Vol. xx, pp. x
© xxxx Society for Industrial and Applied Mathematics
x–x

PR-DAD: Phase Retrieval Using Deep Auto-Decoders

Leon Gugel and Shai Dekel *

Abstract. Phase retrieval is a well known ill-posed inverse problem where one wishes to recover images given only the magnitude values of the Fourier transform as input. In recent years, new algorithms based on deep learning have been proposed, providing breakthrough results that surpass previous results of the classical methods. In this work we provide a novel deep learning architecture whose components are carefully designed based on mathematical modeling of the phase retrieval problem. The architecture provides experimental results that surpass all current results in the supervised setting, but more importantly, provides a robust solution for the semi-supervised setting.

Key words. Phase retrieval, deep learning.

AMS subject classifications. 65T60, 65Y10, 68U10.

1. Introduction .

1.1. The Phase Retrieval problem and classical methods.

The two-dimensional discrete Fourier transform $\mathcal{F}(x)$ of an image $x \in \mathbb{R}^{n\times n}$, can be represented by the magnitude

$$\omega(x) := |\mathcal{F}(x)| \in \mathbb{R}^{n\times n},$$

and the phase

$$\varphi(x) := \arg \mathcal{F}(x) \in [-\pi, \pi]^{n\times n},$$

where $\arg M$ denotes the argument of a complex matrix M applied element-wise. The Fourier phase retrieval is a famous ill-posed inverse problem where the goal is to recover x, or equivalently the phase $\varphi(x)$, only from the input of the magnitude $\omega(x)$. This problem arises arises in many areas in engineering and science and has a rich history tracing back to 1952 [4]. Important examples for Fourier phase retrieval naturally appear in many optical settings since optical sensors, such as a charge-coupled device (CCD) and the human eye, are insensitive to phase information of the light wave. A typical example is coherent diffraction imaging (CDI) which is used in a variety of imaging techniques (see [1] and references therein). In CDI, an object is illuminated with a coherent electro-magnetic wave and the far-field intensity diffraction pattern is measured. This pattern is proportional to the object's Fourier transform and therefore the measured data is proportional to its Fourier magnitude. Phase retrieval also played a key role in the development of the DNA double helix model [3]. Additional examples for applications in which Fourier phase retrieval appear are X-ray crystallography, speech recognition, blind channel estimation, astronomy, computational biology, alignment and blind deconvolution (see [1] and references therein).

The classical techniques for phase retrieval are iterative methods such as the alternating projection (see the survey [1]). The general scheme of the alternating projection is at each step k

(i) compute the Fourier transform $\mathcal{F}(x_k)$ of the current estimated image x_k,

*Shai Dekel, School of mathematical sciences, Tel-Aviv university (shaidekel6@gmail,com).

(ii) keep its phase information $\varphi(x_k)$, and replace the magnitude by the known ground truth magnitude $\omega(x_k) = \omega(x)$,
(iii) compute the inverse Fourier to obtain a temporary estimate $\tilde{x}_{k+1}$,
(iv) impose certain known constraints, if needed, on $\tilde{x}_{k+1}$ (e.g. real non-negative pixel values), to obtain x_{k+1}.

The PhaseCut method [7] is based on the following minimization formulation for the the input modulus ω, unknown image $x = \{x_{j,k}\}$ with unknown phase $\varphi = \{\varphi_{j,k}\}$

$$\min_{x,\varphi} \|\mathcal{F}(x) - \omega \cdot \varphi\|^2, \quad \text{s.t } |\varphi_{j,k}| = 1, \forall j, k.$$

There are several ways to relax this formulation and derive from it a minimization problem in the phase only, especially if x is known to be real.

1.2. The learning setup. When we apply learning methods to an inverse problem such as phase retrieval, we need to clarify if we are attempting to solve the problem in the supervised, semi-supervised or un-supervised setting. First observe that one can easily compute the Fourier magnitude values for any ground truth image and then this pair can be used for supervised training.

- **Supervised:** In this case we provide the trained model access to pairs of Fourier magnitude inputs and their corresponding ground truth images. Using these pairs, one can design a loss function such as Mean Square Error (MSE) that will drive the minimization of a gradient descent method during the training of the model.
- **Semi-supervised:** In this setting only a partial subset of the Fourier magnitude inputs has corresponding ground truth images. This may happen in cases where we have acquired the Fourier magnitude of data through an acquisition process, but we do not have knowledge about the ground truth image, beyond the fact that it is a faithful representative of the given class. This typically implies that to use the Fourier magnitude inputs which have no matching ground truth pixels during the training process, one needs to add additional loss mechanisms. One such loss function is the cycle loss which computes the Fourier magnitude of the images generated by the model and then compares them with the input Fourier magnitudes. Another loss is the adversarial loss where a discriminator network is trained to provide a prediction if the image generated from the Fourier magnitude is plausible, i.e. if it belongs to the given class of images.
- **Un-supervised:** Here, we work with a dataset that has only Fourier magnitude inputs with no ground truth images at all. In this case we can only use loss functions such as the cycle loss to drive the training of our model. One can not use the adversarial loss since there are no ground truth images that can be used as reference for the discriminator. However, in the case if there is some general prior knowledge on the structure of the given class of images, one can potentially transfer this knowledge into the form of a regularization loss function on the model's output images during training.

In this work we assume that we are in the supervised or semi-supervised regimes, where we have a sufficient amount of ground truth image samples from the given class. Indeed, we argue that any inverse method based on learning can truly outperform the classical methods on a given class of images, only if that class has sufficient structure and the learning algorithm can

'study' that structure before it can begin to infer approximate pixel images from inputs it has not previously observed. Moreover, in many practical applications, one can generate synthetic ground truth pixel images that provide faithful representative samples of their class. For example, in the setting of x-ray crystallography one can generate synthetic virtual molecules from which one can compute pixel image slices. It is also possible to simulate and inject various noise models into the Fourier magnitudes of these images.

1.3. Overview of Recent Deep Learning based methods. We now review some recent work where deep learning methods are applied to the problem of phase retrieval.

The DeepPhaseCut architecture [2] starts with a modified U-net generator $\mathcal{G}_\Theta$ that takes as input the Fourier modulus and predicts the Fourier phase. We note in passing that applying a convolutional network, such as a U-net, on a frequency representation, is perhaps not optimal, since there are typically no spatial correlations between 'neighboring' Fourier coefficients or their respective modulus. The predicted phase is then multiplied by the modulus to give a predicted Fourier transform. Then, an inverse Fourier transform is applied to obtain a predicted intermediate image. The intermediate image is then fed into an enhancement network $\mathcal{H}_\Psi$ to obtain the predicted image. The network is trained using several losses such as a cycle consistency / conditional loss, where the Fourier modulus is extracted from the predicted image and compared to the input modulus. The authors also trained discriminators that provide a score relating to the belief that the input is a ground truth image or generated from modulus input. This allows to use adversarial loss during training.

In [6], the same concepts were used, namely, a generator was trained to take as input the Fourier modulus and output a predicted image. The generator was trained with a linear combination of conditional and advesarial losses. Here as well, a discriminator was trained simultaneously to provide the advesarial loss. The authors of [6] note that a generator architecture based on fully connected layers provided better empirical results than a convolutional architecture, which aligns with our understanding.

In [5] the authors propose to use a Cascaded Phase Retrieval (CPR) neural network architecture consisting of a sequence of sub-networks $G^{(1)}, \ldots, G^{(q)}$. Each subnetwork $G^{(i+1)}$ is fed as input the known magnitude $\omega(x)$ and $\hat{x}^{(i)} \in \mathbb{R}^{n_i \times n_i}$, an estimate of the image at some given (lower) resolution which is the output of the subnet $G^{(i)}$. The last subnet $G^{(q)}$ predicts the image x at the full resolution. The CPR network is trained with a loss function that incorporates all of the elements of the sequence of multiresolution approximations $\{\hat{x}^{(i)}\}$.

2. Overview of the PR-DDL architecture . As already stated, in this work we assume that we have a sufficient amount of ground truth image samples from the given class which allows us to first learn some aspects of the structure of the class during a preprocessing stage. As we shall see, we do this by first training a carefully designed auto-encoder/decoder DL architecture. Once this learning mechanism collects important information about the given class of images, we are able to extract the decode part and plug it into our phase retrieval inference network. Now, the central idea of our phase retrieval architecture is to use the prior knowledge about the encoded structure of the class and ensure the network first maps the input Fourier magnitude to a 'Fourier-type' representation of the encoded form of the image. Once this representation is obtained by the first part of the network, it undergoes an inverse transform into the pixel regime, where it is then auto-decoded by the pre-trained decoder and

finally enhanced. It is crucial to observe that it is the existence of the pre-trained decoder component that 'forces' the network to learn to prepare the incoming data in the encoded form we designed.

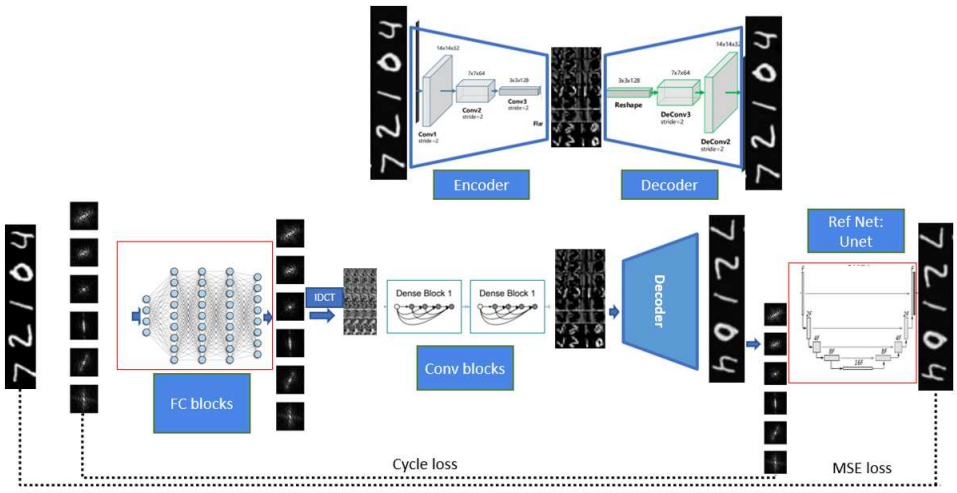

Figure 2.1. *PR-DDL architecture*

2.1. Preprocessing: Training The Auto-Encoder/Decoder Network. We assume we are given a dataset of a certain class of images which contains some ground truth images. During our preprocessing stage, we train a pair of auto encoder/decoder whose goal is to learn a set of nonlinear projections for the given class. The main idea is that the projections will be onto low resolution spaces. A typical example is that a class of images of dimensions 32×32 will be projected by the encoder part onto 128 images of sizes 8×8. In such as case, although the encoding process produces $\times 8$ more pieces of information for an encoded image, the information is represented using a collection of low dimension elements.

the architecture

The training applies the MSE loss between the input images and the outputs of the decoder.

See examples in Figures

2.2. The Fourier Magnitude to Auto-Encoder Representation Subnet. The goal of this subnet is to predict from the Fourier magnitude input, the auto-encoder features of the predicted image in the spatial domain. The subsnet has two parts: the first is a Multi-Layer-Perceptron (MLP) whose output is 'designed' to play the role of the frequency representation of the auto-encoder projections. It is crucial to observe that, in general, in the frequency domain, there is no immediate spatial correlation between neighboring Fourier coefficient values. Therefore, we prefer to process the input Fourier data using a relatively shallow architecture of an MLP over a potentially deeper architecture of convolutional layers. In all our experiments we use an MLP consisting of 4 layers. It is important to point out that the nonlinear activation function we use is the Parametric ReLu (PReLu), given by

$$\sigma_a(z) := \{ z \, , z > 0, az, z \leq 0,$$

where at each layer, the coefficient a is a parameter of the network. The reason is that the output of the MLP subnet is, by design, 'Fourier-type' coefficients of the auto-encoder representations which are real coefficients with potentially negative values. Recall that the 'standard' ReLu activation function only outputs non-negative values and thus not adequate for our case.

The dimensions of the output of the last layer of the MLP are set to the dimensions of the auto-encoder features. For example, if we use a set of 128 auto-encoder feature maps, each with a representation of an 8×8 pixels, then the output of the last MLP layer is then of dimension $128 \times 8 \times 8$, where each 8×8 matrix plays the role of a frequency representation of an auto-encoder's 8×8 feature map.

The second part of the subnet is a carefully chosen fixed inverse transform of 'Fourier-type' that is designed to convert the auto-encoder frequency representations to auto-encoder spatial pixel space representations. In our settings, we deal with datasets of images with real nonnegative pixel values. Therefore, we consider, by design, the output of the MLP subnet to be the coefficients of the real bivariate Discrete Cosine Transform (DCT) of the auto-encoder feature maps of the unknown image. To each such frequency representation we apply the fixed inverse DCT transform that uses fixed transform coefficients (e.g. no learning is applied to them). The inverse DCT transform takes as input real DCT coefficients and outputs real predicted auto-encoder feature map pixels. In the above example of 128 auto-encoder feature maps, each of dimension 8×8, the inverse DCT of dimension 8×8 will be applied 128 times to each feature map. For an input discrete image $f(k_1, k_2)$ of dimension 8×8,the forward DCT transform is

$$F(w_1, w_2) = \frac{1}{4} C(w_1) C(w_2) \sum_{k_1=0}^{7} \sum_{k_2=0}^{7} f(k_1, k_2) \cos \frac{(2k_1 + 1)w_1 \pi}{16} \cos \frac{(2k_2 + 1)w_2 \pi}{16},$$

where $0 \leq w_1, w_2 \leq 7$, and $C(v) = 1/\sqrt{2}$, for $v = 0$, $C(v) = 1$ for $v = 1, \ldots 7$. The inverse transform is given by

$$f(k_1, k_2) = \frac{1}{4} \sum_{w_1=0}^{7} \sum_{w_2=0}^{7} C(w_1) C(w_2) F(w_1, w_2) \cos \frac{(2k_1 + 1)w_1 \pi}{16} \cos \frac{(2k_2 + 1)w_2 \pi}{16}.$$

It is interesting to note that the famous and widely used JPEG image compression standard applies DCT transforms, separately, on each 8×8 pixel block of the compressed image [8].

2.3. The Auto-Encoder Feature Enhancement Subnet.

2.4. The Auto-Decoder Subnet. This subnet is initially a fixed component of the network, whose architecture is the decoder part of the auto-encoder/decoder network that was trained during the preprocessing stage described in Subsection 2.1. This subnet uses the weights that were computed during the preprocessing stage, but as we shall review in the result sections, we may allow this subnet to train in the last few epochs.

2.5. The Auto-Decoder Enhancement Subnet.

3. Experimental Results.

3.1. Overview of Datasets.

3.2. Results in the Supervised Setting. In Table 3.1 we see the results on the MNIST dataset ...

Table 3.1
Quantative comparison on the MNIST dataset

Model	MSE	MAE	SSIM
PRCGAN	0.0168	0.0399	0.8449
CPR	0.123	0.037	0.8756
PR-DAD	**0.0103**	**0.0367**	**0.8849**

Table 3.2
Quantative comparison on the EMNIST dataset

Model	MSE	MAE	SSIM
PRCGAN	0.02390	0.0601	0.8082
CPR	0.0144	0.0501	0.87
PR-DAD	**0.01207**	**0.04527**	**0.87579**

3.3. Results in the Semi-Supervised Setting.

3.4. Ablation Study.

1. Replace auto-encoder by dictionary/wavelets ?
2. Replace DCT with rFFT ?

4. Conclusions.

REFERENCES

[1] T. Bendory, R. Beinert and Y. Eldar, Fourier Phase Retrieval: Uniqueness and Algorithms, In: Compressed Sensing and its Applications (2017), 55-91.
[2] E. Cha, C. Lee, M. Jang and J Ye, DeepPhaseCut: deep relaxation in phase for unsupervised Fourier phase retrieval, https://arxiv.org/abs/2011.10475.
[3] L. Garwin and T. Lincoln, A century of nature: twenty-one discoveries that changed science and the world, University of Chicago Press, 2010.
[4] D. Sayre, Some implications of a theorem due to Shannon. Acta Crystallographica, 5 (1952), 843843.
[5] T. Uelwer, T. Hoffmann and S. Harmeling, Non-iterative phase retrieval with cascaded neural networks, In: Farka I., Masulli P., Otte S., Wermter S. (eds) Artificial Neural Networks and Machine Learning ICANN 2021. ICANN 2021. Lecture Notes in Computer Science, vol 12892. Springer, Cham.
[6] T. Uelwer, A. Oberstra and S. Harmeling, Phase retrieval using conditional generative adversarial networks, ICPR 2021.
[7] I. Waldspurger, A. dAspremont and S. Mallat, Phase recovery, maxcut and complex semidefinite programming, Mathematical Programming 149 (2015), 4781.
[8] G. Wallace, The JPEG still picture compression standard, in IEEE Transactions on Consumer Electronics, 38 (1992), xviii-xxxiv.