

Machine Learning Engineer Course

Day 20

- Deep Learning Framework 1 – TensorFlow -



DIVE INTO CODE

Thursday September 2, 2021
DIOP Mouhamed



Agenda

- 1 Check-in**
- 2 How to proceed**
- 3 Quick Review**
- 4 TensorFlow**
- 5 Assignment**
- 6 TensorFlow– Sample Code**
- 7 Check-out**



Check-in

3 minutes Please post the following point to Zoom chat.

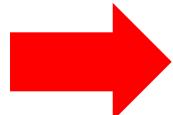
Q. What did you learn in the previous week?
(Anything is fine.)



How to proceed - Objective

What is the purpose?

1. Understand and be able to use TensorFlow
2. Solve the previous problems using TensorFlow



Let's learn the basics of TensorFlow here

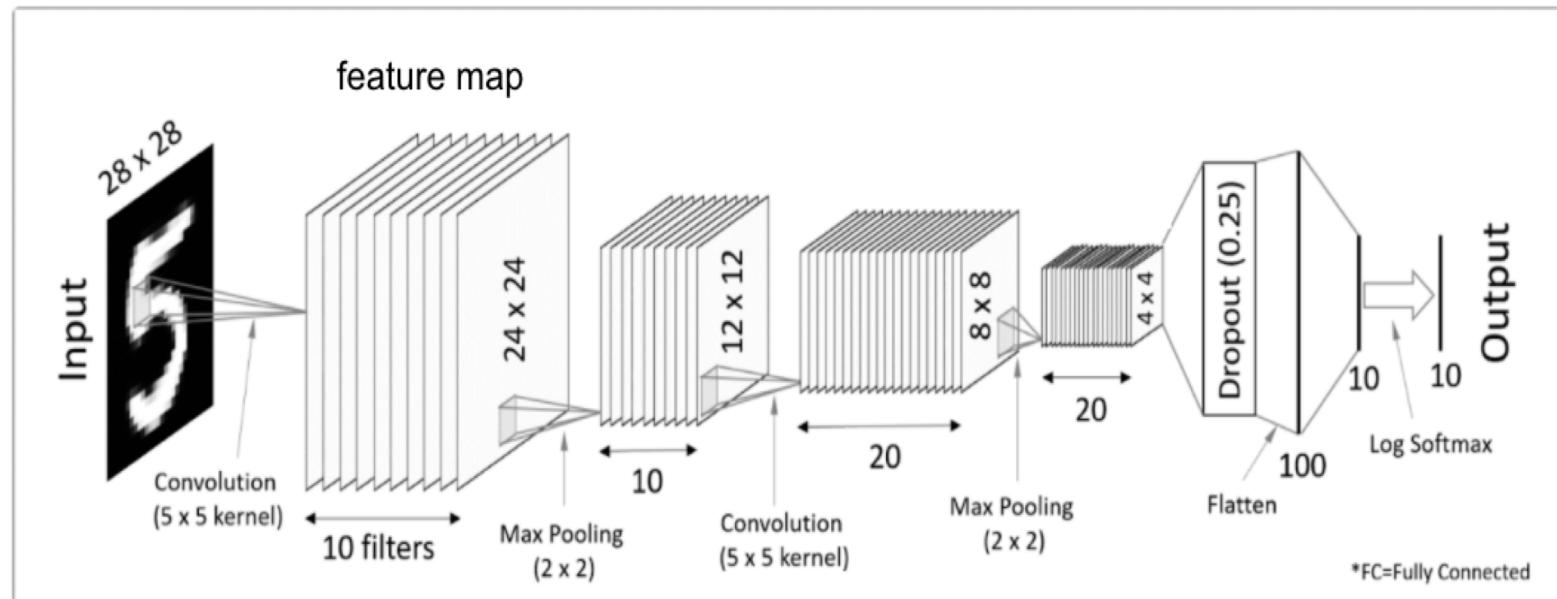


Quick Review (CNN)

What is a convolutional neural network?

A network consisting of convolutional layers with a sparse structure. It was devised to mimic the process by which humans recognize patterns from visual information.

In the architecture, we have a **feature extraction** part and a **classification** part.





What is a framework?

In **Python**, a framework is a piece of software that functions as a framework for developing machine learning, deep learning, and **web** applications using **Python**.

It can also be described as **a set of libraries** (a collection of multiple programs with high versatility) that provides the basic functions of a program, including its control methods.

<https://valiancesolutions.com/difference-framework-library/>



Types of frameworks

For Deep Learning
Frameworks for Deep Learning

Caffe

DL4J
Deeplearning4j

Microsoft
CNTK

mxnet

Purine

torch

Caffe2

TensorFlow

theano

Chainer

PYTORCH

K
KERAS

Mocha.jl
julia

MatConvNet

MINERVA

OpenDeep

Pylearn2

<https://developer.nvidia.com/deep-learning-frameworks>



Types of frameworks

For Deep Learning Frameworks for Deep Learning

Caffe



DL4J
Deeplearning4j



Microsoft
CNTK

theano

K
KERAS



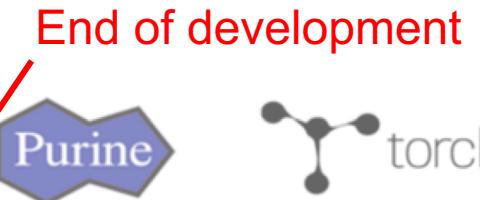
MatConvNet

MINERVA

OpenDeep

Pylearn2

mxnet



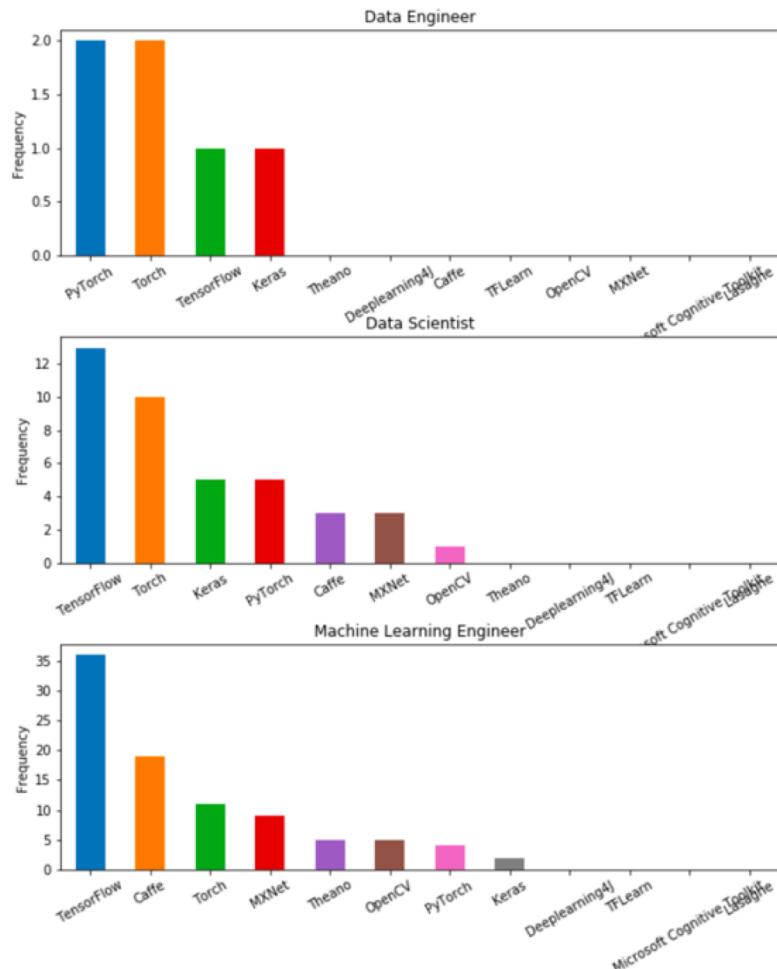
PYTORCH

<https://developer.nvidia.com/deep-learning-frameworks>



How often the framework is used

Deep Learning Frameworks Distribution



※November 2018 article

<https://towardsdatascience.com/what-does-an-ideal-data-scientists-profile-look-like-7d7bd78ff7ab>



What is TensorFlow?

Tensorflow is an open software library for numerical computation and large-scale machine learning developed by Google Brain.

It is a data flow programming language that describes a series of connections between data (directed graph) and expresses the movement of data (data flow).

The previous version of TensorFlow used **Define-and-Run**.
Define-by-Run is adopted in ver. 2.0.



Knowledge used in TensorFlow

Calculation graph (data flow graph)

The flow of computation is described by a directed graph structure.

Idea :

Once a neural network builds a computational graph, it can be trained by back-propagating on it.

Case Study :

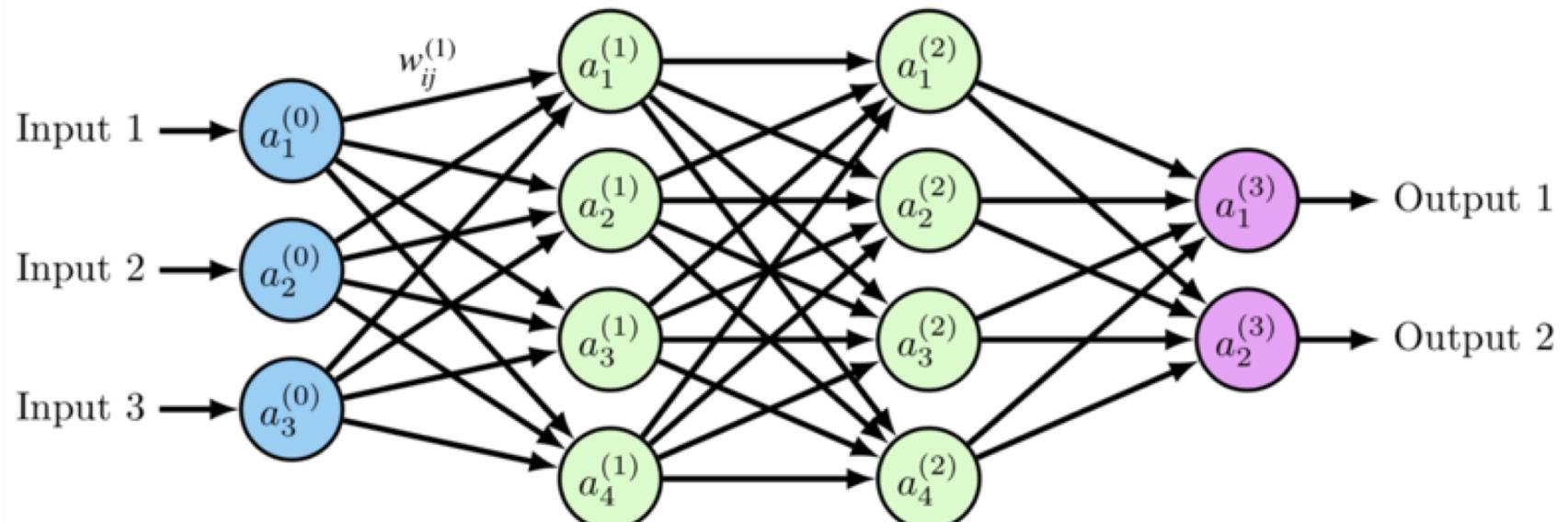
<http://dianne.intec.ugent.be/#gettingstarted>

Prerequisite :

Since the route of the calculation that the data went through is recorded (on the flip side, nothing else is recorded), only the parameters that correspond to that calculation graph are updated.



Knowledge used in TensorFlow

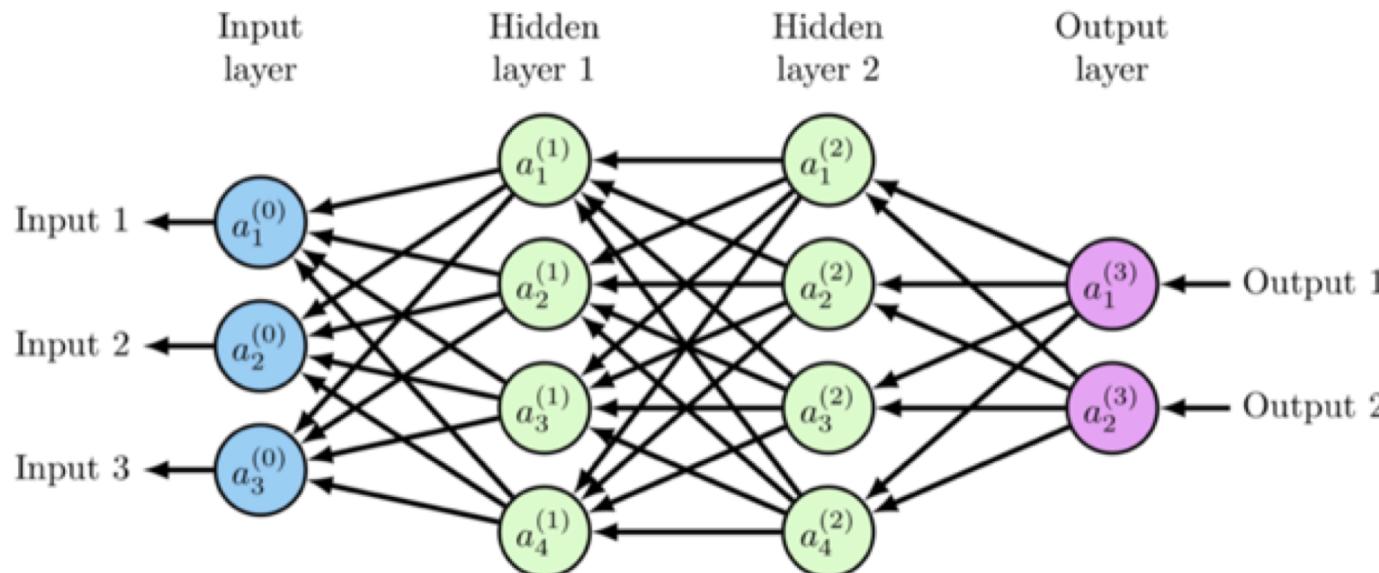


$$\hat{y} = \sigma[W^{(3)}\sigma[W^{(2)}\sigma[W^{(1)}a^{(0)} + b^{(1)}] + b^{(2)}] + b^{(3)}]$$



Knowledge used in TensorFlow

$$\frac{\delta E}{\delta a_j^{(1)}} = \sum_{k \in H2} w_{jk} \frac{\delta E}{\delta z_k^{(2)}} \quad \frac{\delta E}{\delta a_k^{(2)}} = \sum_{l \in out} w_{kl} \frac{\delta E}{\delta z_l^{(3)}}$$



$$\frac{\delta E}{\delta a_l^{(3)}} = a_l^{(3)} - t_l$$

$$\frac{\delta E}{\delta z_l^{(3)}} = \frac{\delta E}{\delta a_l^{(3)}} \frac{\delta a_l^{(3)}}{\delta z_l^{(3)}}$$

$$\frac{\delta E}{\delta z_j^{(1)}} = \frac{\delta E}{\delta a_j^{(1)}} \frac{\delta a_j^{(1)}}{\delta z_j^{(1)}} \quad \frac{\delta E}{\delta z_k^{(2)}} = \frac{\delta E}{\delta a_k^{(2)}} \frac{\delta a_k^{(2)}}{\delta z_k^{(2)}}$$



Building TensorFlow

How to construct a computational graph

Frameworks share the common feature of constructing computational graphs, but depending on how they are constructed, they can be classified into two categories.

Define-and-Run

Execute the calculation after building the calculation graph

Caffe



Microsoft
CNTK

theano

mxnet



Define-by-Run

Construction and execution of calculation graph at the same time



PYTORCH



Building TensorFlow

How to construct a computational graph

① Define - by - RUN system

Define - by - Run

```
1 import numpy as np  
2 a = np.ones(10)  
3 b = np.ones(10) * 2  
4 c = b * a  
5 d = c + 1
```

Idea :

Like Python, it executes one line at a time.

It is said to be interpreted (executed while sequentially interpreting).

case :

When $c = b * a$ is executed, the calculation is carried out and a history of the calculation is generated.



Building TensorFlow

Entering data with `__call__`.
constructing a computational
graph for the first time

```
1 class IrisNN(chainer.Chain):
2     def __init__(self):
3         super(IrisNN, self).__init__(
4             l1 = L.Linear(4,100),
5             l2 = L.Linear(100,100),
6             l3 = L.Linear(100,200),
7             l4 = L.Linear(200,100),
8             l5 = L.Linear(100,3))
9
10    def __call__(self,x):
11        prob = np.random.randn()
12        h = self.l1(x)
13        h = F.relu(h)
14        if prob > 0:
15            h = F.relu(self.l2(h))
16        else:
17            h = F.relu(self.l3(h))
18            h = F.relu(self.l4(h))
19        h = self.l5(F.dropout(F.relu(h),ratio=0.5))
20
21    return h
```



Building TensorFlow

How to construct a computational graph

② Define - and - RUN system

Idea :

The computation procedure is translated into the form of a graph (definition of the computation graph), and then **the computation is executed only after the Session is performed**. It is said to be compiler-like (the program functions by translating it into executable machine language at once and executing that machine language).

case :

When $C = b * a$ is executed, no calculation is done.



Building TensorFlow

Define - and - run (Pseudocode)

```
1 A = Variable('A')
2 B = Variable('B')
3 C = B * A
4 D = C + Constant(1)
5 # compiles the function
6 f = compile(D)
7 d = f(A=np.ones(10), B=np.ones(10)*2)
```



In Tensorflow, a Session object is defined here.
and execute the calculation.



Session

One of the Tensorflow classes that executes the constructed computational graph (or part of it).

Idea :

A session takes an operation object that constitutes a computation graph as an argument and executes it.

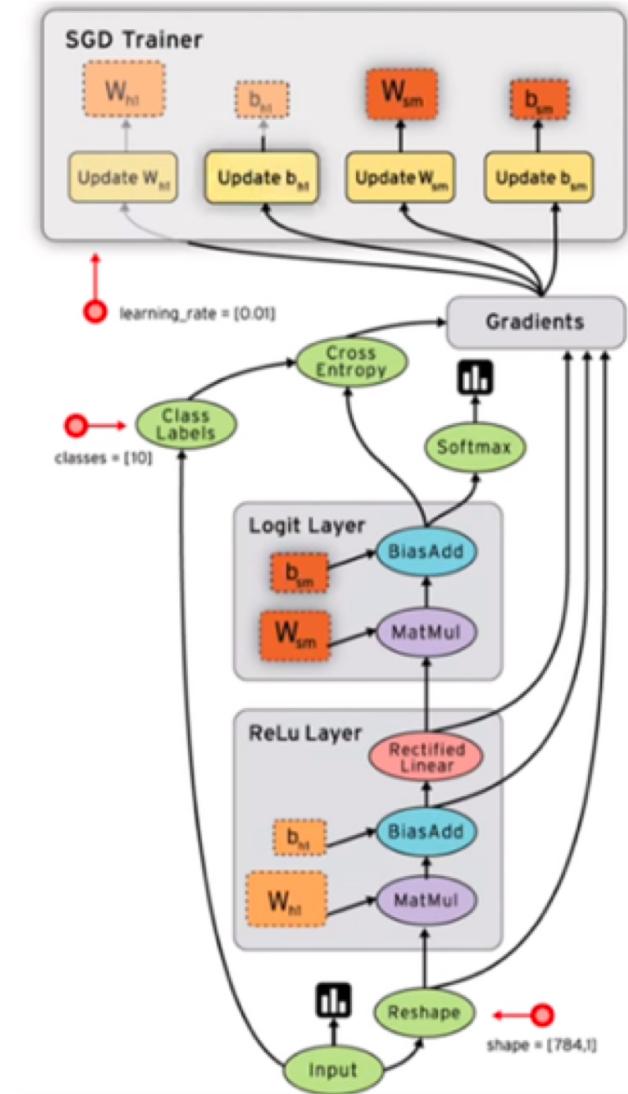
Prerequisite :

The session uses the runtime system (which implements the behavior of the execution model) to access the device. To release the physical resources occupied at this time, use the with block or use sess.close() to terminate this session.

<https://www.tensorflow.org/guide/graphs>



Session





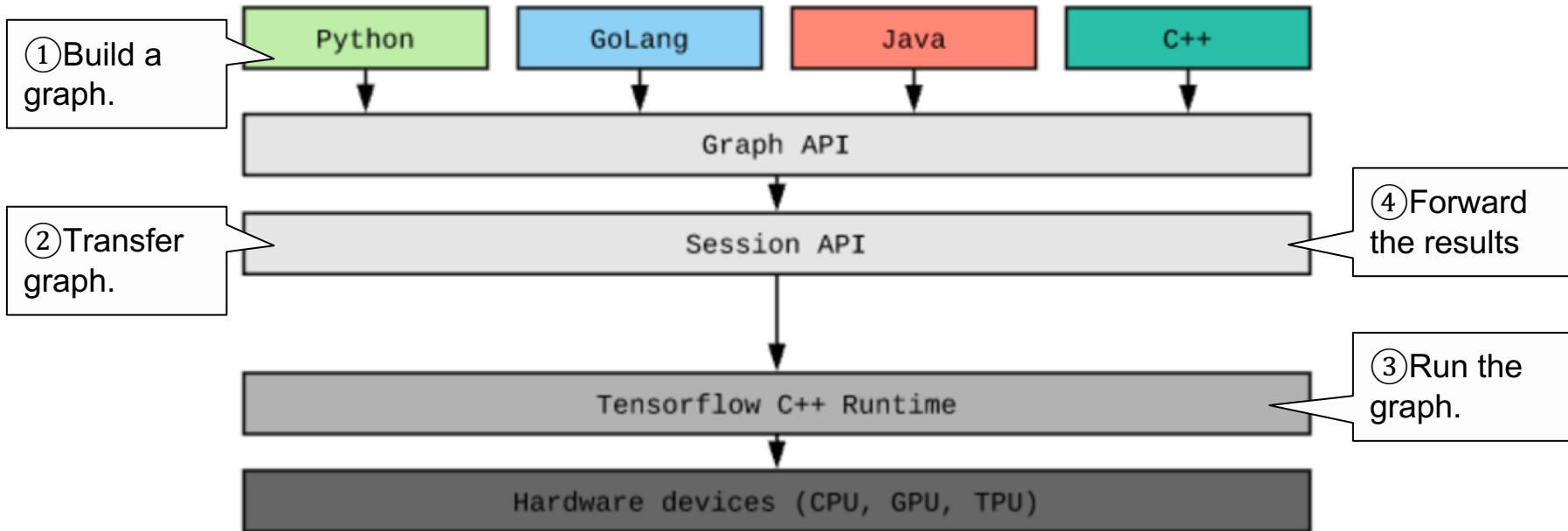
Session

Usage :

The general procedure for using TensorFlow is to first construct a computational graph using Python or other programs, and then execute tf.

low layer :

In this case, the **C++ runtime** (compiled application) is used to perform the actual calculation.
The connection to this runtime is called a session.





Session

TensorFlow session is provided by the following classes.

① tf.Session

- Session is defined independently from the computation graph.

② tf.InteractiveSession

- Default session for interactive operations.
- It is necessary to create an instance before constructing a computation graph.



Session

① tf.Session :

tf.Session

```
a = tf.constant(2.0)
b = tf.constant(1.5)
mul = tf.multiply(a, b)

with tf.Session():
    print(mul.eval())
```

After building the
computational graph
Access the contents

3.0

run() extracts the contents of an Operation
eval() extracts the contents of Tensor



Session

② tf.InteractiveSession :

`tf.InteractiveSession`

```
sess = tf.InteractiveSession()

a = tf.constant(2.0)
b = tf.constant(1.5)
mul = tf.multiply(a, b)
mul.eval()

sess.close()
```

During computational graph construction you can access the contents.

3.0

Contents of Tensor("Mul_10:0", shape=(), dtype=float32)

run() extracts the contents of an Operation
eval() extracts the contents of Tensor



About TensorFlow2.0

In version 2.0, tf.Session will be gone.

Originally, Eager mode was introduced in TensorFlow 1.5 (after January 26, 2018), and there was also a specification that allowed calculations without session; from TensorFlow 2.0, this Eager mode will be the default.

Eager mode

```
import tensorflow as tf
import tensorflow.contrib.eager as tfe

tfe.enable_eager_execution()

a = tf.constant(2.0)
b = tf.constant(1.5)
mul = tf.multiply(a, b)

print(mul)
print(mul.numpy())
```

JupyterNotebook
When running in
Restart the kernel

Graph mode

```
import tensorflow as tf

a = tf.constant(2.0)
b = tf.constant(1.5)
mul = tf.multiply(a, b)

with tf.Session() as sess:
    sess.run(mul)
    print(mul)
    print(mul.eval())
```



About TensorFlow2.0

Instead of tf.Session, we will use tf.function.

<https://github.com/tensorflow/community/blob/master/rfcs/20180918-functions-not-sessions-20.md>

In TensorFlow 2.0, functions with @tf.function (decorator) are compiled. 2.0 also allows you to mix Eager mode and Graph mode.

EagerExecution (TensorFlow1.x)

```
import tensorflow as tf
import tensorflow.contrib.eager as tfe

tfe.enable_eager_execution()

a = tf.Variable(1.0)
b = tf.Variable(1.0)

def f():
    a.assign(2.0)
    b.assign(3.0)
    return a + b
print(f().numpy())
```

5.0

tf.function (TensorFlow2.0)

```
a = tf.Variable(1.0)
b = tf.Variable(1.0)
@tf.function
def f():
    a.assign(2.0)
    b.assign(3.0)
    return a + b
print(f().numpy())
```

5.0



Bonus

Tensorflow official website

<https://www.tensorflow.org/>



Sample code

How to solve problems "Deep Learning Framework 1"

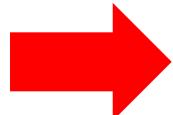
- [Problem 1] Look back at Scratch
- [Problem 2] Consider the correspondence between Scratch and TensorFlow
- [Problem 3] Create a model of Iris using all three objective variables
- [Problem 4] Create a House-Prices model
- [Problem 5] Create a MNIST model



Sprint 13 – TensorFlow

Explanation about this Sprint is given but please try it on your own first.

Sprint 13 – TensorFlow



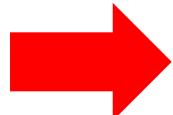
Please work on your own after class and submit your assignments on DIVER.



Sprint 13 – TensorFlow

A Sample Code of this Sprint is given but please try it on your own.

Sprint 13 – TensorFlow



Please work on your own after class and submit your assignments on DIVER.



ToDo by next class

Next class will be Zoom : Thursday September 9, 2021 19:30 ~ 20:30

ToDo: Keras

<https://dive.diveintocode.jp/curriculums/1976>



Check-out

3 minutes Please post the following point to Zoom chat.

Q. Current feelings and reflections
(joy, anger, sorrow, anticipation, nervousness, etc.)



Thank You For Your Attention

