Accelerating DNN Training Through Joint Optimization of Algebraic Transformations and Parallelization

Colin Unger *,1
Sina Lin 6
Vinay Ramakrishnaiah 4
Jamaludin Mohd-Yusof 4
Jongsoo Park 3

Zhihao Jia *,2,3
Mandeep Baines 3
Nirmal Prajapati 4
Xi Luo 7
Misha Smelyanskiy 3

Wei Wu 4,5
Carlos Efrain Quintero Narvaez 3
Pat McCormick 4
Dheevatsa Mudigere 3
Alex Aiken 1
Unity

Accelerating DNN Training Through Joint Optimization of Algebraic Transformations and Parallelization
Unity

Accelerating DNN Training Through Joint Optimization of Algebraic Transformations and Parallelization
1. Algebraic Transformations

2. Accelerating DNN Training Through Joint Optimization of Algebraic Transformations and Parallelization
1. Algebraic Transformations
2. Parallelization
1. Algebraic Transformations
Operator Fusion
Operator Fusion
Operator Splitting
Operator Fusion
Operator Splitting
Operator Reordering

MatMul
+ReLU

Output
ReLU
MatMul
Weight 2
Input
Weight 1
Operator Fusion
Operator Splitting
Operator Reordering
...

“Algebraic Transformations”

Operator Fusion
Operator Splitting
Operator Reordering
...

1. Algebraic Transformations
1. Algebraic Transformations

2. Parallelization
Data Parallel
Data Parallel
Data Parallel

Input → Weight 1 → MatMul 1 → Output

Weight 2 → MatMul 2 → Output
Data Parallel
Data Parallel
Data Parallel

Input → MatMul 1 → Output

Weight 1 → Weight 2 → Output

Weight 1 → Weight 2 → Output

0 / 5 / 2 / 4
Data Parallel

Input

MatMul 1

Weight 1

MatMul 2

Weight 2

Output

0

5

\frac{5}{2}

4
Data Parallel
Data Parallel
Data Parallel
Data Parallel
Data Parallel
Data Parallel

```
Input
  ^ ^ ^ ^ ^
MatMul 1  MatMul 1  MatMul 2
  |       |       |
Weight 1  Weight 1  Weight 1
  |       |       |
MatMul 2  MatMul 2
  |       |
Weight 2
  |
Output
```

```
Input
  ^ ^ ^ ^ ^
Input
  |       |
MatMul 1  MatMul 1
  |       |
MatMul 2  MatMul 2
  |       |
Weight 2
  |
```
Data Parallel
Data Parallel
Data Parallel

Diagram:

Input → MatMul 1 → MatMul 2 → Output

Input → MatMul 1 → MatMul 2

Weight 1 → MatMul 1

Weight 2 → MatMul 2

Output = \frac{3}{2} + 8
Data Parallel

Input → MatMul 1 → Weight 1 → MatMul 1

Weight 2 → MatMul 2 → Output

Input → MatMul 1 → Weight 1 → MatMul 2

Weight 2 → MatMul 2 → Output

Input → MatMul 1 → Weight 1 → MatMul 2

Weight 2 → MatMul 2 → Output

Output = 0 / 5 / 4
Data Parallel

```
Input
Weight 1
MatMul 1
Weight 2
MatMul 2
Output
```

```
Input
Weight 1
MatMul 1
Weight 2
MatMul 2
```

```
Input
Weight 1
MatMul 1
Weight 2
MatMul 2
```
Data Parallel
Data Parallel
Data Parallel
Data Parallel
Data Parallel
Data Parallel
Data Parallel
Data Parallel

Input

MatMul 1

Weight 1

MatMul 2

Weight 2

Output

Weight Synchronization

Input

MatMul 1

Weight 1

MatMul 2

Weight 2

Output

Weight Synchronization

Input

MatMul 1

Weight 1

MatMul 2

Weight 2

Output

Weight Synchronization
Data Parallel
Data Parallel

Model Parallel
Data Parallel
Model Parallel
Attribute Parallel
Data Parallel
Model Parallel
Attribute Parallel
Reduction Parallel
Data Parallel
Model Parallel
Attribute Parallel
Reduction Parallel
Parameter Parallel
Pipeline Parallel
Data Parallel
Model Parallel
Attribute Parallel
Reduction Parallel
Parameter Parallel
Data Parallel
Model Parallel
Attribute Parallel
Reduction Parallel
Parameter Parallel
Pipeline Parallel

...
Data Parallel
Model Parallel
Attribute Parallel
Reduction Parallel
Parameter Parallel
Pipeline Parallel
...

Parallelization
Data Parallel
Model Parallel
Attribute Parallel
Reduction Parallel
Parameter Parallel
Pipeline Parallel
...

Parallelization
Data Parallel
Model Parallel
Attribute Parallel
Reduction Parallel
Parameter Parallel
Pipeline Parallel
...

Algebraic Transformations
Operator Fusion
Operator Splitting
Operator Reordering
...

Parallelization
2 Parallelization

- Data Parallel
- Model Parallel
- Attribute Parallel
- Reduction Parallel
- Parameter Parallel
- Pipeline Parallel
- ...

2 Algebraic Transformations

- Operator Fusion
- Operator Splitting
- Operator Reordering
- ...

- ...
Auto-Parallelization

- FlexFlow [MLSys 19]
- Tofu [EuroSys 19]
- PipeDream [SOSP 19]
- automap [arXiv 19]
- Whale [arXiv 21]
- Alpa [OSDI 22]

...
Auto-Parallelization

FlexFlow  [MLSys 19]
Tofu      [EuroSys 19]
PipeDream [SOSP 19]
automap   [arXiv 19]
Whale     [arXiv 21]
Alpa      [OSDI 22]
...

Algebraic Optimizers

MetaFlow  [MLSys 19]
TASO      [SOSP 19]
PET       [OSDI 21]
Tensat    [MLSys 21]
...

PipeDream  [SOSP 19]
Auto-Parallelization

- FlexFlow [MLSys 19]
- Tofu [EuroSys 19]
- PipeDream [SOSP 19]
- automap [arXiv 19]
- Whale [arXiv 21]
- Alpa [OSDI 22]

Algebraic Optimizers

- MetaFlow [MLSys 19]
- TASO [SOSP 19]
- PET [OSDI 21]
- Tensat [MLSys 21]

...
Auto-Parallelization
Auto-Parallelization

Algebraic Optimizer
Auto-Parallelization

Algebraic Optimizer
MatMul

Output

MatMul

C

MatMul

A

B

0

9

/2

4
MatMul

Output

MatMul

MatMul

C

A

B

“computation graph”
MatMul
A
B
Output
MatMul
C
Data Parallelism
Reduction Parallelism
MatMul
A
B
C
"annotated computation graph"
Auto-Parallelization

Algebraic Optimizer
Auto-Parallelization → Algebraic Optimizer

Output

MatMul

MatMul

A
B
C
Auto-Parallelization

Algebraic Optimizer

MatMul

Output

MatMul

A

B

C
Auto-Parallelization → Algebraic Optimizer

MatMul

Reduction Parallelism

Data Parallelism

Output

C

A

B
Auto-Parallelization → Algebraic Optimizer

Graphical representation:
- Data Parallelism
- Reduction Parallelism
- MatMul A \* B → C
- Output
Auto-Parallelization

Algebraic Optimizer

Auto-Parallelization

Algebraic Optimizer

Output

MatMul

MatMul

MatMul

A

B

C

Reduction Parallelism

Data Parallelism

1 / 2 / 4
Algebraic Optimizer
MatMul
ReLU
MatMul
ReLU

Algebraic Optimizer
Auto-Parallelization

Input
Weight 1
Weight 2
Output

1
1
/2
4
Algebraic Optimizer

Input

Weight 1

MatMul

ReLU

MatMul

ReLU

Weight 2

MatMul

Input

Weight 1

Output

1/2

4
MatMul -> ReLU -> MatMul -> ReLU -> MatMul

Input

Weight 1

Output

1/4

Weight 2

Auto-Parallelization

Algebraic Optimizer
Algebraic Optimizer

Input

Weight 1

Weight 2

MatMul +ReLU

MatMul +ReLU

Output

Weight 2

Weight 1

Auto-Parallelization
Algebraic Optimizer

Auto-Parallelization

Output

MatMul +ReLU

MatMul +ReLU

Input
Weight 1

Weight 2

Output = \frac{1}{2} \cdot 4
Algebraic Optimizer

Auto-Parallelization

Input

Weight 1

MatMul +ReLU

MatMul +ReLU

Output

Data Parallelism

Weight 2

Weight 1

Parallelism
MatMul
ReLU
Algebraic Optimizer
Auto-Parallelization

Input

Weight 1

MatMul
+ReLU

Output

Data Parallelism

ReLU

Weight 2

MatMul

Input

Weight 1

Parallelism

1

1

4
Algebraic Optimizer

Input → MatMul → ReLU → Output

Data Parallelism

MatMul + ReLU

ReLU

MatMul

Input → Weight 1

Weight 2
MatMul + ReLU

Data Parallelism

MatMul

ReLU

Weight 2

Output

Weight 1

Input

Weight 1

Auto-Parallelization

Algebraic Optimizer
≈ 6 × less communication!
Joint Optimization
Joint Optimization

1. 

2. 
1. Representation
1. Representation
2. Scalability
Unity
Unity

Parallel Computation Graph (PCG)

Representation
Unity

- Representation
  - Parallel Computation Graph (PCG)

Scalability
Unity

Representation
Parallel Computation Graph (PCG)

Scalability
Hierarchical Search Algorithm
Parallel Computation Graph (PCG)
Parallel Computation Graph (PCG)

annotated computation graph
Parallel Computation Graph (PCG)

annotated computation graph

parallel computation graph (PCG)
Parallel Computation Graph (PCG)

Data Parallelism

MatMul

A

B

Output

MatMul

C

Reduction Parallelism

annotated computation graph

parallel computation graph (PCG)
Partition

Combine

Replicate

Reduce
Combine
Partition
Replicate
Reduce
Pipeline
Batch

\[ \sum \]

\[
\begin{array}{c}
\text{Combine} \\
\text{Partition} \\
\text{Replicate} \\
\text{Reduce} \\
\text{Pipeline} \\
\text{Batch}
\end{array}
\]
Partition → Replicate → Pipeline
Forward Pass
Combine → Reduce → Batch
Backward Pass
Combine
Replicate
Reduce
Pipeline
Batch
Partition
MatMul
Output
Weights
Input
Data Parallelism
Partition (s)
Combine
Replicate
Reduce
Pipeline
Batch
Partition
MatMul
Partition (s)
Input
Replicate
Weights
Output
Data Parallelism
1 / 2
PCG

Output

MatMul

MatMul

A

B

C
MatMul

Reduce

MatMul

Partition (c)

Partition (r)

Output

C

A

B

PCG

Substitution

\[
\frac{1}{6} - \frac{1}{2}
\]
PCG

Output

Reduce

MatMul

Replicate

C

MatMul

Partition (c)

A

Partition (r)

B
Separation of concerns
Separation of concerns

Automatically generate substitutions
Separation of concerns

Automatically generate substitutions

New operators
Separation of concerns

Automatically generate substitutions

New operators

New forms of parallelism
Separation of concerns

Explicitly represents communication
Separation of concerns

Explicitly represents communication

Concise
Hierarchical Search Algorithm
Hierarchical Search Algorithm

Algebraic Transformation
Hierarchical Search Algorithm

Algebraic Transformation

Parallelism Type
Hierarchical Search Algorithm

Algebraic Transformation

Parallelism Type

Parallelism Degree
Hierarchical Search Algorithm

Algebraic Transformation

Parallelism Type

Parallelism Degree

Device Mapping
Hierarchical Search Algorithm

Algebraic Transformation

Parallelism Type

Parallelism Degree

Device Mapping
Hierarchical Search Algorithm

- Algebraic Transformation
- Parallelism Type
- Parallelism Degree
- Device Mapping

Backtracking Search
Hierarchical Search Algorithm

- Algebraic Transformation
- Parallelism Type
- Parallelism Degree

- Device Mapping

- Backtracking Search

- Dynamic Programming
Evaluation
<table>
<thead>
<tr>
<th>Models</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-Large</td>
<td>(Language Modeling)</td>
</tr>
<tr>
<td>Candle-UNO</td>
<td>(Precision Medicine)</td>
</tr>
<tr>
<td>MLP</td>
<td>(Regression)</td>
</tr>
<tr>
<td>DLRM</td>
<td></td>
</tr>
<tr>
<td>XDL</td>
<td>(Recommendation)</td>
</tr>
<tr>
<td>ResNeXt-50</td>
<td></td>
</tr>
<tr>
<td>Inception-v3</td>
<td>(Computer Vision)</td>
</tr>
<tr>
<td>Models</td>
<td>Baselines</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>BERT-Large (Language Modeling)</td>
<td>TASO → FlexFlow (Sequential Optimization)</td>
</tr>
<tr>
<td>Candle-UNO (Precision Medicine)</td>
<td>TASO+FlexFlow</td>
</tr>
<tr>
<td>MLP (Regression)</td>
<td></td>
</tr>
<tr>
<td>DLRM</td>
<td></td>
</tr>
<tr>
<td>XDL (Recommendation)</td>
<td></td>
</tr>
<tr>
<td>ResNeXt-50</td>
<td></td>
</tr>
<tr>
<td>Inception-v3 (Computer Vision)</td>
<td></td>
</tr>
</tbody>
</table>
Models

<table>
<thead>
<tr>
<th>BERT-Large</th>
<th>(Language Modeling)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candle-UNO</td>
<td>(Precision Medicine)</td>
</tr>
<tr>
<td>MLP</td>
<td>(Regression)</td>
</tr>
<tr>
<td>DLRM</td>
<td>(Recommendation)</td>
</tr>
<tr>
<td>XDL</td>
<td></td>
</tr>
<tr>
<td>ResNeXt-50</td>
<td></td>
</tr>
<tr>
<td>Inception-v3</td>
<td>(Computer Vision)</td>
</tr>
</tbody>
</table>

Baselines

| TASO → FlexFlow   |
|                  |
| TASSO+FlexFlow    | (Sequential Optimization) |
| Expert-Designed or Data Parallel |

20/24
ResNeXt-50

CANDLE-Uno

MLP

Inception-v3

DLRM

XDL

Overall Throughput (samples/second)

Number of GPUs (Number of nodes)
ResNeXt-50

Inception-v3

CANDLE-Uno

MLP

XDL

Search Time

< 20 min
ResNeXt-50

Inception-v3

CANDLE-Uno

MLP

DLRM

XDL

Search Time

< 20 min

Training Time

hours or days
Joint Optimization
Joint Optimization

Unified Graph Representation (PCG)
Joint Optimization
Unified Graph Representation (PCG)
Hierarchical Search Algorithm
https://github.com/flexflow/FlexFlow

Keras

PyTorch

ONNX
Questions?

https://github.com/flexflow/FlexFlow