Analyzing a connectionist model as a system of soft rules

Clayton McMillan & Paul Smolensky

Department of Computer Science & Institute of Cognitive Science
University of Colorado, Boulder

With the rise to prominence of the connectionist or parallel distributed processing approach to cognitive modeling, the issue of the relation of such models to rule-based descriptions has been a consistent source of debate. It is our purpose in this paper to show that rather than regarding these approaches as completely mutually exclusive, there is insight to be gained in viewing standard connectionist models from a rule-based perspective. Our strategy is to show how a classic PDP model can be decomposed and viewed as a kind of rule-based model. We start with a summary of the PDP model we studied: Rumelhart and McClelland's (1986) model of acquisition of the past tense in English. This model is a natural choice, partly because it – and Rumelhart and McClelland's claims about its implications for the relation between connectionist models and rule-based accounts – has recently been the center of considerable controversy (Lachter & Bever, 1988; Pinker & Prince, 1988).

The past tense model

The past tense model simulates how children acquire the past tense of English verbs. It was designed to test the power of the PDP approach in what has long been considered the domain of rule-based models, natural language description and acquisition. Upon repeated presentation of verb stems and their corresponding past tense forms, the model learns a set of weights capable of producing the past tense of all 460 verbs in the corpus, plus many others not in the corpus. We are primarily concerned with the model after learning has been completed by presenting a corpus of verbs over 200 training cycles, and this final set of weights has been achieved.

The model is a bipartite graph. A phonetically spelled representation of the input – the stem of an English verb – is translated into a subset of 460 *Wickelfeatures*. These 460 Wickelfeatures are *position-independent, context-dependent phonetic features* and represent a fine-grained but somewhat restricted representation of phonemes present in the English language. For example, [*Back Vowel Front*] is a Wickelfeature present in any phonetic string that, somewhere, contains a vowel preceded by a back phoneme and followed by a front phoneme. This Wickelfeature is present in /kAm/ (*came*); there are 15 other Wickelfeatures present in this context-dependent representation of the vowel. Each Wickelfeature corresponds to one of 460 input units in the model. The input pattern is presented to the model by activating each of the input units corresponding to Wickelfeatures present in the input stem. Activation then passes once across the connections to a set of 460 output units, each of which also represents a Wickelfeature. As a result, units in the output become either on or off, depending upon the values of the weights on the connections, according to a certain stochastic rule. These output units indicate the Wickelfeatures present in the past tense form of the input verb.

Interpreting the output of the model is a bit complex. The degree to which the model prefers a given target output string over other possible strings is quantified in several ways. The most complex measure is *response strength*, which is computed by a rather complex network for decoding the output Wickelfeatures into output phoneme strings. A simpler measure Rumelhart and McClelland used is

$$1 - [fraction\ of\ target\ 1s\ not\ matched + fraction\ of\ target\ 0s\ not\ matched]$$

We will simply call this the *feature match* between the target and output. This can be defined for a single verb, or for a set of verbs; in the latter case, the two fractions appearing in the formula are each computed once over the whole set of verbs.

In general, if the correct response has a feature match in the range .50–.60, the decoding network tends to produce a response strength for the correct response that is greater than that of any other responses, but its superiority is often weak. With feature matches higher of about .65, the superiority in response strength starts to become pronounced.

The behavior of the model is regulated by the weights on the connections. The weights may be viewed as a matrix, where the weight on the connection between input unit i and output unit j occupies location (j, i) of the matrix: w_{ji}. We will refer to this matrix as $\mathbf{W}_{sim}$ because it is generated by training the model on an entire corpus of verbs simultaneously.

If this were a rule-based system the behavior would be regulated by rules that transform the verb stems into the past tense. After training, rules are *implicitly* present, although inaccessible, in $\mathbf{W}_{sim}$. It is this set of inaccessible rules that we wish to extract from $\mathbf{W}_{sim}$. Our strategy is to decompose $\mathbf{W}_{sim}$ into *several* weight matrices, each of which may be considered to correspond to a rule.

Goal of this research

Rule-based views of the formation of the English past tense have been developed by Bybee and Slobin (1982) and Hoard and Sloat (1973). Although these views are quite different, they share a common ground: Each verb is marked as belonging to a certain verb class. For each verb class there exists one or more rules that transform those verbs into the correct past tense form. The combination of these rules and markings represent a rule-based description of this cognitive task.

Our goal is to decompose $\mathbf{W}_{sim}$ into separate matrices, one for each class of verbs. We use the verb classification of Bybee and Slobin for our decomposition. They have identified eight irregular and three regular classes of verbs, each identified by shared morphological and phonological characteristics. We will therefore decompose $\mathbf{W}_{sim}$ into 11 separate matrices.

In order to view the weight matrices derived from $\mathbf{W}_{sim}$ as rules, each matrix must generate the correct past tense for verbs in its class. Each such matrix will be called a *soft rule matrix*. In order that these define something like a rule *system*, there must be a means of combining these 11 soft rule matrices into a single *composite* matrix, $\mathbf{W}_{com}$. We seek a $\mathbf{W}_{com}$ that performs the task at a level comparable to that of $\mathbf{W}_{sim}$.

Decomposing the weight matrix

We have taken a very simple approach to decomposing $\mathbf{W}_{sim}$. $\mathbf{W}_{sim}$ is generated by training the model on all verbs in all classes simultaneously. In a sense, it is a soft rule matrix that generates the past tense of all verbs in the corpus. Taking this same approach, we have chosen to generate soft rule matrices for each of the 11 classes of verbs by training the model separately on each class of verbs. The result of this approach is 11 soft rule matrices, one for each class.

The exclusive case

In the simplest technique, the training for a given verb class involves exclusively the verbs in that class; this training regime will therefore be called the *exclusive* case.

The behavior of the soft rule matrices in the exclusive case may be illustrated as follows. Bybee and Slobin's verb class 1 is the set of verbs that don't change in the past tense, such as *beat* or *cut*. Given any verb in class 1, $\mathbf{W}_1$ will generate the correct past tense for that verb. For a verb not in class 1, the behavior of the model is unspecified and unpredictable. Generally, the output for a verb outside class 1 will be a pattern containing features common to verbs in class 1. The same behavior is exhibited by $\mathbf{W}_\alpha$ for all classes $\alpha = 1, \ldots, 11$ in the exclusive case.

In this sense these soft rule matrices differ from traditional rules. With a traditional rule one would

expect a conditional clause to determine whether or not the rule should fire. If the rule fires, it will perform a deterministic transformation upon the state of the system. Either the transformation will be performed because the condition is met, or the state of the system will not change. By contrast, the soft rule matrices $\mathbf{W}_\alpha$ have no conditional element: they will always "fire" when presented with input. The output is only predictable when the soft rule matrix is applied to input that it is designed to accept.

The null case

It is possible to change the training regime so that the soft rule matrix corresponding to a given verb class corresponds more nearly to a traditional rule in that it fails to "fire" when presented with input outside the class – that is, for such input it produces *zero* output. Because of this null output specification, we call this training regime the *null* case. In the null case, the matrix $\mathbf{W}_\alpha$ is generated by training the past tense model on verbs in class α with the verb stems and correct past tense as the target, just as in the exclusive case. However, in addition, the model is presented with all verbs from the original training corpus that are *not* in class α, with *null* target patterns instead of the correct past tense.

To implement the null training regime we need to modify the delta learning rule used by Rumelhart and McClelland. In training $\mathbf{W}_\alpha$, the goal for verbs outside class α is that the *net input to each output unit j be zero.* Thus, if the input to the network represents a verb outside class α, let net_j be the net input to unit j: $net_j = \sum_i w_{ji} a_i$ (Here a_i is the activity of input unit i.) Let δ_j be 1 if $net_j < 0$, -1 if $net_j > 0$, and 0 if $net_j = 0$. Then we change the weight according to the usual delta rule: $\Delta w_{ji} = \delta_j a_i$ This will drive the weights towards the desired target of $net_j = 0$.[1] (Here, as throughout, the thresholds on the output units are replaced by weights to a hypothetical input unit that is always on; these weights are modified exactly like all other weights.)

Combining soft rule matrices into a single system

We have adopted two methods for combining the 11 soft rule matrices into a single composite matrix $\mathbf{W}_{com}$: *linear regression* and *straight summation.* In both methods we are assuming that there exists a linear relationship between the separate soft rule matrices and $\mathbf{W}_{com}$.

Linear regression

The idea behind the linear regression technique is to search for some set of weighting coefficients for the 11 soft rule matrices such that the weighted sum of these matrices will generate a single matrix that is close to $\mathbf{W}_{sim}$ and therefore can be expected to behave in a similar fashion. In effect we wish to minimize an *error* which is the mathematical difference between $\mathbf{W}_{com}$ and $\mathbf{W}_{sim}$.

In this analysis, $\mathbf{W}_{sim}$ is viewed as the dependent variable and the 11 soft rule matrices as independent variables. The result of the analysis is a set of 11 coefficients c_α; each coefficient is multiplied by the corresponding soft rule matrix $\mathbf{W}_\alpha$ and the results are summed together to produce $\mathbf{W}_{com}$: $\mathbf{W}_{com} = c_1\mathbf{W}_1 + \cdots + c_{11}\mathbf{W}_{11}$. The error we minimized is the usual sum-squared measure of the difference between $\mathbf{W}_{sim}$ and $\mathbf{W}_{com}$:

$$error = \sum_{i=1}^{460} \sum_{j=1}^{460} \left[w_{ij}^{sim} - w_{ij}^{com} \right]^2$$

where the w_{ij}^{sim} are the weights in $\mathbf{W}_{sim}$ and w_{ij}^{com} are the weights in $\mathbf{W}_{com}$. Minimizing this with respect to the coefficients $(c_1, c_2, \cdots, c_{11}) = \mathbf{c}$ that determine $\mathbf{W}_{com}$ leads to: $\mathbf{c} = \mathbf{X}^{-1}\,\mathbf{x}$, where $\mathbf{X}$ and $\mathbf{x}$ are respectively the 11×11 matrix and the 11-dimensional vector defined by:

$$X_{\alpha\beta} = \sum_i \sum_j w_{ji}^{\alpha} w_{ji}^{\beta} \qquad x_\alpha = \sum_i \sum_j w_{ji}^{\alpha} w_{ji}^{sim}$$

Here the w_{ji}^{α} are the weights in $\mathbf{W}_\alpha$.

Straight summation

The straight summation technique is much simpler than the linear regression technique: $\mathbf{W}_{com}$ is simply the unweighted sum of all 11 soft rule matrices $\mathbf{W}_{\alpha}$. The straight summation of the soft rule matrices is a special case of of the linear combination considered in the previous section: the case in which all coefficients c_{α} have value 1.

In the null case, there is a theoretical basis for expecting that straight summation will produce a $\mathbf{W}_{com}$ that does a reasonable job of combining the capabilities of the individual rules. Imagine an input verb from class β being processed by this $\mathbf{W}_{com}$. The net input to each output unit j is precisely the *sum* of the net inputs contributed by each separate $\mathbf{W}_{\alpha}$: $net_j = \sum_{\alpha=1}^{11} net_j^{\alpha}$. Now for each $\alpha \neq \beta$, by the definition of the null case, $net_j^{\alpha} = 0$. Thus, $net_j = net_j^{\beta}$: the net input to each output unit from $\mathbf{W}_{com}$ is the same as the net input from $\mathbf{W}_{\beta}$ alone; thus, the output of the network under $\mathbf{W}_{com}$ is the same as under $\mathbf{W}_{\beta}$. But, again by the definition of the null case, the output from the soft rule matrix $\mathbf{W}_{\beta}$ for a verb in class β is the correct past tense for that verb.

So, provided it actually is possible to use the null training regime to develop soft rule matrices that satisfy the null specifications, straight summation is a mathematically sound means of soft rule combination.

Null soft rule matrices combined with straight summation can be likened to a rule system in which each rule carries relatively equal weight and functions independently of other rules, and in which rules can be thought of as firing in parallel because the order in which they fire is unimportant. Such a rule system is said to be *free* (Lewis, 1987). Free rule systems probably provide the best analog to the type of rule system we are viewing the connectionist network as embodying.

Summary of the results

To test the hypothesis that $\mathbf{W}_{sim}$ may be viewed as a set of soft rule matrices that have been combined to form a matrix $\mathbf{W}_{com}$, we generated the 11 soft rule matrices $\mathbf{W}_{\alpha}$ for both the exclusive and the null cases, and examined both the linear regression and straight summation techniques for combining them.

To better understand the relationship between the various soft rule matrices we also charted the development during training of the c_{α} coefficients in the linear regression technique. Each c_{α} may be viewed as indicating the weight of the contribution of $\mathbf{W}_{\alpha}$ to the composite matrix $\mathbf{W}_{com}$.

During training of the soft rule matrices we would expect certain soft rule matrices to emerge dominant over others. In particular, we would expect soft rule matrices representing the larger body of regular verbs to dominate soft rule matrices representing the smaller body of irregular verbs. This can be seen in Figure 1. In the first ten training cycles, we trained only on the same ten verbs that Rumelhart and McClelland used for these cycles (those they identified as the ten most frequent verbs in the Kucera and Francis (1967) corpus). Of these ten verbs, eight are irregular; in the first ten cycles, the ratio of the average irregular c_{α} to the average regular c_{α} is very large. As a large body of regular verbs is introduced to the model at cycle 11, this ratio virtually flips, quickly stabilizing at .37 in the exclusive case and 1.01 in the null case.

This reveals three important points: (1) the relative contribution of each soft rule matrix $\mathbf{W}_{\alpha}$ to $\mathbf{W}_{com}$ is established early in training, (2) in the exclusive case there is a substantial imbalance between irregular and regular soft rule matrices, and (3) in the null case the ratio is almost 1 to 1. Thus we might expect a better performance in combining soft rule matrices in the null case than in the exclusive case, because the various soft rule matrices are more nearly alike.

In order to measure the quality of performance of our "rule system," we tested the model separately

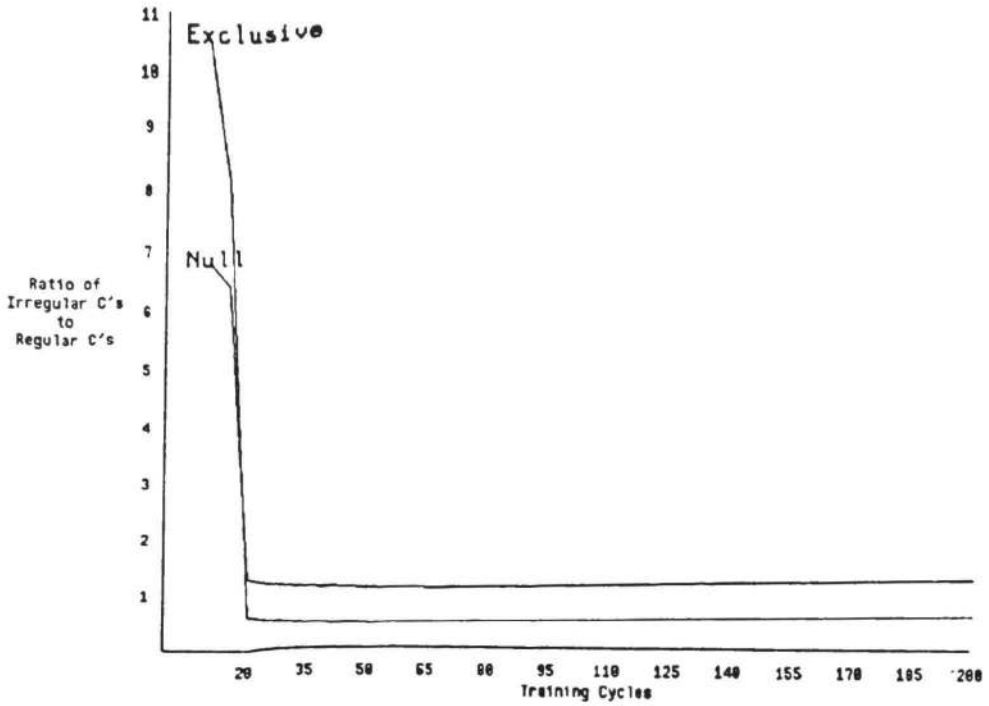

Figure 1: Ratio of average coefficients for regular verbs to average for irregular verbs.

on $\mathbf{W}_{sim}$, on $\mathbf{W}_{com}$, and on each of the soft rule matrices $\mathbf{W}_{\alpha}$. Our measure was the feature match (as defined above) over all the relevant verbs in the training corpus (the entire set for $\mathbf{W}_{sim}$ and $\mathbf{W}_{com}$; the verbs in class α for $\mathbf{W}_{\alpha}$).

We first consider the results in the exclusive case. As shown in Table 1, the average feature match of the soft rule matrices is .97; this is comparable to the feature match of $\mathbf{W}_{sim}$, .95. We may conclude that the soft rule matrices perform on average at the same level as $\mathbf{W}_{sim}$ when presented with words in the corresponding class.

Table 1: Performance (feature match)

Training technique	Original matrix $\mathbf{W}_{sim}$	Average of soft rule matrices $\{\mathbf{W}_{\alpha}\}_{\alpha=1}^{11}$	Composite matrix $\mathbf{W}_{com}$	
			Linear regression	Straight summation
Exclusive	.95	.97	.62	.54
Null	.95	.98	.52	.87

The quality of the two composite matrices in the exclusive case is indicated by a feature match of .54 and .62 for the linear regression and straight summation respectively. Substantially lower than $\mathbf{W}_{sim}$, these figures definitely are on the lower bounds of acceptability, as discussed earlier.

Why is the performance of $\mathbf{W}_{com}$ rather poor when the linear regression produces a matrix that minimizes the difference between its weights and the weights in $\mathbf{W}_{sim}$? Our explanation is as follows: There are an infinite number of matrices that perform the correct input/output transformation on the corpus of verbs across all 11 verb classes. The past tense model employs a method that simply produces *one* such matrix, $\mathbf{W}_{sim}$. Many of the other matrices will have a large difference from $\mathbf{W}_{sim}$, while nonetheless performing identically to $\mathbf{W}_{sim}$. Thus producing a matrix $\mathbf{W}_{com}$ in which there is minimal

difference between the weights in that matrix and $\mathbf{W}_{sim}$ does not guarantee that $\mathbf{W}_{com}$ is the best combination of $\mathbf{W}_{\alpha}$ for approximating the performance of $\mathbf{W}_{sim}$ on the training corpus.[2]

It is perhaps less surprising that the straight summation performs poorly in the exclusive case. While there is a theoretical basis for the soundness of the straight summation combination in the null case, there is none for the exclusive case.

The results of combining the soft rule matrices in the null case are also summarized in Table 1. Again we see that the performance of the individual soft rule matrices is comparable to that of $\mathbf{W}_{sim}$. The performance of $\mathbf{W}_{com}$ generated through linear regression is .52. The same explanation for the rather poor performance of $\mathbf{W}_{com}$ in the exclusive case applies here in the null case.

As predicted by our earlier analysis, the performance of the $\mathbf{W}_{com}$ generated through straight summation of the soft rule matrices is much better: The feature match is .87 for $\mathbf{W}_{com}$ when presented with the full corpus of verbs. In general, with a feature match of .87, we can expect the model's decoding network to consistently generate the correct past tense with very few, and weak, alternatives – if any alternatives are generated at all.

From this result we may conclude that the soft rule matrices in the null case may, in a sense, be viewed as free rules that may be applied separately or combined through straight summation into a single system.

Conclusion

We have shown how a classic PDP model, Rumelhart and McClelland's past tense model, may be decomposed into a set of "soft rule matrices." These rules may be applied separately or combined into a single system. Using the best technique, soft rule matrices trained in the null regime combined using straight summation, we can view the knowledge in this model's weight matrix in four approximately equivalent ways:

(1) Knowledge = $\mathbf{W}_{sim}$: The past tense model is a PDP model consisting of no rules.

(2) Knowledge = $\sum_{\alpha}\mathbf{W}_{\alpha}$: The past tense model is a PDP model in which the knowledge in the weights is a system built (by simple summation) of 11 individual matrices each handling a different subset of the input space.

(3) Knowledge = $\{\mathbf{W}_{\alpha}\}_{\alpha=1}^{11}$: The past tense model is a set of 11 separate noninteracting rules. Each rule is implemented as a PDP network.

(4) Knowledge = $\mathbf{W}_{com} = \sum_{\alpha}\mathbf{W}_{\alpha}$: The past tense model is a rule system combining 11 rules into one single system. The rules, and the way they combine, are defined via connectionist networks. The rules apply independently, in parallel.

In this work we have been exploring the hypothesis that the higher-level perspective provided by rule systems can help us understand the knowledge contained in a PDP network. Another hypothesis worth exploring is that PDP-based "soft" rules of the sort we have been considering might help, in simple domains, to alleviate some of the brittleness that has often plagued systems based on hard rules.

Our explorations clearly constitute the barest beginings. Especially important extensions of our work are to systems with hidden units and to methods for finding rule-decompositions of the sort we used here *automatically* – without the need for a prior (non-connectionist) analysis of the task (provided in our case by Bybee and Slobin). Nonetheless, we are encouraged that our preliminary foray has helped us understand the knowledge contained in a rather inscrutable weight matrix of well over 200,000 weights. We expect further useful results to come from explorations of how conceptual and technical tools of the PDP and rule-based frameworks can be used to strengthen each other.

Acknowledgements

This work has been supported by NSF grants IRI-8609599 and ECE-8617947 to the second author, by a grant to the second author from the Sloan Foundation's computational neuroscience program, and by the Department of Computer Science and Institute of Cognitive Science at the University of Colorado at Boulder.

We thank Dave Rumelhart and Jay McClelland for providing us with the simultation software for their model.

Footnotes

1. In practice, in order to avoid oscillations we set $\delta_j = 0$ if net_j is within a certain tolerance γ of the target value 0. If γ is chosen too small the system will generally fail to converge; on the other hand, $\gamma = \infty$ reduces the null case to the exclusive case. We found that $\gamma = 850$ was an acceptable value; this may seem large, but since there are 460 weights to each output unit, each with an integer weight and many of these greater than 1, the value of 850 is not large compared to a typical net input to an output unit.

2. Recall that our measure of difference is purely based on the weights in the matrices, and not in terms of their responses to the particular inputs in the training or testing sets.

References

Bybee, J. & Slobin, D. (1982). Rules and schemas in the development and use of the English past tense. *Language*, **58**, 265–289.

Hoard, J. & Sloat, C. (1973). English irregular verbs. *Language*, **49**, 107–120.

Kucera, H. & Francis, N. (1967). *Computational analysis of present-day American English.* Providence, Rhode Island: Brown University Press.

Lachter, J. & Bever, T. G. (1988). The relation between linguistic structure and associative theories of language learning—A constructive critique of some connectionist learning models. *Cognition*, **28**, 195–247.

Lewis, C. (1987). Composition of productions. In Klahr, D., Langley, P., & Neches, R. (Eds.), *Production system models of learning and development*, 329–359. Cambridge, MA: MIT Press.

Pinker, S. & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, **28**, 73–193.

Rumelhart, D. E. & McClelland, J. L. (1986). On learning the past tenses of English verbs. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group, *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 2: Psychological and biological models*, 217–271. Cambridge, MA: MIT Press/Bradford Books.