

An Artificial Neural Network Application On Natural Language Recognition

M.Toygar Karadeniz
26 March, 1996

Introduction

Most of the programming language constructs have an inherently recursive structure that can be expressed by the Context-Free-Grammars (CFG). These type of grammars is easily understood and fully examined in several research areas. Many efficient parsing algorithms have been invented to recognize the sentences generated by those CFG's. The programming language compilers, mainly, parse the source code, also with doing some additional processes like symbol table maintenance or code generation.

Although, the main usage area of the CFG's currently, are the programming language structure definitions, some other language structures ,also, obey the rules governed by these type of grammars. One of the most important of these, is the 'Natural Language'. The structure of the Natural Language which we use everyday, is just suitable to be expressed as CFG. But the problem concerned with it, is that the recursive orderings of the terminals and of groups of terminals, has a great deal of importance in that case and recognition of whole sentences may be handled easily if the relative orderings taken into account in the first place rather than the absolute orderings in the full sentence. Artificial neural network application seems to be more suitable to handle the huge recognition phase of a natural sentence than a dedicated parser.

This project proposal is about the implementation of an artificial neural network trained to determine the grammaticality of strings of syntactic categories in the Natural Language.

Natural Language Processor Project

Order is the main concept in making decisions about the recognition of the grammar at hand. For example, 'the ball is dropped' is grammatically correct, but 'dropped the is ball' is incorrect. Our artificial neural network application will capture the order information in a sentence by making pairs, triples etc. from the terminals of the grammar and prepare a node for each of these orderings. For example, consider the following sentences : 'the ball is dropped' and 'the small blue ball is dropped'. Suppose that our artificial neural network was trained on the first sentence. Then, it will be able to recognize the second sentence as well, although in most of the cases concerning CFG's, addition of an extra terminal in front of the sentence will make it grammatically incorrect. In the most general sense, our artificial neural network will be able to learn the k-tuples that are the most strongly indicative of both grammatically correct and incorrect sentences by being trained over a training set. These k-tuples are also the most common ones. Then, it will be easy for it to generalize and recognize the unseen sentences from the same grammar.

Now, a very primitive version of our artificial neural network training algorithm may be given :

- a) A sentence is encountered.

- b) All orderings that constitute the k-tuples, are generated.
- c) All matching k-tuples with the predetermined nodes in the net, is activated with a value of 1.
- d) If there are no matching k-tuples with the new one, a new input node is created and the new one is dedicated to this new node.
- e) The output node that has the largest value is determined.
- f) The response-to-be-given is taken as an input from the training set. If it does not match the node determined in (e), it is added to the output layer. If they match, the weights between the node described in (e) and the input layer, are improved.

This algorithm is iterated through the training set to reach a certain level of stability.

Conclusion

As a final word, the Natural Language Processor algorithm that is proposed in this paper, is still at a very premature state. In fact, it only exists as an idea for the moment. So, everything described, is a good estimation for an implementation rather than the implementation itself. But this particular paper is good at describing the intended application where neural networks are the main concepts and giving the idea about the intended algorithm. Surely, the loose points of the algorithm, will be the subject of our effort.