L_0-regularization induces subregular biases in LSTMs

Charles Torres
University of California, Irvine
Language Science
charlt4@uci.edu

Richard Futrell
University of California, Irvine
Language Science
rfutrell@uci.edu

Introduction Ongoing work attempts to identify the formal language patterns in natural language. In phonology, recent work has identified the subregular languages as a good candidate (Heinz, 2018). However, there remain few explanations for the source of this bias. This abstract proposes a means of investigating formal language learnability. We propose using a variant of minimum description length (MDL) as defined on LSTMs with varying priors on LSTM size. We will show its utility on a test case from Heinz and Idsardi (2013) and Rawski et al. (2017).

Methods The subregular hypothesis is that phonological patterns occupy a well-defined subset of the regular languages (Heinz, 2018). It has enjoyed empirical success, with laboratory experiments demonstrating these preferences in artificial language learning studies (Lai, 2015; Avcu and Hestvik, 2020; McMullin and Hansson, 2019). But explanations for the existence of this bias are lacking. A minimum description length (MDL) or simplicity principle, where the shortest encoding of input data is preferred (Grünwald, 2000; Chater and Vitányi, 2003), is an enticing explanation, but it fails in most explored representation systems (Heinz and Idsardi, 2013). We consider Long Short-Term Memory networks (LSTMs) (Hochreiter and Schmidhuber, 1997) as an alternative representation system for their flexibility in learning formal languages (Weiss et al., 2018), and show that constraining their complexity induces a subregular bias.

With LSTMs, a natural choice of description length is the number of parameters, or number of connections between neurons. Functionally, this means training networks using L_0-regularization, which penalizes for number of nonzero parameters. While it is generally undifferentiable, we use a differentiable sampling technique from Louizos et al. (2017). We keep the architecture fixed (see Appendix A) in order to control for other sources of variation in LSTM complexity.

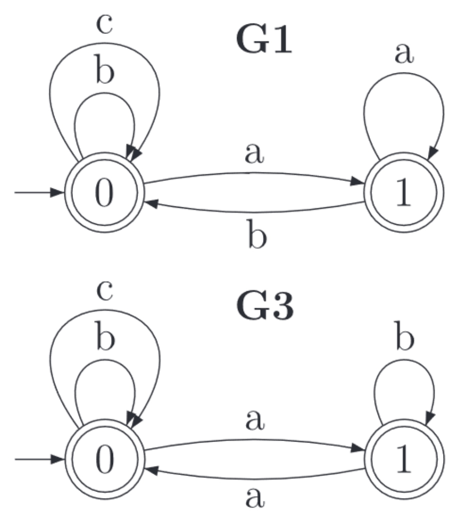

Figure 1: Two regular grammars. $G1$ is subregular and strictly local, $G3$ is a counting language and not subregular. Figures adjusted from Heinz and Idsardi (2013).

Our experiment concerns an open question from Heinz and Idsardi (2013). Consider two formal grammars from their paper (depicted as finite state automata (FSA) in Fig. 1). These have equal description lengths as FSAs, but G_1 is subregular and governed by local constraints whereas G_3 is a counting language and not subregular. G_1 is more language-like and thus its purported preference represents an open puzzle for simplicity-based accounts.

To assess this preference using computational complexity we train 5 LSTMs each with 45 different regularization penalties to vary resulting LSTM complexities (N=225). Each LSTM is trained on words drawn from the intersection between G_1 and G_3 using the cross entropy of the predicted next character together with the regularization penalty as the training objective (details in Appendix B).

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 394-396.
Amherst, Massachusetts, June 15-17, 2023

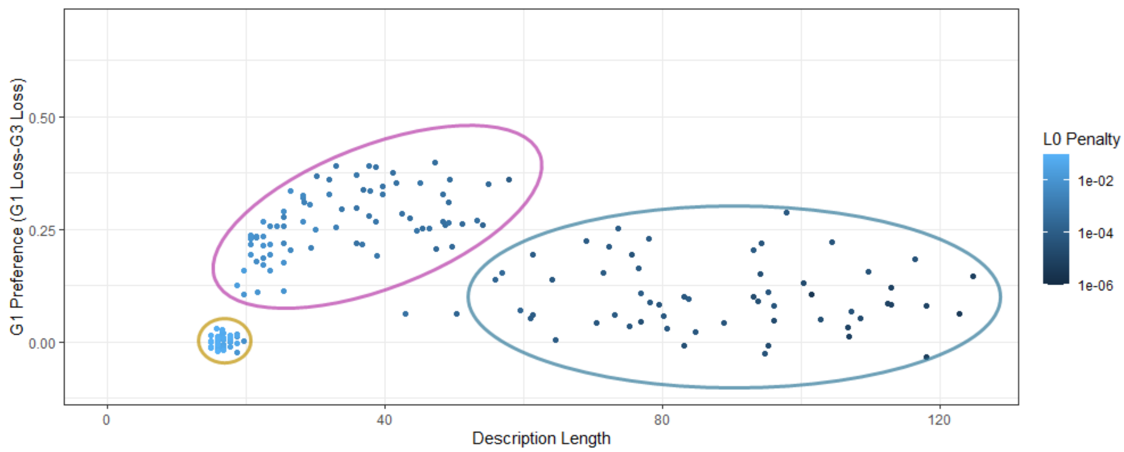

Figure 2: Plot showing relationship between complexity of LSTMs trained on an intersection of G_1 and G_3, and the performance gap between G_1 and G_3 after training (the G_1 preference). When complexity is unconstrained, performance moderately favors G_1 (blue ellipse). As complexity becomes more constrained, favorability of G_1 as a hypothesis increases (purple ellipse). Extremely tight constraints leads to a collapse in preference (gold ellipse).

After training we assess differences in performance on words drawn from G_1 and G_3 separately to assess generalization. If G_1 is preferred for models of constrained complexity, then complexity constraints may result in an inductive bias for subregular languages.

Results Our results show a bias for the subregular grammar G_1 for almost all levels of complexity. But, this preference is responsive to complexity constraints. As complexity of these LSTMs lowers there is an increase in this preference before a subsequent collapse (Fig. 2). A t-test between the purple and blue regions (defined as the range 20-40, and >40, respectively) is statistically significant ($t = -13.79, p < 2.2 \times 10^{-16}$).

What drives this change can be seen in Fig. 3. Regularizing for complexity causes a drop in the cross entropy for the subregular language after training, a pattern which is most extreme when at the 40 parameter mark. In other words, regularization leads to generalization from the intersection of the two grammars to G_1 exclusively.

Discussion Our results show that a preference for simple LSTMs can enhance subregular preferences in at least some cases. Previous work on GRUs (Prickett, 2021), also show subregular biases, but our work contributes a possible additional explanation for this bias: that this preference is downstream of a preference for solutions involving smaller subnetworks. This is consistent with the Lottery Ticket Hypothesis (Frankle and Carbin, 2019), and may function with–or be the underlying cause of–other biases, like the recency bias (Ravfogel et al., 2019).

Though this work reinforces the existence of a subregular bias in neural networks, and offers an explanation for its presence, it still leaves several questions unanswered. Is it really the subregular class that is preferred? It is possible that what appears to be a subregular bias is *only* appearance, and that the real bias has yet to be elucidated by formal language theory. Furthermore, how does this preference under regularization constraints compare with human biases? Further research is warranted to describe this bias, and how it compares with the subregular class and human phonology.

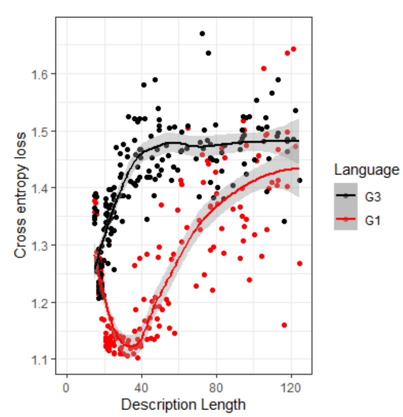

Figure 3: Relationship between description length and cross entropy loss on G_1 and G_3 for LSTMs trained on their intersection.

References

Enes Avcu and Arild Hestvik. 2020. Unlearnable phonotactics. *Glossa: a journal of general linguistics*, 5(1).

Nick Chater and Paul Vitányi. 2003. Simplicity: a unifying principle in cognitive science? *Trends in cognitive sciences*, 7(1):19–22.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*.

Peter Grünwald. 2000. Model selection based on minimum description length. *Journal of mathematical psychology*, 44(1):133–152.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. *Phonological typology, phonetics and phonology*, pages 126–195.

Jeffrey Heinz and William Idsardi. 2013. What complexity differences reveal about domains in language. *Topics in cognitive science*, 5(1):111–131.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Regine Lai. 2015. Learnable vs. unlearnable harmony patterns. *Linguistic Inquiry*, 46(3):425–451.

Christos Louizos, Max Welling, and Diederik P Kingma. 2017. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*.

Kevin McMullin and Gunnar Ólafur Hansson. 2019. Inductive learning of locality relations in segmental phonology. *Laboratory Phonology*, 10(1).

Brandon Prickett. 2021. Modelling a subregular bias in phonological learning with recurrent neural networks. *Journal of Language Modelling*, 9.

Shauli Ravfogel, Yoav Goldberg, and Tal Linzen. 2019. Studying the inductive biases of rnns with synthetic variations of natural languages. In *Proceedings of NAACL-HLT*, pages 3532–3542.

Jon Rawski, Aniello De Santo, and Jeffrey Heinz. 2017. Reconciling minimum description length with grammar-independent complexity measures.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision rnns for language recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 740–745.

A Architecture

Each LSTM is composed of an embedding layer, a single layer LSTM, a linear layer, and decoder layer. The embedding layer has 3 dimensions, and all other layers have 5. The LSTM uses the $tanh$ activation function.

B The training objective

Our training objective is a form of MDL, in particular, a two-part code formed from the sum of the cross entropy and the expected value for number of non-zero parameters. It can be idealized as:

$$J = \frac{1}{N}\sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(x_i) + \beta \sum_{\theta \in \boldsymbol{\theta}} q_{\boldsymbol{\phi}}(\theta \neq 0)$$

Where $p_{\boldsymbol{\theta}}$ is our LSTM, parameterized by parameter vector $\boldsymbol{\theta}$ and $q_{\boldsymbol{\phi}}$ probability of a parameter being masked (see Louizos et al. (2017) for details). The constant β, our regularization parameter, allows us to control for relative preference in LSTM size.