Single Classification: Fully Connected and Convolutional Neural Network on CIFAR-100 Database

Ziwei Zhu

1. Introduction

In the field of computer vision, neural networks own the advantage of representation learning, that is, automatically extracting objects features without manual selection. Unlike classic machine learning methods, such as Regressions and Random Forest, a neural network does not require manually specifying the features of input data and can generate the masks to weigh each feature during the training. In this project, I implemented two typical neural networks, fully connected neural network and convolutional neural network, recognizing bees in order to compare the performances of these two approaches. This project requires downloading the binary files of CIFAR-100 database in advance. The programs are written in python 2.7 and run on Google Colab.

2. Backgrounds

Fully connected is the most basic method in neural networks. It assumes that every neuron, which is a single pixel in image processing, contributes more or less implication to producing all neurons in the next layer. CNN, on the contrast, focuses on generalizing features within proximate neurons. It assumes that close neurons are related significantly in object detection while distant neurons do not necessarily have features in common. Theoretically, with the same number of layers, CNN has fewer memory consumptions and higher speed. Moreover, multiple sizes of masks in CNN can be trained and sifted until the model meets the optimization.

3. Implementation

I first download CIFAR-100 in the binary version from https://www.cs.toronto.edu/~kriz/cifar.html, which contains two files, train and test. There are 1000 pictures in the training process and 200 pictures to be tested. On Google Colab, uploading a file requires importing the library, google.colab first. After that, we use Pickle library to read the binary file into number arrays and resize the arrays into 4D. The first dimension represents the indices of pictures, while the rest three represents x, y coordinates and color channels.

Although CIFAR-100 is widely used as input data in training models, it lacks the generality of the objects in the real world since the pictures are already trimmed into 32*32 pixels. One of the pictures labeled as "bee" is shown below.

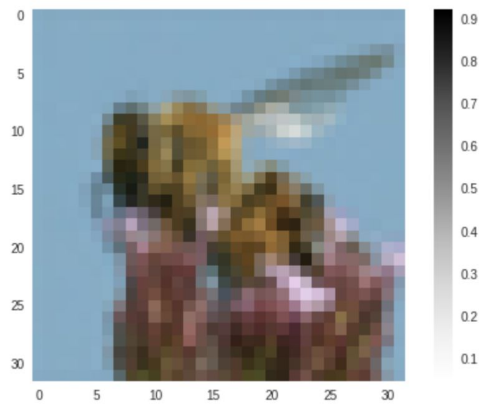

Fig. 1. ne of the picture labeled "bee" in CIFAR-100 database

The two models, fully connected layers and CNN are constructed in the following step. For the fully connected network, I build a dense layer with 128 neurons and another dense layer with 2 neurons since we only need to distinguish bees from other classes. In CNN, total four convolutional layers are built with activation layers in between. The training epochs of the two models are all set to 30. The time lasts in the training process is recorded and the accuracy is calculated after the model is verified on the 200 test pictures. The two models are trained by the GPU provided on Colab, which is Tesla K80 with 2496 CUDA cores. The performances are shown below.

	time consumptio n	accuracy
Fully Connected	4.78s	0.86
CNN	17.75s	0.98

Table. 1. the time consumption and accuracy of CNN

4. Results

From the table, we can conclude that although it takes more time to train the 5 neural network layers, the accuracy is significantly higher than the fully connected layers. Time and accuracy are the trade-offs in training models. CNN usually owns more advantages in practical usage due to its flexibility in dimensions and layers.

Below is the probability histograms of the classification generated from fully connected layers. The left column represents the confidence to recognize an object other than a bee, while the right column represents the confidences to recognize the object as a bee. We can see the discrepancies between the two columns are remarkable in most conditions, but for the object similar to bees in shape and colors, the results fall into false negative error. The visualization of the classification results can provide people with direct awareness of the shortcomings of the model. And this can be utilized when refining the model and introducing more complex models to correct the false results.

Fig. 2. the prediction results of the fully connected layers

5. Conclusion and Limitations

In a practical implementation, single fully connected layers or CNN is no longer effective in tackling with complicated problems. These two methods are served as transition layers or intermediate layers in a complex model. Practical situations usually require models to take more disturbances such as backgrounds and sizes into account and expect models to be more robust. However, it is still critical for us to understand the performance and mathematical meanings behind the models and take them as bedrocks to solve problems.