Convolutional Neural Network for MRI Reconstruction

Problem

Avrajit Ghosh, Siddhant Gautam, Shijun Liang

Computational Mathematics, Science and Engineering,
Michigan State University

December 9, 2020

Main goal: Using a *Convolutional Neural Network* based algorithm for solving ill-posed MRI reconstruction problem

- In an inverse problem, the measurement $Y$ can be written as

$$Y = Ax + \eta$$

where $x$ is the image, $A$ denotes the forward operator and $\eta$ is the noise.

- **Inverse Reconstruction problem:**

$$\hat{x} = \arg\min_x \|Ax - y\|^2_2 + R(x)$$

where $R(x)$ is a regularizer that assumes a prior on $x$.

- If $R(x) = 0$, the solution is given by $\hat{x} = A^\dagger y$ which minimises the $l2$-norm of the difference without assuming any prior on $x$. 

Avrajit Ghosh, Siddhant Gautam, Shijun Liang
The forward or sensing operator $A$ for different cases is given by:

- $A = I$ and $R(x) = \|x\|_1$ for image denoising.
- $A$ is subsampled Fourier transform, $R(x) = \|x\|_{T_v}$ for MRI reconstruction problem.
- $A$ is filtering for deblurring.

The direct inverse solution of the MRI reconstruction is

$$f = H^\dagger Wg$$

where $g = Hf$ and $g$ denotes the measurement, $H$ is the Fourier transform matrix, and $W$ is the inverse sparsifying transform.
Iterative Inversion

Inverse Reconstruction Problem

\[ \hat{a} = \arg \min_a \| y - H W a \|_2^2 + \lambda \| a \|_1 \]

where \( H \) is the forward model (FT), \( y \) is the measured k-space data, \( a \) is transform coefficient vector and \( W \) is the sparsifying transform.

**Desired reconstruction:** \( \hat{x} = W \hat{a} \)

Iterative Shrinkage Thresholding Algorithm (ISTA)

\[ a^{k+1} = S_\lambda^L \left( \frac{1}{L} W^* H^* y + (I - \frac{1}{L} W^* H^* H W) a^k \right) \]

where \( S_\lambda^L \) is the soft-thresholding operator\(^2\) and \( W^*, H^* \) are adjoint operators of \( W \) and \( H \) respectively.

---

Why use a CNN?

Figure: Iterative Unrolling of optimization problem modelled as a CNN

\[ X_0 \rightarrow W \rightarrow I - \frac{1}{L} W^H H^T W \rightarrow + \rightarrow \ldots \rightarrow I - \frac{1}{L} W^H H^T W \rightarrow + \rightarrow W \rightarrow X_* \]

\[ X_0 \rightarrow w_1 \rightarrow + \rightarrow w_2 \rightarrow + \rightarrow \ldots \rightarrow w_{N-1} \rightarrow + \rightarrow w_N \rightarrow + \rightarrow X_* \]

Structure of U-Net\textsuperscript{4}

Components of U-Net:

1. **Double Convolution**: $2 \times (\text{Convolution} + \text{Batch-Normalization} + \text{ReLU})$

2. **Downscaling**: Maxpooling by factor 2

3. **Upscaling**: Upsampling with bilinear interpolation

4. **Skip connection**: The input image is added to the output of the network. So that the network now learns the difference between the true and the aliased image.

5. **Concatenation**: At each subsampled layer, the output of downsample+$\text{Conv2d}$ is concatenated to the upsampled image.

Advantage of U-Net architecture

- **Multilevel decomposition**: Extracts features from each scale
- **Multichannel filtering**: Extracts multiple feature maps from each scale
Implementation Details - CNN

Masks Used:
1. Low Frequency (LF) - Vertical
2. Variable Density Random Sampling (VDRS)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Rate</td>
<td>$5 \times 10^{-3}$</td>
</tr>
<tr>
<td>Epochs</td>
<td>40</td>
</tr>
<tr>
<td>Batch-size</td>
<td>25</td>
</tr>
<tr>
<td>Training/Validation Ratio</td>
<td>645/50</td>
</tr>
<tr>
<td>Training/Testing Ratio</td>
<td>645/5</td>
</tr>
<tr>
<td>CNN Architecture</td>
<td>U-Net(^5)</td>
</tr>
</tbody>
</table>

---

### Implementation Details - Training Data Generation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>OCMR$^6$</td>
</tr>
<tr>
<td>Size of Image</td>
<td>$512 \times 208$</td>
</tr>
<tr>
<td>No. of Total Samples</td>
<td>700</td>
</tr>
<tr>
<td>Undersampling Factors</td>
<td>8x, 16x, 24x</td>
</tr>
<tr>
<td>No. of Coils</td>
<td>18</td>
</tr>
<tr>
<td>Sensitivity Maps</td>
<td>ESPIRiT$^7$</td>
</tr>
<tr>
<td>Decoder</td>
<td>CS-Sigpy$^8$</td>
</tr>
</tbody>
</table>

---


Training Data
Reconstruction Results: Low Frequency @ 8x

Mask: LF, Undersampling factor: 8x, #lines=64

- Applied Mask
- Clean Image
- Aliased Image, PSNR=39.72 dB
- Reconstructed Image, PSNR=29.56 dB
Reconstruction Results: Random Sampling @ 8x

Mask: VDRS, Undersampling factor: 8x, #lines=64

- Applied Mask
- Clean Image
- Aliased Image, PSNR=24.29 dB
- Reconstructed Image, PSNR=24.85 dB
Reconstruction Results: Low Frequency @ 16x

Mask: LF, Undersampling factor: 16x, #lines=32

Applied Mask  Clean Image  Aliased Image, PSNR=35.04 dB  Reconstructed Image, PSNR=30.54 dB
Reconstruction Results: Random Sampling @ 16x

Mask: VDRS, Undersampling factor: 16x, #lines=32

- Applied Mask
- Clean Image
- Aliased Image, PSNR=25.18 dB
- Reconstructed Image, PSNR=29.14 dB
Reconstruction Results: Low Frequency @ 24x

Mask: LF, Undersampling factor: 24x, #lines=21

- Applied Mask
- Clean Image
- Aliased Image, PSNR=28.20 dB
- Reconstructed Image, PSNR=26.74 dB
Reconstruction Results: Random Sampling @ 24x

Mask: VDRS, Undersampling factor: 24x, #lines=21

- Applied Mask
- Clean Image, PSNR=24.92 dB
- Aliased Image, PSNR=27.30 dB
- Reconstructed Image, PSNR=27.30 dB
### Table: PSNR (dB) of the aliased and reconstructed image from the U-Net for different cases

<table>
<thead>
<tr>
<th>Mask Selected</th>
<th>Image</th>
<th>Undersampling Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>8x</td>
</tr>
<tr>
<td>Low Frequency (Vertical)</td>
<td>Aliased</td>
<td>39.72</td>
</tr>
<tr>
<td></td>
<td>Reconstructed</td>
<td>29.56</td>
</tr>
<tr>
<td>Random Sampling</td>
<td>Aliased</td>
<td>24.29</td>
</tr>
<tr>
<td></td>
<td>Reconstructed</td>
<td>24.85</td>
</tr>
</tbody>
</table>

- Better accuracy than the aliased image in case of variable density sampling.
- No improvement in accuracy for 8x undersampling factor.
- Significant improvement for highly undersampled MRI (16x, 24x).
### Reconstruction Accuracy: PSNR

<table>
<thead>
<tr>
<th>Mask Selected</th>
<th>Image</th>
<th>Undersampling Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>8x</td>
</tr>
<tr>
<td><strong>Low Frequency (Vertical)</strong></td>
<td>Aliased</td>
<td>39.72</td>
</tr>
<tr>
<td></td>
<td>Reconstructed</td>
<td>29.56</td>
</tr>
<tr>
<td><strong>Random Sampling</strong></td>
<td>Aliased</td>
<td>24.29</td>
</tr>
<tr>
<td></td>
<td>Reconstructed</td>
<td>24.85</td>
</tr>
</tbody>
</table>

Table: PSNR (dB) of the aliased and reconstructed image from the U-Net for different cases

- Better accuracy than the aliased image in case of variable density sampling
- No improvement in accuracy for 8x undersampling factor
- Significant improvement for highly undersampled MRI (16x, 24x)

Avrajit Ghosh, Siddhant Gautam, Shijun Lia

CNN for MRI Reconstruction

December 9, 2020 18 / 20
Conclusion

- Presented a CNN based MRI-reconstruction with U-Net.
- U-Net extracts features of image at multiple scales. Hence it is ideal for MRI reconstruction.
- For the undersampled MRI dataset, U-Net can extract the fine detail in the image.
- For highly undersampled MRI with reduced time scan, better reconstruction with U-Net.
Fin.

Questions?