Edge視点からの機械学習との付き合い方

2023年2月10日

中本幸一
兵庫県立大学/名古屋大学
yuki.nakamt@gmail.com

自己紹介

- 1982年　大阪大学基礎工学研究科修士課程修了
- 1982年～2004年までNECにて各種組込みシステムのソフトウェアの研究開発に携わる
  - 携帯電話、特に携帯電話のJava、Linuxの開発
  - 工業用コンピュータ、宇宙ステーション（きぼう）のOS、H2Aロケットエンジン制御
  - 社員証、住基カードなどICカード
- 2004年～現在:　兵庫県立大学大学院応用情報科学/情報科学研究科教授
- この間、
  - 2001年:　博士(工学)
  - 2006年～：名古屋大学大学院情報学研究科附属組込み研究センター特任教授
  - 2016～2022年：(株)ヴィッツ　社外取締役

携帯電話Javaの試作
恐らく世界で始めて携帯で
Javaをちゃんと動かした.
(もうなくなったNECのガラケー)

多分世界初のLinux
ケータイの試作機
の待ち受け画面

高信頼リアルタイムUNIX。
今風な言い方だと、準仮想マシン型
リアルタイムUNIX。開発開始から
リリースまで一番長い製品。
この辺に入ってます多分。

私の研究開発人生…

- 一貫して組込みシステムの高度化に携わる

1984～1990年	UNIXの組込み応用(リアルタイム化)
1995～2000年	Javaの組込み/携帯電話/ICカード応用
2000～2005年	Linuxの携帯電話応用
2005～2010年	QoSベース組込みスケジューラ，リアルタイムデータストリーム
2015～	機械学習の組込み応用

今日の内容

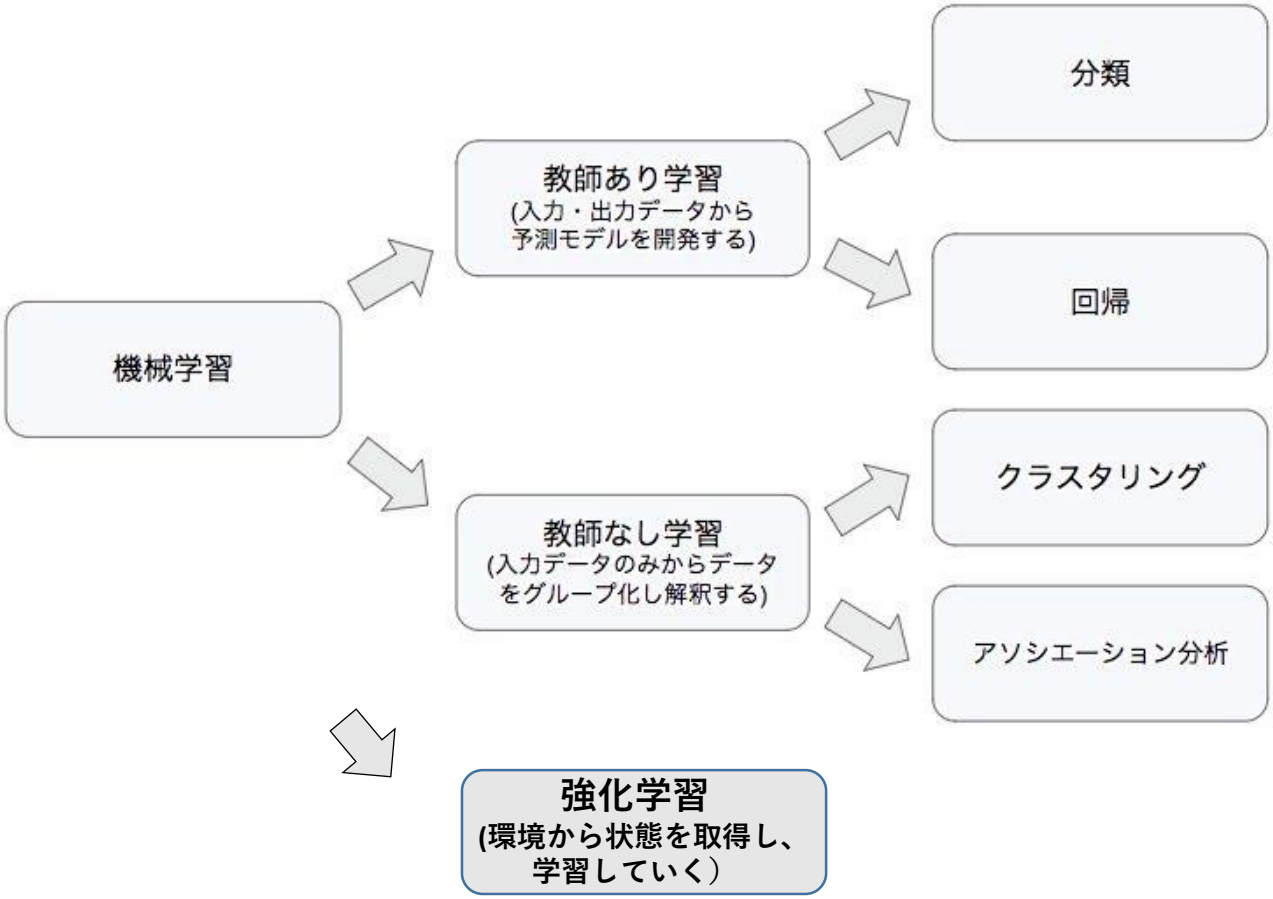
- 機械学習の事例ー機械学習の典型
- 機械学習の組込みシステム応用
  - TensorFlowLiteの応用事例
  - 3値量子化とその応用
  - その他の話題
- 機械学習雑感

クレジットと出典

- 機械学習の事例ー機械学習の典型
  - 吉次，画像分類と物体検出によるCOVID-19胸部X線画像診断の研究，兵庫県立大学応用情報科学研究科修士論文，2022年
- 機械学習の組込みシステム応用
  - TensorFlowLiteの応用事例
    - 甲藤、TensorflowLiteを用いた自転車操作支援組込みシステムの提案，兵庫県立大学応用情報科学研究科修士論文，2023年
  - 3値量子化とその応用
    - T. Zhang et al. Neural networks weights quantization: Target none-retraining ternary (tnt), EMC2-NIPS. IEEE, 2019.
    - Q. Zhao, et al., Optimal Ternarization Method for Federated Model Compression, IEEE International Symposium on low-power and high-speed chips　21, 2022.
  - その他の話題
- 機械学習雑感

- 機械学習の事例ー機械学習の典型
- 機械学習の組込みシステム応用
  - TensorFlowLiteの応用事例
  - 3値量子化とその応用
  - その他の話題
- 機械学習雑感

機械学習でできること



画像などの分類

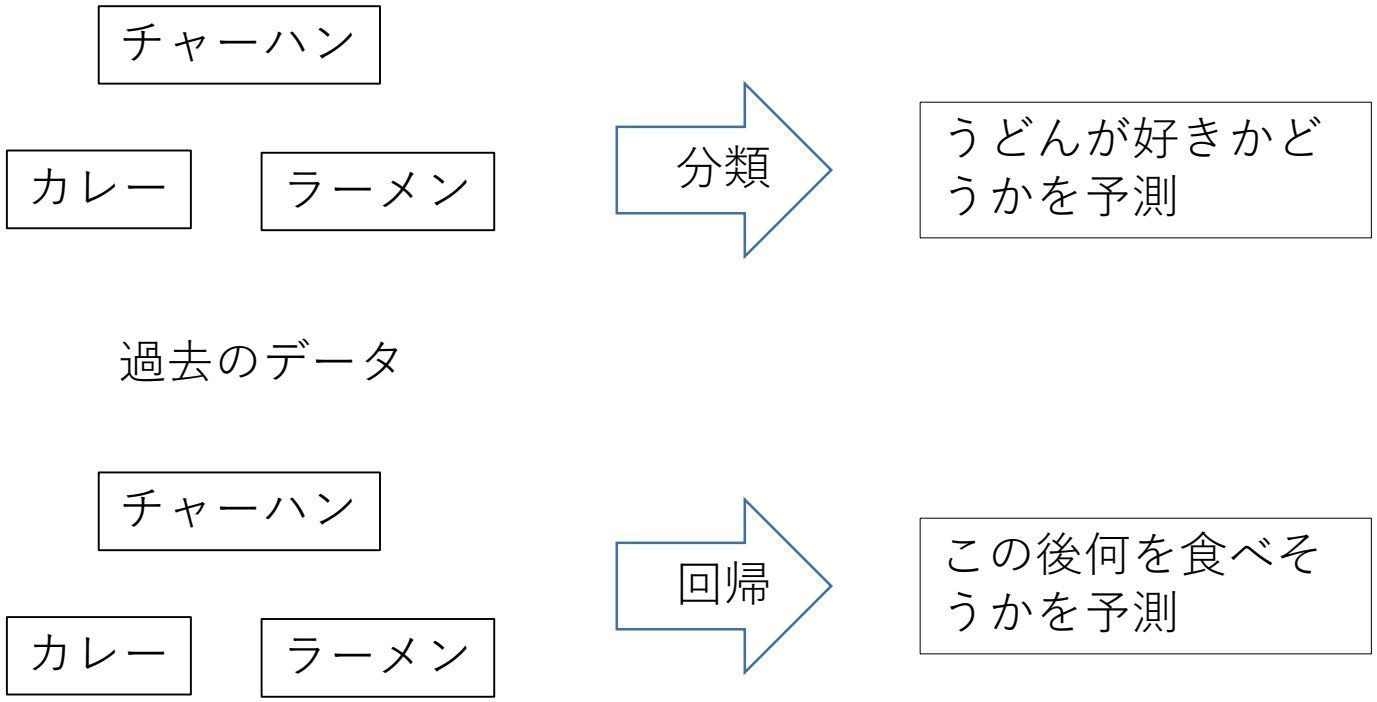
予測

グループ分け

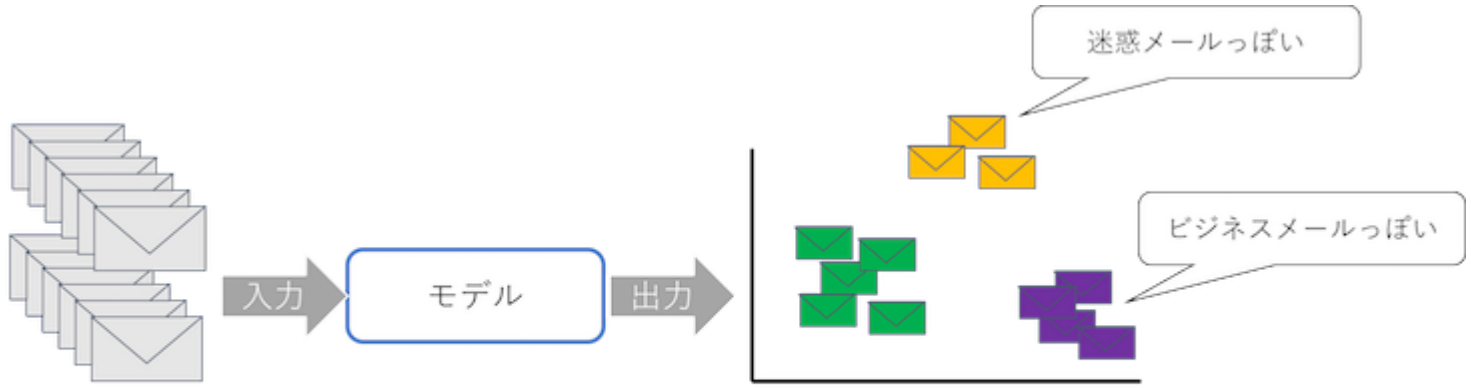
傾向

https://avinton.com/blog/2017/11/supervised-and-unsupervised-machine-learning/を修正

教師あり学習ー分類と回帰



教師なし学習ークラスタリング



https://ledge.ai/unsupervised/

クラスタリング

店舗での売上情報

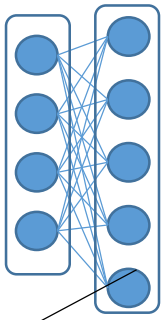
(仮定)ビールと
おむつやペアで
売れるか？

アソシエーション分析

信頼度70％

アソシエーション分析

教師あり学習の流れ



浮動小数点計算

学習

データ

ラベル　はさみ

機械学習ソフトウェア

分類基準

モデルの
学習

分類基準

予測

データ

機械学習
ソフトウェア

分類基準

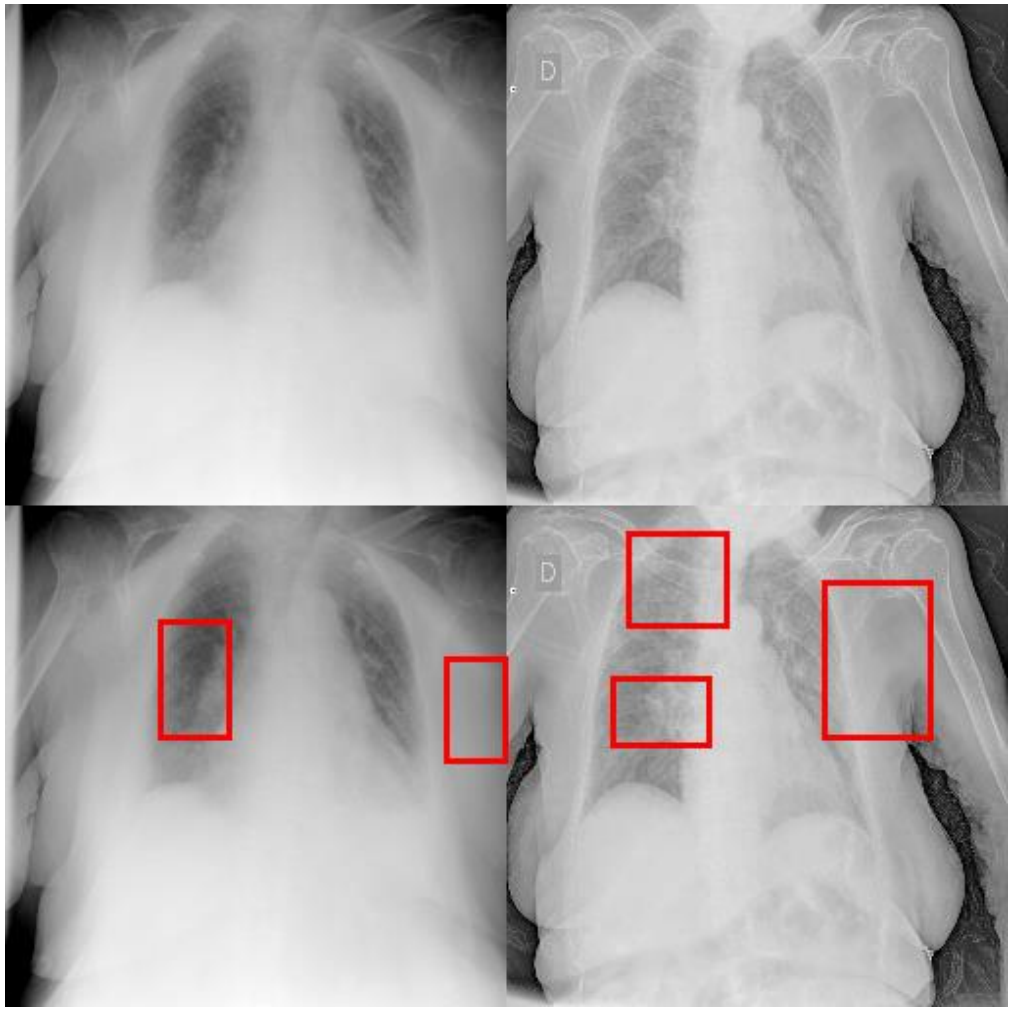
結果

はさみ

機械学習の典型事例

目的

COVID-19胸部X線画像から、臨床所見（４種）と混濁の位置をできるだけ高い精度で検出したい



・臨床所見（４クラス）
陰性
陽性（典型的な外観）
陽性（不確定な外観）
陽性（非典型的な外観）

※軽症、中等症、重症とは異なる分類。

・混濁の位置（左図下段）

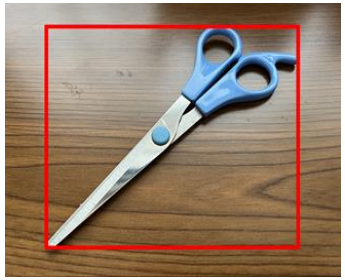
方法

- 画像認識と物体検出を利用

画像認識

物体検出

はさみ
「何」



「何」と
「どこ」

- 画像認識： Keras/EfficientNetとPyTorch/VisionTransformer
- 物体検出：Yolo

データ：kaggle

・kaggleのコンペ用教師データ（train）を利用
画像（dicom形式）、臨床所見＋混濁位置（csv形式）

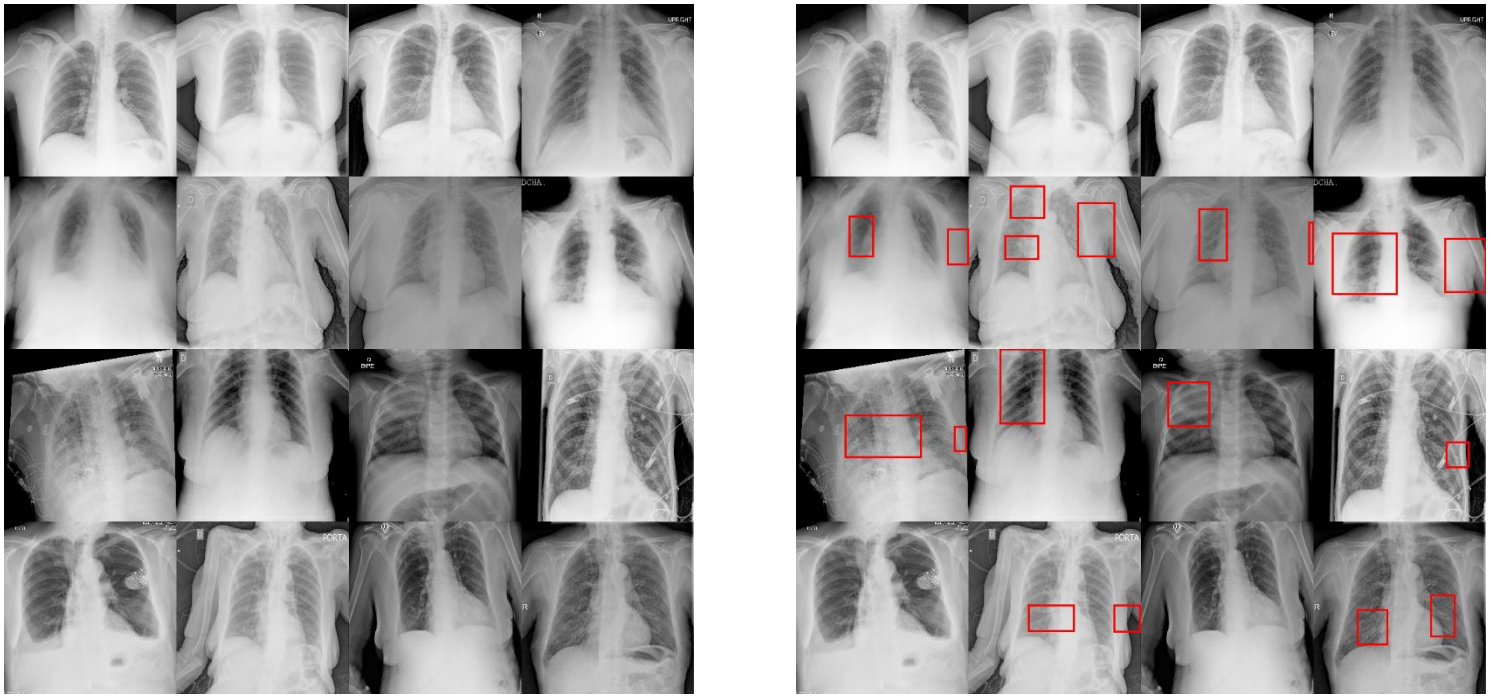
全6334件
5700件（90%）：学習
634件（10%）：予測

※深層学習にとっては、決して多いとは言えないデータ量

「深層学習で学習する場合、大体１クラスに付き5,000件程度のデータがあればまずまずのパフォーマンスが発揮されるが、人間レベルの精度を求めるとすると約10,000,000件という大規模なラベル付きデータが必要になる」

（I. Goodfellow, et al., Deep Learning,　MIT Press, 2016.）

・データ（画像）の例

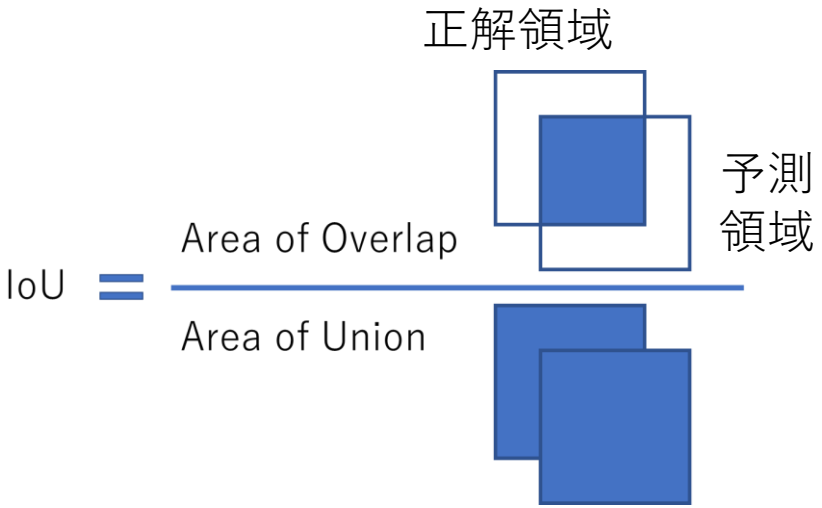


上段から順に、陰性（混濁なし）、陽性（典型的な外観）、陽性（不確定な外観）、陽性（非典型的な外観）で、右の画像は左の画像に混濁の位置を明示した状態。

・物体検出の位置の予測精度：IoU (Intersection of Union)

IoUとは、
正解の矩形と予測の矩形の
面積の和に対する両矩形が
重なった部分の面積比。

・完全に重なると、IoU=1.0
・正解判定の閾値はIoU=0.5



結果

アルゴリズム	方法	学習精度	予測結果
画像分類①4クラス	Keras/EfficientNet	77.71%	**76.03%**
画像分類②4クラス	PyTorch/VisionTransformer	67.03%	64.35%
物体検出　4クラス	YOLO v5	22%	44.16%
物体検出　2クラス	YOLO v5	53%	**46.85%**
最終結果　4クラス	アンサンブル	N/A	**50.47%**

画像分類（４クラス）の予測結果（76.03%）と
物体検出（２クラス）の予測結果（46.85%）を
合成（アンサンブル）することで、**50.47%**という結果を得た（少し向上）。

↓

まだ画像分類（４クラス）の予測結果（76.03%）には及ばない。

IOUを調整

- 画像分類（４クラス）と物体検出（２クラス）の予測結果をアンサンブルしても精度向上が小さかったのは、検出対象である**混濁の境界が周囲に対して曖昧なため**に、検出が困難なのではないか？

  →IoUの値を下げて再計算してみた（下図）

IoU	予測結果（アンサンブル後）
0.5	50.47%（再掲）
0.4	**65.62%**
0.3	**71.14%**
0.2	74.29%

- 機械学習の事例ー機械学習の典型
- 機械学習の組込みシステム応用
  - TensorFlowLiteの応用事例
  - 3ビット量子化とその応用
- 機械学習と組込その他の話題
- 機械学習雑感

機械学習のEdge-IoT適用のための研究

- 必要性
  - デバイス内での予測
    - クラウドでの処理削減による応答時間の確保
  - デバイスにデリバリ後のローカルな環境での学習

- 課題
  - 学習や予測に必要なコンピュータ資源の小型化
  - リアルタイム応答性能
  - デリバリ先での再学習
  - 複数デバイスでの分散学習

方法

学習

データ

ラベル　はさみ

分類ソフトウェア(学習)

分類基準

モデルの
学習

分類基準

クラウド
サーバ

予測

データ

分類ソフトウェア（予測）

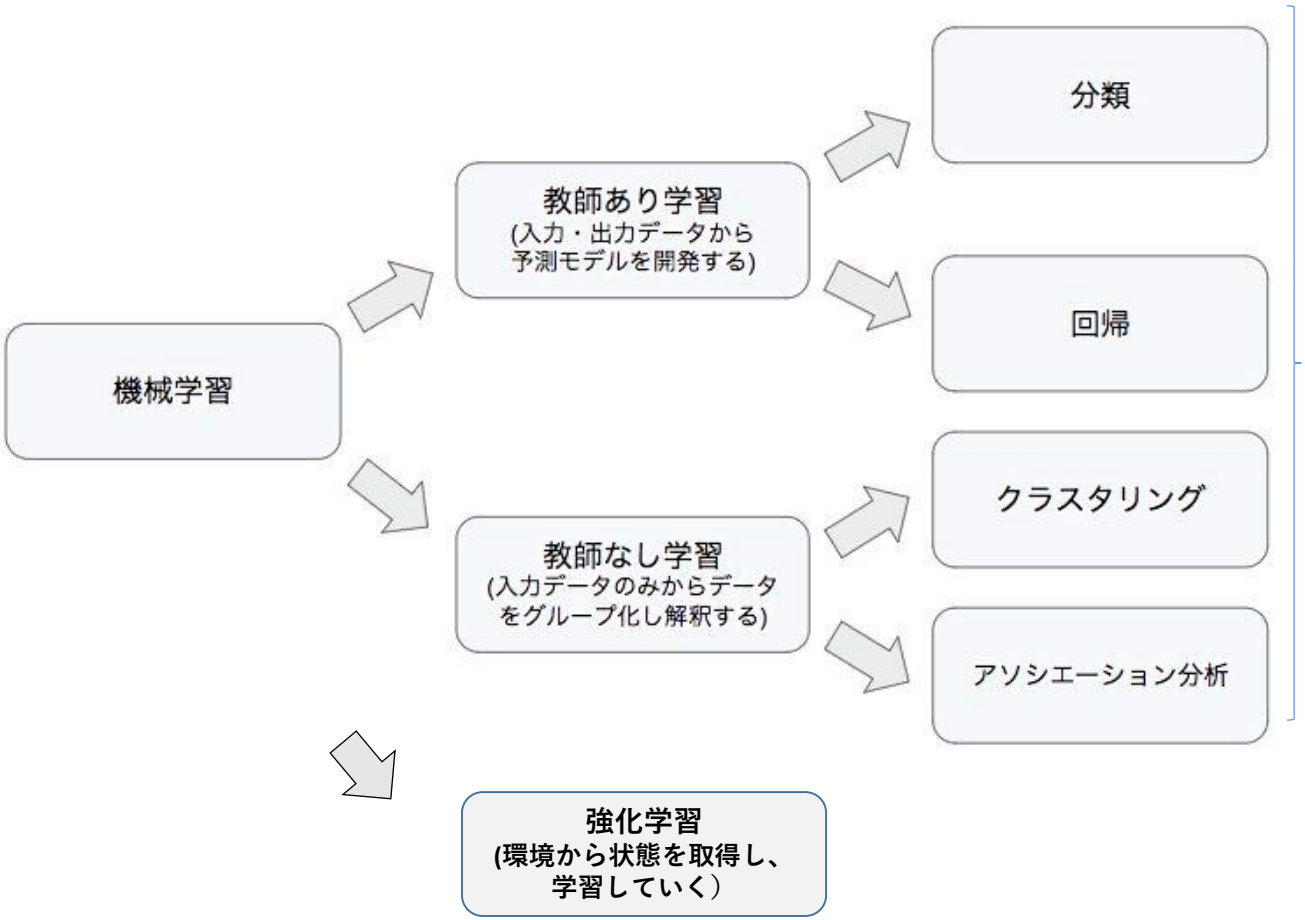
分類基準

ラベル

はさみ

組込み機器

機械学習アルゴリズムとEdge/IoT



デバイスで
ローカル利用

一種の制御システム

https://avinton.com/blog/2017/11/supervised-and-unsupervised-machine-learning/を修正

必要なコンピュータ資源の小型化 ーGreen Machine Learningの視点から

- コンパクトなアーキテクチャ設計
- エネルギー効率のよい学習
- エネルギー効率のよい予測
- 効率的なデータ利用

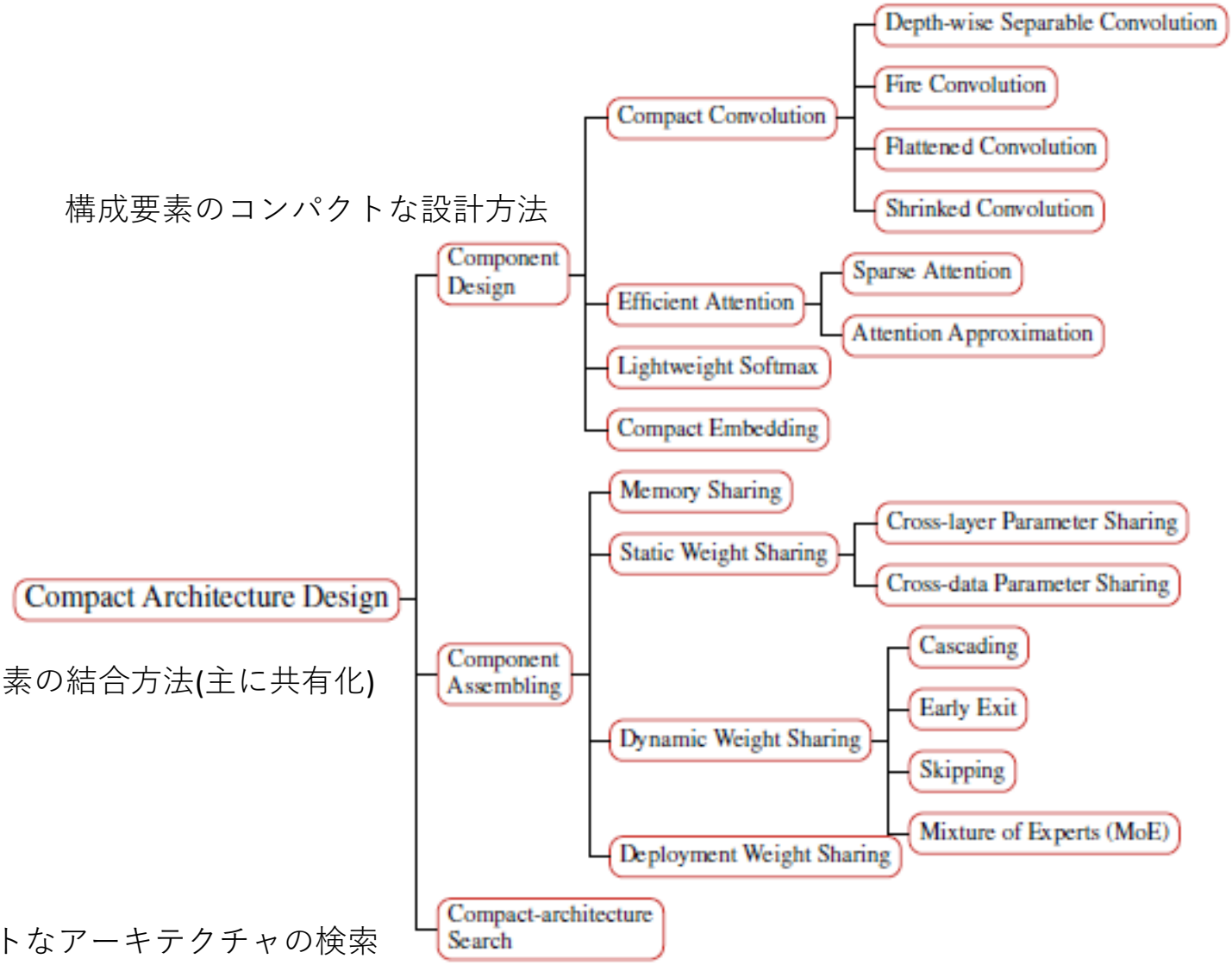
J. Xu, et al, A Survey on Green Deep Learning, CoRR, arXiv:2111.05193, 2021.

コンパクトなアーキテクチャ設計

構成要素のコンパクトな設計方法

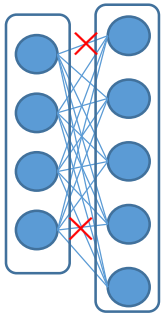
構成要素の結合方法(主に共有化)

コンパクトなアーキテクチャの検索



J. Xu, et al, A Survey on Green Deep Learning, CoRR, arXiv:2111.05193, 2021.

エネルギー効率のよい予測



J. Xu, et al, A Survey on Green Deep Learning, CoRR, arXiv:2111.05193, 2021.

組込みシステム向け機械学習ライブラリ

機械学習の代表的ライブラリ

- Tensorflow: Googleが開発主体。企業で受け入れられている
- Pytorch: FaceBookが開発主体。大学他研究機関で受け入れられている

組込みシステム向け機械学習ライブラリ

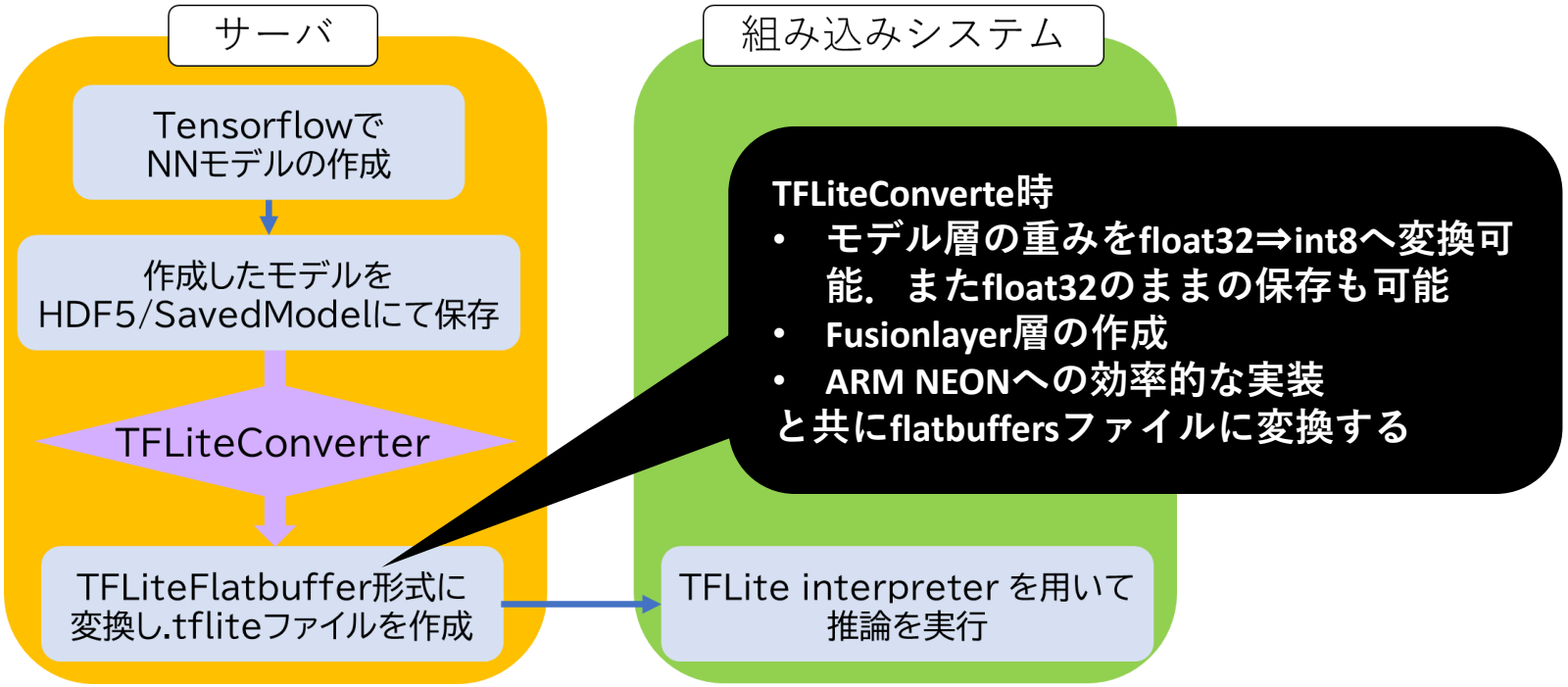
- TensorflowLite: TensorFlowモデルを量子化して小型化、利用オペレータの制限
- Pytorch Mobile: iOS、Androidがターゲット。モバイル用の効率的なインタプリタを具備。

機械学習の組込みシステム応用事例

- TensorFlowLiteとMobileNetの評価とそれらを使った警告システム
  - TensorFlowLite←量子化(32ビット→8ビット)
  - MobileNet←コンパクトな畳み込みと小型アーキテクチャの検索

TensorFlowLite

- TensorFlowモデルの変換
- float32をint8に量子化
- 利用オペレータの制限
- モデルファイルの簡素化



B. Jacob, et al., Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference, IEEE Conference on Computer Vision and Pattern Recognition, pp. 2704 - 2713, 2018.

MobileNet

- モバイルデバイス向け物体検出ライブラリ
- 小型の畳み込み層: フィルタ形状に工夫、動的に計算幅の縮小

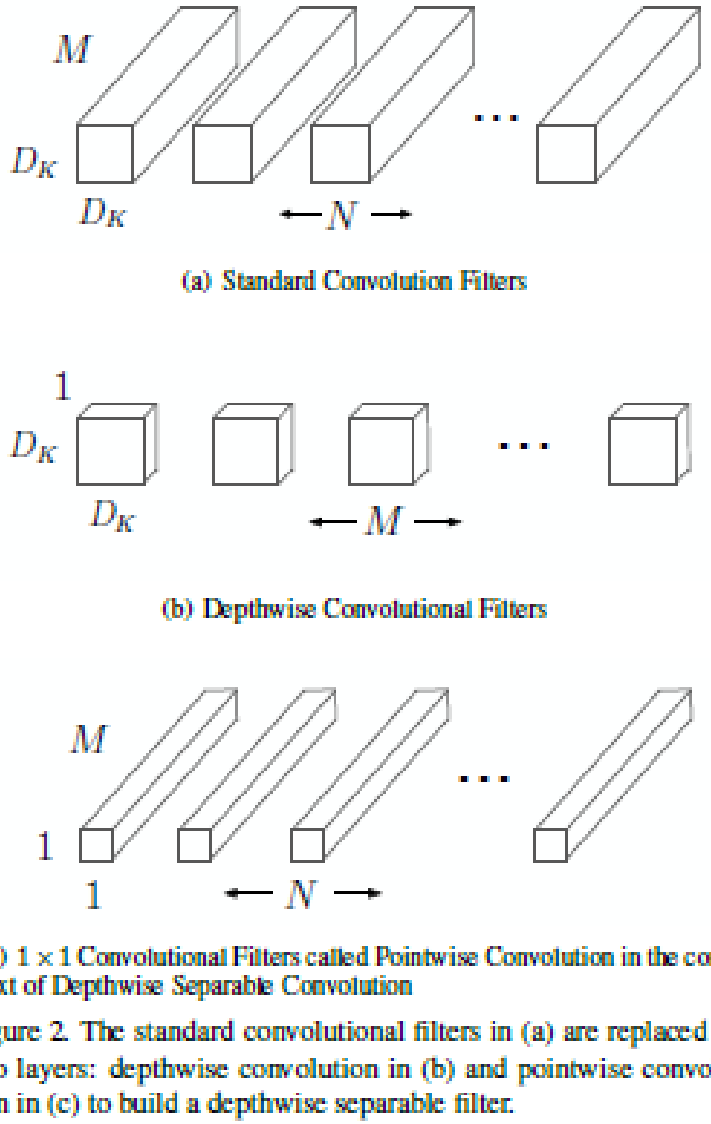


Figure 2. The standard convolutional filters in (a) are replaced by two layers: depthwise convolution in (b) and pointwise convolution in (c) to build a depthwise separable filter.

A. Howard, et al., MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, CoRR, arXiv:1704.04861, 2017.

評価環境

Tensorflowによる機械学習モデルの構築・訓練⇒GoogleColaboratory
TensorflowLiteによる推論　⇒Raspberrypi

使用機器・マシン	GoogleColaboratory	Raspberrypi4B
CPU	Intel(R) Xenon(R)	BroadcomBCM2711
GPU	Tesla T4	-
周波数	2.00GHz	1.50GHz
メモリサイズ	13.3GB	4CG

モデル

- MobileNet
- DenseNet

逆

Mobilenetは組み込みシステム向けのCNN，DenseNetは比較のため

データ

- ラベル３種類それぞれ300枚，合計900枚のRGB画像
- 訓練データ4/5の720枚，テストデータ1/5の180枚用いた

プログラミング言語

- Python

TFLiteの性能評価(画像分類)

モデル	使用機器	TF or TFLite	精度（%）	速度(ms/枚)	パラメタ数8M）	モデルサイズ（MB）
DenseNet	Google colab	TF	91.72	61.76	7.04	28.16
	Raspberrypi	TFLite_int8	91.21	195.0	6.95	7.190
	Raspberrypi	TFLite _float32	90.11	502.6	6.95	27.81
Mobilenet	Google colab	TF	80.11	44.07	3.23	12.93
	Google colab	TFLite_int8	79.12	12.80	3.19	3.240
	Raspberrypi	TFLite_int8	79.12	98.79	3.19	3.240
	Raspberrypi	TFLite _float32	79.67	139.1	3.19	12.80

精度はさほど落ちない

速度はfloat:int8で1:4

モデルのサイズはfloat:int8で1:4

性能評価(物体検出)

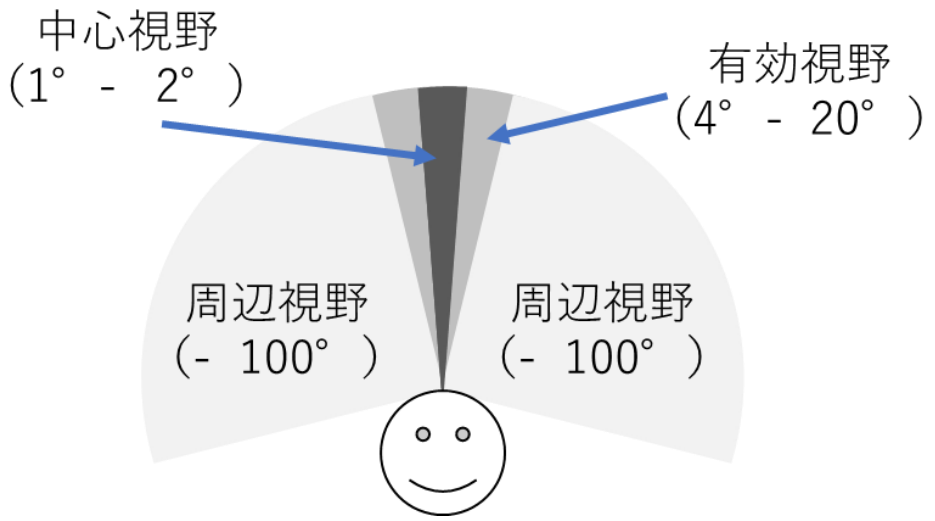
訓練データ、テストデータ；Google の画像検索か ら 3 種類（人形，箱，球）それぞれ 300 枚づつ，合計 900 枚のデータ

モデル	使用機器	TF or TFLite	モデルサイズ（MB)	速度(ms/枚)
YoLov4	Raspberrypi	TFLite_float32	252	9910
YoLov4	Raspberrypi	TFLite_int8	63	6420
SSD/Mobilenet	Raspberrypi	TFLite_int8	4	148

- 1秒で約6枚の画像から物体検出
- 1画像100m秒

自転車運転支援システムの試作

- 自転車の前方・後方の接近自動車を音声で警告
  - カメラの中心視野にない自動車の接近を検出
  - 自動車との距離を知らせる



自転車運転支援システムの試作

- RasbeyPiと単眼カメラで廉価に作成
- TensorFlowLite＋SSD/Mobilenetで作成
- 学習データはCOCO*を使用

カメラ名	画素数	画角	最大fps
OV5647	500万	62°	120

*https://cocodataset.org/#home

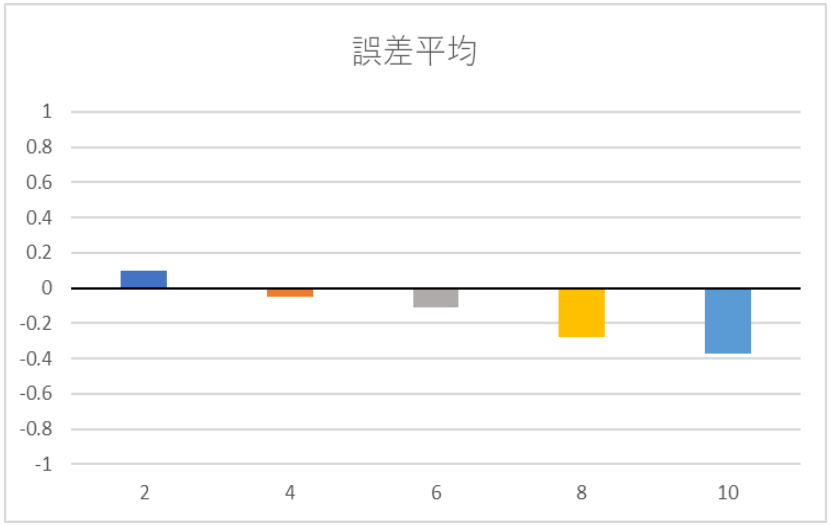
- 自動車の認識

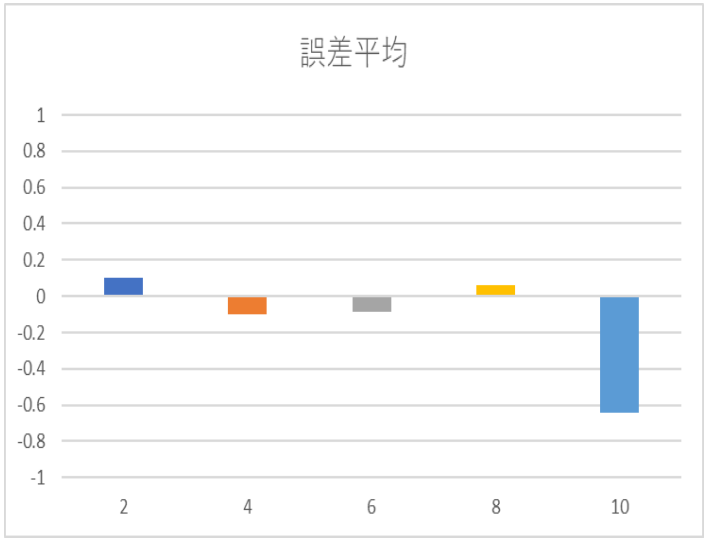
- 夕方でも認識できた！

結果ー認識の精度

- 55％～70％

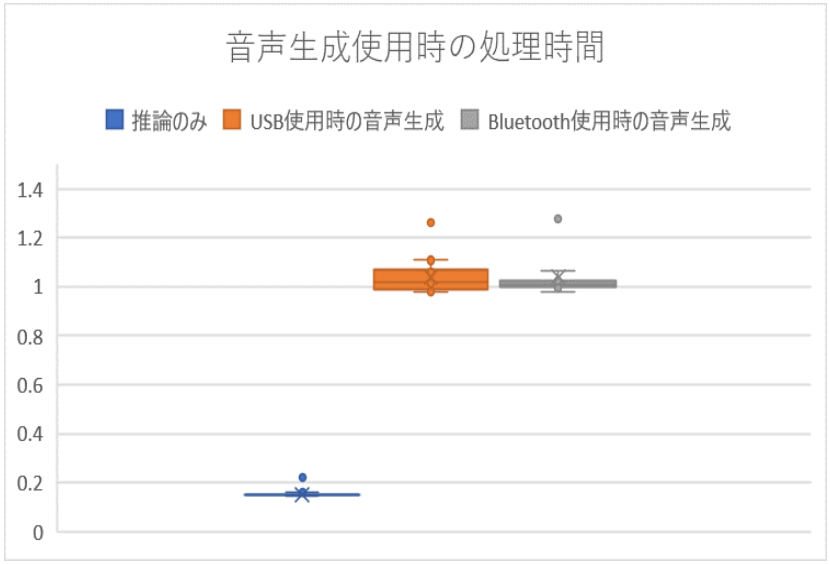
結果ー距離の精度

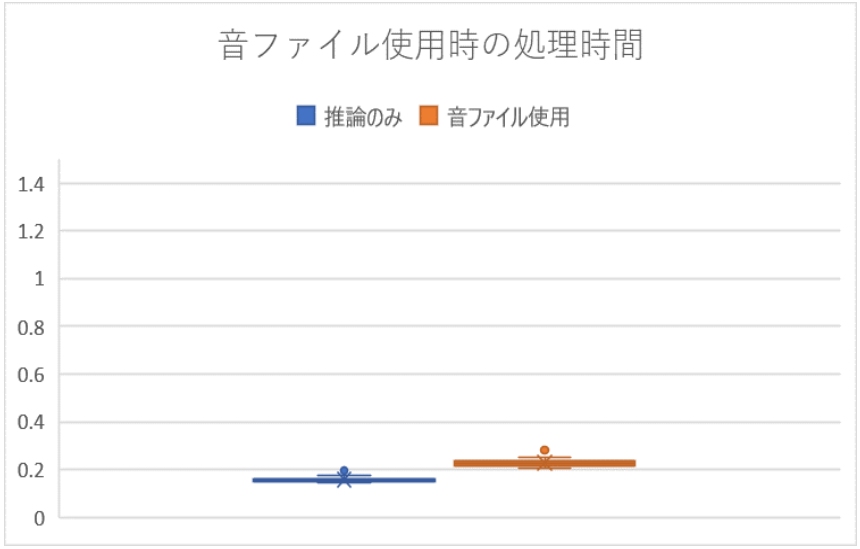




自動車の全体をカメラで捉えた場合　自動車の一部をカメラで捉えた場合

実行時間

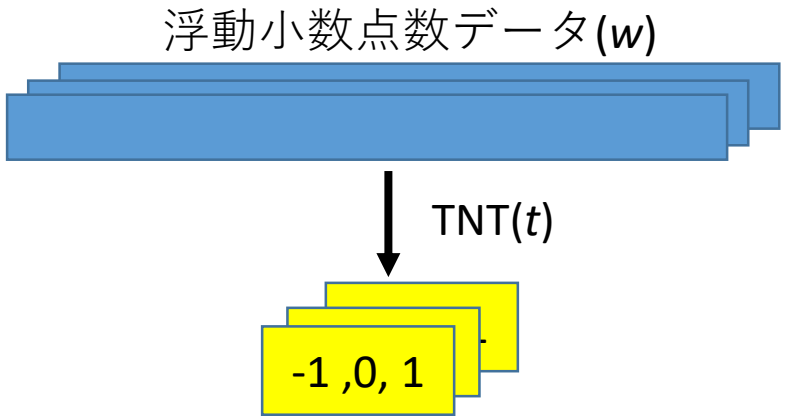

音声生成使用時の処理時間
推論のみ
USB使用時の音声生成
Bluetooth使用時の音声生成
1.4
1.2
1
0.8
0.6
0.4
0.2
0


音ファイル使用時の処理時間
推論のみ
音ファイル使用
1.4
1.2
1
0.8
0.6
0.4
0.2
0

- 機械学習の事例ー機械学習の典型
- 機械学習の組込みシステム応用
  - TensorFlowLiteの応用事例
  - 3値量子化とその応用
  - その他の話題
- 機械学習雑感

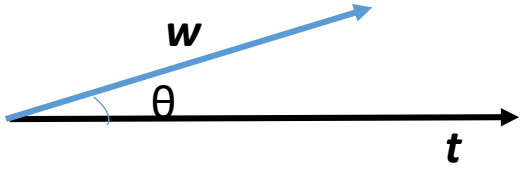
3値量子化(TNT)によるモデル小型化

- DNN浮動小数点計算を数ビットで計算する研究は多数
- 一般に浮動小数点を3値化には時間がかかる(O(3^N))

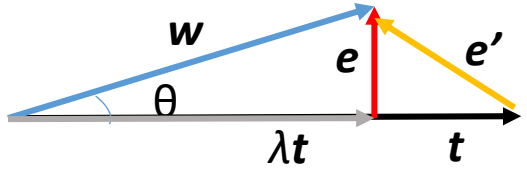


Target None-retraining Ternary(TNT)

- コサイン近似を利用して3値化の時間を短縮
- スケールパラメタを導入して近似差を小さく



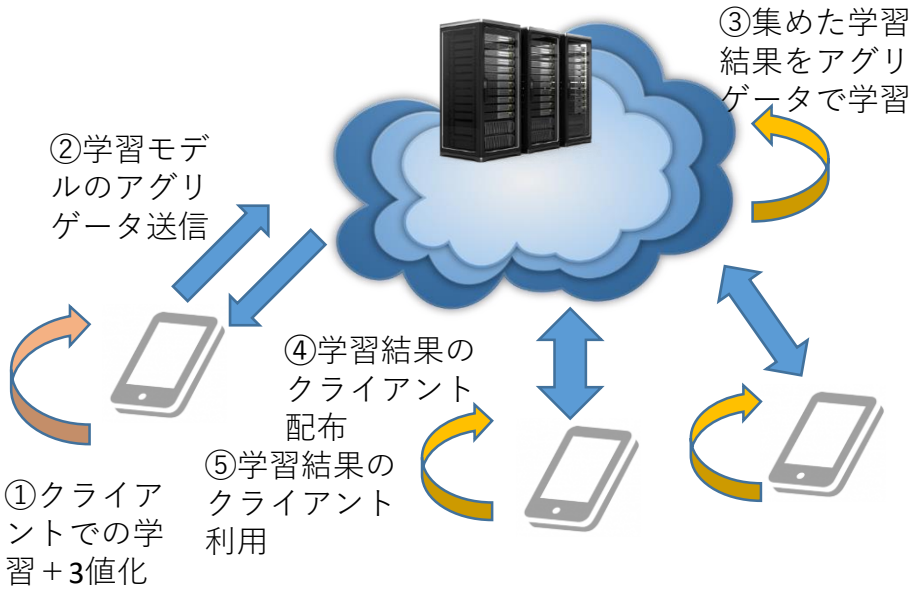
コサイン近似:con θが大きい(θが小さい）ほどよく近似



wと***t***の差が小さくなるようにパラメタλを追加

学習モデル量子化による連合学習

- 連合学習:　分散したクライアントで学習し、それをクラウドでマージして分散環境に配布
- アイデア：　クライアントでの学習を精度の落ちない量子化(TNT+コサイン量子化)を利用して、学習モデルの小型化、高セキュアを目指す



TNTの評価

精度

	ベースライン		TNT	
	MNIST	CIFAR－10	MNIST	CIFAR－10
LeNet	99.18%	-	98.97%	
VGG-7	98.87%	91.31%	98.73%	89.09%

余り精度は落ちない

モデルサイズ（MB)

	浮動小数点	3値化
Alexnet	13.4	4.97
ResNet18	30.1	12.5
ResNet50	60	5.18
Best case	420.07	18.96

サイズは1/3から1/20

TNT＋連合学習の評価

TNT＋連合学習の精度

	浮動小数点のみ	浮動小数点 連合学習	TNTのみ	TNT+連合学習
Alexnet	88.41%	88.64%	87.2%	80.38%
ResNet18	85.4%	93.39%	81.4%	90.75%
ResNet50	81.64	93.92%	80.1%	90.56%
Best case	96.98	97.19	94.38	95.68%

連合学習の方が精度がいい

TNT+連合学習でもそう精度は悪くならない？？

- 機械学習の事例ー機械学習の典型
- 機械学習の組込みシステム応用
  - TensorFlowLiteの応用事例
  - 3値量子化とその応用
  - その他の話題
- 機械学習雑感

その他の話題

- デバイスでの学習の必要性
  - デリバリ先での環境への対応
  - デリバリ時点からの環境変化

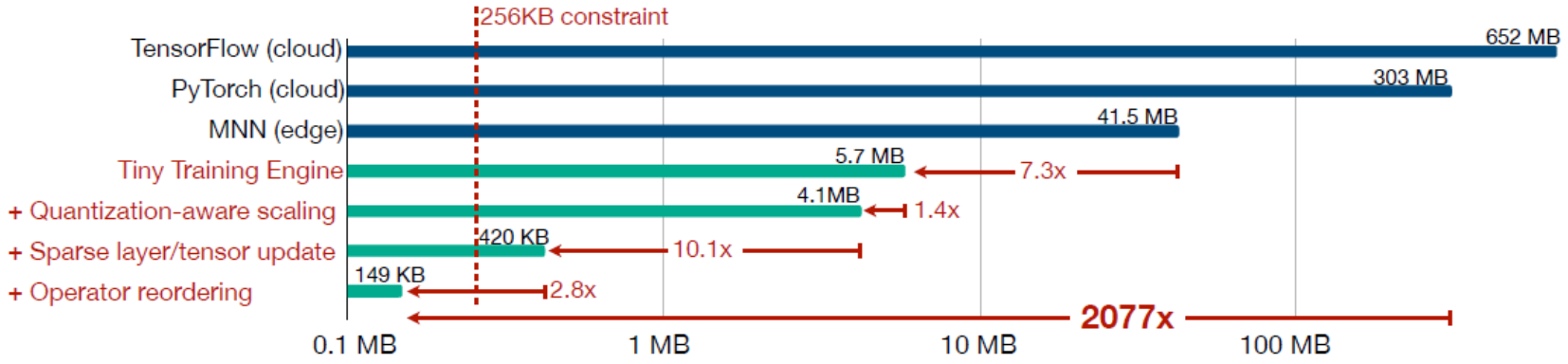
  - デリバリ先でのモデルの更新
    - オンライン学習、逐次学習…
    - 転移学習、ファインチューニング

- リアルタイム性能
  - FPGA化
  - Imprecise computationの応用

- 強化学習の応用

オンデバイス学習

- TinyML@MIT
  - On-Device Training Under 256KB Memory: Tiny Training Engine, Quantization-Aware Scaling, Operator reordering により、小メモリでオンデバイス学習を実現

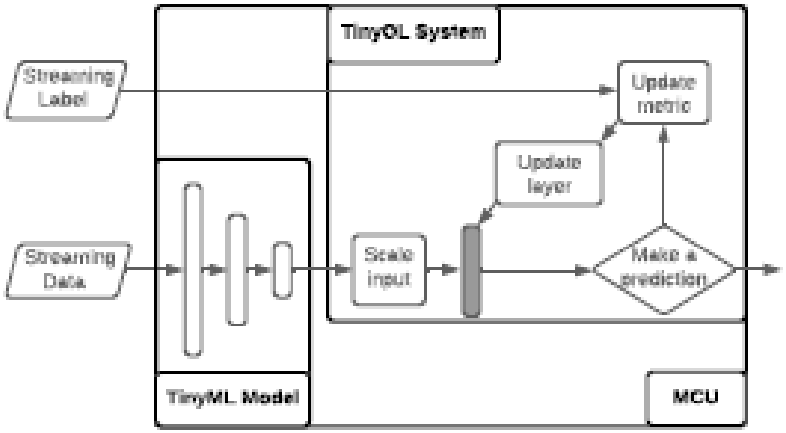


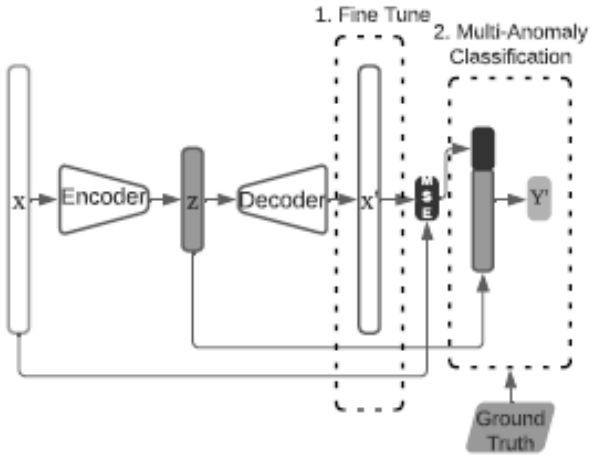
Ji Lin, et al., On-Device Training Under 256KB Memory, NeuroIPS 2022

https://tinyml.mit.edu/

オンデバイス学習

- TinyOL
  - 予測するNNの最後にファインチューニング層を挿入





H. Ren, et al., TinyOL: TinyML with Online-Learning on Microcontrollers, International Joint Conference on Neural Network 2021、IEEE.

機械学習雑感

- 研究開発とビジネス展開のスピードがこれまでと比較にならない
- データについて
  - ビッグデータはそう多くない
  - 教師ありのためのデータの準備(ラベリング)は大変
  - ラベルのつけ方に間違いがあることも
  - データに偏りがある場合
  - 入力データの準備(前処理)が大変
- これまでのソフトウェア工学の手法が通用しないことも
  - 要求他何もかも曖昧
  - 品質保証
  - モデルの説明責任
  - 石川他、機械学習工学、講談社、2022年

ご清聴ありがとうございます。