Beyond Pattern Completion with Short-Term Plasticity

Kevin D. Shabahang (k.shabahang@gmail.com)
Hyungwook Yim (hyungwook.yim@unimelb.edu.au)
Simon J. Dennis (simon.dennis@unimelb.edu.au)
School of Psychological Sciences, The University of Melbourne, Australia

Abstract

In a *Linear Associative Net* (LAN), all input settles to a single pattern, therefore Anderson, Silverstein, Ritz, and Jones (1977) introduced saturation to force the system to reach other steady-states in the *Brain-State-in-a-Box* (BSB). Unfortunately, the BSB is limited in its ability to generalize because its responses are restricted to previously stored patterns. We present simulations showing how a *Dynamic-Eigen-Net* (DEN), a LAN with *Short-Term Plasticity* (STP), overcomes the single-response limitation. Critically, a DEN also accommodates novel patterns by aligning them with encoded structure. We train a two-slot DEN on a text corpus, and provide an account of *lexical decision* and *judgement-of-grammaticality* (JOG) tasks showing how grammatical bi-grams yield stronger responses relative to ungrammatical bi-grams. Finally, we present a simulation showing how a DEN is sensitive to syntactic violations introduced in novel bi-grams. We propose DENs as associative nets with greater promise for generalization than the classic alternatives.

Keywords: Content Addressable Memory; Auto-associative; Recurrent; Short-Term Plasticity; Generalization

Introduction

An account of generalization demands specifying how patterns with little surface-level similarity to traces in memory can be interpreted based on their structural alignment. Associative nets have demonstrated the ability to complete partial input patterns -- they exploit mutual constraints derived from patterns in memory to help fill-in detail not immediate in the stimulus (e.g., Anderson, et al., 1977; Hopfield, 1982). However, current associative networks are driven by surface-level similarity between the input and patterns in memory. It remains an open question whether associative nets can be sensitive to higher-order similarity structures.

In associative nets, synapses between pairs of "neurons" are assumed to strengthen when the neurons activate within a short interval (Hebb, 1949). Assume for the sake of simplicity that a single neuron encodes a single feature in a stimulus (e.g., colour or size). Then the strengthened synapses encode correlations between features composing a stimulus.

In a classic Hebbian system, patterns are represented as vectors and are encoded as a function of the superposition of the outer-product of each pattern vector, with itself, into a single weight matrix. Retrieval is driven by a time-update function of the weight matrix and a state vector, which interact to yield the next state vector.

The state vector is first initialized to the probe. The state at the next time-point is a function of the vector-matrix multiplication of the current state and the weight matrix. The process is carried out iteratively until the system settles to a single state, i.e., when further iterations have no effect on the state vector. Iterative retrieval forces the various neurons to interact until the system reaches an equilibrium (steady) state -- the retrieved pattern.

In LANs, the process pushes the original input towards the dimension that captures maximum variance in the encoded patterns, the dominant *eigenvector* of the weight matrix. Since retrieval always settles to the dominant eigenvector, the system can only generate a single pattern.

Anderson et al. (1977) introduced *saturation* in the BSB by bounding each neuron's activation by a constant. Saturation constrains possible states of activation within a box and forces convergence to one of the corners. The bounding box halts the system before the state gets dominated by the lead eigenvector, thereby allowing a larger number of steady states or responses.

One of BSB's limitations (also true of Hopfield networks; Hopfield, 1982) is that it restricts steady states to single eigenvectors and treats retrieved patterns that do not correspond to a previously stored trace as *spurious*. If the eigenstructure of the system encodes statistical regularities from experience, the capacity to combine constraints from multiple dimensions of variance (eigenvectors) in the encoded patterns may be a more suitable candidate for generalization.

Encoding in associative nets is grounded in Long-Term Potentiation (LTP; Collingridge & Bliss, 1995), however more recent work in neurophysiology also suggests plasticity on shorter time-scales (STP; e.g., Tsodyks, Pawelzik, & Markham, 1998). Miller, Brody, Romo, and Wang (2003) used a recurrent spiking neural network model to explain data from prefrontal cortex neural activity in monkeys performing a somatosensory delayed discrimination task. The task requires the maintenance of the frequency of a vibration during a delay phase for later response, hence it falls in the domain of working memory. They suggested short-term facilitation as a possible mechanism for stabilizing steady (attractor) states. To our knowledge, the consequence of plasticity at both short (e.g., a few milliseconds to seconds) and long time scales (a few

weeks to years) has not been systematically explored in associative nets outside the realm of spiking neuron models.

We explore STP as a control signal for dynamically modulating the contribution of the various dimensions of variance encoded in the weight matrix. We show that STP provides an alternative to saturation for overcoming the dominant eigenvector problem without introducing non-linearities. Importantly, whereas in classic associative nets, the weights combining the eigenvectors -- the eigenvalues -- remain static, introducing STP dynamically weighs the eigenvectors based on their similarity to the current input pattern. The additional control on the eigenstructure enables mixed-eigenstates, or retrieved states that are spread across multiple eigenvectors.

The ability to combine multiple eigenvectors may be one mechanism for exploiting structural-level information. We provide some toy simulations to demonstrate the essential characteristics of DEN. We then scale up the system to encode bigram information from a text corpus. We use the scaled-up DEN to explain a set of JOG data showing slower responses for ungrammatical bi-grams relative to grammatical bi-grams. Next, we provide an account of lexical decision data examining the effect of syntactic violations between a prime and its successor, in two adjacent word-present trials, where there is a speed advantage in verifying the second word if it forms a syntactic bi-gram with its predecessor. Finally, we present evidence for generalizability in a simulation showing that a DEN can distinguish between novel grammatical and ungrammatical bi-grams.

A Toy Example

We now illustrate some key properties of three variants of associative networks. The first variant is a simple LAN, the second is the BSB, and the third is our STP-augmented LAN, which we refer to as a Dynamic-Eigen-Net (DEN).

We specify the encoded patterns and retrieval cues to be identical across the three variants, and only change the iterative retrieval algorithm. Following Anderson et al. (1997), we assume that encoded traces are constructed using vectors with an equal number of -1's and 1's. For simplicity, we define four orthogonal vectors, each with dimensionality four, and take each to stand for a single word in English. We assume that input to the system is an eight-dimensional vector, and construct bi-gram vectors by concatenating pairs of individual word vectors. Hence, bi-grams are coded with the word in the first serial position active in the first slot and the word in the second serial position active in the second slot.

Assume the four primitive word vectors correspond to "the", "a", "cat", and "dog". We construct the bi-grams "the cat" and "a dog" by concatenating the respective word-vectors, in sequence. We encode the two bi-gram vectors with unequal strengths to show how the LAN will always respond with the stronger pattern (strength of 1.2), even when the other pattern is only fractionally weaker (strength of 1.17).

For all following simulations in the toy demonstrations, we assume a weight matrix, $\mathbf{W} = 1.2\mathbf{m}_{\text{the-cat}}\mathbf{m}^{\text{T}}_{\text{the-cat}} + 1.17\mathbf{m}_{\text{a-dog}}\mathbf{m}^{\text{T}}_{\text{a-dog}}$, where the capital "T" denotes the transpose and $\mathbf{m}_{\text{the-cat}}$ and $\mathbf{m}_{\text{a-dog}}$ are the bi-gram vectors corresponding to "the cat" and "a dog", respectively. For later retrieval, we always construct the probe, $\mathbf{p}$, as $\mathbf{p} = 0.5\mathbf{x} + N(0, 0.1)$, where $\mathbf{x}$ is a bi-gram vector and $N(0, 0.1)$ is an eight-dimensional vector of samples from a zero mean Gaussian distribution with standard deviation set to 0.1. We scale down the probe by half prior to adding the noise and unit-normalize the resulting vector before attempting retrieval.

We characterize the system's state at each time-point in terms of the level of activation for the primitive word vectors, yielding four activation values in the first slot and four activation values in the second slot. We segment the state vector from the middle and take the first half as the first slot and the second half as the second slot. We set the activation value for a word in position one (or two) as the absolute value of the vector cosine of its primitive vector and the first slot (or second).

Linear Associative Net

In a LAN, we define the update function as the vector-matrix multiplication of the current state vector and the weight matrix. It is given by, $\mathbf{x}^{\text{T}}_{t+1} = \mathbf{x}^{\text{T}}_{t}\mathbf{W}$. We designate a small criterion, ε, and terminate retrieval when the change from one time-point to the next falls below criterion, $\| \mathbf{x}^{\text{T}}_{t+1} - \mathbf{x}^{\text{T}}_{t} \| < \varepsilon$. After each iteration, we normalize the state-vector to unit-length by dividing it with its vector-length. We set the convergence criterion to 1e-07 for all the following toy simulations.

We can rewrite the weight matrix, $\mathbf{W}$, in terms of its eigenstructure as $\mathbf{W} = \Sigma(\lambda_i \dot{\mathbf{e}}_i \dot{\mathbf{e}}_i^{\text{T}})$, where $\dot{\mathbf{e}}_i$ stands for the *i*'th eigenvector of $\mathbf{W}$. The weight matrix can be decomposed into a superposition of the outer-products of its eigenvectors, $\dot{\mathbf{e}}_i$, weighted by their respective eigenvalues, λ_i. We then have, $\mathbf{x}^{\text{T}}_{t+1} = \mathbf{x}^{\text{T}}_{t}\mathbf{W} = \mathbf{x}^{\text{T}}_{t-1}\mathbf{W}\mathbf{W} = \mathbf{x}^{\text{T}}_{0}\mathbf{W}^{t} = \Sigma_i(\lambda^t_i \mathbf{x}^{\text{T}}_{0}\dot{\mathbf{e}}_i)\dot{\mathbf{e}}_i^{\text{T}}$. In the limit, as we multiply the vector with the matrix and unit-normalize each time, $\mathbf{x}^{\text{T}}_{t}$ converges to the dominant eigenvector, $\dot{\mathbf{e}}_{\max}$, except when $\mathbf{x}^{\text{T}}_{0}\dot{\mathbf{e}}_{\max} = 0$. If we assume any level of noise, the LAN is a single-response memory system.

Table 1 shows the probability of a response (along the columns) as a function of the probe (along the rows). It is clear that the system always converges to the dominant pattern ("the cat") regardless of the input. We provide the probabilities based on 1000 runs with each probe in Tables 1, 2, and 3.

Table 1: Probabilities of response across probes show how a LAN is restricted to a single response.

	Probability of response					
	the cat	a dog	the dog	a cat	dog the	cat a
Probe						
the cat	1	0	0	0	0	0
a dog	1	0	0	0	0	0
the _	1	0	0	0	0	0
a _	1	0	0	0	0	0
the dog	1	0	0	0	0	0
a cat	1	0	0	0	0	0
dog the	1	0	0	0	0	0
cat a	1	0	0	0	0	0

Note. The underscores preceded by determiners, "the" and "a", stand for an empty second slot (i.e., all elements set to 0).

Brain-State-in-a-Box

The BSB remedies the dominant eigenvector problem by introducing saturation, forcing a maximum and minimum over the activations. The modified update function is given by,

$$\mathbf{x}^T_{t+1} = S(\mathbf{x}^T_t \mathbf{W}), \quad \text{where}$$

$$S(\mathbf{x}_i) = \begin{cases} 1, & \text{if } 1 > \mathbf{x}_i \\ \mathbf{x}_i, & \text{if } -1 < \mathbf{x}_i < 1 \\ -1, & \text{if } \mathbf{x}_i < -1 \end{cases}$$

with 1 being the saturation constant. Normalizing is not required when using saturation.

Table 2 has the same form as Table 1, and shows the probability of response as a function of the probes. Whereas the LAN always settled to the dominant pattern, "the cat", the BSB settles to "the cat" with high probability when probed with "the cat" or "the _". It settles to "a dog" when probed with "a dog" or "a _". The underscore denotes an empty slot. The partial probes demonstrate the pattern-completion capabilities of the system. The second two probes, "the dog" and "a cat", are novel combinations of the primitive word vectors. Both "the" and "a" have been encoded in the first slot and both "dog" and "cat" have been encoded in the second slot. Therefore they structurally align with the encoded patterns. However, the responses generated by the BSB are restricted to previously stored patterns.

The final two probes, "dog the" and "cat a", are the same novel combinations, except that the two words have been swapped, so that they are no longer aligned with the encoded structure. Here, the system responds with either "the cat" or "a dog", with the probability for the former being much greater than the latter because of the different encoding strengths. In practice, BSB's sensitivity to strength enables it to track probabilities of stimuli because higher frequency stimuli yield larger strengths for the relevant eigenvectors (Anderson et al., 1977).

Table 2: Probabilities of response across probes show how a Brain-State-in-a-Box is restricted to previously stored patterns.

	Probability of response					
	the cat	a dog	the dog	a cat	dog the	cat a
Probe						
the cat	1	0	0	0	0	0
a dog	0	1	0	0	0	0
the _	.997	.003	0	0	0	0
a _	.042	.958	0	0	0	0
the dog	.782	.218	0	0	0	0
a cat	.785	.215	0	0	0	0
dog the	.667	.333	0	0	0	0
cat a	.645	.355	0	0	0	0

Note. The underscores preceded by determiners, "the" and "a", stand for an empty second slot (i.e., all elements set to 0).

Dynamic-Eigen-Net

We now show how STP enables the system to generalize to novel patterns based on combinations of multiple eigenvectors. We model STP by assuming a temporary increase in the weights corresponding to the active entries in the input. We assume that the temporary change follows the presentation of a new input and spans the duration of the subsequent set of iterations, ending after convergence. We model STP by superimposing the outer-product of the probe, with itself onto the weight matrix. We also normalize the state vector after each iteration, as was done with the LAN.

The update function for a DEN is given by, $\mathbf{x}^T_{t+1} = \mathbf{x}^T_t(\mathbf{W} + \mathbf{x}_0\mathbf{x}^T_0)$. Written in terms of the original input, the state at time t is given by, $\mathbf{x}^T_t = \mathbf{x}^T_0(\mathbf{W} + \mathbf{x}_0\mathbf{x}^T_0)^t = \mathbf{x}^T_0 (\Sigma_i\lambda_i \dot{\mathbf{e}}_i\dot{\mathbf{e}}_i^T + \mathbf{x}_0\mathbf{x}^T_0)^t$. Assume $\mathbf{I}$ is the identity matrix. At time one, $\mathbf{x}^T_0 (\Sigma_i\lambda_i \dot{\mathbf{e}}_i\dot{\mathbf{e}}_i^T + \mathbf{x}_0\mathbf{x}^T_0)^1 = \Sigma_i\lambda_i(\mathbf{x}^T_0\dot{\mathbf{e}}_i)\dot{\mathbf{e}}_i^T + \mathbf{x}^T_0$. At time two, we have $\Sigma_i\lambda^2_i(\mathbf{x}^T_0\dot{\mathbf{e}}_i)\dot{\mathbf{e}}_i^T + \Sigma_i\lambda_i(\mathbf{x}^T_0\dot{\mathbf{e}}_i)^2\mathbf{x}_0^T + \Sigma_i\lambda_i(\mathbf{x}^T_0\dot{\mathbf{e}}_i)\dot{\mathbf{e}}_i^T + \mathbf{x}^T_0 = \Sigma_i\lambda^2_i(\mathbf{x}^T_0\dot{\mathbf{e}}_i)\dot{\mathbf{e}}_i^T + \Sigma_i\lambda_i(\mathbf{x}^T_0\dot{\mathbf{e}}_i)^2\mathbf{x}_0^T + \mathbf{x}^T_1$. Assuming that $\mathbf{x}^T_0$

is unit-normalized, since $\mathbf{x}^T_0\mathbf{x}_0\mathbf{x}^T_0 = \mathbf{I}\mathbf{x}^T_0 = \mathbf{x}^T_0$, any term carried over from a previous time-step ending with $\mathbf{x}^T_0$, persists to the next iteration.

In both a LAN and the BSB, convergence filters out any component orthogonal to the encoded eigenvectors. In a DEN the input persists and is merged with other components that do align with the eigenstructure.

Table 3 shows the probability of response as a function of the probe for a DEN. The first two probes ("the cat" and "a dog") yield a similar pattern of response as the BSB. The pattern completion dynamics are evident in the next two probes, "the _" and "a _", which yield comparable probabilities to the BSB. The second two probes in Table 3 show the response probabilities for "the dog" and "a cat", structurally aligned novel bi-grams. Although the system sometimes converges to the originally encoded patterns, it settles to the novel input with high probability, hence accommodating novelty.

Table 3: Probabilities of response across probes show how a DEN accommodates novel patterns.

	Probability of response					
	the cat	a dog	the dog	a cat	dog the	cat a
Probe						
the cat	1	0	0	0	0	0
a dog	0	1	0	0	0	0
the _	.923	.001	.076	0	0	0
a _	.003	.907	0	.09	0	0
the dog	.051	.03	.929	.002	0	0
a cat	.062	.039	0	.899	0	0
dog the	.635	.308	.027	.03	0	0
cat a	.629	.311	.028	.031	0	0

Note. The underscores preceded by determiners, "the" and "a", stand for an empty second slot (i.e., all elements set to 0).

Finally, the last two probes, "dog the" and "cat a", show the probabilities of responding when the words in the novel bi-grams swap serial-positions, resulting in novel combinations that do not align with the encoded structure. It is clear that the system never settles to the probed patterns (the last two response columns have zero probability). The most likely responses correspond to the previously encoded patterns. Hence, although the system accommodates novel patterns that structurally align with the encoded patterns, it does not do so if they are misaligned.

Figure 1 shows the activations of the vocabulary items as a function of iteration, for two example runs corresponding to the novel probes, "the dog" (top panel) and "dog the" (bottom panel). The pattern of responses shown in the figure has a high probability, as shown in Table 3. The plots on the left side correspond to the first slot whereas the ones on the right correspond to the second slot. When the novel pattern aligns with previously encoded structure (top panel), the system settles to the novel pattern, however, when the same words in the bi-gram are swapped to break the alignment (bottom panel), the system settles to a similar pattern that does align (i.e., "a dog"). Hence, a DEN surpasses the generative complexity of the BSB, while still respecting the statistical regularities encoded in memory.

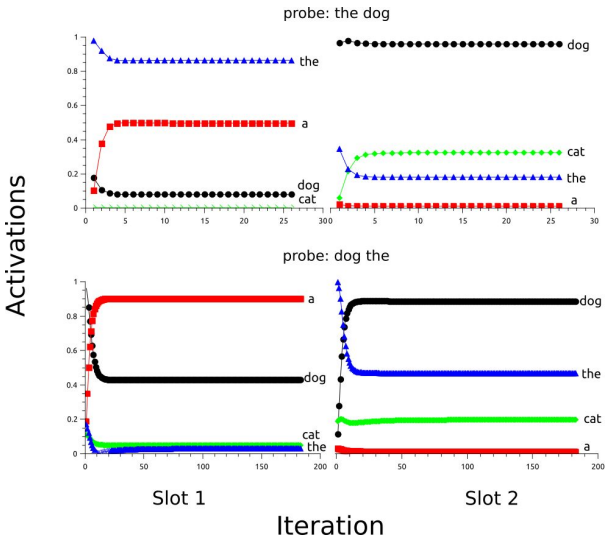

Figure 1. Probing a DEN with a novel bi-gram yields a response corresponding to the initial input, only if it is aligned with the structure encoded in the matrix (top panel) but not when it is misaligned (lower panel). The number of iterations differs between the top and bottom panels because it takes longer for the system to converge to a misaligned input.

From Theory to Data

As a first point of contact between theory and data, we construct a DEN to explain some findings in the JOG and lexical decision tasks. The JOG task requires subjects to decide whether a sequence of words forms a well-formed composite, whereas the lexical decision task requires participants to decide if arrays of letters presented to them are words or nonwords.

Münte and Heinze (1993) constructed bi-grams composed of a pronoun followed either by a noun or a verb, such as "my cat" and ""you walk". They introduced violations by switching possessive pronouns to personal pronouns, or vice versa, as in "me cat" and "your walk". They presented each bi-gram to participants, and asked them to make a speeded grammaticality judgement. They found a 50 msc cost when subjects responded to invalid bi-grams, relative to valid bi-grams, with the invalid bi-grams evoking an event-related

negativity, whose amplitude was much larger than the valid bi-grams, in the 300-600 msc range and maximal along the left frontopolar cortex.

For two adjacent word-present trials, lexical decision is faster for the second word when its syntactic class is predicted by the previous word relative to when it violates the prediction (e.g., Goodman, McClelland, & Gibbs, 1981). Colé and Segui (1994) presented subjects with pairs of letter-strings and asked them to respond "yes", only if both were valid words in French. Colé and Segui report subjects were slower to verify syntactically invalid bi-grams relative to valid bi-grams, with a greater effect for bi-grams whose initial word was closed-class relative to bi-grams whose first word was open-class. Open-class (e.g., adjectives, nouns, verbs etc.) words roughly correspond to content words and closed-class (e.g., determiners, pronouns, prepositions) words to function words. It has been speculated that the two classes may be stored separately (Garrett, 1979).

In the next set of simulations, we trained a DEN by sliding a two-word window across the Touchstone Applied Science Associates (TASA) corpus to encode all the lag-one sequential dependencies in a two-slot system, similar to the one used in the toy examples. We first segmented the corpus by sentence and slid a window across each sentence separately. For each sentence, we superimposed the outer-products of bi-grams into a co-occurrence matrix.

We used the Stanford Part-of-speech (POS) tagger (Toutanova, Klein, Manning, & Singer, 2003) to classify each word in the TASA corpus into a syntactic class. We then grouped bi-grams based on their syntactic composition. For each of nine different syntactic compositions, we used the 150 most frequent bi-grams as our grammatical set. We then constructed an ungrammatical bi-gram for each bi-gram in the grammatical set.

The first two columns in Table 4 show the valid (top) and invalid (bottom) bi-gram compositions used, each with an example. For example, in a bi-gram consisting of a possessive pronoun and a noun (PPRS-NN), we can construct an ungrammatical bi-gram by turning the possessive pronoun into a personal pronoun (i.e., PPR-NN). If the PPRS-NN bi-gram is "her cat", the ungrammatical, PPR-NN, bi-gram would be "she cat". Alternatively, for a determiner-noun bi-gram (DT-NN), such as "the cat", we can construct a corresponding ungrammatical bi-gram, "cat the", by swapping the two words (NN-DT). The PPRS-NN versus PPR-NN and PPR-VBP versus PPRS-VBP comparisons correspond to Münte et al. (1993). We can use the DT-NN versus NN-DT and JJ-NN versus NN-JJ to get a handle on the open- versus closed-class distinction in Colé et al. (1994).

We made several modifications to the DEN presented in the toy examples in order to scale the system. The first modification was a change from distributed-codes to local-codes. Instead of representing a word as a vector of -1's and 1's, we represented it with a vector with all but one element set to zero. The nonzero element, at an index unique to each word, was set to one.

Local-codes ensure orthogonality and facilitate an increase in the vocabulary size. The second modification was to treat a word's activation level as the absolute value of its corresponding element in the state-vector and follows from our change of representation to local-codes. With the localist representation, each cell in the weight matrix is proportional to the co-occurrence count of two words occurring in a bi-gram.

Table 4: The model shows good discriminability between valid and invalid bi-grams across various syntactic violations.

Violation	Examples	Intact	Lesioned
NNS-VBP NN-VBP	animals have animal have	2.359	0.406
IN-VBG IN-VBP	by looking by look	2.011	0.131
NN-VBZ NN-VBP	one knows one know	1.918	1.562
○DT-NN NN-DT	the dog dog the	1.898	1.73
⋄PPRS-NN PPR-NN	his head he head	1.486	1.09
○JJ-NN NN-JJ	long time time long	1.069	0.172
NN-IN IN-NN	look at at look	1.057	0.889
⋄PPR-VBP PPRS-VBP	you think your think	0.558	0.183
VB-RBR RBR-VB	learn more more learn	0.399	−0.094

Note. The abbreviations correspond to the Penn Treebank convention for parts-of-speech. NN: singular or mass noun, IN: preposition, VB: verb, RBR: comparative adverb, VBZ: third-person, singular, and present tense verb, NNS: plural noun, VBP: non-third-person, present tense-verb, VBG: gerund or present-tense verb, DT: determiner, JJ: adjective, PPR: personal-pronoun, PPRS: possessive pronoun. The symbol, ⋄, designates comparisons corresponding to Münte and Heinze (1993) and the symbol, ○, designates comparisons we use to explain the closed-class versus open-class distinction relevant to Colé and Segui (1994).

We apply a threshold to the encoded co-occurrence counts, such that, the (i, j)'th cell is set to one only if its value is greater than 2, and zero otherwise. The thresholding helps prune spurious co-occurrences. Finally, the entire weight matrix was divided by 300, a value slightly higher than the largest eigenvalue of the matrix. Bounding the largest eigenvalue of the system prevents overpowering the contribution of STP during retrieval. The convergence criterion was set to 1e-05.

We compute a familiarity signal for each iteration as the vector length of the state prior to normalization. We tally the familiarity signals in the last iteration for each bi-gram and use the difference in familiarity between the paired valid and invalid bi-grams as a measure of discriminability. For each comparison, we compute discriminability as the quotient of the mean familiarity difference and its standard deviation. Because we subtracted the familiarity for the invalid bi-grams from the valid bi-grams, positive discriminability indicates greater familiarity for the valid relative to the invalid bi-grams.

The third column in Table 3 shows the discriminabilities for the nine syntactic compositions in descending order. It is clear that the discriminability scores all lie above zero, showing that the system can distinguish between valid and invalid bi-grams.

Consistent with Münte et al. (1993) the possessive pronouns followed by a noun (PPRS-NN; e.g., "his head") are more familiar than personal pronouns followed by a noun (PPR-NN; "he head") and the personal pronouns followed by a verb (PPR-VBP; e.g., "you think") are more familiar than possessive pronouns followed by a verb (PPRS-VBP; "your think"). Although Münte et al. did not find a significant difference between the bi-grams containing the nouns relative to verbs, we obtain greater discriminability for the violations in bi-grams containing nouns.

As regards to Colé et al. (1994), both the determiner versus noun swaps (DT-NN vs NN-DT; e.g., "the dog" vs "dog the") and adjective noun swaps (JJ-NN vs NN-JJ; e.g., "long time" vs "time long") show reasonable discriminability in that the valid bi-grams are considered more familiar than the invalid bi-grams. In line with the greater syntactic congruity effect for closed-class relative to open-class bi-grams they reported, we obtain greater discriminability for the closed-class bi-grams relative to the open-class bi-grams.

To determine the extent to which the discriminability between valid and invalid bi-grams is based on encoded structure the last column of Table 4 shows what happens if the system has never encoded a bi-gram. Before probing the system with a valid and invalid bi-gram, we lesion memory by zeroing out the corresponding cell for the valid bi-gram in the weight matrix (and its mirror image in the matrix's transpose). After convergence, we restore the cell to its original value before moving on to a different bi-gram pair. The obtained discriminability scores illustrate the system's generalizability potential. All but one of the discriminability scores fall above zero, showing that the system is able to exploit structure to generalize.

Discussion

We explored the advantage of including both short- and long-term plasticity over the LAN and Anderson et al.'s (1977) BSB model. In a set of toy examples, we showed how augmenting a LAN with STP, modelled by summing in the outer-product of a probe vector to the weight matrix prior to iterative retrieval, overcomes the single-response limitation in the LAN. We further showed how the BSB's responses are limited to previously encoded patterns, and how a DEN can assimilate novel patterns based purely on the structure encoded from previous instances. The DEN can reach equilibria that correspond to novel patterns when the novel patterns align with the eigenstructure of the system but not if they are contradictory to the structure.

We scaled up the system to show that a two-slot Dynamic-Eigen-Net trained on the TASA corpus encodes enough structure to explain some empirical results from judgement-of-grammaticality and lexical decision tasks. Finally, we showed the system is able to distinguish between well-formed and ill-formed bi-grams even when memory for the bi-grams is lesioned. That is, the system is sensitive to various syntactic regularities based purely on the structural statistics of the language environment.

Table 4 shows that the DT-NN versus NN-DT (closed-class) violations are highly discriminable based on encoded structure whereas JJ-NN versus NN-JJ (open-class) violations mainly rely on knowledge of the bi-grams themselves. We propose that the distinction between closed-class and open-class words need not imply distinct representations, a priori, but that the former is better determined by encoded structure relative to the latter, hence explaining the difference in the syntactic congruity effect reported by Colé et al. (1994).

Although a systematic exploration of Short-Term-Plasticity in associative nets has not been presented in the cognitive modelling literature, its use in models has precedence (e.g., Gardner-Medwin, 1989; Burgess & Hitch, 1999; Plaut & Shallice, 1993; Feldman, 1982). The various implementations make different architectural and learning assumptions, but overall show that including Short-Term-Plasticity has several advantages over networks with static connectivity. For instance, in the Hebbian systems, Gardner-Medwin assumes short-term weights are multiplicatively combined with long-term weights whereas Burgess and Hitch assume additive combination of the two. Gardner-Medwin provides information-theoretic analyses showing capacity advantages using their implementation of Short-Term-Plasticity and Burgess and Hitch rely on Short-Term-Plasticity to capture a set of memory benchmarks. Plaut and Shallice additively combine Short-Term-Plasticity to long-term weights, trained using error-driven learning, to model perseveration in their model of deep dyslexia. Feldman suggests Short-Term-Plasticity as one possible mechanism for feature-binding in vision. All of the accounts assume non-linear activation functions that obfuscate the influence of underlying eigenstructure of their systems, and neither provides a direct comparison with classic associative nets such as the Brain-State-in-a-Box.

Our implementation of Short-Term-Plasticity compliments the spiking neuron work (e.g., Miller et al., 2003) by

showing how forcing attractor states provides the kind of stability needed for increasing the generative potential of the system. Wang, Markram, Goodman, Berger, Ma & Goldman-Rakic (2006) suggested a recurrent excitation threshold that must be surpassed in order for the system to transition into a reverberatory loop. In our implementation of Short-Term-Plasticity, we add the auto-association of a pattern into the weight matrix, before running the retrieval iterations. A critical requirement is that the weight of the auto-association exceeds the value of the dominant eigenvalue of the system as a whole. The lead eigenvalue can be considered a kind of recurrent excitation threshold of the system.

Short-term weights have been explored in the broader machine-learning literature, however, the implementations rely on error-driven learning. For instance, Ba, Hinton, Mnih, Leibo, & Ionesco (2016) suggest encoding a set of recent hidden states into a fast-decaying set of short-term weights as a way to model greater facilitation for states corresponding to the system's recent hidden state history. They propose the mechanism for recurrent neural nets which use error-driven learning for training the long-term connectivities. Reliance on error-driven learning limits the system's ability to meet human-level cognitive benchmarks such as one-shot learning. Perhaps the kind of generalization sought in the broad machine-learning literature is better driven through recurrence within a Dynamic-Eigen-Net as opposed to slow error-driven learning.

In conclusion, our demonstrations show that including STP in LANs vastly increases their generative complexity. Given that generalization requires exploiting structural regularities with little surface-level information, and a DEN's ability to assimilate novel patterns based on structure alone, we propose our model as one candidate for a system capable of generalization.

References

Anderson, J. A., Silverstein, J. W., Ritz, S. A., & Jones, R. S. (1977). Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psychological review*, *84*(5), 413-451.

Ba, J., Hinton, G., Mnih, V., Leibo, J. Z. & Ionesco, C. (2016). Using fast weights to attend to the recent past. Advances in Neural Information Processing Systems, 29, 4331-4339

Burgess, N. & Hitch, G. J. (1999). Memory for serial order: a network model of the phonological loop and its timing. Psychological Review, 106(3), 551-581.

Collingridge, G. L., Bliss, T. V. P. (1995). Memories of NMDA receptors and LTP. *Trends in Neurosciences*, 18, 54-56.

Cole, P., & Segui, J. (1994). Grammatical incongruency and vocabulary types. *Memory & Cognition*, 22(4), 387-394.

Feldman, J. A. (1982). Dynamic connections in neural networks. Biological Cybernetics, 46, 27-39.

Gardner-Medwin, A. R. (1989). Doubly modifiable synapses: a model of short and long term auto-associative memory. Proc. R. Soc. Lond. B 238, 137-154.

Garrett, M. (1980). Levels of processing in sentence production. *In Language production Vol. 1: Speech and talk* (pp. 177-220). Academic Press.

Goodman, G. O., McClelland, J. L., & Gibbs, R. W. (1981). The role of syntactic context in word recognition. *Memory & Cognition*, 9(6), 580-586.

Hebb, D. O. *The organization of behavior*. New York: Wiley, 1949.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA*, 79, 2554-2558.

Miller, P., Brody, C. D., Romo, R. & Wang, X. (2003). A recurrent network model of somatosensory parametric working memory in the prefrontal cortex. Cerebral Cortex, 13(11), 1208-1218.

Plaut, D. C. & Shallice, T. Perseverative and semantic influences on visual object naming errors in optic aphasia: a connectionist account. Journal of Cognitive Neuroscience, 5(1), 89-117.

Toutanova, K., Klein, D., Manning, C., & Singer, Y. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *Proc. HLT-NAACL 2003*, pp, 252-259.

Tsodyks, M., Pawelzik, K., Markham, H. (1998). Neural networks with dynamic synapses. *Neural Computation*, 10, 821-835.

Wang, Y., Markram, H., Goodman, P. H., Berger, T. K. & Ma, J., Goldman-Rakic, P. S. (2006). Heterogeneity in the pyramidal network of the medial prefrontal cortex. Nature Neuroscience, 9(4), 534-542.