Sentence processing in spiking neurons: A biologically plausible left-corner parser

Terrence C. Stewart (tcstewar@uwaterloo.ca)
Xuan Choo (fchoo@uwaterloo.ca)
Chris Eliasmith (celiasmith@uwaterloo.ca)
Center for Theoretical Neuroscience, University of Waterloo
Waterloo, ON, Canada N2L 3G1

Abstract

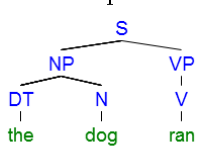
A long-standing challenge in cognitive science is how neurons could be capable of the flexible structured processing that is the hallmark of cognition. We present a spiking neural model that can be given an input sequence of words (a sentence) and produces a structured tree-like representation indicating the parts of speech of that it has identified and their relations to each other. While this system is based on a standard left-corner parser for constituency grammars, the neural natural of the model leads to new capabilities not seen in classical implementations. For example, the model gracefully decays in performance as the sentence structure gets larger. Unlike previous attempts at building neural parsing systems, this model is highly robust to neural damage, can be applied to any binary constituency grammar, and requires relatively few neurons [insert number here].

Keywords: Neural engineering framework; vector symbolic architectures; left-corner parsing; syntax; binary trees; computational neuroscience

Introduction

Human language processing requires not only the ability to represent complex structures, but also the ability to create these representations out of a serial sequence of words. This system is flexible enough to work for a huge variety of possible sentences, including (crucially), sentences that have never been seen before.

Modern linguistics is founded on the principle that these structured representations are tree-like, and that this tree structure imposes useful order on the sentence. For example, "the dog ran" can be parsed as follows:



Here the sentence (S) is divided into a noun phrase (NP) and a verb phrase (VP). The noun phrase is divided into a determiner (DT) "the" and a noun (N) "dog". The verb phrase consists of a single verb (V) "ran".

To describe the space of possible trees, rules as to what constituents can be found within other constituents can be defined. For example, the above sentence is consistent with the following constituency grammar rules:

S → [NP VP]	DT → "the"
NP → [DT N]	N → "dog"
VP → [V]	V → "ran"

By adding more structural rules (on the left), we can build more complex sentences. By adding more words (on the right), we can increase our vocabulary.

Left-Corner Parsing

As more rules are included in the grammar, parsing becomes more complicated. More rules mean more possibilities, and finding a tree that is consistent with the rules can become computationally expensive. For example, there may be the two rules VP → [V] and VP → [V NP], leading to an explosion of possible structures to search through. Furthermore, a single word may have multiple interpretations, such as N → "dog" and V → "dog".

One standard approach to addressing this problem is to use a Left-Corner parsing algorithm. The idea here is to combine bottom-up information with top-down information. The top-down information is what part of speech we are currently looking for (a sentence, a noun phrase, a determiner, etc.). The bottom-up information is the single word we are currently processing. So, if we are currently looking for a noun, we will first try interpreting the word "dog" as a noun, rather than a verb. If this does not lead to a successful parse, then the algorithm will backtrack to this point and try the other option. This approach drastically reduces the amount of backtracking needed.

For "the dog ran", the algorithm proceeds as follows:

- Top-down: look for S
- Bottom-up: see "the"
- Apply rule DT → "the"
- Apply rule NP → [DT N]
- Store the partially completed tree
    - Top-down: look for N for previous rule
    - Bottom-up: see "dog"
    - Apply rule N → "dog"
    - Merge with previously stored tree
- Apply rule S → [NP VP]
- Store the partially completed tree
    - Top-down: look for VP for previous rule
    - Bottom-up: see "ran"
    - Apply rule V → "ran"
    - Apply rule VP → [V]
    - Merge with previously stored tree

As has been pointed out (Johnson-Laird, 1983), this algorithm matches well with observed human sentence comprehension. For example, it has difficulty with "garden-path" sentences such as the classic "the horse raced past the barn fell", which is difficult for most people to interpret even though it has the same form as the easier sentence "the deer shot by the hunter fell". A left-corner parser has the same tendency as people do to connect the ambiguous word "raced" as part of a S → [NP VP] rule.

Previous Models

Rick Lewis' ACT-R model
Neural blackboard architectures
Our previous cogsci paper

Neural Engineering Framework

Vector Symbolic Architectures

Semantic Pointer Architecture

Left-Corner Parsing in SPA

Parsing Results

(show that it works and what sort of sentences and trees it can do)

Parsing Accuracy

(how much accuracy do we need in the representation? What does that translate to in terms of needed numbers of neurons in the buffers? Lots of graphs. Something about depth would be good, too. Maybe another version of the prob/retries plot for a more complex sentence.)

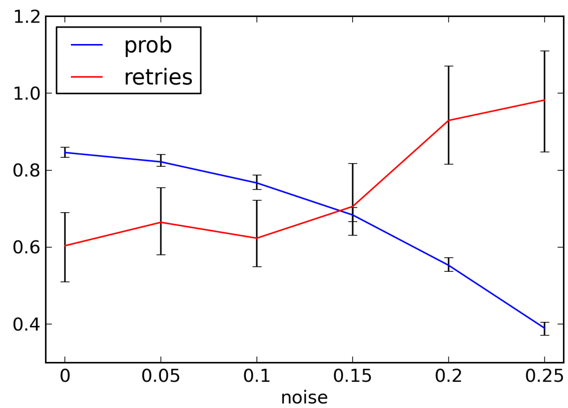
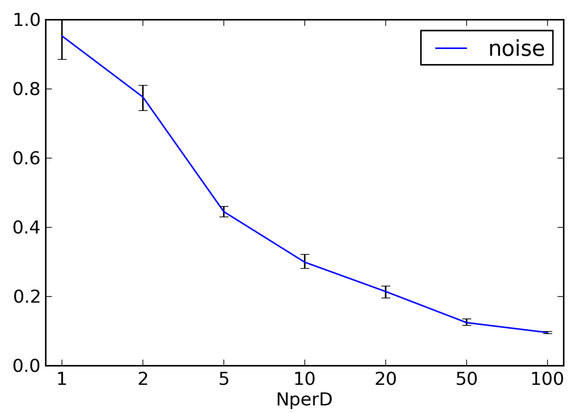


Figure: probability of correctly parsing and number of retries needed as the amount of noise changes



Number of neurons needed per dimension to get a certain level of noise

(mention how this shows that neural death will gradually decrease accuracy)

Using Parsed Commands

(connect to Xuan's stuff)

Future Directions

Automatically learning utilities to handle repair/recovery

Conclusions

Acknowledgments

This work was funded by... (ONR, ?)

References

Ball, J. (2011). A Pseudo-Deterministic Model of Human Language Processing. *33rd Cog. Sci. Society Conference.*

Choo, X and Eliasmith, C. (2013). General Instruction Following in a Large-Scale Biologically Plausible Brain Model. *35th Cog. Sci. Society Conference.*

Eliasmith, C. (2013). *How to build a brain*. Oxford University Press, New York, NY.

Eliasmith, C. & Anderson, C. (2003). *Neural Engineering.* Cambridge: MIT Press.Eliasmith et al., 2012

Gayler, R. (2003). Vector Symbolic Architectures Answer Jackendoff's Challenges for Cognitive Neuroscience, in Slezak, P. (ed). *Int. Conference on Cognitive Science,* Sydney: University of New South Wales, 133–138.

Johnson-Laird, P. N. (1983). *Mental models*. Cambridge, MA: Harvard University Press.

Plate, T. (2003). *Holographic Reduced Representations*, CSLI Publications, Stanford, CA.

Stewart, T.C., Choo, X., and Eliasmith, C. (2010). Dynamic Behaviour of a Spiking Model of Action Selection in the Basal Ganglia. *10th Int. Conf. on Cognitive Modeling.*