

An abstract network diagram with various colored nodes (blue, orange, grey) connected by thin lines, set against a light blue background with a fine grid pattern.

[R Bio⁺] TensorFlow with R

Rloha Corp.
Data Analysis & Education Dept.

김승욱 대표

CEO@Rloha.io

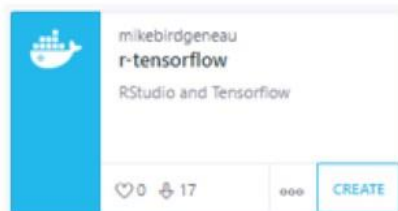
An abstract network diagram on a light blue background. It features several nodes of different sizes and colors (orange, blue, and grey) connected by thin grey lines. The nodes are arranged in a way that suggests a complex, interconnected system. A central white rectangular box with the text "Getting Start" is overlaid on the diagram.

Getting Start

01 초기 세팅



윈도우10 + Docker



이미지(Image) 로드 방법

- ▶ cmd 실행
- ▶ `docker pull mikebirdgeneau/r-tensorflow`
- ▶ `docker run --rm -p 8787:8787 -v /Users/Encaion/Documents/R_docker:/home/rstudio mikebirdgeneau/r-tensorflow tensorflow`
port 설정 연결 할 로컬 주소(작업 폴더) 이미지 이름

01 초기 세팅



로컬 저장소 연결

Settings

General

Shared Drives

Advanced

Network

Proxies

Daemon

Diagnose & Feedback

Reset

Docker is running

Shared Drives

Select the local drives you want to be available to your containers.

Shared	Drive
<input checked="" type="checkbox"/>	C
<input type="checkbox"/>	D

Microsoft PowerShell

> docker run --rm -v c:/Users:/data alpine ls /data

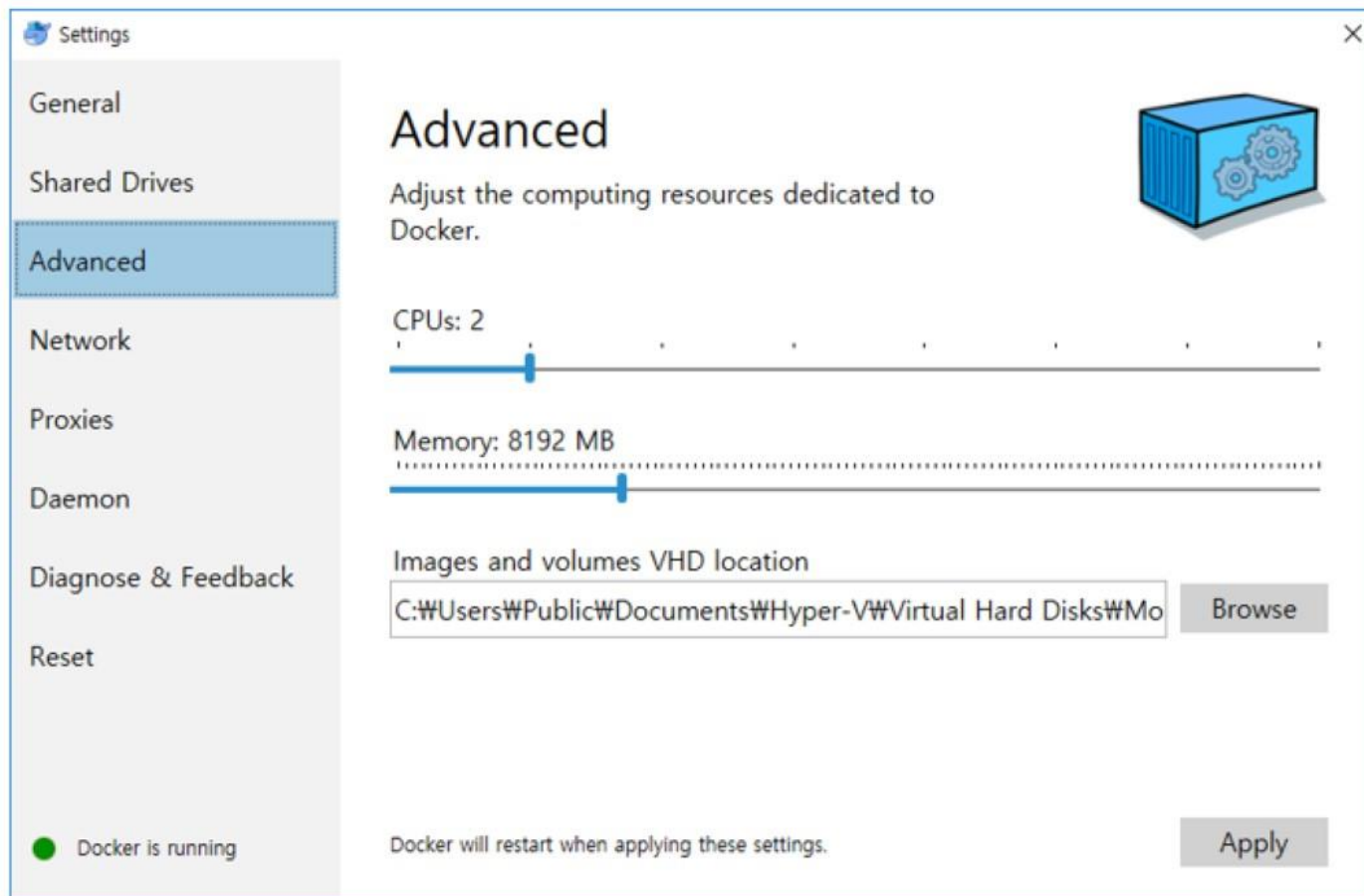
[Reset credentials...](#)

Apply

01 초기 세팅

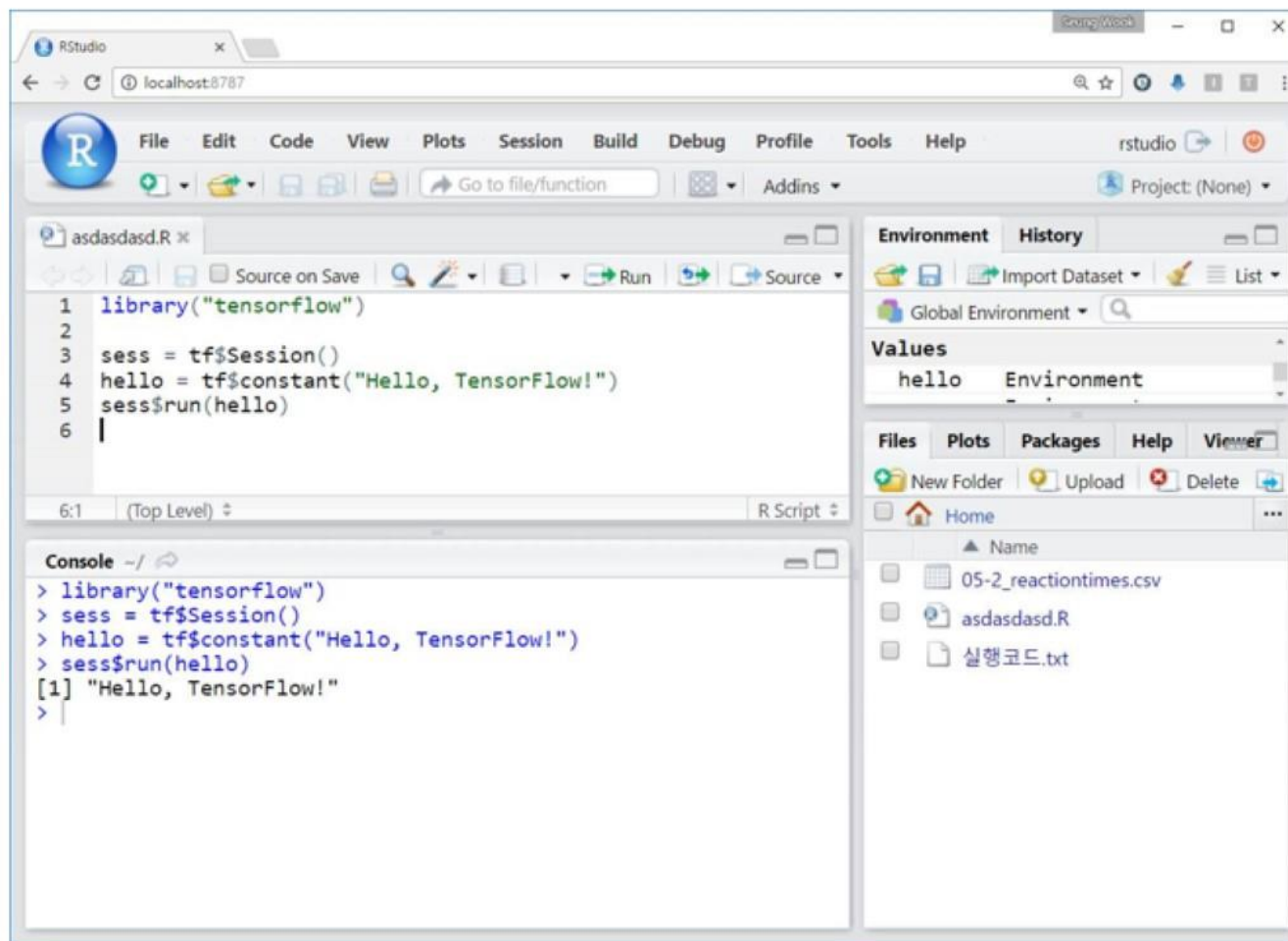


자원 할당



02 시작

</> 예제 코드 실행



The screenshot shows the RStudio interface with a script editor, console, and file explorer. The script editor contains the following R code:

```
1 library("tensorflow")
2
3 sess = tf$Session()
4 hello = tf$constant("Hello, TensorFlow!")
5 sess$run(hello)
6 |
```

The console shows the output of the script:

```
> library("tensorflow")
> sess = tf$Session()
> hello = tf$constant("Hello, TensorFlow!")
> sess$run(hello)
[1] "Hello, TensorFlow!"
> |
```

The file explorer on the right shows the following files:

- 05-2_reactiontimes.csv
- asdasdasd.R
- 실행코드.txt

03 모듈과 함수



모듈(Module)

```
> tf
Module(tensorflow)
```

- ▶ "tf"라고 명명된 TensorFlow 모듈을 기초로 기능 실행 및 문법 구성



함수(Function)

```
> tf$Session
function (...)
{
  dots <- py_resolve_dots(list(...))
  result <- py_call_impl(callable, dots$args, dots$keywords)
  if (convert) {
    result <- py_to_r(result)
    if (is.null(result))
      invisible(result)
    else result
  }
  else {
    result
  }
}
<environment: 0x6a23170>
attr(,"py_object")
<class 'tensorflow.python.client.session.Session'>
```

- ▶ 파이썬의 "."은 R의 "\$"로 대체
- ▶ "\$" 특수 기호로 함수 또는 하위 모듈 사용 가능

04 숫자



숫자형(Numeric Type)

```
> 1L  
[1] 1  
> 1  
[1] 1
```

```
> 2123L  
[1] 2123  
> 2123  
[1] 2123
```

- ▶ R의 편리한 기능(auto casting) 때문에 야기되는 문제를 방지하기 위하여 R integer literal 사용
※ 정수 → 실수, 실수 → 정수 로 자동 변환 되는 경우 발생
- ▶ 특히 tensor dimension과 같이 무조건 정수형이 입력되어야 하는 경우 존재



숫자형(Numeric Type) 예시

```
1 tf$nn$conv2d(x_data, W, strides = c(1L, 1L, 1L, 1L), padding = "SAME")  
2  
3 flags$DEFINE_integer("max_steps", 2000L, "Number of steps to run trainer.")  
4 flags$DEFINE_integer("hidden1", 128L, "Number of units in hidden layer 1.")  
5 flags$DEFINE_integer("hidden2", 32L, "Number of units in hidden layer 2.")
```


04 숫자



숫자 리스트(Numeric Lists)

```
> (x = tf$placeholder(tf$float32, list(NULL, 784L)))  
Tensor("Placeholder_4:0", shape=(?, 784), dtype=float32)  
> (W = tf$Variable(tf$zeros(list(784L, 10L))))  
Variable(shape=(784, 10), dtype=float32_ref)  
  
> (b = tf$Variable(tf$zeros(list(10L))))  
Variable(shape=(10,), dtype=float32_ref)
```

05 배열과 색인



배열(Array)의 종류

1 1차원 – Vector

```
> c(1.0, 2.0, 3.0, 4.0)
[1] 1 2 3 4
```

2 2차원 – Matrix

```
> matrix(c(1.0, 2.0, 3.0, 4.0),
+        nrow = 2,
+        ncol = 2)
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

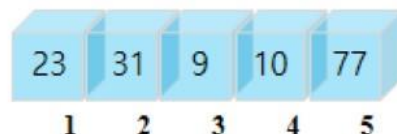
3 3차원 – Array

```
> array(rep(999, 2*2*2), dim = c(2, 2, 2))
, , 1
     [,1] [,2]
[1,]   999   999
[2,]   999   999
, , 2
     [,1] [,2]
[1,]   999   999
[2,]   999   999
```

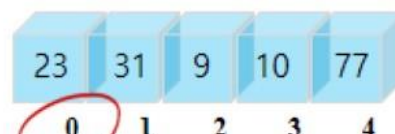


색인(Index)

1 R



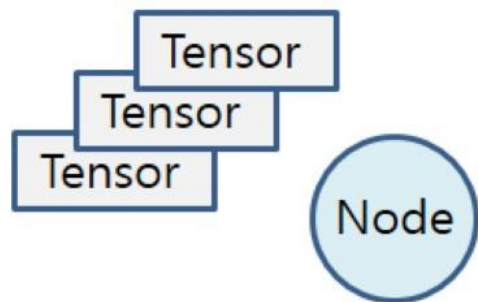
2 TensorFlow



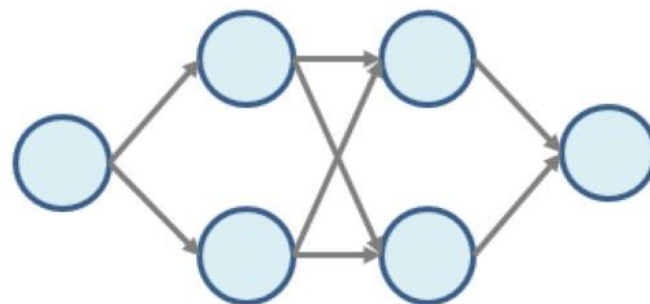
06 기타



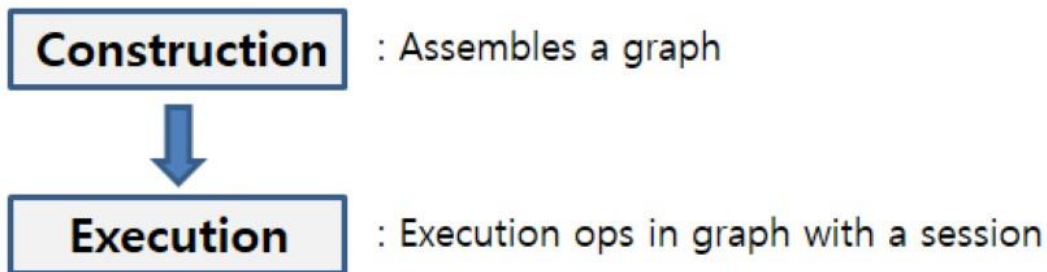
윈도우10 + Docker



Node \rightarrow op(Operation)



Phase of TensorFlow Program



An abstract network diagram with several nodes of different colors (blue, orange, grey) and sizes, connected by thin grey lines. The nodes are scattered across the slide, with some forming small clusters. A central white rectangular box contains the title text.

Variables: Creation, Initialization, Saving, and Loading

07 변수



개요

- ▶ 모델 학습 시 파라미터를 hold 또는 update를 하기 위하여 사용
- ▶ 텐서를 포함하는 메모리 완충 기억 공간(buffer)
- ▶ 초기화(initializing)가 반드시 필요
- ▶ 학습 중 또는 완료 시 저장소(disk)에 저장 가능



학습 범위

- 1 tf\$Variable class
- 2 tf\$train\$Saver class

07 변수

생성(Creation)

```
> weights = tf$Variable(tf$random_normal(shape(784L, 200L),  
+                                     stddev=0.35),  
+                               name="weights")  
> weights  
Variable(shape=(784, 200), dtype=float32_ref)  
  
> biases = tf$Variable(tf$zeros(shape(200L),  
+                               name = "biases"))  
> biases  
Variable(shape=(200,), dtype=float32_ref)
```

tf\$Variable() 의 기능

- ▶ variable op는 변수의 값을 hold
- ▶ 초기화(initializer) op는 대상 변수를 초기값으로 설정하며, 이는 tf\$assign op와 동일
- ▶ zeros, biases 와 같은 초기값 op 또한 그래프에 추가 가능

07 변수



장비 할당(Device Placement)

```
1 # Pin a variable to CPU
2 with(tf$device("/cpu:0"), {
3     v = tf$Variable(tf$zeros(shape(200L),
4                               name = "biases"))
5 })
6
7 # Pin a variable to GPU
8 with(tf$device("/gpu:0"), {
9     v = tf$Variable(tf$zeros(shape(200L),
10                               name = "biases"))
11 })
12
13 # Pin a variable to a particular parameter server task
14 with(tf$device("/job:ps/task:7"), {
15     v = tf$Variable(tf$zeros(shape(200L),
16                               name = "biases"))
17 })
```

참고 사이트: https://www.tensorflow.org/tutorials/using_gpu

07 변수



초기화(Initialization)

- ▶ Variable initializer은 모델 구동 전 실행 필수
- ▶ Variable initializer 대신 checkpoint file로 대체 가능



예제

```
1 # Create two variables.
2 weights = tf$Variable(tf$random_normal(shape(784L, 200L), stddev = 0.35),
3                       name = "weights")
4 biases = tf$Variable(tf$zeros(shape(200), name="biases"))
5
6 # Add an op to initialize the variables.
7 init_op = tf$global_variables_initializer()
8
9 # Later, when launching the model
10 with(tf$Session() %as% sess, {
11     # Run the init operation.
12     sess$run(init_op)
13     # Use the model
14     # ...
15 })
```

07 변수



타 변수를 활용한 초기화(Initialization)

- ▶ 타 변수의 초기값을 활용하여 원하는 변수 초기화
- ▶ `tf$global_variables_initializer()` 함수는 모든 변수 초기화(사용 시 주의)



예제

```
1 # Create a variable with a random value.
2 weights = tf$Variable(tf$random_normal(shape(784L, 200L), stddev = 0.35),
3                        name="weights")
4
5 # Create another variable with the same value as 'weights'.
6 w2 = tf$Variable(weights$initialized_value(), name = "w2")
7
8 # Create another variable with twice the value of 'weights'
9 w_twice = tf$Variable(weights$initialized_value() * 2.0, name = "w_twice")
```

07 변수



저장 및 복구

- ▶ 가장 쉬운 모델 저장 방법은 `tf$train$Saver` 활용
- ▶ `saver` 객체는 `op`를 실행하는 메소드 제공
- ▶ `saver` 객체는 저장하거나 불러오기 위한 체크포인트 파일 주소 제공



예제(저장)

```
1 # Create some variables.
2 v1 = tf$Variable(..., name="v1")
3 v2 = tf$Variable(..., name="v2")
4
5 # Add an op to initialize the variables.
6 init_op = tf$global_variables_initializer()
7
8 # Add ops to save and restore all the variables.
9 saver = tf$train$Saver()
10
11 # Later, launch the model, initialize the variables, do some work,
12 # save the variables to disk.
13 with(tf$Session() %as% sess, {
14     sess$run(init_op)
15     # Do some work with the model.
16     save_path = saver$save(sess, "/tmp/model.ckpt")
17 })
```


07 변수



예제(복구)

```
1 # Create some variables.
2 v1 = tf$Variable(..., name = "v1")
3 v2 = tf$Variable(..., name = "v2")
4
5 # Add ops to save and restore all the variables.
6 saver = tf$train$Saver()
7
8 # Later, launch the model, use the saver to restore variables from disk, and
9 # do some work with the model.
10 with(tf$Session() %as% sess, {
11     # Restore variables from disk.
12     saver$restore(sess, "/tmp/model.ckpt")
13     cat("Model restored.\n")
14     # Do some work with the model
15     ...
16 })
```

07 변수



저장 또는 복구 변수 선택

- ▶ `tf$train$Saver()` 내부 파라미터 미지정 시 해당 함수는 생성된 모든 변수를 저장
- ▶ 변수이름 지정 시 학습된 특정 모델의 파라미터 세트 별도 저장 가능



예제

```
1 # Create some variables.
2 v1 = tf$Variable(..., name = "v1")
3 v2 = tf$Variable(..., name = "v2")
4
5 # Add ops to save and restore only 'v2' using the name "my_v2"
6 saver = tf$train$Saver(dict(my_v2 = v2))
7 # Use the saver object normally after that.
```