Duncan Miller

Brian Morales

Computational Neuroscience

Training Excitatory-Inhibitory RNN

Introduction:

We have chosen H.Francis Song's *Training Excitatory-Inhibitory Recurrent Neural Networks for Cognitive Tasks.* This paper discusses neural networks, specifically recurrent neural networks (RNN's) in their relation to excitatory and inhibitory neurons. Notoriously artificial neural networks have disregarded the process in which the successful outcome was achieved and only focused on positive results. This paper dives into how we can improve RNN models by forcing realistic parameters on our neural networks, such as, only allowing positive firing as negative firing is not observed in nature. Song uses stochastic gradient descent (SGD) training through a Python library called Theano to train these RNN's. He applies these methods to well-known experimental paradigms such as perceptual decision making and multisensory integration. Our goal for this paper is to expand on the results of these training models by either adding inputs, adjusting initial conditions, and/or adjusting the system to different parameters.

The brain carries out computations by populations of interconnected neurons. Many studies have been done to understand the brain function including how brain mechanisms interpret visual, motor, and cognitive tasks, and learn the neural mechanisms underlying the brain. A single-neuron response can reveal a ton of information about the neural mechanisms in sensory, motor, and cognitive processes, however, the brain does not function on a single-neuron. Neural pathways often involve the coordination of many neurons whose complex individual dynamics are not easily explained. The importance of studying neurons at the population-level has become better understood; an increasing number of studies have been published that use large data sets of simultaneously or sequentially recorded neurons to construct neural circuit mechanisms. Many promising approaches have been used to identify the dynamical and computational mechanisms planted in large neural populations.

In this paper we will be focusing on recurrent neural networks (RNN) and training multisensory excitatory and inhibitory neurons. We will first introduce the mathematical background behind the RNN model. Afterwards, we'll introduce Songs method and network structure; comment on his findings and results. Then we will adjust the network structure by editing the number of inputs from 5 to 7, changing the ratio of excitatory and inhibitory hidden units from 0.8 and 0.2 to 0.5 and 0.5, and 0.2 and 0.8, respectively. Lastly,we will alternate the ratio between units receiving visual input , auditory input, and no inputs. From our results we will analyze and comment on the outcome and conclude what network structure we thought worked best.

Mathematics behind RNN:

RNN have three main structural components, first we have N_{in} inputs u(t) which act as the source for our network, N_{out} outputs z(t) which act as the result of our network and a series of weighted matrices W^{in}, W^{rec},W^{out} in the form N x N_{in}, N x N_{rec} and N x N_{out} respectively. Here, W^{in} represents the weights connected to our inputs, W^{rec} represents the recursive weights, and W^{out} represents our weights connected to our outputs. These are matrices of connection weights which, after training, guide the inputs of our network to the correct output. We characterize the network's behavior with N firing rates r(t) with corresponding currents x(t). ζrefers to Gaussian white noise with zero mean and unit variance, which gives our data a more realistic representation of the brain as noise is intrinsic in all systems. Finally we have τ which represents the time constant of the network, and σ_{rec}^{2} represents the unbiased variance. Putting these together gives our equations describing RNN's:

$$\tau\frac{dx_i}{dt} = -x_i + \sum_{j=1}^{N} W_{ik}^{in} u_k + \sqrt{2\tau\sigma_{rec}^{2}}\zeta \qquad (1)$$

$$r_i = [x_i]_+ \qquad (2)$$

$$z_l = \sum_{i=1}^{N} W_{li}^{out} r_i \quad (3)$$

It is important to recognize that equation (2) restricts our system to only positive weights. Traditionally, RNN's are allowed to contain negative firing rates and currents, although this does not represent realistic biological systems. Therefore this restriction reflects a more accurate representation of a real RNN.

Training RNNs with gradient descent:

When training an RNN, we presume that at each time step there is a correct output target z_t^{target} that depends on the present and previous history of inputs $u_{t'}$ for $t' \leq t$.The objective is to find network parameters that reduce our error between our target output and actual output. We would like to minimize our error function without over estimating our learning curve. Song implements an objective function $\epsilon(\theta)$that not only minimizes the difference error but other terms such as L_1regularization terms that influence the types of solutions found by the training algorithm. Additionally, songs implements a loss function $L(\theta)$that measures the difference between the target and actual outputs.

$$\epsilon = \frac{1}{N_{trails}} \sum_{n=1}^{N_{trails}} L_n \quad (4)$$

$$L_n = \frac{1}{N_{out} N_{time}} \sum_{l=1}^{N_{out}} \sum_{t=1}^{N_{time}} M_{tl}^{error} [(z_t)_l - (z_t^{target})_l]^2 \quad (5)$$

$L_n(\theta) \equiv L_n$is the squared sum of differences averaged over N_{trials}, N_{out}, and N_{in}time points. For each trial $n$in equation (5), $z(t)_l$is from equation (3) where $l$is the lth output at time t. $M_{tl}^{error}$is a matrix

of ones and zeros that decides if the error in output $l$at time $t$should be taken into account. The above equations allow us to train networks based on the final time course for the outputs.

In gradient descent training we updated the network parameter iteratively according to the direction of the steepest descent. In Songs algorithm he applies Stochastic Gradient Descent which is similar to normal gradient descent, however, with suitable smoothness properties such as differentiability. SGD is most helpful in scenarios to minimize computational burden where achieving faster iterations is the tradeoff for lower convergence rate.

$$\theta^{i} = \theta^{i-1} + \delta\theta^{i-1} \quad (4)$$

Equation (4) denotes the network being updated iteratively. $\delta\theta$is taken to be proportional to the negative gradient of the objective function with respect to the network parameters as shown below.

$$\delta\theta^{i-1} = -\eta\nabla\epsilon^{i-1} \quad (5)$$

Where $\eta$is the learning rate and $\nabla\epsilon^{i-1}$is the value of the gradient evaluated on the parameters from iteration $i - 1$.Songs utilizes Theano to automatically compute backpropagation through time so that the gradient descent is efficient. Below we show the default results froms Song's paper and expand upon it with our code modifications.

Results

After we have created our network and run a feed forward pass on it, followed by SGD backward propagated training, we are left with a trained network that responds to two sensory inputs (auditory and visual) at varying inputs/second. Our results are split up into three categories of plotted data: Sections A-C. Section A shows example trials of the training data ranging from visual only, to auditory only, and finally both visual and auditory simultaneously (multisensory) with the zero set at the stimulus onset. The stimuli are presented in rates, ranging from 9 events/second to 16 events/second (ie. 9 beeps/flashes per second)(Note: This training data is identical for all of our variations of results). Section B is the key testing data. When presented with the stimuli, the network chooses if the current event rate is above or

below a threshold rate (set to 12.5 events/second). Our plot on B shows the percentage of times a “high” choice is made. Section C gives an important insight into the nature of RNN and neural networks in general. We plot firing rates vs. the time from stimulus onset for 4 variables, visual “high” stimulus, visual “low” stimulus, auditory “high” stimulus and auditory “low” stimulus. The first figure shows how we see coupled behavior between high and low stimulus (choice selectivity) although the second figure shows how we see coupled behavior between auditory and visual behavior (modality selectivity) all while plotting over the same conditions (firing rate vs. stimulus onset). These are conflicting data! The importance of these results is that while RNN’s can be powerful (see plot B), they require a significant amount of data from previous experiments/studies in order to accurately train and reproduce accurate results.

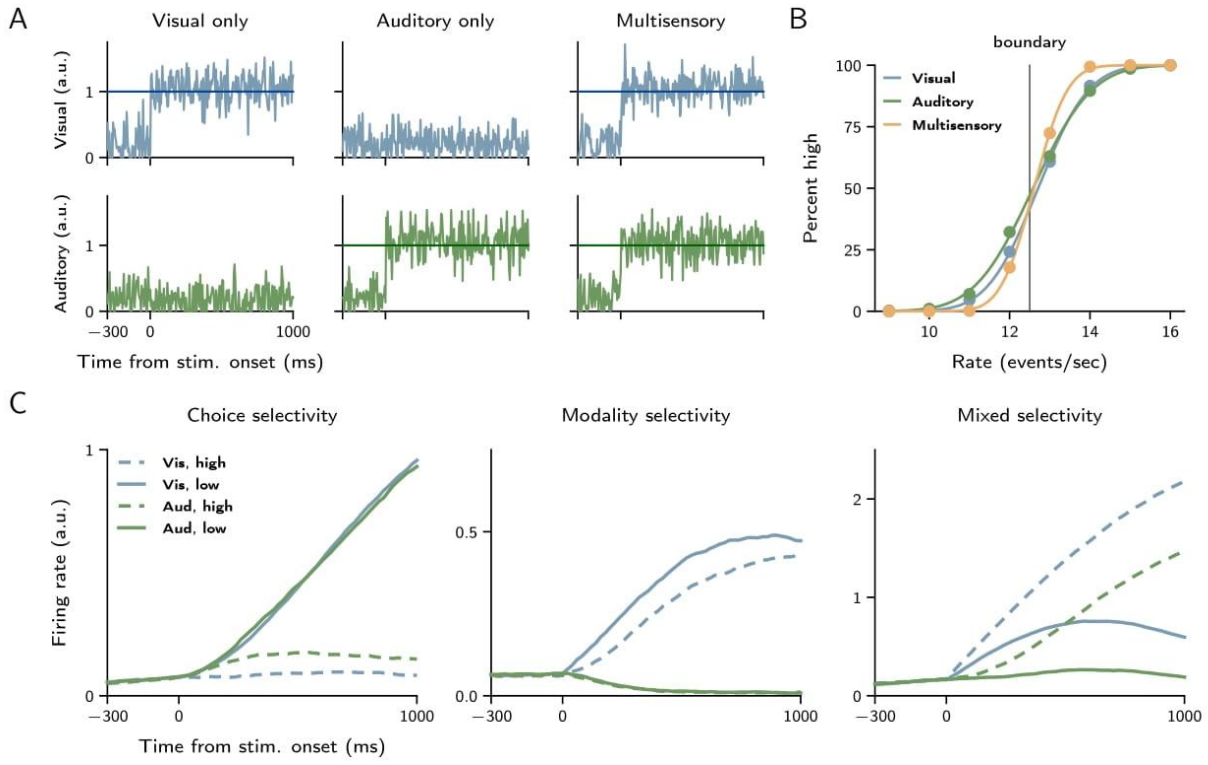

Figure 1 - Default network settings used in the paper

Songs default network structure begins with five input variables, one layer of 150 hidden units, and two outputs for the visual and audio. Default settings for excitatory and inhibit inputs are 80% and 20% therefore 120 inputs are excitatory and 30 are inhibitory. The rationale for 80% excitatory input and 20%

inhibitory inputs is reflective of the 4:1 ratio of biologically excitatory and inhibitory inputs such that this model replicates a ratio of the biological brain. Furthermore, a third of the units receive visual inputs, a third receive audio inputs, and a third receive no inputs. The accuracy of this training is:

Inputs	Accuracy	Variance
Visual	90.33%	0.988
Auditory	88.85%	1.14
Multisensory	94.22%	0.651

Part B graphs the percentage of success when visual, audio, and visual-auditory information are present in different event rates. As stated, the event rates range from 9 events/sec to 16 events/sec and our threshold is 12.5 events/sec. The network was required to determine whether the inputs were below or above the threshold. We see from these data that our multisensory training is performing the best, which is expected as both auditory and visual information are presented at the same rate. Next, we see the visual training is giving incorrect results slightly sooner than expected. Finally, we see that auditory performs the worst in choosing incorrect “lows” and not choosing correct “highs”. Note the analysis on these results are relative, as this is very positive data in regards to the power of RNN.

Part C, as stated before, gives conflicting coupled data. One set couples with choice selectivity while the other couples to modality selectivity. Similarly, the third plot shows (partially) coupled behavior between highs and lows. This chaotic behavior will continue throughout the rest of the report.

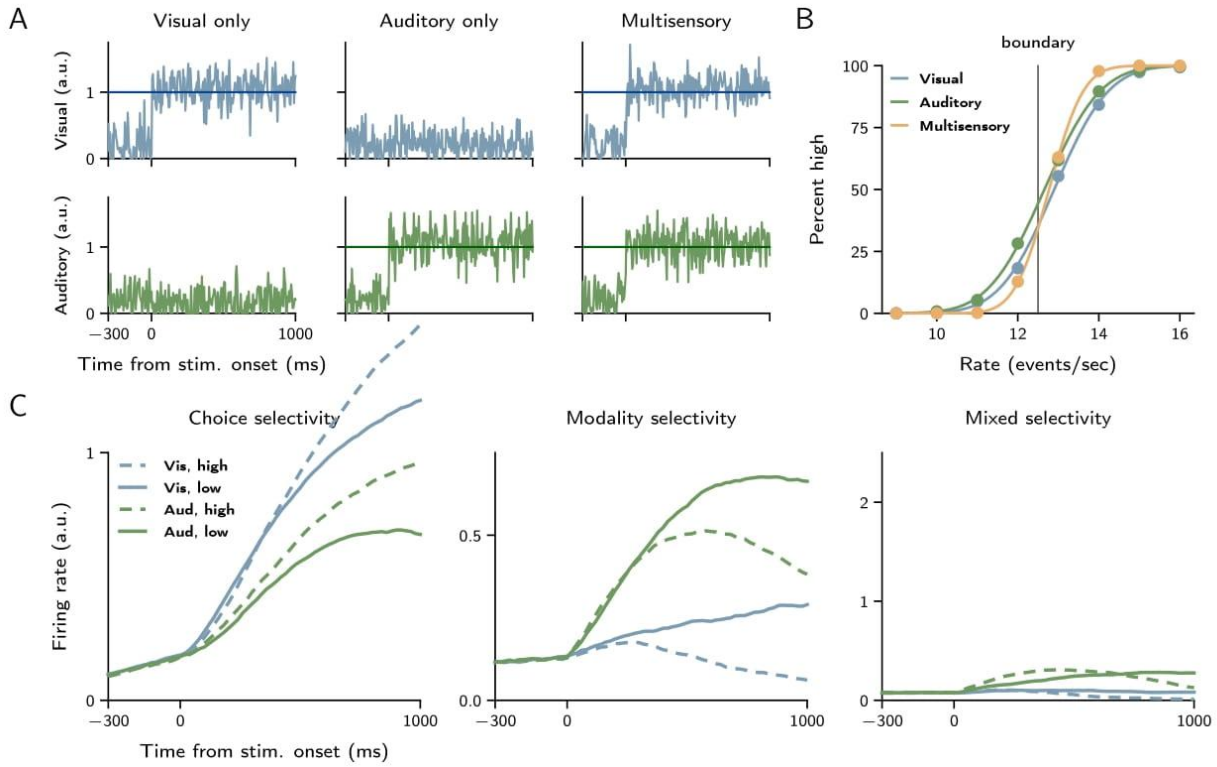

Figure 2 - Increasing the number of inputs to seven.

In this graph we decided to adjust the network structure to have similar default settings to the previous graph, however, instead of five inputs we increased the inputs to seven. The accuracy of this training is:

Inputs	Accuracy	Variance
Visual	90.60%	0.95
Auditory	88.30%	1.20
Multisensory	94.70%	0.60

Part A in Figure 2 is similar to Figure 1, however, Part B is a bit different. As expected, when both visual and auditory inputs are present, the network performs the best and consequently better than Figure 1 Part B. Visual performance is as close to multisensory performance while increasing its accuracy slightly from Figure 1, likewise with auditory performance.

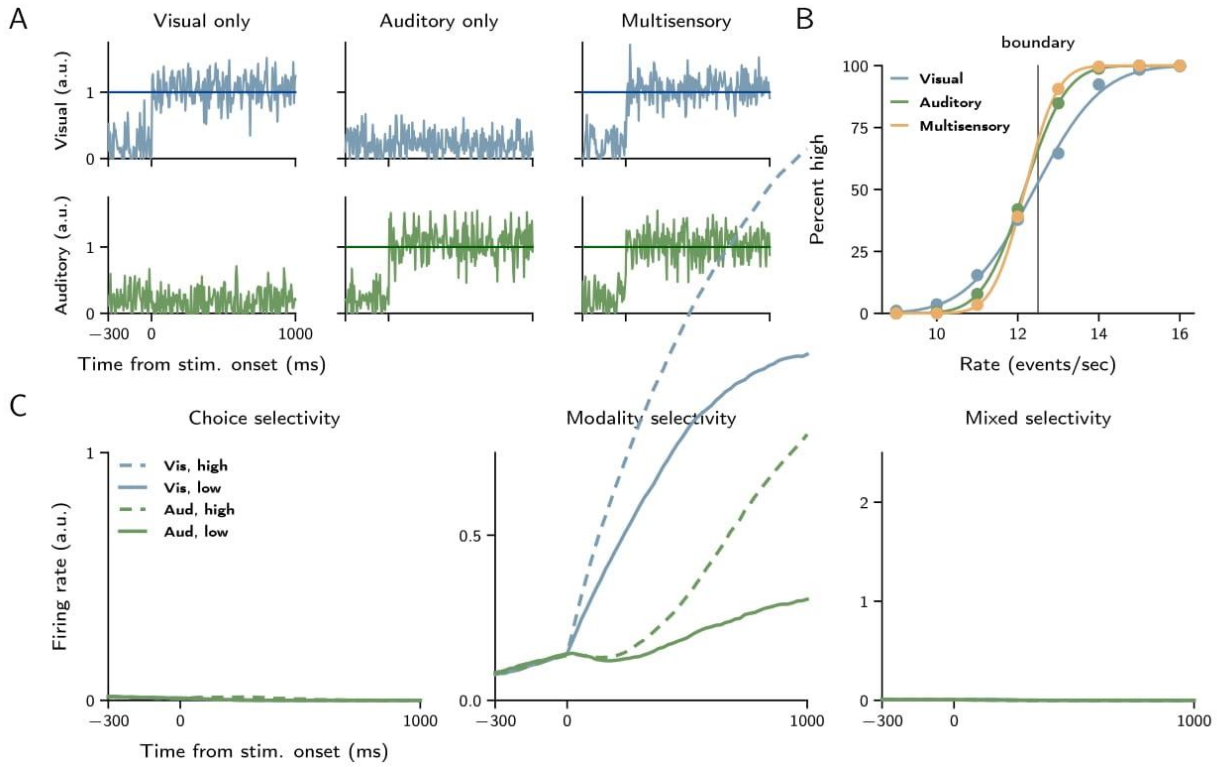

Figure 3 - 60% of the hidden units receive audio input, 30% receive visual inputs, and 10% no inputs

In the above graph we decided to return to five inputs, 150 hidden inputs, and two inputs, however, we modified the amount of units that receive visual inputs, auditory inputs, and no inputs at all. We decided to update the number of units that received audio inputs from a third to 60%, and maintained the number of visual inputs (one third) and decreased the number of units that received no inputs to a tenth. The accuracy of this training is:

Inputs	Accuracy	Variance
Visual	87.17%	1.303
Auditory	91.65%	0.816
Multisensory	93.47%	0.630

Now Figure 2 Part B shows a significant difference with visual and audio accuracy. As mentioned previously, visual-auditory inputs perform the best, but audio accuracy performs better than visual

accuracy. This is the exact conclusion we expected because we increased the number of units that received auditory training and decreased those that received visual training.

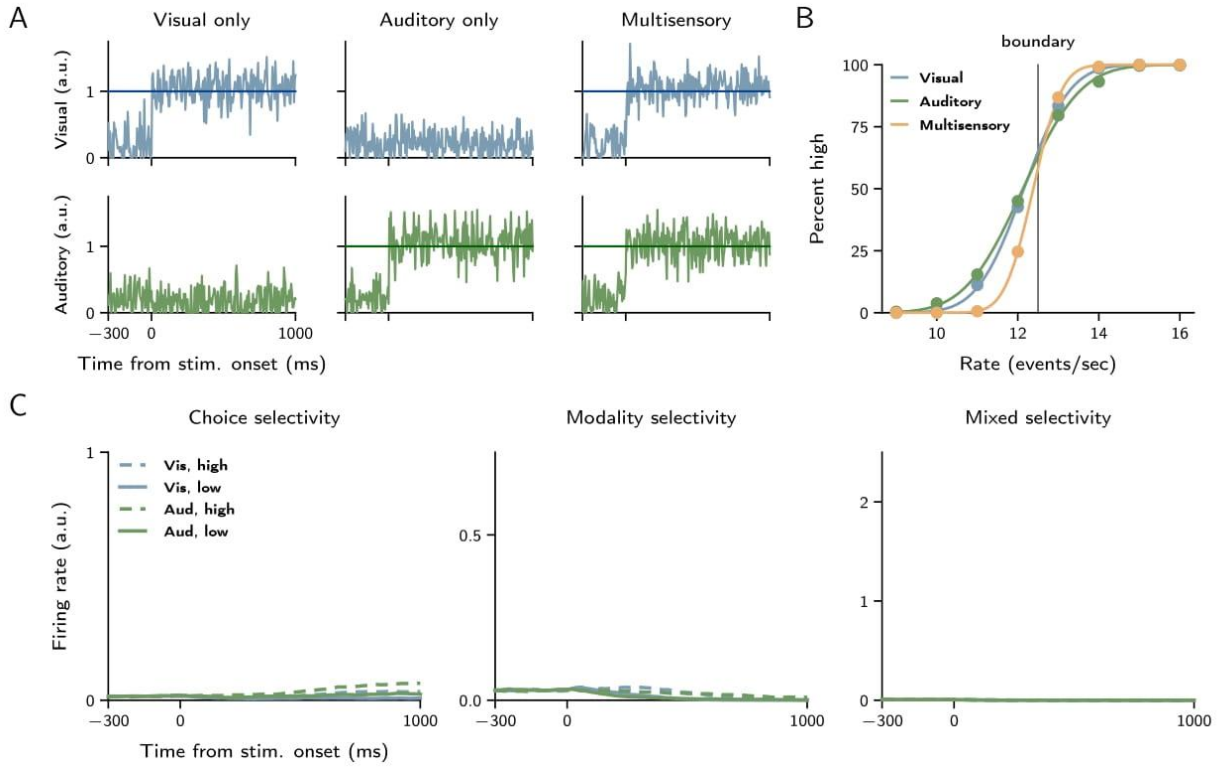

Figure 4 - Hidden units received 60% visual, 30% auditory, and 10% no input

Similarly to Figure 3, we maintained the default settings and again altered the units that are receiving visual inputs, audio inputs, and no inputs. In this run, we switched the number of units that receive visual and audio inputs to 60% of the units receiving visual inputs, a third receiving audio inputs, and a tenth receiving no inputs. The accuracy of this training is:

Inputs	Accuracy	Variance
Visual	90.9%	0.89
Auditory	88.45%	1.12
Multisensory	95.10%	0.55

Similar to Figure 3, Figure 4 Part B shows a significant difference with visual and auditory accuracy. Since we increased the number of units receiving visual training the accuracy of the networks discerning visual events was better than audio results, as expected.

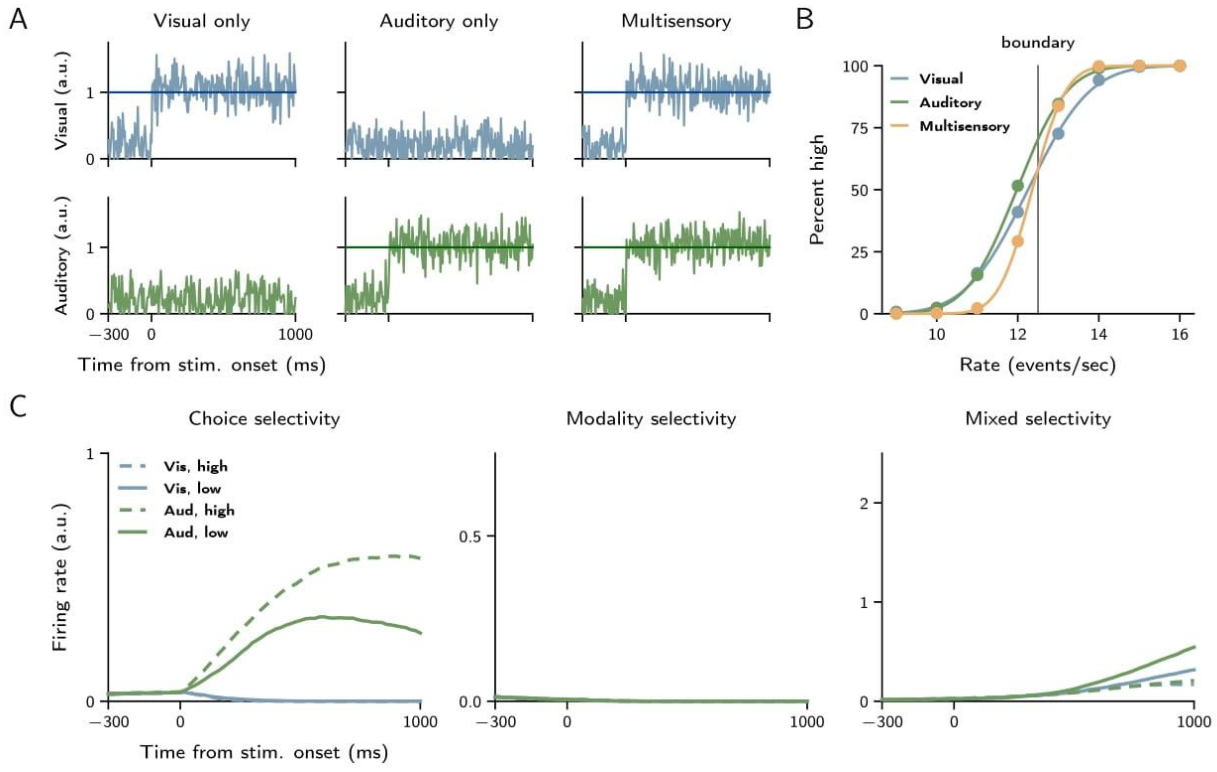

Figure 5 - Half of the hidden inputs receive visual training, half receive auditory training

Lastly, we adopted the same settings as Song's default setting, however, modified the units receiving visual and audio inputs to half and half. We decided to remove units receiving no inputs to

analyze if there would be significant improvement in the network performance. The accuracy rate for this training:

Inputs	Accuracy	Variance
Visual	88.25%	1.20
Auditory	89.20%	0.97
Multisensory	94%	0.65

The results in part B show that visual performance and audio performance increased, however, audio performed better than visual. Overall, the performance of this training ratio was the worst of all scenarios.

Conclusion

Recurrent Neural Networks have the potential to accurately model cognitive processes in the brain. In this expansion of H.Francis Song's paper we explored the relationship between RNN and excitatory/inhibitory neurons. More specifically we analyzed how a RNN, trained using Stochastic Gradient Descent, performed under a multisensory integration task using various ratios of excitatory and inhibitory neurons, as well as editing other hyperparameters in order to achieve the most accurate results.

From all our test results the network structure that performed the best was scenario four where we increased the number of visually trained neurons and decreased the number of auditorily trained neurons. The network structure that performed the worst scenario five when we equalized the number of visual and auditory trained neurons. Evidently, we can conclude that when the audio or visual inputs increase the model will perform better in each corresponding category. Furthermore, we can conclude that multisensory models will always perform best, as we have demonstrated congruent data from both visual and audio inputs.

Each figure has shown the spiking rate of visual presence, audio presence, and visual-audio presence. Part A showing example trails for each sensory, Part B graphs the network's success in discerning visual flashes and audio clicks in a given threshold, and Part C presents insight on how the neuron behaves from choice, modality, and mixed selectivity. Song's research paper and conclusion has demonstrated how close we can emulate an animal's biological brain (specifically a rat) and how we theoretically change the hyperparameters to improve or test the network.