

LG 에너지 솔루션 실습 강의

# Pytorch Tutorial

---

2024. 06. 21

고려대학교 산업경영공학과

Data Mining & Quality Analytics Lab.

임새린

# Pytorch란?

Pytorch

❖ 파이썬 딥러닝 개발 프레임워크



vs



# Pytorch란?

Pytorch

❖ 파이썬 딥러닝 개발 프레임워크



## TensorFlow

vs



## PyTorch

- 다양한 지원 범위(C언어, 하드웨어 등)
- 강력한 배포 기능(TensorFlow Serving, TensorFlow Lite 등)
- 자동화된 최적화

- 직관적인 인터페이스
- 동적 그래프를 활용한 디버깅 및 커스터마이징
- 활발한 커뮤니티 및 오픈소스

# Pytorch란?

Pytorch

❖ 파이썬 딥러닝 개발 프레임워크



## TensorFlow

vs



## PyTorch

- 복잡하고 추상적인 표현
- 정적 그래프로 인한 유연성 부족
- 커뮤니티 분열 (TensorFlow & TensorFlow 2.0)

- 동적 그래프로 인한 속도와 성능 하락
- 배포 및 서비스화 부족
- 문서화된 튜토리얼이 부족

# Pytorch란?

Pytorch

❖ 파이썬 딥러닝 개발 프레임워크

- 연구, 개발을 위해서는 직관성, 쉬운 디버깅, 쉬운 커스터마이징의 특징을 가지는 pytorch가 더 적합함



vs

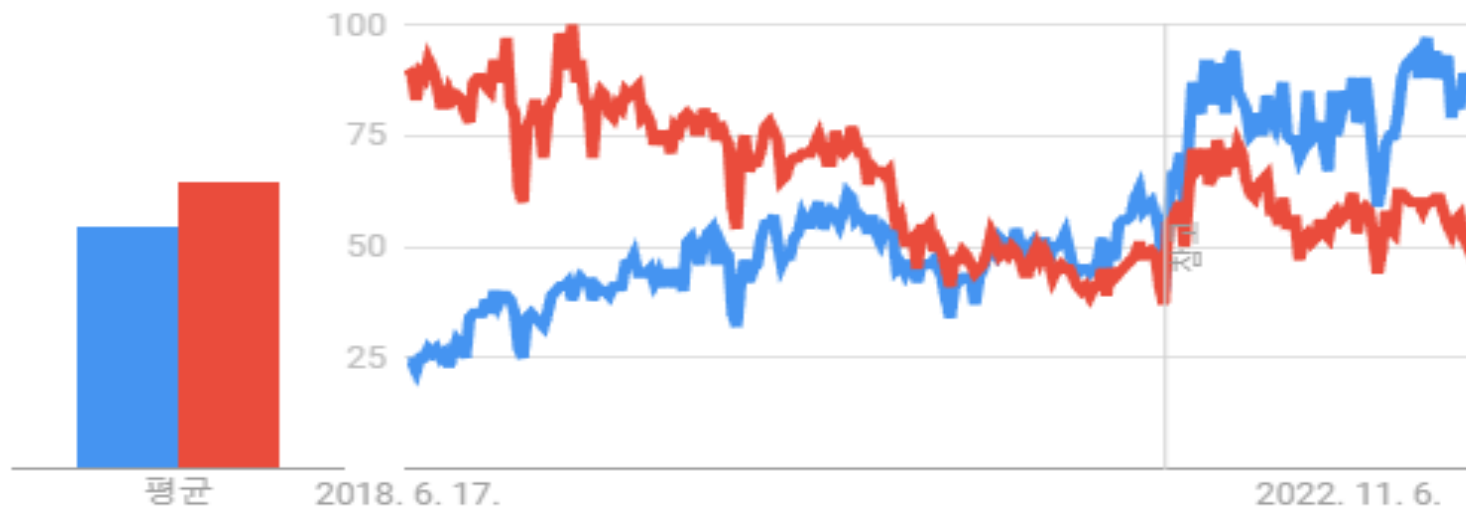


# Pytorch란?

Pytorch

❖ 파이썬 딥러닝 개발 프레임워크

● pytorch ● tensorflow



전 세계. 지난 5년. 웹 검색.

# Pytorch Tutorial

## Tutorial

### ❖ 아나콘다 가상환경에 pytorch와 CUDA 설치하기

- 1. anaconda prompt 창에서 새로운 가상환경 생성 및 실행

Anaconda Prompt

가상환경 이름

파이썬 버전

```
(base) C:\Users\korea>conda create -n mytorch python=3.11
Retrieving notices: ...working... done
Channels:
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done
```

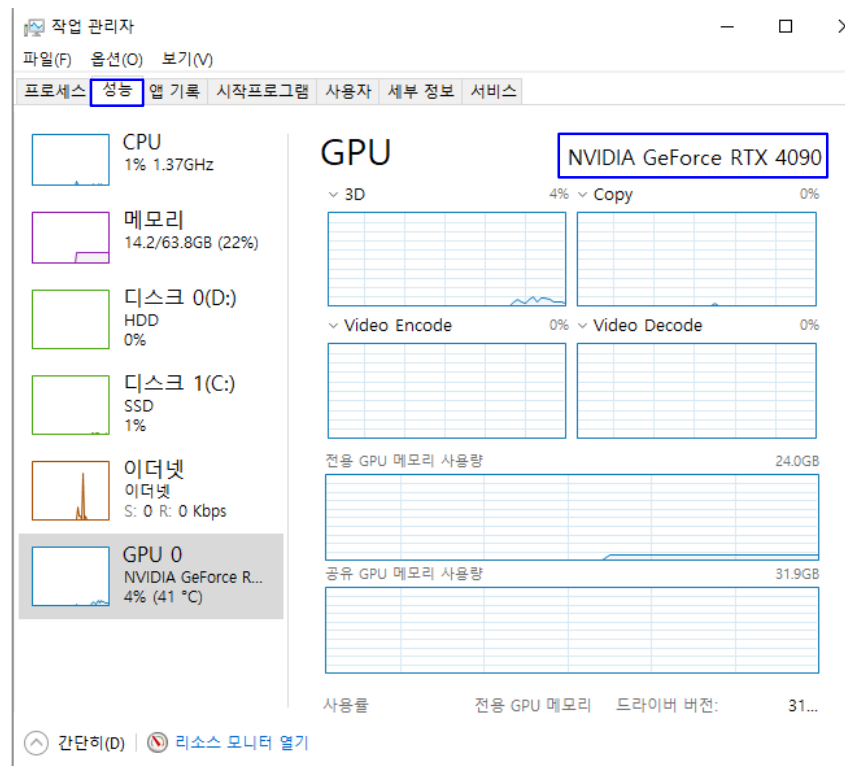
```
(base) C:\Users\korea>conda activate mytorch
(mytorch) C:\Users\korea>_
```

# Pytorch Tutorial

## Tutorial

### ❖ 아나콘다 가상환경에 pytorch와 CUDA 설치하기

- 2. 현재 GPU와 GPU driver에 맞는 CUDA Toolkit 버전 확인
  - 파이썬 버전에 맞는 CUDA toolkit과 cuDNN 설치가 필요
  - 작업 관리자에서 자신의 GPU 확인










# Pytorch Tutorial

## Tutorial

### ❖ 아나콘다 가상환경에 pytorch와 CUDA 설치하기

- 2. 현재 GPU와 GPU driver에 맞는 CUDA Toolkit 버전 확인
  - 파이썬 버전에 맞는 CUDA toolkit과 cuDNN 설치가 필요
  - <https://developer.nvidia.com/cuda-gpus> 에서 현재 GPU가 가지고 있는 Compute Capability 확인

	CUDA-Enabled Datacenter Products
	CUDA-Enabled NVIDIA Quadro and NVIDIA RTX
	CUDA-Enabled NVS Products
	CUDA-Enabled <b>GeForce</b> and TITAN Products
	CUDA-Enabled Jetson Products



CUDA-Enabled GeForce and TITAN Products			
GeForce and TITAN Products		GeForce Notebook Products	
GPU	Compute Capability	GPU	Compute Capability
<a href="#">GeForce RTX 4090</a>	8.9	<a href="#">GeForce RTX 4090</a>	8.9
<a href="#">GeForce RTX 4080</a>	8.9	<a href="#">GeForce RTX 4080</a>	8.9
<a href="#">GeForce RTX 4070 Ti</a>	8.9	<a href="#">GeForce RTX 4070</a>	8.9
<a href="#">GeForce RTX 4060 Ti</a>	8.9	<a href="#">GeForce RTX 4060</a>	8.9
<a href="#">GeForce RTX 3090 Ti</a>	8.6	<a href="#">GeForce RTX 4050</a>	8.9
<a href="#">GeForce RTX 3090</a>	8.6	<a href="#">GeForce RTX 3080 Ti</a>	8.6
<a href="#">GeForce RTX 3080 Ti</a>	8.6	<a href="#">GeForce RTX 3080</a>	8.6
<a href="#">GeForce RTX 3080</a>	8.6	<a href="#">GeForce RTX 3070 Ti</a>	8.6
<a href="#">GeForce RTX 3070 Ti</a>	8.6	<a href="#">GeForce RTX 3070</a>	8.6

# Pytorch Tutorial

## Tutorial

### ❖ 아나콘다 가상환경에 pytorch와 CUDA 설치하기

- 2. 현재 GPU와 GPU driver에 맞는 CUDA Toolkit 버전 확인
  - 파이썬 버전에 맞는 CUDA toolkit과 cuDNN 설치가 필요
  - [https://www.wikiwand.com/en/CUDA#/GPUs\\_supported](https://www.wikiwand.com/en/CUDA#/GPUs_supported) 에서 Compute Capability에 맞는 CUDA Toolkit 버전 확인

Compute Capability (CUDA SDK support vs. Microarchitecture)											
CUDA SDK Version(s)	Tesla	Fermi	Kepler (Early)	Kepler (Late)	Maxwell	Pascal	Volta	Turing	Ampere	Ada Lovelace	Hopper
1.0 <sup>[38]</sup>	1.0 – 1.1										
1.1	1.0 – 1.1+x										
2.0	1.0 – 1.1+x										
2.1 – 2.3.1 <sup>[39][40][41][42]</sup>	1.0 – 1.3										
3.0 – 3.1 <sup>[43][44]</sup>	1.0	2.0									
3.2 <sup>[45]</sup>	1.0	2.1									
4.0 – 4.2	1.0	2.1									
5.0 – 5.5	1.0			3.5							
6.0	1.0		3.2	3.5							
6.5	1.1			3.7	5.x						
7.0 – 7.5		2.0			5.x						
8.0		2.0				6.x					
9.0 – 9.2			3.0				7.0 – 7.2				
10.0 – 10.2			3.0				7.5				
11.0 <sup>[46]</sup>				3.5				8.0			
11.1 – 11.4 <sup>[47]</sup>				3.5				8.6			
11.5 – 11.7.1 <sup>[48]</sup>				3.5				8.7			
11.8 <sup>[49]</sup>				3.5						8.9	9.0
12.0 – 12.5					5.0					9.0	

# Pytorch Tutorial

## Tutorial

### ❖ 아나콘다 가상환경에 pytorch와 CUDA 설치하기

- 2. 현재 GPU와 GPU driver에 맞는 CUDA Toolkit 버전 확인
  - 파이썬 버전에 맞는 CUDA toolkit과 cuDNN 설치가 필요
  - 현재 GPU driver에서 설치 가능한 최대 CUDA Toolkit 버전 확인(최대 버전을 높이려면 더 높은 driver 버전 재설치 필요)

현재 GPU driver에서 설치 가능한 최대 CUDA Toolkit 버전

```
(base) C:\Users\korea>nvidia-smi
Fri Jun 14 11:35:49 2024
```

NVIDIA-SMI 528.02		Driver Version: 528.02		CUDA Version: 12.0	
GPU	Name	TCC/WDDM	Bus-Id	Disp.A	Volatile Uncorr. ECC
Fan	Temp	Perf Pwr:Usage/Cap		Memory-Usage	GPU-Util Compute M. MIG M.
0	NVIDIA GeForce ...	WDDM	00000000:01:00.0	On	Off
0%	41C	P8 26W / 450W	1820MiB / 24564MiB		1% Default N/A

cmd 창 -> nvidia-smi

# Pytorch Tutorial

## Tutorial

### ❖ 아나콘다 가상환경에 pytorch와 CUDA 설치하기

- 3. 설치 가능한 CUDA Toolkit 버전 중 pytorch와 호환되는 CUDA Toolkit 설치
- <https://pytorch.org/get-started/locally/> 에서 Pytorch 버전 확인
- cuDNN은 자동으로 CUDA Toolkit 버전에 맞춰서 설치가 됨

PyTorch Build	Stable (2.3.1)		Preview (Nightly)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python		C++/Java	
Compute Platform	CUDA 11.8	CUDA 12.1	CUDA 12.4	ROCm 6.0
Run this Command:	pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118			

■ Anaconda Prompt - conda install -c conda-forge cudnn - deactivate - deactivate - conda install -c conda-forge cudatoolkit=11.8

```
(mytorch) C:\Users\korea>conda install -c conda-forge cudatoolkit=11.8
Channels:
- conda-forge
- defaults
Platform: win-64
Collecting package metadata (repodata.json): / _
```

Cudatoolkit 버전

```
(mytorch) C:\Users\korea>conda install -c conda-forge cudnn
```

# Pytorch Tutorial

## Tutorial

### ❖ 아나콘다 가상환경에 pytorch와 CUDA 설치하기

- 4. CUDA Toolkit 버전에 맞는 명령어를 통해 원하는 pytorch 설치 (<https://pytorch.org/get-started/previous-versions/>)
- 5. GPU를 사용할 수 있는지 확인

Anaconda Prompt - conda install -c conda-forge cudnn - deactivate - deactivate - conda install -c conda-forge cudatoolkit=11.8 - conda install -c conda-forge cudnn - conda...

```
(mytorch) C:\Users\korea>conda install pytorch==2.3.0 torchvision==0.18.0 torchaudio==2.3.0 pytorch-cuda=11.8 -c pytorch -c nvidia
Channels:
- pytorch
- nvidia
- defaults
- conda-forge
Platform: win-64
Collecting package metadata (repodata.json): 𐄂
```

```
(mytorch) C:\Users\korea>python
Python 3.11.9 | packaged by Anaconda, Inc. | (main, Apr 19 2024, 16:40:41)
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> torch.cuda.is_available()
True
>>> torch.cuda.device_count()
1
>>> torch.cuda.get_device_name(0)
'NVIDIA GeForce RTX 4090'
>>>
```

# Pytorch Tutorial

## Tutorial

- ❖ 아나콘다 가상환경에 pytorch와 CUDA 설치하기
  - .가상환경에 주피터 노트북 설치 및 연결하기

```
(mytorch) C:\Users\korea>pip install jupyter notebook
```

```
(mytorch) C:\Users\korea>python -m ipykernel install --user --name mytorch --display-name mytorch  
Installed kernelspec mytorch in C:\Users\korea\AppData\Roaming\jupyter\kernels\mytorch
```



# Pytorch Tutorial

## Tutorial

### ❖ 파이토치 자료형

- 자료형 : 변수가 메모리공간에 저장되는 형태(Type)로 pytorch는 tensor라는 독자적인 자료형이 있음

실수 변수는 FloatTensor

Data type	dtype	CPU tensor	GPU tensor
32-bit floating point	<code>torch.float32</code> or <code>torch.float</code>	<code>torch.FloatTensor</code>	<code>torch.cuda.FloatTensor</code>
64-bit floating point	<code>torch.float64</code> or <code>torch.double</code>	<code>torch.DoubleTensor</code>	<code>torch.cuda.DoubleTensor</code>
32-bit integer (signed)	<code>torch.int32</code> or <code>torch.int</code>	<code>torch.IntTensor</code>	<code>torch.cuda.IntTensor</code>
64-bit integer (signed)	<code>torch.int64</code> or <code>torch.long</code>	<code>torch.LongTensor</code>	<code>torch.cuda.LongTensor</code>
Boolean	<code>torch.bool</code>	<code>torch.BoolTensor</code>	<code>torch.cuda.BoolTensor</code>

정수 변수는 LongTensor

# Pytorch Tutorial

## Tutorial

### ❖ 파이토치 자료형

- 자료형 : 변수가 메모리공간에 저장되는 형태(Type)로 pytorch는 tensor라는 독자적인 자료형이 있음



- DataFrame 이 기본 자료형
- 3차원 이상의 데이터를 다루기 어려움

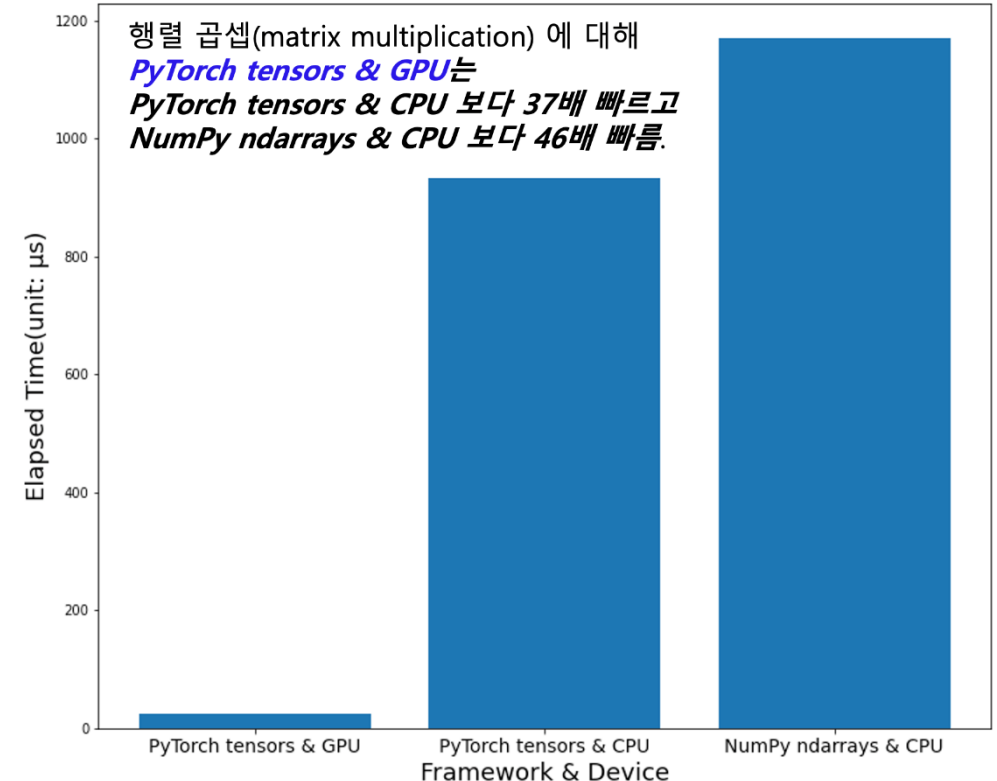


- Narray 가 기본 자료형
- GPU 연산이 불가능
- 행렬 연산에 최적화된 패키지가 아님



- Tensor 가 기본 자료형
- GPU 연산 가능
- 행렬 연산에 최적화
- Numpy와 memory를 공유하여 접근성과 효율성이 좋음

**PyTorch tensors 대비 NumPy ndarrays 의 행렬 곱셈 성능 비교**



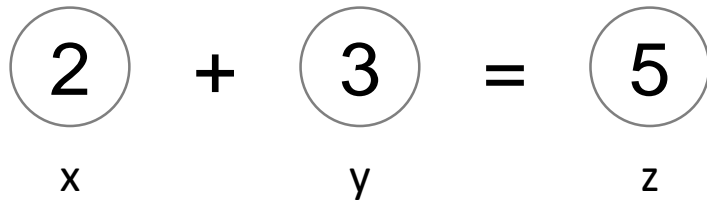


# Pytorch Tutorial

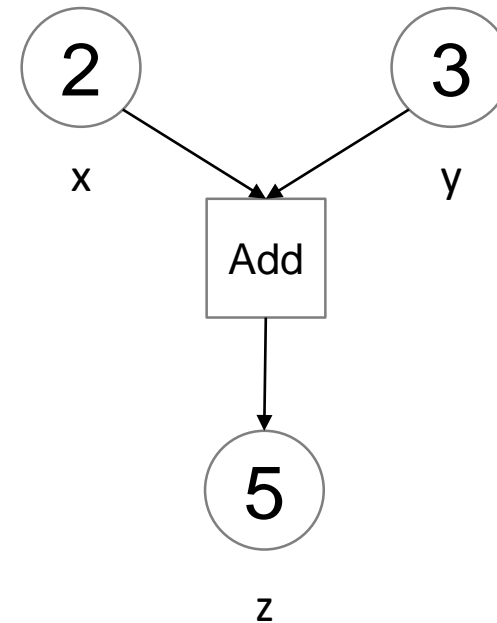
## Tutorial

### ❖ 파이토치 자동 미분 with Computational Graph

- 파이토치는 tensor의 연산이 진행될 때, computational graph를 구성함으로써 어떤 연산이 이루어졌는지를 저장



Traditional computation



computational graph

# Pytorch Tutorial

## Tutorial

### ❖ 파이토치 자동 미분 with Computational Graph

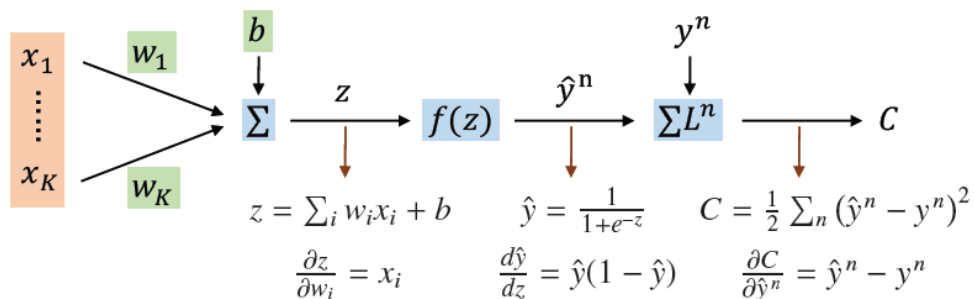
- 파이토치는 tensor의 연산이 진행될 때, computational graph를 구성함으로써 어떤 연산이 이루어졌는지를 저장
- 저장된 연산 정보를 통해서 빠른 backpropagation 연산을 가능케 함
- Training data:  $(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), \dots, (\mathbf{x}^N, y^N)$

- Goal: find the optimal parameter  $\mathbf{w}^*$ , which minimizes the MSE

$$C = \frac{1}{2} \sum_{n=1}^N (\hat{y}^n - y^n)^2 \text{ where } \hat{y}^n = f(\mathbf{w}^T \mathbf{x}^n + b)$$

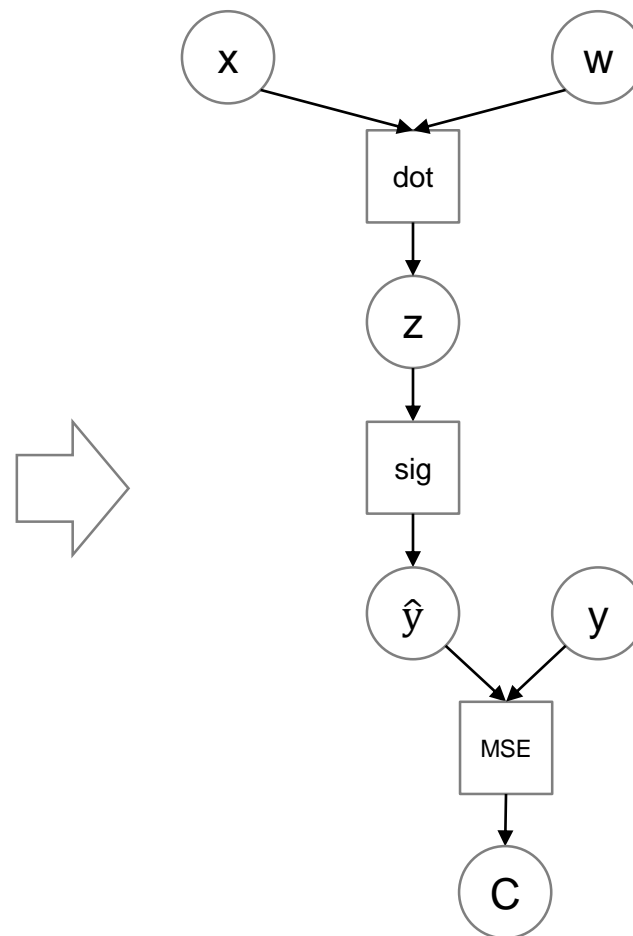
- **Backpropagation**: iteratively updating the model parameters  $\mathbf{w}$  to decrease  $C$  as

$$w_i(t+1) = w_i(t) - \eta \frac{\partial C}{\partial w_i}$$



$$\frac{\partial C}{\partial w_i} = \sum_{n=1}^N \frac{\partial z^n}{\partial w_i} \frac{d\hat{y}^n}{dz^n} \frac{\partial C}{\partial \hat{y}^n} = \sum_{n=1}^N x_i^n \hat{y}^n (1 - \hat{y}^n) (\hat{y}^n - y^n)$$

Chain rule

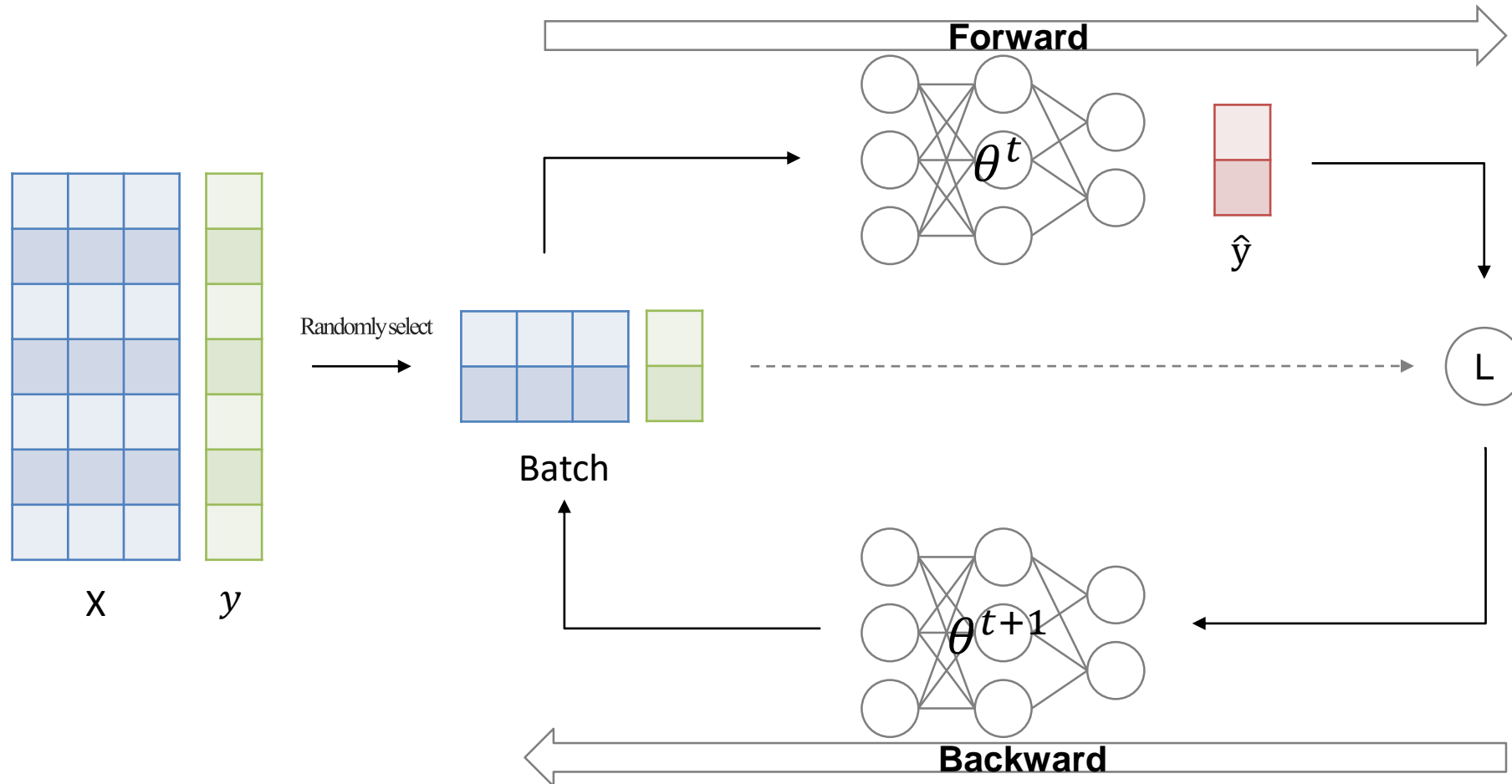


# Pytorch Tutorial

## Tutorial

### ❖ Gradient Descent Algorithm with Pytorch

- Gradient descent algorithm에 필요한 요소는 데이터, 모델, 손실 함수와 기타 하이퍼파라미터(학습률, 학습 에폭 등)

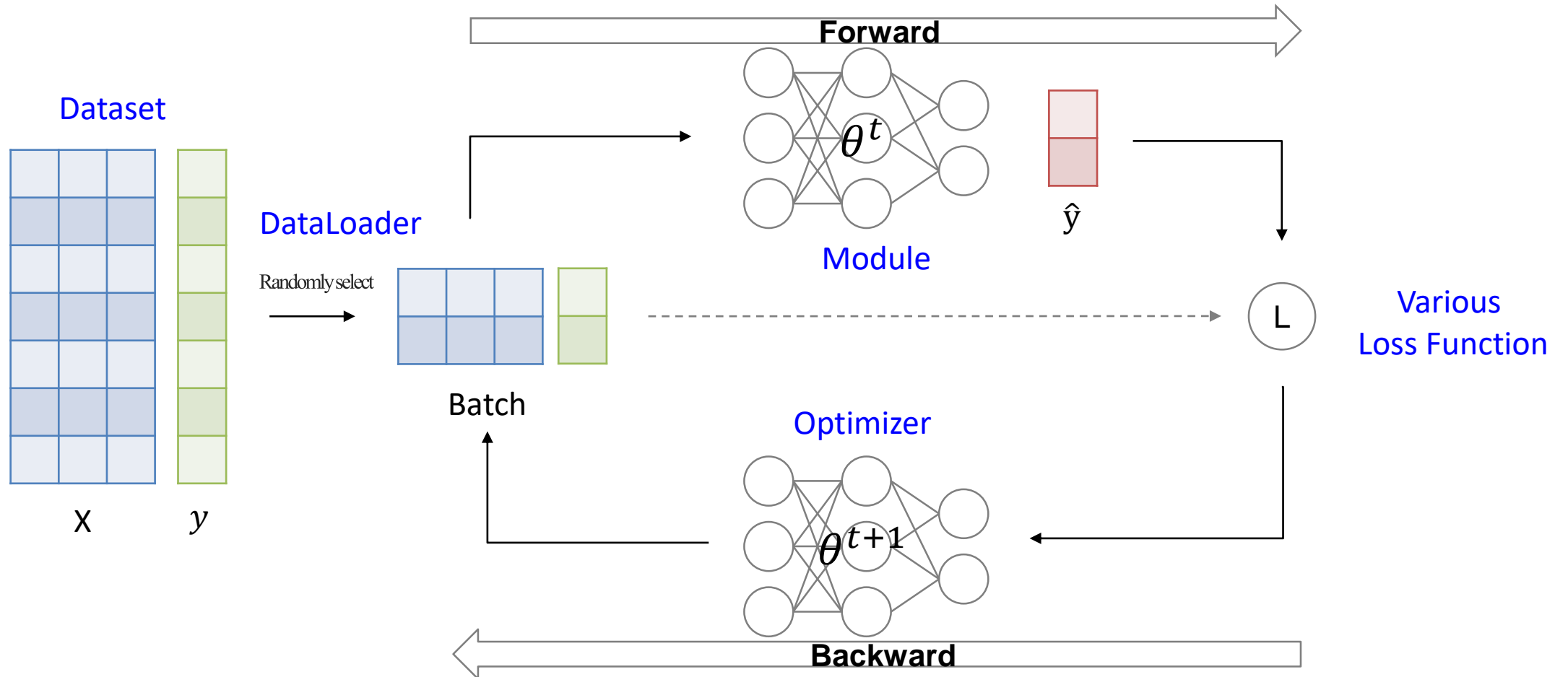


# Pytorch Tutorial

## Tutorial

### ❖ Gradient Descent Algorithm with Pytorch

- Gradient descent algorithm에 필요한 요소는 데이터, 모델, 손실 함수와 기타 하이퍼파라미터(학습률, 학습 에폭 등)

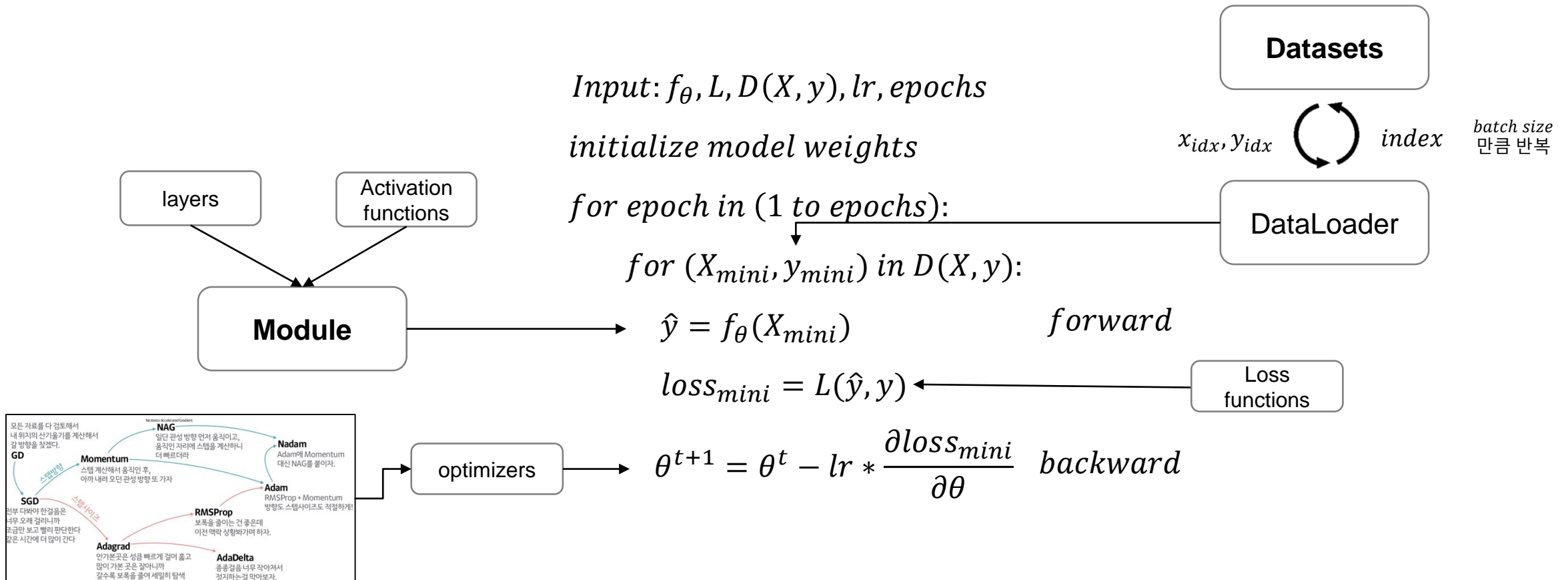


# Pytorch Tutorial

## Tutorial

### ❖ Gradient Descent Algorithm with Pytorch

- 파이토치는 데이터셋, 데이터로더, 모델, 옵티마이저, 손실함수를 통해서 Gradient Descent Algorithm을 구현



감사합니다