

# サイコロの出目を予測するには

黒澤俊

群馬県立高崎高等学校

## 要旨

サイコロの飛び方には出目と関係があり、飛んでいる最中に目を出目も予測することも可能であると考えた。そこで、そのようなことは可能なのかを調べる実験を行った。サイコロの中にジャイロセンサーを入れ、収集したデータに対して、出目を教師データとして機械学習を行い、予測精度を調べた。正六面体のサイコロを地面から離れた状態で投げた場合の予測はできなかったが、条件を絞って、サイコロを地面に設置して倒した場合や、2面のみがでるコインではある程度の予測ができることがわかった。

## 1. はじめに

### 1.1. 研究全体の目的

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がある。もしそうであれば、サイコロの飛び方には出目と関係があり、飛んでいる最中に目を出目も予測することも可能である。本研究の目的は、サイコロの出目をサイコロが飛んでいる最中に高い精度で予測することである。

### 1.2. 研究の仮説

サイコロの中にセンサーを入れ、3軸の角速度、加速度を時間ごとに取得し、そのデータと機械学習モデルを利用して予測が可能なのではないかと考えた。

## 2. 検証実験

### 2.1. 実験1

#### 2.1.1. 仮説

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<sup>1)</sup>があったが、この研究ではサイコロの出目を予測することは失敗に終わった。わたしはこの研究の問題点は、センサーを綿の中に入れてことによって、サイコロの中でセンサーが揺れ動き、サイコロ自身の運動をデータに反映できなかったことだと考えた。そこで、サイコロを発泡スチロール製にし、センサーを安定させることによって予測が可能になると仮説を立てた。

#### 2.1.2. 方法

発泡スチロールを一辺の長さが6.0cmになるように切り出し、中心にジャイロセンサーの機能を持ったM5stickCPlusを入れられるようピッタリと穴を開けた。センサーを入れ、テープで切れ目と垂直に巻き、図1のように正六面体のサイコロを作成した。周囲に障害物のない開けた場所でこのサイコロを30回投げ、それぞれで出た目と手を離れてから地面で静止する直前までのデータを結び付けた。このデータをGoogleColabを用いてLSTMで学習させ、検証データを用いて正解率を調べた。



図1 正六面体のサイコロ

### 2.1.3. 結果

検証データ20件に対して正解率0.0であった。

### 2.1.4. 考察

結果から、機械学習を行ったモデルはまったく予測ができるものではなく、失敗したといえる。正六面体では1, 2, 3, 4, 5, 6の6通りの目が出るため、最初に挑戦するには複雑すぎて、データ数の不足やジャイロセンサーの測定時間間隔 (FPS) の不足等の要因から上手くいかなかったと考えられる。

## 2.2. 実験2

### 2.2.1. 仮説

実験1では条件が複雑すぎてうまくいかなかったと考え、1, 2の2通りの目の出方で実験を行い、また、少しずつ事象の複雑さを上げていことで、何が問題で予測がうまく行っていないのかを調べられると考えた。そこで、よりシンプルな条件である地面からサイコロを離さずに行う実験と、コインを空中から落とす実験を行えば、予測が可能であるという仮説を立てた。

### 2.2.2. 方法

#### ①実験器具の用意

図2のように、中にセンサーを入れるサイコロ、コインは、3DプリンターFLASHFORGE Adventurer4を用いて作成した。樹脂はFlashforge-PLAを使用した。また、モデリング (設計) はFusion360を利用して行った。サイコロの一边は6.0cm、コインは厚さは2.7cm、中心の直径10cm、外側の円の直径8cmのものを作成した。



図2 モデリングと3Dプリンターによる印刷

## ②データ収集

サイコロでは、サイコロの1, 2の面のあいだの一边を地面につけ、5, 6の面のあいだの一边を指で押さえて固定し、角度を $45^\circ$ に固定し、指を離して1, 2のどちらかの面が下になり静止するまでのデータを収集した。コインでは、コインが地面に衝突した際に大きくはねてしまうのを防ぐために、緩衝材を敷き、高さ約10cmまで手で持って持ち上げたあとに手を離し落とした。手から離れたときから地面に転がって表か裏の面が上面になって静止するまでのデータを収集した。

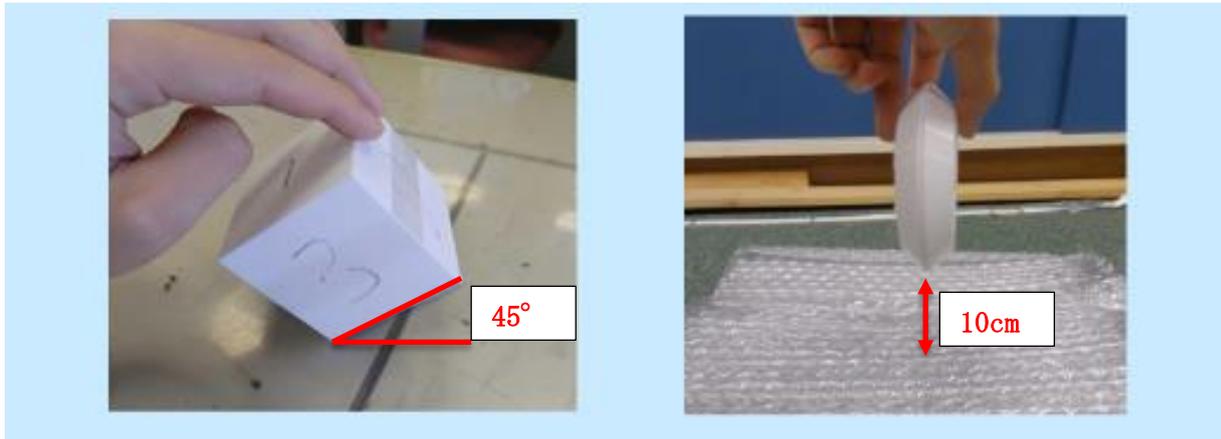


図3 実験の様子

## ③機械学習

ブラウザからpythonを実行できるサービスのgoogle colaboryを用いてTensorFlowのkerasというライブラリを利用した。学習させるデータは、収集したデータでサイコロ、コインが静止した時点から、10件のデータとその際に出た目を一つのデータとして学習を行った。サイコロ、コインが静止したという判定は3軸加速度、角速度の両方が一つ前のデータと全く同じになっているかどうかを見て判断した。図4のデータ例では、10列目のデータが静止した段階でのデータになっている。

	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

図4 収集したデータの例

### 2.2.3. 結果

図5にそれぞれのデータについて行った機械学習の学習曲線を示す。

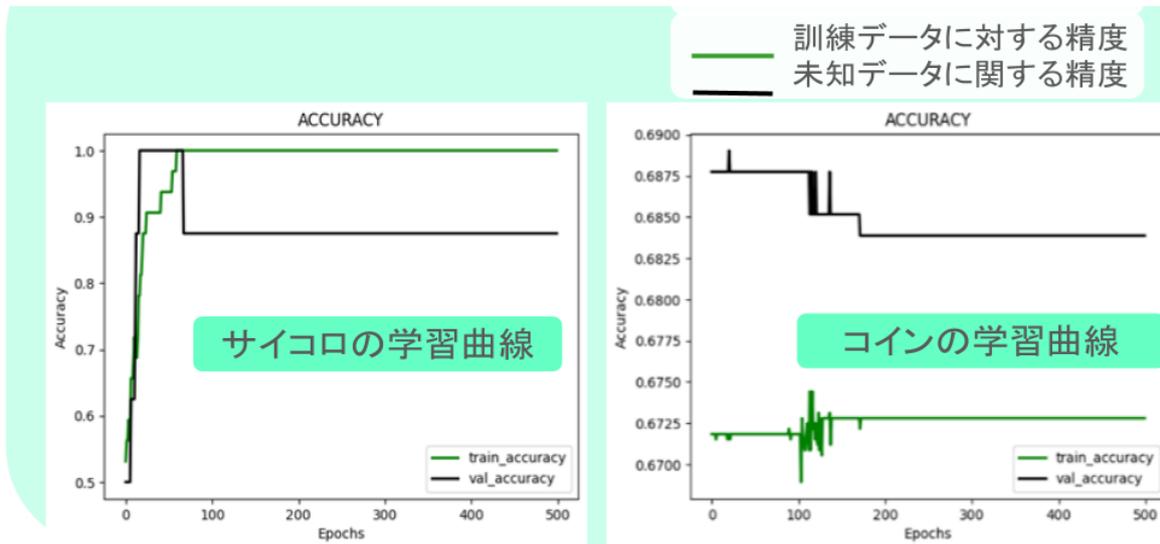


図5 機械学習の学習曲線

それぞれの学習が終了したモデルにおいて、検証用のデータを用いて調べた結果、以下の表1のようになった。

表1

	正解率(%)
一辺を接地させて倒したサイコロ	80
10cmの高さから落としたコイン	63

### 2.2.4 . 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。

サイコロの学習曲線を見ると、トレーニング精度は初期のエポック数で急速に上昇していることがわかる。これは、モデルがトレーニングデータに対して非常に早く適応していることを示している。また、トレーニング精度と検証精度がともに高く、特にトレーニング精度が1.0に達していることから、モデルがトレーニングデータに過度にフィットしている可能性がある。しかし、検証精度も高いため、一般化能力もある程度維持されていると考えられる。

コインの学習曲線では、トレーニング精度はエポック数に応じて小さな変動を繰り返しており、0.67付近で安定している。これは、モデルがトレーニングデータに対して十分な適応を示していないことを示唆している。低いトレーニング精度と検証精度の近さから、バイアスが強くバリエーションが低い状態（アンダーフィッティング）であることが示唆される。モデルがデータの複雑なパターンを捉えきれていないため、全体的なパフォーマンスが低くなっ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。

### 3. まとめ

#### 3.1. 結論

実験から、正6面体のサイコロを地面から離れた状態で投げた場合の予測はできなかったが、条件を絞って、サイコロを地面に設置して倒した場合や、2面のみがでるコインではある程度の予測ができることがわかった。サイコロの予測は倒すだけのときよりも、落とすという事象を加えただけで、大幅に難しくなるということがわかった。

#### 3.1. 今後の展望

今回の実験では、センサーから得たデータに一部不正確なものが含まれていた。具体的には、同じ値がいくつかつながってしまうといったもので、原因はセンサーの時間分解能の不足が考えられる。今後は、センサーを時間分解能にすぐれたものに変え、データ自体を正確なものにすることを考えている。

#### 謝辞

本校物理科教員である岡田直之先生に指導・助言をいただきました。

#### 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