Accelerators for Graph Convolutional Networks

Adityanarayanan Radhakrishnan, Emmanuel Mensah

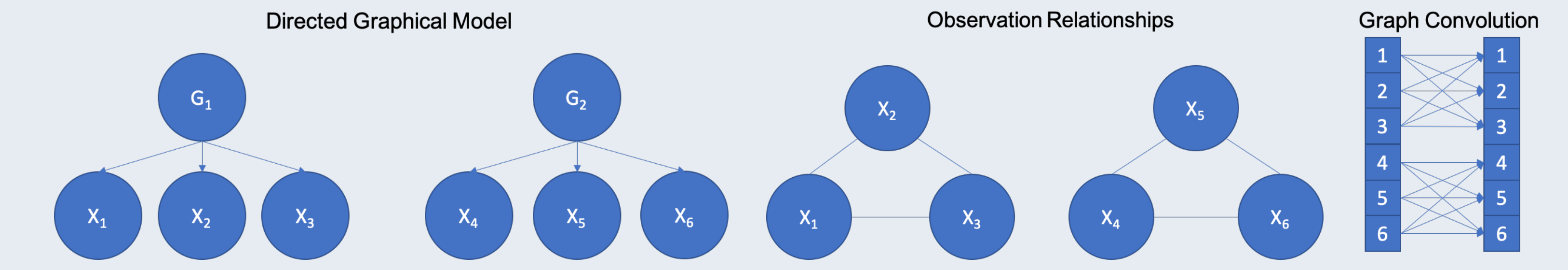
Massachusetts Institute of Technology, 6.888

Abstract

With the success of convolutional networks for image processing and recurrent networks for language/sequential data processing, recent research has turned to constructing architectures for other data modes. In particular, when the data have an underlying graph structure, graph convolutional networks take advantage of this graph structure in data processing. We analyze data flows for graph convolutional networks and provide an algorithm to accelerate graph convolutional networks based on a clique approximation of the underlying graph. When simulated in the lab 4, our method provides a 2x reduction in cycles and energy in comparison to fully connected networks and a 1.5x reduction in cycles in comparison to sparse fully connected networks on a graph that can be decomposed into 2 cliques.

Motivating Problem

- When data are generated from 2 independent factors, are fully connected networks able to learn the structure of the generators?
- If we know the graph relations between our observations, can we provide a consistent estimator for the generators?

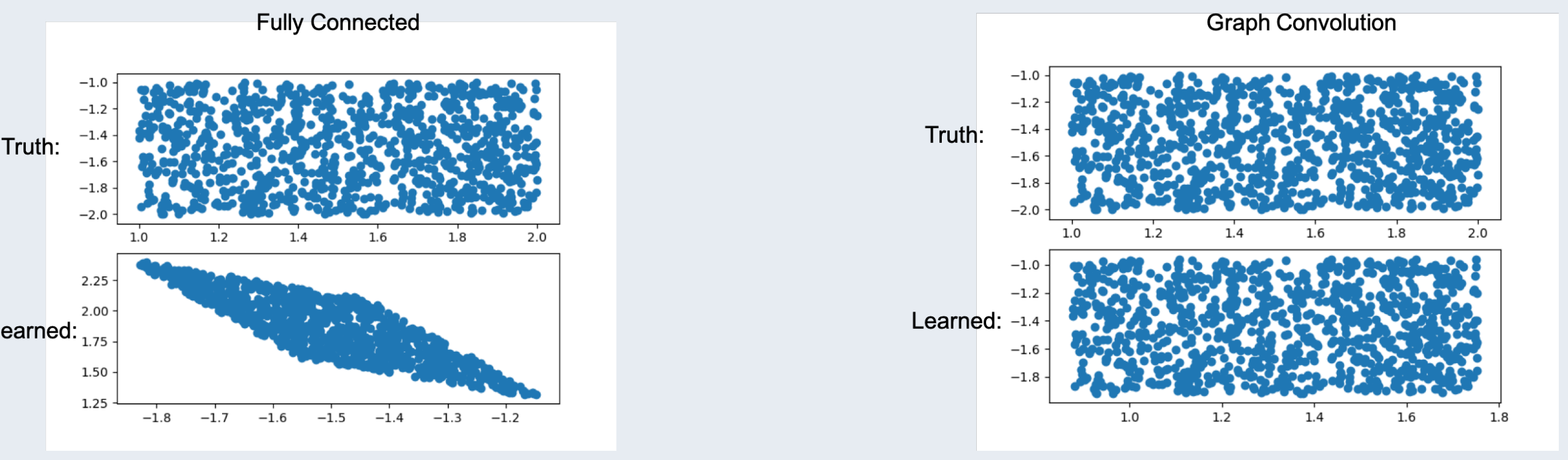


(a) The directed graphical model for generating observations.

(b) The undirected graphical model for observations.

(c) Graph convolution.

Figure: Example of graph convolution.



(a) Learned latent representation of Fully Connected Network.

(b) Learned latent representation of Graph Convolutional Network.

Figure: Learned latent representation of fully connected and graph convolutional networks. The graph convolutional network learns the true independent generative factors nearly perfectly while the fully connected correlates its latent factors.

Method and Dataflow

- As MACs are parallelizable across cliques, use a clique approximation of underlying graph.
- If there are $\mathcal{C}$ cliques of size c:

$$2\mathcal{C}\binom{c}{2} + \mathcal{C}c = \mathcal{C}c^2 \quad \text{for graph convolution} \tag{1}$$

$$(c\mathcal{C})^2 = c^2\mathcal{C}^2 \quad \text{for fully connected} \tag{2}$$

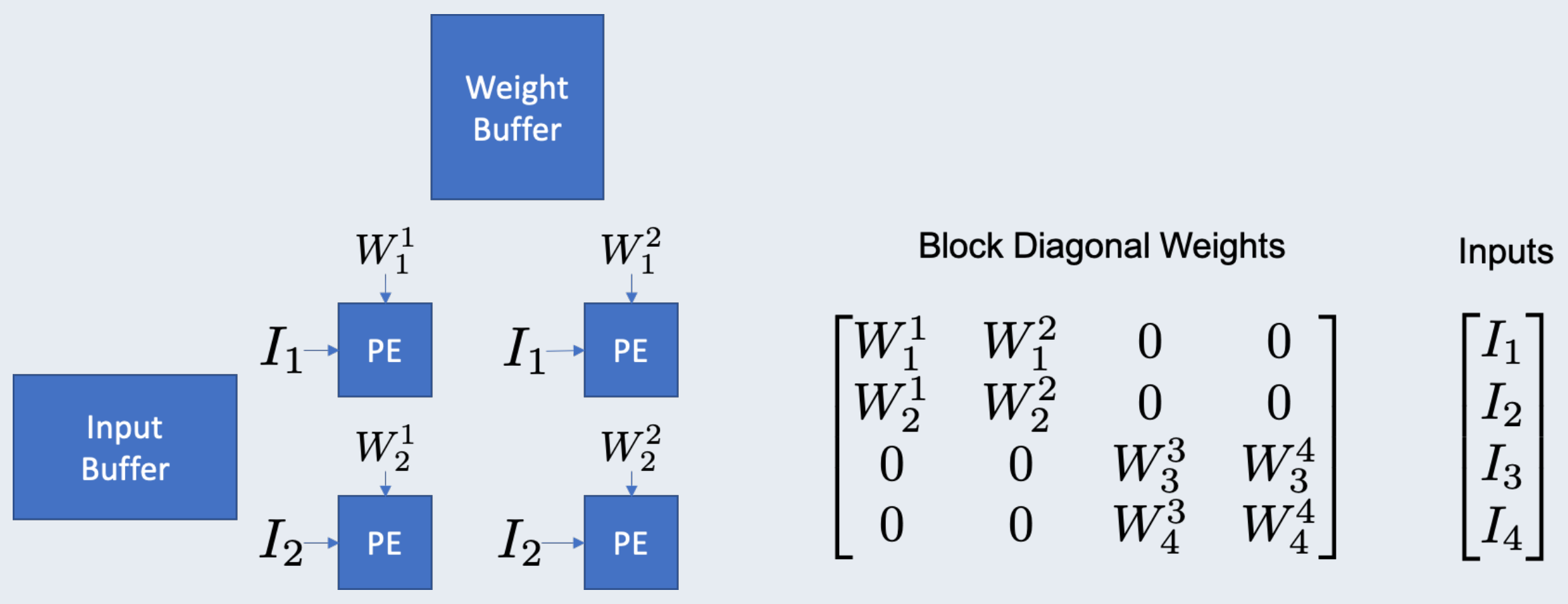

Weight Buffer
Input Buffer
PE
W_1^1
W_1^2
W_2^1
W_2^2
I_1
I_2
Block Diagonal Weights
Inputs
$\begin{bmatrix} W_1^1 & W_1^2 & 0 & 0 \\ W_2^1 & W_2^2 & 0 & 0 \\ 0 & 0 & W_3^3 & W_3^4 \\ 0 & 0 & W_4^3 & W_4^4 \end{bmatrix}$
$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix}$

Figure: Weight stationary dataflow for clique processing: weights form a block diagonal matrix and are loaded onto PEs. Inputs corresponding to a clique are multicast to the PEs.

Mapping

- To take advantage of lab 4's simulator, we reshape inputs and weights into volumes and then treat multiplication as a 1D convolution of inputs across weights.

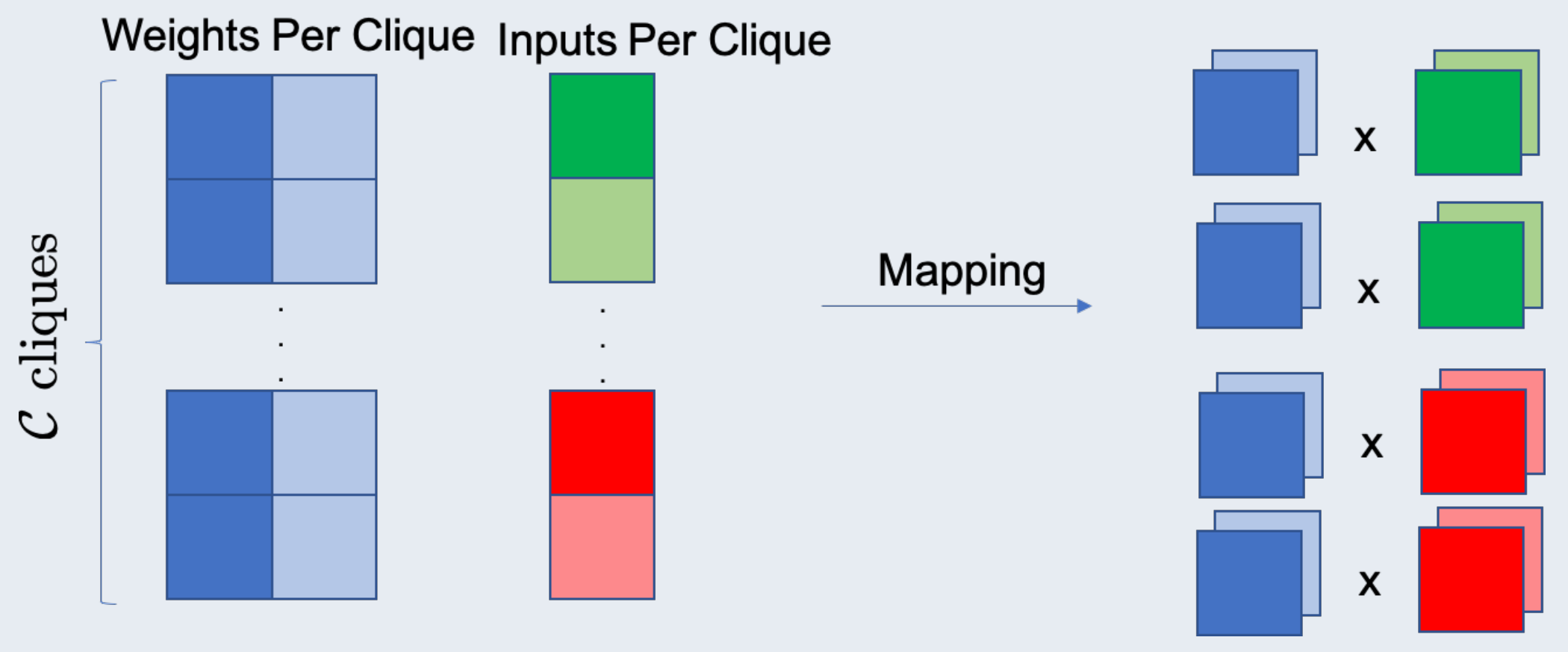


Figure: Transformation of clique processing into one dimensional convolution for processing in lab 4 simulator. Multiplication are treated as 1D convolution of inputs across clique weights.

Evaluation

- Comparison between our method (GCNN) and traditional fully connected for graphs with 2 cliques of equal size on a 4 x 4 PE array. As input size increases, there is a 2x benefit in cycles and energy matching the theory.

Input Size	Clique Size	Network	Cycles	Energy
8	4	GCNN	70	1162
8	-	FCNN	73	2016
18	9	GCNN	276	6284
18	-	FCNN	352	7116
32	16	GCNN	462	10142
32	-	FCNN	883	17027
128	64	GCNN	7062	128214
128	-	FCNN	14203	255419

Table: Comparing total energy consumption and number of cycles between our Graph Convolutional Neural Network method (GCNN) and a traditional Fully Connected Neural Network (FCNN) using a 4x4 PE array.

- With zero-skipping enabled on a single PE, our method still shows roughly a 1.5x speedup in cycles compared to a fully connected network with sparsity structure based on the underlying graph.

Input Size	Clique Size	Zero Skip Enabled	Network	Cycles	Total Energy	MAC Energy	Scratchpad Energy
8	4	No	GCNN	98	200	160	40
8	-	No	SCNN	193	240	160	80
8	-	Yes	SCNN	129	232	160	72
32	16	No	GCNN	1538	3200	2560	640
32	-	No	SCNN	3073	3840	2560	1280
32	-	Yes	SCNN	2049	3712	2560	1152
128	64	No	GCNN	24578	51200	40960	10240
128	-	No	SCNN	49153	61440	40960	20480
128	-	Yes	SCNN	32769	59392	40960	18432

Table: Comparing total energy consumption and number of cycles between our Graph Convolutional Neural Network method (GCNN) and Sparse Fully Connected Neural Network (SCNN) using a single PE element.

Conclusions

- Provide an accelerator for convolutional networks based on approximate clique decomposition of the underlying graph structure.
- When simulated, method provides roughly a 2x benefit in cycles and energy in comparison to fully connected network when underlying graph has 2 cliques.
- Trade-off between accuracy and inference when the underlying graph cannot be decomposed into cliques.