ベイズ・ディープラーニングの量子化フレームワーク

Bayesian-Torch でベイズ・ニューラル・ネットワークを簡単に量子化してデプロイ

Keyur Ranipa インテル コーポレーション AI フレームワーク・アーキテクト
Vrushabh Sanghavi インテル コーポレーション AI フレームワーク・エンジニア
Ranganath Krishnan インテル コーポレーション AI リサーチ・サイエンティスト

ディープ・ニューラル・ネットワーク（DNN）は、優れた精度とパフォーマンスにより、医療診断、自動運転車、天気予報など、さまざまなセーフティー・クリティカル・アプリケーションにおける不可欠なコンポーネントとなっています。しかし、トレーニング・データの過剰適合や、間違った予測を過信してしまうことがあります。これにより、分布外のデータを検出する能力や、信頼できる信頼区間を含む予測の不確実性に関する情報を提供する能力が制限されます。この問題は、説明可能性と安全性が必要な実際のアプリケーションでは極めて重要です。例えば、医療診断では、モデルは予測に不確実性の尺度を示す必要があります。

データに最も適合する重みパラメーターの決定論的ポイントの推定を見つけることを目的とした従来の DNN モデルとは異なり、ベイズ・ディープ・ニューラル・ネットワーク（BNN）モデルは、ベイズの定理に基づいて重みの事後分布を学習することを目的としています（**図 1**）。ベイズモデルの重みは推論中に学習された分布からサンプリングされ、複数のモンテカルロ・サンプルにより予測の不確実性を推定します。

図 1. 大腸組織構造診断における不確実性に基づく選択的予測

BNN は、重みのサンプリングと複数の確率的フォワードパスにより追加のメモリーと計算コストが必要になるため、決定論的モデルよりもモデルのサイズが大きくなり、推論が遅くなります。これらの欠点のため、BNN を実際のアプリケーションにデプロイする際には大きな課題が生じます。ここで量子化の出番です。量子化は、重みと有効性を 8 ビット整数（int8）などの低精度データ型で表すことにより、BNN 推論のメモリーと計算コストを削減するのに役立ちます。

ここでは、広く利用されているベイズ・ディープラーニング（BDL）向けの PyTorch* ベースのオープンソース・ライブラリー、Bayesian-Torch（英語）を使用して構築された量子化モデルを含む BDL ワークロードを紹介します。Bayesian-Torch は、BDL の低精度の最適化をサポートします。Bayesian-Torch を使用して、量子化されたベイズモデルをインテル® アドバンスト・マトリクス・エクステンション（インテル® AMX）を搭載した第 4 世代インテル® Xeon® スケーラブル・プロセッサーにデプロイすることにより、モデルの精度と不確実性の質を犠牲にすることなく、完全な精度のベイズモデルと比較して ImageNet ベンチマークで 6.9 倍の推論速度の向上を達成しました。

BNN の量子化

Bayesian-Torch は、シンプルな API（`dnn_to_bnn()`）を使用して、あらゆる DNN モデルを BNN にシームレスに変換できます。我々は、IISWC 2023（IEEE International Symposium on Workload Characterization）（英語）で発表された論文「Quantization for Bayesian Deep Learning : Low-Precision Characterization and Robustness（ベイズ・ディープラーニングの量子化 : 低精度の特性化とロバスト性）」で、包括的な量子化ワークフローを紹介しました。以下の 3 つの簡単なステップで、トレーニング後の量子化（PTQ）を BNN に適用できます。

1. **準備 :**「オブザーバーの挿入」などの前処理タスクを実行して、静的量子化のモデルを準備します。
2. **キャリブレーション :** 代表的なデータを使用してキャリブレーションを行い、キャリブレーション統計を取得します。
3. **変換 :** モデルのすべてのテンソルと演算についてスケールとゼロ点を計算し、量子化可能な関数を低精度の関数に置換することにより、完全な精度のベイズモデルを量子化モデルに変換します。

Bayesian-Torch 量子化フレームワークには、PyTorch*（英語）のような高水準 API があります（**図 2**）。

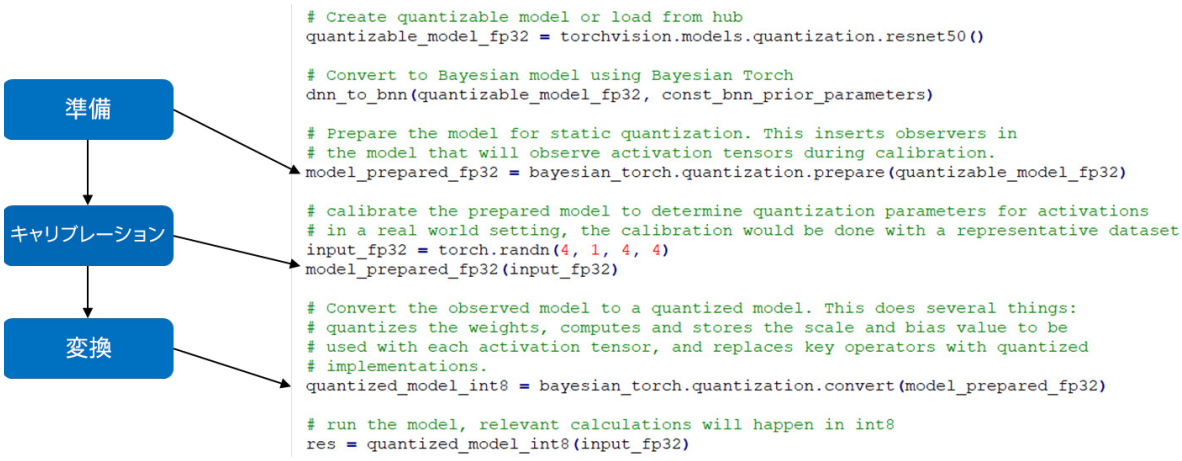


図 2. Bayesian-Torch でトレーニング後の量子化を実装するステップ

ImageNet ベンチマークを使用した ResNet-50 の実験

経験的ベイズ手法（英語）を使用して、重みの事前分布を指定し、事後分布を初期化することにより、平均場変分推論を使用してベイズ ResNet50 モデルをトレーニングしました。重みの事前分布は、Torchvision（英語）ライブラリーで利用可能な事前トレーニング済み ResNet50 決定論的モデルから初期化されます。ImageNet-1K（英語）データセットで Bayesian-Torch（英語）と PyTorch* 2.0 を使用して、初期学習率 0.0001、運動量 0.9、重みの減衰 0.0001 で SGD オプティマイザーを使用し、50 エポック実行してモデルをトレーニングしました。前述したように、トレーニング済みモデルにトレーニング後の量子化を適用し、量子化されたベイズ・ニューラル・ネットワーク（QBNN）を取得しました。

すべての検証は、AI ワークロードを高速化するインテル® AMX が搭載された第 4 世代インテル® Xeon® スケーラブル・プロセッサー上で行われました。インテル® AMX は、推論では INT8 精度を、推論とトレーニングでは Bfloat16 をサポートします。インテル® AMX は、メモリー内の大きな行列の部分行列を保持できる 2 次元レジスター（またはタイル）のセットと、タイル行列乗算アクセラレーターの 2 つのコンポーネントから成る新しい 64 ビット・プログラミング・パラダイムです。インテル® AMX を利用すると、単一の命令をタイルおよびアクセラレーター・ハードウェア上で複数のサイクルで実行できるようになります。

量子化されたベイズモデルの利点を実証するため、ImageNet-1K データセットで 1 つのモンテカルロ・サンプルを使用して、量子化前と量子化後のベイズ ResNet50 モデルの推論スループットを比較しました（**表 1**）。この結果は、量子化されたベイズモデルのスループットが完全な精度のベイズモデルよりも高く、最大 6.9 倍のスピードアップを達成していることを示しています。また、精度をほとんど低下させず、予測される不確実性のキャリブレーション誤差により定量化される不確実性の質を低下させることなく達成されています（**表 2**）。

バッチサイズ	推論スループット ↑ (画像数/秒) BNN (FP32)	QBNN (INT8)	スピードアップ (QBNN vs. BNN)
32	218.5	420.9	1.93x
64	276.4	740.8	2.68x
128	301.1	1171.1	3.89x
256	342.7	1645.6	4.80x
512	366.4	1969.1	5.37x
1024	377.9	2358.3	6.24x
2048	395.1	2735.1	6.92x

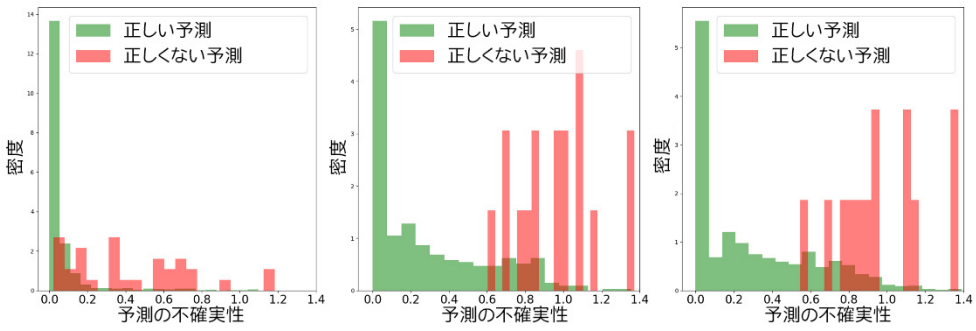
表 1. ImageNet-1K のさまざまなバッチサイズでの BNN と QBNN の推論スループットの比較

	BNN (FP32)	QBNN (INT8)
精度 ↑	76.10	75.71
UCE ↓	8.27	7.26

表 2. ImageNet での BNN と QBNN の精度と不確実性のキャリブレーション誤差（UCE）。矢印は数値が高いほうが良いか低いほうが良いかを示す。

実際のセーフティー・クリティカル・アプリケーション

医療診断、特に大腸組織構造画像分類（英語）に使用される BNN で、この量子化フレームワークを評価します。このデータセットは、ヒト大腸がんの 5,000 枚の組織構造画像のテクスチャーのコレクションです。トレーニングに 4,000 枚の画像、テストに 1,000 枚の画像を使用しました。各 RGB 画像は 150x150x3 です。モデルは画像を 8 つのクラスのいずれかに分類する必要があります。初期学習率 0.0001、運動量 0.9、重みの減衰 0.0001 で SGD オプティマイザーを使用し、50 エポック実行して、ResNet-50 アーキテクチャーの DNN モデルと BNN モデルをトレーニングしました。次に、トレーニング済みモデルにトレーニング後の量子化を適用し、QBNN モデルを取得しました。BNN のロバスト性と信頼性、後述する選択的予測および分布外（OOD）検出タスクについて、量子化後の不確実性の質を評価しました。



(a) DNN (FP32) (b) BNN (FP32) (c) QBNN (INT8)

図 3. 正しい予測と正しくない予測の不確実性の密度ヒストグラム

不確実性の高い予測はドメイン・エキスパートに照会し、不確実性の低い予測に基づいてモデルを評価する、選択的予測（**図 1**）を実行しました。**図 3** は、正しい予測と正しくない予測について DNN、BNN、QBNN から取得された予測の不確実性の密度ヒストグラムです。モデルが正しくない予測を行う場合には高い不確実性が得られ、正しい予測を行う場合には低い不確実性が得られるのが望ましい動作です。ベイズモデルは、DNN よりも信頼できる不確実性の推定を生成し、AI のエキスパートはモデルがいつ失敗するか知ることができることが分かりました。また、**図 3 (c)** から、不確実性の推定の質は量子化の影響を受けないことが分かります。

図 4 (a) は、照会率の関数として精度を比較したものです。モデルの精度が 99.5% を達成するときのドメイン・エキスパートへのサンプルの照会率は、標準 ResNet-50 では 27.5% であるのに対し、BNN と QBNN ではそれぞれ 6% と 8.5% のみです。これは、ドメイン・エキスパートへのデータ照会が少なくなると偽陰性と偽陽性が減ることを示しています。QBNN では DNN と比較して照会率の効率が 78% 向上しています。

Camelyon17-wilds（英語）データセットを使用して、OOD 検出のモデルをテストしました。このデータセットは、リンパ節切片の乳がん転移の全 50 枚のスライド画像から抽出された 450,000 個のパッチで構成されています。検証セットからランダムに選択された 1,000 枚の画像を OOD 検出のテストモデルに使用しました。OOD 検出は、データサンプルが分布データに属しているかどうかを識別する 2 項分類タスクです。教師なしアプローチである（つまり、モデルに OOD データの知識がない）、予測の不確実性の推定を使用して OOD データを検出するモデルの能力を評価します。**図 4 (b)** は、OOD 検出の ROC 曲線下の面積（AUC）の比較です。BNN（AUC 0.83）は、DNN（AUC 0.41）よりも OOD サンプルを検出する能力が優れていることが分かります。また、BNN の量子化は OOD 検出の AUC にほとんど影響を与えていません。

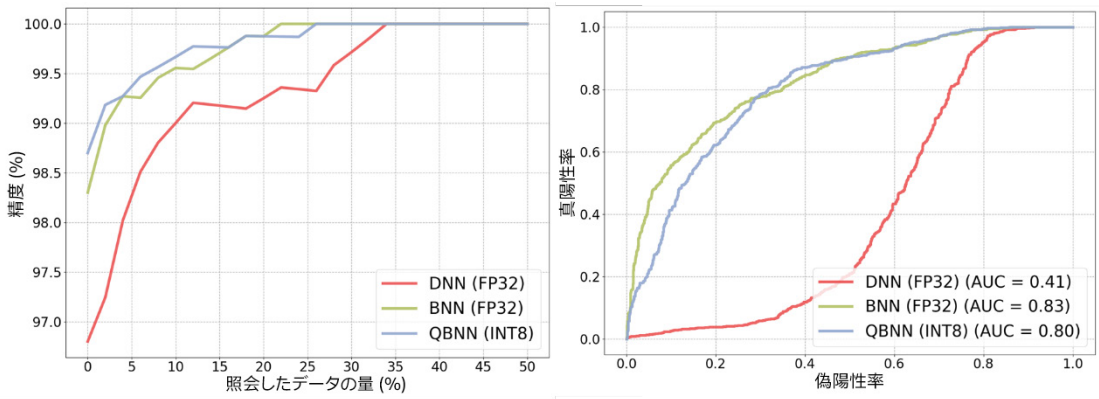


図 4. 予測の不確実性に基づいて照会したデータの量の関数としての精度

まとめ

この記事では、シンプルで使い慣れた API を使用して BNN のトレーニング後の量子化を可能にする、ベイズ・ディープラーニング向けの低精度の最適化フレームワークを紹介しました。Bayesian-Torch 量子化フレームワークを使用することで、最適化された 8 ビット整数（INT8）BNN は、モデルの精度、不確実性の質、ロバスト性を犠牲にすることなく、32 ビット浮動小数点（FP32）BNN と比較して最大 6.9 倍の推論スループットのスピードアップを達成し（**図 5**）、必要なメモリーを 4 分の 1 に減らしました。この結果は、大規模なデータセットと実際のアプリケーションにおける広範な実証分析を通じて得られたものです。量子化フレームワークのコードはオープンソース化されているため、実際のアプリケーションでベイズ・ディープラーニング・モデルが広範にデプロイされることを期待しています。

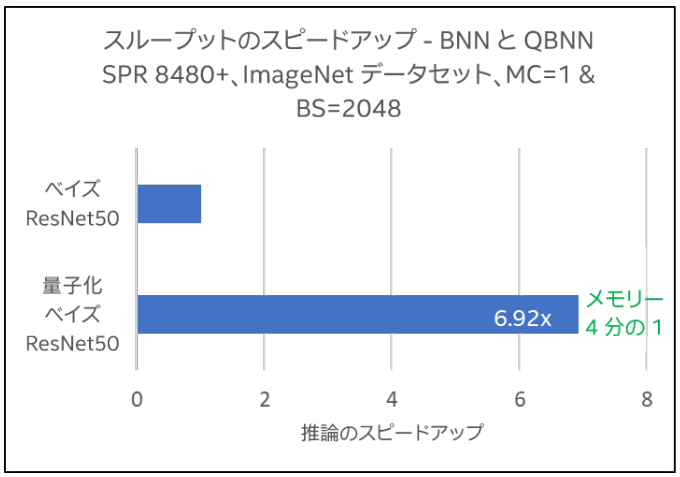


図 5. BNN と QBNN のスループットのスピードアップの比較