Mini-Project 3: Attribute Theory of Associative Recognition and Recall and Hopfield Networks

Michael J. Kahana, Riley D. DeHaan, and Jeremy R. Manning

Part 1: Summed Similarity Model of Associative Recognition and Recall

1. Suppose you have the following vectors stored in memory, where the first four elements represent the first item of a pair, and the last four elements represent the second item. You are further given a test probe that represents a new pair made from the original A and D items.

$$
\begin{array}{ccc}
 & \begin{array}{ccc} \text{A-B} & \text{C-D} & \text{E-F} \end{array} & \text{Test Probe (A-D)} \\
\begin{array}{c} \\ \text{1st Item} \\ \\ \\ \\ \text{2nd Item} \\ \\ \\ \end{array} & \left[\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hdashline 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{array}\right] & \begin{array}{c} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{array}
\end{array}
$$

According to the summed similarity model, will the subject say "yes" to recognition of the test probe with a threshold of 0.9? Assume tau=1. **(1 point)**

As a reminder, for probe g and memories m_i,

$$
\begin{gathered}
SumSimilarity = \sum_{i=1}^{L} e^{-\tau \sqrt{\sum_{j=1}^{N}(g(j)-m_i(j))^2}} \\
Prob(yes) = Prob(SumSimilarity > C)
\end{gathered} \tag{1}
$$

2. In a cued recall paradigm, the subject is cued with the following item:

$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$$

Which item is the subject most likely to recall, and why? 1-2 sentences. **(1 point)**
For similarity S between probe g and memory m_i,

$$\begin{aligned} S(g, m_i) &= e^{-\tau \sqrt{\sum_{j=1}^{N} (g(j) - m_i(j))^2}} \\ Prob(m_i|g) &= \frac{S(g, m_i)}{\sum_{k=1}^{L} S(g, m_k)} \end{aligned} \tag{2}$$

3. Now include a context representation in the model by appending a single temporal context feature to the above feature vectors from Questions 1 and 2 of Part 1. Make the context feature count up from 1 in increments of 1 in the temporal order of E-F, followed by C-D, then A-B (right to left in the memory matrix), and finally the cue from Question 2 of Part 1. Which item is the subject most likely to recall if we include this temporal context feature, and why? **(2 points)**

Part 2: Hopfield Networks

For this problem set, your job is to create your own neural network model of memory (a Hopfield network). Below are two memories, $\mathbf{m}_1$ and $\mathbf{m}_2$ that you will store in your network. Use the techniques we discussed in class (and in the book), along with the provided equations, to answer the following questions. Show your work!

$$\mathbf{m}_1 = \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \\ 1 \\ 1 \end{pmatrix} \quad \mathbf{m}_2 = \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ -1 \end{pmatrix} \quad \mathbf{x}_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{x}_2 = \begin{pmatrix} -1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Hebbian Learning rule:

$$W(i,j) = \sum_{k=1}^{L} a_k(i)a_k(j)$$

Dynamic rule:

$$a(i) = \text{sign}\left(\sum_{j=1}^{N} W(i,j)a(j)\right)$$

If x is greater than or equal to zero, sign(x) is +1, if x is less than zero, sign(x) is -1.

1. Create a weight matrix, using Hebbian learning, that contains both $\mathbf{m}_1$ and $\mathbf{m}_2$ as stable memories. **(2 points)**

2. For each of the partial cues, $\mathbf{x}_1$ and $\mathbf{x}_2$, the activity of the first two neurons is known. Use **asynchronous updating** to calculate the activities of the remaining four neurons (in whatever order you want). Can the network retrieve both memories? Hint: update neurons 3, 4, 5, and 6 (in any order using the Dynamic rule). Then continue updating those 4 neurons until none of the values change to show that the network has stabilized. **(2 points)**

3. Now imagine the subject whose memories are represented by the weight matrix learns some additional information. Represent this information by incorporating the memory $\mathbf{m}_3$ below into the learned memory matrix. Recompute the resulting converged or final activity for the partial cue $\mathbf{x}_1$ after applying asynchronous updating as before. Which memory, if any, does the network retrieve now? What effect(s) that we've discussed in class does this result exhibit? **(2 points)**

$$\mathrm{m}_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$