Running head: SUPERPOSITION CATASTROPHE

PDP Networks Learn Local (Grandmother Cell) Representations in Order to Code for Multiple Things at the Same Time

Jeffrey S. Bowers[1]
Markus F. Damian[1]
Colin J. Davis[2]

1. University of Bristol
2. Royal Holloway University of London

Key Words: Grandmother cells; localist representations; PDP models; Short-term memory; superposition catastrophe

Address correspondence to:
Dr Jeffrey Bowers
University of Bristol, Department of Experimental Psychology
12a Priory Road, Bristol BS8 1TU, United Kingdom
Email: j.bowers@bristol.ac.uk
Tel. (+44) (0)117 – 928 8573

Abstract

One potential limitation with distributed representation is that it is difficult to co-activate multiple things at the same time over the same set of processing units. That is, a superposition of co-active distributed representations results in a blend pattern that is ambiguous; the so-called superposition catastrophe. Thus it is striking that M. M. Botvinick and D. C. Plaut (2006) developed a PDP model of short-term memory that recalls lists of letters based on the superposition of distributed letter codes. Their finding suggests that the distributed representations can solve the superposition catastrophe. However we show that the model's success does not mitigate against this constraint. Under the appropriate training conditions a version of their model can indeed solve (avoid) the superposition catastrophe, but only by learning localist as opposed to distributed representations. Given that many cognitive systems need to code multiple things at the same time, the pressure to learn localist ("grandmother cell") representations is widespread.

PDP Networks Learn Local (Grandmother Cell) Representations in Order to Code for Multiple Things at the Same Time

Neural networks can code for information in two qualitatively different ways. Some models rely on localist representations, such that words, objects, faces, and lexical concepts are coded distinctly, with their own dedicated representations. For example, in localist models of written word identification, the words dog and log are coded with distinct orthographic units, and a word is identified when a single unit (e.g., dog) is activated beyond some threshold (e.g., McClelland & Rumelhart, 1981). Other models rely on distributed representations, such that the words dog and hog are coded as patterns of activation over collections of units, with each unit involved in coding multiple words (e.g., Seidenberg & McClelland, 1989). In fact, various different types of distributed representations can be distinguished, ranging from dense distributed representations in which many hidden units are co-activated in response to a single input and each hidden unit is involved in coding many different things, to sparse distributed coding in which each input activates relatively few units, and each unit is involved in coding relatively few things (but more than one). Nevertheless all versions of distributed coding share one defining feature; that is, it is not possible to interpret the activation of a single unit unambiguously.

The debate regarding the relative merits of localist and distributed coding has taken a variety of forms, including their relative biological plausibility (e.g., Bowers, 2009, 2010ab; Plaut & McClelland, 2010ab; Quian Quiroga & Kreiman, 2010ab), their empirical successes (e.g., Coltheart, Rastle, Perry, Langdon, & Ziegler, 2001; Seidenberg & Plaut, 2006), and their computational capacities (e.g., Botvinick & Plaut, 2009ab; Bowers, Damian, & Davis, 2009ab, Page, 2000; Plaut & McClelland, 2000). In the present paper we focus on the computational capacity of the distributed representations typically learned in PDP models, and argue that they are ill suited to support the co-activation of multiple items at the same time over the same set of units. That is, we claim that these distributed representations cannot overcome the *superposition catastrophe*, contrary to some recent findings that suggest otherwise (Botvinick & Plaut, 2006). We argue that the functional requirement to co-activate multiple things at the same time provides a computational pressure to develop representations that respond highly selectively to inputs. Indeed, we show that PDP models sometimes learn local ("grandmother cell") representations in response to this constraint.

Catastrophic interference

There is already one well known computational limitation of the dense distributed representations often learned in PDP models; namely these representations are poor at supporting rapid one-trial learning, as required for episodic memory. The problem is that each unit in a dense distributed representation is involved in coding many things, and as a consequence, learning something new impacts on pre-existing knowledge (specifically, knowledge coded with overlapping units and connection weights). The greater the learning the greater the impact on pre-existing knowledge, and in the case of rapid one-trial learning, new knowledge can erase prior knowledge, a phenomenon known as catastrophic interference (McClosky & Cohen, 1989; Ratcliff, 1990), or the stability-plasticity dilemma (Grossberg, 1976).

According to McClelland, McNaughton, and O'Reilly (1995), the brain's solution to this problem was to develop sparse coding in the hippocampus and related structures, with relatively few units active at any one point, and each unit involved in coding relatively few things. These sparse codes are not considered local (each unit would be involved in coding for more than one thing), but nevertheless, the solution to catastrophic interference is to reduce the level of distribution. In this way, large changes in connections weights (in response to new learning) have minimal impact on prior knowledge.

However, this solution is itself limited. PDP models generalize on the basis of new inputs activating hidden units that are involved in coding old knowledge, and this overlap is often reduced when knowledge is coded sparsely. This makes the sparse codes in the hippocampus a poor medium for generalizing. For this reason, McClelland et al. (1995) argue that the brain relies on both dense and sparse distributed coding schemes: dense distributed representations in the cortex that support various forms of generalization required for perception, language, and semantic memory, and sparse distributed codes in the hippocampus (and related structures) for episodic memory. This dual solution to fast learning and generalization is the so-called *complementary learning systems hypothesis*.

The key claim we make here is that there is yet another computational constraint that undermines the use of dense distributed representations in the cortex as well. That is, the dense distributed representations learned in PDP networks are not only poor at fast learning, but also poorly suited to the task of co-activating multiple things at the same time. Accordingly, the challenge is to develop neural networks that can both generalize and co-activate multiple things. We show that PDP networks that learn localist representations of letters in response to the superposition constraint can indeed generalize to novel words.

The Superposition Catastrophe

PDP models support generalization in a variety of domains (e.g., reading and semantics), but until recently, there have been no demonstrations that these models can co-activate multiple things at the same time over the same set of units. Given that various cognitive systems support the co-activation of multiple items at the same time (perhaps 4 +/-1; Cowan, 2001), this is a striking omission. Particularly so given that it has been claimed that distributed representations are ill suited for supporting this function, due to the superposition catastrophe (von der Malsburg, 1986).

The superposition catastrophe refers to the observation that co-activating items in a distributed system leads to blend patterns that are ambiguous with regards to what produced the blend in the first place. For example, consider Figure 1, adapted from Page (2000), that depicts distributed patterns for the names Paul, John, George, Ringo, Mick, and Keith. The patterns are distributed given that each name is coded as a pattern of activation of two of four units, and the identity of a person cannot be determined by the activation of a single unit. Although the identity of a name can be determined based on the pattern of activation over the four units, the coding scheme is ambiguous when more than one person is coded at the same time over the same units. For example, the combination of MICK and KEITH co-activates all four units, but so do JOHN and GEORGE. The tempting conclusion is that it is difficult to code for more than one thing at a time over a given set of processing units when relying on distributed representations. By contrast, ambiguity is eliminated with localist coding schemes. For example, as can be seen in Figure 2, if localist Mick and Keith units are co-active, the constituent patterns that produced the blend could only be Mick and Keith. Bowers (2002) used the superposition catastrophe as an argument in favor of localist coding models.

The ambiguity associated with a superposition of dense distributed representations is in fact worse than the example above. In Figure 1, the ambiguity is the product of blending together two familiar names (e.g., Mick and Keith), but blends might just as well be produced by combinations of novel inputs. If novel inputs are free to contribute to blends, then there is an infinity of different combinations of novel constituents that will produce the same blend. In this sense the superposition catastrophe is an ill-posed problem; it is logically impossible to determine the constituent patterns that produced a given blend. This would seem to rule out dense distributed representations as a medium for supporting cognitive processes that deal with more than one thing at a time (that is, most of cognition).

Nevertheless, it is important to note that the brain faces and often solves other ill posed problems. The classic example is the so-called inverse problem in vision, in which an infinite

number of possible shapes in the 3D world are consistent with a given 2D image projected on the retina. The conclusion is not that we cannot perceive in 3D, but rather, that the visual system needs to adopt biases about which 3D shapes are most likely to have produced a given (ambiguous) 2D image on the retina. The biases could be based on innate constraints or learning (Sinha & Poggio, 1996). In a similar way, in order to overcome the superposition catastrophe with dense distributed representations, some sort of adaptive bias might play a role selecting one set of constituent patterns from an infinite possible set that could produce a given blend pattern. The trick, of course, is to include a bias that selects the correct constituents.

The Botvinick and Plaut (2006) PDP model of STM and a potential solution to the superposition catastrophe

Although the superposition catastrophe has largely been ignored in the PDP literature, Botvinick and Plaut (2006) report a PDP model of STM that confronts this issue head on, and the authors show that their model can indeed learn adaptive biases that allow it to accurately encode, store, and recall multiple familiar items based on the superposition of dense distributed representations. The model attempts to account for a standard measure of STM, namely, immediate serial recall. In immediate serial recall, a participant is presented a list of items (e.g., letters, numbers, words, etc.), and is asked to repeat them back in the same order. Participants can retain 7+/-2 (or perhaps 4+/-1; Cowan, 2001) items in short-term memory for a few seconds. The Botvinick and Plaut (2006) model shows a similar STM capacity, and captures a range of important empirical facts about STM.

A schematic diagram of their model (taken from their article) is presented in Figure 3. It includes a set of localist input and output units and an intervening set of hidden units that map between them. As can be seen in the Figure, the hidden units include feedback (recurrent) connections to themselves, and the hidden units are bidirectionally associated with the output layer. The connection weights constitute the LTM of the model, and an activation pattern across the units constitutes the model's STM, with the recurrent connections ensuring that the activation persists in the absence of input.

The key finding reported by the authors is that the trained model is able to code for both item and order information, relying on a distributed pattern of activation over the hidden units. For instance, if the input units that code for the letters A, D, F, Z, Q, R are activated in sequence, the corresponding output units can be recalled (activated) in the correct order with a likelihood that is similar to human performance. The model supports the serial recall of multiple items based on a superposition of distributed activation patterns in the hidden layer. That is, each item in a to-be-remembered list produces its own distributed pattern of activation over the hidden units that codes for a given letter in a given position (e.g., a conjunctive code for A-in-position-1, B-in-position-2, etc.), and it is the combined (superimposed) activation pattern across all items that codes and reproduces the sequence at the output layer. Furthermore, the model can correctly recall many sequences that it has never been exposed to. For example, in Botvinick and Plaut's (2006) first simulation the model was trained on approximately one million letter sequences (of various lengths), and then was tested on random strings of six letters. These test sequences were almost always novel (99.3% of the time), and nevertheless the model succeeded ~50% of the time, which is similar to human performance (but see Bowers et al., 2009ab, for some types of sequences that this model cannot recall).

The success of the model is striking, as it seems to undermine the claim that dense distributed representations are subject to the superposition catastrophe. Indeed, the model succeeded even when noise was added to the distributed patterns. Botvinick and Plaut (2006) are explicit about the way their model retrieves a sequence of items based on a noisy and ambiguous blend pattern. That is, the model makes a response that is most likely to be correct given the blend pattern's similarities to the patterns encountered during training. More specifically, they claim that their model effectively computes a Bayesian analysis in order to

select the most likely set of items when confronted with an ambiguous pattern of activation across a set of hidden units. Just like the inverse problem in vision is solvable with the inclusion of innate or leaned biases, the Botvnick and Plaut (2006) model of STM suggests that PDP models can learn a bias that provides an adaptive solution to the superposition catastrophe.

Two possible problems with Botvinick and Plaut's solution to the superposition catastrophe model

The success of the Botvinick and Plaut (2006) model clearly shows that PDP models can learn adaptive biases that help them overcome the superposition catastrophe. However, there are at least two reasons to think that the solution may not work in general. First, their model succeeded by decomposing the blends into the most likely sequence given its training history. This allows the model to retrieve novel sequences of familiar items, but it may introduce problems if the model was presented with an unfamiliar item. That is, novel items are not the most likely output given the training history, and accordingly, the model might be expected to lexicalize, producing an incorrect familiar item most consistent with a given blend. Although human STM is better for familiar compared to unfamiliar words (Jefferies, Frankish, & Lambon Ralph, 2006), we nevertheless have no difficulty in repeating a few nonwords, such as "blip-blap". The potential difficulty of coding multiple unfamiliar things with distributed representations was highlighted by Bowers (2002):

> Perhaps most problematic, blends are not necessarily the product of combining pre-trained patterns. Imagine the situation in which two words are co-active in a distributed phonological system. Although the blend pattern may be more similar to the two constituent words compared to any other trained word, the pattern is not more similar to many possible items (or possible blends). The blend pattern might have been produced by combining two nonwords, for example, although this possibility cannot be recovered from the blend. But we can co-encode two novel items: e.g., phonologically, as BLIP–BLAP in short-term memory… blend patterns in distributed systems are deeply ambiguous (p. 431)

A second possible limitation concerns familiar items. That is, the bias solution may only work when the model is trained with a small vocabulary of items (e.g., 26 letters). Under these conditions, Botvinick and Plaut have shown that there is only one or two possible sequences of familiar items that could have produced a given blend – which allows the model to succeed most of the time by selecting the most probable solution. However, if the model was trained with a larger set of familiar items, it is not clear that there is only one or two possible sequences of familiar items that could have produced the blend. To the extent that a variety of different sequences of familiar items could have produced the blend, the bias solution would become inadequate.

Only a small number of studies shed some insight into the conditions in which PDP models can address the superposition catastrophe. Bowers et al. (2009a) developed a modified version of the Bovinick and Plaut (2006) STM model in which letters were coded in a distributed rather than localist manner. That is, each letter was defined as a pattern of activation over five input and output units, and each unit contributed to the coding of five letters. Accordingly, it was not possible to interpret a given input unit unambiguously. The critical finding was that the model could learn to recall a series of familiar letters (taken from a vocabulary of 25), but it catastrophically failed when tested on lists that contains a novel letter (that is, a novel pattern of five activated input units). This fits well with the hypothesis that the model developed a bias to recall the most likely sequence given its training history, which rules out lists that contain novel items.

By contrast, Botvinick and Plaut (2009) developed a modified version of their model in which the first 10 units each represented a letter in the onset position of a syllable, the next 10

units each represented a vowel in a syllable, and the final 10 units all represented a coda of a syllable. That is, the model included localist input units for letters, and coded syllables through the co-activation of one onset, vowel, and rhyme. They trained the model on lists of syllables taken from a vocabulary of 999 syllables (out of a possible 10 x 10 x 10, or 1000 syllables). The first key finding was that the model succeeded on lists of familiar syllables (when trained and tested on up to three items). This shows that successful performance on familiar items is not limited to cases in which the model was trained on a small vocabulary. Second, and more strikingly, the model could also repeat sequences that included the (one) untrained syllable. According to Botvinick and Plaut, the failure of the Bowers et al. (2009a) model to generalize to novel letters reflected something idiosyncratic about the simulation rather than some intrinsic limitation of distributed representations.

Similarly, Bowers et al. (2009b) developed a model that included 10, 6, and 10 units devoted to letters in the onset, vowel, and rhyme positions of syllables. We found that the model was equally good at recalling lists of familiar and unfamiliar syllables of up to list length six when it was trained on 500 syllables (out of a possible 10 x 6 x 10 or 600 syllables). That is, the model went from catastrophic failure on novel letters (when trained on 25 items) to striking success on unfamiliar syllables (when trained on 500 items). Indeed, as noted by Bowers et al., the model did too well on novel syllables given that STM in humans is much better for words compared to nonwords.

Do these successes undermine the significance of the superposition catastrophe? Not necessarily. It is unclear how the latter models succeeded, but one possibility is that they learned sparse or localist representations in order to overcome the superposition constraint. Such an outcome would not only highlight the limitation of dense distributed coding schemes, but also the adaptive value of sparse or local coding schemes when coding multiple items simultaneously.

Below we report a series of simulations and analyses in an attempt to gain insight into these contrasting results. To this end we systematically varied the conditions in which the above models failed and succeeded. One key difference was with respect to the input coding schemes. The Bowers et al. (2009a) simulation that failed with a novel letter included a distributed input and output coding scheme in which each unit was involved in representing multiple (five) letters, and where there were no phonotactic constraints (any 5 input and output units were free to be co-activated). By contrast, the subsequent models that succeed with novel syllables included a localist input and output coding scheme, with each input and output unit devoted to coding a specific letter, and where syllables were phonotactically constrained (each syllable was composed of one onset, one rhyme, and one coda; Botvinick and Plaut, 2009; Bowers et al., 2009b). Accordingly, the reliance on a localist input/output coding scheme and the inclusion of phonotactic constraints might play a role in supporting nonword recall. A second key difference was that the Bowers et al. (2009a) model failed when it was trained from a small vocabulary of syllables, and subsequent models succeeded when they were trained on a larger vocabulary. Accordingly, generalization to novel items may simply rely on a larger training set.

In addition to assessing the conditions in which PDP models succeed and fail in recalling lists of familiar and unfamiliar items, we attempt to consider why the models succeed and fail. To this end we systematically analyze the hidden units of the models with localist and distributed input coding schemes in the various training conditions. The key question is whether the models ever succeed with lists of familiar and unfamiliar items by employing dense distributed representations, or whether the models rely on sparse or localist codes to address the superposition catastrophe.

Simulation 1a-b

In Simulations 1a-b we trained two models to recall lists of syllables taken from a small vocabulary of 26 syllables, and at test, assessed their performance on lists of familiar syllables, as well as on a single novel syllable. The two models were structurally identical, with 26 input units, 200 hidden units, and 26 output units (same as in Botvinick and Plaut, 2006). The key difference is in the way in which the syllables are coded in the input layer. In Simulation 1a the input coding scheme was distributed, in that no unit could be interpreted. Each syllable was simply coded by a random pattern of activation over three of 26 input and output units (much like Bowers et al., 2009a, where each letter was coded as a random pattern of activation over five units). From now on we refer to this as the *distributed input (DI) model*. In Simulation 1b, the syllables in the input layer were coded through the co-activation of localist letter codes (same as in Bowers et al., 2009b). The first 10 input (and output) units coded for the onsets of syllables, the next 6 items for vowels, and the final 10 units for codas, and each syllable was coded as one active onset, vowel, and coda unit. Specifically, the 26 input units, organized by onset, rhyme, and coda, coded for the following letters: (b, c, d, f, g, h, j, k, l, m) (a, e, i, o, u, y) (n, p, q, r, s, t, v, w, x, z). From now on we refer to this as the *localist-input (LI) model*. Given that each syllable in the LI model was defined as the co-activation of one onset, one rhyme, and one code unit, there were constraints on what units could be co-activated (a phonotactic constraint). No such constraint applied to the DI model, where any three units could be co-activated. The set of 26 syllables and the input units that they were coded with are listed in Table 1.

The two models were trained on lists of syllables that varied in length from one to nine syllables, and lists were composed of a random selection of syllables (without replacement). The output of the model was determined by comparing the pattern of activation at the output layer with the patterns that defined the 26 familiar syllables. The model was said to recall the syllable with the highest dot product. This is similar to selecting the most active letter in a localist-coding scheme. This is the same procedure as in Bowers et al. (2009ab).

After one million training trials the two models were tested on lists of six familiar syllables. The DI and LI models were correct on 61.7% and 59.3% of the lists, respectively. Accordingly, the nature of the input patterns did not seem to impact on the models' performance when tested on lists of familiar syllables taken from a small vocabulary.

We then took the same models and tested them with novel syllables. For the DI model we generated 1000 syllables by randomly activating three of the input units, and we avoided patterns that were already within the training set. We then tested the model by presenting it with a single item to remember (rather than the standard list of six familiar items). Recall in this case was determined by comparing the pattern of activation at the output layer with the patterns that defined the 26 familiar syllables and the pattern that defined the novel syllable. The DI-model performed very poorly, with an accuracy rate of only 12.1%. When we tested the model on lists of two novel items, its performance dropped to under 1%. Clearly, the ability of the model to code for multiple items at the same time is restricted to familiar items, as the memory span for novel items is approximately zero.

Similar results were obtained for the LI-model. The novel items constituted the entire set of possible syllables other than the 26 trained items. That is, the model was tested on 574 untrained syllables (10 x 6 x 10 possible syllables, minus 26) one at a time. From these, 1000 test patterns were randomly selected. Overall performance was again quite poor, with 15.4% accuracy. We then tested the model on lists of two novel items, and performance dropped to under 1%. Clearly, the structure of the inputs did not impact on the models' ability to recall novel items: In both cases, memory span for novel items was approximately zero.

Simulation 2a-b

One obvious reason why the models may have failed to generalize to novel syllables is that they were trained on a highly restricted set of 26 syllables. Perhaps the limited training set

prevented the model from learning the necessary regularities to support generalization. Another possibility, however, is that the training set is adequate to support generalization, but during training, the models learned a bias to report only familiar items in an attempt to overcome the superposition catastrophe.

In order to distinguish these two hypotheses we trained the two models on the same set of syllables, but during training presented them one a time. That is, the models were trained to have a working memory span of one, much like a model of word naming that is trained to produce a single output given a single input. In this condition the models do not need to learn a lexical bias in order to confront the superposition catastrophe. Accordingly, if the poor generalization was the product of a learned bias, performance with novel syllables in both models should be excellent in Simulations 2a-b. By contrast, if the poor generalization was the product of the small training set, the performance should continue to be poor.

We trained the two models for five million trials on single syllables taken from the same vocabulary of syllables in Simulations 1a-b. This provides a similar amount of training to the amount of training on each syllable in Simulations 1a-b when lists varied on length from one to nine syllables. We then tested the models on the novel syllables taken from Simulation 1a-b. Performance in the two models was much better. The DI- and LI-models achieved 90.9% and 90.5% correct for the novel items, respectively. Note, performance was already excellent after 1 million training trials, with overall performance at 86.6% and 82.4%, in the two models, respectively.

To summarize, the DI- and LI-models can generalize to novel items when trained on a small vocabulary of items one item at a time. However, when models are trained to recall a series of familiar items, a task that requires the model to interpret an ambiguous superposition of distributed patterns, the model fails with the same unfamiliar items. This is a manifestation of what we will call the *generalization-superposition trade-off*: Models with distributed representations can either generalise to new items, or they can solve the superposition catastrophe, but not both. If these models need to co-activate multiple items at the same time over the same set of units, they develop a bias to recall only familiar items, restricting generalization.

Simulation 3a-b

Although the above simulations appear to provide evidence for the superposition trade-off, Botvinick and Plaut (2009) and Bowers et al. (2009b) found that modified versions of the Botvinick and Plaut (2006) model could recall lists of familiar and unfamiliar syllables when they were trained on a larger set of syllables. Accordingly, this trade-off may only be expressed under limited (and unnatural) conditions in which training is restricted to a small vocabulary.

In Simulation 3a we took the same DI-model as in Simulation 1a and trained it to recall lists of items taken form a vocabulary of 300 syllables (that is, syllables defined by a random pattern of 3 input and output units). An additional random set of 300 syllables constituted the novel syllables for testing purposes. In Simulation 3b we took the same LI-model as in Simulation 1b, and trained it to recall lists of syllables taken from a vocabulary of 300 (out of a possible 600 syllables). The remaining 300 syllables were defined as the novel syllables for testing.

In both cases, training was carried out for five million trials on lists of familiar syllables (with lists varying from one to nine syllables), and at test, we assessed the models' performance on 1000 lists of familiar or unfamiliar syllables, with lists varying in length from between one to six items. The models' response was defined as the syllable most similar to the output produced by the model (when considering the 300 familiar and the critical novel syllables), based on the dot-product measure. As can be seen in Figure 4, the DI-model trained on 300 syllables performed equally well on lists of familiar and unfamiliar letters. In addition, as can

be seen in Figure 5, the LI-model trained on 300 syllables performed similarly with lists of familiar and unfamiliar syllables (although here there was a slight advantage for the familiar items for the longer lists). These results are similar to Bowers et al. (2009b) where performance was the same for familiar and unfamiliar syllables after the LI-model was trained on 500 out of the possible 600 syllables.

To summarize, the DI- and LI-models performed radically differently with the novel items in the different conditions. When trained on a small vocabulary of 26 syllables they catastrophically failed with novel syllables (Simulations 1a-b), and when trained on 300 familiar syllables, they recalled lists of unfamiliar syllables almost as well as familiar ones. Accordingly, recall of novel syllables depends on the size of the training set and not the nature of the input coding scheme (distributed vs. localist).

How has a PDP model overcome the generalization-superposition trade-off?

Based on the above findings it appears that the DI- and LI-models can overcome the generalization-superposition trade-off when trained on a large vocabulary. However, there are two quite different conclusions that can be drawn from this: a) either there are no fundamental constraints associated with co-activating multiple things with distributed representations, or b), the models abandoned dense distributed representations in favour of sparse or localist ones when trained on a large vocabulary of syllables, effectively avoiding (rather than solving) the superposition catastrophe. Such a finding would not only highlight the computational limitations of distributed coding schemes, but also provide a computational argument in support of sparse or localist codes. Indeed, the results would provide a computational explanation as to why neurons in the brain (both hippocampus and cortex) respond so sparsely and so selectively to inputs (Bowers, 2009).

When considering any potential changes in the learned representations in the different training conditions, it is useful to consider a distinction drawn by Plaut and McClelland (2010a). They argued that representations vary along two dimensions, namely, perplexity and sparseness. On the perplexity dimension, the interpretability of individual units varies. On one extreme, it is possible to interpret the output of a single unit unambiguously. We would label this a localist (grandmother) representation. On the other extreme, each unit responds to a wide range of inputs, such that it is impossible to interpret the output of a given unit (the response is perplexing). The sparseness dimension is conceptually distinct. Here, the proportion of active units in a network varies. On one extreme a single unit is active at a time, and at the other extreme, a high proportion of units are active at any point in time.

It is possible that PDP models learn various combinations of sparseness and perplex representations, depending on the details of the model and the training conditions. For example, in one condition a model might learn highly perplex and dense representations (so-called dense distributed representations), and in another condition, learn interpretable and sparse representations (in the limit, a single localist or "grandmother" representation is active, and nothing else). At the same time, another PDP model might learn sparse representations that are highly perplex (that is, relatively few units are active at a given time, but it is not possible to interpret the output of any given item). This would constitute a distributed representation, but the sparseness might have some functional implications for the model's performance. Yet another model might learn interpretable but dense representations (that is, multiple units might all redundantly code for the same thing). We would call these units highly redundant local (grandmother) representations.

The question here is what coding schemes have been employed by the DI- and LI-models in order to solve the superposition-generalization trade-off. If the models can co-activate multiple items and generalize to novel items employing dense distributed representations, then the superposition-catastrophe does not pose a fundamental constraint to theorizing.

Analysis of LI model

First we carried out an analysis introduced by Berkeley, Dawson, Medler et al. (1995). On this analysis, a separate scatter plot for each hidden unit is constructed, with each point in a scatter plot corresponding to a unit's activation in response to one input (e.g., a syllable). All the relevant inputs can then be presented to the network, and the response of each hidden unit recorded. Level of activation is coded along the x-axis, with a random value assigned to each point on the y-axis (in order to prevent points from overlapping). This effectively provides a singe-cell (or in the case, a single unit) recording for each hidden unit to a large range of inputs.

We carried out this analysis on the LI-model in four conditions: When the model was trained on 26 and 300 familiar syllables presented one at a time, and when the model was trained on 26 and 300 familiar syllables presented in lists. We then plotted the activation of the hidden units in response to single syllable, with all the syllables in the vocabulary presented once. Figure 6a-b presents the scatter plots of the first 30 hidden units (out of 200) when the model was trained on a vocabulary of 26 and 300 syllables one syllable at a time. As is clear from these plots (and equally true of the remaining plots not shown in the Figure), many units are active in response to a given syllable, and it is not possible to interpret the output of any hidden unit given that each unit responded in the same way to many different syllables. That this, the model has learned to repeat syllables on the basis of dense distributed representations.

Next consider the LI-model trained on lists of syllables (a model of STM). Figure 7a-b presents the scatter plots of the first 30 hidden units (out of 200) when the model was trained to recall lists of syllables taken from a vocabulary of 26 and 300 syllables. As above, the scatter plots reflect the activation of the units to a single test syllable. When the model was trained on 26 syllables the model continued to rely on a dense distributed coding scheme. However, the plots look very different when the model was trained on a vocabulary of 300 syllables. That is, many of the units did not respond to any input, but a subset of units responded in a systematic manner to a subset of the syllables, as reflected in a banding pattern.

This banding pattern was first reported by Berkeley et al. (1995), and it is often possible to identify what a given hidden unit codes for by considering all the inputs that contribute to a given band. In the present example, consider hidden unit 64 that has a clear banding pattern. The inputs that produced this pattern are as follows: *HUR, MYR, KIR, GAR, KYR, LAR, JAR, COR, CIR, CUR, LOR, KAR, K0R, MAR, CAR, BER, JUR, DUR, JER, LYR, MIR*. As is clear from this list, this unit responds in a systematic manner to the letter R (all 21 syllables that contained the letter R are found in the band, and no other syllables are). The tempting conclusion is that this unit is a localist representation for the letter R.

Figure 8a-b characterizes the selectivity of all 200 hidden units when the model was trained to recall lists of syllables taken from the small and large vocabulary, respectively. In order to fit all the data on a single figure we presented the LI-model with single letters rather than syllables (so that each scatter plot includes 26 rather than 300 data points), and we dropped the y-axis (that only plotted random scatter for the sake of distinguishing the 300 data points). Furthermore, rather than labelling each letter with a dot, we presented the letters themselves (letters with the same level of activation are illegible, as they overlap with one another).

Figure 8b provides converging evidence that hidden units are selectively responding to letters following training on lists of syllables taken from a large vocabulary. Indeed, most letters are selectively associated with one (often more) hidden units. In some cases selective responding was characterized by an increased response to a given letter, and in other cases, by a decrease in firing. For example, units 4 and 22 selectively responded to the letter S through their increased activation, and unit 28 selectively responded to S through its decreased activation. In Table 2 we have provided a rough summary of how each letter induced selective responses in the hidden units. We defined a unit's response to be selective if its activation to

one letter was .2 different compared to all other letters, and defined a unit's response as useful if its activation was consistent with only a few letters.

It is important to note that the same set of selective units can be identified when we presented the LI-model with syllables (that is, we activated 3 input units at a time; Figure 7b) and single letters (that is, we activated 1 input unit at a time; Figure 8b). For example, in Figure 7b, hidden unit 4 showed a banding pattern composed of a set of syllables containing the letter S, and in Figure 8b, the selectivity of this same unit can be seen by its selective response to the letter S (this close correspondence was found throughout). This highlights the fact that the model analyzed the syllables through their component letters (rather than as complex holistic patterns). If the activation of these hidden units were the product of many co-active input units, then we would not find the units to show the same selectivity in response to syllables and single letters.

Although the model appears to have learned a number of localist representations for letters (e.g., unit 64 for R), it is important to note that not all letters were associated with a given unit so unambiguously (e.g., there is no unit selective for the letter H). In these cases, the model was presumably relying on some version of distributed coding in order to code the letter. Still, in these cases, the level of distribution is much reduced compared to the condition in which the LI-model was trained on letters one at a time (Figure 9). That is, although some letters were not associated with a selective unit, the identity of every letter could be highly constrained by considering the activation of a single unit (e.g., when unit 67 is highly active, the letter is either a B or K). Accordingly, the co-activation of a relatively small number of these units could code for a letter. By contrast, little information can be gathered from the output of a single unit when the model was trained on one syllable at a time. The implication is that the superposition catastrophe poses a pressure to learn highly selective representations, but that selectivity is not always at the extreme level of a local (grandmother cell) representation.

One surprising outcome of these analyses is that some units selectively responded to a given letter by decreasing their activation. For example, the reduced activation of the hidden unit 14 corresponded with the letter V. This raises two obvious questions: (1) How does a selective reduction in activation function to activate the correct output? (2) Is this a localist grandmother unit for V? With regards to the first question, the reduced activation of the hidden unit 14 in response to the letter V (input unit 23) presumably reflects a learned inhibitory connection between the input unit 23 and hidden unit 14. If hidden unit 14 is in turn connected to the output unit 23 through an inhibitory connection, then turning off hidden 14 would facilitate the activation of output unit V (through disinhibition). If this is the correct characterization of these units, then it is unclear whether or not to call them grandmother cells. On the one hand, it is possible to determine the identity of the input based on the output of a single unit. This sounds like a grandmother cell. On the other hand, this unit alone cannot drive the model to output a letter. For instance, selectively turning off hidden unit 14 only provides a "permission" for V to be output – the V output unit still needs positive inputs from other sources. If a grandmother cell needs to drive behaviour by itself, then units that code for information through decreased activation do not satisfy the definition. It is nevertheless an interesting type of selective response that may have some correspondence in the cortex given the prevalence of inhibitory connections in the brain.[1]

The analyses thus far have focused on the interpretability of hidden units, and they suggest that selective (sometimes local) codes play a causal role in the solution to the superposition catastrophe. However, it is also worth considering the role (if any) that sparseness plays. As noted above, perplexity and sparseness are distinct dimensions, and it is possible that the learned representations were also highly sparse, and that these contributed to the solution as well.

In order to assess the potential role of sparseness to the solution we plotted the activation of the hidden units in response to the 26 letters presented individually (as in the analyses depicted in Figure 8ab). However, in Figure 10ab, each scatter plot corresponds to a single letter (with 26 plots in all), and each point in each plot corresponds to the activation of a different hidden unit (with 200 data points in all). Once again, activation is plotted on the x-axis, and the value on the y-axis is random in order to avoid units overwriting each other. This, in contrast to the figures above, provides a depiction of how many hidden units are active in response to a given input, and to what extent. In Figure 10a we have plotted the activation of hidden units after the model was trained to recall lists of syllables taken from the small vocabulary (that is, when the model did not solve the superposition catastrophe), and in Figure 10b we plotted the corresponding activations for the model trained on the large vocabulary (that is, when the model did solve the superposition catastrophe).

The key result of this analysis is that there is no obvious difference in the plots in the two models. That is, the level of sparseness of the two models is similar despite the striking contrast we observed when we measured the selectivity of hidden units in the two models. Indeed, when the model was trained on the small vocabulary, 25% of the hidden units were activated beyond 0.1 (out of 1), whereas when the model was trained on the large vocabulary, 29% of the hidden units were activated beyond 0.1. That is, if anything, when the model learned localist representations it coded information in a less sparse format. This clearly indicates that it is the selectivity, not the sparseness, of units that is required to solve the superposition catastrophe. Interestingly, sparseness in the LI-model was also similar when trained on a small (28%) and large (31%) vocabulary of syllables one at a time.

This analysis also highlights the importance in distinguishing the selectivity of neurons and the sparseness of neural coding when evaluating the neural plausibility of grandmother cells. Grandmother cells are the neurobiological implementation of localist representations in cognitive theories, and as such, they constitute a theory of the selectivity of single neurons in coding for particular things (words, objects, faces, etc.). Nevertheless, grandmother cell theories have often been rejected as implausible on the basis of analyses that suggest that many (perhaps millions) of neurons respond to a given input (e.g., Waydo, Kraskov, Quian Quiroga, Fried, & Koch, 2006). However, as demonstrated here, when the LI-model learned highly selective representations of letters, it did not learn corresponding sparse representations. Indeed, the sparseness in our model (~25% of the units responding to a given input) was much lower than the sparseness in medial temporal lobe (<0.1% of neurons responding to an image; Waydo et al., 2006), and at the same time, it would be a mistake to reject our conclusion that the model has learned localist (grandmother cell) representations of letters. This makes it more difficult to distinguish between distributed and grandmother cell theories of brain function, but the distinction is nevertheless important, and highlights the need to make estimates of selectivity rather than sparseness of neural firing in the brain (cf. Bowers, 2009, 2010ab; Foldiak, 2009).

In the above analyses we characterized the activation patterns of hidden units in response to inputs. That is, the analyses assessed how information is coded in STM in the LI-model in the various training conditions. However, these activation patterns depend on the learned connection weights in the network; that is, how information is coded in long-term memory (LTM). Accordingly, when the LI-model learns localist representations of letters it should be possible to identify corresponding connection weights that mediate the model's performance. For instance, the letter Q in the network was coded by the input and output units 19, and the hidden unit 40. Accordingly, it should be predicted that there are strong and interpretable connections between these units. By contrast, when the LI model failed to develop localist representations (when trained on a small vocabulary of items), these interpretable connection weights should be lost.

In Figures 11 and 12 we depict the strength of the connection weights between input and hidden units and between hidden and output units when the LI-model was trained on lists of syllables taken from the small and large vocabulary, respectively (the Figures only include the connections between the input units and the first 75 hidden units). Just as with the activation plots (Figure 7ab) the connection weights were only interpretable when the model was trained on a large vocabulary. And these connection weights have exactly the characteristics that should be predicted given the selectivity of the activation (e.g., input unit 19 has a strong positive connection to hidden unit 40, which in turn has a strong positive connection to output 19). Consistent with our explanation regarding the role of the selective non-active hidden units, these units were connected to both input and output units through inhibitory connections (e.g., V is coded with input unit 23, which has an inhibitory connection to hidden unit 14, which in turn has an inhibitory connection to output unit 23). Accordingly, turning off hidden unit 14 allows output unit 23 to become active. In Figure 12 we've labelled a few of the strong positive and negative connection weights that are devoted to coding a specific letter.

The connection weights displayed in Figure 12 also help explain how the network can learn units that are selectively activated by a given letter, and at the same time, not affect the overall sparseness of activation. Consider the row 40 that depicts all the connection weights between the input units and hidden unit 40. It is clear that most of the links are weak, apart from the link to input unit 19. This explains why hidden unit 40 selectively codes for the letter Q. Next, consider column 19 that depicts the connection weights between input unit 19 and hidden units 1-75. Here is it clear that input 19 is also connected to a number of hidden units (e.g. 46), which will result in multiple hidden units responding to a given input (resulting in a non-sparse representation of Q).

Analyses of the distributed network

As with the LI-model, we first analyzed the activation of the hidden units of the DI-model by computing a scatter plot for each hidden unit in response to all the syllables presented one at a time. Figure 13ab depicts the plots from the first 30 units when the model was trained with 26 and 300 distributed syllables one at a time, and Figure 14ab depicts the corresponding plots when the model was trained on lists of syllables. In one respect, the results are just the same as those of the LI model. That is, the DI model learned dense distributed representations in all conditions but one, namely, when trained to recall sequences of syllables when taken from a large vocabulary. In this latter condition, many of the hidden units showed a banding pattern, again highlighting the conclusion that PDP networks need to code for information in a selective manner in order to address the superposition catastrophe. Furthermore, the inputs associated with a band are systematically related. For example, consider the set of syllables that are part of the band in unit 14 (Figure14b). The syllables are: 1-2-4, 1-2-10, 1-2-12, 2-3-6, 2-3-12, 2-3-14, 2-3-15, 2-4-5, 2-4-5, 2-4-15, 2-4-17, 2-5-7, 2-5-15, 2-6-8, 2-6-13, 2-7-13, 2-8-10, 2-8-16, 2-10-20, 2-10-26, 2-11-19, 2-12-16, 2-12-17, 2-12-19, 2-12-24, 2-13-18, 2-14-26, 2-15-21, 2-16-23, 2-16-25, 2-16-26, 2-17-23, 2-18-24, 2-19-23, 2-22-26, 2-23-24. That is, this unit appears to be an "input unit 2 detector". As with the LI-model, banding patterns were associated with most input units.

However, there is an important difference in the selective representations learned in the LI- and DI-models. It was possible to unambiguously interpret many of the hidden unit in the LI-model because each input unit was itself associated with a letter. For instance, the second input unit in the LI model codes for the letter C, and accordingly, hidden units that selectively respond to this input unit were described as localist representations for the letter C. But in the DI-model, input unit 2 is not associated with anything meaningful. For instance, in Table 1, input unit 2 in the DI coding scheme is associated with the following collection of three syllables: CIT, BAR, and JAW. These syllables were just randomly assigned to their input patterns, and accordingly, there is no meaningful interpretation for input unit 2. In the same

way, the 36 input patterns above are associated with a random collection of syllables, and accordingly, input unit 2 has no interpretation (by design).

We next assessed the sparseness of the DI-model (as opposed to the selectivity of individual units) when it was trained to recall lists of items taken from a small and large vocabulary. As above, we presented the networks with a single letter and measured the level of activation of all the hidden units. Just as with the LI-model, the level of sparseness was similar in the two training conditions, with 28% and 31% of the hidden units activated beyond 0.1, respectively. So once again, the superposition catastrophe was solved by learning selective rather than sparse representations of the inputs.

The implication seems to be that the DI-model has solved the superposition catastrophe by learning representations that are selective but at the same time, uninterpretable. What type of representation is this? In one sense the learned representations appear to be localist – the hidden units often respond in selective and predictable ways (e.g., hidden unit 14 is only on in response to input unit 1). However, in another sense, the representations appear to be distributed, as the individual hidden units are themselves perplexing (hidden unit 14 is associated with a random pattern of syllables). As detailed below, the problem in characterizing this type of representation is that we have not made enough distinctions regarding the types of representations that PDP models can learn. That is, it is not enough to categorize representations along the dimensions of sparseness and perplexity. Another dimension, sometimes called "explicitness", needs to be considered as well (cf. Foldiak, 2009). An explicit representation maps onto meaningful categories in the world, whereas implicit representations do not. On our view, the DI-model has in fact learned "implicit local" representations, whereas the LI-model has learnt "explicit local" representations. On this account, in order to solve the superposition catastrophe it is necessary to learn local (or highly selective) representations, but explicitness is irrelevant. This argument is described in detail below.

General Discussion

PDP theories of cognition are associated with a fundamental claim; namely, knowledge is coded in a distributed rather than a localist format. Indeed, it is commonly argued that the brain relies on two different types of distributed representations, namely, sparse distributed representations in the hippocampus, and dense distributed representations in the cortex. This so-called *complementary learning system hypothesis* was advanced based on contrasting computational limitations of sparse and dense distributed representations. That is, sparse representations can learn fast, but are poor at generalizing, and dense distributed representations can generalize but cannot learn quickly without erasing previous knowledge (dense distributed representations suffer from catastrophic interference).

The present set of simulations highlight another computational constraint; namely, dense distributed representations cannot support the co-activation of multiple things at the same time (due to the superposition catastrophe). The problem is that co-active things (e.g., words) in a dense distributed coding scheme produce blend patterns that are ambiguous. In order to overcome this ambiguity, highly selective representations need to be learned, and strikingly, this is exactly what is learned in LI- and DI-models of STM. Indeed, the models often learned localist representations of letters. Note, this same computational constraint applies to perceptual, semantic, and language systems given that they all support the co-activation of knowledge (cf. Cowan, 2001). Accordingly, we conclude that the cortex needs to learn highly selective representations in order to solve the superposition constraint, much like the hippocampus needs to learn sparse representations in order to address catastrophic interference. This helps explain why neurons in cortex do in fact respond so selectively to high-level perceptual categories, such as objects and faces (cf. Bowers, 2009).

Summary of findings

These conclusions are supported by a series of simulations in which we employed two different versions of the Botvinick and Plaut (2006) model of STM: One version that included a distributed input coding scheme (the so-called DI-model), another with a localist coding scheme (the so-called LI-model). In both models we identified two manifestations of the superposition catastrophe.

First, when the DI- and LI-models were trained to recall a series of syllables taken from a small vocabulary (26 syllables), the models catastrophically failed in recalling single *novel* syllables (Simulations 1a-b). In both cases, STM for novel items was ~ 0 items (also see Bowers et al., 2009ab). This failure was not due to the small training set, as the two models succeed with novel syllables when they were trained to recall one syllable at a time rather than lists of syllables (Simulations 2ab). That is, the DI- and LI-models could either recall a single novel syllable (generalization), or recall multiple familiar syllables (solve the superposition constraint), but not both. We labelled this the *generalization-superposition trade-off*.

The cause of this trade-off was in fact identified by Botvinick and Plaut (2006). That is, they noted that a superposition of distributed patterns can often be decomposed into the correct set of constituent patterns by adopting a bias; namely, recalling the sequence of trained patterns that most likely produced the ambiguous blend. Just like the inverse problem in vision is solvable with the inclusion of innate or leaned biases, the DI- and LI-models learned a bias that provided a (partial) solution to the superposition catastrophe. However, Botvinick and Plaut did not consider the down side of this solution, namely, that the bias prevents the model from recalling *novel* items. A sequence that includes a novel syllable is not the most likely solution to the blend, so the model selects the most plausible (but incorrect) sequence of familiar items.

The second manifestation of the trade-off is observed with *familiar* items. That is, when the DI- and LI-models were trained on a larger vocabulary, the bias to retrieve only familiar syllables was no longer sufficient to support good performance with familiar items, as there are too many possible sequences of the familiar patterns that will produce a given blend. Under these conditions, the models gave up on dense distributed coding (the source of the problem), and instead, learned representations that responded highly selectively to inputs. Indeed, the LI-model appeared to learn some localist representations, with single units devoted to a specific letter (e.g., unit 64 coding for the letter R). The behavioural manifestation of this selective responding was dramatic: Performance of the LI- and DI-models went from catastrophic failure with novel syllables to striking success (with performance the same for lists of familiar and unfamiliar syllables).

Although both the DI- and LI-models solved the superposition catastrophe, there are interesting differences between the models. We turn to these differences next.

<u>Comparing the DI and LI-models</u>

The LI- and DI-models solved the superposition catastrophe by learning localist representations in the hidden layer that respond highly selectively to inputs, but nevertheless, the representations in the two models are quite dissimilar. That is, in the LI-model, the hidden units were often interpretable in a semantically meaningful fashion (e.g., hidden unit 4 codes for an S), whereas the hidden units in the DI-model did not correspond to anything meaningful in the world (they were associated with random collections of syllables). Accordingly, the hidden units in the DI-model shared some key characteristics with localist representations (they are selective) and with distributed representations (they do not represent anything meaningful).

When considering how to best characterize these two types of representations, it is important to realize that the interpretability of single hidden units in the DI- and LI-models has nothing to do with the models themselves. Indeed, these two sets of representations can be exactly the same, and at the same time, differ in how we interpret them. To see this, consider the following thought experiment. Imagine that we trained the DI-model with the same set of 300 syllable patterns as used in the LI-model (that is, the inputs follow the same phonotactic

constraints of the LI-model, with one active unit from the first 10 units, one active unit from the next 6, and one active unit from the final 10 units). The only difference is that we randomly reassign the input patterns and the syllable names in the DI-model. For instance, in the LI-model, the letters B, C, A, I, O, Q, and R were coded by the input units 1, 2, 11, 13, 14, 19, and 20 (localist letter units), and accordingly, the syllables BAQ, BIQ, and COR were defined by the co-activation of units {1, 11, 19}, {1, 13, 19}, and {2, 14, 20}, respectively. In this way, similar syllables names are coded with overlapping units (syllables BAR and BIQ are coded by two overlapping letter units). In the thought experiment, we assign these same patterns to syllable names randomly, so now BAQ might be coded by {2, 14, 20}, BIQ by {1, 11, 19}, and COR by {1, 13, 19}. Admittedly this is an odd coding scheme in which similar syllable names are often coded by unrelated patterns (e.g., BAQ and BIQ are unrelated in the input layer), and dissimilar syllable names can be coded by similar patterns (e.g., BIQ and COR overlap in two out of three input units). But the critical point for present purposes is that the LI- and the DI-models include the same set of syllable names and syllable patterns, but the DI-model includes a distributed input coding scheme given that it is not possible to interpret any unit in isolation (e.g., unit 1 is involved in coding many unrelated syllables).

How will the two models compare? The answer is straightforward: They will perform in exactly the same way, and they will learn exactly the same internal representations, as they are the same model. We don't even need to run the two simulations, as they both are trained on the same set of input patterns in the same number of times. So, the conclusion that one model learns localist and one model learns distributed representations is not actually a claim about differences in the model, but rather, a claim about how the model relates to the world (in this case, the mappings between the input units and the letter names). [2]

This insight raises a new possibility regarding the conditions in which models succeed or fail to address the superposition catastrophe. That is, the solution to the superposition catastrophe may still rely on learning highly selective and interpretable hidden units, but interpretable from the perspective of the model, not the modeller. Indeed, from the model's point of view, the hidden units in the LI- and DI-model are equally interpretable. For example, in both cases, input unit 1 might selectively activate hidden unit 2, which in turn selectively activates output unit 1. As long as the model learns one-to-one mappings between input and hidden units, and a corresponding one-to-one mapping between hidden and output units (or at least highly selective mappings), then the superposition catastrophe is avoided.

These considerations suggest that Plaut and McClelland's (2010) description of the different types of representations that are learnable in PDP networks needs to be amended. They argued that representations can vary along two dimensions, sparsity (the proportion of units that are active by a given input) and perplexity (the number of different things in the world that a single unit represents). But this categorization does not capture the representations learned in the DI-model in which the single units are perplex with respect to the observer, but transparent with respect to the model. In order to capture all the relevant types of possible representations in neural networks, the dimension "explicitness" needs to be added (cf. Foldiak, 2009). Explicitness refers to the nature of the mapping between representations in a model and meaningful categories in the world. In an explicit local representation, a representation is local to the model (e.g., hidden unit 4 is selectively active in response to input unit 1 being active) and an observer can interpret units meaningfully because the model learns to map units to meaningful categories in the world (as in the LI-model), whereas in an implicit local representation, a representation is local to the model (e.g., hidden unit 4 is selectively active in response to input unit 1 being active), but an observer cannot interpret units meaningfully because the model has learned mappings between units and meaningless categories in the world. On this view, in order to overcome the superposition catastrophe, models rely on learning either explicit or implicit local (or highly selective) codes.

But what type of local code is used by the brain? It is relevant to note that explicit local codes were learned when the input units were related to meaningful things in the world (e.g., input unit 1 in the LI model responded to the letter B), and implicit local codes were learned when the input units were not associated with meaningful categories (e.g., input unit 1 in the DI-model responded to a collection of unrelated syllables). That is, in both models, the explicitness of the hidden units mirrored the explicitness of the input units. Indeed, it is hard to see how hidden units could develop explicit representations based on inputs from implicit units. Thus it is relevant to note that single neurons in primary perceptual systems respond to semantically meaningful categories in the world. For example, simple cells in primary visual cortex selectively respond to lines at a given orientation in a given position in space (and not some arbitrary unrelated set of line orientations). If high level representations in the brain inherit the explicitness of their inputs, then high-level local representations should be explicit, not implicit. And indeed, one of the striking findings in single cell neurophysiology is that neurons in high-level visual systems also respond highly selectively to meaningful categories in the world (cf., Bowers, 2009).

One noteworthy property of the learned explicit localist (grandmother) representations in the LI-modal is that they corresponded to letters rather than syllables. There was no evidence that an individual unit was involved in coding a complete syllable, for instance. Why is this? In our view, the models learned localist representations for letters because the input-output mappings were entirely systematic, and under these conditions, mappings between individual input-output units were sufficient to solve the superposition catastrophe. However, these same representations may not suffice if the input-output mappings were more arbitrary. We suspect that the joint requirement to perform arbitrary mappings (e.g., mappings between phonology and semantics) while coding multiple things at the same time would result in conditions in which localist word (or syllable) representations result.

In sum, the superposition catastrophe provides a computational pressure to learn highly selective (perhaps local) representations. Furthermore, given that low-level perceptual systems code for information in an explicit fashion, there are good reasons to assume that the brain learns explicit local representations. In addition, it is worth noting there are physiological constraints that provide a strong pressure to learn sparse representations. That is, the metabolic cost of firing neurons is high. Lennie (2003) estimated that these costs restrict the activation of neurons in cortex to about 1% of neurons concurrently (compared to the ~20% of hidden units that are active in response to an input in the present models). Together, these computational and physiological constraints may produce highly selective and highly sparse representations of meaningful things in the world.

Generalization in the LI- and DI- models

It is often argued that PDP models with dense and sparse distributed representations are good and poor at generalizing, respectively. This is the logic of the complementary learning systems hypothesis according to which generalization is mediated by networks in the cortex, and episodic memory is mediated by networks in the hippocampus (McClelland et al., 1995). However, it is interesting to note that generalization in the current simulations did not follow this prediction. That is, the level of sparseness in the LI- and DI-models were similar when they were trained to recall a series of syllables taken from a small and large vocabulary of syllables, but generalization to novel syllables was much better in the latter case.

Why did the networks trained on a large vocabulary generalize so much better? As noted above, the improved performances cannot be attributed to a general principle that networks generalize better when trained on more items: The LI- and DI-models generalized just fine when trained on a small vocabulary of syllables as long as they were trained to recall one syllable at a time (Simulations 2ab). Rather, the good generalization was a product of the DI-

and the LI-networks learning highly selective and localist representations of letters when trained on a large vocabulary.

To appreciate the relevance of local coding, it is important to note that generalization in PDP (or any network) relies on the extent to which new and old items share overlapping representations. In the absence of any additional constraints, the overlap between two patterns will be greater for dense compared to sparse representations (just by chance). This presumably is why dense distributed representations in PDP networks tend to generalize better. However, the local representations learned in the LI- and DI-models ensure that similar things (from the model's point of view) are coded with overlapping representations. For example, in the DI-model, the network learned to selectively map between input unit 3, hidden unit 34, and output unit 3. Accordingly, any novel input that activates input unit 3 will overlap strongly with pre-existing knowledge (regardless of the sparseness of the model). An obvious demonstration that local but sparse coding schemes can generalize comes from models of word (and nonword) naming. The DRC model of Coltheart et al. (2001) relies on local coding of letters and words, and level of sparseness in the model is extremely high (more so than here). Nevertheless it generalizes well to novel items. Accordingly, there is no conflict in learning selective (even sparse) representations in response to the superposition catastrophe, and at the same time, maintaining the capacity to generalize. All that is required is a hierarchy of local codes, such that similar inputs activate overlapping units.

That said, not all forms of local coding are equally adept at generalization. For example, there are computational reasons to prefer explicit as opposed to implicit local codes for the sake of generalization. As noted by Hummel (2000) local representations that represent meaningful categories in the world can generalize over these meaningful categories. He gives the example of a network learning to categorize colored shapes, with shape irrelevant but free to vary. So the task might be to put all red shapes in category A. If the network learns to represent color with one unit and shape with another, then the model can learn a mapping from the A-unit to a category-A unit. But if the model learns a localist representation that codes a random collection of things in the world (e.g., *red*, *square*, and *green)*, then generalization is problematic. That is, strengthening the link between this unit and the A-category would not only increase the likelihood of categorizing a new red object as a member of category A, but also of categorizing squares in the same way.

To take a more relevant example, imagine that the LI- and DI-models were trained on a set of syllables that activated unit 1 relatively infrequently. This constitutes a coherent set of syllables in the LI-model (the set of syllables that start with the letter B), but constitute a random set of unrelated syllables in the DI model. In the LI-model this would result in a systematic reduction in performance on the set of syllables starting with a B (compared to other syllables; cf. Bowers et al., 2009), and an impairment on an idiosyncratic collection of syllables in the DI-model (those that happen to include input unit 1). Furthermore, in the case of the LI-model, additional training on a subset of syllables starting with a B would selectively improve performance on all syllables starting with a B. In the case of the DI-model, training on one syllable that includes the unit 1 will improve performance on all syllables that include input unit 1, even those that are phonologically unrelated (e.g., training on BAR would improve performance on DIZ if both syllables include input syllable 1). The former but not the latter type of generalization seems more plausible.

There are also limitations with the explicit localist representations learned in the LI-model. For instance, Bowers et al. (2009b) found that the LI-model trained on a large vocabulary of items failed to recall syllables that included the letter R if the model was not trained with these syllables in a given position. For example, if the model was trained on the syllable BAR in positions 2-9 in a list, but not in position 1, the model would selectively fail on lists that included BAR in position 1 (assuming no other syllable with an R was trained in

position 1). As a consequence, the model could recall the sequence BAR-COR but not COR-BAR, which also seems unnatural. What may be needed in order to generalize in a human-like way are explicit localist representations that are also context independent (so-called *symbolic* representations). That is, the same localist representation should be involved in coding letters or syllables in whatever position in a list, and these representations can be recruited to encode the syllable in any position. Most models of STM have exactly this property (e.g., Burgess & Hitch, 1999; Grossberg, 1978; Page & Norris, 1998), and they do not suffer the generalization constraints of the Botvinick and Plaut model (cf., Bowers et al., 2009ab, but see Botvinick and Plaut, 2009ab).

The contrast between explicit local representations that are context independent (symbolic) and context dependent (non-symbolic) has been studied extensively in other domains, including written word identification (e.g., Davis, 1999; Grainger, Granier, Farioli, Van Assche, & van Heuven, 2006) and semantics (Hummel & Holyoak, 1998; St. John & McClelland, 1990), and the key issue in each case is the relative ability of these models to generalize. Whatever the merits of symbolic and non-symbolic representations, the current findings provide a computational reason to take seriously the hypothesis that the brain learns local (grandmother) representations of meaningful things in the world. As noted by Hummel (2000), this is a first step in learning symbolic models of cognition.

Summary

The current simulations highlight a computational pressure to learn highly selective, perhaps local representations of letters (and meaningful categories more generally). Just as sparse representations are better at learning quickly without suffering catastrophic interference, highly selective representations are better at coding multiple things at the same time. Critically, these representations also supported generalization, highlighting the fact that generalization is not restricted to dense distributed coding schemes often learned in PDP networks.

It is important to emphasize that all PDP models prior to Botvinick and Plaut (2006) activated only one thing at a time over a common set of processing units (no previous model confronted the superposition catastrophe). The current simulations highlight not only how the learned representations in PDP networks are fundamentally restructured in the context of encoding multiple things at the same time (dense distributed representations were replaced with highly selective, and sometimes local representations), but also, that this restructured knowledge impacts on the performance of a model (e.g., generalization to novel syllables often depended on learning localist representations of letters). Given that most cognitive and perceptual systems co-activate multiple things (e.g., Cowan, 2001), this raises concerns regarding existing PDP models in all domains.

References

Berkeley, I. S. N., Dawson, M. R. W., Medler, D. A., Schopflocher, D. P., & Hornsby, L. (1995). Density plots of hidden unit activations reveal interpretable bands. *Connection Science, 7*, 167–186.

Bowers, J. S. (2002). Challenging the widespread assumption that connectionism and distributed representations go hand-in-hand. *Cognitive Psychology, 45*, 413-445.

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

Bowers, J. S. (2010a). More on grandmother cells and the biological implausibility of PDP models of cognition: A reply to Plaut and McClelland (2010) and Quian Quiroga and Kreiman (2010). *Psychological Review.*

Bowers, J. S. (2010b). Postscript: Some final thoughts on grandmother cells, distributed representations, and PDP models of cognition. *Psychological Review.*

Bowers, J. S., Damian, M. F., & Davis, C. J. (2009a). A fundamental limitation of the conjunctive codes learned in PDP models of cognition: Comments on Botvinick and Plaut (2006*). Psychological Review, 116*, 986-995.

Bowers, J. S., Damian, M. F., & Davis, C. J. (2009b). Postscript: More problems with Botvinick and Plaut's (2006) PDP model of short-term memory. *Psychological Review, 116,* 995-997.

Botvinick, M. M., & Plaut, D. C. (2006). Short-term memory for serial order: A recurrent neural network model. *Psychological Review, 113*, 201-233.

Botvinick, M. M., & Plaut, D. C. (2009a). Empirical and computational support for context-dependent representations of serial order: Reply to Bowers, Damian, and Davis (2009). *Psychological Review, 116*, 998-1001.

Botvinick, M. M., & Plaut, D. C. (2009b). Postscript: Winnowing out some take-home points. *Psychological Review, 116,* 1001-1002.

Burgess, N., & Hitch, G. J. (1999). Memory for serial order: A network model of the phonological loop and its timing. *Psychological Review, 106*, 551-581.

Brown, G. D. A., Preece, T., & Hulme, C. (2000). Oscillator-based memory for serial order. *Psychological Review, 107*, 127-181.

Burgess, N., & Hitch, G. J. (1999). Memory for serial order: A network model of the phonological loop and its timing. *Psychological Review, 106*, 551-581.

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,* 204–256.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences, 24*, 87-114.

Davis, C. J. (1999). *The Self-Organising Lexical Acquisition and Recognition (SOLAR) model of visual word recognition.* Unpublished doctoral dissertation, University of New South Wales.

Grainger, J., Granier, J.P., Farioli, F., Van Assche, E., & van Heuven, W. (2006). Letter position information and printed word perception: The relative-position priming constraint. *Journal of Experimental Psychology: Human Perception and Performance, 32*, 865-884.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding, II: Feedback, expectation, olfaction, and illusions. *Biological Cybernetics, 23*, 187–203.

Grossberg, S. (1978). A theory of human memory: self-organization and performance of sensory-motor codes, maps, and plans. In R. Rosen & F. Snell (Eds.), *Progress in theoretical biology* (pp. 233-374). New York: Academic Press.

Hummel, J. E. (2000). Localism as a first step toward symbolic representation. *Behavioral and Brain Sciences, 23*, 480.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review, 104*, 427-466.

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.

Lennie, P. (2003). The cost of cortical computation. *Current Biology, 13*, 493–497.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks. The sequential learning problem. In G. H. Bower (Ed.), *The psychology of learning and motivation*. New York: Academic Press.

Page, M. P. A. (2000). Connectionist modeling in psychology: A localist manifesto. *Behavioral and Brain Sciences, 23*, 443–512.

Page, M. P. A., & Norris, D. (1998). The primacy model: A new model of immediate serial recall. *Psychological Review, 105*, 761-781.

Plaut, D. C., & McClelland, J. L. (2000). Stipulating versus discovering representations. *Behavioral and Brain Sciences, 23*, 489-491

Plaut, D.C., & McClelland, J. L. (2010a) Locating object knowledge in the brain: A critique of Bowers' (2009) attempt to revive the grandmother cell hypothesis. *Psychological Review*.

Plaut, D. C., & McClelland, J. L. (2010b). Postscript: Parallel distributed processing in localist models without thresholds. *Psychological Review.*

Quiroga, R. Q., & Kreiman, G. (2010a). Measuring sparseness in the brain: Comment on Bowers (2009). *Psychological Review.*

Quiroga, R. Q., & Kreiman, G. (2010b). Postscript: About Grandmother cells and Jennifer Aniston neurons. *Psychological Review.*

Ratcliff, R. (1990). Connectionist models of recognition memory - constraints imposed by learning and forgetting functions. *Psychological Review, 97*, 285–308

Seidenberg, M. S., & McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review, 96*, 523-568.

Seidenberg, M. S., & Plaut, D. C. (2006). Progress in understanding word reading: Data fitting versus theory building. in S. Andrews (Ed.), *From Inkmarks to ideas: Current issues in lexical processing*. Psychology Press: Hove, UK.

Sinha, P., & Poggio, T. (1996). The role of learning in 3-D form perception. *Nature, 384, 6608*, 460-463.

St. John, M. F., & McClelland, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence, 46*, 217–457.

Von der Malsburg, C. (1986). Am I thinking assemblies? In G. Palm and A. Aertsen (Eds.), *Brain Theory.* 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.

Footnotes

[1]The possibility that units (neurons) code for information selectively through the absence of firing raises some conceptual and practical difficulties in any search for grandmother cells in the brain. Imagine that a researcher has identified a single neuron that responds selectively to a given face (amongst thousands of faces), and further, activating this single neuron induces a percept to that face. This might appear to be evidence for a grandmother cell representation. But the current observation raises the possibility that in addition to this neuron being active, some other inhibitory neuron also selective to this face must be inactive. If the perception of the face requires the coordinated response of these two selective neurons, then the recording from the activated neuron is misleading. That is, even though the neuron is perfectly selective for this and only this face, the identification does not rely on this neuron alone.

[2] Note, this input coding scheme is uncharacteristic of the coding schemes employed in PDP models. Indeed, most PDP models include localist and interpretable input units. Nevertheless, the distributed coding scheme used in the DI model (and in the thought experiment) serves to highlight the consequences of including input units that cannot be interpreted.

Table 1
The small vocabulary of 26 syllables, and their input coding in the LI and DI model.

26 Syllables	LI-input units	DI-input units
BYX	1, 16, 25	1, 20, 22
CIT	2, 13, 22	2, 6, 8
DIN	3, 13, 17	3, 18, 24
FUN	4, 15, 17	4, 7, 15
GAS	5, 11, 21	5, 16, 17
HAP	6, 11, 18	5, 6, 24
JUR	7, 15, 19	7, 14, 17
KIX	8, 13, 25	1, 8, 25
LER	9, 12, 20	5, 9, 21
MIQ	10, 13, 19	9, 10, 19
BAR	1, 11, 20	2, 11, 21
JEP	7, 12, 18	1, 12, 26
DIW	3, 13, 24	13, 18, 25
KOT	8, 14, 22	6, 11, 14
MUW	10, 15, 24	3, 15, 23
CYV	2, 16, 23	9, 13, 16
CAN	2, 11, 17	11, 12, 17
LEP	9, 12, 18	10, 18, 22
FAQ	4, 11, 19	8, 15, 19
MYR	10, 16, 20	3, 20, 23
FOS	4, 14, 21	13, 20, 21
HUT	6, 15, 22	3, 19, 22
BYV	1, 16, 23	14, 16, 23
JAW	7, 11, 24	2, 24, 26
LEX	9, 12, 25	4, 7, 25
DOZ	3, 14, 26	10, 12, 26

Table 2
Hidden units that selectively respond to single letters in the LI model.

Letter	Hidden unit(s) selectively off	Hidden unit(s) selectively on	Some other highly informative hidden units
A		124	
B			67, 139
C			119
D			119
E	83	144	
F			66
G			139
H		157	181
I	10		
J	37		
K			67
L		158	181
M			
N		103, 127	195
O	62		121, 153
P	123		149
Q		40	149
R		64	
S	28	4, 22	
T		82	
U	56		
V	14	159	196
W		95	
X		147	130
Y	160	164	121
Z			130

Figure Captions

Figure 1. Distributed patterns for the names Paul, John, George, Ringo, Mick, and Keith.

Figure 2. Localist patterns for the names Paul, John, George, Ringo, Mick, and Keith.

Figure 3. Diagram of the Botvinck and Plaut (2006) recurrent PDP model of immediate serial recall. The model includes a set of 27 input and output units (one for each letter of the alphabet plus a unit in the input layer that cues recall, and a unit in the output layer that codes end of list) plus a set of 200 hidden units. Arrows indicate connections between and within layers.

Figure 4. Performance of the DI model trained on 300 words presented in lists, tested on familiar (word) and novel (nonword) items in lists varying from one to six.

Figure 5. Performance of the LI model trained on 300 words presented in lists, tested on familiar (word) and novel (nonword) items in lists varying from one to six.

Figure 6. LI model: Hidden units 1-30 (out of 200) when trained on 26 (top panel) or 300 (bottom panel) syllables one at a time. Within each scatterplot, each dot represents the unit's response to a particular syllable.

Figure 7. LI model: Hidden units 1-30 (out of 200) when trained on 26 (top panel) or 300 (bottom panel) syllables, with list length varying from 1-9. Within each scatterplot, each dot represents the unit's response to a particular syllable.

Figure 8. LI model: Hidden units 1-200 when trained on 26 (top panel) or 300 (bottom panel) syllables, with list length varying from 1-9, and tested on single letters.

Figure 9. LI model: Hidden units 1-200 when trained on 300 input patterns one at a time, and tested on single letters.

Figure 10. LI model: Hidden unit activations as a function of a single letter input when trained on 26 (top panel) or 300 (bottom panel) syllables with list length varying from 1-9. Within each scatterplot, each dot represents a hidden unit's response to a particular letter.

Figure 11. LI model trained on 26 syllables presented in lists: Connection weights between 26 input and 75 of the 200 hidden units; and connection weights between these hidden units and 26 output units. Bright squares represent large positive weight values, dark squares represent large negative values, and gray squares represent intermediate values.

Figure 12. . LI model trained on 300 syllables presented in lists: Connection weights between 26 input and 75 of the 200 hidden units; and connection weights between these hidden units and 26 output units. Bright squares represent large positive weight values, dark squares represent large negative values, and gray squares represent intermediate values.

Figure 13: DI model: Hidden units 1-30 (out of 200) when trained on 26 (top panel) or 300 (bottom panel) training patterns, with list length 1. Within each scatterplot, each dot represents the unit's response to a particular input pattern.

Figure 14. DI model: Hidden units 1-30 (out of 200) when trained on 26 (top panel) or 300 (bottom panel) training patterns, with list length varying from 1-9. Within each scatterplot, each dot represents the unit's response to a particular input pattern.

Figure 1

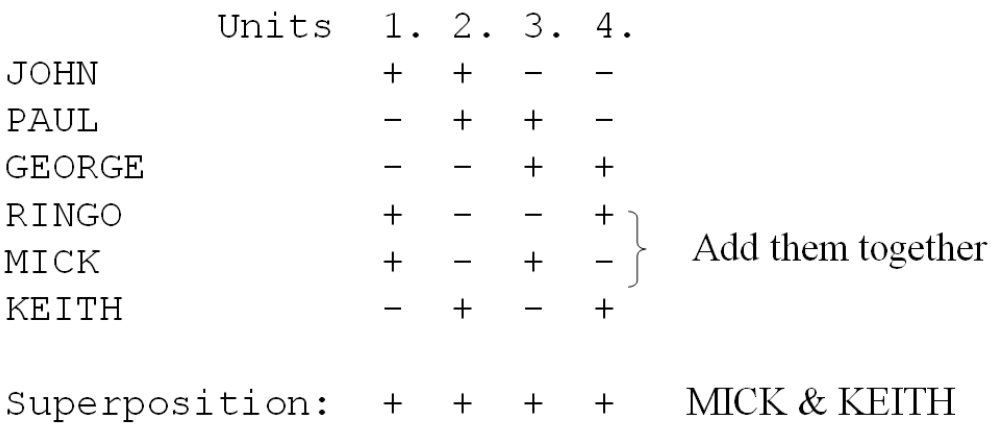

Units 1. 2. 3. 4.
JOHN + + − −
PAUL − + + −
GEORGE − − + +
RINGO + − − +
MICK + − + −
KEITH − + − +
Add them together
Superposition: + + + + MICK & KEITH

Figure 2

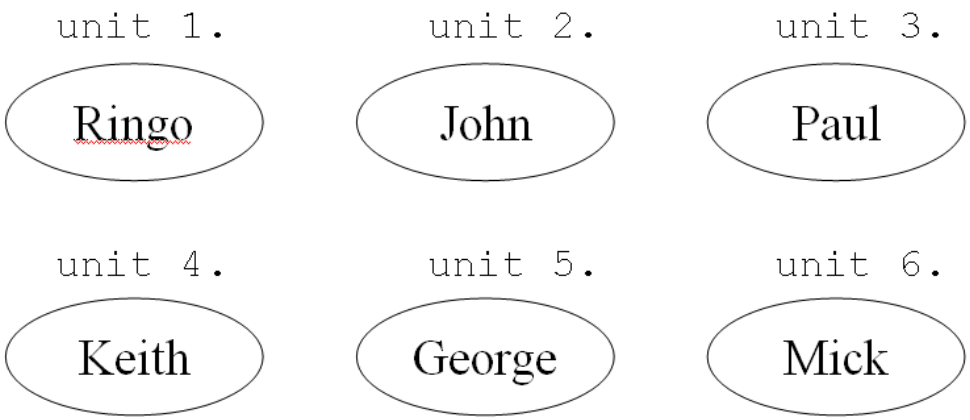

unit 1.
Ringo
unit 2.
John
unit 3.
Paul
unit 4.
Keith
unit 5.
George
unit 6.
Mick

Figure 3

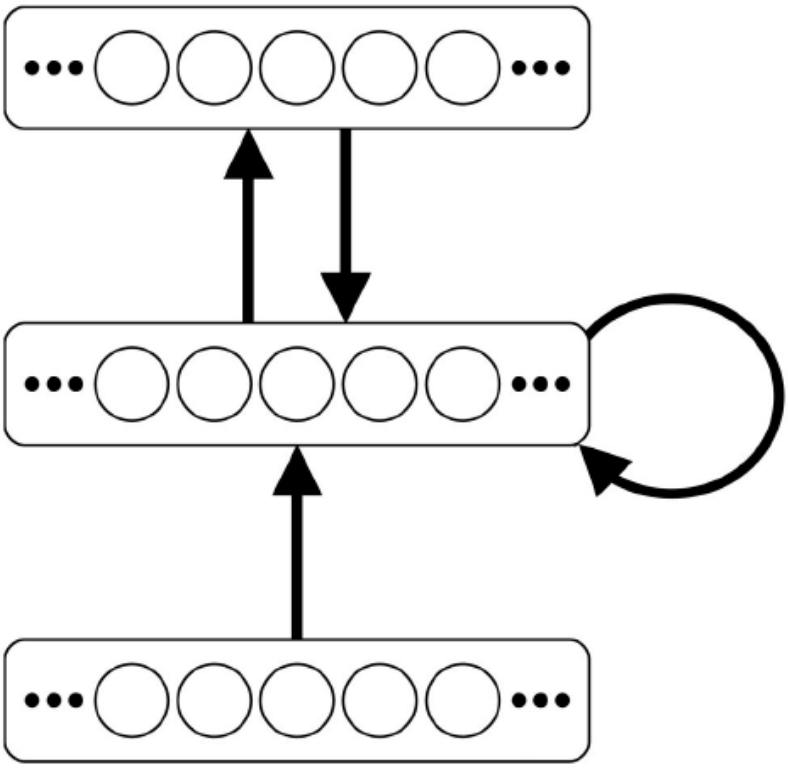


Figure 4

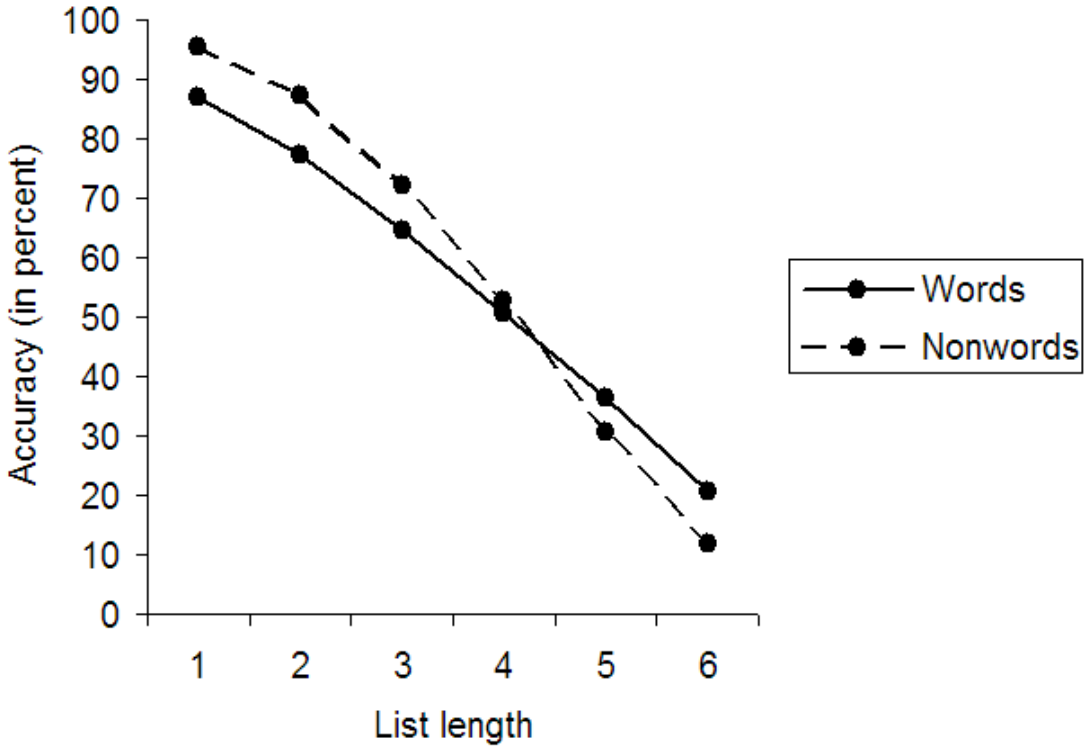

100
90
80
70
60
50
40
30
20
10
0
Accuracy (in percent)
1
2
3
4
5
6
List length
Words
Nonwords

Figure 5

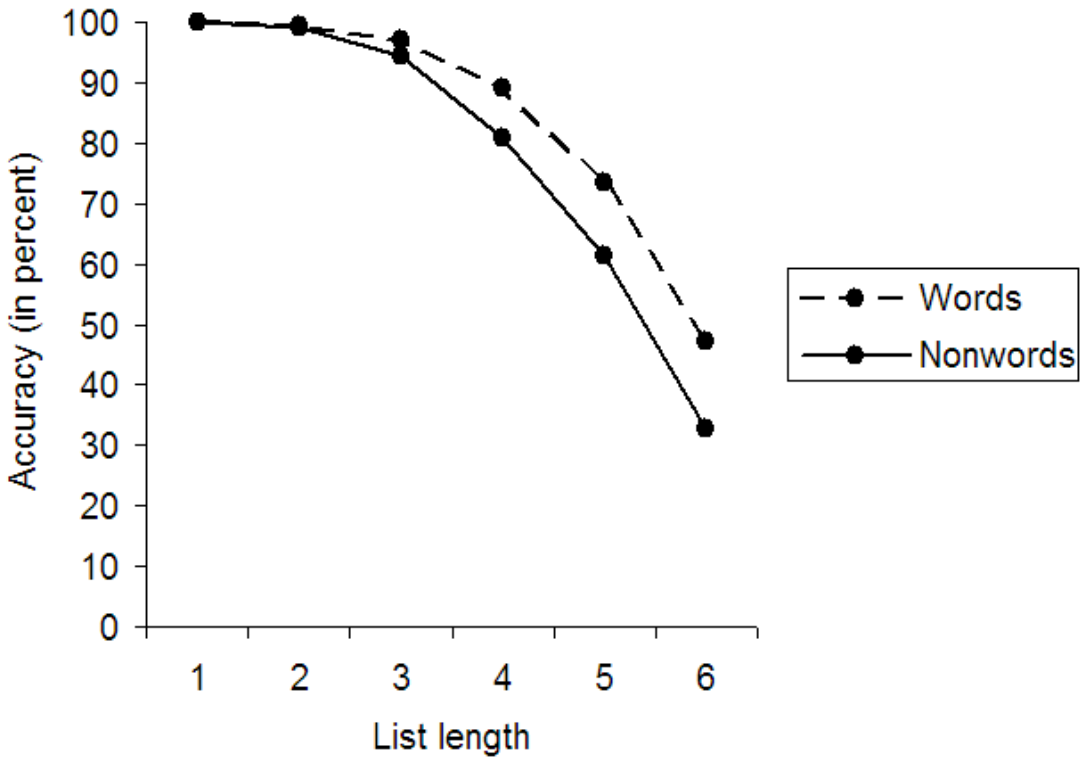
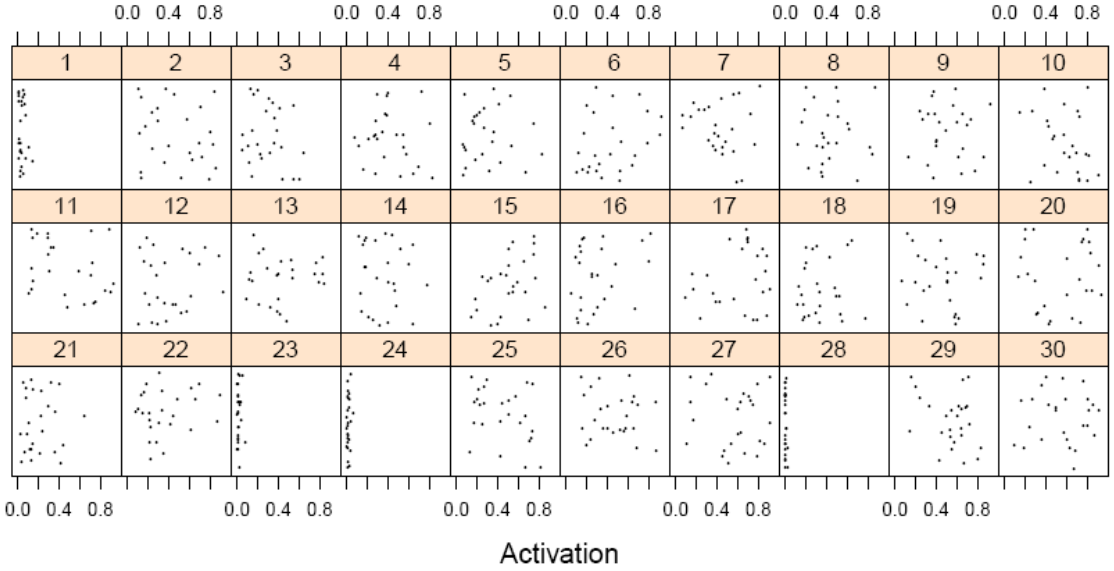

100
90
80
70
60
50
40
30
20
10
0
Accuracy (in percent)
1
2
3
4
5
6
List length
Words
Nonwords

Figure 6


0.0 0.4 0.8
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
Activation

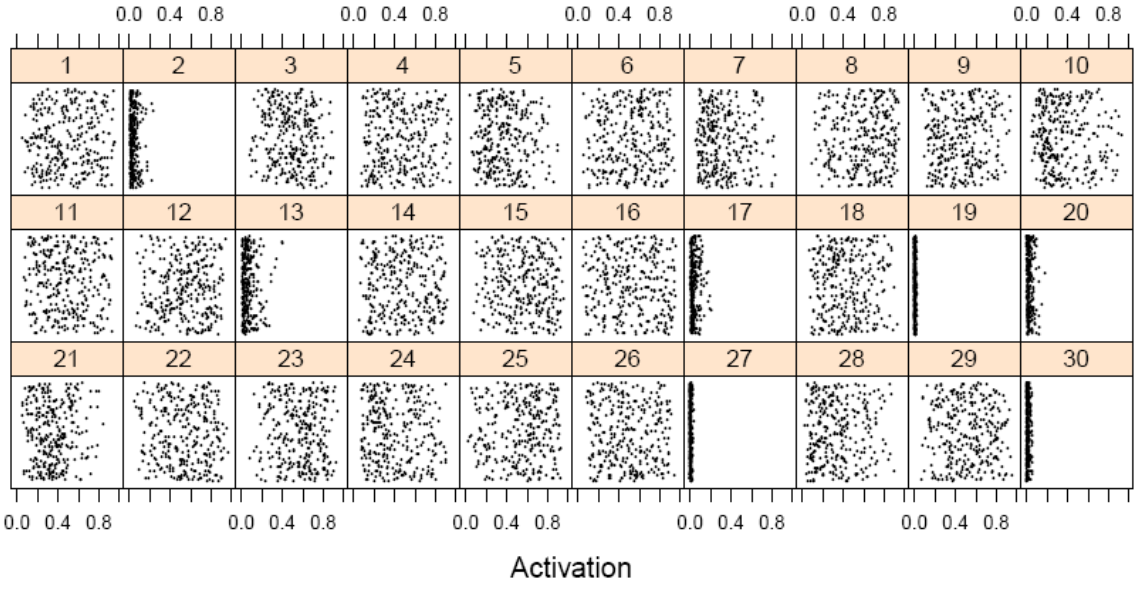
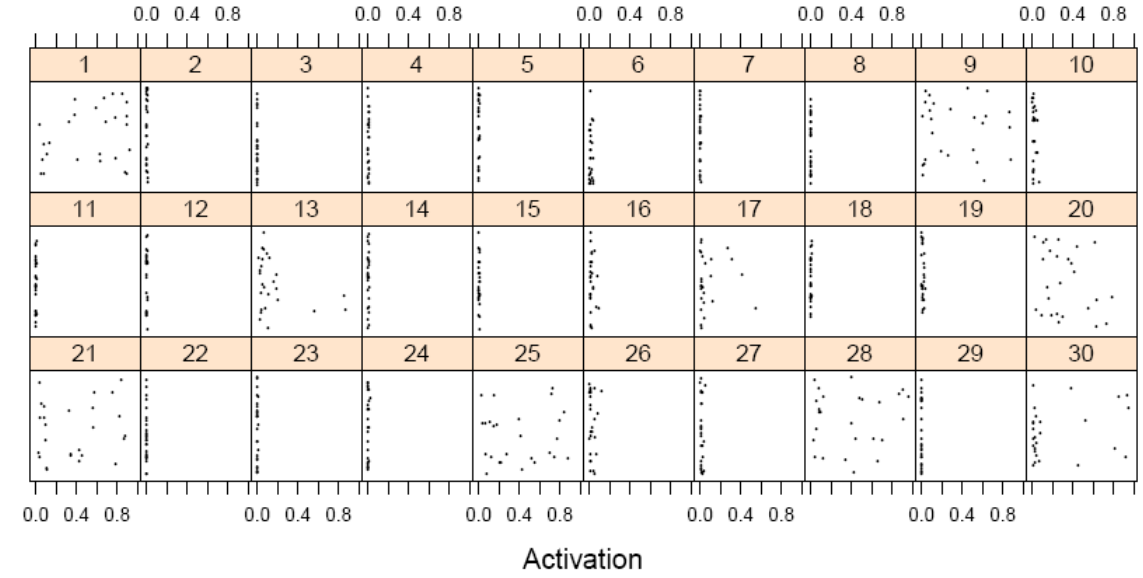

0.0 0.4 0.8
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
Activation

Figure 7


0.0 0.4 0.8
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
Activation

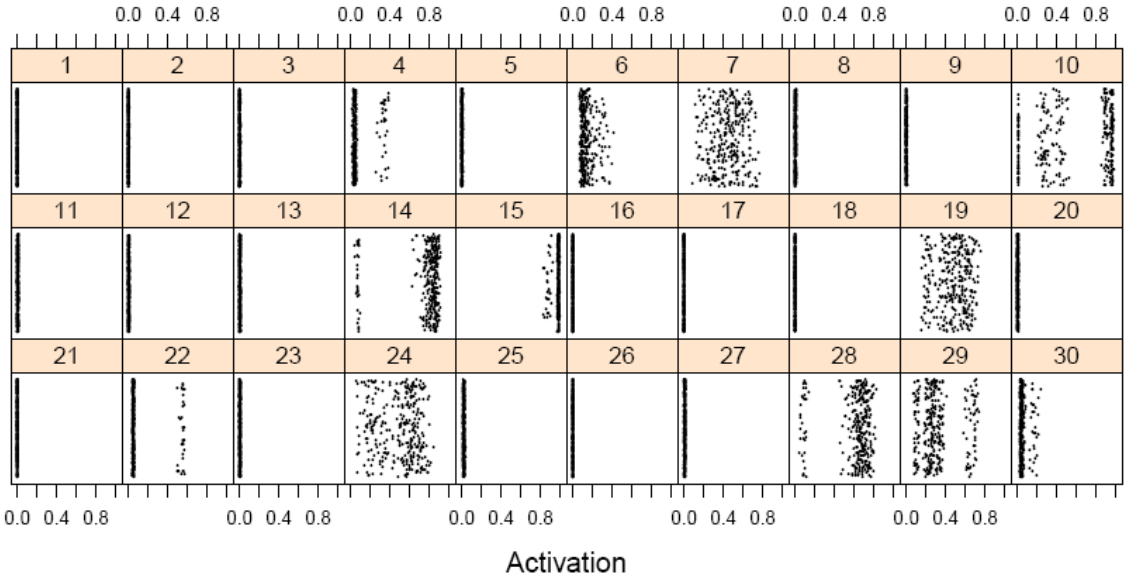
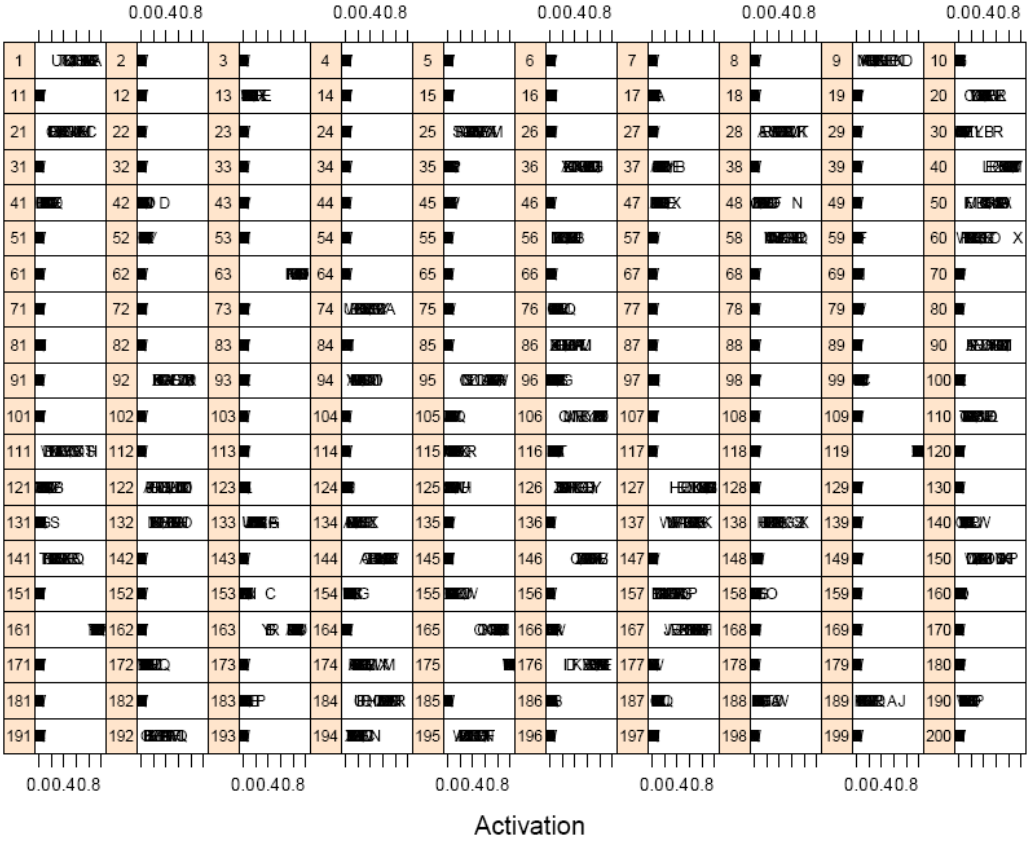

0.0 0.4 0.8
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
Activation

Figure 8



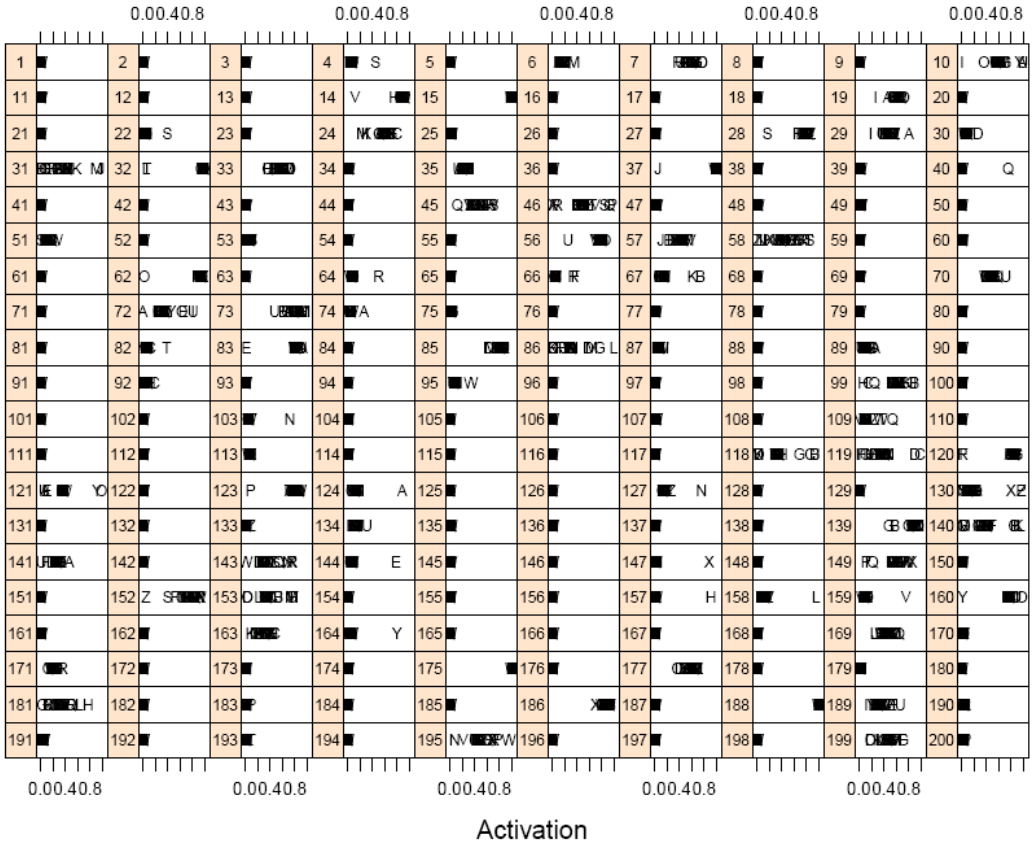


Figure 9

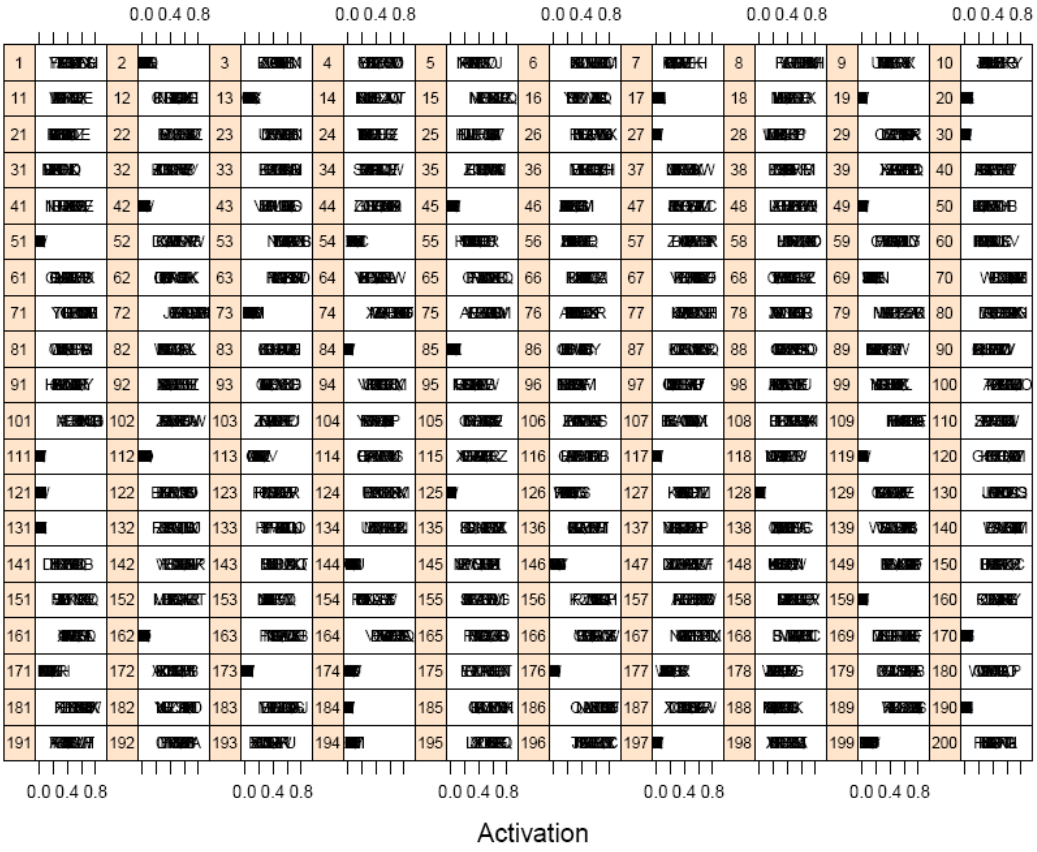
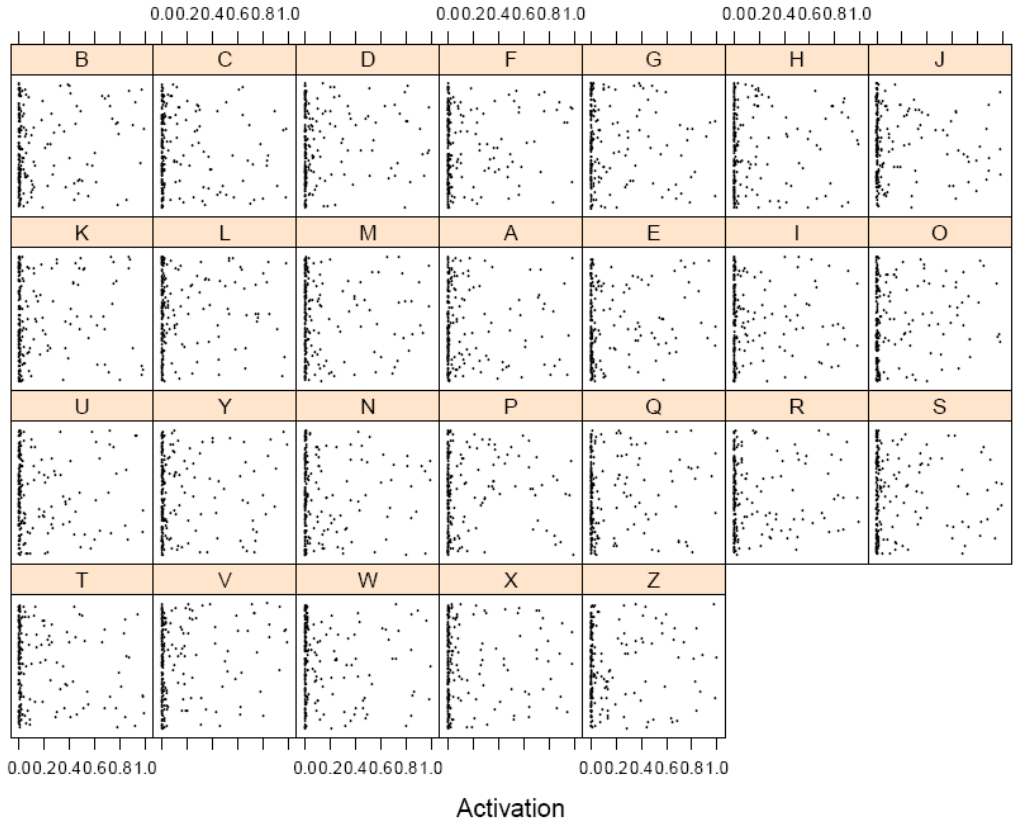


Figure 10


0.00.20.40.60.81.0
B
C
D
F
G
H
J
K
L
M
A
E
I
O
U
Y
N
P
Q
R
S
T
V
W
X
Z
Activation

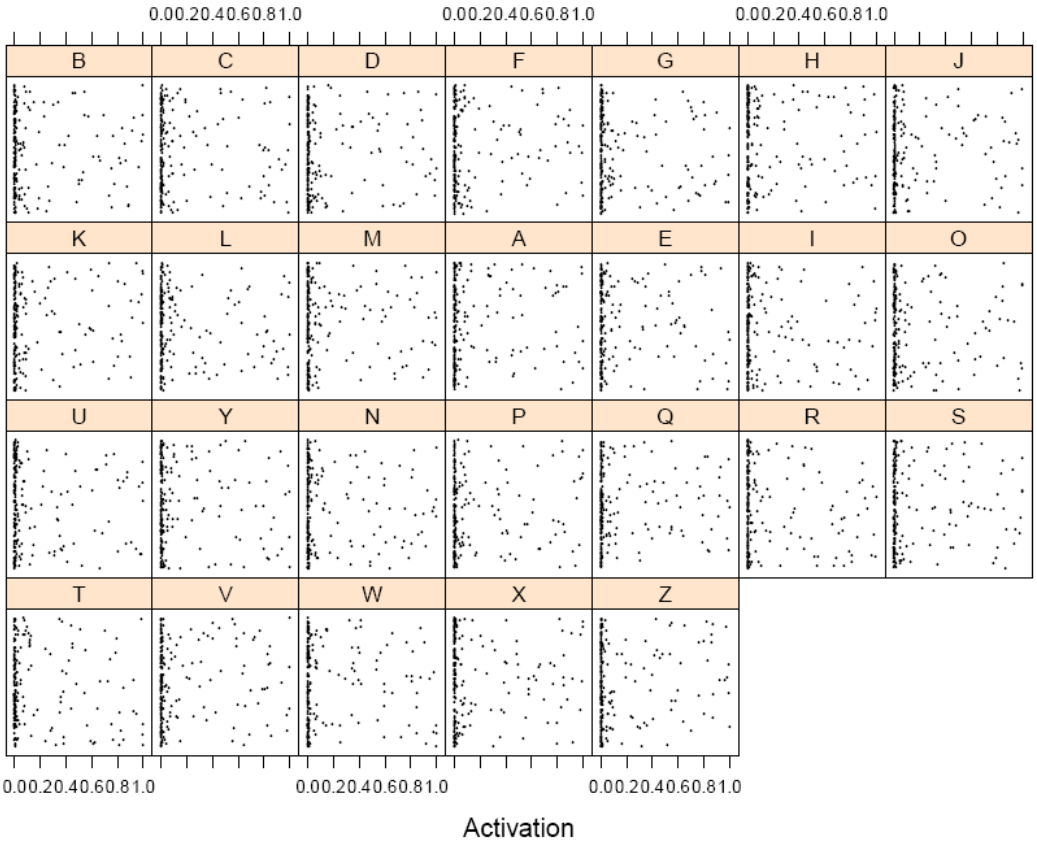

0.00.20.40.60.81.0
B
C
D
F
G
H
J
K
L
M
A
E
I
O
U
Y
N
P
Q
R
S
T
V
W
X
Z
Activation

Figure 11

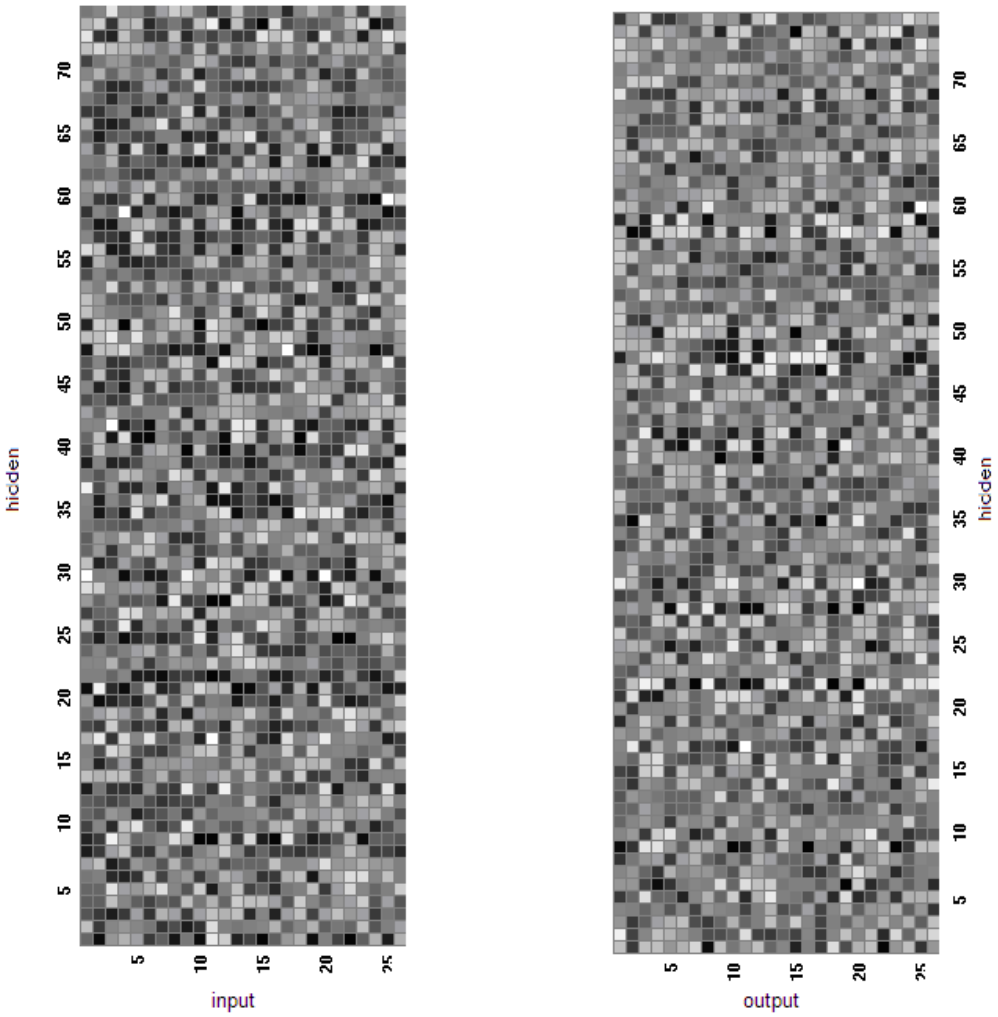


Figure 12

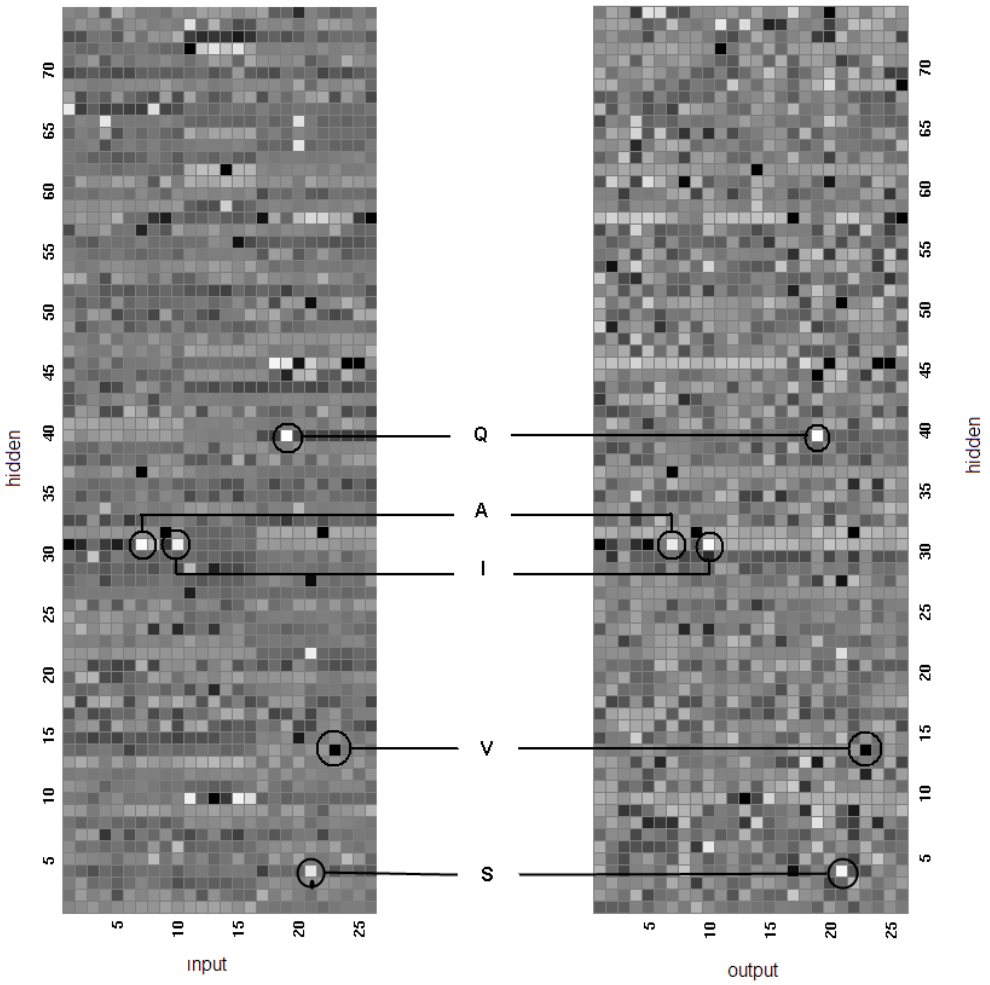
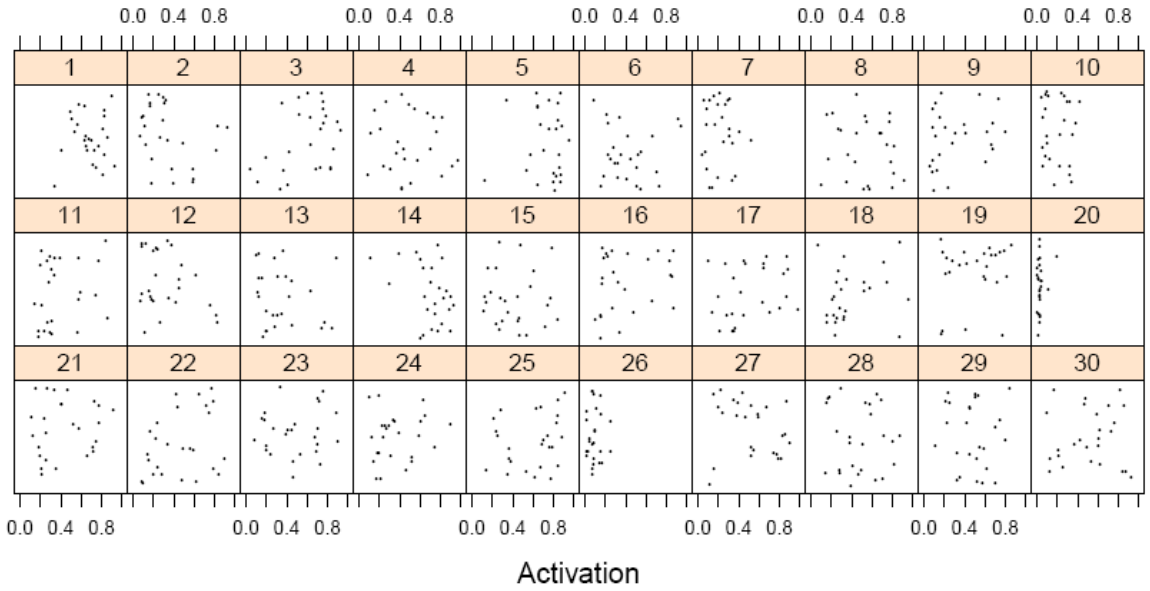


Figure 13


0.0 0.4 0.8
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
Activation

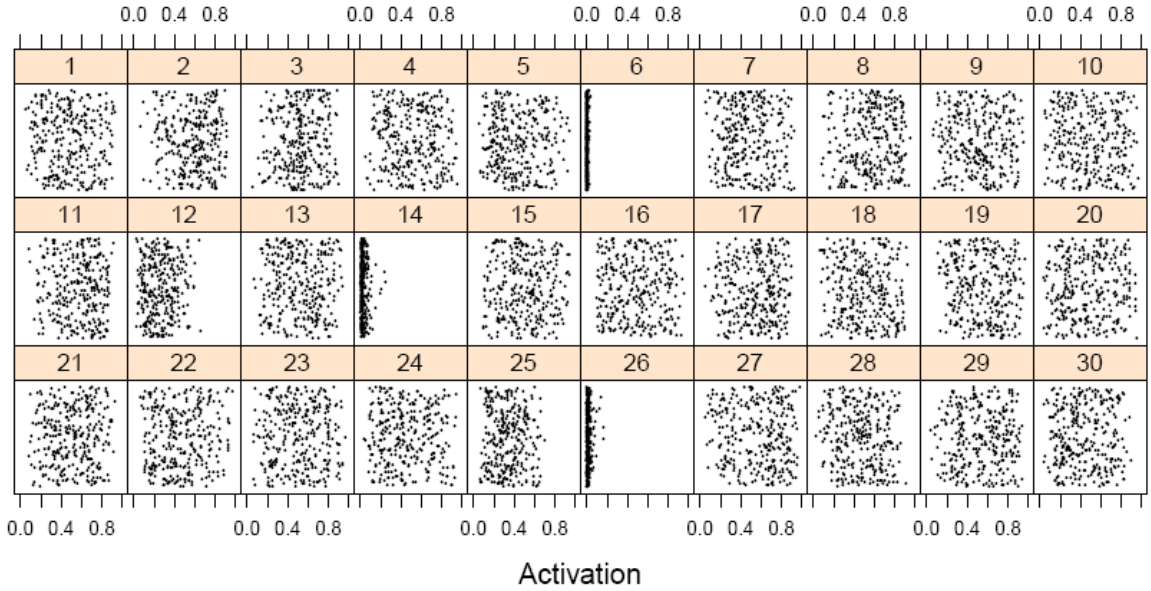
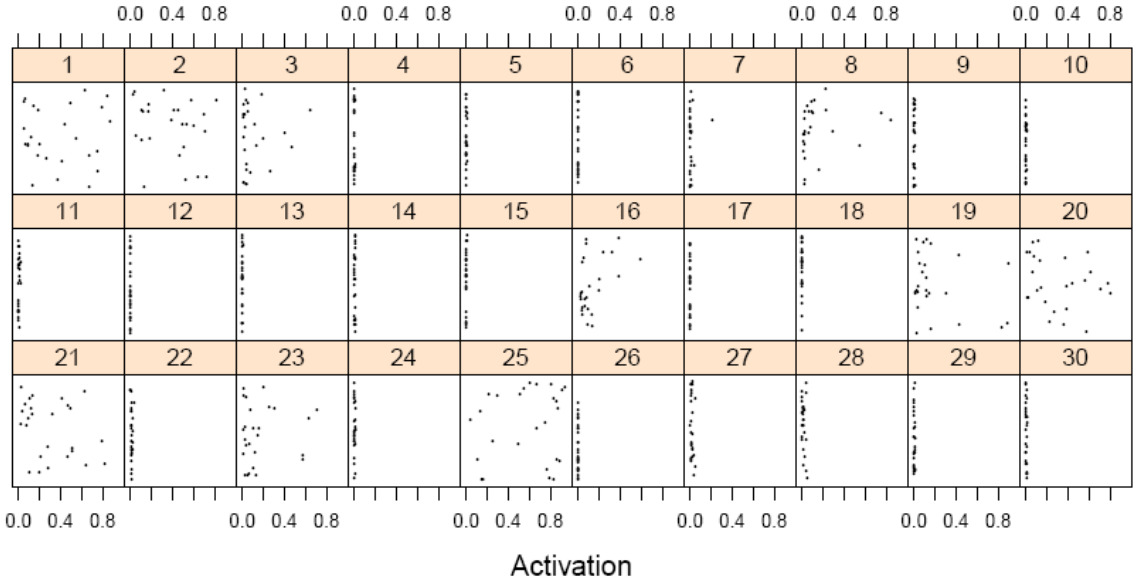
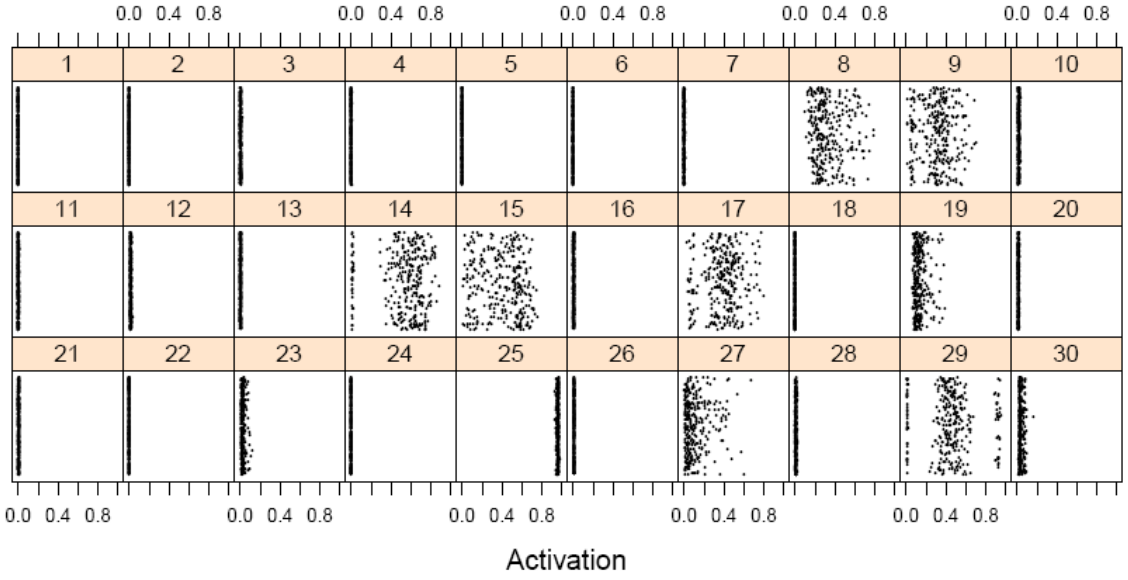

0.0 0.4 0.8
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
Activation

Figure 14


0.0 0.4 0.8
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
Activation


0.0 0.4 0.8
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
Activation