The CoRa Tensor Compiler: Compilation for Ragged Tensors With Minimal Padding

Pratik Fegade¹, Tianqi Chen¹², Phillip B. Gibbons¹, Todd C. Mowry¹

¹Carnegie Mellon University ²OctoML
Ragged Tensors in Deep Learning

- Natural language processing

```python
input_batch = [
    [Dogs, bark, .],
    [Maine, is, a, state, .],
    [The, song, rocks, !],
    [Hello]
]
```

- Image processing

Ragged Tensor
Limited Support for Ragged Tensor Operators

- Limited support for ragged tensors
- Extensive support for dense tensors

Logos and icons representing various software and hardware components.
Padding Leads to Wasted Computation

1.07 - 2.41X wasted computation for a transformer encoder layer!
Ragged Computations Are Similar to Dense Computations

Ragged Tensor A

\[ B[i, j] = 2 \times A[i, j] \]

Ragged Tensor B
CoRa: a Tensor Compiler for Ragged Tensors

```
for i in 0:32:
    for j in 0:s(i):
        B[i,j] = 2*A[i,j]
```

```
loop_exts = [32, lambda b: s(b)]
A = input_tensor(loop_exts)
O = compute(loop_exts,
            lambda i,j: 2*A[i,j])
```

```python
i, j = 0.axis
s.padding(j, 32)
jo, ji = s.split(j, nparts=32)
s.bind(i, 'blockIdx.x')
s.bind(jo, 'threadIdx.x')
```
Transformer Layer Layer Forward Latencies on Nvidia V100

Lower is better

Datasets

Normalized Execution Time

- PyTorch
- FasterTransformer
- CoRa

RACE Wiki512 SQUADv2 Wiki 125 MNLI XNLI MRPC CoLA Overall

1.58X
0.98X
More details can be found in our paper.