MPEG Compression of Neural Network (NNC) 국제표준 기술 동향

□ 문현철, 정진우, 김성제 / 한국전자기술연구원

요 약

인공신경망 모델이 다양한 분야에서 뛰어난 성능을 보이고 있지만, 동시에 모델의 복잡도도 크게 증가하였다. 따라서, 모바일 같은 저전력 디바이스에 인공신경망 모델이 실시간으로 추론/배포되기 위해서는 모델의 가중치 파라미터의 수 혹은 메모리 소모량을 줄이는 경량화 기술이 필수적이다. 이에 MPEG에서는 인공신경망 모델을 다양한 프레임워크에서 상호 운용 가능하고 파라미터를 압축 표현하는 NNC (Compression of Neural Networks) 표준화를 진행 중에 있다. 본고에서는 NNC 표준의 개요와 가중치 파라미터를 압축하는 압축 기술, 그리고 HLS (High-Level Syntax)들을 소개하고자 한다.

I. 서 론

최근 DNN (Deep Neural Network)을 기반으로 하는 인공신경망은 최근 컴퓨터 비전, 객체 인식, 의료 영상, 그리고 음성 및 자연어 처리 등 다양한 분야에서 뛰어난 성능을 보이고 있다. 그러나, 성능의 향상을 위해 계층의 깊이 및 학습할 가중치의 수가 크게 증가하여 인공신경망 모델의 크기 및 추론 과정에서 접근해야 할 특징 맵(Feature Map)의 메모리 크기가 크게 증가하였다. 따라서, 실제 연산 속도나 메모리가 제한된 모바일 및 IoT (Internet of Things) 기기 등에서 인공신경망을 적용하기에는 제한이 따른다. 특히, 단일 영상이 아닌 비디오로 입력해야 하는 인공신경망 같은 경우 실시간성이 더욱 필수적이다. 따라서, 기존의 학습된 인공신경망 모델의 성능을 최대한 유지하면서 모델 크기 및 연산량을 줄이는 경량화 연구가 진행되고 있다[1].

인공신경망을 다양한 방식으로 개발하기 위해 다양한 인공지능 프레임워크(Tensorflow[2], Pytorch[3] 등) 및 HW 플랫폼이 생성되고 있다. 이때, Tensroflow[2],

※ 이 연구는 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2021-0-00802, 속성을 유지하는 지능적 미디어 화면비 변환 기술 개발)

Pytorch[3] 같은 딥러닝 프레임워크들은 모델의 개발 및 공유시에 자체 포맷을 사용하게 된다. 학습을 통해 인공신경망 모델을 만들어 내는 개발자와 추론 플랫폼을 만들어 내는 개발자들은 각자의 환경에 맞는 프레임워크들을 사용하는 것이 일반적이기 때문에 이들의 포맷이 상이한 경우 개발에 큰 어려움이 따른다. 이를 위해 다양한 딥러닝 프레임워크들 간의 상호 운용 가능한 포맷을 만드는 연구와 표준화가 진행되고 있다[4-5].

이러한 2가지 요구들을 바탕으로 국제 표준화 기구인 ISO/IEC JTC1/SC29/WG04 MPEG (Moving Picture Experts Group) Video Group에서는 NNC (Compression of Neural Networks for Multimedia Content Description and Analysis, MPEG-7 Part 17)이라는 이름으로 인공신경망 모델을 상호 운용 가능한 형태로 파라미터를 압축 표현하기 위한 표준화가 진행되고 있다[6]. NNC 표준은 2017년 처음 제안되어, 2018년부터 테스트 데이터 수집 및 CfE (Call for Evidences)를 진행하였다. 이후 2019년부터 인공신경망 모델 압축을 위한 CfP (Call for Proposals)를 게시하여 다수의 기관 및 기업에서 참여하여 본격적인 표준화가 시작되었다. CfP의 응답으로 HHI, Nokia, Technicolor 등 기업에서 총 9개의 기술들을 기고하여 기술에 대한 성능 평가를 받았고, 이를 바탕으로 제안 기술들을 3가지 Category로 나누어 CE (Core Experiments) 및 참조 SW인 NCTM (Neural network Compression Test Model)을 발간하였다. NCTM은 크게 가중치 수를 줄이거나 압축 효율을 증대시키는 가지치기, 행렬분해, Unification 같은 전처리 기술과 가중치 표현하는 비트수를 줄이는 양자화 및 엔트로피 부호화 방법으로 구성되어 있다. 이와 같은 지속적인 표준화 활동으로 2020년 CD (Committee Draft), 2022년 FDIS (Final Draft International Standard) 발간으로 현재 NNC 표준이 마무리 단계에 있는 상태이다. 더불어 데이터 저작권 이슈를 해결하기 위해 나온 연합학습 시나리오를 반영하기 위한 NNC 표준의 phase 2인 INNC (Incremental Compression of Neural Networks)의 CfP를 2020년에 게시하였으며[7], CfP의 응답으로 HHI, Tencent, Nokia 등 3개 기관에서 기술들을 기고하여 성능 평가를 받았다. 현재 INNC는 DIS (Draft International Standard)를 발간한 상태이며, 2023년 FDIS 발간을 목표로 표준화 진행이 되어 이 역시도 표준이 마무리 상태에 있다.

II. MPEG NNC 개요

1. Phase 1 NNC 개요

인공신경망 모델 추론/배포 관점에서의 압축 필요성과 전송 관점에서의 상호 운용 포맷의 필요성의 요구에 따라 2018년 NNC 표준이 시작되었다. 즉, NNC 표준은 다양한 인공신경망 모델과의 호환성을 충족시킴과 동시에 모델의 가중치 파라미터에 대한 수 혹은 메모리를 줄이는 것을 목적으로 하였다. 본 장에서는 NNC 표준의 시나리오를 검증할 수 있는 유즈 케이스 및 성능 평가 방법에 대해 기술한다.

1) MPEG NNC 유즈 케이스

MPEG NNC는 다양한 적용 분야의 시나리오에 대해 검증을 위해 유즈 케이스 및 테스트 데이터를 수집하였고, 그 결과 ETRI, Nokia, Huawei, Mitsubishi, 항공대 등 11개 기관에서 16개의 유즈 케이스 기고를 제출하였다[8]. NNC에서는 제출된 유즈 케이스들을 크게 각각의 특성에 따라 3개의 카테고리로 구분하였다. 유즈 케이스의 카테고리는 크게 모델의 일반적인 배포, 클라우드를 활용한 모델의 지속적인 업데이트, 그리고 영상 처리

<표 1> MPEG NNC 유즈 케이스 목록[8]

Category	Use cases
NN distribution/ deployment	UC1 Installing NN-based applications UC2 Camera app with object recognition UC3 Translation app UC4 Large-scale public surveillance - image classification UC5 Visual pattern recognition (VPR) UC6 NN representation for devices with limited memory and bandwidth UC9 Efficient re-use of neural networks among different media applications UC14 Electronic health record and genomic data - phase 2 - 14A: Federated learning for Medical Applications UC15 Dynamic adaptive media streaming UC16 Audio classification / Acoustic scene classification
NN (re)training	UC7 Deep NN Factory UC8 Personalized machine reading comprehension (MRC) application UC10 Distributed training and evaluation of neural networks for media content analysis - phase 2
Image/video processing and coding	UC11 Compact descriptors for video analysis (CDVA) UC12 Image/Video Compression - 12A: tool-by-tool based, 12B: end-to-end UC13 Distribution of neural networks for content processing

<표 2> MPEG NNC Phase 1 테스트 데이터

Use cases	Datasets	Retraining (optional)	Evaluation metrics	Model
UC4 Image Classification & Detection	ILSVRC2012 (224x224)	Training dataset	Top-5 Accuracy (required) Top-1 accuracy (optional)	VGG16 ResNet50 MobileNetV2
UC16 Audio classification	DCASE2017 (40x500)		Top-1 accuracy (required)	DCase model (HHI)
UC12-B Image compression	CIFAR100 (32x32)		PSNR (required) SSIM (optional)	Autoencoder (KAU)
UC 12-A Video compression	JVET CTC		PSNR (required) BD-rate (required)	In-loop filter for JEM/VTM (KAU)
UC 11 CDVA	CDVA dataset		TP/FP (required)	VGG-16 Alexnet ResNet50

및 압축으로 구성되어 있고, 다양한 인공신경망 모델 응용에서 NNC를 검증하기 위해 NNC CfE 단계에서 3개 카테고리에 대한 시나리오에 대해 각각 검증을 받았다.

CfP 평가를 위해, 앞서 테스트 데이터로 제출된 5가지 모델에 대하여 테스트 환경을 정의하였다. 〈표 2〉는 MPEG NNC Phase1에서의 테스트 데이터에 대해 기술하였다. 여기서 일반적으로 공개된 인공신경망 모델 이외에 자체적으로 제작한 모델에 대해서는 기관을 같이 표시하였다. CfP 응답을 위해 NNC에서는 모델 및 데이터의 접근성 등을 자체적으로 고려하여 응답 시 필수적으로 성능 평가를 해야 하는 데이터와 선택적 제출이 되는 데이터를 구분하였다. CfP 응답 시 필수로 성능 평가해야 할 테스트 데이터는 이미지 및 오디오 분류(UC4, UC16), 그리고 이미지 압축(UC12B)데이터

로 정의하였고, 나머지 데이터인 비디오 압축(UC 12-A), CDVA(UC 11)에 대해선 선택적 제출 데이터로 정의하였다.

2) NNC 평가 방법

2019년 MPEG NNC에서 CfP를 발간하기 위해서 그 이전 회의에서 CTC (Common Test Conditions)를 포함한 평가 방법(Evaluation framework) 문서를 발간하였다[9]. 본 절에서는 각 유즈 케이스의 제출된 테스트 데이터에 대한 평가 방법을 포함한다. 〈그림 1〉은 NNC의 평가 방법에 대한 전반적인 개요를 나타낸 그림이다. 〈그림 1〉에서 표현된 NNC의 평가 방법을 요약하면 다음과 같다.

- 추론 시 모델을 불러올 때 발생하는 메모리에 대한 모델의 압축률(O_size, R_size)
- 배포시 모델을 불러올 때 발생하는 메모리에 대한 모델의 압축률(Os_size, Cs_size)
- 네트워크 성능(e.g. classification의 accuracy)
  - O_Per: 압축되지 않은 원본 모델의 성능
  - R_Per: Reconstructed된 모델의 성능 (decoded)
  - RR_Per: 재학습(Re-training)이 포함된 복원된 모델의 성능

표현된 NNC의 평가 방법을 요약하면 다음과 같다.

여기서 원본 및 복원 모델은 인공신경망 모델 파일 포맷, 압축 모델은 이진화된 비트스트림 형태로 저장된다. 즉, NNC에서의 평가 방법은 크게 원본 모델과 bitstream을 비교한 압축률과 원본 모델 대비 복원된 모델의 성능을 측정하는 것으로 구성된다.

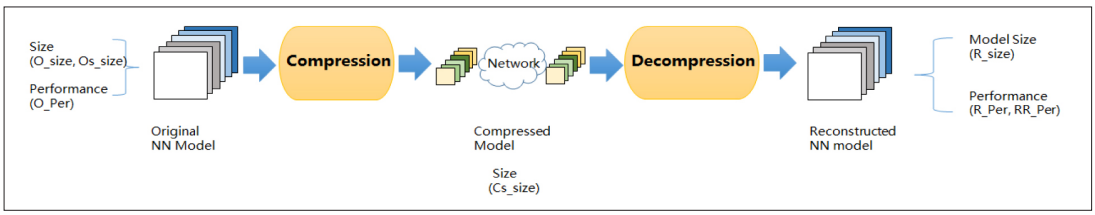
2. Phase 2 NNC 개요

인공신경망의 큰 성능 발전에 빅데이터의 존재는 필수적이었다. 그러나, 의료 영상 같은 개인 신상정보가 있는 데이터가 공유가 어려운 경우에 성능 발전에 제한이 따른다. 이를 해결하기 위해, 연합학습(Federated Learning)의 개념이 등장하였다[10]. 연합학습은 데이터 공유 없이 각 유저들이 학습한 모델을 중앙 서버로 전송하여 하나의 모델로 통합하는 것을 의미한다. 예를 들어, 의료 영상 같은 경우 각 병원에서 각자의 데이터로 학습한 모델을 중앙서버로 전송하고, 중앙 서버에서는 받은 모델의 전부를 가중치를 평균하는 방식 등으로 하나의 모델로 생성한다. 이를 통해 별도의 학습 데이터 공유 없이 모델의 성능 최적화가 가능하게 되었다.

이러한 요구들을 바탕으로 MPEG NNC에서는 2020년 10월에 연합학습 시나리오를 고려하는 NNC의 Phase 2인 INNC에 대한 유즈 케이스 및 CTC와 평가 방법에 대한 문서를 공표하였다. 이번 절에서는 INNC의 유즈 케이스와 평가 방법에 대해 기술한다.

1) INNC 유즈 케이스

INNC CfP 평가를 위해 연합학습 시나리오를 반영



<그림 1> MPEG NNC 평가 방법 개요

<표 3> INNC Test data

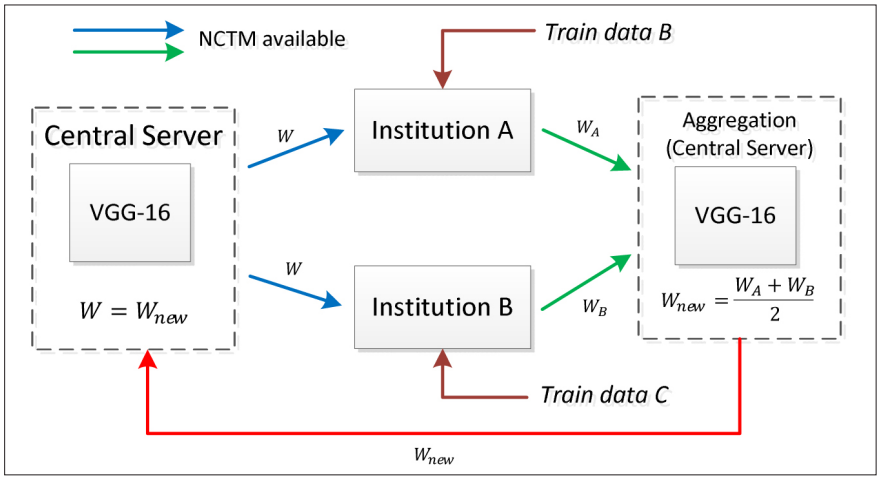
Use cases	Datasets	Retraining (optional)	Evaluation metrics	Model
UC10 Federated training and evaluation	PASCAL VOC12 (20 classes)	Training dataset	Top-5 Accuracy (required) Top-1 accuracy (optional)	VGG11 ResNet18 MobileNet
UC14A Federated learning for medical applications	Pneumonia detection dataset		Top-1 accuracy (required) Precision, recall, F1 score (optional)	VGG16 (KAU)

한 테스트 데이터들이 제출되었고, 〈표 3〉은 INNC 테스트 데이터를 나타낸다. 여기서 일반적으로 공개된 신경망 모델 이외에 자체적으로 기관에서 제작한 모델에 대해서는 기관도 같이 표시하였다. 2가지 테스트 데이터 모두 기존의 접근성이 용이한 미리 학습된(Pre-trained) 이미지 분류 모델 기반으로 제작되었다. 또한 INNC에서는 NNC와 달리 CfP 응답 단계에서 선택적 제출 데이터가 없고, 〈표 3〉의 2가지 테스트 데이터 모두에 대한 압축 성능 평가를 한다.

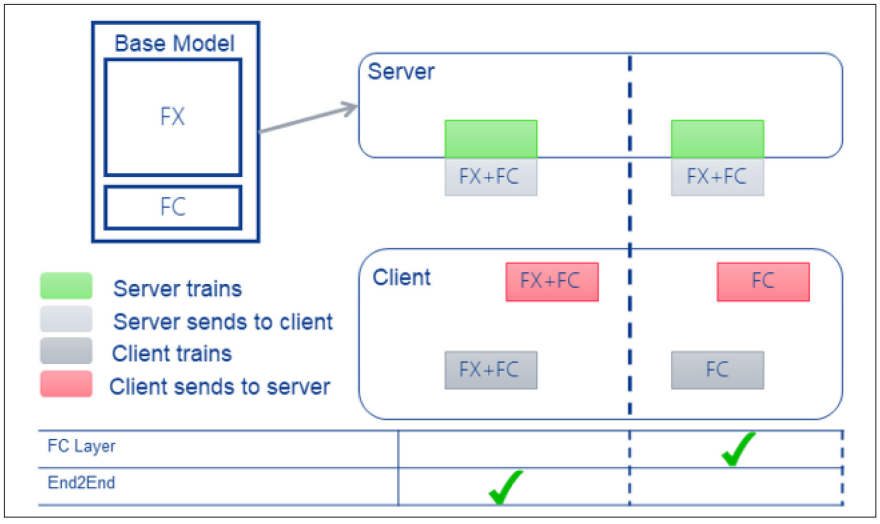
〈그림 2〉는 Phase 2의 주요 테스트 데이터인 연합학습을 위한 의료영상(UC 14A)에 대한 흐름도를 나타낸 것이다[11]. 연합학습의 시나리오를 위해 본 기고에서는 Kaggle에 있는 흉부 X-ray 사진의 폐렴 탐지 데이터에서 중앙서버와 2개 기관 등 총 3개의 데이터로 나누었다. 이러한 3종류의 학습데이터를 이용한 NNC에서의 연합학습 시나리오는 다음과 같다. 우선, 중앙서버에 있는 데이터로 학습을 진행하고 나서 각 기관에 전송을 한다. 그 다음 각 기관(A, B)에서도 각자 가지고 있는 데이터로 학습을 진행하고 중앙 서버의 결과와의 잔차 가중치 정보를 다시 중앙서버로 전송하게 한다. 마지막으로, 중앙 서버에서는 각 기관에서 받은 잔차 가중치를 평균하여 새로운 모델을 생성하고, 이를 계속 반복하게 한다.

2) INNC 평가 방법

2020년 INNC에서 CfP를 발간하기 위해서 그 이



<그림 2> Phase2 연합학습 유즈 케이스 흐름도[11]



<그림 3> INNC Pipeline[24]

전 회의에서 CTC문서를 발간하였다[24]. 〈그림 3〉은 INNC의 평가 방법에 대한 개요도를 나타낸 것이다. 전반적인 INNC의 평가 방법은 연합학습을 고려한 각각의 시나리오에 대해 모두 압축률 및 성능 지표를 제출해야 한다. INNC에서 고려해야 할 시나리오는 다음과 같이 요약된다.

- 학습 방법
  - Bidirectional 가중치 업데이트에서 각 client에서는 epoch 1만큼만 학습을 진행하여 server로 전송. 이 때의 과정을 하나의 단계(stage)로 정의하고, 15번 단계만큼 반복하여 각 단계마다 성능 평가 진행
  - Training stage는 15로 설정
- 가중치 전송 방법
  - 이전 단계와의 잔차 가중치 정보를 대상으로 압축 및 전송
  - Server → Client만을 고려한 단방향 가중치 업데이트(기존 NNC)
  - Server → Client, Client → Server를 모두 고려한 양방향 가중치 업데이트
- 모델의 압축 범위(모델의 구성은 〈그림 3〉에서의 FX+FC)
  - 모델 전체
  - 모델 일부 - 완전 연결 계층(〈그림 3〉의 FC)

여기서 모델의 압축 범위 같은 경우는 일반적으로 Tensorflow, Pytorch에서 VGG, ResNet 모델의 특징 추출 계층의 가중치 정보를 제공하기 때문에 각 서버에서 해당 계층의 가중치를 가지고 있는 점을 고려하여 INNC에서는 모델 전체와 모델 일부인 완전 연결 계층만을 압축했을 때의 성능을 각각 평가한다.

III. MPEG NNC Coding Tools

MPEG NNC CfP의 응답에 대한 각 제안 기술들을 평가 방법에 따라 각 유즈 케이스에서의 성능을 측정하였다. 이 때, 주요 특성에 따라 3개의 CE를 설정하였다.

<표 4> MPEG NNC CE 구성

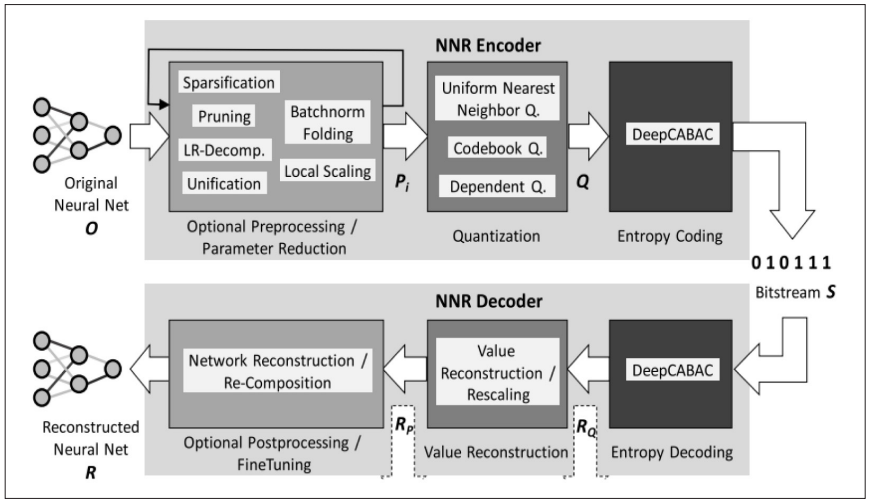
CE	Features	Methods	Organization
CE1	Parameter reduction	Sparsification/pruning Matrix decomposition Unification	Nokia, HHI, ZJU Interdigital, KAU, Insiginal, Tencent, ZJU
CE2	Quantization	Uniform/codebook quantization Dependent quantization Local quantization	Nokia, Interdigital, PKU HHI, Tencent, KAU, KHU, KETI
CE3	Entropy coding	CABAC Arithmetic coding	HHI, Nokia, interdigital

〈표 3〉은 NNC CE에 대한 구성 및 참여기관들을 요약하였다. CE1은 모델의 가중치 파라미터의 수를 줄이는 특징을 가지고 있으며, 원본 모델 대비 압축된 모델의 압축률과 실제 추론 시 발생하는 메모리 및 실행 시간들을 고려하는 기술들이다. 반면에 CE2와 CE3인 양자화와 엔트로피 부호화 방법은 모델의 가중치 파라미터를 bitstream으로 만들어 표현하는 비트 수를 줄이는 기법으로써 입력으로 들어가는 모델 대비 bitstream 크기인 압축률과, 복원된 모델의 성능을 고려한다.

이번 장에서는 NNC 표준에서 채택된 압축 기술들에 대해 기술한다. 우선 1절에서는 채택된 압축 기술들로 구성된 NNC의 참조 SW인 NCTM에 대한 개요를 설명하고, 이후 2~7절에서는 채택된 기술의 구체적인 알고리즘에 대해 기술한다.

1. NCTM Overview

CfP 응답 및 CE 단계에서 제안된 기술 중 채택된 기술들로 NNC에서는 참조 SW인 NCTM을 발간하였고, 〈그림 4〉는 NCTM에 대한 전반적인 개요를 나타내었다[12]. 우선, NNC에서는 크게 파라미터의 수를 줄이거나 선택적인 후처리 모듈 같은 전처리 기술과 각 가중치



<그림 4> NCTM 개요[23]

파라미터가 표현하는 비트 수를 줄이는 양자화, 엔트로피 부호화 같은 코딩 기술들로 구성되어 있다. 일반적인 NCTM의 흐름도는 우선 원본 모델을 전처리 기술을 통해 파라미터 수를 줄인다. 그 다음 전처리된 모델을 양자화 및 엔트로피 부호화를 통해 비트스트림을 생성하고, 비트스트림을 복호화하기 위해 엔트로피 복호화 및 역 양자화를 수행하고 선택적으로 후처리 미세조정(Fine-tuning)을 통해 복호화된 인공신경망 모델을 생성하게 된다. 다음 절부터는 채택된 기술들을 CE 순서대로 기술한다.

2. Sparsification/Pruning

가지치기 기법은 일반적으로 〈그림 5〉와 같이 가중치와 노드의 연결을 제거하여 인공신경망 모델에서 표현하는 가중치의 수를 줄인다[1]. 여기서, 가중치와 노드의 연결을 제거 기준을 각각의 가중치에 대하여 모델의 성능에 미치는 중요도로 판단되어진다. 대부분의 가지치기법에서는 가중치의 값이 매우 작을 경우 모델의 성능에 내성이 있다고 간주하고, 해당 가중치 값을 0으로 만들어 노드와의 연결을 제거한다. 또한, 모델 내의 압축률 조절을 위해 수식 (1)처럼 입력되는 희소도(Sparsity)에 충족하는 임계 값을 구하고 난 뒤 해당 임계 값보다 낮은 값을 가진 가중치의 값을 0으로 만든다. 또한, 모델 성능의 감소를 최소화하기 위하여 별도의 미세 조정(fine-tuning)과 가지치기를 반복하게 된다.

$$w = \begin{cases} 0, & if\ |w| < \varepsilon \\ w, & otherwise \end{cases} \quad (1)$$

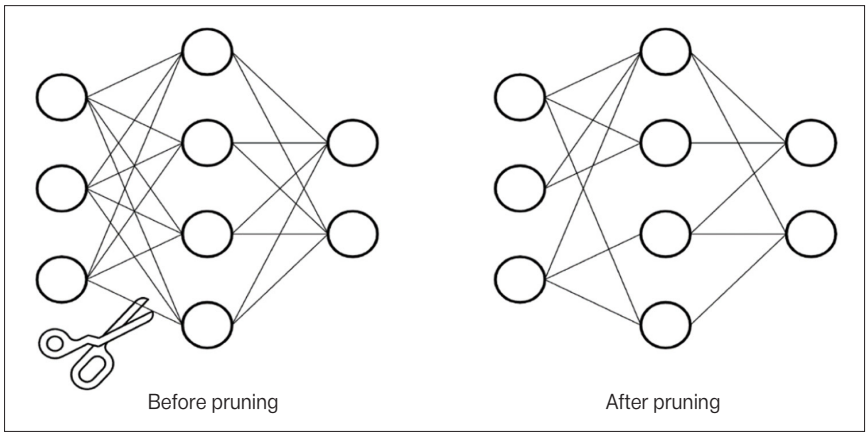
1) 데이터 기반 변환(Data Dependent Transformation)

데이터 기반 변환 기법은 기존의 가지치기 기법에서 가중치와 노드의 연결을 제거하는 기준인 가중치의 대소 관계인 부분은 동일하지만 이후 미세조정하는 손실함수(Loss function)에서 차이를 보인다[13].

$$L_{total} = L_{train} + \lambda L_{sparsity} \quad (2)$$

$$L_{sparsity} = \frac{|\boldsymbol{w}|}{\|\boldsymbol{w}\|} + \partial \frac{\|\boldsymbol{w}\|}{|\boldsymbol{w}|} \quad (3)$$

여기서 ∂은 학습 과정에서 $\frac{1}{3}\frac{|\boldsymbol{w}|}{\|\boldsymbol{w}\|} = \frac{\|\boldsymbol{w}\|}{|\boldsymbol{w}|}$에 맞게 값이



<그림 5> 가지치기 예시

설정된다. 특히 수식 (3)은 가중치 행렬의 L1-norm 대비 L2-norm의 크기로 판별하며, 해당 값의 최소화는 가중치 행렬의 대부분의 가중치 크기를 작게 만드는 것과 동시에 특정 값으로 에너지가 모인다는 것을 의미한다. 즉 NCTM의 데이터 기반 변환 기법은 수식 (2)와(3)처럼 모델의 성능은 최대한 유지하면서 가지치기를 더 효율적으로 할 수 있게 가중치 행렬의 희소화를 유도하는 과정이다.

2) Micro-structure 가중치 가지치기

Micro-structure 가중치 가지치기 기법은 가중치 행렬을 임의의 몇 개의 micro-block으로 분할 후 각 block 단위로 가지치기를 수행한다[14]. 해당 기법의 예시는 〈그림 6〉으로 나타내었으며, 절차는 다음과 같다.

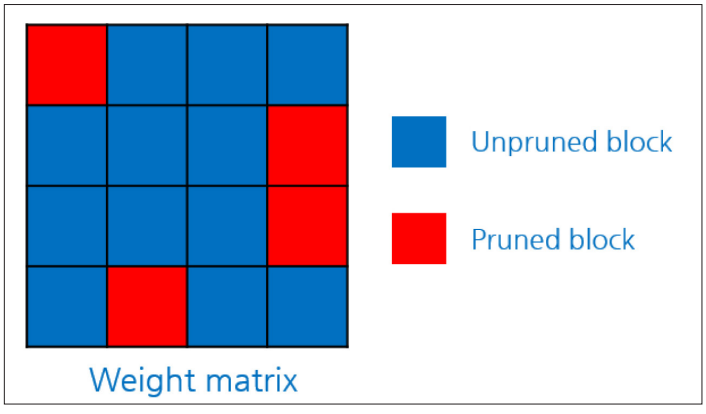
- 1) 가중치 행렬을 몇 개의 micro-block으로 나눌지 설정(〈그림 5〉의 경우 16개로 설정)
- 2) 각 block의 L2-norm을 계산 후, 해당 값의 오름차순으로 하이퍼 파라미터 q% 만큼 가지치기 (〈그림 5〉의 경우 25%로 설정)
- 3) 재학습을 수행 - 이때, 가지치기된 블록(〈그림 6〉의 빨간색 블록)의 가중치들은 고정하고, 나머지 블록의 가중치에 대해서만 가중치 업데이트

즉, Micro-structure 가중치 가지치기 기법은 micro-block 단위로 더 효율적인 가지치기를 통해 압축 효율을 증대시키는 기법이다.

3. Matrix Decomposition

행렬 분해 기법은 입력되는 인공신경망 모델 내 완전연결 계층과 합성곱 계층에 대한 가중치 행렬을 2개 이상의 행렬로 분해하여 가중치의 수 및 연산량을 줄이는 기법이다[15]. 인공신경망 모델 내의 각 계층의 유형에 따라 가중치들을 저장하는 행렬의 차원의 형태가 다르다. 예를 들어, 완전 연결층은 입력 노드의 수와 출력 노드의 수 곱의 형태인 2차원으로 저장되어 있으며, 합성곱 층은 필터의 크기와 입력 및 출력의 특징 맵의 수 곱 형태로 4차원으로 저장되어 있다. 따라서 NCTM에서의 행렬분해 기법은 완전 연결 및 합성곱 층에 낮은 순위 근사 기법(Low-Rank Approximation) 기법이 적용이 되지만, 각각의 층에서의 적용 방법이 다르다.

먼저, 완전연결 층의 낮은 순위 근사 방법은 식 (4)와 같이 2차원 행렬을 SVD 분해 기법 등을 이용하여 2차원 행렬 2개로 분해하는 기법이다[16].

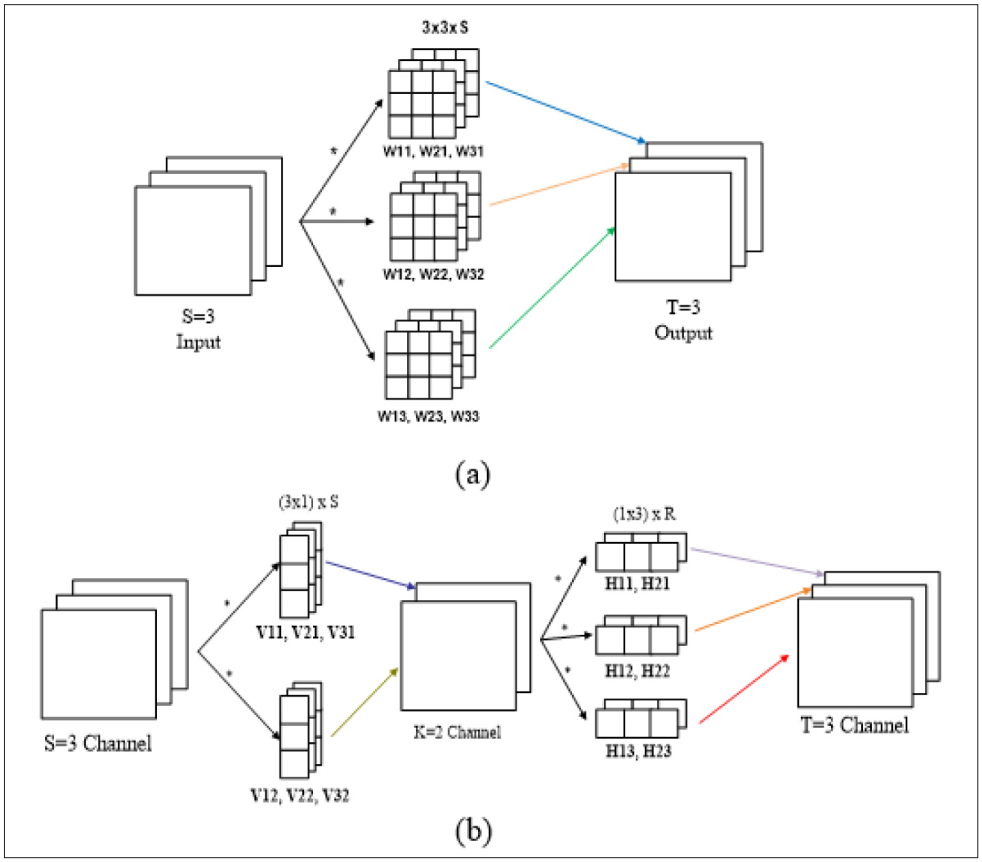


<그림 6> Micro-structure Pruning 예시

$$W_i = U_i V_i^T \quad (4)$$

여기서 W_i, U_i, V_i^T는 각각 $M \times N$, $M \times R$, $R \times N$ 크기를 가지며, W_i는 각 CNN 모델의 i번째 층의 가중치 행렬을 의미하며, U_i, V_i^T는 각각 i번째 층의 가중치 행렬 W_i로부터 분해되는 2개의 행렬을 의미한다. 즉 해당 기법은 하나의 가중치 행렬을 2개의 행렬로 분해함으로써 가중치 파라미터 수를 줄인다.

반면에, 합성곱 층의 경우는 가중치 행렬이 $T \times S \times D \times D$ 4차원 형태로 되어 있고, 여기서 T, S, D는 각각 입력 특징 맵 채널 수, 출력 특징 맵 채널 수, 필터 크기이다. 따라서, 4차원 행렬에서의 제안 기법의 적용이 안되기 때문에 4차원 행렬을 2차원으로 재배치하고 나서 적용을 하며[17], 〈그림 7〉은 해당 기법의 개요도를 나타낸다. 즉 가중치 행렬 W를 $W' = \begin{pmatrix} W_{11} & \cdots & W_{1t} \\ \vdots & \ddots & \vdots \\ W_{s1} & \cdots & W_{st} \end{pmatrix}$ 로 재배치하며, 여기서 W_{st}는 입력 특징 맵의 s번째 채널과 t번째 출력 특징 맵 사이의 필터를 의미한다. 재배치하고 나서, 식 (5), (6)과 같이 행렬을 분해하며, 여기서 K의 값은 중간 특징 맵 채널 수를 결정하는 rank 값이다. 〈그림 6〉은 필터의 크기가 3인 합성곱 층을 낮은 순위 근사 기법을 적용 전/후의 그림을 나타낸다. 즉 낮은 순위 근사 방법은 2D filter를 2개의 1D filter로 구성된 합성곱 층으로 분해하



<그림 7> (a) 원본 합성곱 층, (b) 낮은 순위 근사를 적용한 합성곱 층

며, 이 때 가중치의 수 및 해당 층에서의 연산량을 줄일 수 있음을 확인할 수 있다.

$$W' = VH \quad (5)$$

$$W_{st} \cong \sum_{k=1}^{K} V_{sk} H_{kt} \quad (6)$$

4. Unification

가중치 unification 기술은 가중치 행렬을 임의의 몇 개의 하위 블록(Sub Block)으로 분할 후 각 하위 블록 단위로 가중치를 하나의 값으로 통합하는 기술이다[18]. 즉, 특정 가중치에 대한 확률을 높임으로써 엔트로피 부호화의 효율성을 증대하는 기술이다. 해당 기법의 예시는 〈그림 8〉로 나타냈으며, 절차는 다음과 같다.

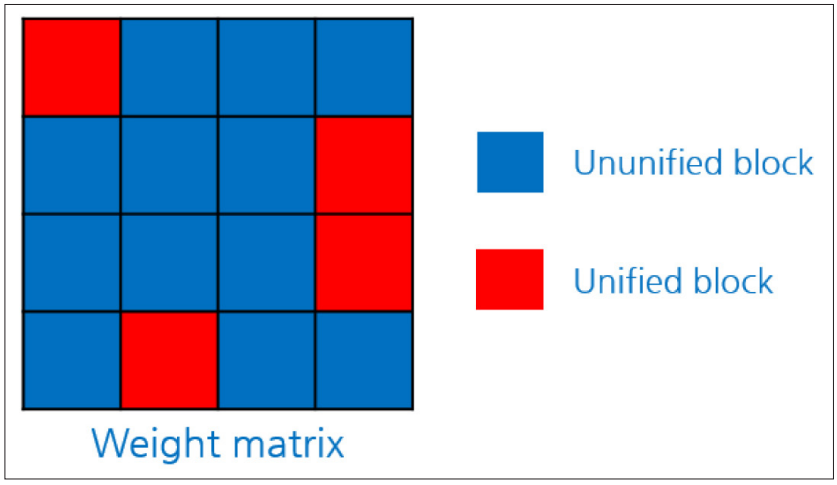
- 1) 가중치 행렬을 몇 개의 하위 블록으로 나눌지 설정(〈그림 7〉의 경우 16개로 설정)
- 2) 각 블록의 unify loss를 계산 후(수식7), 해당 값의 오름차순으로 하이퍼 파라미터 q%만큼 unification 수행(〈그림 7〉의 경우 25%로 설정)
  - $L_{unify}=max(B_{ij})-abs(B_{ij})$, 여기서 B_{ij}은 하위 블록을 의미
  - $UV_{ij}=mean(abs(B_{ij}))$, 여기서 UV_{ij}는 해당 하위 블록에서 통합할 값을 의미
- 재학습을 수행 - 이때, unification이 적용된 블록(〈그림 8〉의 빨간색 블록)의 가중치들은 고정하고, 나머지 블록의 가중치에 대해서만 가중치 업데이트

절차 2)의 unify loss 및 unification 값을 결정한 수식은 다음과 같다.

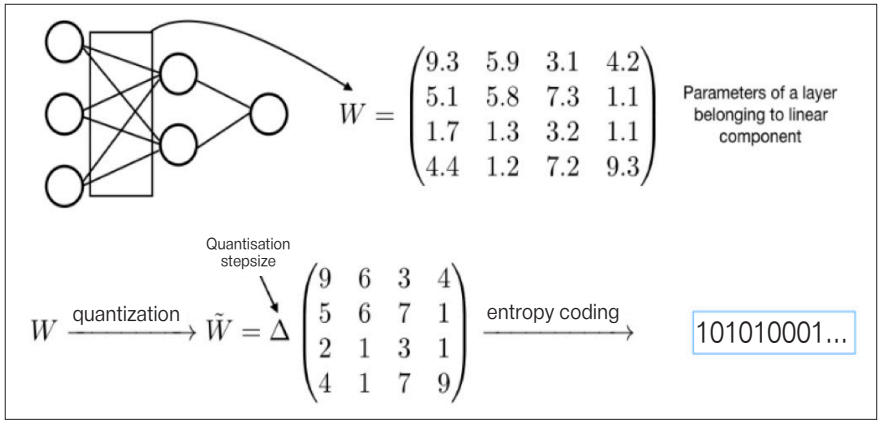
$$L_{unify}=max(B_{ij})-abs(B_{ij}), \quad (7)$$

$$UV_{ij}=mean(abs(B_{ij})), \quad (8)$$

여기서 B_{ij}, UV_{ij}은 각각 하위 블록, 해당 하위 블록에서 통합할 값을 의미한다. 즉 해당 기법은 해당 하위 블록들을 하나의 값으로 통합시켰을 때 발생하는 손실이 적은 블록들만 unification 과정을 수행하게 된다. 또한 하위 블록 내 양수와 음수가 모두 존재하므로, 추후 양자화 에러를 줄이기 위해 부호 값도 같이 전송하게 된다.



<그림 8> Unification 예시



<그림 9> NCTM에서의 균일 양자화 및 엔트로피 부호화 예시

5. Quantization

〈그림 9〉는 NCTM에서의 양자화 및 엔트로피 부호화에 대한 간단한 예시를 나타내었다. NCTM의 부/복화기의 입력으로는 앞서 언급한 전처리된 가지치기 및 행렬분해된 모델을 입력을 받는다. 전처리된 모델의 가중치행렬들은 32 비트 부동소수점(floating point)으로 표현되며, 이를 압축하기 위해 양자화로 각각의 가중치들을 정수(integer) 형태로 변환한다. 그리고 나서, 양자화 후에 나오게 되는 정수기반의 값들의 통계적 빈도에 따라 가변적인 길이의 코드로 표현하게 하는 엔트로피 부호화로 압축을 한다. NCTM에서 사용되는 양자화 기법은 크게 균일, 코드북 기반 비선형 양자화, 의존 양자화 등이 있다.

1) 균일/코드북 양자화

먼저, 균일 양자화는 수식 (9), (10)처럼 모든 가중치에 대한 계단 크기가 균일하게 적용이 되며, 수식은 다음과 같다.

$$Q=\Delta W \tag{9}$$

$$W' = Q/\Delta \tag{10}$$

여기서, Q는 양자화된 가중치 행렬, Δ는 step size, W와 W'은 각각 가중치 행렬과 복호화된 가중치 행렬을 의미한다.

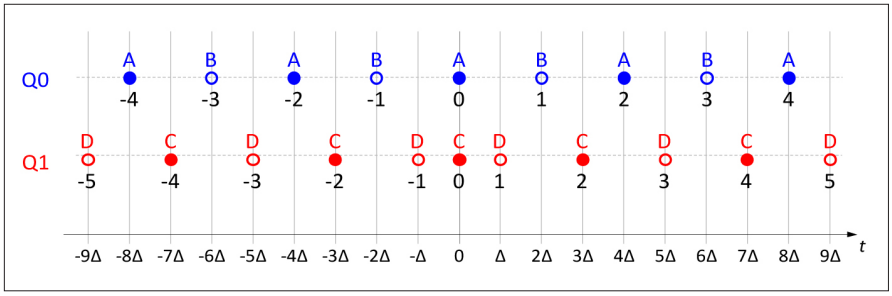
코드북 기반의 양자화는 수식 (11)처럼 가중치에 대한 계단 크기를 양자화 에러를 최소화하는 대표 값으로 구성하며, 수식으로 표현하면 다음과 같다.

$$C = \underset{c_1,\cdots,c_k}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{w_j \in c_i} |w_j - c_i|^2 \tag{11}$$

여기서 C는 코드북, k는 코드북의 인덱스, w_j, c_i는 각각 가중치 행렬에서의 i번째 가중치와 w_j에 상응하는 코드북 내의 대표 값을 의미한다.

2) 의존 양자화

NNC에서는 트렐리스 부호 기반 양자화로 알려진 의존 양자화(DQ, Dependent scalar Quantization)를 지원한다. 의존 양자화는 MPEG의 비디오 부호화 표준인 VVC(Versatile Video Coding)에서도 채택된 기법이며, 두 개의 양자화를 사용하는 방식으로 기존 양자



<그림 10> DQ 양자화기 예시

<표 5> MPEG NNC DQ 상태 전이 모델

current state	next state for		Quantizer (Q0/Q1) for current param.
	(k & 1) == 0	(k & 1) == 1	
0	0	2	Q0
1	7	5	Q1
2	1	3	Q0
3	6	4	Q1
4	2	0	Q0
5	5	7	Q1
6	3	1	Q0
7	4	6	Q1

화기법 대비 압축 효율에 대한 성능이 뛰어난 것으로 보고되었다[19].

의존 양자화는 두개의 양자화기와 상태 전이모델 같은 두 가지의 주요 요소로 구성되며, 두 개의 양자화기 예시는 〈그림 10〉으로 나타내었다. 여기서 상태 전이 모델은 현재 가중치의 양자화 상태(State)와 양자화 계수의 패리티(Parity)에 따라서 다음 가중치에 사용할 상태를 선택하는 알고리즘을 모델 형태로 표현하였다. VVC에서는 4개의 상태만을 정의하는 것과 다르게 NNC에서는 8개의 상태를 정의하였으며, NNC에서의 상태 전이 모델은 〈표 5〉와 같다[20].

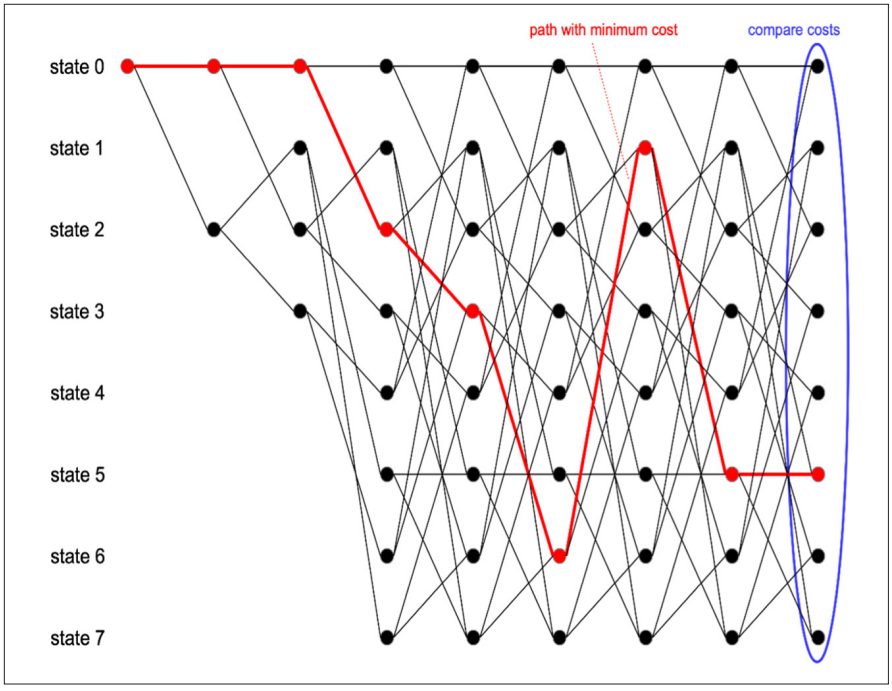
NNC에서의 의존 양자화를 위한 부호화 절차는 다음과 같으며, 그 예시는 〈그림 11〉로 나타내었다.

1) 첫 번째 가중치의 상태는 0으로 고정
2) 스캔 순서에 따라 가중치를 부호화
3) 각 양자화된 가중치에서는 각 subset(〈그림 9〉에서의 A,B,C,D)에 대해 양자화 인덱스를 결정
4) 각 가중치에서는 모든 트렐리스 노드(State 0~7)에 대해 이전 상태에선 온 경로 중 양자화 에러가 큰 경로를 제거(prune)
5) 마지막 가중치에서 각 트렐리스 노드 중 양자화 에러가 최소화되는 경로를 선택

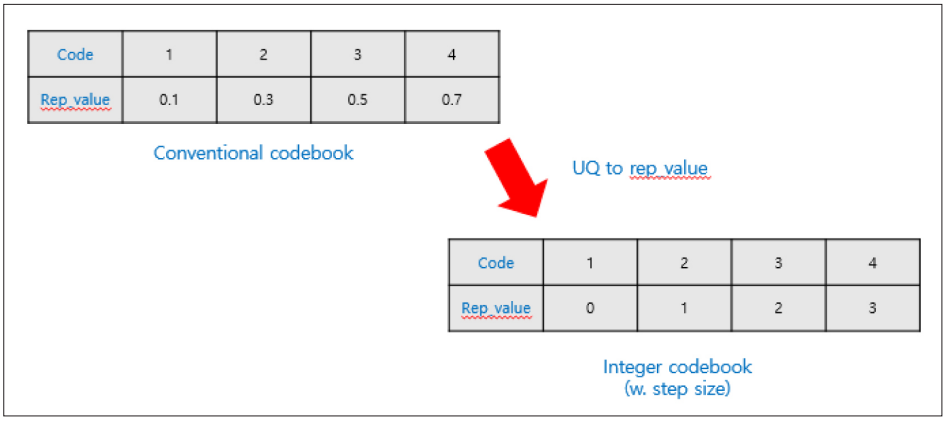
즉 해당 기법은 복호화를 위해 전송할 정보는 상태 전이 모델 및 마지막 가중치의 상태(state) 값 및 전체 가중치에 대한 양자화 인덱스로 구성된다.

3) 정수 기반 코드북 양자화

코드북 기반 양자화는 일반적으로 코드북 내 양자화 인덱스, 대표값으로 구성되어 있다. 그러나, 일반적으로 코드북 내 대표값은 32-bit 부동소수점으로 표현되기 때문에, 부호화 효율 측면에서 비효율적이다. 따라서 정수 기반 코드북 양자화는 이를 해결하기 위해 대표값에 균



<그림 11> NNC DQ 부호화 예시



<그림 12> 정수 기반 코드북 양자화 예시

일 양자화를 적용하여서 코드북의 구성요소를 모두 정수로 표현한다[21]. 〈그림 12〉는 정수 기반 코드북 양자화를 나타내었다.

정수 기반 코드북 양자화를 복호화하기 위해선 대표값의 양자화에 사용된 step size를 필요로 하며, HLS에 다음과 같이 표현(굵게 표시)되었다.

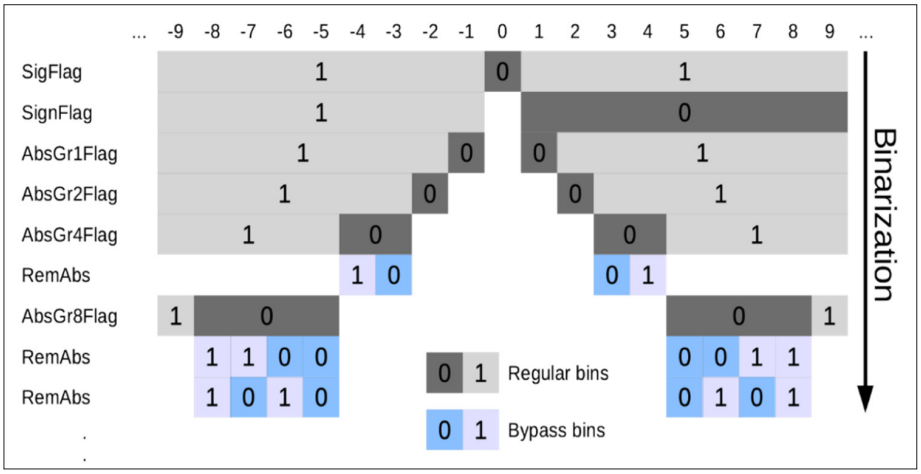
- NNR_compressed_data_unit_payload_type(uint5): NNR_NDU
  - 0: PT_INT32, 1: PT_FLOAT32, 2: CB:FLOAT32, **3: CB_INT32**

• 여기서 PT와 CB는 Parameter Tesnor, CodeBook을 의미한다.

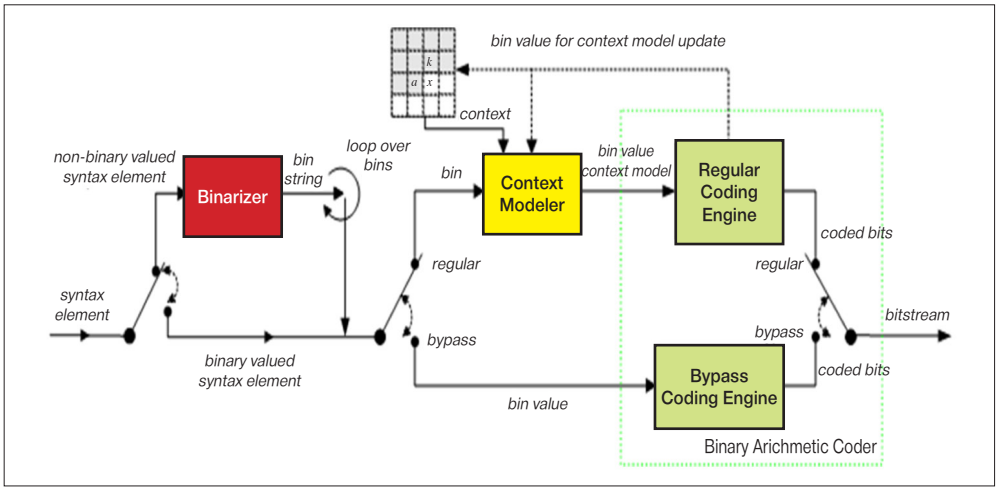
6. Entropy Coding (CABAC)

CABAC은 양자화된 가중치를 이진화(Binarization)를 한 뒤에 컨텍스트 모델을 기반으로 엔트로피 부호화를 수행한다[22]. 〈그림 13〉은 CABAC에서의 이진화 예시이다. 여기서 상단에 있는 값은 양자화된 정수 계수이며, 아래의 Flag 계수는 상응하는 양자화 계수의 전송 값을 의미한다.

SigFlag는 해당 양자화된 가중치 값이 0의 값을 가지는지에 대한 정보를 나타내며, SignFlag는 양자화된 계수의 부호 정보를 의미한다. 여기서 값이 "0"인



<그림 13> NCTM CABAC의 이진화 예시[22]



<그림 14> NCTM CABAC 흐름도

경우는 양수, “1”인 경우는 음수 값을 의미한다. 또한, AbsGrXFlag는 양자화된 계수의 크기를 나타내는 정보이며, 해당 flag 값이 1인 경우는 양자화된 계수의 절대값이 “X”보다 크다는 것을 의미한다. 마지막으로, RemAbs는 AbsGrXFlag==0 에서의 X값과 이전 X값 사이의 나머지 값을 표현하며, 이때의 부호화는 bypass 방식으로 전송된다. 그리고, 각 계층별로 AbsGrXFlag에서 전송되는 “X”값의 최대값을 나타내는 MaxNumNoRem 값을 보낸다.

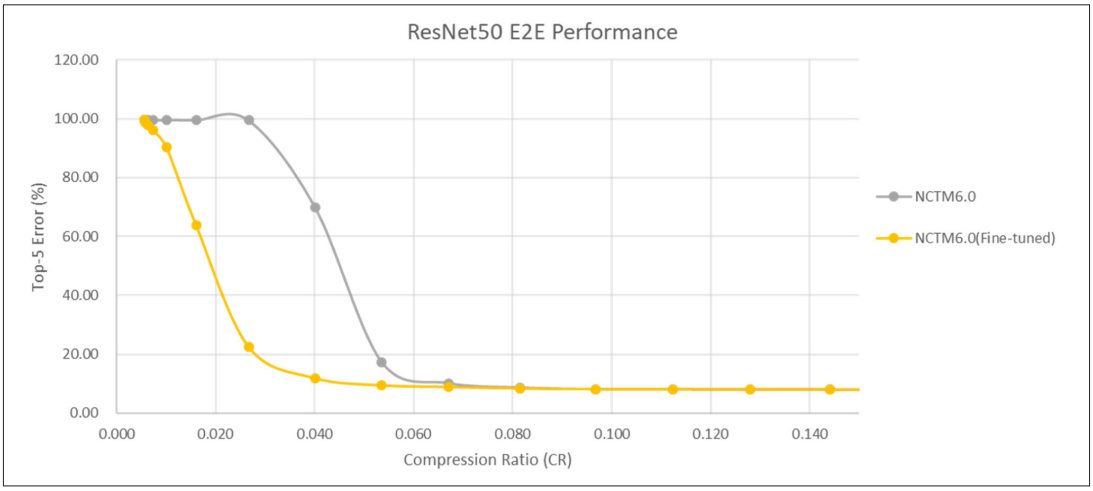
〈그림 14〉는 NCTM에서의 CABAC의 흐름도를 나타낸다. 앞서 이진화 과정에서 RemAbs 값을 제외한 나머지 값들은 Regular coding (context modeling)을 수행하며, RemAbs는 Bypass로 부호화한다.

7. 후처리 미세조정

인공신경망 모델 중 파라미터는 크게 가중치와, 바이어스(Bias) 값으로 구성되어 있다. NNC에서는 부/복호화 대상을 가중치 혹은 가중치와 바이어스 모두를 선택할 수 있다. 일반적으로 바이어스가 차지하는 메모리가 가중치 대비 매우 작기 때문에, 경우에 따라서 NNC에서는 전자를 선택하는 경우가 있다. 후처리 미세조정 기법은 엔트로피 부호화까지 다 끝난 가중치를 고정한 상태로 바이어스 값만을 미세조정을 통해 업데이트하여 부호화 성능을 높인다[23]. 즉, NCTM 내에 구현되지 않은 후처리 기술이므로, NCTM의 복잡도를 증가시키지 않는다. 〈그림 15〉에서도 알 수 있듯이, 약간의 바이어스의 미세조정만으로 부호화 성능이 크게 향상됨을 확인할 수 있다.

IV. High-Level Syntax

일반적으로 압축된 정보들을 시스템 단위에서 효율적으로 처리하도록 비트스트림을 구성하는 기술들을 High-level Syntax라 한다. 현재 MPEG NNC에서의 비트스트림은 NNR Unit 형태로 이루어져 있으며, 구성 예시는 〈그림 16〉과 같다[6]. 여기서 NNR Unit은 Unit의 크기를 나타내는 size와 metadata를 표현하는 header, 그리고 payload로 나눌 수 있다.

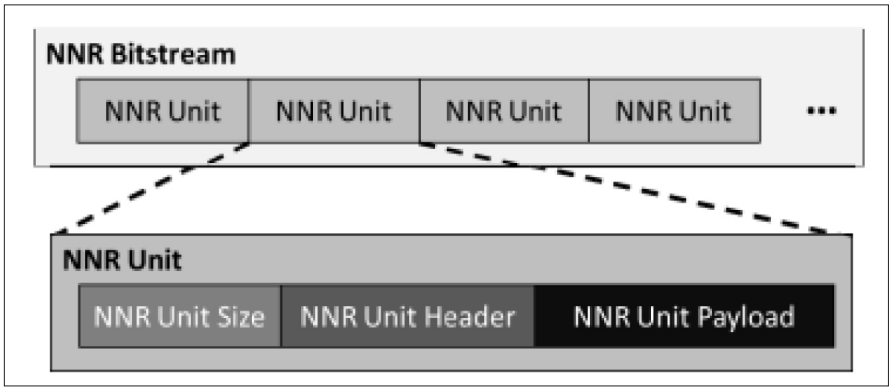


<그림 15> 후처리 미세조정 기법 성능

〈표 6〉은 NNR unit 내 타입(Type)에 대해 설명하였다. NNR_STR의 경우 NNR 비트스트림의 시작점을 알려주는 지시자(Indicator)이다. 또한, NNC 부호화 관점에서 해당 압축 기술들이 모델 전체 혹은 일부에 적용될 수 있으므로, 이를 구분할 수 있는 타입들을 정의하였다. MPS의 경우는 모델 전체에 해당하는 메타 데이터(Meta data)를 의미하는 반면에, LPS의 경우 하나의 계층 같은 모델의 일부에 해당하는 메타 데이터를 의미한다. 예를 들어, 가지치기가 계층 단위로 독립적으로 적용되었다면 관련 메타 데이터를 LPS 타입으로 전송해야 한다. 그 다음으로, TPL과 QNT는 NNC의 상호운용성을 나타내는 정보이며, 관련 메타 데이터를 표현하는 방식을 전송한다. 예를 들어, 일부 메타데이터 정보를 다른 상호 운용 포맷인 ONNX, NNEF을 사용한다면 해당 타입에서 정의하고 전송해야 한다. 마지막으로, NDU는 부호화된 가중치들을 인덱스 값 등을 전송하며, AGG는 NNR Unit 내 다수의 NNR unit이 존재하는지에 대한 지시자이다.

NNR Unit이 올바르게 복호화하기 위해서 다음과 같은 규칙들이 적용된다.

- 비트스트림은 반드시 NNR_STR로 시작
- MPS 타입은 반드시 NDU보다 앞서 정의되어야 함
- LPS는 다음 LPS 혹은 AGG NNR Unit이 끝날 때까지 유효
- TPL과 QNT는 가중치를 어떻게 저장할지에 대한 정보이므로 반드시 NDU보다 앞서 정의되어야 함
- TPL은 Tensorflow/keras, Pytorch, ONNX, NNEF 포맷 이상을 지원해야 함



<그림 16> NNR 비트스트림 구성 예시[23]

<표 6> MPEG NNC HLS 구성[6]

Identifier	Unit Type	Description
NNR_STR	NNR start unit	Bitstream start indicator
NNR_MPS	NNR model parameter set	NN global metadata
NNR_LPS	NNR layer parameter set	Partial NN metadata
NNR_TPL	NNR topology data unit	NN topology data information
NNR_QNT	NNR quantization data unit	NN quantization information
NNR_NDU	NNR compressed data unit	Compressed NN data
NNR_AGG	NNR aggregation unit	NNR unit with multiple NNR units

V. 결 론

지금까지 인공신경망 모델 압축 표준인 NNC의 기술의 개요 및 동향을 살펴보았다. NNC는 인공신경망 모델의 압축 뿐만 아니라 다양한 딥러닝 프레임워크 포맷과의 상호 운용하는 것을 목표로 하고 있고, 표준화 시작 대비 다양한 부호화 기술들이 채택되어 보다 높은 압축 성능 향상을 이끌어냈다. 또한, NNC에서는 상호 운용 측면에서의 폭넓은 활용을 위해 기존 상호 운용 포맷 및 표준인 NNEF와 ONNX와도 긴밀히 협력 중에 있고, 인공지능 학습 데이터 저작권 이슈를 해결하기 위해 나온 연합학습 시나리오를 고려한 INNC 표준이 진행 중에 있다. 현재 최종 국제표준안인 FDIS 단계로 마무리되는 단계에 들어섰으며, 앞으로 다양한 인공신경망 응용 측면에서의 활용도가 기대되는 표준이라고 하겠다.

참 고 문 헌

[1] S. Han, et al, "Deep Compression: Compressing Deep Neural Net-works with pruning, trained quantization and Huffman coding," In Proc. CVPR, Jun. 2015.

[2] https://www.tensorflow. org/

[3] https://pyotrch.org/

[4] Neural Network Exchange Format (The Khronos NNEF Working Group), [Available at Online] https://www.khronos.org/registry/NNEF/specs/1.0/nnef-1.0.3.pef

[5] Open Neural Network Exchange, [Available at Online] https://github.com/onnx/onnx/blob/ master/onnx/onnx.proto

[6] W. Bailer, et al, "Text of ISO/IEC FDIS 15938-17 Compression of Neural Networks for Multimedia Content Description and Analysis," ISO/IEC JTC1/SC29/WG4, N20331, Jun. 2021.

[7] W.Bailer, et al, "Draft Call for Proposals on Incremental Compression of Neural Networks for multimedia content description and analysis," ISO/IEC JTC1/SC29/WG11, N19515, Jul. 2020.

[8] W. Bailer, et al, "Use cases and requirements for Compressed Representation of Neural Networks," ISO/IEC JTC1/SC29/WG11, N17924, Oct. 2019.

[9] W. Bailer, et al, "Evaluation Framework for Compression of neural networks for multimedia content description and analysis," ISO/IEC JTC1/SC29/WG11, N18575, Jul. 2019.

[10] S. Niknam, et al, "Federated learning for wireless communications: Motivation, opportunities, and challenges," IEEE Communications Magazine, 58(6): 46-51, 2020.

[11] H. Moon, et al, "Test Data for Incremental NNR: Federated Learning for Medical Applications (UC 14A)," ISO/IEC JTC1/SC29/WG11 m55054, Oct. 2020.

[12] W. Bailer, et al, "Test Model 6 of Compression of Neural Networks for Multimedia Content Description and Analysis," ISO/IEC JTC1/SC29/WG11, N19765, Oct. 2020.

[13] C. Aytekin, et al, "Response to the Call for Proposals on Neural Network Compression: Training Highly Compressible Neural Networks," ISO/IEC JTC1/SC29/WG11, m47379, Mar. 2019.

[14] W. Jiang, et al, "[NNR] CE1 result: micro_structured_pruning," ISO/IEC JTC1/SC29/WG11, m55022, Oct. 2020.

[15] S. Lin, et al, "Holistic CNN Compression via Low-Rank Decomposition with Knowledge Transfer," IEEE transaction on pattern analysis and machine intelligence, 41(2): 2889-2905, 2019.

[16] M. Jaderberg, et al, "Speeding up Convolutional Neural Networks with Low Rank Expansions," In Proc. CVPR, Jun. 2014.
[17] H. Moon, et al, "Response to the Call for Proposals on Neural Network Compression: Quantization and Low-Rank Approximation," ISO/IEC JTC1/SC29/WG11, m47704, Mar. 2019.
[18] W. Jiang, et al, "NNR non-CE1 related: Data-dependent transformation for highly unified Neural Networks," ISO/IEC JTC1/SC29/WG11, m52631, Jun. 2020.
[19] H. Schwarz, T. Nguyen, D. Marpe and T. Wiegand, "CE7: Transform Coefficient Coding and Dependent Quantization (Tests 7.1.2, 7.2.1)", JVET-K0071, 2018.
[20] P. Hasse, et al, "[NNR] CE2-related: Dependent scalar quantization for neural network parameter approximation," ISO/IEC JTC1/SC29/WG11, m52358, Jun. 2020.
[21] P. Hasse, et al, "[NNR]: HLS adaptation for integer codebook representation," ISO/IEC JTC1/SC29/WG11, m54937, Jun. 2020.
[22] S. Wiedemann, et al, "DeepCABAC: Context-adaptive Binary Arithmetic Coding for Deep Neural Network Compression," International Conference on Machine Learning (ICML), May. 2019.
[23] H. Kirchhoffer, et al, "Overview of the Neural Network Compression and Representation (NNR) Standard," IEEE transcation on circuits and systems for video technology, 32(5): 3203-3216, 2022.
[24] W. Bailer, et al, "Common Test Conditions for Incremental Neural Network Compression," ISO/IEC JTC1/SC29/WG4, N0123, Aug. 2021.

필 자 소 개

문 현 철

- 2018년 : 한국항공대학교 항공전자정보공학과 학사
- 2020년 : 한국항공대학교 항공전자정보공학과 석사
- 2021년 ~ : 한국전자기술연구원 연구원
- 주관심분야 : 인공지능 기반 미디어 신호 처리, 인공지능 경량화

정 진 우

- 2011년 : 연세대학교 전기전자공학과 박사
- 2015년 : 삼성전자 VD사업부 책임연구원
- 2016년 ~ : 한국전자기술연구원 책임연구원
- 주관심분야 : 인공지능 기반 미디어 신호 처리, 수중 영상 처리

필 자 소 개

김 성 제

- 2011년 : 연세대학교 전기전자공학과 박사
- 2015년 : 삼성전자 S.LSI 책임연구원
- 2015년 ~ : 한국전자기술연구원 책임연구원
- 주관심분야 : 미디오 신호처리, 인공지능