Inverse Problems, Deep Learning, and Symmetry Breaking

Kshitij Tayal [1] **Chieh-Hsin Lai** [2] **Raunak Manekar** [1] **Vipin Kumar** [1] **Ju Sun** [1]

Abstract

In many physical systems, inputs related by intrinsic system symmetries are mapped to the same output. When inverting such physical systems, i.e., solving the associated inverse problems, there is no unique solution. This causes fundamental difficulty in deploying the emerging end-to-end deep learning approach. Using the generalized phase retrieval problem as an illustrative example, we show that careful symmetry breaking on training data can help remove the difficulty and significantly improve the learning performance. We also extract and highlight the underlying mathematical principle of the proposed solution, which is directly applicable to other inverse problems. A full-length version of this paper can be found at https://arxiv.org/abs/2003.09077.

1. Introduction

For many physical systems, we observe only the output and strive to infer the input. The inference task is called *inverse problem*. Formally, the underlying system is modeled by a forward mapping f, and solving the inverse problem amounts to identifying the inverse mapping f^{-1}.

Let $\boldsymbol{y}$ denote the observed output. Traditionally, inverse problems are phrased as regularized optimization problems: $\min_{\boldsymbol{x}} \ \ell(\boldsymbol{y}, f(\boldsymbol{x})) + \lambda\Omega(\boldsymbol{x})$, where $\boldsymbol{x}$ represents the input to be estimated, $\ell(\boldsymbol{y}, f(\boldsymbol{x}))$ ensures $\boldsymbol{y} \approx f(\boldsymbol{x})$, and $\Omega(\boldsymbol{x})$ encodes structural information about $\boldsymbol{x}$ to make the problem well posed. Recently, with the resurgence of deep learning, data-driven functions are proposed to replace parts or the entirety of ℓ, Ω, and numerical methods for solving the regularized optimization. The most radical is using neural networks to directly approximate f^{-1}, also known as the end-to-end approach. Several review articles (McCann et al., 2017; Lucas et al., 2018; Arridge et al., 2019; Ongie et al., 2020) have covered these recent developments. We note that most of the success examples are about *linear* inverse problems, where the forward mapping f is linear.

In this paper, we focus on the end-to-end approach applied to nonlinear inverse problems. This combination is attractive because when f is nonlinear, in practice it may be tricky to gather complete information of f, rendering alternatives that need accurate modeling of f ineffective. We are interested in f's with symmetries that systematically map different $\boldsymbol{x}$'s to the same $\boldsymbol{y}$, e.g., $f(x) = x^2$ with a sign symmetry, $f(\boldsymbol{A}, \boldsymbol{X}) = \boldsymbol{A}\boldsymbol{X}$ with scaling and permutation symmetries, and f that is invariant to translation and rotation of the input (translation and rotation symmetries). Such symmetric forward mappings are prevalent in nonlinear inverse problems.

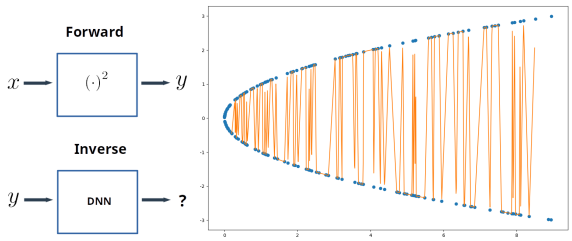

Figure 1. Learn to take square root. (Left) The forward and inverse models; (Right) The function (in orange) determined by the training set.

Symmetries can cause significant difficulty for the end-to-end approach. To see this, suppose we randomly sample real values x_i's and form a training set $\{x_i, x_i^2\}$ and try to learn the square-root function, allowing both positive and negative outputs, using the end-to-end approach. Now if we think of the function determined by the training set, which the neural network is trying to approximate, it is highly oscillatory (see Fig. 1)[1]: the sign symmetry dictates that in the training set, there are frequent cases where x_i^2 and x_j^2 are close but x_i and x_j have different signs and are far apart. Although in theory neural networks with adequate capacity are universal function approximators, in practice they will struggle to learn such irregular functions. For general inverse problems,

[1]Department of Computer Science and Engineering, University of Minnesota, Twin Cities, USA. [2] School of Mathematics, University of Minnesota, Twin Cities, USA. Correspondence to: Kshitij Tayal <tayal007@umn.edu>, Raunak Manekar <manek009@umn.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020.

[1]Interestingly, the more train samples one gathers, the more serious the problem is.

so long as the forward symmetries can relate remote inputs to the same output, similar problems can surface.

In learning the square root, one can quickly eliminate the difficulty by fixing the signs of x_i's to either positive or negative—which breaks the sign symmetry, and effectively learn the positive or negative square root function. In this paper, we show that similar symmetry breaking can be performed for nontrivial problems, and also identify the underlying mathematical principle. We take the generalized phase retrieval problem as a first illustrative example.

2. Generalized Phase Retrieval

Phase retrieval (PR) is a classic problem in computational imaging with stacks of applications (Bendory et al., 2017). This paper works with two modified (also simplified) versions of PR to illustrate our symmetry breaking idea:

Real Gaussian PR The forward model: $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2$, where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$, and $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is iid real Gaussian. The absolute-square operator $|\cdot|^2$ is applied elementwise. The only symmetry is sign, and $\boldsymbol{x}$ and $-\boldsymbol{x}$ are mapped to the same $\boldsymbol{y}$.

Complex Gaussian PR The forward model: $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2$, where $\boldsymbol{x} \in \mathbb{C}^n$, $\boldsymbol{y} \in \mathbb{R}^m$, and $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ is iid complex Gaussian. The modulus-square operator $|\cdot|^2$ is applied elementwise. The only symmetry is global phase shift, and $e^{i\theta}\boldsymbol{x}$ for all $\theta \in [0, 2\pi)$ are mapped to the same $\boldsymbol{y}$.

These two versions have been intensively studied in the recent developments of generalized PR; see, e.g., (Candes et al., 2015; Sun et al., 2017; Fannjiang & Strohmer, 2020).

2.1. Real Gaussian PR

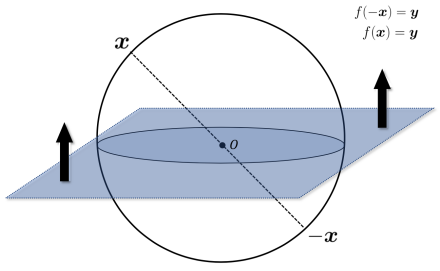

Figure 2. Symmetry breaking for real Gaussian phase retrieval.

In learning the square root, there is a sign symmetry and we can break it by restricting all desired network outputs to be positive. Here, the symmetry is the global sign of vectors and antipodal points are mapped to the same observation. Thus, an intuitive generalization is breaking antipodal point pairs, and a simple solution is to make a hyperplane cut and take samples from only one side of the hyperplane! This is illustrated in Fig. 2 where we use the xy-hyperplane in $\mathbb{R}^3$.

In $\mathbb{R}^3$, the upper half space cut out by the xy-plane is connected. Moreover, it is representative as any point in the space (except for the plane itself) can be represented by a point in this set by appropriate global sign adjustment, and it cannot be made smaller to remain representative. The following proposition says that these properties also hold for high-dimensional spaces.

Proposition 2.1. *Let* $\mathscr{R} \doteq \{\boldsymbol{x} \in \mathbb{R}^n : x_n > 0\}, Z \doteq \{\boldsymbol{x} \in \mathbb{R}^n : x_n = 0\}$. *The following properties hold: 1)* ***(connected)*** $\mathscr{R}$ *is connected in* $\mathbb{R}^n$*; 2)* ***(representative)*** Z *is of measure zero (Rudin, 2006) and for any* $\boldsymbol{x} \in \mathbb{R}^n \setminus Z$, *either* $\boldsymbol{x} \in \mathscr{R}$ *or* $-\boldsymbol{x} \in \mathscr{R}$*; and 3)* ***(smallest)*** *No* $\boldsymbol{x} \in \mathscr{R}$ *can be represented by points in* $\mathscr{R} \setminus \{\boldsymbol{x}\}$.

The coordinate hyperplane Z we use is arbitrary, and we can prove similar results for arbitrary hyperplanes. The set Z is negligible, as the probability of sampling a point exactly from Z is zero. In fact, we can break the symmetry in Z also by recursively applying the current idea. For the sake of simplicity, we will not pursue it here.

Now we can apply the above result to preprocess the training samples $\{\boldsymbol{x}_i, |\boldsymbol{A}\boldsymbol{x}_i|^2\}$ for symmetry breaking: for all $\boldsymbol{x}_i$'s, if $\boldsymbol{x}_i$ lies above Z, we simply leave it untouched; if $\boldsymbol{x}_i$ lies below Z, we switch the sign of $\boldsymbol{x}_i$; if $\boldsymbol{x}_i$ happens to lie on Z, we make a small perturbation to $\boldsymbol{x}_i$ and then adjust the sign as before. Now $\boldsymbol{x}_i \in \mathscr{R}$ for all i. Since $\mathscr{R}$ is a connected set, when there are sufficiently dense training samples, small perturbations to $|\boldsymbol{A}\boldsymbol{x}_i|^2$ always lead to only small perturbations to $\boldsymbol{x}_i$. So we now have a nicely behaved underlying function.

The three properties are also necessary for effective symmetry breaking. Being representative is easy to understand. If the representative set is not the smallest, symmetry remains for certain points in the set and so symmetry breaking is not complete. Now the set can be smallest representative but not connected. An example in the setting of Proposition 2.1 would be taking out a $B \subsetneq \mathscr{R}$, and considering $M \doteq (-B) \cup (\mathscr{R} \setminus B)$. It is easy to verify that M is smallest representative, but not connected. This leaves us the trouble of approximating (locally) oscillatory functions.

2.2. Complex Gaussian PR

We now move to the complex case and deal with a different kind of symmetry. Recall that in the complex Gaussian PR, $e^{i\theta}\boldsymbol{x}$ for all $\theta \in [0, 2\pi)$ are mapped to the same $|\boldsymbol{A}\boldsymbol{x}|^2$, i.e., global phase shift is the symmetry. These "equivalent" points form a continuous curve in the complex space, contrasting the antipodal point pairs in the real case. Inspired by the real version, we seek the three properties in symmetry

breaking. It turns out we can perform effective symmetry breaking as follows.

Proposition 2.2. *The set* $\mathscr{R} \doteq \{\boldsymbol{x} \in \mathbb{C}^n : \mathrm{Im}(x_1) = 0, x_1 > 0\}$ *is a connected, smallest representative set for* $\mathbb{C}^n \setminus Z$ *with* $Z \doteq \{\boldsymbol{x} \in \mathbb{C}^n : x_1 = 0\}$. *Moreover,* Z *is a measure-zero subset of* $\mathbb{C}^n$.

Here Z is a coordinate hyperplane in $\mathbb{C}^m$ and $\mathscr{R}$ is restricted half-space. $\mathscr{R}$ enjoys the three desired properties, similar to the real case, despite the different symmetry. If we emulate the argument for the real case, the construction leads to effective symmetry breaking for the current complex case.

For general inverse problems, although the symmetries might be very different than here and the sample spaces could also be more complicated, the three properties, which concern only the geometric and topological aspects of the space, can be generalized as a basic mathematical principle for effective symmetry breaking. Also, although the symmetry breaking schemes for our couple of examples seem simple, this is the first time this general issue is addressed. Symmetry breaking for general nonlinear inverse problems that potentially contain multiple symmetries (e.g., Fourier PR (Bendory et al., 2017)) might not be as straightforward.

3. Numerical Experiments

Simple comparison Our focus here is to illustrate how symmetry breaking can lead to more efficient learning than without for the end-to-end approach, and not to pick the best approach for solving Gaussian PR. Hence we will not compare our method with regularized optimization or its hybrid with data-driven approach.

Learning models We set up an end-to-end pipeline and use neural network models to approximate the inverse mappings. Learning models we consider are: **Neural Network (NN)**: Feedforward network with architecture m-256-128-64-n. **Wide Neural Network (WNN)**: The architecture is m-512-256-128-n. **Deep Neural Network (DNN)**: The architecture is m-2048-1024-512-256-128-n. **K-Nearest Neighbors (K-NN)** K-NN regression, where output is the average of the values of $K = 5$ nearest neighbors.

Data We take $m = 4n$, which is just above the threshold for ensuring injectivity (up to the symmetry) (Balan et al., 2006). To generate data samples, we draw iid uniformly random data points $\boldsymbol{x}_i$'s from the unit ball and consequently generate $\{(\boldsymbol{x}_i, |\boldsymbol{A}\boldsymbol{x}_i|^2)\}$ as the simulated datasets. All the datasets are split into 80% training and 20% test, and 10% of the training data are used for validation.

We conduct experiments with varying input dimension n and dataset size. Specifically, we experiment with $n = 5, 10, 15$ and dataset size of $2e4$, $5e4$, $1e5$, $1e6$, respectively. We do not test higher dimensions, in view of the exponential growth of sample requirement to cover the high-dimensional ball. For most practical inverse problems where the input data tend to possess low-dimensional structures despite the apparent high dimensions, the sample requirement will not be an issue and our symmetric breaking scheme naturally adapts to the structures.

For all neural network models, we train them based on two variants of the training samples: one with symmetry untouched (i.e., before symmetry breaking) and one with symmetry breaking (i.e., after symmetry breaking). To distinguish the two variants, we append our neural network model names with "-A" to indicate *after* symmetry breaking and "-B" to indicate *before* symmetry breaking.

Why no real-data experiments? One may suggest performing a real-data experiment on natural images, as was done in numerous previous papers (Candès et al., 2012; Candes et al., 2015; Sun et al., 2017). This is sensible, but not interesting in the Gaussian PR context, as the sign (resp. phase transfer) symmetry for real (resp. complex) PR is naturally broken due to the restriction of the image values to be real nonnegative. Moreover, the Gaussian settings erase the essential difficulties of the classic PR, which entails in addition flipping and translation symmetries than complex Gaussian PR. One needs to work out the symmetry breaking in order to perform real-data experiments under the classic setting; we leave it as future work.

Training and error metric The mean loss is used in the objective. We use the Adam optimizer (Kingma & Ba, 2014) and train all models for a maximum of 100 epochs. The learning rate is set as 0.001 by default and training is stopped if the validation loss does not reduce for 10 consecutive epochs. The validation set is also used for hyperparameter tuning. To train the models for the complex PR, real and complex parts of any complex vector are concatenated into a long real vector. The K-NN model is fit on the whole training dataset and serves as a baseline.

To imitate the real-world test scenario, we do not perform symmetry breaking on the test data. To measure performance, we use the normalized mean square error (MSE) which is rectified to account for the symmetry:

$$\varepsilon_{\text{real}} = \min_{s\in\{+1,-1\}} \frac{||\widehat{\boldsymbol{x}}s - \boldsymbol{x}||^2}{n}, \qquad \text{(real)} \quad (3.1)$$

$$\varepsilon_{\text{comp}} = \min_{\theta\in[0,2\pi)} \frac{||\widehat{\boldsymbol{x}}e^{i\theta} - \boldsymbol{x}||^2}{n}., \qquad \text{(complex)} \quad (3.2)$$

where $\widehat{\boldsymbol{x}}$ is the prediction by the learned models.

Quantitative Results Table 1 provides test errors for all models trained for real PR. All models for the same combination of input dimension n and sample size use the same

Table 1. Summary of results in terms of test error for real Gaussian PR. These numbers needs to be scaled by 10^{-4}. Blue coloring indicate the best performance in each row.

n	**Sample**	**NN-A**	**K-NN**	**NN-B**	**WNN-A**	**K-NN**	**WNN-B**	**DNN-A**	**K-NN**	**DNN-B**
5	2e4	10	17	283	**8**	18	283	10	19	284
	5e4	**6**	12	282	8	17	284	7	14	285
	1e5	**5**	10	284	**5**	12	283	13	18	284
	1e6	**4**	7	283	5	6	283	7	8	283
10	2e4	11	20	82	9	22	82	**8**	21	82
	5e4	9	16	82	**6**	18	82	9	20	82
	1e5	9	16	82	**6**	15	82	8	17	82
	1e6	7	13	82	**5**	10	82	9	11	82
15	2e4	12	17	38	**9**	16	38	**9**	16	38
	5e4	11	14	38	9	14	38	**8**	15	38
	1e5	10	13	38	8	13	38	**7**	13	38
	1e6	8	9	38	**7**	10	38	9	10	38

set of data. Blued numbers in the tables indicate the best-performing models across all the models in each row.

We first note that for the same NN architecture with any dimension-sample combination, symmetry breaking always leads to substantially improved performance. Without symmetry breaking, i.e., as shown in the $(\cdot)$-B columns, the estimation errors are always worse, if not significantly so, than the simple baseline K-NN model. By contrast, symmetry breaking as shown in the $(\cdot)$-A columns always leads to improved performance compared to the baseline. To rule out the possibility that the inferior performance of $(\cdot)$-B's is due to the capacity of the NNs, we compare the performance of NN with that of WNN and DNN, where the latter two have $4\times$ and $50\times$ more parameters than the plain NN.

From Table 1, it is clear that larger capacities do not lead to improved performance, suggesting that our plain NNs are probably already powerful enough. Moreover, for a fixed learning model, increasing the number of samples beyond $2e4$ yields only marginally improved errors, suggesting that bad performance cannot be attributed to lack of samples. These observations together show that $(\cdot)$-B's are inefficient learners, as they do not explicitly handle the symmetry issue.

Moreover, as the dimension n grows, there is a persistent trend that the $(\cdot)$-B models performs incrementally better. This might be counter-intuitive at first sight, as the coverage of the space becomes sparser as the dimension grows with same number of random samples and hence a reverse trend could be expected. But as we hinted in Section 1, more training samples also cause more wildly behaved functions—the problem becomes less severe when the dimension grows, as the density of sample points becomes smaller. In fact, when the sample density is extremely low, the other trend that is dictated by the lack of samples could reveal. Nonetheless, here we focus on the data-intensive regime. Overall, the difficulty of approximating highly oscillatory functions and the benefit of breaking breaking are evident.

We obtain similar results on complex PR, which can be found in our full paper (Tayal et al., 2020).

4. Related Work

The end-to-end approach is attractive when (i) the forward model is unknown—e.g., for complex imaging systems (Horisaki et al., 2016; Li et al., 2018); (ii) or alternatives rarely work, e.g., PR (Fienup, 1982; Sinha et al., 2017).

The end-to-end approach has been applied to a number of nonlinear inverse problems with symmetries, e.g., blind image deblurring (i.e., blind deconvolution) (Tao et al., 2018), real-valued Fourier phase retrieval (Sinha et al., 2017), 3D surface tangents and normal prediction (Huang et al., 2019). We believe our work represents the first time that the difficulty of learning with symmetries is articulated and addressed.

Mathematically, points related by symmetries form an equivalence class and these equivalence classes form a partition of the input space for the forward model. Our symmetric breaking task effectively consists in finding a *consistent* representation for the equivalence classes, where the consistency here requires the set of the representatives to be topologically connected.

One limitation of our approach is figuring out the symmetry, which requires full or partial (maybe through experimentation) knowledge about the forward model which may not be available in general. As such, we refer the readers to the exciting work of (Krippendorf & Syvaeri, 2020) which present methods to detect symmetries based on the learning approach.

Acknowledgments

KT and VK are supported by NSF BIGDATA 1838159. The authors acknowledge the Minnesota Supercomputing

Institute (MSI) at the University of Minnesota for providing resources that contributed to the research results reported within this paper.

References

Arridge, S., Maass, P., Öktem, O., and Schönlieb, C.-B. Solving inverse problems using data-driven models. *Acta Numerica*, 28: 1–174, may 2019. doi: 10.1017/s0962492919000059.

Balan, R., Casazza, P., and Edidin, D. On signal reconstruction without phase. *Applied and Computational Harmonic Analysis*, 20(3):345–356, may 2006. doi: 10.1016/j.acha.2005.07.001.

Bendory, T., Beinert, R., and Eldar, Y. C. Fourier phase retrieval: Uniqueness and algorithms. In *Compressed Sensing and its Applications*, pp. 55–91. Springer International Publishing, 2017. doi: 10.1007/978-3-319-69802-1_2.

Candès, E. J., Strohmer, T., and Voroninski, V. PhaseLift: Exact and stable signal recovery from magnitude measurements via convex programming. *Communications on Pure and Applied Mathematics*, 66(8):1241–1274, nov 2012. doi: 10.1002/cpa.21432.

Candes, E. J., Li, X., and Soltanolkotabi, M. Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007, apr 2015. doi: 10.1109/tit.2015.2399924.

Fannjiang, A. and Strohmer, T. The numerics of phase retrieval. *arXiv:2004.05788*, 2020.

Fienup, J. R. Phase retrieval algorithms: a comparison. *Applied Optics*, 21(15):2758, aug 1982. doi: 10.1364/ao.21.002758.

Horisaki, R., Takagi, R., and Tanida, J. Learning-based imaging through scattering media. *Optics Express*, 24(13):13738, jun 2016. doi: 10.1364/oe.24.013738.

Huang, J., Zhou, Y., Funkhouser, T., and Guibas, L. J. Framenet: Learning local canonical frames of 3d surfaces from a single rgb image. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8638–8647, 2019.

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

Krippendorf, S. and Syvaeri, M. Detecting symmetries with neural networks. *arXiv preprint arXiv:2003.13679*, 2020.

Li, Y., Xue, Y., and Tian, L. Deep speckle correlation: a deep learning approach toward scalable imaging through scattering media. *Optica*, 5(10):1181, sep 2018. doi: 10.1364/optica.5.001181.

Lucas, A., Iliadis, M., Molina, R., and Katsaggelos, A. K. Using deep neural networks for inverse problems in imaging: Beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1): 20–36, jan 2018. doi: 10.1109/msp.2017.2760358.

McCann, M. T., Jin, K. H., and Unser, M. Convolutional neural networks for inverse problems in imaging: A review. *IEEE Signal Processing Magazine*, 34(6):85–95, nov 2017. doi: 10.1109/msp.2017.2739299.

Ongie, G., Jalal, A., Metzler, C. A., Baraniuk, R. G., Dimakis, A. G., and Willett, R. Deep learning techniques for inverse problems in imaging. *arXiv:2005.06001*, 2020.

Rudin, W. *Real and complex analysis*. Tata McGraw-hill education, 2006.

Sinha, A., Lee, J., Li, S., and Barbastathis, G. Lensless computational imaging through deep learning. *Optica*, 4(9):1117, sep 2017. doi: 10.1364/optica.4.001117.

Sun, J., Qu, Q., and Wright, J. A geometric analysis of phase retrieval. *Foundations of Computational Mathematics*, 18(5): 1131–1198, aug 2017. doi: 10.1007/s10208-017-9365-9.

Tao, X., Gao, H., Shen, X., Wang, J., and Jia, J. Scale-recurrent network for deep image deblurring. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2018. doi: 10.1109/cvpr.2018.00853.

Tayal, K., Lai, C.-H., Kumar, V., and Sun, J. Inverse problems, deep learning, and symmetry breaking. *arXiv preprint arXiv:2003.09077*, 2020.