Deep learning model compression H/W design

Taking advantage of the Bit_shift operator to perform multiplication obviates the need for expensive digital multipliers.

201301178 홍성욱
201301212 홍태양
201501187 양혜린

Team

System-on-Chips

Lab

개요

문제제기

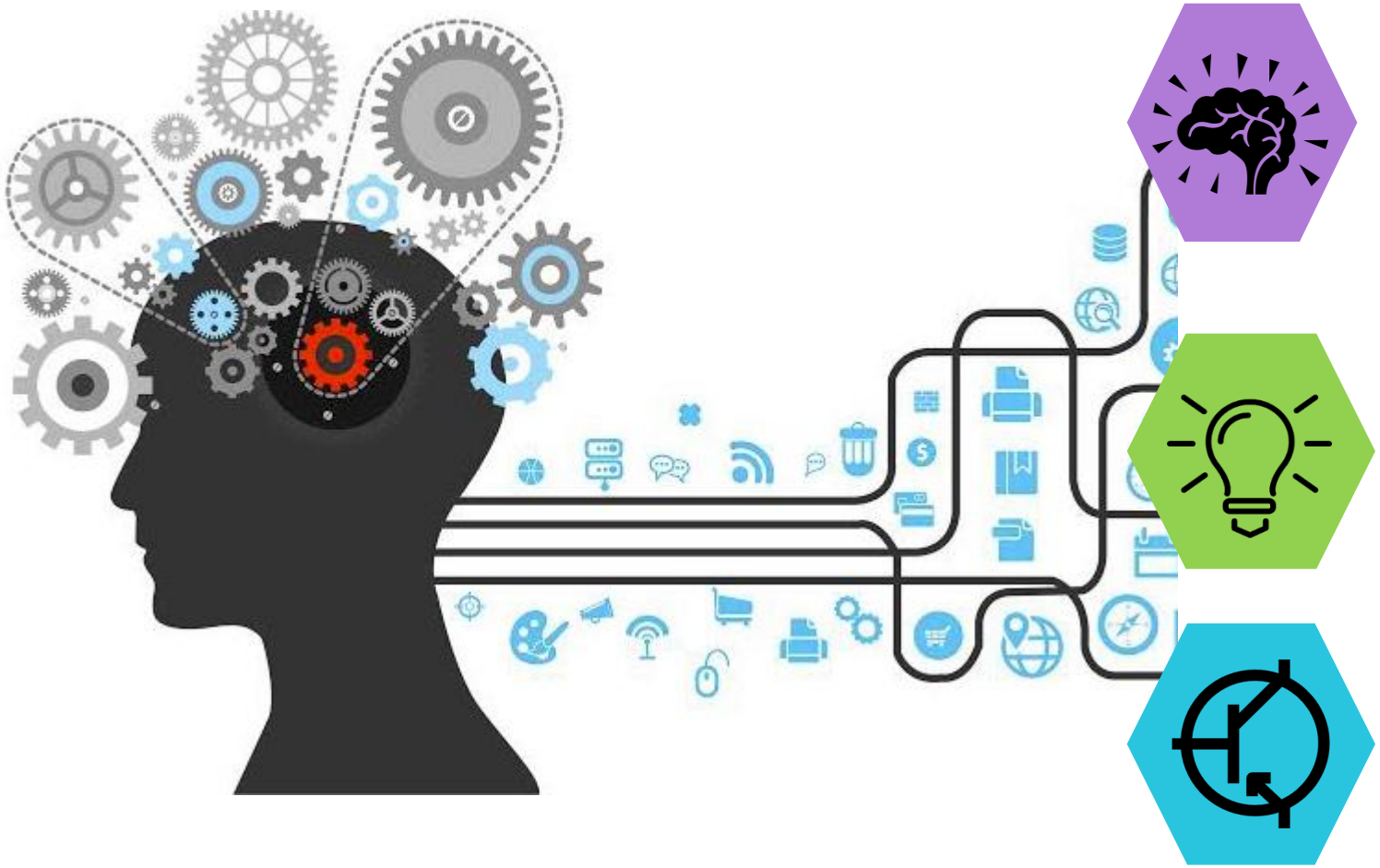
논문조사

목표 및 내용

업무분장

주간일정표

개요



Development in AI

- 고성능의 프로세서가 필요하다
- 알파고
  - 1202개 CPU
  - 176개 GPU => 48개 TPU

Computation Speed Up

- 연산과정 많은 전력과 시간 소요

Model Compression

- 전력과 시간 줄이는 방법이 필요

문제제기: Why Deep leaning?

Big Data

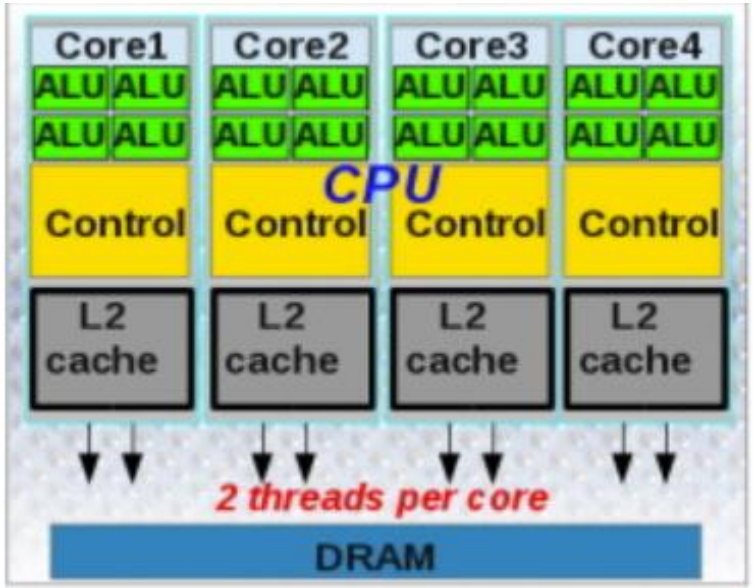
Artificial intelligence

- 4차 산업혁명 시대는 데이터 혁명의 동의어이며 따라서 빅데이터 해석이 중요시 되고 있다.
- 딥러닝이란 Training Data를 기반으로 예측 및 분류하는 인공신경망과 결합된 알고리즘으로서 빅데이터를 분석하는 핵심 기술이다.

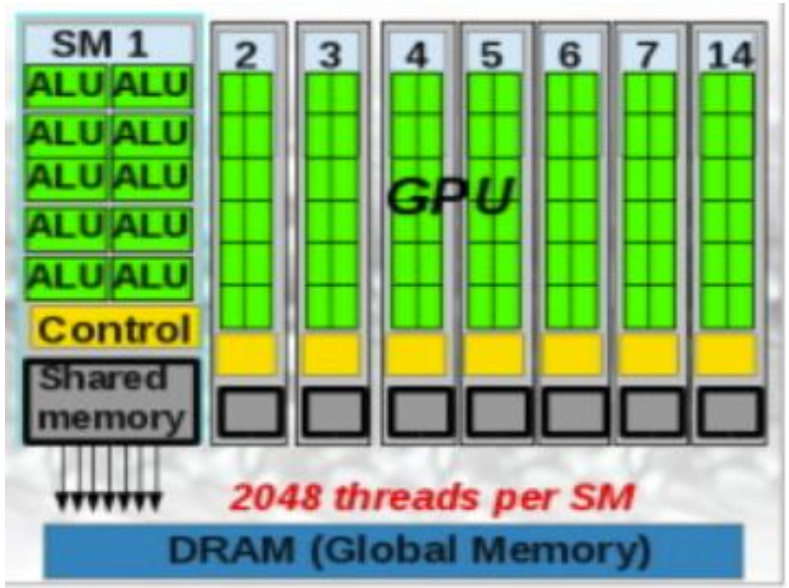
문제제기: Processor Unit

*NPU: Deep Neural Network 전용 Processor Unit

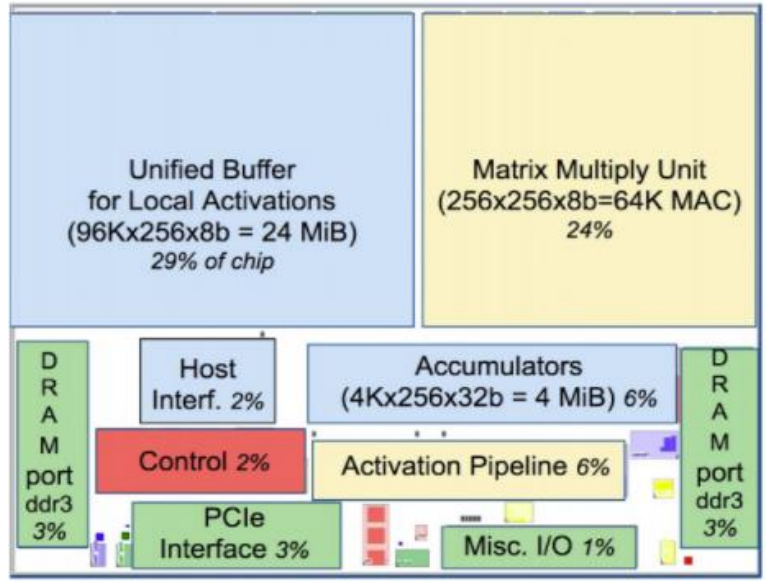
CPU



GPU



NPU(TPU)



Complex Control logic

Low Computation Density

Parallel Computation

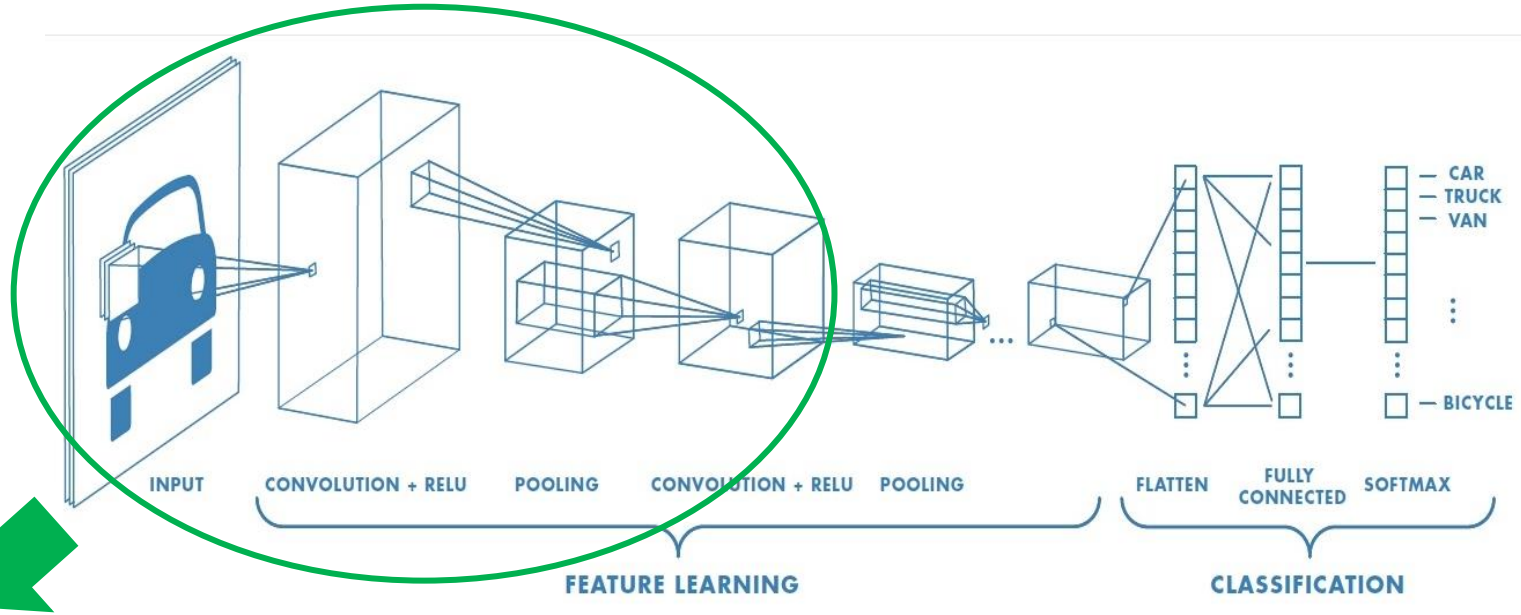
Floating Point data type

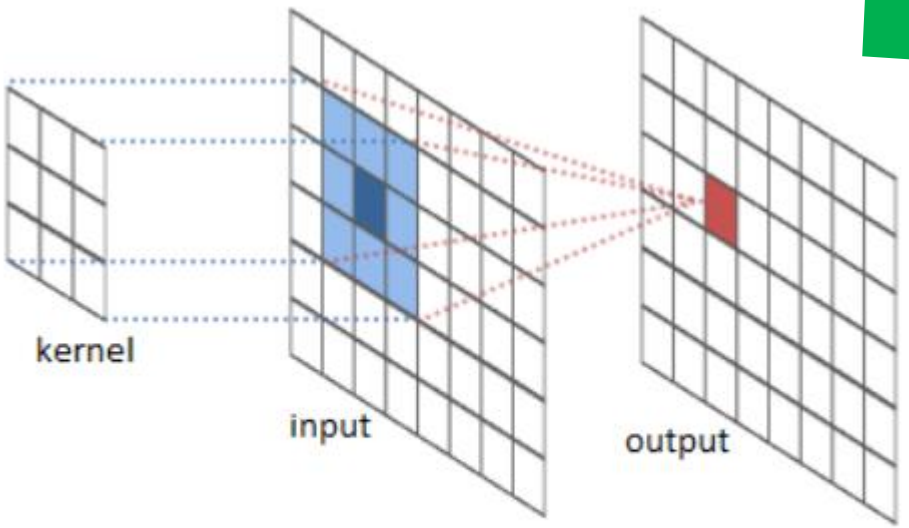
Parallel MMU (Systolic)

8bit Integer data type

Common : Deep learning 과정중 Convolution연산에서 Dot-Product를 사용한다.

문제제기: **Convolution_Matrix Multiplication**





$$F(x,y) = I * W$$

$$\text{Output}_0 = i_1w_1 + i_2w_2 + \cdots + i_9w_9$$

$$\begin{array}{r} 1010 \\ \times\ \ 10 \\ \hline 0000 \\ 1010 \\ \hline 10100 \end{array} + \begin{array}{r} 1010 \\ \times\ \ 10 \\ \hline 0000 \\ 1010 \\ \hline 10100 \end{array} + \cdots + \begin{array}{r} 1010 \\ \times\ \ 10 \\ \hline 0000 \\ 1010 \\ \hline 10100 \end{array}$$

1. Bit수가 커지면 걸리는 시간은?
2. '1'에 따른 곱셈 연산의 시간은?

문제제기: 필요성 및 동기

01

앞선 CNN의 연산과정에서 볼 수 있듯이, CNN에는 처리해야할 데이터의 양과 그에 따른 연산양이 매우 많다.

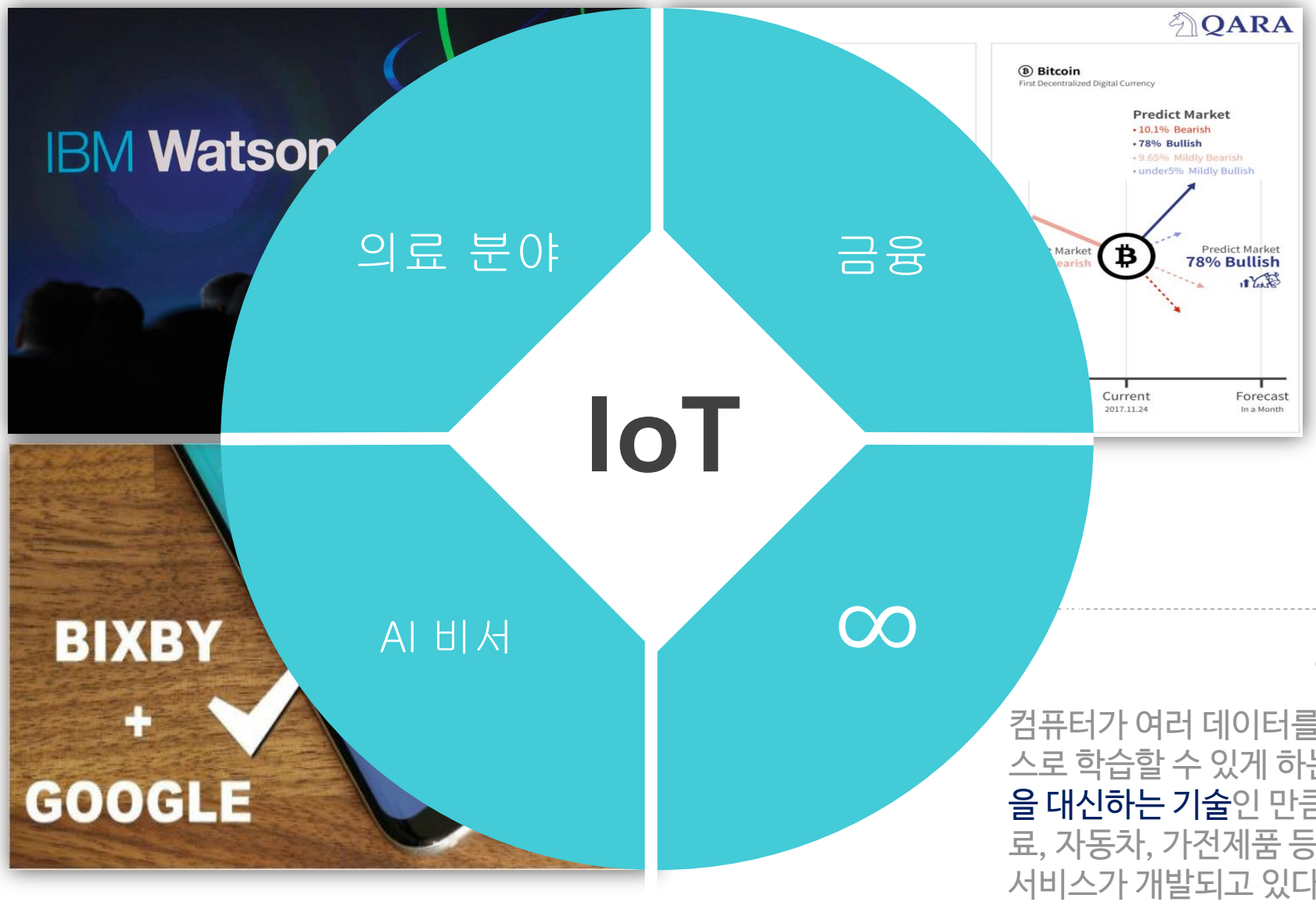
02

또한 Convolution과 Matrix Multiplication 기법은 많은 곱셈을 포함하고 있는데, 하드웨어 면에서 곱셈은 다른 연산에 비해 더 많은 수의 Clock을 소모한다.

03

곱셈기의 피처 사이즈 또한 크기 때문에 곱셈기를 개선한다면 크기 및 연산 속도 향상효과를 기대할 수 있다.

시장현황 : 시장조사



컴퓨터가 여러 데이터를 이용해 마치 사람처럼 스스로 학습할 수 있게 하는 **딥러닝의 기술은 사람을 대신하는 기술**인 만큼 분야를 가리지 않고 의료, 자동차, 가전제품 등 **다양한 시장**에서 다양한 서비스가 개발되고 있다.

시장현황 : 논문조사

Google

Deep learning model compression

Google 검색 I'm Feeling Lucky

Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding

Song Han, Huizi Mao, William J. Dally

(Submitted on 1 Oct 2015 (v1), last revised 15 Feb 2016 (this version, v5))

https://arxiv.org/abs/1510.00149

***Pruning**
0에 가까운 가중치 값을 0으로 치환

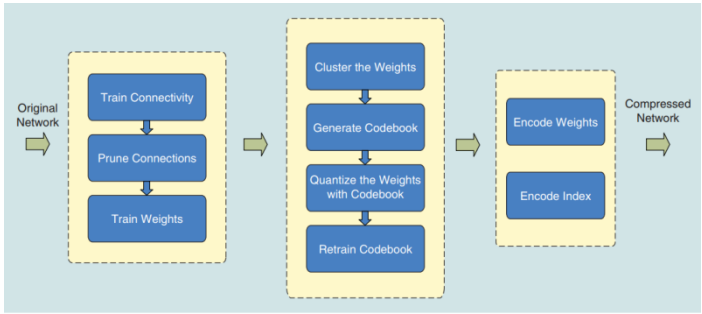
***Huffman Coding**
Encoding 방법중 하나로
빈도가 많은 Data에 대해
적은 Bit를 할당하는 기법

Model Compression and Acceleration for Deep Neural Networks

The principles, progress, and challenges

Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang

http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8253600



시장현황 : 논문조사

$1 \times 2^a = 1 \ll a$
"$Shift$"

01

$A \times B \rightarrow 2^a \times 2^b = 2^{a+b}$

03

Multiplication using Shift

Quantization

Multiplication

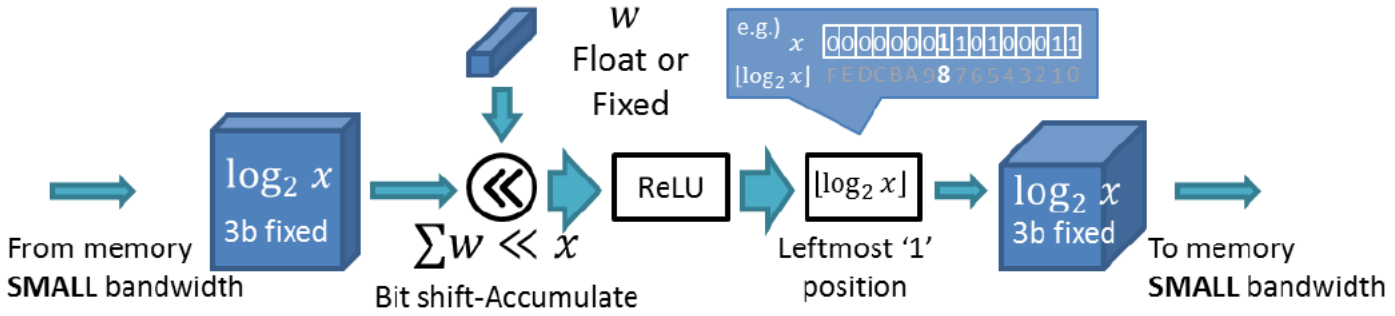
Encoding

이론

Activation과 Weight를 log도메인으로 표현하기 위해 2^a로 양자화 한다.

02

Index화를 통한 Multiplication 대체



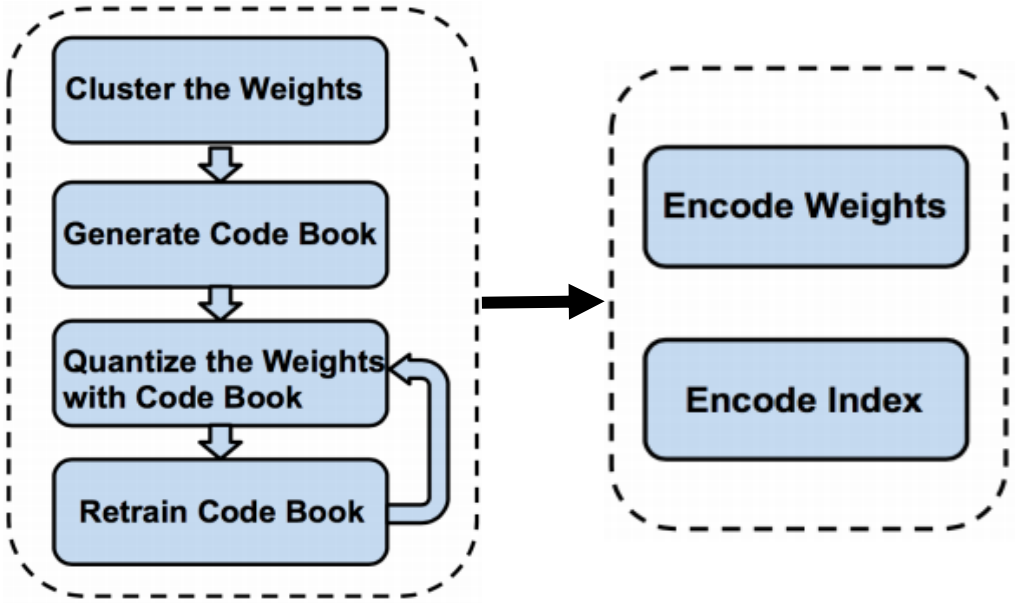
For Computation Speedup

1. **양자화(Quantization)**를 통한 Bitshift 곱셈 적용을 위한 Data 표현
2. Dot Product가 아닌 **BitShift** 곱셈 적용한 H/W(MAC) 개발

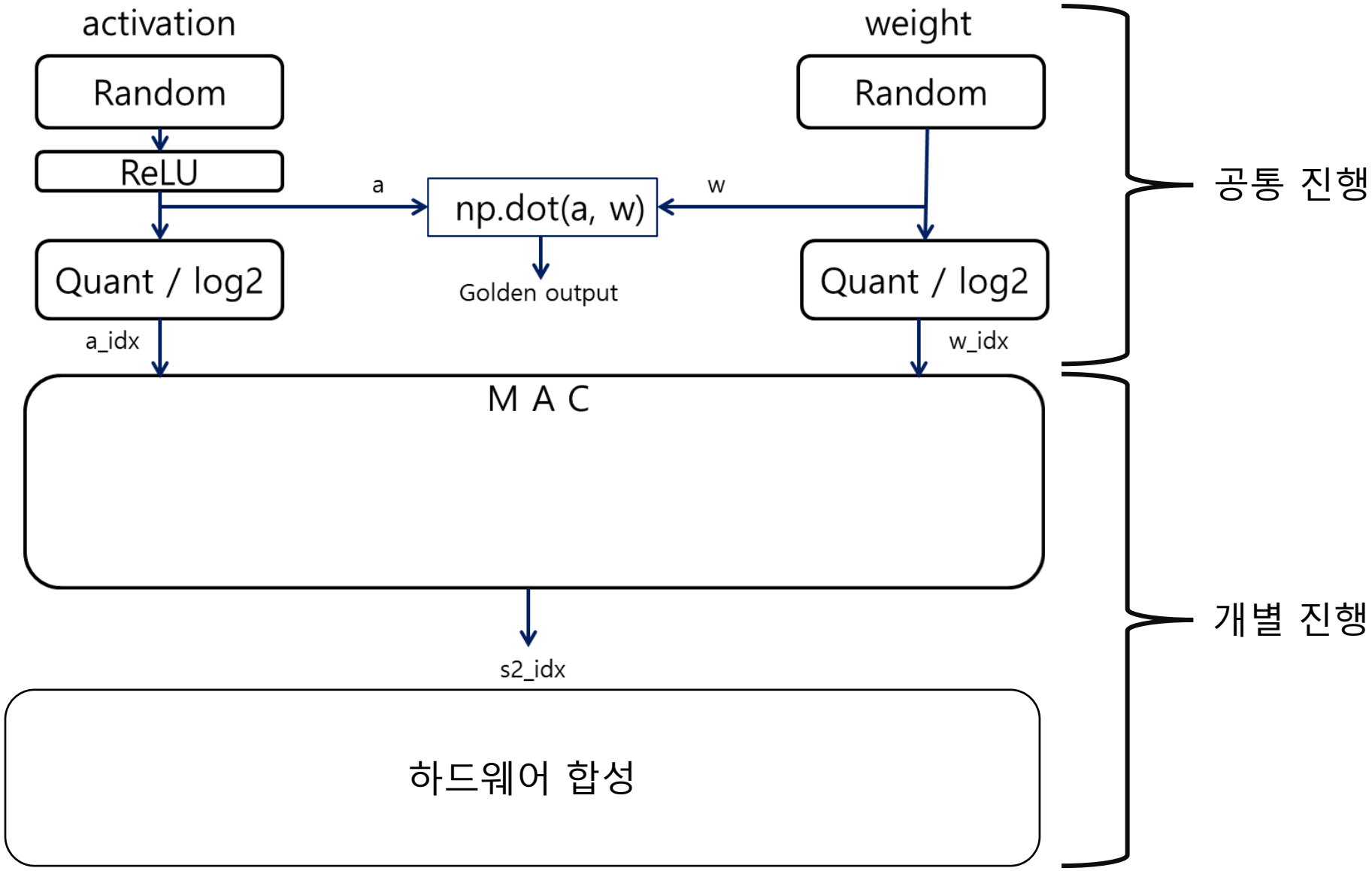
0	1	0	0	2 << 1
0	0	1	0	2
0	0	0	1	2 >> 1

For Efficiently & Flexibility

1. 양자화된 Data를 **Indexing**하여 단순한 data
2. **_Accumulation** 된 data 도 Data Range 움직여 적용 가능
3. Index를 효율적으로 해석하는 **Encoding & Decoding** 개발



업무분장


activation
weight
Random
Random
ReLU
a
np.dot(a, w)
w
Quant / log2
Golden output
Quant / log2
a_idx
w_idx
M A C
s2_idx
하드웨어 합성
공통 진행
개별 진행

주간일정표

3월	
1-2주차	Activation 및 Weight의 Log 도메인화 및 sign 분리
3-4주차	Activation 및 Weight의 효율적인 Index화 연구
4월	
1-2주차	Index화의 함수구현을 통한 Encoding 및 연산 수행
3-4주차	Operation Time 및 각 단계의 Error rate 측정
5월	
1-2주차	Error 분석을 통한 보완 연구
3-4주차	Verilog를 이용한 H/W 구현
6월	
1-2주차	H/W 합성
3-4주차	최종적인 연구 점검

출처

Model Compression and Acceleration : http://ieeexplore.ieee.org/document/8253600/

NVIDIA GPU Architecture : https://goo.gl/XsMbZN

Convolutional Neural Network : https://kr.mathworks.com/discovery/convolutional-neural-network.html

Q&A