インテル® Gaudi® 2 AI アクセラレーター上での Prediction Guard のプライバシー保護 LLM プラットフォームのスケーリング

この記事は、インテルのサイトで公開されている「Scaling Prediction Guard's Privacy-Conserving LLM Platform on an Intel® Gaudi® 2 AI Accelerator」の日本語参考訳です。原文は更新される可能性があります。原文と翻訳文の内容が異なる場合は原文を優先してください。

はじめに

プライバシーを優先する AI プラットフォームの需要が高まっており、インテル® Gaudi® 2 AI アクセラレーター上で動作する Prediction Guard[1] の大規模言語モデル (LLM) プラットフォームは、このニーズを満たす先駆けとなっています。LLM は、強力な情報抽出、チャット、コンテンツ生成、自動化アプリケーションを可能にすることで、業界を変革しています。しかし、精度とスケーラビリティーを犠牲にすることなく、データのプライバシーとセキュリティーを確保することが課題となり、LLM の企業への普及は依然として遅れています。Prediction Guard は、Meta* Llama* 3、Neural-Chat-7B、DeepSeek* など最先端のオープンソース LLM をホスティングすることで、この両方のニーズに対応する LLM プラットフォームのパイオニアです。Prediction Guard のプラットフォームは、プライバシーを保護しながら、出力検証 (事実の一貫性など) と入力フィルター (PII[2] やプロンプト・インジェクションなど) をシームレスに統合します。

大規模で強力なプラットフォームを実現するため、Prediction Guard は、インテル® Tiber™ デベロッパー・クラウド内のインテル® Gaudi® 2 AI アクセラレーター上で LLM 推論の展開を最適化しました。この記事では、LLM アプリケーション向けにインテル® Gaudi® 2 AI アクセラレーターの圧倒的なパフォーマンスを解き放つため Prediction Guard チームが行った作業について説明します。

> **訳者注:**
> 現在、インテル® Tiber™ デベロッパー・クラウドでは、Enterprise ELA プランの利用者向けにインテル® Gaudi® 2 AI アクセラレーターへのアクセスを承認制で提供しています。
>
> インテル® Tiber™ デベロッパー・クラウドの詳細については、以下のサイトを参照してください。
>
> - サービス概要
>   https://www.xlsoft.com/jp/products/intel/devcloud/index.html
> - Enterprise ELA プランの特長
>   https://www.xlsoft.com/jp/products/intel/devcloud/services.html

---

[1] Prediction Guard の API プラットフォームは、幻覚、有害な出力、プロンプト・インジェクションなどのセキュリティーと信頼性の問題を軽減しながら、企業が大規模な言語モデルの可能性を最大限に活用できるようにします。

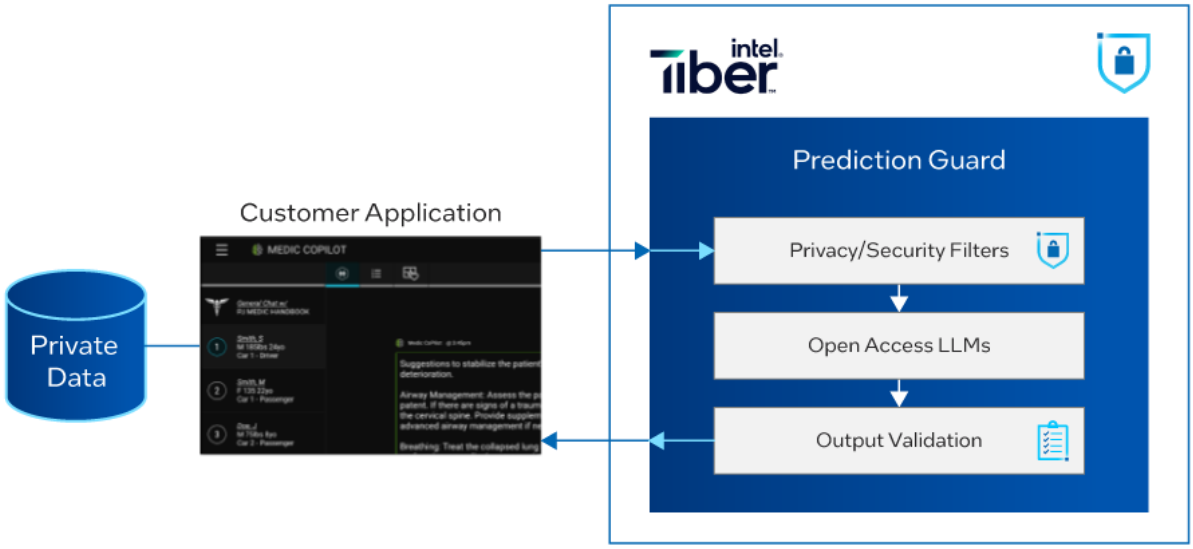
[2] PII (個人識別用情報: Personally Identifiable Information)

背景

企業は、LLM を統合することの価値を明確に理解していますが、LLM が有用な出力を生成できるようにするには、多くの場合、企業のプライベートな機密データを統合し、企業に関連する分野で LLM 出力の「基盤」を構築して改善する必要があります。例えば、既存のサポートチケットと顧客からのサポートメッセージを LLM アプリケーションに統合することで、LLM によって生成される新しい顧客応答を改善できます。また、ヘルスケア企業では、ケア・ガイドラインや顧客の病歴に関する情報を統合して、意思決定を支援するアプリケーションを改善できます。

プライベートなデータを統合するため、企業は使用する LLM が安全でプライベートな方法でホスティングされることを保証する必要がありますが、低レイテンシー、高スループット、費用対効果に優れた方法で LLM をホスティングすることは容易ではありません。ローカル LLM システムを使用すると、プライバシーは確保できますが、レイテンシーとスループットが犠牲になります。代わりに、クラウドベースのモデル・ホスティング・プラットフォームを使用すると、スケーリングは保証されますが、コストが非常に高くなります。

ホスティングの課題は、企業による AI の採用を遅延させる可能性があります。モデル・ホスティングの課題が解決しても、モデルの出力精度に関する新たな懸念や、プライバシーやセキュリティーの新たな侵害方法に直面する可能性があります。LLM は不正確なテキストを生成することがあるため (ハルシネーションまたは幻覚と呼ばれる現象)、何らかの方法でモデル出力を検証する必要があります。さらに、LLM への入力には、アプリケーション出力に漏洩する可能性のある個人情報や、悪意のある命令 (プロンプト・インジェクションと呼ばれる) が含まれる可能性があります。



Prediction Guard プラットフォーム

Prediction Guard プラットフォームは、クラス最高のフィルターを統合して、PII を検出して編集し、悪意のあるプロンプトや有害な生成をブロックし、出力を信頼できるデータソースと照合します。費用対効果に優れたスケーラブルなインテル® Gaudi® 2 AI アクセラレーターによるホスティングと組み合わせることで、金融、医療、法務などの業界向けにプライバシーが保護された LLM アプリケーションを実現します。

Prediction Guard はどのようにしてインテル® Gaudi® 2 AI アクセラレーターで成功を収めたのか?

Prediction Guard は、インテル® Tiber™ デベロッパー・クラウドのインテル® Gaudi® 2 AI アクセラレーター・インスタンスに LLM モデルをデプロイしたパイオニアであり、この構成を使用して有償顧客をサポートした最初の企業です。Optimum for Intel Gaudi ライブラリー、NVIDIA* Triton Inference Server*、およびインテルの Gaudi 製品チームからの情報提供により、次の最適化を行いました。

1. 推論リクエストの動的バッチ処理
2. 複数のインテル® Gaudi® 2 AI アクセラレーター上のモデルレプリカ間で推論を負荷分散
3. バッチ間で静的な形状を維持するプロンプトの最適化とパディング
4. インテル® Gaudi® AI アクセラレーターのガイダンスに基づくキー値 (KV) キャッシュおよびその他のハイパーパラメーターのチューニング

Prediction Guard システムへのリクエストは、最大の遅延で動的にバッチ処理されます。つまり、サーバーは、リクエストがシステムに届くまで、最大しきい値の時間待機します。この時間内に届いたリクエストは 1 つのバッチにまとめられ、実行されます。Prediction Guard API レプリカは、Google* Go* プログラミング言語で記述され、マルチプロンプト・リクエストを受信すると、リクエストを分散します。分散されたリクエストは、インテル® Gaudi® 2 AI アクセラレーターで動作する複数のモデルレプリカにリクエストを分散するロードバランサーを通過します。

以下は、PyTriton サービング・フレームワークを使用した動的バッチ処理の例です。

```
# インポート
# stdlib インポートとその他のインポート...

import habana_frameworks.torch.core as htcore
import habana_frameworks.torch.hpu as torch_hpu
from habana_frameworks.torch.hpu import wrap_in_hpu_graph, hpu
from optimum.habana.utils import set_seed
from optimum.habana.transformers.modeling_utils import adapt_transformers_to_gaudi

from pytriton.decorators import batch, first_value, group_by_values
from pytriton.model_config import DynamicBatcher, ModelConfig, Tensor
from pytriton.triton import Triton, TritonConfig


# モデル推論関数/クラス
def generate_completion(prompt, temperature=0.1, top_p=0.75, max_new_tokens=100) -> str:
    # モデル推論
    # ...
    return completions

@batch
@group_by_values("max_tokens", "temperature", "top_p")
@first_value("max_tokens", "temperature", "top_p")
def _infer_fn(
    prompt: np.ndarray,
    max_tokens: np.int32,
    temperature: np.float32,
    top_p: np.float32
):
```

```
    # 入力処理
    # ...
    return {"completion": np.array([[np.char.encode(c, "utf-8")] for c in completions])}


# PyTriton サービングと動的バッチ化
def main():
    """Initialize server with model."""
    args = _parse_args()

    log_level = logging.DEBUG if args.verbose else logging.INFO
    logging.basicConfig(level=log_level, format="%(asctime)s - %(levelname)s
- %(name)s: %(message)s")

    log_verbose = 1 if args.verbose else 0
    config = TritonConfig(exit_on_error=True, log_verbose=log_verbose)

    with Triton(config=config) as triton:
        LOGGER.info("Loading the model and starting the inference server...")
        triton.bind(
            model_name=model_name.split("/")[-1],
            infer_func=_infer_fn,
            inputs=[
                Tensor(name="prompt", dtype=bytes, shape=(1,)),
                Tensor(name="max_tokens", dtype=np.int32, shape=(1,)),
                Tensor(name="temperature", dtype=np.float32, shape=(-1,)),
                Tensor(name="top_p", dtype=np.float32, shape=(-1,)),
            ],
            outputs=[
                Tensor(name="completion", dtype=bytes, shape=(1,)),
            ],
            config=ModelConfig(
                max_batch_size=int(os.environ["BATCH_SIZE"]),
                batcher=DynamicBatcher(
                    max_queue_delay_microseconds=100000,
                ),
            ),
            strict=True,
    )
    triton.serve()
```

インテル® Gaudi® 2 AI アクセラレーターで最適なパフォーマンスを達成するには、バッチ全体およびバッチ間で入力形状を静的に維持する必要があります。Prediction Guard は、プロンプトのバッチを分析し、プロンプトを一定の長さにパディングし、NVIDIA* Triton Inference Server* から動的にバッチ化されたリクエスト間でも静的な形状を維持できる特殊なロジックを実装しました。

インテル® Gaudi® 2 AI アクセラレーターで推論を実行する際にプロンプト入力をパディングする簡単な方法の 1 つは、プロンプトの最大長パラメーターにパディングすることです。この機能を含むテキスト生成機能については、GitHub* の Optimum for Intel Gaudi ライブラリーの例 (英語) で実証されています。

以下に例を示します。

```
# トークン化
if args.max_input_tokens > 0:
    input_tokens = tokenizer.batch_encode_plus(
        input_sentences,
        return_tensors="pt",
        padding="max_length",
        max_length=args.max_input_tokens,
        truncation=True,
    )
else:
    input_tokens = tokenizer.batch_encode_plus(
        input_sentences,
        return_tensors="pt",
        padding=True
    )
```

最後に、Prediction Guard は、インテルの Gaudi 製品チームのガイダンスに基づいて、KV キャッシュサイズ、数値精度、およびその他のハイパーパラメーターをチューニングしました。これにより、インテル® Gaudi® 2 AI アクセラレーター独自のアーキテクチャーを最大限に活用できるようになりました。

Prediction Guard のハードウェア構成

Prediction Guard モデルサーバーは、インテル® Tiber™ デベロッパー・クラウドの次のインテル® Gaudi® 2 AI アクセラレーター構成のインスタンスで動作しています。

- 8 x インテル® Gaudi® 2 AI アクセラレーター (トレーニング用)
- 2 x 第 3 世代インテル® Xeon® スケーラブル・プロセッサー
- 各インテル® Gaudi® 2 AI アクセラレーターに統合された 24x 100Gb RoCE ポートによる拡張ネットワーク容量
- 700GB/s スケール (サーバー内)、4TB/s スケールアウトを実現

インテル® Gaudi® ソフトウェア・スイートによりシステムの構築と移行を容易に行うことができます。

結果

Prediction Guard は、インテル® Gaudi® 2 プロセッサー上で LLM 推論を実行する最適化により、画期的なパフォーマンス向上を実現しました。モデルサーバーのデプロイメントを NVIDIA* A100 Tensor Core GPU からインテル® Gaudi® 2 AI アクセラレーターに移行したことで、ファイン・チューニングした Llama* 2 や Mistral* AI など、特定のモデルで最大 2 倍のスループットを達成しました。

さらに、Prediction Guard はインテル® Gaudi® 2 AI アクセラレーターを搭載したストリーミング・エンドポイントで業界トップクラスのレイテンシーを実現しました。Neural-Chat-7B モデルでは、最初のトークンまでの平均時間はわずか 174 ミリ秒に短縮されました。チャットボットなどのリアルタイム・アプリケーションにとって重要なこのメトリックは、LLMPerf ベンチマークで測定された Anyscale、Replicate*、Together AI などの業界をリードするクラウド・プロバイダーと同等かそれ以上で、Neural-Chat-7B モデルでは 200 ミリ秒から最大 3.68 秒の時間でした。

LLMPerf ベンチマーク・スイートは、出力トークン・スループット (要約などの高スループット・ユースケースの場合) と最初のトークンまでの時間 (ストリーミング・ユースケースの場合) という 2 つの主要なメトリックで LLM 推論プロバイダーを評価します。インテル® Gaudi® 2 プロセッサーを使用した Prediction Guard AI アクセラレーション・プラットフォームは、両方の面で優れたパフォーマンスを発揮しました。

Prediction Guard は、エンタープライズ・カスタマー・アプリケーションを強化するだけでなく、3 つの主要なハッカソンで 4,500 人を超えるユーザーをサポートすることで、プラットフォームのスケーラビリティーを検証しました。これには、米国のさまざまな大学から 1,000 人の学生が参加した Purdue 大学の Data 4 Good Case Competition、世界中の 2,000 人の開発者が参加したインテルの Advent of GenAI Hackathon、1,500 人の学生が参加したスタンフォード大学の TreeHacks イベントが含まれます。ピーク時の需要も、Prediction Guard のインテル® Gaudi® 2 プロセッサー上のデプロイメントは、シームレスに負荷を処理しました。

まとめ

Prediction Guard は、インテル® Gaudi® 2 AI アクセラレーター上に LLM をデプロイしたパイオニアとして、バッチ処理、負荷分散、静的シェーピング、ハイパーパラメーターに関する最適化を緻密に実装することで、言語 AI の世界に革命をもたらしました。革新と効率性への取り組みにより、Prediction Guard はこれまで成しえなかったレベルのパフォーマンスとスケーラビリティーを達成することに成功しました。データ・プライバシー対策とインテル® Gaudi® 2 プロセッサーを利用した最先端の LLM 推論機能をシームレスに統合することで、Prediction Guard は、さまざまな業界の企業で採用される、カスタマイズされた安全で高性能な言語 AI ソリューションの最先端を行くリーダーとしての地位を確立しました。

関連情報

インテル® Gaudi® 2 AI アクセラレーター

エコシステム開発ハブ (英語)

Prediction Guard (英語)

Neural-Chat-7B Model (英語)

製品および性能に関する情報

[1] 性能は、使用状況、構成、その他の要因によって異なります。詳細については、http://www.intel.com/PerformanceIndex/ (英語) を参照してください。