

# サイコロの出目予測



チーム名: 高高Alea 群馬県立高崎高等学校2年 黒澤駿 常見健太

## 研究の背景

この研究に取り組みようと考えたきっかけは、ギャンブルなどのディーラーがサイコロをふるときにどの目を出すかを考えて、狙った目を出しているという話を聞き、そのようなことは可能なかと疑問に思ったことだ。

先行研究として、綿を詰めた布製のサイコロの中に、M5stick core2というセンサーをいれ、そのサイコロが投げられてから静止するまでの加速度、角速度を調べ、出る目を予測するという研究があったが、この研究では**サイコロの出る目を予測することは失敗に終わった**。

## 予備実験

まず、発泡スチロールを1辺が6cmの立方体に切り取り、中に「M5stickCPlus」というセンサーをいれて、約1mの高さから、不規則に投げ、サイコロが投げられてから静止するまでのデータを「TensorFlow」という機械学習モデルを利用して学習させ、どの程度の確率で出た目を当てられるかを調べた。



しかし、学習させたモデルをテストしたところ、正解できた数は0個であり、サイコロの出る目を予測することは完全に失敗した。

## 実験

予備実験から、始めから6面のサイコロを用いて6通りの結果が出る実験で、さらにあまりにも不規則に投げてしまうと、サイコロの出る目を予測することができなかった理由を把握することが難しかったため、下記の2つの実験を行った。

### 方法

実験1では3Dプリンターで図①のようなサイコロを印刷し、中に「M5stickCPlus」を入れ、サイコロでは一辺を地面につけ、約45度に傾かせた状態で手を放して出目が2通りになるようにして実験を行った。実験2では空中から落とした場合にも出目を2通りにするためにコインを模したものを印刷し、衝撃があまり大きくない緩衝材を3枚重ねて敷いて地面との距離約10cmから縦向きで静かに落とした。その時の手を離れた瞬間から静止したときまでの加速度、角速度を記録し、そのうち静止したことが判断された列のデータを含まずにその上の10列を学習データ、出た目をラベルデータとした。そのデータを、「TensorFlow」という機械学習モデルを利用し時系列性は持たせず、特徴量として学習させ、そのモデルをテストして正解率を算出した。



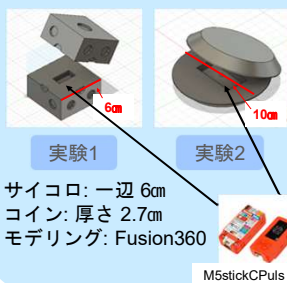
- 学習はgoogle collaboratoryを用いて行った。
- TensorFlowのkerasというライブラリを利用した。

データは下のようにして扱った

x	y	z	wX	wY	wZ	
1	-0.77	0.25	0.02	-10.47	-85.64	-50.88
2	-0.84	0.25	0.04	-15.29	-94.3	-49.42
3	-0.51	0.19	0.27	-178.19	-641.3	-17.8
4	-0.51	0.19	0.27	-178.19	-641.3	-17.8
5	0.02	-0.04	0.35	120.76	42.84	-60.65
6	0.02	-0.04	0.35	120.76	42.84	-60.65
7	0.01	-0.02	0.43	118.99	14.03	-56.56
8	0.05	0.03	1.28	30.61	-5.5	-15.48
9	-0.02	0.06	1.13	28.72	-0.61	-16.64
10	-0.02	0.06	1.13	28.72	-0.61	-16.64

10列目のデータは静止した直前のデータを表している

### モデリング



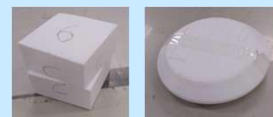
サイコロ: 一辺 6cm  
コイン: 厚さ 2.7mm  
モデリング: Fusion360

M5stickCPlus



FLASHFORGE  
Adventurer4

### 完成品



- 実験1
- 実験2
- 充填率: 15%
- 内部の充填パターン: 六角形
- 樹脂: Flashforge-PLA

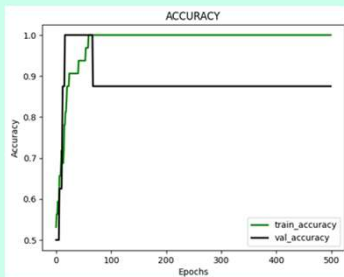
## 結果

### 最終的な正解率(1.00が最大)

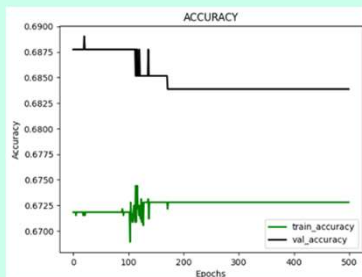
- 1辺を接地して倒した「サイコロ」: **0.80**
- 10cmの高さから落とした「コイン」: **0.63**

正解率 = 正解数 / 全データ数  
(出目は2通りのため、期待値は0.50)

- 訓練データに対する精度
- 未知データに関する精度

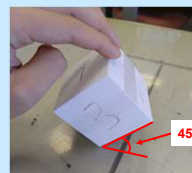


サイコロの学習曲線

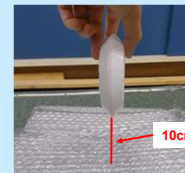


コインの学習曲線

## 実験の様子



実験1



実験2

サイコロを倒したり、コインを落下させたりした後は、センサーの値を確認し、静止したことを確認してから約3秒待ち、次の試行を行った。

## センシング



M5stickCPlus



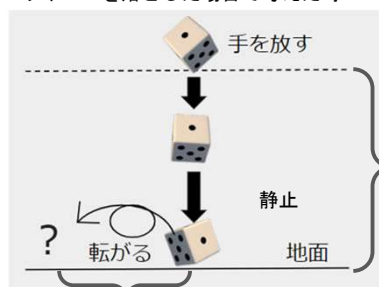
- arduinoIDEを利用して、M5stickCPlusのコーディングを行っている
- キャリブレーション(センサー校正)を行い、誤差を小さくしている。

## 考察

上の学習曲線から二つの実験では、共に訓練データに対する精度と未知のデータに対する精度は直線になっているため、正確な正解率が求められていると考えられる。

また、コインを落とす実験の方がサイコロを倒す実験に比べて正解率が下がっている。これは、コインの場合には空中で手を放してから、静止するまでのデータで学習しており、サイコロの場合は地面で接地した状態から倒れて静止するまでのデータのみで学習していることが影響していると考えられる。

### サイコロを落とした場合で考えた時



サイコロの実験で用いたデータ

コインの実験で用いたデータ

## 今後の展望

- 手を放してから静止するまでのデータのうちのどの部分を学習させると正解率が高くなるかを調べる。
- LSTMなどの時系列データを扱えるモデルについても検討していきたい。

## 参考文献

1) Interface Pythonと実データで未来予測！ データサイエンス入門, CQ出版社, 令和4年5月1日発行, p29~p46