2016 LUNA Challenge Description Document

January 18, 2017

Abstract

This document contains brief introduction to models we applied and the fully detailed experiment procedure description.

1 Introduction to Models

We adopt deep neural network model into detection tasks, based on Overfeat[4], TensorBox[6] with a specially designed loss function. For CNN substructure to extract features from raw image input, Google Inception V1[7] is used. LSTM[2] is used for detection and merge object proposed by region proposing network. This model is built and running using Tensorflow[1].

Inception V1 are designed for ImageNet challenge which is highly complicated. With such complexity, it would be easily to overfit the data since the task is considerable simple. We fine-tune the structure and choose lower layers architecture for feature extractor. Also larger dropout ratio is used for better generalization. Adam[3] is used for training with default learning rate and other parameters.

2 Experiment Design Details

We use the dataset provided by LUNA[5]. The dataset is divided into 10 subsets with annotations given. The procedures for training and validation are described below:

1. Slice all the data into 512×512 sized image noted as I, using annotations given in the dataset, and here we use I_i for images set from the ith subset. And slices with annotations are noted as A and A_i for ith subset.

2. Perform CV of i round using all sliced image with annotation A **except** the i-th subset A_i, with **argumentation using rotation and flipping**.

3. After training of each round, model learned is tested and evaluate on the images sliced from CT file **without** annotations.

4. We use the same hyper-parameters for training on each subset. The only variable is the number of training epochs.

5. Model and results submitted is the best average result from LUNA evaluation scripts on all subsets of the **same training epoch cycles** .

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[3] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[4] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[5] A. A. A. Setio, A. Traverso, T. de Bel, M. S. Berens, C. v. d. Bogaard, P. Cerello, H. Chen, Q. Dou, M. E. Fantacci, B. Geurts, et al. Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the luna16 challenge. *arXiv preprint arXiv:1612.08012*, 2016.

[6] R. Stewart and M. Andriluka. End-to-end people detection in crowded scenes. *arXiv preprint arXiv:1506.04878*, 2015.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.