

Capstone Project

Ryosuke Honda

August 3, 2016

1 Definition

1.1 Project Overview

Student provides a high-level overview of the project in layman 's terms. Background information such as the problem domain, the project origin, and related data sets or input data is given.

The improvement on the camera and the widespread of larger capacity hardwares lead us to store more digital pictures. However computers can't understand the meaning of the pictures, so it is essential for us humans to classify or search images. Image recognition(Figure.1) will help us to classify or search pictures without our intervention. These days, because of the development of the computational capacity, we can process large number of pictures with high level accuracy(nearly the human's recognition). In this project, I'll discuss the image recognition algorithm which will be useful in the classification of large number of images.

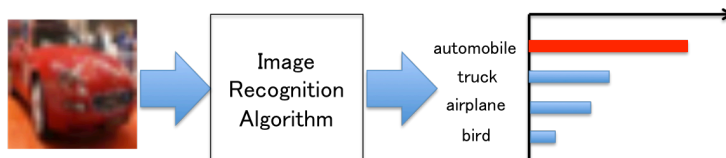


Figure 1: Image of the Algorithm

1.2 Problem Statement

The problem which needs to be solved is clearly defined. A strategy for solving the problem, including discussion of the expected solution, has been made.

The CIFAR-10 dataset

The CIFAR-10 dataset consists of 60,000 32x32x3(width=height=32 and RGB=3) color images in 10 classes, with 6,000 images. 50,000 images are the training images and 10,000 images are the test images.

The classes are completely mutually exclusive. There is no overlap within each image class.

In this task, I'll discuss the image recognition by Convolutional Neural Network(CNN).

1.3 Metrics

Metrics used to measure performance of a model or result are clearly defined. Metrics are justified based on the characteristics of the problem.

The objective in this problem is multi class classification. Therefore, the metrics will be 'cross entropy'. The objective of the learning in this task is to minimize the cross entropy. Cross entropy is defined as the following formula.

$$C = - \sum_{j=1}^n d_j \log p_j \quad (1)$$

d_1, \dots, d_n is the optimal output (Correct output). p_1, \dots, p_n is the probability of for the output class. This probability is calculated by softmax function which is defined as below.

$$p_j = \frac{e^{u_j}}{\sum_{k=1}^n e^{u_k}} \quad (2)$$

Classification Error is calculated by the difference between the optimal output d_1, \dots, d_n and predicted output p_1, \dots, p_n . The target output d_1, \dots, d_n takes the representation of 1-of-n. That means that only the correct label j becomes $d_j = 1$ and the others $k (\neq j)$ become $d_k = 0$.

2 Analysis

2.1 Data Exploration

If a dataset is present, features and calculated statistics relevant to the problem have been reported and discussed, along with a sampling of the data. In lieu of a dataset, a thorough description of the input space or input data has been made. Abnormalities or characteristics about the data or input that need to be addressed have been identified.

CIFAR-10 dataset consists of 50,000 images of training data and 10,000 images of test data. For each data, it contains 10 different kind of classes (discussed in the next chapter) and the distribution of each class is the same in both training and test data. That means 5,000 images for each class in the training data and 1,000 images for each class in the test data. (Figure.3 and Figure.4).

2.2 Exploratory Visualization

A visualization has been provided that summarizes or extracts a relevant characteristic or feature about the dataset or input data with thorough discussion. Visual cues are clearly defined.

There are 60,000 of images in total. Figure (50,000 images for training data and 10,000 images for test data). 2 shows the samples of the images. I plotted 10 images for each class. Figure.3 and Figure.4 shows the distribution of the data. For the training dataset, each class has 5,000 images and for the test dataset, each class has 1,000 images. The label 0 to 9 corresponds to 'Airplane', 'Automobile', 'Bird', 'Cat', 'Deer', 'Dog', 'Frog', 'Horse', 'Ship', and 'Truck'.

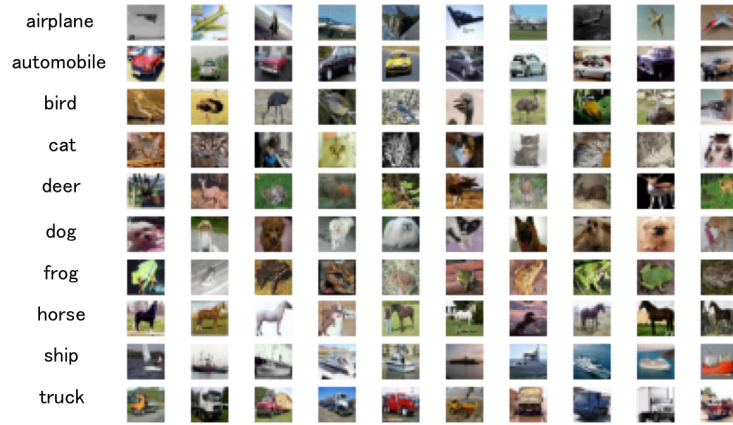


Figure 2: Sample of the Images

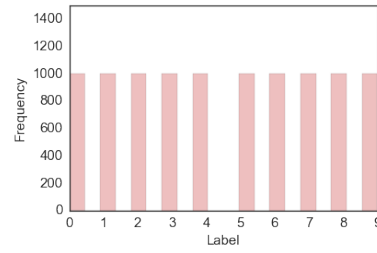
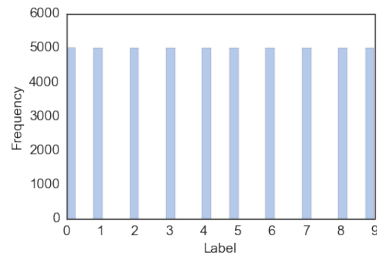


Figure 3: Distribution of Training Data Figure 4: Distribution of Test Data

2.3 Algorithms and Techniques

Algorithms and techniques used in the project are thoroughly discussed and properly justified based on the characteristics of the problem.

In this task, I'll use Convolutional Neural Network(CNN). CNN has been successful in practical applications for image recognition. CNN consists of convolution layer, pooling layer and fully-connected layer and sometimes contains local contrast normalization(LCN). In this chapter, I'll discuss the convolution layer and pooling. As the name 'convolution' suggests, the network employs a mathematical operation called convolution.

Figure.5 is the example of the architecture of the CNN.

Pooling layer is put after the convolution layer. The function of the pooling layer is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting. By introducing pooling layer, not only the architecture will be more robust but also the dimensionality will be reduced. Max pooling and Average pooling are the typical pooling which are generally utilized. Figure.7 is the example of the pooling. The original map is the size of 4x4. The stride

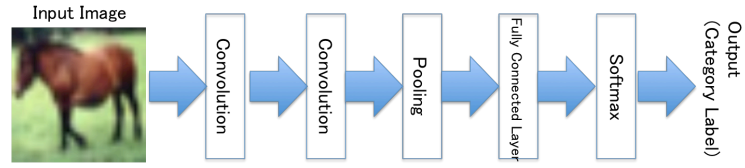


Figure 5: An example of CNN Architecture

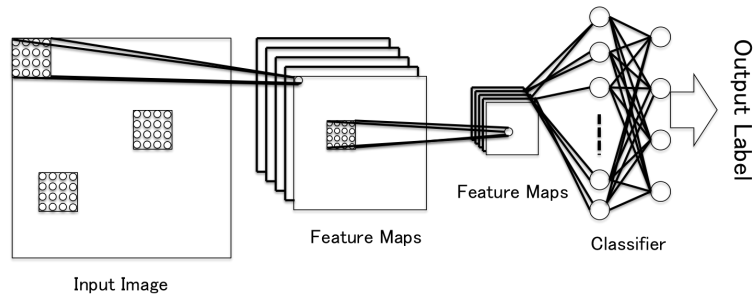


Figure 6: Example of Max Pooling and Average Pooling

for the pooling is 2 and the pooling size is 2×2 . "Max pooling" is to extract the maximum pixel from each region and "Average pooling" is to calculate the average value for each region. In this task, I utilized max pooling at the pooling layers.

2.4 Benchmark

Student clearly defines a benchmark result or threshold for comparing performances of solutions obtained.

For the CNNs architecture, it is quite important to decide the number of layers. Therefore, I first choose several convolutional layers and decide one of them as a benchmark.

When deciding the architecture, I set the mutual parameters as follows. The number of batch size is 32. The number of filters of each convolutional layer is 32. and finally the number of epochs is 20.

I tried 4 architecture of CNNs.

1. 1 Convolutional layer 2 Fully connected layers
The simplest version in these model. The input and output dimension on each layer are below. The accuracy rate of training and validation data are also below.
2. 2 Convolutional layers 2 Fully connected layers
The input and output dimension on each layer are below. The accuracy rate of training and validation data are also below.
3. 3 Convolutional layers 2 Fully connected layers
The accuracy rate of training and validation data are also below.

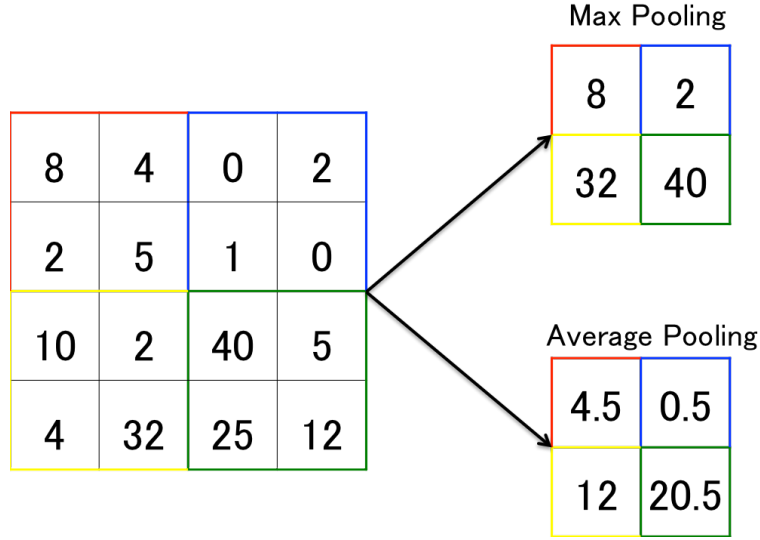


Figure 7: Example of Max Pooling and Average Pooling

4. 4 Convolutional layers 2 Fully connected layers
The accuracy rate of training and validation data are also below.

3 Methodology

3.1 Data Preprocessing

All preprocessing steps have been clearly documented. Abnormalities or characteristics about the data or input that needed to be addressed have been corrected. If no data preprocessing is necessary, it has been clearly justified.

Apart from the models of image processing or computer vision, CNN doesn't need complex preprocessing. However, when analyzing the data, data preprocessing plays a crucial role. One of the first steps is the normalization of the data. This step is essential when dealing with parameters of different units and scales. In this dataset, pixel values range from 0 to 255. I process normalization to this dataset. Normalization scales all numeric variables in the range of [0,1]. The formula is given below.

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (3)$$

The minimum pixel value is 0 and maximum pixel value is 255. Therefore, I normalized the data by following.

$$x_{new} = \frac{x}{255} \quad (4)$$

3.2 Implementation

The process for which metrics, algorithms, and techniques were implemented with the given datasets or input data has been thoroughly documented. Complications that occurred during the coding process are discussed.

3.3 Refinement

The process of improving upon the algorithms and techniques used is clearly documented. Both the initial and final solutions are reported, along with intermediate solutions, if necessary.

Finding the optimal parameters for deep learning is quite difficult though it is important. When it comes to typical machine learning algorithm(Decision tree, Support Vector Machine etc.),grid search is taken to search the optimal parameters. However, it's almost impossible to apply grid search in deep learning because of the computational time. Therefore, other methods are indispensable. A good choice is Bayesian optimization, which has been shown to outperform other state of the art global optimization algorithms on a number of challenging optimization benchmark functions.

4 Result

4.1 Model Evaluation and Validation

The final model 's qualities such as parameters are evaluated in detail. Some type of analysis is used to validate the robustness of the model 's solution.

4.2 Justification

The final results are compared to the benchmark result or threshold with some type of statistical analysis. Justification is made as to whether the final model and solution is significant enough to have adequately solved the problem.

5 Conclusion

5.1 Free-Form Visualization

A visualization has been provided that emphasizes an important quality about the project with thorough discussion. Visual cues are clearly defined.

5.2 Reflection

Student adequately summarizes the end-to-end problem solution and discusses one or two particular aspects of the project they found interesting or difficult.

5.3 Improvement

Discussion is made as to how one aspect of the implementation could be improved. Potential solutions resulting from these improvements are considered and compared/contrasted to the current solution.