

Google Colab Free GPU Guide

A comprehensive guide to using Google Colab's free GPU resources for machine learning and data science projects.

Quick Start

1. Visit colab.research.google.com
2. Sign in with your Google account
3. Create a new notebook
4. Enable GPU: **Runtime → Change runtime type → GPU (T4)**

Table of Contents

- [Features](#)
- [Getting Started](#)
- [Verifying GPU Access](#)
- [Working with Files](#)
- [Best Practices](#)
- [Common Commands](#)
- [Limitations](#)
- [Troubleshooting](#)

Features

- **Free GPU Access:** NVIDIA T4 GPU with 16GB VRAM
- **Pre-installed Libraries:** TensorFlow, PyTorch, NumPy, Pandas, and more
- **Google Drive Integration:** Easy file storage and persistence
- **12 Hours Runtime:** Maximum session time per instance
- **12GB RAM:** System memory available
- **No Setup Required:** Browser-based, works on any device

Getting Started

Step 1: Access Colab

Navigate to [Google Colab](https://colab.research.google.com) and sign in with your Google account.

Step 2: Create a Notebook

Click + New Notebook or File → New Notebook

Step 3: Enable GPU

1. Go to **Runtime** → **Change runtime type**
2. Select **GPU** under Hardware accelerator
3. Choose **T4 GPU** (free tier)
4. Click **Save**

🔍 Verifying GPU Access

PyTorch

```
python
import torch

print(f"CUDA Available: {torch.cuda.is_available()}")
print(f"GPU Name: {torch.cuda.get_device_name(0)} if torch.cuda.is_available() else 'No GPU'")
print(f"GPU Memory: {torch.cuda.get_device_properties(0).total_memory / 1e9:.2f} GB")
```

TensorFlow

```
python
import tensorflow as tf

print(f"GPUs Available: {len(tf.config.list_physical_devices('GPU'))}")
print(tf.config.list_physical_devices('GPU'))
```

Check GPU Utilization

```
bash
!nvidia-smi
```

📁 Working with Files

Upload Files

```
python
```

```
from google.colab import files  
uploaded = files.upload()
```

Mount Google Drive

```
python  
  
from google.colab import drive  
drive.mount('/content/drive')  
  
# Access files at /content/drive/MyDrive/
```

Download Files

```
python  
  
from google.colab import files  
files.download('filename.txt')
```

Clone GitHub Repository

```
bash  
  
!git clone https://github.com/username/repository.git
```

Best Practices

1. **Save Regularly:** Mount Google Drive and save checkpoints frequently
2. **Monitor Memory:** Use `!nvidia-smi` to check GPU utilization
3. **Clear Cache:** Free up GPU memory when needed

```
python  
  
import torch  
torch.cuda.empty_cache()
```

4. **Use Smaller Batches:** Reduce batch size if you encounter memory errors
5. **Disconnect When Done:** Free resources via **Runtime → Disconnect**
6. **Backup Your Work:** Save important files to Google Drive before session ends

Common Commands

```
bash

# Check GPU status
!nvidia-smi

# Check disk space
!df -h

# Check RAM usage
!free -h

# Install packages
!pip install package-name

# Install from requirements
!pip install -r requirements.txt

# List installed packages
!pip list

# Check Python version
!python --version

# Download files from URL
!wget https://example.com/file.zip

# Unzip files
!unzip file.zip
```

Limitations

Free Tier Constraints

- **Session Duration:** Maximum 12 hours per session
- **Idle Timeout:** 90 minutes of inactivity disconnects session
- **Usage Caps:** Heavy users may face temporary restrictions
- **GPU Access:** Not guaranteed during peak hours
- **Storage:** Temporary; files deleted after session ends
- **No Background Tasks:** Cannot run after closing browser

Resource Specifications

Resource	Free Tier
GPU	NVIDIA T4 (16GB VRAM)
RAM	~12GB
Disk	~100GB temporary
Runtime	12 hours max

Troubleshooting

GPU Not Available

```
python
# Check runtime type
# Runtime → Change runtime type → Select GPU
```

Out of Memory Error

- Reduce batch size
- Clear GPU cache: `torch.cuda.empty_cache()`
- Restart runtime: **Runtime → Restart runtime**

Session Disconnected

- Reconnect and mount Google Drive again
- Reload necessary data and checkpoints

Slow Performance

- Check if GPU is being utilized with `!nvidia-smi`
- Ensure tensors are on GPU: `tensor.to('cuda')`
- Monitor resource usage in the Resources tab

Additional Resources

- [Official Colab Documentation](#)
- [Colab FAQ](#)

- [PyTorch Documentation](#)
- [TensorFlow Documentation](#)

Quick Example

```
python

# Simple GPU test with PyTorch
import torch
import torch.nn as nn

# Create tensors on GPU
x = torch.rand(1000, 1000).cuda()
y = torch.rand(1000, 1000).cuda()

# Perform GPU computation
z = torch.mm(x, y)
print(f"Result shape: {z.shape}")
print(f"Device: {z.device}")
print("✅ GPU computation successful!")
```

License

This guide is open source and available for educational purposes.

Contributing

Feel free to suggest improvements or report issues!

Happy Coding! 

For more advanced features and longer runtimes, consider [Colab Pro](#) or [Colab Pro+](#).