A STICKY-BIT APPROACH FOR LEARNING TO REPRESENT STATE. Jonathan R. Bachrach.

(bachrach@cs.umass.edu) Computer and Information Science Department, University of Massachusetts, Amherst, MA 01003

Many tasks require knowledge of past events in order to successfully choose actions. For example, current percepts are inadequate in speech recognition and language understanding. Since only a finite amount of state may be stored, it is important for a system to remember only significant past events. Furthermore, it is crucial that learning algorithms develop these state representations given only delayed and possibly noisy teaching information: the system cannot know what state will be useful in the future. Connectionist modeling is just beginning to address these issues and researchers have produced various solutions. One approach is to use tapped delay lines, or variations thereof, in which weights are assigned to inputs at successive delays, making their representations necessarily dependent on fixed delays. This approach is subject to the following limitations: learning scale-invariant representations is quite difficult, and the depth of the memory is bounded. Another approach involves setting memory variables that remain set until altered by other processes. This state representation approach has the potential to yield temporal scale and translation invariances important in speech recognition and permits memory of events that ocurred arbitrarily long in the past.

We employ a version of this latter scheme on the following class of learning problems: learning to recognize temporal sequences of events based on training information that arrives at the end of each admissible sequence, where admissible sequences are not defined in terms of specific temporal durations between events. In the general case this is equivalent to recognizing regular languages given only positive and negative examples of strings in the language and given n or fewer states, a problem known to be intractable [Gol78]. Therefore, we restrict our attention to very simple regular languages.

The main component of our approach is a teachable flip-flop, called a sticky-bit, that records the occurrence of significant events. In their most connectionistic form, sticky-bit's can be created out of semi-linear units with a self-recurrent connection and symmetric logistic squashing function that ranges from -1 to 1. With the proper setting for the feedback weight (namely > 2), these elements have two attractors of the same magnitude but different sign, permitting the storage of one bit of state information. By adjusting the input weights, the sticky-bit can divide the input patterns into those events that cause it to converge to the first attractor, converge to the second attractor, or stay at the same attractor. We derived a gradient-descent learning rule that generalizes back-propagation to work with sticky-bit's. Instead of back-propagating in an unfolded network (as described by [RHW85]), this learning rule keeps a modulated trace on each input line and updates the weights based on this trace (Mozer independently discovered the same algorithm). This trace captures all the information derived from the unfolding process but without the excess space and time penalties. Conceptually, the trace remembers what event caused the state change, since the sticky-bit must decide on a state before the state is used and thus before the sticky-bit can receive a teaching signal. Even though the modulated trace itself will eventually forget events, it can be used to learn state representations that can remember indefinitely.

Although this real-time back-propagation learning algorithm represents an advance in learning algorithms for recurrently connected networks, we discovered several limitations of this algorithm. This feedback unit is unable to learn to latch events coded as spatial conjunctions of inputs because of false triggering on temporal conjunctions. This remains true despite attempts to increase the "stickiness" of the attractors (in order to make it converge quicker) or to increase the space between inputs (in order to give it more time to converge). Finally, although this algorithm will scale to two layers of sticky-bit's as in [WS86], the extension to layered sticky-bit's poses major obstacles.

Watrous and Shastri's Temporal Flow Model [WS86] is a two layered network of self-recurrently connected back-propagation units, like our original design but with adaptive feedback weights. This network can be configured (with proper feedback weights) to exhibit both temporal scale and translation invariance, and, by unfolding the network in time (as described in [RHW85]), they are able to perform gradient descent. Unfortunately, this approach appears to be prone to the same problems that the feedback sticky-bit exhibits. Additionally, their finite network unfoldings only approximates the true gradient and involves more space and time overhead than our algorithm. Furthermore, learning the feedback weight takes an excessive amount of time, since only a very small range of values are useful.

We abandoned the feedback sticky-bit approach for a simpler algorithm that involves a predictor and a latch. The predictor attempts to forecast what state will be required next and the latch holds onto strong predictions. Learning only occurs in the predictor, and it utilizes a simplified version of the tracing approach of the previous design. This approach solves the false triggering problem with (1) an instantaneous latch and (2) the separation of latch from predictor. It also scales quite nicely. Unfortunately, this state representation has trouble capturing certain temporal properties of events. In summary, our event-based approach overcomes some of the limitations of the tapped delay line and "temporal flow" approaches, but would benefit from additional representational power.

[Gol78] E. Mark Gold. Complexity of automaton identification from given data. *Information and Control*, 37, 1978.

[RHW85] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning Internal Representations by Error Propagation.* Technical Report 8506, Institute for Cognitive Science, C-015; University of California, San Diego; La Jolla, CA 92093, 1985.

[WS86] Raymond L. Watrous and Lokendra Shastri. *Learning Phonetic Features Using Connectionist Networks: An Experiment in Speech Recognition.* Technical Report, Department of Computer and Information Science; Moore School; University of Pennsylvania, 1986.