

人工知能（機械学習）応用論 I の演習 「野生動物検出」と「自動運転」への応用を学ぶ

門 勇一*・齋藤 寛**・奥山 祐市***・富岡 洋一***
kado@kit.ac.jp, hiroshis@u-aizu.ac.jp, okuyama@u-aizu.ac.jp, ytomioaka@u-aizu.ac.jp

1. はじめに -人工知能（機械学習）応用論 I 開講の背景-

人間の脳の神経細胞（ニューロン）の働きを理想化した数学的ニューロン・モデルが1940年代に提案されて以来、TPU（Tensor Processing Unit）等の機械学習専用ハードウェア技術が進展し、ニューロンが相互に繋がった多層構造の大規模なニューラルネットワーク（Neural Network, NN）の学習が現実的な時間で行えるようになってきました。多数のサンプルを学習データに用いて機械学習（Machine Learning, ML）させることによって、大規模NNは画像認識、自然言語処理、音声認識、自動運転、ロボティクス、医療診断、材料設計、等の広い分野で革新的な性能を発揮しています。MLはデータから自動的にパターンを学習し、その学習結果をもとに新しいデータの予測や分類を行う技術や手法の総称です。最近、注目されているテキストベースの対話が可能なChatGPT [1] も自然言語処理で高いパフォーマンスを示すアーキテクチャが実装された大規模NNで動作しています。

こうした状況を鑑み、令和3年度から全専攻の大学院生を対象にした講義として大学院専攻共通科目・人工知能（機械学習）応用論 I [2] を開講しています。講義の目的は受講生が修士論文や博士論文研究にMLを応用するための基礎知識の習得とGoogleのクラウド上でMLを学べるGoogle Colaboratory（略称Colab）[3] 活用法の習得です。受講生には電子・情報系専

* 未来デザイン工学機構 特定教授

** 会津大学 コンピュータ理工学科 教授

*** 同大学 コンピュータ理工学科 上級准教授

攻だけでなく機械系、材料系、生物系、デザイン・建築系専攻の大学院生及び学部生がいます。講師陣はコンピュータ理工学分野における研究・教育で優れた実績をもつ会津大学・コンピュータ理工学科 [4] の6名の教員により構成されています。講義は学習管理システムMoodleを活用して、双方向のWebExミーティングによるオンライン講義で提供されています。また、Colabを活用して、MLの基礎や応用を学び、NNの訓練や評価を行う方法を学びます。受講生は講義録画で反復学習が可能であり、MoodleのQ & Aフォーラムの機能を使って講義内容や毎回出題されるHomeworkに関する質問が出来ます。

本講義前半の構成は図-1に示す様に、工学における様々な問題に人工知能（ML）を応用する際に重要な知識の習得を目指しており、人工知能の歴史、問題の定式化、探索、推

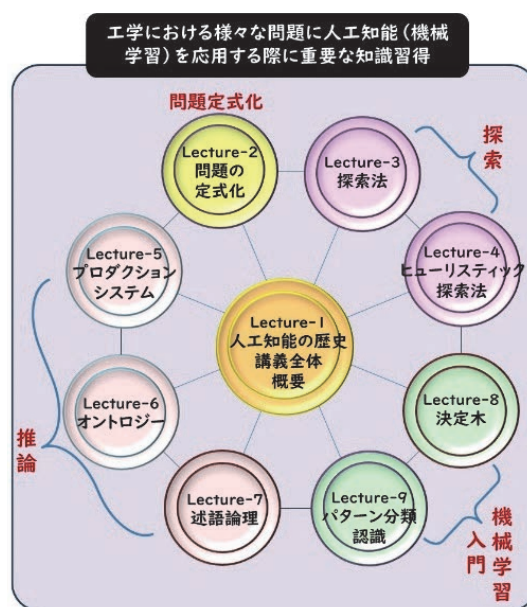


図-1 「人工知能（機械学習）応用論 I」前半の構成

論を学んだ後、パターン分類とパターン認識の概念を学びます。これらの概念は、データ解析、画像処理、音声認識、自然言語処理など、様々な領域で広く応用されています。次に、MLの例として、決定木とNNと言うアプローチの異なる手法の基礎を学びます。更に、Support Vector Machine (サポートベクターマシン)、Ensemble Learning (アンサンブル学習)、Random Forest (ランダムフォレスト)、K-means (k平均法)等、人工知能応用の視点で重要なMLモデルを学びます。

「ものづくりプロジェクト」と言う実践的なML応用演習を含む本講義後半の構成を図-2に示します。後半はColabを用いて大規模NNの基礎とMLメカニズムを学び、深層学習(Deep Learning, DL)を「野生動物検出」と「自動運転」に応用する演習を情報科学センター [5]・演習室 [6]にて対面形式で実施しています。

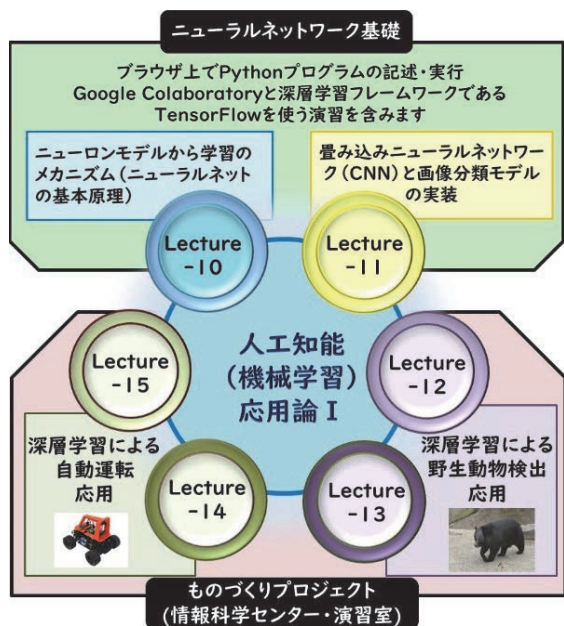


図-2 「人工知能 (機械学習) 応用論 I」後半の構成

2. 演習目的と概要

紹介する「ものづくりプロジェクト (対面形式のML応用演習)」は情報科学センター・演習室の情報環境を利用して実施しています (図-3の演習風景参照)。大規模NNを使ったMLの知識や理論を座学で学ぶことは重要ですが、実際にその知識を具体的な応用タスク (野生動物検出、自動運転) に適用する演習を通じ

て、以下の4点の目的を達成します。

- (1)「理論と実践のギャップの認識」; 座学だけでは理解できない実践的な問題や課題に取り組み、理論と実際の応用とのギャップを体感し、それを解決する方法を学ぶ。
- (2)「実践的なスキルの習得」; プログラミング、データ前処理、モデルの選択とチューニング、結果の評価・解釈など、MLプロジェクト全体の流れを体験し、実践的なスキルを習得する。
- (3)「問題解決能力の向上」; 実際の応用には多くの予期しない問題や挑戦が伴います。これらの問題に対処することで、柔軟な問題解決能力を向上させる。
- (4)「実社会への適用」; 実世界の問題に対する解決策としてMLを使用する際の課題や制約を理解し、実社会での適用を視野に入れたアプローチを学ぶ。

これらの目的を達成するために、最初に演習に向けた事前学習として、多層ニューラルネットワークを用いたDLの理論と知識を学び、Colabを活用して、DLライブラリであるTensorFlow [7]を利用した学習モデルの定義、学習のプロセス、及び評価について学びます。DLとは複数の隠れ層を持つNNを使用してデータから特徴やパターンを学習する手法を指し、MLの一分野です。その後、情報科学センター・演習室にてDLを「野生動物検出」と「自動運転」に応用する対面型演習を実施しています (図-3参照)。各受講生は2面のディスプレイを活用して、演習用説明資料とColabのインタフェース画面を表示させて効率的に演習を進められます。



図-3 深層学習応用の演習風景 (情報科学センター・演習室)

3. 事前学習 (演習に向けた準備)

対面型応用演習に向けた事前学習では深層 NN (Deep Neural Network, DNN) に関する基礎理論を学び、Colab を用いて DNN の定義、学習、評価を行うスキルを習得します。担当教員は会津大学の富岡・上級准教授です。

最初に、人間の神経細胞 (ニューロン) の働きを理想化した数学的ニューロン・モデルの提案から DNN に至る歴史を学び、DNN の学習メカニズムを知ることから始まります。学習プロセスの計算に関わる逆誤差伝搬法 [8]、計算グラフ、及び自動微分について理解し、学習プロセスを進める上での勾配消失問題 [8] への対策例を学びます。次に、画像認識が得意な畳み込みニューラルネットワーク (Convolutional Neural Network, CNN) の構成と画像パターン検出を例に各層の機能を理解します。CNN の学習原理は DNN と同じですが、学習データに対応したモデルを作成し過ぎて未知のデータを予測できなくなる過学習問題とその対策法学びます。

次に、受講生は Google が提供するクラウドベースのノートブック環境である Colab を活用して、予め準備された DL ライブラリ TensorFlow (Keras) を使って DNN モデルの定義、学習、評価をするためのスキルを身に付けます。受講生はブラウザを通じて Python コードを実行し、結果の解析や可視化を行えます。

具体的には後述する「自動運転」の演習で活用する自動運転のフレームワークである Donkeycar [9] というプラットフォームを使います。このプラットフォームは DL を自動運転やロボティクスに応用して、学習や実験を行うためのツールとして広く使用されています。Donkeycar は Donkey Simulator [10] というシミュレーション・ツールを提供し、実際に自律走行 RC カーを使用せずに、自動運転モデルの開発とテストを行う仮想環境を提供します。従って、このツールは実世界での学習データ収集や実験に比べてコストが低く、安全であり、効率的なモデルの訓練と調整を可能にします。以上述べた Colab と Donkeycar の統合演習環境を図 4 に示します。

演習手順を図 5 に示します。演習のステッ

プは 5 段階です。「学習データ準備」から始まります。Donkey Simulator で取得した仮想 DonkeyCar の走行データを使用します。データには走行画像と操作情報 (ステアリングとアクセル) が含まれ、画像サイズは 96×96 画素のカラー画像であり、画像データの shape は $(96, 96, 3)$ となります。次に「NN モデルの定義」を行います。Colab 内の TensorFlow (Keras) ライブラリを用いて、6 層の畳み込み層、3 層の Max プーリング層、2 層の全結合層で構成される CNN モデルを定義します。最終的に、ステアリング操作に関する左方向、直進、右方向の 3 クラスに分類するため最終層のニューロン数は 3 となります。次が「学習実行」であり、Keras [11] の fit 関数を用いて CNN モデルを学習させます。この関数は与えられた学習デー

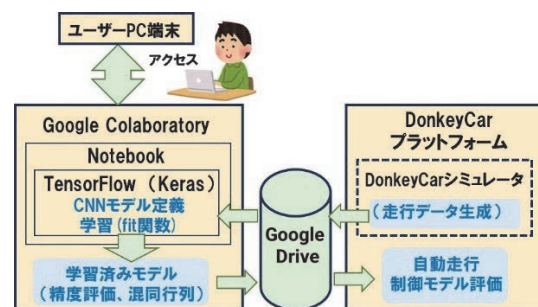


図-4 Colab と Donkeycar が連携した統合演習環境

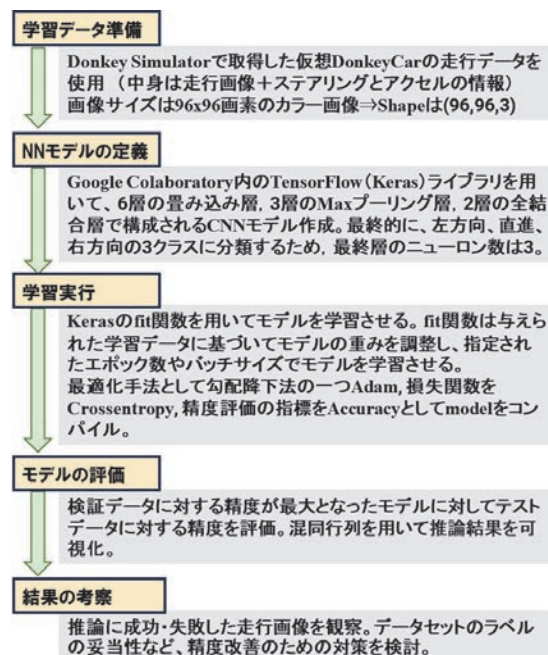


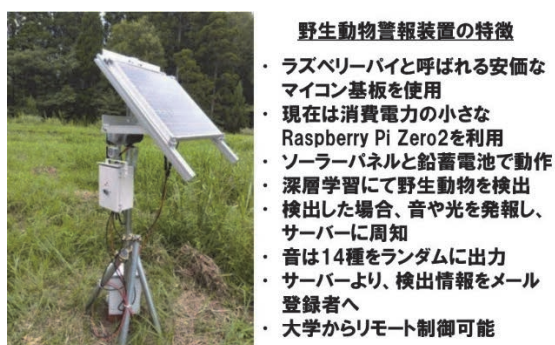
図-5 Colab と Donkeycar による演習手順

タに基づいてモデルの重みを調整し、指定されたエポック数やバッチサイズでCNNモデルを学習させます。最適化手法として勾配降下法の一つ Adam [12]、損失関数を Crossentropy [13]、精度評価の指標を Accuracy [14] としてモデルをコンパイルします。次が「モデルの評価」のステップで、検証データに対する精度が最大となったモデルに対してテストデータに対する精度を評価します。更に、混同行列を用いて推論結果を可視化します。最後に「結果の考察」であり、推論に成功・失敗した走行画像を観察し、データセットのラベルの妥当性など、精度改善のための対策を検討します。

4. 「野生動物検出」の演習

「ものづくりプロジェクト」実施日の午前、会津大学・斎藤教授が企画・運営された「深層学習（DL）を野生動物検出に応用する演習」が実施されました。最近、クマやイノシシが生活空間に現れ、人身被害や農作物被害が頻発しています。会津大学では、DNNにより画像データの複雑なパターンを学習できるDLを用いて、会津大学で開発された野生動物警報装置にて撮影した画像に野生動物が写っているかどうかを判断して、警報を出すシステムを研究・開発しています。既に、福島県会津地方に複数の野生動物警報装置を設置して、その有効性が実証されています（図-6参照）。

野生動物を検出する学習モデルの構築はサーバーで行われ、学習済みのモデルを屋外設置する低消費電力のマイコン基板（Raspberry Pi Zero2 [15]）に実装して推論させる方法を採用しています。屋外設置の野生動物警報装置はマイコン基板、トレイルカメラ、無線通信機、等



野生動物警報装置の特徴

- ・ ラズベリーパイと呼ばれる安価なマイコン基板を使用
- ・ 現在は消費電力の小さなRaspberry Pi Zero2を利用
- ・ ソーラーパネルと鉛蓄電池で動作
- ・ 深層学習にて野生動物を検出
- ・ 検出した場合、音や光を発報し、サーバーに周知
- ・ 音は14種をランダムに出力
- ・ サーバーより、検出情報をメール登録者へ
- ・ 大学からリモート制御可能

図-6 実証中の野生動物警報装置

で構成され、太陽光パネルと蓄電池を電源としています。人工知能とセンシング機能（トレイルカメラ）を組み合わせる屋外に遍在させるインテリジェントなIoT（Internet of Things）応用への先駆けとも言えます。

野生動物検出の演習概要を簡潔に説明します。まず、ColabとPyTorch [16]を用いて、関連する画像データにより野生動物を検出するための学習（訓練）を行います（Step1 図-7参照）。次に、学習済みの深層学習モデルを本学が準備したRaspberry Pi 4 [15]に実装して、与えられた画像データから野生動物を検出（推論）させます（Step2 図-8参照）。最後に、検出結果を評価します。

Step1 モデル作成（学習）

主にサーバーで開発
(演習では、Google Colabをクラウド環境を使用)

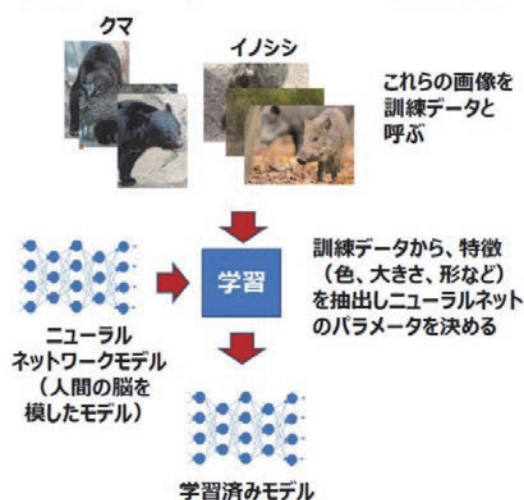


図-7 NNモデルの訓練データによる学習

Step2 モデルの実装（推論）

運用するマシンに実装
(演習では、ラズベリーパイを使用)

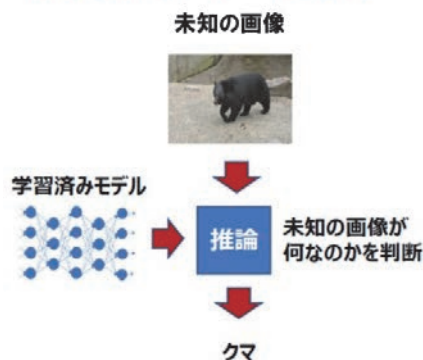


図-8 学習済み NNモデルによる推論

本演習の Step1 に相当する NN モデル作成と学習に関する情報環境を図-9に、演習手順を図-10に示します。最初に、演習環境等の準備を行います。Colab の環境で、予め準備された Notebook 等を開き、YOLOv5 [17] に関するプログラム・推論結果、学習用 Python コード、推論用 Python コード、及び物体検出モデル作成用のクマ・イノシシ・それ以外の動物・背景のカラー写真とラベル付けに関する数値データを統合したデータ（図-11の学習用画像データ例参照）を My Drive に展開します。物体検出の畳み込みニューラルネットワークとして、YOLOv5（You Only Look Once ver.5）を

物体検出モデルに対するデータセット （ラベル付けた画像）



図-11 学習用画像データセット例

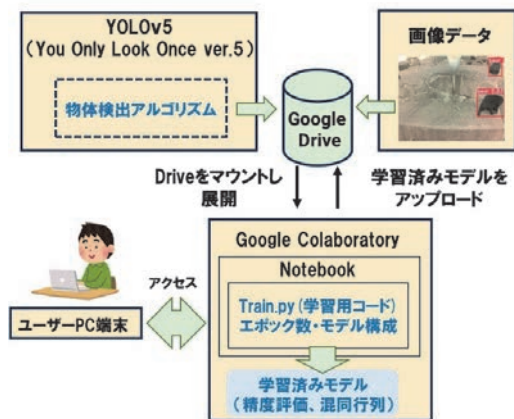


図-9 モデル作成と学習の演習環境

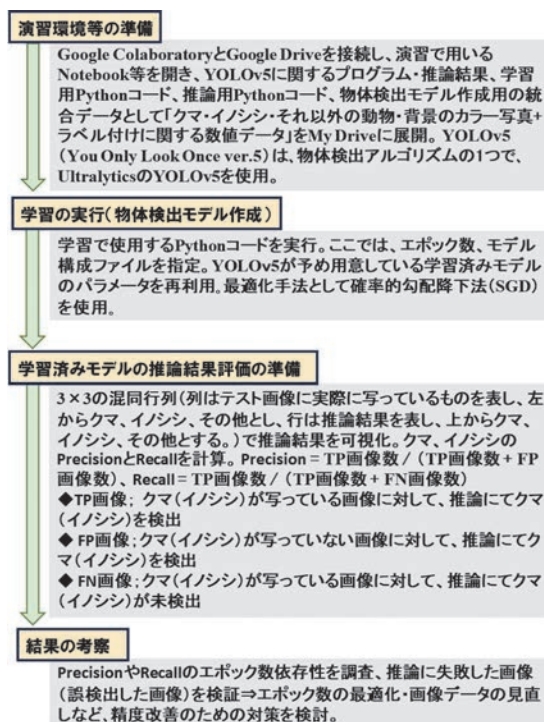


図-10 野生動物検出演習（Step1）の手順

使います。YOLO は人間のようにより目ただけで物体検出ができるという意味です。次が学習実行のステップで、学習用の Python コードを実行します。ここでは、エポック数、モデル構成ファイルを指定します。最適化手法には確率的勾配降下法（SGD）[18] を使います。

次に、学習済みモデル評価の準備を行います。3 × 3 の混同行列（列はテスト画像に実際に写っているものを表し、左からクマ、イノシシ、その他とし、行は推論結果を表し、上からクマ、イノシシ、その他とする。）で推論結果を可視化します。混同行列の Epoch 数依存性の例を図-12に示します。混同行列より学習したモデルの精度評価として、クマ、イノシシの Precision と Recall を計算します。Precision = TP 画像数 / (TP 画像数 + FP 画像数)、Recall = TP 画像数 / (TP 画像数 + FN 画像数) と定義します。TP 画像、FP 画像、及び FN 画像の定義は図-10に示しました。最後に、学習結果の考察で、Precision や Recall のエポック数依

Epoch数 = 50

クマ	14	1	0
イノシシ	0	0	0
その他	8	13	2
	クマ	イノシシ	その他

Epoch数 = 100

クマ	20	0	0
イノシシ	1	10	2
その他	1	4	0
	クマ	イノシシ	その他

図-12 3 × 3 混同行列の Epoch 数依存性

存性を調査、推論に失敗した画像（誤検出した画像）を検証します。更に、エポック数の最適化・画像データの見直しなど、精度改善のための対策を検討します。演習室における学習用画像データセットの確認シーンを図-13に示します。

本演習の Step2 に相当する学習済みの物体検出モデルをマイコン基板（Raspberry Pi 4 Model B [15]、図-14）に実装して推論させ、野生動物検出モデルとしての性能評価プロセスについて説明します。Step2 の演習に関する情報環境を図-15に、演習手順を図-16に示します。

最初に、演習に必要なキーボード、マウス、ディスプレイ、USBカメラをマイコン基板に接続します。また、Step1 で作成した学習済みモデルをマイコン基板にダウンロードします。次に、予め準備したテスト画像を用いて推論を実行させます。具体的には、Raspberry Pi の統合開発環境を用いて、推論で使用する Python コードを実行します。この時、テスト画像サイズやスコアの閾値を 0.6 に設定します。推論結果は作成されたフォルダに自動保存されます。次に、USBカメラで野生動物の画像を撮影し、この画像データに対して推論を実行し



図-13 学習用画像データセットの確認シーン

Raspberry Pi 4 Model B



仕様
CPU: ARMCortex-A72
(quad core 1.5GHz)
Memory: 4GB
Current: 459mA (idle, 2.3W)
Power supply: 2.5A

図-14 学習済みモデルを実装したマイコン基板仕様

ます。図-16の下部に示されるケースでは、学習用データセットに大きく撮影されたクマが含まれていないため、大きく撮影されたクマを検出出来ていない結果（クマに対して赤枠が付かない）を確認できます。また、画像サイズやスコアを変えて推論を実行します。最後に、結果の考察を行います。Step1 同様に混同行列を用いることで、画像サイズやスコアの閾値が結果に及ぼす影響を検証します。

Step2 モデルの実装（推論）



図-15 学習済みモデルによる推論の演習環境

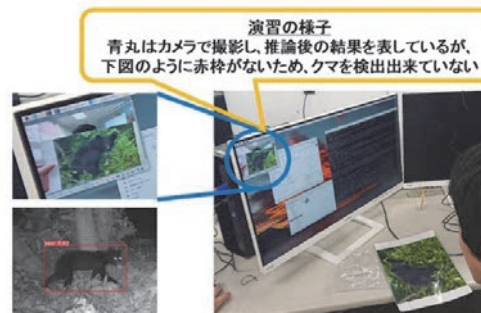
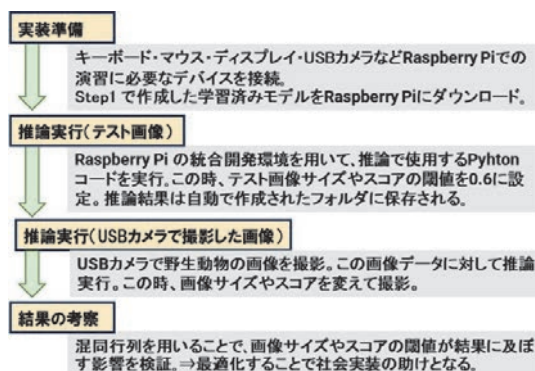


図-16 学習済みモデルによる推論演習手順

5. 「自動運転」の演習

自動運転の演習は会津大学の奥山・上級准教授により企画・運営されています。演習の概要は既に図-4を用いて説明した様に Colab、Donkeycar、及び Google Drive を連携させた統合演習環境で、シンプルな自動運転のフレームワークである Donkeycar を用いて、DL に必要な学習データの収集と学習の方法を理解します。統合演習環境で Donkey Simulator と Python プログラムを連携させて、DL の活用方法の具体的なイメージを体験することになります。

自動運転の演習環境を図-17に、自動運転演習の手順を図-18に示します。最初に、演習環境の準備で、Colab のノートブックの記述に従い、AI 自動走行車用の DNN を訓練するために必要な Donkeycar、Tensorflow、学習スクリプトをインストールし Google Drive を Colab に接続します。Donkeycar については既に説明しました。次に、学習モデルの定義を行います。この演習では予め定義された TensorFlow ライブラリ [7] を用いて5層の畳み込み層と2層の全結合層で構成された NN モデルを利用します。演習で利用した NN モデル構成を図-19に示します。各畳み込み層では特徴マップを生成しますが、入力画像に異なるフィルターを適用します。異なるフィルターは、画像の異なる特徴（エッジ、テクスチャなど）を検出します。最後の全結合層は、畳み込み層からの特徴を受け取り、それらを用いて最終的な分類を行います。全結合層の出力は2つの実

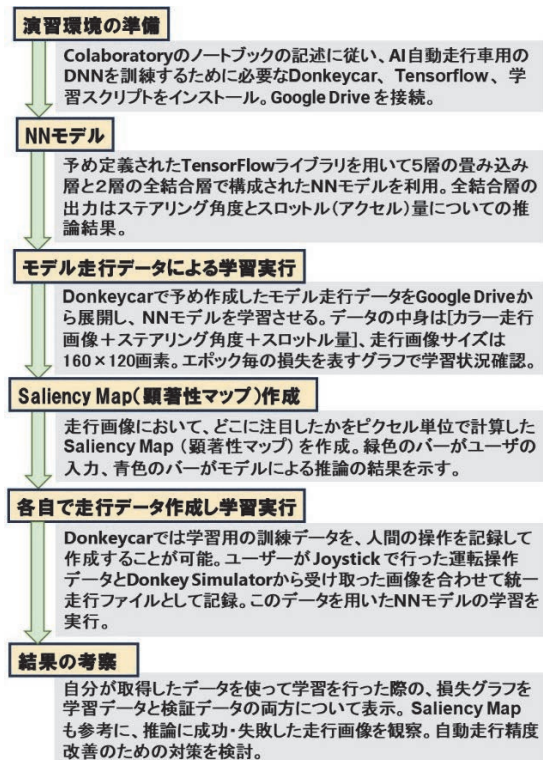


図-18 自動運転演習の手順



図-19 演習で利用した NN モデル構成

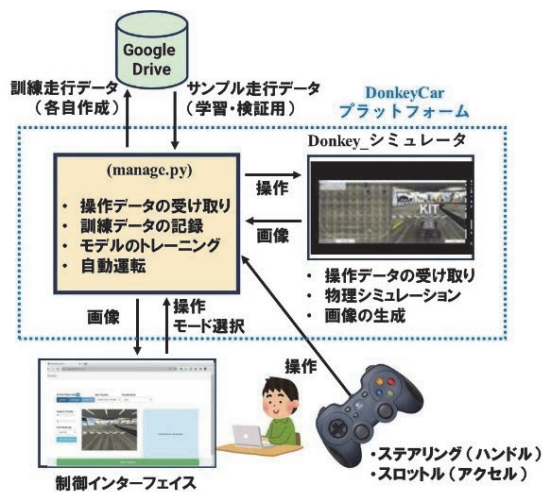


図-17 自動運転の演習環境

数値で、それぞれステアリング角度とスロットル(アクセル)量について推論結果を出力します。

次に、走行画像において、NN モデルがどこに注目したかをピクセル単位で計算した Saliency Map (顕著性マップ) [19] を作成します。緑色のバーがユーザの入力、青色のバーがモデルによる推論の結果を示します。次のステップでは、各自が Joystick を用いて Donkey Simulator の特定のコースで自動車を

走らせることにより走行データ作成し、そのデータを用いて学習を実行させます。この演習では Donkey Simulator に Donkeycar を接続し、Donkeycar にシミュレータ内の小型車を自動運転させます。Donkeycar での自動運転の実行中、Donkey Simulator では Donkeycar から受け取った操作を元に、自動車の移動の物理計算を行い、画像を生成します。Donkeycar は Donkey Simulator が生成した画像を受け取り、学習済みモデルを実行し、操作を生成します。また、Donkeycar では、自動運転を学習するための訓練データを人間の操作を記録して作成することもできます。Donkeycar を Joystick モードで起動するとユーザが Joystick で行った操作データと Simulator から受け取った画像データが統合され統一ファイルとして記録されていきます。学習用の走行データ作成状況を図-20 に示します。

最後に推論結果の考察を行います。自分が取得したデータを使って学習を行った際の、損失グラフを学習データと検証データの両方について表示させます。各自の走行データによる学習曲線例を図-21 に示します。Saliency Map [19]



図-20 学習用の走行データ作成状況

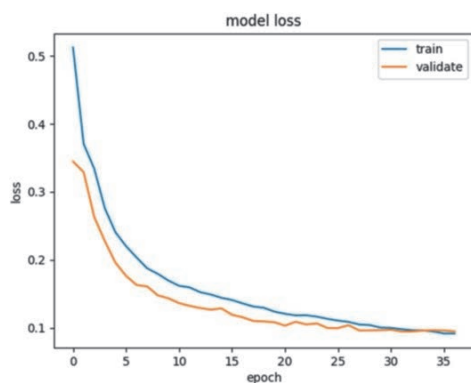


図-21 各自の走行データによる学習曲線例

も参考に、推論に成功・失敗した走行画像を観察し、自動走行精度改善のための対策を検討します。

6. まとめ

受講生は本講義・演習を通して Colab を活用して、「野生動物検出」と「自動運転」と言う具体的なタスクに ML を応用するプロセスを実践的に学びました。その結果、(1) 座学だけでは理解できない実践的な課題に取り組み、理論と実際の応用とのギャップを体感し、それを解決する方策を学びました。(2) Colab を活用して、プログラミング、データ前処理、モデルの選択・定義、結果の評価・解釈など、ML プロジェクト全体の流れを体験し、実践的スキルを習得しました。(3) 実際の応用で直面する予期せぬ問題に対処する柔軟な問題解決の対策例を学びました。これにより、実世界の問題に ML を応用して解く際に直面する課題や制約を理解し、実社会の問題を柔軟な思考で取り扱う姿勢を学びました。演習に参加した受講生の演習プログラムに対する満足度は非常に高く、次年度の演習に TA として参加したいと言う積極的な受講生が複数名いました。

今後、材料研究を含む様々な専攻分野の大学院生が博士・修士論文研究に DL を応用する力を獲得することを目指し、講義・演習プログラム内容を毎年見直していきます。また、情報科学センター・演習室の優れた情報環境を活用して、Colab 等のクラウドサービスを活用するスキルを習得し、実世界の具体的問題に DL を応用して解決する実践的プロセスを学ぶ機会を提供していきます。何よりも受講生が大規模 NN による DL の高い課題解決能力に関心を持ち、目を輝かせて演習に取り組める演習内容を継続していきます。

謝辞

「人工知能(機械学習)応用論 I & II」の企画・運営に日頃から丁寧なご指導をいただいております会津大学の趙強福・副理事長(大学院コンピュータ理工学研究科長)、及び講義をご担当されている先生方に深謝致します。また、本演習に毎年 TA を派遣していただいている本学

の増田新・副学長に感謝致します。最後に、大学の榊田・情報科学センター長を初めスタッフの皆様による演習実施に向けた支援に御礼を申し上げます。

参考情報

- [1] <https://chat.openai.com/auth/login>
- [2] <https://www.syllabus.kit.ac.jp/?c=detail&pk=2024>
- [3] <https://colab.research.google.com/?hl=ja>
- [4] <https://u-aizu.ac.jp/intro/faculty/cse/>
- [5] <https://cis.kit.ac.jp/>
- [6] https://cis.kit.ac.jp/services/seminarroom/covid-19_eduseguide/
- [7] <https://www.tensorflow.org/?hl=ja>
- [8] https://free.kikagaku.ai/tutorial/basic_of_deep_learning/learn/tensorflow_network_basic_backward
- [9] <https://www.donkeycar.com/>
- [10] https://docs.donkeycar.com/guide/deep_learning/simulator/
- [11] <https://keras.io/about/>
- [12] https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam
- [13] https://www.tensorflow.org/api_docs/python/tf/keras/losses/CategoricalCrossentropy
- [14] https://www.tensorflow.org/api_docs/python/tf/keras/metrics/Accuracy
- [15] https://ja.wikipedia.org/wiki/Raspberry_Pi
- [16] https://colab.research.google.com/github/DeepInsider/playground-data/blob/master/docs/articles/pytorch_neuralnetwork.ipynb
- [17] <https://github.com/ultralytics/yolov5>
- [18] <https://keras.io/ja/optimizers/>
- [19] https://en.wikipedia.org/wiki/Saliency_map