Detecting Asteroids with Neural Networks using PyBrain

Dustin Ingram
dustin@drexel.edu
di @ GitHub

Philly Python Users Group

May 21, 2013

The goal

Build and train a neural network to correctly identify asteroids in astrophotography data, using **PyBrain**, a modular machine learning library for Python.

Disclaimer

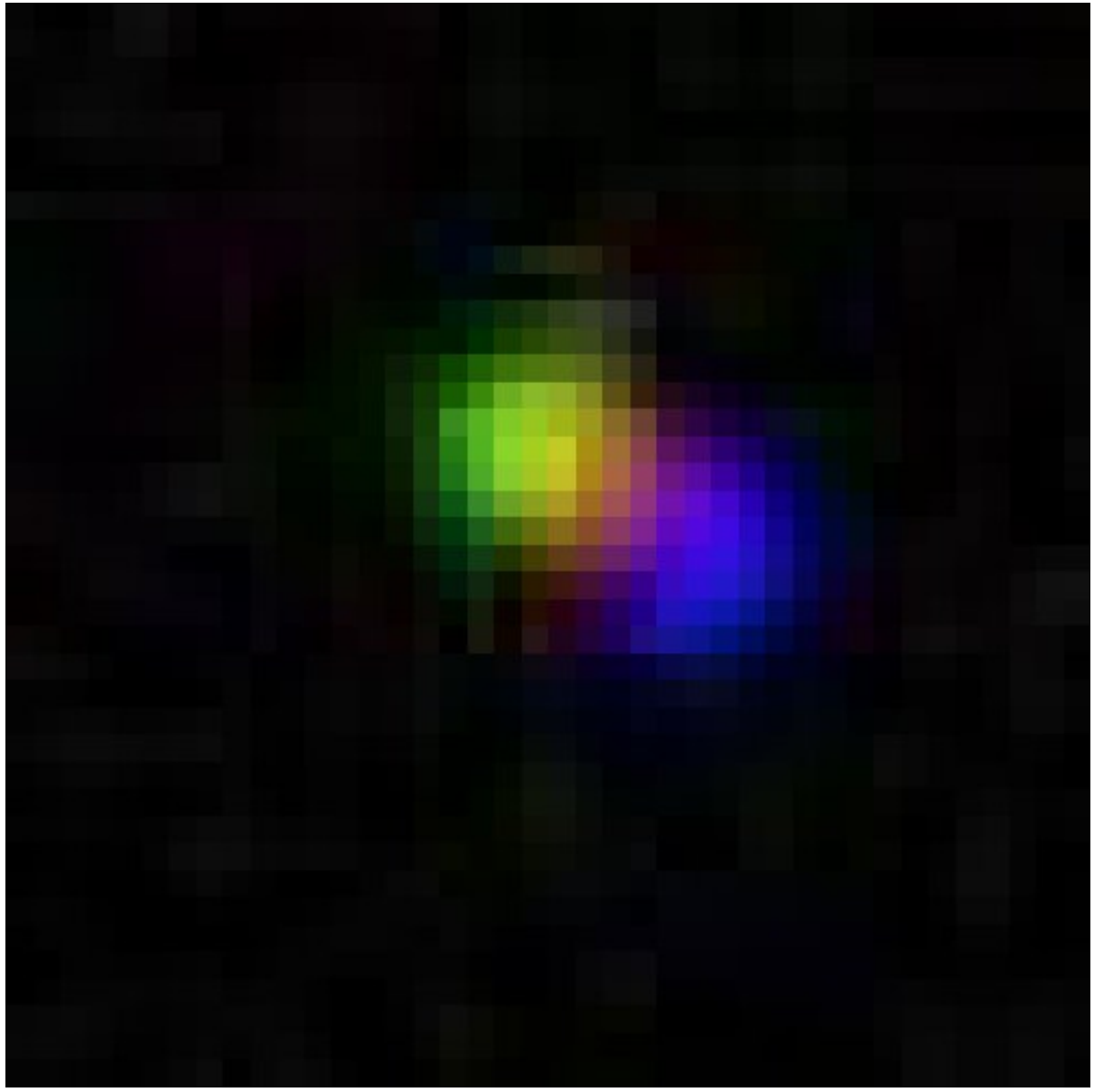
- I am not an expert;
- This is not (quite) my field;
- Some things might be wrong!

The data

The Sloan Digital Sky Survey:

- "One of the most ambitious and influential surveys in the history of astronomy."
- Approx 35% of sky;
- Largest uniform survey of the sky yet accomplished;
- Data is freely available online;
- Each image is 922x680 pixels.

An example asteroid



Why use a Neural Network?

This type of classification is well suited for a neural network:

- We have a clear set of training data;
- There is a small amount of input features which can accurately define an item:
  - Ratio valid hues to non-valid hues
  - Best possible cluster collinearity
  - Best possible average cluster distance
- Each of the input features can be resolved to a $0 \rightarrow 1$ metric;
- The output is either affirmative (1) or negative (0);
- Neural network activation will be fast!

Getting started

Getting the initial training data:

- Small tool to extract potential candidates from full-scale images;
- Extremely naïve, approx 100:5 false positives to actual positives;
- Very low false negatives (approx 1:1000);
- Incredibly slow (complex scan of 100Ks of potentials);
- Manual classification, somewhat slow;
- Yields approx 250 valid items, 500 invalid items;
- Form is a set of 20x20px images.

Making the data set

```
2  from pybrain.datasets import SupervisedDataSet
```

```
15 def make_dataset(source):
16     data = SupervisedDataSet(3, 1)
17
18     print("Adding valid training data")
19     for i in glob(source + "valid/*.jpg"):
20         data.addSample(functions.values(i), [1])
21
22     print("Adding invalid training data")
23     for i in glob(source + "invalid/*.jpg"):
24         data.addSample(functions.values(i), [0])
25
26     return data
```

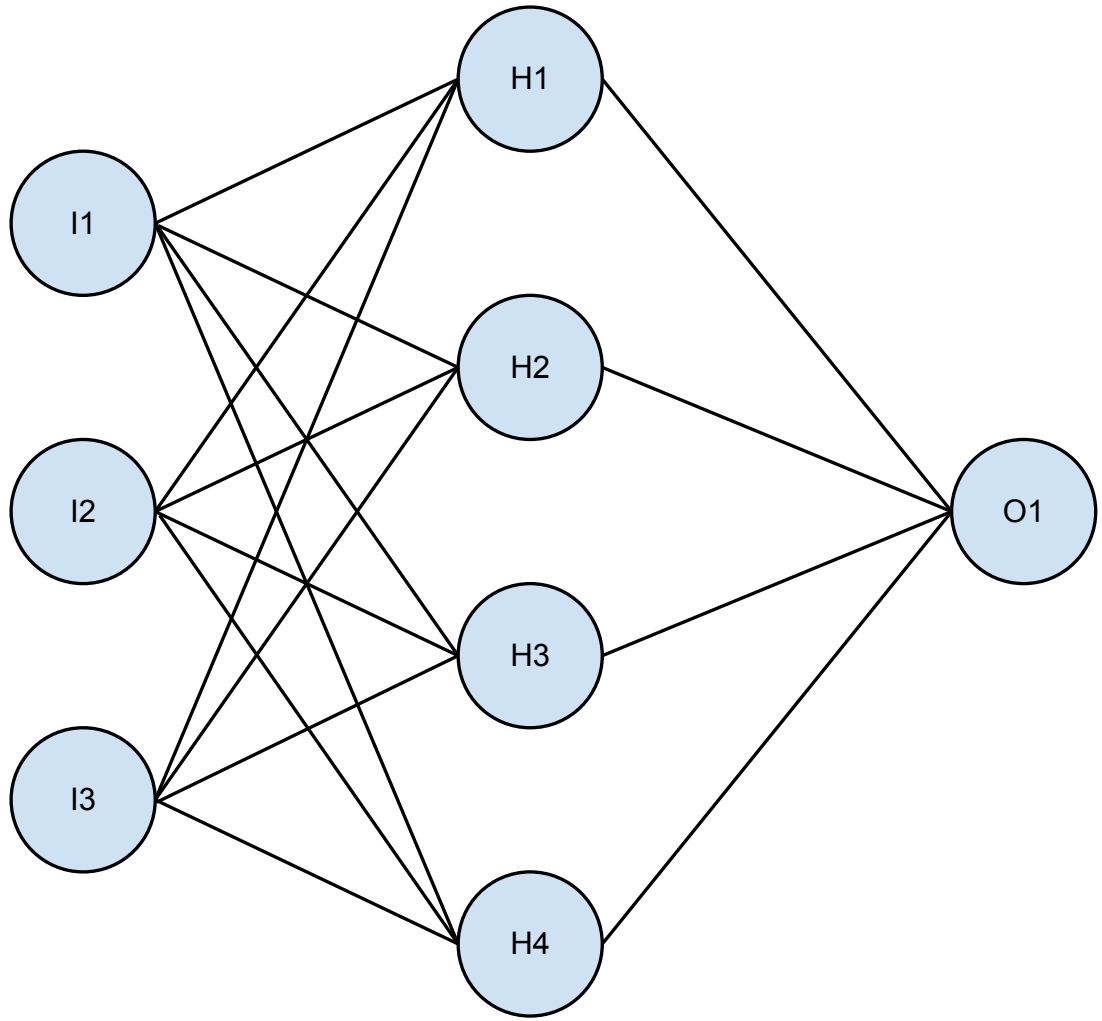
Building and training the network

```
3  from pybrain.tools.shortcuts import buildNetwork
4  from pybrain.supervised import BackpropTrainer
```

```
29 def train_network(d, iterations):
30     print("Training")
31     n = buildNetwork(d.indim, 4, d.outdim, bias=True)
32     t = BackpropTrainer(
33         n,
34         d,
35         learningrate=0.01,
36         momentum=0.99,
37         verbose=False)
38     for epoch in range(iterations):
39         t.train()
40     return n
```

Building the neural network

The resulting neural network:



Training the network

- Approx 250 valid items;
- Approx 500 invalid items;
- Trained for 5,000 iterations;
- Took approx. 3 hours;
- Probably could have gotten by with less iterations.

Testing the network

```
9  import shutil
10 import os
```

```
43 def test(path, source, net, cutoff):
44     val = net.activate(functions.values(path))
45     base = os.path.basename(path)
46     if val > cutoff:
47         print path, val, "(Valid)"
48         shutil.copy(path, source + 'valid/' + base)
49     else:
50         print path, val, "(Invalid)"
51         shutil.copy(path, source + 'invalid/' + base)
```

Putting it all together

```
59  data = make_dataset('./training_data')
60  net = train_network(data, iterations=5000)
```

```
66  for path in glob('./' + sys.argv[1] + "*.jpg"):
67      test(path, './' + sys.argv[1], net, cutoff=0.9)
```

Storing your neural network

```
8  import pickle
```

```
53 if __name__ == "__main__":
54     try:
55         f = open('_learned', 'r')
56         net = pickle.load(f)
57         f.close()
58     except:
59         data = make_dataset('./training_data')
60         net = train_network(data, iterations=5000)
61         f = open('_learned', 'w')
62         pickle.dump(net, f)
63         f.close()
```

PUBLIC

di / astro

Code | Network | Pull Requests 0

branch: **master** | Files | Commits | Branches 1

Added pickled brain :)

master

di authored 2 months ago

Showing **1 changed file** with **927 additions** and **0 deletions**.

927 _learned

```
...  ...  @@ -0,0 +1,927 @@
     1    +ccopy_reg
```

Thanks!

- Contact me:
  `dustin@drexel.edu`
- Source for this talk:
  `https://github.com/di/astro`
- The Sloan Digital Sky Survey:
  `http://www.sdss.org/`
- PyBrain:
  `http://pybrain.org/`