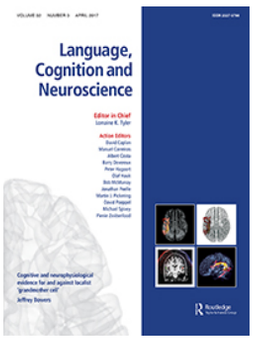

Routledge
Taylor & Francis Group

Language, Cognition and Neuroscience

ISSN: 2327-3798 (Print) 2327-3801 (Online) Journal homepage: http://www.tandfonline.com/loi/plcp21

Do arbitrary input–output mappings in parallel distributed processing networks require localist coding?

Ivan I. Vankov & Jeffrey S. Bowers

To cite this article: Ivan I. Vankov & Jeffrey S. Bowers (2017) Do arbitrary input–output mappings in parallel distributed processing networks require localist coding?, Language, Cognition and Neuroscience, 32:3, 392-399, DOI: 10.1080/23273798.2016.1256490

To link to this article: http://dx.doi.org/10.1080/23273798.2016.1256490

Published online: 02 Dec 2016.

LANGUAGE, COGNITION AND NEUROSCIENCE, 2017
VOL. 32, NO. 3, 392–399
http://dx.doi.org/10.1080/23273798.2016.1256490

Routledge
Taylor & Francis Group

REGULAR ARTICLE

Do arbitrary input–output mappings in parallel distributed processing networks require localist coding?

Ivan I. Vankov[a] and Jeffrey S. Bowers[b]

[a]Department of Cognitive Science and Psychology, New Bulgarian University, Sofia, Bulgaria; [b]School of Experimental Psychology, University of Bristol, Bristol, UK

ABSTRACT

The Parallel Distributed Processing (PDP) approach to cognitive modelling assumes that knowledge is distributed across multiple processing units. This view is typically justified on the basis of the computational advantages and biological plausibility of distributed representations. However, both these assumptions have been challenged. First, there is growing evidence that some neurons respond to information in a highly selective manner. Second, it has been demonstrated that localist representations are better suited for certain computational tasks. In this paper, we continue this line of research by investigating whether localist representations are learned in tasks involving arbitrary input–output mappings. The results imply that the pressure to learn local codes in such tasks is weak, but still there are conditions under which feed-forward PDP networks learn localist representation. Our findings further challenge the assumption that PDP modelling always goes hand in hand with distributed representations and provide directions for future research.

ARTICLE HISTORY
Received 24 March 2016
Accepted 20 October 2016

KEYWORDS
Localist representations; distributed representations; neural networks; PDP; arbitrary input–output mapping

Introduction

One of the main assumptions of the Parallel Distributed Processing (PDP) approach to understanding cognition is that information is represented in a distributed manner (McClelland, Rumelhart, & PDP Research Group, 1986). That is, each piece of meaningful information (e.g. letter, word, object, face, etc.) is coded by the simultaneous activation of multiple processing units (i.e. neurons) and, at the same time, each unit is involved in representing many different things. As a result, it is impossible to identify what a single-unit codes for. Distributed representations are thus different from both localist representations in psychology and grandmother cells in neuroscience, where the activation of units is interpretable.

Distributed representations are often claimed to enjoy a number of computational advantages over localist ones (Hinton, McClelland, & Rumelhart, 1986). For example, with distributed representations, similar items are coded with similar patterns of activation over the same set of units. This property makes it possible to interpolate between learned examples and thus to generalise to novel stimuli. For example, novel and familiar views of objects produce a similar pattern of activation over a set of units, and accordingly, it is possible to categorise objects presented in novel views. By contrast, localist and grandmother representations are often characterised as look-up tables that do not have the capacity to generalise (e.g. French, 1992; Poggio & Bizzi, 2004). Distributed coding also supports graded learning in which performance improves gradually over time and "graceful degradation" in which lesioning a few units leads to a small overall decrement on performance, rather than a catastrophic forgetting of specific items. Localist representations, by contrast, are claimed to have difficulties explaining both these results (e.g. Rogers & McClelland, 2014). Distributed coding schemes are also claimed to have a greater representational capacity than a localist coding schemes (Hinton et al., 1986).

In addition to these computational arguments, it is widely claimed that the brain relies on distributed representations. Indeed, grandmother cells are widely dismissed as untenable in the neuroscience literature (e.g. Rolls, in press), and this is taken as evidence against localist representations in psychology (Plaut & McClelland, 2010). Given these computational and biological claims, it is no surprise that many theorists have endorsed distributed representations.

Nevertheless, there are reasons to question some of these conclusions. First, there is a growing body of evidence suggesting that some neurons code for information in a highly selective manner. A striking example

CONTACT Ivan I. Vankov i.i.vankov@cogs.nbu.bg

was reported by Quiroga, Reddy, Kreiman, Koch, and Fried (2005) who found some neurons in the hippocampus of humans that selectively responded to specific persons, objects or scenes (e.g. a neuron that selectively responded to photographs of Jennifer Aniston). Other studies have found neurons in the visual cortex of monkey that selectively responded to images of body parts, objects, or faces (Desimone, Albright, Gross, & Bruce, 1984; Logothetis & Sheinberg, 1996; Logothetis, Pauls, & Poggio, 1995). Bowers (2009) took these findings to be consistent with localist representations in the brain, although the findings are generally taken in support of sparse distributed representations (Quiroga, 2012; Waydo, Kraskov, Quiroga, Fried, & Koch, 2006). In either case, the findings do suggest that some neurons represent high-level information in a highly selective manner, and that localist models in psychology should not be ruled out on the basis of neuroscience.

Second, under some circumstances, PDP models learn highly selective (localist) representations that resemble the highly selective neurons found in the hippocampus and cortex (Bowers, Vankov, Damian, & Davis, 2014, 2016). For example, we found that PDP models learn localist codes when trained to code for multiple items at the same time. In these simulations we used recurrent PDP models of short-term memory (STM) based on Botvinick and Plaut (2006). In both papers we varied the number of items (words in this case) that were co-activated in STM and carried out single-unit recordings in the hidden layer (analogous to single cell recording studies in neuroscience) in order to characterise the learned representations. The key finding is that the models learned many localist representations of letters and words, and that the number of localist codes scaled with the number of items that had to be kept active in STM.

We argued that these findings not only highlight the general plausibility of localist coding schemes, but also provide an explanation as to why the brain might learn localist codes under some conditions. Specifically, it has been claimed that co-activating distributed patterns at the same time leads to a blend pattern that is ambiguous in that is not possible to reconstruct the constituent patterns that produced the blend – the so-called superposition catastrophe (Von der Malsburg, 1986). We took our findings to support this hypothesis, and that the solution to this might be that the brain learns localist codes when it is important to code multiple items at the same time (Bowers, 2002).

Arbitrary input–output mapping

In this article, we consider another condition that may lead to the development of localist codes, namely, when networks are trained on arbitrary input–output mappings. Indeed, it has long been hypothesised that localist (or highly sparse and selective representations) codes are better suited for representing arbitrary bits of information. For example, Marr (1971) and McClelland, McNaughton, and O'Reilly (1995) argued that the knowledge in the hippocampus is coded in a highly sparse and selective (although not localist) manner in order to encode new and arbitrary episodic memories quickly without suffering catastrophic interference (e.g. learning to associate a name to a new face). Coltheart, Rastle, Perry, Langdon, and Ziegler (2001) argued that localist word codes support the naming of exception words when the mappings between letters and sounds are irregular. Similarly, Farah (1990) noted that more selective (although not localist) codes are needed to support the mapping between objects and their names compared to objects and their meanings because the former mappings are more arbitrary. This was used to explain the finding that acquired neuropsychological disorders in object naming can be highly selective (e.g. the selective inability to name fruits and vegetables; Hillis & Caramazza, 1991).

What are the computational advantages of localist coding that make it more suitable for representing arbitrary input–output mappings? As already discussed, one of the fundamental characteristics of distributed representations is that similar things are represented in similar ways. While this feature facilitates generalisation and the parsimonious use of existing computational resources, it introduces a problem when similar inputs have to be mapped to very different outputs. In such cases, it may be beneficial to keep the internal representations of similar inputs as different as possible. When this computational pressure is pushed to the extreme, the solution is to have (close to) orthogonal internal representations, and the most efficient way to implement this is localist codes.

There have already been a few attempts to characterise what sorts of representations PDP models learn when trained on arbitrary input–output mappings and the results have been mixed. Hinton et al. (1986) trained a three-layered network to map letters to word meanings (an arbitrary mapping). The network included 30 input units, 20 hidden units, and 30 semantic units, and each word in the letter layer was coded by 3 letter units, and the meaning of each word was coded as a random pattern of activation over the 30 semantic units (the meaning of each word was defined by randomly setting a semantic unit active with probability 0.2). After training the model to associate 20 words with 20 different meanings the authors analysed the connection weights between units and measured performance after

lesioning single units in the network. The authors found that each hidden unit was strongly connected to multiple output units (suggesting distributed coding), and that lesioning single units tended to cause a slight rise in the error rate for several different words rather than the complete failure on one specific word (again suggesting distributed representations). These findings were taken to support the conclusion that distributed codes support arbitrary mappings (for similar conclusions, see Plaut & Shallice, 1993).

By contrast, Berkeley (1995) reported that a feed-forward network trained to solve a set of logic problems (that involved a complex set of input–output mappings) learned localist representations. After training the network, Berkeley carried out single-unit recording on each hidden unit in response to all the trained inputs. He found that the hidden units were activated in a "banded" manner (such that a subset of the trained inputs all drove the hidden unit to a similar level of activation), and that these bands could often be assigned featural interpretations. For example, in one of his simulations, all of the input patterns that fell into a band included the feature unit OR (one feature common to many of the logical problems). That is, the model seemed to learn localist representations (in this case an "OR" unit), with individual units coding for discrete parts of the logical problem. We found similar banding patterns in our superposition simulations using this same method of recording hidden units (Bowers et al., 2014, 2016).

More recently, a variety of "deep" convolutional networks that perform state-of-the-art object recognition have been shown to learn highly selective object representations in their upper layers (Yosinski, Clune, Nguyen, Fuchs, & Lipson, 2015). These networks consist of multiple pooling and special "convolutional" layers which serve as feature maps. Importantly, the units in the subsequent layers are not fully interconnected and the weights of the convolutional layers are shared across many units. This architecture provides an engineering solution to solving spatial invariance as well as reduce the number of number of connections that need to be trained (and thus speeds up training). But importantly, these networks do not have any built-in mechanisms designed to lead to selective coding (e.g. "winner-takes-all" lateral inhibition connections), and accordingly, these results again suggest that the local codes were learned for adaptive reasons.

Together, these studies show that networks trained on complex (or arbitrary) input–output patterns one-at-a-time learn distributed representations under some conditions and localist representations under others. However, it is unclear why the different results were obtained. The goal of the current paper is to explore the conditions that drive PDP networks to learn selective representations when trained on arbitrary input–output mappings one-at-a-time.

Our simulations and findings are as follows. In Simulation 1, in order to manipulate the difficulty of the input–output mappings, we varied the density and the similarity of input patterns, the size of the hidden layer, and the training set size. In Bowers et al. (2014), we found that the number of localist representations learnt in a PDP model of STM increased when the input representations were denser and had increased overlap, and we hypothesised this could explain the previous contrasting results. However, for Simulation 1, we found no local codes in any of our conditions. In Simulation 2, rather than using a single input, we investigated whether a PDP network would learn localist codes for a category of inputs (i.e. localist codes for many-to-one mappings), as this condition is more similar to those faced by deep networks trained to categorise input patterns. Again, no local codes were learned. Finally, in Simulation 3 we used more natural stimuli – photos of human faces – and trained the network to learn both one-to-one and many-to-one mappings. The analysis revealed that the network learned localist representations only in the many-to-one task. Overall, it is clear that PDP networks can learn localist codes under some conditions, but we do not yet have a good understanding of the exact conditions required.

Simulation 1

A feed-forward PDP network was trained to store a variable number of arbitrary input–output mappings. The network consisted of an input layer having 500 units, a hidden layer of a size that was systematically varied between 100, 500 and 1000 units, and an output layer of 50 units. The input layer was fully connected to the hidden layer and the hidden layer was fully connected to the output layer. There was also a bias unit, connected to all the units in the hidden and the output layers. The activation of *i*th unit, a_i, in the hidden or he output layer was computed using the sigmoid function:

$$a_i = \frac{1}{1 + e^{-net_i}},$$

where net_i is the net input to unit *i*.

The performance of the network during training was assessed by applying the cross entropy function to the output layer:

$$error = \sum_i t_i \ln\left(\frac{t_i}{a_i}\right) + (1 - t_i)\ln\left(\frac{1 - t_i}{1 - a_i}\right),$$

where t_i is the target value of the ith unit of the output layer. The network was trained by using a backpropagation algorithm until the difference between the activation of the output layer and the target pattern, as measured by the cross entropy function, dropped below 0.01 per output unit. The learning rate was set to 0.01 in all of the simulations.

The training sets were constructed by generating a set of random input and target patterns and arbitrarily pairing them. Both the input and the target output patterns were binary (the input and the target unit activations were either 0 or 1). The training set was either small (100 input-target vector pairs), medium (500 pairs) or large (1000 pairs). The target patterns were distributed – they were generated by randomly turning on half of the output units. The density of the input patterns was manipulated by randomly turning on 20%, 40%, 60% or 80% of the input units. In sum, there was a total of 36 conditions under which the network was trained: 4 levels of input density (20%, 40%, 60%, 80%) × 3 levels of hidden layer size (100, 200, 400 units) × 3 training set sizes (100, 500 and 1000). The simulation was repeated 10 times for each of these conditions, each time generating a new training set. The average number of trials needed to train the network varied as a function of condition. With regards to density, 19,303, 17,035, 18,651 and 25,215 training trails were required for densities of 20%, 40%, 60% and 80%, respectfully. With regarding to the training set size, 9767, 17,315 and 34,512 training trials were required on average for 100, 500 and 1000 training examples, respectively. Finally, with respect to the size of the hidden layer, 27,301, 16,986 and 14,492 training trials were required for 100, 500 and 1000 hidden units.

Once training was complete the selectivity of the units in the hidden layer of the network was analysed by using the metric proposed in Bowers et al. (2014, 2016). That is, the selectivity of a unit with respect to a particular input vector was quantified as the minimal difference in its activation when the network was presented with this input compared to all other inputs. Thus, the selectivity of a unit varied between 1 (the unit was maximally activated only by a given input vector and all other inputs didn't activate it all) to −1 (the units was maximally activated by all inputs but one, which set its activation to 0). As in our previous work (Bowers et al., 2014, 2016) we characterised a unit if the absolute value of its selectivity exceeded 0.5

In order to demonstrate that the localist codes we found are not spurious, we calculated the number of local codes before the network was trained and made sure no local codes were found even then the selectivity criterion is reduced to 0.1.

The selectivity analysis revealed that the networks did not learn any localist representations in any of the conditions that varied the difficulty of the arbitrary input–output mapping. In order to make sure that this conclusion is not related to the specific value of the selectivity criterion we adopted (0.5), we also checked the number of units which absolute selectivity exceeded 0.4 and 0.3 – no such units were found in any of the conditions.

This finding is surprising given evidence that local codes do develop in deep networks when trained to categorise pictures of objects (a case of arbitrary input–output mapping). The results suggest the variables we manipulated (input density, hidden layer size and training set size) are not crucial for learning local coding schemes. There is another variable that may play a role in these unexpected results. Deep learning networks networks are usually trained to map together many different instances of a given category onto a single output category (e.g. many different images of cats are all mapped to the category "cat"). In Simulation 1 each input had its own arbitrary output (i.e. a 1-to-1 mapping) instead. Therefore, in Simulation 2 we explored the possibility that local codes emerge when networks area trained on many-to-one mappings.

Simulation 2

In the simulation reported above we found no examples of localist representations for specific inputs. However, the highly selective neurons found in neuroscience often respond to a variety of things belonging to a particular category (i.e. all photos of Jennifer Aniston). This raises the question as to whether we will find a similar outcome when we train the network to map multiple similar inputs onto the same output.

We also consider the possibility that the emergence of localist codes for categories of inputs depends on the similarity of categories to one another (between-group similarity) and the similarity of the exemplars belonging to the same category (within-group similarity). We conducted a simulation to assess whether a PDP network will indeed learn category specific localist representations when confronted with an arbitrary many-to-one mapping problem, and, in addition, whether the number of such representations will be modulated by between-group and within-group similarity. The training set consisted of 500 bipolar input patterns organised into 50 categories, 10 patterns per category. All input patterns belonging to a category were mapped to the same target pattern. The target patterns were the same as in Simulation 1. In order to manipulate between-group similarity, we first generated 50 "prototype" bipolar input patterns,

one for each category. The cosine similarity of the prototype patterns was low (average cosine similarity 0.2), or high (average cosine similarity 0.8). Then, for each prototype pattern we generated 10 patterns in which average similarity was either low (0.2) or high (0.8). The simulation was repeated 10 times, each time generating a new training set. The average number of training trials was 14,525 for the high between-group similarity condition, 8975 for low between-group similarity, 8050 for the high within-group similarity and 15,450 for the low-within-group similarity condition.

The results of the selectivity analysis revealed no local codes at any reasonable selectivity level (0.3, 0.4 or 0.5), suggesting that the nature of the arbitrary input–output mapping (one-to-one or many-to-one) does not drive learning localist representations.

Simulation 3

The goal of simulation 3 was to investigate if a PDP network can learn localist representations if given a naturalistic training set, rather than random input patterns. To this end, we used a publicly available database of photographs of human faces (AT&T Laboratories Cambridge, n. d.). The training set consisted of 400 photos of faces belonging to 40 people (10 images per person). The images were black and white and were scaled down to 45×37 pixels. Each pixel was coded by an input unit with an activation value that varied continuously from 0 to 1. rather than discretely (either 0 or 1). The target patterns were binary and were generated by randomly turning half of the output units on.

The simulation was run using both a one-to-one and a many-to-one mapping task. In the one-to-one mapping each image of a person was mapped to a unique distributed target pattern (resulting in 400 target patterns). In the many-to-one mapping task, we mapped all the images belonging to a given person to a single target pattern (which required 40 target patterns, one for each person). A feed-forward PDP network with 1665 (45×37) input units, 500 hidden units and 50 output units was trained to perform the 2 tasks until the error at the output layer dropped below 0.01 per unit. Apart for the larger input layer size, the settings of the network and the training procedure were the same as in the previous simulations. The simulation was repeated 10 times; it took 130,228 trials on average to train the network in the one-to-one mapping condition and 22,560 in the many-to-one mapping condition

The analysis of the hidden layer revealed that the network learnt local representations for specific people in the many-to-one condition, and no local codes at all in the one-to-one condition. In Figure 1(a) we depict the selectivity values of the 500 hidden units as well as the images which some of the most selective units responded to. The results show that PDP networks can learn some localist representations when the training set is composed of more naturalistic input patterns in which the input units coded for information in a continuous (grey-scale) rather than discrete manner. Interestingly, this only happened in the many-to-one mapping condition and only local codes for groups of things (i.e. all images of the same person), rather than local codes of individual images, were found. Figure 1b shows an example of a highly selective unit that responded only to the images of a specific person.

General discussion

Our first key result is that PDP models trained on arbitrary input–output mappings learned distributed rather than localist codes when the input and output units took on discrete values (standard in most PDP modelling in psychology). This was the case in Simulation 1 in which the model was trained to map inputs onto unique outputs (one-to-one mappings) when we varied similarity of the input patterns, the number of input patterns, as well as the number of hidden units (varying the difficulty of the task), and in Simulation 2 in which the model was trained to map multiple input patterns to a common output (many-to-one mappings) when we varied the similarity of the patterns within and between categories as well as the number of input patterns. This pattern of results contrasts with our previous work where we found that PDP models learned localist codes when trained to co-activate multiple items at the same time in STM using input and output units that took on binary values.

The second key result was that PDP models did learn localist representations under some conditions when trained on arbitrary mappings with photographs of faces where the input units took on continuous rather than binary values. When the model was trained on one-to-one mappings the model continued to learn no local codes, but when trained on many-to-one mappings, the model learned an average of 6 person-specific localist representations (out of 20 people). That is, the model learned localist codes for categories (persons) rather than local codes for individual photographs.

What is to be made of this pattern of results? The fact that distributed codes supported a wide range of complex arbitrary input–output mappings clearly demonstrates that the task of mapping between arbitrary domains does not constitute such as strong a pressure to learn localist codes as the task of co-activating multiple items at the same time (Bowers et al., 2014, 2016). In this

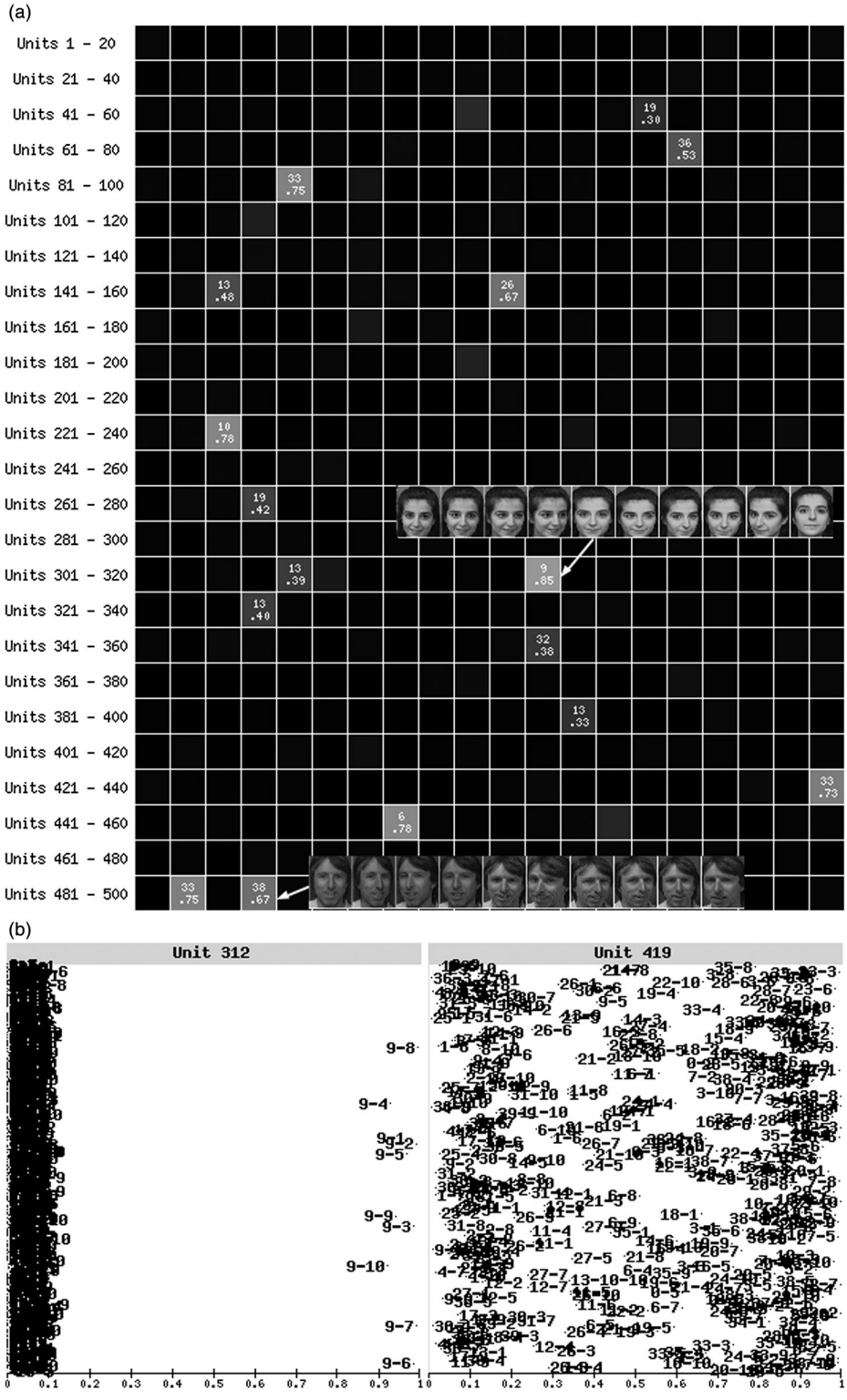

Figure 1. (a) Selectivity of the hidden layer units in Simulation 3. Each unit is represented by a square and its selectivity is indicated by the degree of lightness of the square, with light grey referring to a unit with high selectivity and black referring to a unit that is non-selective. The units which selectivity exceeds .3 are labelled with the number of the person that the unit responds to (there were images of 40 people in the training set, ten images per person). The numbers under the labels show the exact selectivity of the units. Two examples of the images that selectively activate hidden units are given. (b) Labelled scatter plots of a highly selective (left) and a non-selective unit (right) in Simulation 3. Each dot in the scatter plots represents a single input pattern and the label indicates the identification of the input. The activation of the unit varies along the *x*-axis. Unit 312 responded only to the images of the ninth person in the faces database.

earlier work, localist codes were always found when the model succeeded in co-activating and recalling multiple familiar and unfamiliar patterns. Indeed, we argued that localist codes are required in order to solve the superposition constraint. By contrast, localist codes are not required for many complex input–output mappings.

At the same time, our findings again highlight that standard feed-forward PDP networks learn localist codes under some conditions when trained on items one-at-a-time. This is in line with earlier work by Berkeley (1995) who trained networks on logical problems (when inputs took on discrete values) as well as recent work with "deep networks" (networks with many hidden layers) that support state-of-the-art recognition of images in scenes (Le et al., 2012; Yosinski et al., 2015). For example, Yosinski et al. (2015) analysed individual units in deep network that came in first place ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2012 (Krizhevsky, Sutskever, & Hinton, 2012). They found that the best image for driving a single neuron was often an easily interpretable image (a localist code). Perhaps even more impressive, Le et al. (2012) trained a deep network on 10 million images in an unsupervised manner, and found a unit that selective coded for faces. The authors wrote:

> But perhaps more importantly, it answers an intriguing question as to whether the specificity of the "grandmother neuron" could possibly be learned from unlabeled data.

The answer appears to be "yes" given their analyses.

What remains unclear to us, however, is why local codes are learned in some conditions and not others when trained on items one-at-a-time. Interestingly, we observed local codes under the same conditions that these deep networks learn local codes, namely, when detailed images (with input units that take on continuous values) are involved in many-to-one mappings. But what is it about these conditions that lead to local codes whereas the one-to-one mappings with these same images (Simulation 3), or many-to-one mappings with dense, but discrete input patterns (Simulation 2) do not? Similarly, what it is about the input–output mappings employed by Berkeley (1995) that lead to localist input codes, given that he used sparse and discrete input patterns? We do not have any good intuitions about why these contrasting results are observed.

There are not esoteric questions only relevant to theorists interested in specific computational models. We have used PDP models because the learned representations are not stipulated (Plaut & McClelland, 2000). That is, models learn the representations that are best or most efficient in solving a given task. This opens up the possibility that PDP models can be used to advance hypotheses as to when localist or distributed codes are best suited for solving tasks more generally, including when the tasks are performed by the brain. We think we have developed a good understanding regarding the conditions in which recurrent PDP models learn localist codes when trained on multiple items at the same (Bowers et al., 2014, 2016), and argue that this provides a possible explanation as to why cortical systems learn selective codes. Our hope is that the current results provide a first step in developing a better understanding of the conditions in which PDP models learn localist codes when trained on items one-at-a-time may, with similar implications for neuroscience.

It is perhaps worth emphasising that in that all our previous work, as well as here, we always find a majority of units that do not respond selectively to inputs. Accordingly, our findings are consistent with the common claim that PDP networks learn distributed representations, but contrary to the standard assumption, we have found that PDP models learn a mixture of localist and distributed codes under a range of conditions. We would hypothesise that the biological neural systems may also rely on mixed (localist/distributed) representations when coding multiple items at the same time (Bowers et al., 2014, 2016) or when performing some sorts of arbitrary input–output mappings.

In summary, we take the neuroscience and computational results to challenge the two main reasons why distributed representations are favoured over localist coding. Contrary to the common claim that PDP models learn distributed codes, and this reflects the computational advantages of distributed coding, we and others have shown that a range of neural networks often learn localist codes, and this reflects the computational advantages of localist coding across a range of conditions. Second, in contrast with the common claim that the brain codes information in a distributed manner, a wide variety of data now highlight how selective neural coding can be. Our computational work suggests some hypotheses as to why these results may be observed, and challenge the widespread view in psychology that distributed representations should be preferred on computational and biological grounds. More generally, we hope our results inspire more work on the analysis of hidden layer representations across a range of network architectures in order to better characterise the computational tasks that generate distributed and localist coding. In our view, a better understanding of these constraints will provide important insights for theories in both neuroscience and psychology.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

AT&T Laboratories Cambridge. (n.d.). The database of faces. Retrieved from http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

Berkeley, I. N. (1995). Density plots of hidden value unit activations reveal interpretable bands. *Connection Science*, *7*(2), 167–187. doi:10.1080/09540099550039336

Botvinick, M. M., & Plaut, D. C. (2006). Short-term memory for serial order: A recurrent neural network model. *Psychological Review*, *113*(2), 201–233. doi:10.1037/0033-295X.113.2.201

Bowers, J. S. (2002). Challenging the widespread assumption that connectionism and distributed representations go hand-in-hand. *Cognitive Psychology*, *45*(3), 413–445. doi:10.1016/S0010-0285(02)00506-6

Bowers, J. S. (2009). On the biological plausibility of grandmother cells: Implications for neural network theories in psychology and neuroscience. *Psychological Review*, *116*(1), 220–251. doi:10.1037/a0014462

Bowers, J. S., Vankov, I. I., Damian, M. F., & Davis, C. J. (2014). Neural networks learn highly selective representations in order to overcome the superposition catastrophe. *Psychological Review*, *121*(2), 248–261. doi:10.1037/a0035943

Bowers, J. S., Vankov, I. I., Damian, M. F., & Davis, C. J. (2016). Why do some neurons in cortex respond to information in a selective manner? Insights from artificial neural networks. *Cognition*, *148*, 47–63. doi:10.1016/j.cognition.2015.12.009

Coltheart, M., Rastle, K., Perry, C., Langdon, R., & Ziegler, J. (2001). DRC: A dual route cascaded model of visual word recognition and reading aloud. *Psychological Review*, *108*(1), 204–256. doi:10.1037/0033-295X.108.1.204

Desimone, R., Albright, T. D., Gross, C. G., & Bruce, C. (1984). Stimulus-selective properties of inferior temporal neurons in the macaque. *The Journal of Neuroscience : The Official Journal of the Society for Neuroscience*, *4*(8), 2051–2062.

Farah, M. J. (1990). *Visual agnosia: Disorders of object recognition and what they tell us about normal vision*. Cambridge, MA: MIT Press.

French, R. M. (1992). Semi-distributed representations and catastrophic forgetting in connectionist networks. *Connection Science*, *4*(3–4), 365–377. doi:10.1080/09540099208946624

Hillis, A. E., & Caramazza, A. (1991). Category-specific naming and comprehension impairment: A double dissociation. *Brain*, *114*(5), 2081–2094. doi:10.1093/brain/114.5.2081

Hinton, G. E., McClelland, J. L., & Rumelhart, D. E. (1986). Distributed representations. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition*, (vol. 1 pp. 77–109). Cambridge, MA: MIT Press.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *NIPS 2012: Neural Information Processing Systems*, Lake Tahoe, Nevada

Le, Q., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G., ... Ng, A. (2012). Building high-level features using large scale unsupervised learning. In J. Langford & J. Pineau (Eds.), *Proceedings of the 29th International Conference on Machine Learning (ICML-12), ICML '12* (pp. 81–88). New York, NY: Omnipress.

Logothetis, N. K., Pauls, J., & Poggio, T. (1995). Shape representation in the inferior temporal cortex of monkeys. *Current Biology*, *5*(5), 552–563. doi:10.1016/S0960-9822(95)00108-4

Logothetis, N. K., & Sheinberg, D. L. (1996). Visual object recognition. *Annual Review of Neuroscience*, *19*, 577–621. doi:10.1146/annurev.ne.19.030196.003045

Marr, D. (1971). Simple memory: A theory for archicortex. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 23–81. doi:10.1098/rstb.1971.0078

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning-systems in the Hippocampus and Neocortex – Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, *102*, 419–457. doi:10.1037/0033-295X.102.3.419

McClelland, J. L., Rumelhart, D. E., & PDP Research Group. (1986). *Parallel distributed processing: Psychological and biological models* (Vol. 2). Cambridge, MA: MIT Press.

Plaut, D. C., & McClelland, J. L. (2000). Stipulating versus discovering representations. *Behavioral and Brain Sciences*, *23*(4), 489–491. doi:10.1017/S0140525X00473358

Plaut, D. C., & McClelland, J. L. (2010). Locating object knowledge in the brain: Comment on Bower's (2009) attempt to revive the grandmother cell hypothesis. *Psychological Review*, *117*, 284–288. doi:10.1037/a0017101

Plaut, D. C., & Shallice, T. (1993). Deep dyslexia: A case study of connectionist neuropsychology. *Cognitive Neuropsychology*, *10*(5), 377–500. doi:10.1080/02643299308253469

Poggio, T., & Bizzi, E. (2004). Generalization in vision and motor control. *Nature*, *431*(7010), 768–774. doi:10.1038/nature03014

Quiroga, R. Q. (2012). Concept cells: The building blocks of declarative memory functions. *Nature Reviews Neuroscience*, *13*, 587–597. doi:10.1038/nrn3251

Quiroga, R. Q., Reddy, L., Kreiman, G., Koch, C., & Fried, I. (2005). Invariant visual representation by single neurons in the human brain. *Nature*, *435*(7045), 1102–1107. doi:10.1038/nature03687

Rogers, T. T., & McClelland, J. L. (2014). Parallel distributed processing at 25: Further explorations in the microstructure of cognition. *Cognitive Science*, *38*, 1024–1077. doi:10.1111/cogs.12148

Rolls, E. (in press). Cortical coding. *Language, Cognition and Neuroscience*. doi:10.1080/23273798.2016.1203443

Von der Malsburg, C. (1986). Am I thinking assemblies? In G. Palm & A. Aertsen (Eds.), *Brain theory* (pp. 161–176). Berlin: Springer.

Waydo, S., Kraskov, A., Quiroga, R. Q., Fried, I., & Koch, C. (2006). Sparse representation in the human medial temporal lobe. *Journal of Neuroscience*, *26*, 10232–10234. doi:10.1523/jneurosci.2101-06.2006

Yosinski, J., Clune, J., Nguyen, A. M., Fuchs, T., & Lipson, H. (2015). Understanding neural networks through deep visualization. *CoRR*. Retrieved from http://arxiv.org/abs/1506.06579