

# WICWIU: 가독성과 확장성이 우수한 C++ 기반 딥러닝 오픈소스 프레임워크

박천명<sup>○</sup>, 김지웅, 기윤호, 김지현, 김인중

한동대학교 전산전자공학부

21400337@handong.edu, 21100162@handong.edu, 21500043@handong.edu,

21500165@handong.edu, ijkim@handong.edu

## WICWIU: C++ based Deep Learning Open Source Framework with Readability and Extensibility

Chunmyong Park<sup>○</sup>, Jeewoong Kim, Yunho Kee, Jihyeon Kim, Injung Kim

School of Computer Science and Electrical Engineering, Handong Global University

### 요 약

국내 대학으로는 최초로 공개한 오픈소스 딥러닝 프레임워크 WICWIU를 소개한다. WICWIU는 다양한 신경망 계층을 제공하고 일반적인 그래프 형태의 네트워크 구조를 지원하여 ResNet, DenseNet, Inception 등 최신 CNN 모델들을 구현할 수 있다. 또한, GPU 기반 고성능 컴퓨팅을 지원해 빠른 학습이 가능하다. 모든 API가 C++로 제공되어 C++ 개발자들이 쉽게 적응할 수 있으며, 파이썬 기반의 프레임워크에 비해 메모리 및 성능 최적화에도 유리하다. 특히, 응용시스템의 개발 외에도 프레임워크 자체를 모바일과 같은 특수한 환경에 맞도록 수정 및 확장이 가능하다. 일관성 높은 코드와 API로 구성되어 가독성과 확장성이 우수하며, 한국어 문서를 제공해 국내 개발자들이 쉽게 접근할 수 있다. WICWIU는 Apache 2.0 라이선스를 적용해 연구 목적 및 상용 목적으로 제약 없이 활용할 수 있다.

### 1. 서론

딥러닝은 최근 세계적으로 활발한 연구가 이루어지고 있을 뿐만 아니라 산업계에서도 많은 주목을 받고 있다. 그러나, 현재 국내에서 수행중인 딥러닝 연구의 대부분은 해외에서 개발된 프레임워크들이 사용되고 있으며, 딥러닝 프레임워크를 자체 개발한 기업이나 대학은 매우 희소하다. 그 결과 딥러닝 알고리즘까지 개발한 경험을 보유한 핵심 기술자들이 매우 적어서 해외 기업에 의한 기술적 종속이 우려된다. 본 논문에서는 이러한 문제를 극복하기 위해 순수 국내 기술로 개발한 오픈소스 딥러닝 프레임워크 WICWIU(위큐)를 공개한다. WICWIU는 국내 대학에서 최초로 공개하는 딥러닝 오픈소스 프레임워크이며, 연구 목적 및 상용 목적으로 제약 없이 활용 가능하다.

대부분의 기존 딥러닝 오픈소스들은 파이썬 환경을 중심으로 설계되어 응용시스템 개발이 편리하다는 장점이 있으나, 사용자가 프레임워크 자체를 수정/확장하거나 모바일 등 다양한 환경에 이식하기 쉽지 않다는 단점이 존재한다. WICWIU는 일관성 있는 구조를 제공하고 있어 가독성과 확장성이 우수하다. 모든 API를 C++로 제공하여 C++개발자들이 쉽게 접근 가능하며, 파이썬 기반의 프레임워크에 비해 메모리 및 성능 최적화에 유리하다. 특히, 응용시스템의 개발 외에도 프레임워크 자체를 모바일 등 특수한 환경에 맞도록 수정/확장이 가능하다.

제안하는 프레임워크는 한글 문서를 제공하고, 주요 딥러닝 알고리즘을 저수준 코드까지 직접 살펴보면서 이해하거나 수정할 수 있어 딥러닝에 입문하는 국내 연구자들이 딥러닝 알고리즘을 직접 체험하거나 새로운 아이디어를 테스트하기에도 적합하다.

본 프레임워크의 명칭 WICWIU는 “What I can Create is What I Understand”의 약자이다. WICWIU는 Apache 2.0 라이선스로 공개되었으며, 소스코드는 GitHub<sup>1</sup>에서 다운로드 가능하다.

### 2. 설계 목표 및 접근 방법

본 프레임워크는 딥러닝 개발자들이 쉽게 이해하고 직접 수정/확장할 수 있도록 설계되었다. 첫째, 기존 딥러닝 프레임워크들이 주로 파이썬 환경을 중심으로 개발된 것에 비해 WICWIU는 국내 개발자들에게 친숙한 C++ 언어로 내부 코드와 API를 제작하였다. C++ 환경은 파이썬 환경에 비해 메모리 및 성능 최적화에 용이하며 모바일과 같은 다양한 환경에 이식하기 쉽다. 본 프레임워크는 C++98을 지원하며, 이후 버전은 모두 호환 가능하다.

둘째, WICWIU는 코드의 가독성을 높이기 위해 많은 노력을 기울였으며, 문서와 주석 등을 한글로 제공하여 국내 개발자가 처음 딥러닝 분야를 학습할 때 이해하기 쉽도록 제작되었다.

<sup>1</sup> <https://github.com/wicwiu/wicwiu/>

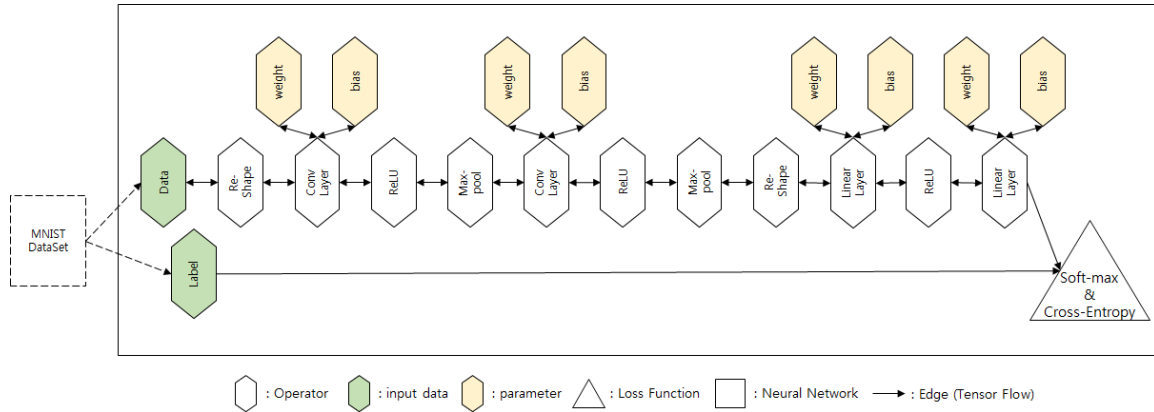


그림 1 WICWIU를 이용해 구현한 CNN의 예 (MNIST 필기 숫자 인식기)

셋째, 사용자가 새로운 알고리즘을 적용하기 용이하도록 설계하였다. WICWIU는 사전 단계에서 확장성을 고려하여 설계하였으며, 이를 통해 코드와 API의 일관성을 확보하였다. 이를 통해 사용자가 쉽게 자신만의 연산자와 신경망 계층을 쉽게 구현 가능하다.

마지막으로, 고성능 계산을 위해 GPU를 이용하여 많은 계산을 빠르게 수행할 수 있도록 설계하였다. GPU연산은 cuDNN을 이용해 구현하였다. 또한, 호스트 메모리와 GPU 메모리 간 동기화를 자동으로 수행하여 사용자의 편의를 증진시켰다.

### 3. 프레임워크 구조

WICWIU는 데이터를 표현하기 위한 Tensor 및 Shape, 신경망 연산 지원을 위한 Operator 및 Layer, 학습을 위한 Loss Function과 Optimizer, 신경망 모델을 표현하기 위한 Neural Network 등으로 구성된다. 각 클래스의 역할과 특징은 다음과 같다.

#### 3.1 Tensor 및 Shape 클래스

WICWIU의 Tensor 클래스는 최대 5차원 텐서데이터를 저장 및 관리하며, 모든 신경망 연산은 Tensor에 대해 수행된다. Tensor 내부에는 모든 데이터를 1차원 배열의 형태로 저장하나 Shape 클래스를 이용하여 외부로는 1~5차원 구조의 인터페이스를 제공한다.

Tensor의 데이터는 호스트 메모리와 GPU 메모리 양쪽에 존재할 수 있으며, 두 메모리 간 동기화는 모델이 동작하는 디바이스에 따라 자동으로 수행된다.

#### 3.2 Operator 클래스 (저수준 기본 연산)

Operator 클래스는 본 프레임워크에서 딥러닝 모델을 만들기 위해 필수적인 연산자를 표현하기 위해 제공하는 추상 클래스이다. Operator 클래스는 순전파(forward propagation)와 역전파(back-propagation)를 수행하는 저수준 연산들을 포함하며, 각 연산의 결과 값은 각 객체의 멤버 변수로 저장된다. 저장된 결과는 연결된 다른 Operator나 Loss Function의 피연산자로 사용 가능하다.

각 연산자는 Operator 클래스를 상속받은 클래스로 정의되며, WICWIU는 Add, MatMul, ReLU, Sigmoid, Tanh, Convolution, Max-Pooling, Batch-Normalization 등 CNN에 자주 사용되는 기본 연산들을 내장한다. 또한, 이들은 순전파 및 역전파 연산을 제공하고 있어 사용자가 모델을 구성하면 자동으로 학습을 수행할 수 있도록 돕는다. 마지막으로, 사용자는 Operator 클래스를 상속받아 새로운 연산자를 정의할 수 있다.

#### 3.3 Layer 클래스 (고수준 복합 연산)

Layer 클래스는 복잡한 신경망 계층을 편리하게 구현하기 위해 설계된 추상 클래스이다. 복잡한 신경망 모델을 저수준 연산에 해당하는 Operator 클래스만을 이용하여 구현하는 것은 불편하다. 이를 극복하기 위해 복수의 Operator를 조합해 복잡한 계층을 구현할 수 있도록 설계된 클래스가 Layer 클래스이다.

Layer 클래스의 연산은 복수의 Operator들을 그래프 구조로 조합함으로써 정의된다. 또한, Layer 클래스는 Operator 클래스의 자식클래스이기 때문에 다른 Layer를 포함하는 재귀적 구조도 구성 가능하다.

WICWIU는 Convolution, Batch-Normalization, Linear(fully-connected layer) 등 자주 사용되는 계층들을 내장하고 있으며, 사용자가 Layer 클래스를 상속하여 커스텀 Layer를 정의할 수 있다.

#### 3.4 Loss Function 클래스 (학습을 위한 손실 함수)

딥러닝 모델의 학습은 손실 함수를 정의한 후 이를 최소화하도록 파라미터를 최적화함으로써 수행된다. Loss Function 클래스는 본 프레임워크에서 손실 함수를 표현하기 위한 추상 클래스이다.

WICWIU는 MSE(Mean Squared Error), cross entropy, hinge loss (max-margin loss) 등 널리 사용되는 손실 함수들을 기본으로 제공하며, 사용자가 개별적으로 Loss Function 클래스를 상속받아 새로운 손실 함수를 구현할 수 있다.

3.5 Optimizer 클래스 (파라미터 최적화 알고리즘)

Optimizer 클래스는 경사도 벡터(Gradient)를 이용하여 파라미터를 최적화하는 경사하강법 및 파생 알고리즘을 표현하는 추상클래스이다. 본 프레임워크의 Operator 클래스들은 모두 역전파 연산을 내장하여 경사도 벡터를 자동으로 계산할 수 있다.

WICWIU는 SGD(Stochastic Gradient Descent), 모멘텀 등 기본적인 최적화 알고리즘들을 제공하고 있다. Optimizer 클래스 역시 상속이 가능하여 사용자가 새로운 최적화 방식을 구현할 수 있다.

3.6 Neural Network 클래스 (신경망 모델)

Neural Network 클래스는 신경망 모델을 표현하기 위한 클래스이다. Operator와 Layer를 조합해 신경망 모델을 구성하고, 데이터를 입력 받아 결과를 출력하거나 모델을 학습하기 위해 다음과 같은 기능을 제공한다.

첫째, 사용자는 Neural Network 클래스를 이용해 Operator와 Layer로 구성된 그래프 형태의 신경망 모델을 정의한다. Neural Network 클래스는 너비 우선 탐색(breadth-first search)을 통해 순전파 및 역전파 연산 순서를 자동으로 결정한다.

둘째, 연산을 진행하는 연산자와 파라미터를 관리한다. Optimizer에게 파라미터들을 리스트 형태로 전달할 수 있으며, 학습을 진행하는 동안 중간 결과들을 조회/수정/초기화 가능하다.

셋째, 호스트 메모리와 GPU 메모리 간 데이터 동기화를 비롯하여 신경망의 연산이 GPU에서 동작하게 하는 기능을 자동으로 수행한다.

마지막으로, Neural Network 클래스는 모델의 학습 및 검증 작업을 위해 학습 및 테스트 함수를 제공한다. 먼저, 학습 함수는 순전파 및 역전파 연산, 파라미터 최적화를 포함한다. 테스트 함수는 순전파 연산만을 수행하며, 모델을 검증, 평가하거나 실제 문제에 적용하는데 사용된다.

4. 프레임워크 유효성 검증

WICWIU의 유효성을 검증하기 위해 다음과 같은 작업을 진행하였다. 모델 학습 및 테스트 데이터로는 MNIST 필기 숫자 데이터를 사용하였으며, 제공되는 바이너리 데이터를 Tensor로 전환하는 전처리 클래스를 구현하여 모델에 적용하였다. 그 후 WICWIU를 적용하여 그림 1과 같은 구조의 신경망 모델을 구현하여, 앞에서 준비한 데이터를 이용하여 학습을 진행하였다. 또한 검증을 위해 널리 사용하는 딥러닝 프레임워크인 TensorFlow로 동일한 구조의 신경망 모델을 구현하여 학습시킨 후, 테스트 데이터에 대한 각 모델의 인식률을 비교하였다[1].

표 1 WICWIU와 TensorFlow의 인식률 비교

	Accuracy (%)
Tensorflow	99.12
WICWIU	99.12

표 1로 WICWIU와 TensorFlow의 모델이 거의 동일한 성능을 보인다는 것을 알 수 있으며, 이를 통해 WICWIU를 이용하여 정의/학습된 CNN 모델이 잘 동작함을 확인하였다.

그림 1의 모델 외에도 ResNet, DenseNet 등 복잡한 최신 신경망 모델도 WICWIU를 이용해 구현 및 학습을 진행함으로써 WICWIU가 최신 CNN 모델들을 구현하는데 충분한 기능을 제공함을 확인하였다[2][3].

5. 결론 및 향후 연구

본 논문에서는 딥러닝 오픈소스 프레임워크 WICWIU를 소개하였다. 제안하는 프레임워크는 데이터를 표현하기 위한 Tensor 데이터형과 다양한 저수준 기본 연산 및 고수준 복합 계층, 일반적인 그래프 형태의 신경망 구조를 지원하여 최신 CNN 모델들을 구현할 수 있으며, 이 외에도 사용자가 기본 연산자를 사용하거나 새로 정의하는 방식을 통하여 다양한 모델을 구성할 수 있도록 설계하였다.

향후 LSTM 등 RNN 모듈, 생성적 딥러닝 모델(GAN, VAE) 등을 추가하여 더 많은 딥러닝 모델들을 지원하도록 확장할 예정이다.

사 사 의 글

본 연구는 과학기술정보통신부와 정보통신기술진흥센터의 2018년도 소프트웨어중심대학지원사업(2017-0-00130)의 지원을 받아 수행됨

참 고 문 헌

[1] Abadi, M., and Agarwal A. Barham P. , "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.", Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI'16, Savannah, GA, USA, 265-283, 2016.

[2] He, Kaiming, et al. "Deep residual learning for image recognition.", Proceedings of the IEEE conference on computer vision and pattern recognition, 770-778, 2016.

[3] Huang, Gao, et al. "Densely connected convolutional networks.", Proceedings of the IEEE conference on computer vision and pattern recognition, Vol. 1, No. 2, pp.3, 2017.