

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

박천명, 김지웅, 기윤희, 김지현, 김인중

Handong Global University

2018. 6. 21.

# Agenda

---



- Introduction
- WICWIU Overview
- Components of WICWIU
- Demo
- Conclusion & Future Works

# Introduction

- 파이썬 중심 딥러닝 프레임워크 (Pytorch, TensorFlow)
  - 응용시스템 개발에 편리
  - 우수한 성능
  - 엔진의 이해 및 소스 수정은 쉽지 않음
  - 기술적 종속 등에 대한 우려
- WICWIU: 국내 대학 최초의 딥러닝 오픈소스 프레임워크
  - 한동대 딥러닝 연구실에서 연속성을 가지고 개발
  - WICWIU = What I Create is What I Understand.
    - “What I cannot create, I do not understand” [Richard Feynman]  
에서 영감

# WICWIU Overview

## ■ WICWIU의 특징

- C/C++ 개발자에게 친숙한 C++기반 프레임워크
- Hands-on을 통해 엔진 및 알고리즘 자체를 깊이 이해하기 좋음
- 메모리, 속도 최적화에 유리

가독성	딥러닝 초보자들을 위한 가독성 높은 코드 (주요 딥러닝 알고리즘)
-----	--------------------------------------

확장성	잘 정의된 API를 통해 자신만의 Operator, Layer 추가 용이
-----	--

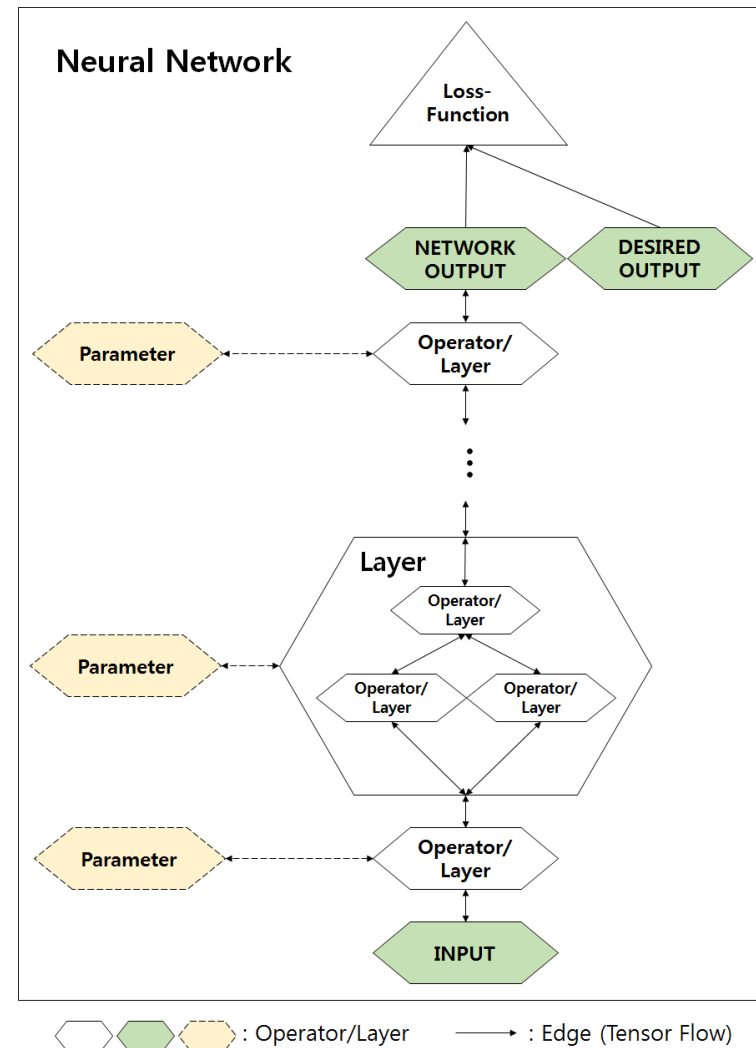
고성능	GPU 기반 대규모 병렬처리 (cuDNN)
-----	-------------------------

일반성	Linear 구조 뿐 아니라 일반적인 Graph 구조의 신경망 지원
-----	---------------------------------------

편리성	Auto-differentiation 지원 (Gradient 계산 자동화)
-----	---

# Components of WICWIU

- Data
  - Tensor
  - Shape
  - Long Array
- Model Building Block
  - Operators
  - Layers
- Learning
  - Loss Functions
  - Optimizers
- Neural Network



# WICWIU Components: Data

- **Tensor** class – 다차원 Tensor 데이터 저장/관리

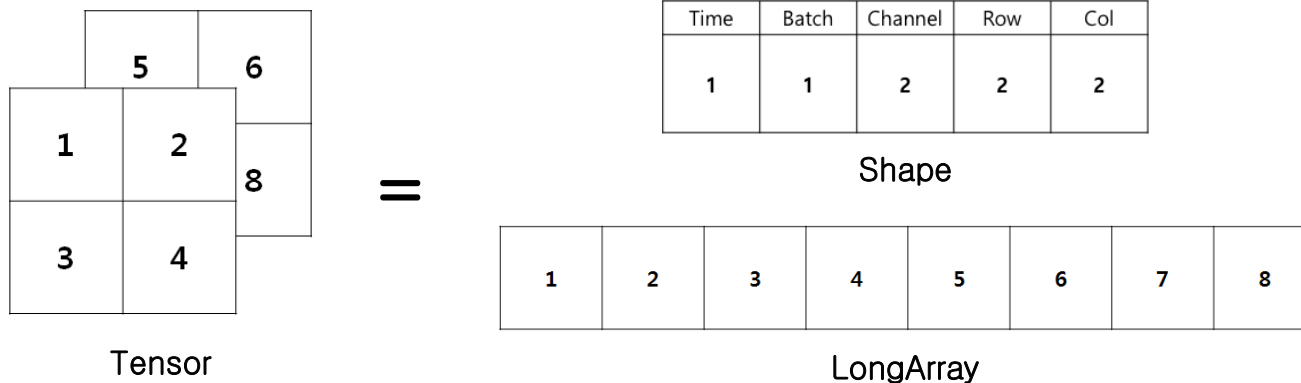
- 최대 5D : Time, Batch, Channel, Row, Col
- Host <-> Device 메모리 Auto Sync

- **Shape** class – 다차원 Tensor의 형태 표현

- Rank 및 각 차원의 길이

- **Long Array** class – 대규모 1D 데이터 저장

- 내부적으로는 대규모 메모리 할당을 위해 2차원 구조 사용

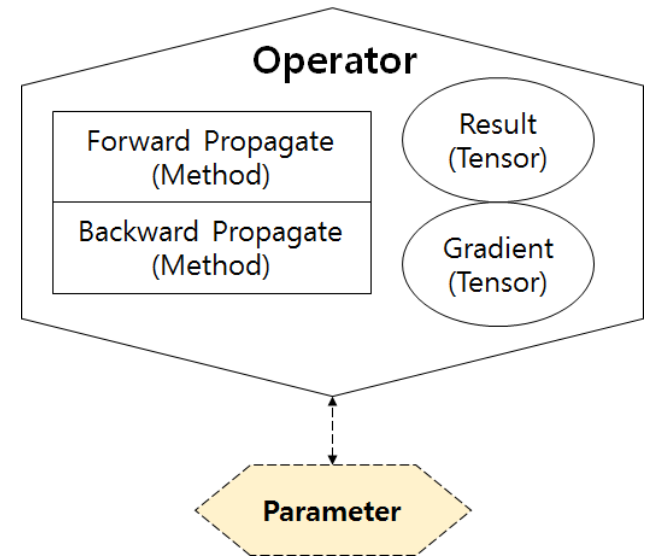


# WICWIU Components: Operators

- **Operators** class: 저수준 기본 연산자를 위한 클래스
  - 모델을 구성하는 연산자를 표현하기 위한 추상 클래스
    - Operator 공통 인터페이스
    - 순전파와 역전파 연산 함수 포함
    - 연산 결과값 및 경사도 벡터를 Tensor로 저장
    - 연산자와 파라미터 구분

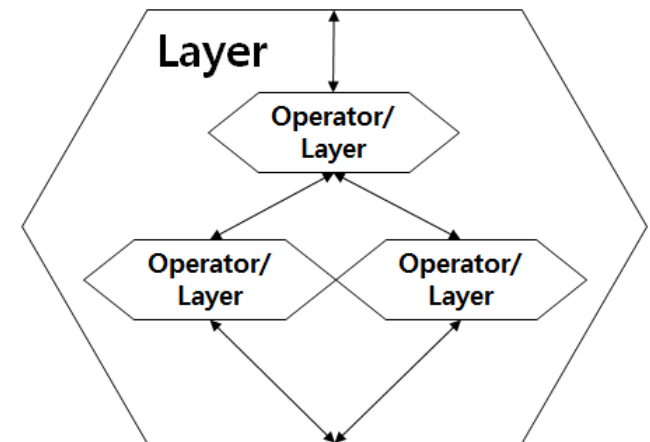
- Built-in Operators (Subclasses)

- |                |                       |
|----------------|-----------------------|
| ■ TensorHolder | ■ Tanh                |
| ■ Add          | ■ Convolution         |
| ■ MatMul       | ■ Max-pooling         |
| ■ ReLU         | ■ Average-pooling     |
| ■ Sigmoid      | ■ Batch-Normalization |



# WICWIU Components: Layers

- **Layers** class: 고수준 복합 연산 계층을 위한 클래스
  - Operator를 조합해 신경망 계층을 구현하기 위한 추상 클래스
    - Layer 공통 인터페이스
    - 복수의 Operator를 조합하여 Subgraph 구성 및 연산
    - 다른 Layer를 포함하는 재귀적 구조 (Operator를 상속)
- Built-in Layers (Subclasses)
  - Convolution Layer
  - Batch-Normalization Layer
  - Linear(fully-connected) Layer
- Example Layers
  - Resnet Basic Block
  - Densenet Basic Block (under construction)

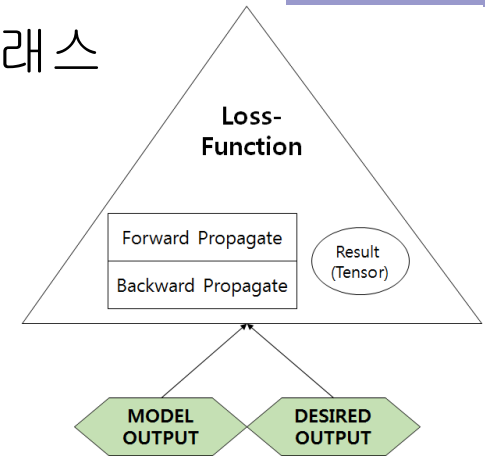




# WICWIU Components: Learning

## ■ Loss Function class: 학습을 위한 손실 함수 클래스

- Loss Function 공통 인터페이스
- 순전파와 역전파 연산 함수 포함
- Built-in Loss Functions
  - Mean Squared Error
  - Cross Entropy
  - Hinge Loss (under construction)

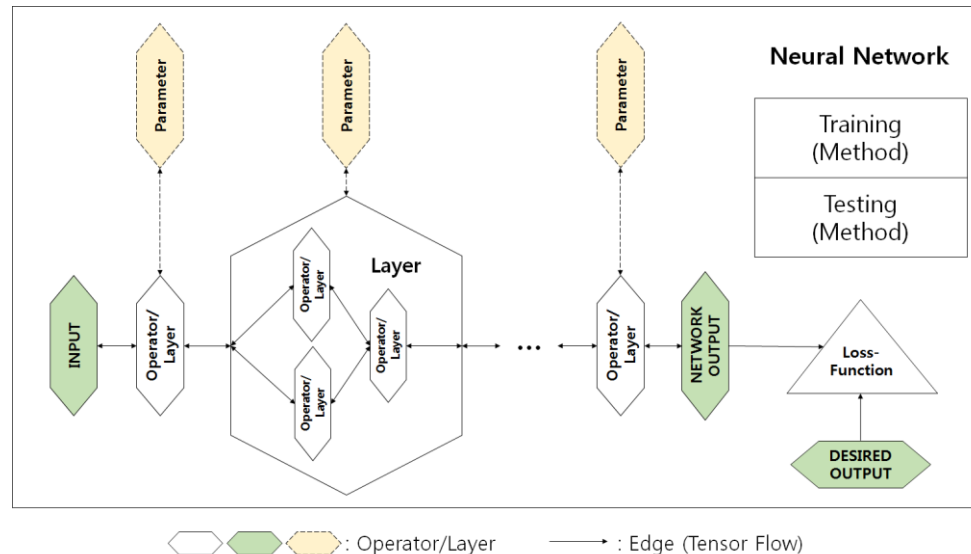


## ■ Optimizer class: 경사도 벡터 (gradient)를 이용한 모델 파라미터를 최적화 알고리즘

- Optimizer 공통 인터페이스
- Parameter Update function
- Built-in Optimizers
  - Stochastic Gradient Descent (SGD)
  - Momentum
  - RMSProp (under construction)
  - ADAM (under construction)

# WICWIU Components: Neural Network

- **Neural Network** class: 신경망 모델을 표현하기 위한 추상 클래스
  - Operator와 Layer를 조합해 신경망 모델을 구성 (일반적인 Graph구조)
  - Loss Function과 Optimizer등을 이용한 모델 학습
  - 모델 구성, 연산 및 학습을 위한 다양한 기능 제공
    - Operator와 Layer 추가/연동
    - 너비 우선 탐색(Breadth-first search)으로 연산/학습 순서 결정
    - 연산자와 모델 파라미터를 구분하여 관리
    - 모델의 학습 및 검증 작업을 위한 학습 및 테스트 함수 제공

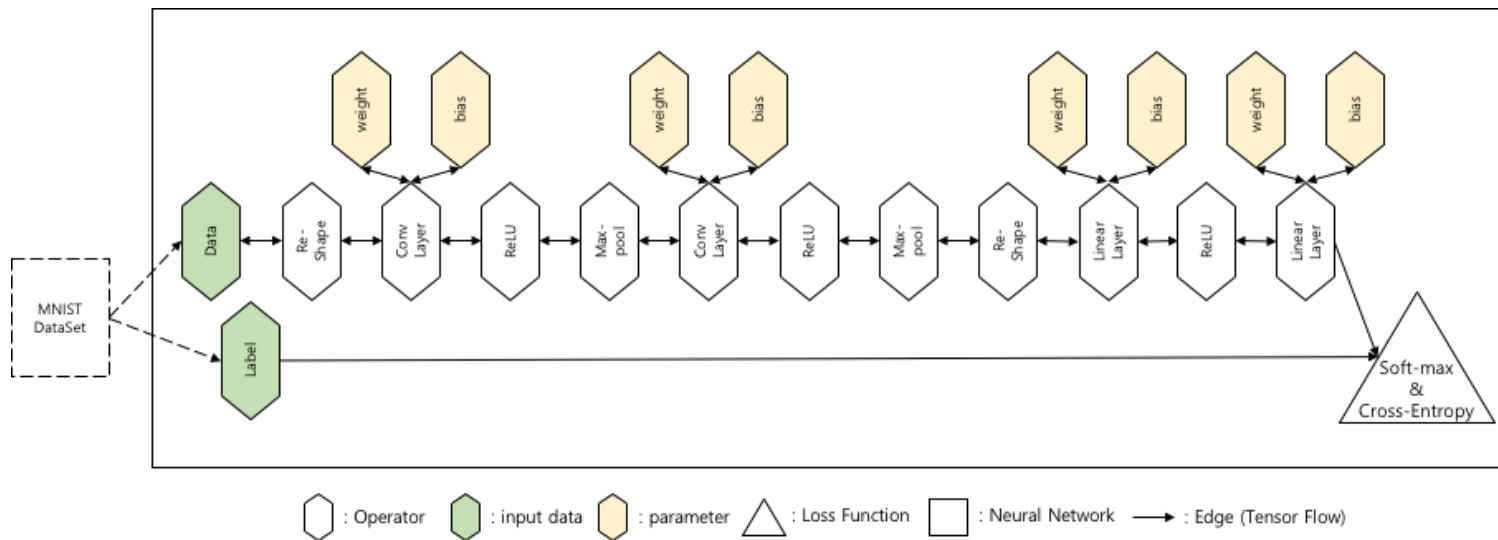


# Demo

## ■ 튜토리얼 – MNIST 필기 숫자 인식기

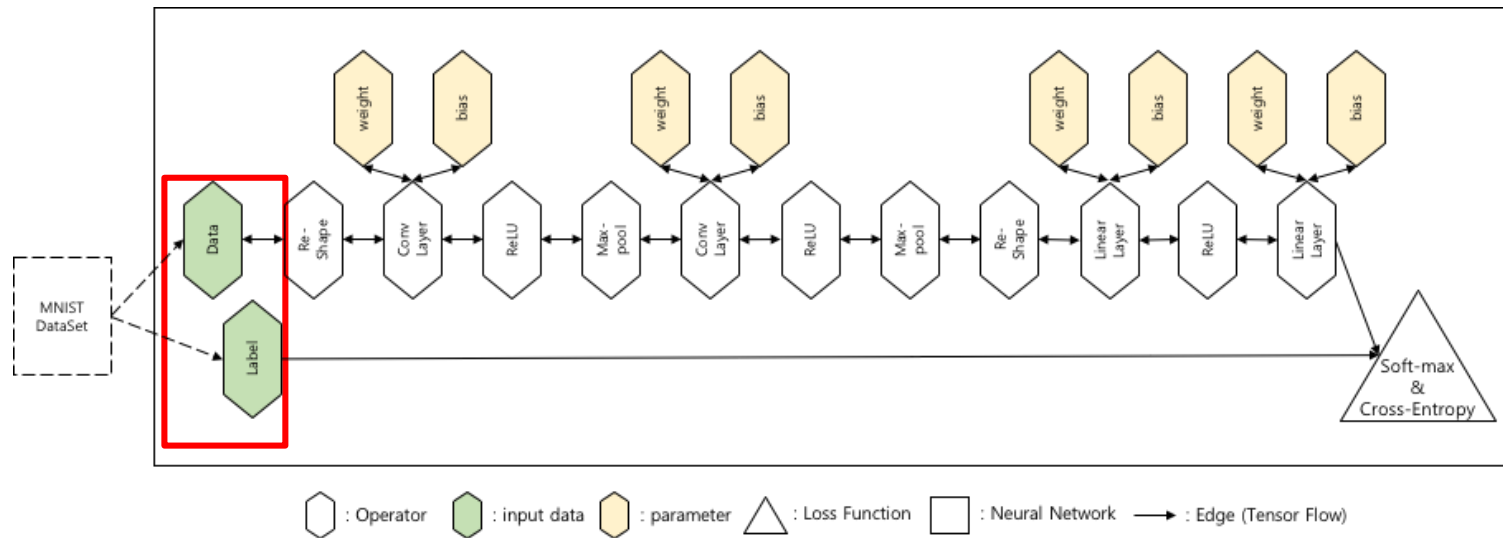
- <https://github.com/wicwiu/wicwiu/tutorials/MNIST>

### MNIST 인식을 위한 CNN 모델



# Demo: CNN Model Code

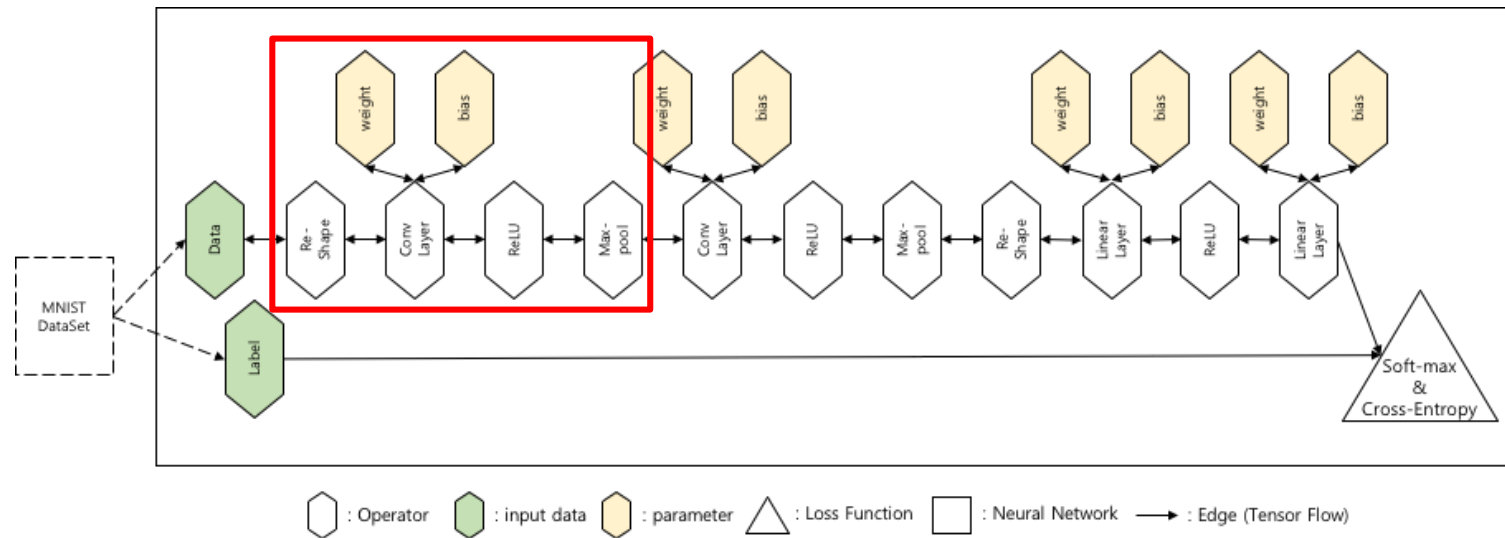
```
class my_CNN : public NeuralNetwork<float>{  
private:  
public:  
    my_CNN(Tensorholder<float> *x, Tensorholder<float> *label) {  
        SetInput(2, x, label);  
    }  
};
```



# Demo: CNN Model Code

```
out = new ReShape<float>(x, 28, 28, "Flat2Image");

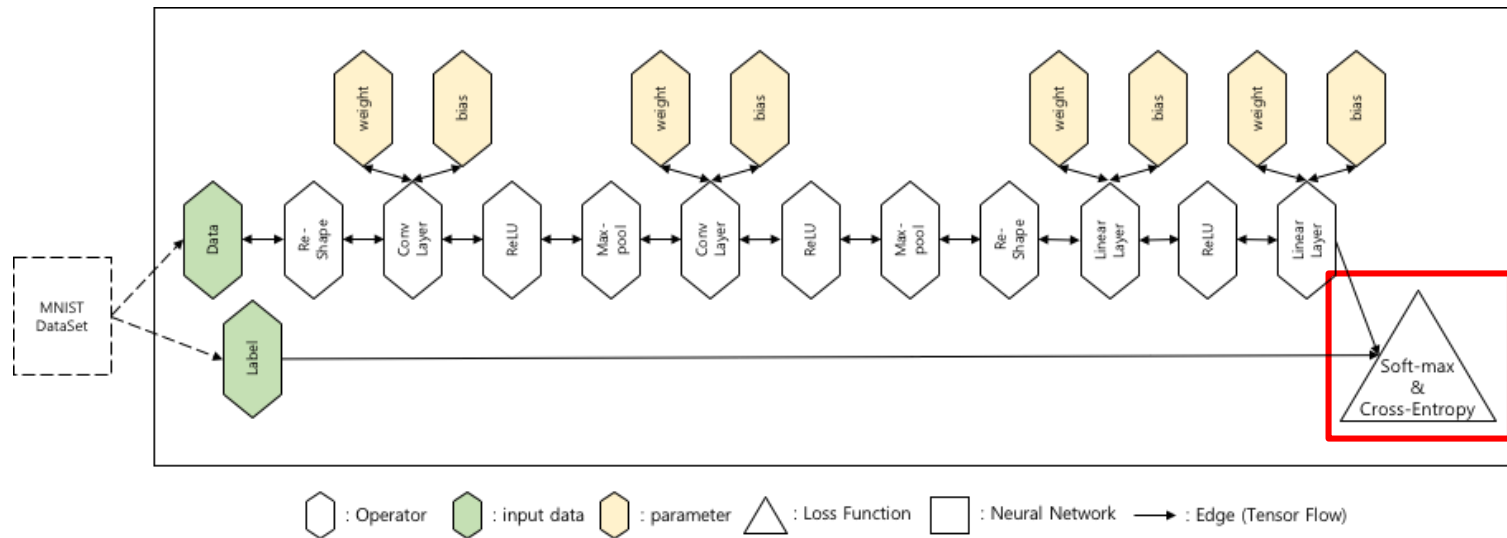
// ===== layer 1=====
out = new ConvolutionLayer2D<float>(out, 1, 10, 3, 3, 1, 1, 0, TRUE, "Conv_1");
out = new Relu<float>(out, "Relu_1");
out = new Maxpooling2D<float>(out, 2, 2, 2, 2, "MaxPool_1");
```



# Demo: CNN Model Code

```
AnalyzeGraph(out);
```

```
// ===== Select LossFunction Function =====  
SetLossFunction(new SoftmaxCrossEntropy<float>(out, label, "SCE"));  
  
// ===== Select Optimizer =====  
SetOptimizer(new GradientDescentOptimizer<float>(GetParameter(), 0.04, MINIMIZE));
```



# Demo: Training Code

```
// ----- Select net -----  
NeuralNetwork<float> *net = new my_CNN(x, label);  
  
// ===== Prepare Data =====  
MNISTDataSet<float> *dataset = CreateMNISTDataSet<float>();  
  
for (int i = 0; i < EPOCH; i++) {  
    for (int j = 0; j < LOOP_FOR_TRAIN; j++) {  
        dataset->CreateTrainDataPair(BATCH);  
  
        Tensor<float> *x_t = dataset->GetTrainFeedImage();  
        Tensor<float> *l_t = dataset->GetTrainFeedLabel();  
  
        net->FeedInputTensor(2, x_t, l_t);  
        net->ResetParameterGradient();  
        net->Training();  
    }  
}
```

# Demo: 실행

A terminal window with a title bar showing 'cmpark — chunjee@deep300: ~/example — ssh chunjee@203.252.106.38 — 98x24'. The terminal content shows the command 'chunjee@deep300:~/example\$ git clone https://github.com/wicwiu/wicwiu' being entered. The prompt 'chunjee@deep300' is in green, the path '~/example' is in blue, and the rest of the command is in black. A cursor is at the end of the command.

```
chunjee@deep300:~/example$ git clone https://github.com/wicwiu/wicwiu
```



# Demo: 성능

- 프레임워크 유효성 검증

- Tensorflow를 사용하여 동일한 모델 학습 및 인식을 비교

	Accuracy(%)
Tensorflow	99.12
WICWIU	99.12

- WICWIU를 이용하여 정의/학습된 CNN 모델이 잘 동작함을 확인

- ResNet 모델도 예제에 포함

# Download and License



- WICWIU 사용을 위해 필요한 S/W
  - gcc/g++
  - CUDA Toolkit 9.0 이상
  - cuDNN 7.0.5 이상
- Download URL
  - <https://github.com/wicwiu/wicwiu>
  - \$ git clone https://github.com/wicwiu/wicwiu
- License
  - Apache 2.0
  - 연구 목적 및 상용 목적으로 제약 없이 활용 가능

# Conclusion & Future Works



- 국내 대학 최초의 딥러닝 오픈소스 프레임워크
  - 주요 딥러닝 알고리즘에 있어서 가독성 높은 코드 제공
  - GPU기반 고성능 병렬 연산 지원
  - 확장성을 고려해 설계된 클래스 구조
- Future Works
  - Advanced CNN Modules
  - RNN Module (LSTM 등)
  - Generative Models (GAN, VAE, pixel RNN/CNN)



Thank you  
for your attention!

