

Part. 01

Deep Learning

| Tensorflow 2.0의 이해

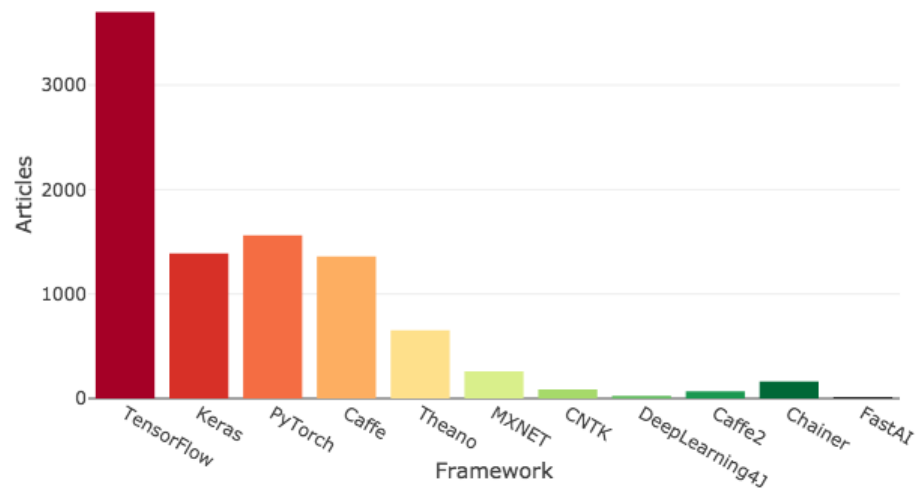
FASTCAMPUS
ONLINE

강사. 신제용

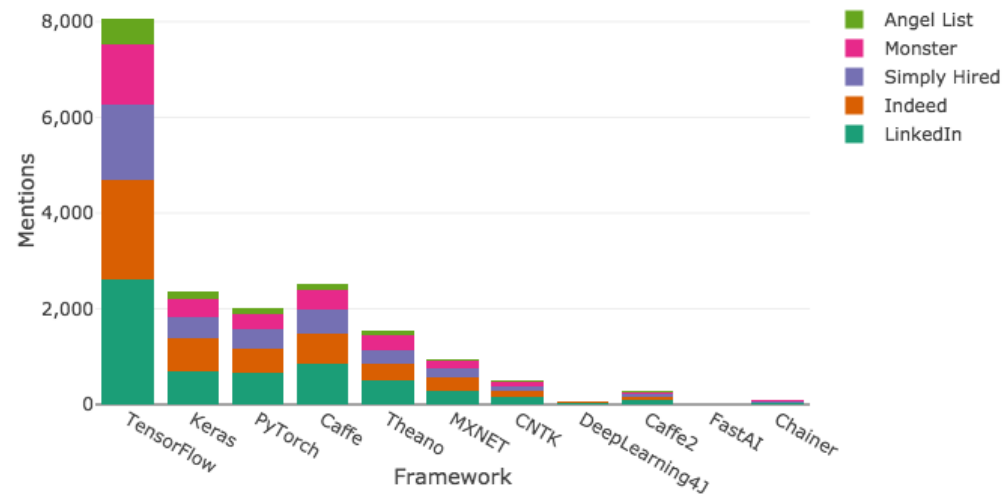
I TensorFlow 2.0



ArXiv Articles



Online Job Listings



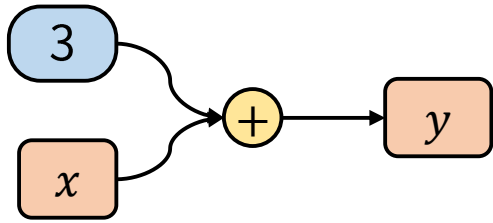
- 구글에서 꾸준히 개발/배포하는 오픈 소스 프레임워크
- 2.0 알파 버전이 미리 공개되어 개발자들이 테스트해볼 수 있음
- 학계는 물론, 산업계에서도 가장 많이 사랑받는 프레임워크




I TensorFlow 1.x

Python

```
x = tf.placeholder(tf.int16)           그래프 정의
y = tf.add(tf.Variable(3), x)
sess = tf.Session()                   세션 생성 및 초기화
sess.run(tf.global_variables_initializer())
y_val = sess.run(y, feed_dict={x: 5}) 연산 실행
```

그래프 정의 (Graph Definition)



- 변수 (Variables) 
- 연산 (Operations) 
- 플레이스홀더 (Placeholders) 

세션 (Session)

- 변수 생성 및 초기화
- 메모리 할당 및 해제
- 연산 리소스 할당 (CPU, GPU 등)
- 연산 실행
- 학습된 변수 저장, 불러오기



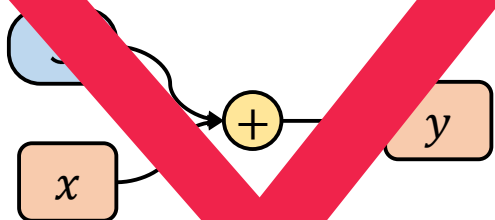
I TensorFlow 2.x

Python AutoGraph와 Eager execution으로 간단하게 구현

```
@tf.function
def op(x):
    return 3 + x

y = op(5)
```

그래프 정의 (Graph Definition)



- 변수 (Variables)
- 연산 (Operations)
- 플레이홀더 (Placeholders)

세션 (Session)

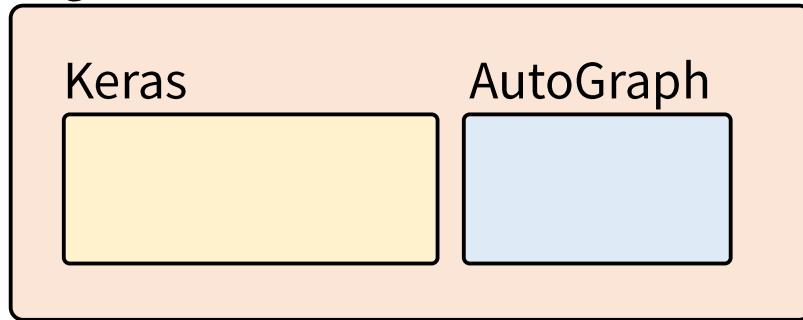
- 변수 생성 및 초기화
- 메모리 할당 및 해제
- 연산 디바이스 할당 (CPU, GPU 등)
- 연산 실행
- 학습된 변수 가져오기 (복합)



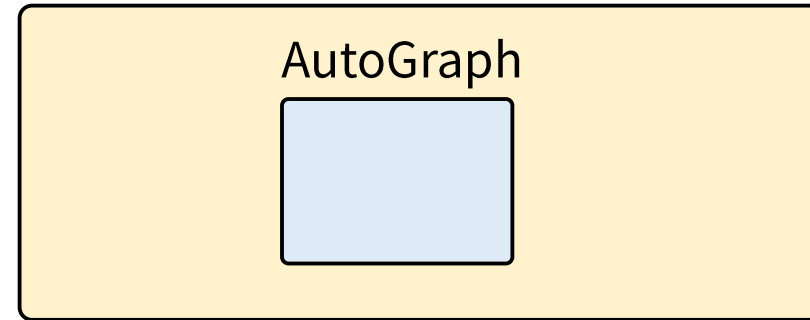
I TensorFlow 2.x의 장점

- Eager execution으로 곧바로 실행되는 코드 구현 (PyTorch의 직관성)
- AutoGraph를 통한 자동 코드 변환 기능으로 기존 세션 방식의 장점을 유지함
- 1.x 버전에서 여러 구현이 혼재하던 API를 Keras를 주축으로 깔끔하게 정리
- Keras 방식의 간단한 구현과 TensorFlow 방식의 유연한 구현을 모두 사용 가능
- Dataset 및 Log 관리의 편의성 향상
- 사용자 수준에 따라 직관성, 편리성, 유연성을 조절하여 자유롭게 사용 가능

Eager execution



Keras



어떤 형태로든, 사용자가 편한대로 유연하게 구성할 수 있다.