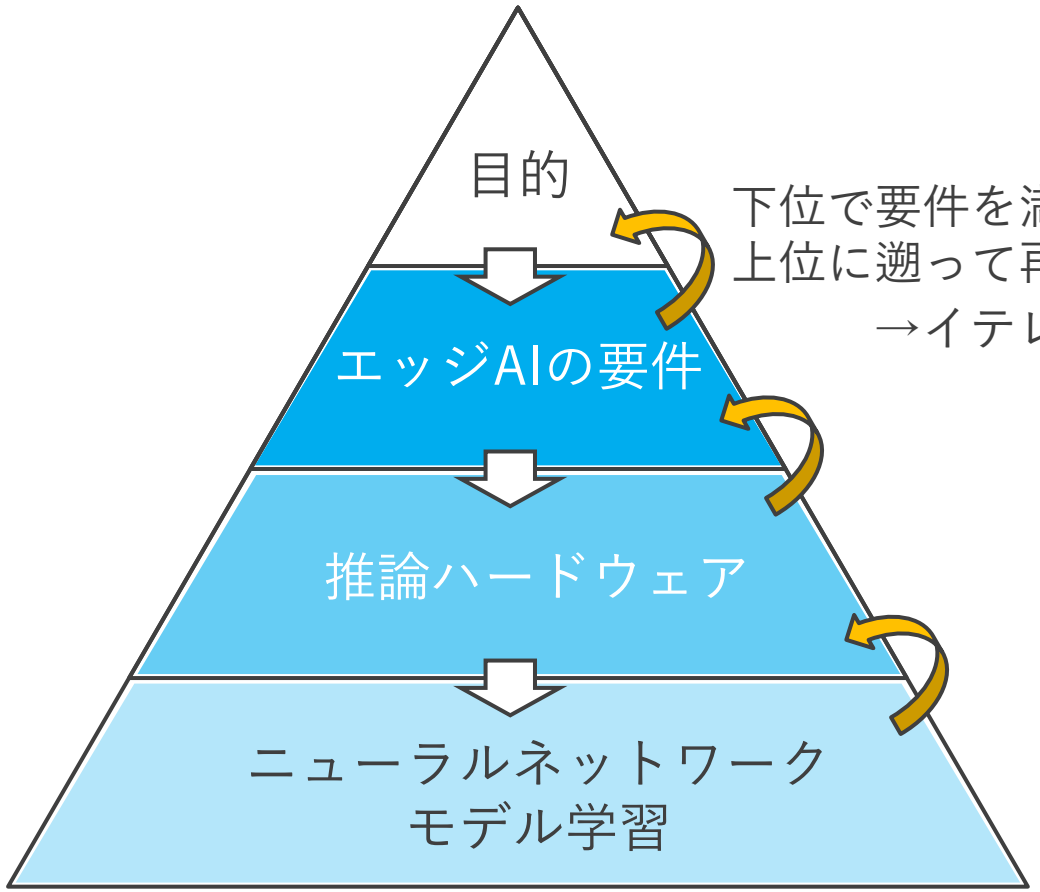
エッジAI実装でおさえておくべき
7つの観点

この資料でお話しさせて頂くこと

ARAYA

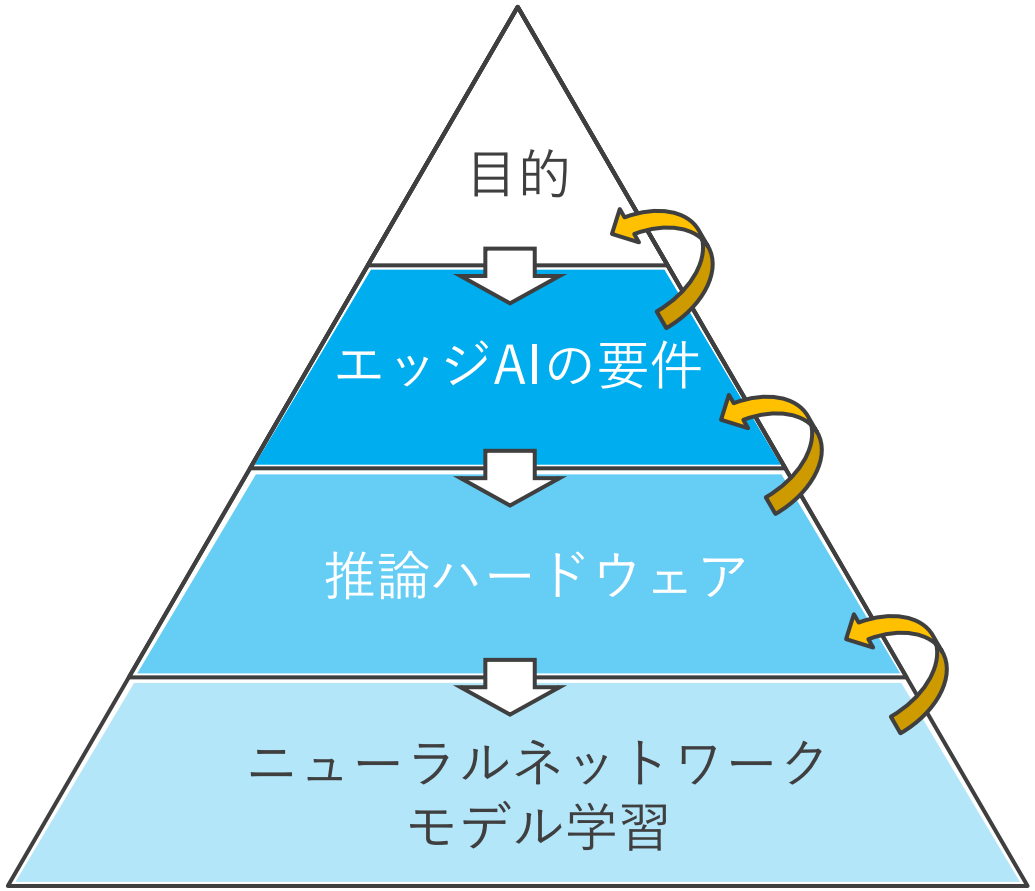
- エッジAIを実装するときの開発プロセス

エッジAI実装の開発プロセス



下位で要件を満たせない場合は
上位に遡って再検討する
→イテレーションが発生する

エッジAI実装の開発プロセス



各フェーズでの検討・実施事項

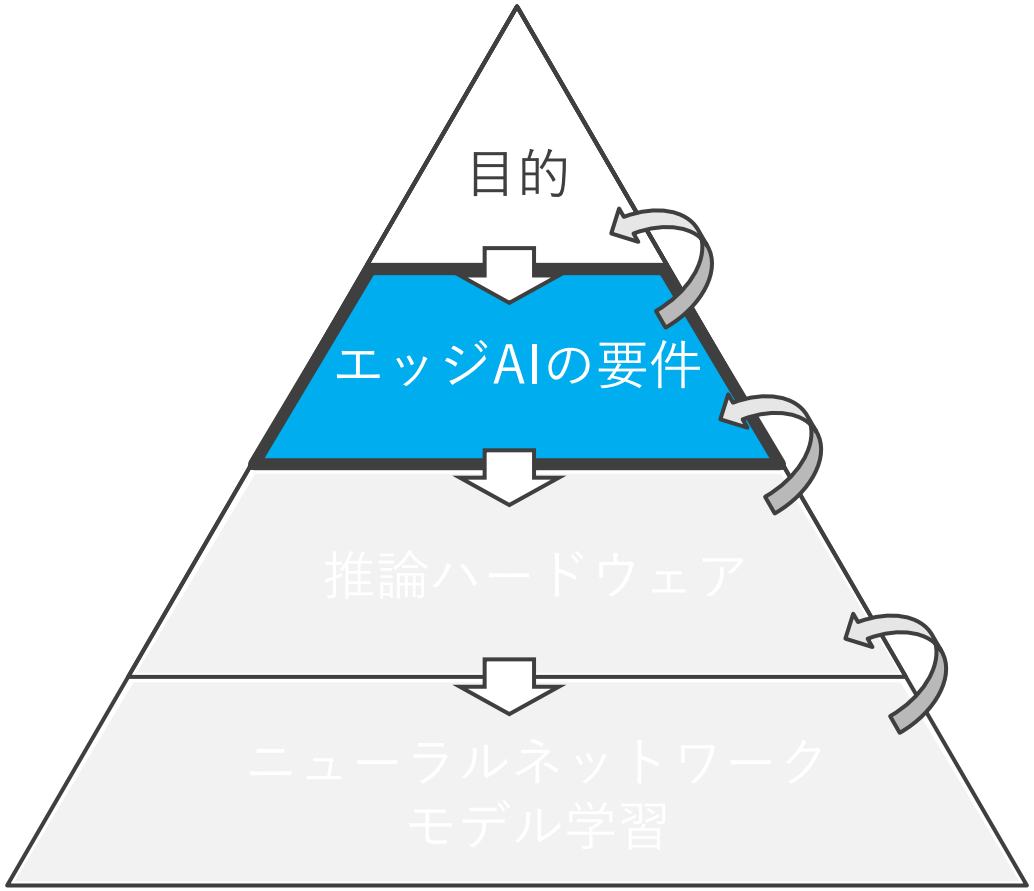
エッジAIの要件	• 目的達成のために、必要な推論タスクを検討 • 目的達成のための推論精度・速度目標を検討
推論ハードウェア	• 目標推論速度を満たすモデル・推論ハードウェアの組み合わせを探索。これにより速度が決まる。
モデル学習	• 通常の学習に加え、推論ハードウェアのための量子化や高速化のためのモデル圧縮を行う。これにより精度が決まる。

この資料でお話しさせて頂くこと

ARAYA

- エッジAIを実装するときの開発プロセス

- 各開発フェーズでおさえておくべき
  観点とベストプラクティス



観点①

本当にエッジAIが必要か？

観点①　本当にエッジAIが必要か？

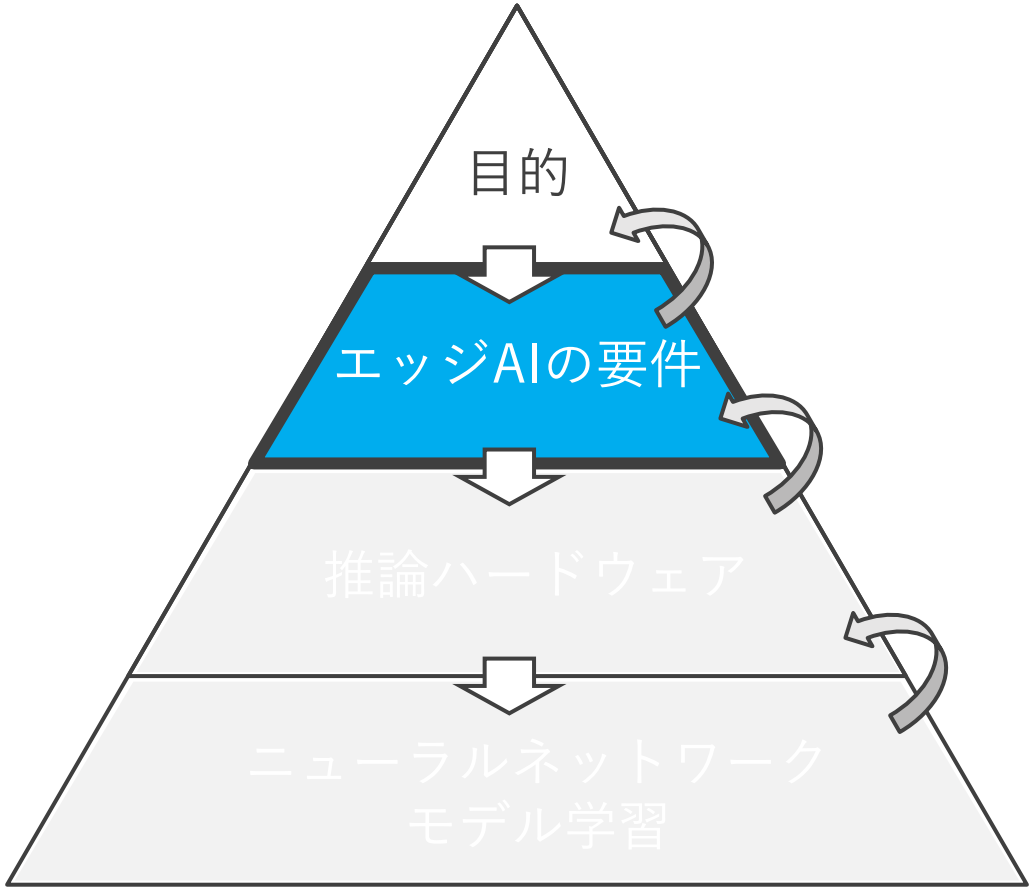
- エッジで実装することのメリットだけでなくデメリットも含めて検討し、メリットがデメリットを上回る場合にのみ、エッジ上での実装を検討すべき。

メリット

- リアルタイム性：通信遅延がないため、毎回ほぼ同じ周期、時間で実行できる。
- 低レイテンシ：エッジで実行した方が、通信に要する時間がかからない分、比較的低いレイテンシで実行できる。
- プライバシーの保護：エッジで処理した結果だけをクラウドに送ることで、プライバシー情報への不正なアクセスを防ぐ。
- 通信費の削減：生データをクラウドに送る必要がないため通信費を削減できる。

デメリット

- モデル管理の煩雑化：モデルを更新する場合は、各エッジデバイスにモデルを配信し更新を確認する必要がある。
- モデルの改善が難しい：生データをエッジ側で処理して捨ててしまうため、エッジで収集したデータを活用してモデルの精度を向上させることができない。
- 初期投資が必要：性能が必要な場合はNPU/GPUなどの購入が必要となるが、そのための初期費用が必要となる。



観点②

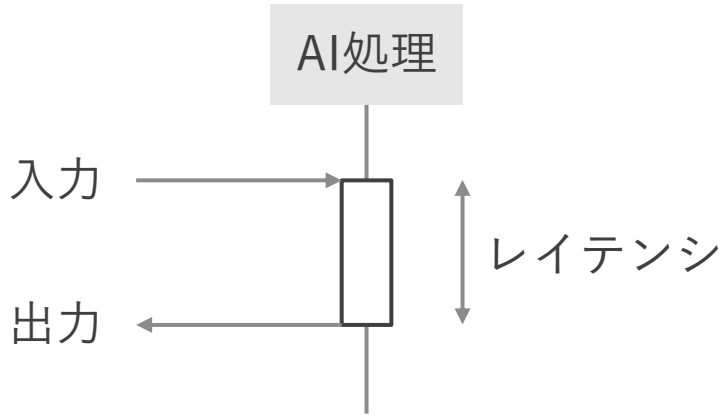
レイテンシと
スループットを
考える

観点②　レイテンシとスループットを考える

ARAYA

レイテンシ

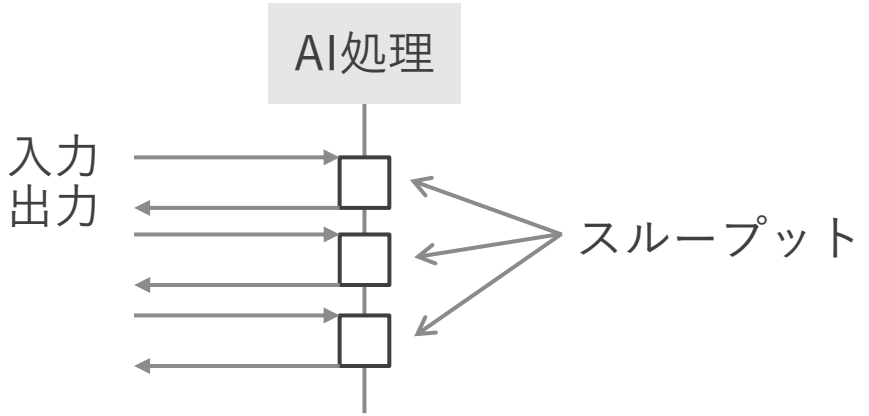
- 入力を入れてから結果が返ってくるまでの時間。



- レイテンシは、HWのスケールアップや、モデルの小型化によって短くすることが可能。

スループット

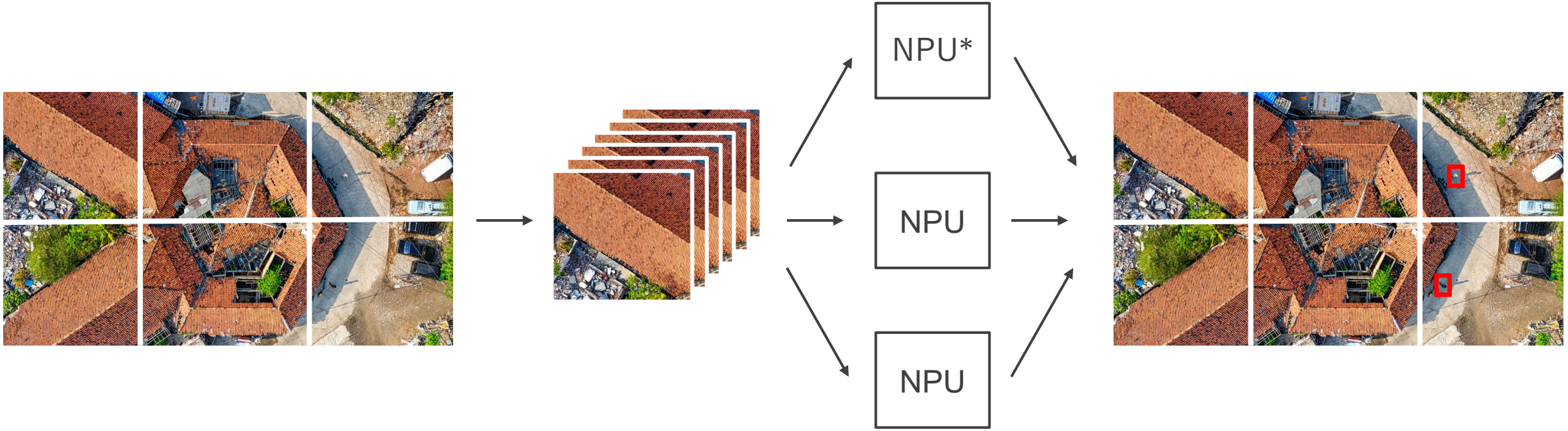
- ある一定の時間内に処理可能な入力の量。FPSなどの指標で測られることが多い。



- スループットは、HWのスケールアップとスケールアウト(並列処理)、モデルの小型化、バッチサイズ最適化によって向上させることが可能。

スループットが大事になるユースケース

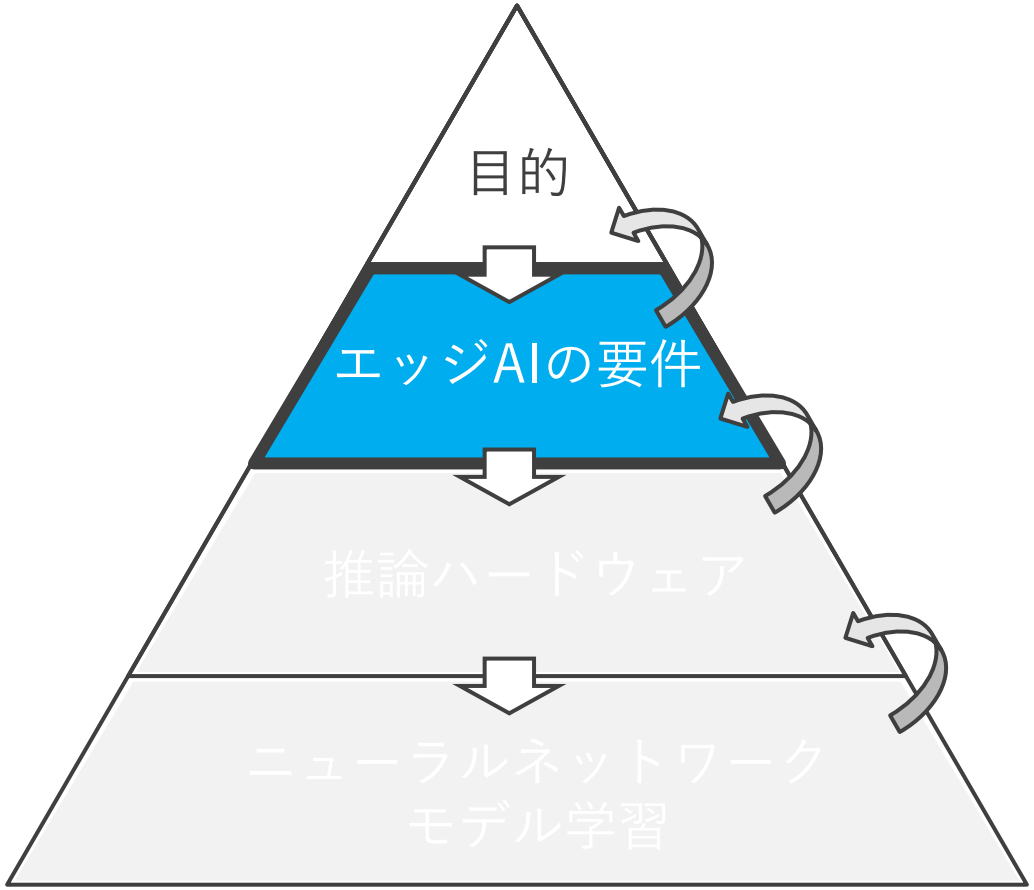
- 例えばFull HDまたは4K画像に対する物体検知を考える。
- 大きい画像をタイル状に分割し、各タイルを別々のAI演算機に割り当てて計算させることで、スループットを向上させ、結果として、1枚の画像を処理する時間(レイテンシ)も短くすることができる。



NPU: Neural Processing Unit

NPU並列使用によるスループットの向上

- 例えば、PC(Raspberry Pi等)に複数のNeural Compute Stickを接続し、並列で演算できるようにすることで、スループットを向上させることができる。
- 2本指すことで、ほぼx2のスループットを得られる。一方、３～4本と本数を増やしていくと、並列化のオーバーヘッドが大きくなるため、単純にx3、x4倍の速度は得られなくなっていく。



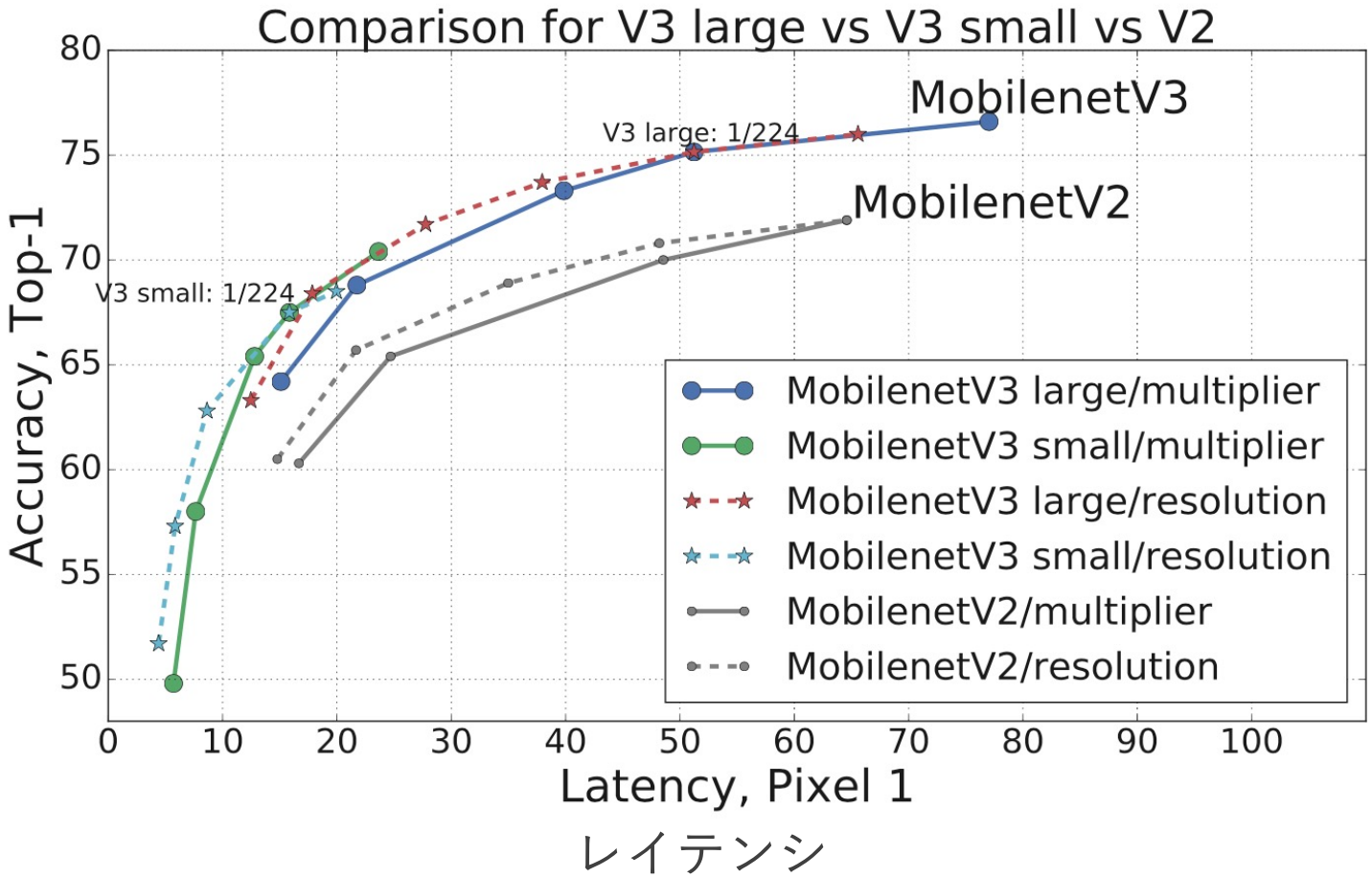
観点③

精度と速度の
トレードオフを
考える

観点③　精度と速度のトレードオフを考える

- 一般的に、モデルを小型化することで、推論の速度は速くなるが、精度は劣化する。

精度

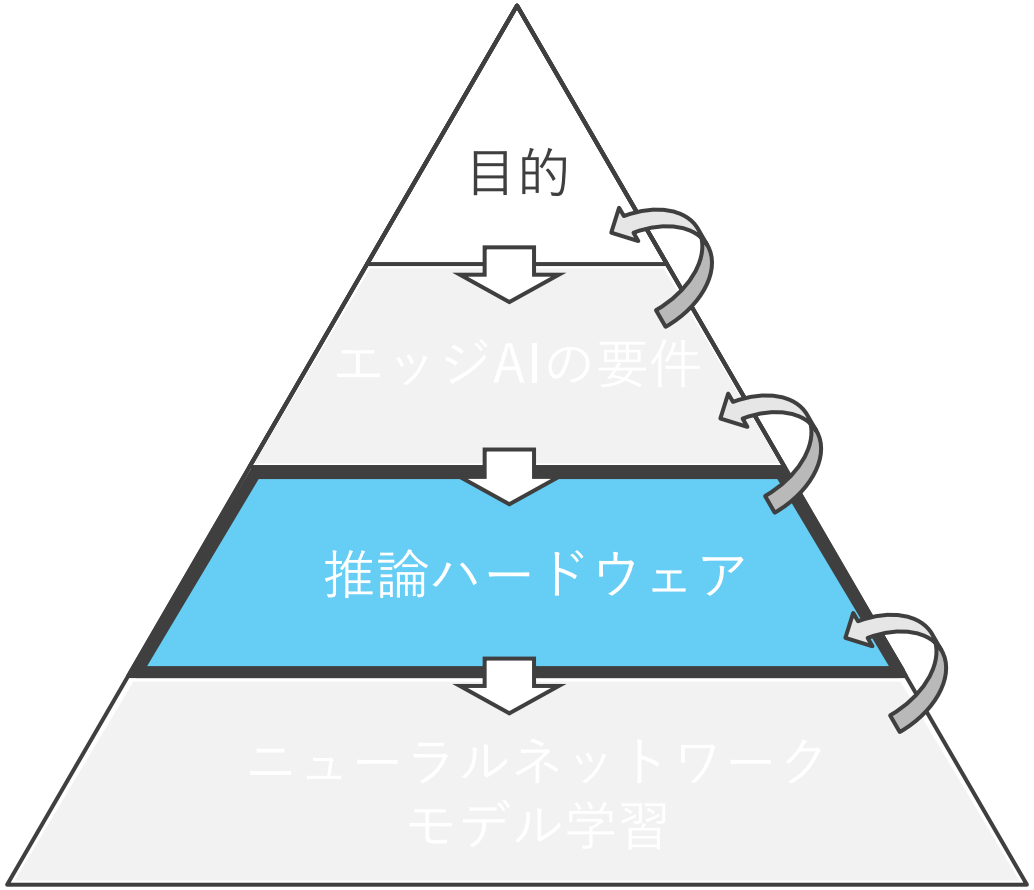


レイテンシ

A. Howard et al., "Searching for MobileNetV3," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 1314-1324, doi: 10.1109/ICCV.2019.00140.

観点③　精度と速度のトレードオフを考える

- そのため、
  - どの程度の精度劣化であれば許容可能か？
  - どの程度の速度が求められるか？
- を考慮してモデルの小型化を検討する必要がある。



観点④

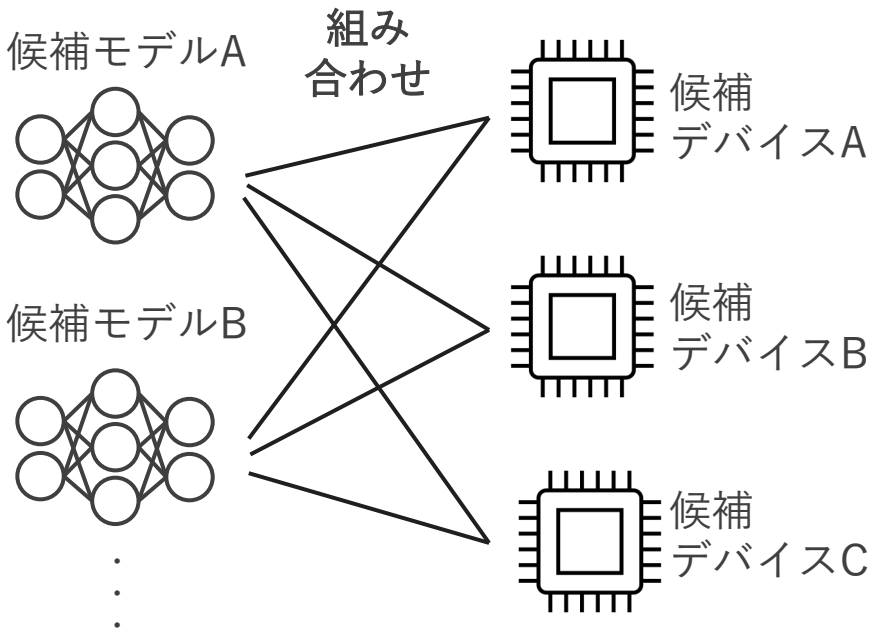
エッジデバイスの
ベンチマークをとる

観点④　エッジデバイスのベンチマークをとる

- 様々なデバイスとモデルの組み合わせがあり、これらの中から最適なものを選択する必要があるが、様々な課題がある。

様々なモデル

- パラメータが少なく推論効率のよいものや、パラメータが多く精度は高いが効率は低いものなど色々存在する。
- また１つのモデルでも、パラメータを変えることで複数のモデルとできるものもある。(MobileNet、EfficientNet)



適切なデバイスの選定

- 最適なデバイスを見出すためには、まず複数のデバイスを用意する必要があり、ものによってはセットアップに時間を要するものもある。
- また、デバイスごとに制約や高速化のノウハウが存在し、これを考慮しながら実装するのは労力を要する。

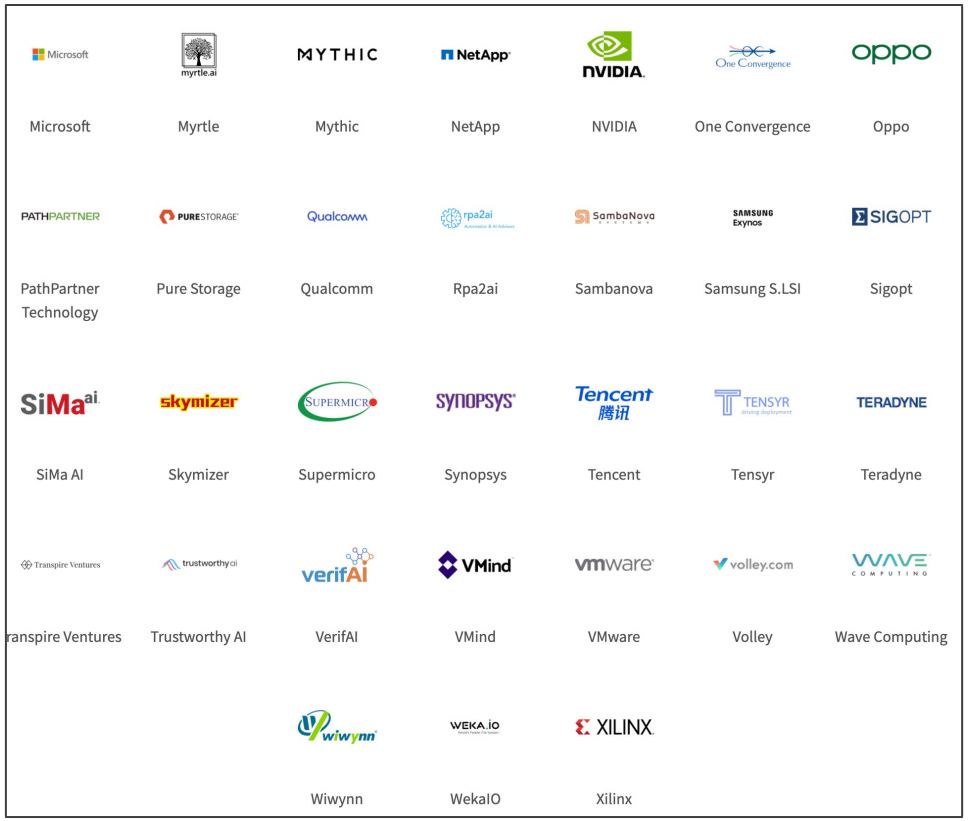
膨大な組み合わせの数

- モデルとデバイスの種類が多いため、取り得る組み合わせの数が膨大となり、最適な組み合わせを見つけだすのに労力がかかる。

公開されているベンチマークの利用：MLPerf

- まずは、公開されているベンチマークの利用を検討する。
- 例えば、デバイスベンダーが協力して、MLPerfと呼ばれるベンチマークを公開している。

Supported Companies



https://mlperf.org/

公開されているベンチマークの利用：測定結果

- 様々なProcessor、Acceleratorを使ったベンチマーク結果が掲載されている。
- ターゲットタスクとして、画像分類以外に、物体検出、音声認識、自然言語処理もある。

ターゲットタスク

測定条件(HW/SWスペック)

								Results										
								Task	Image classification			Object detection (small)			Object detection (large)			Medical ima
								Data	ImageNet			COCO			COCO			BraTS 2019
								Model	ResNet			SSD-small			SSD-Large			3D-UNet
								Accuracy (%FP32 re			99.00			99.00			99.00	
								Scenario	Single Stream	Multiple Stream	Offline	Single Stream	Multiple Stream	Offline	Single Stream	Multiple Stream	Offline	Single Stream
ID	Submitter	System	Processor	#	Accelerator	#	Software	Units	latency in ms	streams	samples/s	latency in ms	streams	samples/s	latency in ms	streams	samples/s	latency in ms
CATEGORY: Available																		
0.7-130	Atos	BullSequana Edge server (2 x T4)	Intel(R) Xeon(R) D-2187NT CPU @ 2.00GHz	1	NVIDIA T4	2	CUDA 11.0		1.26		11,762				6.84		266	
0.7-131	CentaurTechnology	Centaur Technology Reference Design v1.0D	Centaur Integrated x86 CPUs	1	Centaur Integrated AI Coprocessor	1	TensorFlow-Lite 2.2.0 + custom patch		1.05		1,254	1.33		1,938				
0.7-132	dividiti	Firefly-RK3399 (firefly)	Arm Cortex-A72 MP2 (big); Arm Cortex-A53 MP4 (LITTLE)	1	Arm Mali-T860 MP4	1	ArmNN v20.08 (OpenCL)		458.25		2							
0.7-133	dividiti	NVIDIA Jetson AGX Xavier	NVIDIA Carmel (ARMv8.2)	1	NVIDIA AGX Xavier	1	TensorRT v6.0		2.29	70	1,301	2.25		1,244				
0.7-134	dividiti	Firefly-RK3399 (firefly)	Arm Cortex-A72 MP2 (big); Arm Cortex-A53 MP4 (LITTLE)	1			ArmNN v20.08 (Neon)		367.53		3							
0.7-135	dividiti	Firefly-RK3399 (firefly)	Arm Cortex-A72 MP2 (big); Arm Cortex-A53 MP4 (LITTLE)	1			TFLite v2.2.0 (ruy)		544.73		2							
0.7-136	dividiti	Raspberry Pi 4 (rpi4)	Arm Cortex-A72 MP4	1			ArmNN v20.08 (Neon)		418.83		2							
0.7-137	dividiti	Raspberry Pi 4 (rpi4)	Arm Cortex-A72 MP4	1			TFLite v2.2.0 (ruy)		759.54		1							
0.7-138	dividiti	Raspberry Pi 4 with Coral (rpi4coral)	Arm Cortex-A72 MP4	1			ArmNN v20.08 (Neon)		363.62		3							
0.7-139	dividiti	Raspberry Pi 4 with Coral (rpi4coral)	Arm Cortex-A72 MP4	1			TFLite v2.2.0 (ruy)		612.46		2	170.99		6				
0.7-140	dividiti	NVIDIA Jetson AGX Xavier	NVIDIA Carmel (ARMv8.2)	1			ArmNN v20.08 (Neon)		73.28		17							
0.7-141	dividiti	NVIDIA Jetson AGX Xavier	NVIDIA Carmel (ARMv8.2)	1			TFLite v2.2.0 (ruy)		86.82		13							
0.7-142	dividiti	NVIDIA Jetson AGX Xavier	NVIDIA Carmel (ARMv8.2)	1			TFLite v2.3.0 (ruy)		119.35		9							
0.7-143	Inspur	NF5488A5	AMD EPYC 7742	2	NVIDIA A100-SXM4	1	TensorRT 7.2, CUDA V11.0.221								1.77		993	
0.7-144	Inspur	NF5468M5	Intel(R) Xeon(R) Platinum 8268 CPU @ 2.90GHz	2	NVIDIA T4	1	TensorRT 7.2, CUDA V11.0.221		0.88		6,205				8.22		140	
0.7-145	Lenovo	Lenovo ThinkSystem SE350 Edge Server (1x T4)	Intel(R) Xeon(R) D-2123IT CPU @ 2.20GHz	1	NVIDIA T4	1	TensorRT 7.2, CUDA 11.0 Update 1		0.88	272	6,084	0.52	368	7,706	8.17	8	134	156.18
0.7-146	NVIDIA	Gigabyte G482-Z52 (1x A100-PCIe, TensorRT)	AMD EPYC 7742	2	NVIDIA A100-PCIe	1	TensorRT 7.2, CUDA 11.0 Update 1		0.52	1,344	32,008	0.35	1,600	44,371	1.99	48	852	35.28
0.7-147	NVIDIA	Gigabyte G482-Z52 (1x A100-PCIe, TensorRT, Triton)	AMD EPYC 7742	2	NVIDIA A100-PCIe	1	TensorRT 7.2, CUDA 11.0 Update 1		0.56	1,344	33,423	0.37	1,600	44,134	2.04	48	852	44.94
0.7-148	NVIDIA	NVIDIA DGX-A100 (1x A100-SXM4, TensorRT)	AMD EPYC 7742	2	NVIDIA A100-SXM4	1	TensorRT 7.2, CUDA 11.0 Update 1		0.51	1,760	37,480	0.31	2,368	51,205	1.76	60	981	27.42
0.7-149	NVIDIA	NVIDIA DGX-A100 (1x A100-SXM4, TensorRT, Triton)	AMD EPYC 7742	2	NVIDIA A100-SXM4	1	TensorRT 7.2, CUDA 11.0 Update 1		0.53	1,440	37,558	0.33	1,920	50,796	1.95	56	979	37.21
0.7-150	NVIDIA	NVIDIA DGX-A100 (1x A100-SXM4-MIG-1x1g.5gb, TensorRT)	AMD EPYC 7742	2	NVIDIA A100-SXM4 (1x1g.5gb MIG)	1	TensorRT 7.2, CUDA 11.0 Update 1		0.74		4,966	0.49		6,939	8.34		134	192.92
0.7-151	NVIDIA	NVIDIA DGX-A100 (1x A100-SXM4-MIG-1x1g.5gb, TensorRT, Triton)	AMD EPYC 7742	2	NVIDIA A100-SXM4 (1x1g.5gb MIG)	1	TensorRT 7.2, CUDA 11.0 Update 1		0.76		4,907	0.53		6,842	8.51		135	203.00
0.7-152	NVIDIA	NVIDIA Jetson AGX Xavier 32GB (TensorRT)	NVIDIA Carmel (ARMv8.2)	1	NVIDIA AGX Xavier	1	20.09 Jetson CUDA-X AI Developer F		2.14	96	2,075	1.25	107	2,533	28.53	2	51	439.27
0.7-153	NVIDIA	Supermicro 4029GP-TRT-OTO-28 (1x T4, TensorRT)	Intel(R) Xeon(R) Platinum 8280 CPU @ 2.70GHz	2	NVIDIA T4	1	TensorRT 7.2, CUDA 11.0 Update 1		1.09	272	6,036	0.55	368	7,960	8.42	8	135	157.76
0.7-154	NVIDIA	Supermicro 4029GP-TRT-OTO-28 (1x T4, TensorRT, Triton)	Intel(R) Xeon(R) Platinum 8280 CPU @ 2.70GHz	2	NVIDIA T4	1	TensorRT 7.2, CUDA 11.0 Update 1		1.19	264	6,010	0.61	360	7,963	8.49	8	135	173.68
0.7-155	NVIDIA	NVIDIA Jetson Xavier NX (TensorRT)	NVIDIA Carmel (ARMv8.2)	1	NVIDIA Xavier NX	1	20.09 Jetson CUDA-X AI Developer F		3.21	50	1,032	1.67	60	1,267	46.91	1	27	771.46
CATEGORY: Research, Development, or Internal																		
0.7-156	IVATech	iva-fpga-1	Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz	1	IVA_FPGA_1	1	Tensorflow 1.15.0		12.23		89							
0.7-157	Mobilint	Mobilint Edge	Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz	1	Xilinx Alveo U250	1	Internal Framework		37.46		107	10.64		410				

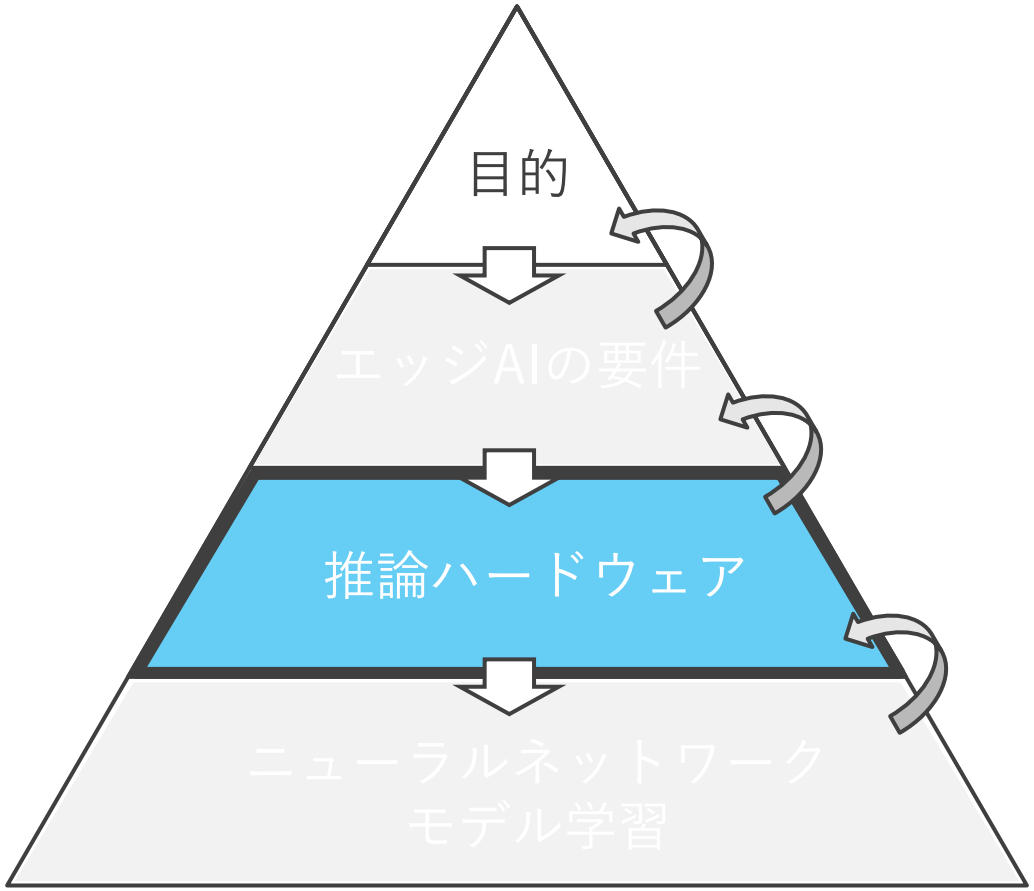
公開されているベンチマークの利用：注意点

ARAYA

- すべてのデバイスで測定されているわけではない。
  - 例えば2020/10/21にリリースされた、最新のMLPerf Inference v0.7 Resultsを見ると、Coral(Edge TPU)は載っているが、Myriad(Intel)は載っておらず、比較ができない。

- 基本的に精度の低下を考慮したモデルでは試されていない。
  - 元の精度の99%を担保したモデルのみ結果を登録できる仕組み。
  - なので、先ほどの精度・速度トレードオフは見れない。

- なお、アラヤでは上記の観点も考慮したベンチマークを自社で作成しております。詳細につきましては、最後に説明するお問い合わせ窓口よりお問い合わせ下さい。



観点⑤

エッジデバイスに
載るモデルと
載らないモデル

観点⑤　エッジデバイスに載るモデルと載らないモデル

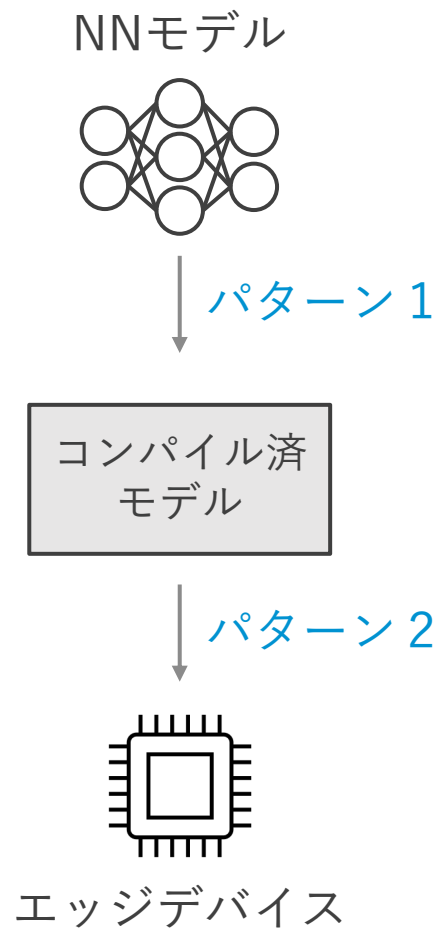
- 学習したモデルをエッジデバイスに載せようとする場合、そのデバイスでサポートしているオペレーション(あるいはレイヤ)の種類を考慮する必要がある。

パターン１：エッジデバイス向けにモデルをコンパイルできない。

- エッジデバイス向けコンパイラを使って変換できないネットワークのアーキテクチャ、あるいはオペレーション／レイヤも存在する。特にカスタムレイヤ。
- そのようなレイヤがある場合は、他のレイヤを組み合わせて代替する、あるいはカスタムレイヤを別途定義できるものについては、定義する必要がある。

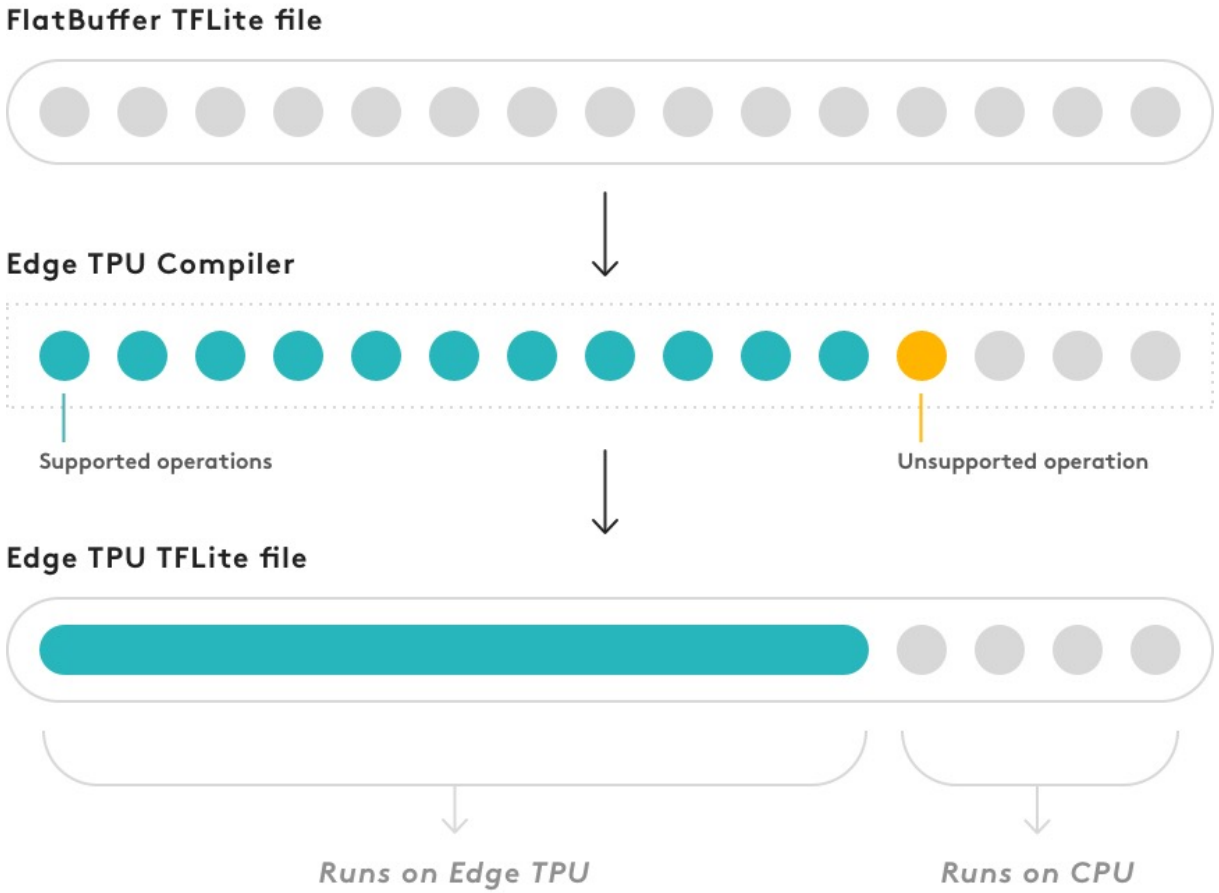
パターン２：コンパイルできても、デバイスで高速化されない。

- コンパイルできたとしても、高速に動作しない場合がある。これは、HWによっては、高速化をサポートしていないオペレーション／レイヤがあるためである。

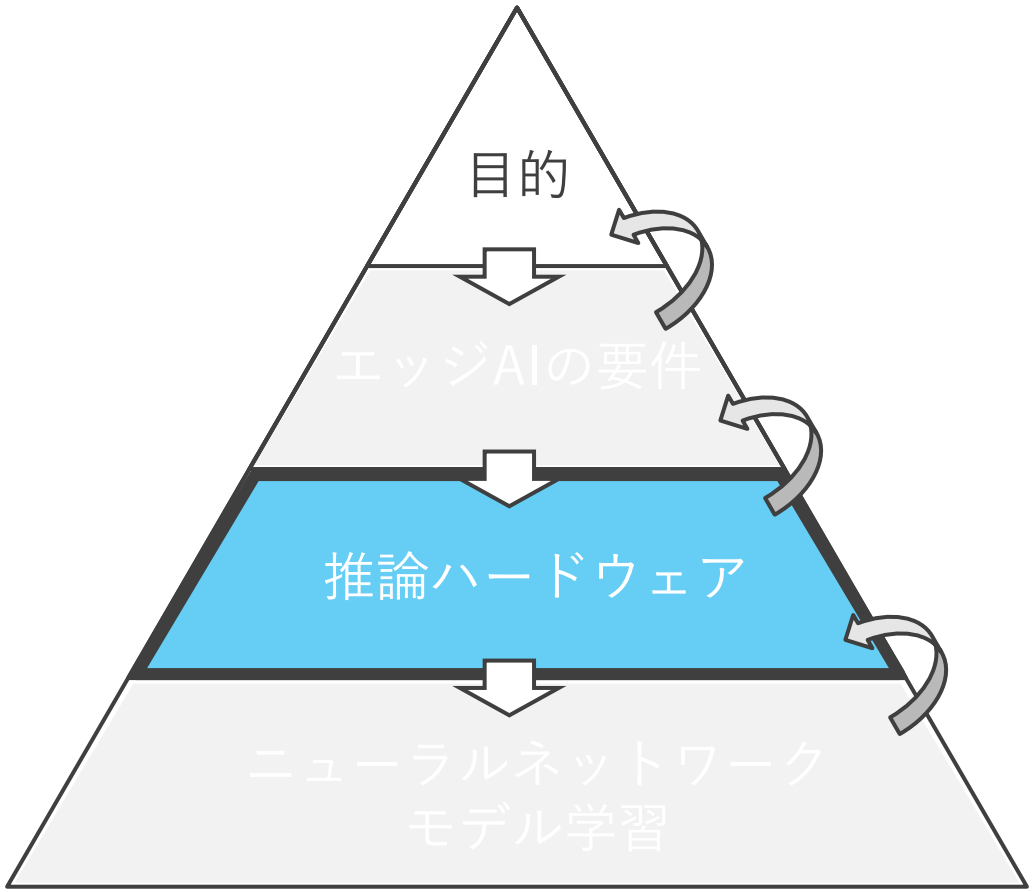


パターン２の例：Coralのケース

- 例えばCoralでは、ニューラルネットワークのモデルの途中にサポートされていないオペレーションが存在した場合は、それ以降のオペレーションはCPUで実行される。



https://coral.ai/docs/edgetpu/models-intro/#compiling



観点⑥

前処理・後処理を
見直す

観点⑥　前処理・後処理を見直す

ARAYA

- 推論のレイテンシ短縮を考えるとき、よくモデルの実行時間のことを考えがちだが、実際には前後の処理に時間がかかっている場合も多い。
- そのような場合には、いくらモデルの実行時間を短縮しても、全体の時間が短縮されないため、注意が必要である。

弊社の事例

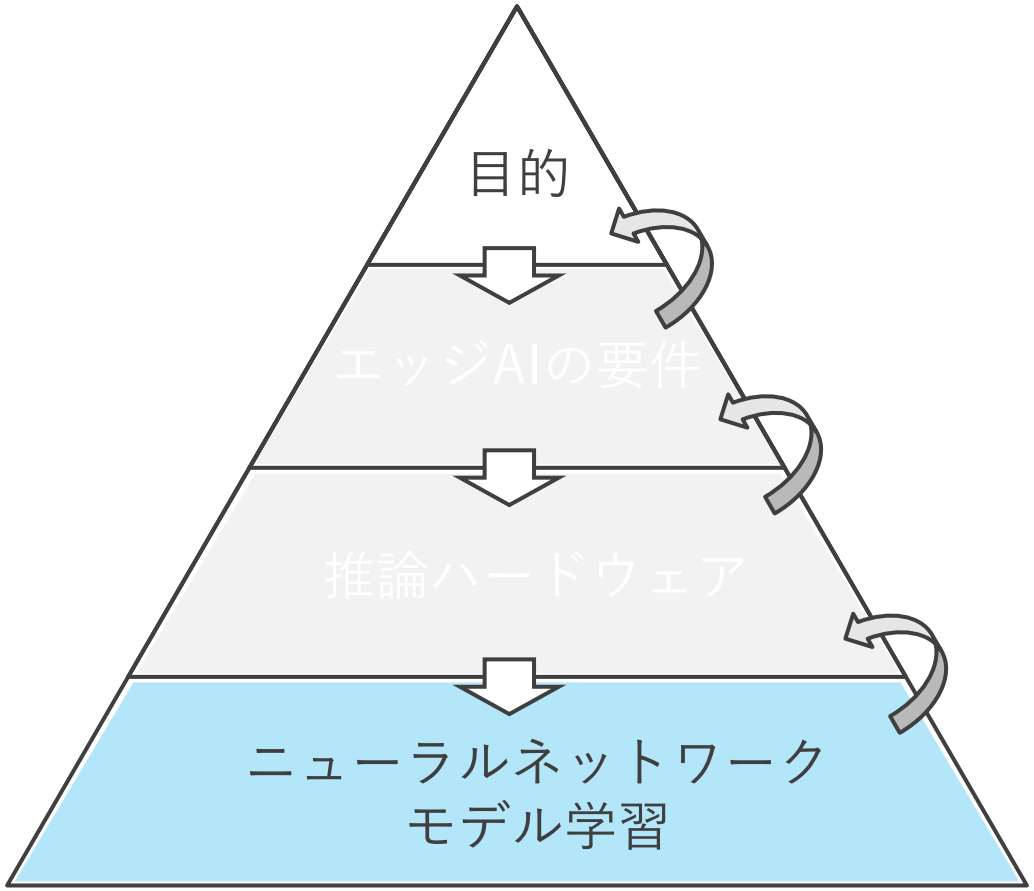
- ドローン撮影画像(Full HD)をタイル分割して推論する際、肝心のニューラルネットワークの推論処理よりも、画像をタイル分割し、結果をマージする処理がボトルネックとなっていることが分かった。
- まずはプロファイルを取得することで、処理のボトルネックを把握することが重要。

最適化前

推論の各パート	レイテンシ
SSDLite (without NMS)	36 ms x 32
分割画像のBoxのFullHD画像へのマージと整列	**668 ms**
NMS (Non-Maximum Suppression)	3 ms
合計	1823 ms

最適化後

推論の各パート	レイテンシ
SSDLite (without NMS)	7.8 ms x 32
分割画像のBoxのFullHD画像へのマージと整列	**76.4 ms**
NMS (Non-Maximum Suppression)	2.5 ms
合計	352.2 ms



観点⑦

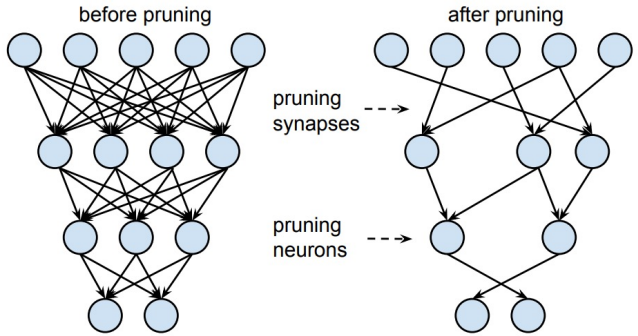
ニューラルネットの小型化を検討する

観点⑦　ニューラルネットの小型化を検討する

- 一般的に、ニューラルネットワークを小型化することで、速度を向上させることができるが、うまく小型化をすれば、精度劣化を小さく抑えることができる。
- そのための手法として、例えば、枝刈りや蒸留、量子化といった方法がある。

枝刈り(Pruning)

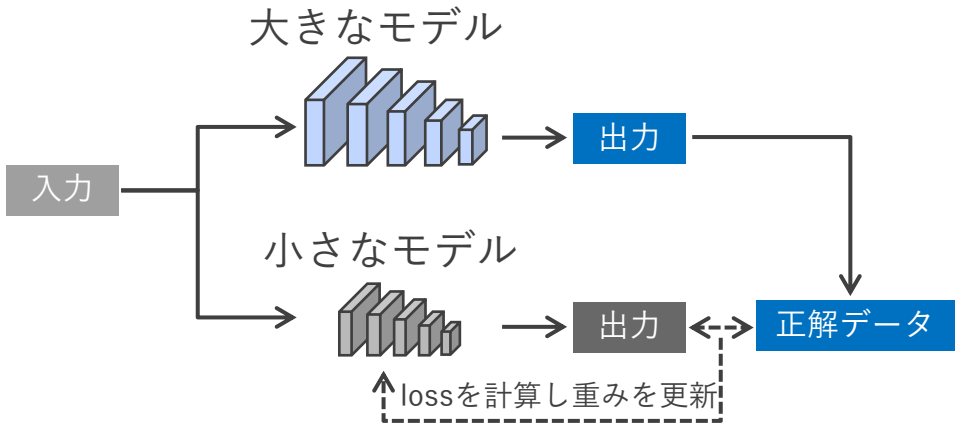
- 重要度の低い重みの値をゼロとすることで、ネットワークを疎にする手法。



https://papers.nips.cc/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf

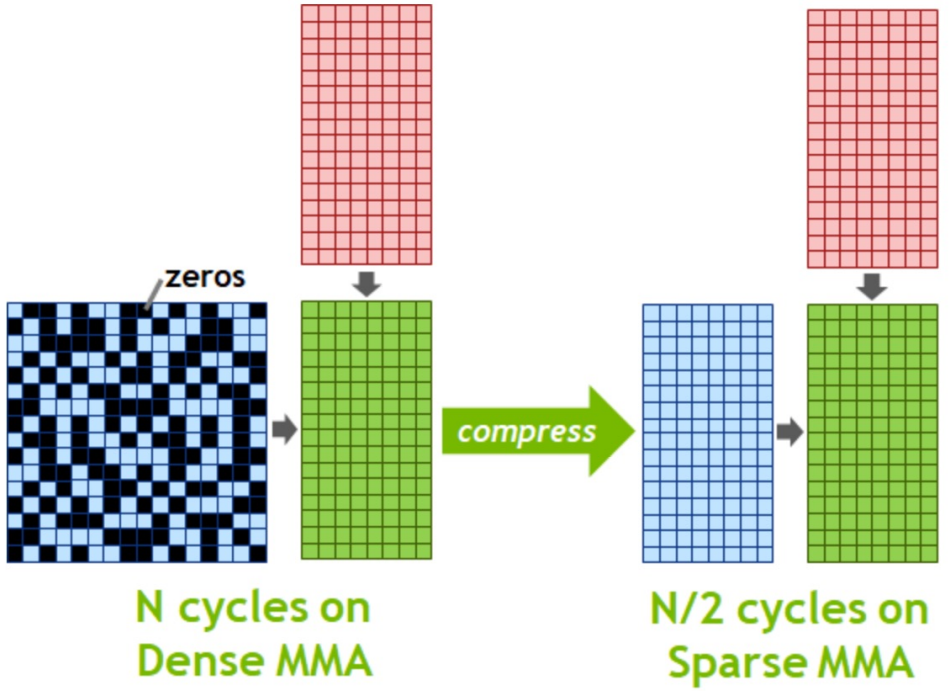
蒸留(distillation)

- 大きなモデルの入出力を教師データとして小さなモデルを学習することで、小さいモデルを一から学習するのと比較して、高い精度を実現する手法。



NVIDIA A100で採用された 2-out-of-4 pruning

- エッジではないが、NVIDIAの最新GPU、A100では、2-out-of-4 pruningをサポート。
- これにより、最大x2の推論速度を実現。



Example Dense MMA and Sparse MMA operations using 16x16 sparse matrix (Matrix A), multiplied by a dense 16x8 matrix (Matrix B). Sparse MMA operation on right doubles throughput by skipping compute of zero values

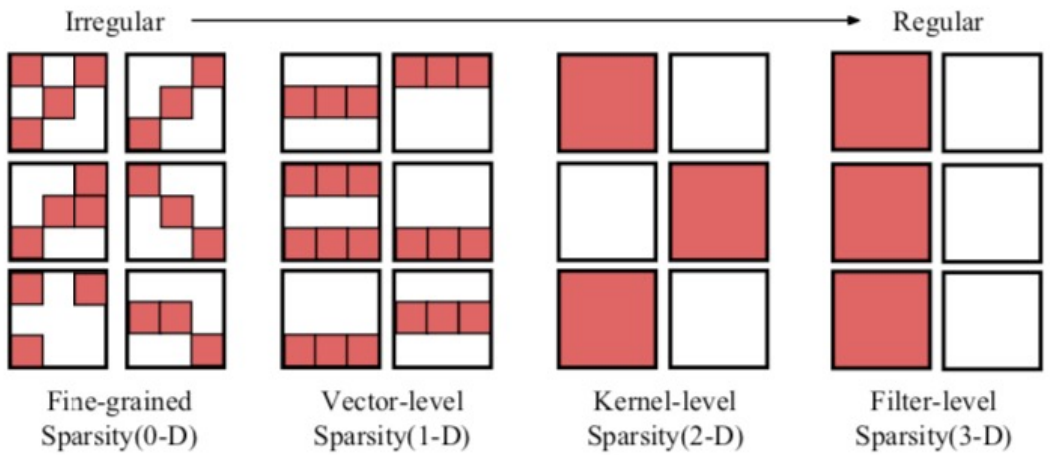
Figure 13. Example Dense MMA and Sparse MMA operations

https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/nvidia-ampere-architecture-whitepaper.pdf

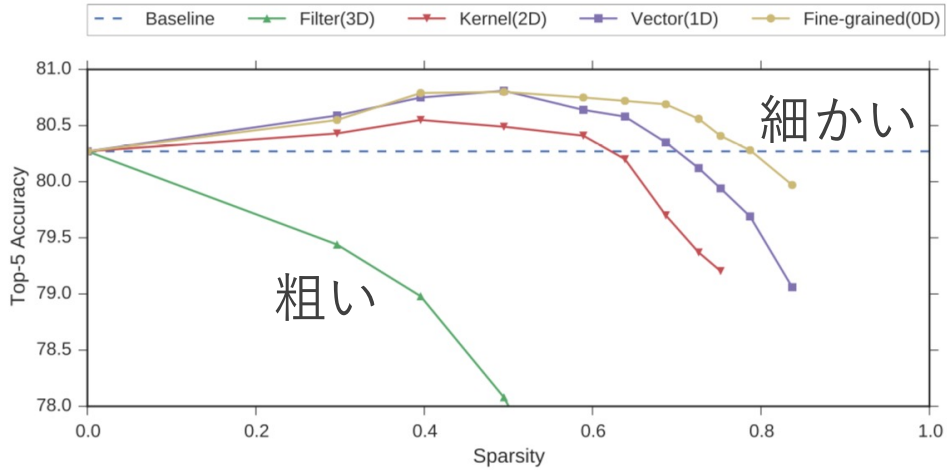
枝刈りの粒度と精度のトレードオフ

- 枝刈りに用いるマスクのスパース度のパターンの粗さ　・　細かさ(粒度)とモデルの性能にもトレードオフ関係があり、粒度が粗ければ粗いほどモデルの性能は下がりやすいという傾向がある。
- 一方、粒度が粗い方がCPU/GPUでの高速化が期待できることが知られている。

細かい　　粗い



精度



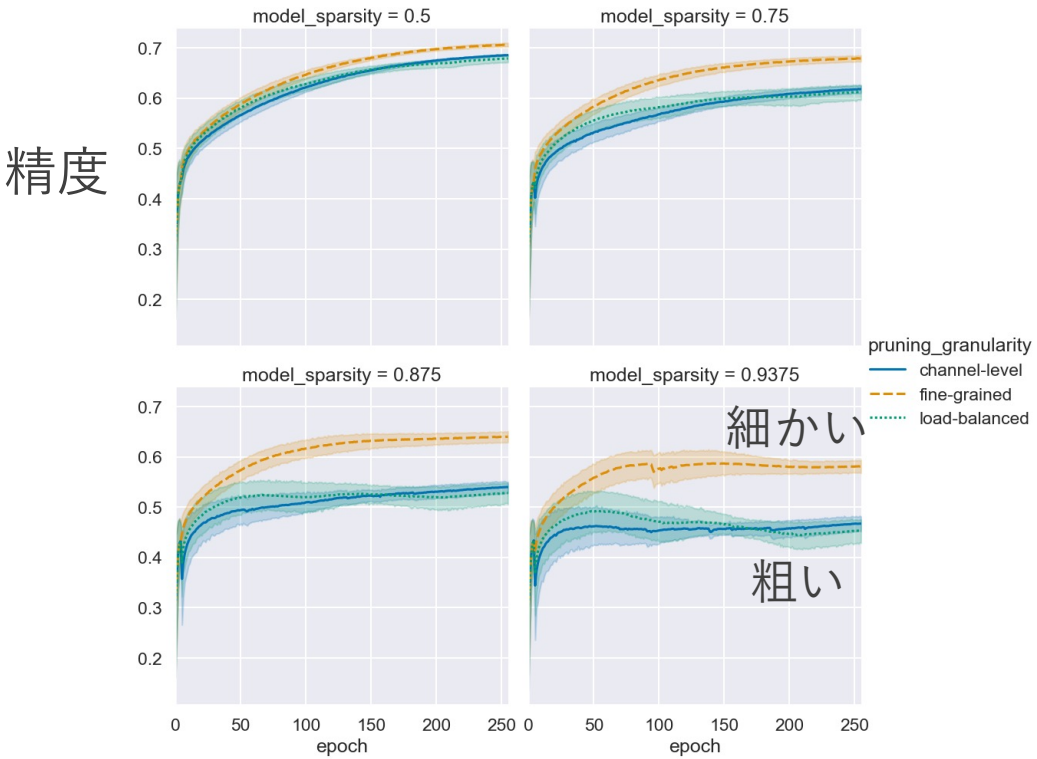
圧縮率(スパース度)

Figure 1, 3 from "Exploring the Granularity of Sparsity in Convolutional Neural Networks" by Song Han, William Dally et al.

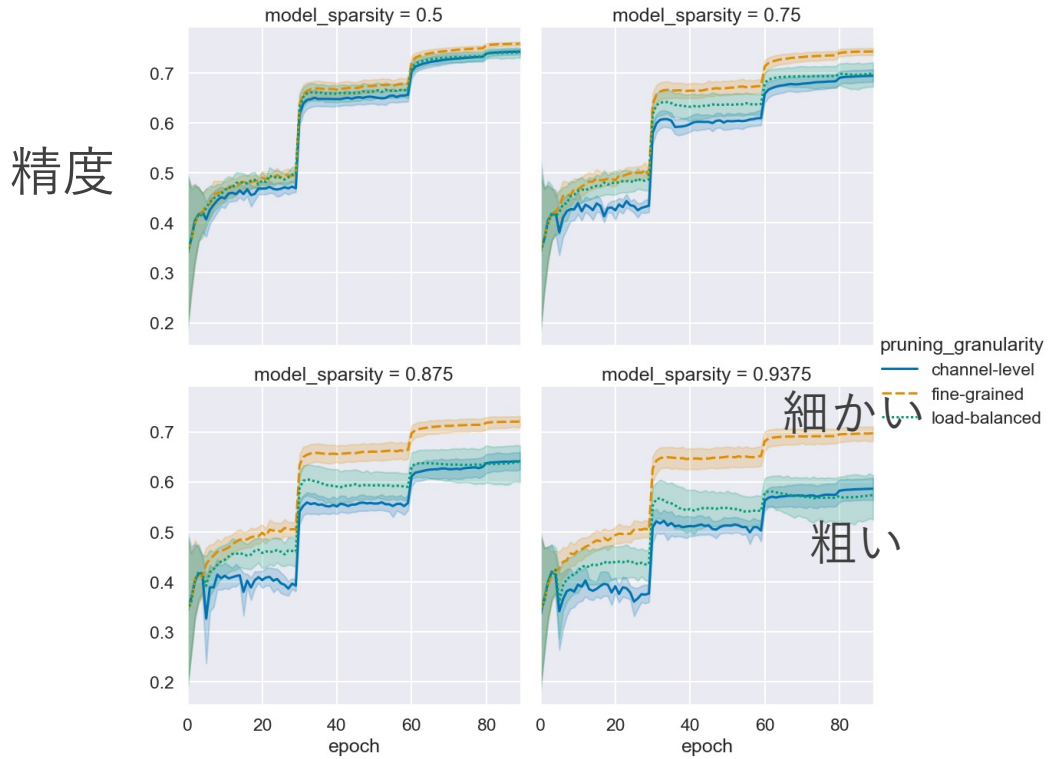
枝刈りの粒度と精度のトレードオフ

- アラヤでは、ImageNet(1,281,167枚の画像の1000クラス分類)での枝刈り実験を実施した。
- MobileNet V1、ResNet-50の各モデルで、計96試行の学習を伴う枝刈りを行い、粒度とスパース度と精度のトレードオフ関係を炙り出すことができた。

MobileNet V1の学習曲線



ResNet-50の学習曲線



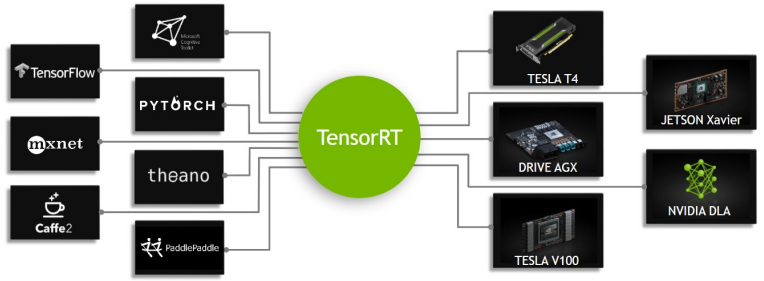
この実験はABCI(産総研のAIプラットフォーム)を使って行われました。

量子化による小型化

- 量子化による小型化は、既に多くのデバイス・ツールで一般的となっている。

IntelのNNコンパイルツールであるOpenVINOでは、8bitまでの量子化をサポート。

Google Coralでは、8bit integerでの量子化による高速演算を提供。



NVIDIAが提供するNN推論最適化ツールのTensorRTで、8bit integerでの量子化をサポート。

まとめ

- エッジAIの開発プロセス：目的→要件→デバイス選定→モデル学習

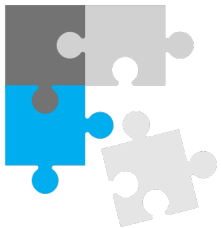
- エッジAIの開発で考慮すべき７つの観点

要件
① 本当にエッジAIが必要か？
② レイテンシとスループットを考える
③ 精度と速度のトレードオフを考える

デバイス選定
④ エッジデバイスのベンチマークをとる
⑤ エッジデバイスに載るモデルと載らないモデル
⑥ 前処理・後処理を見直す

モデル学習
⑦ ニューラルネットの小型化を検討する

アラヤのエッジAIコンサルティングサービス



エッジのことをすべてお任せしたいお客様むけ

1.トータルソリューションサービス

- 弊社の知見をもとに、有効と思われる圧縮手法や最適化、デバイスを試すことで、高速なイテレーションを回し、お客様の目的を達成可能なエッジAIを開発します。
- また、必要に応じて要件定義から一緒に入って実施させて頂くことも可能です。
- すべてを弊社でお引き受け致します。

イテレーションの一部をお任せしたいお客様むけ

2.個別コンサルティングサービス

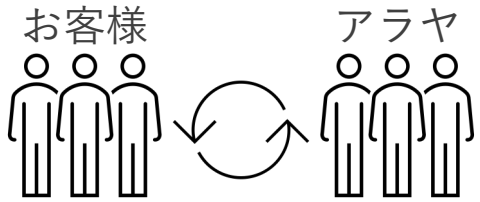
- ネットワーク圧縮の適用や、特定デバイスへの実装について、弊社がサポート致します。
- 実装の各フェーズに対応する形で、以下の3つのサービスをご提供致します。

2-1. 要件定義支援サービス
2-2. エッジ実装・最適化サービス
2-3. モデル小型化サービス

1. トータルソリューションサービス

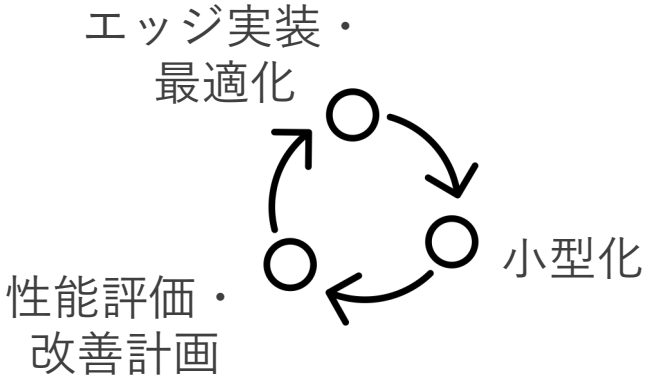
- お客様と弊社でOne teamとなり、要件定義からモデルの小型化、エッジデバイスへの実装まで、一気通貫でご支援させて頂きます。
- 期間は2ヶ月間～となりますが、一般的には以下のようなスケジュールを想定しています。

要件定義フェーズ（約2週間）



2週間集中して議論し、要件をFixする。

開発フェーズ（約1.5ヶ月～）



- 実装と評価のイテレーションをなるべく早く回すようにPJを進めていく。
- 1イテレーションが2～3週間として、最低2回程度、イテレーションを回すことができると想定。

2-1. 要件定義支援サービス

お客様の課題

- モデルを学習させたことはあるが、エッジへの実装は初めてのため、実装にあたってどういった点を考慮すべきかが分からない。
- エッジAIの要件を作成したが、網羅的に検討できているかどうか不安。

ご提供サービス

- お客様の目的に対し、複数の観点から要件を定義する作業の支援をさせて頂きます。
- 実際に実施する作業や観点としましては、例えば以下のようなものがございます。

現状のプロファイリング

- 現状の推論処理のプロファイリングを行います。
- 特に、前処理、推論、後処理にかかっている時間、メモリフットプリント、使っているレイヤの種類は、今後の高速化やコンパイルの際に重要となる要素です。

求められるFPS/レイテンシの定義

- リアルタイム性を求められるアプリケーションではレイテンシが最も重要となります。またFPSも重要な指標の一つです。
- アプリケーションに求められるレイテンシ/FPSの整理と定義を支援します。

求められる精度性能

- ネットワーク圧縮の過程では、速度向上の代償として精度が低下する場合があります。
- その可能性に対し、どの程度の精度低下までであれば耐えうるかの検証と整理を支援します。

モデルサイズとアップデート

- モデルは一度作れば終わりではなく、データ分布のシフトが発生した場合はアップデートが必要になります。
- このような場合に備え、アップデートの方法や、モデルの重みサイズに求められる要件を整理します。

エッジ vs クラウドの役割検討

- 必ずしも全ての処理をエッジで行う必要はなく、クラウドとの分担も考慮すべきです。
- エッジ・クラウドのメリット・デメリットを考慮し、どの処理をエッジで行い、どの処理をクラウドで行うべきかの整理と定義を支援します。

2-2. エッジ実装・最適化サービス

お客様の課題

- 適切なエッジデバイスが分からない。
- 選定の際、複数のデバイスを用意する必要がある。
- デバイスごとに制約や高速化のノウハウが存在し、これを考慮しながら実装するのに労力を要する。

ご提供サービス

- 深層学習モデルを、搭載したいエッジ環境に適したモデルに変換し、実際のデバイスを用いて速度を評価します。以下がその流れとなります。

①モデル用意	お客様からモデルファイル、また必要に応じてモデル作成コードをいただく。 （弊社でモデルを用意する場合）：要求精度・速度を満たす見込みのあるモデルを用意する。必要に応じ、ハードウェアに合わせてlayerの変更を行う
②変換・最適化	候補となるエッジデバイス用にモデルの変換・最適化を行う。 前処理・後処理の高速化も合わせて高速化する。また、複数デバイスの利用・並列化処理を行う。デバイスは弊社で用意する。
③評価	推論速度を計測し、要求推論速度を満たすかどうか評価する。 必要に応じてモデル小型化サービスをご提案させて頂く。

2-2. エッジ実装・最適化サービス

弊社では、以下のデバイスをご用意しております。

#	種別	デバイス	メーカー	スペック
1	NPU (Neural Processing Unit)	Myriad X	Intel	4 TOPS。1.5W。USBアクセラレータタイプのものを使用。コネクタはUSB 3.0 Type-A。OpenVINOを用いて最適化・コンパイルを行う。
2		Edge TPU	Google	4 TOPS (int8); 2 TOPS per watt、USBアクセラレータタイプのものを使用。コネクタはUSB 3.0 Type-C* (data/power)。
3	GPU	Jetson nano	NVIDIA	GPUは、128 Core Maxwell 0.5 TFLOPS (FP16)搭載。CPUは、Quad-core ARM A57 (1.5 GHz)搭載。
4		Jetson TX2	NVIDIA	GPUは、256 Core Pascal 1.33 TFLOPS (FP16)搭載。CPUは、Dual-core Denver and Quad-core A57 2GHz (2x) 2MB L2搭載。
5		Jetson Xavier NX	NVIDIA	GPUは、NVIDIA Volta architecture with 384 NVIDIA CUDAR cores and 48 Tensor coresで、21TOPS (Int8)。CPUは、6 コア NVIDIA Carmel 64 ビット ARMv8.2 @ 1400MHz* (6MB L2 + 4MB L3)。
6	CPU	Raspberry Pi 4	ラズベリーパイ財団	CPUは、Broadcom BCM2711， quad-core Cortex-A72 (ARM v8) 64-bit SoCを搭載。
7	Cloud	AWS lambda	AWS	AWSのサーバレスサービス。シングルコアCPUでの実行となる。
8	Smartphone	iPhone/iPad	Apple	Neural Engine搭載のA13 Bionicで、5 TOPSを実現。
9		Pixelシリーズ	Google	CPUとして、Snapdragon 765G（2.4GHz＋2.2GHz＋1.8GHz、オクタコア）を搭載。

※）こちらのリストは今後も拡充予定です。ご要望等ありましたらお気軽にお申し付け下さい。

2-3. モデル小型化サービス

お客様の課題

- モデルが大きすぎて目的のデバイスに載らない。
- 推論速度を上げることで前処理や後処理にもっとリソースを割きたい。
- 推論速度や消費電力に不満がある。

ご提供サービス

- モデル、入出力データ、学習コード等を拝見した上で、モデルを小さくするための方法についてご提案いたします。ご要望に応じて実装も承ります。
- 小型化の手法は様々ありますが、ここでは代表的な５つの手法を説明いたします。

#	名称	説明
1	モデルの枝刈り	重要度の低い重みの値をゼロとすることで、重みテンソルのスパース化を達成する、様々な手法が考案されている。スパース性のパターンの粒度に応じて細粒度/チャネル/ロードバランスプルーニングなどが存在し、それぞれ一長一短がある。
2	モデルの量子化	種々のテンソルのデータ型を変換する。学習を行いながら量子化する方法とそうでない方法がある。量子化後のデータ型はFP16、INT16、INT8となることが多い。
3	モデルの知識の蒸留	大きなモデルの入出力を教師データとして小さなモデルを学習することで、小さいモデルを一から学習するのと比較して、高い精度を実現する。
4	レイヤのフュージョン	複数のレイヤの処理を一つのレイヤにまとめられる場合がある（例えばBNレイヤ）。
5	アーキテクチャの変更	パラメタ数や演算回数を少なくするように設計されたモデルのアーキテクチャ（例えばMobileNetやEfficientNet）を選ぶことによって、モデルの小型化を図る。

実施フロー

トータルソリューション サービス・ 要件定義サービス

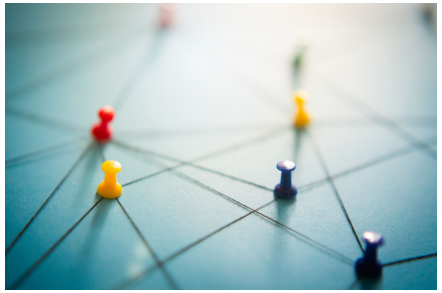
- まずはお客様からのご要望をお伺いした上で、期間内に可能なスコープを見積もり、ご提案させて頂きます。
- ご提案にご納得頂けましたら、その後は、お客様と弊社をOne teamとし、緊密に連携を取りながらPJを進めます。

エッジ実装・最適化 サービス

- まずはお客様からモデルの情報(データは不要です)と、ターゲットとするデバイスを指定頂いた上で、実施内容をご提案させて頂きます。
- 弊社からデバイスをご提案させて頂くことも可能です。

モデル小型化 サービス

- コンサル型：　基本的にはお客様主導で進めて頂き、適宜弊社からアドバイスをさせて頂きます。
- 受託型：　まずお客様からモデルの情報を頂き、期間内に可能なスコープを見積もり、ご提案させて頂きます。



各サービスに含まれる作業一覧

#	名称		複数回のイテレーション	要件定義	デバイス実装			モデル小型化		
					ご提案	モデル変換	コード最適化	ご提案	実装	学習
1	トータルソリューションサービス		✓	✓	✓	✓	✓	✓	✓	✓
2-1	個別コンサルティングサービス	要件定義支援サービス		✓						
2-2		エッジ実装・最適化サービス			✓	✓	✓			
2-3-1		モデル小型化サービス(コンサル型)						✓		
2-3-2		モデル小型化サービス(受託型)						✓	✓	✓

各サービスでお客様にご提供頂くもの・納品物・期間

#	名称		お客様にご提供頂くもの	弊社からの納品物	期間
1	トータルソリューションサービス		• 案件の概要説明(目的など) • エッジ搭載したいNNモデル • 学習用コード・データ(*1)	• 小型化・最適化済のNNモデルと推論コード • 報告書(最終的なモデルの精度・速度、作業プロセスを含む)	1ヶ月～
2-1	個別コンサルティングサービス	要件定義支援サービス	• 案件の概要説明(目的など) • エッジ搭載したいNNモデル	• 要件定義書	2週間~
2-2		エッジ実装・最適化サービス	• ターゲットデバイス(*2) • エッジ搭載したいNNモデル(*2)	• 最適化後のNNモデルと推論コード • 変換・最適化手順書 • 報告書(速度計測結果)	1週間～
2-3-1		モデル小型化サービス(コンサル型)	• 小型化の要件 • 小型化したいNNモデル	• 報告書(小型化に関する提案のまとめ)	2ヶ月～
2-3-2		モデル小型化サービス(受託型)	• 小型化の要件 • 小型化したいNNモデル • 学習用コード・データ(*1)	• 小型化後のNNモデルと推論コード • 報告書(最終的なモデルの精度、作業プロセスを含む)	1ヶ月～

(*1) 学習用データは実際に使われているものをご提供いただけるのがベストですが、難しい場合は、公開データセット等で代替頂いても問題ございません。
(*2) 弊社側からご提案させて頂くことも可能です。

アラヤの強み

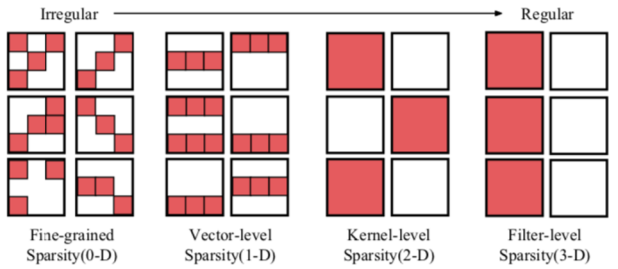
様々な受託案件で培ってきた豊富な実績と自社での研究開発力を活かし、お客様のAIエッジ実装をご支援いたします。

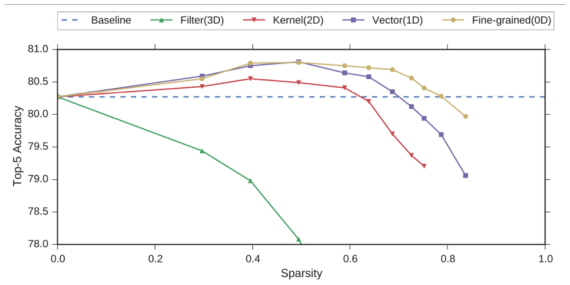
実績

- 特にドローンや監視カメラなどのエッジデバイスへの実装を行ってきました。
- 次頁以降にて、これまでの実績についてご紹介させて頂きます。

研究開発力

- 常に最新の小型化手法や、エッジデバイス・フレームワークについて動向を調査し、必要に応じて再現実験も行ってきました。





株式会社アラヤの会社概要

認知神経科学の研究者が設立した脳技術を併せ持ったAI開発企業

KANAI Ryota

金井 良太

株式会社アラヤ

代表取締役

Career

- 京都大学理学部卒業
- オランダ・ユトレヒト大学で実験心理学PhD取得
- 米国カリフォルニア工科大学にて、視覚経験と時間感覚の研究に従事
- 前英国サセックス大学准教授（認知神経科学）

会社概要

社名	株式会社アラヤ
設立	2013年12月
所在地	東京都千代田区神田佐久間町1-11 産報佐久間ビル6F
社員数	約70名
事業内容	AIプロダクト開発/ ニューロテックプロダクト開発事業

《お気軽にお問い合わせください》

ホームページ:https://www.araya.org/

E-mail:support@araya.org

Tel:03-6426-5144

アラヤでは大手製造/建設/物流業など
幅広い業界にソリューションを提供しています

AeroEdge
KDDI
KONOIKE GROUP
Signpost
SAMSUNG
住友商事マシネックス
ANA Inspiration of JAPAN
DAIKIN
iSiD
東北大学 TOHOKU UNIVERSITY
TOYOINKGROUP
TOYOTA
ひとの ときを、想う。JT
HITACHI Inspire the Next
HONDA

＊50音順に掲載

