Classifying Television Commercials by Convolutional Neural Network with Evaluation of Activation Functions

Ryu Nagabayashi s1230179 Supervised by Prof. Kazuyoshi Mori

Abstract

In this paper, we describe the development of a system to classify television commercial categories. In modern times, people watch televisions on a daily basis. Investigating the television commercials can be useful for social analysis because the television commercials have influences on people and the social culture. So we decided to develop a system to classify television commercial by using Convolutional Neural Network (CNN). On the other hand, many activation functions of CNNs have been proposed. As in the development, we evaluated the system by various activation functions. In this study, the good classification accuracy was obtained when ReLU and its derivatives(ELU, SELU, Leaky ReLU, ReLU6) was used as the activation function.

1 Introduction

In modern times, many people have television and often watch television commercials. Television commercial provide information on various products and services to us. It affects buying intention of viewers. We consider that investigating television commercials is useful for social analysis because television commercials reflect the social situation and trends at the time. In order to investigate television commercials, it is important to classify television commercials according to their categories.

We decided to develop a system that automatically classifies television commercials using convolution neural network because it is inefficient to classify television commercials manually. CNN has been used frequently recently. In many cases, images are input one by one to CNN. In this study, we implement CNN with video input by using several consecutive images for input. This will enable us to obtain a temporal features of television commercials. Also, in order to develop a better system, we evaluate the activation functions in CNN in the viewpoint of the recognition accuracy by employing various activation functions. They have been proposed for used in CNN so far, but any activation functions have its own some problems. Therefore, it is important to investigate the optimal activation function for the CNN we developed.

2 Neural Network

Neural network [1] is a computing system modeled human on the brain and nervous system, which is often used in the field of pattern recognition such as character recognition and speech recognition. The neural network consists of an input layer, a hidden layer, and an output layer. Machine learning by a neural network with multiple hidden layers is called deep learning. An example of neural network is shown in Fig. 1. The circle in Fig. 1 is called a neuron. It is calculated with a weight and a bias and passes the value to the next neuron. Adjusting this parameter makes it possible to develop a neural network that eventually output the expected result.

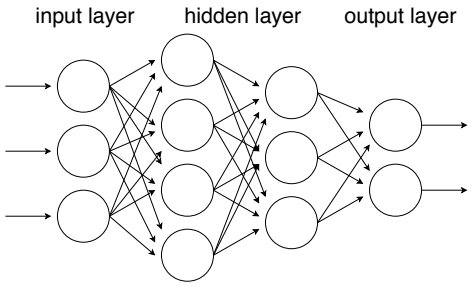

Figure 1: Example of Neural NetWork

2.1 Convolutional Neural Network

CNN is one of neural networks with deep learning. It is a neural network with convolution layers and pooling layers. Neural network is often used in the field of image recognition. In the convolution layer, CNN calculates the input data with convolution and obtains local feature of the data. The data calculated in this way is

called a feature map [2]. The pooling layer is usually applied behind the convolution layer and compresses the information to transform the input data into a more manageable form. The pooling layer has several types such as average pooling [2] and max pooling [2]. In this research, max pooling, which outputs maximum value of input, is employed.

2.2 OpenCV and TensorFlow

The Open Source Computer Vision Library (OpenCV) [3] is an open source library that summarizes the functions for processing images and movies. This supports a wide variety of programming languages such as C, C++, Python, and Java. This is used for video and image processing in this research.

TensorFlow [4] is also an open source software library used in machine learning developed by Google. This supports the programming languages, C, C++, and Python. This is used for neural network constructions in this research.

3 Activation function

Activation function is a function that activates the neuron output. For example, the calculation of neurons in the output layer in the neural network of Fig. 1 is shown in Fig. 2. The input y to the neuron can be expressed as

$$y = x_1 w_1 + x_2 w_2 + x_3 w_3 + b, \tag{1}$$

where x_1, x_2, and x_3 are outputs of previous layer, w_1, w_2 and w_3 are weights and b is the bias. The output from the neuron is derived by applying the activation function.

Denoting $h(\cdot)$ by the activation function, the output from the neuron is given as $h(y)$ (See Fig. 2). Various activation functions have been proposed so far. The following seven activation functions are employed in this research.

3.1 Sigmoid function and Tanh function

Sigmoid function (Fig. 3) [2] is a function whose output value range is 0 to 1. Sigmoid function can be expressed as

$$h(x) = \frac{1}{1 + \mathrm{e}^{-x}}. \tag{2}$$

Activation function is said that it is better to pass through the origin, the activation function that improved this problem is the tanh function (Fig. 4) [2]. Tanh function (Fig. 3) is a function whose output value range is -1 to 1. Tanh function can be expressed as

$$h(x) = \frac{\mathrm{e}^{x} - \mathrm{e}^{-x}}{\mathrm{e}^{x} + \mathrm{e}^{-x}}. \tag{3}$$

In the past, sigmoid function and tanh function were often used, but recently ReLU function (Fig. 5) [5] is often used in many studies [6]. This is because vanishing gradient problem is improved in ReLU [2].

3.2 ReLU

ReLU outputs 0 if the input value x is negative and x if the input value x is positive. This ReLU function can be expressed as

$$h(x) = \max(0, x). \tag{4}$$

In many studies, it is said that we can obtaine the best recognition accuracy by employing ReLU [5]. However there is a problem that ReLU always outputs 0 when the input is a negative value. Several activation functions have been proposed to improve it.

3.3 Derived form of ReLU function

In order to solve the problem in ReLU, several functions with negative argument values are proposed. ELU (Fig. 6) [7], SELU (Fig. 7) [8], Leaky ReLU (Fig. 8) [9] and ReLU6 (Fig. 9) [10] are used in this research.

In contrast to ReLU, SELU have negative values. SELU function can be expressed as

$$h(x) = \lambda \begin{cases} x & (x \geq 0) \\ \alpha(\mathrm{e}^{x} - 1) & (x \leq 0). \end{cases} \tag{5}$$

Example of λ and α is $\lambda = 1.0607$ and $\alpha = 1.6732$ [8]. ELU is set λ and α of SELU function to 1. ELU function can be expressed as

$$h(x) = \begin{cases} x & (x \geq 0) \\ \mathrm{e}^{x} - 1 & (x \leq 0). \end{cases} \tag{6}$$

In contrast to ReLU, Leaky ReLU also has negative values. Leaky ReLU can be expressed as

$$h(x) = \begin{cases} x & (x \geq 0) \\ \alpha x & (x \leq 0), \end{cases} \tag{7}$$

where $\alpha = 0.5$ [9]. RELU 6 has a maximum of 6 and does not increase any further. ReLU6 can be expressed as

$$h(x) = \min(\max(0, x), 6). \tag{8}$$

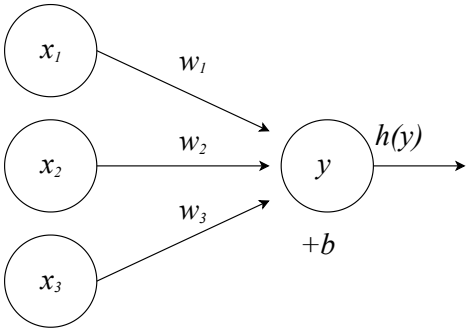

Figure 2: Neuron Output

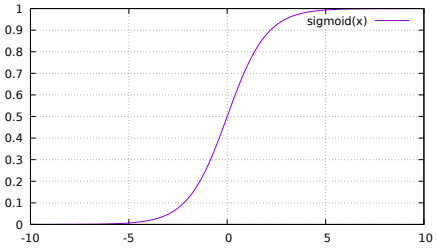

Figure 3: sigmoid

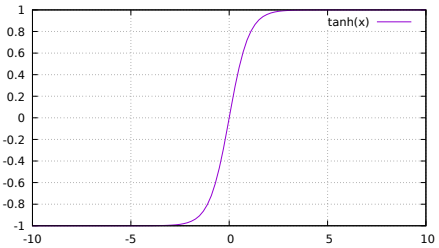

Figure 4: tanh

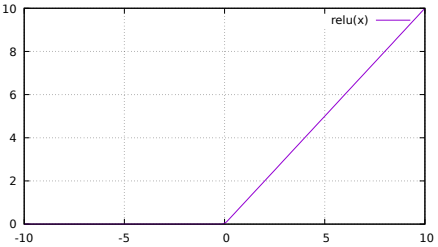

Figure 5: ReLU

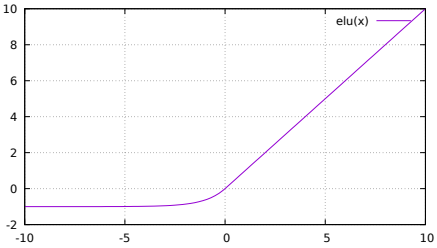

Figure 6: ELU

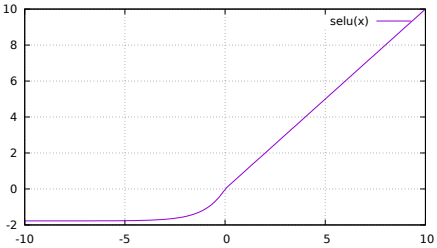

Figure 7: SELU

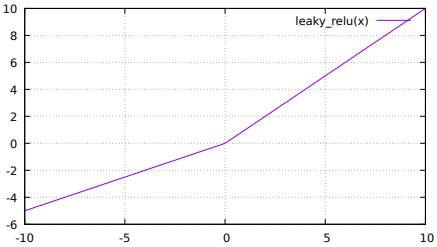

Figure 8: Leaky ReLU

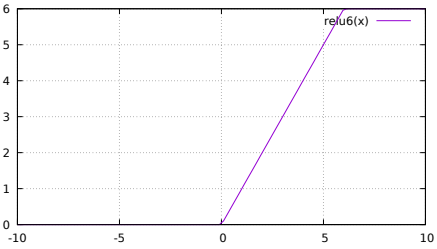

Figure 9: ReLU6

4 Experiments

4.1 Preparation

In order to training CNN, we have so far collected 245 of various television commercials. We use only 15 seconds of television commercials in this study. These CMs are labeled as a category with “food,” “car,” “cosmetic,” and “other.” After this task, we got training videos of food (103), car (49), cosmetic (41), and other (52). In the same way, in order to test CNN, we have so far collected 40 of various television commercials and labeled according to categories food (10), car (10), cosmetic (10), and other (10).

4.2 Structure of CNN

We developed CNN (Fig. 10) that categorizes television commercials into four categories (“food,” “car”, “ cosmetic ”, “ other ”). It consists of input layer, 3D convolutional layer, fully connected layer and output layer. In the input layer, we have used 30 images as input that were taken out from one television commercial (Sampling every 0.5 seconds). All of CMs are originally recorded as 1920 x 1080 resolution (It depends on the broadcast media of CM such as terrestrial broadcasting, BS broadcasting and CS broadcasting), but for the computer source limitation, we have resized them to 80 x 45. 3D convolution layers consist of four layers, and each pooling layer is applied behind each 3D convolution layers (Fig. 11). We employed max pooling in pooling layer. In this research, we evaluate the activation functions by changing activation functions in 3D convolution layer. The output layer gives the decision of the classification into four categories. For activation functions, the softmax function is employed on the output layer. Learning rate set to 0.00001. Training is terminated when training accuracy reaches 98%.

4.3 Results and Discussion

In the experiment, CNN was trained for each of the seven activation functions described in Chapter 3. First, The CNN we developed categorizes television commercials into four categories (“ food ”, “ car ”, “ cosmetic ”, “ other ”) and trained it. The result are shown in Table. 1. As a result, the average value of the test accuracy was about 46.8%, which is less than half of the training accuracy.

In order to investigate the cause of low accuracy, we considered that we need to classify even the number of categories other than four categories. Therefore, we newly developed CNN which categorize into two categories (“food”, “car”) and three categories (“food”, “ car ”, “ cosmetic ”). Thereafter, television commercials were classified by using a trained model, and accuracy was measured. Table. 2 shows test accuracy each activation functions and their average accuracy. On the results of Sigmoid, its accuracy is 15 to 20% lower than the average, which shows that the accuracy is much lower than the other activation functions.On the results of activation functions other than Sigmoid, there is no conspicuous trend and there are almost same value.

	4categories
Sigmoid	25%
Tanh	52.5%
ReLU	50%
ELU	50%
SELU	47.5%
Leaky ReLU	52.5%
ReLU6	50%
Average	46.8%

Table 1: Test Accuracy every Activation Function for 4 Category

In the training, recognition accuracy increased to 98%, but the results show that the test accuracy is low throughout. For example, the average of the test accuracy of the four categories decreased to less than half 46.8%. We considered the possibility that this low accuracy is based on overfitting. Therefore, we changed the maximum accuracy of training and checked whether overfitting occurred. The results are shown in Table. 3 and Fig. 12. Although there are some errors, the test accuracy tends to increase as the training accuracy increases. Therefore, overfitting seems not to have occurred.

5 Conclusion and Future Work

In this study, we developed CNN system that classifies television commercial and evaluated the system by various activation functions. The results show that the sigmoid function is a particularly low accuracy compared with other activation functions, and the other activation functions are almost the same accuracy. Some derivation forms made to improve the problem of ReLU have recognition accuracy that is almost the same as ReLU. It cannot be said that which activation function is the

	2categories	3categories	4categories
Sigmoid	65%	43.3%	25%
Tanh	80%	66.7%	52.5%
ReLU	80%	63.3%	50%
ELU	85%	73.3%	50%
SELU	85%	63.3%	47.5%
Leaky ReLU	90%	56.7%	52.5%
ReLU6	85%	50%	50%
Average	81%	59.5%	46.8%

Table 2: Test Accuracy every Activation Function

best is unconditionally decided. However, currently it seems to be the best to employ ReLU and its derivation activation functions.

Besides, as future works, better system should be developed by increasing the training data and adjusting the structure of CNN such as learning rate, the size of input images, the parameter of hidden layers and the number of layers of neural net. Further, the transfer learning [11] will be employed to obtain high accuracy from small number of data.

References

[1] Hiromi Hirano,“C でつくるニューラルネットワーク，パーソナルメディア株式会社,”1991.

[2] 原田達也,“画像認識”, 講談社, 2017.

[3] “OpenCV documentation Index,” http://docs.opencv.org/

[4] “MNIST For ML Beginners,”Dec.2016, https://www.tensorflow.org/versions/r0.11/tutorials/mnist/beginners/index.html#the-mnist-data

[5] V.Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In ICML, 2010.

[6] Yann LeCun, Yoshua Bengio and Geo ffrey Hinton “Deep learning, ” Nature 521 (7553), 436/444. May. 2015.

[7] D.-A. Clevert. T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units(ELUs). In ICLR, 2016

[8] Gunter Klambauer, Thomas Unterthiner, Andreas Mayr and Sepp Hochreiter “Self-Normalizing Neural Networks”2017

[9] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In the ICML 2013 workshop on Deep Learning for Audio, Speech and Language Processing, 2013.

[10] Alex Krizhevsky “Convolutional Deep Belief Networks on CIFAR-10”2012

[11] lan Goodfellow, Yoshua Bengio, and Aaron Courville “DEEP LEARNING”, 2016

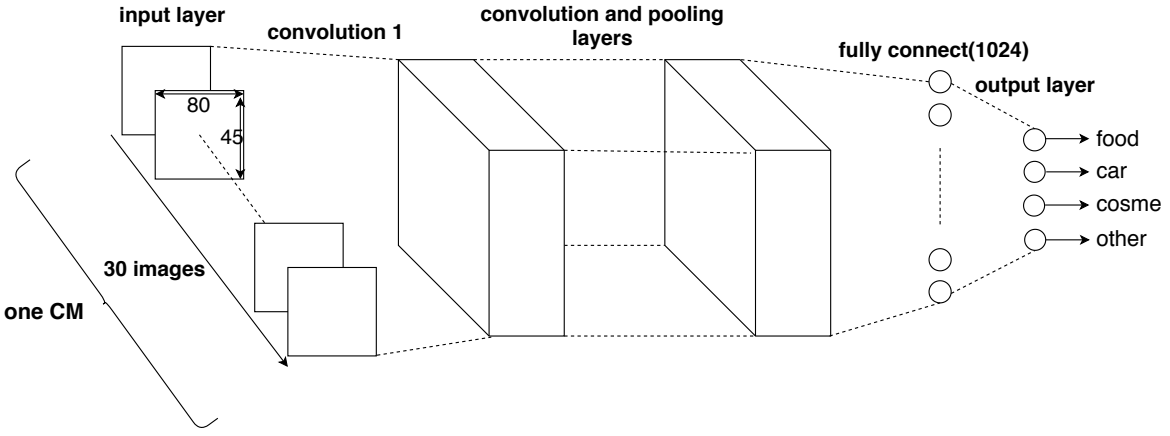

Figure 10: Convolutional Neural NetWork

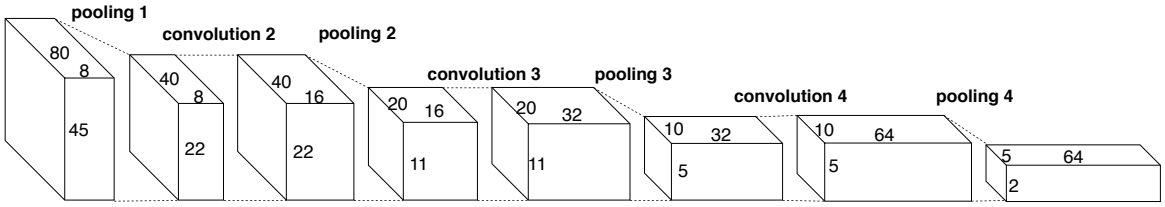

Figure 11: Convolution and Pooling Layer

	0.5	0.75	0.875	0.9375	0.96875	0.984375	0.992188	0.996094	0.998047
Sigmoid	37.5%	42.5%	37.5%	45%	45%	42.5%	50%	42.5%	40%
Tanh	45%	37.5%	50%	50%	55%	47.5%	50%	65%	60%
ReLU	35%	45%	60%	52.5%	65%	50%	57.5%	57.5%	42.5%
ELU	37.5%	52.5%	37.5%	60%	47.5%	62.5%	55%	65%	60%
SELU	35%	50%	50%	50%	47.5%	52.5%	57.5%	60%	55%
Leaky ReLU	35%	45%	50%	50%	57.5%	60%	55%	62.5%	62.5%
ReLU6	42.5%	45.5%	62.5%	42.5%	50%	45%	52.5%	57.5%	60%

Table 3: Overfitting Test every Max Training Accuracy

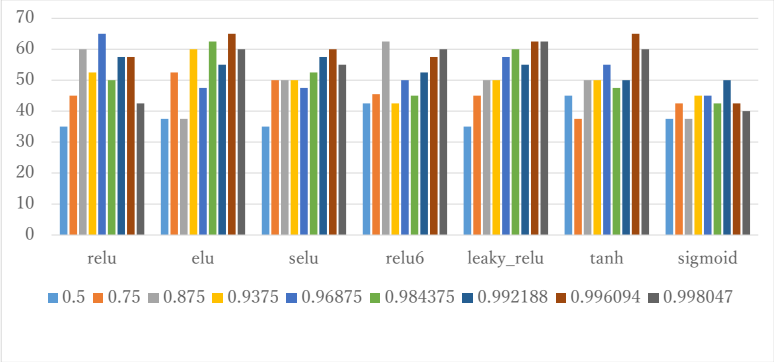

Figure 12: Overfitting Test