

12.1~12.2 텐서플로 훑어보기, 넘파이처럼 사용하기



홍혜선 1:1 채팅

2021.03.06. 01:28 조회 16

댓글 1 URL 복사

□ 12장에서는 텐서플로 저수준 API를 살펴봄

- 자신만의 손실 함수, 지표, 층, 모델, 초기화, 규제, 가중치 규제 등을 만들때 필요할 때를 대비

12. 1. 텐서플로 훑어보기

□ 텐서플로 제공 기능 요약

- (1) GPU 지원 (NumPy는 GPU위에서의 계산을 지원하지 않음)
- (2) (여러 장치/서버에 대한) 분산 컴퓨팅 지원
- (3) 계산 최적화 (ex. 사용하지 않는 노드 가지치기, 독립적 연산을 자동으로 병렬 실행 등)
- (4) 다른 환경에서 호환성 (ex. 리눅스와 안드로이드)
- (5) 모든 종류의 손실함수 쉽게 최소화 (자동 미분, 고성능 옵티마이저 제공)

□ 텐서플로 파이썬 API

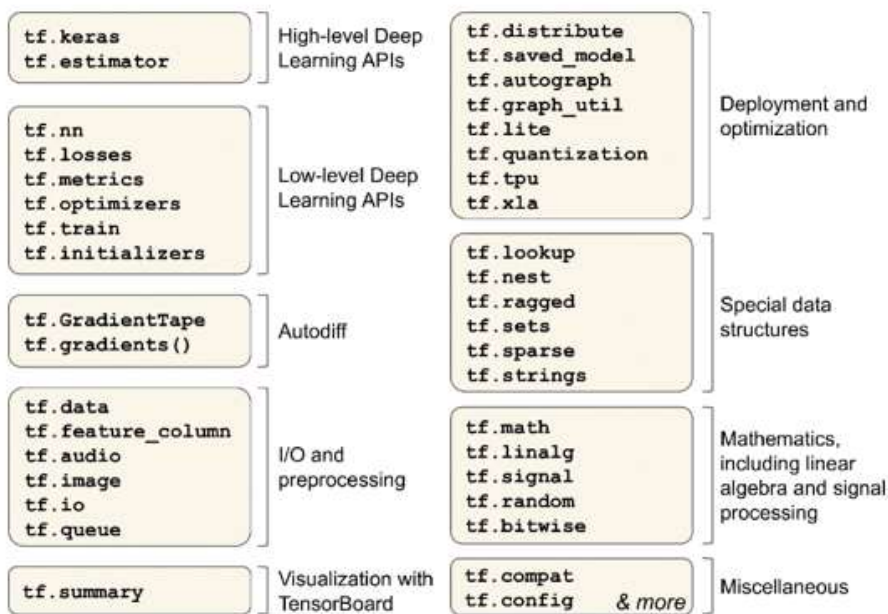


Figure 12-1. TensorFlow's Python API

그림 12-1 텐서플로 파이썬 API

- 많은 연산은 커널이라 부르는 여러 구현을 가짐

- 각 커널은 CPU, GPU, TPU와 같은 특정 장치에 맞추어 만들어짐 (19장에서 GPU와 TPU 사용 방법 설명함)

※ TPU : 딥러닝 연산을 위해 특별하게 설계된 칩

- 텐서플로의 실행 엔진은 여러 디바이스와 서버로 이루어진 분산 환경에서도 효율적 연산을 진행

□ 텐서플로 구조

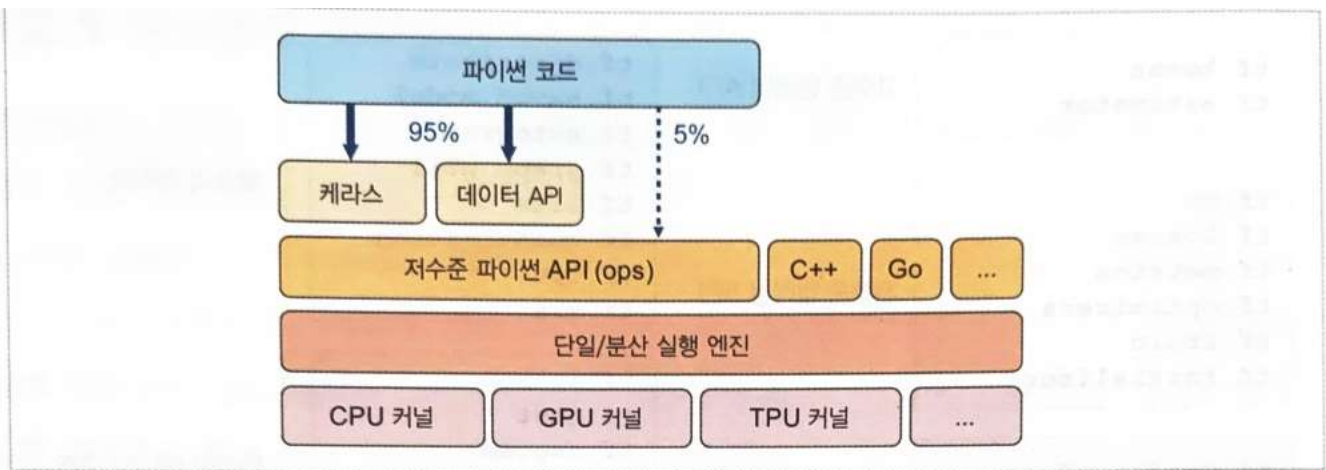


그림 12-2 텐서플로 구조

▣ 참고 사이트

- 시각화를 위한 텐서보드도 제공
- 텐서플로 허브를 사용하면 사전훈련된 신경망을 다운로드하여 재사용 가능
 - : 텐서플로 모델 저장소(<https://github.com/tensorflow/models>)
- 텐서플로 기반 프로젝트 참고 사이트
 - : 텐서플로 리소스 페이지 (<https://www.tensorflow.org/resources> + <https://github.com/joy/awesome-tensor>)
- 머신러닝 논문 참고 사이트
 - : <https://paperswithcode.com/>
- 기술적인 질문에 유용한 사이트
 - : 스택오버플로(<http://stackoverflow.com>)
- 텐서플로 버그 제보 or 새로운 기능 요청 時
 - : <https://github.com/tensorflow/tensorflow>
- 텐서플로에 대한 논의하기 좋은 사이트
 - : <https://hml.info/41>

12. 2. 넘파이처럼 텐서플로 사용하기

- 텐서플로 API는 텐서를 순환시킴
(한 연산에서 다른 연산으로 흐르기 때문에 텐서플로라고 부름)
- 넘파이 ndarray와 비슷
- 텐서는 일반적으로 다차원 배열 (스칼라 값도 가질 수 있음)

12.2.1 텐서와 연산

▣ tf.constant : 텐서 만들기

- tf.Tensor는 크기와 데이터 타입을 가짐
- 인덱스 참조도 넘파이와 비슷
- 모든 종류의 텐서 연산이 가능

```
[4] ▶ ML
#행렬
tf.constant([[1.,2.,3.], [4.,5.,6.]])

<tf.Tensor: shape=(2, 3), dtype=float32, numpy=
array([[1., 2., 3.],
       [4., 5., 6.]], dtype=float32)>

[5] ▶ ML
#스칼라
tf.constant(42)

<tf.Tensor: shape=(), dtype=int32, numpy=42>
```

tf.constant로 텐서를 만들 수 있다.

```
[6] ▶ ML
t = tf.constant([[1.,2.,3.], [4.,5.,6.]])
t.shape

TensorShape([2, 3])
```

tf.Tensor는 데이터 타입과 크기를 가진다.

```
[7] ▶ ML
# tf.add(t,10)과 동일
# -, *와 같은 연산도 지원
t+10

<tf.Tensor: shape=(2, 3), dtype=float32, numpy=
array([[11., 12., 13.],
       [14., 15., 16.]], dtype=float32)>

[8] ▶ ML
tf.square(t)

<tf.Tensor: shape=(2, 3), dtype=float32, numpy=
array([[ 1.,  4.,  9.],
       [16., 25., 36.]], dtype=float32)>

[10] ▶ ML
# @는 행렬곱셈
# tf.matmul() 함수 호출과 동일
t @ tf.transpose(t)

<tf.Tensor: shape=(2, 2), dtype=float32, numpy=
array([[14., 32.],
       [32., 77.]], dtype=float32)>
```

모든 종류의 텐서 연산이 가능하다

□ 기본 수학 연산

- tf.add()
- tf.multiply()
- tf.square()
- tf.exp()
- tf.sqrt()

□ 넘파이와 비슷한 함수

- tf.reshape()
- tf.squeeze()
- tf.tile()

□ 넘파이와 이름 다른 함수

- tf.reduce_mean() = np.mean()
- tf.reduce_sum() = np.sum()
- tf.reduce_max() = np.max()
- tf.mat.log() = np.log()
- tf.transpose(t) = t.T

※ 두 함수에 차이가 있기 때문에 이름이 다름!

ex1) `tf.transpose(t) = t.T`

`tf.transpose(t)` : 전치된 데이터 복사본으로 새로운 텐서가 만들어짐

넘파이의 `t.T` : 동일한 데이터의 전치된 View

ex2) `tf.reduce_sum()` = `np.sum()`

`tf.reduce_sum()` : GPU커널이 원소가 추가된 순서를 보장하지 않는 리듀스 알고리즘 사용

▣ 케라스의 저수준 API

- `keras.backend`에 자체적 저수준 API를 가지고 있음
- `square()`, `exp()`, `sqrt()` 같은 함수 포함
- 단, 텐서플로에서 제공하는 함수의 일부만 지원

```
[12] ▶ ▶ Mi
from tensorflow import keras
K = keras.backend #보통 별칭으로 K를 사용
K.square(K.transpose(t)) + 10

<tf.Tensor: shape=(3, 2), dtype=float32, numpy=
array([[11., 26.],
       [14., 35.],
       [19., 46.]], dtype=float32)>
```

12.2.2 텐서와 넘파이

- 함께 사용하기 편리
- 넘파이 배열로 텐서를 만들 수 있고 반대로 가능, 연산도 호환됨
- 단, 넘파이는 기본 64비트 정밀도 사용, 텐서플로는 32비트 정밀도 사용
(넘파이 배열로 텐서를 만들때, `dtype=tf.float32`로 지정)

```
[14] ▶ ▶ Mi
a = np.array([2.,4.,5.])
tf.constant(a) #텐서로 변환

<tf.Tensor: shape=(3,), dtype=float64, numpy=array([2., 4., 5.])>

[15] ▶ ▶ Mi
t.numpy() #위에서 만든거 넘피로 변환

array([[1., 2., 3.],
       [4., 5., 6.]], dtype=float32)

[16] ▶ ▶ Mi
tf.square(a) #넘파이 배열을 텐서플로 연산을 적용

<tf.Tensor: shape=(3,), dtype=float64, numpy=array([ 4., 16., 25.])>

[19] ▶ ▶ Mi
np.square(t) #텐서를 넘파이 연산에 적용

array([[ 1.,  4.,  9.],
       [16., 25., 36.]], dtype=float32)
```

넘파이와 텐서는 서로 호환이 된다

12.2.3 타입 변환

- 텐서플로는 어떤 타입 변환도 자동으로 수행되지 않음 (성능 감소시킬 수 있기 때문)
- 호환되지 않는 타입의 텐서의 연산은 수행되지 않음

ex) 실수 텐서 + 정수 텐서, 32비트 실수 + 64비트 실수 등 (타입변환 필요 시 `tf.cast()` 사용)

```
[21] > ▶ MI
      t2 = tf.constant(40., dtype=tf.float64)

[22] > ▶ MI
      tf.constant(2.0) + tf.cast(t2, tf.float32)

<tf.Tensor: shape=(), dtype=float32, numpy=42.0>
```

텐서플로는 다른 타입끼리 연산이 허용되지 않으나, `tf.cast()` 함수를 사용하면 가능하다

12.2.4 변수

- `tf.Tensor`는 변경 불가능한 객체. (즉, 텐서의 내용을 바꿀 수 없음)

▣ `tf.Variable`는 변경 가능함

- 변수값 변환 : `assign()` 메서드
- 변수값 증감 : `assign_add()`, `assign_sub()`
- 개별 원소/슬라이스 수정 : `scatter_update()`, `scatter_nd_update()`

```
[23] > ▶ MI
      v = tf.Variable([[1.,2.,3.],[4.,5.,6.]])
      v

<tf.Variable 'Variable:0' shape=(2, 3) dtype=float32, numpy=
array([[1., 2., 3.],
       [4., 5., 6.]], dtype=float32)>

[24] > ▶ MI
      # assign() 메서드 사용하여 변수값 변경 가능
      v.assign(2*v)

<tf.Variable 'UnreadVariable' shape=(2, 3) dtype=float32, numpy=
array([[ 2.,  4.,  6.],
       [ 8., 10., 12.]], dtype=float32)>
```

```
[26] > ▶ MI
      # 변수값 변환
      v[:,2].assign([0.,1.])

<tf.Variable 'UnreadVariable' shape=(2, 3) dtype=float32, numpy=
array([[ 2., 42.,  0.],
       [ 8., 10.,  1.]], dtype=float32)>

[27] > ▶ MI
      # 개별 원소 수정
      v.scatter_nd_update(indices=[[0,0],[1,2]],updates=[100.,200.])

<tf.Variable 'UnreadVariable' shape=(2, 3) dtype=float32, numpy=
array([[100., 42.,  0.],
       [ 8., 10., 200.]], dtype=float32)>
```

`tf.Variable`은 변경 가능하다.

12.2.5 다른 데이터 구조

- 주피터 노트북의 '텐서와 연산' + 부록 F참고

(1) 희소 텐서 : `tf.SparseTensor`

- 0으로 채워진 텐서

(2) 텐서 배열 : tf.TensorArray

- 텐서의 리스트
- 기본은 고정 길이이나 동적으로 변경 가능
- 단, 리스트 內 텐서 크기와 데이터 타입 동일해야 함

(3) 래그드 텐서 : tf.RaggedTensor

- 리스트의 리스트를 나타냄
- 데이터 타입은 동일해야 하나, 리스트의 길이는 다를 수 있음

(4) 문자열 텐서

- tf.string 타입의 텐서
- 유니코드가 아닌 바이트 문자열을 나타냄 (자동으로 UTF-8로 인코딩)
- tf.strings 패키지 : 텐서 사이의 유니코드와 바이트 문자열 변환 (끝에 string's' 주의)

(5) 집합

- 일반적인 텐서로 나타냄

ex) tf.constant([[1,2], [3,4]]) 는 집합 {1,2}와 {3,4}임

- tf.sets 패키지 : 집합 연산을 다룸

(6) 큐 (tf.queue)

- 큐는 단계별로 텐서를 저장
- FIFOQueue, PriorityQueue, RandomShuffleQueue, PaddingFIFOQueue 등

댓글 등록순 최신순

댓글알림



백관구

466쪽에 t[..., 1, tf.newaxis]의 결과가 왜 저렇게 나오는지 궁금합니다!

```
>>> t[..., 1, tf.newaxis]
<tf.Tensor: id=15, shape=(2, 1), dtype=float32, numpy=
array([[2.],
       [5.]], dtype=float32)>
```

2021.03.07. 11:30 답글쓰기

⋮

백관구

댓글을 남겨보세요



등록