COMP5329 - Deep Learning Assignment 2 Report

Zhecheng Zhong(SID:490319299, Unikey:zzho7727)

Semester1 2020

1 Introduction

1.1 Aim of study

Given a set of images, with one or more labels and a short caption that summarizes the image, the goal of this project is to implement and combine an image classifier containing recurrent neural networks (RNNs) (LSTM and GRU) for caption-label analysis with EfficientNet convolution networks for multi-label image classification.

1.2 Importance of the study

There are several ways to encode the label, we simply transfer labels to one-hot form and decrypt during predictions. While deep convolution neural networks (CNNs) has shown great achievements in single-label image classification, it is significant to recognize that real-world images generally contain multiple labels, which could correspond to different objects, scenes, actions and attributes[2]. EfficientNet improves both the efficiency and accuracy on the baseline network. We embed it to our project with RNN analyzing the caption. Both of them produce the probability matrix. The output comes with a combination result.

2 Related works

This section looks back at the related research on architecture and framework. The core part is having EfficientNet as image training model and cooperate with caption classification.

2.1 EfficientNet

In the paper given by Mingxing Tan and Quoc V.Le[3], they propose a new scaling method which scales all dimensions and design a new baseline network model, called EfficientNets. As an outstanding example, EfficientNets -B7 achieves 97.1 percentage accuracy on ImagaeNet, while being six times faster than the best existing ConvNet.

The most common and traditional way of scaling up ConvNets is by depth[1], width[6] or resolution. The problem is that conventional ways only scale one of these three dimension, never quantify the relationship among them. The author found that scaling up any dimension of network width, depth, or resolution improves accuracy, but the accuracy gain diminishes for bigger models. Besides, it is practically critical to balance all dimensions of network width, depth, and resolution during ConvNet scaling. As a solution to these problems, researchers propose a compound scaling method, using a compound coefficient ϕ to uniformly scales network where depth: d = α^{ϕ}, width: w = β^{ϕ}, and resolution: r=γ^{ϕ}, where a, b, y are constants that can be determined by a small grid search.

In EfficientNets, they leverage a multi-objective neural architecture search to develop the baseline network. Take EfficientNet-B0 as an example. They apply one Conv3x3, six MBConv accompanying with pooling and FC layers. In conclusion, powered by this compound scaling method, a EfficientNet model can be scaled up very effectively with decent high accuracy.

2.2 EfficientNet

Since images are giving more than one core information, multi-label classification task is receiving increasing attention[5]. General methods of dealing the multi-label, is to transform the multi-label into multi single-label and training it with the ranking loss or corss entropy loss. However, this method ignores the strong label co-occurrence dependencies[4], since usually, sky and cloud are coming together.

In this paper[2], researchers utilize recurrent neural networks (RNNs) to model the label dependencies, obtained higher-order label relationships while keeping the computational complexity tractable. And they design the RNNs to adapt the image features by encoding the attention information, thus creating a CNN-RNN model. Another topic discussed in the paper is handling labels' overlapping.

A unified CNN-RNN framework asks CNN for help first. CNN generates image embedding vectors, following RNN learning a low-dimension image-label joint matrix, and model the semantic relevance between them. The high-order label co-occurrence dependency in this low-dimensional space is modeled with the long short term memory(LSTM) recurrent neurons.

3 Techniques

In this section, we dive to the details of project design and implementation. Starting from explicitly conceptual architecture design. Then follows the label encoding and the principle of methods chosen in the project.

3.1 Architecture

The project architecture is shown in figure 1. Here an image is firstly processed and transformed into a restricted rotated size (224x224). And via EfficientNet model, it then generates a probability tensor(matrix) applying sigmoid as activation function and BCEloss as multi-label loss function. Each elements in the predict result representing a probability which is likelihood to be at a specific label.

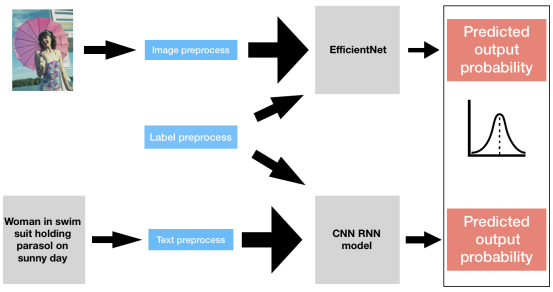

Figure 1: High level project architecture

ImageID	Label	Encoding
0.jpg	1	1000000000000000000
1.jpg	1 19	1000000000000000001
2.jpg	1	1000000000000000000
3.jpg	8 3 13	0010000100001000000

Figure 2: Label encoding

The bottom streamline reveals the process of caption-label processing. Via creating a vocabulary we use the features of LSTM cell to construct an output tensor. Similarly as mentioned in the above, the tensor records a probability of label. Finally, we joint these two tensor together. By setting a threshold value, each element(with probability) is re-assigned with value one or zero(the label encoding details will be explained in next subsection). And finally get the result.

3.2 Label encoding

The original label format contains one or more number ranging from 1 to 19, separating by empty space. Label encoding processing generates the label into one-hot form, whose size is 19 binary numbers(figure 2). Every digit of the 19-dimension binary represents a valid label. However, this label processing does not take the label-co-relation into consideration and ignoring the CNN-RNN benefits. More discussion about this will be in latter part.

3.3 Image procession: EfficientNet

While determining the structure of convolution networks, it is commonly agreed that a larger scale networks usually implies to a better training model. However, one can be on the horns of dilemma that how to define a better scale network, and how to measure the running time and storage cost. EfficientNet is the solver here. In the project we implement EfficientNet-b4 which has a better performance with a reasonable size (less than 100MB).

3.4 Activation and Loss Function

The collaboration information output of Sigmoid is Bernoulli distribution. Each elements illustrates the probability of having that value. Multi-label classification should use sigmiod as activation function, and binary cross entropy as corresponding loss function. Pytorch offers BCELoss().

3.5 Caption Procession: RNN(with LSTM

RNN maintains internal hidden states to model the dynamic temporal behaviour of sequences with absolute arbitrary lengths. As an inprovement, LSTM extends RNN by putting in three more gates to an RNN neuron: a forget gate to decide whether to forget the current state; an input gate to indicate if it should read the input; and an output gate to control whether to output the state. LSTM takes the embedding of the predicted label at each time step and maintains a hidden state to model the label co-occurrence information.

4 Experiments

4.1 Results

4.1.1 Results - Accuracy

For each caption and image classification, they produce probability label matrix and predict classification results. As shown in figure 3, the RNN caption classification itself does not have good test results since only implementing caption classification bringing weak accuracy to the multi-label image classification. EfficientNet-b0 and EfficientNet-b4 having a decent performance. Besides, combining RNN with CNN, the system model reaches another high performance level.

Model	Model Size	Train Accuracy	Loss	Time Cost	Test Accuracy
RNN(LSTM)	-	46.4%	-	< 1 min	65.4%
EfficientNet-b0	23MB	66.73%	0.1968	< 4 hours	83.8%
EfficientNet-b4	74.4MB	68.75%	0.1962	<20 hours	85.2%
EfficientNet-b0 + RNN	23MB	72.57%	-	< 4 hours	84.9%
EfficientNet-b4 + RNN	74.4MB	79.68%	-	<20 hours	87.8%

Figure 3: Summary of Experiment Result

4.1.2 Results - Efficiency

Running an image classifier is time consuming, images would go through tens of convolution layers to process. As a results, time is proportional to the scale of networks. As figure 3 illustrated, even EfficientNet-b4 bringing a little better results, the time consumption is mammoth and not feasible nor consistent. On the other site, EfficientNet-b0 gives still decent accuracy within training in four hours.

4.2 Extensive Analysis

4.2.1 Hyper Parameter Analysis

Since the data set has approximate 30,000 samples, the batch size and number of epoch mainly decide the time cost of the training process. Usually the batch size is preferred to be at some 2 to n power value. The total time cost is acceptable when batch size = 16. For some model, we set epoch to be 5 to 10 where some denser layer EfficientNet, we prefer the epoch to be 1 in most cases. The learning rate is compared between 0.01, 0.001, the evaluation of 0.01 tends to have a slightly better performance than the others.

5 Conclusion

In this experiment, we generate a CNN-RNN model including the EfficientNet and LSTM, for multi-label

classification. The total training data is approximate 30,000 images, and the output data achieve around 0.88 mean F1 scores. The training process is within 4 hours using common laptop with prediction process within 3 hours.

6 Other (LATEX)

LaTeX is a high-quality typesetting system; it includes features designed for the production of technical and scientific documentation. This report is written and compiled using online LATEX application.

7 Instructions on how to run the code

To start with, split images into test image and train image, separately putting them under the "data/image/test_image" and "data/image/train_image". For the train process, go to train_efficientnet and run the python file, the output model will be stored under the Weights directory. In the process of prediction, run inference.py, the result will be stored in output folder.

References

[1] Zhang X. Ren S. He, K. and J. Sun. *Deep residual learning for image recognition.* 2016.

[2] Yi Yang Jiang Wang. *CNN-RNN: A Unified Framework for Multi-label Image Classification.* 2016.

[3] Quoc V. Le Mingxing Tan. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.* ICML 2019, 2019.

[4] J. Zhang B. Wu J. Fan X. Xue, W. Zhang and Y. Lu. *Correlative multi-label multi-instance image annotation.* 2011 IEEE International Conference on, pages 651–658., 2011.

[5] T. Leung A. Toshev Y. Gong, Y. Jia and S. Ioffe. *Deep convolutional ranking for multilabel image annotation.* 2013.

[6] S. Zagoruyko and N. Komodakis. *Wide residual networks.* 2016.