SEMINAR 21 SEP 2020

PROPOSAL S7 CSE B

TOPIC:

**NEURAL TURING MACHINES**

**SUBMITTED BY**

VISHNUCHANDRA MC

**ABSTRACT**

We extend the capabilities of neural networks by coupling them to external memory resources, which they can interact with by attentional processes. The combined system is analogous to a Turing Machine or Von Neumann architecture but is differentiable end-to end, allowing it to be efficiently trained with gradient descent. Preliminary results demonstrate that Neural Turing Machines can infer simple algorithms such as copying, sorting, and associative recall from input and output examples.

Computer programs make use of three fundamental mechanisms: elementary operations (e.g., arithmetic operations), logical flow control (branching), and external memory, which can be written to and read from in the course of computation (Von Neumann, 1945). Despite its wide-ranging success in modelling complicated data, modern machine learning has largely neglected the use of logical flow control and external memory. Recurrent neural networks (RNNs) stand out from other machine learning methods for their ability to learn and carry out complicated transformations of data over extended periods of time. Moreover, it is known that RNNs are Turing-Complete (Siegelmann and Sontag, 1995), and therefore have the capacity to simulate arbitrary procedures, if properly wired. Yet what is possible in principle is not always what is simple in practice. We therefore enrich the capabilities of standard recurrent networks to simplify the solution of algorithmic tasks. This enrichment is primarily via a large, addressable memory, so, by analogy to Turing’s enrichment of finite-state machines by an infinite memory tape, we 1 arXiv:1410.5401v2 [cs.NE] 10 Dec 2014 dub our device a “Neural Turing Machine” (NTM). Unlike a Turing machine, an NTM is a differentiable computer that can be trained by gradient descent, yielding a practical mechanism for learning programs. In human cognition, the process that shares the most similarity to algorithmic operation is known as “working memory.” While the mechanisms of working memory remain somewhat obscure at the level of neurophysiology, the verbal definition is understood to mean a capacity for short-term storage of information and its rule-based manipulation (Baddeley et al., 2009). In computational terms, these rules are simple programs, and the stored information constitutes the arguments of these programs. Therefore, an NTM resembles a working memory system, as it is designed to solve tasks that require the application of approximate rules to “rapidly-created variables.” Rapidly-created variables (Hadley, 2009) are data that are quickly bound to memory slots, in the same way that the number 3 and the number 4 are put inside registers in a conventional computer and added to make 7 (Minsky, 1967). An NTM bears another close resemblance to models of working memory since the NTM architecture uses an attentional process to read from and write to memory selectively. In contrast to most models of working memory, our architecture can learn to use its working memory instead of deploying a fixed set of procedures over symbolic data. The organization of this report begins with a brief review of germane research on working memory in psychology, linguistics, and neuroscience, along with related research in artificial intelligence and neural networks. We then describe our basic contribution, a memory architecture and attentional controller that we believe is well-suited to the performance of tasks that require the induction and execution of simple programs. To test this architecture, we have constructed a battery of problems, and we present their precise descriptions along with our results. We conclude by summarizing the strengths of the architecture.

**REFERENCES**

* <https://arxiv.org/abs/1410.5401>
* <https://en.wikipedia.org/wiki/Neural_Turing_machine>
* <https://github.com/MarkPKCollier/NeuralTuringMachine>