Interpolation of regularly sampled prestack seismic data with self-supervised learning

Satyakee Sen, Sribharath Kainkaryam, Cen Ong, Arvind Sharma, TGS*

Summary

We describe a framework for interpolation of broadband, prestack seismic data using a deep learning pipeline trained in a self-supervised manner. Unlike classification or segmentation tasks, image generation with convolutional neural networks (CNNs) is inherently more complex as the networks needs to learn the high dimensional distributions of broadband prestack gathers to be able to accurately reconstruct the missing traces. We highlight two main challenges specific to prestack seismic data for this task: (1) the choice of the loss function and (2) the lack of suitable training data in the form of image-label pairs to cast the problem as a supervised deep learning task.

We show that a naïve implementation of standard loss measures like sample-wise L2 or L1 leads to cycle skipping issues at high frequencies and undesirable smoothness in the mid-frequency reconstruction. To resolve this, we use a perceptual loss function computed using a pretrained variational auto-encoder (VAE) that penalizes differences between the interpolated and the input gathers in the high dimensional feature space of these gathers. To account for the lack of high-quality labelled training data we use a self-supervised learning scheme where a generator network is trained to read in random noise and produce the desired super-resolution output.

Introduction

Seismic data interpolation is one of the classical inverse problems in seismic processing that can be formulated as:

$$\boldsymbol{min\, f(m,d) + J(m)} \;, \qquad (1)$$

where *f(m,d)* is a data term measuring the discrepancy between the input data (*d*) and the interpolation result (*m*), while *J(m)* is a regularization term which tries to constrain the estimated model according to some apriory knowledge about the system (e.g. sparsity constraints). For irregularly sampled seismic data, equation (1) can be efficiently solved using iterative greedy solvers (Abma et al., 2006, and Xu et al., 2005) and produces state-of-the art results on production surveys. However, when the sampling scheme is regular, the sparsity constraint breaks down and complicated heuristics need to be designed to make sparsity-promoting methods work on such regularly sampled data (e.g., Curry et al., 2010). An alternative way of interpolating regularly sampled prestack data is using a deep learning framework to reformulate equation (1) as:

$$\boldsymbol{min\, f(L\{h_\theta(z)\}, d)} \;, \qquad (2)$$

where h_θ is a suitably designed convolutional neural network (CNN) trained as a generator, i.e. given *z* an image of random numbers of size (2N X 2N) drawn from a multi-dimensional gaussian distribution, one tries to optimize the weights of the generator such that the discrepancy between the network's super-resolution (by a factor of 2) output $h_\theta(z)$ and the original data *d* (N X N) is minimized after application of a standard subsampling operator *L*. In the remainder of the paper we describe some of the issues and practical solutions for solving equation (2) for field data

Theory

The first problem that needs to be addressed is what kind of training data should one use. While using synthetic data is a viable option, but it is well known that when the application domain is complex CNN models trained on synthetic data suffer from domain shifting issues even for far more simpler tasks like classification or segmentation (e.g. Hoffman et al., 2018). Another option is to use simpler interpolators (e.g. Wang et al., 2019) like bicubic or dip-based to generate the super-resolution data to act as labels. However, when the data is broadband and the structure is complex, such simple interpolators would usually produce sub-optimal results making them unsuitable to be used as ground truths. Instead we propose using a generator framework as shown in the top half of Figure 1 in the red box. The generator is a U-shaped encoder-decoder network that uses 4 encoder blocks with {32,64,128,256} channels along with *tanh* as non-linearities. The decoder uses nearest neighbor interpolation followed by 3X3 convolutions to upsample the feature maps. The generator's output is subsampled to make it the same size as the target input gather that needs to be interpolated to compute the loss which can be backpropagated through the generator. Notice that in this setup no ground truth label set is required, instead the generator is trained with the low-resolution input only in a fully self-supervised manner. Figure 3c shows the interpolation result (super sampling by a factor of 2) for a complex channel gather (Figure 3a) using a conventional loss function that measures the per pixel L2 loss between the generator's output and the target input image to update the weights of the network. From the interpolated result it is clear that while the events are reconstructed, excessive jitters are seen in the reconstruction (yellow ovals, zoomed view in Figure 4c). We attribute this to cycle-skipping issues at high frequencies. Since the L2 loss is computed in the low-dimensional space of the gathers, unless the reconstruction is within half a cycle of the input, the loss is extremely inaccurate.

10.1190/segam2019-3215774.1

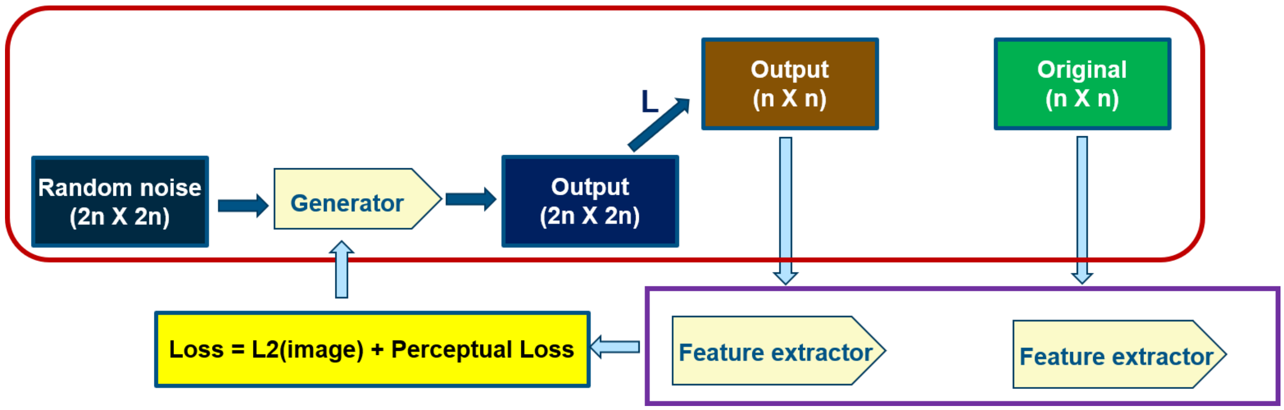

Figure 1: CNN pipeline for prestack interpolation. Top Red box: a *U shaped generator* network reads in random noise to produce a super-resolution gather which is subsampled with a sampling operator (L) and compared with the original gather (green-box) to compute a *per-pixel L2 loss*. Bottom Purple box: Both the generator's reconstruction and the original image are passed to fixed pretrained encoder network to extract high dimensional feature maps at different levels of this *feature extractor network*. Differences of these feature maps (*peceptual loss*) are used to augment the L2 loss (yellow box) and backpropagated through the generator to update its weights.

Perceptual Loss with pretrained VAE

In order to combat the problems of measuring the discrepancy between gathers in a low-dimensional 2D space, we use a setup similar to the one described in Gatys et al., 2015 and Johnson et al., 2016. The key idea is to use loss functions that penalize reconstruction errors on relevant high dimensional features of the gathers. If the features of the gather are adequately encoded, it is expected that such high dimensional feature representation would show adequate disentaglement such that loss functions can be computed which are not susceptible to cycle skipping.

To extract such high dimensional feature maps we use a custom VAE (Kingma et al, 2014), with residual blocks (He et al., 2016), trained with prestack seismic gathers. Once the VAE is trained, the decoder units are removed while the encoder part of the VAE (total of 35 layers) are retained to be used as an efficient feature extractor network, ϕ (indicated in the purple box in Figure 1). Figure 2b and 2c shows a sample feature map extracted for a broadband channel gather (Figure 2a) at layer 9 and layer 26 using ϕ. Notice that for the lower layer (Figure 2b) the feature map resembles the input image somewhat, while for the higher layer all spatial information has been lost while semantic information has been enriched. Using this pretrained feature extractor (ϕ) we extract all the n-feature maps ($n=64$) at layer-9 and construct a high dimensional content loss (L_c) between the features of the input and the generator's subsampled reconstruction as :

$$\boldsymbol{L_c} = \frac{1}{n}[\![\boldsymbol{\varphi(inp) - \varphi(recon)}]\!]^2 \ , \qquad (3)$$

The loss function in equation (3) encourages the reconstructed gather to have a similar representation as the input gather in the high dimensional feature space, rather than trying to enforce a match (in L2 sense) in the image space. Consequently we can expect that due to better separation of the key features of the gathers in this layer, the content differences between the input and the reconstruction can be computed in a more robust fashion when compared to a conventional per-sample L2 loss.

Unlike the lower layers, the feature maps extracted from the higher layer of a network encode the *style* of the image (Gatys et. Al., 2015), i.e. information about the gather's statistics. This information can be used to further improve the reconstruction task, by using the feature maps, F_l, at some level, l, of the network to construct Gram matrices, G_l, which are proportional to the covariance of these high dimensional encodings of the image. Figure 2d shows the Gram matrix for the higher layers' feature maps (Figure 2c) at layer 26. We utilize these Gram matrices (G) to compute a *style loss* as:

$$\boldsymbol{L_s} = \sum_l (\boldsymbol{G(inp_l) - G(recon_l)})^2, \boldsymbol{G_l = F_l^T F_l}\ , \qquad (4)$$

In the above equation the Gram matrices are computed at 3 high level layers (l) in the network, i.e. $\{l=14, 26, 35\}$ and are summed to produce the style loss. Notice that by using equations (3) and (4) we have effectively separated the loss computation between a reconstructed and original gather into two components: (1) a content loss, equation (3), that can efficiently track key content difference between the gathers and (2) the style loss, equation (4) which can be used to reduce the higher order statistical differences

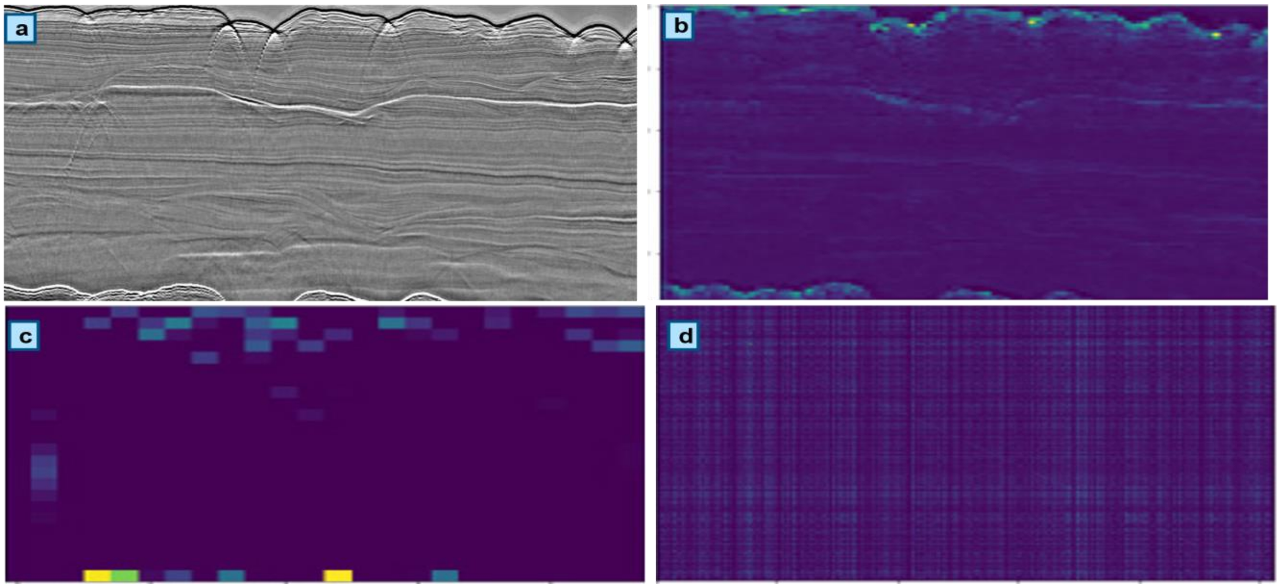

Figure 2: (a) channel gather, sample feature map extracted at (b) layer-9 , (c) layer-26 of the feature extractor network and (d) Gram matrix for all 256 feature maps at layer-26 which is proportional to the covariance of the 256-dimensional feature space at layer-26.

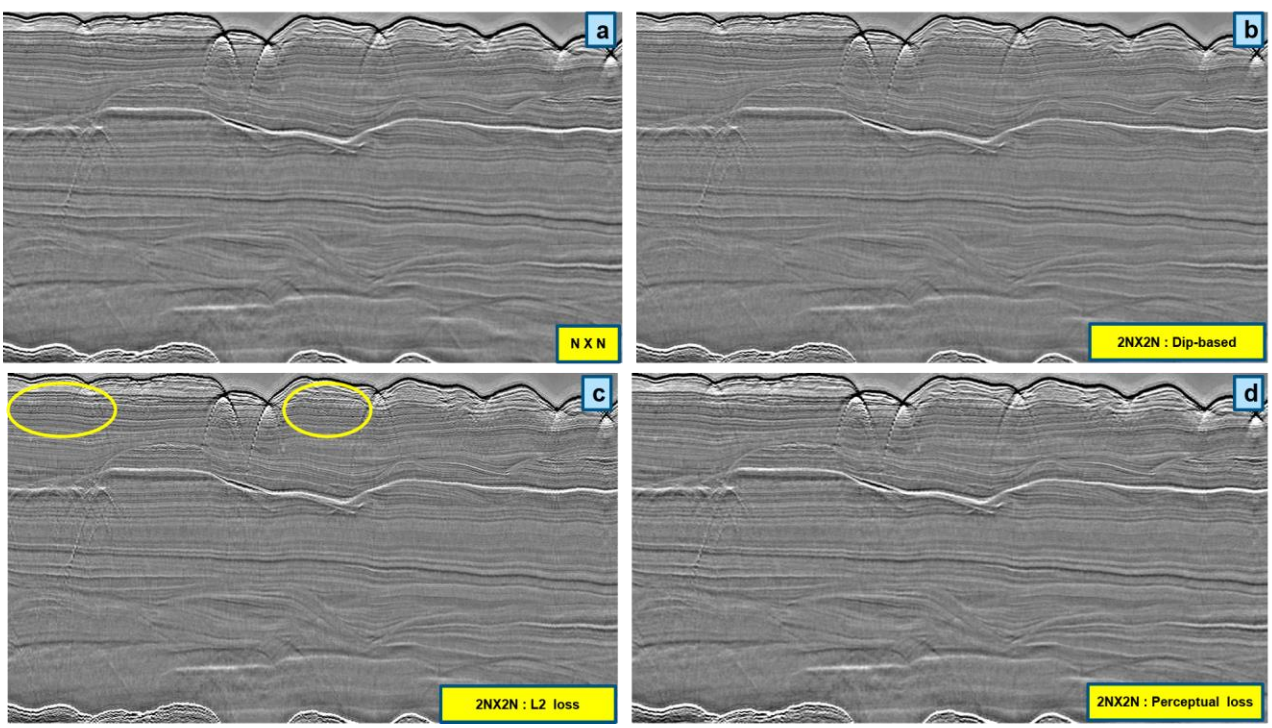

Figure 3: (a) original channel gather, interpolation with (b) conventional dip based, self supervised learning with (c) L2 loss only and (d) L2+perceptual loss.

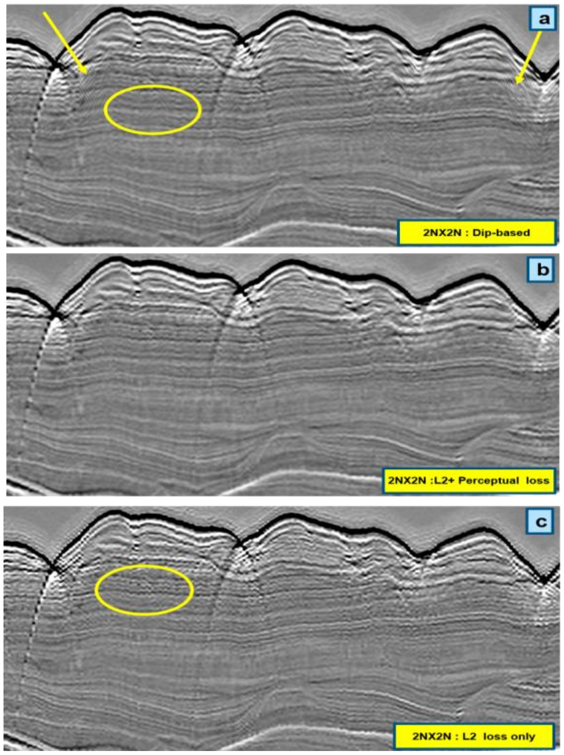

Figure 4: Zoomed view of interpolation by (a) dip based, CNN using (b) L2 + perceptual loss and (c) L2 loss only.

between the most relevant features of the reconstruction and the input. Together these two losses are referred to as the perceptual loss. The final loss that can now be used to update the generator's weights during the optimization process is given by:

$$\boldsymbol{Loss} = \boldsymbol{w_1}\|\boldsymbol{Lm} - \boldsymbol{d}\|^2 + \boldsymbol{w_2 L_c} + \boldsymbol{w_3 L_s}\,, \quad (5)$$

where w_i are the weights used to scale the conventional L2 reconstruction loss (first term), the content loss (second term) and the style loss (the third term). We note that during the training process, the feature extractor network is not updated, it is fixed and precomputed on a survey-by-survey basis. Only the generator's weights are updated to produce the interpolated gather. Figure 3d shows the interpolated gather using the loss function defined n equation (5). Notice that the cycle skipping issues (Figure 3c) have been effectively dealt with and undesirable smoothness in the reconstruction are also removed. Figure 4 compares a zoomed in view of the interpolation with a traditional dip-based interpolator the L2+perceptual loss based and L2-only reconstruction. Notice that the reconstruction (Figure 4b) is extremely accuracte and undesirable artefacts that a dip-based interpolator might produce (yellow oval and arrows, Figure 4a) in areas where the dip semblances are uncertain are effectively handled, while jitters due to cycle skipping (Figure 4c) are also removed by the CNN based interpolator described in this paper

One problem that our current implementation has and which has not been dealt with in this paper, is the lack of explicit anti-aliasing. Consequently our tests revealed, that the interpolation framework in some places do not handle aliased data adequately. In order to introduce anti-aliasing in this framework we plan to use a discriminator similar to a Generative Adversarial Network (GAN) within our framework. The task of the discriminator would be to compute a loss that explicitly penalizes reconstructed gathers that show aliasing. This loss can be futher used to augment the loss function described in equation (5) to introduce anti-aliasing into the framework.

Conclusions

We have described an efficient framework for interpolation of prestack, broadband seismic data using deep learning techniques. We have shown the use of a new loss function (the perceptual loss) computed using high -dimensional feature maps that a CNN produces which can overcome one of the perennial problems of computing the discrepancies between two seismic gathers (or images) that likely exhibit cycle-skipping issues. We anticipate that use of this kind of loss-functions can be used in other areas of seismic processing (e.g. FWI loss) where the optimization objective suffers from similar issues. Further for complex seismic signal processing tasks like denoise, we also anticipate that investigating the use of feature maps that a CNN produces can be an attractive alternative to the more traditional linearity based transform domain implementations (e.g. Fourier or Curvelet transform). In the context of interpolation, future work would include introducing explicit aliasing controls via a GAN framework to make the results more robust.

Acknowledements

The authors would like to that TGS management for permission to publish the abstract.

10.1190/segam2019-3215774.1

REFERENCES

Abma, R., and N., Kabir, 2006, 3D interpolation of irregular data with a POCS algorithm: Geophysics, **71**, no. 6, E91–E97, doi: https://doi.org/10.1190/1.2356088.

Curry, W., 2010, Interpolation with Fourier-radial adaptive thresholding: Geophysics, **75**, no. 6, WB95–WB102, doi: https://doi.org/10.1190/1.3500977.

Gatys, L., A. S., Ecker, and M., Bethge, 2015, A neural algorithm of artistic style: arXiv Preprint arXiv:1508.06576.

He, K., X., Zhang, S., Ren, and J., Sun, 2016, Deep residual learning for image recognition: IEEE Conference on Computer Vision and Pattern Recognition, 770–778.

Hoffman, J., E., Tzeng, T., Park, J., Zhu, P., Isola, K., Saenko, A., Efros, and T., Darrell, 2018, CyCADA: Cycle consistent adversarial domain adaptation: Proceedings of the 35th International Conference on Machine Learning, 1989–1998.

Johnson, J., A., Alahi, and F. F., Li, 2016, Perceptual losses for real time style transfer and super resolution: European Conference on Computer Vision, 694–711.

Kingma, D. P., and M., Welling, 2014, Auto-encoding variational Bayes: Proceedings of the 2nd International Conference on Learning Representation.

Wang, B., N., Zhang, W., Lu, and J., Wang, 2019, Deep-learning based seismic data interpolation: A preliminary result: Geophysics, **84**, no. 1, V11–V20, doi: https://doi.org/10.1190/geo2017-0495.1.

Xu, S., Y., Zhang, D., Pham, and G., Lambare, 2005, Antileakage Fourier transform for seismic data regularization: Geophysics, **70**, no. 4, V87–V95, doi: https://doi.org/10.1190/1.1993713.