СКЛАДЧИК

КУПЛЕНО НА

SKLADCHIK.ORG

Все самые свежие
курсы по лучшей цене!

SKL.TO

Install Visual Studio with:

- •C++ CMake tools for Windows.
- •C++ core features
- •Windows 10/11 SDK.

Visual Studio 2022 Enterprise with required components installed.

2. CUDA Toolkit:

- •Download and install CUDA Toolkit 12.2 from NVIDIA's official website.
- •Verify the installation with `nvcc --version` and `nvidia-smi`.
- •Add CUDA_PATH (C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v12.2) to your environment variables.

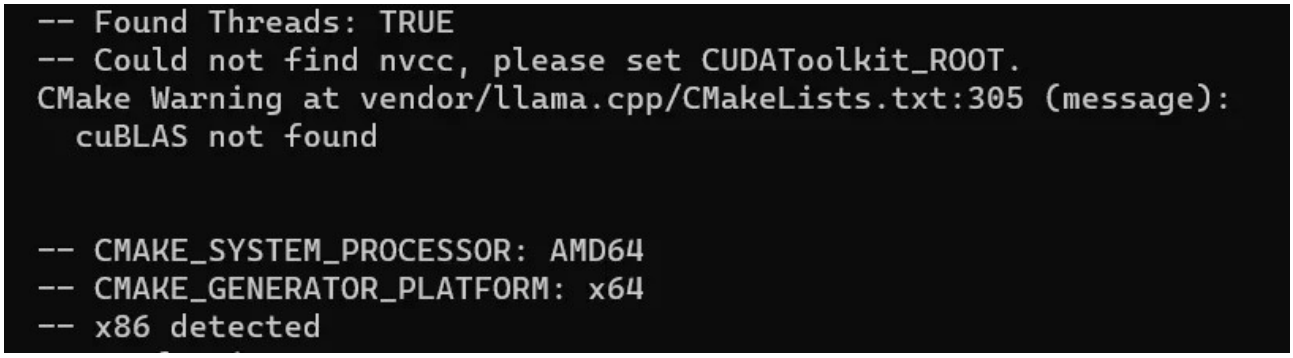
Installation Steps:

Open a new command prompt and activate your Python environment (e.g., using conda). Run the following commands:

```
set CMAKE_ARGS=-DLLAMA_CUBLAS=on
set FORCE_CMAKE=1
pip install llama-cpp-python --force-reinstall --upgrade --no-cache-dir
```

```
# Use --verbose for extra assurance that cuBLAS is being used in
compilation.
```

Add the `--verbose` option during installation if you want to ensure that CUDA is being used in compilation.



If CUDA is not configured correctly, llama-cpp-python will be installed without Hardware Acceleration.

If Cuda is detected but you get `No CUDA toolset found`error do the following:

- •Copy files from: C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v12.2\extras\visual_studio_integration\MSBuildExtensions
  to
  (For Enterprise version) C:\Program Files\Microsoft Visual Studio\2022\Enterprise\MSBuild\Microsoft\VC\v170\BuildCustomizations
  or
  (For Community version)C:\Program Files\Microsoft Visual Studio\2022\Community\MSBuild\Microsoft\VC\v170\BuildCustomizations

```
copy "C:\Program Files\NVIDIA GPU Computing
Toolkit\CUDA\v12.2\extras\visual_studio_integration\MSBuildExtensions"
"C:\Program Files\Microsoft Visual
Studio\2022\Enterprise\MSBuild\Microsoft\VC\v170\BuildCustomizations"
```

(Adjust the paths based on your installation)

Testing

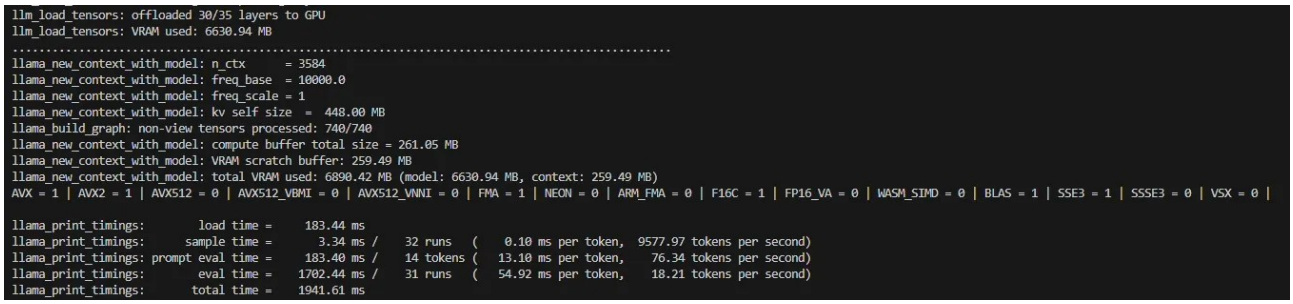
- Verify the installation by running the following Python code:

```
from llama_cpp import Llama
llm = Llama(model_path="model.gguf", n_gpu_layers=30, n_ctx=3584,
n_batch=521, verbose=True)

llm = Llama(model_path="Wizard-Vicuna-13B-Uncensored.Q4_0.gguf",
n_gpu_layers=30, n_batch=521, verbose=True)


# adjust n_gpu_layers as per your GPU and model
output = llm("Q: Name the planets in the solar system? A: ", max_tokens=32,
stop=["Q:", "\n"], echo=True)
print(output)

A chat between a curious user and an artificial intelligence assistant. The
assistant gives helpful, detailed, and polite answers to the user's
questions. USER: {prompt} ASSISTANT:
```



Using LLama2–7B-Chat with 30 layers offloaded to GPU

If the installation is correct, you'll see a `BLAS = 1` indicator in the model properties.

Conclusion:

By following these steps, you should have successfully installed llama-cpp-python with cuBLAS acceleration on your Windows machine. This guide aims to simplify the process and help you avoid the common pitfalls.

Now you’re ready to dive into local llama development with enhanced performance. Happy GPU Offloading!

СКЛАДЧИК

ЭТОТ КОНТЕНТ КУПЛЕН НА САЙТЕ SKLADCHIK.ORG

ЧТО ТАКОЕ КЛУБ «СКЛАДЧИК»?

Платформа

Это платформа, где каждый день тысячи людей собираются вместе, чтобы общими усилиями находить, приобретать и изучать курсы интересные именно им.

Сообщество

Это сообщество из 500 000 людей, открывших для себя выгодный способ быть в тренде самых актуальных и получать ценные знания по минимальной цене.

Библиотека

Это крупнейшая библиотека инфопродуктов в которой можно найти практически любой курс или тренинг продававшийся за последние 10 лет, а также уникальные авторские инфопродукты, которые не получиться найти больше нигде.

SKL.TO

ТЫ ЕЩЕ НЕ В КЛУБЕ?

ПРИСОЕДИНЯЙСЯ И НАЧИНАЙ ЭКОНОМИТЬ!