Learning a Deterministic Finite Automaton with a Recurrent Neural Network

* Laura Firoiu, Tim Oates, and Paul R. Cohen

Computer Science Department, University of Massachusetts at Amherst
lfiroiu,oates,cohen@cs.umass.edu

Abstract. We consider the problem of learning a finite automaton with recurrent neural networks from positive evidence. We train an Elman recurrent neural network with a set of sentences in a language and extract a finite automaton by clustering the states of the trained network. We observe that the generalizations beyond the training set, in the language recognized by the extracted automaton, are due to the training regime: the network performs a "loose" minimization of the prefix DFA of the training set, the automaton that has a state for each prefix of the sentences in the set.

1 Introduction

1.1 The Problem of Inducing a Deterministic Finite Automaton (DFA)

Our interest in DFA inference is partly induced from the larger goal of explaining how humans learn the grammar rules of their native language. There have been debates on whether children learn in an unsupervised mode, just by listening to other language speakers, or if they have innate knowledge of language. Therefore, it is an interesting problem to see what can be learned just by "listening to others", that is, from a set of grammatically correct sentences. While the complex syntactic rules of natural language cannot be encoded efficiently as regular grammar productions, fragments of language can be represented by finite automata. Thus this representation, which we prefer due to its simplicity, is adequate for some language fragments. Throughout the paper we will call the symbols of the alphabet "words" and the strings over the alphabet "sentences".

1.2 Deterministic Finite Automata and Recurrent Neural Networks

The problem of learning finite automata is difficult. Gold ([7]) showed that if the language is not known to be finite, then learning from positive evidence is...
not always possible, even from infinite sequences of words. Some other hardness (NP-complete) results concern finding the minimum, or a polynomial approximation of the automaton consistent with a finite sample([11]). In this work we study the problem of using simple recurrent neural networks (RNN) to induce a regular grammar that is consistent with a training set of sentences from the language, i.e. from positive evidence only. We use an Elman recurrent network to induce a DFA, and are interested in understanding what the network learns.

The contribution of this paper is in informally characterizing the automaton extracted from the network states as a “loose” minimization of the prefix DFA (the automaton that has a state for each prefix of the sentences in the set).

The absence of negative examples renders the problem of learning a DFA underspecified and may lead to one of two extreme assumptions. One assumption is that negative examples do not exist, so the language consists of all the sentences over the alphabet and the minimum DFA that represents the language is the automaton with just one state. The other assumption is that the negative examples are all the sentences that are not in the training set. The language is then the training set and it is accepted by the prefix DFA. This DFA has a unique state for each prefix of the sentences in the training set and can be constructed and minimized in polynomial time (on the size of the training set). We call the minimized automaton the training set DFA. Since we do not make any explicit assumptions about the target language, the problem setting resembles the conditions of Gold’s theorem on the impossibility of language learning from positive evidence, with the provision that the sequence of examples cannot be infinite.

Recurrent neural networks and deterministic finite automata are both state devices. It has been shown (see [13]) that there is an immediate encoding of a DFA with \( n \) states and \( m \) input symbols into a simple recurrent neural network with \( mn \) state units and integer weights. A DFA can be easily extracted from such an RNN. Because neural networks can be trained with the backpropagation algorithm they have been a natural choice for the tasks of DFA induction. For accepting automata, the network target output can be either the word following the current input, i.e. (the prediction task) ([4], [2]), or if both positive and negative examples are present, a value encoding the membership in the language of the current string ([6]). As in [2] and [4] we use an Elman recurrent network trained on the prediction task to induce a DFA, and are interested in understanding what the network learns. In section 2 we describe in more detail the setting of the learning task, the experiments and the results. In section 3 we observe that the network learns an approximation of the training set DFA of the training set and conclude that this network architecture and training regime are biased towards the extreme case where the training set is the entire language.

2 The Learning Task

2.1 The Grammar and the Training Sets

We wrote a small context free grammar (CFG), that generates natural-sounding sentences, similar to the one used in [5]. From this CFG we obtained regular