Comment on the Now-or-Never Bottleneck by Christiansen and Chater

Neural Constraints and Flexibility in Language Processing

Christian R. Huyck

Middlesex University

London, NW4 4BT, UK

44-208-411-5412

c.huyck@mdx.ac.uk

www.mdx.ac.uk/chris

Abstract

Humans process language with their neurons. Memory in neurons is supported by neural firing, and short and long-term synaptic weight change; the emergent behaviour of neurons, synchronous firing and cell assembly dynamics, is also a form of memory. As the language signal moves to later stages, it is processed with different mechanisms that are slower but more persistent.

The proposed Now-or-Never bottleneck in language processing has a great deal of evidence to support it. Like all cognitive processes, language processing must be implemented in neurons, and the bottleneck is a neural one. Signals from the environment must be processed by neurons, and those neurons must keep a memory trace of those signals, or they will be lost. Moreover, any processing mechanism must not only be implemented by the behaviour of neurons, but in the case of language, the process must be learned by those neurons.

Neural memory comes in several forms. Neurons spike propagating signals across their synapses to post-synaptic neurons taking tens of milliseconds. Neurons can be wired into cell assemblies (Hebb 49) that can persistently fire for seconds. Synaptic weights can be modified for seconds to minutes via short-term potentiation (STP), or for days, months or longer, via long-term potentiation (LTP). The formation of a cell assembly, via potentiation, can form a circuit that can last indefinitely. When that long-term memory is activated, by a cascade of neural firing in the cell assembly, the long-term memory is also an active short-term memory.

When a sentence is parsed, either in speech or text, it is generally one pass. This can be seen via eye tracking evidence, especially when repairs are needed (Just and Carpenter, 1980). This is typically simulated with a stack, but a memory based mechanism (Lewis and Vasishth, 2005) can eliminate the need for a stack. This has been implemented in a neural parsing model (Huyck, 2009), with the persistence of the cell assembly showing the strength and duration of the memory. The author is unaware of any existing simulated neural mechanism for backtracking in parsing.

One important aspect of eliminating the stack in parsing is that it reduces the need for binding. Binding is another type of neural memory mechanism that, though needed in standard computational models, is typically overlooked. In a standard program, if a variable is assigned a value, the two are bound. This is usually a primitive operation so it is ignored. Binding in a neural system is more difficult because it is not primitive. There are various binding mechanisms with synchronous firing being most widely used in the literature (Fuster and Alexander, 71). Two bound assemblies fire in roughly the same firing pattern, while another pair (or more) can be bound in a different pattern. Synchronous binding requires the neurons to continue firing. Moreover, there are a small number of patterns that can be supported simultaneously so there are a limited number of bindings; all of the bound neurons do not fire at the exact same time, so separate patterns must be quite distinct. Another option is to bind via STP. This has neither of these limits with a much larger number of bindings supported, and the duration being up to minutes; it does however take longer to form. Binding can also be done via LTP, but this shades into permanent associative memory.

When processing language, it is faster and safer to avoid binding. When it is necessary, lower level processing is likely to use synchrony. Higher level processing is likely to use STP. So the speech signal uses synchrony; neurons representing the prime formants fire synchronously in the auditory cortex (Eggermont, 2001). The simulated neural parser (Huyck, 2009) used STP for binding the slots in the neural implementation of verb frames associated with sentences. These could be used immediately after sentence processing to retrieve the meaning of the sentence, but they were gradually erased by the STP fading. The neurons that supported the binding were reused later for processing other sentences.

Finite state automata (FSA) do not require binding. Evidence from text engineering to support the bottleneck, is that the Message Understanding Competitions for Text Extraction (Appelt et al, 93) converged on a cascade of FSAs to solve the problem of processing text. One FSA separated words, a second categorised them lexically, a third did simple phrase parsing, and a fourth combined phrases. These could be run in a cascade, and perhaps this is the basic mechanism that the brain uses.

As Christiansen and Chater note, the bottleneck also has ramifications for learning. Firstly, the whole language cascade (whatever that may be) is being learned simultaneously. Initially, low level phenomena, like morphemes, are learned. Later, larger systems like simple phrase grammars begin to be learned, but the lower level systems are still being developed. It is not known how these biological neural systems work, much less how they are learned. One mechanism may be that things are being learned, CAs are formed; CAs can be connected to form FSAs. Binding may be involved initially, and the synapse can then be modified to combine CAs into FSAs; STP can support reverberation, which can then lead to LTP. While one FSA in the cascade is being learned, both FSAs above and below it can be learned so that the whole system continues to improve.

At the highest level, dialogue and above, the bottleneck begins to disappear. Rich cognitive maps support this kind of processing and memory is mostly via LTP and CA circuit dynamics. Since these CAs can persistently fire, and the circuits can be reactivated via associative memory, it is possible to remember large amounts of things. I can still remember some of the dialogue from the movie I saw this weekend, and the plot.

There is solid support for Now-or-Never bottleneck in language processing, though the duration of the bottleneck is reduced as the signal passes through stages of language processing. The distributed nature of neural processing supports multiple stages in processing, and the simultaneous learning of these stages. Processing and learning is implemented in neurons, though CA dynamics and binding issues are often not considered by researchers. By expanding understanding and modelling at the neural level, language processing can be better understood, and more robust language processing systems constructed.

References

D. Appelt, J. Hobbs, J. Bear, D. Israel, and M. Tyson (1993) FASTUS: A finite-state processor for information extraction from real-world text. IJCAI pp. 1172-1178

J. Eggermont (2001) Between sound and perception: reviewing the search for a neural code. Hearing research 157(1), pp. 1-42

J. Fuster and G. Alexander (1971) Neuron Activity Related to Short-Term Memory. Science 173, pp. 652-654

D. Hebb (1949) The Organization of Behavior. John Wiley and Sons, New York.

C. Huyck (2009) A Psycholinguistic Model of Natural Language Parsing Implemented in Simulated Neurons. Cognitive Neurodynamics 3(4), pp. 316-330

M. Just and P. Carpenter (1980) A theory of reading: From eye fixations to comprehension. Psychological Review, 87(4), pp. 123-154

R. Lewis and S. Vasishth (2005) An activation-based model of sentence processing as skilled memory retrieval. Cognitive Science 29:3, pp. 375-419

Acknowledgements: This work was supported by the Human Brain Project Grant 604102, Neuromorphic Embodied Agents that Learn

Main 977/1000 words

Abstract 58/60

References 133

Total 1227