YouTube Video Classification

Arian Shariat
August 2019

Abstract

Categorizing web-based videos is an important yet challenging task. People often tend to retrieve videos from internet by searching specific categories. As a result it is very important for website's beneficial purposes to satisfy user's demands. Further it affects the interest level of the users. With growth of videos on the Internet, organizing videos into categories is massively important for improving user experience and website monetization. This paper presents a solution for classifying YouTube videos based on their categories by using human made tags as inputs and passing it to an DNN (Deep Neural Network).

1. Introduction

As we discussed classifying videos is vital for multi media content service providers. Extracting internal features of a video is a complex job and brings some problems.

1.1. The Problem with videos

Difficulties are numerous, including large data diversity within a category, lack of manually labeled video data, and degradation of quality in some videos. Different patterns are generated for each class for classification. Hue, Saturation, Value color model is used to extract color features from each frame. Unlike text data, Multi-Media Content is way too much rich and expressive. The paper aims to tackle these issues and tends to classify videos based on a different aspect.

1.2. Tag, A new source of information

Tags have emerged as very powerful mechanism of information from users perspective and a service providers (like YouTube). Over the web not only the textual information, but also the tiniest video information has been tagged. Users constantly use tags as means of finding and re-finding the content they are interested in as well as the resources they want to be viewed by other users. Tags represent a social classification of the content.

1.3. Utilization of tags from aspect a user

- Lots of users use tags to recall information and it makes the process less time consuming.
- Tags are widely used for searching purposes.
- Some users tend to store videos that are personal to them. In such cases, they would like to use tags that have very low or no discovery for other users.
- Tags are one of the most valuable means of marketing these days. Users want the information submitted by them to be discovered by other users with maximal ease.

2. Methodology

A deep neural network is trained for classifying YouTube videos based on their categories. Code implementations can be found here.
A brief explanation of neural networks is given for unfamiliar readers.

2.1. Background

What is an Artificial Neural Network (ANN) ? Artificial neural networks are computing systems roughly inspired by the biological neural networks that constitute animal brains. A neural network has input and output neurons, which are connected by weighted synapses. The weights affect how much of the forward propagation goes through the neural network. The weights can then be changed during the back propagation (this is the part where the neural network is now learning). This process of forward propagation and backward propagation is conducted iteratively on every piece of data in a training data set. The greater the size of the data set and the greater the variety of data set that there is, the more that the neural network will learn, and the better that the neural network will get at predicting outputs.

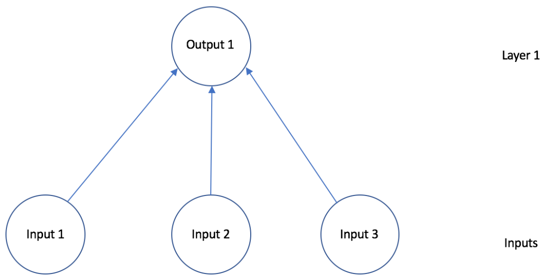


Figure 1. This neural network has one layer, three inputs, and one output. Any neural network can have any number of layers, inputs, or outputs.

Simply , a neural network is a connected graph with input neurons, output neurons, and weighted edges. Let's go into detail about some important components:

- Neurons: A neural network is a graph of neurons (neurons are nodes of the graph). A neuron has inputs and outputs. Similarly, a neural network has inputs and outputs. The inputs and outputs of a neural network are represented by input neurons and output neurons. Input neurons have no predecessor neurons, but do have an output. Similarly, an output neuron has no successor neuron, but does have inputs.

- Connections and Weights: A neural network consists of connections, each connection transferring the output of a neuron to the input of another neuron. Each connection is assigned a weight.

- Propagation Function: The propagation function computes the input of a neuron using the outputs of predecessor neurons and the connection's weight. The propagation function is leveraged during the forward propagation stage of training.

- Learning Rule: The learning rule is a function that modifies the weights of the connections. This serves to produce a favored output for a given input for the neural network. The learning rule is leveraged during the backward propagation stage of training.

- Overfitting: One of the problems that occur during neural network training. The error on the training set is driven to a very small value, but when new data is presented to the network the error is large. The network has memorized the training examples, but it has not learned to generalize to new situations.

- Dropout: Dropping out units in a neural network. Simply put, dropout refers to ignoring neurons during the training phase of certain set of neurons which is chosen at a random rate.

So now that we know what a Neural Network is. What is a Deep Neural Network? A Deep Neural Network simply has more layers than smaller Neural Networks. A smaller Neural Network might have 1-3 layers of neurons. However, a Deep Neural Network (DNN) has more than a few layers of neurons. A DNN might have 20 or 1,000 layers of neurons.

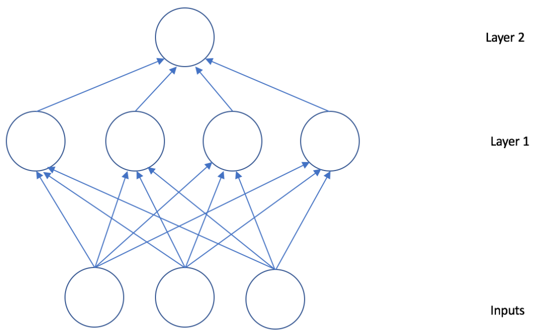


Figure 2. This neural network has two layers, three inputs, and one output. Any neural network can have any number of layers, inputs, or outputs. The layers between the input neurons and the final layer of output neurons are hidden layers of a deep neural network.

2.1.1. Summary of a DNN

DNN is a set of connected neurons organized in layers:

- input layer: brings the initial data into the system for further processing by subsequent layers of artificial neurons.

- hidden layers: layers between input layer and output layer, where artificial neurons take in a set of weighted inputs and produce an output through an activation function.

- output layer: the last layer of neurons that produces given outputs for the program.

2.2. Data Requirement

40K Top trending YouTube videos of Canada, USA and Great Britain (120K totally) are used as training, validating and testing purposes.
Data includes the video title, channel title, publish time, tags, views, likes and dislikes, description, and comment count and category id.

For the purpose of the classification the following fields were collected:

- Tags

- Number of likes

- Number of comments

- Views

- Category ID as label of the videos (17 distinct categories)

The main idea was to use tags as the only feature, but data duplication (discussed later) led to use of other features as well.

2.3. Experiments

At the beginning, only Canada's trending YouTube videos were used as dataset. A neural network with 2 hidden layer with 0.01 learning rate was used as base network. Due to dependency of input video tags (tags of a video are meaningful only if given to network together), One-Hot-Encoding is used for normalizing tags. Encoding is applied using a vocabulary of all tags in the dataset pruned by frequency (tags with frequency less than 30 are dropped). Results are shown. (Figures 3,4 and 5)

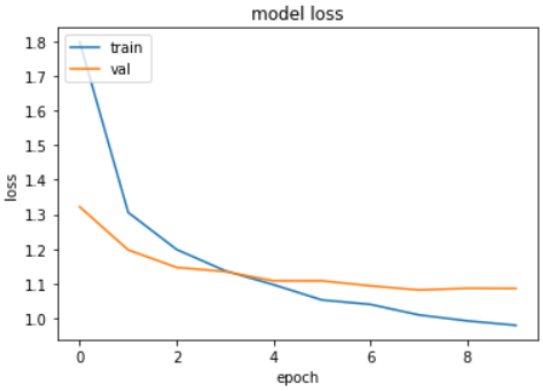


Figure 3. Model Loss

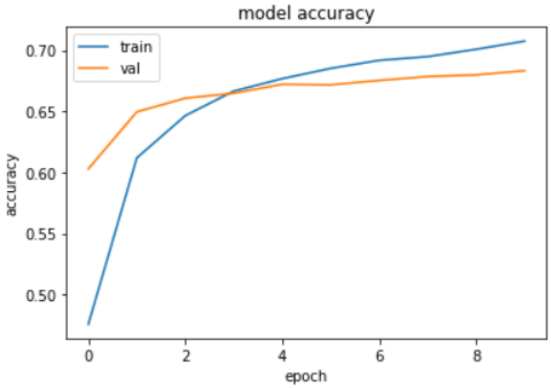
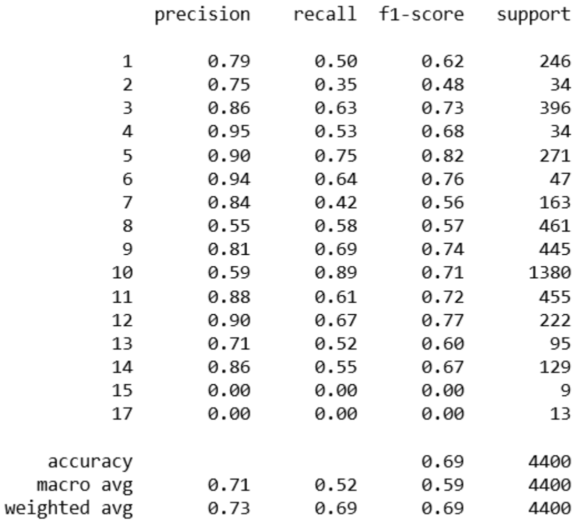


Figure 4. Model Accuracy.



```
              precision    recall  f1-score   support

           1       0.79      0.50      0.62       246
           2       0.75      0.35      0.48        34
           3       0.86      0.63      0.73       396
           4       0.95      0.53      0.68        34
           5       0.90      0.75      0.82       271
           6       0.94      0.64      0.76        47
           7       0.84      0.42      0.56       163
           8       0.55      0.58      0.57       461
           9       0.81      0.69      0.74       445
          10       0.59      0.89      0.71      1380
          11       0.88      0.61      0.72       455
          12       0.90      0.67      0.77       222
          13       0.71      0.52      0.60        95
          14       0.86      0.55      0.67       129
          15       0.00      0.00      0.00         9
          17       0.00      0.00      0.00        13

    accuracy                           0.69      4400
   macro avg       0.71      0.52      0.59      4400
weighted avg       0.73      0.69      0.69      4400
```

Figure 5. Test report before balancing data

Linearity of validation accuracy and values of train data support states presence of unbalanced data. (some categories have more train and test data. e.g category '17' have zero precision.) Thus Great Britain and USA datasets were added to former dataset for balancing purposes. Only 25K of final data (120K) were usable due to data duplication, Also several videos had the exact same tags. In order to not to loose mentioned videos, other features (Views, Likes and comments) were required to use. Views, number of likes and number of comments are normalized using Standard Scalar:

Standardization:

$$z = (x - \Theta)/\sigma$$

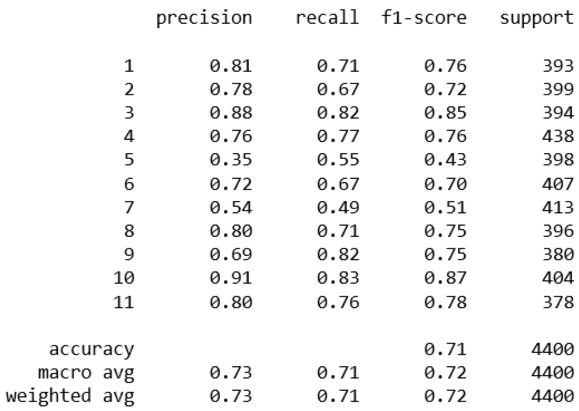
with mean:

$$\mu = 1/N\ \Sigma(x)$$

and standard deviation:

$$\sigma = (\sqrt{1/N\ \Sigma(x-\mu)^2})$$

6 categories dropped due to lack of frequency (less than 700 videos). Remaining categories down-sampled or up-sampled to obtain 2000 samples from each resulting in 22K balanced data. Sampling was necessary due to shortage of data. A new model trained with new data. Results are shown in Figure 6.



```
              precision    recall  f1-score   support

           1       0.81      0.71      0.76       393
           2       0.78      0.67      0.72       399
           3       0.88      0.82      0.85       394
           4       0.76      0.77      0.76       438
           5       0.35      0.55      0.43       398
           6       0.72      0.67      0.70       407
           7       0.54      0.49      0.51       413
           8       0.80      0.71      0.75       396
           9       0.69      0.82      0.75       380
          10       0.91      0.83      0.87       404
          11       0.80      0.76      0.78       378

    accuracy                           0.71      4400
   macro avg       0.73      0.71      0.72      4400
weighted avg       0.73      0.71      0.72      4400
```

Figure 6. Test report after balancing data

Train support values indicates that data are balanced now and equal data from each class is obtained. building up the model by altering layers, neuron sizes, regularization and etc happens to be the next step.

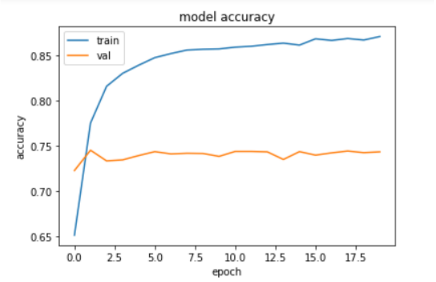


Figure 7. learning rate = 0.001

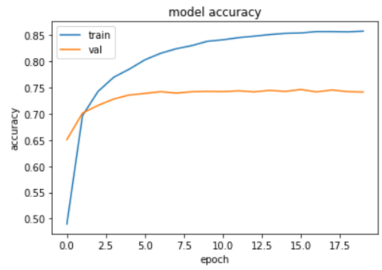


Figure 8. learning rate = 0.0001

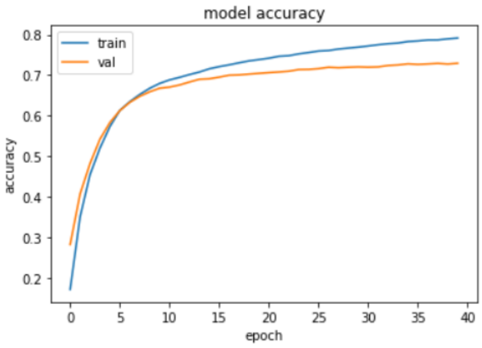


Figure 9. learning rate = 0.00001

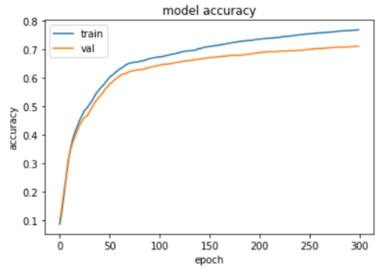


Figure 10. learning rate = 0.000001

Learning rates 0.001 and 0.0001 did not produce desired result as validation accuracy is too low. Learning rates 0.00001 and 0.000001 have similar results, Learning rate = 0.00001 is chosen for now due to faster training process. Figure 9 indicates overfitting after some epochs as validation accuracy is converging but training accuracy is still improving. Before performing regularization to overcome overfitting, lets try different models by altering layers.

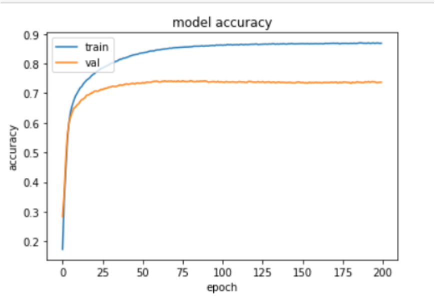


Figure 11. Training model with 3 hidden layers and learning rate = 0.0001

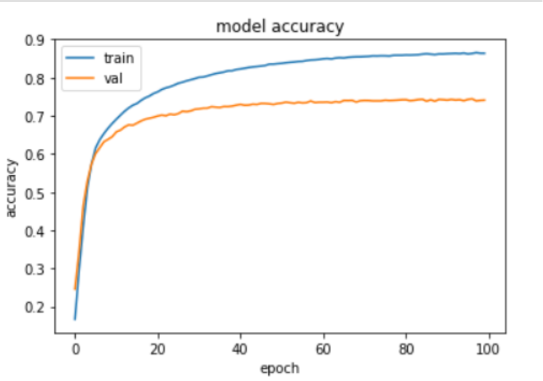


Figure 12. Training model with 4 hidden layers and learning rate = 0.0001

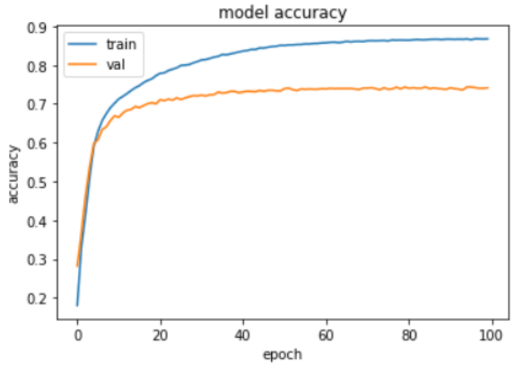


Figure 13. Training model with 5 hidden layers and learning rate = 0.0001

As shown in figures 11,12 and 13, validation accuracy is less linear in networks with 4 and 5 layers. At this point we go further with 4 layers network to speed up training process. Also overfitting can be diminished by using weight decay and drop out. Average of views, likes and comments also added to network as a feature for reducing overfitting. Final model training details is shown in figures 14 and 15.

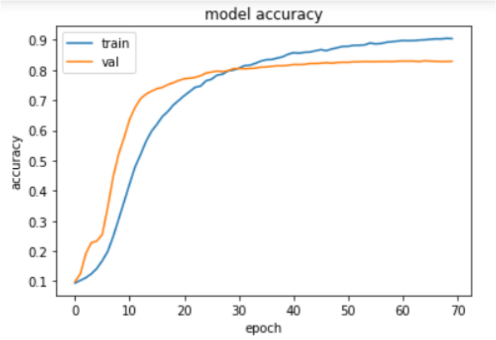


Figure 14. Final trained model accuracy

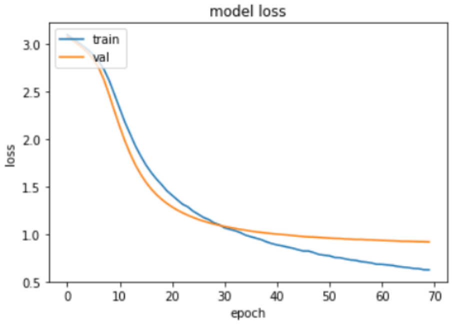
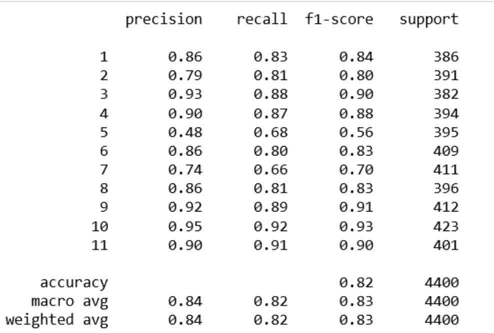


Figure 15. Final trained model loss

Superiority of validation accuracy in early epochs is due to resampled data as validating data and training data have duplications in some classes. This problem can be solved by using (or generating) lots of unique data. The model is evaluated using test data. (Figures 17 and 18)



```
              precision    recall  f1-score   support

           1       0.86      0.83      0.84       386
           2       0.79      0.81      0.80       391
           3       0.93      0.88      0.90       382
           4       0.90      0.87      0.88       394
           5       0.48      0.68      0.56       395
           6       0.86      0.80      0.83       409
           7       0.74      0.66      0.70       411
           8       0.86      0.81      0.83       396
           9       0.92      0.89      0.91       412
          10       0.95      0.92      0.93       423
          11       0.90      0.91      0.90       401

    accuracy                           0.82      4400
   macro avg       0.84      0.82      0.83      4400
weighted avg       0.84      0.82      0.83      4400
```

Figure 16. Final trained model reports

3. Conclusions

a DNN model trained using top trending YouTube videos given tags. Tags are great features to use for classify video but some shortcomings occur. On one hand a massive dataset of tags are required for obtaining desired accuracy and loss. On the other hand, the usage of tags lies in the looseness of the representation. As tags are created by humans they represent the human interpretation of the multimedia content, the personal bias of the human being comes into the context, which in turn acts as noise in this scenario. Thus, it becomes difficult to identify tags that are relevant to the multimedia content. here is where importance of pruning and normalizing comes in sight. For improving result;

- Use larger dataset. 22K data is not enough for training a robust system.
- User usually use several words in a tag and separate them with underscore. words of tags can be separated in to new tags and can be pruned by frequency. An embedding layer as first layer of network can lead to better results given mentioned data.