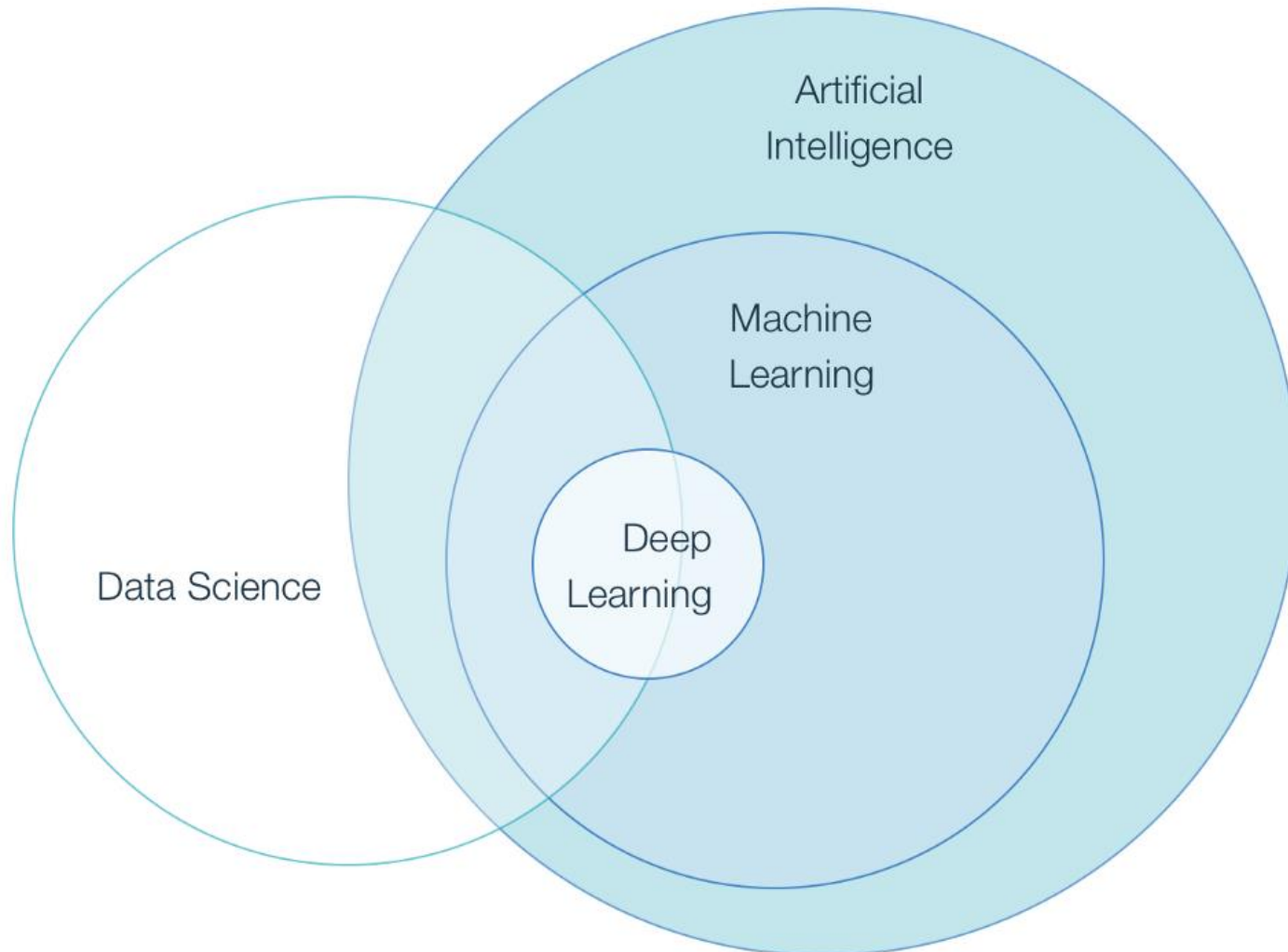


코랩으로 시작하는 텐서플로 기초 프로그래밍

Data Science & AI

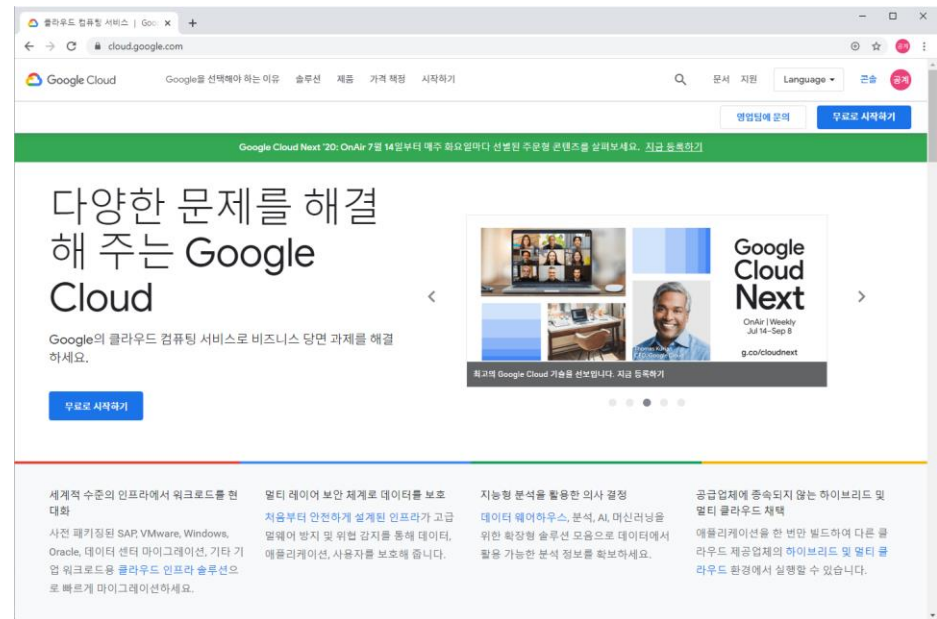


텐서플로

- **Tensorflow**
 - 머신러닝을 위한 오픈소스 플랫폼
 - 가장 널리 쓰이는 딥러닝 프레임워크
 - 지원 언어
 - 파이썬, C++, 자바스크립트, 자바, Go, Swift
- **버전 변화**
 - 2017, 1월: 1.0 알파
 - 2019, 3월: 2.0 알파
 - 2019, 9월: 2.0 정식
 - 즉시 실행 모드(eager execution)가 기본
 - 세션 대신 함수 사용
 - 정식으로 TPU 지원
- **홈페이지**
 - <http://tensorflow.org>

구글의 여러 서비스

- **텐서플로**
- **구글 코랩**
 - 노트북 ipynb의 자유로운 공유
- **구글 클라우드**
 - <https://cloud.google.com/>
 - BigQuery
 - AutoML



추천 제품

Compute Engine
Google의 데이터 센터에서 실행되는 가상 머신

Cloud Storage
안전하고 내구성과 확장성이 뛰어난 객체 스토리지

Cloud SDK
Google Cloud용 명령줄 도구 및 라이브러리

Cloud SQL
MySQL, PostgreSQL, SQL Server 용 관계형 데이터베이스 서비스

Google Kubernetes Engine
컨테이너식 앱 실행을 위한 관리형 환경

BigQuery
비즈니스 민첩성과 통찰력을 위한 데이터 웨어하우스

Cloud CDN
웹 및 동영상 전송을 위한 콘텐츠 전송 네트워크

Dataflow
스트림 및 일괄 처리를 위한 스트리밍 분석

작업
모니터링, 로깅, 애플리케이션 성능 제품군

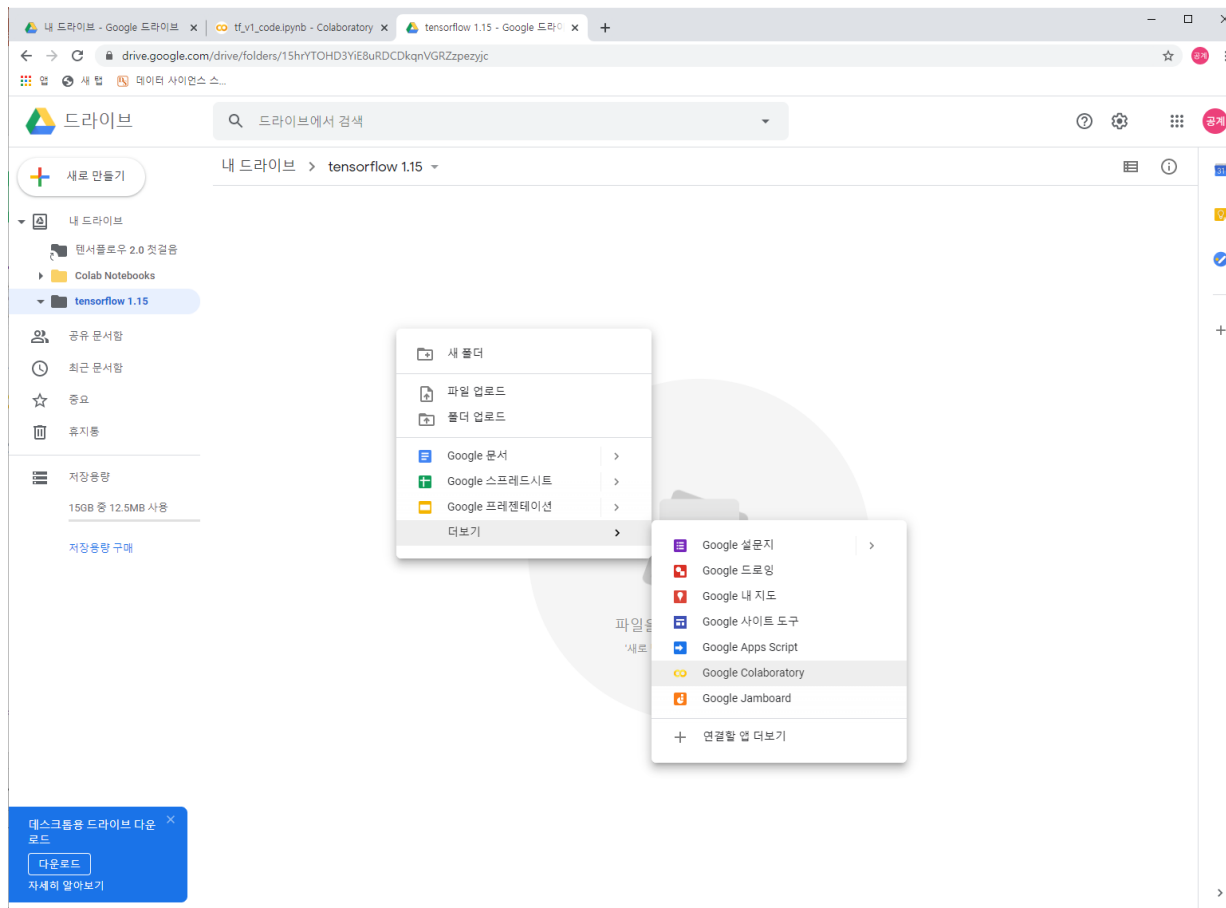
Cloud Run
컨테이너식 앱 실행을 위한 완전 관리형 환경

Cloud Functions
클라우드 서비스 및 앱을 위한 이벤트 기반 컴퓨팅 플랫폼

원하는 사항을 찾을 수 없으신가요?
[모든 제품 보기\(100개 이상\)](#)

코랩 드라이브에서 노트북 파일 열기

- 폴더 tensorflow 1.15 하부에 작성
 - 파일 적당히, tf_1_code.ipynb



Tensorflow 불러오기

- `import tensorflow as tf`
 - `tf.__version__`
- 현재는 자동으로 2.2.0

매직 명령어로 버전 바꾸기

- 코랩에서 쉽게 버전 사용방법
 - %tensorflow_version 1.x
 - %tensorflow_version 2.x
- Import 하기 전에 위 매직 명령어 사용
 - 사용 중에 바꾸려면 '런타임 다시 시작' 후 바로
 - %tensorflow_version 1.x
 - %tensorflow_version 2.x
- 2.2 사용 중에 1.x으로 변경
 - 1. 메뉴, 런타임 | 런타임 다시 시작
 - 단축키: ctrl+M .
 - 2. 바로 실행
 - %tensorflow_version 1.x

```
[1] 1 import tensorflow as tf
```

```
[2] 1 tf.__version__
```

```
↳ '2.2.0'
```

```
[3] 1 # 3.4 텐서플로 2.0 버전 선택
2 try:
3     # %tensorflow_version only exists in Colab.
4     %tensorflow_version 1.x
5 except Exception:
6     pass
7 import tensorflow as tf
8 tf.__version__
```

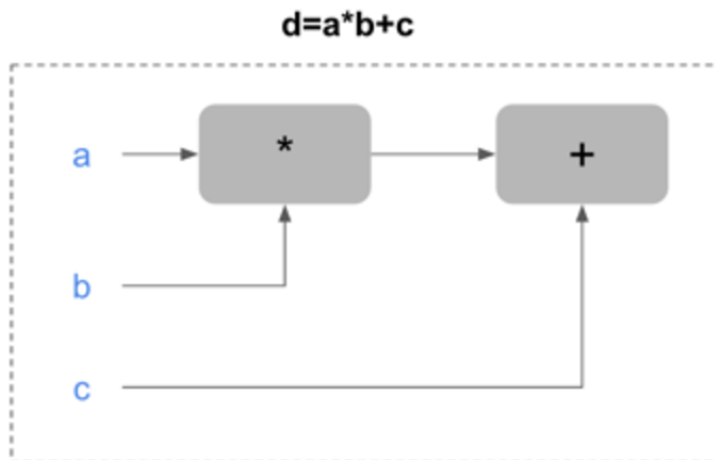
```
↳ TensorFlow is already loaded. Please restart the runtime to change versions.
'2.2.0'
```

```
[1] 1 # 3.4 텐서플로 2.0 버전 선택
2 try:
3     # %tensorflow_version only exists in Colab.
4     %tensorflow_version 1.x
5 except Exception:
6     pass
7
8 import tensorflow as tf
9 tf.__version__
```

```
↳ TensorFlow 1.x selected.
'1.15.2'
```

코랩으로 텐서플로 1.15 코딩

- 코랩에서 1.x 사용 지정
- Hello World에서 시작
- Session
 - 그래프를 실행시키는 객체
 - 만들어진 그래프에 실제 값의 흐름을 수행해 결과가 나오도록 하는 객체



```
[1] 1 # 텐서플로 1.0 버전 선택
    2 try:
    3     # %tensorflow_version only exists in Colab.
    4     %tensorflow_version 1.x
    5 except Exception:
    6     pass
    7
    8 import tensorflow as tf
    9 tf.__version__
```

TensorFlow 1.x selected.
'1.15.2'

```
[2] 1 import tensorflow as tf
    2
    3 hello = tf.constant('Hello World!')
    4 sess = tf.Session()
    5
    6 print(sess.run(hello))
    7
    8 sess.close()
```

b'Hello World!'

```
[3] 1 a = tf.constant(5)
    2 b = tf.constant(3)
    3 sess = tf.Session()
    4
    5 print(sess.run(a+b))
    6 sess.close()
```

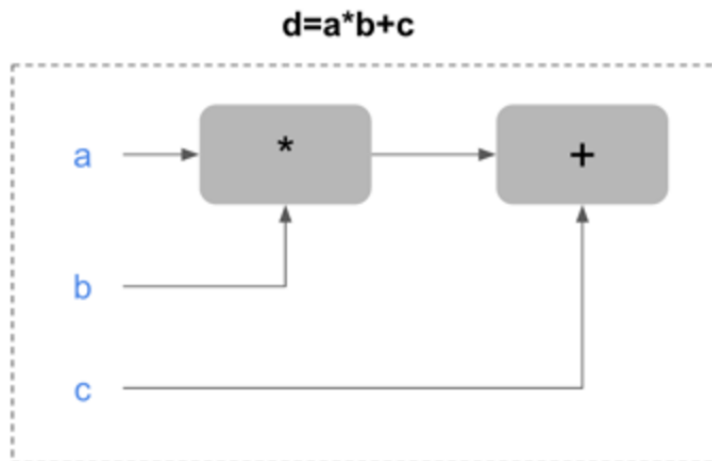
8

구문 with

```
[4] 1 c = tf.constant('Hello, world!')
    2
    3 with tf.Session() as sess:
    4     print(sess.run(c))
```

↳ b'Hello, world!'

```
[5] 1 x = tf.constant(1.)
    2 bool = tf.constant(True)
    3 res = tf.cond(bool, lambda: tf.add(x, 1.), lambda: tf.add(x, 10.))
    4
    5 with tf.Session() as sess:
    6     print( sess.run(res))
```



```
[6] 1 a = tf.constant([5], dtype=tf.float32)
    2 b = tf.constant([10], dtype=tf.float32)
    3 c = tf.constant([2], dtype=tf.float32)
    4
    5 d = a * b + c
    6
    7 with tf.Session() as sess:
    8     print(sess.run(d))
```

↳ [52.]

```
[11] 1 a = tf.constant(5, dtype=tf.float32)
    2 #a = 5
    3 with tf.Session() as sess:
    4     print(sess.run(a))
```

↳ 5.0

텐서를 세션 없이 실행

• 텐서

- 그대로 출력
- 그래프의 정보 출력

```
[ ] 1 x = tf.constant([1, 2, 3])
    2 y = tf.constant([5, 6, 7])
    3
    4 with tf.Session() as sess:
    5     print(sess.run(x+y))
    6
```

➞ [6 8 10]

```
[ ] 1 x = tf.constant([1, 2, 3])
    2 y = tf.constant([[1], [2], [3]])
    3
    4 with tf.Session() as sess:
    5     print(sess.run(x+y))
```

➞ [[2 3 4]
[3 4 5]
[4 5 6]]

```
[ ] 1 a = 2
    2 b = 3
    3 c = tf.add(a, b, name='Add')
    4 print(c)
```

➞ Tensor("Add_4:0", shape=(), dtype=int32)

```
[ ] 1 a = 2
    2 b = 3
    3 c = tf.add(a, b, name='Add')
    4
    5 sess = tf.Session()
    6 print(sess.run(c))
    7 sess.close()
    8
```

➞ 5

```
[ ] 1 a = 2
    2 b = 3
    3 c = tf.add(a, b, name='Add')
    4
    5 with tf.Session() as sess:
    6     print(sess.run(c))
```

➞ 5

다양한 연산

- `tf.add`
- `tf.reduce_mean`

```
[ ] 1 import tensorflow as tf
    2 x = 2
    3 y = 3
    4 add_op = tf.add(x, y, name='Add')
    5 mul_op = tf.multiply(x, y, name='Multiply')
    6 pow_op = tf.pow(add_op, mul_op, name='Power')
    7 useless_op = tf.multiply(x, add_op, name='Useless')
    8
    9 with tf.Session() as sess:
   10     pow_out, useless_out = sess.run([pow_op, useless_op])
   11     print(pow_out, useless_out)
```

➞ 15625 10

```
[ ] 1 a = tf.constant(2)
    2 b = tf.constant(3)
    3 c = tf.constant(5)
    4
    5 # Some more operations.
    6 mean = tf.reduce_mean([a, b, c])
    7 sum = tf.reduce_sum([a, b, c])
    8
    9 with tf.Session() as sess:
   10     print("mean =", sess.run(mean))
   11     print("sum =", sess.run(sum))
```

➞ mean = 3
sum = 10

즉시 실행

- 1.0에서는 즉시 실행이 기본이 아님
- 다음을 실행, 즉시 실행 가능
 - `tf.InteractiveSession()`
 - 결과 보기
 - 메소드 `eval()`

```
[ ] 1 tf.executing_eagerly()
```

↪ False

```
[2] 1 interactive_sess = tf.InteractiveSession()
    2 a = 2
    3 b = 3
    4 c = tf.add(a, b, name='Add')
    5 print(c.eval())
```

↪ 5

```
[3] 1 x = tf.constant(1.)
    2 bool = tf.constant(True)
    3 res = tf.cond(bool, lambda: tf.add(x, 1.), lambda: tf.add(x, 10.))
    4
    5 print(res.eval())
```

↪ 2.0

텐서플로 2.0으로 실행

- 매직 명령어 사용
 - 또는 그대로 import
- 기본이 즉시 실행

```
[1] 1 # 텐서플로 2.0 버전 선택
    2 try:
    3     # %tensorflow_version only exists in Colab.
    4     %tensorflow_version 2.x
    5 except Exception:
    6     pass
    7
    8 import tensorflow as tf
    9 tf.__version__
```

↪ '2.2.0'

```
[6] 1 tf.executing_eagerly()
```

↪ True

좀 더 쉬운 2.0

- **텐서 출력**
 - 첫 항목에 내용
 - 값만 보려면
 - **메소드 numpy()**

```
[7] 1 import tensorflow as tf
    2
    3 hello = tf.constant('Hello World!')
    4 print(hello)
    5 print(hello.numpy())
```

```
↳ tf.Tensor(b'Hello World!', shape=(), dtype=string)
   b'Hello World!'
```

```
[8] 1 a = tf.constant(5)
    2 b = tf.constant(3)
    3 print(a+b)
    4 print((a+b).numpy())
```

```
↳ tf.Tensor(8, shape=(), dtype=int32)
   8
```

```
[9] 1 c = tf.constant('Hello, world!')
    2
    3 with tf.Session() as sess:
    4     print(sess.run(c))
```

```
↳ -----
AttributeError                                Traceback (most recent call last)
<ipython-input-9-7a2fa5301d92> in <module>()
      1 c = tf.constant('Hello, world!')
      2
----> 3 with tf.Session() as sess:
      4     print(sess.run(c))
```

AttributeError: module 'tensorflow' has no attribute 'Session'