

8.6 使用GPU



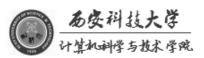
■ 导入TensorFlow, 查看版本

```
In [1]: import tensorflow as tf
print(tf. __version__)
2.0.0
```

查看当前主机上的运算设备

```
In [2]: gpus = tf.config.experimental.list_physical_devices(device_type='GPU')
    cpus = tf.config.experimental.list_physical_devices(device_type='CPU')
    print(gpus)
    print(cpus)

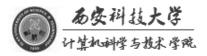
[PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
    [PhysicalDevice(name='/physical_device:CPU:0', device_type='CPU')]
```





■ 指定在CPU上执行

```
In [3]: # 指定在CPU上执行
        with tf. device ('/cpu:0'):
             cpu a = tf. random. normal([10000, 1000])
             cpu b = tf. random. normal([1000, 2000])
             cpu c=tf. matmul(cpu a, cpu b)
In [4]: print ("cpu a:", cpu a. device)
        print ("cpu b:", cpu b. device)
        print ("cpu c:", cpu c. device)
        cpu a: /job:localhost/replica:0/task:0/device:CPU:0
        cpu b: /job:localhost/replica:0/task:0/device:CPU:0
        cpu c: /job:localhost/replica:0/task:0/device:CPU:0
```



■ 指定在GPU上执行

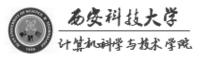
```
In [5]: #查看GPU是否可用
        tf. test. is gpu available()
Out[5]: True
In [6]: # 指定在GPU上执行随机数操作
        with tf. device ('/gpu:0'):
            gpu a = tf. random. normal([10000, 1000])
            gpu b = tf. random. normal([1000, 2000])
            gpu c = tf. matmul(gpu a, gpu b)
In [7]: print("gpu_a:", gpu_a. device)
        print ("gpu b:", gpu b. device)
        print("gpu c:", gpu c. device)
        gpu a: /job:localhost/replica:0/task:0/device:GPU:0
        gpu_b: /job:localhost/replica:0/task:0/device:GPU:0
        gpu_c: /job:localhost/replica:0/task:0/device:GPU:0
```





■ 创建函数cpu_run()和gpu()_run

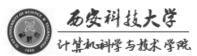
```
In [8]: # 函数cpu run()
         def cpu run():
             with tf. device ('/cpu:0'):
                 cpu a = tf. random. normal([10000, 1000])
                 cpu b = tf. random. normal([1000, 2000])
                 c = tf. matmul(cpu a, cpu b)
             return c
In [9]: # 函数gpu run()
         def gpu run():
             with tf. device ('/gpu:0'):
                 gpu a = tf. random. normal([10000, 1000])
                 gpu b = tf. random. normal([1000, 2000])
                 c = tf. matmul(gpu a, gpu b)
             return c
```





■ 比较在CPU和GPU上执行乘法操作的时间

```
In [10]: import timeit # 导入timeit模块
In [11]: # 使用timeit工具来统计执行10次的时间
         cpu_time = timeit.timeit(cpu_run, number=10)
         gpu_time = timeit.timeit(gpu_run, number=10)
         print("cpu: ", cpu_time, "gpu: ", gpu_time)
         cpu: 1.6976891000000052 gpu: 0.002923800000004917
In [12]: # 使用timeit工具来统计执行100次的时间
         cpu_time = timeit.timeit(cpu_run, number=100)
         gpu_time = timeit.timeit(gpu_run, number=100)
         print("cpu: ", cpu_time, "gpu: ", gpu_time)
         cpu: 16.43405639999999 gpu: 0.0255496999999912
```





- 在TensorFlow中,张量可以运行在CPU、GPU或TPU上
- 一般无需指定设备, TensorFlow会自动调用所有可用资源进行计算, 决定执行操作的设备, 并在需要时将操作复制到该设备

