# LUT-NN: Empower Efficient Neural Network Inference with Centroid Learning and Table Lookup

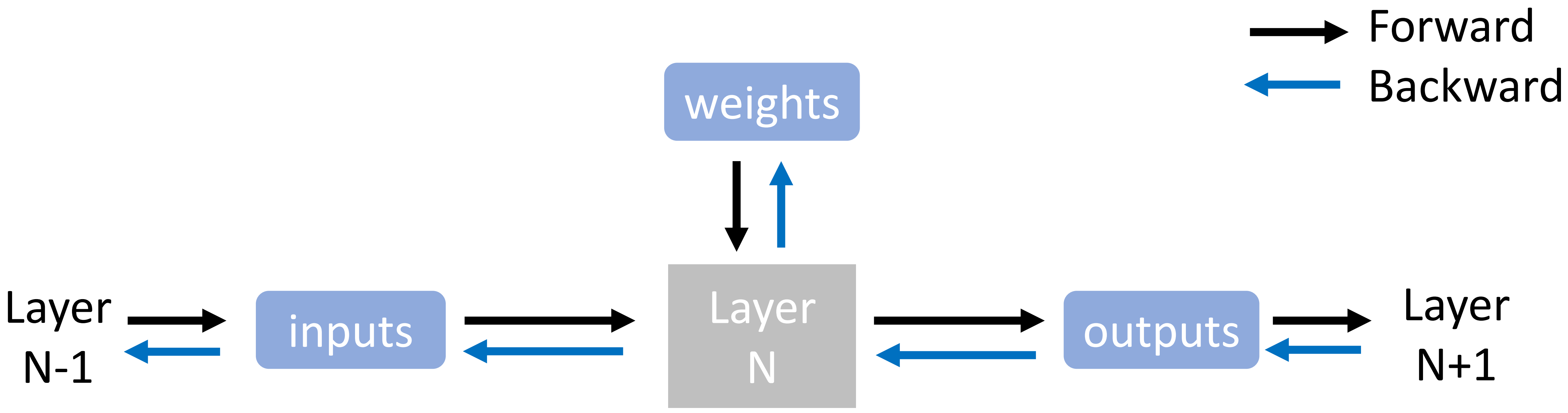
MobiCom'23

Xiaohu Tang, Yang Wang, Ting Cao, Li Lyna Zhang, Qi Chen, Deng Cai, Yunxin Liu, and Mao Yang

Zhejiang University, Microsoft Research, Tsinghua University

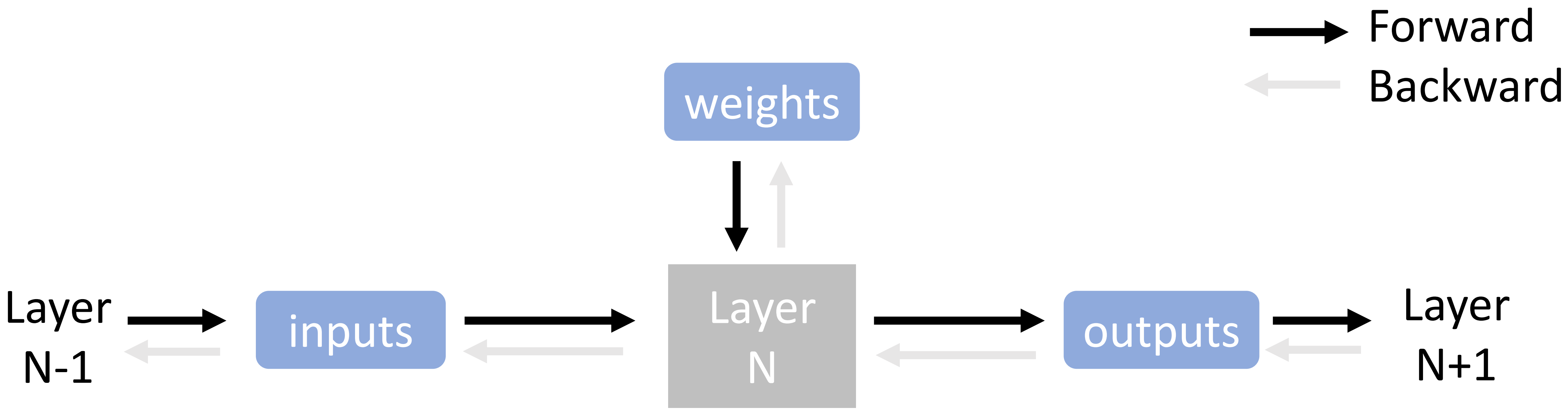
# Introduction

- DNN training



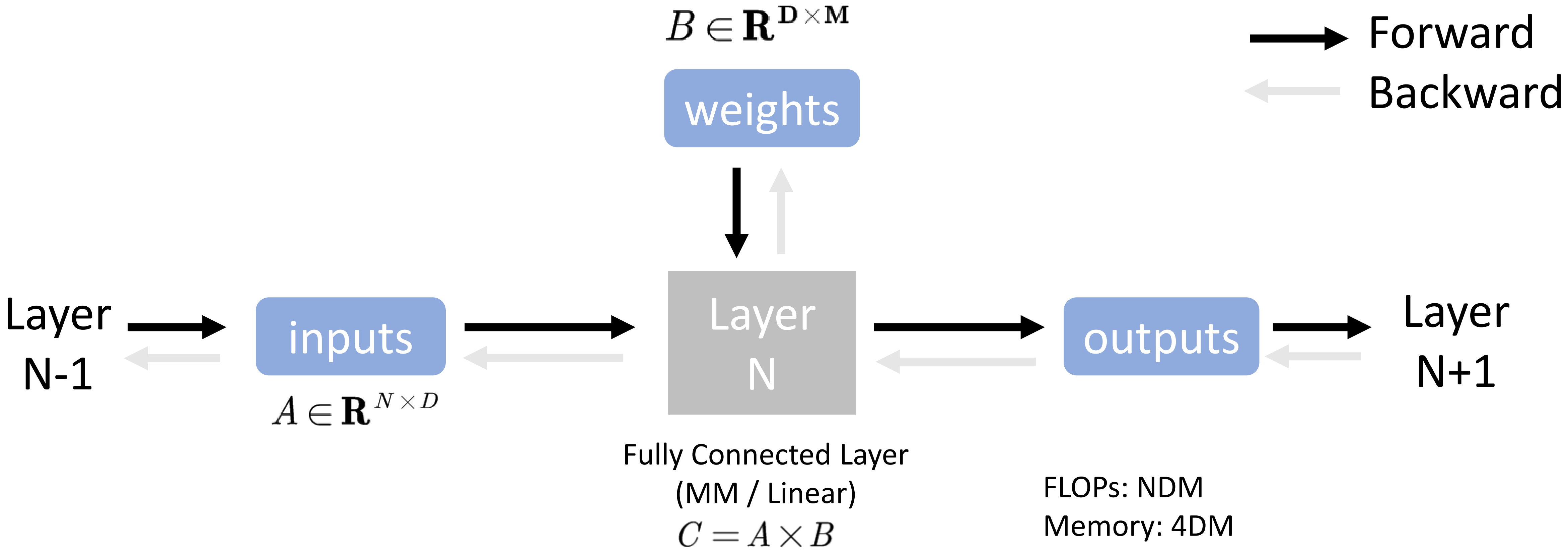
# Introduction

- DNN inference



# Introduction

- DNN inference



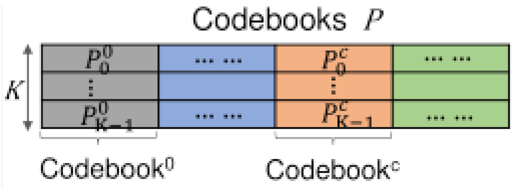
# Introduction

- Product Quantization
  - Centroid learning
    - Finding **K** centroids for the cth sub-vector with **V** dimension divided from the original vector with **D** dimension (**D=C*V**) as the cth codebook
  - Sub-vector encoding
    - The input vector will be decomposed into C sub-vectors and then clustered into different centroids

# Introduction

- Product Quantization
  - Centroid learning
    - Finding **K** centroids for the cth sub-vector with **V** dimension divided from the original vector with **D** dimension (**D=C*V**) as the cth codebook

$$A \in \mathbf{R}^{N \times D}$$

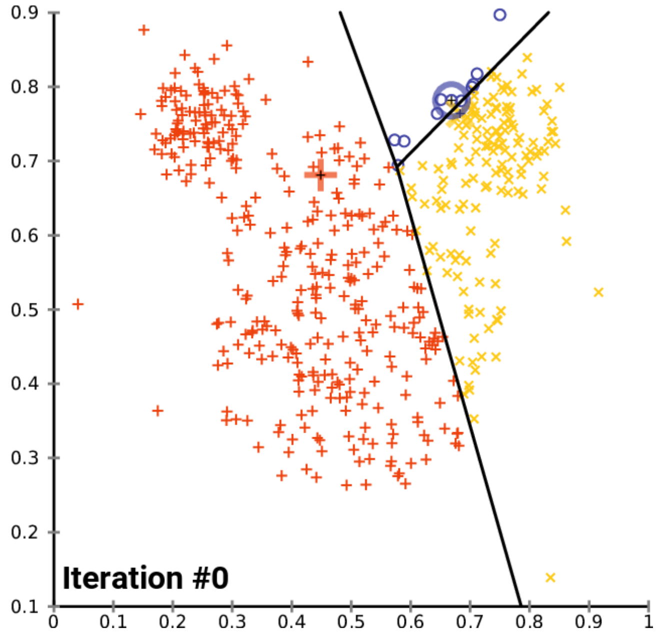



$$P \in \mathbf{R}^{K \times C}$$

# Introduction

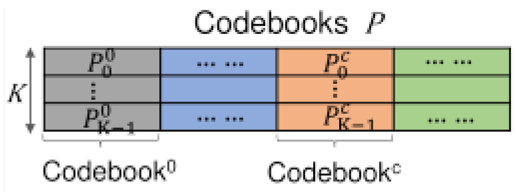
- Product Quantization
  - Centroid learning
    - Distance-based method: **k-means clustering**

$$\arg\min_{P} \sum_{c} \sum_{i} ||\hat{A}_i^C - P_k^C||^2$$



# Introduction

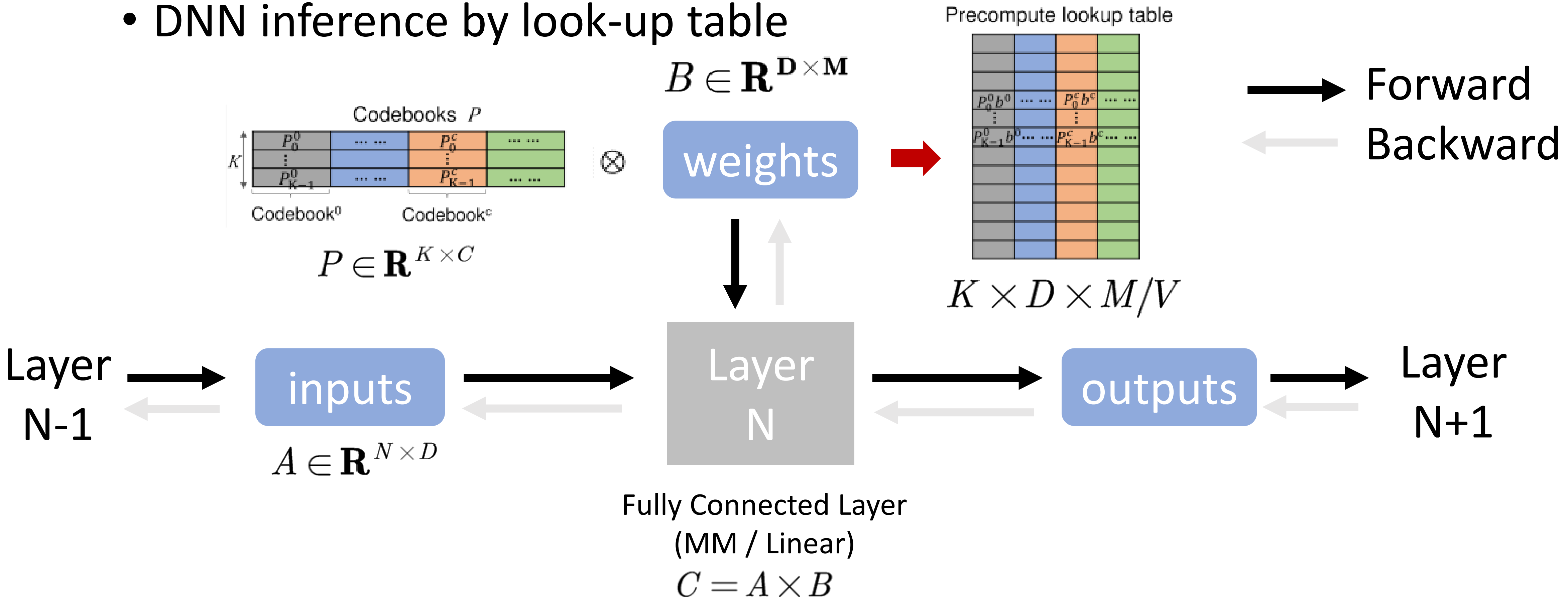
- Product Quantization
  - Sub-vector encoding
    - The input vector will be decomposed into C sub-vectors and then clustered into different centroids




$$g^C(a^C) = \arg\min_k ||a^C - P_k^C||^2$$

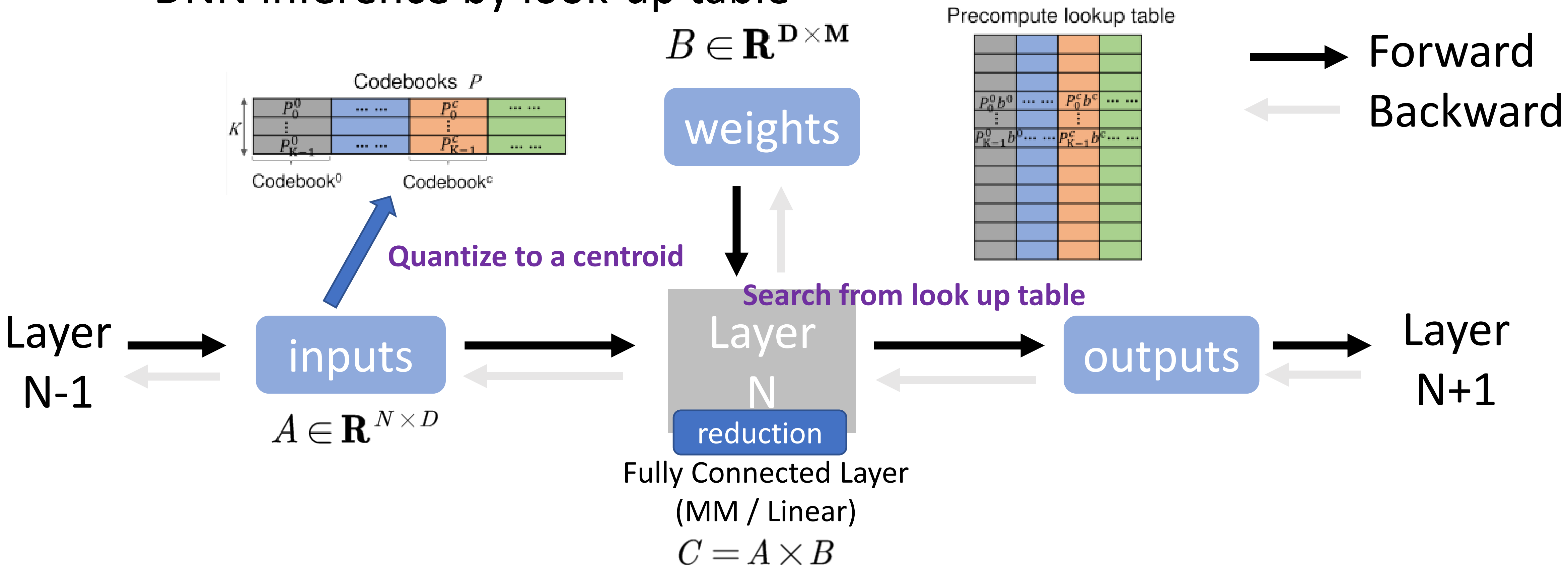
# Introduction

- DNN inference by look-up table



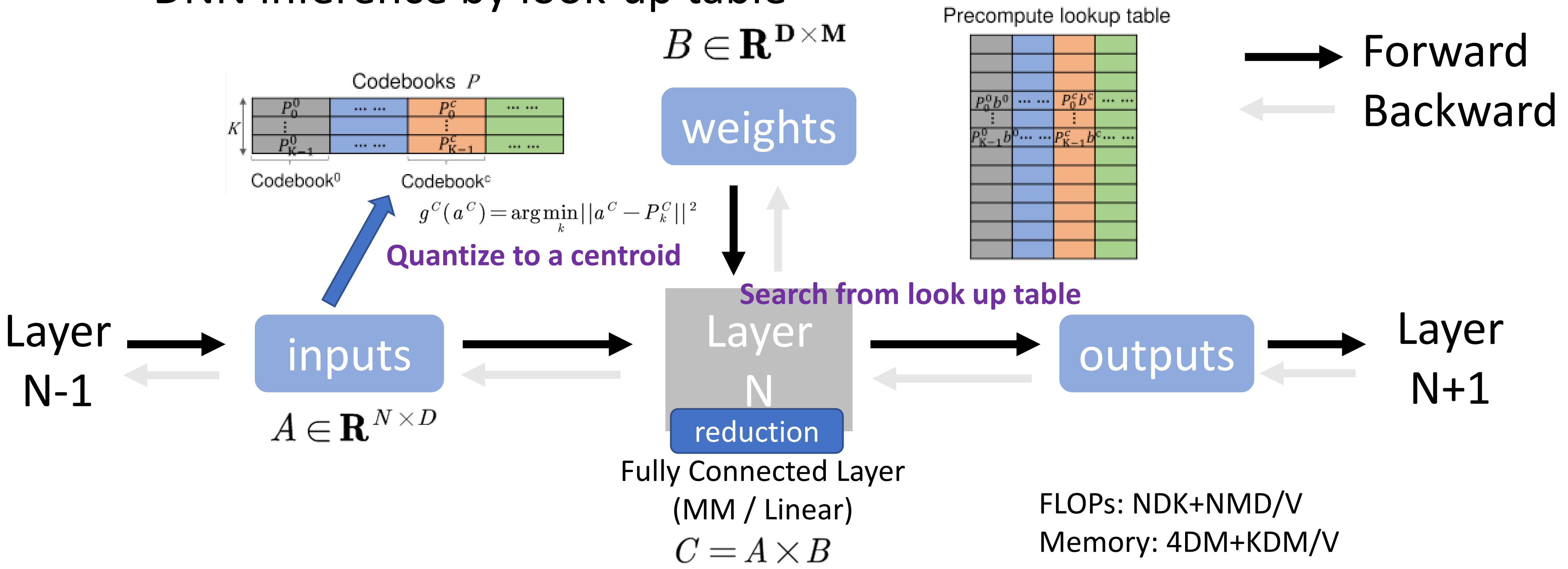
# Introduction

- DNN inference by look-up table



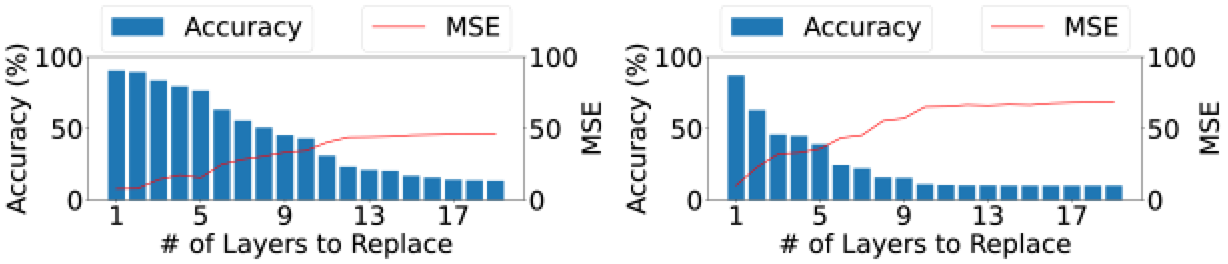
# Introduction

- DNN inference by look-up table



# Motivation

- Existing methods perform bad in accuracy




**(a) Vanilla PQ-based AMM** **(b) MADDNESS AMM**

MADDNESS [ICML'21] used hash-based centroid learning rather than k-means.

# Motivation

- Results for poor accuracy
  - The optimization goal of PQ and DNN learning is different. The **approximation error** will be accumulated  from the first layer to the last layer.
- Challenge
  - Indifferentiable of Product Quantization

$$\arg\min_{P} \sum_{c} \sum_{i} ||\hat{A}_i^C - P_k^C||^2$$

# Contribution 1: Differentiable Centroid Learning

- Soft-PQ:
  - Use of soft-max operator rather than max operator.

$$\arg\min_{P} \sum_{c} \sum_{i} ||\hat{A}_i^C - P_k^C||^2$$

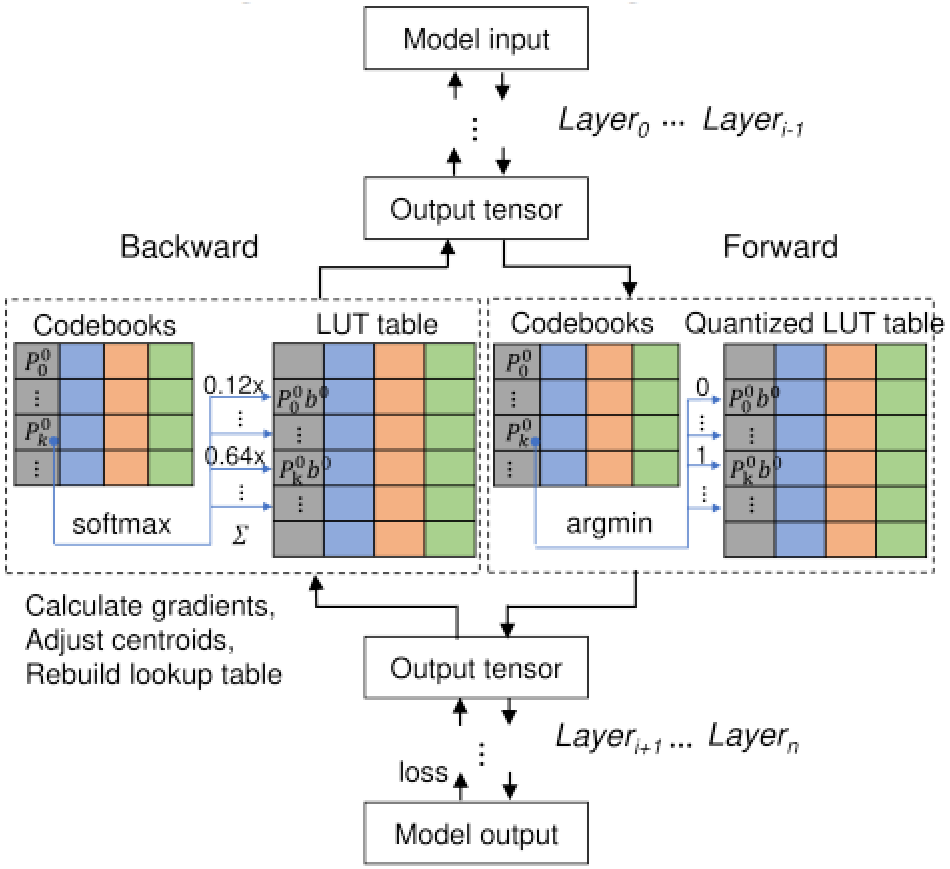
$$\Downarrow$$

$$\tilde{g}^C(a^C) = \text{softmax}(-||a^C - P_K^C||^2/t)$$

  - t represents the temperature hyperparameter. The concept is that the closer the centroid is to the sub-vector, the higher the probability will be. The encoding is transformed from a deterministic onehot vector into a probability vector. For the sub-vector AMM, the result is calculated by a dot product of the probability vector and the lookup table entries.

# Contribution 1: Differentiable Centroid Learning

- Soft-PQ:



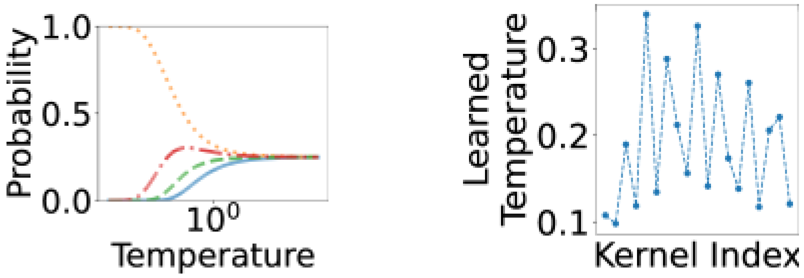

In forward pass, for simplicity, the argmin function is still utilized to calculate the model output and loss

In backward pass, calculating softmax result and its gradients, adjust centroids via gradient descent, and rebuild lookup tables with the updated centroids for the next training iteration

The **initial value is critical** for learning convergence and accuracy, using vanilla PQ to initialize the centroids and lookup tables

# Contribution 1: Differentiable Centroid Learning

- Learned-temperature:
  - Existing works are setting fixed value such as 1 or anneal it from a large number to a small one, they never analyze how to set it reasonably. This problem can be omitted in DNN training for only used softmax in one layer, but in centroid learning, **this approximation is used in each layer which may incur accumulated error.**

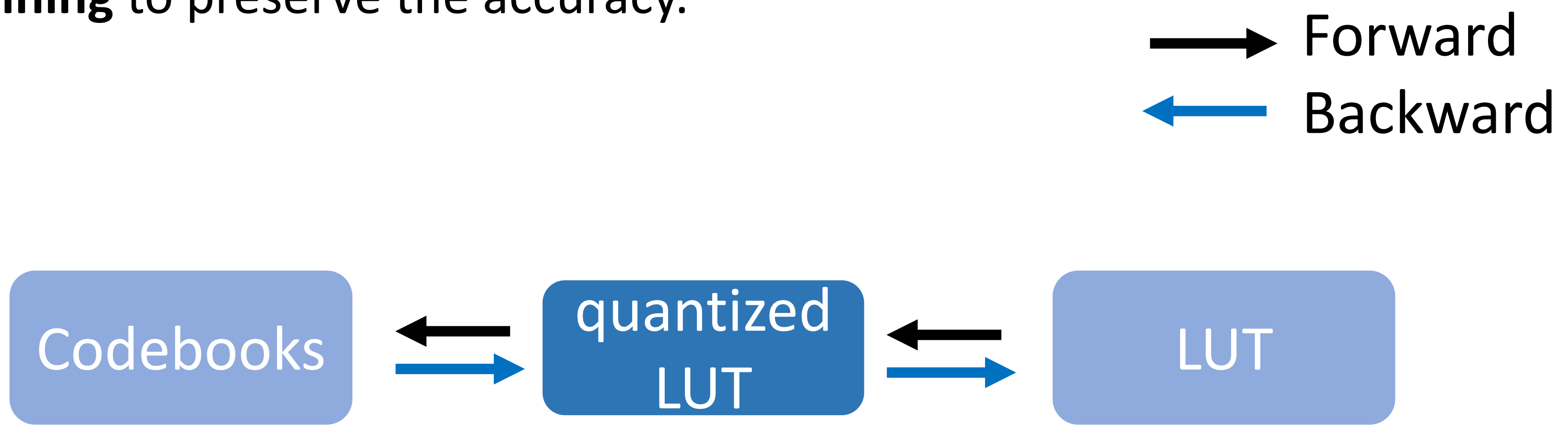



(a) (b)

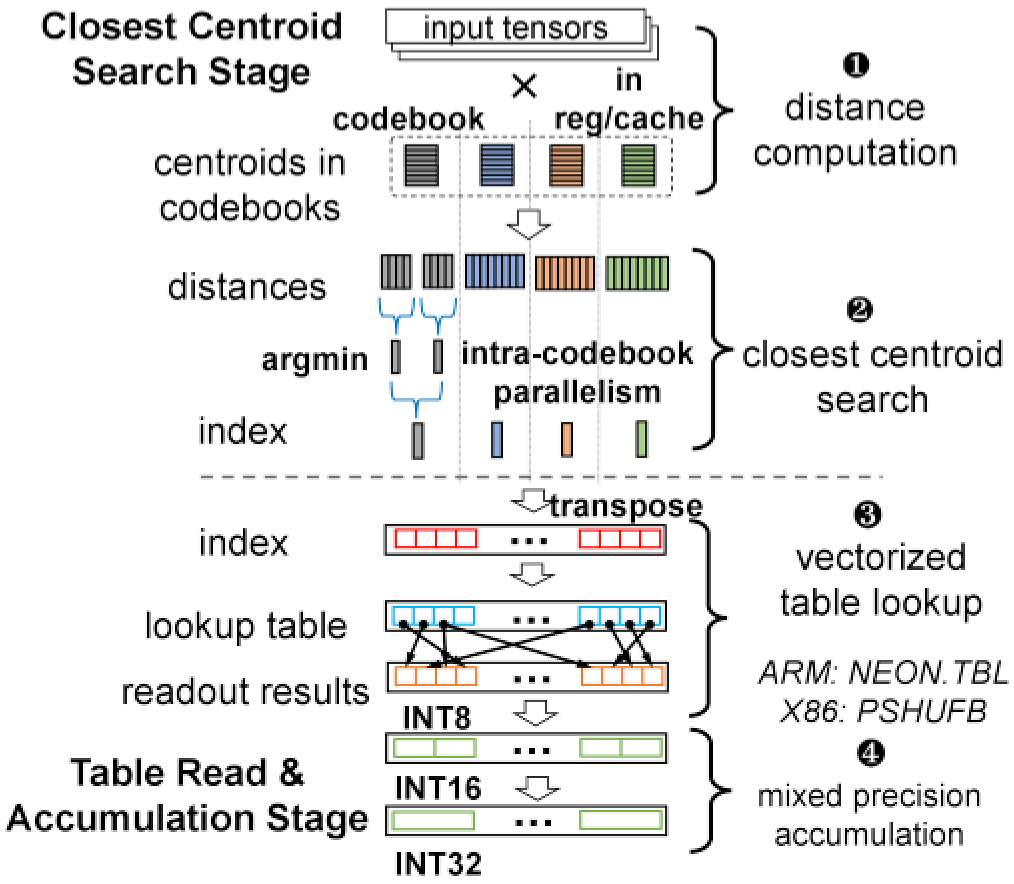
Spend less than 1 iteration of training.

# Contribution 1: Differentiable Centroid Learning

- Scalar quantization:
  - Using scalar quantization in the lookup tables with **the classic range-based linear quantization in symmetric quantization**. Using **quantization-aware training** to preserve the accuracy.



# Contribution 2: Cost reduction for LUT inference




**Centroid-stationary computation scheme**: keep centroid in reg/cache

Split the single codebook into multiple sub-codebooks for searching in parallel.

Using SIMD *shuffle* instruction to read table.

Using mixed precision accumulation (INT8 -> INT16 -> INT32) to improve accumulation throughput.

# Results

- Setup
  - Tasks: image recognition, speech recognition and NLP tasks
  - Models: VGG, ResNet, SENet and BERT
  - Datasets: CIFAR-10, GTSRB, Google Speech Command, SVHN, UTKFace, ImageNet and GLUE
  - Using age prediction task to test the regression ability
  - Metric: Mean Average Error (MAE)
  - KV setting: (16,9)
  - Devices:
    - Two mobile devices: Google Pixel 4 and 6, which are equipped with Cortex-A76 (2.42 GHz) and Cortex-X1 (2.8 GHz)
    - A desktop CPU: Intel Core i7-4790 (3.6 GHz)
    - A server CPU: Xeon Silver 4210 (2.2 GHz)

# Results

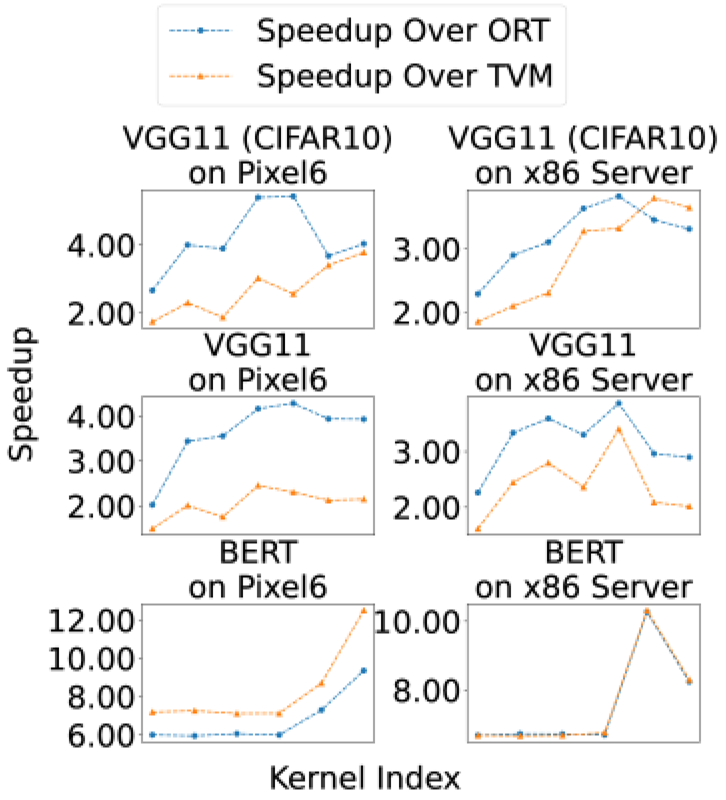
- Accuracy

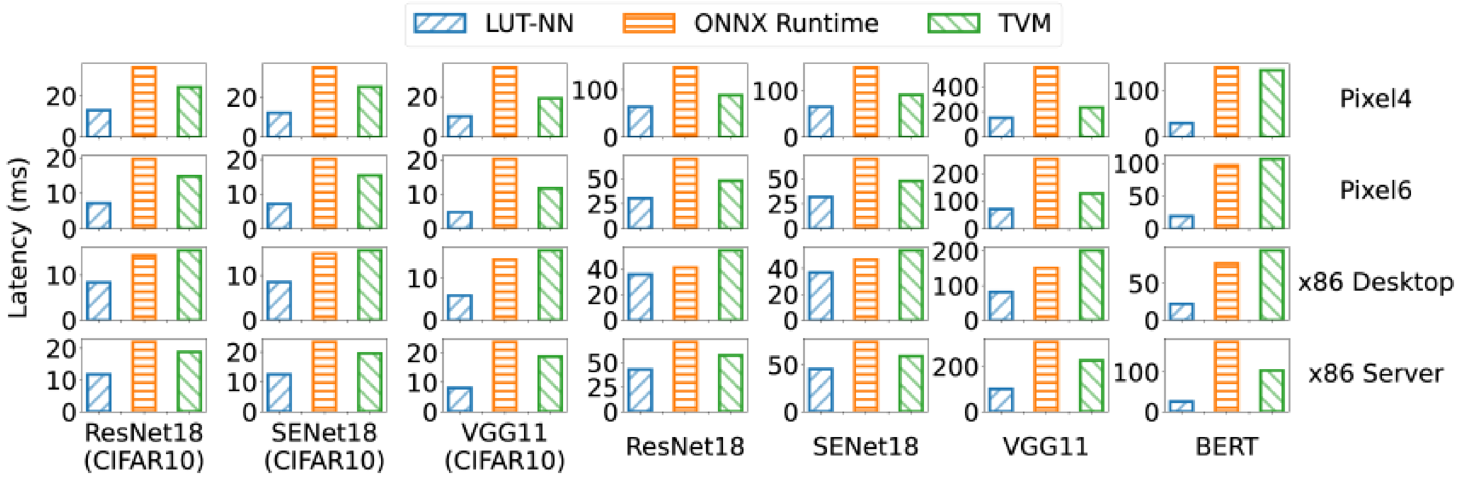
| Model | ResNet18 | | | SENet18 | | | VGG11 | | |
|---|---|---|---|---|---|---|---|---|---|
| Dataset | LUT-NN | MADDNESS | baseline | LUT-NN | MADDNESS | baseline | LUT-NN | MADDNESS | baseline |
| CIFAR10 | 94.40 | 10.01 | 95.26 | 94.48 | 10.65 | 95.47 | 93.89 | 22.87 | 95.04 |
| GTSRB | 98.73 | 4.53 | 98.80 | 98.36 | 5.68 | 98.84 | 98.55 | 5.70 | 99.22 |
| Speech Commands | 93.70 | 1.49 | 91.72 | 93.04 | 1.49 | 94.36 | 93.38 | 1.49 | 93.11 |
| SVHN | 96.00 | 20.68 | 96.67 | 96.22 | 20.12 | 96.60 | 96.23 | 29.97 | 96.62 |
| UTKFace | 4.91 | 10.51 | 5.57 | 4.74 | 11.02 | 5.46 | 5.69 | 24.57 | 5.85 |
| ImageNet | 67.38 | 0.10 | 69.76 | 68.21 | 0.17 | 70.63 | 68.04 | 0.16 | 68.33 |

| Dataset Task | Single Sentence | Similarity and Paraphrase | Natural Language Inference | | |
|---|---|---|---|---|---|
| | SST-2 | QQP | QNLI | RTE | Average |
| Training Dataset Size | 67k | 364k | 105k | 2.5k | |
| Test Dataset Size | 1.8k | 391k | 5.4k | 3k | |
| BERT base (%) | 93.5 | 71.2 | 90.5 | 66.4 | 80.4 |
| LUT-NN (%) | 92.4 | 69.6 | 87.4 | 64.7 | 78.5 |

# Results

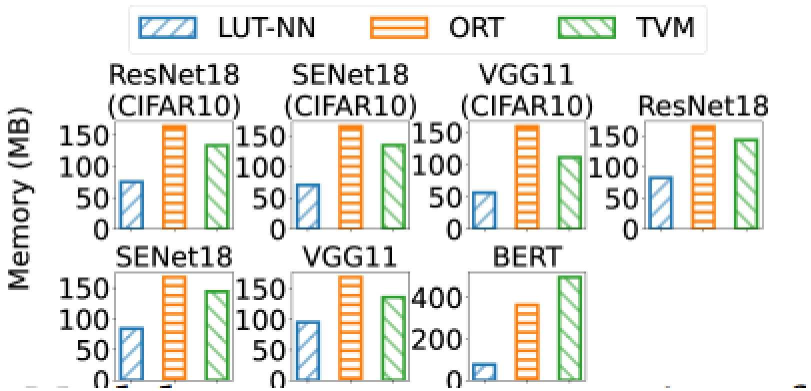
- Latency



# Results

- Memory and power




| Model | LUT-NN v.s. TVM Avg. power (W) |
|---|---|
| BERT | 2.6/3.7 |
| ResNet18 | 2.6/3.0 |
| ResNet18 (CIFAR) | 2.6/3.3 |
| SENET18 | 2.6/2.9 |
| SENET18 (CIFAR) | 2.8/3.2 |
| VGG11 | 2.3/2.9 |
| VGG11 (CIFAR) | 2.7/3.3 |

# Thoughts

- The concept of Table lookup based DNN inference is similar with codebook-based quantization.
- Engineering efforts is important in system papers especially on library optimization.

Thank You!

Oct 10, 2023

Presented by Mengyang Liu