

# 2장 실습 환경 설정과 파이토치 기초

🕒 생성일	@2024년 9월 22일 오후 2:11
🏷 태그	Week 1

## 2-1 파이토치 개요

- 간결하고 빠른 구현성에 장점을 둔 딥러닝 프레임워크



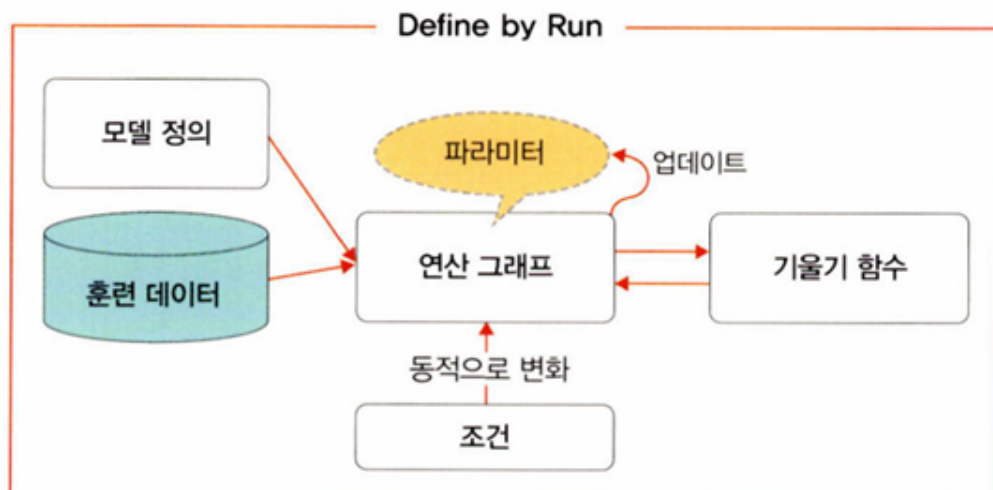
GPU에서 텐서 조작 및 동적 신경망 구축이 가능한 프레임워크

→ GPU: 병렬 연산에서 GPU의 속도는 CPU의 속도보다 훨씬 빠르기 때문에 딥러닝 학습, 기울기를 계산할 때 미분을 쓸 때 빠른 계산이 간으

→ 텐서: 파이토치의 데이터 형태로 단일 데이터 형식으로 된 자료들의 다차원 행렬임. 텐서 뒤에 .cuda()를 사용해서 GPU로 연산을 수행하게 할 수 있음

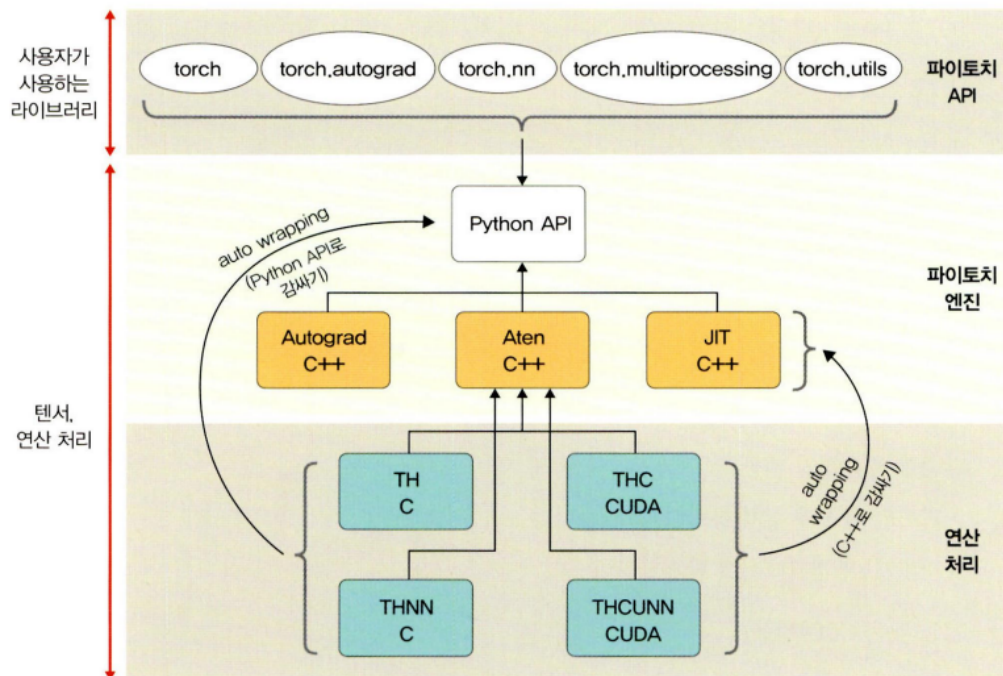
→ 동적신경망: 훈련을 반복할 때마다 네트워크 변경이 가능한 신경망을 의미. 학습 중 은닉층을 추가하거나 제거하는 등 모델 네트워크 조작이 가능. 연산 그래프를 정의하는 것과 동시에 값을 초기화하는 Define by Run 방식을 사용함. 따라서 연산 그래프와 연산을 분리해서 생각할 필요가 없기 때문에 코드를 이해하기 쉬움

▼ 그림 2-1 파이토치 'Define by Run'



인공지능에서 데이터는 **Vector**(숫자들의 리스트, 1차원 배열 형태로 구성됨. 이때 **Matrix**은 2차 배열 형태고 **Tensor**는 3차원 이상의 배열 형태로 행렬의 다차원 표현

## 파이토치의 아키텍처



## 파이토치 API

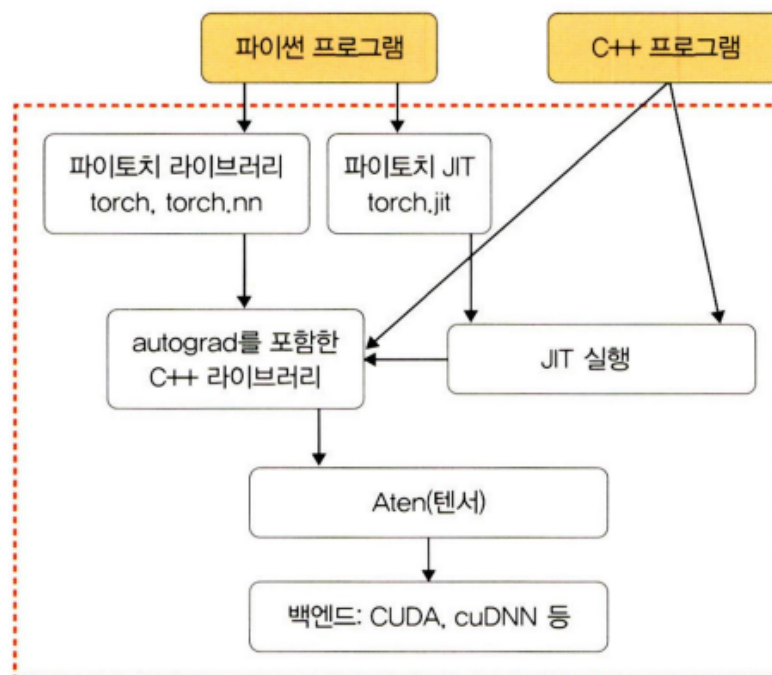
- **\*API: Application Programming Inference**, 두 소프트웨어가 서로 상호작용할 수 있게 도와주는 인터페이스 의미. 하나의 프로그램이 다른 프로그램의 기능이나 데이터를 사용할 수 있도록 미리 정의된 규칙과 명령어 집합을 제공하는 방식. 개발자가 어떤 웹 서비스에서 정보를 가져오고 싶을 때 그 서비스에서 제공하는 API를 사용하면 필요한 데이터를 쉽게 가져올 수 있음. 예시) 날씨 정보를 제공하는 웹사이트에서 다른 앱이 날씨 데이터를 가져오는 경우)
  - Pytorch API는 딥러닝을 위한 오픈소스 라이브러리 Pytorch에서 제공하는 함수와 클래스의 집합
1. **torch**: GPU 지원: 다차원 텐서를 기반으로 다양한 수학적 연산이 가능하도록 함
  2. **torch.autograd**: 자동미분: Tensorflow와 가장 차별화되는 패키지로 일반적으로 신경망에 사소한 조작이 있다면 신경망 구축을 처음부터 다시 시작해야 함. 하지만 파이토치는 자동 미분이라고 하는 기술로 미분 계산을 효율적으로 처리함. 연산 그래프가 즉시 계산되기 때문에 사용자는 다양한 신경망을 적용할 수 있음
  3. **torch.nn**: 신경망 구축&훈련: 합성곱 신경망, 순환 신경망, 정규화 등이 포함됨

4. torch.multiprocessing: 프로세스 전반에 걸쳐 텐서의 메모리 공유가 가능함. 서로 다른 프로세스에서 동일한 데이터/텐서에 대한 접근과 사용에 용이
5. torch.utils: 모델에 데이터를 제공하거나 병목현상을 디버깅하거나 모델의 일부를 검사하기 위한 모듈이 포함됨

## 파이토치 엔진

- Autograd C++: 가중치, 바이어스를 업데이트하는 과정에서 필요한 미분을 자동으로 계산
- Aten C++: C++ 텐서 라이브러리 제공
- JIT C++: 계산을 최적화하기 위한 Just In-Time 컴파일러

(\*컴파일러" 프로그래밍 언어로 작성된 소스코드를 기계가 이해할 수 있는 실행 파일로 변환해주는 프로그램)



## 연산 처리

- 가장 아래 C,Cuda 패키지는 상위의 API에서 할당된 거의 모든 계산을 수행함.
- CPU, GPU를 사용하여 효율적인 다차원 텐서에 대한 연산을 처리함



텐서를 저장할 때는 항상 1차원 배열 형태가 됨

→ 1차원 배열 형태여야만 메모리에 저장할 수 있고 이때 변환된 1차원 배열을 storage라고 함.

→ offset: 텐서에서 첫번째 요소가 storage에 저장된 index

→ stride: 각 차원에서 다음 요소를 얻기 위해 skip이 필요한 storage의 요소 개수로 메모리에서 텐서 레이아웃을 표현하는 것. 요소가 연속적으로 저장되기 때문에 행 중심으로 stride는 항상 1임.

