An incremental processing system must solve two computational problems: (TA) temporary ambiguity and (CD) context dependency. Consider an artificial language, \( L_1 = \{ [a \ b], [a \ c], [d \ b], [d \ c] \} \). A first word input ‘a’ creates temporary ambiguity and the system needs to allow for possible continuations \([a \ b], [a \ c]\) until it encounters disambiguating information. When the input changes to ‘b’, the system needs to exclude \([d \ b]\) in favor of \([a \ b]\), as the former is inconsistent with the preceding context (which is no longer in the input itself). Solving these two problems (TA, CD) is a challenge for neural-network models that utilize highly local constraints, operate over distributed representations, and do not explicitly monitor complete structures. We present a Gradient Symbolic Computation (Smolensky, Goldrick, & Mathis, 2014) model that meets the challenges.

The proposed model performs stochastic gradient ascent in harmony (Smolensky, 1986), a measure of constraint satisfaction. Harmony is defined by two components. The first harmony component is the grammar, which is specified by local constraints. These turn out to assign highest harmony to a blend state—a conjunction of partially-active discrete structures. The second harmony component, quantization, penalizes blend states. The relative strength \( q \) of quantization’s contribution to the harmony increases during parsing. Fig 1a illustrates the dynamic changes to the harmony surface during processing of ‘a b’ based on word input and \( q \).

We argue that the appropriate control of \( q \) addresses the two computational problems (TA, CD). At low \( q \) values, with ‘a’ presented, the model handles temporary ambiguity (TA) by moving to a blend state in which possible continuations \([a \ b], [a \ c]\) are most strongly active (Fig 1a1). As \( q \) increases, local maxima appear at all states encoding purely discrete structures, and between these, harmony valleys emerge. When the input changes to ‘b’, \([d \ b]\) has the same harmony as \([a \ b]\) but the harmony valley between these states prevents the system from mis-parsing (CD; Fig 1a2). When \( q \) increases too slowly, the model can fail to reject context inappropriate \([d \ b]\). When \( q \) increases too quickly, the model can commit too early to one predicted structure \([a \ c]\) over another \([a \ b]\), before the disambiguating information ‘b’ comes in.

Preliminary results suggest that optimal control of \( q \) can be learned by reinforcement learning. We constructed a deep Q network (DQN) (Mnih et al., 2015) (see Fig 1b) that takes the current state, current external input, and current \( q \) value of the GSC model as input and computes the reward values of possible actions: holding or increasing \( q \) by different amounts for the next GSC network iteration. For each action taken, the DQN received a reward at the final iteration: harmony of the final state plus a reward based on parsing accuracy. This allowed the model to parse the sentences of \( L_1 (\text{acc} = 1; \text{of 100 trials for each sentence}). Learning also acquired appropriate \( q \) policies for two more complex languages: \( L_2 = \{ [a \ b], [a \ c], [d \ e], [d \ f]\} \) where \( P(b|a) > P(c|a) \) and \( P(e|d) = P(f|d): acc = .998 \). \( L_3 = \{ [[a \ b] \ c], [[a \ d] \ c], [b \ c]] \}: acc = .993 \). We note that, with random initialization, the model often failed to learn an optimal \( q \) policy on \( L_2 \) and \( L_3 \). We are exploring alternative reward regimes and model architectures.

In sum, the GSC model allows a distributed model with local constraints to solve two key computational problems (TA, CD). Bottom-up input activates all locally coherent structures. Quantization dynamics, under a control schedule (policy) acquired via reinforcement learning, separates structures that are globally coherent vs. incoherent.

References

