Designing Transformer networks for sparse recovery of sequential data using deep unfolding

Brent De Weerdt, Yonina C. Eldar, Nikos Deligiannis
Use sparse priors to recover signals
from compressed measurements

▶ Compressed measurement: \( x_t = A s_t + \eta_t, \quad A \in \mathbb{R}^{m \times n} (m \ll n), t = 1, \ldots, T \)

▶ Assume a sparse representation \( h_t \) in some dictionary: \( s_t = D h_t \)

▶ Assume some correlation over time: \( C(h_t, h_{t-1}) \)

▶ Solve \( \min_{h_1, \ldots, h_T} \sum_t \left( \frac{1}{2} \| x_t - A D h_t \|_2^2 + \lambda_1 \| h_t \|_1 + \lambda_2 C(h_t, h_{t-1}) \right) \)

▶ The final reconstructed signal is \( s_t^* = D h_t^* \)
Deep unfolding Intro

- Deep unfolding designs neural network models by:
  1. Unrolling an iterative algorithm
  2. Mapping the algorithm’s (sub)steps to neural network layers
  3. Training the resulting model on data

- Deep unfolding models have lower reconstruction errors and less iterations than the original iterative algorithm
LISTA

★ Optimization problem: \( \min_h \frac{1}{2} \|x - ADh\|_2^2 + \lambda \|h\|_1 \)

★ Iterative Soft Thresholding Algorithm (ISTA):

\[ h^{(k+1)} = \phi_{\lambda/c} \left( h^{(k)} + \frac{1}{c} D^T A^T (x - ADh^{(k)}) \right) \]

★ Deep unfolding model: Learned ISTA (LISTA)

\[ h^{(k+1)} = \phi_{\lambda/c} \left( Sh^{(k)} + Wx \right) \]

Deep unfolding RNNs

▶ SISTA-RNN:
\[
\sum_t \left( \frac{1}{2} \| x_t - ADh_t \|^2_2 + \lambda_1 \| h_t \|_1 + \frac{\lambda_2}{2} \| Dh_t - FDh_{t-1} \|^2_2 \right)
\]

▶ \(\ell_1-\ell_1\)-RNN
\[
\sum_t \left( \frac{1}{2} \| x_t - ADh_t \|^2_2 + \lambda_1 \| h_t \|_1 + \lambda_2 \| h_t - Gh_{t-1} \|_1 \right)
\]

▶ Reweighted-RNN
\[
\sum_t \left( \frac{1}{2} \| x_t - ADZh_t \|^2_2 + \lambda_1 \| g \circ Zh_t \|_1 + \lambda_2 \| g \circ (Zh_t - Gh_{t-1}) \|_1 \right)
\]

Deep unfolding for a vanilla Transformer

Optimization problem designed to unfold into a Transformer architecture:

\[
Y = [y_1, \ldots, y_N], \quad \psi(u) = \begin{cases} 
+\infty & \text{if } u < 0 \\
0 & \text{if } u \geq 0 
\end{cases}
\]

\[
\min_Y \sum_{i,j} - \exp \left( -\frac{1}{2} \| W_a y_i - W_a y_j \|_2^2 \right) + \frac{1}{2} \| W_a Y \|_F^2 + \frac{1}{2} \text{Tr} \left( Y^T W_b Y \right) + \frac{1}{2} \| Y \|_F^2 + \psi(Y)
\]

- Design minimization steps for each part separately
- Alternating between these two steps minimizes the total optimization problem:

\[
Y^{(k+1)} = \text{ReLU} \left( W_b Y^{(k)} \softmax_\beta \left( Y^{(k)^T} W_a Y^{(k)} \right) \right)
\]

Our deep unfolding Transformer for sparse recovery

- Incorporate priors for sequential sparse recovery
  - Model correlations across the whole video
  - Retain the sparsity constraint and data fidelity term

\[
\min_{h_1, \ldots, h_T} \sum_t \lambda_2 \left( \sum_{\tau} - \exp \left( -\frac{1}{2} \| D h_t - D h_\tau \|_2^2 \right) + \| D h_t \|_2^2 \right) + \frac{1}{2} \| x_t - A D h_t \|_2^2 + \lambda_1 \| h_t \|_1
\]

- temporal correlations
- data fidelity and sparsity
The optimization algorithm

\[
\min_{\mathbf{h}_1, \ldots, \mathbf{h}_T} \sum_t \lambda_2 \left( \sum_{\tau} - \exp \left( -\frac{1}{2} \| \mathbf{D}\mathbf{h}_t - \mathbf{D}\mathbf{h}_{\tau} \|_2^2 \right) + \| \mathbf{D}\mathbf{h}_t \|_2^2 \right) + \frac{1}{2} \| \mathbf{x}_t - \mathbf{A}\mathbf{D}\mathbf{h}_t \|_2^2 + \lambda_1 \| \mathbf{h}_t \|_1
\]

- **First part: softmax self-attention**

\[
\mathbf{H}^{(k+\frac{1}{2})} = \lambda_2 \mathbf{H}^{(k)} \text{softmax}_\beta \left( \mathbf{H}^{(k)T} \mathbf{D}^T \mathbf{D}\mathbf{H}^{(k)} \right), \quad \mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \ldots & \mathbf{h}_T \end{bmatrix}
\]

- **Second part: parallel ISTA operations**

\[
\mathbf{h}_t^{(k+1)} = \phi \frac{\lambda_1}{c} \left( \mathbf{h}_t^{(k+\frac{1}{2})} + \frac{1}{c} \mathbf{D}^T \mathbf{A}^T \left( \mathbf{x}_t - \mathbf{A}\mathbf{D}\mathbf{h}_t^{(k+\frac{1}{2})} \right) \right) \forall t
\]
DUST: Deep Unfolding Sparse Transformer

- **Start from:**
  \[ x_t = A s_t, \ h_t^{(0)} = 0 \quad \forall t \]

- **For** \( K \) times:
  \[
  H^{(k+\frac{1}{2})} = \lambda_2 H^{(k)} \text{softmax} \left( H^{(k)^T} D^T D H^{(k)} \right) \\
  h_t^{(k+1)} = \phi \lambda_1/c \left( U h_t^{(k+\frac{1}{2})} + V x_t \right) \quad \forall t
  \]

- **Final reconstruction:**
  \[ s_t^* = D h_t^{(K)} \]
Experimental results

Average video reconstruction quality (PSNR) on the Avenue, UCSD and ShanghaiTech dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Avenue</th>
<th>UCSD</th>
<th>ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>SISTA-RNN</td>
<td>35.73</td>
<td>34.13</td>
<td>34.90</td>
</tr>
<tr>
<td>$\ell_1$-$\ell_1$-RNN</td>
<td>36.51</td>
<td>34.34</td>
<td>35.56</td>
</tr>
<tr>
<td>Reweighted-RNN</td>
<td>36.94</td>
<td>35.22</td>
<td><strong>36.03</strong></td>
</tr>
<tr>
<td>ViT</td>
<td>36.04</td>
<td>34.79</td>
<td>35.91</td>
</tr>
<tr>
<td>Unfolded Transformer</td>
<td>34.36</td>
<td>32.94</td>
<td>34.25</td>
</tr>
<tr>
<td>DUST (proposed)</td>
<td><strong>37.61</strong></td>
<td><strong>35.98</strong></td>
<td>35.94</td>
</tr>
</tbody>
</table>

Average video reconstruction quality (PSNR) on the Avenue dataset for different compression rates.

<table>
<thead>
<tr>
<th>Method</th>
<th>50%</th>
<th>40%</th>
<th>30%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>SISTA-RNN</td>
<td>41.89</td>
<td>39.92</td>
<td>37.99</td>
<td>32.01</td>
</tr>
<tr>
<td>$\ell_1$-$\ell_1$-RNN</td>
<td>42.86</td>
<td>40.90</td>
<td>38.89</td>
<td>32.98</td>
</tr>
<tr>
<td>Reweighted-RNN</td>
<td>43.23</td>
<td>41.16</td>
<td>39.12</td>
<td>33.88</td>
</tr>
<tr>
<td>ViT</td>
<td>39.53</td>
<td>38.28</td>
<td>37.12</td>
<td>33.85</td>
</tr>
<tr>
<td>Unfold. Transf.</td>
<td>39.66</td>
<td>37.93</td>
<td>36.07</td>
<td>32.11</td>
</tr>
<tr>
<td>DUST (proposed)</td>
<td><strong>43.32</strong></td>
<td><strong>41.47</strong></td>
<td><strong>39.67</strong></td>
<td><strong>34.71</strong></td>
</tr>
</tbody>
</table>
Model size and computation complexity

- DUST and the other Transformer models can process videos twice as fast compared to the deep unfolding RNNs
  - More parallel computation
  - Less complex calculations
- DUST has 1.4M parameters, significantly smaller than the next best performing model, reweighted-RNN (2.5M parameters)
Conclusion

- We designed a deep unfolding Transformer architecture for sparse recovery of sequential data.
- This model has improved reconstruction quality and lower computational cost compared to deep unfolding RNNs.
- Future work: different attention mechanisms, longer sequences, denoising, super-resolution.