Google AI Open Images - Visual Relationship Track

Winning Model Documentation

Author

User Name : toshif https://www.kaggle.com/toshif

Full Name : Toshiyuki Fukuzawa

Email : tfukuzawa@live.jp

Tel : 44 (UK) 7493 581306

A. MODEL SUMMARY

A1. Background on you/your team

•Competition Name: Google AI Open Images - Visual Relationship Track

•Team Name: toshif

•Private Leaderboard Score: 0.22832

•Private Leaderboard Place: 4th

•Name: Toshiyuki Fukuzawa

•Location: London UK

•Email: tfukuzawa@live.jp

A2. Background on you/your team

•What is your academic/professional background?

I studied Physics at Tokyo Institute of Technology to obtain my master degree and am currently a software engineer.

•Did you have any prior experience that helped you succeed in this competition?

No. In general, my prior Kaggle competitions helped me learn a lot and achieve this success.

•What made you decide to enter this competition?

The high quality dataset of Open image V4 and GCP 500 Credits.

•How much time did you spend on the competition?

I worked on the competition for a month, spending 2 hours per day on average.

A3. Summary

My approach consists of 2 models. The first model is to solve "is" visual relationship. The second is "non-is". I took this way as I wanted to start simple in the beginning and to leverage a model established for the object detection track.

1. “is” relation

The distinct triplets of “is” relation were only 42 therefore I treated each of them as a separate class and trained SSD Resnet50 (Retinanet) with the 42 target classes. I chose this model as it was relatively lightweight and it was compatible with GCP TPU.

https://github.com/tensorflow/models/blob/master/research/object_detection/samples/configs/ssd_resnet50_v1_fpn_shared_box_predictor_640x640_coco14_sync.config

In the given triplet labels challenge-2018-train-vrd.csv, there were 194K “is” relations out of 374K samples. I generated my “is” only model training set based on these triplets for the positive labels, mixing-in 10K images which don’t have any “is” relations as negative labels.

This first model gave me 0.09 on LB.

2. “non-is” relations

Next, I moved to “non-is” relations such as “under”, “on”, “holds” etc. I trained another Retinanet with all the 62 target classes just to detect single boxes. This second training set was all the images with challenge-2018-train-vrd-bbox.csv positive labels. Then I calculated the probabilities of possible valid box combinations in each image for the most 100 confident boxes. In other words, for each image, I got the most confident 100 combinations out of 10000 combinations of boxes (100 x 100). I used this formula to estimate the confidence for each combination C_c.

C_c = F_r x sqrt (C_box1 x C_box2)

Where C_box1 and C_box2 are the confidences for each box given by the box detection model. F_r is the relationship coefficient function for each box1-box2-relation combination. I used GBDT (LightGBM) to calculate this coefficient using simple features like box labels, relation label, Euclidean distance of boxes, relative distance (distance divided by sum of total box area), relative x-y position of box1 to box2 and raw box cordinates.

This second model gave me 0.16 on LB.

3. Ensemble

This is just a concatenation of the first and second model outputs. This is simply ok as they detect different types of relations. There shouldn't be any conflict or performance degradation.

The result was 0.25 on LB.

This summary also has been posted onto Kaggle Discussion .

https://www.kaggle.com/c/google-ai-open-images-visual-relationship-track/discussion/64642

A4. Features Selection / Engineering

For the main models of neural nets, I didn’t do any feature engineering. The inputs were the given 500GB+ images and labels. I didn’t use any external data.

For the sub model of LightGBM part, I used the simple features mentioned above but this part was far less important than the main models.

A5. Training Method(s)

•What training methods did you use?

I trained my 2 Retinanets on TPUs on Google Compute Platform. The training configuration is the same as this config file on github.
https://github.com/tensorflow/models/blob/master/research/object_detection/samples/configs/ssd_resnet50_v1_fpn_shared_box_predictor_640x640_coco14_sync.config

•Did you ensemble the models?

Yes, I just concatenated 2 models’ outputs as mentioned above.

•If you did ensemble, how did you weight the different models?

The 2 models were with the same weights.

A6. Interesting findings

•What was the most important trick you used?

* trained 2 neural nets models (“is” and “non-is” relations)

* used GBDT to calculate confidence coefficient

•What do you think set you apart from others in the competition?

* The training datasets which I generated from the original dataset for the 2 neural nets.

* Neural net architecture (Retinanet SSD Resnet50)

* used GBDT (LightGBM) to calculate the confidence coefficient

* Other Notes

My local environment was MacBook Pro and I managed everything on GCP within the given 500 GCP credits. This was challenging but it proved that if you use GCP resources efficiently, you can compete and win a prize of a competition even with such a huge dataset.

A7. Simple Features and Methods

Many customers are happy to trade off model performance for simplicity. With this in mind:

•Is there a subset of features that would get 90-95% of your final performance? Which features? *

No. Feature engineering was not important as the main models were neural nets.

•What model that was most important? *

Retinanet

A8. Model Execution Time

Many customers care about how long the winning models take to train and generate predictions:

•How long does it take to train your model?

“is” relation Retinanet : 6 hours on GCP TPU

“non-is” relation Retinanet : 12 hours on GCP TPU

•How long does it take to generate predictions using your model?

18 hours on a 2 core CPUs machine with a Nvidia Tesla K80 GPU. It can be improved with a better resource. I chose a cheap option as I wanted to manage everything within the 500 GCP credit.

•How long does it take to train the simplified model (referenced in section A6)?

The model is already simple. I chose a lightweight neural net architecture to mange everything within the given 500 GCP credit.

A9. References

Citations to references, websites, blog posts, and external sources of information where appropriate.

https://storage.googleapis.com/openimages/web/challenge.html

https://github.com/tensorflow/models/tree/master/research/object_detection