Journal of The Institute of Electronics and Information Engineers Vol.61, NO.1, January 2024 https://doi.org/10.5573/ieie.2024.61.1.27
ISSN 2287-5026(Print) / ISSN 2288-159X(Online)

논문 2024-61-1-4

디바이스 적응형 신경망 생성 및 배포 구현

(Implementation of Device-adaptive Neural Network Generation and Deployment)

김 선 태*, 조 창 식**

(Seon-Tae Kim© and Changsik Cho)

요 약

디바이스의 성능이 높아짐에 따라 인공지능 응용도 급속도로 다양한 디바이스에 적용되고 있다. 디바이스에 적합한 빠른 신경망 개발은 기업의 경쟁력을 좌우하여, 이를 지원하기 위해 개발함과 동시에 바로 적용이 가능한 프레임워크인 MLOps(Machine Learning Operations)이 클라우드 서비스를 제공하는 글로벌 기업들에 의해 제공되고 있다. 하지만, 현재 제공되는 프레임워크는 고성능 자원을 유료로 제공하는 클라우드 서비스를 이용하였으며, 개발자가 원하는 타겟 디바이스에 최적인 신경망을 개발하는데 한계가 있었다. 본 논문에서는 추론 신경망이 구동되는 디바이스를 고려하여 최적의 신경망 및 응용이 개발될 수 있도록 프레임워크를 구현하였다. 제안하는 프레임워크는 기존의 프레임워크에 비해 개발자가 인공지능에 다소 전문성이 부족하더라도 쉽고 빠르게 신경망을 개발할 수 있는 기능을 포함하였다. 그리고 개발된 프레임워크는 GitHub에 공개하여 관심있는 개발자 및 산업 적용에 애로점이 있는 개발자에게 소스코드를 제공하고 있다.

Abstract

As device performance improves, artificial intelligence applications are also rapidly being applied. Rapid neural network development suitable for devices determines a company's competitiveness, and to support this, MLOps (Machine Learning Operations), a framework that can be developed and applied immediately, is being provided by global companies that provide cloud services. However, the currently provided framework uses a cloud service that provides high-performance resources for a fee, and there are limitations in developing a neural network optimal for the target device desired by the developer. In this study, a framework was implemented so that optimal neural networks and applications can be developed by considering the devices on which the inference neural network runs. Accordingly, compared to existing frameworks, it includes a function that allows developers to develop neural networks easily and quickly even if they have less expertise in artificial intelligence. Additionally, the developed framework is made public on GitHub, providing source code to interested developers and developers who face difficulties in industrial application.

Keywords : Device-adaptive, Neural network generation, Neural network deployment, MLOps, Neural network model recommendation

Ⅰ. 서 론

스마트폰 등 디바이스들의 연산 성능이 고속화되면서, 수많은 인공지능 응용 서비스들이 다양한 추론 디바이스에서 제공되고 있다. 즉, 이미지 화질 보정 기능, 슈퍼 해상도 기능, 카메라 영상 감지, 언어와 음성 지원 기능에 대한 다양한 인공지능 응용들이 기존 고성능의 PC 기반 디바이스를 넘어서 산업형 혹은 휴대형 디바이스의 컴퓨팅 연산 고성능화로 다양한 디바이스에서 구동되고 있다[1]. 따라서 관련 추론 디바이스에 맞는 신경망 모델 확보를 통해 응용 서비스 개발이 활발히 진행되고 있다.

*평생회원, **비회원, 한국전자통신연구원 인공지능컴퓨팅연구소(AI Computing Research Lab., ETRI)
© Corresponding Author(E-mail : stkim10@etri.re.kr)
※ 이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (2021-0-00766, 신경망 응용 자동생성 및 실행환경 최적화 배포를 지원하는 통합개발 프레임워크 기술개발).
Received ; November 7, 2023 Revised ; November 22, 2023 Accepted ; November 30, 2023

하지만, 다양한 디바이스에서 구동되는 신경망 모델은 디바이스 성능과 응용 종류에 따라 응용 개발자가 그동안 경험했던 노하우에 따라 신경망 모델이 결정하고 관련 모델 복잡도를 결정하였다. 이를 위해서는 응용에 사용될 데이터 종류 및 규모를 파악해야 하며, 다양한 종류의 신경망에서 수많은 학습 경험을 통한 지식 습득으로 최적의 신경망 모델을 개발하였다. 여기에 추가적으로 학습 하이퍼 파라미터를 설정 및 결정하고 최적 신경망을 생성하게 된다.

한편, 위와 같은 반복적인 작업을 개발자의 경험에 의해 수행하는 것보다 다수의 GPU 가속기를 갖는 클라우드 서버 등 학습 디바이스의 고성능화에 따른 연산량으로 극복하는 기술이 제시되었다[2, 3]. 즉, 폭넓은 검색공간을 제시하고 수많은 변수를 적용하여 최적의 신경망을 도출해 내는 자동생성 알고리즘이 출현하였다. 자동생성 알고리즘은 인간의 노하우보다는 뛰어난 성능을 얻었지만, 이런 결과물을 얻기 위해서는 상당한 연산량이 요구되었다.

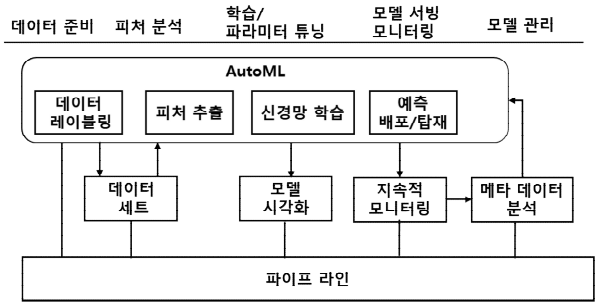


그림 1. 기존 MLOps 구조
Fig. 1. Previous MLOps structure.

인공지능 응용은 기존의 응용의 적용과는 달리 지속적인 배포 관리가 요구되는데, 신속한 개발을 위해서 MLOps(Machining Learning Operations)[4] 기술을 적용한다. 그림 1에서 보여준 바와 같이, 데이터 준비부터 신경망 생성, 신경망 배포, 신경망 관리 등의 모듈로 구성되고 이를 파이프라인 형식으로 수행하여 간단히 신경망을 생성하고 신경망을 최적화하는 기능을 제공하고 있다. 클라우드 서비스를 제공하는 글로벌 기업의 Google Vertex AI[5], MS Azure Machining Learning[6], Amazon SageMaker[7]와 공개 프로젝트인 Kubeflow[8] 등이 대표적인 것들이다.

위에서 언급한 바와 같이, 현재 인공지능 응용이 PC 및 스마트 폰, 산업 분야 디바이스 등 다양한 디바이스에 적용되고 있는데, 이는 학습 서버의 고성능화로 AutoML 기술의 적용으로 최적 신경망 생성이 용이해졌으며, 응용 서비스의 빠른 적용을 위해 MLOps의 프레임워크를 제공하여, 인공지능 서비스 개발 및 품질 유지의 효율성을 제공하고 있기 때문이다.

하지만, 다양한 디바이스에서 원하는 사양에 최적의 신경망을 신속하게 생성하기 위해서는 주어진 입력 사항을 잘 분석하여 신경망 모델과 복잡도를 예측하는 기술이 필요하게 된다. 이를 위해서 본 논문에서는 개발자가 원하는 인공지능 태스크(응용 서비스)와 디바이스 사양을 바탕으로 디바이스에 최적인 신경망 모델을 추천하는 기능을 추가하였으며, 인공지능에 대한 경험은 미흡하여 응용개발에 힘들어하는 산업 분야에 특화된 개발자를 위해서 손쉽게 신경망을 생성할 수 있는 프레임워크를 개발하였다.

본 논문은 이러한 산업환경의 디바이스에 적응적 최적 신경망을 생성하는 기술을 개발하는 것으로, 구성은 II장에서 산업 현장을 위한 MLOps에서 산업현장 개발자가 자신에 맞는 응용을 보다 쉽게 개발할 수 있도록 필요한 추가적인 기능을 제안하고, III장에서는 구현한 내용과 결과를 서술하고, IV장에서는 결론을 맺는 순으로 서술하였다.

II. 산업 현장을 위한 MLOps

기존의 MLOps는 신경망 생성을 하고 신경망 배포를 각각 독립적으로 수행하여 디바이스에 최적화된 신경망을 만드는데 애로점이 있었다. 본 연구에서는 신경망 모델을 생성하고자 할 때, 추론 디바이스 성능을 고려하여 신경망을 생성하되 디바이스의 구동환경(OS 및 Pytorch, TensorRT, TVM 등)에 맞게 다시 한번 최적화를 수행하는 프로세스를 거쳐 디바이스 최적 신경망을 생성 및 구동할 수 있는 프레임워크를 구현하였다.

그림 2는 기존 MLOps의 기능을 포함하면서, 추가되어야 할 모듈과 보완이 되어야 할 모듈에 대해서 하이라이

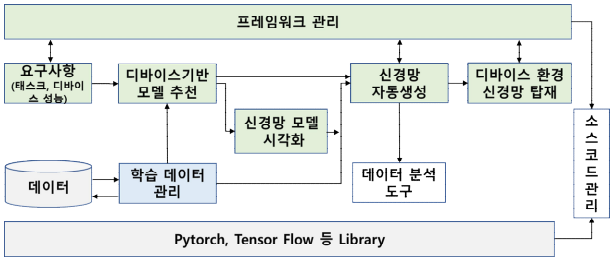


그림 2. 제안하는 프레임워크 구조
Fig. 2. Proposed framework structure.

트로 표시하였다. 다음은 주요 모듈에 대해서 구체적인 특징 및 기능을 서술하였다.

1. 제안하는 프레임워크 구조 및 관리

제안하는 MLOps 프레임워크는 Docker 기반 MSA (MicroService Architecture)로 각 모듈이 연동되며, 각 모듈은 파이프라인 구조로 실행이 되도록 설계되었다. 즉, 입력 처리 모듈, 신경망 모델 추천, 신경망 생성 및 신경망 배포의 구성 요소들을 각각의 프로세스로 설정하여 순차적으로 처리할 수 있도록 설계하였다. MSA 장점에 맞게 각 모듈은 독립적으로 구동 및 디버깅이 가능하며, 대체하고자 하는 기존 모듈들은 독립적으로 교체될 수 있도록 하였다.

그리고, 향후에 구성 모듈들을 추가할 경우에 다른 모듈에 영향이 미치지 않도록 설계하였으며, 신경망 생성의 경우 같은 위치에 다양한 신경망 모델을 배치할 수 있도록 하였다. 즉, 이미지 분류 신경망과 객체 탐지 신경망이 각 모듈로서 존재하지만, 수행 시에는 필요 모듈이 선택되어 구동되도록 하였다.

2. UI 기반 입력 정보 명세

제안 프레임워크는 먼저 신경망 생성 프로젝트를 만들고 사용자의 입력 사항을 UI 기반으로 관련 정보를 공유 저장소에 저장하고 이 정보를 구성모듈의 입력으로 주고 다음 모듈은 전달받은 정보를 바탕으로 구동되는 파이프라인 형식으로 구현하였다. 신경망 생성을 위해서는 기본적으로 요구되는 정보가 있는데, 신경망의 종류를 결정하는 Task 타입(예:이미지 분류 vs 영상 객체 탐지)에 대한 정보와 디바이스의 연산 처리 속도에 대한 성능이다. 또한, 신경망 배포를 위해서는 추론 디바이스에서의 HW 및 SW 구동 환경이 제공되어야 하는데, 이 또한 입력 UI를 통해 구현하였다. 그림 3은 입력 정보를 입력하는 것으로, 데이터 타입 및 디바이스의 사양을 결정하는 것을 보여주고 있으며, 타겟 디바이스 설정에서 SW 환경을 설정 및 탑재를 위한 고급 옵션들이 포함되어 있다.

3. 신경망 생성 모듈

이미지 분류의 경우 현재 SOTA(State-of-the-Art) 모델보다는 산업환경에서 기본적으로 많이 사용되는

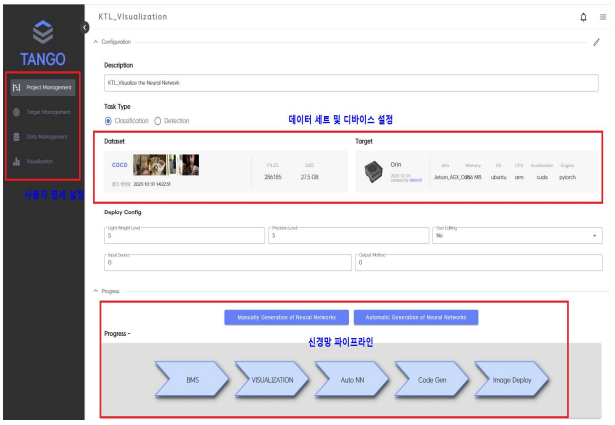


그림 3. 제안 프레임워크에서 태스크 타입, 데이터 세트와 타겟 디바이스 설정 UI

Fig. 3. Task type, data set and target device setting UI in proposed framework (TANGO*[9]).

Resnet과 Densenet기반으로 설계되었으며, 디바이스 성능에 따른 추론 시간을 고려하여 5개의 복잡도를 제공하도록 하였다. 앞서 언급했듯이, 제안 프레임워크 구조가 MSA 방식으로 되어 있어, 향후에 응용 및 데이터에 맞게 신경망 모델을 추가하거나 교체할 수 있도록 하였다.

영상의 객체 탐지의 경우에는 정확도뿐만 아니라 빠른 추론 시간을 고려하여 one-stage로 처리하는 Yolo 모델을 기반으로 하여 구현하였다. 신경망 생성을 위한 입력 형식 파일로 YAML 파일을 받아서 학습을 수행하였으며, 다음의 3가지 방식으로 기능을 확장하였다. 첫째로, 기존 Yolo 모델보다 성능 향상을 위해서 Neck 구조에서 피처의 크기를 재조절하고 3D 공간으로 확장하는 알고리즘을 추가(Bag of Specials)하여 정확도를 높였다. 두 번째로, 한 번의 학습으로 다양한 디바이스에 적합한 신경망 모델을 생성하는 Supernet 기반 NAS(Neural Architecture Search) 구조로 설계하여, 다수의 스마트 폰 디바이스를 대상으로 한번 학습으로 각각 최적 신경망이 생성되도록 하였다. 마지막으로 모델별로 최적의 학습 파라미터가 존재하는데, 본 연구에서도 학습율 등 하이퍼 파라미터를 최적화하여 학습의 속도와 정확도를 높였다.

신경망 모델은 제공하고자 하는 응용(task)과 사용하는 데이터에 따라 다양하게 변경이 될 수 있다. 본 논문에서 제안하는 프레임워크에서는 산업 현장에서 많이 적용되는 카메라에서 취득한 이미지 데이터를 기반으로, 이미지 분류와 객체 탐지에 관한 신경망 생성 모듈을 구현하였다.

* TANGO : Target Aware No-code neural network Generation and Operation framework 약자로 노코드 기반으로 디바이스에 최적화된 신경망을 생성하는 프레임워크임

4. 디바이스 적응형 신경망 모델 추천

기본적으로 신경망을 생성하기 위해서는 신경망 모델을 결정하고 최적화 작업을 수행한다. 한번 학습을 통해 생성된 신경망은 디바이스에 맞게 생성되지 않으면 변경하기 어려우며, 학습 환경이 바뀌면 재학습을 하게 된다. 이런 문제를 해결하기 위해서 한번 학습으로 다양한 디바이스에 맞는 신경망을 생성하는 One-shot 학습이나 수많은 NAS 검색 공간에 최적의 신경망을 찾는 방법이 시도되었다. 하지만, 이 방법은 최적에 가깝지만, 디바이스에 맞는 최적화된 신경망을 생성하는데 애로점이 있었고 수많은 Trial-error를 통한 신경망을 생성하게 된다. 이에 본 논문에서는 신경망을 생성하기 전에 사용자가 원하는 인공지능 응용과 디바이스 성능에 따라 신경망 모델을 결정하고 복잡도를 예측 결정하는 방법을 제시하였다. 신경망 종류는 사용되는 학습 데이터나 task 타입을 통해 결정하였으며, 복잡도는 디바이스의 성능에 따라 결정하였다.

표 1. 디바이스 적응형 신경망 모델 추천
Table 1. Device adaptive neural network model recommendation.

타겟 디바이스		신경망 모델 추천	
등급	세부 디바이스	객체 탐지 (6 종류)	이미지분류 (5 종류)
Cloud	Cloud	Yolov7_E6E	Resnet203
K8S (Kubernetes)	K8S_PC	Yolov7_W6	Resnet152
	K8S_Jetson_Nano	Yolov7_Tiny	Resnet34
PC	PC_Server	Yolov7_E6	Resnet152
	PC	Yolov7_W6	Resnet152
On Device	Jetson_AGX_Orin	Yolov7_W6	Resnet101
	Jetson_AGX_Xavier	Yolov7_X	Resnet50
	Jetson_Nano	Yolov7_Tiny	Resnet34
	Galaxy_S22	Yolov7_Tiny	Resnet34
	Odroid_N2	Yolov7_Tiny	Resnet34

표 1은 이미지 분류나 영상의 객체 탐지에 대해서 디바이스 성능에 따른 신경망 모델의 추천 예를 보여준다. 다양한 성능을 갖는 10개의 디바이스에 대해서 4가지 등급으로 분류하였으며, 이미지 분류와 객체 탐지에 대해서 각각 5개의 신경망 모델과 6개의 신경망 모델을 제시하였다. 신경망 모델 추천은 학습의 횟수가 증가함에 따라 경험 데이터가 누적되고 이 정보를 바탕으로 추천 모델 학습을 하게 되면 효율적인 추천 알고리즘이 가능하다. 이를 위해서 그림 2의 학습 데이터 관리 모듈을 추가하였다.

5. 신경망 시각화

전문가인 개발자가 신경망의 생성할 경우, 신경망 모델 추천 모듈에서 자신의 전문가적 경험치를 반영할 수 있는 단계가 필요하다. 이런 경우, 신경망 모델 추천 모듈에서 추천하는 신경망 구조를 수정하거나 파라미터를 직접 수정해야 하는데, 이를 위해 신경망 시각화 모듈을 추가하였다. 이 모듈에서는 추천된 신경망 모델을 시각화하고 파라미터를 변경할 수 있도록 편집 기능을 제공하여, 개발자가 손쉽게 신경망 모델을 수정하고 이를 학습에 적용하도록 하였다.

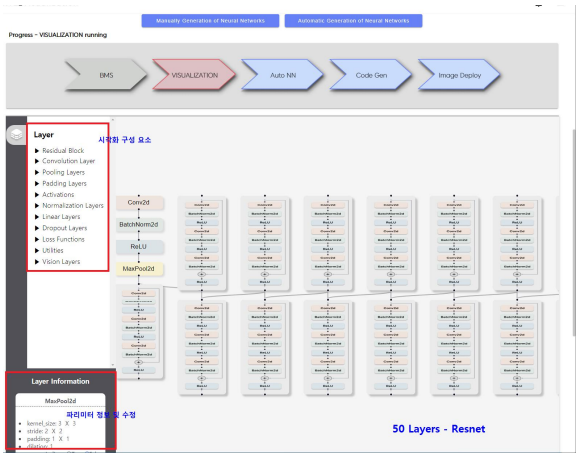


그림 4. Resnet-50 신경망의 시각화
Fig. 4. Visualization of Resnet-50 neural network.

그림 4는 Resnet-50에 대한 신경망 모델을 시각화해주고 각 레이어의 파라미터를 수정할 수 있는 UI 인터페이스를 보여주고 있다. 신경망 시각화에서는 신경망을 구성하는 레이어 구성요소를 갖추고 있어 원하는 모듈을 드래그앤드랍 방식으로 새로 만들거나 추가할 수 있다. 내부 모듈을 클릭하면, 수정할 수 있는 파라미터를 보여주어 편집할 수 있는 기능을 제공하고 있으며, 복잡한 신경망의 경우 블록화 기능을 제공함으로써, 구조의 추상화를 통해 바로 신경망 구조를 확인할 수 있도록 하였다.

본 논문의 신경망 시각화에서는 신경망 모델의 백본망(backbone network)으로 VGG, Resnet 및 Densenet이 지원되며, 넥망(neck network) 및 헤드를 포함하는 신경망에서는 Yolo 모델의 시각화 기능을 지원한다. 그림 4에서는 Resnet50을 시각화한 것으로 16개의 블록에 블록당 3개의 Conv가 포함되어 있으며, 블록 외부에 2개가 레이어가 존재하는 구조를 보여주고 있다.

Ⅲ. 구현 및 결과

제안한 프레임워크를 보여주는 그림 5는 고성능 PC 혹은 서버 기반에서 구동되는 프레임워크의 실제 실행되는 화면이다. 응용 서비스 선택을 위한 Task Type을 입력하고, 신경망 학습을 위한 데이터 세트를 입력한 후 신경망이 구동될 추론 디바이스 환경을 입력하는 구조이다.

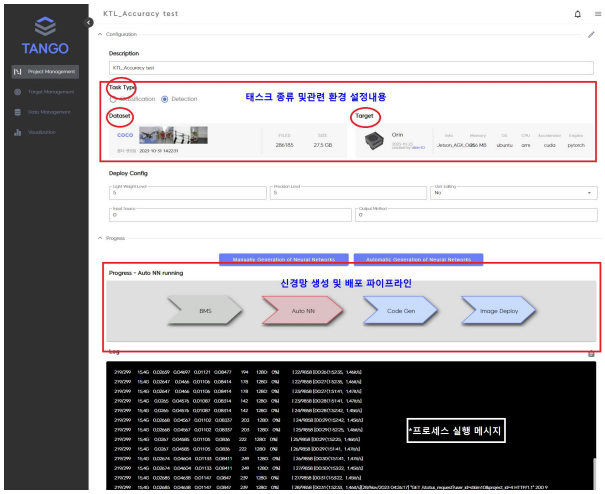


그림 5. Jetson Orin 디바이스를 위한 신경망 생성 프레임워크 구동 장면

Fig. 5. Running scene of the neural network generation framework for Jetson ORIN device.

여기서 사용자에 의해 입력되는 정보는 프로젝트 내 모든 모듈이 사용할 수 있도록 공유 파일로 저장되며, 이 저장 파일을 해독하여 신경망 모델 추천에서는 신경망 모델과 복잡도 기반으로 신경망이 추천되고 이 모델을 입력으로 하여 신경망 학습이 수행된다. 학습이 종료되면 출력 결과인 가중치 파일을 이용하여 추론 디바이스 환경에 맞게 신경망을 양자화, 추론 엔진에 맞는 형식변환 등의 최적화를 거친 후에 실제 추론 디바이스에 최적화된 신경망이 탑재되고 구동이 된다. 파이프라인 내 각 모듈의 구동 중에는 각 모듈에서의 구동 메시지를 콘솔창으로 출력하여 해당 모듈의 프로세스 구동 상태를 확인할 수 있다. 사용자의 편의성 제공을 위해서 수동으로 한 스텝씩 모듈별로 실행을 할 수 있으며, 사용자가 원하는 환경 정보 입력 후 자동적으로 신경망 모델을 추천하고 생성 및 배포까지 실행할 수 있도록 구현하였다.

신경망 모델 학습은 Nvidia RTX 4090 GPU 2개를 이용하여 다양한 Yolo 모델을 학습하였는데, Yolov7-x 모델의 경우, 300 Epochs을 학습 수행하는데 14일 정도 걸렸다. 정확도는 Yolov7 공개 프로젝트[10]와 같이 5천장의 Evaluation 데이터와 2만 장의 Test 데이터 세트를 이용해서 구하였는데, 각각 0.1% 이내로 큰 차이가 없었다.

표 2. Nvidia Jetson AGX ORIN 디바이스에서 추론 성능

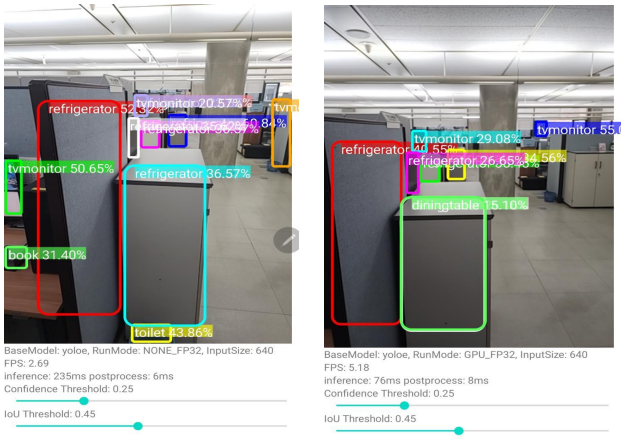
Table 2. Inference performance on Nvidia Jetson AGX ORIN devices.

입력 해상도 & 엔진 / 신경망 모델	해상도(320)		해상도(640)	
	Pytorch (ms)	Tensor RT (ms)	Pytorch (ms)	Tensor RT (ms)
Yolov7-Tiny	16.21	4.19	17.29	9.62
yolov7	25.3	9.39	27.4	26.17
yolov7-x	34.25	14.62	54.05	43.52
ylov7-w6	33.22	9.79	55.4	23.81
yolov7-e6	40.02	13.87	46.95	42.34
yolovy-d6	45.66	16.59	42.12	35.08
yolov7-e6e	65.49	20.97	82.92	52.16

본 실험에서는 Yolov7의 다양한 신경망 모델 기반으로 신경망을 생성하고, Jetson Orin 디바이스 타겟에서 추론 실행 속도를 살펴보면 표 2와 같았다. 해상도 입력을 320과 640으로 했을 때와 Pytorch 및 Nvidia 최적 환경인 TensorTR로 최적화했을 경우를 비교하였다. Nvidia 디바이스에서 제공하는 연산기능으로 최적화를 수행했을 때 현저한 연산 속도 향상이 있었으며, 해상도 320 입력 데이터는 모두 TensorRT로 최적화된 신경망 모델을 적용하며 실시간 추론이 가능하며, 해상도를 640으로 했을 때는 Yolov7-W6 정도의 신경망 모델을 적용해야 실시간 추론이 가능함을 확인할 수 있었다.

또한, Nvidia GPU 뿐만 아니라, NPU(Neural Processing Unit) 디바이스인 Oroid-N2에서 RKNN 엔진에 탑재하여 구동하였는데, 640 이미지에 대해서 14FPS 정도 추론 성능이 나왔다.

그림 6은 삼성 휴대폰 Galaxy S22에 실시간으로 객체 탐지 응용을 구동했을 때 Adreno GPU의 적용 전후의 추론 시간을 살펴보았다. GPU를 사용했을 때 추론 시간이 76ms임에 반해 사용하지 않았을 때 추론 시간이 235ms로 나와 GPU 사용으로 연산 가속이 3배 이상 되었음을 확인하였다.



(a) GPU 가속기 사용 전 (b) GPU 가속기 사용 후

그림 6. 휴대 단말에서의 GPU 가속기 사용 전후 추론 시간 비교

Fig. 6. Comparison of inference time before and after using GPU accelerator on mobile devices.

Ⅳ. 결 론

본 논문에서는 인공지능 응용이 다양한 디바이스에서 구동되는 상황에서 전문가가 아니더라도 산업현장의 디바이스에 최적화된 신경망을 손쉽게 생성할 수 있는 프레임워크를 제안하였다. 기존의 MLOps 구조에 신경망 모델 추천 및 신경망 생성 모듈에 기능을 추가하여 편하고 효율적인 신경망 개발을 할 수 있도록 제안하였다. 본 연구개발 내용은 공개 프로젝트(TANGO)로 수행되고 있으며 Github에 매년 2회 릴리즈를 하고 있다.

제안한 신경망 생성 프레임워크는 산업 분야의 다양한 곳에 적용될 수 있도록 향후 신경망 모델 추천에 디바이스의 검색 기능과 데이터 기반 추천 알고리즘을 추가할 예정이다. 또한 제안 프레임워크에서는 산업에서 사용하는 산업 데이터를 이용한 응용 서비스를 개발할 수 있게 전이학습 기능이 제공하도록 하며, 입력 정보로 들어오는 다양한 디바이스에 대해 세밀하고 보다 정밀한 신경망을 생성할 수 있도록 할 것이다.

REFERENCES

[1] Andrey Ignatov, Radu Timofte, Andrei Kulik, Seungsoo Yang, Ke Wang, Felix Baum, Max Wu, Lirong Xu, and Luc Van Gool, “AI Benchmark: All About Deep Learning on Smartphones in 2019,” ICCV 2019, pp. 3617-3635, Oct. 2019

[2] Yeon-bok Lee, “Trends in hardware platforms and software frameworks for AI Inference,” The Magazine of the IEIE, Vol. 50, No. 2, pp. 37-46, Feb., 2023

[3] Xin He, Kaiyong Zhao, and Xiaowen Chu, “AutoML: A survey of the state-of-the-art,” Knowledge-based systems, Vol 212, article 06622, Jan. 2021

[4] Dominik Kreuzberger, Niklas Kuhl , and Sebastian Hirschl, “Machine Learning Operations (MLOps): Overview, Definition, and Architecture,“ IEEE Access, Vol. 11, pp. 31866-31879, Mar. 2023

[5] Google Vertex AI, https://cloud.google.com/vertex-ai?hl=ko

[6] MS Azure Machine Learning, https://azure.microsoft.com

[7] Amazon SageMaker, https://aws.amazon.com/pm/sagemaker

[8] Young-Sok Park, “Kubernetes-based open source AI/ML platform,“ The Magazine of the IEIE, Vol. 50, No. 2, pp. 57-62, Feb., 2023

[9] TANGO Project, https://github.com/ML-TANGO/TANGO

[10] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object Detectors,” Arxiv, 2207-02696, Jul. 2022

저 자 소 개

김 선 태(평생회원)

1997년 KAIST 전자및전자공학과 학사
2000년 서울대학교 전기공학부 석사
2012년 고려대학교 메카트로닉스 박사
2000년~현재 ETRI AI컴퓨팅시스템SW연구실 책임연구원
2021~현재 UST ETRI 스쿨 인공지능전공 교수
<주요관심분야 : 인공지능 프레임워크, AutoML, 경량 OS, 저전력 IoT 네트워킹, 실시간 처리기술>

조 창 식(비회원)

1993년 경북대학교 컴퓨터공학과 학사
1995년 경북대학교 컴퓨터공학과 석사
2011년 충남대학교 컴퓨터공학과 박사
1995년~현재 ETRI AI컴퓨팅시스템SW연구실 책임연구원/실장
<주관심분야 : AutoML, MLOps, 노코드 신경망 개발 도구>