IFT6390 - Report of Homework4

Classification de dessins

Team: YLW_IFT6390_UdeM

Qiang, Ye (20139927); Lifeng, Wan (20108546); Jinfang, Luo (20111308)

December 1, 2018

1 Introduction

This report briefly presented our team work during the Kaggle In-class competition as homework 4. We nee to predict the categories of 10000 hand-drawn but noised images called Quick, Draw! based on another 10000 training images and their labels. The team has been working together for more than two weeks in a very friendly atmosphere, climbing the accuracy on test dataset. After removing the noises from the images, resizing each image from size of 100x100 to 30x30, augmenting original data , designing a competitive CNN model, and a long time model training procedure, finally we obtained an accuracy 83.40% on partial test dataset.

2 Feature Design(Pre-processing)

2.1 Data Pre-processing

- **Noise reduction:** Deep first search(DFS) algorithm is used to find the maximal connected points in the original image assuming that a meaningful part in an image has the largest connected points, which may not always be true in this task. DBSCAN is an another algorithm capable of handling this work if parameters are set to: epsilon = 1, and min_samples = 2. We implemented both and chose DFS for further work.

- **Re-sizing** A meaningful part of an image only occupies small area of the original image with 100x100 pixels. Brief statistics shows that most of meaningful part are within a size 30x30 pixels. Therefore, we cropped and re-sized each image into a smaller size: 30x30 pixels.

- **Normalization** This operation scales and translates each feature individually in the given range between 0 and 1.

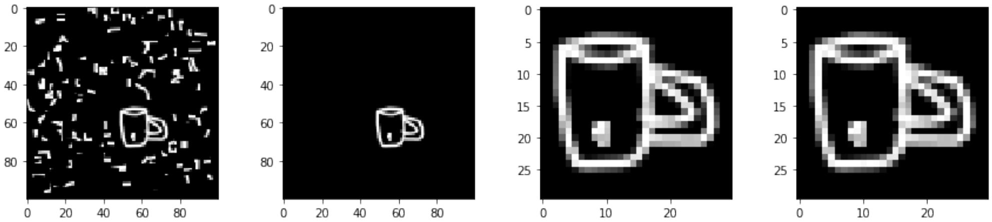

Figure1. a mug(id:7346) image sample and outputs of pre-processing

Figure1 shows the effect of above procedures on an image sample(id = 4962).

- **Washing**

  We manually removed some samples which are not hand drawings at all.

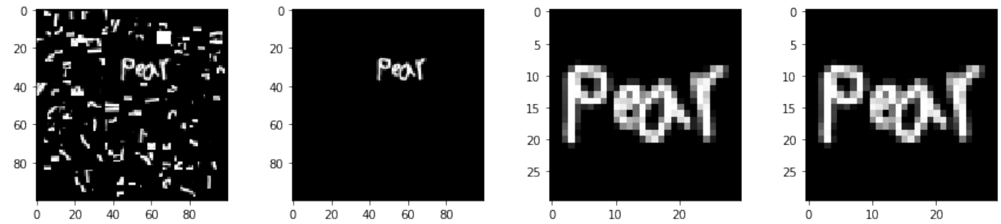

Figure2. Is this a hand-drawn pear? (id: 341, category: pear)

Figure2 shows one of the images washed out: a handwritten characters of pear.

2.2 Data Augmentation

Several data augmentation strategies are used: left-to-right-flipping, rotation with tiny degrees on all samples. Additionally, we carefully performed some large degrees of rotation on some special samples because these kinds of pictures can be easily recognized no matter how many degrees they are rotated, such as: shovels, screwdrivers, etc.

3 Algorithms

3.1 Linear SVM as Baseline Model

Linear SVM is used as our baseline model before data pre-processing. After data pre-processing, we chose a rbf kernelized SVM model, trained it under different value of hyper-parameter C to get the best prediction possible. The performance of Linear SVM and kernelized SVM will be reported later in the ”Results” section.

3.2 CNNs as Competitive Models

Convolutional neural network is competitive for image recognition tasks. We implemented a tiny residual CNN with 5 convolutional layers where the shortcut is connected between the input and the second convolutional layer’s output. We named it Net5.

3.3 Bagging

Although it took us much time finding a best parameter set of Net5, We still obtained several well trained Net5s with different dropout configurations on different train / validate datasets. The predicting of all these well trained models are very good resources for bagging. We built two accesses to get the final predict of bagging: one access is to directly combine the predictions of different models, and the other access is to import the different predictions from a submission files(.csv), given the name lazy bagging by its inventor: Lifeng Wan.

4 Methodology

4.1 Training / validation split

Before performing data augmentation, we need to split the whole training dataset which contains 10000 training samples. Usually, we split the whole training dataset into 8000 training samples and 2000 for validating. When training different models, the whole dataset is shuffled before splitting to the two datasets.
Sometimes, we seperated 1000 samples from the validating set as our private test samples to test the performance of ensemble models and of our best models which are selected by the remaining 1000 validating samples.

4.2 Hyper-parameters

In Net5, hyper-parameters are set to the following values: kernel size: 3x3; padding = 0 or 1; stride = 1; max_pool kernel size: 2x2 with stride: 2; 1000 neurons in hidden full connected layer with a dropout probability of 0.5.
Learning rate is initially set to 0.001 and has a schedule to be reduced; batch size = 128; Weight decay = 0, and early stopping is True.

5 Results

5.1 Linear SVM

Linear SVM, as the baseline model, achieved an accuracy around 4.60% before data pre-processing, and climbed to an accuracy of 48.40% when kernelized by rbf and C = 2.0 on pre-processed dataset.

5.2 Net5

Net5 is our the most competitive model architecture, which brought us best performance with 84.93% accuracy on our validate set and 80.90% on public test dataset.

Figure3. learning curve of Net5

5.3 Ensemble of Net5

35 well trained Net5 models were used to make an ensemble prediction with bagging. the accuracy was improved to 83.40% on public test dataset.

5.4 Other Models

We also tried MLP with two hidden layers, and it provided us an accuracy around 50-60%.

6 Discussion

1. Data pre-processing is very important. Without any data processing, linear SVM only got an accuracy 4%. After pre-processing data, the accuracy increased to 45%. Noise reduction brought us a relatively clean training set; resizing the training samples to be a small size allows us to benefit from a small CNN model, thus then it speeds up the training.
2. Data augmentation is useful. Reasonable data augmentation is always useful; but sometimes, augmented data may not comes from the true distribution of training and testing dataset, which should be avoided, In our codes, we set many switches to open or close one type of data augmentation. We can run a model again and again with different settings to observe the difference with each or combinations of different data augmentations.
3. Model selection is crucial. Some models have their own bottle neck, limiting them from getting better predictions no matter how data are pre-processed. For example, the prediction accuracy of a simple MLP is only around 50 - 60%. CNN is the best model for image recognition tasks. Detection and abstraction layer by layer of Spatial features are the two key points for CNN models to overbear others.
4. Finding a good model architecture is not easy, so is tuning hyper-parameters. Even now, we believe Net5 could still be improved if more tuned.
5. Bagging is useful. Compared with the best predictor of single well trained models, Bagging still significantly elevated the accuracy.

Through this competition, many theoretical knowledge become more impressive and practical to us.

7 Statement of Contributions

Defining the problem: Lifeng, Wan; Qiang, Ye; Jinfang, Luo
Developing the methodology: Qiang, Ye; Lifeng, Wan; Jinfang, Luo
Coding the solution: Qiang, Ye; Lifeng, Wan; Jinfang Luo
Performing data analysis: Jinfang, Luo; Lifeng, Wan; Qiang, Ye
Writing the report: Jinfang, Luo, Qiang, Ye; Lifeng, Wan
We hereby state that all the work presented in this report is that of the authors.