**Project Outline – changed stuff**

**Project 1: Comparing theoretical input weights to weights generated from Hebbian learning rules**

1. Pick filters, these have been provided, these filters are used as integration kernels.
2. Select a collection of natural images. The neural network will be constructed to optimally encode these natural images.
3. Without context integration the neurons in the network simply respond to their receptive field. With context integration neurons from the network are fed both their classical receptive field as well as inputs from the surrounding neurons in a nonlinear manner.
4. Integrate 18 filters against all of the natural images. These are the fc’s. The size of the filter = 15x15 pixels. The spacing between filter centers is 7 pixels, ex/ 1, 8, 15…

This is true for VISp.

More Detailed List for Project 1

**Project 2: Compute the Exponential decay constant for the theoretical input weights, take this constant and take ratio of exponential constant to cortex magnification in each of the layers. Compare computed ratios, they should be the same.**

1. Calculate fc from the convolution of the 18 filters with the whole image data set (500 natural images)
2. Use eq #26 to calculate the synapse weights “W”
3. Find the average synaptic weight for a given receptive field distance between any two filters. Plot W(\delta) as a function of the distance is \delta, which should hopefully follow an exponential decay. Exp[-\delta/(\alpha)] Fit the data to a function (find decay constant **\alpha**)

****

1. For each visual layer, compute the ratio between \alpha and the cortex magnification.

Cortex magnification -> Cm = (angle picked)/(distance between neurons in mm)

1. Are these ratios the same? They should be!

1. The size of the filter = 15x15 pixels. The spacing between filter centers is 7 pixels, ex/ 1, 8, 15… This is true for VISp. We will need to rescale the filter size for VLM (VisL), VAL, VRL, VAM, VPM. In order to know how much to rescale each filter for these different brain areas, we need from the Python notebook the variable **onofft** (receptive field size) from In[77]. We need to double check with someone that **onofft** is area instead of length (we aren’t sure what the units are)

More Details

1. Images – we obtained the 500 natural images from the Berkley Segmentation Data Set.

We turn the data to greyscale by averaging the three RGB values and then/255.

1. Filters – we used the 18 filters generated from 8 on/off Gaussian distributions, 8 off/on Gaussian distributions, 1 on Gaussian distribution, and 1 off Gaussian distribution.