Replication of Tyler et al. (2000)

Dr. Olivia Guest

November 5, 2015

1   Save Outputs

Download the slightly different version of `network.py`. Go to where it says

```
self.save_outputs_button = gtk.Button("Save Outputs")
self.save_outputs_button.connect("clicked", self.save_outputs, None)
self.hbox2.pack_end(self.save_outputs_button, expand, fill, padding)
self.save_outputs_button.show()
```

What this line is saying is equivalent to the lines above that are identical for saving the weights, but with a different function being connected to the button with label "Save Outputs". In other words, it is saying that when that button is pressed both these two functions will be called and both the weights, as we saw last time, and now the outputs will be saved to a file. Go to the function **save_weights** and use that to help you write a *new* function called **save_outputs**. This function should save the outputs of the network *for every pattern* to a file.

Hint: Perhaps collect them all up in a single variable and dump that into a pickle file.

Use just the `pickle.dump` function (no dialogue or button!) to dump the errors you are calculating after training:

```
pickle.dump(variable_you_saved_errors, 'name_of_file.pkl', "wb" ) )
```

NB: **variable_you_saved_errors** is whatever you named it and **'name_of_file.pkl'** is a string containing the name of the file.

2   Open and Graph Variables

To open the weights, in **open_weights**, we use the following to load them back into our network:

```
self.network.weights_i2h, self.network.weights_h2o = pickle.load(open( dialog.get_f
```

Make sure you understand what is going on here. Now start a new file completely called e.g., `graph.py` and use that to load in the network errors and then decide what you want to do with them. Remember you have to tell python to import the same libraries as we used previously (see the `import` statements at the top).

If you want to graph them see the tutorial here:
`http://matplotlib.org/users/pyplot_tutorial.html`

This should be easy for the errors and harder for the outputs, since the outputs will need to be somehow evaluated/compressed down to something more easy to graphically depict. Graphically depicting the errors after every 10 epochs of trainign for example might be useful. Thanks about what you need to do to make that happen.