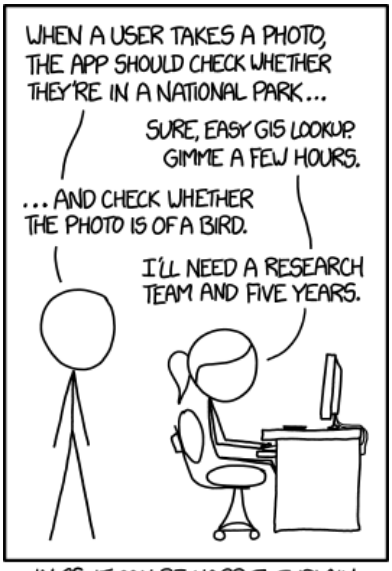
CS30 - Neural Network Lab



http://www.xkcd.com/1425/

In this lab, we will experiment with a few simple neural networks. The idea is to obtain preliminary answers to questions like:

- "How quickly can a network be trained?"
- "How reliable (or repeatable) is the training?"
- "How accurate is the resulting network?"

Although there is nothing to submit for this lab, I highly recommend that you document what you do for this lab for future reference. That includes both any code that your write as well as a couple of short sentences for each question. At the end of the lab we will discuss as a group what you find out!

Use the neural network software that is described in the handout *Neural Networks in Python* linked on the course web page as reading for the lab today (which you should have already read!).

1. Construct a network with two hidden nodes and train it on the XOR data below. Notice that we have complete information about the function we are trying to approximate—an unusual

situation for a neural network. How quickly do the weights converge? How well does the resulting network perform? Try training several different networks to see the variation in the convergence rate. You may also want to reduce `printInterval` to get finer information about the changes in the error.

inputs		output
0	0	0
0	1	1
1	0	1
1	1	0

2. Repeat part 1 with a network with eight hidden nodes. Does the convergence go faster? Is the resulting network a better approximation to the XOR function?

   Notice that "faster" can mean "fewer iterations" or "less clock time." A network with more nodes will have more weights to adjust and will take more clock time for each training cycle—but it *may* require fewer cycles and the total time *may* be shorter.

3. See what happens when you repeat part 1 using a network with just one hidden node. The XOR function cannot be computed with a single node. Why is a network with one hidden node equivalent to a single node?

4. Table 1 contains a sampling of voter opinions. The idea is to deduce a voters party affiliations from their views on the importance of various topics. Six voters were asked to rate the importance of five issues on a scale from 0.0 to 1.0 and to identify themselves as Democrat (0.0) or Republican (1.0).

   Write a neural network and train it on the data in Table 1. Then try it on the samples from Table 2 or other cases of your own creation. Can you explain the conclusions drawn by the network?

budget	defense	crime	environment	social security	party
0.9	0.6	0.8	0.3	0.1	1.0
0.8	0.8	0.4	0.6	0.4	1.0
0.7	0.2	0.4	0.6	0.3	1.0
0.5	0.5	0.8	0.4	0.8	0.0
0.3	0.1	0.6	0.8	0.8	0.0
0.6	0.3	0.4	0.3	0.6	0.0

Table 1: A sampling of voter opinions. This example is taken from notes by Dave Reed of Creighton University.

budget	defense	crime	environment	social security	party
1.0	1.0	1.0	0.1	0.1	?
0.5	0.2	0.2	0.7	0.7	?
0.8	0.3	0.3	0.3	0.8	?
0.8	0.3	0.3	0.8	0.3	?
0.9	0.8	0.8	0.3	0.6	?

Table 2: Some test cases for the voter network.