Probing Neural Network Generalization using Default Patterns

Brandon Prickett[*1], **Tianyi Niu**[*2], **Katya Pertsova**[2]
University of Massachusetts Amherst[1], University of North Carolina at Chapel Hill[2]
bprickett@umass.edu, tianyin4@email.unc.edu, pertsova@unc.edu

Abstract

Whether neural-net models can learn minority-default patterns has been a matter of some controversy. Results based on modeling real human language data are hard to interpret due to their complexity. Therefore, we examine the learning of a simple artificial language pattern involving defaults using three computational models: an Encoder-Decoder RNN, a Transformer Encoder, and a Logistic Regression. Overall, we find that the models have the hardest time with minority defaults, but can eventually learn them and apply them to novel words (although not always extend them to completely novel segments or novel CV-sequences). Type-frequency has the largest effect on learning in all models, trumping the effect of distribution. We examine the weights of two models to provide further insights into how defaults are represented inside the models.

1 Introduction

Some linguistic patterns (e.g., English past tense regular inflection) are best stated using *default* rules – rules for a heterogeneous category that apply as a last resort when more specific rules are not applicable.

An early connectionist model of morphological learning by Rumelhart and McClelland (1986) was criticized by proponents of rule-based approaches for its inability to appropriately handle defaults (Pinker and Prince, 1988). In particular, it was claimed that the default behavior of the model could only hold under the condition of high frequency of the default (regular) category (although see work by Hare et al. (1995)).

More recent work by Kirov and Cotterell (2018) using an Encoder-Decoder (ED) RNN model also focused on the English past tense problem. However, Corkery et al. (2019) found that this model was unstable across simulations and the averaged aggregated performance still did not provide a good fit to human data. Similarly, McCurdy et al. (2020), using the same model as Kirov and Cotterell (2018) investigated the learning of the German plural allomorphy, which has been claimed to involve a minority default pattern (Marcus et al., 1995). They found that while their model achieved 88.8% accuracy on the held-out dataset, it did not match human performance in a few critical ways. In particular, unlike humans, it failed to generalize the two apparent default suffixes -en and -s (where -s is the least frequent suffix) to unusual novel words. Another study compared the behavior of an ED model and a Transformer model on both the English past tense (majority default) and the German plural (minority default) (Beser, 2021). The fit to human data was still weak, and the models were susceptible to some analogical errors.

The modeling work described above involving actual languages is hard to interpret since so many factors play a role: for example, it is controversial to what extent the German -s suffix is in fact a default (Zaretsky et al., 2016), the specific patterns are often lexically idiosyncratic and the factors conditioning the suffixes range from semantics to phonology and not all of them are included in models' training. In this work, we aim to address these problems by focusing instead on artificial languages designed to isolate the specific patterns of interest. Our goal is to abstract away from the irrelevant complexities of natural language and focus on understanding how default patterns (including minority defaults) are represented and learned in a neural net model. We define "default" to be a category whose distribution cannot be described with a simple conjunction of features, but can be stated either as (a) a complement of other conjunctive categories or (b) a complement of the distribution of a set of idiosyncratic lexical items (exceptions). Note that this is a definition of what a default distribution

[*]Equal contribution
[1]Code and data available at https://github.com/tianyiniu/Min_Default.

looks like in the data, not how it is represented in the mind/model. In this paper, we focus on testing case (a) above. Our research questions are:

1. Under what conditions are defaults generalized to a wider set of contexts than the trained contexts and what does this tell us about how they are represented?

2. How does frequency interact with "defaultness"? E.g., what is the difference in learning defaults when they are most frequent, least frequent, and have the same frequency as other categories? Does the default status (e.g., heterogeneous distribution wrt. to other categories) trump frequency or the reverse?

What we find is that all of the models we tested are able to learn and represent default patterns, including the minority default. However, minority defaults are learned slower compared to any other category. For all of our models frequency has the largest effect on learning: low-frequency categories are learned slower compared to high-frequency ones. However, the effect of the distribution also plays a role, albeit a smaller one: patterns that have a wide heterogeneous distribution (defaults) are learned slower compared to those with a more narrow distribution. These results are in line with human learning as we will discuss later. Additionally, the models made some errors on novel stimuli that included segments not seen in the training or CV-combinations not seen in the training. The rule-based models predict that true defaults should occur in all unusual circumstances, or when the learner does not know what to do. From the three models we tested, the Transformer model was the closest to this type of behavior (but more testing is required). Overall, we conclude that neural networks can represent minority defaults, but these patterns are difficult to learn both because they have low frequency and because they are more variable by having wider distribution.

2 The artificial language stimuli

2.1 The general pattern

Our artificial language pattern was modeled on the allomorphy of the English regular plural suffix -/z/ which has the following general structure:

1. **If** a word ends in a sibilant, use suffix A (natural class "conjunctive" category)

2. **ELSE if** a word ends in a voiceless consonant, use suffix B (intermediate default)

3. **ELSE** use suffix C (global default)

However, unlike English, where A, B, C are phonological variants of each other, we used three phonologically distinct strings [wa] (suffix A), [ji] (suffix B), and [lej] (suffix C), as though this was an instance of phonologically conditioned suppletive allomorphy.

We then generated several versions of this pattern varying the frequency of the global default category. In the Equal Frequency condition, each suffix was equally frequent, in the Majority Default condition, 90% of words had suffix C, and in the Minority Default condition, suffix C occurred 10% of the time. Table 1 summarizes the number of items and item types in each condition during training.

2.2 The stimuli

The stimuli were generated using a bigram model trained on the CMU pronouncing dictionary. The words were restricted to the following inventory of segments: consonants [p, t, tʃ, k, b, d, ʒ, g, f, s, ʃ, v, z, dʒ, m, n, ŋ, l, h] and vowels [i, ɪ, u, ʊ, ej, ɛ, o, ɑ]. The templates for the words came from this list: CVC, CVCVC, VCVC, CVCV, VC, CVCC, CCV (where C=consonant and V=vowel).

We used several different types of stimuli to test the models' generalization behavior: novel words of the same types seen in training (same segments and syllable templates), novel words of unseen templates (VC, CVCC, and CCV shapes were withheld from training), novel words ending in unseen segments (words ending in [l] and [h] were withheld from training), and novel words which we call "mutants"– these were derived by taking words from the training and changing their last segment to a different segment that categorized the word into a different suffix-class. These stimuli were designed to tease apart several possible generalization strategies a model could use. The stimuli ending in novel segments were meant to test whether the model would generalize based on phonetic similarity of these segments to those seen during training. The novel templates were included to test whether the model learned to generalize based on the last segment alone, or whether it was taking the word-shape into account. Mutants were included to test whether overall similarity (not just last-segment features) would play a role.

Condition	Category A (natural class)	Category B (default 1)	Category C (default 2)
Equal frequency 33.3% each	132 of each cvc, cvcvc, vcvc (total 396)	132 of each cvc, cvcvc, vcvc (total 396)	99 of each cvc, cvcvc, vcvc, cvcv (total 396)
Majority default default: 90%	39 of each cvc, cvcvc, vcvc (total 117)	39 of each cvc, cvcvc, vcvc (total 117)	238 of each cvc, cvcvc, vcvc, cvcv (total 952)
Minority default default: 10%	177 of each cvc, cvcvc, vcvc (total 531)	177 of each cvc, cvcvc, vcvc (total 531)	31 of each cvc, cvcvc, vcvc, cvcv (total 124)

Table 1: Distribution of training stimuli.

3 The models

We used three classification models, an Encoder-Decoder RNN, a Transformer Encoder and a multi-class logistic regression.

For all of these models, the inputs was a sequence of numerical feature vectors representing sequences of sounds. For example, the sound [t] corresponds to a vector that includes a value of -1 for the feature corresponding to its voicing, a value of +1 for the feature consonantal, a 0 for the feature corresponding to its roundness (since that feature is only relevant to vowels), and so on (see Appendix for the features we used). Outputs used a single vector of three numerical features, each of which corresponded to one of the suffixes. An output value of 1 in training indicated that that feature's suffix was being assigned to the input stem.[1]

3.1 Encoder-decoder neural network

Following Kirov and Cotterell (2018) and McCurdy et al. (2020), we used an ED neural network, implemented using Keras (Chollet et al., 2015). These differ from the kind of feed-forward networks used by Rumelhart and McClelland (1986) in two crucial ways: (1) they have recurrent connections, which act as a kind of memory, allowing the model to process data sequentially, and (2) they are made up of two separate networks (the encoder and the decoder). The encoder processes an input into an intermediate representation that is passed through recurrent connections to the decoder, which process that into an appropriate output.

Due to the fact that we were using a simpler artificial language and a categorization task, our ED model needed fewer parameters than the ones used by Kirov and Cotterell (2018) and McCurdy et al. (2020). The encoder and decoder had only a single layer each, with 32 nodes in each layer.

For training, we used a learning rate of 0.0005, the algorithm RMSProp (Hinton et al., 2012), batch sizes of 32, 100 epochs, no attention, and hyperbolic tangent as the activation function for all layers.[2] We ran our model for 10 separate runs in each condition, with each run using a unique, randomly sampled set of starting weights. Performance on test data was evaluated after each epoch to track the model's generalization throughout learning.

3.2 Transformer Encoder neural network

We also explored whether the Transformer architecture proposed in Vaswani et al. (2017) can successfully learn various default conditions. Since we currently frame our experiment as a classification problem, we use only the encoder. The Transformer architecture differs from the ED network significantly in that it does not rely on recurrent connections. Instead, using self-attention allows the model to attend to all positions in the input simultaneously. Consequently, the Transformer uses positional encoding to represent order, rather than relying on the inherent sequential nature of the input sequence.

For comparability of results, we match the hyperparameters of the Transformer encoder model and the ED model as closely as possible. The model contains only 1 layer, with 1 attention head. We use a 32-dimensional internal representation and a 64-dimensional hidden layer for the feed-forward block. Using these hyperparameters, the Transformer encoder model contains 8.5K parameters, which is comparable to the 10K of the ED model. During training, we used the Adam optimizer (Kingma and Ba, 2015) with a default learning rate of 0.001, a batch size of 32, and 100 epochs.

[1] Early results suggest that the models' learning trajectories are sensitive to this output representation. However, we leave a proper search over potential output representations to future work.

[2] After running models with and without attention (Bahdanau et al., 2014) in all conditions, we found no significant effects on the tasks explored here. Other hyperparameters were not exhaustively tested (due to limited resources) and were chosen based on a small amount of pilot testing.

We trained our model for five separate runs in each condition, using a randomized set of starting weights. Similar to the ED model, we evaluated performance on test data after each epoch.

3.3 Logistic regression

Since complex neural networks can be difficult to interpret, we also ran a multi-class logistic regression classifier (LR) on the same pattern. LR models are closely related to Maximum Entropy Grammars (Goldwater and Johnson, 2003; Moreton et al., 2017), a common theoretical approach to morphological and phonological learning, and are relatively easy to interpret because of the straightforward relationship between the weights they learn and the patterns those weights represent.

As logistic regression requires all inputs to be flattened into a one-dimensional vector, a pooling function is required. We tested two versions of the LR model, one *pool-concat* (LR-PC), in which the data was presented as a concatenation of feature-vectors for each segment. We also introduced a padding segment (where all features values are set to 0) to pad the length of each word to 5 segments (the longest word in our corpus). Additionally, we trained a simpler model, *pool-last* (LR-PL), that was trained only on the last segment of each word. Since the default pattern we are examining is solely dependent on the final character, this pooling method directed the model to look only at the relevant features making the task easier (at the cost of being less analogous to the other models' input).

Each LR model was randomly initialized and trained for 3 epochs, using a batch size of 10 and a fixed learning rate of 0.01[3]. Because the model performs a step of gradient descent after being exposed to a batch of 10 random words, there is no guarantee that all three suffixes are represented in a single batch. For reliability and robustness, 10 models were trained in parallel. Accuracy for each suffix was calculated using the held-out test stimuli after each batch of training. We then averaged the accuracy for each of the 10 models to create the plots discussed in §4. This procedure was repeated for all three conditions.

[3]The current learning rate selected to balance between learning the suffix pattern efficiently and obtaining a gradual enough learning curve to differentiate between the different default conditions.

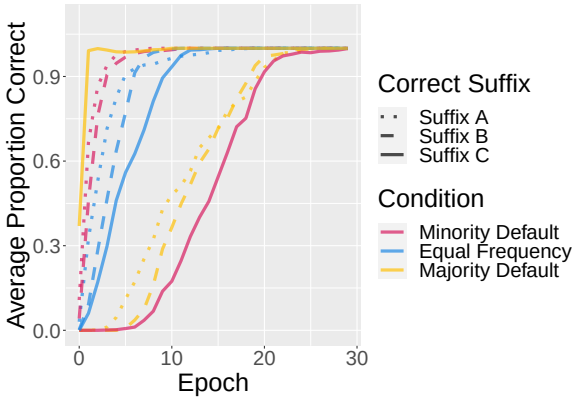

Figure 1: ED network's average performance on the test data for the first 30 epochs of learning, broken down by condition and the correct suffix for each word.

4 Results

The accuracy at the end of learning for the three models (omitting LR-PC) is summarized in Table 2. The performance of all models is very high in all conditions. Note that this high performance is not indicative of overfitting as the model was tested on different stimuli than the stimuli in the training. The perfect performance on the regular test stimuli (the column "Test") shows that all models learned the relationship between the suffix and the features of the last segment. This relationship was for the most part successfully generalized to novel segments ('h' and 'l') not present in the training and to novel templates with a few exceptions discussed below. Novel words, templates, and segments represent the levels of generalization that have been demonstrated in humans for this kind of task (for more discussion on the scopes of generalization in morphology, see Prickett et al., 2022; Berent, 2013).

4.1 ED model results

After about 25 epochs of training, the ED model performs perfectly except when tested on new templates (which will be discussed below).

4.1.1 Performance over the course of learning

Figure 1 shows the ED model's generalization over time to novel words that were made up of the same segments and templates as the training data. These results demonstrate the model's sensitivity to suffix frequency. Within each condition, the most frequent suffix is always the one that's learned the quickest (where 'learned' is defined as having a test accuracy close to 1). There's also an effect of

Model	Language	Train	Test	Mutant	New Template	[h]-final	[l]-final
ED	Maj. Def.	1.00	1.00	0.97	0.86	0.96	1.00
ED	Equal Freq.	1.00	1.00	0.98	0.81	1.00	0.99
ED	Min. Def.	1.00	1.00	1.00	0.83	1.00	0.89
Transformer Encoder	Maj. Def.	1.00	1.00	1.00	0.82	0.73	0.96
Transformer Encoder	Equal Freq.	1.00	1.00	1.00	0.76	1.00	0.97
Transformer Encoder	Min. Def.	1.00	1.00	1.00	0.83	1.00	0.84
LR (pool-last)	Maj. Def.	1.00	1.00	n/a	n/a	1.00	1.00
LR (pool-last)	Equal Freq.	1.00	1.00	n/a	n/a	1.00	1.00
LR (pool-last)	Min. Def.	1.00	1.00	n/a	n/a	1.00	0.00

Table 2: Accuracy results, by model and condition. The "Test" column refers to testing data that used novel words with the same segments and templates seen in training. The "Mutant" and "New Template" statistics for LR (pool-last) is n/a due to the pooling function cropping out all but the last segment, making these two tests irrelevant.

distribution—when frequency doesn't play a role (e.g., in the equal frequency condition), the default suffix is still at a disadvantage compared to the suffixes with more narrow distributions. However, the model *is* eventually able to learn the minority default and correctly generalize it to novel words.

Another aspect of the ED model's generalization that is not shown in the graph is that, at the end of learning, it tends to do worse on novel items that end in the unseen last segment [l] than it does on novel items that end in the similarly unseen segment [h] in the Minority Default condition. The other models show similar behavior which we will discuss later.

4.1.2 Performance on Mutants

The so-called "mutants" were the words that were generated by mutating the final segment in one of the training words. These items were included to test whether the model was driven by the overall similarity to the training data (which should be high for the mutant words), or by the relevant properties of the last segment. Given high performance on these items, we conclude that the ED model was not significantly affected by the overall segmental similarity to the training data and made its decisions based on the last segment.

4.1.3 Performance on Novel Templates

Table 2 shows that, in general, the ED model performs relatively well, but not perfectly on words that consist of unseen shapes/templates (that is, new orderings of consonants and vowels). When we investigated the model's performance on this set of test data, we saw that the model performed perfectly on the novel templates VC and CCV. The only template the model struggled to correctly classify was CVCC. At the end of the learning, the model's output for CVCC inputs was seemingly random. We could find no simple explanation for why the model had trouble with this specific kind of test datum, and leave further investigation of this to future work. It is possible that the model's generalizations were based not just on the last segment, but last two segments.

4.2 Transformer Encoder results

The accuracy results for the Transformer Encoder model at the end of learning and the learning curves are similar to the ED model (see Figure 2). The default suffix (suffix C) is overgeneralized in the Majority Default condition at the early stages of learning. In the other two conditions, Suffix A and B are learned first, equally well and relatively fast. Thus, this model also shows a very pronounced frequency effect: the suffixes that have the lowest frequency (suffix C in the minority default and suffixes A and B in the majority default conditions) are learned the slowest. Likewise, suffixes with the highest frequency (suffix C in the majority default and suffixes A and B in the minority default) are learned the fastest. However, the effect of distribution is also present: it can be seen in the Equal Frequency condition, where the global default Suffix C is learned a little bit slower compared to the other two suffixes.

4.2.1 Performance on Novel Templates

The Transformer Encoder model, like the ED model, makes some mistakes when exposed to novel word templates. In all three stimuli conditions, the VC template results in the most frequent mistakes. Interestingly, this is different from the ED model for which the CVCC template was

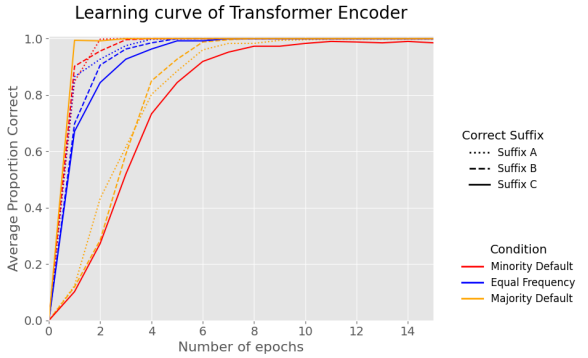

Figure 2: Transformer Encoder's performance on the test data after the first 15 epochs of training, broken down by condition and the correct suffix for each word.

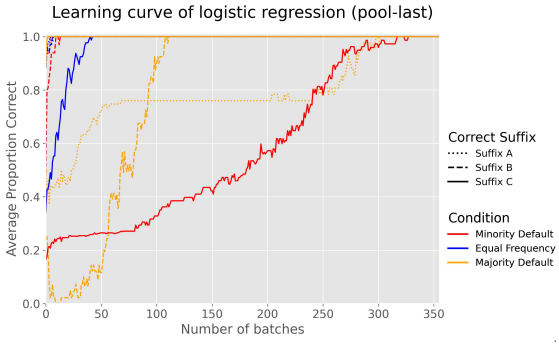

Figure 3: LR-PL average performance on the test data after each batch, broken down by condition and the correct suffix for each word.

more problematic. Thus, the presence of consonant clusters not seen in the training did not present a problem for the Transformer. These differences between the models might be due to how they treat inputs: Transformer encoders are able to process all input segments at once, while the ED model processes inputs sequentially.

The most common mistake the model made on the VC templates was classifying words that should get Suffix A or Suffix B as suffix C stems (the global default). Summing over conditions and runs, we see 127, 169, 120 incorrect predictions in the new template test items in the Maj Def, Equal Freq, and Min Def conditions, correspondingly. Within these, 80%, 86%, 83% are due to incorrectly predicting the default suffix C. (All of these incorrect predictions occur when the input template is VC.) This result suggests a tendency of the model to overgeneralize the default when exposed to certain types of novel data. Note that the VC template not only was absent in the training, it also presented a new word-length: none of the words in the training were shorter than 3 segments. Our current guess is that this fact is responsible for why the model struggled with this template (see more on this in section 4.4.1).

4.2.2 Performance on Mutants

The model achieves perfect accuracy on the mutant test stimuli in all default conditions. This suggests that similar to the results obtained from the ED model, the Transformer does not seem to be affected by segmental similarity. Rather, it has learned to correctly identify suffix based on the final segment.

4.2.3 Performance on Novel Segments

The model can consistently generalize to novel segments when trained in the Equal Frequency condition. However, the model fails to consistently predict the correct suffix for some novel segments in the Majority Default and Minority default conditions. In the Minority Default condition, the model sometimes predicts suffix A for [l], which should be classified as suffix C (since it is a voiced non-strident). This behavior is explained in the next section when we examine a more transparent logistic regression model. In the Majority Default condition, the model makes mistakes by predicting suffix C for [h], which should belong to low-frequency category B (since [h] is a voiceless non-strident). Again, this suggests that the default suffix is overgeneralized in a situation that is novel or unusual.

4.3 Logistic regression results

LR-PC with left padding correctly identified that the final segment determines the suffix. Moreover, LR-PC with either left or right padding identified the same feature patterns for distinguishing between suffixes as LR-PL (see Figure 4). For this reason (and for reasons of space) we will focus on LR-PL here, the model trained on the last segment only. All references to LR will exclusively refer to LR-PL.

LR generalization proceeds mostly in the order of frequency (as shown in Figure 3). One difference between the neural net models and the LR model is the relatively slow acquisition of suffix A in the Majority default condition by the LR model. In the LR model suffix B is learned a lot faster compared to suffix A, despite the fact that they have equal frequency and suffix A forms a natural class.

4.3.1 Performance on Novel Segments

The LR completely fails to correctly categorize the novel segment [l] in the minority default condition despite successful generalization in the equal frequency and majority default conditions. This section analyzes this phenomenon, which may also shed light onto how training-data distribution affects learning in neural-based models. Recall that those models also made mistakes with [l] in the minority default condition (although they still performed over 80%).

The aforementioned inability to correctly classify [l] is due to slight differences in learned weights across the three default conditions, as well as the unique phonological features of [l]. The segment [l] is underspecified for [voice] and [strid], the two most significant features identified by the model (as discussed in the next section). Although two other features are highly weighted, [son] and [cor], with [+cor] being strongly correlated with suffix A and [+son] strongly correlated with suffix C. As [l] is positive for both features, the model must make a classification based on minute differences between the weights. In the Minority Default condition, the conflict is not resolved correctly because the weight of [son] is lower (+1.4) than the weight of [cor] (+1.8). This difference is due to the model not seeing as many sonorant sounds in the minority default condition, so the weight for this feature does not move as fast.

Interestingly, the model does not make the same misclassification for the segment [n] despite it also being [+son, +cor]. We believe this is expected as [n] was present in the training data (unlike [l]), and hence the composite effects of all learned features allows [n] to overcome the lower suffix C weight for [+son]. On the other hand, in both equal and majority default distributions, [nas] has a medium-strong positive weight for suffix C (0.65), thus contributing to the correct classification between suffix A and suffix C.

Since this model was trained on the last segment only, we did not have the novel template and the mutant test-conditions.

4.4 Model Analysis

4.4.1 Learned Representation Across Models

We propose two methods for understanding how the different models learned the phonological patterns. Since the weight matrix of LR is easily extractable and interpretable, we plot the learned weights as a heat map (see Figure 4). However, the same cannot be done for the neural net models. Instead, we plot a saliency map from the Transformer encoder, which is obtained by calculating the gradient of a particular output class with respect to an input. In figure 5, we first separate all test inputs into their ground truth suffix class. For each suffix class, we then calculate the saliency map of all input-features and average the gradient matrix. Because all training inputs are comprised of either 3, 4, or 5 segments, we do not see any meaningful patterns in the saliency map in time steps 0 and 1. This is most likely why the Transformer model fails on the novel template VC which is 2 segments long. In effect, the model cannot represent the concept “last segment".

It must be noted that while the weight matrix from the LR model and the saliency maps from the Transformer encoder model are not obtained by the same mechanism, they are comparable in the sense because they both offer insights into which features are relevant to their respective predictions.

These plots reveal that both models learned similar patterns. For words that used suffix A, a large positive weight is assigned to [strid] and [cor] (that is, “strident” and “coronal”). This is expected, since the natural class of sibilants is defined as [+strident], and all stridents happen to also be [+coronal] (although, as in English, not all coronals are strident). For the default suffixes B and C, [voice] determines which class they are assigned to. Consequently, we see that [voice] ends up with a high negative weight for class B and a high positive weight for class C. Moreover, a strong positive weight is assigned to [son] in Class C due to the fact that vowels which are sonorants, but are underspecified for voice, also belong to the default class C.

Importantly, these graphs show that the LR model and the neural networks capture a default pattern through competition among multiple classes. For example, take a segment like [s], which is both voiceless (so could potentially fit into class B) and sibilant (which means it fits into class A). As long as the cumulative weight of [strid] and [cor] is greater than the weight of [voice], this segment will be assigned to the correct class, A.

5 Discussion

It appears that all models we looked at were able to learn the simple artificial language pattern they

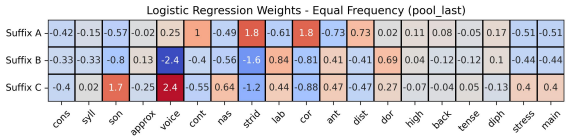

Figure 4: Final weights for pool-last in the Equal Frequency condition. A deeper shade of blue suggests a large negative weight, and a deeper shade of red suggests a large positive weight.

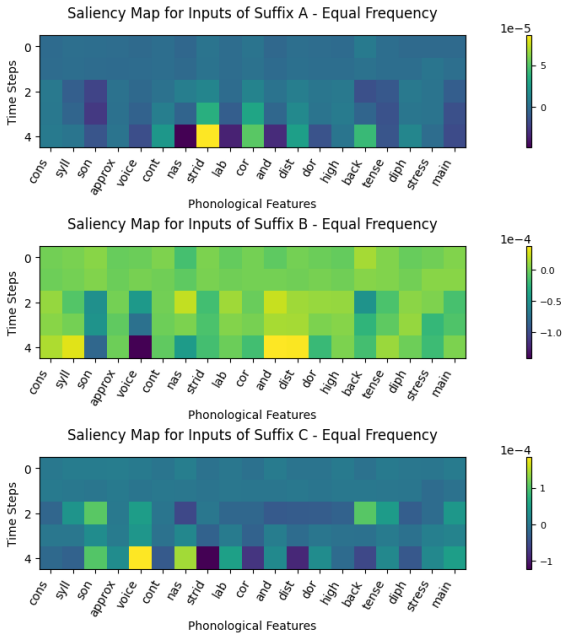

Figure 5: The averaged saliency map for inputs of each suffix class. A deeper shade of blue suggests higher sensitivity of the output to a negative change in the features, and a deeper shade of yellow suggests a higher sensitivity of the output to a positive change.

were trained on, and successfully generalize to novel items that looked similar to training data. However, when tested on items that contained novel word-final segments, or novel CV-templates, the models produced some errors. In particular, all models struggled with the novel segment 'l' in the Minority Default condition, because this segment is expected to belong to the default pattern which was exemplified only by 10% of the data and because this segment is featurally similar to segments in the sibilant class (by being a coronal consonant). When the default class is more frequent, the models are able to overcome this difficulty and correctly categorize inputs ending in 'l'. The Transformer model also had some difficulties with the novel segment 'h' in a condition in which the category to which it belongs (suffix B) is low frequency. Another source of errors were words from different templates: the ED model struggled with the novel CVCC template, while the transformer model struggled with the VC template. We hypothesize that the humans would not have difficulties with such novel templates and would generalize based on the features of the last segment. However, the current models are either not abstracting away the concept "last", or are making their decisions based on the shape of seen words, and not just the features of the last segment.

Our analysis of the two models' weights revealed that different categories have different assignment of feature weights so as to maximize the probability of an input falling into a specific class if it shares more highly weighted features with that class compared to all the other classes. This inherent competition among classes is important for creating the apparent default structure. The networks find such weight configurations so as to ensure that inputs sharing features with both a default and a narrow class, will be more likely to be assigned to the narrow class (capturing the principle that the more specific classes "block" to the more general ones). This typically means that the cumulative weight of all relevant features in the default class will be lower than the cumulative weight of features in the non-default class.

Overall, we also see that the frequency of a pattern has the largest effect on learning: unsurprisingly the more frequent the pattern, the faster it is learned. When frequency is held constant, patterns that have a more narrow distribution in the phonological space are learned better. This is consistent with human data. For example, studies on category learning in the domain of visual perception report that more diverse categories with a wide distribution are learned worse than those with a narrow, clustered distribution (Fried and Holyoak, 1984; Hahn et al., 2005). Artificial language learning studies that investigated the effects of frequency and distribution on learning defaults find the same thing (Nevat et al., 2018; Pertsova et al., 2024). The reliance of network models on both frequency and distribution has also been found before, including in work on modeling minority defaults. In particular, Feldman (2005) found that testing a simple three-layer perceptron trained on German plural with inputs that had all features removed produced responses that were either -n (the most frequent suffix) or -s (the most widely distributed suffix).

We are pursuing several directions for extending this work. First, we are considering alternative

ideas for testing the “across-the-board" property of defaults that are predicted by some rule-based models. In particular, it has been argued that regular inflection (of the default type) will automatically apply to abbreviations, acronyms, proper names, non-native sounds, quotations, and when memory or processing fail (Marcus et al., 1995). It is not obvious what the best analog of such circumstances would be for our models. But one idea we are pursuing is testing our models on some words with an entirely new subset of features which were set to 0 in the training, but which are contrastive in the testing. In our current setup, the closest test for generalization to "unusual" cases are the novel templates and novel segments. As previously mentioned, the ED model produced seemingly random behavior when tested on CVCC templates, affected by the absence of word-final clusters in the training data. On the other hand, the Transformer model struggled with the novel VC words and ‘h’-final words in one of the conditions. Interestingly, this model tended to use the default suffix when making errors (although not always), which is more similar to the default-behavior one would expect from a rule-based model.

Thus, we take this as preliminary evidence that global defaults (those that would apply to any new or unusual situation) can arise in a non-symbolic system. More testing (along the lines suggested above) would be required to confirm this claim.

Another extension of our work is to make the target pattern gradually more complex, and therefore more realistic. Instead of a structure where the “exceptions" (the more narrow categories) are deterministically defined, we’d like to test a situation that is more similar to the English past tense, namely when the exceptions do not form a natural class and partially overlap with the default category (e.g., monosyllabic verbs ending in coronal stops are typically irregular like *put*, but can also be regular like *bat*).

Finally, in our experiments we only varied type-frequency, while the token-frequency of each word was the same (=1). In the future, we would like to also investigate how the models respond to changes in both type and token frequencies as these are known to play different roles in language acquisition.

Acknowledgments

The authors would like to thank Joe Pater, Michael Becker, Gaja Jarosz and the students in UMass’s Fall 2024 Morphophonology Seminar for feedback on an early version of this project.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Iris Berent. 2013. The phonological mind. *Trends in cognitive sciences*, 17(7):319–327.

Deniz Beser. 2021. Falling through the gaps: Neural architectures as models of morphological rule learning. *ArXiv*, abs/2105.03710.

François Chollet et al. 2015. Keras. `https://keras.io`.

Maria Corkery, Yevgen Matusevych, and Sharon Goldwater. 2019. Are we there yet? encoder-decoder neural networks as cognitive models of english past tense inflection. *arXiv preprint arXiv:1906.01280*.

Naomi Feldman. 2005. Learning and overgeneralization patterns in a connectionist model of the german plural. MA thesis, U. of Wien.

Lisbeth S Fried and Keith J Holyoak. 1984. Induction of category distributions: a framework for classification learning. *Journal of experimental psychology: Learning, memory, and cognition*, 10(2):234.

Sharon Goldwater and Mark Johnson. 2003. Learning ot constraint rankings using a maximum entropy model. In *Proceedings of the workshop on variation within Optimality Theory*, pages 111–120.

Ulrike Hahn, Todd M Bailey, and Lucy BC Elvin. 2005. Effects of category diversity on learning, memory, and generalization. *Memory & cognition*, 33(2):289–302.

Mary Hare, Jeffrey L Elman, and Kim G Daugherty. 1995. Default generalisation in connectionist networks. *Language and cognitive processes*, 10(6):601–630.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. 2012. Neural networks for machine learning lecture 6a, overview of mini-batch gradient descent.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Christo Kirov and Ryan Cotterell. 2018. Recurrent neural networks in linguistic theory: Revisiting pinker and prince (1988) and the past tense debate. *Transactions of the Association for Computational Linguistics*, 6:651–665.

Gary F. Marcus, Ursula Brinkmann, Harald Clahsen, Richard Wiese, and Steven Pinker. 1995. German inflection: The exception that proves the rule. *Cognitive Psychology*, 29(3):189–256. Copyright: Copyright 2017 Elsevier B.V., All rights reserved.

Kate McCurdy, Sharon Goldwater, and Adam Lopez. 2020. Inflecting when there's no majority: Limitations of encoder-decoder neural networks as cognitive models for german plurals. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1745–1756.

Elliott Moreton, Joe Pater, and Katya Pertsova. 2017. Phonological concept learning. *Cognitive Science*, 41(1):4–69.

Michael Nevat, Michael T Ullman, Zohar Eviatar, and Tali Bitan. 2018. The role of distributional factors in learning and generalising affixal plural inflection: An artificial language study. *Language, Cognition and Neuroscience*, 33(9):1184–1204.

Katya Pertsova, Esther Chen, and Brandon Prickett. 2024. Effects of frequency and distribution on learning minority defaults. *Supplemental Proceedings of the 2023 Annual Meeting on Phonology*.

Steven Pinker and Alan Prince. 1988. On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, 28(1-2):73–193.

Brandon Prickett, Aaron Traylor, and Joe Pater. 2022. Learning reduplication with a neural network that lacks explicit variables. *Journal of Language Modelling*, 10(1):1–38.

David E Rumelhart and James L McClelland. 1986. On learning the past tenses of english verbs. *Psycholinguistics: Critical Concepts in Psychology*, 4:216–271.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Eugen Zaretsky, Hans-Helge Müller, and Benjamin P Lange. 2016. No default plural marker in modern high german. *Italian Journal of Linguistics*, 28(1):1–28.

Appendix

	cons	syll	son	approx	voice	cont	nas	strid	lab	cor	ant	dist	dor
p	+	0	-	0	-	-	0	0	+	-	0	0	-
b	+	0	-	0	+	-	0	0	+	-	0	0	-
t	+	0	-	0	-	-	0	-	-	+	+	-	-
d	+	0	-	0	+	-	0	-	-	+	+	-	-
k	+	0	-	0	-	-	0	0	-	-	0	0	+
g	+	0	-	0	+	-	0	0	-	-	0	0	+
ŋ	+	0	+	-	0	0	+	0	-	-	0	0	+
m	+	0	+	-	0	0	+	0	+	-	0	0	-
n	+	-	+	-	0	0	+	0	-	+	+	-	-
l	+	-	+	+	0	0	0	0	-	+	+	-	-
f	+	0	-	0	-	+	0	0	+	-	0	0	-
v	+	0	-	0	+	+	0	0	+	-	0	0	-
s	+	0	-	0	-	+	0	+	-	+	+	-	-
z	+	0	-	0	+	+	0	+	-	+	+	-	-
ʃ	+	0	-	0	-	+	0	+	-	+	-	+	-
ʒ	+	0	-	0	+	+	0	+	-	+	-	+	-
tʃ	+	0	-	0	-	-	0	+	-	+	-	+	-
dʒ	+	0	-	0	+	-	0	+	-	+	-	+	-
h	+	0	-	0	-	+	0	-	-	-	0	-	-

Table 3: Feature values, for the first 14 features. Vowels and features that were only relevant to vowels (high, back, tense, diphthong, and main stress) are not given here, since they did not play a role in the patterns we used.