Fall 2018

Reading Responses To Journal Articles, Computational Emulation Of Published Research

Vidya Ganapati
Swarthmore College, vganapa1@swarthmore.edu

Reading Response Template: These are general guidelines, you may customize as you see fit; complete 1 reading response for each week's group of papers.

Journal Paper Information: (complete the following for each paper)
Author(s):
Date of publication:
Title:
Journal:
DOI:
Link (if applicable):

Key Words:

General subject:

Specific subject:

Hypothesis:

Methodology:

Result(s):

Summary of key points:

Context (how this article relates to other work in the field; how it ties in with key issues and findings by others):

Significance and broader impact to society:

Relevant sketches:

Cited References to follow up on:

Other Comments:

E30: Computational Optics

Lab 1, due 9/27

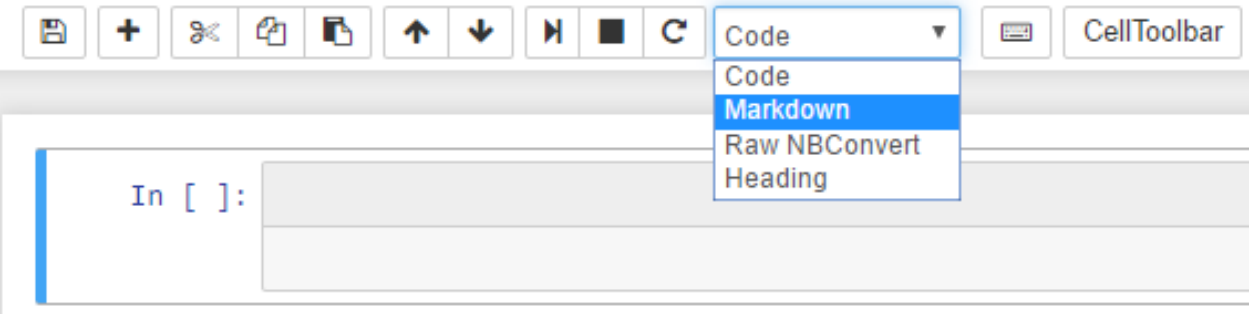
Objective

In this project, you will strengthen your programming skills and explore numerical scalar optics simulations. The functions and code you create in this lab will be a foundation you can build on for later labs. This lab is based off of the book "Computational Fourier Optics," posted on Moodle.

Logistics

For this project, you should use an IPython notebook to write your code and analysis. For details on installation and use, see Linge and Langtangen, p. 215.

To answer the text questions, change the cell from "Code" to "Markdown" as shown below:



If you run a Markdown cell (Shift-Enter), it will display regular text. For more details on formatting your text with Markdown, follow the following link:

http://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Working With Markdown Cells.html

For all code, include descriptive comments.

Please submit BOTH your IPython notebook file (.ipynb) and a PDF of your IPython notebook to Moodle by 5 pm on 9/27.

To save as PDF, go to File --> Download as --> PDF via LaTeX (.pdf)

Tasks

All chapter references are to the "Computational Fourier Optics" book posted on Moodle. Though this book uses Matlab, please use python for the following tasks. You may find the

numpy, scipy, and imageio packages helpful. For a refresher on python, see the book by Linge and Langtangen, also posted on Moodle.

Read Chapter 2 and complete the following exercises at the end of the chapter:

Exercise 2.1
Exercise 2.3
Exercise 2.4

Read Chapter 3 and replicate the following figures using python/numpy/scipy (and any other python packages you find helpful):

Figure 3.6
Figure 3.9
Figure 3.11

Complete the following exercise at the end of Chapter 3 again using python:

Exercise 3.3

Read Chapter 5 and replicate the code from the following sections in python:

5.1 Fresnel Transfer Function (TF) Propagator
5.2 Fresnel Impulse Response (IR) Propagator

Replicate the following figure:

Figure 5.5

Complete the following exercises at the end of Chapter 5:

Exercise 5.1
Exercise 5.3

Read Chapter 7. Download the Air Force Target image posted on Moodle to complete the following (you may find the package imageio helpful for uploading the image into your python code):

Replicate the following figures:

Figure 7.7
Figure 7.12

Complete the following exercise at the end of Chapter 7:

Exercise 7.9 (use python)

E30: Computational Optics
Lab 2, due 11/1 at 5 pm

Objective

In this lab, you will computationally optimize the phase pattern on a spatial light modulator in the Fourier plane of a 4-F system. Your goal is to find the phase pattern that allows for the best 3-D localization of a single point emitter.

You will use deep learning to figure out the optimal phase pattern. In this lab, you will gain experience in using TensorFlow, strengthen your knowledge of numpy, and further develop your understanding of optics in the scalar wave domain.

Logistics

For this project, you can use the Python code editor of your choice. You should submit both your code (.py files) and your writeup (.pdf) on Moodle by 5 pm on 11/1.

For all code, include descriptive comments. You may work in a pair or triple.

Overview

Note: For computationally intensive code, it is best to run code on the command line (Terminal for MacOS; please come see me if you are unsure how to do this on your computing setup).

You may need to reference the TensorFlow documentation (https://www.tensorflow.org/api_docs/python/tf) for this lab. Also, don't be shy about using Google Search and Stack Overflow to learn about TensorFlow and find the appropriate TensorFlow op.

In the following steps, you will write code for a neural network that does the following:

1. Takes as input the coordinates of a randomly selected point x, y, z
2. Finds the image (intensity) of the point once it goes through a 4-F system with a phase only spatial light modulator with a given numerical aperture
3. Add Poisson noise to the image
4. Put the image through a series of convolutional layers
5. Puts the output of the convolutional layers into a dense layer that outputs the guess of the coordinates x, y, and z
6. Trains the weights, biases, kernels, and phase of the spatial light modulator, in order to minimize the squared error between the actual x, y, z coordinates and the guess of the x, y, z coordinates

Tasks

1. Sketch out pseudocode or a diagram for the neural network.
    a. Plan out the following for your TensorFlow graph:
        i. What are your trainable variables?
        ii. What are your placeholder variables?
        iii. What are your constant variables?
    b. What functions do you need to write?
2. Code the TensorFlow computational graph:
    a. Start by calculating or defining any constants. Use the following values of physically relevant quantities:
        i. Number of pixels in the x-direction = 2^7
        ii. Number of pixels in the y-direction = 2^7
        iii. Wavelength of light = 600 nm
        iv. Numerical aperture = 1.25
        v. Spacing in the x- and y-directions = 50 nm
        vi. Number of pixels in the z-direction = 20
        vii. Spacing in the z-direction = 0.1 μm
        viii. Maximum z value = 1 μm
        ix. Minimum z value = -1 μm
    b. Write your tf.Graph()
    c. Write your tf.Session()
    d. Write any auxiliary functions.
3. Include the following features/make the following decisions in your code:
    a. Decide how to initialize the phase only spatial light modulator values, as well as how to initialize all the other trainable variables in your graph.
    b. When randomly selecting x, y points, only choose points in the center 2^6 by 2^6 pixels to minimize periodic edge effects
    c. Feed in Poisson noise to the intensity image as described out in page 7 of this paper: https://arxiv.org/pdf/1807.04813.pdf. You may need to add a regularization term to the square root: explain why.
    d. Add a feature to save and restore your model, see the TensorFlow documentation here: https://www.tensorflow.org/guide/saved_model
    e. At the end of training, create a 2D plot (use matplotlib.pyplot.imshow) of the phase of the spatial light modulator.
    f. Add feature to plot the final 3D Point Spread Function at the end of training:
        i. Make a 2D plot of the image of a point for every z distance considered.
4. Train your neural network, considering the following options. You do not have to test every combination of options. I suggest starting with one option, and then using that as a jumping off point to adjust your option values. Please justify which combination of options you pick. Plot both the log of the error as a function of iteration, the final phase on your spatial light modulator, and the final 3D Point Spread Function.
    a. Experiment with initial values
    b. Number of iterations: 1e2, 1e3, 1e4, 1e5 (You may want to consider using Amazon Web Services or XSEDE for longer training runs. We will go over these options in class.)

    c. Different learning rates: 1e-1, 1e-2, and 1e-3
    d. Number of convolutional layers: 2, 3, 4
    e. Kernel length: 2^4, 2^5, 2^6
    f. Different Poisson noise multipliers: 1e5, 1e6, 1e7
    g. Bonus: Experiment with different neural network architectures including residual layers, batch normalization, and maxout neurons. Search the internet and Stack Overflow to learn more about these features. What worked and what didn't?
5. Consider your results. How successful was your method for single point localization (quantify any claims you make)? What options worked best? Why do you think they worked? Why did you pick the options that you picked (consider the tradeoffs)?
6. How does this method of 3D Point Spread Function design compare to the method of Fisher information maximization we discussed in class? Compare the pros and cons of the two methods. What are some other possible methods to design the 3D Point Spread Function (you may cite references from the literature here)?
7. List ways that you could improve and build upon the work you've done in this lab. What did we not consider that could be important? As a bonus, try to implement some of your suggestions.