Using IBM Watson and TensorFlow to Build Landmark Classifier

Ping-Han Lei
University of Washington-Tacoma
leiph@uw.edu

Eyhab Al-Masri
University of Washington-Tacoma
ealmasri@uw.edu

ABSTRACT

Machine learning and Deep learning have plenty of applications nowadays, especially in image data like facial recognition and object detection. One of the fields that are closely related to everyone's life yet not being under the spotlight is landmark recognition.

The purpose of this project is to explore the methods and tools and to build a well-performed classifier for landmarks. In this project, we explored a major cloud service: IBM Watson for this task. Also, we build a landmark classifier with the help of TensorFlow and a Convolutional Neural Network model named Inception-ResNet model.

KEYWORDS

Machine Learning, Deep Learning, CNN, TensorFlow, IBM Watson, Cloud Service

1 INTRODUCTION

During the last few years, Data Science and Machine Learning become one of the hottest fields for research and also for industry. More and more applications utilize the power of machine learning to get insights of data. In the branch of machine learning which deals with images and visual data, researchers have made tremendous progress and now can tackle complicated problems thanks to the fast development of computational ability in computers and the machine learning tools and libraries.

Now we can render the power of machine learning to teach machines to recognize not only objects but also landmarks in image or visual data.

In this project, we explore the famous and widely used machine learning framework: TensorFlow, and one of the major cloud services for machine learning: IBM Watson Visual Recognition, then we build classifiers with both of them and examine the performances.

2 Related Works

2.1 Deep Learning

Deep learning is part of the family of machine learning, it has been widely used in fields like Computer Vision and Natural Language Processing and Bioinformatics. Just like machine learning, there are two types of learning for deep learning: supervised and unsupervised. This project aims to build a classifier hence we use supervised learning techniques.

Deep learning uses series of layers of nonlinear processing units to extract useful features based on the training data, then add in transformation functions for the further learning [7]. Each layer will take in the output from previous layer and see it as input.

Deep learning architectures have been showing impressive performance in image related problems, especially vision classification problems [4, 5, 6].

2.2 Convolutional Neural Network

Convolutional Neural Network (CNN) is one of the classes of feed-forwarding neural network. CNN architecture is made of one Input Layer, one Output Layer and several hidden layers that lay between input and output. The hidden layers contains a variation of layers including Convolution Layer, Pooling Layer, Normalization Layers and Fully Connected Layers.

The major advantage of CNNs is that it requires little pre-processing effort compared to other image classification algorithms.

There are several models have been proved to be effective for the researches of image recognition or object recognition. In this project, we use the pre-trained Inception-ResNet-v2 model to be the foundation of our landmark classifier. Inception-ResNet-v2 model has been proved as a state-of-the-art model in ILSVRC 2012 image classification benchmark.

By combining residual connections with Inception architecture, the training of the model can be significantly accelerated [3].

2.3 Cloud-based Machine Learning Services – IBM Watson Visual Recognition

IBM Watson is a famous AI-based cloud platform developed by IBM. It not only allows you to build your own machine learning models on their cloud platform, you can also use their pre-trained models to create custom models and classifiers with little efforts. IBM Watson provides powerful computation ability for you to quickly create, evaluate and manage your models [2].

Visual Recognition Service is part of IBM Watson, it understands the contents of images and it could give you the insights of your visual data. You can also make use of the powerful pre-trained model within IBM Watson and train it to learn to recognize your custom objects/labels.

2.4 TensorFlow

TensorFlow is an open source machine learning framework developed and maintained by Google [1]. It is designed and developed by Google Brain team within Google AI's organization to do high performance numerical computation.

Since TensorFlow is open sourced and it has an active and robust community, it now supports deep learning and you can use the numerical computation core across many other domains including image processing and computer vision.

3 Experiments

3.1 Experiment using IBM Watson

We use IBM Watson Visual Recognition service to build custom classifiers with Watson's pre-trained model. We use IBM Watson's API to communicate with the service and extract the information we need from the responds which is in JSON format.

In Experiment 1, we trained the classifier with 18 labels and 100 images for each label, makes it 1800 training images in total. The ratio of train and validation is 90% and 10% in this experiment. Thus there are 200 validation data.

In Experiment 2, we want to see how the model performs with only small size of training data but large number of labels. We trained the classifier with only 10 images for each label, and there are 40 labels in total.

3.2 Experiments using TensorFlow on local computer

Other than using the very powerful and fast cloud service provided by IT giant to build the classifier, we also built a classifier with smaller scale of dataset on a local machine. We use TensorFlow, a well-known machine learning framework to build our classifier.

The reason we choose TensorFlow for this project is that there are many pre-trained models could be used on image processing and classification. By using pre-trained models, we don't have to reinvent the wheel and train the model from scratch. It's very time-consuming to train from scratch when your local machine doesn't have powerful GPU in it. Pretrained models are already trained by a large size of dataset and researchers already spent very long time to train those models. We can do transfer learning with TensorFlow and easily utilize these pre-trained models to make our customized classifier.

In experiments using local machine, we use Inception-ResNet-v2 as our pre-trained model, and we train the model with our landmark dataset to build a classifier. The dataset in this experiment contains 7 labels and there are 200 images for each label. We split the data into 70% of training data and 30% of validation data. The image size is 128x128, we make it small to speed up the training and validation process.

In the first few experiments, we run the training process with 6 epochs to quickly have some preliminary understanding of how the model works with our landmark dataset. We found that although small number of epoch can make the training process finish in short period, the validation results showed that the classifier was overfitted by our training data.

After we ran the fewer epochs experiments, we increased the training epochs to see if we can eliminate the effect of overfitting. We then trained the classifier with the same training data but in 20 epochs rather than just 6 epochs. From the validation results, we found that the training accuracy and the validation accuracy are close and both achieved very high percentage.

4 RESULTS AND DISCUSSION

4.1 Results of Experiments with IBM Watson

Experiment #	Number of Labels	Size of Training Data	Validation Accuracy
Experiment 1	18	100 per label	98.33%
Experiment 2	40	10 per label	89.54%

From the results of Experiment 1 and 2 we can see that the classification performance of IBM Watson is excellent, especially for the first experiment.

The validation accuracy in experiment 2 went down to 89.54% comparing to our first experiment with Watson. The classifier in Experiment 2 still works really well considering how small the training dataset is.

4.2 Results of Experiments with TensorFlow

Experiment #	Number of Epoch	Learning Rate	Training Accuracy	Validation Accuracy
Experiment 3	6	0.0002	84.39%	67.27%
Experiment 4	6	0.002	72.30%	13.43%
Experiment 5	20	0.0002	93.50%	97.71%
Experiment 6	20	0.0001	91.54%	97.93%
Experiment 7	20	0.00002	83.22%	96.79%

4.2.1 Overfitting

From the results of Experiment 3 and 4, we can see there could be an overfitting issue with less training epochs. We have decent training accuracies but much lower validation accuracies.

4.2.2 Underfitting

From the results of Experiment 5, 6 and 7, we can see there could be an opposite issue: Underfitting, comparing to prior experiments. Although we have very high accuracies in both training and validation, the training accuracies are lower than the validation accuracies.

Also, from the figures 1 to 3, we can see that the loss of training process drops normally until certain point. There are uprising curves during the dropping of the loss. In our experiments, neither increase nor decrease the learning rate could eliminate the little hill in loss curves.

Figure 1. Loss Curve of Experiment 5

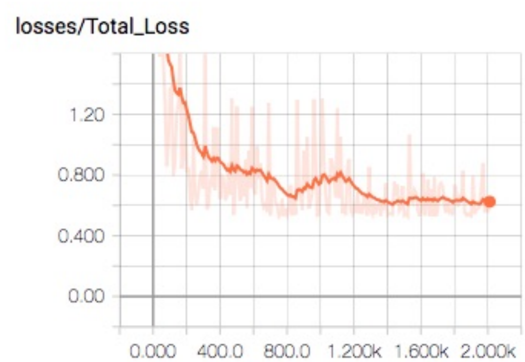

Figure 2. Loss Curve of Experiment 6

Figure 3. Loss Curve of Experiment 7

4.2.3 Effect of Epoch

From the results of Experiment 3 and 5, with everything be the same except the number of epoch, we can see that when we train the model with more epochs, the accuracies in both training and validation

are significantly higher, which means the classifier performs better.

4.2.4 Effect of Learning Rate

From the results of Experiment 5, 6 and 7, we can see that the lower learning rate could causes lower training accuracy. We didn't find a specific effect of learning rate on validation accuracy. In our experiments, we couldn't eliminate the underfitting by having lower learning rate.

5 CONCLUSIONS

In summary, although our best result in Experiment 5 has signs of slightly underfitting, we can still say that the Inception-ResNet model can bring us desirable classification accuracy with appropriate parameter. Our classifier depends on a relatively small size of dataset, we have 200 images for each label and we have 7 labels in total.

The image data we use has very compact size, which is 128x128. Comparing to larger image size like 800x600 or 1920x1080, the compact size of images could lead to loss of information. However, our classifier still achieves high training accuracy (93.5%) and validation accuracy (97.71%).

In this project, we successfully build a landmark classifier that can make use of small size images and recognize the landmark in the image, then classify the images to correct labels.

REFERENCES

[1] TensorFlow Website https://www.tensorflow.org

[2] IBM Watson Website https://www.ibm.com/watson/

[3] Szegedy, Christian & Ioffe, Sergey & Vanhoucke, Vincent. (2016). *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.* AAAI Conference on Artificial Intelligence.

[4] Nguyen A, Yosinski J, Clune J. *Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images*. In Computer Vision and Pattern Recognition (CVPR '15), IEEE, 2015.

[5] Y. Bengio. *Learning deep architectures for ai*. Foundations and trends in Machine Learning, 2(1):1–127, 2009.

[6] G. E. Hinton. *Learning multiple layers of representation*. Trends in cognitive sciences, 11(10):428–434, 2007.

[7] Deng, L.; Yu, D. (2014). *"Deep Learning: Methods and Applications"* (PDF). Foundations and Trends in Signal Processing. 7 (3–4): 1–199. doi:10.1561/2000000039.