Report of Computational Imaging Project

Author: Xu Xie

Supervisor:Zhenguo Tan

1. Introduction

1.1 Modl

MoDL presents a systematic approach for developing deep architectures for solving inverse problems by combining the strengths of model-based reconstruction and deep learning. The approach involves a CNN-based regularization prior within a variational framework, leading to an architecture that interleaves CNN blocks for image set information capture and data consistency blocks for measurement consistency. This method enables the use of complex forward models and the incorporation of additional image priors, demonstrating enhanced performance and efficiency, particularly in settings with limited training data.

1.2 Varnet

This paper introduces a variational network framework for MRI image reconstruction that integrates a convolutional neural network (CNN)-based regularization prior into a model-based image reconstruction process. By explicitly accounting for the forward model, the proposed method requires a smaller network with fewer parameters than direct inversion approaches, reducing the need for extensive training data and training time. The network is trained end-to-end with weight sharing across iterations, which allows the CNN weights to be customized to the forward model, leading to improved performance. Experiments demonstrate benefits such as lower demand for training data, reduced risk of overfitting, and efficient memory usage.

1.3 SSDU

This study introduces a self-supervised learning method for MRI reconstruction that doesn't rely on fully sampled reference data. By partitioning available measurements into disjoint sets for data consistency and training loss, the method trains a physics-guided neural network to enhance MRI reconstruction quality. This approach outperforms traditional compressed-sensing and parallel imaging techniques, offering a viable solution for scenarios where fully sampled data are unavailable, thus paving the way for efficient and accurate MRI reconstructions without the need for extensive ground-truth datasets.

2. Tasks

2.1 Convert Fastmri Dataset

First the task is to use the Fastmri dataset to make a new dataset. Initially, data is selected based on certain criteria, such as the number of coils, for example 16. Following the selection, the next step involves computing the Csm Org Mask for train and test. Once computed, the data undergoes a transformation phase where it is resized, the data type is converted, and normalization is applied to ensure data uniformity. After these adjustments, the processed data is added back to the selected data, enhancing the original dataset with the computed and normalized values. The final step in this process is to combine the data along a slice, which implies integrating the data across a particular dimension or axis. This process ensures that the data is well-structured and primed for further analysis or application. Figure 1 is the result.(I use possion undersample method and the acceleration factor is 6)

```
Keys: <KeysViewHDF5 ['trnCsm', 'trnMask', 'trnOrg', 'tstCsm', 'tstMask', 'tstOrg']>
Dataset: trnCsm
Data type: complex64
Shape: (400, 16, 768, 396)
Dataset: trnMask
Data type: int8
Shape: (400, 768, 396)
Dataset: trnOrg
Data type: complex64
Shape: (400, 768, 396)
Dataset: tstCsm
Data type: complex64
Shape: (96, 16, 768, 396)
Dataset: tstMask
Data type: int8
Shape: (96, 768, 396)
Dataset: tstOrg
Data type: complex64
Shape: (96, 768, 396)
```

Fig. 1 Details of the new dataset

2.2 Modify Varnet's code

Figure 2 shows the architecture of Varnet, it doesn't share weight paramater like Modl, instead in this project I used a list of Unet to implement. As we can see the code in figure 3 and figure 4, I use nn.Modulelist to define the structure and use it in the forward function. This part is actually a nonlinear denoising part.

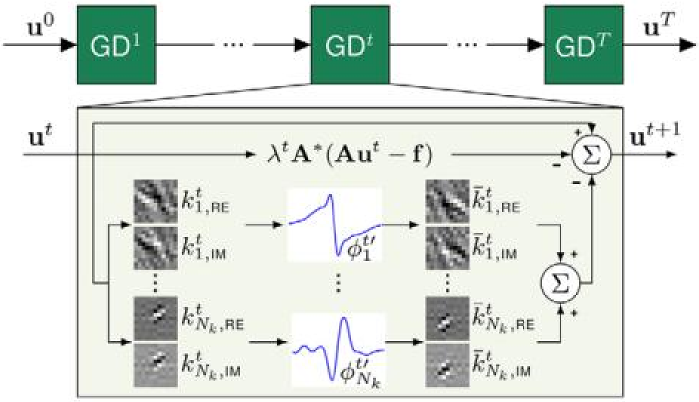


Fig. 2 Yaman B, Hosseini SAH, Moeller S, Ellermann J, Uğurbil K, Akçakaya M. Self-supervised learning of physics-guided reconstruction neural networks without fully sampled reference data. Magn Reson Med. 2020;84:3172–3191. https://doi.org/10.1002/mrm.28378

```python
class VarNet(nn.Module):
    def __init__(self, n_layers, k_iters) -> None:

        super().__init__()

        self.n_cascades = k_iters
        self.dc = data_consistency()
        self.dw = unet.Unet(2, 2, num_pool_layers=n_layers)

    def forward(self, x0, coil, mask):

        x0 = r2c(x0, axis=1)
        x = x0.clone()

        for c in range(self.n_cascades):

            x = self.dc(x, x0, coil, mask)
            x = x - r2c(self.dw(c2r(x, axis=1)), axis=1)

        return c2r(x, axis=1)
```

Fig. 3 Original code of Varnet

```python
class VarNet(nn.Module):
    def __init__(self, n_layers, k_iters) -> None:
        super().__init__()

        self.n_cascades = k_iters

        self.dc = data_consistency()

        self.unets = nn.ModuleList([unet.Unet(2, 2, num_pool_layers=n_layers) for _ in range(self.n_cascades)])

    def forward(self, x0, coil, mask):
        x0 = r2c(x0, axis=1)
        xk = x0.clone()

        for i in range(self.n_cascades):
            xk = self.dc(xk, x0, coil, mask)

            xk = xk - r2c(self.unets[i](c2r(xk, axis=1)), axis=1)

        return c2r(xk, axis=1)
```

Fig. 4 Modified code of aVarnet

2.3 Transport SSDU to MODL

2.3.1 Introduction

The original code was based on Tensorflow and when I ran it, I encountered so many problems, first is the environment is outdated, I need to create a new conda environment with python = 3.6, tensorflow = 1.13, then I need to make a new dataset that fits the requirements, and after that finally I can run the code, but it was really slow. So I use another method, transport SSDU to Pytorch_Modl.

2.3.2 Methods

1. Implement neural network architecture(Resnet) in ssdu.py.

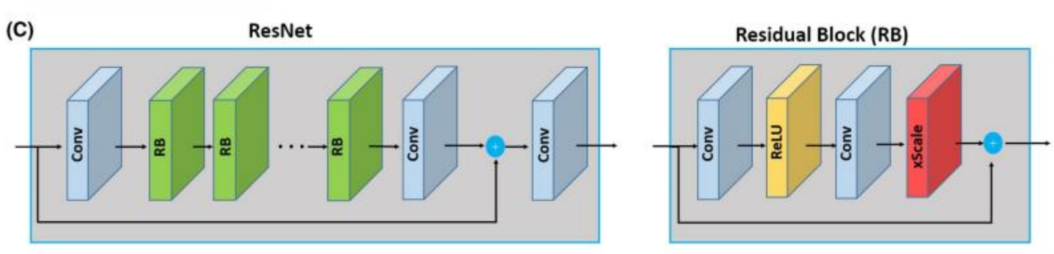


Fig. 5 Structure of Resnet in the paper SSDU

2. Implement a new dataloader in ssdu_dataset.py.

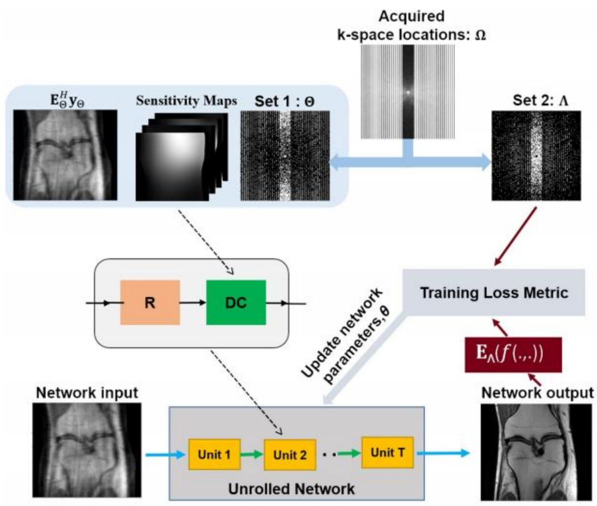


Fig. 6 The self-supervised learning scheme to train physics-guided deep learning without fully sampled data.

As we can see in figure 6, what we need to load about the data becomes more, included trn_mask and loss_mask now, and during the loss calculation we need the undersample kspace.

3. Implement L1L2 loss function in the paper in utils.py

A normalized ℓ_1-ℓ_2 loss, defined as

$$\mathcal{L}(\mathbf{u},\mathbf{v}) = \frac{\|\mathbf{u}-\mathbf{v}\|_2}{\|\mathbf{u}\|_2} + \frac{\|\mathbf{u}-\mathbf{v}\|_1}{\|\mathbf{u}\|_1}, \qquad (11)$$

Fig. 7 L1L2 function in the paper SSDU

4. Modify code in train.py
5. Apply normalization on kspace

3. Result

3.1 Modl

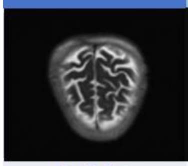
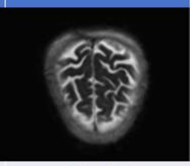
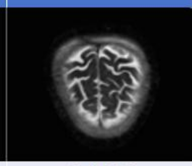
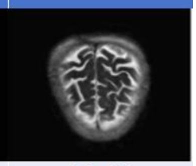
GT	layer:05 epoch:50 k_iteration:10	layer:05 epoch:50 k_iteration:5	layer:05 epoch:50 k_iteration:1	layer:10 epoch:50 k_iteration:1	layer:10 epoch:80 k_iteration:1
					
PSNR	29.40	30.21	31.55	32.01	32.87
SSIM	0.84	0.87	0.91	0.91	0.95

Fig. 8 Result of Modl

In Figure 8, we observe a series of experiments designed to understand the effects of various parameters—such as the number of iterations (k_iteration), the number of layers, and epochs—on the performance of the MoDL architecture for MRI image reconstruction.

The baseline, depicted by the first image labeled "GT" (Ground Truth), is followed by reconstructions under varying parameters. The first variation involves a decrease in k_iteration from 10 to 5 at a constant epoch setting of 50 and layer configuration of 5. This change leads to an improvement in image quality, reflected by an increase in PSNR from 29.40 to 30.21, and an uplift in SSIM from 0.84 to 0.87. The improvement suggests that a higher number of iterations does not necessarily contribute to better reconstruction in this scenario, and a reduced iteration count might actually enhance the performance.

Further adjustments include modifying the number of layers from 5 to 10, keeping both epochs at 50 and k_iteration at 1. This alteration leads to a slight increase in PSNR to 32.01 and maintains the SSIM at 0.91, indicating that increasing the network's depth has a positive but not overly pronounced effect on the reconstruction quality.

The final configuration explores the effect of prolonging the training process by increasing epochs from 50 to 80 while maintaining the number of layers at 10 and k_iteration at 1. The result is a notable improvement, with the highest PSNR of 32.87 and SSIM of 0.95 across all tested configurations. This outcome confirms the hypothesis that training for a more extended period allows the deep learning model to better learn from the data, enhancing its ability to reconstruct higher fidelity images.

Collectively, these experiments suggest that while the MoDL architecture benefits from a tailored approach to parameter tuning, certain parameters such as training duration (epochs) have a more substantial impact on performance than others. Additionally, this information could guide the design of more efficient training regimes and architectures, focusing on the most impactful elements of the model.

3.2 Varnet

GT	layer:05 epoch:50 k_iteration:10	layer:05 epoch:50 k_iteration:5	layer:05 epoch:50 k_iteration:1
			
PSNR	19.32	31.73	38.72
SSIM	0.78	0.92	0.98

Fig. 9 Result of Varnet

In the exploration of VarNet's performance, Figure 9 demonstrates how the model responds to variations in the k_iteration parameter. The first column presents the "GT" (Ground Truth) as the benchmark for comparison. Subsequent columns show the reconstruction results with the number of layers and epochs held constant at 5 and 50, respectively, while the k_iteration parameter is systematically reduced from 10 to 5, and then to 1.

At the highest k_iteration of 10, the PSNR is at a lower 19.32, and the SSIM at 0.78, indicating a significant departure from the ideal reconstruction seen in the ground truth. When the k_iteration is halved to 5, there's a marked improvement in the quality of image reconstruction, as evidenced by the PSNR jumping to 31.73 and SSIM to 0.92. This trend continues as we further reduce k_iteration to 1, which yields the best results amongst the tested configurations, with a PSNR of 38.72 and an SSIM of 0.98, closely approaching the fidelity of the ground truth.

This set of results reveals an intriguing inverse relationship between the number of iterations and the quality of image reconstruction for VarNet, which is consistent with the results observed for the MoDL framework. It suggests that beyond a certain point, more iterations can lead to overfitting or other inefficiencies that degrade the quality of the reconstructed image.

The comparison between VarNet and MoDL could imply that both architectures share a common characteristic in that a more concise iterative process can be more beneficial than longer, potentially redundant iterative sequences. This could be due to various factors including the ability of the model to rapidly converge to a good solution or due to inherent properties of the model architecture that make longer iteration sequences less effective.

By reducing k_iteration, VarNet seems to become more efficient, possibly avoiding the redundancy of processing without substantial information gain, leading to a faster and more effective convergence. This result is particularly important because it indicates that the optimization landscape of the model is such that it can achieve high performance without the need for extensive iterative refinement.

3.3 SSDU

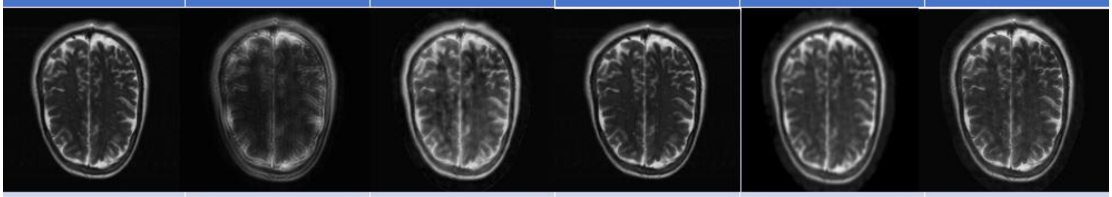
GT	lr=0.001 epoch:50 k_iteration:1	lr=0.001 epoch:50 k_iteration:5	lr=0.001 epoch:50 k_iteration:10	lr=0.0001 epoch:50 k_iteration:10	lr=0.0005 epoch:50 k_iteration:10
					
PSNR	21.10	26.04	31.41	13.36	27.56
SSIM	0.71	0.86	0.91	0.35	0.87

Fig. 10 Result of SSDU

From the first image with a learning rate of 0.001 and k_iteration set to 1, there's a gradual improvement in image quality as k_iteration increases to 5 and then 10, with PSNR values rising from 21.10 to 26.04, and then to 31.41, and SSIM increasing correspondingly from 0.71 to 0.86, and then to 0.91. This indicates that, for a consistent learning rate, increasing the number of iterations can enhance the reconstruction, aligning with the pattern observed in the VarNet framework. However, when the learning rate is decreased tenfold to 0.0001 with k_iteration fixed at 10, there's a significant drop in the reconstruction quality, with the PSNR plummeting to 13.36 and SSIM to 0.35. This suggests that too small a learning rate may impede the network's ability to adjust its weights effectively during training, leading to a suboptimal solution.

Conversely, with a slightly higher learning rate of 0.0005 and k_iteration at 10, there's a recovery in image quality, as shown by the improvement in PSNR to 27.56 and SSIM to 0.87. This suggests that there is an optimal range for the learning rate that balances convergence speed and training stability, ensuring the network effectively captures the underlying data distribution without overshooting or getting stuck in local minima.

These results imply that in SSDU, the parameter k_iteration is influential in determining image quality, corroborating the hypothesis that more iterations, to an extent, are beneficial. However, this effect is highly dependent on an appropriate learning rate, indicating a complex interaction between the two parameters. The findings underscore the necessity of fine-tuning both the learning rate and the number of iterations to achieve the best possible image reconstruction quality in self-supervised deep learning models.

4. Conclusion

1.Iteration Parameter (k_iteration) Impact:

In the context of Self-Supervised Learning of Physics-guided Reconstruction Neural Networks (SSDU), increasing the number of iterations (k_iteration) tends to improve the reconstruction quality. This is likely because additional iterations allow the network to refine its understanding and application of the physical model and data consistency constraints, thus progressively enhancing the image reconstruction from undersampled MRI data. The iterative nature allows the model to better learn and adapt to the nuances of MRI data, making the most of the available undersampled datasets.

However, for MoDL (Model-Based Deep Learning Architecture for Inverse Problems) and VarNet (Learning a Variational Network for Reconstruction of Accelerated MRI Data), the benefit of increasing iterations is not as clear-cut. These architectures integrate deep learning models with iterative reconstruction processes. Beyond a certain point, additional iterations do not necessarily translate to significant improvements in reconstruction quality for these models. This plateau effect might be due to the models reaching an equilibrium where the benefits of additional iterations are offset by the risk of overfitting or by computational inefficiencies.

2.Epoch Parameter Impact:

Both MoDL and VarNet benefit from a higher number of training epochs. Training for more epochs allows these networks to better learn the complex mappings from undersampled to fully sampled MRI spaces, due to their deep learning components' ability to adaptively refine the reconstruction process over time. This prolonged training leads to more accurate image reconstructions, capturing finer details and reducing artifacts, as the models iteratively adjust their parameters towards optimal values for the given reconstruction task.

3.Comparison between VarNet and MoDL:

VarNet's performance exceeding that of MoDL can be attributed to the specific architecture and learning strategy employed by VarNet. VarNet uses a variational network approach, which might offer a more robust framework for integrating data consistency and learning-based priors in the reconstruction process. This architecture potentially allows VarNet to more effectively leverage both the physics of MRI acquisition and the patterns learned from training data, leading to superior reconstruction results compared to MoDL, which also seeks to blend model-based and deep learning approaches but might do so in a less optimized manner.

4.Comparative Analysis of SSDU, MoDL, and VarNet:

MoDL and VarNet exhibit better performance than SSDU, especially when SSDU employs a uniform selection strategy for data partitioning. This discrepancy in performance can be understood considering the inherent differences in how these models approach the reconstruction problem. While SSDU focuses on a self-supervised learning paradigm without requiring fully sampled data—relying on splitting available undersampled data for training and validation—MoDL and VarNet incorporate more sophisticated blending of deep learning with explicit modeling of the MRI acquisition process. This allows MoDL and VarNet to more accurately reconstruct images from undersampled data by exploiting learned priors and data consistency in a more effective manner. The choice of data partitioning strategy in SSDU, particularly the uniform selection, might not fully capture the complex spatial-frequency relationships in MRI data, leading to inferior reconstruction quality compared to the more integrated and adaptive approaches of MoDL and VarNet.

5. Reference

1. Yaman B, Hosseini SAH, Moeller S, Ellermann J, Uğurbil K, Akçakaya M. Self-supervised learning of physics-guided reconstruction neural networks without fully sampled reference data. Magn Reson Med. 2020;84:3172–3191. https://doi.org/10.1002/mrm.28378

2. H. K. Aggarwal, M. P. Mani and M. Jacob, "MoDL: Model-Based Deep Learning Architecture for Inverse Problems," in IEEE Transactions on Medical Imaging, vol. 38, no. 2, pp. 394-405, Feb. 2019, doi: 10.1109/TMI.2018.2865356.keywords: {Image reconstruction;Training data;Training;Optimization;Imaging;Numerical models;Machine learning;Deep learning;parallel imaging;convolutional neural network},

3. Hammernik K, Klatzer T, Kobler E, et al. Learning a variational network for reconstruction of accelerated MRI data[J]. Magnetic resonance in medicine, 2018, 79(6): 3055-3071.