Development of alternative approach (Version 3) of BTS compression algorithm

Drazen Prelec, February 19, 2019

1 OLD, Section 1 unchanged

- 4 nodes or 'players,' X, Y, Z, W
- the input to the network is one of 8 possible stimuli s_i, $i = 1, 2, .., 8$. Later on we could have more complex stimuli or combinations, $s_i, s_j, s_k...$, and noise. One can think of the stimulus as a single 'pixel' with eight possible values.
- the probability of each stimulus is $p(s_i) \geq 0$, such that $\sum_{i=1}^{8} p(s_i) = 1$, (e.g., $p(s_i) = 1/8$, $i = 1, .., 8$)
- each node is a 'player' with a binary response, "1" or "2". We designate the two possible responses of player X as $x = 1$ and $x = 2$, and similarly for Y, Z, W.
- The strategy for each player is therefore a vector of 8 probabilities, indicating the probability that he will respond with a 1 (instead of a 2) to any of the four possible stimuli:
  - Strategy for X: $x_1(s_i) = \Pr(x = 1|s_i),\ i = 1, .., 8$
  - Strategy for Y: $y_1(s_i) = \Pr(y = 1|s_i),\ i = 1, .., 8$
  - Strategy for Z: $z_1(s_i) = \Pr(z = 1|s_i),\ i = 1, .., 8$
  - Strategy for W: $w_1(s_i) = \Pr(w = 1|s_i),\ i = 1, .., 8$
  - The strategies for the opposite response is then just the complement, $x_2(s_i) = 1 - x_1(s_i)$, etc..
- We will model how strategies change as result of reinforcement. We will not model the sampling of stimuli per se, but assume that strategies are fixed in each learning iteration, and compute the long-run average reinforcement score that is associated with each of the two possible responses, for that stimulus. If the long-run reinforcement score for response $x = 1$ (for a particular stimulus) is greater than the reinforcement score for response $x = 2$, then the strategy on the next iteration is adjusted to increase the probability of responding 1 to that stimulus.

- To implement this, we need to compute at each iteration the joint probabilities of any pair of responses by any pair of nodes. Given above four strategy vectors (32 values in total, 8 for each of the 4 nodes), we can compute for each pair of nodes the 2x2 joint probability response matrix. For example, for nodes X and Y,

$$\Pr(x = j, y = k) = \sum_{i=1}^{8} x_j(s_i) y_k(s_i) p(s_i)$$

  This is just a dot product, of the two strategy vectors, weighted by stimulus probabilities. From this, one can also compute the pairwise conditional probabilities which will drive the scores. For example,

$$\Pr(x = j | y = k) = \frac{\Pr(x = j, y = k)}{\Pr(y = k)} = \frac{\sum_{i=1}^{8} x_j(s_i) y_k(s_i) p(s_i)}{\sum_{i=1}^{8} y_k(s_i) p(s_i)}$$

  This where we are short-circuiting the second learning process. In a later implementation, these probabilities would be updated using some kind of reinforcement learning.

- On each iteration of the learning we compute the expected Bayesian truth serum (BTS) score that each player would receive for responding 1 and 2, assuming that the other three players' strategies are fixed. For example, for player X the expected score for responding 1 or 2 for stimulus s_i is:

$$\begin{aligned} EV(x = 1|s_i) = \lambda \log(&\frac{x_1(s_i) + y_1(s_i) + w_1(s_i) + z_1(s_i)}{4}) \\ &- \sum_{k=1}^{2} y_k(s_i) \log \Pr(x = 1|y = k) \\ &- \sum_{k=1}^{2} z_k(s_i) \log \Pr(x = 1|z = k) \\ &- \sum_{k=1}^{2} w_k(s_i) \log \Pr(x = 1|w = k) \end{aligned}$$

$$
\begin{aligned}
EV(x=2|s_i) = \lambda \log(\frac{x_2(s_i)+y_2(s_i)+w_2(s_i)+z_2(s_i)}{4}) \\
- \sum_{k=1}^{2} y_k(s_i) \log \Pr(x=2|y=k) \\
- \sum_{k=1}^{2} z_k(s_i) \log \Pr(x=2|z=k) \\
- \sum_{k=1}^{2} w_k(s_i) \log \Pr(x=2|w=k)
\end{aligned}
$$

- The expression: $(x_1(s_i)+y_1(s_i)+w_1(s_i)+z_1(s_i))/4$ is the expected fraction of 1 votes consistent with these strategies, and can be viewed as neural 'excitation' of responding '1'. The remaining terms penalize predictability, according to the BTS formula, and represent neural 'inhibition.' The parameter λ controls the weight of inhibition. $\lambda = 3$ is excitation-inhibition balance, which is theoretically required by BTS.

- We then adjust the strategy for the next iteration, using the logit (softmax) function with β the speed of learning parameter:

$$
x_1(s_i) \rightarrow \frac{x_1(s_i)\exp(\beta(EV(x=1|s_i)-EV(x=2|s_i)))}{1-x_1(s_i)+x_1(s_i)\exp(\beta(EV(x=1|s_i)-EV(x=2|s_i)))}
$$

  Two final tweaks, are a jitter parameter J , and bounds $[\varepsilon, 1-\varepsilon]$ for the strategies. The final expression for the revised strategy is:

$$
x_1(s_i) \rightarrow \max\left\{\varepsilon, \min\left\{1-\varepsilon, J+\frac{x_1(s_i)\exp(\beta(EV(x=1|s_i)-EV(x=2|s_i)))}{1-x_1(s_i)+x_1(s_i)\exp(\beta(EV(x=1|s_i)-EV(x=2|s_i)))}\right\}\right\}
$$

  J is sampled independently on each trial from a uniform distribution on $[-\delta, +\delta]$. Typical value of δ might be .001, and the same for ε.

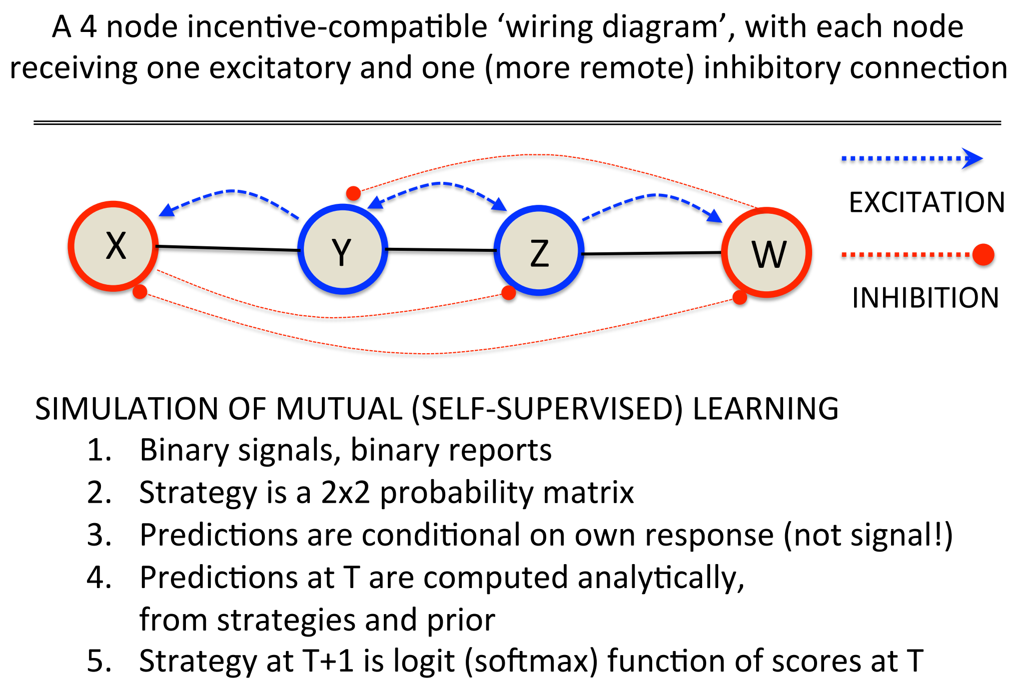

Figure 1: X and W are inhibitory, Y and Z excitatory.

2 NEW material, Version 3, 4 nodes

- Each node receives one excitatory and one inhibitory connection, indicated by arrows:

$$X \leftarrow Y,\ X \leftarrow W$$

$$Y \leftarrow Z,\ Y \leftarrow W$$

$$Z \leftarrow Y,\ Z \leftarrow X$$

$$W \leftarrow Z,\ W \leftarrow X$$

- The expected scores are then a sum of positively signed log-conditional probabilities for each excitatory incoming connection, and negatively signed conditional probabilities for each inhibitory incoming connection. For example, node X is the target of excitation from node Y and a target of inhibition from node W, hence:

$$EV(x=1|s_i) = \lambda \sum_{k=1}^{2} y_k(s_i) \log \Pr(x=1|y=k) - \sum_{k=1}^{2} w_k(s_i) \log \Pr(x=1|w=k)$$

$$EV(x=2|s_i) = \lambda \sum_{k=1}^{2} y_k(s_i) \log \Pr(x=2|y=k) - \sum_{k=1}^{2} w_k(s_i) \log \Pr(x=2|w=k)$$

For completeness, these are the equations for the other nodes:

$$EV(y=1|s_i) = \lambda \sum_{k=1}^{2} z_k(s_i) \log \Pr(y=1|z=k) - \sum_{k=1}^{2} w_k(s_i) \log \Pr(y=1|w=k)$$

$$EV(y=2|s_i) = \lambda \sum_{k=1}^{2} z_k(s_i) \log \Pr(y=2|z=k) - \sum_{k=1}^{2} w_k(s_i) \log \Pr(y=2|w=k)$$

$$EV(z=1|s_i) = \lambda \sum_{k=1}^{2} y_k(s_i) \log \Pr(z=1|y=k) - \sum_{k=1}^{2} x_k(s_i) \log \Pr(z=1|x=k)$$

$$EV(z=2|s_i) = \lambda \sum_{k=1}^{2} y_k(s_i) \log \Pr(z=2|y=k) - \sum_{k=1}^{2} x_k(s_i) \log \Pr(z=2|x=k)$$

$$EV(w=1|s_i) = \lambda \sum_{k=1}^{2} z_k(s_i) \log \Pr(w=1|z=k) - \sum_{k=1}^{2} x_k(s_i) \log \Pr(w=1|x=k)$$

$$EV(w=2|s_i) = \lambda \sum_{k=1}^{2} z_k(s_i) \log \Pr(w=2|z=k) - \sum_{k=1}^{2} x_k(s_i) \log \Pr(w=2|x=k)$$

- There is some freedom in how to set up the 6 node version. The basic principle is that (a) each node must receive at least on E and one I input; (b) I-I and E-E reciprocal connections are allowed, (b) there are no reciprocal E-I connections (these principles are respected in Figure 1). The simplest extension is to have
  - 3 excitatory nodes, E_1, E_2, E_3 all pairs reciprocally connected (targeting each other);
  - 3 inhibitory nodes, I_1, I_2, I_3 all pairs reciprocally connected (targeting each other);,
  - node E targeting only node I_i, $i = 1,2,3$.
  - node I_i targeting two nodes E_j and E_k, $i = 1,2,3$, $j,k \neq i$.