using deep learning tools and techniques. arXiv preprint:1801.07232.

Rucker, D.F., Fink, J.B., Loke, M.H. (2011) Environmental monitoring of leaks using time-lapsed long electrode electrical resistivity. Journal of Applied Geophysics, 74(4), 242-254.

Sambridge, M., and Mosegaard, K. (2002). Monte Carlo methods in geophysical inverse problems. Reviews of Geophysics, 40(3), 3-1.

Sætrom, J., and Omre, H. (2013). Uncertainty quantification in the ensemble Kalman filter. Scandinavian Journal of Statistics, 40(4), 868-885.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018). How does batch normalization help optimization?. Advances in neural information processing systems, 31, 2483-2493.

Sen, M. K., and Stoffa, P. L. (2013). Global optimization methods in geophysical inversion. Cambridge University Press.

Sun, J., Slang, S., Elboth, T., Larsen Greiner, T., McDonald, S., and Gelius, L. J. (2020). A convolutional neural network approach to deblending seismic data. Geophysics, 85(4), WA13-WA26.

Tarantola, A. (2005). Inverse problem theory and methods for model parameter estimation. Society for Industrial and Applied Mathematics.

Uhlemann, S., Wilkinson, P.B., Chambers, J.E., Maurer, H., Merritt, A.J., Gunn, D.A. and Meldrum P.I. (2015). Interpolation of landslide movements to improve the accuracy of 4D geoelectrical monitoring. Journal of Applied Geophysics, 121, 93-105.

Vinciguerra, A., Aleardi, M., Hojat, A., and Stucchi, E. (2021). Discrete Cosine Transform for Parameter Space Reduction in Bayesian Electrical Resistivity Tomography. Geophysical Prospecting. Doi: 10.1111/1365-2478.13148.

Vrugt, J. A. (2016). Markov chain Monte Carlo simulation using the DREAM software package: Theory, concepts, and MATLAB implementation. Environmental Modelling & Software, 75, 273-316.

Vu, M. T., and Jardani, A. (2021). Convolutional neural networks with SegNet architecture applied to three-dimensional tomography of subsurface electrical resistivity: CNN-3D-ERT. Geophysical Journal International, 225(2), 1319-1331.

Waldeland, A. U., Jensen, A. C., Gelius, L. J., and Solberg, A. H. S. (2018). Convolutional neural networks for automated seismic interpretation. The Leading Edge, 37(7), 529-537.

Wang, B., Zhang, N., Lu, W., and Wang, J. (2019). Deep-learning-based seismic data interpolation: A preliminary result. Geophysics, 84(1), V11-V20.

Whiteley, J., Chambers, J.E., and Uhlemann, S., (2017). Integrated monitoring of an active landslide in lias group mudrocks, north yorkshire, UK. In: Hoyer, S. (Ed.), GELMON 2017: 4th International Workshop on GeoElectrical Monitoring, 27.

Wu, Y., and McMechan, G. A. (2019). Parametric convolutional neural network-domain full-waveform inversion. Geophysics, 84(6), R881-R896.

Zhao, Z., and Sen, M. K. (2021). A gradient-based Markov chain Monte Carlo method for full-waveform inversion and uncertainty analysis. Geophysics, 86(1), R15-R30.

APPENDIX A

We here represent the final RMSE values computed on the training and validation sets for four different network configurations. These are variants of the network represented in Figure 2 in which only one parameter has been modified. All these networks consider the same training and validation set previously used to train the finally selected network.

- Network 1: The same as Figure 2 but with a filter size of 5×5 in all the layers;
- Network 2: The same as Figure 2 but all the convolutional layers use a total number of filters increased by 5 (e.g., the CONV 2 and CONV 3 layers use 10 filters; see Figure 2);
- Network 3: The same as Figure 2 but without the first fully connected layer;
- Network 4: The same as Figure 2 but with the ReLU activation function instead of the LeakyRelu.

Figure 25 represents the final RMSE values computed on the validation and training sets for the four previously considered networks. All the configurations provide quite satisfactory predictions especially the Networks 2-4 with prediction errors lower than the optimal value of 30-35. This figure also shows that the selected network architecture (Figure 2) yields slightly better results than the four considered alternatives. Finally, Figure 26 illustrates a comparison of the apparent resistivity pseudosections predicted by Network 1 and the desired response (FE output). Compare this figure with Figure 8 to appreciate the better predictions provided by the selected network.

FIGURES AND CAPTIONS

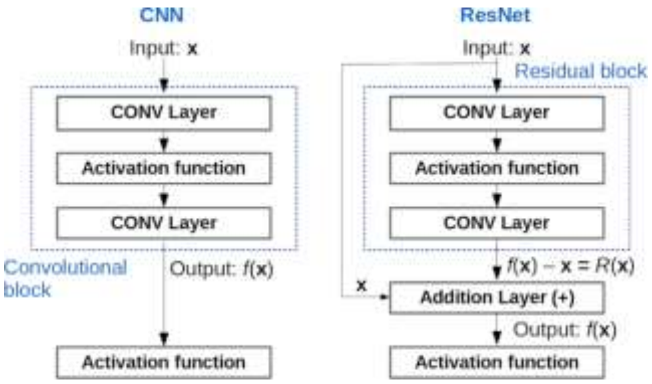

Figure 1: Comparison between a convolutional block in CNN (left) and a residual block in ResNet (right).

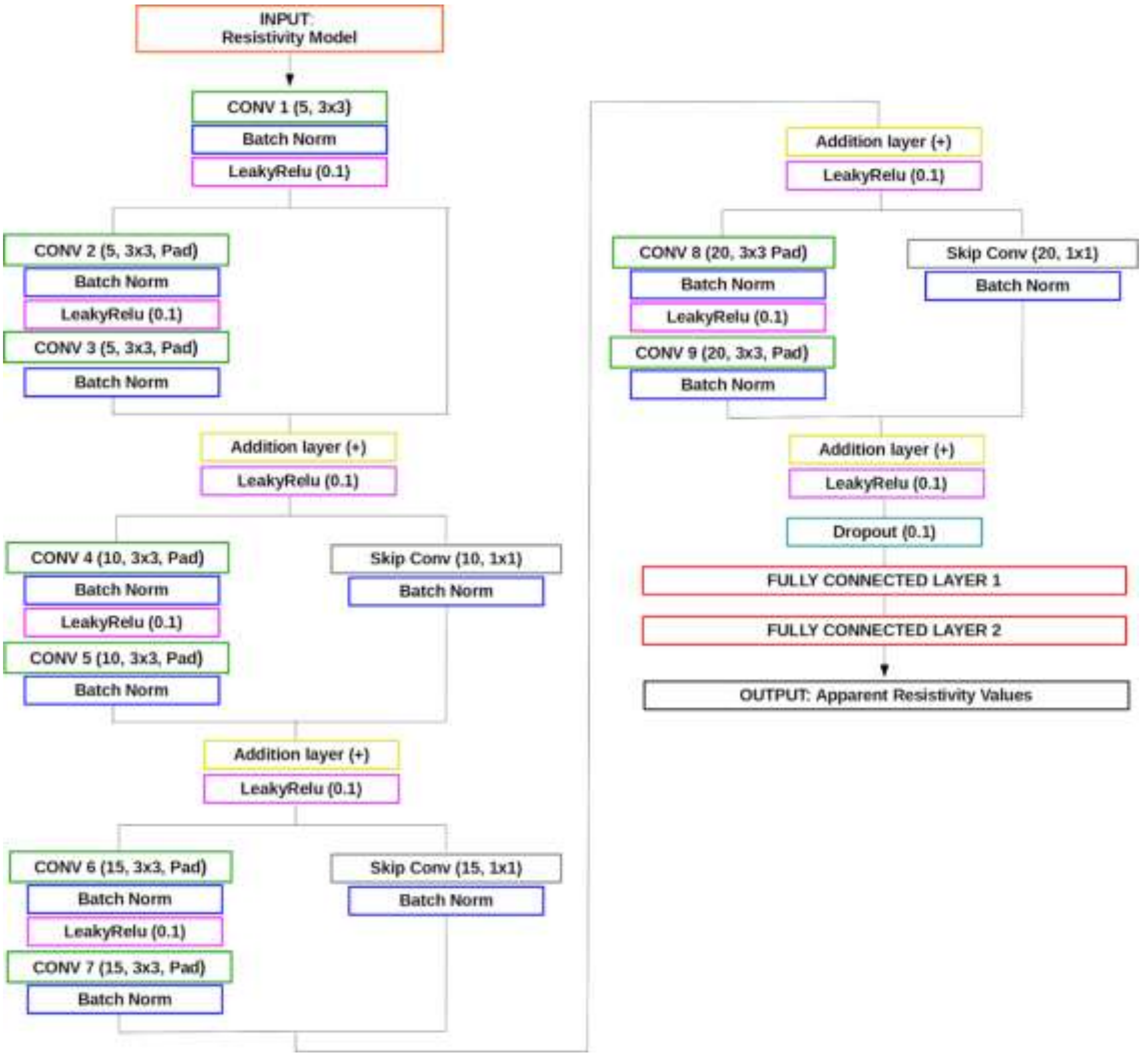

Figure 2: Representation of the employed ResNet architecture annotated with key parameters. For example, in the second convolutional layer “CONV 2” the term within bracket (5, 3x3, Pad) indicates that we employ 5 convolutional filters with size 3x3 and that zero-padding is applied. The only difference in the synthetic and field data applications concerns the dimension of input and output response. See the text for details.

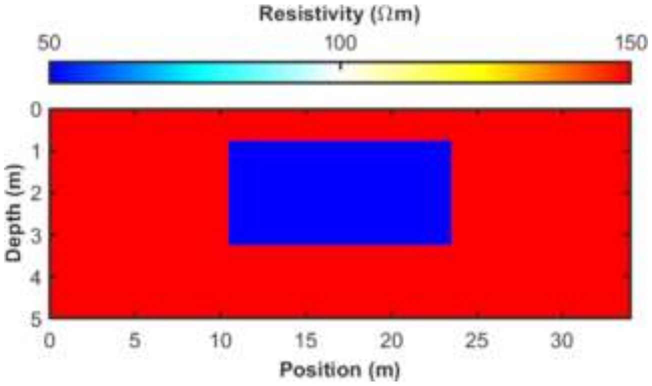

Figure 3: The true model for the synthetic inversion.

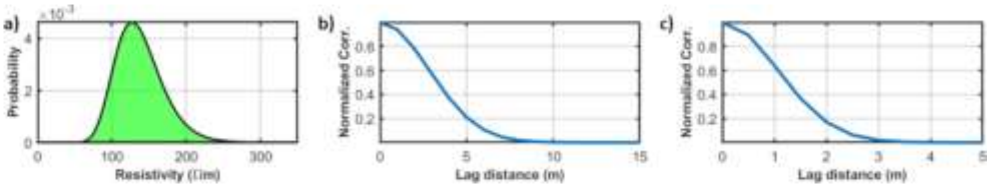

Figure 4: a) Log-Gaussian prior distribution for the synthetic example. b), and c) spatial correlation functions associated with the assumed 2-D variogram model for the horizontal and vertical directions, respectively.

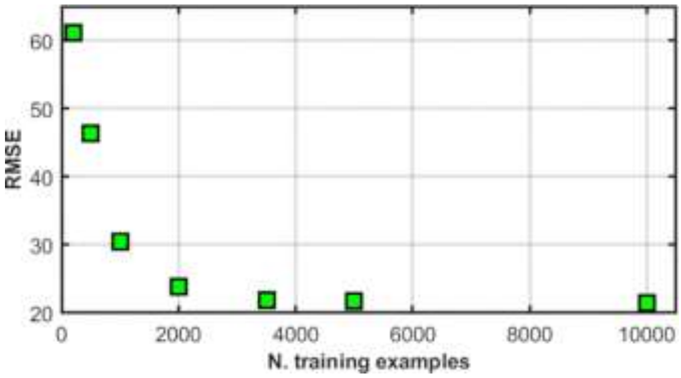

Figure 5: RMSE values computed on the validation set for different sizes of the training ensemble (200, 500, 1000, 2000, 3500, 5000, and 10000 examples)

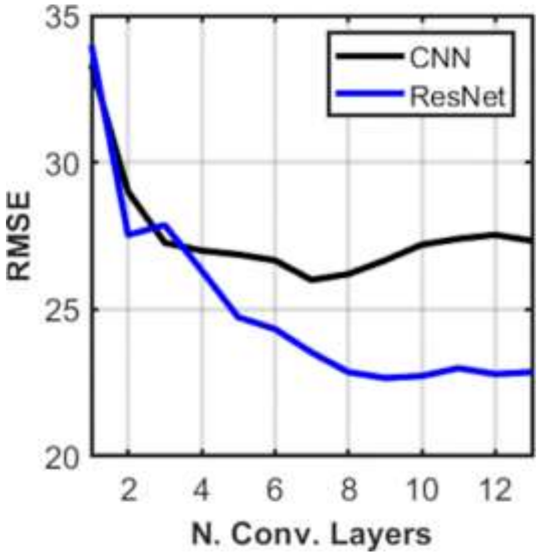

Figure 6: Final RMSE values computed on the validation set for the CNN and ResNet as the number of convolutional layers varies.

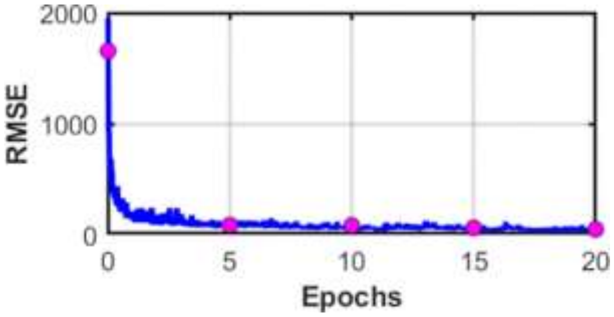

Figure 7: Evolution of the RMSE on the training (blue line) and validation sets (magenta circles).

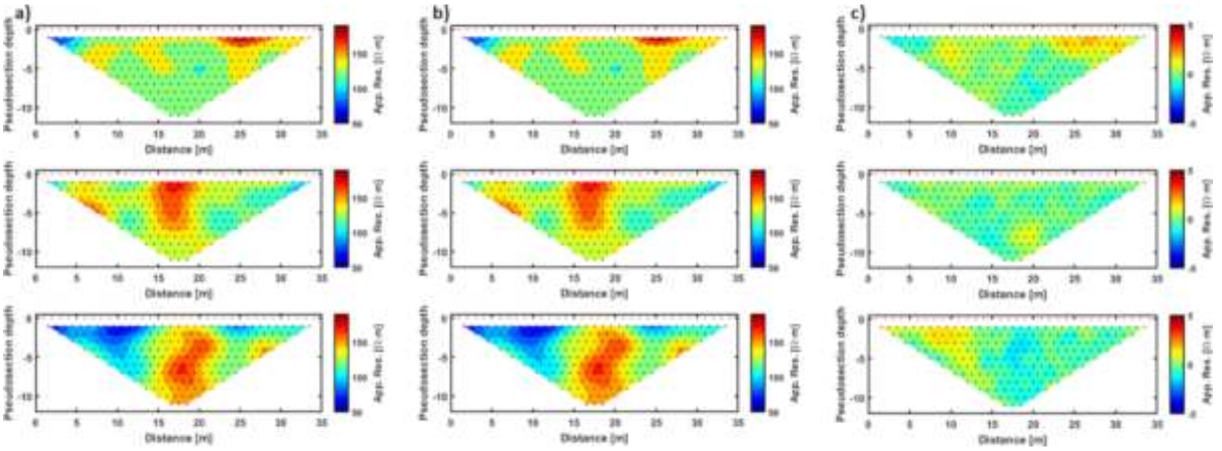

Figure 8: Some comparisons between (a) the network responses and (b) the desired (i.e., FE) output from the validation set. In c) we represent the sample by sample difference between a) and b). Note the different color scale in c) with respect to a) and b).

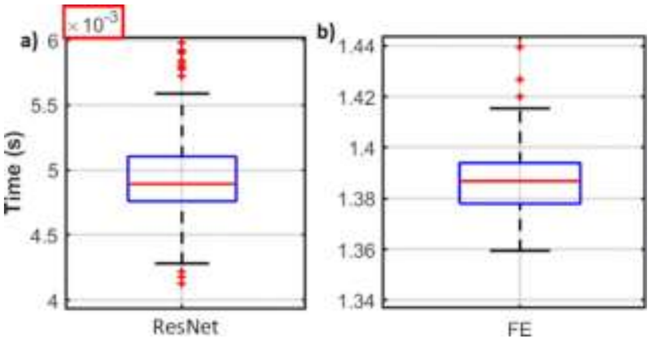

Figure 9: Box plots derived from 100 runs of the ResNet and FE forward operators (a, and b, respectively). The red crosses represent outliers. Note the different scales on the vertical axes.

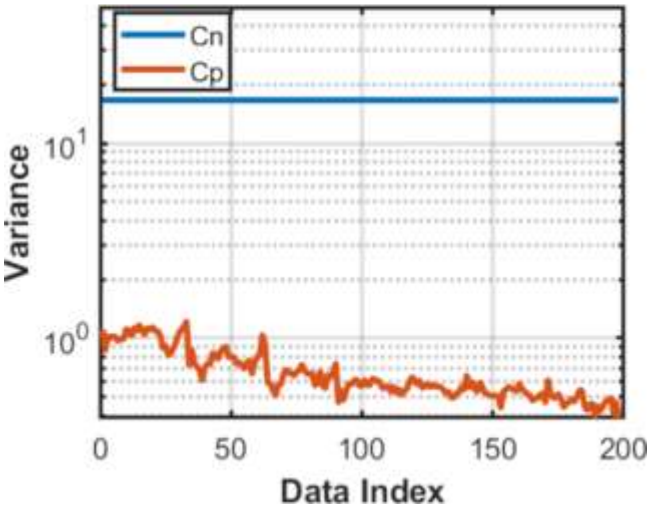

Figure 10: Comparison between the diagonal entries of the $\mathbf{C_n}$ and $\mathbf{C_p}$ matrices.

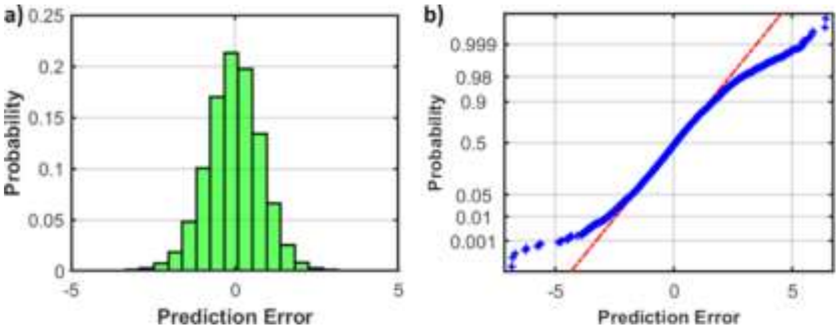

Figure 11. a) Histogram of the prediction error computed on the validation set. b) Normal probability plot derived from the prediction error shown in a). The red line depicts the theoretical trend for a Gaussian distribution.

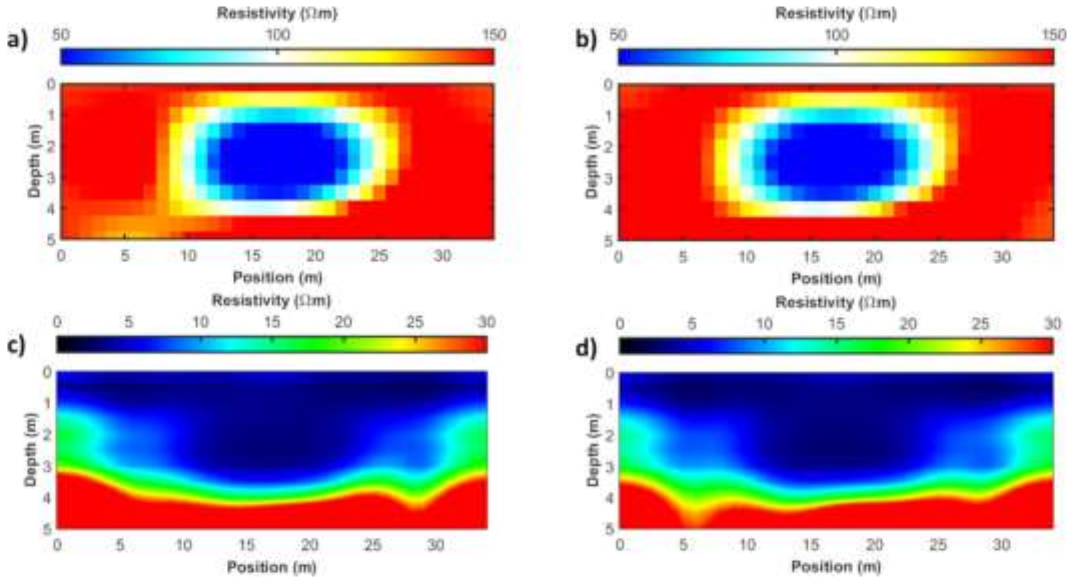

Figure 12: a) and b) the most likely models predicted by the DEMC inversions when the ResNet and FE forward operator are employed, respectively. c) and d) Posterior standard deviations associated with the solution shown in a) and b), respectively.

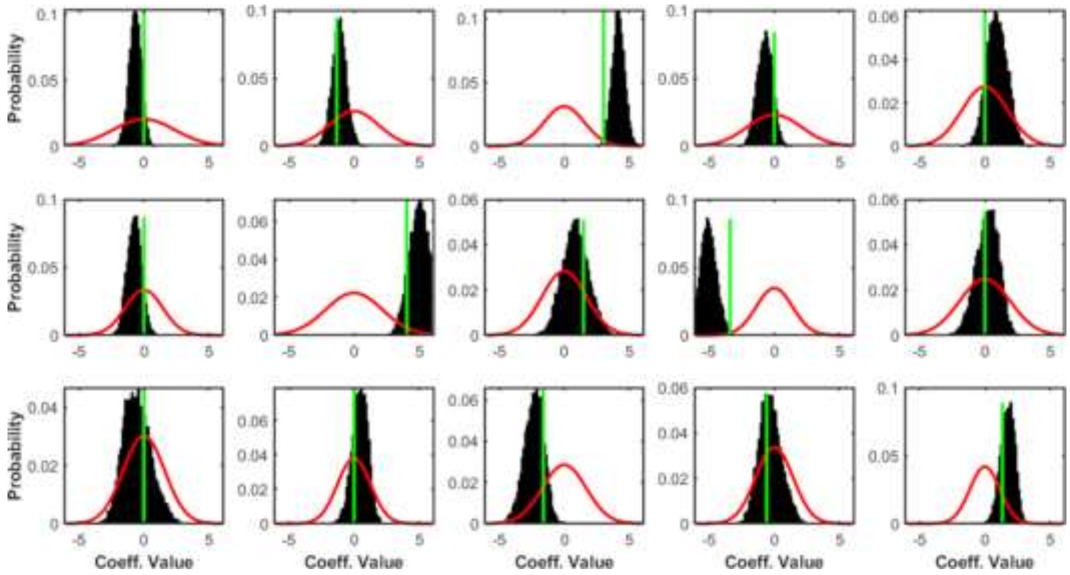

Figure 13: Inversion results in the DCT space when the ResNet forward modeling is employed. Each plot refers to one of the fifteen considered DCT coefficients. The red lines represent the marginal priors, the black bars are the marginal posteriors, whereas the green lines indicate the true DCT parameter values.

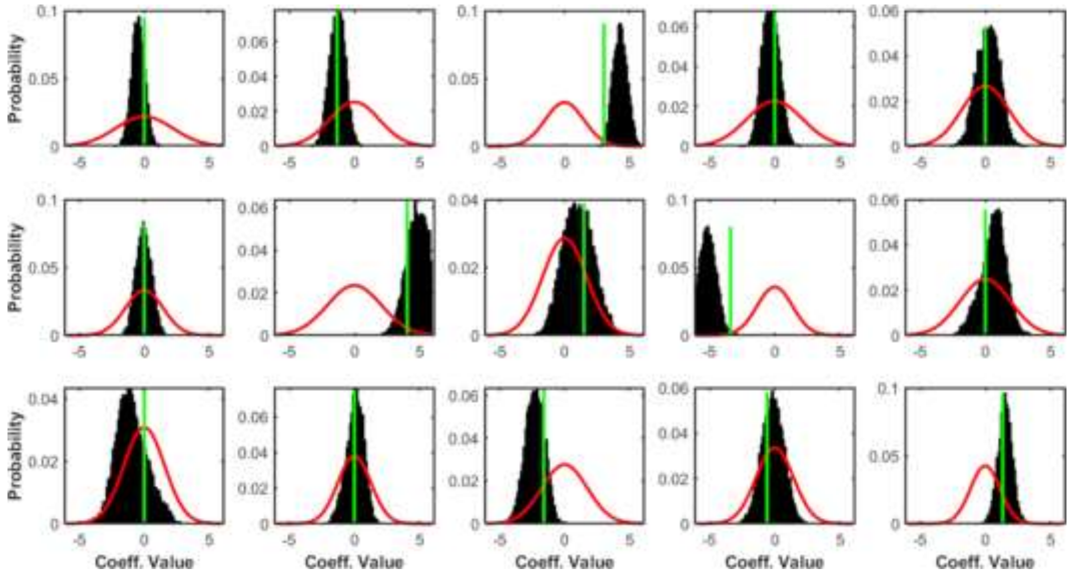

Figure 14: As in Figure 13 but when the FE forward code is used.

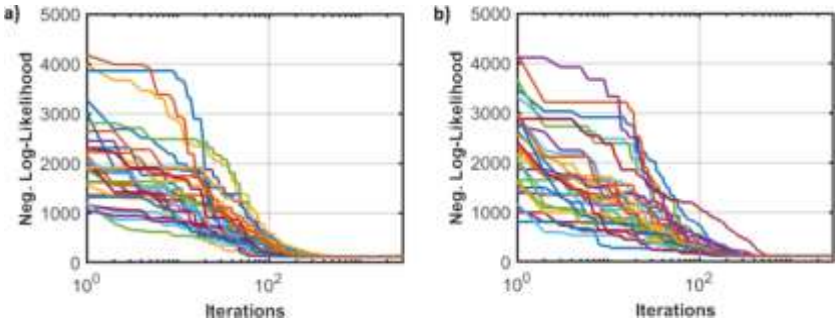

Figure 15: Evolution of the negative log-likelihood during the DEMC inversion for the 30 chains and the first 3000 iterations. a) With the ResNet forward operator. b) With the FE forward operator.

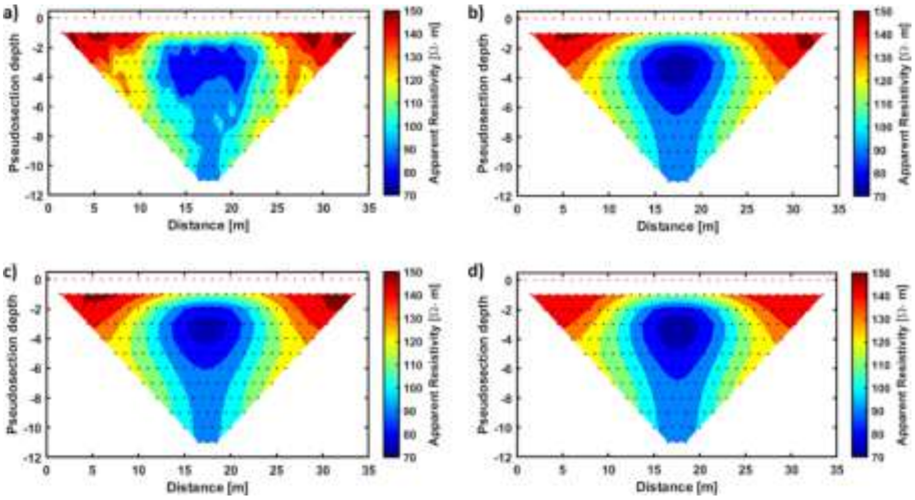

Figure 16: a) Observed data. b) Apparent resistivity pseudosection computed on the models shown in Figure 12a when the ResNet forward operator is used. c) Apparent resistivity pseudosection computed on the models shown in Figure 12a when the FE forward operator is used. d) Predicted data computed on the most likely model of Figure 12b with the FE code.

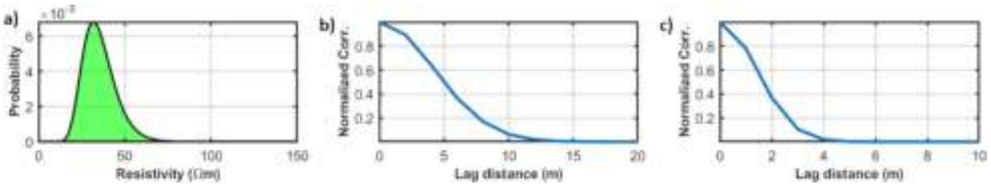

Figure 17: a) Log-Gaussian prior distribution for the field data inversion. b), and c) the spatial correlation functions associated with the assumed 2-D variogram model for the horizontal and vertical directions, respectively.

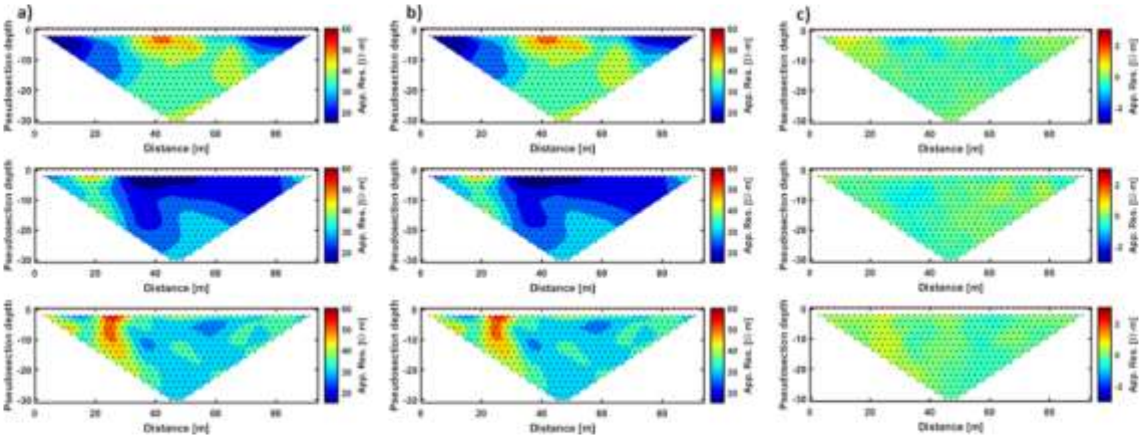

Figure 18: Some comparisons between (a) the network responses and (b) the desired output extracted from the validation set. In c) we represent the sample by sample difference between a) and b). Note the different color scale in c) with respect to a) and b).

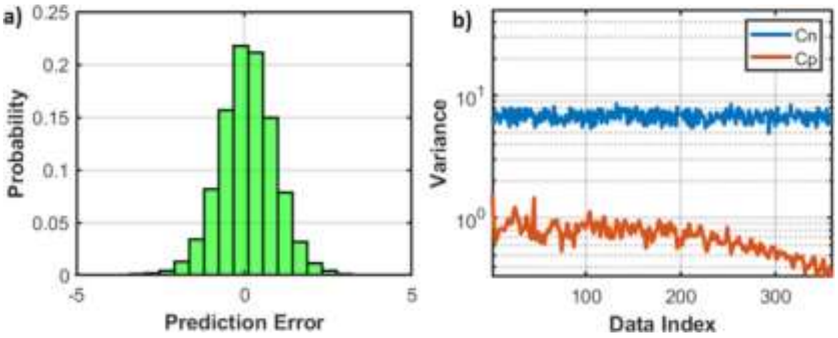

Figure 19: a) Histogram of the modeling error computed on the validation set. b) Diagonal entries of the $\mathbf{C_n}$ and $\mathbf{C_p}$ matrices in the field data application.

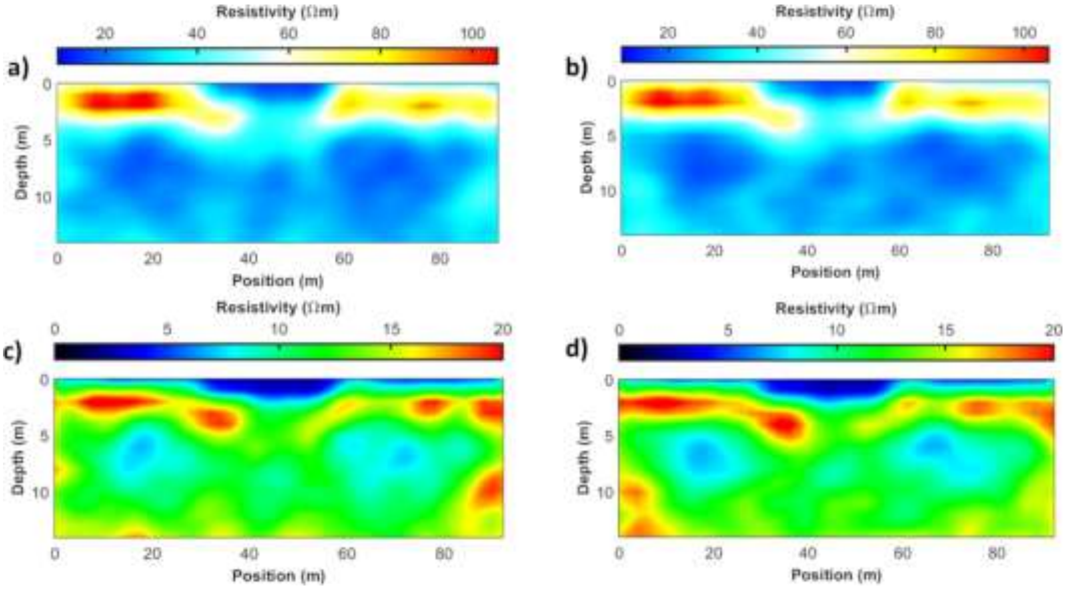

Figure 20: a) and b) the most likely solutions provided by the ES-MDA inversions when the ResNet and FE forward operator are employed, respectively. c) and d) Posterior standard deviations associated with the models shown in a) and b), respectively.

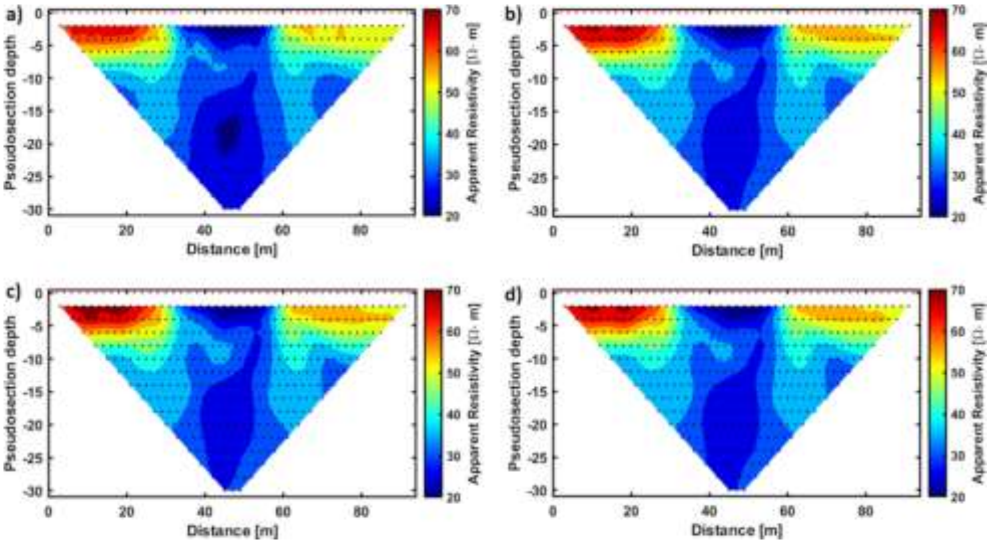

Figure 21: a) Observed data. b) Apparent resistivity pseudosection computed on the model shown in Figure 20a when the ResNet forward operator is used. c) Apparent resistivity pseudosection computed on the model shown in Figure 20a when the FE forward operator is used. d) Predicted data computed on the most likely model of Figure 20b with the FE code.

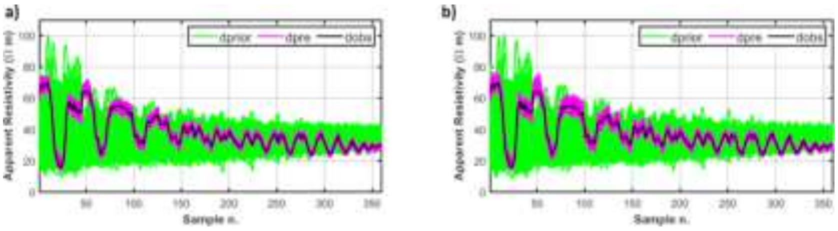

Figure 22: Comparison between the observed data (black line), the data computed on the initial ensemble of models (green lines), and the data associated with the models at the last ES-MDA inversion (magenta lines). For graphical convenience, the pseudo sections have been flattened to 1D vectors. a) and b) refer to the same models but when the ResNet and FE codes are employed, respectively.

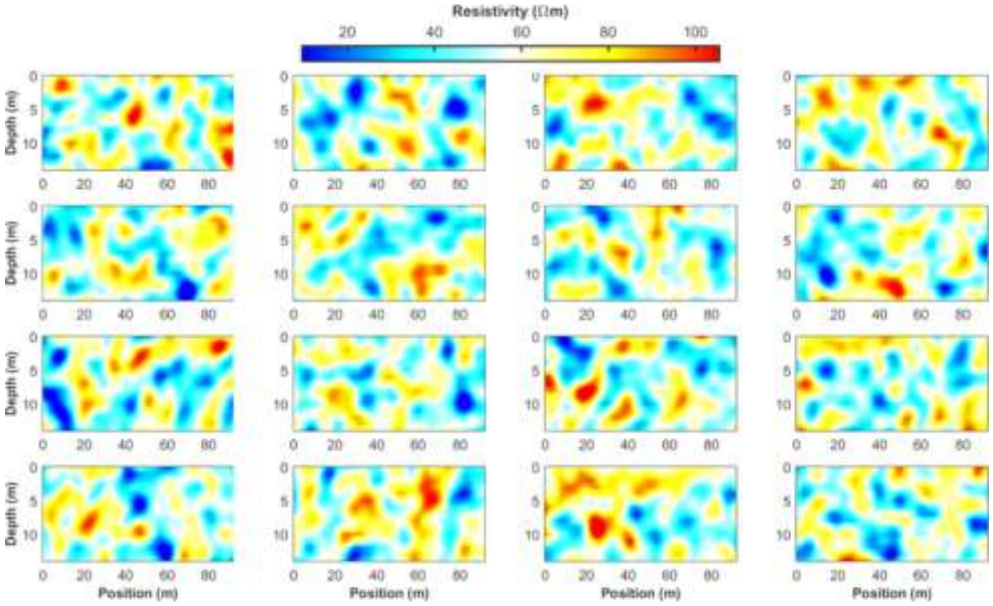

Figure 23: Examples of prior realizations forming the starting ensemble of the ES-MDA inversion with the ResNet forward operator.

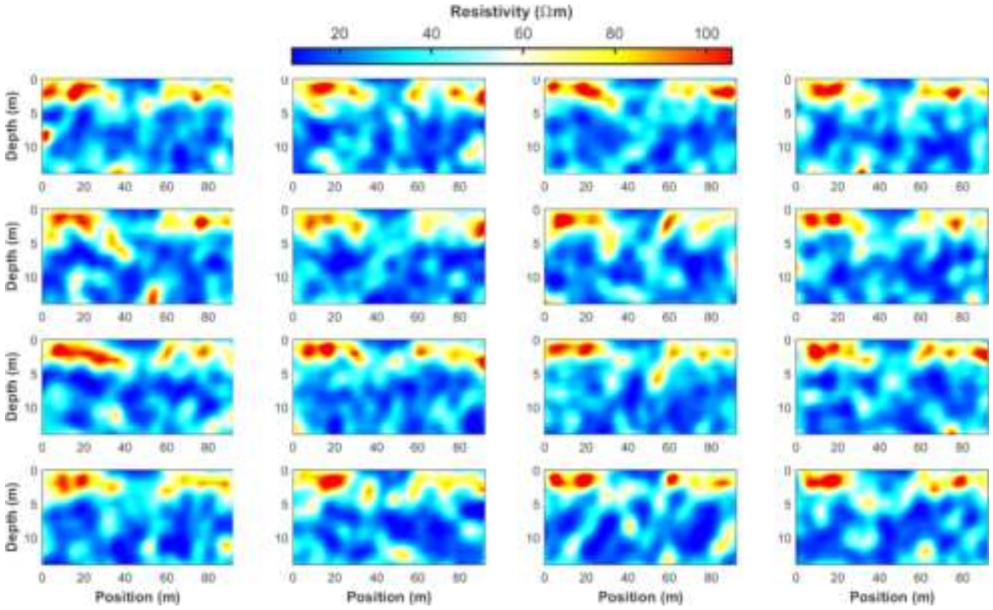

Figure 24: Models extracted from the ensemble at the last ES-MDA iteration when the the ResNet forward is used.

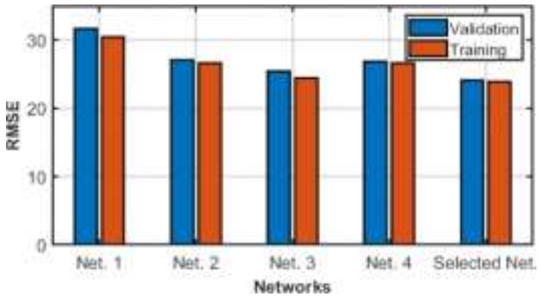

Figure 25: RMSE on validation and training sets for the four different network architectures considered and the finally selected configurations (represented in Figure 2).

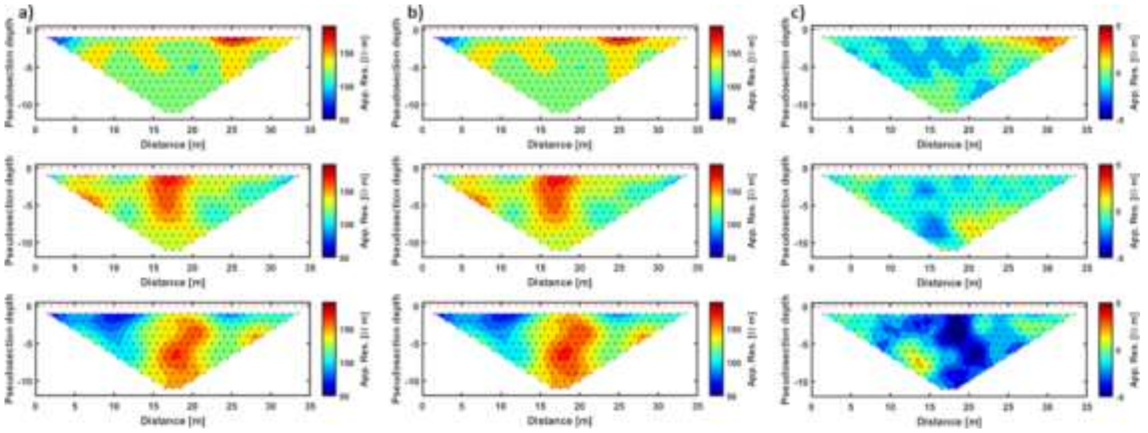

Figure 26: Some comparisons between (a) the network predictions provided by Network 1 and (b) the desired output extracted from the validation set. In c) we represent the sample by sample difference between a) and b). Note the different color scale in c) with respect to a) and b). Compare with Figure 8.

TABLE

	RMSE Model	RMSE Data	90% coverage ratio	Computing Time
DEMC with the ResNet forward operator	117.81	6.23	86.71%	≈ 10 minutes
DEMC with the FE forward operator	116.34	6.12	86.98%	≈ 20 hours

Table 1: Table listing for the two DEMC inversions, the RMSE values between predicted and true models, observed and predicted data (see Figure 12), the 90 % coverage ratios, and the computing time for the inversion.