組み込みシステムへのディープ・ラーニングの導入

Texas Instruments

Mark Nadeski
Embedded Processing
Texas Instruments

ディープ・ラーニングの将来性の高さは疑いがないでしょう

ディープ・ラーニングは、インターネット、いえ、それ以前のトランジスタと同じくらい世の中に大きなインパクトを与えるであろう基盤技術だと言われてきました

コンピュータの処理能力が大幅に向上し、ラベル付きデータセットを大量に利用できるようになったことで、ディープ・ラーニングはこれまでも、画像分類、仮想アシスタント、ゲーム・プレイングに大きな進歩をもたらしており、今後も数知れないほどの業界でそうなるでしょう。従来の機械学習と比べて、ディープ・ラーニングは高い精度をもたらし、汎用性を高め、ビッグデータがより活用できるようになり、すべての側面で特定分野の専門知識の必要性は減少しています。

幅広い業態でディープ・ラーニングがその期待に応えられるようにするには、ディープ・ラーニング推論（トレーニングされたディープ・ラーニングのアルゴリズムを実行する部分）を組み込みシステムに展開できるようにする必要があります。このように展開するにあたっては、特有の課題や要件があります。このホワイト・ペーパーでは、組み込みシステムにディープ・ラーニングを展開する上での課題と、ディープ・ラーニングに適した組み込みプロセッサを選択するときに主に考慮すべきことを取り上げます。

トレーニングと推論

ディープ・ラーニングは、トレーニングと推論という主に2つのフェーズから成り立ちます。以下の**図1**で示すとおり、この2つはまったく別々の処理プラットフォームで行うことが可能です。ディープ・ラーニングのトレーニング・フェーズは通常、オフラインのデスクトップPC上かクラウド上で行われ、大量のラベル付きデータセットをディープ・ニューラル・ネットワーク（DNN）に流し込みます。このフェーズではリアルタイムの処理性能や電力消費は問題とされません。トレーニング・フェーズの結果が、トレーニング済みニューラル・ネットワークになります。このニューラル・ネットワークを展開すると、組み立てラインでボトルを検分したり、部屋の中にいる人物の数を数えたりトラッキングしたり、偽造紙幣かどうかを見分けたりというような、特定の作業を実行することが可能になります。

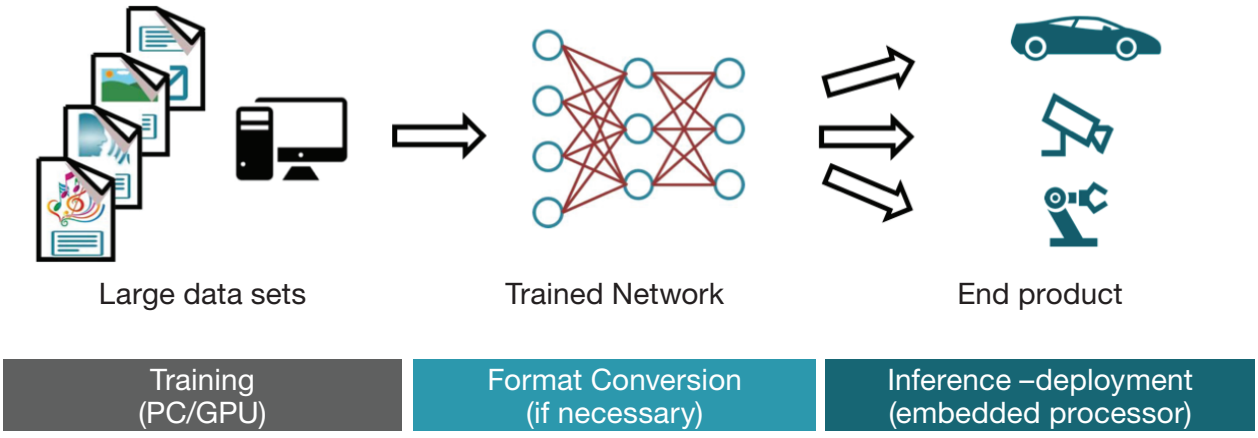

図1：*従来のディープ・ラーニング開発の流れ*

アルゴリズムを実行する機器にトレーニング済みニューラル・ネットワークを展開したものが、推論とも呼ばれます。組み込みシステムに課される制約を考えると、ニューラル・ネットワークのトレーニングは、推論を実行するプラットフォームとは別の処理プラットフォームで行われることが多いでしょう。このホワイト・ペーパーでは、ディープ・ラーニングの推論部分に対応するプロセッサの選択に焦点を当てます。このホワイト・ペーパーで「ディープ・ラーニング」というときは、推論のことを指します。

エッジでのディープ・ラーニング

センサーが情報を収集する場所のなるべく近く、つまりネットワークのエッジで演算処理を行うという考え方は、近年の組み込みシステムで中心となるポイントです。ディープ・ラーニングについては、ネットワーク・エッジでのインテリジェンスの利用や自律運用を可能にするために、この考え方がより重要になります。工場フロアの自動化機械や産業用ロボットから、自動で動く家庭用のロボット掃除機、農地で利用する農業用トラクターにいたるまでの多くの用途で、ローカルで処理が行われなければなりません。

ローカル処理を行う理由は用途により多岐にわたります。以下に挙げるのは、ローカル処理の必要性を高める原因となる考慮事項のうちのごく一部です。

- **信頼性**　インターネット接続が必ずしも当てにできるわけではありません。
- **低レイテンシ**　多くの用途で、即時応答が求められます。用途によっては、どこか別の場所にデータを送信して処理してもらうために生じる遅延時間が許容できないかもしれません。
- **プライバシー**　データには機密性が高いものがあり、その場合は外部に送信したり保管したりすべきではありません。
- **帯域幅**　ネットワークの帯域幅の効率は、しばしば主要な懸念事項となります。すべてのユースケースで持続的にサーバーに接続できるわけではありません。
- **電力**　組み込みシステムでは消費電力が常に優先課題です。データの移動には電力を消費します。より遠くまでデータを移動するには、より多くのエネルギーが必要です。

ディープ・ラーニング向けの組み込みプロセッサの選択

ローカル処理を必要とする考慮事項の多くは、組み込みシステムが本来持っている課題と重なります。特に電力と信頼性がそうです。他にも、組み込みシステムにはシステムの物理的制約により考慮すべき要素がいくつかあります。サイズ、メモリ、電力、温度、製品寿命に関連して、もちろんコストも含めて、変えられない要件があることがしばしばです。

特定の組み込みアプリケーションでこれらすべての要件と課題のバランスをとる中で、エッジでディープ・ラーニング推論を実行するプロセッサを選択する際にいくつか考慮すべき重要な要素があります。

- **アプリケーション全体を考慮**　処理ソリューションを選ぶ前に最初に理解しておくことの1つは、全体的なアプリケーションの範囲です。必要な処理は推論実行だけなのか、それとも従来のマシン・ビジョンにディープ・ラーニング推論を追加して組み合わせて使用することになるのか、といったことです。従来のコンピュータ・ビジョン・アルゴリズムをシステムが高レベルで実行し、それから必要なときにディープ・ラーニングを実行する方が効率性が高いことがよくあります。例えば、高fps（フレーム数／秒）の入力イメージ全体で古典的なコンピュータ・ビジョン・アルゴリズムを実行してオブジェクトをトラッキングし、識別されたイメージの特定のサブ領域に低fpsでディープ・ラーニングを使用してオブジェクト分類を実行できます。この例では、複数のサブ領域にまたがるオブジェクト分類には推論インスタンスが複数必要になるか、または各サブ領域で別々の推論を実行する可能性があります。各サブ領域で別々の推論を実行する場合、従来のコンピュータ・ビジョンとディープ・ラーニングの両方を実行することに加えて、別々のディープ・ラーニング推論のインスタンスを複数実行できる処理ソリューションを選択する必要があります。**図2**は、イメージのサブ領域での複数のオブジェクトのトラッキングと、トラッキングされる各オブジェクトで分類を実行する場合の使用例です。

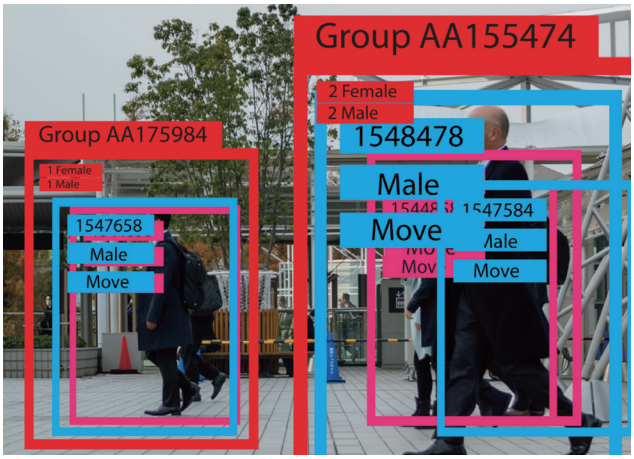

図2：組み込みディープ・ラーニングを使用したオブジェクト分類の例

- **適正な性能ポイントの選択**　アプリケーション全体の範囲の感覚がつかめたところで、アプリケーションのニーズを満たすためにどれくらいの処理性能が必要かを理解することが重要になってきます。性能のかなりの部分がアプリケーションに応じて異なるため、ディープ・ラーニングに関してはこれを理解するのは難しいことがあります。例えば、ビデオ・ストリーミングでオブジェクトを分類する畳み込みニューラル・ネットワーク（CNN）の性能はさまざまな要素に左右されます。ネットワークで使われるレイヤ、ネットワークの深さ、ビデオ解像度、fps要件、ネットワークの重みで使われるビット数などはそのごく一部です。しかし組み込みシステムでは、試行しながら必要な性能の程度を把握することが重要です。というのも、一般的に課題に対してプロセッサ性能が高すぎることは、消費電力やサイズ、コストの増加とのトレードオフとなるからです。プロセッサがResNet-10の1080p/30fpsに対応し、高電力で集中型のディープ・ラーニング・アプリケーションで一般的によく使われるニューラル・ネット・モデルを扱えるとしても、目的とする244×244領域で、より組み込みに適したネットワークを運用するアプリケーションには行き過ぎでしょう。

- **組み込みを考慮**　適正なネットワークを選択することは、適正なプロセッサの選択と同じくらい重要です。すべてのニューラル・ネット・アーキテクチャが組み込みプロセッサに適合するわけではありません。より動作の少ないモデルに限定した方が、リアルタイムの性能を達成するのには役立ちます。組み込みスペース向けに設計されていない、AlexNetやGoogleNetのような名の知れたネットワークに代わって、精度を犠牲にして演算処理を大幅に省力化するような、より組み込みに適したネットワークのベンチマークを優先すべきです。同様に、これらのネットワークを組み込み領域に取り入れるツールを効果的に活用可能なプロセッサを探しましょう。例えば、ニューラル・ネットワークは多数のエラーを許容できます。量子化の利用は、なるべく精度を落とさずに性能要件を抑えるためによい方法です。組み込み領域では、動的量子化をサポートでき、スパース性（非ゼロの値の数を制限）などのその他の手法を効果的に利用できるプロセッサが、よい選択肢となります。

- **使いやすさを確保**　使いやすさとは、開発が容易であることと、評価が容易であることを意味します。先に述べたように、プロセッサ性能を適正なサイズにすることは、設計の上で重要な考慮事項です。このことを正しく行うには、選択したネットワークを既存のプロセッサ上で実行することが最善の方法です。ネットワーク・トポロジを仮定して、ある特定のプロセッサで実現可能な処理能力を表示するツールも提供されています。これにより実際のハードウェアや確定されたネットワークを必要とせずに性能評価ができます。開発では、CaffeやTensorFlowといった一般的に使用されるフレームワークからトレーニング済みネットワーク・モデルを簡単にインポートできることが不可欠です。

ディープ・ラーニングに適したプロセッサを選択するとき、検討すべきプロセッサの種類がさまざまあり、そのすべてに強みや弱点があります。グラフィックス処理ユニット（GPU）はネットワーク・トレーニングに広く使われているため、通常は最初に検討されます。GPUは処理能力は非常に高いのですが、組み込みアプリケーションでよく見られる消費電力やサイズ、コストの制約を考えると、組み込み領域で本格的に広まるにはいたっていません。

ディープ・ラーニングが普及するにつれて、消費電力とサイズが最適化された「推論エンジン」もだんだんと提供されるようになってきました。推論エンジンは、特にディープ・ラーニング推論を実行することを目的とした専門のハードウェア製品です。エンジンによっては1ビットの重みを使用するよう最適化され、キーフレーズ検出のような単純な機能を実行できますが、省電力化と演算処理の省力化のために大幅に最適化すると、システム機能や精度が犠牲となってしまいます。推論エンジンが小さいと、オブジェクトの分類や精巧な作業が必要なアプリケーションには十分な能力が得られないかもしれません。これらのエンジンを評価するときは、アプリケーションに対して適切なサイズかどうか確認してください。ディープ・ラーニング推論の他にも処理が必要なアプリケーションのとき、これらの推論エンジンの限界が来ます。往々にして、システム内の他のプロセッサとともに、ディープ・ラーニングのコプロセッサとして機能するようエンジンを使用することが必要になります。

統合されたシステム・オン・チップ（SoC）は、多くの場合組み込み領域ではよい選択肢です。その理由は、ディープ・ラーニング推論を実行可能なさまざまな処理要素を収容することに加えて、SoCは組み込みアプリケーション全体をカバーするのに必要な多数のコンポーネントも統合しているからです。統合SoCの中には、ディスプレイ、グラフィック、ビデオ・アクセラレーションおよび産業ネットワークの機能を内蔵するものがあり、ディープ・ラーニングを実行するにとどまらないシングルチップ・ソリューションを実現します。

ディープ・ラーニング向けに高度に統合されたSoCの一例が、テキサス・インスツルメンツのAM5749デバイスです（**図3**を参照）。AM5749は、システム処理用にArm® Cortex® -A15コアを2個、従来のマシン・ビジョン・アルゴリズム処理用にC66xデジタル信号プロセッサ（DSP）コアを2個、推論実行用に組み込みビジョン・エンジン（EVE）を2個備えています。TIディープ・ラーニング（TIDL）ソフトウェア製品にはTIDLライブラリが含まれており、C66x DSPコアかEVEのいずれかで稼働し、デバイス上で同時に複数の推論を実行することが可能です。その上、AM5749は豊富なペリフェラル・セットとして、EtherCATといった工場フロアのプロトコルを実装するための産業用通信サブシステム（ICSS）、ビデオのエンコード／デコードと3Dおよび2Dグラフィックのためのアクセラレーションを備えて、ディープ・ラーニングの実行も行う組み込み領域でのこのSoCの使用を促進します。

多くの場合で、組み込みアプリケーションに対応するプロセッサを選択することが、製品用の部品選択時に最も重要となりますが、ネットワーク・エッジにディープ・ラーニングを導入するような、業界に変化をもたらす多くの製品でもそのようなことが言えます。アプリケーション全体を考慮すること、適正な性能ポイントを選択すること、組み込みを考慮すること、そして使いやすさを確保することなど、このホワイト・ペーパーで述べてきたこれらの議題が、プロセッサを選択する上で検討すべき課題に対する重要な指標となれば幸いです。

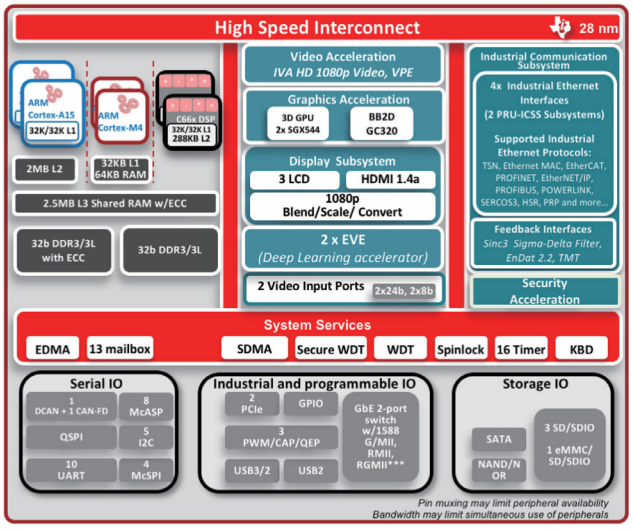

図3：*Sitara™ AM5749 SoCのブロック図*

関連Webサイト：

- Sitara AM57xプロセッサについてはこちらをご覧ください。
- 組み込みアプリケーション向けディープ・ラーニングを行うSitara AM57xプロセッサに対応したプロセッサ・ソフトウェア開発キット（SDK）のダウンロードはこちら。
- 組み込みアプリケーション用ディープ・ラーニング推論のリファレンス・デザインのダウンロードはこちら。

Printed in Japan 2018.10

JAJY083

重要なお知らせと免責事項

TI は、技術データと信頼性データ（データシートを含みます）、設計リソース（リファレンス・デザインを含みます）、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションが適用される各種規格や、その他のあらゆる安全性、セキュリティ、またはその他の要件を満たしていることを確実にする責任を、お客様のみが単独で負うものとします。上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、TI の販売条件（www.tij.co.jp/ja-jp/legal/termsofsale.html）、または ti.com やかかる TI 製品の関連資料などのいずれかを通じて提供する適用可能な条項の下で提供されています。TI がこれらのリソースを提供することは、適用されるTI の保証または他の保証の放棄の拡大や変更を意味するものではありません。