Chapter eight from the *Theoretical Neuroscience* book starts the discussion on learning and activity-dependent synaptic plasticity behind different types learning including supervised, unsupervised and reinforcement learning. The chapter presents an abstract mathematical representation of Hebb’s rule where synapse between two neurons should be strengthened if one neuron frequently contributes to the firing of the other neuron. The synaptic plasticity is described in a linear firing rate model to avoid complexity related to any nonlinear dependence. I am mostly lost in the model application and derivation, but the big picture of simulating a neuron network with feedback and recurrent mechanism and adjusting a weight matrix according to the data is very similar to a lot of ideas used in machine learning and neuron network training.

Chapter eight from the MATLAB book introduces the notion of psychophysics where a quantified mental phenomena is elicited when physical stimuli is presented and the stimuli are generally easily described by certain parameter such as luminance, contrast or spatial frequency. The origin of psychophysics theory was from Gustav Theodor Fechner and his “identity theory” that all things have souls sounds almost like superstition to me. It is interesting to see how these philosophical and seemingly non-scientific ideas could lead experimental researches into mental domains such as the perception of heavy mass. The MATLAB tool is basically a Photoshop in MATLAB and would allow us to have precise control over the visual stimuli presented in our experiments. These functions would not help directly with the summer projects, but would be fun tools to use for photography.

I learned some new practical mathematical ideas this week. One big challenge we faced last week was to convert the regression in log linear time but leave the remaining part in regular exponential decay meanwhile making sure everything is bounded between zero and one. Dr. Reber showed us a few ways around the problem. We could treat the value between zero and one like a probability function and use logistic regression from there. Or we could always keep in mind the inverse function. That is, we could create an intermediate variable holding the value of the raw data in log or any function that fits our needs, do a regression analysis or some function to update the intermediate value, and lastly use the inverse function to turn the intermediate variable back to its original format and unit. Using this approach, we could transform the data into the right space and time within a specific window and return back to the original condition to continue the regular analysis afterwards.

I also coded a golden section search algorithm into the model. It is an iterative method of finding minimum of a function, a topic generally covered in a numerical method class. The model, even with only one parameter, became fairly complicated after the introduction of the non-linear component. The tmr boost value is solved from a univariate higher power equation with the highest power equals to the number of cues played during sleep. The solutions from the math solver in python contain complex roots that would not make sense in the context. So we turned to a numerical solution by using the iterative golden section search method. It is included in python’s optimization package, but Dr. Reber thought it would be a good practice to write your own version and I enjoyed it. Besides, it would give us more control over the script and make troubleshooting easier in the future.