Perception, Control and Cognition Lab

Objective:

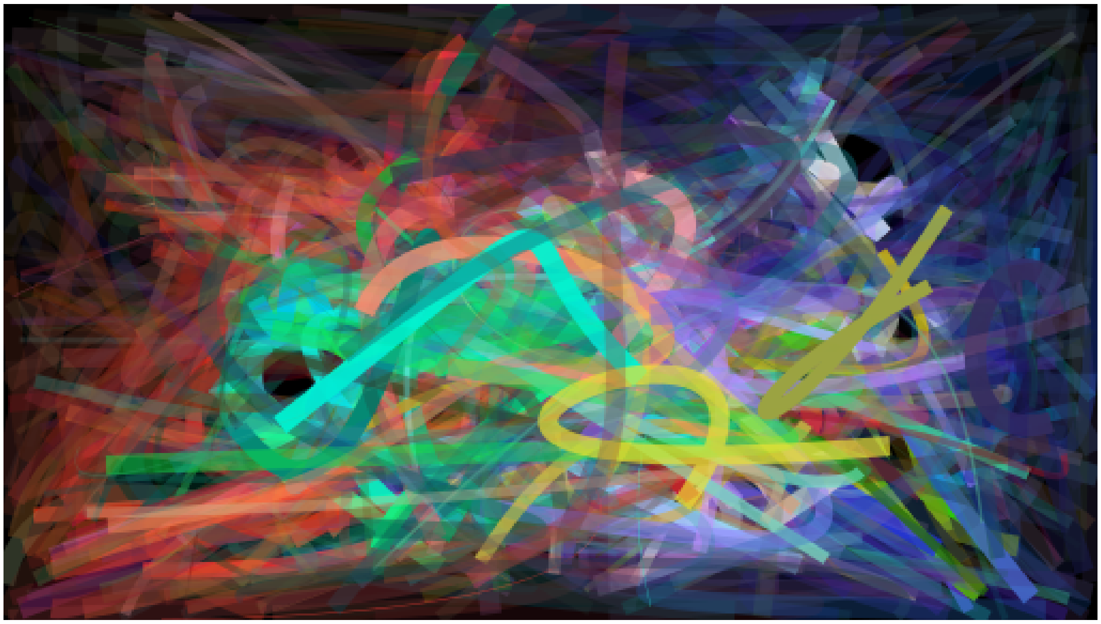
Get started with anaconda, python, ipython notebooks, and pandas. Begin producing simple visualizations of data and images.

Deliverable:

For this lab, you will submit an ipython notebook. This notebook will have two parts:

Part 1: Your notebook should generate a random image. We will run this notebook 5 times; it should generate 5 different, moderately complex images. Each image should be 512 x 288. Have fun with it!

The resulting image could, for example, look like this:



Part 2: You must play with the Tensorflow playground neural network, and figure out how to create a classifier that successfully classifies the “spiral” dataset.

Tensorflow playground [http://playground.tensorflow.org/]

Grading standards:

Your notebook will be graded on the following:

- 20% Successfully turned in a notebook with working code
- 35% Random image with 50 random elements
- 35% Image indicating tensorflow success
- 10% Tidy and legible figures, including labeled axes where appropriate

Description:

Throughout this class, we will be using a combination of ipython notebooks, Tensorflow and the anaconda python distribution. For this lab, you must install anaconda, and write a simple python program (using ipython notebooks).

As described above, the notebook should do two things: 1) generate simple random images, and 2) display an image that you generate using the Tensorflow playground.

For part 1, you can generate any sort of random image that you want – consider random lines, random curves, random text, etc. Each time the program is run, it should generate a different random image. Your image should have at least 50 random elements (they can all be the same type, such as random lines, and can be created in a loop). We won't count the number of elements; this is just to encourage you to create random images with moderate complexity.

For part 2, you should visit the Tensorflow playground (see link above), and play with different settings. Most of it will be unfamiliar, but don't worry – you can't break it!

Once you have a working classifier, take a screenshot. Then use your ipython notebook to display that image in-line.

Installing anaconda:

http://docs.continuum.io/anaconda/install [http://docs.continuum.io/anaconda/install]

To generate images, check out PIL.

To generate random numbers, check out the numpy.random [http://docs.scipy.org/doc/numpy-1.10.0/reference/routines.random.html] module.

To create a new notebook, run:

```
jupyter-notebook
```

This should start an ipython kernel in the background, set up a webserver, and point your browser to the webserver. In the upper-right corner, you will see a “new” menu; under that menu you should see “Notebook” and “Python 2”. This will create a new notebook.

Note: When you turn in your notebook, you should turn in the `.ipynb` file. Do not take a screen shot, or turn in an HTML page.

Here's some starter code to help you generate an image. The `nbimage` function will display the image inline in the notebook:

```
import cairo
import numpy

# A simple function to display an image in an ipython notebook
def nbimage( data ):
    from IPython.display import display, Image
    from PIL.Image import fromarray
    from StringIO import StringIO

    s = StringIO()
    fromarray( data ).save( s, 'png' )
    display( Image( s.getvalue() ) )
```

```
WIDTH = 512
HEIGHT = 288

# this is a numpy buffer to hold the image data
data = numpy.zeros( (HEIGHT,WIDTH,4), dtype=numpy.uint8 )

# this creates a cairo context based on the numpy buffer
ims = cairo.ImageSurface.create_for_data( data, cairo.FORMAT_ARGB32, WIDTH, HEIGHT )
cr = cairo.Context( ims )

# draw a blue line
cr.set_source_rgba( 1.0, 0.0, 0.0, 1.0 )
cr.set_line_width( 2.0 )
cr.move_to( 0.0, 0.0 )
cr.line_to( 100.0, 100.0 )
cr.stroke()

# display the image
nbimage( data )
```

Hints:

The following python functions might be helpful:

```
import matplotlib.pyplot as plt
plt.plot_date

plt.legend
plt.xlabel
plt.ylabel

plt.tight_layout
```

cs501r_f2016/lab1.txt · Last modified: 2016/08/29 10:32 by admin