



Notebooks

“

*1-click Jupyter notebooks with
zero configuration required*

”

Feature Description

Jupyter notebooks are one of the most popular developer tools. Gradient makes it easy for users to access high-performance CPU or GPU-enabled computing resources capable of running ML workflows within a fully hosted and managed development environment.

Key Benefits

Easy setup

Allocating compute resources for model training can be an enormous headache. Gradient makes it easy to focus on building models, not managing your environment. Launch a fully configured notebook that includes all the frameworks, libraries, and drivers you need for machine learning and deep learning -- and install any custom dependencies that are compatible with Jupyter.

Job runner architecture

Gradient ships with a job runner to allocate compute resources automatically. Models can be taken from research to production to distributed training with 1-click deployment.

Framework, library, and driver support

Notebooks can be pre-loaded with nearly any framework, library, or driver. Custom libraries and dependencies are also supported.

How it works

01

Choose a pre-configured template or custom container

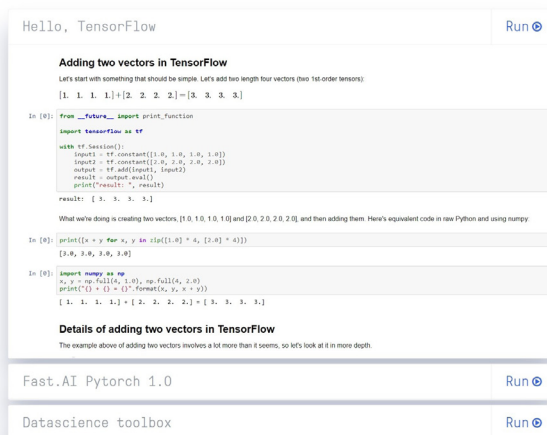
02

Select a CPU, GPU, or TPU instance -- you can toggle a new instance on demand

03

Start, clone, or stop your notebook at any time

Feature Screenshots



Hello, TensorFlow Run

Adding two vectors in TensorFlow

Let's start with something that should be simple. Let's add two length four vectors (two 1st order tensors):

```
[1, 1, 1, 1] + [2, 2, 2, 2] = [3, 3, 3, 3]
```

```
In [10]: from __future__ import print_function
import tensorflow as tf

with tf.Session():
    input1 = tf.constant([1, 0, 1, 0], dtype=tf.float32)
    input2 = tf.constant([2, 0, 2, 0], dtype=tf.float32)
    output = tf.add(input1, input2)
    result = output.eval()
    print("result: ", result)

result: [ 3.  3.  3.  3.]
```

What we're doing is creating two vectors, [1.0, 1.0, 1.0, 1.0] and [2.0, 2.0, 2.0, 2.0], and then adding them. Here's equivalent code in raw Python and using numpy:

```
In [8]: print([x + y for x, y in zip([1, 0] * 4, [2, 0] * 4)])
[3, 0, 3, 0, 3, 0, 3, 0]
```

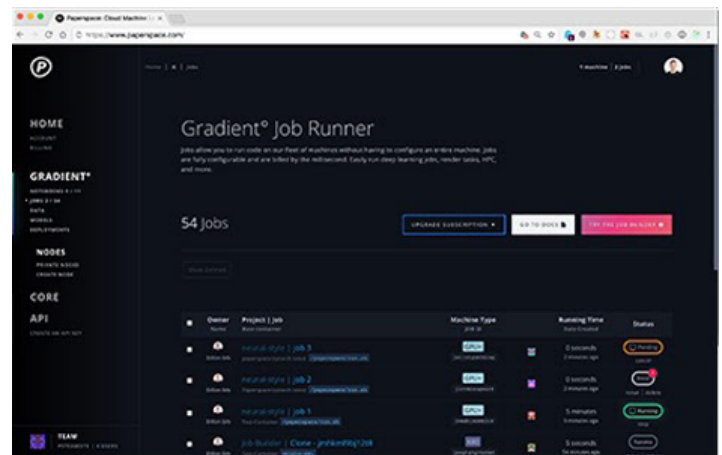
```
In [9]: import numpy as np
x, y = np.full([4, 1], 1), np.full([4, 1], 2)
print([i + j for i, j in zip(x, y)])
[ 3.  3.  3.  3.] + [ 2.  2.  2.  2.] = [ 3.  3.  3.  3.]
```

Details of adding two vectors in TensorFlow

The example above of adding two vectors involves a lot more than it seems, so let's look at it in more depth.

Fast.AI Pytorch 1.0 Run

Datascience toolbox Run



Jupyter notebooks form the basis of the Gradient enterprise machine learning pipeline and come pre-configured to support any framework, library, or driver needed for machine learning or deep learning.

The Gradient job runner makes machine learning as effortless as modern web development by providing 1-click deployment of models to compute resources. The job runner architecture supports deployments on any hardware type and any framework.

More Information

- <https://gradient.paperspace.com/notebooks>