NEU KRON: Constant-Size Lossy Compression of Sparse Reorderable Matrices and Tensors

Taehyung Kwon*
Jihoon Ko*
Jinhong Jung
Kijung Shin
Sparse matrices from Web applications

<table>
<thead>
<tr>
<th>Friendship in Social Media</th>
<th>Counting Clicks on Ads By Search Engine</th>
<th>Publication Records from Academic Databases</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Social Media Network" /></td>
<td><img src="image2" alt="Ad Clicks" /></td>
<td><img src="image3" alt="Publication Records" /></td>
</tr>
</tbody>
</table>

**Introduction**

**Proposed Method**

**Experiments**

**Conclusion**

<table>
<thead>
<tr>
<th>Publication 1</th>
<th>Publication 2</th>
<th>Publication 3</th>
<th>Publication 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image4" alt="Publication 1" /></td>
<td><img src="image5" alt="Publication 2" /></td>
<td><img src="image6" alt="Publication 3" /></td>
<td><img src="image7" alt="Publication 4" /></td>
</tr>
</tbody>
</table>

Authors:

- Publication 1: B, D
- Publication 2: C, D
- Publication 3: A, B, C, D
- Publication 4: D

**Publication Records**

<table>
<thead>
<tr>
<th>Publication 1</th>
<th>Publication 2</th>
<th>Publication 3</th>
<th>Publication 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image8" alt="Publication 1 Records" /></td>
<td><img src="image9" alt="Publication 2 Records" /></td>
<td><img src="image10" alt="Publication 3 Records" /></td>
<td><img src="image11" alt="Publication 4 Records" /></td>
</tr>
</tbody>
</table>

**Authors**

- Publication 1: B, D
- Publication 2: C, D
- Publication 3: A, B, C, D
- Publication 4: D

**Counting Clicks on Ads**

- Publisher: Search Engine

**Friendship in Social Media**

- Publisher: Friendships

Real-world sparse matrices are **large-scale**

- Real-world sparse matrices often containing billions of rows or columns
  - requires heavy memory or network I/O usage
  - compressing these large sparse matrices is important!
Our goal: **constant-size** compression

\[ A \in \mathbb{R}^{N \times M} \] / a constant \( k = O(1) \)

- **Given:** a sparse and reorderable matrix \( A \) whose size is at most \( k \)
- **Find:** a model \( \Theta \) whose size is at most \( k \)
- **To minimize:** the approximation error \( \| A - \tilde{A}_\Theta \|_F^2 \)
Overview of NEUKRON

• **Recurrent Neural Network:** having a constant number of parameters but also expressive power

• **Reordering:** extract and exploit structural patterns for better compression

### Diagram

- **Real-valued Outputs**
  - LSTM Cell Constant size
  - LSTM Cell Constant size
  - LSTM Cell Constant size

- **Target Entry**
  - (top, left)
  - (bottom, right)
  - (top, right)

- **Encoded Sequence of position**

- **Sharing Parameters**

- **Experiments**

- **Conclusion**
Model of **NeuKron**

- **Encode** the position in a sequence by recursively dividing the input matrix.
Model of **NEUKRON**

- Feed the sequence to LSTM to compute seed matrices
- Approximate the entry by multiplying the outputs of the LSTM cells

![Diagram showing Kronecker Graphs and NEUKRON](image)
Order optimization

• Many real-world sparse matrices are reorderable
  ⇒ Exploit structural patterns for compression!
Order optimization

• Step 1. Find similar pairs of slices using Min-Hashing

• Step 2. Exchange slices with the neighboring slices when loss decreases
Overall training procedure

Iterative update until convergence

Model optimization

Order optimization

Real-valued Outputs

Encoded Sequence of position

LSTM Cell Constant size

Sharing Parameters

(top, left) (bottom, right) (top, right)
Experimental settings

• 10 real-world datasets: 6 sparse matrices and 4 sparse tensors (up to 233M non-zeros)

• 9 SOTA competitors

Email Communication  Twitch Watch History  Publication Record  And Others...
Experimental settings

• 10 real-world datasets: 6 sparse matrices and 4 sparse tensors

• 9 SOTA competitors
  • Factorization-based matrix compression
    • T-SVD, CMD, CUR
  • Co-clustering-based matrix compression
    • ACCAMS, bACCAMS
  • Kronecker product-based matrix compression
    • KronFit
  • Factorization-based tensor compression
    • CP, Tucker
  • Lossless tensor compression
    • CSF (Compressed Sparse Fiber)
**NeuKron is compact and accurate**

- The outputs of NeuKron are up to 5 orders of magnitude smaller.
- The approximation error was up to 10.1X smaller in the outputs of NeuKron.

---

- **twitch**
  - Compressed Size (Bytes): $3 \times 10^{10}$
  - Approximation Error: $463697.1X$
  - NeuKron (Proposed): $1.3X$

- **kasandr**
  - Compressed Size (Bytes): $3 \times 10^{6}$
  - Approximation Error: $921.6X$
  - NeuKron (Proposed): $10.1X$

- **tky**
  - Compressed Size (Bytes): $1 \times 10^{6}$
  - Approximation Error: $300.6X$
  - NeuKron (Proposed): $12X$

- **nips**
  - Compressed Size (Bytes): $7 \times 10^{7}$
  - Approximation Error: $982.9X$
  - NeuKron (Proposed): $6.6X$
NeuKron is scalable

- Compression by NeuKron scaled linearly with the number of non-zeros
Ablation Study

- All components of NEUKRON are effective
  - the variants of NEUKRON with missing components (NEUKRON-H, -A, -F, -I) were outperformed by the original NEUKRON, equipped with all components
Conclusion

• We propose NeuKron, a lossy compression algorithm for reorderable and sparse matrices and tensors

Code and datasets are available at https://github.com/kbrother/NeuKron
NeuKRON: Constant-Size Lossy Compression of Sparse Reorderable Matrices and Tensors

Taehyung Kwon*
Jihoon Ko*
Jinhong Jung
Kijung Shin