インテルと Facebook* の協力により PyTorch* の CPU パフォーマンスを向上

この記事は、インテル® デベロッパー・ゾーンに公開されている「Intel and Facebook* collaborate to boost PyTorch* CPU performance」の日本語参考訳です。

世界中で、テキスト、写真、ビデオなど多くの情報が日々生成されています。ここ数年のディープラーニングの進歩により、最先端の音声認識および合成、画像/ビデオ認識、個人認識を含む情報を人々が理解できるように支援するアプリケーションは大きく改善されました。

ディープラーニングを適用した新しいモデルの開発は、モデルのトレーニング段階から始まります。目標のパフォーマンスが達成されたら、推論にこれらのモデルを配備してアプリケーションで新しい予測を行います。トレーニングの完了までは、通常、数時間から数日かかります。状況によっては数週間かかることもあります。一方、推論は通常、数ミリ秒単位で完了し、多くの場合アプリケーション・ワークフロー内の大きなプロセスの 1 ステップを構成します。推論は通常、大量のデータセットに対して行われますが、推論の計算強度はトレーニングよりもはるかに低いため、推論に費やされる計算リソースはトレーニングに費やされる計算リソースよりも少なくなります。具体的な例として、Facebook* データセンターで処理する多くの分野の推論ワークロードの数は増加し続けており (英語)、毎日 200 兆以上の予測と 60 億以上の言語変換が行われています (英語)。

現在、インテルは、インテル® ディープラーニング・ブースト (インテル® DL ブースト) テクノロジーを実装した第 2 世代インテル® Xeon® スケーラブル・プロセッサー (開発コード名 Cascade Lake) をリリースしています。エンドユーザーは、コードに最小限の変更を加えるだけでこのテクノロジーを活用できます。最適化は抽象化され、PyTorch* などのディープラーニング・フレームワークに統合されます。

この記事では、第 2 世代インテル® Xeon® スケーラブル・プロセッサー (開発コード名 Cascade Lake) におけるハードウェアの進歩、インテルと Facebook* の協力により PyTorch* コミュニティーにもたらされた、最適化されたハードウェアを活用するために必要なソフトウェアの最適化、ディープラーニング・ワークロードにおけるこれらの進歩の結果を詳しく説明します。

ハードウェアの進歩

2017 年 7 月、インテルは、次の新機能を含むインテル® Xeon® スケーラブル・プロセッサー (開発コード名 Skylake) をリリースしました。

- 512 ビット幅の FMA (Fused Multiply Add) 命令 (https://www.isus.jp/technical-document/ にある「インテル® 64 および IA-32 アーキテクチャー最適化リファレンス・マニュアル参考訳」の 14.15 節を参照) を含むインテル® アドバンスト・ベクトル・エクステンション 512 (インテル® AVX-512) 命令セット
- 32 個のレジスター (従来の 2 倍)
- キャッシュメモリーも追加
- 高いメモリー帯域幅
- コアあたり 2 つの FMA 実行ユニット (一部のプロセッサーを除く)

現在、インテルは、第 1 世代インテル® Xeon® スケーラブル・プロセッサーのすべての機能に、インテル® DL ブースト・テクノロジーの一部としてインテル® AVX-512 VNNI (Vector Neural Network Instruction) 拡張命令 (図 1 を参照) を追加した、第 2 世代インテル® Xeon® スケーラブル・プロセッサー (開発コード名 Cascade Lake) をリリースしています。VNNI は 8 ビットの FMA と 32 ビットの融合操作を 1 つの命令で行うことができます。32 ビットの FMA では FMA の処理能力は 4 倍になります。

低い精度は 2 つの点でパフォーマンス向上をもたらします。

1. FMA スループットの向上により、計算依存の操作を高速化します。
2. フットプリントの減少 (32 ビットではなく 8 ビットを使用) により、メモリー階層間で高速にデータ移動を行い、メモリー帯域幅依存の操作を高速化します。

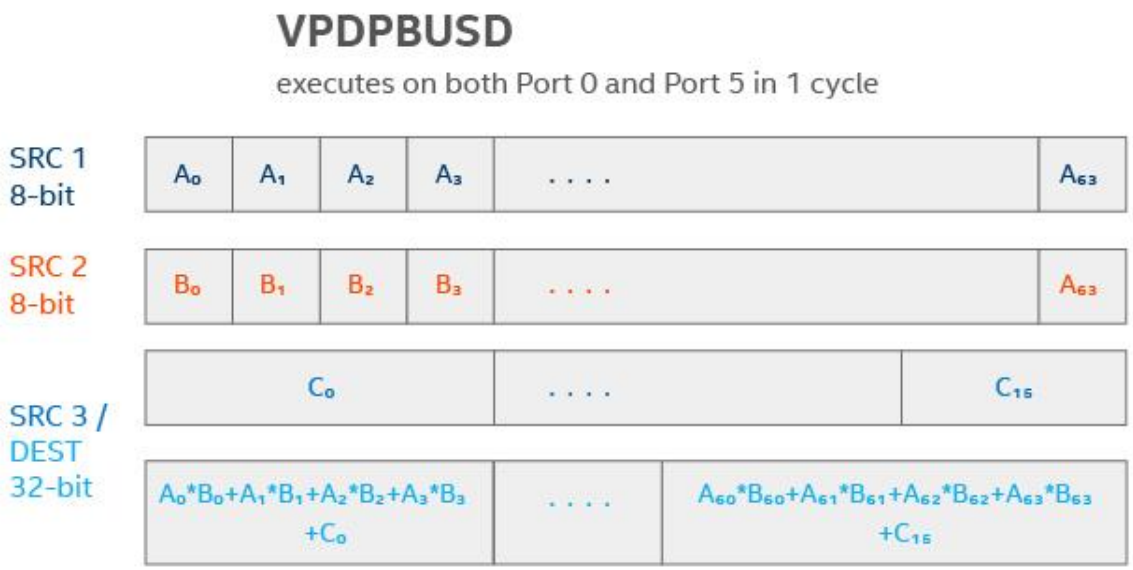

図 1. インテル® AVX-512 VNNI VPDPBUSD 命令は FMA ユニットごとにクロックサイクルあたり 64 の符号付き 8 ビット値と 64 の符号なし 8 ビット値を乗算して 16 の符号付き 32 ビット値に融合 (FMA ユニット数はコアあたり最大 2 つ)。謝辞: Israel Hirsh

ソフトウェアの進歩

インテルと Facebook* は、PyTorch* の CPU パフォーマンスを向上するため協力して作業を行っています。通常、これらの最適化を行うためにデータ・サイエンティストが PyTorch* スクリプトを変更する必要はありません。

ディープラーニング・ネットワークは、さまざまなレイヤーやノードで構成される計算グラフです。最適化はノードレベルとグラフレベルで行われました。ノードレベルでは、インテルは CPU パフォーマンス向けに畳み込み、行列乗算、ReLU、プーリングなど、さまざまなレイヤーを最適化し、ディープ・ニューラル・ネットワーク向けインテル® MKL (インテル® MKL-DNN) の最適化を行いました。これらの最適化は、データ転送を最小限に抑え、SIMD 命令、実行ユニット、レジスター、メモリーキャッシュ階層を効率良く利用することを保証します。グラフレベルでは、さまざまなデータ順序手法とレイヤー融合 (例えば、ReLU を畳み込みに融合して、データがレジスターに存在する間に最後の畳み込みサイクルで ReLU 操作を実行) によりノードのグループを最適化しました。

インテル® MKL-DNN は、インテル® MKL-DNN API を使用して最もパフォーマンス・クリティカルな DNN レイヤーを実装し、PyTorch* と Caffe2 バックエンドの両方に統合されました。PyTorch* と Caffe2 への統合に際しインテル® MKL-DNN の統合も整理され、インテル® MKL-DNN ライブラリーは PyTorch* 1.0 バイナリーに CPU のデフォルトとしてビルトインされました。インテル® MKL-DNN のテンソル表現は、PyTorch* と Caffe2 (C2) バックエンドの両方で動作するように再設計されました。また、FBGEMM とインテル® MKL-DNN の両方で Caffe2 *int8* モデルを実行できるように、*int8* 操作の FBGEMM のセマンティクスと量子化手法を調整しました。

結果

Tioga Pass は、さまざまな計算サービスをサポートするために Facebook* で使用されている Open Compute Project (OCP) プラットフォームです。デュアルソケットのマザーボードに、インテル® Xeon® Gold 6139 プロセッサー (開発コード名 Skylake) を搭載しています。表 1 は、インテル® MKL-DNN ライブラリーが統合された PyTorch* (C2 バックエンド) の使用結果をまとめたものです。ソケットあたりのバッチサイズ 1 および 32 で ResNet50 推論を実行したところ、*fp32* で *fp32* ベースラインの 5.4 倍および 8.0 倍、*int8* で *fp32* ベースラインの 9.3 倍および 15.6 倍のパフォーマンス・ゲインが得られました。

表 1. シングルソケットのインテル® Xeon® Gold 6139 プロセッサー (開発コード名 Skylake) 上でバッチサイズ 1 および 32 を使用して ResNet50 を実行した場合の、インテル® MKL-DNN が統合された PyTorch* の *fp32* および *int8* のベースライン (*fp32*、インテル® MKL-DNN なし) に対するパフォーマンス・ゲイン。

ソケットあたりの ResNet50 推論画像数/秒					
バッチサイズ	インテル® MKL-DNN なし FP32	インテル® MKL-DNN FP32	ゲイン	インテル® MKL-DNN INT8	ゲイン
1	18.90	101.36	5.4x	175.16	9.3x
32	21.18	169.49	8.0x	331.12	15.6x

現在リリースされている第 2 世代インテル® Xeon® スケーラブル・プロセッサー (開発コード名 Cascade Lake) では、さらに高いゲインが得られます。表 2 は、これらのゲインをまとめたものです。インテル® Xeon® Platinum 8280 プロセッサー (開発コード名 Cascade Lake) およびインテル® MKL-DNN ライブラリーが統

合された PyTorch* (C2 バックエンド) を使用して、ResNet50、Faster R-CNN (ResNext101-64x4d バックボーン、800x1333 解像度入力)、および RetinaNet (ResNet101 バックボーン、800x1333 解像度入力) を実行したところ、*fp32* で *fp32* ベースラインの 7.7 倍、47 倍、23.6 倍、*int8* で *fp32* ベースラインの 19.5 倍、105.1 倍、58.9 倍のパフォーマンス・ゲインが得られました。

表 2. シングルソケットのインテル® Xeon® Platinum 8280 プロセッサー (開発コード名 Cascade Lake) 上でバッチサイズ 1 を使用して ResNet50、Faster R-CNN、RetinaNet を実行した場合の、インテル® MKL-DNN が統合された PyTorch* の *fp32* および *int8* のベースライン (*fp32*、インテル® MKL-DNN なし) に対するパフォーマンス・ゲイン。

ソケットあたりの推論画像数/秒					
バッチサイズ = 1	インテル® MKL-DNN なし FP32	インテル® MKL-DNN FP32	ゲイン	インテル® MKL-DNN INT8	ゲイン
ResNet50	21.88	167.51	7.7x	427.15	19.5x
Faster R-CNN	0.02	1.08	47.0x	2.42	105.1x
RetinaNet	0.18	4.27	23.6x	10.69	58.9x

まとめ

インテルと Facebook* は協力して、PyTorch* エコシステム全体に利益をもたらす、CPU 向けの PyTorch* 1.0+ の高速化を続けています。インテル® MKL-DNN は PyTorch* (英語) にディープラーニング向けのデフォルトの数学カーネル・ライブラリーとして含まれています。ディープラーニングの推論とトレーニングの数値精度の低下に関する情報は、こちら (英語) を参照してください。

著者紹介

Andres Rodriguez 博士: インテル コーポレーションのデータセンター・グループ (DCG) のシニア主席エンジニア兼リードクラウド AI アーキテクト。クラウド顧客向けにディープラーニング・ソリューションの設計に取り組み、ディープラーニング製品についてインテルで技術的なリーダーシップをとっています。AI 分野で 15 年の経験があり、マシンラーニングの研究でカーネギーメロン大学から博士号を取得しています。マシンラーニングに関して、雑誌、会議、書籍で 20 を超える査読論文を発表しています。

Jianhui Li 博士: インテル コーポレーションのインテル® アーキテクチャー、グラフィックス & ソフトウェア・グループのシニア主席エンジニア。ディープラーニング・フレームワークの統合およびワークロードの最適化に取り組んでいます。バイナリー変換と JIT コンパイラーのソフトウェア開発者であり、AI ベースのプラットフォームに匹敵するユーザー・エクスペリエンスで Android* ARM アプリケーションを実行できる Houdini の開発にも携わりました。Fudan University でコンピューター・サイエンスの博士号を取得しています。バイナリー変換と実在アプリケーションの最適化に関する 21 の米国特許を保有しています。

Shivani Sud: システム・アーキテクト。クラウド・テクノロジーおよび ML システム・アーキテクチャーに取り組んでいます。NFV、SDN およびクラウド・テクノロジーを利用したソフトウェア定義インフラストラクチャーに対する Telco ネットワーク変換の主要な貢献者です。彼女の研究は、次世代のモバイルデバイス、マルチデバイスの利用法およびプラットフォーム・セキュリティーにも貢献しています。

システム構成

インテル® Xeon® Platinum 8280 プロセッサー:
2019 年 3 月 25 日現在のインテル社内での測定値。2x インテル® Xeon® Platinum 8280 プロセッサー、28 コア、インテル® ハイパースレッディング・テクノロジー有効、インテル® ターボ・ブースト・テクノロジー有効、合計メモリー 384GB (12 スロット/32GB/2933MHz)。BIOS: SE5C620.86B.0D.01.0271.120720180605 (ucode: 0x4000013)。Ubuntu* 18.04.1 LTS、kernel 4.15.0-45-generic。ディープラーニング・フレームワーク: PyTorch* および ONNX/Caffe2 バックエンド: https://github.com/pytorch/pytorch.git (英語) (コミット: 4ac91b2d64eeea5ca21083831db5950dc08441d6)、プル・リクエスト・リンク: https://github.com/pytorch/pytorch/pull/17464 (英語) (アップストリーミング向けにサブミット)、gcc (Ubuntu* 7.3.0-27ubuntu1~18.04) 7.3.0。インテル® MKL-DNN バージョン: v0.17.3 (コミットハッシュ: 0c3cb94999919d33e4875177fdef662bd9413dd4)、mkl 2019.1.144。ResNet50: https://github.com/intel/optimized-models/tree/master/pytorch (英語)、BS=1、合成データ、2 インスタンス/2 ソケット。データ型: INT8 および FP32

Faster R-CNN:
https://github.com/intel/Detectron/blob/master/configs/12_2017_baselines/e2e_faster_rcnn_X-101-64x4d-FPN_1x.yaml (英語)、BS=1、合成データ、2 インスタンス/2 ソケット。データ型: INT8 および FP32

RetinaNet:
https://github.com/intel/Detectron/blob/master/configs/12_2017_baselines/retinanet_R-101-FPN_1x.yaml (英語)、BS=1、合成データ、2 インスタンス/2 ソケット。データ型: INT8 および FP32

インテル® Xeon® Gold 6139 プロセッサー:
2019 年 3 月 1 日現在のインテル社内での測定値。2x インテル® Xeon® Gold 6139 プロセッサー、18 コア、インテル® ハイパースレッディング・テクノロジー有効、インテル® ターボ・ブースト・テクノロジー有効、合計メモリー 128GB (4 スロット/32GB/2.30 GHz)。BIOS: F08_3A13、CentOS* 7、kernel 3.10.0-957.e17.x86_64。ディープラーニング・フレームワーク: PyTorch* および C2 バックエンド。プル・リクエスト・リンク: https://github.com/pytorch/pytorch/pull/17464 (英語)、gcc (Red Hat* 5.3.1-6) 5.3.1 20160406、インテル® MKL-DNN バージョン: v0.17.3 (コミットハッシュ: 0c3cb94999919d33e4875177fdef662bd9413dd4)、mkl 2019.1.144

ResNet50:
https://github.com/intel/optimized-models/tree/master/pytorch (英語)、BS=1/32、データレイヤーなし、1 ソケット。データ型: INT8 および FP32

法務上の注意書きと最適化に関する注意事項

性能に関するテストに使用されるソフトウェアとワークロードは、性能がインテル® マイクロプロセッサー用に最適化されていることがあります。SYSmark* や MobileMark* などの性能テストは、特定のコンピューター・システム、コンポーネント、ソフトウェア、操作、機能に基づいて行ったものです。結果はこれらの要因によって異なります。製品の購入を検討される場合は、他の製品と組み合わせた場合の本製品の性能など、ほかの情報や性能テストも参考にして、パフォーマンスを総合的に評価することをお勧めします。詳細については、http://www.intel.com/performance (英語) を参照してください。

コンパイラーの最適化に関する詳細は、最適化に関する注意事項を参照してください。