An Adaptive Resonance Theory-based Neural Network Capable of Learning via Representational Redescription

Guszti Bartfai

Analogical and Neural Computing Systems Laboratory
Computer and Automation Research Institute of the Hungarian Academy of Sciences
H-1111 Budapest, Kende u. 13-17, Hungary.
On leave from Computer Science, Victoria University, P.O. Box 600, Wellington, New Zealand.
E-mail: guszti@mcs.vuw.ac.nz

Abstract

This paper introduces a neural network architecture called R2MAP, which builds upon the Representational Redescription (RR) hypothesis in cognitive science and Adaptive Resonance Theory (ART) neural networks. The R2MAP network learns to classify arbitrary sequences of input patterns using a re-iterative process whereby knowledge that gets embedded in the network via ARTMAP-style error-driven learning is redescribed and becomes available to it for further learning. The knowledge redescription phase is triggered when the perceived level of difficulty of the given task — which is proportional to the number of input categories developed in the previous phase — exceeds a certain threshold, and is achieved through the dynamic creation of new features that better distinguish between output classes. This way the R2MAP network is capable of learning complex, relational input-output dependencies that cannot be represented efficiently using solely features extracted through ordinary learning of statistical relationships. A simple proof-of-concept example is presented to illustrate the main ideas. Some related work is also discussed.

1. Introduction

Artificial neural networks are well known to be useful alternative tools to traditional AI methods mainly due to their inherent parallelism, adaptability and ability to handle noisy and incomplete data. Despite their impressive range of successful real-world applications, neural networks are also known to have several drawbacks, most notably long training time and difficulty of understanding the effect of parameter changes as well as knowledge encoded in trained networks (which is often buried in the connection weights). They typically have difficulty coping with complex and non-stationary environments (see, for example, "catastrophic forgetting" [8]), something that humans can do naturally. This has motivated researchers to look for ideas by studying living organisms, the human brain and cognitive system in particular, in order to improve the capabilities of neural network models. The present paper proposes a neural network architecture, R2MAP, whose development was motivated by two theories, each of which has proven to be a significant contribution in its own field. One is the Representational Redescription (RR) hypothesis in cognitive science [11], and the other is the Adaptive Resonance Theory (ART) neural network [2]. Besides being successful in explaining many cognitive and neural phenomena, they each pose further research questions: how to model RR in a connectionist framework, and how to make ART networks capable of coping with difficult learning problems more effectively. The R2MAP model presented here offers a contribution to both theories by creating a link between them and thereby pointing out their relevance to each other. The two theories are described first (RR in section 2, ART in section 3) at a level sufficient for understanding the motivation for this work as well as the key ideas of the proposed neural network model, which is introduced in section 4. Section 5 presents a simple proof-of-concept example that illustrates the main ideas underlying the operation of R2MAP. Some related work and further interesting issues are discussed in section 6, while conclusions are drawn in section 7.

2. Representational Redescription

The Representational Redescription hypothesis [11] emphasises a developmental perspective on cognitive science. Central to the RR theory is the assumption of an internally-driven and domain-general re-iterative process whereby knowledge that is stored initially in the mind — as a re-
result of mainly error-driven learning — gets re-described into a new representational form, and becomes accessible to the mind for further cognitive manipulation (including re-description!). The theory has been tested and its suitability demonstrated on a range of cognitive tasks performed by children at various stages of development [11].

The RR theory distinguishes the following three phases of the re-iterative process:

**Phase 1** External information is processed, and learned knowledge is added to the existing stock without affecting it. The learner reaches “behavioural mastery” in the given domain via mainly failure-driven learning.

**Phase 2** Attention is focussed on internal representation while external features are disregarded. Behavioural success decreases temporarily as new relationships are being explored.

**Phase 3** The newly discovered internal representation and external data are reconciled. Performance level reaches its maximum again.

The theory also distinguishes four formats (or levels) of knowledge representation as follows:

**Level I (Implicit)** At the end of phase 1, learned knowledge gets stored in the system as a whole, which can be repeated/processed, but the parts are not available. Typical classical connectionist models stop at this point.

**Level E1 (Explicit-1)** Via a repeated process of redesicription, condensed (abstracted) representations are explicitly defined. These are still not available to conscious access and verbal report, but are reduced descriptions that lose many of the details of the procedurally-encoded information.

**Level E2 (Explicit-2)** At this level, it is hypothesised, representations are available to conscious access but not to verbal report (which is possible only at level E3).

**Level E3 (Explicit-3)** At this level, knowledge is recoded into a cross-system code. This common format is hypothesised to be close enough to natural language for easy translation into stable, communicable form and therefore becomes available to verbal report.

There are several possible relationships of these levels (see p24 in [11]). Here we shall concentrate on the distinction between the implicit (I) and explicit (E1-E2-E3) levels.

Most current connectionist models are typically good at implementing phase-1 error-driven learning to achieve “behavioural mastery” (i.e., minimising an error function, for example, by gradient descent like back-propagation nets) in a given problem. Representation of learned knowledge is largely implicit and hardly accessible (through hidden layer activations and connection weights), thus cannot be used by the network to learn further, or transferred to other networks to be used for solving other problems (see, for example, chapter 8 in [11]).

To elaborate their point, Clark and Karmiloff-Smith define a class of first-order networks [5], which are characterised as follows:

- The network’s learning is purely example-driven. Any change in internal representation reflects a change in the statistical properties of the input-output pairings used in training.

- Knowledge of rules in such networks is always emergent. They do not depend on symbolic expressions that stand for the elements of a rule. Instead they exploit a multitude of subsymbolic representations whose complex interaction produces behaviour which, in central cases, fits the rule.

- They have no self-generated means of analysing their own activity so as to form symbolic representations of their own processing. Their knowledge of rules always remains implicit unless an external theorist intervenes.

They also propose that connectionist systems to model human cognitive development must be able to:

- treat their own representations as objects for further manipulation;

- do so independently of prompting by continued training inputs;

- retain copies of the original networks;

- form new structured representations of their own knowledge which can be manipulated, recombined and accessed by other computational processes.

Although a few neural networks have been shown to exhibit certain characteristics of the RR process (see, e.g., [12]), no connectionist models have yet been proposed that directly address the implementation of the essential features of RR (namely, the three phases and the implicit-explicit distinction). This is partly due to the biological implausability of back-propagation type networks [9], which has been used predominantly in connectionist cognitive modelling. Our proposed model builds upon a neural network that is based on node interactions, network equations and learning laws more closely related to those found in vivo [2], and is able to capture many cognitive phenomena [10] at the same time.
3. Adaptive Resonance Theory (ART) networks

Adaptive Resonance Theory (ART) networks [2] were designed to overcome the stability-plasticity dilemma every learning system has to face. Namely, how can a system be stable against noisy or irrelevant data and yet remain plastic enough to learn novel inputs without affecting already learned knowledge (i.e., to avoid catastrophic forgetting)? Most neural networks can either be set plastic (i.e., during the learning phase), or stable (during recall, when the weights are frozen), but not both. An ART network is capable of fast and stable learning of clusterings of arbitrary sequences of input vectors by self-organisation. Upon presentation of an input pattern, the network attempts to categorise it by first comparing it against the stored prototypes of existing categories. If a category is found with the required matching level (which is controlled by a system parameter called *attentional vigilance*), the network enters into a so-called resonant state (hence the name *adaptive resonance*) and learns by modifying its prototype to keep only the critical features for the selected category. If no existing matching prototype is found, the network considers the input novel, and generates a new category node that learns the current input. An extension of ART called ARTMAP has been proposed [3], which is capable of supervised learning of arbitrary pattern classifications (or mappings). The ARTMAP network is built up of an input and an output ART module (ARTa and ARTb, respectively), and an association (or map) field, which links each of the input categories to one of the output classes. The network learns an input-output pair by first searching for an existing category for the current input vector in ARTa, and then making a prediction — through the map field — for the target class registered at ARTb. If the prediction is correct, normal ART learning (see above) takes place in ARTa. If the prediction is incorrect, a so-called “match-tracking” process is triggered whereby the inter-ART map field reset signal causes the ARTa module to search for another category by raising its vigilance just above the current matching level. The network then either finds another existing (and matching) category, or creates a new one and links it with the desired output class. This way, given enough ARTa (and ARTb) category nodes, the network can learn arbitrary input-output mappings in a way that each output class is covered by one or more input categories. See [3] for details of the architecture, network equations and application.

The ARTMAP network possesses a number of distinctive features that are in direct contrast to classical (back-propagation type) feedforward neural networks. The most important of them are:

- the ability to create new input categories and output classes dynamically;
- localised internal representation (hidden nodes represent input categories);
- continuous (on-line), fast and stable learning (no separate on-line and off-line learning modes);
- ability to learn novel inputs and exceptions fast, and without destroying previously learned information;
- co-existing processes that occur at different time scales (category search, resonance, learning, match-tracking);
- completely autonomous operation (no need for external control or synchronisation);
- modular design with clear interfaces that allows easy extension.

Despite its apparent appeal and advantages over many other types of networks, ARTMAP is still essentially a first-order network (according the above criteria). Most notably, it has no self-generated means of analysing its own activity (i.e., to review or modify its input categories and associations), and its knowledge — however much easier it is to understand and extract it [4] — remains implicit and cannot be transferred to other tasks. Due to many of its useful properties, however, it appears to be overall a better candidate for going beyond “behavioural mastery”.

4. The R2MAP network

This section introduces the R2MAP network, an ART-based model capable of learning difficult problems by applying the essential ideas of the RR hypothesis. The overall architecture of the R2MAP network is shown in figure 1. It resembles the ARTMAP network with the main difference being the use of a new ART-based module (called ARRTa) on the input side instead of ARTa. The ARRT (for “ART with RR”) network is similar to ART except that it is capable of creating new features (FRi) to extend the original input vector (stored in the F1 layer) before categorisation (in the F2 layer). These additional FRi modules carry out a kind of recombination of their input (hence the name FR for “Feature Recombinator”, which is termed after its use in [1]) to provide new features that cannot be extracted through ordinary ART-type learning. In other words, the role of the FR modules is to capture relational dependencies that might be present in a data set to help represent it better for unsupervised competitive learning of statistical dependencies [14]. In the R2MAP model, the creation of these new features is triggered by the perceived difficulty level of the given problem, which is in turn signalled by the creation of an excessive number of input category nodes (in the F2 layer) during ordinary ARTMAP learning. A new
Figure 1. Overall architecture of the R2MAP network. The full connectivity between the F1 and F2 layers in ARRTa is shown as a bus structure for clarity. Category nodes in the F2 layer receive the original inputs plus the output of each FR box. The addition of an input line to a node/module is represented by a diagonal ‘/’. The ‘Map’ field contains the links from input categories to output classes (represented as category nodes in the ARTb module). Network signals like F2 reset, match-tracking and internal saturation are ignored for clarity.

A candidate FR box is accepted for inclusion in the network only if it causes a reduction in the number of input categories (with respect to its current level), which is equivalent to the new FR module producing a better distinguishing feature for one of the output classes. The network’s representation is re-described (at the end of phase 2) in a sense that knowledge that was previously contained in the F2 layer as a collection of input category nodes (with associated links to output classes) becomes available to the F2 layer as a new input feature (capturing another relational aspect of the task), which is available for further redescription.

An outline of the R2MAP learning algorithm is shown in figure 2. It runs in an endless loop implementing the main re-iterative process of RR. The three main phases of this process can be identified in it as follows:

**Phase 1** Ordinary ARTMAP-type error-driven learning of the current task (data set). If the task is “easy”, the network will complete learning during this phase. If the network detects that the task is “difficult”, it will enter phase 2. The problem is considered “difficult” (for the given iteration of phase 1) when a certain limit for the number of input categories is reached, i.e., the F2 layer becomes full. This means that the network needs better distinguishing (non-statistical) features to separate out the output classes with fewer input categories.

**Phase 2** Allocate candidate FR box
Search for “good” FR function
Incorporate new FR

**Phase 3** Re-learn task

Figure 2. Outline of the R2MAP learning algorithm.
Phase 2  In this phase, the network focuses on its internal representation since it could not learn the task well enough with the given feature selection and available resources (i.e., total number of F2 category nodes). It creates a candidate “feature recombinator” (FR) module whose inputs are all the components of the original input vector plus the outputs of previously created FR modules. The output of the module will be added to the current input vector to the F2 layer (which is the original inputs plus the outputs of all existing FR modules). The goal of the FR module is to provide a good discriminatory feature for one of the target classes, which allows the network to learn the same task with a fewer number of input categories. It is important that the FR module does this not by matching its input with a template, or prototype, pattern (which can be done by the F2 layer!), but to match parts of the input vector with other parts in it. This way the FR box can represent “relational” dependencies (which a simple ART module cannot) as opposed to “statistical” ones [6]. The measure of goodness is the correlation of the output of FR with one of the target classes. When a good feature recombination is found, the network extends its original feature set with it, and enters phase 3.

Phase 3 The network with the newly added feature (FR) re-learns the original task (or, more precisely, learning whatever the task is at hand by attending to the external environment again). At the same time, it will also release (reset) the original input categories. As a result, the network will represent the same task with fewer F2 category nodes (and thus with more available free nodes for further phase 1 learning). It is this phase and the previous one where a temporary drop in performance can be observed (with respect to one achieved in phase 1). The reason for this is that the new FR may be good in most cases, but there may still be a need to learn new “exceptions”.

The learning algorithm allows the recursive application of the three phases in the RR process, which can discover ever deeper relationships in a given problem. Note that this model only makes a distinction between the implicit (I) and explicit (E) levels, and does not address the issue of different E levels (E1, E2 and E3). Since the FR modules implement relational operators on its inputs (which is task-independent), they are potentially transferable to other R2MAP networks for solving different problems.

5. An example

The following is a simple proof-of-concept example that illustrates how different FR functions can affect the number of input categories developed in an R2MAP network in different iterations of phase 1. The task is to learn the following four-input Boolean function:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

We can see that this is a “difficult” classification task in that statistical properties reveal very little about the true input/output relationship: the output is 1 IFF exactly two neighbouring features are 1.

For this example, we use the following three feature recombinator modules (ignoring the issue of how these could be found):

- **FRshr1**: “right shift 1”, the output of which is 1 IFF \( A = B \) or \( B = C \) or \( C = D \) = 1.
- **FRshr2**: “right shift 2”, which is 1 IFF \( A = C \) = 1 or \( B = D \) = 1.
- **FRopt**: the optimal feature, which is 1 IFF \( Y = 1 \).

The following table shows the average number of ARRTa categories an R2MAP network created (over 100 runs of phase 1) when trained on the above problem with different features added to the original input vector.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>ARRTa categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-B-C-D</td>
<td>8.59</td>
</tr>
<tr>
<td>A-B-C-D-FRshr1</td>
<td>6.36</td>
</tr>
<tr>
<td>A-B-C-D-FRshr1-FRshr2</td>
<td>6.09</td>
</tr>
<tr>
<td>A-B-C-D-FRopt</td>
<td>3.07</td>
</tr>
<tr>
<td>A-B-C-D-FRopt-FRopt</td>
<td>2.10</td>
</tr>
</tbody>
</table>

The table shows that the additional features “FRshr1” and “FRshr2” help reduce the number of ARRTa categories, but clearly they still lack the discriminatory power (i.e., the network still had to capture quite a few “exceptions” using up additional input categories). The “FRopt” feature causes a dramatic improvement, but it is still above
the theoretical minimum of 2 (since there are two classes here). This level was nearly achieved by duplicating “FRopt”, which essentially gave double weight to this feature.

6. Discussion

The R2MAP model presented here has some relevance to the Cascade-Correlation (CasCor) neural network [7] as well as the idea of “constructive induction” [14] where transformation of the representation space is sought by introducing new features through a process of exploiting relational effects recursively. CasCor and R2MAP are similar in that both learn by creating “hidden nodes” (i.e., FR modules in R2MAP) whose roles are to represent ever deeper relationships in the given task. Even though CasCor can be viewed as a “ redescribing” network [12], the main difference is that, besides being based on a biologically more plausible neural network model (ARTMAP), R2MAP makes the redescriptions process as well as the three phases and two levels (I and E) of RR more explicit. In a recent paper, Schyns et al. have proposed that fixed feature sets are not sufficient to learn complex tasks. In their model, new features are created and incorporated into the system during development “to the extent that they distinguish between object categories” [13]. In the R2MAP model, the role of FR modules (which are features created dynamically) is exactly that.

One of the issues the R2MAP model raises — which has been ignored so far — is searching the space of possible relational operators in order to find “good” distinguishing features. This task, in general, is intractable since the space of such operators is infinite [6]. One, however, may apply a genetic algorithm (GA) with the fitness function being the degree by which the resulting number of ARTa category nodes decreases after re-learning. Another possible approach is to restrict the search space to a finite set of operators; for example, to those that can be implemented in an ARTMAP-like structure whereby parts of an input vector, which are presented at both the input and output sides, are matched using an appropriate wiring of the map field. (The “FRshr1” and “FRshr2” features fall into this type.) The matching level at the map field will then indicate the presence or absence of the given relation (which can then be the output of the FR module).

7. Conclusions

In this paper, we have proposed a new ART-based neural network architecture called R2MAP, which is capable of going beyond “behavioural mastery” on a given task by recursively re-describing its own knowledge via the dynamic creation of “feature recombination” modules. The redescription process follows the three main phases of the RR theory. The model also makes a concrete suggestion for the distinction between the implicit (I) and explicit (E) levels in a network context.

We hope that the R2MAP architecture provides a framework in which further challenging questions about learning difficult problems with neural networks can be posed, the investigation of which may help us understand the capabilities and limits of the models better.

References