Deep Learning in seismic inverse problems with Recurrent Inference Machines

Introduction

With the recent rise of deep learning, we have seen several successful applications in seismic processing and imaging. These include data denoising, deblending, interpolation, velocity-model building and automated seismic interpretation - with many of these applications being well-known inverse problems. In most such cases where supervised learning is employed, attention has been mostly placed on choosing a suitable learning architecture and pre-processing pipeline. This way, users often completely forego of conventional forward-operator-based optimisation techniques; instead replacing them by deep-learning architectures of choice. In that replacement, the known physics or forward operators pertaining to the geophysical or data-processing problem of choice is usually not invoked. Based on the recognition that a known forward operator can be a powerful constraint in solving inverse problems, Putzky and Welling (2017) proposed the Recurrent Inference Machines (RIM): a recurrent learning architecture that relies on known forward operators to build data-driven inference models for inverse problems. Here, we briefly introduce the original RIM as well as the more recent invertible RIM (iRIM) (Putzky and Welling, 2019; Putzky and Karkalousos et al., 2019) and discuss their use for seismic data interpolation and image-domain deblurring.

Recurrent Inference Machines (RIMs)

Within the context of inverse problems, the goal to estimate output model parameters \( p \) from input data \( d \) can be mathematically described as

\[
p = \mathcal{A}^{-1} d \quad \text{with} \quad d = \mathcal{A}(p, n),
\]

where the known forward operator \( \mathcal{A} \) allows computing \( d \) from \( p \) in the presence of noise \( n \). As such, RIMs are designed to impose the known forward operator as a constraint within training, by building on the recurrent neural net architecture to mimic the role of gradient-based optimisation schemes routinely applied to both linear and nonlinear inverse problems. To accomplish this, within the core of recurrent units are the so-called iterative inverse models:

\[
p_{t+1}, s_{t+1} = h_\phi(\mathcal{A}, d, p_t, s_t),
\]

where the local model state \( s_{t+1} \) can be thought of as a RIM-weighted model back-projection, and \( p_{t+1} \) the associated update to the model parameters. \( h_\phi \) is a parametric (e.g., convolutional) function within each recurrent unit acting as the proxy for a single step in an iterative inversion scheme - analogous to a single CG or l-BFGS iteration. Once presented with a set of training pairs (models and associated data) and the corresponding forward operators, training seeks to update the RIM-network model \( \phi \) by minimising the network loss function

\[
\mathcal{L}(p; d, \mathcal{A}, \phi) = \sum_t w_t \mathcal{L}_t(p, \hat{p}_t(\mathcal{A}, d, \phi)),
\]

Figure 1: Schematic representations of the recurrent units in RIMs. **A)** shows the non-invertible RIM recurrent unit, while **B)** and **C)** depict the forward and reverse recurrent units of the invertible RIM.
where $L_t$ and $w_t$ are user-defined, and $\hat{p}_t$ is a model prediction for the current network model $\phi$. Once the net work is trained, the output of $h_\phi$ can be seen as

$$s_{t+1} = f(\nabla D(d, \mathcal{A}(\mathcal{M}(\eta_t))), \eta_t, s_t),$$

$$\eta_{t+1} = \eta_t + g(\nabla D(d, \mathcal{A}(\mathcal{M}(\eta_t))), \eta_t, s_{t+1}),$$

(4)

where $\nabla D$ is the gradient of the data misfit (objective function) chosen for the inverse problem in eq. 1 and $\mathcal{M}$ is a link function mapping the RIM unit’s estimate $\eta_{t+1}$ to the parameter model $p_{t+1}$. In this form, it is clear that the parameters in eq. 4 carry meaning within the RIM, where $s_{t+1}$ is a local-unit estimate of a gradient back-projection, and $\eta_{t+1}$ is the corresponding local parameter model estimate. Unlike deterministic iterative optimisation approaches, the final form of $f$ and $g$ in eq. 4 is ultimately determined by the choices made for the training data, and architecture parameters such as number of recurrent units, loss functions, and training epochs. Below we discuss two distinct applications of RIMs in seismic processing, each using a different version of the architecture as depicted in Figure 1.

Given how RIMs are designed to mimic inference in inverse problems by iterative optimisation - informed by known forward operator as a constraint - they aim to learn:

- **Data-driven regularisation.** While deterministic optimisation requires user-supplied preconditioners, regularisers and often weight parameters, the RIM network model learns to regularise inference solutions from the data and noise scenarios provided during training.

- **Implicit model shaping.** Given the user-chosen training parameter models - including choices in parameterisation, features and perhaps geologic context - the RIMs learn to shape and constrain the behaviour of the output $p$ in inference that would otherwise appear as model-space constraints or preconditioners in traditional schemes.

Next, we discuss the application of RIM-based inference to seismic interpolation and image-domain deblurring. In both cases a classical end-to-end learning approach based on a UNet architecture (i.e., unaware of the modelling operator) is used as a benchmark against RIMs.

**Seismic interpolation**

As our first application, we investigate the use of RIMs to retrieve missing seismic traces, i.e., to infill spatially-subsampled wavefields (Kuijpers et al., 2021). In this case, when training the RIM network (Fig. 1A), the models $p$ consist of regularly well-sampled seismic data, $d$ are made up of several subsampled versions of $p$ using many choices of user-defined restriction operators $\mathcal{A} = \mathcal{R}$. The very same data and models are provided to train the UNet model.

In Fig. 2, we show inference examples of reconstructing a benchmark shot gather from a complex subsalt model, generated both by RIM and UNet. The inferences result from each network being trained with different dataset choices. Note that the data in Fig. 2A is not included in any of the training data sets, but other gathers from the same model are included in training leading to Figs. 2C and 2D. When the features of the target data used for inference are suitably represented by the training data (Figs. 2C and 2D), both RIM and UNet perform well in retrieving the missing data. However, when the training data do not closely resemble the target data (Figs. 2B and 2E), RIM significantly outperforms UNet in terms of data reconstruction: this is likely due to the role of the forward operator acting as a constraint in addition to the choice of the training data.

**Image-domain deblurring**

Next, we consider the problem of deblurring Kirchhoff prestack depth-migrated seismic images: this can be thought of as a multi-dimensional extension of poststack-migration impedance inversion. Here, $d$ are depth-migrated images, $p$ is a wide-bandwidth subsurface model (e.g., reflectivity or impedance),

83rd EAGE Annual Conference & Exhibition
Figure 2 RIM application to seismic data interpolation. Panel A shows the fully-sampled benchmark data from a numerical subsalt model. Panels B - E display reconstructions using either RIM (top) or UNet (bottom) inference networks. Each reconstruction panel corresponds to different data sets used in training: B used a field dataset (Gulf of Suez); C used modelled reflection data (Rdat) from the same numerical model used to generate the data in A; D combines both the field and numerical training data from B and C; E relies on data from a numerical model of ocean turbulence (i.e., a near-constant-velocity water layer with complex internal scattering). The input data are irregularly subsampled at 62% with the black sections in the bar above the data plots indicating the location of missing data. The numbers above each panel shows the misfit between their respective ML-based reconstructions and the benchmark data in A, according to the structure-similarity index metric (SSIM).

and $\mathcal{B} = \mathbf{B}$ is a blurring operator tied to the migration operation, e.g., by means of $\mathbf{B} = \mathbf{L}_B \mathbf{L}_B^\dagger$ with $\mathbf{L}_B$ a Born modelling operator (Fletcher et al., 2012). In this application, we rely on iRIM (Fig. 1) as well as on an invertible UNet for benchmarking. For training purposes, we approximate the response of $\mathbf{B}$ by local phase-weighted convolutions of the desired reflectivity with point-spread functions, generated by modelling and migration of point scatterers placed within the migration model used to compute $\mathbf{d}$.

Though our research on iRIM-based image deblurring is at an early stage, in Fig. 3 we show preliminary image deblurring results. In these examples, where both iRIM and iUnet are trained with noise-free data, they tend to perform well in retrieving wide-bandwidth reflectivity estimates; iRIM deliveres consistently wider-bandwidth results, particularly at the low-frequency end. However, when deblurring images in the presence of band-limited noise (not included in training), iRIM inferences perform noticeably better than iUNet ones due the action of the forward operator as a constraint in training.

Conclusions

When addressing inverse problems with machine learning, Recurrent Inference Machines (RIMs) represent a powerful solution in ensuring that a known forward operator is employed as a constraint during training process. When training RIMs/iRIMs, the choice of data and model scenarios, together with the design properties of a chosen forward operator, determine how the networks learn to jointly perform data-driven regularisation and implicit model shaping. Though it is still early days for these types of ar-
chitectures in geophysical problems, our initial experiences in the context of seismic data interpolation and image-domain deblurring show promise: where RIMs/iRIMs consistently deliver reliable inference results in cases beyond those included in the training data, due to the role of the forward operator as a training constraint.

Acknowledgements

We are grateful to Patrick Putzky and Max Welling (AMLab, Amsterdam) for supporting the initial implementation of their RIM architecture to seismic problems.

References