Baidu* の PaddlePaddle における最適化されたディープ・アテンション・マッチング・モデル

この記事は、インテル® AI Blog に公開されている「Optimized NLP/Deep Attention Matching Model in Baidu's PaddlePaddle」の日本語参考訳です。

自然言語処理 (NLP) は、コンピューターが人間の言語を理解し処理することに注目した人工知能 (AI) 技術のサブセットです。中国の大手インターネットおよび AI サービス企業の Baidu* (英語) は、NLP 技術を導入した 100 を超えるアプリケーションをサポートしており、一部のモジュールは 1 日に 1,000 億回以上呼び出されます。Baidu の NLP 技術の導入例として、オンラインのカスタマー・サポート・チャットボットが挙げられます。Baidu のチャットボットは、アテンション・メカニズムをベースとするディープ・アテンション・マッチング (DAM) (英語) ネットワーク・モデルを採用しています。チャットボットの重要なタスクの 1 つは、会話の内容に応じて候補のセットから最適な応答を選択することです。

PaddlePaddle (英語) (並列分散ディープラーニング) は、Baidu によって開発されたディープラーニング・フレームワークで、同社のオンラインおよびオフラインのサービスと製品で広く使用されています。チャットボットを PaddlePaddle フレームワークに統合するため、Baidu とインテルのエンジニアは協力して、インテル® アーキテクチャー上で DAM モデルのパフォーマンスの最適化に取り組みました。インテルによるソフトウェアの最適化により、表 1 に示すようにインテル® Xeon® Gold 6148 プロセッサー・ベースのシステムで PaddlePaddle (英語) のパフォーマンスが向上しました。

レイテンシー (サンプルごと) (ms)	PP レイテンシー (サンプルごと) - ベースライン	PP レイテンシー (サンプルごと) - 最適化	ゲイン
バッチサイズ = 1	174.22	62.35	2.79 倍
バッチサイズ = 300	169.85	56.46	3.01 倍

表 1: DAM モデルの推論レイテンシー (サンプルごと)。システム構成: インテル® Xeon® Gold 6148 プロセッサー @ 2.40GHz。環境設定: OMP_NUM_THREADS=1。ベースライン・ベンチマークは、2018 年 11 月 8 日現在のインテル社内の測定値。最適化ベンチマークは、2018 年 12 月 12 日現在のインテル社内の測定値。システム構成の詳細は、「法務上の注意書き」を参照してください。

モデルのプロファイルと解析

インテル® アーキテクチャー・ベースの最適化されたインテリジェント・サービスをサポートするため、インテルは Baidu* と協力 (英語) して作業に取り組みました。このケースでは、最も時間を費やしている操作 (「ホットスポット」) の解析することで DAM モデルの最適化に取り掛かりました。図 1 に示すように、*layer_norm*、*softmax*、*stack*、および *conv3d* がホットスポットです。これらは、モデルのすべての操作の約 80% を占めており、最初に最適化すべきです。

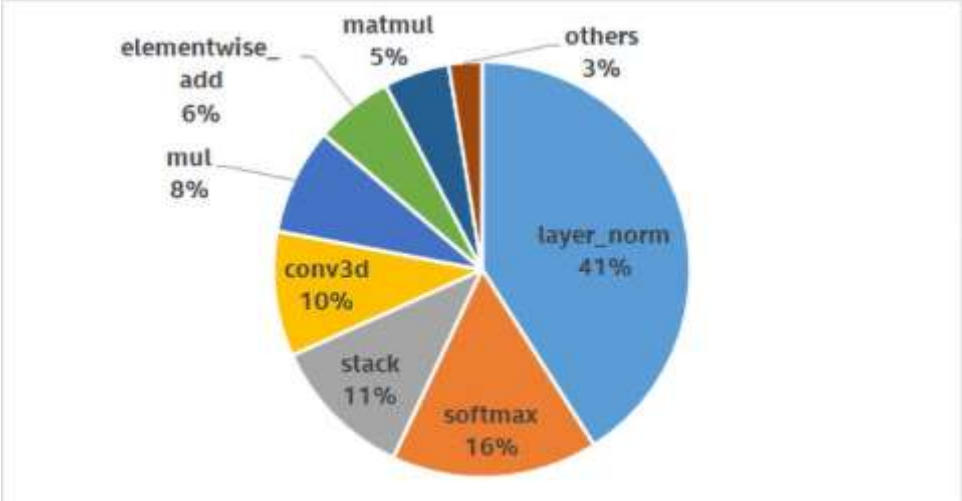

図 1: 最も時間を費やしている操作を示す初期の DAM ホットスポット・プロファイル解析結果

Baidu の DAM ネットワーク・モデル全体の構造に従って、これらの処理を解析しました。

- **表現:** 表現は、単語や文章と意味的な依存関係を取得する反復可能なアテンティブ・モジュール (図 2) で構成されます。*layer_norm op* は、この反復可能なモジュールで使用され (どこ)、計算式が複雑な (なぜ) 勾配の消滅や急増を防ぎます (なに)。

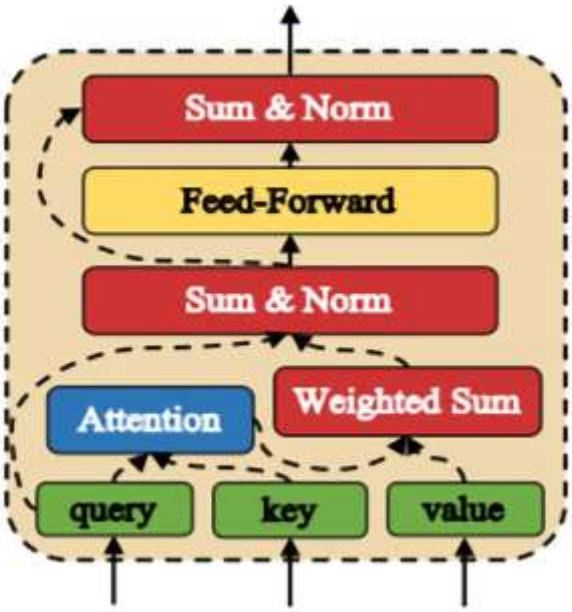

図 2: アテンティブ・モジュール[1]

- **マッチング:** 3D 畳み込みの入力として発話と応答を格納したセグメント-セグメント類似性行列を使用して、発話と応答がマッチングされます (なに)。*stack op* は、このモジュールで使用される (どこ) メモリーレベルの操作です (なぜ)。

- **集約:** 図 3 に示すように、最終的に DAM は各発話と応答のすべてのセグメント一致度を 高次元 (なぜ) の 3D マッチングイメージ Q (なに) に集約します。2 つの層 *conv3d* と *pool3d* (どこ) はこのネットワークの最後で使用されます。

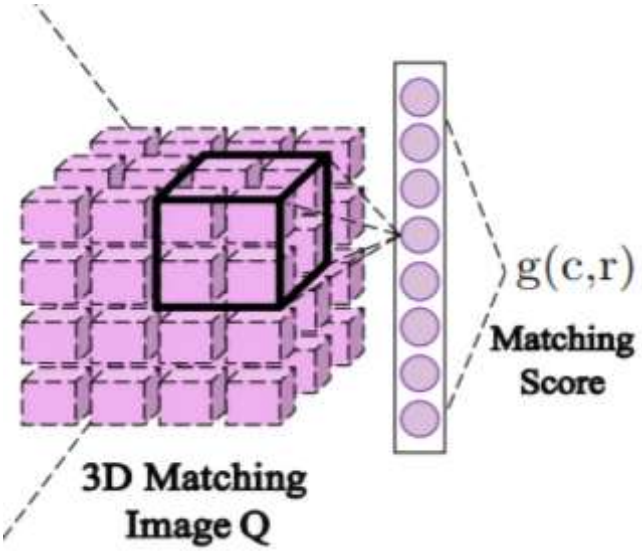

図 3: 集約。3D 畳み込みの入力となる 3D マッチングイメージ。[1]。

ワークロードの高速化による操作の最適化

インテル® マス・カーネル・ライブラリー (インテル® MKL)、ディープ・ニューラル・ネットワーク向けインテル® マス・カーネル・ライブラリー (インテル® MKL-DNN) (英語)、およびインテル® アドバンスト・ベクトル・エクステンション (インテル® AVX) は、マシンラーニングにおけるワークロードの高速化に役立ちます。表 3 と図 4 に示すように、適切な最適化を選択することで、操作レベルで最大のパフォーマンス・ゲインが得られます。

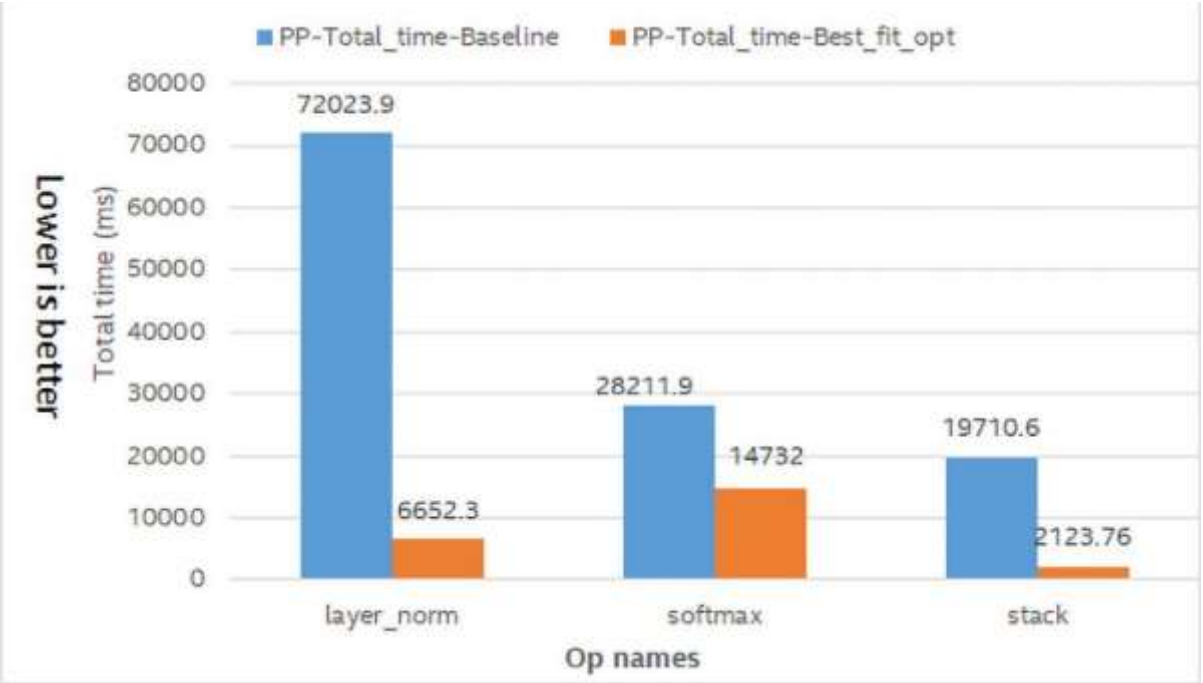

図 4: ベースラインと最適化の合計操作時間の比較。システム構成: インテル® Xeon® Gold 6148 プロセッサー @ 2.40GHz。環境設定: OMP_NUM_THREADS=1。ベースライン・ベンチマークは、2018 年 11 月 8 日

現在のインテル社内の測定値。最適化ベンチマークは、2018 年 12 月 12 日現在のインテル社内の測定値。システム構成の詳細は、「法務上の注意書き」を参照してください。

操作を最適化したところ、DAM モデルのレイテンシー (サンプルごと) はほぼ半分に減りました。表 4 は、各操作レベルの最適化によるモデルのパフォーマンス・ゲインを示しています。

バッチサイズ	ベースライン (ms)	最適な最適化 (ms)	ゲイン
1	174.22	73.49	2.37 倍
300	169.85	67.83	2.50 倍

表 4:「最適な」操作レベルの最適化によるモデルのパフォーマンス・ゲイン。システム構成: インテル® Xeon® Gold 6148 プロセッサー @ 2.40GHz。環境設定: OMP_NUM_THREADS=1。ベースライン・ベンチマークは、2018 年 11 月 8 日現在のインテル社内の測定値。最適化ベンチマークは、2018 年 12 月 12 日現在のインテル社内の測定値。システム構成の詳細は、「法務上の注意書き」を参照してください。

ライブラリーを使用した最適化 (PR#14437 (英語)): softmax

softmax 操作の実装をプロファイルしたところ、softmax の実行時間の 50% 以上が「exp」で、30% が「$1/\Sigma$」で費やされていることが分かりました。そのため、「e^x」とそれに続く合計と要素単位の除算の最適化を目標としました。インテル® MKL は、これらを最適化する BLAS とスパース BLAS ルーチンを実装 (英語) します。

複雑な計算操作の最適化 (PR#14417 (英語)): レイヤーの正規化

式 1 は、レイヤーの正規化の式です。インテル® MKL とインテル® MKL-DNN には、これらの計算を直接最適化できる数学関数はありません。最近のコンパイラーは、最適化されたアセンブリー・コードを生成しますが、インテル® AVX により多くのディープラーニング・プリミティブが向上します。また、ベクトル命令を直接使用することで、図 4 に示すように、層の正規化のパフォーマンスが 7 倍向上します。

$$\mu^l = \frac{1}{H}\sum_{i=1}^{H} a_i^l \qquad \sigma^l = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(a_i^l - \mu^l)^2}$$

(式 1)

メモリー依存操作の最適化 (PR#14488 (英語)): スタック

スタック操作は、1 つの軸にすべての入力を格納します。これはメモリーコピー操作ですが、これらのメモリー依存操作を最適化するには、次の 2 つの方法でメモリーの書き込みと読み取り操作の数を減らします: 1) 作成したメモリーを最大限に利用する、2) 最適化されたメモリー関数を使用する。このケースでは、「memcpy」関数を使用してスタックの実装をリファクタリングすることで、図 4 に示すパフォーマンス・ゲインが得られます。

インテル® MKL-DNN を使用して 3D 畳み込みをさらに最適化

インテル® MKL-DNN の 3D 畳み込みを使用して畳み込み操作を向上

プロファイル結果から、conv3d はモデルの実行時間の 10% を占めていることが分かります。インテル® MKL-DNN は、ディープラーニングの畳み込みを高速化するオープンソースのパフォーマンス拡張ライブラリー (英語) です。そのため、インテル® MKL-DNN を使用して conv3d のパフォーマンスを向上します。インテル® MKL-DNN により、表 5 に示すように、インテル® Xeon® プロセッサー E5-2650 v4 ベースのプラットフォームで 3D 畳み込みのパフォーマンスが約 4 倍向上します。

バッチサイズ	PP-インテル® MKL 合計時間 ベースライン (ms)	PP-インテル® MKL-DNN 合計時間 最適化 (ms)	ゲイン
conv3d	17140.9	4334.55	3.95 倍

表 5: 最適化の前と後の conv3d 操作の合計時間 DAM モデル、バッチサイズ = 1。システム構成: インテル® Xeon® Gold 6148 プロセッサー @ 2.40GHz。環境設定: OMP_NUM_THREADS=1。ベースライン・ベンチマークは、2018 年 11 月 8 日現在のインテル社内の測定値。最適化ベンチマークは、2018 年 12 月 12 日現在のインテル社内の測定値。システム構成の詳細は、「法務上の注意書き」を参照してください。

バッチサイズ	インテル® MKL を使用する DAM conv3d のレイテンシー (サンプルごと) (ms)	インテル® MKL-DNN を使用する DAM conv3d のレイテンシー (サンプルごと) (ms)	ゲイン
1	73.49	62.35	15.16%
300	67.83	56.46	16.76%

表 6: conv3d の最適化によるモデルのパフォーマンス・ゲイン。システム構成: インテル® Xeon® Gold 6148 プロセッサー @ 2.40GHz。環境設定: OMP_NUM_THREADS=1。ベースライン・ベンチマークは、2018 年 11 月 8 日現在のインテル社内の測定値。最適化ベンチマークは、2018 年 12 月 12 日現在のインテル社内の測定値。システム構成の詳細は、「法務上の注意書き」を参照してください。

融合命令によりさらにパフォーマンスを向上

PaddlePaddle では、バイアスと ELU を使用する畳み込みは、*conv3d op*、*elementwise add op*、および *elu op* の 3 つの操作で計算されます。インテル® MKL-DNN は、バイアスと ELU を使用する畳み込みをサポートしているため、この 3 つの操作を conv3d 操作に融合して、バイアスと RELU を使用する畳み込みの計算をサポートできます。これにより、フレームワークのオーバーヘッドが軽減されます。

これらすべての最適化を適用すると、次の表に示すように、モデル操作の 95% (時間比率による) が最適化された高速なツールで実行されます。

操作名	モデル内の時間比率	最適化
fc	27%	インテル® MKL GEMM
softmax	18%	インテル® MKL BLAS
layer norm	15%	Math JIT
conv3d	14%	インテル® MKL-DNN
matmul	13%	インテル® MKL バッチ GEMM
elementwise add	5%	インテル® MKL VADD
stack	3%	Memcpy

表 7:「最適な」最適化を適用した操作のリスト

まとめ

インテルは、ディープラーニングのパフォーマンスを向上するため、さまざまな フレームワークの最適化 (英語)、ツール (英語)、ソフトウェア・ライブラリー (英語) を提供してきました。1 つのライブラリーまたは 1 つのメソッドでは、必ずしも最高のパフォーマンスが得られません。さまざまな操作に対し、1 種類の最適化をすべてに適用するのではなく、最適な最適化方法を選択します。

グラフ融合の目標は、アルゴリズムの不要な計算とメモリーアクセスを最小限に抑えることです。融合により時間のかかる計算を減らすことができれば 合理的です。そうでない場合、このような融合はスキップできます。インテル AI チームからのパフォーマンスの最適化に関する最新情報は、@IntelAIResearch (英語) をフォローしてください。

法務上の注意書き

[1] Zhou, X., Lu, L., Dong, D., Liu, Y., Chen, Y., Zhao, X., Yu, D. and Wu, H. Multi-turn Response Selection for Chatbots with Deep Attention Machine Network, P18-1103, 2018.

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、www.intel.com/benchmarks (英語) を参照してください。

システム構成: インテル® Xeon® Gold 6148 プロセッサー @ 2.40GHz。システム構成: インテル® Xeon® Gold 6148 プロセッサー @ 2.40GHz。ベースライン・ベンチマークは、2018 年 11 月 8 日現在のインテル社内の測定値。最適化ベンチマークは、2018 年 12 月 12 日現在のインテル社内の測定値。

パフォーマンス結果は 2018 年 12 月 12 日時点のテスト結果に基づいたものであり、公開されている利用可能なすべてのセキュリティー・アップデートが適用されていない可能性があります。絶対的なセキュリティーを提供できる製品またはコンポーネントはありません。

テスト環境を再現するには、最初に PaddlePaddle を格納するパスを選択し、次のコマンドを使用して PaddlePaddle のソースコードを GitHub* からローカルの現在のディレクトリーにある Paddle という名前のフォルダーにクローンします: git clone

https://github.com/PaddlePaddle/Paddle.git。Paddle ディレクトリーに移動します (Docker* でコンパイルするか、ローカルコンパイルを選択できます)。詳細は、Paddlepaddle のドキュメント (英語) を参照してください。

Baidu* より提供された画像を使用: docker run -name paddle-test -v $PWD:/paddle -network=host -it hub.baidubce.com/paddlepaddle/paddle:latest-dev /bin/bash

ローカルコンパイルを選択: cmake -DWITH_TESTING=ON -WITH_FLUID_ONLY=ON -DWITH_GPU=OFF -DWITH_MKL=ON -WITH_SWIG_PY=OFF -DWITH_INFERENCE_API_TEST=ON -DON_INFER=ON ..

ベースライン・ベンチマーク: commit id: 1001f8e1dbd913a3560f067f39a19f1dde7bae19

Paddle でベースラインの DAM ベンチマークを実行するコマンド:

./paddle/fluid/inference/tests/api/test_analyzer_dam -infer_model=third_party/inference_demo/dam/model/ -infer_data=third_party/inference_demo/dam/data.txt -gtest_filter=Analyzer_dam.profile -paddle_num_threads=1 -repeat=5 -batch_size=1 -use_analysis=false -test_all_data

最適化ベンチマーク: commit id: acc6ae49b18cb55db4dd84cd09069ebe01a1b54a

Paddle で最適化した DAM ベンチマークを実行するコマンド:

./paddle/fluid/inference/tests/api/test_analyzer_dam -infer_model=third_party/inference_demo/dam/model -infer_data=third_party/inference_demo/dam/data.txt -gtest_filter=Analyzer_dam.profile_mkldnn -paddle_num_threads=1 -batch_size=1 -repeat=5 -test_all_data