Notes on Python and NumPy

- In NumPy, obj.sum(axis = 0) sums the columns while obj.sum(axis = 1) sums the rows.

- In NumPy, obj.reshape(1,4) changes the shape of the matrix by broadcasting the values.

- Reshape is cheap in calculations so put it everywhere you're not sure about the calculations.

- Broadcasting works when you do a matrix operation with matrices that doesn't match for the operation, in this case NumPy automatically makes the shapes ready for the operation by broadcasting the values.

- In general principle of broadcasting. If you have an (m,n) matrix and you add(+) or subtract(-) or multiply(*) or divide(/) with a (1,n) matrix, then this will copy it m times into an (m,n) matrix. The same with if you use those operations with a (m , 1) matrix, then this will copy it n times into (m, n) matrix. And then apply the addition, subtraction, and multiplication of division element wise.

- Some tricks to eliminate all the strange bugs in the code:
  - If you didn't specify the shape of a vector, it will take a shape of (m,) and the transpose operation won't work. You have to reshape it to (m, 1)
  - Try to not use the rank one matrix in ANN
  - Don't hesitate to use assert(a.shape == (5,1)) to check if your matrix shape is the required one.
  - If you've found a rank one matrix try to run reshape on it.
- Jupyter / IPython notebooks are so useful library in python that makes it easy to integrate code and document at the same time. It runs in the browser and doesn't need an IDE to run.
  - To open Jupyter Notebook, open the command line and call: jupyter-notebook It should be installed to work.
- To Compute the derivative of Sigmoid:

```
s = sigmoid(x)
ds = s * (1 - s)       # derivative  using calculus
```

- To make an image of (width,height,depth) be a vector, use this:
  v = image.reshape(image.shape[0]*image.shape[1]*image.shape[2],1) #reshapes the image.

- Gradient descent converges faster after normalization of the input matrices.