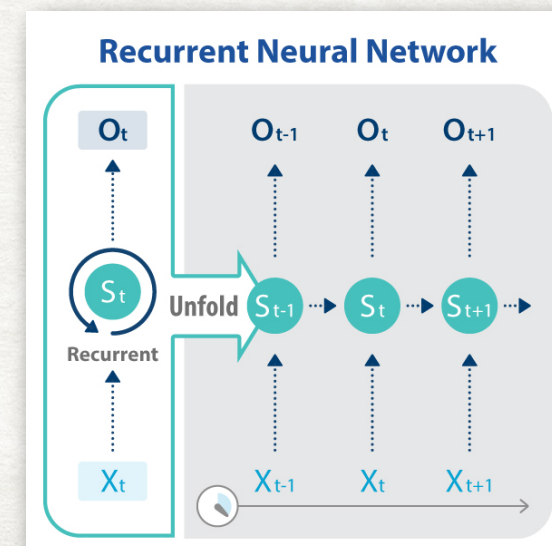
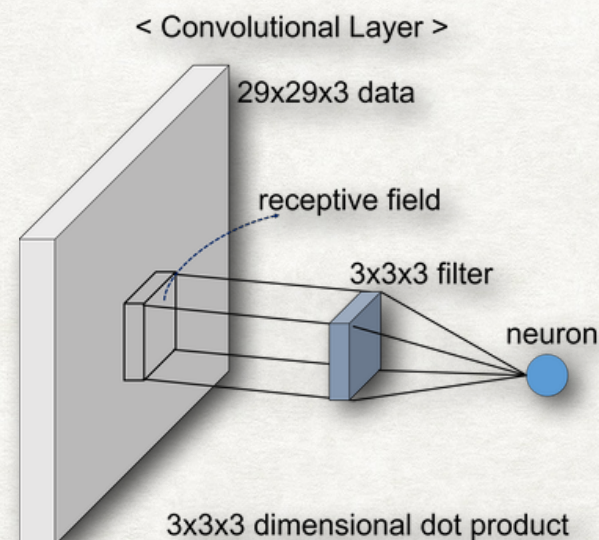
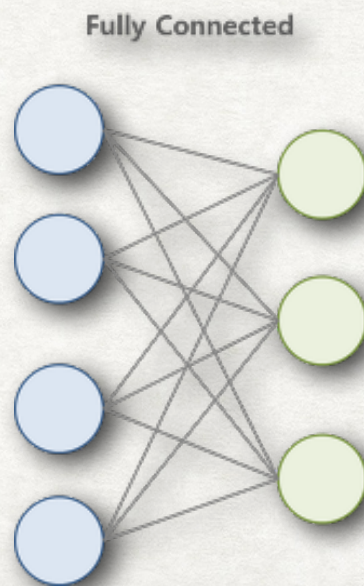


텐서플로

텐서플로

- 딥러닝 아키텍처를 쉽게 구축할 수 있도록 구글에서 만든 머신러닝 프레임워크
- **머신러닝 프레임워크란?**
머신러닝을 위한 다양한 기능들을 제공하는 환경
- **선언형 프로그래밍 언어**



명령형 vs 선언형

```
a = 3  
b = 4  
c = a + b  
print(c)
```

```
a = tf.constant(3)  
b = tf.constant(4)  
c = a + b  
print(c)
```

7

Tensor("add:0", shape=(), dtype=int32)

선언과 동시에 계산을 수행하는 **명령형**
계산에 대한 선언만 하고 **계산은 나중에** 수행하는 **선언형**

TensorFlow 버전확인

```
>>> import tensorflow as tf  
>>> print(tf.__version__)
```

2.0.0

TensorFlow 실행

```
import tensorflow as tf_new
tf = tf_new.compat.v1

g = tf.Graph()
with g.as_default() as graph:
    hello = tf.constant("Hello TensorFlow!")
    sess = tf.Session()
    print(sess.run(hello))
```

b'Hello TensorFlow!'

텐서(Tensor) 실습

```
import tensorflow as tf_new
tf = tf_new.compat.v1

g = tf.Graph()
with g.as_default() as graph:
    node1 = tf.constant(3.0, tf.float32)
    node2 = tf.constant(4.0, tf.float32)
    node3 = tf.add(node1, node2)
    print("node1:", node1)
    print("node2:", node2)
    print("node3: ", node3)
```

```
node1: Tensor("Const:0", shape=(), dtype=float32)
node2: Tensor("Const_1:0", shape=(), dtype=float32)
node3: Tensor("Add:0", shape=(), dtype=float32)
```



덧셈 결과를 보려면?

텐서(Tensor) 실습

```
import tensorflow as tf_new
tf = tf_new.compat.v1

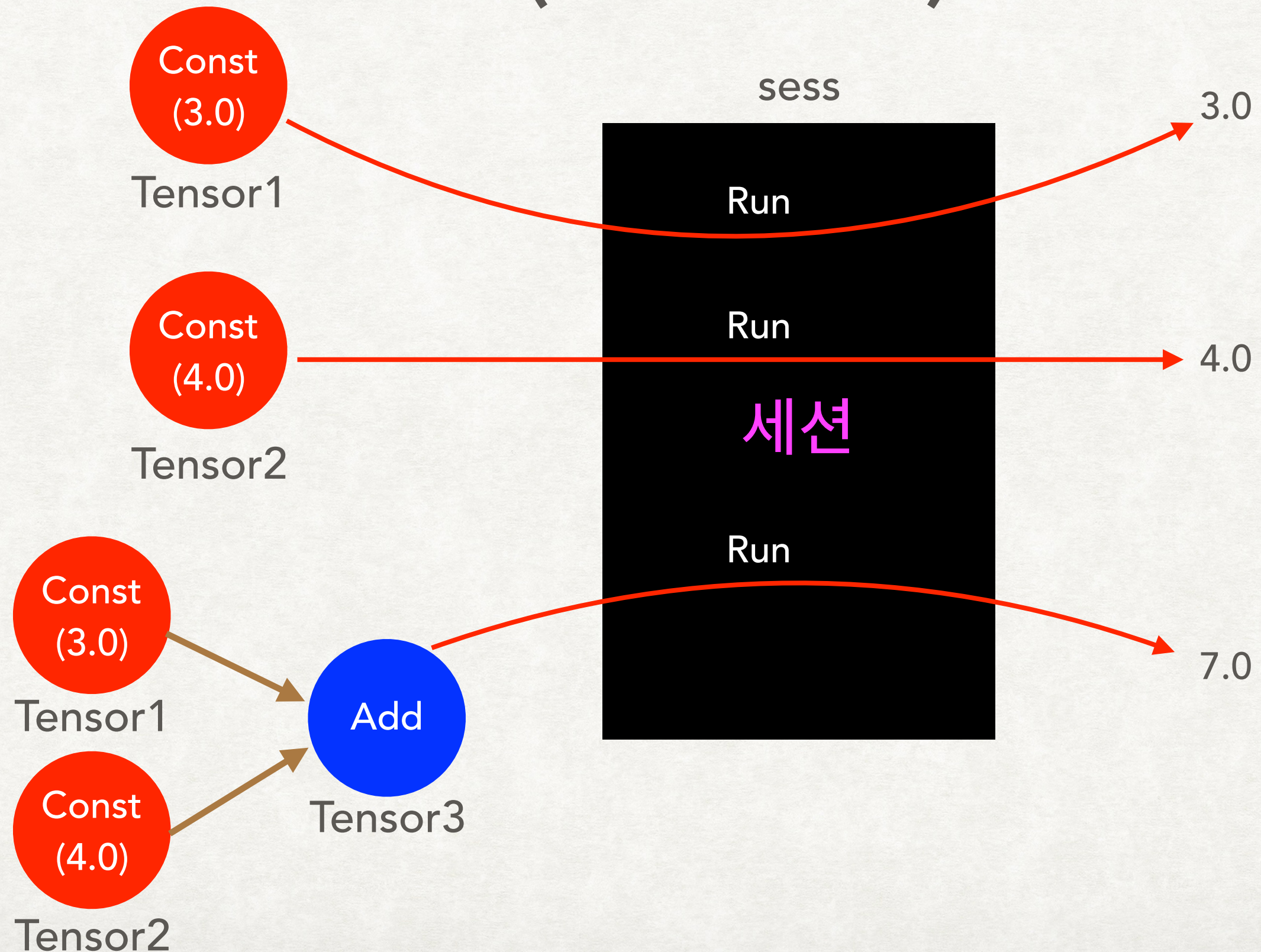
g = tf.Graph()
with g.as_default() as graph:
    node1 = tf.constant(3.0, tf.float32)
    node2 = tf.constant(4.0, tf.float32)
    node3 = tf.add(node1, node2)
    sess = tf.Session()
    print("node1:", sess.run(node1))
    print("node2:", sess.run(node2))
    print("node3:", sess.run(node3))
```

그래프를 실행할 수 있는
환경(세션) 만들기

세션을 통해서
그래프를 실행(Run)

```
node1: 3.0
node2: 4.0
node3: 7.0
```


세션(Session)



Placeholder

- 프로그래머가 관리하는 변수
- 그래프를 실행할 때 직접 변수에 값을 넣어야 한다.(feed_dict 옵션 사용)

```
import tensorflow as tf_new
tf = tf_new.compat.v1
```

```
g = tf.Graph()
```

```
with g.as_default() as graph:
```

```
    x1 = tf.placeholder(tf.float32)
```

```
    x2 = tf.placeholder(tf.float32)
```

```
    c = x1 + x2      # c = tf1.add(a, b)
```

```
    sess = tf.Session()
```

```
    print(sess.run(c, feed_dict={x1: 3, x2: 4.5}))
```

```
    print(sess.run(c, feed_dict={x1: [1, 3], x2: [2, 4]}))
```

실수형(float32) 데이터를
저장할 수 있는 x1, x2 만들기

feed_dict 옵션을 이용해서
데이터 넣어주기

Variable

- 텐서플로가 관리하는 변수
- 텐서플로가 계산해서 수정하는 변수(주로 학습에 사용)

```
import tensorflow as tf_new
tf = tf_new.compat.v1
tf.set_random_seed(777)
```

텐서플로가 관리하는 변수 생성
초기값 필요

```
g = tf.Graph()
with g.as_default() as graph:
```

(초기값에 대한 선언만 한 상태)

```
    a = tf.Variable(tf.random_normal([10]))
```

```
    b = tf.Variable(tf.zeros([10]))
```

```
    sess = tf.Session()
```

```
    sess.run(tf.global_variables_initializer())
```

```
    print(sess.run(a))
```

선언한 초기값 적용

```
    print(sess.run(b))
```