

PyTorch Basics

TEAMLAB director

최성철

**딥러닝을 할 때
코드는 처음부터 다 짤다?**

죽을 수도 있습니다.

그래서 남이 만들 것을 쓴다.
프레임워크



DEEP
LEARNING




theano



PYTORCH



TensorFlow

	Keras 	TensorFlow 	PyTorch 
Level of API	high-level API ¹	Both high & low level APIs	Lower-level API ²
Speed	Slow	High	High
Architecture	Simple, more readable and concise	Not very easy to use	Complex ³
Debugging	No need to debug	Difficult to debugging	Good debugging capabilities
Dataset Compatibility	Slow & Small	Fast speed & large	Fast speed & large datasets
Popularity Rank	1	2	3
Uniqueness	Multiple back-end support	Object Detection Functionality	Flexibility & Short Training Duration
Created By	Not a library on its own	Created by Google	Created by Facebook ⁴
Ease of use	User-friendly	Incomprehensive API	Integrated with Python language
Computational graphs used	Static graphs	Static graphs	Dynamic computation graphs ⁵

<https://datasciencecareer.wordpress.com/2020/12/09/ml03-pytorch-vs-tensorflow/>

```

# Start training
with tf.Session() as sess:

    # Run the initializer
    sess.run(init)

    # Fit all training data
    for epoch in range(training_epochs):
        for (x, y) in zip(train_X, train_Y):
            sess.run(optimizer, feed_dict={X: x, Y: y})

        # Display logs per epoch step
        if (epoch+1) % display_step == 0:
            c = sess.run(cost, feed_dict={X: train_X, Y: train_Y})
            print("Epoch:", '%04d' % (epoch+1), "cost=", "{:.9f}".format(c), \
                  "W=", sess.run(W), "b=", sess.run(b))

    print("Optimization Finished!")
    training_cost = sess.run(cost, feed_dict={X: train_X, Y: train_Y})
    print("Training cost=", training_cost, "W=", sess.run(W), "b=", sess.run(b), '\n')

    # Graphic display
    plt.plot(train_X, train_Y, 'ro', label='Original data')
    plt.plot(train_X, sess.run(W) * train_X + sess.run(b), label='Fitted line')
    plt.legend()
    plt.show()

```



```

train_X = numpy.asarray([3.3, 4.4, 5.5, 6.71, 6.93, 4.168, 9.779, 6.182, 7.59, 2.167,
                          7.042, 10.791, 5.313, 7.997, 5.654, 9.27, 3.1])
train_Y = numpy.asarray([1.7, 2.76, 2.09, 3.19, 1.694, 1.573, 3.366, 2.596, 2.53, 1.221,
                          2.827, 3.465, 1.65, 2.904, 2.42, 2.94, 1.3])
n_samples = train_X.shape[0]

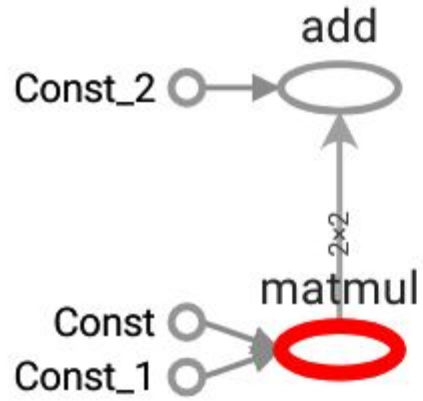
# tf Graph Input
X = tf.placeholder("float")
Y = tf.placeholder("float")

# Set model weights
W = tf.Variable(rng.randn(), name="weight")
b = tf.Variable(rng.randn(), name="bias")

# Construct a linear model
pred = tf.add(tf.multiply(X, W), b)

# Mean squared error
cost = tf.reduce_sum(tf.pow(pred-Y, 2))/(2*n_samples)
# Gradient descent
# Note, minimize() knows to modify W and b because Variable objects are trainable=True by default
optimizer = tf.train.GradientDescentOptimizer(learning_rate).minimize(cost)

```



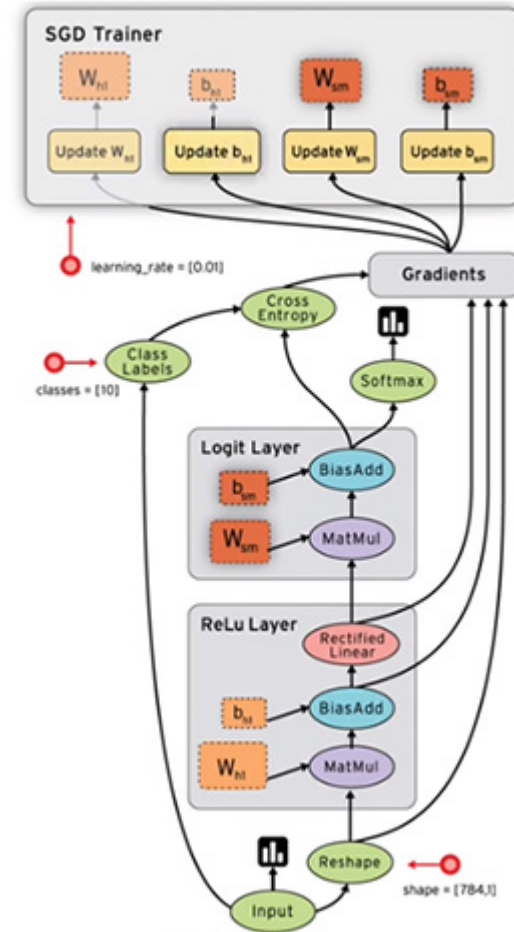
matmul
 Operation: MatMul

Attributes (3)
 T {"type": "DT_FLOAT"}
 transpose_a {"b": false}
 transpose_b {"b": false}
 Device /device:GPU:0

Inputs (2)
 ○ Const 2x2
 ○ Const_1 2x2

Outputs (1)
 ○ add 2x2

Remove from main graph



When you try to print
something in Tensorflow



* 🔄 PyTorch released *

DL practitioners -





Andrej Karpathy

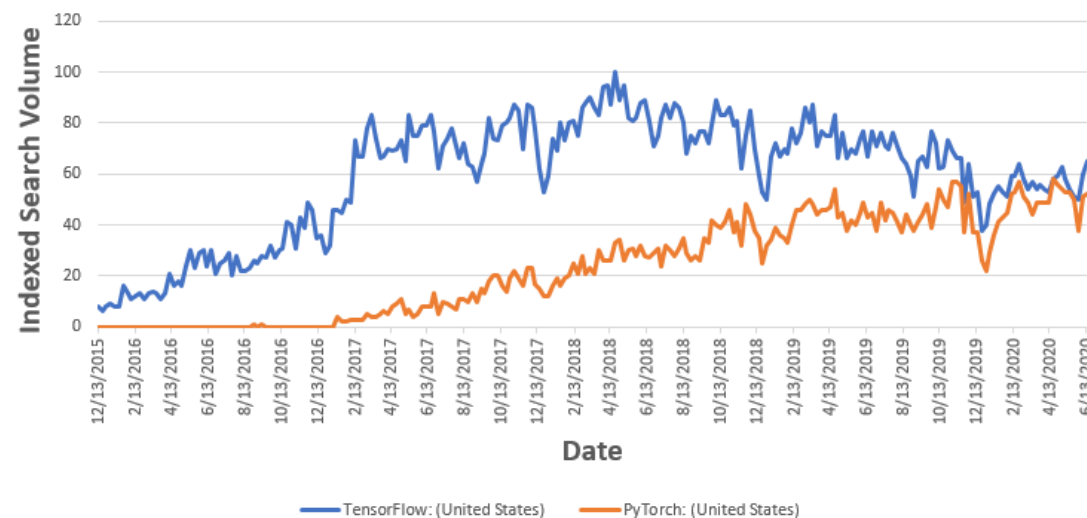
@karpathy

Following

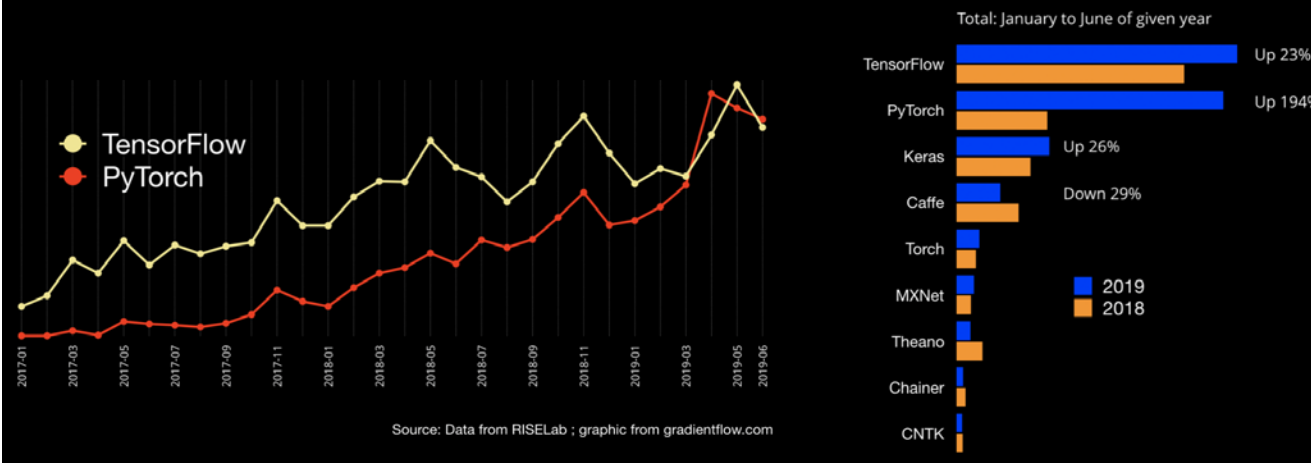
I've been using PyTorch a few months now and I've never felt better. I have more energy. My skin is clearer. My eye sight has improved.

11:56 AM - 26 May 2017

Google Search Trends



Number of papers on arxiv.org that mention a given framework



Numpy + AutoGrad + Function

- Numpy 구조를 가지는 Tensor 객체로 array 표현
- 자동미분을 지원하여 DL 연산을 지원
- 다양한 형태의 DL을 지원하는 함수와 모델을 지원함

Tutorials > Welcome to PyTorch Tutorials

WELCOME TO PYTORCH TUTORIALS

New to PyTorch?

The 60 min blitz is the most common starting point and provides a broad view on how to use PyTorch. It covers the basics all the way to constructing deep neural networks.

[Start 60-min blitz >](#)

PyTorch Recipes

Bite-size, ready-to-deploy PyTorch code examples.

[Explore Recipes >](#)

All	Audio	Best Practice	C++	CUDA	Frontend APIs	Getting Started
Image/Video	Interpretability	Memory Format	Model Optimization			
Named Tensor	Parallel and Distributed Training	Production	Profiling			
Quantization	Reinforcement Learning	Tensorboard	Text	TorchScript		

PyTorch로 시작하는 딥 러닝 입문

[WikiDocs](#)

PyTorch로 시작하는 딥 러닝 입문



지은이 : [유원준](#)

최종 편집일시 : 2020년 12월 30일 9:51 오전

저작권 :

395 명이 추천

이 책은 딥 러닝 프레임워크 PyTorch를 사용하여 딥 러닝에 입문하는 것을 목표로 합니다.

현재 작성 중에 있습니다.

많은 피드백 부탁드립니다.

감사합니다.

End of Document
Thank You.