Optimizing Initial Configurations of Neural Networks for the Task of Natural Language Learning

Jaime J. Dávila

Graduate School and University Center, City University Of New York
33 West 42nd Street
New York, NY 10031, USA
jdavila@broadway.gc.cuny.edu

One approach used to develop computer systems for natural language processing (NLP) is that of Artificial Neural Networks (NNs). Because of the large number of parameters a NN has (e.g. network topology, learning algorithm, transfer functions) and the way they interact, finding optimal parameter values for any particular task can be extremely difficult.

Topology can greatly affect the performance of a NN. Stolcke (1990) found that simple sentences could be processed by NNs with one hidden layer. Performance degraded, however, when the NN was presented with embedded sentences, for which more complex topologies were needed. What topology to use for any particular task is still an open question. Most decisions regarding NN topology are based on ideas of how the problem should be tackled. Jain (1991) used a NN that first decomposed a sentence into phrases and later defined relationships between the phrases. Miikkulainen (1996) divided a NN into a parser, a segmenter, and a stack. These NN vary in the number of layers, the number of nodes in each layer, and how these layers are connected to each other.

Another aspect affecting performance is the corpus used to train the NN. Nenov and Dyer (1994) found that training with individual words before showing sentences increased performance. Elman (1993) found that training a NN with simple sentences first and complex sentences later produced better results than presenting all sentences in one session.

Although researchers have been successful in using NN for NLP, choosing an initial configuration is quite complex. An automated process that can find an optimal set of parameters would be helpful in designing such a system.

In my research I use Genetic Algorithms (GAs) to find what NN parameter values produce better performance for a particular NL task. There is one input node for each word in the vocabulary. A sentence is presented to the NN one word at a time by activating the corresponding node at the input layer. The NN incrementally generates a description of the input by correctly identifying the parts of the sentence and how they relate to each other. For example, in the sentence <the boy runs> the NN should respond by (1) identifying each word in the sentence, e.g. <runs> is a movement verb; (2) identifying phrases, e.g.<the boy> is a noun phrase; and (3) identifying the relationships among phrases, e.g. <the boy> is the agent of <runs>. This is similar to Jain (1991).

The NN has 75 hidden nodes, divided into between 1 and 30 layers, as determined by the GA. The GA also determines how these layers are connected to each other, and which learning algorithm and transfer functions to use.

The NN is tested on a language of 508 sentences with three different complexity levels. The basic training set is 20% of the complete language. The GA makes three major decisions about training: (1) whether to train with sentences from all complexity levels at once, or with sentences from the simpler level first and more complex sentences later, (2) whether to train with sentences from all complexity levels in the same proportion, or with more sentences from one level or another; (3) whether the training set should be increased past the basic 20% . If it is, the NN's fitness is decreased by a factor equal to the increase in the training set.

By studying the configurations chosen by the GA, I hope to identify which parameters are critical for the NLP task and which values for these parameters produce the best performance. These results will provide a better understanding of NN behavior and point to improved NN configurations.

References

Elman, J. 1993. Learning and Development in Neural Networks: The Importance of Starting Small. *Cognition* 48:71-99.

Jain, A. 1991. Parsing Complex Sentences with Structured Connectionist Networks. *Neural Computation* 3:110-120.

Miikkulainen, R. 1996. Subsymbolic Case-Role Analysis of Sentences with Embedded Clauses. *Cognitive Science* 20:47-73.

Nenov, V.; and Dyer, M. 1994. Perceptually Grounded Language Learning. *Connection Science* 6.

Stolcke, A. 1990. Learning Featured-based Semantics with Simple Recurrent Networks, Technical Report, TR-90-015. International Computer Science Inst., Berkeley, CA.