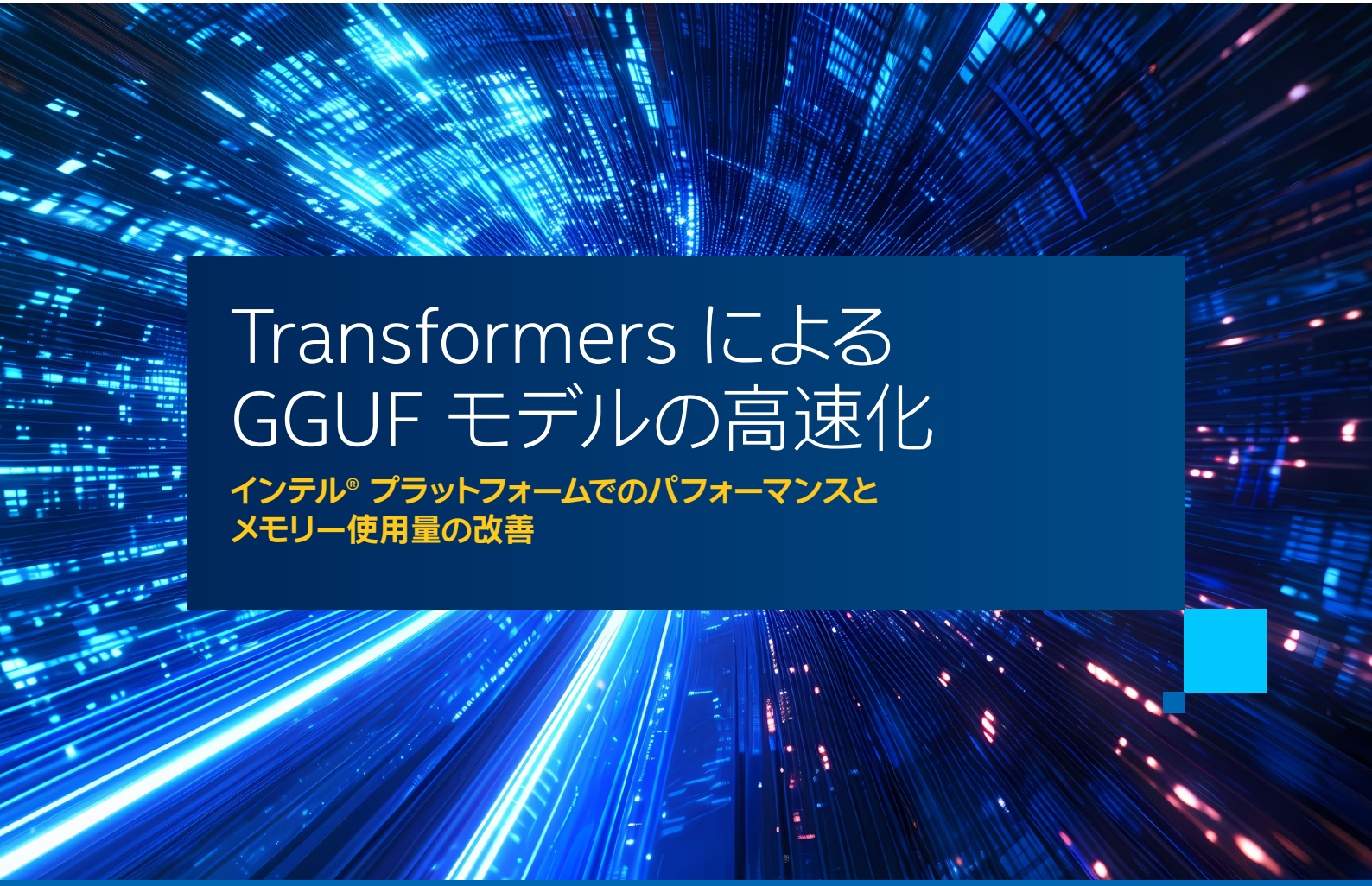

Transformers による GGUF モデルの高速化

インテル® プラットフォームでのパフォーマンスとメモリー使用量の改善

Bo Dong、Jun Lin、Zhentao Yu、Zhenzhong Xu、Yu Luo、Hanwen Chang、Haihao Shen
インテル コーポレーション

GGUF（GPT-Generated Unified Format、GPT によって生成された統一形式）は、ファイル内のテンソルとメタデータを素早く検査できる新しいバイナリー形式です（**図 1**）。これは言語モデルファイル形式の大きな飛躍であり、GPT などの大規模言語モデル（LLM）の保存と処理の効率を最適化します。PyTorch* モデルを GGUF 形式に変換するのは簡単です。

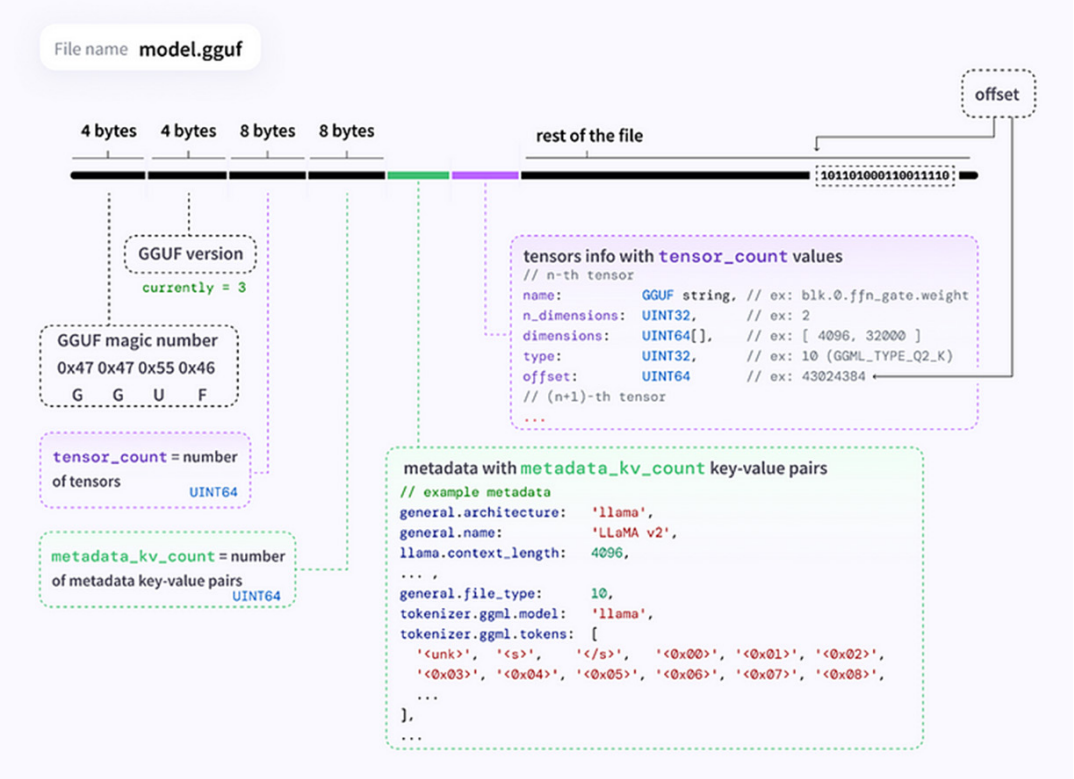

図 1. GGUF 形式（出典（英語））

Hugging Face* Transformers は最近、Transformers PR (英語)で GGUF をサポートしました。Transformers は、PyTorch* を使用して推論を実行する前に、GGUF モデルを FP32 に逆量子化します。使い方は簡単で、**図 2** に示すように、`from_pretrained` で `gguf_file` パラメーターを指定するだけです。

```
from transformers import AutoTokenizer, TextStreamer, AutoModelForCausalLM

model_name = "TheBloke/TinyLlama-1.1B-Chat-v1.0-GGUF"
gguf_file = "tinyllama-1.1b-chat-v1.0.Q4_0.gguf"
model = AutoModelForCausalLM.from_pretrained(model_name, gguf_file = gguf_file)
```

図 2. Hugging Face* Transformers での GGUF モデルの使用

Transformers 向けインテル® エクステンション（英語）は低ビットの LLM 推論を高速化します。Hugging Face* Transformers を拡張し、インテル® プラットフォーム上でパフォーマンスを向上します。幅広いモデルで GGUF 推論をサポートしており、使い方も簡単です（**図 3**）。

```
from intel_extension_for_transformers.transformers import AutoModelForCausalLM
from transformers import AutoTokenizer, TextStreamer


model_name = "TheBloke/TinyLlama-1.1B-Chat-v1.0-GGUF"
gguf_file = "tinyllama-1.1b-chat-v1.0.Q4_0.gguf"
model = AutoModelForCausalLM.from_pretrained(model_name, gguf_file = gguf_file)
```

図 3. Transformers 向けインテル® エクステンション**（英語）での GGUF モデルの使用**

現在、Transformers は Llama* や Mistral* など、50 を超える一般的な LLM をサポートしています（**表 1**）。

	LLM	中国語モデル	コーディング・モデル	音声モデル
モデル	Meta-Llama-3-8B-Instruct, TinyLlama-1.1B, LLaMA2-tB, LLaMA2-13B, LLaMA2-70B, LLaMA-7B, LLaMA-13B, , Solar-10.7B, Mistral-7B, Mistral-7B-Instruct-v0.2, Mixtral-8x7B, , GPT-J-6B, GPT-NeoX-20B, Dolly-v2-3B, MPT-7B, MPT-30B, Falcon-7B, Falcon-40B, BLOOM-7B, OPT-125m, OPT-1.3B, OPT-13B, , phi-2, phi-1_5 phi-1, phi-3-128k, phi-3-48k, StableLM-2-1_6B, StableLM-3B, StableLM-2-12B, gemma-2b-it , gemma-7b, Neural-Chat-7B-v3-1, Neural-Chat-7B-v3-2,	Qwen-7B, Qwen-14B, Qwen1.5-7B, Qwen1.5-0.5B, ChatGLM-6B, ChatGLM2-6B, ChatGLM3-6B, Baichuan-13B-Chat,Baichuan2-13B-Chat,Baichuan2-7B-Chat	Magicoder-6.7B, StarCoder-1B, StarCoder-3B, StarCoder-15.5B, CodeLlama-7b	Whisper-tiny, Whisper-base Whisper-small Whisper-medium Whisper-large

表 1. サポートされる LLM

セットアップは簡単です。最初に、Neural Speed をインストールします。

```
git clone https://github.com/intel/neural-speed.git
cd neural_speed
python setup.py install
```

次に、Transformers 向けインテル® エクステンション（英語）をインストールします。

```
git clone https://github.com/intel/intel-extension-for-transformers.git
cd intel_extension_for_transformers
python setup.py install
```

最後に、Transformers をインストールします。

```
pip install transformers
```

図 4 のコードを使用して簡単にテキストを生成できます。

```python
from transformers import AutoTokenizer, TextStreamer
from intel_extension_for_transformers.transformers import AutoModelForCausalLM

# Specify the GGUF repo on the Hugginface
model_name = "TheBloke/TinyLlama-1.1B-Chat-v1.0-GGUF"
# Download the the specific gguf model file from the above repo
gguf_file = "tinyllama-1.1b-chat-v1.0.Q4_0.gguf"
# make sure you are granted to access this model on the Huggingface.
tokenizer_name = "TinyLlama/TinyLlama-1.1B-Chat-v1.0"
prompt = "Once upon a time, there existed a little girl,"
tokenizer = AutoTokenizer.from_pretrained(tokenizer_name, trust_remote_code=True)
inputs = tokenizer(prompt, return_tensors="pt").input_ids
streamer = TextStreamer(tokenizer)
model = AutoModelForCausalLM.from_pretrained(model_name, gguf_file = gguf_file)
outputs = model.generate(inputs, streamer=streamer, max_new_tokens=300)
```

図 4. テキストを生成するサンプルコード

インテル® Xeon® Platinum プロセッサー / インテル® Core™ Ultra 5 プロセッサー 125H ベースのシステム（最大 4400MHz、最小 400MHz、合計メモリー 32GB（8 x 4GB LPDDR5、[7467 MT/s]）、Ubuntu* 13.2.0–9ubuntu1）で Transformers のパフォーマンス比較を実施しました。メモリーが小さいため、比較は TinyLlama-1.1B-Chat（英語）に限定されました。Transformers 向けインテル® エクステンションは、Transformers と比較して優れたレイテンシーとメモリー使用量を提供しました（**表 2**）。

TinyLlama in=32, out=32	最初のトークンのレイテンシー	次のトークンのレイテンシー	メモリー使用量	Lambada 精度
Transformers 向けインテル® エクステンション	275ms	12.15ms	641MB	59.07
Hugging Face* Transformers	302ms	88.7ms	4416MB	58.9

表 2. パフォーマンス、メモリー使用量、精度の比較

Transformers 向けインテル® エクステンションは、わずかなコード変更で、メモリー制約のあるシステムでトークンのレイテンシーを最小限に抑える優れた方法です。カーネルは 1 ~ 8 ビットをサポートします。StreamingLLM や Tensor Parallelism などのより高度な機能もあるため、Transformers 向けインテル® エクステンション（英語）を確認することを推奨します。