

令和3年度 特別研究報告書

教師あり学習を用いたごみの均一化作業の 自動化の検討

龍谷大学 理工学部 情報メディア学科

学籍番号 T180468 廣瀬 大河

指導教員 三好 力 教授

内容梗概

日常生活を送っていると、どうしても出てしまうごみ。そんな現在の産業廃棄物処理業界は深刻な人手不足に悩まされている。近年機械の技術が発達してきており、ごみの種類を分別する作業は自動的に可能になっている。しかし、分別された後、ごみが含んでいる水分量に応じて燃えやすいごみと混ぜて、ごみを均一化させる作業は手動で行っている。この作業を自動化させることで人手不足の緩和に繋られるのではないかと考える。実際に京都市南部クリーンセンターに見学、インタビューをしたことを基に現状を把握することで問題点を解決できるのではないかと考えた。

本研究では、現在手動で行われているごみの均一化作業を教師あり学習を用いて、各ごみの量に応じて混ぜ合わせる比率を最適化して、必要なら燃料を追加するというシステムを提案し、実験によりその実現性を確認する。

目次

第 1 章 はじめに	1
1.1 研究の背景と目的.....	1
1.2 現状 - 既存技術.....	1
1.3 問題点.....	4
第 2 章 関連技術	5
2.1 機械学習.....	5
2.2 AI の三大分類.....	5
2.3 教師あり学習.....	6
2.4 Softmax 関数.....	7
第 3 章 提案手法	9
第 4 章 実験	10
4.1 実験目的.....	10
4.2 実験概要.....	10
4.3 実験結果.....	11
第 5 章 考察	15
第 6 章 まとめ	16
謝辞	17
参考文献	18
付録 1	19

第1章 はじめに

1.1 研究の背景と目的

日常生活を送っていると、どうしても出てしまうごみ。私たちが出しているごみを集めると膨大な量になり、社会問題にもなっている。日本全体で1年間に家庭から排出されるごみの量は、4,272万トン(家庭から出される一般廃棄物のみの量であるため、産業廃棄物は除く)。ある日、ごみ処理施設の実態に密着取材する番組を見ていた。そこでは工場内でごみを自動で分別している映像や大量のごみを人がクレーンで操作している映像があった。ごみは、リサイクルできるごみとリサイクルできないごみに分けられる。リサイクルできるごみは、工場に運ばれて新たなものへと生まれ変わる。リサイクルできないごみは、埋め立てをするために最終処分場に運ばれる。最終処分場にごみを運び込む前に、焼却工場でごみを燃やして体積を大きく減らしている。[1]

現在、廃棄物処理業界は人手不足である。ある日、ごみ処理施設の実態に密着取材する番組を見ていた。そこでは工場内でごみを自動で分別している映像や大量のごみを人がクレーンで操作している映像があった。近年機械の技術が発達してきており、ごみの種類を分別する作業は自動だったが、ごみを効率よく燃やすために行うごみの質の均一化作業はまだ手動で行っていた。というよりも今ある自動運転に任せるよりも手動の方が高い精度で均一化できるということが現実だ。そこでまだ手動で行っている作業を自動にすることで人手不足の緩和に繋がるのではないかと考えた。

本研究では、実際にごみ処理施設である京都市南部クリーンセンターへ見学に行き、ごみの分別、均一化作業の実態や現場の人のインタビュー、見学したのを基に、現在手動で行われているごみの均一化作業を教師あり学習を用いて自動化するシステムを提案する。

1.2 現状 - 既存技術

実際に工場の中で行われている工程を既存技術として以下に示す。我々が実際出しているごみの大まかな流れをさすてな京都の資料を基に説明する。コンベアで高速回転式破砕機に送り、細かく砕く。その後、磁選機で磁石の働きにより鉄を選別する。次に粒度選別機に送り、木や紙などの大きな可燃物を取り除く。残ったものからアルミ選別機で磁界を利用してアルミニウムだけを選別して回収し、残ったものは、ごみピットに送り、焼却処理する。選別された鉄とアルミニウムは資源としてリサイクルするために運び出される。



図1 ごみピット(さすてな京都の資料より引用)

ごみピットとは、ごみを溜めておくことができる場所であり、一時的に置くことで、ごみの水分が下に落ちてごみがより燃えやすくなるようにしている。また、ごみクレーンでかき混ぜることで固まっているごみをほぐす効果やごみの質を均一化している。ごみピットの大きさは幅 1.6m、長さ 42mで、ピットの底から、ホッパまでの高さは 35mある。ごみを貯めることができる有効深さは 18mで、容量は 12000 m³(小学校のプールでおよそ 36 杯分)である。ごみクレーンの一掴みのごみの量は、4 トン前後である。ごみクレーンの運転は勤務時間の関係で日中は人が行っている。人間のオペレータ勤務時間内に行っている作業はごみピットにあるごみを固まらないようにほぐしたり、ごみの質を見て度のごみを混ぜるかを判断してクレーンを操作している。またごみの水分が多かったら燃えやすくなるように燃料を入れている。だが、夜間は自動運転しており、その運転自体はごみを混ぜる程度である。その運転は日中でのごみピットに溜めたごみは、ごみクレーンで窓の正面に見える焼却ポツパ(投入口)から焼却炉に投入される。また、右側に見えるポツパは、バイオガス化施設に投入される。選別資源化施設やバイオガス施設で処理されたごみをバイオガスポツパの中に入れて、自動で(重さによって)分別される。分別されたごみは①～⑥の穴から出てくる仕組みになっている。

- ①大きなゴミ(固いごみ)を破砕機で細かく砕いたごみ
- ②③バイオガス化施設で選別された、発酵に適さない軽いプラスチックごみ
- ④バイオガス化施設で発酵後に残った不要なごみ
- ⑤大きなごみ(柔らかいごみ)を切断機で細かく切り刻んだごみ
- ⑥1階のプラットフォームにある扉から投入された燃やすごみ

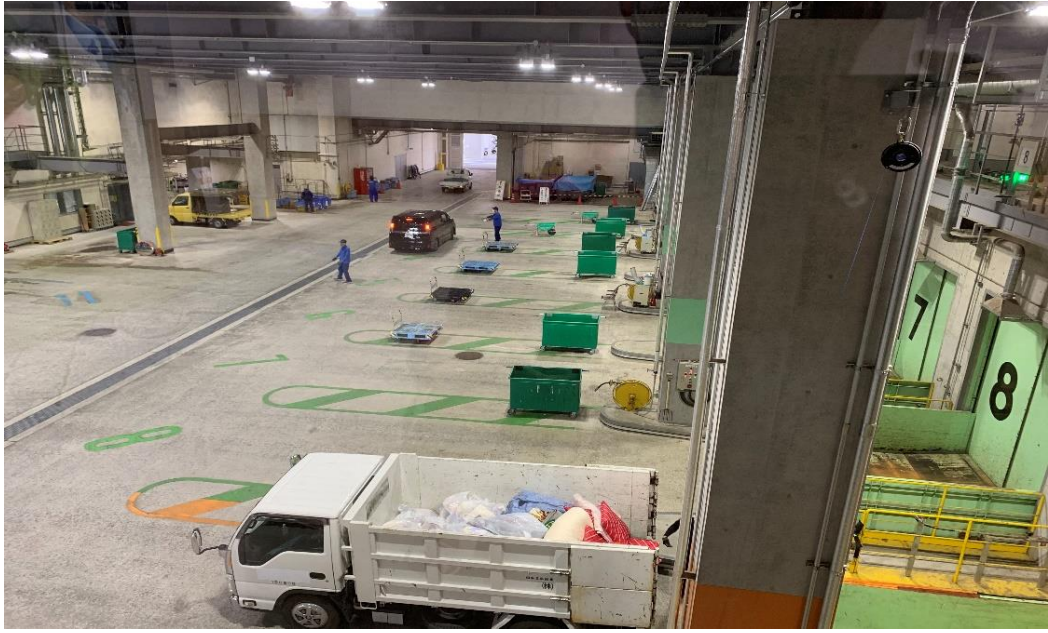


図 2 1階のプラットフォーム(さすてな京都の現地の写真)



図 3 実際のごみピットの写真(さすてな京都の現地の写真)

1.3 問題点

1.2 節で述べた既存技術には問題点がある。夜間の自動運転ではごみが固まらないように混ぜている程度であるということだ。夜間は自動運転だが、現状人が行った方が効率よく均一化作業が行える。3 交替勤務でも回せるが、シフトの勤務時間の関係により夜は機械による自動運転プログラムに任していることが多い(京都市南部クリーンセンター内のさすてな京都で現地の人にしたインタビューより)。本来、ごみを上手く混ぜて燃えるようにしたいというのが目的だが、ごみを混ぜるだけで上手く燃えるようにするのが中々難しい。また人間が行っている作業の一つにごみを混ぜた上に燃えやすくなる燃料を混ぜるという処理を行っている。そのため、現在人間が行っている「ごみが燃えやすくなるように燃料を追加する」という作業を自動化するには、燃料の適切な追加を自動で行うことも必要である。

第2章 関連技術

本章では提案手法に利用する関連技術の概要について述べる。

2.1 機械学習

機械学習とは、人間が行う学習能力と同様の機能をコンピュータで実現する技術のことをいう。具体的には、コンピュータに大量のデータから指定したアルゴリズムを基に特徴を発見させたり(モデル化)、その特徴に基づいて、コンピュータが最適化や推論、判断など自動的に行うという仕組みとなっている。

コンピュータは、人間の目や脳と異なりデジタルな情報しか扱えない。例えば、コンピュータが猫と犬を見分けるには、猫については、体長が〇〇cm～〇〇cm 程度、足の長さは〇〇cm～〇〇cm 程度という具合に、数字でルールを定義することが必要になる。ところが動物には個体差があるため、それぞれの特徴を一つ一つ人間がコンピュータに教えることは不可能である。そこで詳細なルールを与える代わりに、大量のデータと分析・解析ができるアルゴリズムを与えることにより、コンピュータが人間の学習能力と似たような判断作業を行えるようにするのが機械学習である。[2]

2.2 AIの三大分類

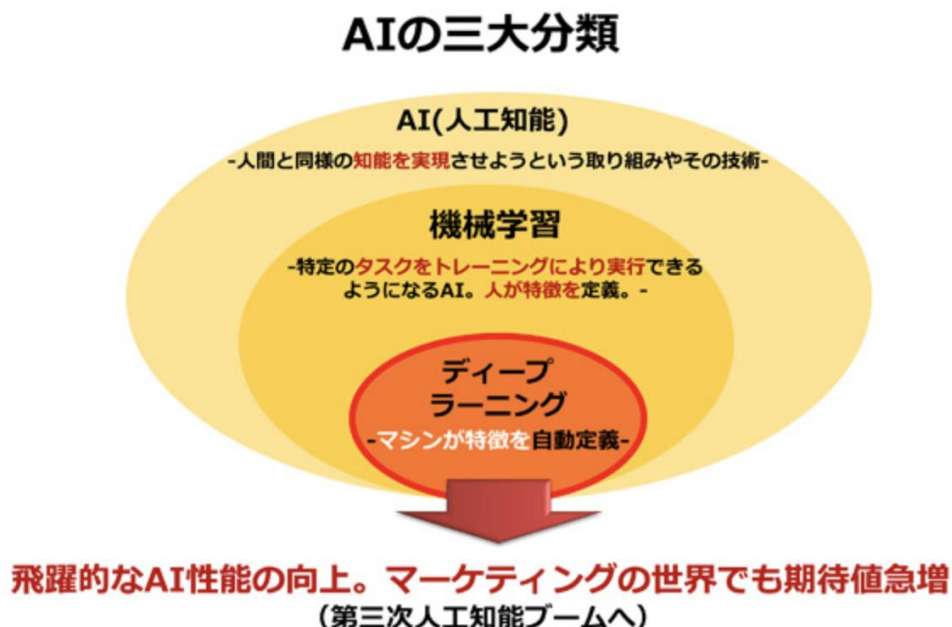


図4 機械学習・AI・ディープラーニング・ニューラルネットワークの違い

AI(人工知能)とは、機械に人間のような動作や判断をさせようとする試み全般を指す。人間の自然に行っている思考や行動などを、アルゴリズムや数値などで置き換えることで、機械が自動的に実行することを目指す。機械学習は、AI(人工知能)の中のひとつのカテゴリ。人が定義した特徴をもとに、トレーニングを重ねて機械が実行できるようになる仕組みになる。ニューラルネットワークは、機械学習の手法の一つの手法であり、人間の脳のニューロンを模倣したモデルを使用するものである。入力層、出力層、隠れ層の3層で構成されている。

図表4 ニューラルネットワークと層構造

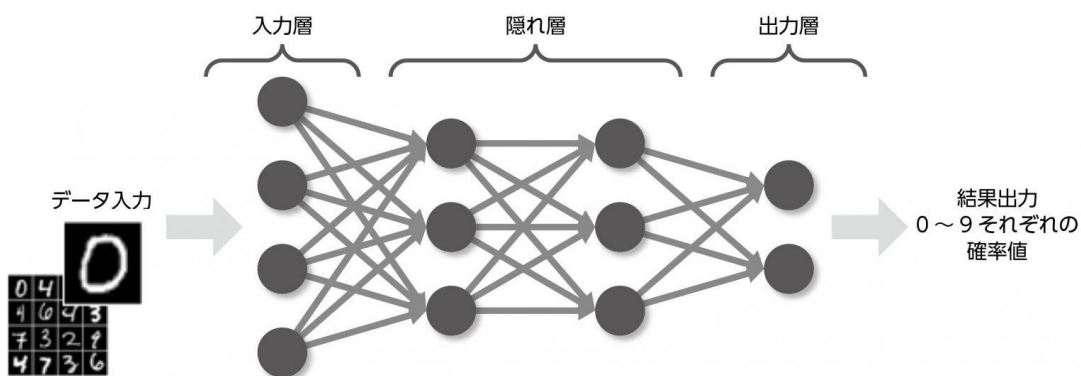


図5 ニューラルネットワークと層の構造

そして、ディープラーニングは、ニューラルネットワークの層を多層化したものを用いて、より高度な学習や分析を行えるようにしたアルゴリズムである。人間が答えを定義しなくてもコンピュータが自動でパターンの定義を行うことができ、現在ニューラルネットワークの主流として用いられている。[2]

2.3 教師あり学習

教師あり学習は、学習データに政界を与えた状態で学習させる手法である。学習データに正解を与えない「教師なし学習」と対をなす機械学習の学習方法となっている。教師あり学習では、トレーニングデータや教師データなどと呼ばれる正解となる学習データを利用する。つまり、正解・不正解が明快な問題の解決に利用できる学習手法なのである。例えば、システムの不正行為の検出や、おすすめ製品のパーソナライズなどを実現できる。教師あり学習で利用されるアルゴリズムとしては、回帰と分類が代表的な例であり、回帰は物価の予測や気象分析に利用され、分類は植物や動物などのカテゴリ化を実現するアルゴリズムである。教師あり学習は学習と認識・予測の2段階のプロセスで構成されており、このプロセスを実現するアルゴリズムとして回帰と分類が使用される。なお、ディープラーニング(深層学習)は基本的に教師あり学習を発展させたものである。

また、教師あり学習と教師なし学習は学習データに正解を与えるか否かという違いがありますが、正解が明確である場合には教師あり学習・不明確な場合には教師なし学習を

用いる。そのため、教師あり学習の方が学習精度は高く、基本的には教師あり学習が用いられる機会が多いと言える。

また教師あり学習では、電子メールのスパム判定、物価や住宅価格の予測、工業機械の故障予測、顧客の障害価値の判定、店舗の最適な人員配置の予測など様々な用途で活用できる。正解や最適とされる答えが名カウになる問題に関しては、教師あり学習は大きな効果を発揮する。主に予測や判定として活用される例が多いと言える。[3]

2.4 Softmax 関数

ソフトマックス関数とは、

$$f_i(x) = \frac{e^{x_i}}{\sum_{k=1}^n e^{x_k}}$$

で表される関数です。

ソフトマックス関数はディープラーニングの活性化関数の1つである。

N 個のデータがある時に、その合計を 1(100%)になるように調整してくれる関数である。3 個のデータの場合を例にグラフで表すとこのような形になる。

(x_1 は変動させ、x_2=1, x_3=2 に固定)

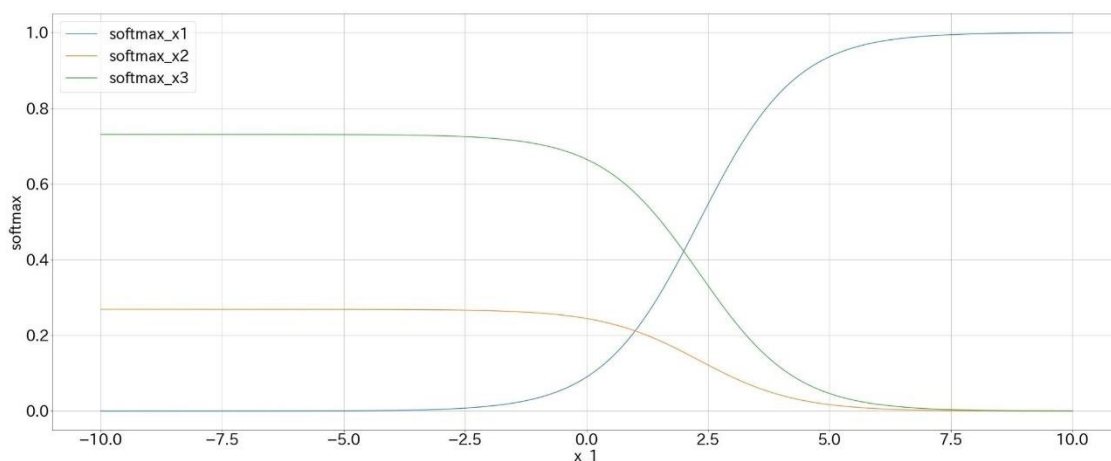


図6 ソフトマックス関数のグラフ

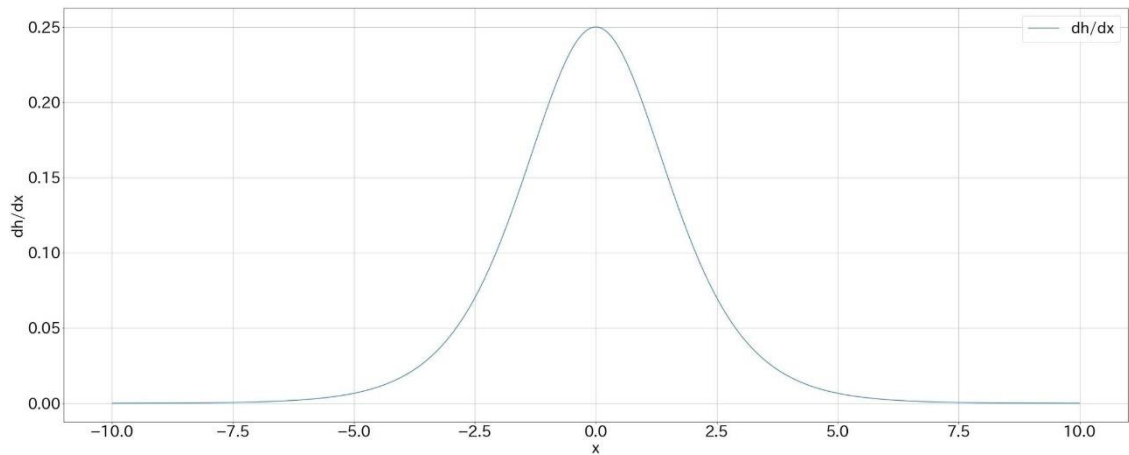
ソフトマックス関数の特徴は、上のグラフを見ると、ソフトマックス関数では 0 以上 1 以下の値で出力されていることが分かる。それぞれの出力データを合計すると 1(100%)となる。またデータが 2 個の場合にはシグモイド関数になる。

ソフトマックス関数は他クラス分類の出力層として利用される。出力層の合計は 1(100%)となるように調整されるため、そのまま結果(確率)として使用することができる。

ソフトマックス関数の微分

$$\frac{h_i}{dx_j} = \begin{cases} h_i(1 - h_j) & (i = j) \\ -h_i h_j & (i \neq j) \end{cases}$$

$i = j$ の時にはシグモイド関数の微分と同様の結果となる。 $i = j$ のときの微分後の関数をグラフで表すとこのような形になる。



<https://nisshingeppe.com/ai/>

図7 シグモイド関数のグラフ

[4]

第3章 提案手法

1.3 節の問題点で述べた通り、夜の自動運転ではごみが固まらないようにほぐしているだけで、人間のオペレータ勤務時間内に行っている作業はごみを燃えるように水分の多いごみと燃えやすいごみを混合して投入口から投入必要があれば燃焼を助けるために適切な分量の燃料を追加している。そのため、自動運転の方でもごみが燃えやすいように混合して投入口に持っていき、適切な分量の燃料を追加できるようにしたい。

本研究では、実現させたいシステム 3 つの機能が必要と考える。1 つ目はごみの量が多いところからたくさんごみを取って、ごみの種類ごとに溜まっている分量をできるだけ均等になるようにもやす分量を調節する機能。2 つ目は燃やす際にあまり追加の燃料を使いたくないため、燃えるように燃えにくいごみと燃えやすいごみをうまく混合することで燃料の追加を抑える機能。3 つ目は燃料が必要な場合、どれくらいの量が必要なのかを予測する機能。この 3 つの機能を持ったシステムが提供できると現在オペレータが勤務時間内で行っている作業と同じようなシステムを提案する。

追加する燃料を計算する際、式に必要なパラメータは季節によって動的に変動していて、毎日オペレータに聞きに行かないと燃えやすさのデータ分からないため大変である。そのため、オペレータの業務を妨げずに業務の負担を減らしたいと考えた。オペレータがどのような割合で混ぜているかのデータは自動的に取れるため、今までのごみを混ぜてきた割合をデータとして使うと上手くいく教師あり機械学習を用いる。

第4章 実験

本章では実験目的、実験概要、実験結果について述べる。

4.1 実験目的

本研究では「生ごみ」、「紙ごみ」、「プラスチックごみ」3種類のごみを扱う。3種類のごみの量に応じてどれくらい燃料が必要かということデータを正しく学習して、人間と同じような業務ができているかを確認するためにニューラルネットワークで実験をする。

第3章で述べた、3つ目の提案システムのみを実験する。なぜなら、3つ目の提案システムだけは燃料を追加する際はその分コストがかかるため、それを優先して実験することにした。

4.2 実験概要

プログラムの実験環境は Google Colaboratory で教師あり機械学習を行った。まず学習データは本来、オペレータからのデータが必要なのだが今回は自分でモデルを作成し学習データをモデルに従って合成した学習データ、教師データ、入力データ、出力データを表1に示す。

表1 実験データ

教師データ	燃やすために必要な燃料の計算結果
入力データ	全体のごみの率が1として、3種類のごみの0から1の比率
出力データ	投入する燃料

ごみ全体での燃えやすさを f 、生ごみ、紙ごみ、プラスチックごみそれぞれの燃えやすさ係数を m_1 、 m_2 、 m_3 として必要な燃料を求める時の式を以下の i)、ii)に示す。

$$i) f = \text{生ごみ} \times m_1 + \text{紙ごみ} \times m_2 + \text{プラスチックごみ} \times m_3$$

$$ii) \text{必要な燃料} = \text{燃える基準} - f \text{ [付録 1]}$$

学習条件はミニバッチ数は32、エポック数は80とする。入力データは5151通りで、そこからランダムに32個のデータを取り出して学習を80回繰り返す。検証には5151件全データを用いて損失を計算する。損失関数には平均二乗誤差を使う。

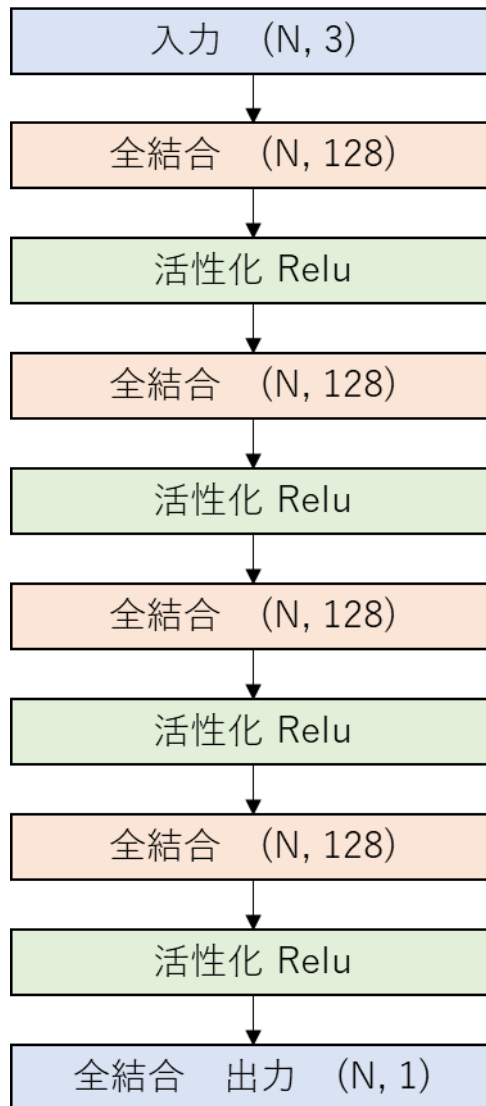


図 8 ニューラルネットワークの構造

4.3 実験結果

燃えやすさ係数 $[m_1, m_2, m_3]=[1, 10, 7]$ の時、燃える基準を 8 と 16 の時の損失の変化を比較したのを図 8、9 に示す。図 8 は学習回数 40 回を境目に収束しており、図 9 は学習回数 50 回を境目に 0 に収束していることが分かる。

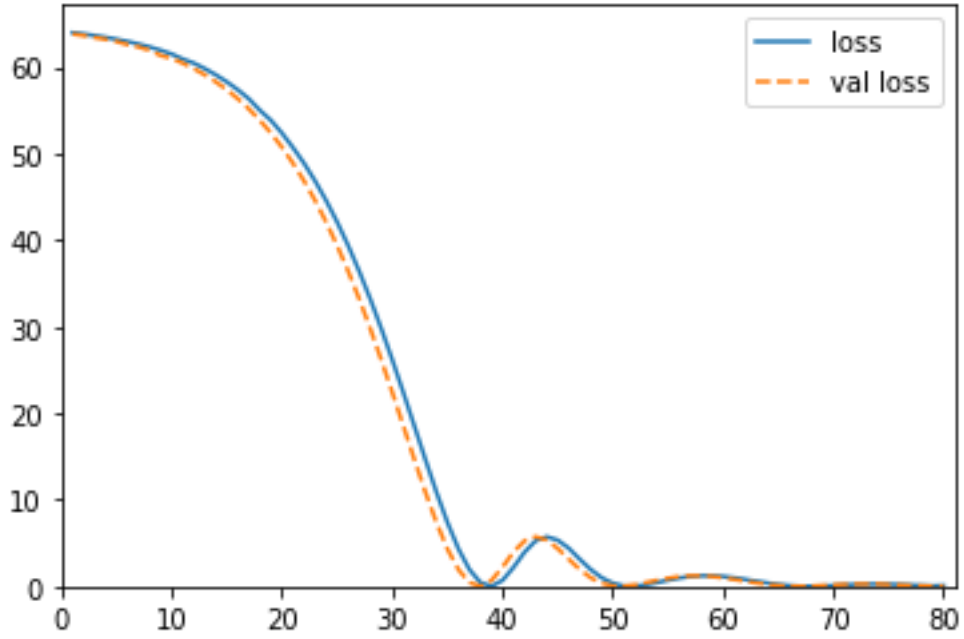


図 9 燃える基準 8 の時の損失の変化

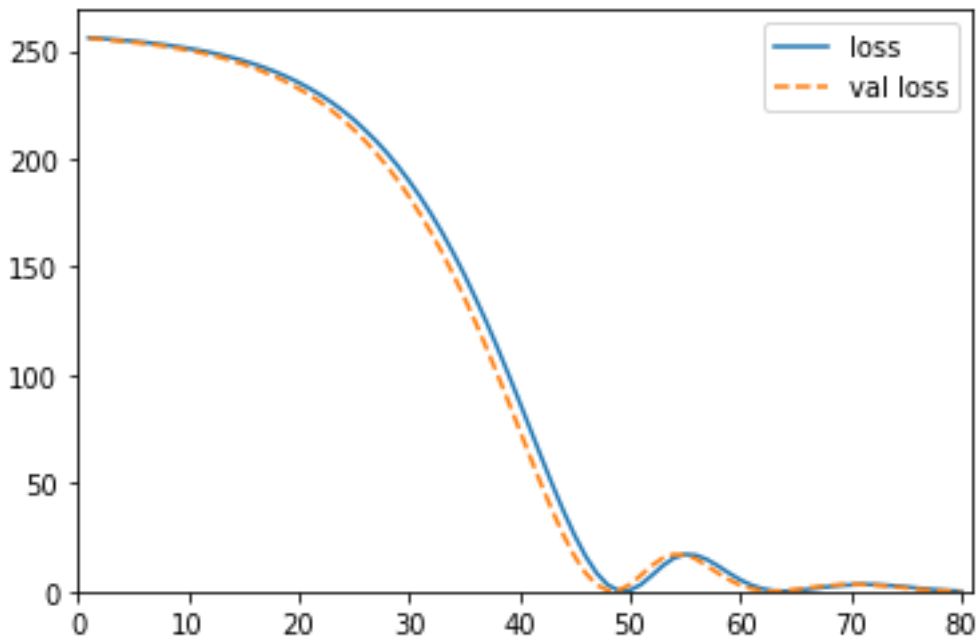


図 10 燃える基準 16 の時の損失の変化

燃える基準を 8 の時、燃えやすさ係数[2, 20, 14]と[3, 20, 21]の時損失の変化を比較したのを図 10、11 に示す。学習回数 40 回を境目に 0 に収束していることが分かる。

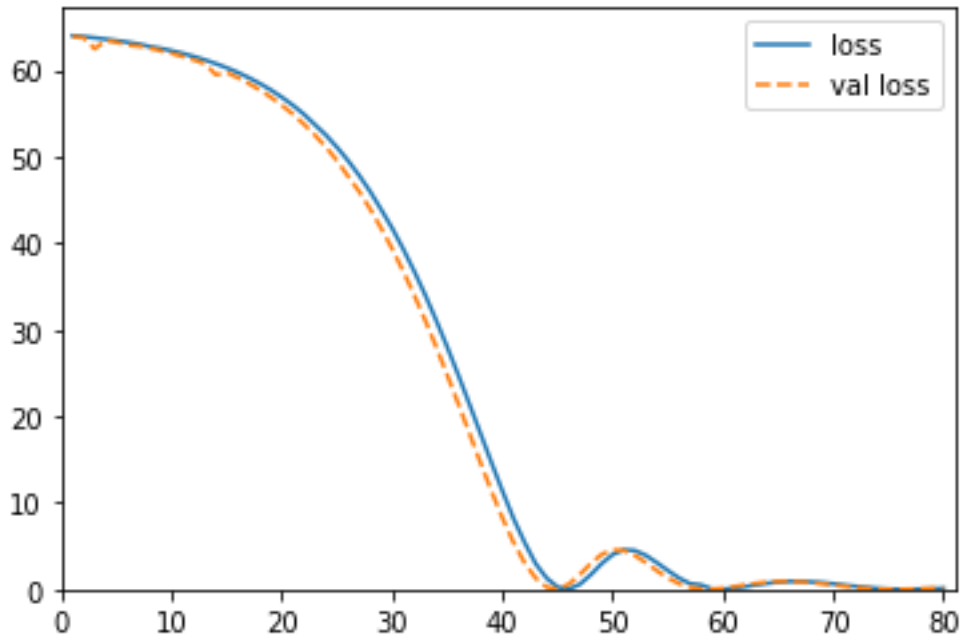


図 11 燃えやすさ係数[2, 20, 14]の時の損失の変化

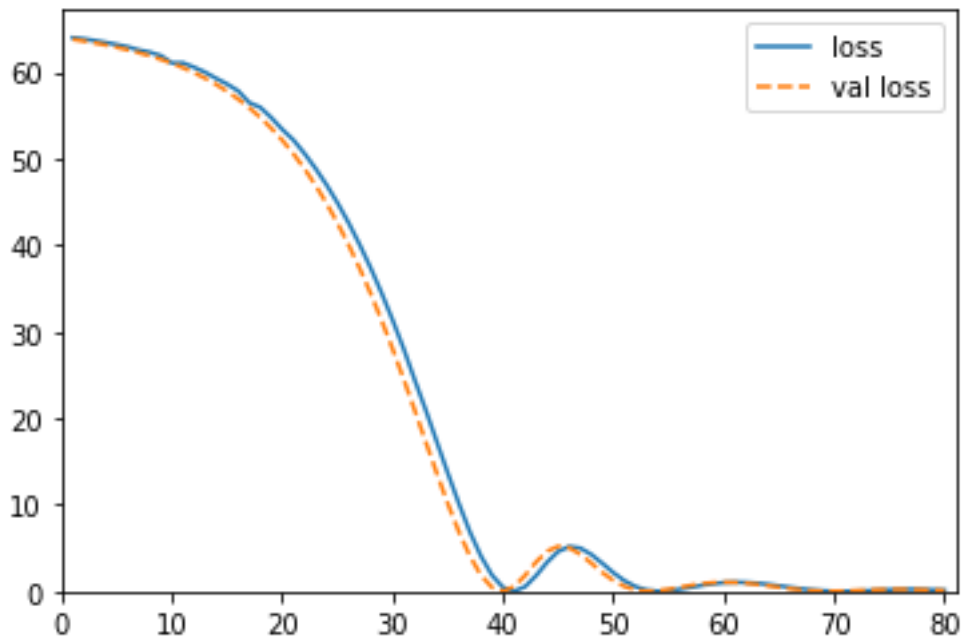


図 12 燃えやすさ係数[3, 20, 21]の時の損失の変化

燃えやすさ係数と燃える基準両方の数字を変えて行った損失の変化を図 12 に示す。学習回数 50 回を境目に 0 に収束していることが分かる。

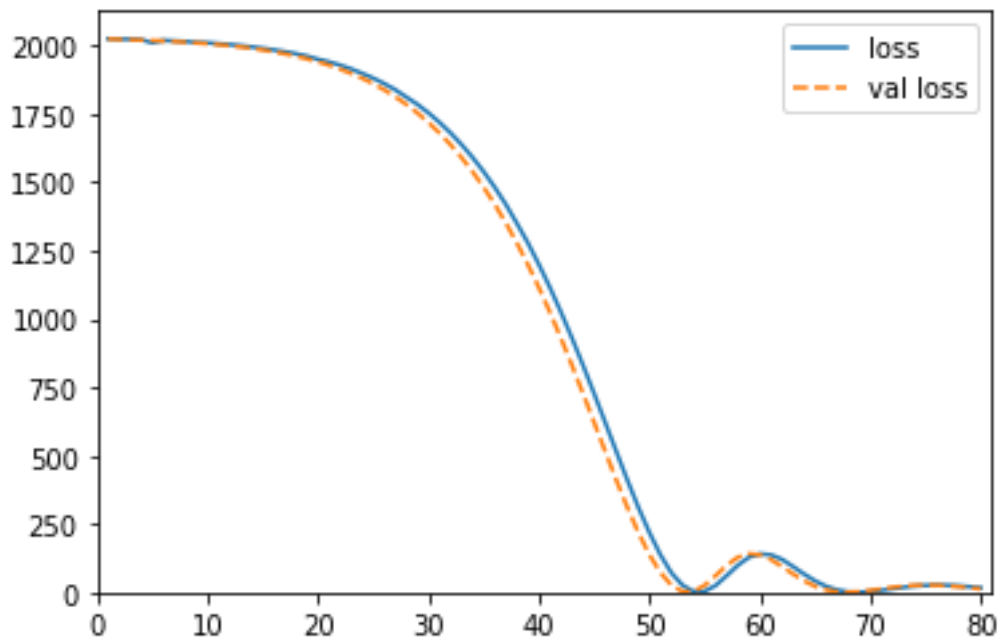


図 13 燃えやすさ係数[4, 40, 28]、燃える基準 45 の時の損失の変化

第5章 考察

燃えやすさ係数を固定して燃える基準を変えて行った実験、燃える基準を固定して燃えやすさ係数を変えて行った実験、燃えやすさ係数も燃える基準どちらも変えて行った実験は、全てにおいて損失が 0 に収束している。検証用データの損失 `val_loss` においても同様に収束が見られた。本研究では実験の条件を変えながら学習しており、全てにおいて 0 に収束しているため、また実際にごみの比率が変わった時もこの本研究で使用したプログラムで代用できると考える。

第6章 まとめ

ごみの量が多いところからたくさんごみを取って、ごみの種類ごとに溜まっている分量をできるだけ均等になるようにもやす分量を調節し、燃やす際にあまり追加の燃料を使いたくないため、燃えるように燃えにくいごみと燃えやすいごみを合わせ、燃料が必要な場合、どれだけの量が必要なのかを出力するシステムを提案した。提案手法のシステムとしてニューラルネットワークを用いて3つのごみの量に応じてどれだけの燃料が必要となるのかを計算するプログラムを実験した。実験結果より、複数の条件下においても利用できることが確認できた。本研究では、ごみの体積を一定として入力としてごみの比率を変化させたため、必要な燃料を求めるためには計算結果にごみの体積を掛ける必要がある。今後の課題としては、システム全体を構築し実用化することを検討する。

謝辞

本論文を作成するにあたり、多くのご指導、ご助言をいただきました三好力教授に深く感謝の意を表します。また京都市南部クリーンセンターのさすてな京都の皆様、インタビュー、見学にご協力ありがとうございました。

参考文献

[1]ごみ問題の現状は？原因や対策から私たちにできることを考えよう！

<https://naruhodosdgs.jp/garbage/>

[2]機械学習とは？できることや事例を初心者向けに分かりやすく解説 | 侍エンジニア編集部

<https://www.sejuku.net/blog/11595>

[3]教師あり学習とは？覚えておきたい機械学習の学習手法概要 | コラム | クラウドソリューション

<https://business.ntt-east.co.jp/content/cloudsolution/column-161.html>

[4]ソフトマックス関数(softmax 関数)とは？機械学習の視点で分かりやすく解説!!

<https://nisshingeppo.com/ai/softmax-function/>

[5]可視化 - Keras Documentation

<https://keras.io/ja/visualization/>

インタビュー、見学場所:京都市南部クリーンセンター さすてな京都(インタビュー、見学をした日にち:2021年10月21日)

付録 1

```
import random
import numpy as np
import tensorflow
from tensorflow.keras.layers import Input, Dense
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import plot_model
import matplotlib.pyplot as plt

def softmax(x):
    exp = np.exp(x)
    return exp / np.sum(exp)

def create_outputs(inputs, factor, condition):
    L = []
    for input_data in inputs:
        #ごみ全体での燃えやすさ
        #input_data["生ごみ", "紙ごみ", "プラごみ"]
        fuel = input_data[0] * factor[0] + input_data[1]
        * factor[1] + input_data[2] * factor[2]

        #最大が0か自分で決めた値の大きい方の値を取る
        fuel = max(0, condition - fuel)
        L.append([fuel])

    return np.array(L)

model = Sequential()

#入力層
model.add(Input((3,)))

#中間層
model.add(Dense(128, activation='relu'))
model.add(Dense(128, activation='relu'))
model.add(Dense(128, activation='relu'))
model.add(Dense(128, activation='relu'))

#出力層
model.add(Dense(1, activation='linear'))
print(model.summary())
plot_model(model, to_file="model.png",
show_shapes=True, show_layer_names=False)

#平均二乗誤差 (mse)
model.compile(tensorflow.keras.optimizers.Adam(),
loss="mse")

#学習

#配列の初期化(グラフを作るために値を1つ1つ残しておく)
losses = []
val_losses = []

#50回学習でfor文で繰り返し
epochs = 80

garbage = [[i, j-i, 100-j] for i in range(101) for j
in range(i, 101)]
garbage = softmax(np.array(garbage))

condition = 45

#燃える基準に近づけるようにしている
print(garbage.shape[0])

for i in range(1, epochs+1):

    #ランダムなデータを作成
    index = np.random.randint(0, garbage.shape[0],
(32,))

    #ランダムな入力データの時にどれだけ燃料があるのか
    を y とする
    x = garbage[index]
    y = create_outputs(x, factor, condition)

    #学習する
    loss = model.train_on_batch(x, y)

    #学習をしない evaluate
    # index = np.random.randint(0, garbage.shape[0],
(32,))
    # val_x = garbage[index]
    # val_y = create_outputs(val_x, factor, condition)
    val_y = create_outputs(garbage, factor, condition)

    #x と y を使って評価する、評価する時の loss を出して
    くれる
    val_loss = model.evaluate(garbage, val_y,
verbose=0)
    print("epoch: {} loss: {:.4f}
val_loss: {:.4f}".format(i, loss, val_loss))

    #評価する時に出てきた loss, val_loss を配列 loss,
val_losses に追加
    losses.append(loss)
    val_losses.append(val_loss)

#グラフ作成
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.plot(range(1, epochs+1), losses, label="loss",
linestyle="solid")
ax.plot(range(1, epochs+1), val_losses, label="val
loss", linestyle="dashed")
ax.set_xlim(0, epochs+1)
ax.set_ylim(bottom=0)
ax.legend()
fig.show()
```