Biologically Inspired Cognitive Architectures (2014) 9, 71–81

Available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/bica

RESEARCH ARTICLE

Symbolic neural networks for cognitive capacities

Tsvi Achler

ITOP Corporation, 3168 South Court, Palo Alto, CA 94306, United States

Received 19 April 2014; received in revised form 1 July 2014; accepted 3 July 2014

KEYWORDS
Symbolic neural networks;
Auto-dissociative networks;
Iterative neural networks;
Regulatory feedback;
Pattern recognition with symbolic networks;
Gradient descent during recognition

Abstract
Pattern recognition (recognizing a pattern from inputs) and recall (describing or predicting the inputs associated with a recognizable pattern) are essential for neural-symbolic processing and cognitive capacities. Without them the brain cannot interact with the world e.g.: form internal representations and recall memory upon which it can perform logic and reason. Neural networks are efficient, biologically plausible algorithms that can perform large scale recognition. However, most neural network models of recognition perform recognition but not recall: they are sub-symbolic. It remains difficult to connect models of recognition with models of logic and emulate fundamental brain functions, because of the symbolic recall limitation.
I introduce a completely symbolic neural network that is similar in function to standard feedforward neural networks but uses feedforward-feedback connections similar to auto-associative networks. However, unlike auto-associative networks, the symmetrical feedback connections are inhibitory not excitatory. Initially it may seem counterintuitive that recognition can even occur because the top-down connections are self-inhibitory. The self-inhibitory configuration is used to implement a gradient descent mechanism that functions *during recognition* not learning. The purpose of the gradient-descent is not to learn weights (weights are still learned during learning) but to find neuron activation. The advantage of this approach is the weights can now be symbolic (representing prototypes of expected patterns) allowing recall within neural networks. Moreover, considering the costs of both learning and recognition, this approach may be more efficient than feedforward recognition. I show that this model mathematically emulates standard feedforward model equations in single layer networks without hidden units. Comparisons that include more layers are planned in the future.

Tel.: +1 650 493 4867.
E-mail address: achler@gmail.com.

http://dx.doi.org/10.1016/j.bica.2014.07.001
2212-683X/

Introduction

The neural networks responsible for pattern recognition determine the form of information and connection weights required for recognition, memory, and further processing. However, certain forms may be more optimal for certain tasks than others. Popular feedforward neural networks are efficient recognition algorithms however are not optimal for recall: describing or predicting the inputs associated with a recognizable pattern. Thus feedforward networks are considered sub-symbolic (e.g. Fodor & Pylyshyn, 1988; Sun, 2002). In this work we will discuss and show how it possible to change the form of information and achieve both recognition and recall with symbolic neural networks. Moreover, our current simulations, limited single layer, show symbolic networks may be even more efficient for recognition than feedforward networks.

First let us define the meaning of symbolic. A symbolic connection can be thought of as the relationship between an input and the output node that does not depend on any other inputs and outputs. For example, suppose there is an input node that represents legs and an output node that represents zebra. The symbolic relationship for zebra and legs is that it has 4 legs. In a symbolic network, the value 4 can be the weight that represents the connection strength between legs and zebra. This represents a prototype description of a zebra (node) where it does not matter whether there exist other animals (other nodes) that also use the same input (and may also have 4 legs such as dogs, cats, rats, and giraffes, or may have none, 2, 6, 8 or any other number legs). The symbolic weight between zebra and legs remain the same regardless of other nodes.

However, it is important for recognition that certain information will be unique. For example, if the only information available at the input is 4 legs, it would not be possible to recognize a zebra from any other animal with 4 legs, regardless of the type of network. Unique information (such as distinctive stripes) is needed to properly perform recognition regardless whether the network is feedforward or symbolic.

A major difference between feedforward networks and the proposed symbolic network is how uniqueness is evaluated and encoded. Feedforward networks solve recognition with one multiplication per layer. In order to recognize correctly within one multiplication, they encode uniqueness information in the weights (more than just symbolic information). To incorporate the uniqueness information into weights, feedforward methods use a gradient descent error-driven mechanism during learning, which rehearses the patterns of the training set in a uniform identical and independently distributed (*iid*) fashion to incorporate how relevant (unique) each input is. Thus in feedforward networks, weights also depend on input and output nodes other than the immediate input and output node that they connect. This is why error-driven learning is sometimes referred to as global learning. Returning to our example, in feedforward networks, the weights between legs and zebra will be dependent on whether other animals exist and whether other animals have legs. If no other animals have legs, the final weight between legs and zebra will be different than if all other animals have legs.

By definition, symbolic weights cannot incorporate whether information is unique, because uniqueness depends on whether other nodes use that information (and by definition symbolic information must be independent of other outputs).

However, it is necessary to determine uniqueness for recognition, thus the proposed model uses a gradient descent mechanism during recognition to determine how a unique piece of information is based on the other nodes that are currently active (e.g. the other animals that are also being considered) and modulates the relevance of the input (e.g. stripes) accordingly. In effect the proposed model is doing what feedforward learning algorithms do during learning (modulate weights based on uniqueness) but during recognition (modulating inputs based on uniqueness). The gradient descent of the proposed model does not learn weights (weights do not change) but determines uniqueness.

The symbolic model does not require a gradient descent mechanism during learning and subsequently its learning is much easier. Moreover, during recognition the current test pattern is available (while not available during learning). This translates into better efficiency. Feedforward learning algorithms have to perform a gradient descent using the all the patterns in the training set (in *iid* form) and determine overall how relevant each input is. In contrast, the proposed model only performs a gradient descent using the current input pattern. This requires much less gradient descent iterations, iteration times, and allows a simpler, quicker and recallable form of learning.

Clearly our zebra example is a gross oversimplification; there are many types of legs and features that go into legs and so on. However, this generalizes to more complex networks. Networks that can be described in a feedforward manner (including multilayer networks) can also be described in a symbolic manner. Even if nodes are hidden, they can still have symbolic weights. The difference is in our symbolic network uniqueness information is not incorporated into weights. Our analysis in this paper is currently limited to single layer networks, but the symbolic properties are generalizable to hidden nodes as well. The differences between the networks will hopefully become clearer in the examples.

Let us more formally establish the standard notation for neural networks and pattern recognition on which we shall build, and then review and compare neural network models. Let vector Y represent the activity of a set of labeled nodes that may be called output neurons, or classes and individually written as $Y = (y_1, y_2, y_3, \ldots, y_h)^T$. Supervised neurons identify patterns using labels or guide behavior. ''Supervised'' tasks are explicitly naming patterns or behaviorally responding to the environment (without explicit naming). The definition of supervised includes any representation that is behaviorally relevant or action-able. This definition may be a broader than other authors' definitions. For example labels can include: y's representing ''*dog*, *cat*'', predator, prey, or food, mate, danger, and so on. Without labeled associations the brain cannot interact with the world, for example find: food, mates, and hazards. All of these must be correctly labeled by the organism (whether they are explicitly named or not) and responded with the appropriate behavior. Thus supervised recognition is a broad and essential foundation