









MacBook Pro 内蔵 GPU

では、一部のライブラリや機能が「未対応」または「制限付きで動作」する場合があります。以下に**使用が難しい/不可能な主なライブラリ**やその理由をまとめます。

使用困難・未対応なライブラリー一覧（内蔵 GPU 向け）

ライブラリ / フレームワーク	状況と理由
CUDA 関連（全般）	 使用不可 ：CUDA は NVIDIA 専用で、Apple Silicon の GPU（Metal ベース）とは非互換。 例：torch.cuda, cupy, tensorflow-gpu（CUDA 版）
cuDNN / cuBLAS	 使用不可 ：上記と同様に NVIDIA GPU 専用。
TensorRT	 使用不可 ：NVIDIA GPU 向けの推論最適化ライブラリ。
JAX（GPU 利用）	 Metal 未対応 ：CPU 版は使用可能だが、Apple GPU では JAX の GPU 加速はできない。
DeepSpeed / Megatron	 GPU 加速非対応 or ビルド困難 ：GPU 要件が NVIDIA 前提。
Large-scale Hugging Face Transformers (with CUDA)	 大規模モデルの GPU 加速は不可。Metal 非対応で CPU 推論は可能だが遅い。
Stable Diffusion Web UI (AUTOMATIC1111)	 CUDA 依存。MPS パッチなどもあるが動作不安定。
cuGraph / RAPIDS	 CUDA 依存のため、Apple GPU では完全に使用不可。

使用可能な代替 or 特別対応ありのライブラリ

ライブラリ	備考
tensorflow-macos + tensorflow-metal	 Apple 公式。Metal API 経由で GPU 加速が可能。
torch + mps backend	 PyTorch で Apple Silicon GPU を使うには mps バックエンドを使用：torch.device("mps")

scikit-learn / pandas / numpy	✅ CPU ベースだが最適化されており、高速。
ONNX Runtime	✅ CPU 対応。GPU は未対応 (Apple) 。
coremltools / Core ML	✅ Apple ネイティブで最適。モデル変換して利用可能。
Metal Performance Shaders (MPS)	✅ Apple GPU 向けの低レベル API (PyTorch が一部で使用中) 。

🔍 GPU 利用可能か確認する方法 (PyTorch 例)

```
import torch
```

```
print(torch.backends.mps.is_available()) # True なら Apple GPU が利用可能
```

👉 補足 : Kaggle や Colab を併用する手もあり

Apple 内蔵 GPU で無理な処理は、以下のクラウド GPU で行くと便利です :

- 🧠 **Kaggle Notebooks** (無料 / Tesla P100)
- 💻 **Google Colab** (無料～有料 / T4, A100 など)

必要であれば、「Apple Silicon で使えるライブラリセットアップ手順」や「Metal 対応の学習コード例」も用意できますよ。欲しい？