August 9th 2020

Jack Schenkman
Tactile Stimulation Project

Words

- There are N words
- The neural network accepts a one-hot $N \times 1$ vectorized representation of a word as input

$$V = \begin{bmatrix} \\ \\ \\ \end{bmatrix} \overset{1}{} \Big\} N$$

Data Structure

- initially assume there is only one actuator
- To generalize to multiple actuators, simply let each actuator have its own instance of this data structure

Case 1

- Set of K pairs of values

pattern = $[(a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k)]$

a_i: time between $(i-1)$th pulse and the ith pulse
- for a_1 this quantity is the time since the stimulation pattern began

b_i: intensity or magnitule of the ith pulse

Note: The ~~~~ system can be specified to only pulse if $b_i >$ threshold
- This allows the system to output patterns that are perceived by the user as having different numbers of pulses

pros: Can represent basically any set of pulses as long as the number of pulses is less than or equal to k

Con: If one intends for the user to feel a lot of high frequency vibrations, k must be very large

Data Structure (continued)

Case 2

- set of K triplets of values

pattern $= [(a_1, b_1, c_1), (a_2, b_2, c_2), \ldots, (a_K, b_K, c_K)]$

a_i: time between $(i-1)$th tuple and ith tuple

b_i: frequency of ith tuple

c: magnitude of ith tuple

Note: This is similar to my ELE464 data structure, if the c-values are ignored

Pros: K does not need to be very large, frequencies are explicitly specified

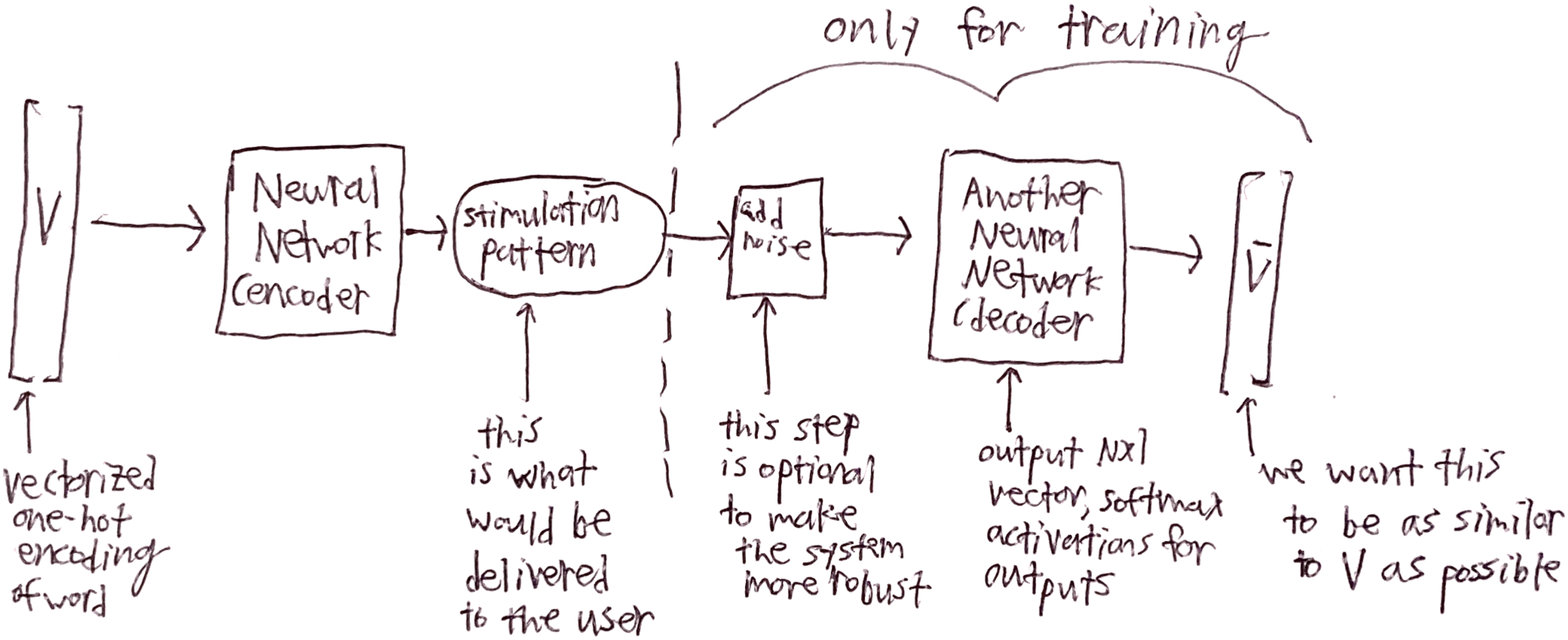
Cons: Only one frequency at a time, less robust patterns, frequency and magnitude are dependent on one another

Machine Learning

Purpose: To derive an intuitive, semantically-aware mapping from words to stimulation patterns

- Eventual ability to scale beyond the vocabulary on which the person was taught

- Rather than explicitly defining a metric of how similar two stimulation patterns are to one another, compute this implicitly



- The similarity between V and $\bar{V}$ is assessed as follows:
    - Compute the distance between every word in the vocabulary and the given word V
        - => store these distaces in an $N \times 1$ vector D
    - define the error as $D \cdot \bar{V}$ (the ~~weighted~~ dot product)
- Update the weights of both neural networks to try to minimize this error
- V with some noise added could be fed in:
    - Rather than feeding $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, feed $\begin{bmatrix} 0.95 \\ 0.025 \\ 0.025 \end{bmatrix}$
    - This could make the model more robust