

# 深層学習を用いた動画のタイトルからタグを予測する 自動ジャンル分け機能

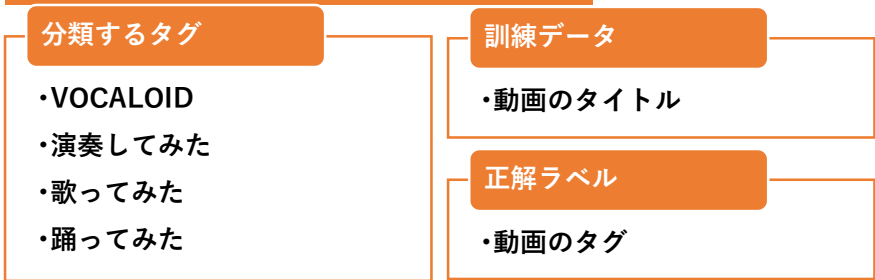
佐口 航（日本工学院専門学校）

※以下の内容の著作権は佐口航に帰属します。

## 目的

- 従来よりも視聴者のニーズに合う動画を表示する
- 投稿されたばかりの動画が視聴者に再生されやすくする
- 共通するメジャーなタグを登録する手間を省く

## 訓練データと正解ラベル



今回は、図1のように"VOCALOID", "演奏してみた", "歌ってみた", "踊ってみた"の4種類のタグに絞って分類しました。深層学習に使用する訓練データが「動画のタイトル」で、正解ラベルが「動画のタグ」です。対象は、2007年03月06日から2018年11月08日までに投稿された16,703,325件の動画のうち、上記の4種類のタグが1つ以上登録されている動画です。2つ以上のタグが登録されている場合は、動画に登録されているタグのうち先頭に近いタグを正解ラベルとして認識するようにプログラムを作成しました。

図1 概要

## 開発環境

- Python 3.8
- Anaconda3 (2020.07)
- Spyder 4.1.4
- TensorFlow (GPU) 2.0.0 (Keras を GPU で動作させるため)
- Keras 2.3.1
- NVIDIA cuda 10.0
- NVIDIA cuDNN 10.0
- NVIDIA Graphics Driver 441.28
- Microsoft Visual Studio 2017 C++ (Keras を動作させるため)
- Microsoft Windows 10 Pro 64bit 1909

## 学習モデル(ニューラルネットワーク)の仕様

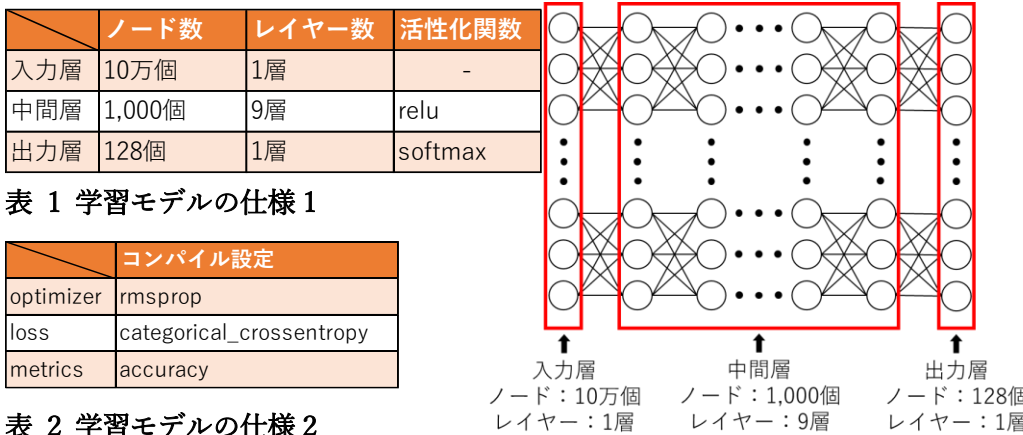


表1 学習モデルの仕様1

|           | コンパイル設定                  |
|-----------|--------------------------|
| optimizer | rmsprop                  |
| loss      | categorical_crossentropy |
| metrics   | accuracy                 |

表2 学習モデルの仕様2

仕様は表1、表2、図2のようになっています。入力層は、Unicodeのコードポイントを10万まで許容しているためノード数が10万になっています。中間層はパソコンの性能が許す限りノードとレイヤーを増やしたので、学習モデルのファイルサイズは約825MBになりました。

図2 ニューラルネットワーク

## 結果

正解率の求め方は、4種類のタグが1つ以上登録されている動画のタイトルから予測されたタグが、実際の動画のタグに含まれているかで判定しました。例えば、図5のような動画のタイトルがあったとき、この動画にはVOCALOIDのタグが含まれているので、VOCALOIDのタグが出れば正解と判定します。もしも、それ以外のタグが予測された場合、この動画には"演奏してみた", "歌ってみた", "踊ってみた"のタグは含まれていないので不正解と判定します。このルールを基に正解率を求めたところ、表3のように約92.6%の正解率を出すことができました。



図5 タイトルとタグの例

| 出題件数     | 正解数      | 正解率    |
|----------|----------|--------|
| 1746221件 | 1617648件 | 約92.6% |

表3

動画投稿サイトに投稿されたばかりの動画は、登録されているタグが少ないという問題があります。そこで、この問題を解決するために深層学習を使用し動画のタイトルから適切なタグを予測し自動でジャンル分けする機能を作りました。

## データの変換方法

図3のようにString型の文字列を、int型のUnicodeのコードポイントに変換することで機械が学習を行えるようにしました。ベクトル化にはvectorize\_sequences関数を、カテゴリ化にはto\_one\_hot関数を使用しました。



図3 データの変換方法

## 過学習を防ぐ

epochsの値に注目しました。epochsとは、「同じ訓練データを何回繰り返して学習させるのか」の回数の事です。当初はepochsの値を50に設定していたところ、図4の損失関数のグラフのようにepochsの値が15を超えたあたりから、学習の精度の変化が鈍くなることが分かりました。過学習を防ぐために、学習に効果のある回数で止めました。過学習とは、規則ではなく答えを覚えてしまう現象です。つまり、過学習が起きるとテストケースでは正解率が高くて、本番環境では正解率が下がってしまいます。そこで、epochsの値を15に下げ過学習を防ぎました。

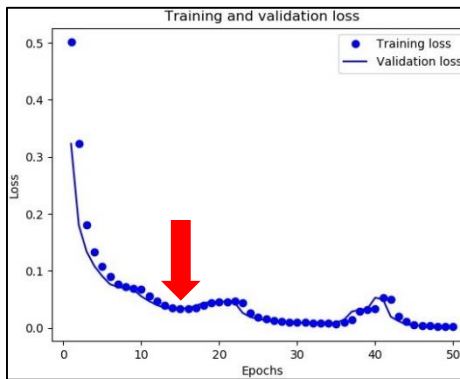


図4 損失関数のグラフ

## 考察・今後の展望

以上の正解率から、動画のタイトルと動画のタグに関係性があることがわかりました。今回は文字単位で学習を行いました。今後は文字単位ではなく、単語単位で学習させると精度が上がるのか下がるのか実験してみたいと思いました。特に、"【】"(すみつきかっこ)や"."(ドット)は単語ではないので、どのような結果になるのか興味深いです。また、タグの種類を増やして学習させようと思いました。さらに、訓練データと正解ラベルを別のデータセットに置き換えてより実用的な機能を作ろうと思いました。

## ソースコードと学習モデルの公開先

今回作ったソースコードと学習モデルはGitHubにて公開しております。環境構築すれば誰でも同様の実験が行えるので、ご興味のある方は是非私的利用の範囲内でお試ください。

※著作権は佐口航に帰属します。

<https://github.com/SaguchiWataru>

## 参考文献

- よくわかるPython[決定版]
- PythonとKerasによるディープラーニング



図6 GitHubのQRコード