TIDLを活用した組込み型の低消費電力ディープ・ラーニング

Texas Instruments

Manu Mathew
Principal Engineer &
Member Group Technical Staff

Kumar Desappan
Member Group Technical Staff

Pramod Kumar Swami
Principal Engineer &
Member Group Technical Staff

Soyeb Nagori
Senior Principal Engineer &
Senior Member Technical Staff

Biju Moothedath Gopinath
Engineering Manager

Automotive Processors
Texas Instruments

概要

コンピュータ・ビジョンのアルゴリズムは、実装ごとにかなり大きな違いがあります。たとえば、線や円を検出するアルゴリズムはハフ（Hough）変換を使用することが多いのに対し、関心のある画像を検出するアルゴリズムは、方向性グラデーションのヒストグラムなど別の手法を必要とする可能性が高く、セマンティック・セグメンテーション（画像セグメンテーション、個別のピクセルがどの物体に所属するかを識別）の場合はさらに異なるアルゴリズムが必要と考えられます。

CNN（convolutional neural networks、畳み込みニューラル・ネットワーク）などのディープ・ラーニングの各種手法は、機械のインテリジェンスに革新をもたらし、アルゴリズムの精度、多様性、自律性の向上に貢献しています。ディープ・ラーニングは、車載アプリケーションの革新も実現してきました。先進運転支援システム（ADAS）向けの多くの最新アルゴリズムは、車線表示の検出、また歩行者、自動車、自転車、信号機などさまざまな物体の検出を含め、ディープ・ラーニングの手法を必要とするようになっています。ディープ・ラーニングは、このようなアルゴリズムの大半で、最善の精度を実現する重要なテクノロジーとしてその存在感を高めてきました。このホワイト・ペーパーで説明するツールを使用すると、テキサス・インスツルメンツ（TI）の車載プロセッサ上でADASアルゴリズムを実現できるようになります。

ディープ・ラーニングは、さまざまなアルゴリズムを実現するための体系的な方法を提供します。たとえば、多くのアルゴリズムではディープ・ラーニング構成は互いにかなり類似した方法で動作するので、ソフトウェア最適化とハードウェア最適化の手法を通じて処理速度を高速化するうえで、ディープ・ラーニングは非常に優れた候補と言えます。さらにこのホワイト・ペーパーでは、特にソフトウェア最適化に注目します。高度に最適化されたソフトウェア・コンポーネントは、利用可能なハードウェアの効率を最大限に高めるので、ディープ・ラーニング・アルゴリズムの動作速度を向上させることができます。アルゴリズムの最適化には、同じ最終結果をより短い時間で求めることができる高速にアルゴリズム、またはより良い結果を得ることができる改良アルゴリズムの開発が関係します。使いやすく、既存のシステム・フレームワークへの統合が容易なライブラリとコンポーネントを提供すると、改廃期間を短縮できます。

TIのJacinto™ TDA2、TDA2P、TDA3という各車載プロセッサを使用すると、カメラ、レーダ、超音波センサから受信したデータの処理と融合（フュージョン）を実現し、ADAS機能をサポートすることができます[1]。これらのセンサにより、物体の検出、分類、追跡を行うアルゴリズムを実装し、自動型の緊急ブレーキ、ドライバー監視、複数のカメラ・ストリームを組み合わせたサラウンド・ビューを開発することができます。想定される他のアルゴリズムは、車線の検出による車線逸脱防止支援や、3D（立体）構造物の検出によるパーキング・アシストです。

これらのプロセッサは、セマンティック・セグメンテーションも実行でき、この手法により、画像を形成するピクセルのうち、道路に属するピクセルと、道路以外のピクセルを分類する方法で、運転に進行可能な道路の識別を支援します。

TIディープ・ラーニング（TIDL）は、TIの組込みデバイスでディープ・ラーニングを実現する一連のコンポーネントで形成されています。TIDLは、高度に最適化された一連のディープ・ラーニング・プリミティブ（基本構成要素）で形成されていり、精度、速度、メモリ使用量に関する最善のトレードオフを実現します。また、非常に一般的に使用されているディープ・ラーニング・トレーニング・フレームワークの1つに基づくモデルを簡単に使用して、TDAベースの組込みプラットフォームでそのモデルを非常に高速に実行する方法を実現します。使いやすさと高性能は、TIDLの2つの重要な要因であるといえます。

図1に、一連のTIDLコンポーネントを示します。開発フローの最初の作業は、ネットワーク・モデルのトレーニングであり、一般的なトレーニング・フレームワーク内で実施するのが最善です。次の手順は、TIDLデバイス変換ツールを使用して、ネットワーク・モデルを内部形式に変換することです。これは、TIDLライブラリ内での使用に最善の形式です。最初のステップは、TIDLに付属のAPI（アプリケーション・プログラミング・インターフェイス）を使用して、組込み用のTDAデバイス上で、変換後のネットワーク・モデルを実行することです。

TIDLは、物体検出やセマンティック・セグメンテーションのような一部のADASアルゴリズムが必要とする、フレーム全体のCNNを実行できます。また、TIDLは画像のうち関心のある小規模な領域に対して機能させる、物体分類アルゴリズムを実行することもできます。

低消費電力デバイス向けのディープ・ラーニング

ディープ・ラーニングには、トレーニングと推論が関係します。トレーニングは通常、外部GPU（グラフィックス処理ユニット）を使用し、サーバーまたはPCに格納されている大規模なデータ・セットに対してオフラインで実行します。このフェーズでは、リアルタイム性能や消費電力を問題視しません。ただし、実際の推論を実施する、つまり、低消費電力デバイスが車線検出のようなアルゴリズムを実行するときは、リアルタイム性能と低消費電力が重要になります。一般に入手できるいくつかのディープ・ラーニング・フレームワークを使用すると、CNNまたは他のディープ・ラーニング・モデルのトレーニングを実施できます。一般的なフレームワークとして、Caffe、TensorFlow、CNTK、MxNet、PyTorchを挙げることができます。

これらのプラットフォームの大半は、CPU（中央演算装置）またはGPU向けに最適化されており、特にGPUを使用する場合は非常に高速に動作します。ただし、DSP（デジタル信号プロセッサ）のような低消費電力の組込みデバイスはサポートされていません。DSPの消費電力はGPUよりかなり小さいので、DSPプロセッサを使用するシステムは小型のケース内に配置するのが普通で、この場合は放熱性能が限定されています。また、ポータブル・デバイスに搭載する場合、バッテリから供給される電力が限定されています。

TIはDSPサポートに関するギャップに取り組む目的で、TIDLコンポーネント・スイートを開発しました。TIDLは、ディープ・ラーニング・モデルのトレーニングに対応しておりません。この機能は、一般的なディープ・ラーニング・フレームワークで処理するのが最善だからです。代わりに、TIDLはディープ・ラーニングの推論部分に対応します。サポートされているネットワークでトレーニングが完了したモデルを使用し、そのモデルを、非常に高い速度で、TIのTDAファミリに属する製本など、サポート対象の低消費電力組込みプロセッサで実行します。

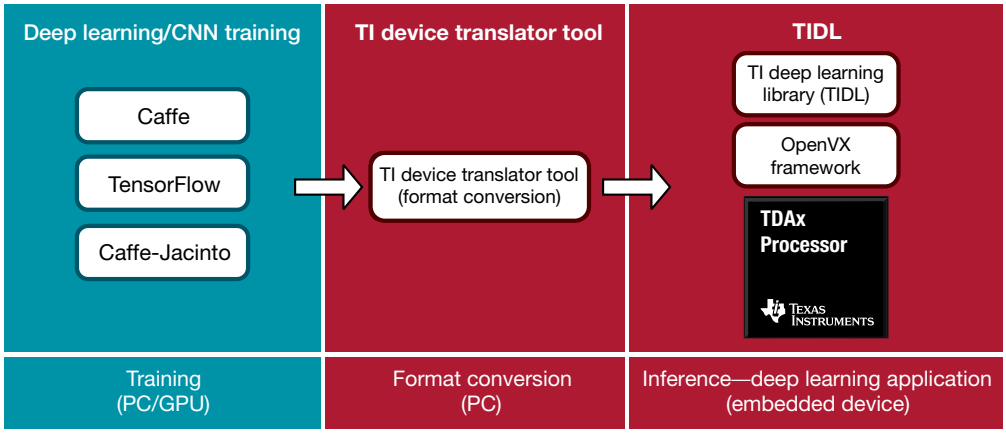

図 1. TIDL 開発フロー。

TIデバイス変換ツールを使用すると、オープン・フレームワークを開発し、PCから組込みデバイスへの移植を実行するプッシュボタンを利用することができます。TIDLは組込みデバイス向けの開発を抽象化し、高効率の実装を実現するほか、プラットフォームのスケーラビリティが高いという特長もあります。

機能

すでに説明したように、TIDLの目的は、使いやすさを向上させること、および最適化された推論機能を実現することです。使いやすさを向上させるために、トレーニングの完了したネットワーク・モデルをTIDLライブラリ内で使用する方法を提供しています。したがって、主な特長の1つは、一般的なフレームワークからのトレーニング完了済み出力を、TIDLが理解できることです。

TIDLは、ソフトウェア最適化を実施します。具体的には、最適化された推論を実現し、基盤となるハードウェア・リソースを最適な方法で使用します。また、CNNで必要とされる処理数を低減するスパース（疎空間）畳み込みのようなアルゴリズム簡略化も実現します。

TIDLは、以下の機能も実現します。

- **複数の層タイプ：**CNNのようなディープ・ラーニング・モデルは、複数の層で形成されています。個別の層は通常、フィルタ、ReLU（rectification linear unit、調整線形ユニット）演算、ダウンサンプリング演算（通常の呼び名は平均プーリング、最大プーリング、またはストライド化）、要素単位の加算、連結、一括正規化、全結合行列乗算のような特定の算術演算で形成されています。TIDLは、CaffeやTensorFlowなどのフレームワーク内に存在している一般的なCNN層の大半をサポートしています。
- **スパース畳み込み：**大部分の重みが0のときに、値が0である複数の係数を活用し、より高速に動作する畳み込みアルゴリズムを、スパース（疎空間）畳み込みと呼びます。TIDLが使用するのは、効率的な畳み込みアルゴリズム、つまりスパース・モデルを使用して、より高速に動作するアルゴリズムです。スパース性が高いときは、速度向上がかなり顕著になる可能性があります。
- **量子化推論とリアルタイム量子化：**Cトレーニング完了モデルは、浮動小数点モデルです。ただし、浮動小数点は、低消費電力の組込みデバイスの実行速度を考慮すると、最善の方法ではありません。したがって、浮動小数点モデルを変換し、推論が固定小数点演算を実行できる形態にすることが重要です（例を挙げると、8ビットまたは16ビットの整数乗算を使用する畳み込み）。TIDLとそのデバイス変換ツールは、浮動小数点から固定小数点への変換を自動的に実行するので、TIDLで使用する固定小数点の推論を実施する目的で、トレーニング側のアルゴリズムやフレームワークは何も特別な処理を行う必要はありません。この動作を、リアルタイム量子化と呼びます。これは、実行速度を大幅に向上させる洗練された機能であり、入力信号の特性や中間層の出力が変動する場合にも対処できます。TIDLは、8ビットと16ビット両方の量子化をサポートしています。いくつかの一般的なネットワークの場合、量子化に伴う精度の低下は小規模で済みます。
- **ホスト・エミュレーション：**実際にTIDLは組込みデバイス上で動作するのに対し、ホスト・エミュレーション・モードを使用すると、正常性チェックを実行できます。ホスト・エミュレーション・モードでは、TIDLはホストPCで動作し、CNNネットワークの各層のエミュレーションを実施して、予期される出力を生成します。したがって、実際に組込みデバイスを使用することなく、デバイスで予期される結果を確認することができます。
- **さまざまなトレーニング・フレームワークのサポート：**TIDLデバイス変換ツールは、BVLC/Caffe、TensorFlow、NVIDIA/Caffe、TIDSP/Caffe-Jacintoのトレーニング完了モデルとの互換性があります。これらの各ツールには、独自の長所があります。実際の要件に最適なツールを選択することができます。

- **低消費電力:** フレーム全体のセマンティック・セグメンテーションを15fps（フレーム／秒）で実行する場合、TDA2x SoC（システム・オン・チップ）の演算消費電力はわずか2.5Wで済みます。

スパース畳み込み

ターゲット・デバイスの計算能力で適切に賄える範囲まで、ネットワーク全体の複雑度を制限することが望まれます。通常、畳み込みの各層は最も演算集中度（計算負荷）が高く、推論の実行速度を決定する要因になります。したがって、畳み込みの各層の複雑度を低減することが重要です。TIDLはスパース畳み込みをサポートしています。この方式は、0の係数が多数存在する場合は、かなり高速に推論を実行できます。

スパース畳み込みアルゴリズムを使用すると、重みが0である場合は常に、乗算が不要になります。スパース・トレーニングの手法により、大半の畳み込みの層で80%以上のスパース性を実現できます。つまり、80%の畳み込みの重みを0にすることができます。TIで実験したところ、畳み込みの各層で重みの80%近くが0の場合、実行速度が4倍に上昇するという結果が得られました。スパース性はオプションとして利用できますが、TIDLはスパースを使用しない従来のモデルで動作させることもできます。

トレーニング

現時点でサポートしているトレーニング・フレームワークは、CaffeとTensorFlowです。言い換えると、TIDLはデバイス変換ツールを使用して、これらのフレームワークでトレーニングが完了したモデルをインポートすることができます。すでに説明したように、TIDLは、BVLC/Caffe、NVIDIA/Caffe、TIDSP/Caffe-Jacintoなど、Caffeのさまざまな派生版をサポートしています。

Caffe-Jacintoは、Caffeのカスタム派生版であり、スパース性を考慮してモデルのトレーニングを実施するツールを提供します。Caffe-Jacintoモデルは、スパース性を考慮したトレーニングの開始に役立ち、トレーニングを進める方法に関する詳細なドキュメントも付属しています。

畳み込みの重みを強制的に0にすると、導入するアルゴリズムの精度が低下する可能性があります。たとえば、スパースを採用した結果、トレーニングが完了した画像分類用ネットワーク・モデルで、精度の低下が25%に達することを避けたいとしましょう（実際は、精度低下を1%か2%にとどめることが望まれます）。精度の大幅な低下を招かずに、スパース性を採用するトレーニング・モデルは、トレーニング・フェーズの重要な要素の1つです。参考文献[2]は、スパース化を使用するトレーニングに関する付加的な詳細を掲載しています。

トレーニング時のスパース化が役に立つのは、推論フレームワーク（この例ではTIDL）が、スパース畳み込みを効率的に実行する能力がある場合のみです。Caffe- Jacintoは、TIDLでかなり高速に実行できるスパース・モデルを生成するための適切なトレーニング・フレームワークです。

デバイス変換ツール

トレーニングは浮動小数点で実行してもかまいません。浮動小数点モデルから固定小数点モデルへの変換は、デバイス変換ツールとTIDLの内部でリアルタイム実行されます。この方法は、最大の使いやすさを実現します。量子化に関する懸念なしでトレーニングを進めることができるからです。

結果

参考文献[3]と[4]は、TDA2車載用プロセッサでリアルタイム・セマンティック・セグメンテーションを実行する目的でTIDLを使用したデモを掲載しています。**図2**は、セマンティック・セグメンテーションの出力をカラーで示したサンプル・フレームです。赤紫の色は道路に分類されたピクセル、青紫は車両に分類されたピクセル、赤は歩行者と自転車に分類されたピクセル、黄色は信号機（この図では表示されていない）に分類されたピクセルを示しています。

図 2. *TDA2 SoC で TIDL を使用したセマンティック・セグメンテーション。*

Inference method	Configuration for inference	Giga multiply accumulations per second (MACs)	Giga cycles	Time (ms)	Frames per second (fps)
Dense	JSegNet21 nonsparse	8.843	0.700	194.44	5.14
Sparse	JSegNet21 sparse (80%)	1.540	0.188	52.22	20.22

表 1. *1,024 x 512 ピクセルの解像度の画像 1 枚に対するセマンティック・セグメンテーションの推論を TDA2x SoC で実行する性能の測定。*

この実験で観察された結果では、80%のスパースを生成したところ、代表的なCNNネットワークの分類精度低下は約1%でした。スパース化と量子化に起因する合計の精度低下は2%以内でした。セマンティック・セグメンテーションに関する同様の観察も類似の結果になりました。CNNネットワーク構造の詳細と結果は、[2]と[5]に掲載されています。

表1に、TIDLを使用してTDA2 SoCでセマンティック・セグメンテーション・ネットワークを実行した結果を示します。表からわかるように、約80%のスペース性を達成すると、1,024×512ピクセルでフレーム全体のセマンティック・セグメンテーションを実施するアプリケーションで、推論速度は約5fpsから約20fpsに向上します。

ネットワーク構成の選択方法

一般的なネットワークはTIDLで実行できますが、低消費電力の組込みデバイスには大電力（なおかつ高コスト）のGPUと同等の演算能力はありません。導入するネットワークは、このデバイスの能力の範囲内に収まることが必須です。この能力は、組込みデバイスによって異なります。

アルゴリズム・デベロッパーは、時にはモデル・サイズ（モデル内で使用するパラメータの数）に注目して推論の複雑度を判定することがあります。ただし、車載アプリケーションの場合、これは小さな課題です。推論の複雑度は、乗算の回数、入出力操作や重みの転送に伴うデータ転送の要件など、複数の要件に依存します。住宅内ネットワーク（residential networks、ResNets）のようなモデルでは、負荷が重い包括的なネットワーク接続型の多数の層を使用することはないので、特定サイズの画像の推論を実施する場合に必要とされる乗算の回数が、多くの場合は複雑度に関する適切な指標になります。

Caffe-Jacintoモデルが示す例を観察して、TDA2xデバイスで実施する推論に適したネットワークを理解することもできます。CNNに使用できる演算能力の向上が予期される将来は、TIのADAS SoCがかなり複雑なモデルを実行できる可能性が高いと考えられます。

TIDLの入手方法

TIDLは、TIのビジョン用プロセッサ・ソフトウェア開発キット（SDK）の一部であり、セマンティック・セグメンテーションをベースとした、すぐに使用できるディープ・ラーニングのデモが付属しています。ビジョンSDKのうち、<VSDK>\ti_components\algorithms_ codecs [6] でTIDLが見つけることができます。

TIDLパッケージには、使用方法に関する資料、さまざまな層の性能、変換と推論を示すためのサンプル・ネットワーク、その他の関連情報が付属しています。組込みビジョン・エンジン（EVE）と、TDA2、TDA2P、TDA3の各デバイスに搭載されているC66x DSPコアの両方がTIDLをサポートしています。また、他のシステム複雑度を理解していなくても、開発中のネットワークの実行と性能測定を自動的に処理するスタンドアロンのテスト・ベンチも付属しています。

リファレンス

1. Jacinto TDAx ADAS SoCs、ADAS用のヘテロジニアス・ハードウェアとソフトウェアのアーキテクチャを採用
2. “Sparse, Quantized, Full Frame CNN for Low Power Embedded Devices”（英語）プロセス全体のうちトレーニング部分に注目し、スパース性を導入する際の詳細について説明。
3. “TI’s Deep Learning-Based Semantic Segmentation on TDA Processors.”（英語）
4. “TI のディープ・ラーニング・ベース・セマンティック・セグメンテーションに関するデモ.”（英語）
5. “Caffe-Jacinto – embedded deep learning framework.”（英語）
6. “TI Vision SDK, Optimized Vision Libraries for ADAS Systems.”（英語）

S-0107

Printed in Japan 2018.02

JAJY077

TIの設計情報およびリソースに関する重要な注意事項

Texas Instruments Incorporated ("TI")の技術、アプリケーションその他設計に関する助言、サービスまたは情報は、TI製品を組み込んだアプリケーションを開発する設計者に役立つことを目的として提供するものです。これにはリファレンス設計や、評価モジュールに関係する資料が含まれますが、これらに限られません。以下、これらを総称して「TIリソース」と呼びます。いかなる方法であっても、TIリソースのいずれかをダウンロード、アクセス、または使用した場合、お客様(個人、または会社を代表している場合にはお客様の会社)は、これらのリソースをここに記載された目的にのみ使用し、この注意事項の条項に従うことに合意したものとします。

TIによるTIリソースの提供は、TI製品に対する該当の発行済み保証事項または免責事項を拡張またはいかなる形でも変更するものではなく、これらのTIリソースを提供することによって、TIにはいかなる追加義務も責任も発生しないものとします。TIは、自社のTIリソースに訂正、拡張、改良、およびその他の変更を加える権利を留保します。

お客様は、自らのアプリケーションの設計において、ご自身が独自に分析、評価、判断を行う責任がお客様にあり、お客様のアプリケーション(および、お客様のアプリケーションに使用されるすべてのTI製品)の安全性、および該当するすべての規制、法、その他適用される要件への遵守を保証するすべての責任をお客様のみが負うことを理解し、合意するものとします。お客様は、自身のアプリケーションに関して、(1) 故障による危険な結果を予測し、(2) 障害とその結果を監視し、および、(3) 損害を引き起こす障害の可能性を減らし、適切な対策を行う目的での、安全策を開発し実装するために必要な、すべての技術を保持していることを表明するものとします。お客様は、TI製品を含むアプリケーションを使用または配布する前に、それらのアプリケーション、およびアプリケーションに使用されているTI製品の機能性を完全にテストすることに合意するものとします。TIは、特定のTIリソース用に発行されたドキュメントで明示的に記載されているもの以外のテストを実行していません。

お客様は、個別のTIリソースにつき、当該TIリソースに記載されているTI製品を含むアプリケーションの開発に関連する目的でのみ、使用、コピー、変更することが許可されています。明示的または黙示的を問わず、禁反言の法理その他どのような理由でも、他のTIの知的所有権に対するその他のライセンスは付与されません。また、TIまたは他のいかなる第三者のテクノロジまたは知的所有権についても、いかなるライセンスも付与されるものではありません。付与されないものには、TI製品またはサービスが使用される組み合わせ、機械、プロセスに関連する特許権、著作権、回路配置利用権、その他の知的所有権が含まれますが、これらに限られません。第三者の製品やサービスに関する、またはそれらを参照する情報は、そのような製品またはサービスを利用するライセンスを構成するものではなく、それらに対する保証または推奨を意味するものでもありません。TIリソースを使用するため、第三者の特許または他の知的所有権に基づく第三者からのライセンス、もしくは、TIの特許または他の知的所有権に基づくTIからのライセンスが必要な場合があります。

TIのリソースは、それに含まれるあらゆる欠陥も含めて、「現状のまま」提供されます。TIは、TIリソースまたはその仕様に関して、明示的か暗黙的かにかかわらず、他のいかなる保証または表明も行いません。これには、正確性または完全性、権原、続発性の障害に関する保証、および商品性、特定目的への適合性、第三者の知的所有権の非侵害に対する黙示の保証が含まれますが、これらに限られません。

TIは、いかなる苦情に対しても、お客様への弁護または補償を行う義務はなく、行わないものとします。これには、任意の製品の組み合わせに関連する、またはそれらに基づく侵害の請求も含まれますが、これらに限られず、またその事実についてTIリソースまたは他の場所に記載されているか否かを問わないものとします。いかなる場合も、TIリソースまたはその使用に関連して、またはそれらにより発生した、実際的、直接的、特別、付随的、間接的、懲罰的、偶発的、または、結果的な損害について、そのような損害の可能性についてTIが知らされていたかどうかにかかわらず、TIは責任を負わないものとします。

お客様は、この注意事項の条件および条項に従わなかったために発生した、いかなる損害、コスト、損失、責任からも、TIおよびその代表者を完全に免責するものとします。

この注意事項はTIリソースに適用されます。特定の種類の資料、TI製品、およびサービスの使用および購入については、追加条項が適用されます。これには、半導体製品(http://www.ti.com/sc/docs/stdterms.htm)、評価モジュール、およびサンプル(http://www.ti.com/sc/docs/sampterms.htm)についてのTIの標準条項が含まれますが、これらに限られません。