PHILOSOPHICAL TRANSACTIONS B

royalsocietypublishing.org/journal/rstb

Research

Cite this article: Vankov II, Bowers JS. 2019 Training neural networks to encode symbols enables combinatorial generalization. *Phil. Trans. R. Soc. B* **375**: 20190309.
http://dx.doi.org/10.1098/rstb.2019.0309

Accepted: 16 September 2019

One contribution of 16 to a theme issue 'Towards mechanistic models of meaning composition'.

Subject Areas:
cognition

Keywords:
symbols, neural networks, combinatorial generalization

Author for correspondence:
Ivan I. Vankov
e-mail: i.i.vankov@cogs.nbu.bg

Electronic supplementary material is available online at https://doi.org/10.6084/m9.figshare.c.4723922.

THE ROYAL SOCIETY PUBLISHING

Training neural networks to encode symbols enables combinatorial generalization

Ivan I. Vankov[1] and Jeffrey S. Bowers[2]

[1]Department of Cognitive Science and Psychology, New Bulgarian University, Sofia, Bulgaria
[2]School of Psychological Science, University of Bristol, Bristol, UK

IIV, 0000-0002-2772-5365

Combinatorial generalization—the ability to understand and produce novel combinations of already familiar elements—is considered to be a core capacity of the human mind and a major challenge to neural network models. A significant body of research suggests that conventional neural networks cannot solve this problem unless they are endowed with mechanisms specifically engineered for the purpose of representing symbols. In this paper, we introduce a novel way of representing symbolic structures in connectionist terms—the vectors approach to representing symbols (VARS), which allows training standard neural architectures to encode symbolic knowledge explicitly at their output layers. In two simulations, we show that neural networks not only can learn to produce VARS representations, but in doing so they achieve combinatorial generalization in their symbolic and non-symbolic output. This adds to other recent work that has shown improved combinatorial generalization under some training conditions, and raises the question of whether specific mechanisms or training routines are needed to support symbolic processing.

This article is part of the theme issue 'Towards mechanistic models of meaning composition'.

1. Introduction

Recent advances in neural network modelling have led to impressive results in fields as diverse as object, face and scene recognition [1], reasoning [2], speech perception [3], machine translation [4], playing computer games [5] and producing art [6]. These successes have relied on a restricted set of tools (e.g. the back propagation learning algorithm or the convolutional network architecture), and principles (all learning and computations take place in links between units), and are consistent with the claim that all forms of cognition rely on a small set of general mechanisms in which minimal innate structure needs to be included. We will call this class of connectionist models currently popular in computer science the 'conventional connectionist framework'.[1] However, there are still fundamental disagreements concerning the limitations of this framework, with some researchers claiming that it cannot account for a range of core cognitive capacities (e.g. [9,10]).

One of the main criticisms of the conventional connectionist approach is that it lacks the ability to represent symbolic structures and is thus unsuitable for modelling tasks requiring symbolic operations (e.g. [11]). It is important to emphasize that the critics of this approach are not claiming that connectionist systems are, in principle, unable to account for symbolic processing, but rather, that models need to be augmented in order to explicitly implement symbolic computation [8]. This can involve introducing new computational mechanisms above and beyond the modification of weights between units, such the synchrony of units firing [12–15] or the reliance of delay lines [16], introducing new inductive biases specifically designed to support symbolic computation, such as graph-based structures [17], hybrid symbolic connectionist units [18]

or introducing new dedicated circuits for the sake of symbolic computation [19,20]. We will refer to models that adopt some or all of these solutions as the 'symbolic connectionist approach' [21]. Central to the symbolic approach is the claim that conventional connectionist models will fail on symbolic reasoning tasks that humans can perform.

Here we show that conventional connectionism systems can support at least some forms of symbolic processing when trained to output symbolic structures. We achieve this by introducing the vector approach to representing symbols (VARS) that encodes symbolic structures of varying complexity as a static numeric vector at the output layer. We show that VARS representations can be learned and that this enables conventional neural networks to achieve combinatorial generalization, a core capacity of symbolic processing. It is important to emphasize that we do not take our findings to rule out symbolic connectionist architectures—there may well be functional and biological pressures that lead the brain to adopt special mechanisms devoted to symbolic computation. However, our findings do undermine one of the motivations for this approach—that dedicated mechanisms are necessary to support tasks that require combinatorial generalization.

The structure of the paper is as follows. We start by briefly reviewing the limited successes of conventional networks in modelling tasks that require combinatorial generalization. We then describe VARS and demonstrate how it can be used to represent symbolic structures in connectionist terms and report two simulation studies showing that neural network models trained to output VARS alongside conventional output representations are able to support an impressive degree of combinatorial generalization in short-term memory and visual reasoning tasks. Importantly, not only do the VARs output representations themselves support combinatorial generalization, but so do the conventional output codes when trained in parallel with the VARS representations. By contrast, the same conventional output codes fail to support combinatorial generalization when the task to output VARS representations is omitted. We argue that our approach does a better job than others existing symbolic and non-symbolic models, and highlights the importance of training conventional neural networks on tasks requiring the explicit representation of symbols.

2. Review of previous studies assessing combinatorial generalization in conventional connectionist architectures

Fodor & Pylyshyn [11] provided an early seminal criticism of the conventional connectionist framework. They argued that a wide range of cognitive capacities rest on the fact that the mind is compositional, with a small set of context-independent elements (e.g. words, units of meaning) used to productively compose more complex representations in limitless ways. On their view, conventional connectionist models that fail to build in mechanisms to explicitly code for the compositional structure of cognition are doomed to fail in combinatorial generalization tasks in which networks are required to produce novel outputs based on novel combinations of familiar symbols. For example, after training a network on the symbols John', 'Mary', 'loves' and the relation 'loves (John, Mary)' a conventional network would not be able to output the relation 'loves (Mary, John)' in response to any query. Combinatorial generalization is at the heart of what Fodor and Pylyshyn call the systematicity and productivity of thought.

In subsequent debates surrounding this issue, a number of authors highlighted the generalization capacities of conventional connectionist models (e.g. [22]) and others have highlighted their limitations [23]. But the conclusions one should draw with regards to Fodor and Pylyshyn's critique are far from obvious for a number of reasons. In some cases, the apparent success of a model is not relevant because the model was not actually tested in a condition that required combinatorial generalization ([24,25]; see below for more details). In other cases, the failure of a model is taken as a virtue as it is thought to mirror limited human performance (e.g. [26]). However, the more basic problem in reaching any strong conclusion is that the failure of a given conventional connectionist model does not provide a demonstration that all such models will fail. Indeed, many of the early failures of conventional connectionist models that have been used to motivate symbolic models were carried out prior to the development of the more versatile modelling tools and much larger datasets used today. It is therefore important to assess whether current conventional networks can support combinatorial generalization in tasks that humans can straightforwardly perform. If these models succeed, then symbolic models cannot be motivated on the basis of computational necessity.

As we summarize next, both earlier and current conventional networks show limited capacity to support combinatorial generalization. We briefly review these limitations in the domain of short-term memory and visual reasoning tasks, the two domains that we test our VARS model in.

(a) Combinatorial generalization in sequence learning tasks

An example of a failure to support combinatorial generalization in an earlier generation of connectionist models was reported by Bowers *et al.* [27,28] in the context of modelling short-term memory. Botvinick & Plaut [24] had developed simple recurrent model of immediate serial recall that could correctly repeat a sequence of six letters approximately 50% of the time (a level of performance that matches human performance). In addition to accounting for a range of empirical phenomena, the authors had emphasized that their model could support widespread generalization in that it could recall sequences of letters it had never been trained on. However, the model was only tested on a limited form of generalization in which each letter was trained in each position within a list. When Bowers *et al.* [27,28] excluded specific letters in specific positions during training (e.g. the letter A was trained in all positions apart from position 1) and then included them in that position at test, the model did poorly, highlighting the model's failure in combinatorial generalization. Similar limitations in related networks were observed by other authors [19,29].

Do these findings pose a challenge to the conventional connectionist approach to explaining human cognition? Botvinick & Plaut [24] defended this approach by arguing that the restricted generalization was a strength based on their claim that humans would also fail under similar training conditions. Alternatively, the conventional approach might be supported by noting that Bowers *et al.* [27,28] only observed limited performance with a simple recurrent network. More recent and powerful recurrent networks that include long

short-term memory (LSTM) circuits [30] might well overcome these. However, both of these lines of defence are difficult to maintain given similar findings have been reported by Lake & Baroni [31] using state-of-the-art recurrent networks trained on tasks that humans can transparently perform. They trained LSTM models to translate a series of commands to a series of actions when the commands were composed of actions (e.g. RUN, WALK) and modifiers of the actions (LEFT, TWICE). The model was unable to perform the correct series of actions if the model had not been trained an all the relevant combinations of actions and modifiers. For instance, if the model had never been trained on LEFT–RUN it could not perform the appropriate action despite being trained on LEFT and RUN in other combinations (e.g. LEFT–WALK, TWICE–RUN).

Still, there are some successes of generalization in sequence learning tasks that appear to require some degree of combinatorial generalization. For example, Gulordava *et al*. [32] trained conventional recurrent neural networks (RNNs) to predict long-distance number agreement in various constructions in four different languages (e.g. predict the verb in: 'The girl the boys like: IS or ARE?'). The model was trained on a corpera of text in each language, and critically, succeeded at near human levels not only when tested on sentences composed of meaningful sentences (where predictions might be based on learned semantic or distributional/frequency-based information rather than abstract syntactic knowledge), but also on nonsense sentences that are grammatical but completely meaningless (motivated by the classic sentence by Chomsky: 'Colorless green ideas sleep furiously'). The authors took these findings to provide tentative support for the claim that RNNs can construct some abstract grammatical representations. Nevertheless, when more challenging forms of combinatorial generalization are required, current state-of-the art conventional connectionist systems continue to struggle, as detailed next.

(b) Recent explorations of generalization using conventional neural networks in the domain of visual relational reasoning

Barrett *et al*. [33] assessed the capacity of various networks to perform abstract reasoning problems analogous to the Raven-style Progressive Matrices, a well-known human IQ test. In this task, a panel of images are presented that vary according to a rule such as 'progression' (e.g. in a panel of 3×3 images in which there is an increasing number of items per image along the first two columns), and the model is trained to select the image that satisfies this rule in order to complete the third column of images (select the target image that has more items). The authors assessed various forms of generalization, including combinatorial generalization (e.g. puzzles in which the progression relation was only encountered when applied to the colour of lines and then tested when the progression was applied to the size of shapes). Several state-of-the-art neural network models were tested and they all performed poorly in the generalization conditions. The authors were able to improve performance somewhat by adding a 'relation network' module specifically designed to improve relational reasoning, and more relevant to the current paper, further still by augmenting the training procedure so that the model outputted 'meta-targets' that specified the relevant dimensions for correctly responding. That is, the model was trained not only to select the correct image but also the reason why the image was the correct answer. Nevertheless, the modified model with the augmented training still performed 'strikingly poorly' in the conditions that most relied on combinatorial generalization.

In a closely related paper, Hill *et al*. [34] tested the ability of conventional neural networks to perform analogical reasoning on a set of visual problems. In this case, the model was presented with a 'source' set of three images that shared a given relation (e.g. the number of items in each image increased by one) and a 'target' set of two images along with a set of images, only one of which shared the same underlying relation (e.g. progression). The task of the model was to select the correct image. Again, the models were tested across a range of conditions, including conditions that required combinatorial generalization, such as (again) applying a familiar relation to new domains. The authors used a conventional convolutional neural network that provided input to a recurrent layer without any special mechanisms for relational reasoning and found that the type of training had a significant impact on the model's performance. When trained in a standard manner in which the model was trained to discriminate the target from foil images that different from one another in various ways the model performed poorly in the combinatorial generalization condition. The important finding, however, was that the model did better in some (but not all) conditions that required combinatorial generation when the training foils were carefully selected so that the model was forced to learn to encode the relevant relations. The fact that performance continued to be poor in conditions requiring extrapolation highlights how difficult generalization outside the training space can be, but at the same time, the improved performance with carefully crafted training foils suggests that conventional connectionist models can be more successful in such tasks than critics often assume. The benefits of manipulating the pressure to learn to encode relations for the sake of combinatorial generalization (e.g. [33,34]) provides the motivation of our approach which we now introduce.

3. The vector approach to representing symbols

The goal of the current paper is to further explore whether neural network architectures without dedicated mechanisms for symbolic processing can solve the combinatorial generalization problem if they are pushed to learn to explicitly represent symbols. To this end, we train neural network models on two separate tasks—a main task that answers a query in a standard format (a 'one hot' encoding of the answer) and a secondary task that outputs a symbolic encoding of the problem at hand. We consider not only whether the model is successful in outputting symbolic representations, but also, whether training on this secondary task leads to success on the main task (the standard one hot encoding output that typically does not support combinatorial generalization).

In order to train on the secondary task we needed a way to represent symbolic structures as numeric vectors that can be used at the output of standard connectionist models. There have already been several proposals for constructing such representations, including tensor products [35] and holographically reduced representations [36] that are discussed (and criticized) by Hummel [37]. However, to our

royalsocietypublishing.org/journal/rstb Phil. Trans. R. Soc. B 375: 20190309

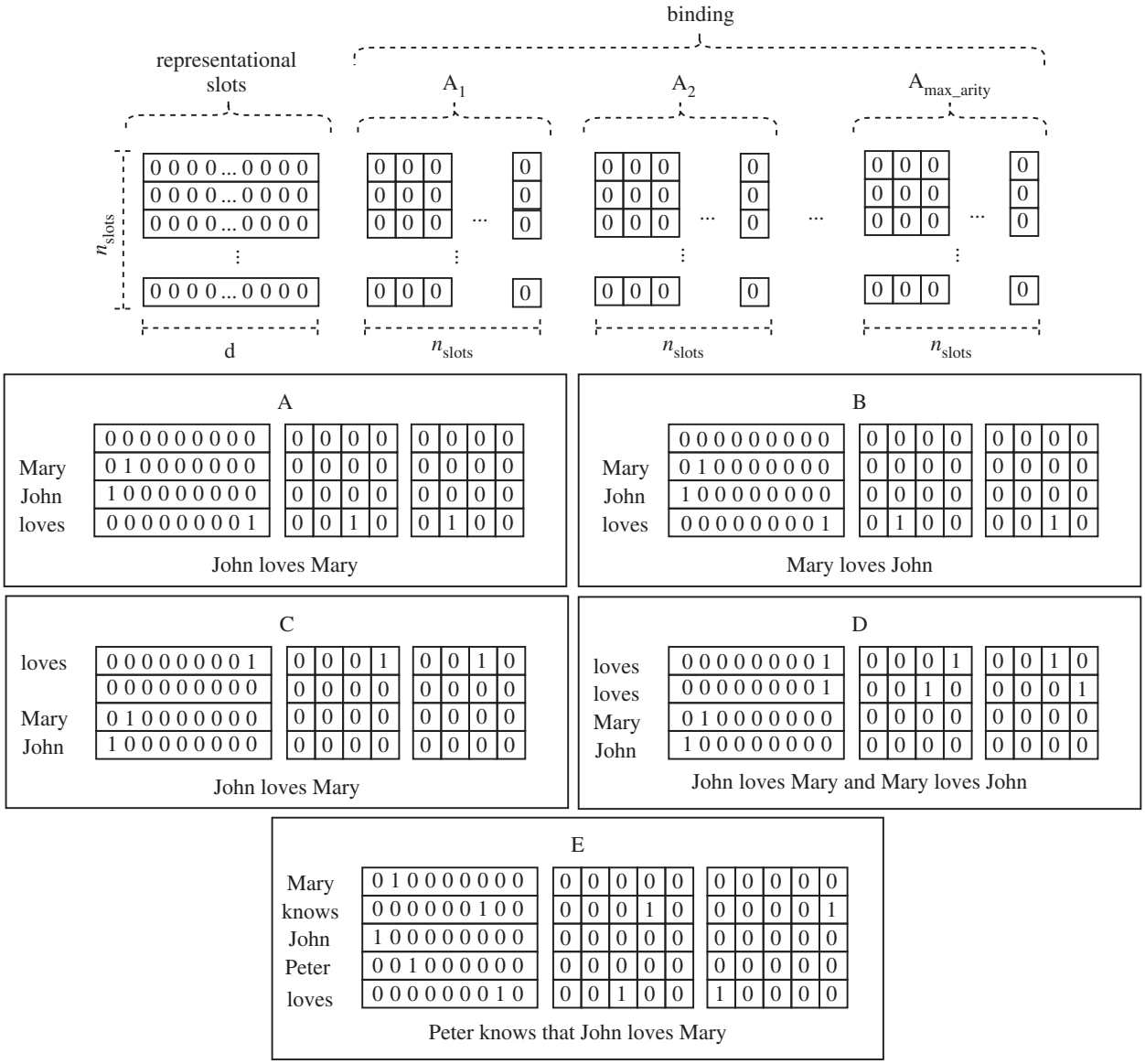

Figure 1. The vector approach to representing symbols (VARS). Top: the structure of VARS representations. The number of symbols that can be represented at the same time is limited by the number of representational slots (n_{slots}). The maximum arity of the symbols which can be represented is also limited. The symbols are represented by vectors with fixed dimension *d*, which can be either localist or distributed. For each argument position, there is a separate $n_{slots} \times n_{slots}$ binding matrix (A_1, A_2, … A_{max_arity}). Five examples of VARS representations are given. Examples A and B demonstrate how the roles of the 'loves' relation are bound to the corresponding fillers. In example A, the first argument of 'loves' is bound to 'John' and the second one is bound to 'Mary'. In example B, the first argument of 'loves' is bound to 'Mary' by activating the second unit in the fourth row of A_1 ('Mary' is represented at the second slot) and the second argument is bound to 'John'. Examples A and C represent the same information, although symbols have been located in different slots. Example D demonstrates how two instances of the same type ('loves) are represented. Example E shows the representation of a second order relation (the second argument of 'knows' is bound to another relation - 'loves'). More examples are available at https://vankov.github.io/combgenvars.

knowledge, there have been no demonstrations so far that conventional neural networks can learn such representations in order to achieve combinatorial generation (but see [38], for a trainable RNN architecture specifically designed for processing tensor products and [39], for a spiking neural network capable of learning holographically reduced representations). We propose a novel way of representing symbolic structures of varying complexity using numeric vectors, which we believe is more suitable for training conventional neural networks on symbolic tasks—VARS (figure 1). Our choice to use VARS does not rule out other approaches—a systematic evaluation of the alternative ways to represent symbols in connectionist terms is outside the scope of the current paper.

The main assumptions of VARS are twofold. First, we assume that the meaning of symbols (i.e. the representation of an item independent of its relation to other items) can be encoded at multiple spatial locations within a VARS vector (in either localist or distributed manner). We will refer to these locations as representational slots. The allocation of symbols per slots is arbitrary which means that a symbol can be represented at any slot without affecting its interpretation (figure 1, examples A and C). The arbitrary allocation of symbols to slots does not imply that neural representations of symbols can move freely around as the contents of memory cells can move in a computer system. Instead, we assume that there exist redundant representations of symbols which are functionally equivalent, i.e. activating any of them encodes the same knowledge (there are multiple ways to represent the same thing). In order to achieve such functional equivalence, a system has to be able to represent each

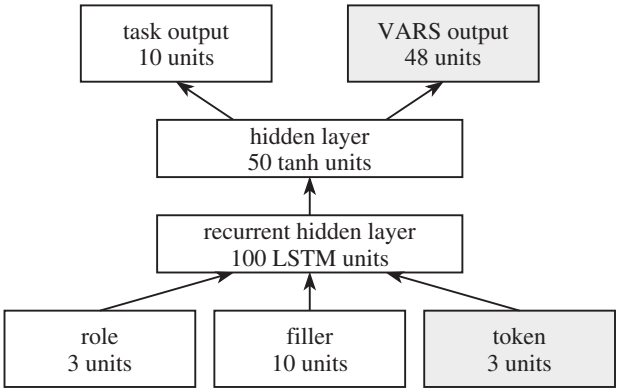

Figure 2. Model architecture in simulation 1. Two grey components were used only when the model was trained to output VARS representations. The randomly generated token was used to assign the current symbol to the corresponding slot in the VARS output.

symbol at each representational slot and keep the contents of the slot independent of the slot identity (i.e. the representation of 'John' should not depend on whether it is activated in slot 1 or slot 2). For models trained to output VARS representations, this can be achieved by learning to encode the symbols at different slots across trials, while making sure that the allocation of symbols to slots is independent of the information which has to be represented. In this way, it is assured that the system will treat the knowledge represented by a symbol independently of the slot which the symbol is allocated to in a given trial. We show how this can be implemented in neural network models in the subsequent simulations.

Being able to encode a symbol at several representational slots makes it possible to represent multiple instances of the same type (see example D in figure 1). Moreover, the address of a representational slot, which we will refer to as a 'token', can be used to bind the argument of a predicate to its corresponding filler. The second main principle of VARS is that binding information is represented explicitly and separately from the representation of the meaning of symbols which makes it possible to bind the arguments of predicate symbol to any other symbol, thus ensuring role-filler independence. For each argument position a, there is a separate $n_{\text{slots}} \times n_{\text{slots}}$ binding matrix A_a, where n_{slots} is the number of representational slots (figure 1, top). Binding the nth argument of the predicate represented at slot i to the symbol at slot j is encoded by activating the unit at the ith row and jth column of A_n. For example, to represent the binary relation 'John loves Mary' (example A in figure 1), one would need at least three representational slots and two binding matrixes—A_1 and A_2, one for each argument position. If 'John' is represented at slot 3, 'Mary' at slot 2 and 'loves' at slot 4, then binding the first argument of 'loves' to 'John' is implemented by activating the third unit (because 'John' is at slot 3) of the fourth row ('loves' is at slot 4) of A_1 (because it represents the binding of the first argument position). Accordingly, activating the second unit of the fourth row of A_2 binds the second argument of 'loves' to 'Mary'.

The complexity of symbolic structures that can be encoded using VARS is constrained by two parameters: the number of addressable representational slots and the maximum arity of the predicate symbols. The ability of VARS to support role-filler independence and its inherent capacity limits make it a plausible account of how symbolic knowledge is represented in the human mind. However, in this work we use VARS only as a computational pressure to train artificial neural networks to encode knowledge symbolically and we therefore refrain from discussing its cognitive plausibility.

Using VARS to represent symbolic knowledge bears resemblance to other approaches which use space in order to enable encoding of multiple instances of the same type and role-filler bindings [19,20,40–42]. However, none of these methods result in fixed size vector representations which can be used to train a conventional neural network architecture. The idea of VARS is also similar to the semantic pointers approach [39].

4. Simulations

Two series of simulations were conducted in order to test the ability of conventional neural networks to account for combinatorial generalization. Our definition of combinatorial generalization follows Kriete *et al.* [19] and relates to the ability to process relations between objects which have not been experienced in the corresponding relational roles before. A critical test of this ability will be to answer questions regarding novel role-filler combinations. For example, if the system is presented with a sentence 'John loves Mary' and asked 'Who is the lover?', it should be able to answer 'John' even if it has never seen John in the role of a lover before and therefore has no existing representation of 'John as a lover'.

In both of the simulations, we first show that conventional neural networks fail the combinatorial generalization tests if no pressure to encode information symbolically is enforced. We then retrain the same models on the same problems, but with the additional task to output a VARS representation of the presented information (analogous to the 'meta-targets' described above by Barrett *et al.* [33]). Thus, the models are trained on two tasks in parallel—the main task and the VARS task. The results show that the models not only learn to encode the correct VARS representations for inputs which constitute novel combinations of familiar items, but in doing so they also pass the combinatorial generalization test on the main tasks. In this way, we show that providing a pressure to output symbolic representations allows conventional network architectures with conventional output coding schemes to support a form of generalization that is often claimed to rely on special purpose symbolic machinery.

(a) Simulation 1

In this simulation, we replicate a combinatorial generalization task in the context of short-term memory as developed by Kriete *et al.* [19]. The model was given a sequence of role-filler pairs and then cued to recall the filler that was associated with a given role (e.g. after encoding a sequence DOG-SUBJECT, EAT-VERB, STEAK-PATIENT the model was cued to recall the filler EAT when probed with the role VERB). In the combinatorial generalization condition, one of the fillers was never paired with a specific role during training (for example, the filler DOG was never presented in the role of SUBJECT, in any sentence). The authors reported that a simple recurrent network failed on this task, whereas a network with an architecture specifically designed to support symbolic processing was successful.

In order to test whether a RNN with a conventional architecture can also achieve combinatorial generalization in this task we contrasted the performance of two models

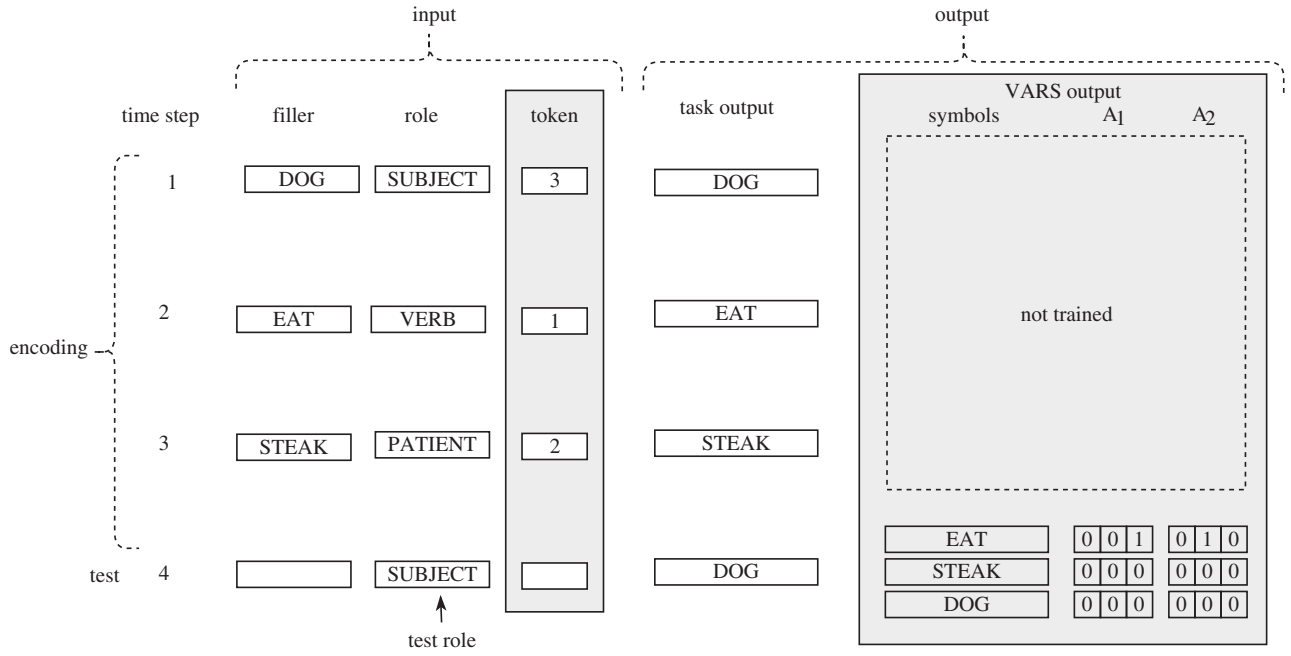

Figure 3. Description of a sample trial in simulation 1. During encoding, the model was presented with a series of three role-filler pairs (SUBJECT - DOG, VERB - EAT, PATIENT - STEAK). At the last (fourth) time step, the model was only presented with a test role (in the example above: SUBJECT) and it had to output the corresponding filler which was associated with it (DOG). When the model was trained to produce VARS representations, the role-filler pairs were accompanied by a random[2] permutation of tokens during encoding which determined the allocation of symbols to representational slots (for example, the fact that DOG was paired to token 3 meant that DOG has to be represented in the third representational slot). In this way, the random token input ensured that each filler has been allocated to each slot during training. The VERB was treated as a binary relation and its arguments (i.e. the relational roles SUBJECT and PATIENT) were bound to the corresponding fillers (in this example, subject to DOG and PATIENT to STEAK). Note, the VARS output was only trained in time step 4, which means that no error was computed during the encoding stage (the 'not trained' area).

Table 1. Combinatorial generalization means accuracy rates in simulations 1 and 2. (In both simulations, the models achieved combinatorial generalization only when trained to explicitly represent symbolic structures. The standard deviation is shown in parentheses. CNN, convolutional neural network.)

		combinatorial generalization accuracy (s.d.)	
	model	main task	VARS task
simulation 1	LSTM	0.30 (0.15)	n.a.
	LSTM + VARS	0.74 (0.18)	0.93 (0.10)
simulation 2	CNN	0.29 (0.05)	n.a.
	pre-trained VGG 16	0.24 (0.05)	n.a.
	CNN + VARS, no binding	0.34 (0.11)	1.00 (0.00)
	CNN + VARS	0.99 (0.01)	0.99 (0.01)

(figure 2). Both of them included the same conventional LSTM architecture and were trained to solve the same task, but only one was also trained to output a VARS representation of the structural information provided during the encoding phase (figure 3). In this way, the model had to solve two tasks in parallel—the main task which required to output the filler corresponding to the requested role and the VARS representation of the three encoded symbols and their relationship. Performance in combinatorial generalization was measured in both tasks: in the main task, this was simply checking whether the correct filler was at the output and in the VARS task we checked whether the slots have been filled and bound correctly. In order to train the model on two tasks, we defined the loss function as a sum of the error on the main task and the VARS task. More details about the simulation are provided in the electronic supplementary material.

The results of the simulation clearly show that a neural network model without dedicated mechanisms for symbolic processing is much better at combinatorial generalization when pressure to represent knowledge symbolically is enforced (table 1). Importantly, the model trained with VARS outputs not only managed to output correct VARS representations of untrained role-filler binding over 90% of the time, but performed over twice as well on the main task (74%) compared to the network without VARS (30%). Performance of the model on the main task in the VARS condition was comparable to the Kriete *et al.* [19] model that included specialized mechanisms to support symbolic computations. This finding suggests that training a neural network model to explicitly output

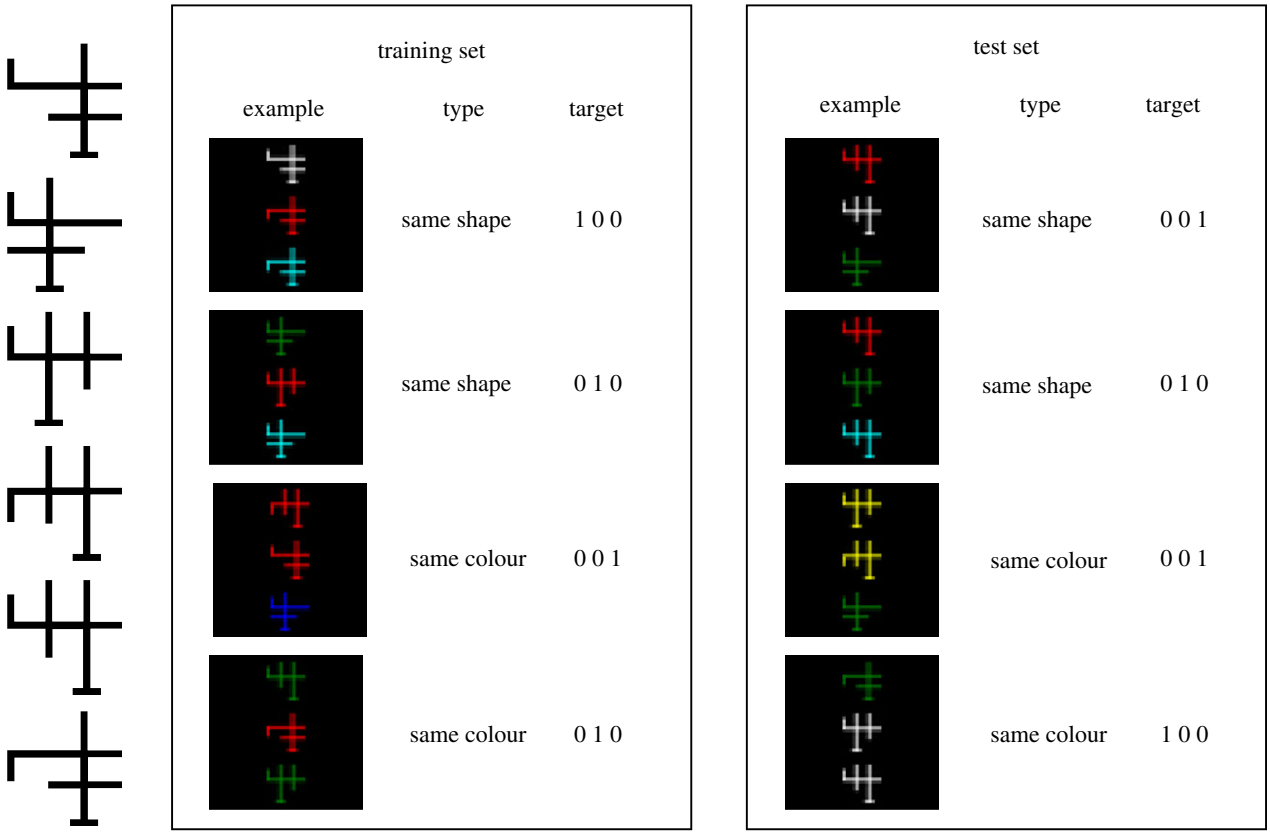

Figure 4. Images used in simulation 2. The six object shapes are presented on the left. The training set consisted of triplet of objects, arranged in vertical order, such that exactly two of the objects shared a single feature (either same colour or same shape). The task of the model was to output the position of the 'odd' object by turning the corresponding bit in the target on. For example, in the first example from the training set presented above, the red and the cyan objects have the same shape, so the model has to output the position of the white object (1 0 0). In all of the test examples, the model had to report the position of the green object, which never happened during the training position. Note that a green object did appear during training, but they were never in the 'odd' role (see the second and the fourth training examples).

symbols qualitatively changes the nature of its internal representations, allowing it to solve problems it otherwise fails on.

(b) Simulation 2

The goal of our second simulation is to confirm and further extend our finding that conventional neural architectures can achieve combinatorial generalization when pressed to encode knowledge symbolically. Here, we assessed whether training a feed-forward convolutional neural network (CNN) on VARS representations improves combinatorial generalization in a visual reasoning task.

In each trial, the model was presented with three objects and had to choose which one of them was 'the odd man out'. Each object had three features: position (top, middle or bottom), shape (one out of six) and colour (one out of six). In each triplet of objects, exactly two of the objects had either the same colour or the same shape, but not both (figure 4). The task of the model was to output the position of the 'odd' object, i.e. the one which shared neither shape nor colour with the other two. In order to test combinatorial generalization in this task, we excluded all examples in which the 'odd' object was green from the training set and tested the model on these examples. In other words, during training the model never had to report the position of a green object. Green objects did appeared in the training set, but never in the 'odd' role. The arbitrary allocation of symbols to representational slots, needed in order to make sure that each symbol can be represented at each slot, was implemented by feeding a random sequence of tokens to the fully connected layers of the network (figures 5 and 6).

In order to assess the ability of neural networks to solve the 'odd man out' problem with or without the pressure to represent knowledge symbolically, we constructed a convolution neural network model displayed in figure 5. To make sure that a failure in this task cannot be attributed to the details of our custom CNN architecture, we also tested VGG 16 [43]—a state-of-the-art model of visual object recognition, which has been pre-trained on the ImageNet dataset. The pressure to encode knowledge symbolically was implemented by making our CNN model output VARS parallel to the main task (figure 6). Just as in simulation 1, the loss function the of model trained on VARS was a sum of the error on the main task and the VARS task (more details about the training procedure are available in the electronic supplementary material) The VARS representation contained the following symbols: *X, Y, Z, different from(X, (Y, Z))*, where X, Y, Z belonged to the set ('top', 'middle' and 'bottom'). In other words, the VARS output represented information such as 'the middle object is different from the top and from the bottom one'. We also trained the CNN model on VARS representations without binding information (i.e. without the 'different-from' symbol) in order to make sure that binding is essential for combinatorial generalization.

The results of the simulation clearly show that the pressure to encode knowledge symbolically enables it to solve the combinatorial generalization problem (table 1). The inability of the model using VARS targets with no

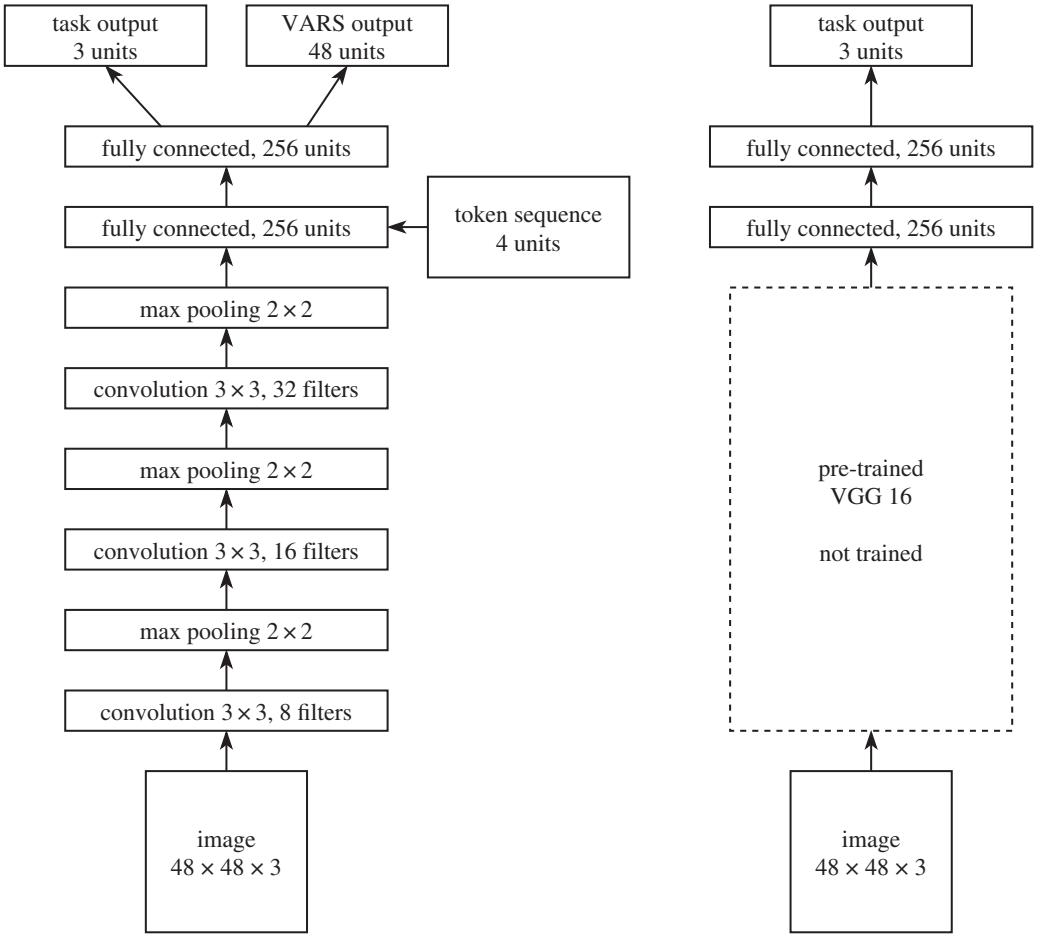

Figure 5. Model architectures in simulation 2. Left: CNN model with three convolutional layers. The components in the grey boxes were only used in the conditions with VARS pressure. Right: pre-trained VGG 16 model with two fully connected layers added at the top. The convolutional layers of VGG 16 were not trained in this simulation. More details about the model are available in the electronic supplementary material.

binding to account for combinatorial generalization suggests that encoding symbolic structure is indeed what drives the model improvement in generalization and not some idiosyncratic effect of forcing the network to represent the objects at different slots in the VARS output. On the other hand, the failure of the pre-trained VGG 16 model suggests that the difficulty of combinatorial generalization in this task cannot be attributed to the relatively limited visual experience our CNN model was exposed to.

5. Discussion

The two simulations above show that combinatorial generalization can be greatly improved in conventional neural networks trained on simple short-term memory and visual reasoning tasks. In order to achieve this, we introduced VARS that encodes symbolic structures of varying complexity as a static numeric vector in the output layer. The output VARS representations led the models to learn internal representations that better supported combinatorial generalization, not only in the VARS output codes, but also in the main task that used standard 'one hot' encoding output units. This suggests that one of the hallmarks of symbolic computation can be performed without adding new special purpose mechanisms or processes that are often claimed to be necessary.

The current findings are consistent with some recent work that also observed improved combinatorial generalization in conventional connectionist models trained in specific ways designed to improve symbolic computations. This includes training on carefully tailored training sets that force networks to induce relational representations in order to succeed [34], or including output codes that explicitly code for the relevant dimensions of the input patterns, the so-called meta-targets [33]. The latter finding is similar to our own, although we believe using VARS is more promising as it makes it possible to train the network on a broader variety of tasks requiring combinatorial generalization, as well as other forms of symbolic processing.

It is important to be clear what we take our contribution to be. We are not claiming that VARS is necessarily the best way to make conventional networks learn internal representations capable of supporting combinatorial generalization. There may be other approaches that are as good or better in achieving this goal. But we are claiming that our findings provide the best evidence to date that conventional neural networks can learn to support combinatorial generalization, and this challenges one of the main motivations for introducing special mechanisms and processes in symbolic networks.

We are also not claiming that our findings rule out symbolic theories of the mind. Despite our findings and other recent research [32–34], it is still far from clear whether mainstream connectionism can address all the concerns that Fodor & Pylyshyn [11] raised so long ago. There could be other cases of combinatorial generalizations which cannot be accounted for by purely connectionist models, even when pressures to encode knowledge symbolically are applied. Moreover, there are other and more sophisticated manifestations of

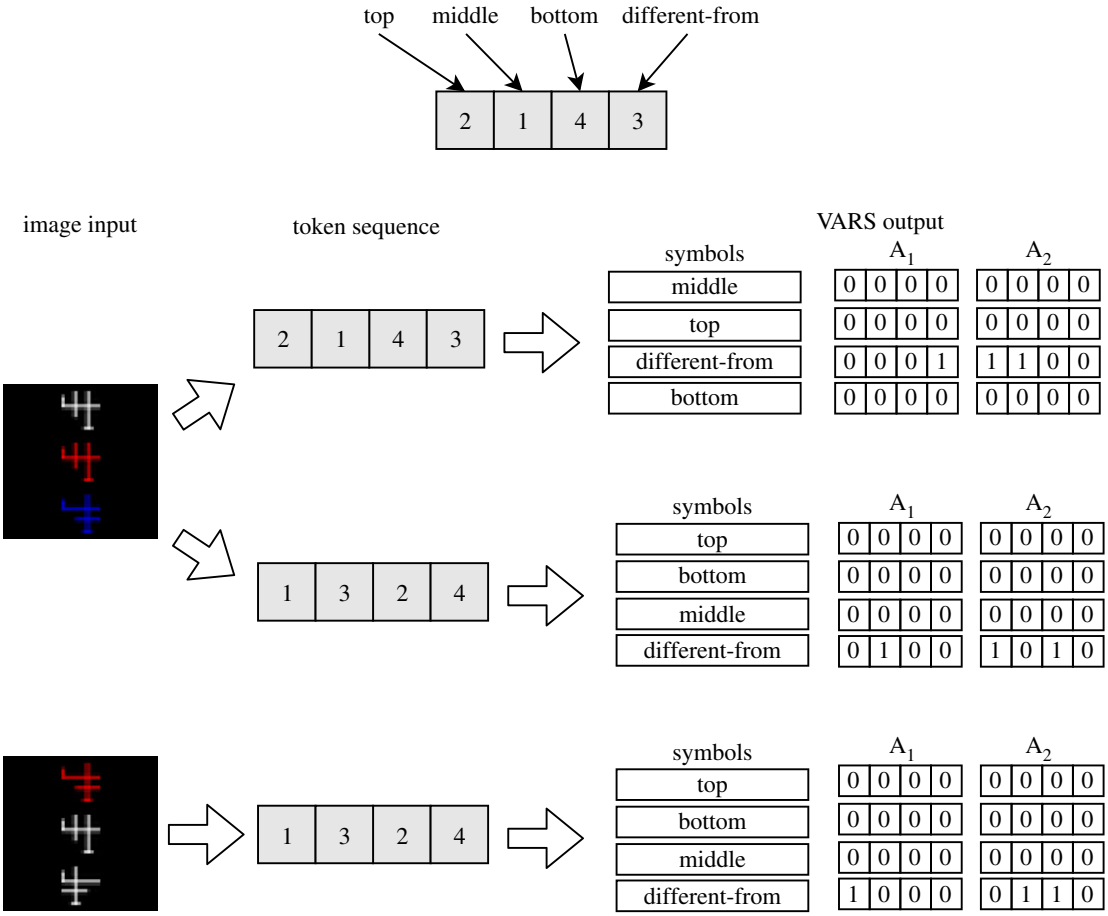

Figure 6. Encoding VARS representations in simulation 2. Top: the random sequence of tokens fed to the fully connected layers of the CNN model determined the allocation of symbols to representational slots. In this example, 'top' should go to the second slot, 'middle' to the first one, 'bottom' to the fourth and 'different-from' to the third. The first two examples in the bottom pane show how the same image can be encoded in different ways, depending on the order of tokens. The difference between the second and the third examples is only in the binding information, although the odd objects are at different positions, which demonstrate the importance of binding information in order to solve the problem.

symbolic thought, such as language and analogy-making, which still constitute a serious challenge to sub-symbolic approaches. We look forward to more research pushing the boundaries of what neural networks can do without implementing dedicated mechanisms for symbolic processing. Even if such models are shown to support human-level generalization, this alone does not rule out symbolic models, but then, the debate will not be about whether conventional neural networks are capable of symbolic computation, but about what kind of models provide the best account of human learning and performance.

Data accessibility. The source code of the simulations is publicly available at https://github.com/vankov/combgenvars.

Authors' contributions. I.V. and J.B. identified the theoretical problem and conceived the project. I.V. designed and conducted the computer simulations. I.V. and J.B. participated in interpreting the results and writing them up. Both authors gave final approval for publication.

Competing interests. We declare we have no competing interests.

Funding. I.V. has been supported by the European Office for Aerospace Research and Development (grant no. FA9550-15-1-0510). J.B. is grateful for support from the ERC Advanced grant no. 741134 (M and M).

Endnotes

[1]An earlier generation of models that adopted this approach has been characterized as 'eliminative connectionism' [7] or 'non-symbolic connectionism' [8].

[2]The results of the simulations reported in this paper do not depend on whether the sequence of tokens is random or fixed. However, the arbitrary allocation of symbols to slots is a general assumption of VARS which may be decisive in other simulations.

References

1. LeCun Y, Bengio Y, Hinton G. 2015 Deep learning. *Nature* **521**, 436–444. (doi:10.1038/nature14539)
2. Silver D *et al.* 2016 Mastering the game of go with deep neural networks and tree search. *Nature* **529**, 484–489. (doi:10.1038/nature16961)
3. Graves A, Mohamed A-R, Hinton G. 2013 Speech recognition with deep recurrent neural networks. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing, May 2013, Vancouver, British Columbia*, pp. 6645–6649. Piscataway, NJ: IEEE. (doi:10.1109/ICASSP.2013.6638947)
4. Wu Y *et al.* 2016 Google's neural machine translation system: bridging the gap between human and machine translation. *arXiv* 1609.08144. (http://arxiv.org/abs/1609.08144)
5. Antonoglou I *et al.* 2015 Human-level control through deep reinforcement learning. *Nature* **518**, 529–533. (doi:10.1038/nature14236)
6. Gatys L, Ecker A., Bethge M. 2015 A neural algorithm of artistic style. *arXiv* 1508.06576. (http://arxiv.org/abs/1508.06576)
7. Pinker S, Prince A. 1988 On language and connectionism: analysis of a parallel distributed

processing model of language acquisition. *Cognition* **28**, 73–193. (doi:10.1016/0010-0277(88)90032-7)

8. Holyoak KJ, Hummel JE. 2000 The proper treatment of symbols in a connectionist architecture. In *Cognitive dynamics: conceptual and representational change in humans and machines* (eds E Dietrich, AB Markman), pp. 229–263. Mahwah, NJ: Lawrence Erlbaum Associates Publishers.
9. Bowers JS. 2017 Parallel distributed processing theory in the age of deep networks. *Trends Cogn. Sci.* **21**, 950–961. (doi:10.1016/j.tics.2017.09.013)
10. Marcus G. 2018 Deep learning: a critical appraisal. *arXiv* 1801.00631. (http://arxiv.org/abs/1801.00631)
11. Fodor JA, Pylyshyn ZW. 1988 Connectionism and cognitive architecture: a critical analysis. *Cognition* **28**, 3–71. (doi:10.1016/0010-0277(88)90031-5)
12. Hummel JE, Biederman I. 1992 Dynamic binding in a neural network for shape recognition. *Psychol. Rev.* **99**, 480. (doi:10.1037/0033-295X.99.3.480)
13. Hummel JE, Holyoak KJ. 1997 Distributed representations of structure: a theory of analogical access and mapping. *Psychol. Rev.* **104**, 427–466. (doi:10.1037/0033-295X.104.3.427)
14. Holyoak K, Hummel J. 2003 A symbolic-connectionist theory of relational inference and generalization. *Psychol. Rev.* **110**, 220–264. (doi:10.1037/0033-295X.110.2.220)
15. Doumas LAA, Hummel JE, Sandhofer CM. 2008 A theory of the discovery and predication of relational concepts. *Psychol. Rev.* **115**, 1–43. (doi:10.1037/0033-295X.115.1.1)
16. Davis CJ. 2010 The spatial coding model of visual word identification. *Psychol. Rev.* **117**, 713–758. (doi:10.1037/a0019738)
17. Battaglia PW *et al*. 2018 Relational inductive biases, deep learning, and graph networks. *arXiv* 1806.01261. (http://arxiv.org/abs/1806.01261)
18. Kokinov B, Petrov AA. 2001 Integrating memory and reasoning in analogy-making: the AMBR model. In *The analogical mind: perspectives from cognitive science* (eds D Gentner, K Holyoak, B Kokinov), pp. 59–124. Cambridge, MA: MIT Press.
19. Kriete T, Noelle DC, Cohen JD, O'Reilly RC. 2013 Indirection and symbol-like processing in the prefrontal cortex and basal ganglia. *Proc. Natl Acad. Sci. USA* **110**, 16 390–16 395. (doi:10.1073/pnas.1303547110)
20. van der Velde F, de Kamps M. 2006 Neural blackboard architectures of combinatorial structures in cognition. *Behav. Brain Sci.* **29**, 37–70. (doi:10.1017/S0140525X06009022)
21. Holyoak KJ. 1991 Symbolic connectionism: toward third-generation theories of expertise. In *Toward a general theory of expertise: prospects and limits* (eds A Ericsson, J Smith), pp. 301–355. Cambridge, UK: Cambridge University Press.
22. McClelland JL, Botvinick MM, Noelle DC, Plaut DC, Rogers TT, Seidenberg MS, Smith LB. 2010 Letting structure emerge: connectionist and dynamical systems approaches to cognition. *Trends Cogn. Sci.* **14**, 348–356. (doi:10.1016/j.tics.2010.06.002)
23. Marcus GF. 1998 Rethinking eliminative connectionism. *Cogn. Psychol.* **37**, 243–282. (doi:10.1006/cogp.1998.0694)
24. Botvinick M, Plaut D. 2006 Short-term memory for serial order: a recurrent neural network model. *Psychol. Rev.* **113**, 201–233. (doi:10.1037/0033-295x.113.2.201)
25. O'Reilly RC. 2001 Generalization in interactive networks: the benefits of inhibitory competition and Hebbian learning. *Neural Comput.* **13**, 1199–1242. (doi:10.1162/08997660152002834)
26. Thomas MSC, McClelland JL. 2008 Connectionist models of cognition. In *The Cambridge handbook of computational psychology* (ed. R Sun), pp. 23–58. New York, NY: Cambridge University Press.
27. Bowers JS, Damian MF, Davis CJ. 2009 A fundamental limitation of the conjunctive codes learned in PDP models of cognition: comments on Botvinick and Plaut. *Psychol. Rev.* **116**, 986–995. (doi:10.1037/a0017097)
28. Bowers JS, Damian MF, Davis CJ. 2009 More problems with Botvinick and Plaut's (2006) PDP model of short-term memory. *Psychol. Rev.*, **116**, 995–997. (doi:10.1037/0033-295X.116.4.995)
29. van der Velde F, van der Voort van der Kleij GT, de Kamps M. 2004 Lack of combinatorial productivity in language processing with simple recurrent networks. *Connect. Sci.* **16**, 21–46. (doi:10.1080/09540090310001656597)
30. Hochreiter S, Schmidhuber J. 1997 Long short-term memory. *Neural Comput.* **9**, 1735–1780. (doi:10.1162/neco.1997.9.8.1735)
31. Lake BM, Baroni M. 2018 Generalization without systematicity: on the compositional skills of sequence-to-sequence recurrent networks. In *Proc. 35th Int. Conf. on Machine Learning, ICML 2018, 10–15 July 2018, Stockholm, Sweden*, pp. 2879–2888. ICML.
32. Gulordava K, Bojanowski P, Grave E, Linzen T, Baroni M. 2018 Colorless green recurrent networks dream hierarchically. In *Proc. 2018 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 1195–1205. (doi:10.18653/v1/N18-1108)
33. Barrett DG, Hill F, Santoro A, Morcos AS, Lillicrap T. 2018 Measuring abstract reasoning in neural networks. *arXiv* 1807.04225. (http://arxiv.org/abs/1807.04225)
34. Hill F, Santoro A, Barrett DG, Morcos AS, Lillicrap T. 2019 Learning to make analogies by contrasting abstract relational structure. *arXiv* 1902.00120. (http://arxiv.org/abs/1902.00120)
35. Smolensky P. 1990 Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artif. Intellig.* **46**, 159–216. (doi:10.1016/0004-3702(90)90007-M)
36. Plate TA. 1995 Holographic reduced representations. *IEEE Trans. Neural Network* **6**, 623–641. (doi:10.1109/72.377968)
37. Hummel JE. 2011 Getting symbols out of a neural architecture. *Connection Science* **23**, 109–118. (doi:10.1080/09540091.2011.569880)
38. Schlag I, Schmidhuber J. 2018 Learning to reason with third order tensor products. In *Advances in neural information processing systems* (eds S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, R Garnett), pp. 10 003–10 014. Montreal, Canada: MIT Press.
39. Eliasmith C. 2013 *How to build a brain: a neural architecture for biological cognition*. Oxford, UK: Oxford University Press.
40. Marcus G. 2001 *The algebraic mind: integrating connectionism and cognitive science*. Cambridge, MA: MIT Press.
41. Bowman H, Wyble B. 2007 The simultaneous type, serial token model of temporal attention and working memory. *Psychol. Rev.* **114**, 38–70. (doi:10.1037/0033-295X.114.1.38)
42. Swan G, Wyble B. 2014 The binding pool: a model of shared neural resources for distinct items in visual working memory. *Attent. Percept. Psychophys.* **76**, 2136–2157. (doi:10.3758/s13414-014-0633-3)
43. Simonyan K, Zisserman A. 2015 Very deep convolutional networks for large-scale image recognition. In *Proc. of the 3rd Int. Conf. on Learning Representations (ICLR 2015), May 2015, San Diego, CA*. ICLR.