

**令和 3 年度 卒業研究**

**AI によるゲーム音楽のリアルタイム自動生成**

**Real-time automatic generation of game music by AI**

**函館工業高等専門学校**

**生産システム工学科 情報コース 5年**

**竹谷大地**

**指導教員 東海林智也**

# 目次

第 1 章 はじめに	2
1.1 概要 (Abstract).....	2
1.2 背景 .....	2
1.3 目的 .....	2
第 2 章 使用技術	3
2.1 開発環境 .....	3
2.2 Keras.....	3
第 3 章 提案手法	4
3.1 ニューラルネットワークの仕様.....	4
第 4 章 実験	5
4.1 実験概要 .....	5
4.2 結果と考察 .....	5
第 5 章 まとめ	7
参考文献	8

# 1 はじめに

## 1.1 概要 (Abstract)

The goal of this research is to devise a way to automatically generate game music with AI in real time. Ultimately, we would like to find a way to add game music to user-generated game screens. This is because I thought that there are many people who are interested in composing music, even those who can create games but not game music. In this research, I mainly used deep learning using Keras, which can learn relationships with images. As a result, we were able to discriminate between the two types of screens and successfully sorted them into two different scenes. However, the aforementioned final goal has not been achieved.

Key words: AI, Game music, Real time

## 1.2 背景

今日までゲームは作れるが、ゲーム音楽を作ることが出来ない人を対象に自動で作曲する研究やアルゴリズムが研究されてきた。しかし、似たような曲が出来てしまったり、ゲームジャンルによってはゲームの場面の雰囲気合わない曲が出来てしまったりなどの課題点があった。

## 1.3 目的

当研究の最終的な目的は、ゲームの場面を判別してその場面に合ったゲーム音楽をリアルタイムで自動生成することであるが、まずはその前段階として複数のゲーム画面(今回は戦闘画面とフィールド画面の2パターンとした)を学習させ、それらのデータを元に戦闘画面かフィールド画面かを判別するニューラルネットワーク[1]を構築することにした。

## 2 使用技術

### 2.1 開発環境

使用言語： Python 3.7 (Spyder)

使用ライブラリ： Keras[2], numpy[3], tensorflow[4]

入出力： CSV ファイル

学習データ： 画像データ 16 枚

テストデータ： 画像データ 12 枚

### 2.2 Keras

Keras は Python で書かれたオープンソースニューラルネットワークライブラリである [2]。ディープニューラルネットワークを用いた迅速な実験を可能にするよう設計され、最小限、モジュール式であることから、このライブラリを使用するに至った。

### 3 提案手法

#### 3.1 ニューラルネットワークの仕様

当研究では入力された画像が戦闘画面かフィールド画面かの判別という機械学習を行うニューラルネットワークを構築する(図1)。

入力データを作成するために、まず画像データに対して RGB ごとに得られる 256 段階の濃度値データを 32 段階のヒストグラムデータに圧縮する[5]。それと同時に平均値も RGB ごとに計算する。この様にして得られた画像の RGB ごとの 96(=32×3)個のヒストグラムと 3 個の平均値を合わせ計 99 個の数値データを入力データとした。よって入力層のユニット数は 99 個となる。

ラベルは入力によって得られたデータから戦闘画面かフィールド画面かを判別するため戦闘画面を 1、フィールド画面を 0 とする one-hot ベクトル形式とした。よって出力層のユニット数は 2 個となる。中間層のユニット数は 100 個にして実験を行った。更に Dropout 層を追加した。学習率は 0.001 とエポック数は 1000 とした。

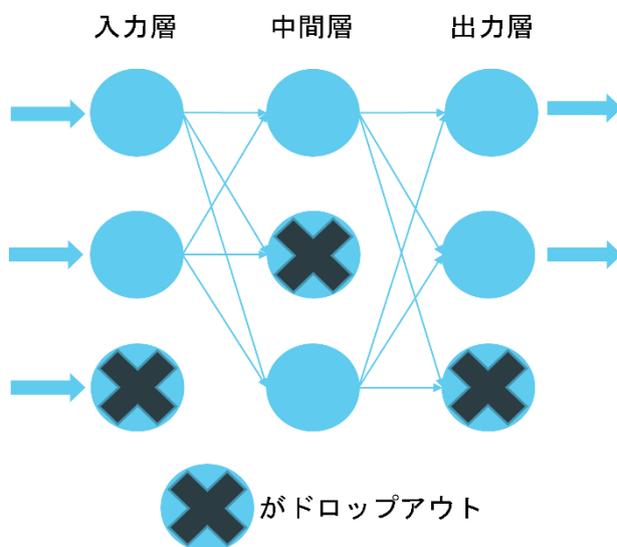


図1 ニューラルネットワーク構成図

## 4 実験

### 4.1 実験概要

今回の実験では、8枚の戦闘画面の画像と8枚のフィールド画面の画像の計16枚の画像を学習データセットとした。またテスト用のデータとして戦闘画面の画像とフィールド画面の画像、6枚ずつをテストデータセットとした。これらのデータセットを元にニューラルネットワークを用いて戦闘画面かフィールド画面かの判別を行った。

### 4.2 結果と考察

学習データセットの学習前と学習後の正解率、さらにテストデータセットの正当率を評価した。目安としては、50%ほどがランダムであり、75%以上が安定して続けば学習が出来ている基準となり、90%以上のほぼミスが無い状態になると正確に判別されたという指標になる。

学習データセットに関しては、学習前よりも正解率は平均的に上がっていたのでうまく学習出来ていることが分かる(図2、図3)。

またテストデータセットでも正解率は75~99%となっていた(図4)。よって学習前よりも学習後の正解率が上がったことから学習に効果が表れていることが分かった。

```
学習前
学習データセット : 損失=0.87, 正解率=0.56
予測値
[[ 0.326  0.674]
 [ 0.016  0.984]
 [ 0.392  0.608]
 [ 0.582  0.418]
 [ 0.602  0.398]
 [ 0.208  0.792]
 [ 0.261  0.739]
 [ 0.231  0.769]
 [ 0.679  0.321]
 [ 0.091  0.909]
 [ 0.116  0.884]
 [ 0.027  0.973]
 [ 0.146  0.854]
 [ 0.461  0.539]
 [ 0.102  0.898]
 [ 0.085  0.915]]
```

図2 学習データセットの学習前の正解率

```

学習後
学習データセット : 損失=0.00, 正解率=1.00
予測値
[[ 0.999 0.001]
 [ 1.000 0.000]
 [ 0.997 0.003]
 [ 1.000 0.000]
 [ 0.998 0.002]
 [ 0.999 0.001]
 [ 0.997 0.003]
 [ 0.998 0.002]
 [ 0.001 0.999]
 [ 0.001 0.999]
 [ 0.005 0.995]
 [ 0.001 0.999]
 [ 0.000 1.000]
 [ 0.000 1.000]
 [ 0.001 0.999]
 [ 0.000 1.000]]

```

図3 学習データセットの学習後の正解率

```

テストデータセット : 損失=0.25, 正解率=0.92
予測値
[[ 0.921 0.079]
 [ 0.963 0.037]
 [ 0.066 0.934]
 [ 0.998 0.002]
 [ 1.000 0.000]
 [ 0.999 0.001]
 [ 0.148 0.852]
 [ 0.002 0.998]
 [ 0.000 1.000]
 [ 0.020 0.980]
 [ 0.000 1.000]
 [ 0.000 1.000]]

```

図4 テストデータセットの正解率

## 5 まとめ

最終目標であるゲーム音楽のリアルタイム自動生成を行うために、当研究ではその前段階として戦闘画面かフィールド画面かを判別しその正解率を出力するニューラルネットワークを構築した。

今後は構築したネットワークを元に実際に音楽をゲームに付ける研究を行う。具体的には、戦闘画面と判別された場面には戦闘用のBGMを、フィールド画面にはフィールド用のBGMをそれぞれ合わせて音楽を自動生成する。

## 参考文献

[1] ニューラルネットワーク:

<https://atmarkit.itmedia.co.jp/ait/articles/1811/20/news012.html> (最終参考 2022 年 2 月)

[2] Keras: <https://keras.io/ja/> (最終参考 2022 年 2 月)

[3] numpy: <https://numpy.org/> (最終参考 2022 年 2 月)

[4] tensorflow: <https://www.tensorflow.org/?hl=ja> (最終参考 2022 年 2 月)

[5] ヒストグラム: <https://tomomai.com/python-histogram/> (最終参考 2022 年 2 月)