Writing Scientific Papers and Software

Cheng Soon Ong
Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—A critical part of scientific discovery is the communication of research findings to peers or the general public. Mastery of the process of scientific communication improves the visibility and impact of research. While this guide is a necessary tool for learning how to write in a manner suitable for publication at a scientific venue, it is by no means sufficient, on its own, to make its reader an accomplished writer. This guide should be a starting point for further development of writing skills.**

I. Introduction

The aim of writing a paper is to infect the mind of your reader with the brilliance of your idea [1]. The hope is that after reading your paper, the audience will be convinced to try out your idea. In other words, it is the medium to transport the idea from your head to your reader's head. In the following section, we show a common structure of scientific papers and briefly outline some tips for writing good papers in Section **??**.

At that point, it is important that the reader is able to reproduce your work [2], [3], [4]. This is why it is also important that if the work has a computational component, the software associated with producing the results are also made available in a useful form. Several guidelines for making your user's experience with your software as painless as possible is given in Section **??**.

This brief guide is by no means sufficient, on its own, to make its reader an accomplished writer. The reader is urged to use the references to further improve his or her writing skills.

II. Models and Methods

Describe your idea and how it was implemented to solve the problem. Survey the related work, giving credit where credit is due.

III. Results

Show evidence to support your claims made in the introduction.

IV. Discussion

Discuss the strengths and weaknesses of your approach, based on the results. Point out the implications of your novel idea on the application concerned.

V. Summary

Summarize your contributions in light of the new results.

VI. Obtaining Heat-Maps

In this section we are going to describe the first component of our framework which is a method to transform a given input image into (how we call it) a *Road Heat Map*. More precisely suppose an image I is given of dimensions $n \times n$, then the output of our method is an array $x \in [0,1]^{n \times n}$ with the intent that $x(i,j)$ corresponds to the probability that the pixel (i,j) is part of a road. Clearly this is the essential part of the task and the accuracy here crucially affects the final accuracy of our method, no matter how we proceed after that.

A. Basic CNN

The provided sample code *tf_aerial_images.py* of a Convolutional Neural Network for solving this task serves as good starting point for our final approach. Let us describe it briefly here. To compute a prediction for a given input picture of dimension $n \times n$, the picture is first divided into $\frac{n}{16} \times \frac{n}{16}$ chunks of size 16×16 each. Subsequently, every such chunk C is being input into a neural network $\mathcal{N}$, which outputs a number $p(C) \in [0,1]$ which is then used as the road heat map value for the corresponding 16×16 chunk. What is left to explain is how is $\mathcal{N}$ constructed and how is it trained. The network $\mathcal{N}$ has the following layers: (*Convolutional, RELU, Max Pooling*) repeated twice in this order. To calculate losses, the cross entropy function is used. The network is trained using training pairs (x,y) (input and output) of the form: x is a 16×16 chunk, part of a training picture and $y \in \{0,1\}$ is a suitably rounded mean of the corresponding ground truth chunk.

To deal with the network $\mathcal{N}$ efficiently, the *TensorFlow* library is used. In particular, crucially, this provides us with a ready-to-apply optimization primitive *MomentumOptimizer*. We do not go into details on the choice of parameters on which the method is invoked.

B. Big CNN

Our way of obtaining Heat-Maps builds upon the approach described above, but includes several new insights and modifications which we are going to describe below and briefly discuss how do they affect the results and accuracy.

The first modification we employ concerns the general philosophy how the heat map is obtained. Note that in the basic approach above, the heat map is "discrete" in the sense that it is constant on 16×16 chunks. Another disadvantage is that the result for a single chunk depends *only* on the pixels within it, not at all on the neighboring ones, whereas in reality such a prediction cannot be performed locally.

To fix these issues we propose and implement a different method for computing the Heat Map. For every pixel (i, j) in the picture it takes its square neighborhood of size 48×48

VII. The Post-Processing Phase

Acknowledgements

References

[1] S. P. Jones, "How to write a great research paper," 2008, microsoft Research Cambridge.

[2] M. Schwab, M. Karrenbach, and J. Claerbout, "Making scientific computations reproducible," *Computing in Science and Engg.*, vol. 2, no. 6, pp. 61–67, 2000.

[3] J. B. Buckheit and D. L. Donoho, "Wavelab and reproducible research," Stanford University, Tech. Rep., 2009.

[4] R. Gentleman, "Reproducible research: A bioinformatics case study," *Statistical Applications in Genetics and Molecular Biology*, vol. 4, no. 1, 2005. [Online]. Available: http://www.bepress.com/sagmb/vol4/iss1/art2