# 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）



## 目的

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適切なタグを予測して自動でジャンル分けする機能を作りました。期待できる効果は、従来よりも視聴者のニーズに合った動画が Web サイトに表示されることです。

## 訓練データと正解ラベル

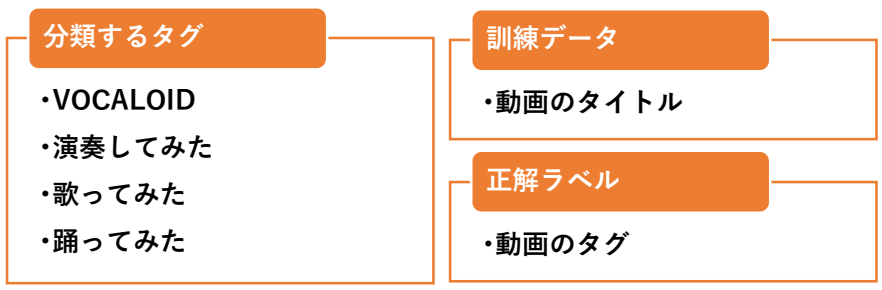


図 1 概要

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。

## 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

| | ノード数 | レイヤー数 | 活性化関数 |
|---|---|---|---|
| 入力層 | 10万個 | 1層 | - |
| 中間層 | 1,000個 | 9層 | relu |
| 出力層 | 128個 | 1層 | softmax |

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

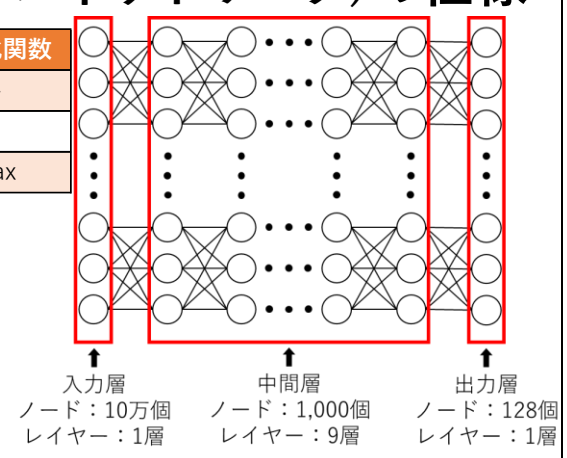


図 2 ニューラルネットワーク

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のでノード数が 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。

## 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

## 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 関数を使いました。

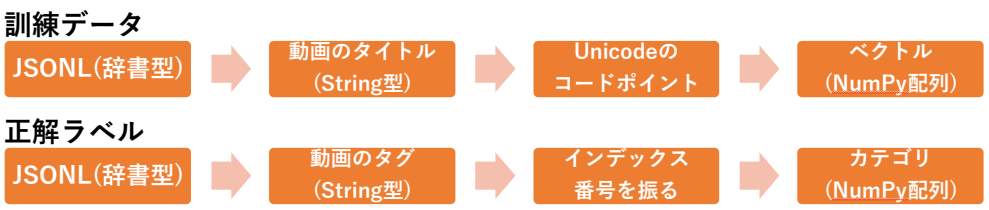


図 3 データの変換方法

## 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も、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。

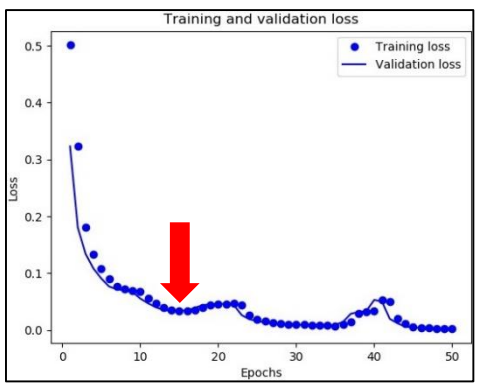


図 4 損失関数のグラフ

## 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が、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を公開しております。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お試しください。※私的利用の範囲でご使用ください。以下の内容の著作権は佐口航に帰属します。学習モデルは分割されたファイルを 7-Zip を使って展開してください。

https://github.com/SaguchiWataru/Deep_learning_of_titles_and_tags_using_Keras

## 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