インテル® MKL の DNN プリミティブ

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「Introducing DNN primitives in Intel® Math Kernel Library」の日本語参考訳です。

ディープ・ニューラル・ネットワーク (DNN) は、マシンラーニング分野の最先端のアルゴリズムです。1990 年代後半に業界での採用が進み、当初は銀行小切手の手書き文字認識などのタスクに利用されていました。ディープ・ニューラル・ネットワークは、このタスクにおいて人と同等またはそれ以上の能力を提供します。今日 DNN は、画像認識、ビデオおよび自然言語処理、自動運転などに必要な複雑な視覚理解の問題の解決に使用されています。DNN には、非常に多くの計算リソースと処理データが必要です。一例として、最新の画像認識トポロジー AlexNet のトレーニングには最新の計算システムで数日を要し、1400 万を上回る画像を使用します。この複雑さに対応するには、トレーニング時間を短縮し、産業分野のニーズに応えることができる高度に最適化されたビルディング・ブロックが必要です。

インテル® マス・カーネル・ライブラリー (インテル® MKL) 2017 で追加された DNN ドメインには、AlexNet、VGG、GoogLeNet、ResNet を含む主要画像認識トポロジーの高速化に必要な関数が含まれます。

これらの DNN トポロジーは、テンソルと呼ばれる多次元データを扱う、多くの標準のビルディング・ブロックやプリミティブに依存します。プリミティブには、畳み込み、正規化、アクティベーション、および内積に加えて、テンソルの操作に必要な関数が含まれます。インテル® アーキテクチャー上で計算を効率良く実行するには、ベクトル化により SIMD 命令を利用したり、スレッド化により複数の計算コアを利用する必要があります。最近のプロセッサーは、最大 512 ビットのベクトルデータ (16 の単精度数) を処理でき、1 サイクルごとに最大 2 つの積和演算 (FMA: Fused Multiply-Add) 命令を実行できるため、ベクトル化は非常に重要です。ベクトル化による利点を得るためには、データを連続するメモリー位置に配置する必要があります。通常、テンソルの次元は小さいため、データレイアウトの変更により大きなオーバーヘッドが生じます。そのため、プリミティブ間でデータレイアウトを変更せずに、トポロジーのすべての操作を実行するようにします。

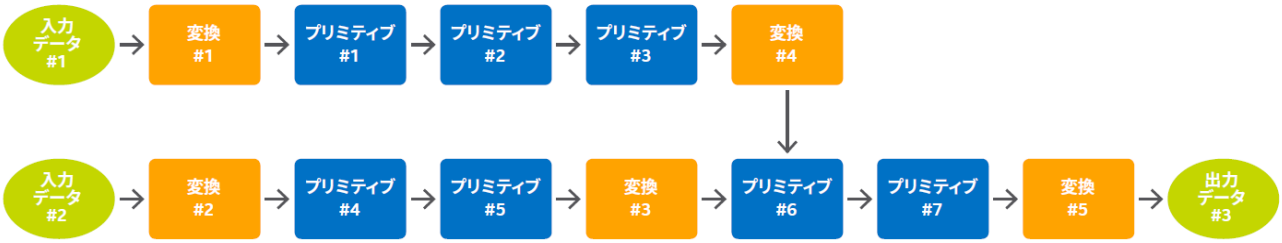
インテル® MKL は、よく使用される操作をベクトル化に対応したデータレイアウトで実装したプリミティブを提供します。

- 直接バッチ畳み込み
- 内積
- プーリング: 最大、最小、平均
- 正規化: チャネル、バッチ正規化にわたる局所反応正規化 (LRN)
- アクティベーション: 正規化線形関数 (ReLU)
- データ操作: 多次元転置 (変換)、分割、連結、合計、スケール

プログラミング・モデル

ニューラル・ネットワーク・トポロジーの実行フローは、セットアップと実行の 2 つのフェーズで構成されます。セットアップ・フェーズでは、スコアリング、トレーニング、その他のアプリケーション固有の計算を実装するのに必要なすべての DNN 操作の説明をアプリケーションが作成します。1 つの DNN 操作から次の DNN 操作へデータを渡すため、適切な出力と入力データレイアウトが一致しない場合、アプリケーションは中間データを変換し、一時配列を割り当てることがあります。このフェーズは、通常のアプリケーションでは一度だけ実行され、その後、複数の実行フェーズで実際の計算が行われます。

実行フェーズでは、データは BCWH (バッチ、チャネル、幅、高さ) などのプレーンなレイアウトでネットワークに渡され、SIMD 対応レイアウトに変換されます。層と層の間のデータの伝達では、データレイアウトが保持され、既存の実装でサポートされていない操作を実行するのに必要な場合は変換されます。



インテル® MKL の DNN プリミティブは、既存の C/C++ DNN フレームワークで利用可能な、単純な C アプリケーション・プログラミング・インターフェイス (API) を実装します。インテル® MKL の DNN 関数を呼び出すアプリケーションは、次のフェーズで構成されるべきです。

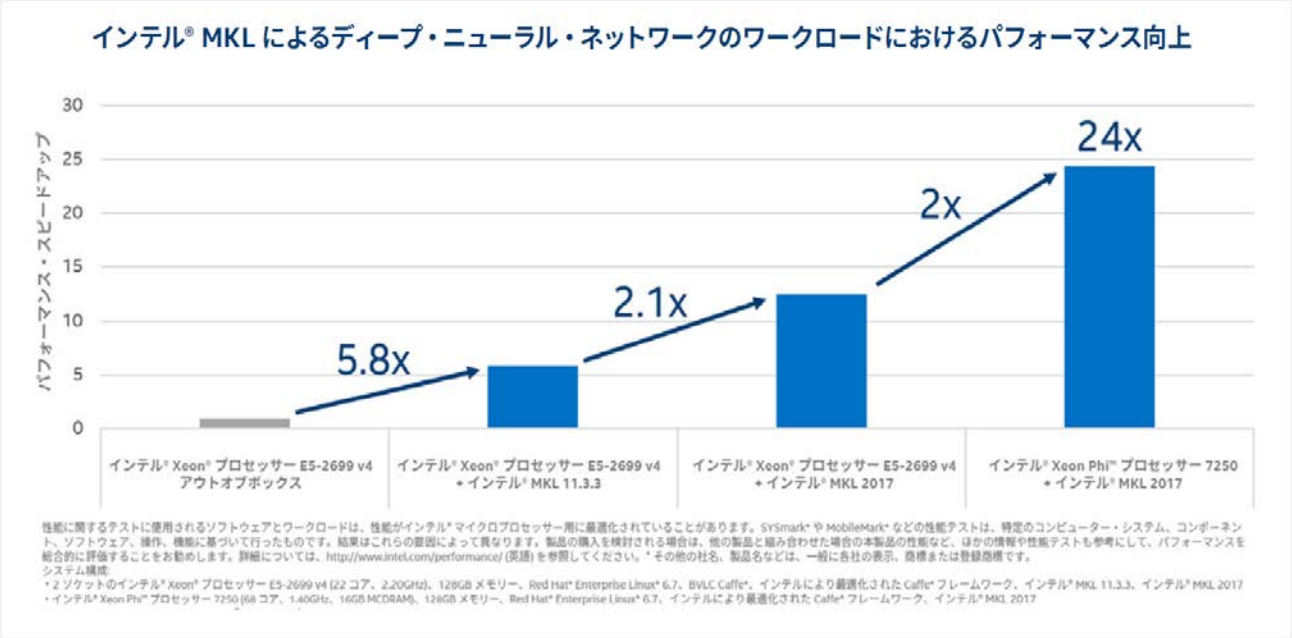
セットアップ・フェーズ: 指定された DNN トポロジーついて、アプリケーションはスコアリング、トレーニング、その他のアプリケーション固有の計算を実装するのに必要なすべての DNN 操作を作成します。1 つの DNN 操作から次の DNN 操作へデータを渡すため、適切な出力と入力データレイアウトが一致しない場合、アプリケーションは中間データを変換し、一時配列を割り当てることがあります。

実行フェーズ: アプリケーションは、入力、出力、一時配列の必要な変換を含む DNN 操作を適用する DNN プリミティブを呼び出します。

トレーニングとスコアリングのサンプルコードは、インテル® MKL パッケージのディレクトリーにあります: <mklroot>\examples\dnnc\source。

パフォーマンス

Caffe* は、Berkeley Vision and Learning Center (BVLC) 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、最もよく使用されている画像認識コニュニティー・フレームワークの 1 つです。画像認識ニューラル・ネットワーク・トポロジーの AlexNet とラベル付き画像データベースの ImageNet とともに、Caffe* はベンチマークとしてよく使用されます。次のグラフは、インテル® Xeon® プロセッサー E5-2699 v4 (開発コード名 Broadwell) とインテル® Xeon Phi™ プロセッサー 7250 (開発コード名 Knights Landing) 上で、オリジナルの Caffe* 実装とインテルにより最適化されたバージョン (最適化された行列-行列乗算と新しいインテル® MKL の DNN プリミティブを利用) のパフォーマンスを比較したものです。



まとめ

インテル® MKL 2017 で利用可能な DNN プリミティブは、インテル® アーキテクチャー上でディープラーニングのワークロードを高速化します。詳細は、インテル® MKL デベロッパー・リファレンス・マニュアルとサンプルコード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。