LANGUAGE, TRANSLATION AND NEURAL NETWORKS: OBSTACLES AND LIMITATIONS

Masoud Azizinezhad
Department of English Language and Literature, Toyserkan Branch,
Islamic Azad University, Toyserkan, IRAN
mazizinezhad@gmail.com

Masoud Hashemi
Department of English Language and Literature, Toyserkan Branch,
Islamic Azad University, Toyserkan, IRAN
masoudhashemi38@yahoo.com

ABSTRACT

The neurons are regarded as the simple units of processing in the brain which help it order information in an spontaneous manner. In the late 1940s Donald Hebb made one of the first hypotheses of learning with a mechanism of cortical remapping also known as neural plasticity, according to which brain is regarded as a structure with the ability to change when affected by experience (Begley, 2007). Since then many interdisciplinary attempts have been made to develop more capable units of artificial language processing. In this article attempts has been made to provide the readers with a summary of the application areas of artificial neural networks as well as the problems researchers face in dealing with designing artificial neural networks for language processing. First a short introduction to the artificial intelligence and neural networks and differences among them is presented, then attention has been paid to the limitations related to simulation of levels of processing in natural language focusing on listening and reading (the linearity issue, the non-invariance issue, the normalization issue, the accommodation issue, etc (Field, 2003). A number of the related theories and their loopholes will also be discussed and elaborated.

Key words: Neural network, Translation, Language, Obstacles, Limitations

INTRODUCTION

What are neural networks?

The term neural network was traditionally used to refer to a network or circuit of biological neurons (HOPFIELD, 1995). The modern usage of the term often refers to artificial neural networks, which are composed of artificial neurons or nodes. Thus the term has two distinct usages:

- Biological neural networks are made up of real biological neurons that are connected or functionally related in a nervous system. In the field of neuroscience, they are often identified as groups of neurons that perform a specific physiological function in laboratory analysis.
- Artificial neural networks are composed of interconnecting artificial neurons (programming constructs that mimic the properties of biological neurons). Artificial neural networks may either be used to gain an understanding of biological neural networks, or for solving artificial intelligence problems without necessarily creating a model of a real biological system. The real, biological nervous system is highly complex: artificial neural network algorithms attempt to abstract this complexity and focus on what

may hypothetically matter most from an information processing point of view. Good performance (e.g. as measured by good predictive ability, low generalization error), or performance mimicking animal or human error patterns, can then be used as one source of evidence towards supporting the hypothesis that the abstraction really captured something important from the point of view of information processing in the brain. Another incentive for these abstractions is to reduce the amount of computation required to simulate artificial neural networks, so as to allow one to experiment with larger networks and train them on larger data sets.

History of the Neural Network Analogy

In the brain, spontaneous order appears to arise out of decentralized networks of simple units (neurons).

Neural network theory has served both to better identify how the neurons in the brain function and to provide the basis for efforts to create artificial intelligence. The preliminary theoretical base for contemporary neural networks was independently proposed by Alexander Bain (1873) and William James (1890). In their work, both thoughts and body activity resulted from interactions among neurons within the brain.

For Bain, every activity led to the firing of a certain set of neurons. When activities were repeated, the connections between those neurons strengthened. According to his theory, this repetition was what led to the formation of memory. The general scientific community at the time was skeptical of Bain's[3] theory because it required what appeared to be an inordinate number of neural connections within the brain. It is now apparent that the brain is exceedingly complex and that the same brain "wiring" can handle multiple problems and inputs.

James's theory was similar to Bain's, however, he suggested that memories and actions resulted from electrical currents flowing among the neurons in the brain. His model, by focusing on the flow of electrical currents, did not require individual neural connections for each memory or action.

C. S. Sherrington (1898) conducted experiments to test James's theory. He ran electrical currents down the spinal cords of rats. However, instead of the demonstrating an increase in electrical current as projected by James, Sherrington found that the electrical current strength decreased as the testing continued over time. Importantly, this work led to the discovery of the concept of habituation.

In the late 1940s psychologist Donald Hebb created a hypothesis of learning based on the mechanism of neural plasticity that is now known as Hebbian learning. Hebbian learning is considered to be a 'typical' unsupervised learning rule and its later variants were early models for long term potentiation. These ideas started being applied to computational models in 1948 with Turing's B-type machines.

Farley and Clark (1954) first used computational machines, then called calculators, to simulate a Hebbian network at MIT. Other neural network computational machines were created by Rochester, Holland, Habit, and Duda (1956).

Rosenblatt (1958) created the perceptron, an algorithm for pattern recognition based on a two-layer learning computer network using simple addition and subtraction. With mathematical notation, Rosenblatt also described circuitry not in the basic perceptron, such as the Exclusive Or circuit, a circuit whose mathematical computation could not be processed until after the backpropogation algorithm was created by Werbos (1975).

Neural network research stagnated after the publication of research of machine learning research by Minsky and Papert (1969). They discovered two key issues with the computational machines that processed neural networks. The first issue was that existing neural networks were incapable of processing the Exclusive Or circuit. The second significant issue was that computers were not sophisticated enough to effectively handle the long run time required by large neural networks. Neural network research slowed until computers achieved greater processing power. Also key in later advances was the backpropogation algorithm which effectively solved the Exclusive Or problem (Werbos 1975).

The cognitron (1975) designed by Kunihiko Fukushima was an early multilayered neural network with a training algorithm. The actual structure of the network and the methods used to set the interconnection weights change from one neural strategy to another, each with its advantages and disadvantages. Networks can propagate information in one direction only, or they can bounce back and forth until self-activation at a node occurs and the network settles on a final state. The ability for bi-directional flow of inputs between neurons/nodes was produced with the Hopfield's network (1982), and specialization of these node layers for specific purposes was introduced through the first hybrid network.

The parallel distributed processing of the mid-1980s became popular under the name connectionism. The text by Rummelhart and McClelland (1986) provided a full exposition of the use connectionism in computers to simulate neural processes.

The backpropagation network generated much enthusiasm at the time and there was much controversy about whether such learning could be implemented in the brain or not, partly because a mechanism for reverse signaling was not obvious at the time, but most importantly because there was no plausible source for the 'teaching' or 'target' signal. However, since 2006, several unsupervised learning procedures have been proposed for neural networks with one or more layers, using so-called deep learning algorithms. These algorithms can be used to learn intermediate representations, with or without a target signal, that capture the salient features of the distribution of sensory signals arriving at each layer of the neural network.

The Brain, Neural Networks and Computers

Neural networks, as used in artificial intelligence, have traditionally been viewed as simplified models of neural processing in the brain, even though the relation between this model and brain biological architecture is debated, as little is known about how the brain actually works.

A subject of current research in theoretical neuroscience is the question surrounding the degree of complexity and the properties that individual neural elements should have to reproduce something resembling animal intelligence.

Historically, computers evolved from the von Neumann architecture, which is based on sequential processing and execution of explicit instructions. On the other hand, the origins of neural networks are based on efforts to model information processing in biological systems, which may rely largely on parallel processing as well as implicit instructions based on recognition of patterns of 'sensory' input from external sources. In other words, at its very heart a neural network is a complex statistical processor (as opposed to being tasked to sequentially process and execute).

Neural coding is concerned with how sensory and other information is represented in the brain by neurons. The main goal of studying neural coding is to characterize the relationship between the stimulus and the individual or ensemble neuronal responses and the relationship among electrical activity of the neurons in the ensemble. It is thought that neurons can encode both digital and analog information.

Neural Networks and Artificial Intelligence

A *neural network* (NN), in the case of artificial neurons called *artificial neural network* (ANN) or *simulated neural network* (SNN), is an interconnected group of natural or artificial neurons that uses a mathematical or computational model for information processing based on a connectionistic approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network.

In more practical terms neural networks are non-linear statistical data modeling or decision making tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

However, the paradigm of neural networks - i.e., *implicit*, not *explicit* , learning is stressed - seems more to correspond to some kind of *natural intelligence* than to the traditional symbol-based *Artificial Intelligence*, which would stress, instead, rule-based learning.

Applications of Natural and Artificial Neural Networks

The utility of artificial neural network models lies in the fact that they can be used to infer a function from observations and also to use it. Unsupervised neural networks can also be used to learn representations of the input that capture the salient characteristics of the input distribution, e.g., see the Boltzmann machine (1983), and more recently, deep learning algorithms, which can implicitly learn the distribution function of the observed data. Learning in neural networks is particularly useful in applications where the complexity of the data or task makes the design of such functions by hand impractical.

The tasks to which artificial neural networks are applied tend to fall within the following broad categories:

- Function approximation, or regression analysis, including time series prediction and modeling.
- Classification, including pattern and sequence recognition, novelty detection and sequential decision making.
- Data processing, including filtering, clustering, blind signal separation and compression.

Application areas of ANNs include system identification and control (vehicle control, process control), game-playing and decision making (backgammon, chess, racing), pattern recognition (radar systems, face identification, object recognition, etc.), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications, data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering.

Neural Networks and Neuroscience

Theoretical and computational neuroscience is the field concerned with the theoretical analysis and computational modeling of biological neural systems. Since neural systems are intimately related to cognitive processes and behaviour, the field is closely related to cognitive and behavioural modeling.

The aim of the field is to create models of biological neural systems in order to understand how biological systems work. To gain this understanding, neuroscientists strive to make a link between observed biological processes (data), biologically plausible mechanisms for neural processing and learning (biological neural network models) and theory (statistical learning theory and information theory).

Types of Models

Many models are used in the field, each defined at a different level of abstraction and trying to model different aspects of neural systems. They range from models of the short-term behaviour of individual neurons, through models of how the dynamics of neural circuitry arise from interactions between individual neurons, to models of how behaviour can arise from abstract neural modules that represent complete subsystems. These include models of the long-term and short-term plasticity of neural systems and its relation to learning and memory, from the individual neuron to the system level.

Neural networks and logical reasoning systems, because they both rely on Turing models of computation, are equivalent in the sense that whatever is computable in one framework must also be computable in the other. How to establish the equivalence in each particular case is however a non-trivial and interesting issue, because problems that are addressed in a simple manner in one approach turn out to be intractable in the other and vice-versa. For example, combinatorial optimization problems are efficiently handled by neural networks, whereas logical systems succumb to combinatorial explosion.
Conversely, for highly structured problems where appropriate heuristics are known, it is simpler and faster to use a logical reasoning system, the corresponding neural network system taking a long time to tune up, because rule-based information is in general not so easy to build in the network architecture. In an attempt to relate these two computational models, some authors have tried to identify the nature of the rule-based information that may be extracted from the networks as well as the network structures that correspond to particular logical operations. On the other hand the use of hybrid systems has been proposed to solve complex problems. Extraction of rules from networks is an issue of practical importance in the construction of expert systems from example-trained networks. On the other hand when some prior rule-based information is known about a problem but nevertheless a network implementation seems appropriate, it would be useful to have some simple rules to implement the symbolic information on the architecture of the network. Neural networks take advantage of parallel VLSI hardware implementations, which largely improve the processing speed and it is not so clear how to take advantage of similar implementations J. Martins & R. V. Mendes for symbolic processing. Therefore even for problems that are naturally stated in logic terms, it might be useful to have a translation dictionary for hardware implementation purposes. The establishment of a concise way to go back and forth between symbolic and network

formulations is also welcome in learning-oriented systems and in the design of multicomputer networks. Different architectures make different types of learning easier or harder to design and in multicomputer networks it is essential for algorithms and architectures to fit together as well as possible. Finally and independently of the practical issues, the establishing of a compact translation dictionary between the two paradigms might be a useful step in the development of a unified language for cognitive processes. Doyne Farmer, in a classical paper , has shown that there is a common framework in which neural networks, classifier systems, immune networks and autocatalytic reaction networks, may be treated in a unified way. The general model, to which all these models are mapped, provides then an extension of the scope of neural network models . In a similar way, providing a bridge between neural networks and logical systems (a new entry to the Rosetta stone") might suggest new algorithms and applications in both domains. Past attempts at establishing a logic-networks dictionary concern either the question of the architecture required to represent some types of logical operations or the refining of the numerical part of the knowledge base. In some cases, an extension of the usual connectionist framework is required and a full correspondence is not established. Here we describe an attempt to establish a correspondence involving all the basic elements that are present in logical systems (knowledge base; rules; inference; recursion and handling of queries). In a neural network one has a (distributed) memory on the connection strengths (synapses), a learning dynamics on the synapses and an activation dynamics of the node intensities. In a logical reasoning system there is some set of ground facts about the objects in the domain, a set of rules which are potential knowledge concerning relations between the objects and an inference mechanism (backward or forward chaining) allowing for the extraction of further information and the answering of queries. It has been argued that trying to isolate, in a network, the structures that correspond to specific logical statements or operations is a waste of time because everything in a network (memory, rules and inference) is distributed everywhere and forever inseparable. This may well be true for some architecture and some classes of concepts. However even the identification of the modular structures that correspond to the logic elements will be a useful step. For example, we conclude here that a natural network representation of an atomic proposition is a node with nth-order synapses. Because of the universal approximation properties of neural networks, that same proposition might as well be represented by first-order synapses, the proposition corresponding then to a sub-module of several neurons. However the identification of the kind of minimal sub-module that corresponds to a specific logical element is already a useful step. On the other hand, establishment of rules, inference mechanisms and queries are related to dynamical laws in the corresponding network, thus providing a dynamical systems view of the logical reasoning process.

The Network Representation of a Logic System

For definiteness, the scope of the logical system that is considered is a subset of the Prolog logic programming language. Namely the logic system is specified by a set of Horn clauses which are constructed from terms which are either constants or variables or structures. A constant represents some concrete object in the domain of the problem. It is represented in the logical system by an indecomposable elementary symbol (an atom). Structures are restricted to be atomic propositions of the general form called functor. The functor is an atom and the parameter list is any list of atoms, variables or other atomic propositions. Finally a variable is an entity that can at any time be bound to any atom (constant or functor). Small letters will be used for the atoms and capitals for the variables. The first step is to find a network representation of the basic entities of the logical system. Each atom will correspond to a node.

MACHINE TRANSLATION AND NEURAL NETWORKS

Example-Based (EB) Machine Translation (MT) systems have recently led to successful limited domain applications [Brown et al., 1993, Vogel et al., 1996]. *Sub sequential Transducers* (SSTs) [Berstel, 1979], which are also within the EB framework and are a class of Finite-State Models (FSMs), have become an interesting approach to both Language Understanding (Castellanos et al., 1993) (considered as a particular case of translation), and MT (Castellanos et al., 1994, Oncina et al., 1994, Vilar et al., 1995). The appeal of transducer learning is in part due to the fact that very accurate translation models can be obtained when enough training examples are presented (Oncina et al., 1993). However, the amount of data required is sometimes quite large. Moreover, *Neural Networks* (NNs), so-called *Connectionist Models,* can also be considered as an encouraging approach to EB MT. On that score, NNs have also shown empirical success
dealing with Language Understanding tasks (Stolcke, 1990). However, only a few connectionist MT systems have been developed in the literature. PARSEC (Jain, 1991), which was used in the JANUS project (Waibel et al., 1991), follows this approach. Another effective and more simple EB connectionist translator for text-to-text applications has been recently introduced in (Castaño & Casacuberta, 1997). The preliminary results presented in that paper indicated that translations from the source to the target language can be automatically and successfully approached. In addition, these findings suggested that small corpora are required to train the neural models. FSMs had also been previously applied to the same task considered in these pilot experiments. However, SSTs were not trained and tested on the same data as those employed for NNs. Consequently, both connectionist and transduction models could not be precisely compared. Appropriate experiments which compare both MT methodologies are carried out and are discussed in this paper. The paper is organized as follows: First, the MT task in which NNs and SSTs are compared is described. Section 3 presents the neural architecture employed, as well as the procedure used to train the net. Section 4 briefly describes some concepts related to SSTs and the different experiments which were carried out with these models. In Section 5, the experimental results achieved with the two techniques are reported.

NEURAL NETWORK: PROBLEMS AND OBJECTIONS

A common criticism of neural networks, particularly in robotics, is that they require a large diversity of training for real-world operation. This is not surprising, since any learning machine needs sufficient representative examples in order to capture the underlying structure that allows it to generalize to new cases. Dean Pomerleau, in his research presented in the paper "Knowledge-based Training of Artificial Neural Networks for Autonomous Robot Driving," uses a neural network to train a robotic vehicle to drive on multiple types of roads (single lane, multi-lane, dirt, etc.). A large amount of his research is devoted to (1) extrapolating multiple training scenarios from a single training experience, and (2) preserving past training diversity so that the system does not become over-trained (if, for example, it is presented with a series of right turns – it should not learn to always turn right). These issues are common in neural networks that must decide from amongst a wide variety of responses, but can be dealt with in several ways, for example by randomly shuffling the training examples, by using a numerical optimization algorithm that does not take too large steps when changing the network connections following an example, or by grouping examples in so-called mini-batches.

A. K. Dewdney, a former Scientific American columnist, wrote in 1997, "Although neural nets do solve a few toy problems, their powers of computation are so limited that I am surprised anyone takes them seriously as a general problem-solving tool." (Dewdney, p. 82)

Arguments for Dewdney's position are that to implement large and effective software neural networks, much processing and storage resources need to be committed. While the brain has hardware tailored to the task of processing signals through a graph of neurons, simulating even a most simplified form on Von Neumann technology may compel a NN designer to fill many millions of database rows for its connections - which can lead to abusive RAM and HD necessities. Furthermore, the designer of NN systems will often need to simulate the transmission of signals through many of these connections and their associated neurons - which must often be matched with incredible amounts of CPU processing power and time. While neural networks often yield effective programs, they too often do so at the cost of time and money efficiency.

CONCLUSION

Regardless of the problems and shortcomings in the field, which are the result of its being an interdisciplinary discipline, it can be claimed that it has gained much attention in the academic arena. It is especially interesting in that it has applications in artificial intelligence, machine translation, neuroscience, data processing, etc. Much of the criticism centers around the fact that developing studies and projects relating to neural networks in time-consuming. It is not surprising as long as the natural neural networks are comparatively, if not more, as complicated. Neural networks are in the dock not only because they have been hyped to high heaven, (what hasn't?) but also because you could create a successful net without understanding how it worked: the bunch of numbers that captures its behavior would in all probability be "an opaque, unreadable table...valueless as a scientific resource". In spite of his emphatic declaration that science is not technology, Dewdney seems here to pillory neural nets as bad science when most of those devising them are just trying to be good engineers. An unreadable table that a useful machine could read would still be well worth having.

In response to this kind of criticism, one should note that although it is true that analyzing what has been learned by an artificial neural network is difficult, it is much easier to do so than to analyze what has been learned by a biological neural network. Furthermore, researchers involved in exploring learning algorithms for neural networks are gradually uncovering generic principles which allow a learning machine to be successful.

REFERENCES

Bain (1873) *Mind and Body: The Theories of Their Relation*. New York: D. Appleton and Company.

Berstel,(1979)J. Berstel: *Transductions and Context-Free Languages,* Teubner. (1979)

Brown et al., (1993) P.P.Brown, S.A. Della Pietra, V.J. Della Prieta and R.L. Merser: "The Mathematics of Statistical Machine Translation: Parameter Estimation". *Computational Linguistics,* 19, pp. 263-311. (1993) [Castellanos et al., 1993] A.

Brown EN, Kass RE, Mitra PP. (2004). "Multiple neural spike train data analysis: state-of-the-art and future challenges". *Nature Neuroscience* **7** (5): 456–61. doi:10.1038/nn1228. PMID 15114358.

Castellanos, E. Vidal and J. Oncina: (1993) "Language Understanding and Subsequential Transducer Learning". *Proceedings of the 1st International Colloquium on Grammatical Inference (ICGI-93),* pp.11/1-11/10, Colchester, UK.

Castellanos et al., (1994) A. Castellanos, I. Galiano and E. Vidal: "Application of Ostia to Machine Translation Tasks". In *Lecture Notes in Computer Science-Lecture Notes in Artificial Intelligence: Grammatical Inference and Applications,* vol 862, pp.93-105, R.C.Carrasco and J. Oncina (Eds), Springer-Verlag, Berlin.

Castaño & Casacuberta, (1997) M.A. Castano and F. Casacuberta: "A Connectionist Approach to Machine translation". *To appear in Proceedings of the 5th European Conference on Speech Communication and Technology (EUROSPEECH-97),* Rhodes, Greece.

Ebb, Donald (1949). *The Organization of Behavior*. New York: Wiley.

Farley, B; W.A. Clark (1954). "Simulation of Self-Organizing Systems by Digital Computer". *IRE Transactions on Information Theory* 4: 76–84.

Fukushima, Kunihiko (1975). "Cognitron: A self-organizing multilayered neural network". *Biological Cybernetics* 20 (3-4): 121–136. doi:10.1007/BF00342633. PMID 1203338.

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

James (1890). *The Principles of Psychology*. New York: H. Holt and Company.

J. J. Hopfield Neural networks and physical systems with emergent collective computational abilities. Proc. NatL Acad. Sci. USA Vol. 79, pp. 2554-2558, April 1982 Biophysics Arbib, p.666

Minsky, M.; S. Papert (1969). *An Introduction to Computational Geometry*. MIT Press. ISBN 0262630222.

Oncina et al., (1993) J. Oncina, P. Garcia, and E. Vidal: "Learning Sub sequential Transducers for Pattern Recognition Interpretation Tasks". *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 15, pp. 448-458. (1993)

Rochester, N.; J.H. Holland, L.H. Habit, and W.L. Duda (1956). "Tests on a cell assembly theory of the action of the brain, using a large digital computer". *IRE Transactions on Information Theory* 2: 80–93.

Rummelhart, D.E; James McClelland (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge: MIT Press.

Rosenblatt, F. (1958). "The Perceptron: A Probalistic Model For Information Storage And Organization In The Brain". *Psychological Review* 65: 386–408.

Sherrington, C.S.. "Experiments in Examination of the Peripheral Distribution of the Fibers of the

Posterior Roots of Some Spinal Nerves". *Proceedings of the Royal Society of London* 190: 45–186.

ISSN: 2223-9553
Volume 1, Issue 3, November 2011

Stolcke, 1990 A. Stolcke: "Learning Feature-based Semantics with Simple Recurrent Networks". Technical Report no. TR-90-015, International Computer Science Institute, Berkeley, California. (1990)

Waibel et al., 1991 A. Waibel, A.N. Jain, A.E. Mcnair, H. Saito, A.G. Hauptmann and J. Tebelskis: "Janus: A Speech-to-Speech Translation System using Connectionist and Symbolic Processing Strategies". *Proceedings of the 1991 International Conference on Acoustics, Speech and Signal Processing (ICASSP-91),* pp. 793-796, Toronto, Canada. (1991)

Werbos, P.J. (1975). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*.